ABSTRACT

WEATHERFORD, SHAWN A. Student Use of Physics to Make Sense of Incomplete but Functional VPython Programs in a Lab Setting. (Under the direction of Ruth Chabay.)

Computational activities in Matter & Interactions, an introductory calculus-based physics course, have the instructional goal of providing students with the experience of applying the same set of a small number of fundamental principles to model a wide range of physical systems. However there are significant instructional challenges for students to build computer programs under limited time constraints, especially for students who are unfamiliar with programming languages and concepts. Prior attempts at designing effective computational activities were successful at having students ultimately build working VPython programs under the tutelage of experienced teaching assistants in a studio lab setting. A pilot study revealed that students who completed these computational activities had significant difficultly repeating the exact same tasks and further, had difficulty predicting the animation that would be produced by the example program after interpreting the program code.

This study explores the interpretation and prediction tasks as part of an instructional sequence where students are asked to read and comprehend a functional, but incomplete program. Rather than asking students to begin their computational tasks with modifying program code, we explicitly ask students to interpret an existing program that is missing key lines of code. The missing lines of code correspond to the algebraic form of fundamental physics principles or the calculation of forces which would exist between analogous physical objects in the natural world. Students are then asked to draw a prediction of what they would see in the simulation produced by the VPython program and ultimately run the program to evaluate the students' prediction. This study specifically looks at how the participants use physics while interpreting the program code and creating a whiteboard prediction. This study also examines how students evaluate their understanding of the program and modification goals at the beginning of the modification task.

While working in groups over the course of a semester, study participants were recorded while they completed three activities using these incomplete programs. Analysis of the video data showed that study participants had little difficulty interpreting physics quantities, generating a prediction, or determining how to modify the incomplete program. Participants did not base their prediction solely from the information from the incomplete program. When participants tried to predict the motion of the objects in the simulation, many turned to their knowledge of how the system would evolve if it represented an analogous real-world physical system. For example, participants attributed the real-world behavior of springs to helix objects even though the program did not include calculations for the spring to exert a force when stretched. Participants rarely interpreted lines of code in the computational loop during the first computational activity, but this changed during latter computational activities with most participants using their physics knowledge to interpret the computational loop.

Computational activities in the Matter & Interactions curriculum were revised in light of these findings to include an instructional sequence of tasks to build a comprehension of the example program. The modified activities also ask students to create an additional whiteboard prediction for the time-evolution of the real-world phenomena which the example program will eventually model. This thesis shows how comprehension tasks identified by Palinscar and Brown (1984) as effective in improving reading comprehension are also effective in helping students apply their physics knowledge to interpret a computer program which attempts to model a real-world phenomena and identify errors in their understanding of the use, or omission, of fundamental physics principles in a computational model.

© Copyright 2011 by Shawn A. Weatherford

All Rights Reserved

Student Use of Physics to Make Sense of Incomplete but Functional VPython Programs in a Lab Setting

by
Shawn A. Weatherford

A dissertation submitted to the Graduate Faculty of
North Carolina State University
in partial fulfillment of the
requirements for the Degree of
Doctor of Philosophy

Physics

Raleigh, North Carolina

2011

APPROVED BY:

Robert Beichner

David McConnell

John Risley

Ruth Chabay
Chair of Advisory Committee

DEDICATION

This body of work is dedicated in loving memory and honor of

Christopher Price Weatherford

brother, friend, and awesome *U.S.S Enterprise NCC-1701* captain

(1981 - 2006)

BIOGRAPHY

Shawn Weatherford was born in Norfolk, Virginia to Randy and Pam Weatherford on September 18, 1979. After moving to Corapeake, North Carolina, a small rural farm town at age 4, Shawn completed his entire grade school education in the Gates County Public School system. After graduating with a high school diploma, Shawn enrolled into a bachelor's degree program at Elon College in Elon College, North Carolina to focus study in secondary science education and physics. As a North Carolina Teaching Fellow at Elon, Shawn was privy to internship experiences in Washington, D.C. and a semester in London, England. After adapting to the culture shock in public policy towards secondary education in England, Shawn returned to the States and joined the North Carolina Section of the Association of Physics Teachers. The following semester, he began a student teaching semester at Walter M. Williams High School in Burlington, North Carolina. Shawn was offered the physics teaching job at Williams upon graduating from Elon in 2001.

Shawn found tremendous challenge and reward teaching high school physics and physical science, but quickly found that he lacked the training to develop and evaluate research-based instructional materials which seek to improve student learning. Shawn took a distance learning course from Bruce Sherwood after learning of the Matter & Interactions curriculum at a NCS-AAPT meeting. Although he was initially impressed by the computational component of the course, Shawn quickly realized that the course was more than simply an integration of computational physics in the introductory physics sequence. Shawn was again excited about learning new physics. Although Shawn was able to substantially grow the physics program at Williams two-fold, including tripling the number of females choosing to take physics, he was not satisfied by incremental improvements with student performance in learning physics concepts. Shawn enrolled into a graduate program in physics at North Carolina State University. It was important to Shawn to enroll in a physics program and not an education program. Although effective pedagogy is important, Shawn's first priority was learning physics and he thoroughly enjoyed revisiting old and forgotten skills as well as learning new physics that was previously beyond his reach. After joining the Physics Education Research and Development group, he turned his research focus on the component of physics instruction that refreshed his interest in physics: computation. Under the direction of Ruth Chabay and the support of the faculty in the PERD group, Shawn picked a research focus that, to this day, remains as exciting as it did at the beginning of his training.

After successfully passing his preliminary oral exam, Shawn was recruited to bring the Matter & Interactions curriculum back to his alma mater, now called Elon University. Hired as a full-time instructor in physics, Shawn spent the last year of his graduate program teaching the introductory physics with calculus sequence to majors and engineers while finishing up his dissertation work. Shawn had a unique opportunity and pleasure of witnessing the fruits of his research deployed in the course that initially exposed him to physics over a decade ago.

ACKNOWLEDGEMENTS

Although there's only one author on the title page, there's a massive supporting cast behind the work to whom I owe a tremendous amount of gratitude that I will attempt to acknowledge.

First, I should acknowledge the Physics Education Research and Development group for their willingness to have thoughtful conversations about my research and and endless amount of support. Ruth Chabay has been such an encouraging mentor and colleague and has provided endless support for this research as well as my desire to serve as a contributing member of AAPT. Bruce Sherwood, the person who introduced me to Matter & Interactions many years ago through a NCS-AAPT presentation and later my instructor for the course, was always excited about showcasing new features of Visual or a new demo to measure the effects of surface charge on a thin strip of aluminum foil. I appreciate Bob Beichner for his support to attend the Gordon Research Conference on computational physics in the physics curriculum which has had a substantial influence on my research and greatly increased my network of researcher contacts. I also owe Dr. Beichner thanks for exposing me to SCALE-UP and allowing me to serve as his TA. Thanks to John Risley and the team at WebAssign for the experience coding questions during two summers. I should note that there's no conflict of interest: I ended my employment at WebAssign two years before I asked Dr. Risley to serve on my committee. And thanks to David McConnell for his willingness to provide time to serve on my committee and offer substantive comments on the research as well as bring his expertise in the affective domain. The faculty and staff in the NCSU physics department has been very supportive of this research and I thank specifically David Brown, Karen Daniels, David Haase, Jenny Allen for their assistance and guidance during my graduate program.

I also acknowledge the contributions from each student member, past and present, of the PERD group. Matt Kohlmyer was very supportive of the efforts of this research to help develop reformed instructional materials for the M&I course and was very flexible as the course coordinator to provide support and critiques on the new course materials. Evan Richards and

Jon Gaffney both served as great mentors through their willingness to involve my participation in their own research projects. Mary Bridget Kustusch has always been there to listen and talk about life's unexpected traumatic events. Jeff Polak, Brandon Lunk, and Meghan West have all been willing to help validate the coding scheme and serve as independent raters for reliability measurements.

I offer a great amount of thanks to the officers and members of the North Carolina Section of the American Association of Physics Teachers. Over the past 13 years, I have enjoyed my membership to the organization and the many section meetings I've attended from across the state from Pembroke to Boone. Perhaps the most difficult part of moving away from North Carolina is moving out of driving distance from these meetings. The section has time after time supported my own scholarship and professional development through lunch conversations, presentations, and workshops. The opportunity to present my research and ideas over the years has provided public speaking practice that has had an ample effect on my performance. The section has provided leadership opportunities and provided a great amount of trust to allow a graduate student to plan and run a joint section meeting with the Southeast Section of the American Physical Society. Mario Belloni, Wolfgang Christian, Tony Crider, Jose D'Aruda, Joe Heafner, John Hubisz, Martin Kamela, Tatlock Lauten, Don Smith, Gordon Sheppard, Aaron Titus, Larry Ward have all offered wonderful conversation on the teaching and learning of physics.

Finally, I'd like to offer thanks to my support network outside of academics. Peter Dittmar, my partner and the best thing that's come out of my graduate studies, has always had time to listen and offer optimism during difficult times when I had none. My parents who have been extremely supportive of my decision to interrupt my professional career to go back to school and who always offer a warm and inviting retreat from the city. Tanner, the puppy who is always excited to see papa come home even if he's wearing a frown has always reminded me of the joys of life and the fleeting moments we have in it.

And last but not least, my brother, whose time on this Earth was too short, who always

looked up to his older brother. His encouragement pushed me to strive farther and work on setting a better example. Chris was quite amused by calling me Dr. Weatherford before his passing and he's still pushing me to this day to make every moment of this life count.

To everyone listed here, and those I erroneously forgot: Thank you.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Computation offers physics education the possibility of examining complex, unidealized systems which are out of reach of traditional analytical methods. By applying an algorithm consisting of the iterative application of physics principles, students can explore and predict the time-evolution behavior of two- and three-body systems interacting through gravitational or electrostatic forces and the dissipative effects of friction on falling or sliding objects such as springs. Both of these systems are commonly discussed in varying levels of detail in the introductory physics course, and a computational approach offers an open-ended time-evolution analysis of these more complex systems. Recognizing the usefulness of computation to offload the heavy iterative workload of an algorithmic approach to open-ended solutions, introductory and intermediate physics courses are integrating computation into their curriculums. There are two approaches to using computation in the introductory calculus-based physics curriculum: 1) provide students with simulations that model complex systems and the freedom to explore the effects of varying initial conditions on a system of interest, or 2) provide students with a programming environment in which they may build or alter a computational model to examine a complex system.

Matter & Interactions, a curriculum designed for the introductory calculus-based two-semester physics sequence relies heavily on computation to 1) visualize abstract physics quantities and 2) provide a medium through which the open-ended application of physics principles are used to evolve a system's dynamics. As such, students are asked to complete activities where they are tasked with building computational models. The hope is that by building computational models which involve the application of fundamental physics principles, students gain a greater conceptual understanding of how a few fundamental principles can apply to a wide variety of systems.(Chabay & Sherwood, 2008) Prior research provides some evidence in support of this endeavor in pictorial programming environments such as STELLA (e.g. Costanza, 1987) or spatial programming environments such as Boxer (e.g. diSessa & Abelson, 1986). This research is examined in greater detail in Chapter 2 of this dissertation.

It is an open question as to how students use their knowledge to interpret and apply physics principles across different programming environments. In Matter & Interactions, students use the VPython programming environment to build computational models. VPython is a 3D graphics module that provides Python programmers with access to a navigable 3D display and create animations with ease. VPython provides the option of importing graphics libraries to draw 3D objects within a virtual 3D window, or visual scene. Students have the freedom to control the visual attributes of these 3D objects by changing their default attributes in a text editing environment. Combined with the iterative application of physics principles, students can create time-evolving models of the positions of these 3D objects to produce simulations corresponding to analogous real-world behavior of interacting systems. Computational activities were developed to guide students through building a computational model and have been used in studio-style lab settings where students, working in collaborative groups of three, complete scaffolded tasks to achieve a valid and working program. These computational activities are based on research investigating the difficulties students have in creating a computational model from scratch, providing extra information to aid in the completion of the programming task when students most often need help (M.Kohlmyer, 2005).

The instructional model for designing computational activities is nearly identical to the way in which computational physicists build models of the natural world, as summarized by

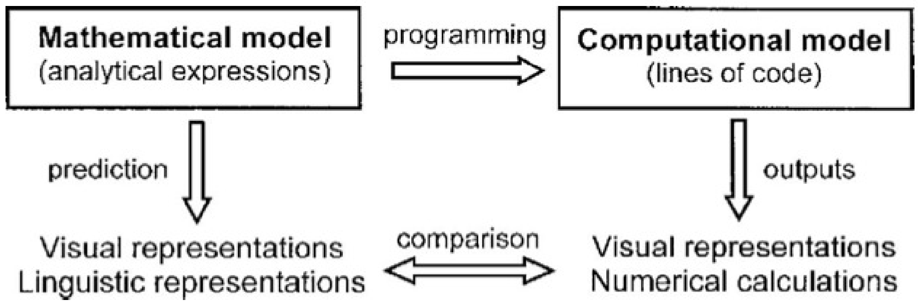


Figure 1.1: Buffler's model for computational activities in physics (Buffler et al, 2008, p.432).

Buffler et al.(2008) in Figure 1.1. Buffler shows the process to produce a computational model of the mathematical model which eventually runs to generate a visualization of the numerical output in the form of an animation. First, you start with a mathematical model for the system. From the mathematical model, one translates initial state variables and system of equations into programming syntax which iteratively calculates and updates the values of these state variables to produce a numerical output. With VPython, 3D objects are used to visualize these quantities to produce an animation of the numerical output at runtime. In Buffler's computational activities, he asked his students to predict features of the dynamics of the system based on the mathematical model. Then, after the students translated the mathematical model into a computational model, he asked students to compare the prediction of the mathematical model to the visual output.

Given the limited programming experience of introductory calculus-based physics students, the task of translating a mathematical model into a computational model is incredibly challenging. Buffler et al.(2008) reported that very few students (20 of 51) were able to generate both the mathematical model and computational model during the reported activity. Kohlmyer's (2005) dissertation research investigating the difficulties students have programming in VPython has led to revised computational activities to reduce this burden. However, students interpret the goal of these activities as one where their task is to create a program that runs to produce a visual scene and an animation without receiving any error messages from the compiler. The intended goal of the computational activity was to provide students with experience generating a computational model based on the iterative use of fundamental physics principles and explore the model by changing initial values and observing the effects of these changes to the simulation in the navigable 3D visual output. That is, students focus on the programming task and not how the model generates a prediction for the evolution of a system of interest.

A pilot study was conducted to investigate further the difficulties students have completing tasks which focus on building the computational model in VPython. Students were recruited who completed the first semester of the Matter & Interactions course and therefore the computational activities included in the first semester. One major finding of the study, which is discussed further in Chapter 3, is that students who were capable of interpreting the individual lines of code in an example program had a difficult time thinking through how the program's execution could display the events which unfolded in the visual scene. Student predictions of the visual output were mostly correct, including the effects of interactions between 3D objects and how the object's motion would evolve in the simulation. However, it's not clear that the predictions were a reflection of an interpretation of the program code or, alternatively, an expectation of system behavior unrelated to thinking about how the program code evolves the

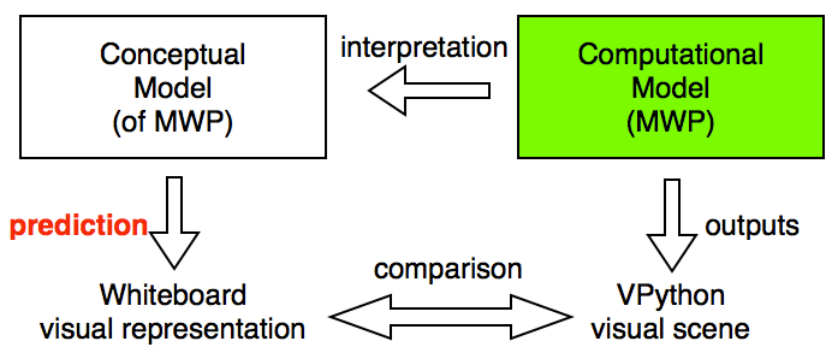


Figure 1.2: New model for computational activities in physics beginning with a comprehension task.

dynamics of the system. Investigating this question further provides an opportunity to add to the physics education literature on how students create an informed comprehension of system dynamics using evidence contained in an accessible computational model juxtaposed with their own conceptual understanding of physics quantities and principles. This led to the generation of a new type of computational activity which initially focuses on comprehension of programming code rather than generating programming code. As a result, the instructional model for the computational activities is a modified version of Buffler's Model (2008), and is presented in Figure 1.2. Before asking students to create program code, the activity begins with students interpreting a program that is incomplete, but still functional. The program is incomplete with regards to the computational model not including all the necessary physics calculations to generate an accurate prediction of the system. The program is functional in the sense that the program will produce a navigable 3D visual scene and an animation of the numerical solution generated by the computational model in its incomplete form. While students interpret the program, they are asked to generate a drawing to represent their prediction of what they might see happen in the visual scene or animation before they are allowed to run the program for the first time. Chapter 4 elaborates further on this new activity as well as the motivation behind asking students to interpret a program that isn't finished. This dissertation explores the primary question underlying the interpretation of this new type of program:

> **How do students use their physics knowledge to make sense of incomplete, but functioning VPython programs?**

The analysis consists of investigating how participants in a research study use physics to build a conceptual model of the incomplete, but functional program while working together in collaborative groups. The analysis will also investigate how participants evaluate their understanding of the program code based on comparisons between the output from the computational model as displayed in the visual scene with the visual representation generated from the prediction task. And finally, the analysis will investigate the goals of each collaborative group as they begin the modification task to complete the computational model.

Chapter 4 defines the characteristics of an incomplete, but functional program and provides three example activities, each incorporating a different program. Recordings of participants in a research laboratory environment are used to serve as the data to answer the research question. Chapter 4 continues to report the details the experimental study and its design goals as well as the techniques used in analysis of the data generated during the study. Chapter 5 presents the results from the analysis of the data, comparing student groups to one another as they work on the instructional tasks. Chapter 6 concludes the study with a discussion of what was learned, what new knowledge is now part of the research literature, further implications for instructional design of computational activities, and the avenues for future research to examine how to use computation to enhance the conceptual understanding of physics.

But first, this chapter will detail the terminology used in this dissertation when referring to the fundamental physics principles or programming concepts.

1.1 Terminology

Matter & Interactions emphasizes the use of a small set of fundamental principles which are valid for all systems. The Momentum Principle is particularly useful in building computational models due to the vector nature of the physics quantities which, combined with the Position Update formula, provides enough information to evolve the spatial attributes of 3D objects in a 3D visual scene. Eqn (1.1) shows the algebraic representation of the Momentum Principle in its update form. The update form of the Momentum Principle is useful as an interpretation of how the new value of the momentum of the system equals the previous value at the beginning of an interval plus a quantity that quantifies the nature of the interaction between it and its surroundings.

$$\vec{p}_{\text{final}} = \vec{p}_{\text{initial}} + \vec{F}_{\text{net}} \Delta t \tag{1.1}$$

$$\texttt{pcraft} = \texttt{pcraft} + \texttt{Fnet*deltat} \tag{1.2}$$

The computational analog, or Eqn (1.2) shows how this algebraic equation might appear in a computational model. The interpretation of Eqn (1.2) is somewhat different from its algebraic interpretation.(Sherin, 1996) Each of these variable names represents stored values that were initially given some value. If these variables represent vector quantities for the momentum and net force on a spacecraft, and the `deltat` represents the interval of time during which the force is approximated as a constant value, then this line of code represents the algebraic form of the Momentum Principle with the function of updating a value store in system memory which represents the value for the momentum of the spacecraft.

The position update formula computes the new position of a system of interest due to the average velocity over a time interval. Eqn (1.3) shows the algebraic representation of the Position Update formula. The computational analog to this formula is shown in Eqn (1.4) and updates the attributes of the 3D object named craft based on the current value for the momentum of the craft, the mass of the craft, and the interval over which the momentum is approximated as a constant value. Using the updated value of the momentum instead of the average of the old and updated values produces an algorithm which is stable.(Timberlake & Hasbun, 2008)

$$\vec{r}_{\text{final}} = \vec{r}_{\text{initial}} + \frac{\vec{p}_{\text{avg}}}{m}\Delta t \tag{1.3}$$

$$\texttt{craft.pos} = \texttt{craft.pos} + \texttt{pcraft/mcraft*deltat} \tag{1.4}$$

The second fundamental principle, the Energy Principle shown in Eqn (1.5) describes changes in energy within a system due to inputs and outputs from the system's surroundings. The Energy Principle is a scalar formula and therefore it is not productive for introductory physics students to use as a method for constraining the dynamical quantities of 3D objects defined on a 3D space.

$$E_{\text{sys,final}} = E_{\text{sys,initial}} + \Delta E_{\text{surroundings}} \tag{1.5}$$

The computational loop, or iterative loop, refers to a section of program where lines of code are calculated in sequence until a specified condition is met. The computational loop is the workhorse for the program and controls the animation viewed in the visual output. The algorithm refers to the specific sequence of calculations within the computational loop. In Matter & Interactions, students are taught the Euler-Cromer algorithm for updating the momentum and position values assigned to 3D objects for two reasons: 1) the algorithm obeys Liouville's theorem when applied to conservative systems (Timberlake & Hasbun, 2008) and produces numerical results which conserve phase space and therefore conserve energy, and 2) the sequence of calculations closely resembles the analytical solution to many of the quantitative homework problems students work on during the course.

Chapter 2

Literature Review

2.1 Introduction

This literature review is segmented into three sections, each discussing a relevant sector of literature relating to this research project:

2.2 Computation in Introductory Physics

2.3 Program Comprehension & Reading Comprehension

2.4 Sensemaking in Physics Education Research

Section 2.2 contains the bulk of this review, summarizing previous research investigating the use of computation as an activity to learn physics. This section explores relevant prior research, defined as research which involves student analysis of computational models. It will not include research where graphical or numerical representations are displayed to students without access to the computational model creating these representations. The section is further divided into subsections exploring: computational thinking/reasoning, computational environments in physics, and research of multiple representations.

Section 2.3 analyzes the research by cognitive psychologists to develop a model of the mind using a computational metaphor by studying how programmers complete programming tasks. For this dissertation, the comprehension of computer programs is of particular interest. The section analyzes the program comprehension models for similarities to the primary functions of reading comprehension, as identified by Palinscar & Brown (1984).

Section 2.4 examines the physics education research literature on sensemaking. Recently, researchers have focused their interest on how students work towards identifying goals and complete tasks during an instructional activity. This section reports how these researchers identify sensemaking and distinguish sensemaking from other forms of student activity, and how this identification is a useful measure of student engagement.

2.2 Computation in Introductory Physics

This section begins by exploring a fundamental shift in instructional goals by computer scientists.

2.2.1 Computational Thinking

Abelson & Sussman (1984), in the preface of their procedural textbook pointed to the computer revolution as a "revolution in the way we think and express what we think." They referred to this metacognitive analysis as a procedural epistemology, "the study of the structure of knowledge from an imperative point of view." Meaning, just as mathematics provides a framework for "what is", computation provides a framework for "how to" (Abelson & Sussman, 1984).

Procedural epistemology is more commonly known as algorithmic thinking, which involves using computers as machines to run algorithms and programming languages as tools for expressing these algorithms to computers and programmers (Syslo & Kwiatkowska, 2008). Specifically, algorithmic thinking involves the ability to analyze problems from an algorithmic point of view, apply existing algorithms for their solution, develop new algorithms for unique problem situations, implement the solution and finally run and test the solution on a computer (Syslo & Kwiatkowska, 2008).

Jeanette Wing's (2006) essay argues that computational thinking is a fundamental skill, useful in applications outside of computer science. With the introduction of object-oriented languages, Wing elevates the role of creating abstractions and layers of abstractions with the knowledge of appropriate algorithms to deploy as a way to think like computer scientists. Wing argues that computational thinking in analytical sciences extends the domain of creative problem solving. By providing opportunities to ask questions otherwise unimaginable, computational methods extends the reach of analytical sciences to produce simulations which evolve analytical models.

Qualities of computational thinking includes: 1) thinking at multiple levels of abstraction, 2) having the freedom to construct virtual worlds and engineer systems beyond the physical world, 3) using imagination to tackle inconceivable problems through reduction and task analysis, 4) using computational concepts to solve everyday problems and manage daily lives (Wing, 2006). Note that computational thinking is more than knowledge about different types of algorithms. Computational thinking includes initiating a range of mental tools and concepts (abstraction, induction, heuristics) applied to a specific problem to develop a process to efficiently and approximately makes the problem tractable in order to compute a solution.

2.2.2 21st Century Curriculum Re-Design

Physicists increasingly rely on computation for scientific purposes to visualize physical phenomena and simulate the time-evolution through representations of physical systems. The increase in the use of computation for physics has led to revisions of the physics curriculum to include computational activities to prepare future physicists for these tasks. Over the past 40 years, the education research literature has explored the roles computation can play in a formal education setting to display abstract concepts and provide the opportunity for students to build computational models. Physics education researchers have explored the use of computation as a tool to enhance qualitative conceptual learning through interactive simulations (e.g. Perkins et al., 2006; Linn, Lee, Tinker, Husic, & Chiu, 2006; deJong, 2006; Christian & Belloni, 2000; Singh, Belloni, & Christian, 2006). Others use programming languages as a representational language in introductory physics courses to enhance the conceptual learning goals of the curriculum (e.g. E. Redish & Wilson, 1993; Chabay & Sherwood, 2004, 2008; Buffler et al., 2008). Finally, others explicitly teach computational methods in parallel with introductory and upper level physics courses for majors (e.g. Landau, 2006; McIntyre, Tate, & Manogue, 2008; Roos, 2006; Taylor & King, 2006). Since the focus of this dissertation is on how students use physics to understand computational models, this section will restrict its breadth to how researchers have studied and used computer modeling environments in educational settings.

2.2.3 Computer-Based Modeling Environments

This section explores the different types of microworld environments researchers have used to explore how students apply physics to such environments.

M.U.P.P.E.T.

The physics department at the University of Maryland in 1985 implemented a curriculum that integrated computational physics with the introductory calculus-based physics sequence for majors (E. Redish & Wilson, 1993). The M.U.P.P.E.T. environment provided a library of input/output devices to be used by programmers. This library included data input screens, graph windows, menus, and parsing procedures. Figure 2.2 is an example of the input/output screens which are actuated by the program shown in Figure 2.1. Students study example programs to learn or become familiar with the Pascal programming language (Turbo or Think's

Pascal on Macintoshes, or Turbo Pascal on PC). The curriculum augmented the traditional physics sequence to include more realistic problems and situations beyond the limitations of students' formal mathematical training. The focus of the project was to expose students to a class of physics problems designed to analyze real systems using a discrete form of fundamental principles. Through discrete representations of physical laws combined with the added power of computation, the introduction of physically important ideas may occur at an earlier stage than otherwise possible. An early introduction to these ideas promotes a hierarchical ordering of material to help students decide on "the relative strengths of the various concepts, principles, and techniques presented in the curriculum." (MacDonald, Redish, & Wilson, 1988)

The M.U.P.P.E.T. project followed three guiding principles of design: 1) rethink the curriculum entirely assuming the availability of the computer (what can we teach that we couldn't teach before?), 2) the computer should not replace the teacher, textbook, or the laboratory, and 3) the student should run the computer, rather than the other way around. (MacDonald et al., 1988)

Redish(1993) reported on the capabilities of the M.U.P.P.E.T. environment to provide opportunities for students to complete research projects of interest, somewhat connected to the topics contained in the introductory physics course. The goal of these projects was to provide an open-ended investigation fashioning itself on the methods and techniques of doing science. Prior to M.U.P.P.E.T., Redish reported that student work lacked the desired characteristics of scientific research. Using M.U.P.P.E.T., two-thirds of the students produced projects Redish considers "valuable and interesting."

Johnson & McPhedran completed a study involving three research trials investigating the effects of integrating M.U.P.P.E.T. with the second and third year physics curriculum at the University of Sydney. Specifically, Johnson & McPhedran (1993) were interested in the answers to three research questions: 1) Can the subject material be learned in an computational environment as well as from lecture, 2)are students who use M.U.P.P.E.T. able to solve a wider

```
PROGRAM Projectile1D;        {PROJ1D.PAS}
{**************************************}
{      Program to calculate motion of    }
{      a particle in 1D with gravity     }
{      and air resistance.               }
{**************************************}

USES Crt, Dos, Graph, Printer, MUPPET;

CONST
   NumData    = 200; {Number points to plot}
   g          = 9.8; {m/sec/sec}

VAR
   t, x: DataVector;  { time, position    }
   v, a: DataVector;  { velocity, accel   }
   x0, v0 : Real;     { initial conds.    }
   m      : Real;     { mass              }
   b      : Real;     { air resis. coeff. }
   dt     : Real;     { time step         }
   i      : Integer;  { loop variable     }
   IC     : Screen;   { data screen       }
   act    : Char;     { control char.     }

  { The types 'DataVector' and 'Screen' }
  { are defined in the unit MUPPET      }


 {*-----        Physics Procedure        -----*}

 FUNCTION Force(x, v, t: Real) : Real;
 BEGIN
    Force := -m*g - b*v*abs(v)
 END;

 {*-----        MathematicsProcedure     -----*}

 PROCEDURE StepEuler(xIn, vIn, tIn, aIn,tStep:
        Real; VAR  xOut,vOut,tOut,aOut: Real);
 BEGIN
    {STUDENT TO FILL IN THIS PROCEDURE}
 END;


 {*-----     Data Screen Procedure       -----*}

 PROCEDURE MakeDataScreen;
 BEGIN
 DefineInputPort(0,0.45,0,0.9);
 _A[01]:=''*M.U.P.P.E.T.*                      ';
 _A[02]:=''*University of Maryland*            ';
 _A[03]:='                                     ';
 _A[04]:=''*PROJECTILE PROGRAM: 1D*            ';
 _A[05]:=''*F = -mg - bv*abs(v)*               ';
 _A[06]:='                                     ';
 _A[07]:=''*PARAMETERS*                        ';
 _A[08]:=' *Mass          m = * 1+++++ *kg*    ';
 _A[09]:='                                     ';
 _A[10]:=' *Air Resistance*                    ';
 _A[11]:=' *Coefficient, b = * 0+++++ *kg/m*   ';
 _A[12]:='                                     ';
 _A[13]:=' *Time step,  dt = * 0.05++ *sec*    ';
 _A[14]:='                                     ';
 _A[15]:=''*INITIAL CONDITIONS*                ';
 _A[16]:=' *Position:   x0 = * 0++++ *m*       ';
 _A[17]:=' *Velocity:   v0 = * 40+++ *m/sec*   ';
 LoadScreen(IC,17);
 END;

PROCEDURE GetScreenData
          (VAR m,b,x0,v0,dt: Real);
BEGIN
  ClearMUPPETport;
  OpenInputPort;
  PutDate;        {puts date on data screen}
  Message('Press <ENTER> to plot,
                 <ESC> to quit');
  Accept(IC);                {display screen}
  m    := Valu(IC,1);
             {accept 1st entry on IC to m}
  b    := valu(IC,2)
           {accepts 2nd entry on IC to b}
  dt   := valu(IC,3);
  x0   := valu(IC,4);              {etc...}
  v0   := valu(IC,5);
END;

{*-----       Graphics Procedures      -----*}

PROCEDURE GraphSetUp;
BEGIN
  GraphBackColor:=DarkGray;
  DefineViewport(1, 0.55,1, 0.5,0.9);
  DefineViewport(2, 0.55,1, 0.05,0.45);
  DefineScale(1, 0,10,-75.0,120);
  DefineScale(2, 0,10, -75.0,75);
END;

PROCEDURE PlotIt(viewport, color: Integer;
       x,y: DataVector;  namelabel: BigStr);
VAR
   xmin,xmax,ymin,ymax:real;
BEGIN
   Setcolor(color);
   SelectScale(viewport);
   OpenViewport(viewport);
   Axis(0,0,1,20);
   PlotData(x,y,NumData);
   Putlabel(Inside,namelabel);
END;

{*-----         Main program           -----*}

BEGIN
MUPPETinit;
MakeDataScreen;
GraphSetUp;
REPEAT
  GetScreenData(m,b,x0,v0,dt);
  IF EscapedFrom(IC) THEN EXIT;
  t[1] := 0;          {initialize first step}
  x[1] := x0;
  v[1] := v0;
  a[1] := -g - b*v0*abs(v0)/m;

  FOR i := 2 to NumData DO   {solve the eqn}
    StepEuler(x[i-1],v[i-1],t[i-1],a[i-1],dt,
              x[i],v[i],t[i],a[i]);

  Message('Press <ENTER> for new data,
                 <ESC> to quit');
  PlotIt(1, LightGreen, t, x, 'X vs T');
  PlotIt(2, LightRed,   t, v, 'V vs T');

  act := ReadKey;
UNTIL ord(act) = 27;
MUPPETdone;
END.
```

Figure 2.1: Source code for Pascal program modeling projectile motion(Redish & Wilson, 1993)

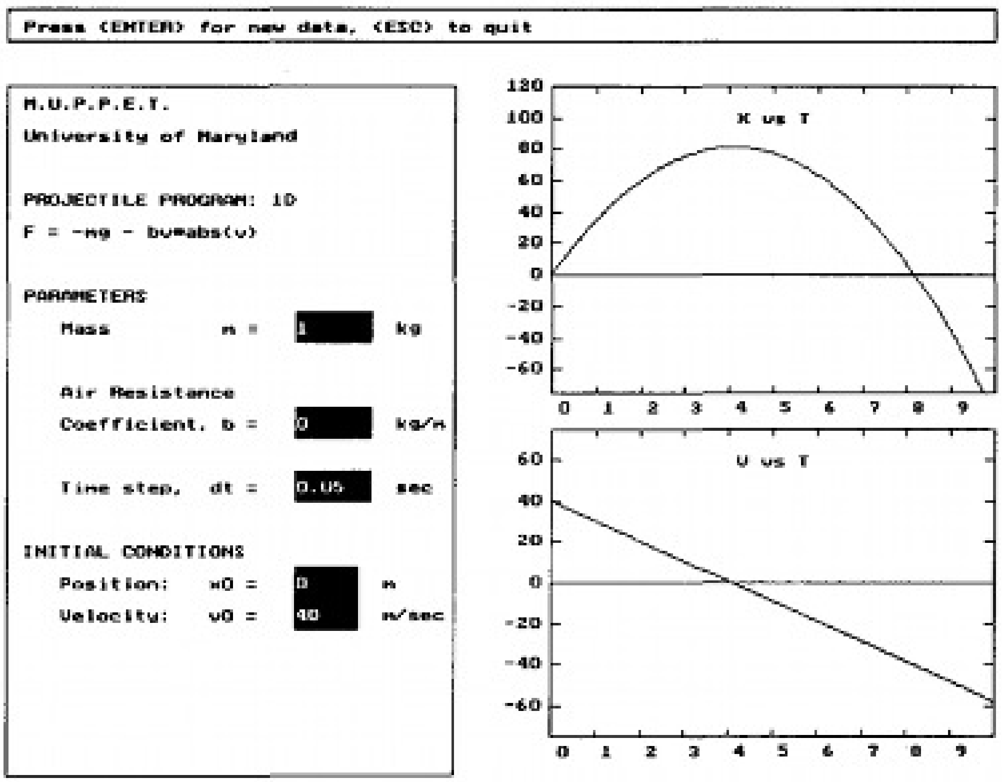


Figure 2.2: Input/Output screens from projectile motion program (Redish & Wilson, 1993)

range of physics problems, and 3)will the computational course shift attitudes towards physics. The computational activities took the form of additional work to the normal curriculum on quantum mechanics for research trials 1 and 2. After the first and second trials, the authors concluded 1) "students do not need to be able to program before handling these materials" and 2) "the students' understanding of a number of traditional subjects was significantly improved by adding computer modeling problems as shown in a comparison of the students in the test and traditional groups on traditional tests." Johnson & McPhedran attributed this performance gain to student exposure to additional physical situations beyond the limitations of analytical cases.

STELLA

STELLA (Structural Thinking Experimental Learning Laboratory with Animation)[1] provides programmers a method to represent state variables and their relationships as a collection of iconic structures in a concept-map-like environment. STELLA was designed to minimize the demands on any programming and mathematical skills. (Costanza, 1987) Figure 2.3 provides an example of how this representation describes the quantitative relationships between acceleration, velocity, and position. (Schecker, 1993) After defining initial conditions and form of graphical representation to display, STELLA extracts the differential equations from the map and evolves the system. STELLA provides a collection of visual icons for creating structural relationships, which define the dynamics of a system.

The design of STELLA allows students to focus on the qualitative relationships between quantities and how these quantities change values through time. It is argued to serve as an advantageous environment to deal with conceptual relations over mathematical formalism. Additionally, Chi (1981) shows, from her expert-novice research, that experts often tend to first use a qualitative approach when solving problems.

[1]STELLA is sold by ISEE Systems (formerly High Performance Systems). It was introduced in 1987.

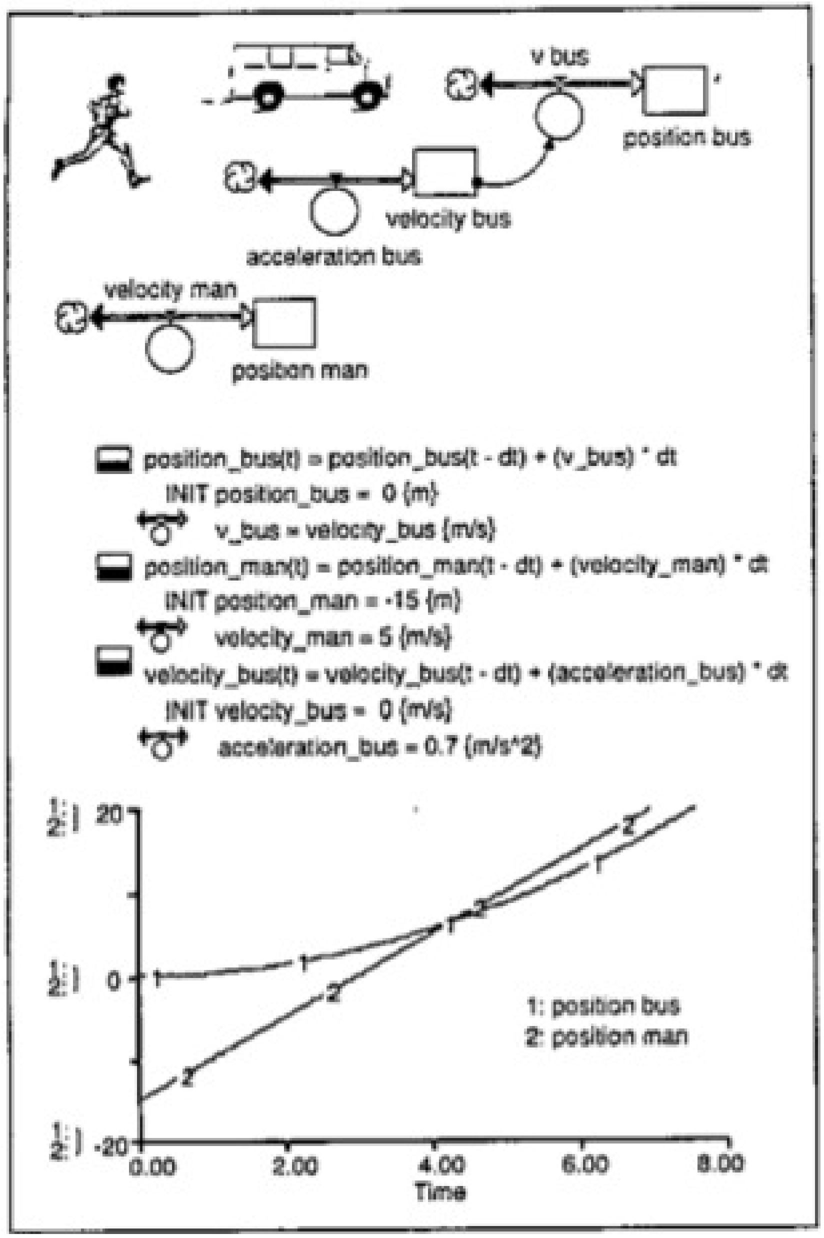


Figure 2.3: The STELLA model (top), the equations and procedures (middle), and output

One drawback of STELLA is the limited set of objects and relations that can be expressed in the environment. This limits the conceptual tools available to students in building models to the following: 1) *stocks*, or representations of things that build up over time, 2) *flows*, allows the control of stocks, 3) *converters* to generate outside influences and graphical relationships, and 4) *connectors*, to show the direction of proposed relationships. In order to use STELLA, students must conceptualize systems using these productions. Students who have alternative models of relationships may have trouble constructing a complete and coherent model with the provided tools. (Penner, 2000) Further, any evaluation of the STELLA model from graphical outputs may not reflect the student's conceptual thinking.

Schecker (1993) developed an empirical study to investigate how students use STELLA to study motion. His goal was to help students realize that the same qualitative model may be used to explore a wide range of dynamical systems. That is, students can model a different phenomena by changing the forces involved for each system, without altering the top-level process. Figure 2.4 shows the STELLA model for a meteor falling in the atmosphere. After completing an experimental semi-quantitative activity to determine the effects of velocity, air density, and cross-sectional area on friction acting on paper cones, students took 20 minutes to create a STELLA model of a parachutists. Finally, students modified this program to extend the model to an asteroid falling through the atmosphere. Schecker claims "by working on complex examples for which standard calculations fail, students realize that it is essential to have a qualitative understanding of physical structures - these can be applied universally, special equations cannot."

Hogan and Thomas (2001) compare the cognitive behaviors between five pairs of high school students using STELLA to develop a quantitative ecological model. Table 2.1 summarizes the comparisons between the productive approaches to building a quantitative model to less productive approaches. After accounting for possible differences between the groups of students, the researchers determined that the productive students maintained a systematic view of the modeling activity, which helped them view each element in the system as a whole and repetitively call upon the behavior of the system to provide feedback on their progress. Hogan and Thomas offer suggestions for designing educational activities and formative feedback during its completion, including: 1) Provide explicit and ongoing emphasis on system principles, and 2) focus on system behavior where the output is used to reform the model.

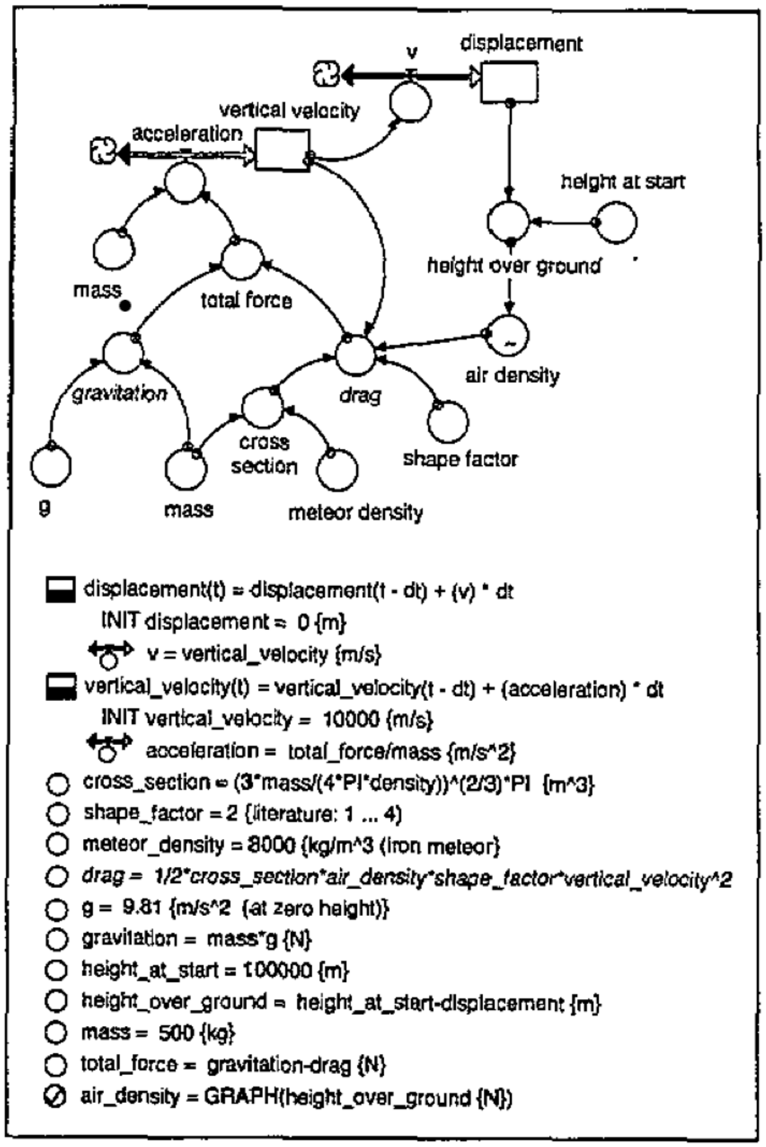


Figure 2.4: STELLA model for an asteroid moving through Earth's atmosphere. (Schecker, 1993)

Table 2.1: Table II from Hogan and Thomas(2001).

Phases of modeling	More productive approaches	Less productive approaches
Model construction	Thinking about system behavior and the burden of interpreting model output while specifying model parts and relationship	Tending to focus on representing real system ingredients without anticipating system behavior when constructing model parts and relationships
Model quantification	Taking a budget view of all inputs and outputs, and thinking about continuous relationships, when choosing values and building equations for model parts	Considering only one variable or set of equations at a time, instead of thinking about the dynamic interaction of all model quantities over time, and using constant to represent varying quantities
Model interpretation	Using output to explore the relationship between how a model functions and how it is structured and specified	Failing to investigate carefully why a model yielded particular output
Model revision	Using output to guide a range of types of revisions, including model parts, relationships, and quantities	Not engaging in extensive model revisions to learn more about how to affect and produce certain system behaviors

Boxer

Boxer[2] takes its name from its design, where all computational objects within the program are represented by boxes containing text. (diSessa & Abelson, 1986) The environment takes advantage of the computer monitor as a valid representational tool and uses it as an opportunity to create a 2D expressive environment. This environment was designed with two principles in mind - spatial metaphor and naive realism. Spatial metaphor refers to the relative location of programming objects. For example, smaller boxes can be nested into larger boxes, representing hierarchical structures. While a cursor is inside of a box, information pertaining to the box is accessed and can be treated as a separate environment. Figure 2.5 details the nested nature of boxes to create a hierarchical loop structure for repetitive calculations, the current values for the variables representing position, velocity, and acceleration, and a display of the printed position of the object after each repetition of the tick procedure. Naive realism refers to the belief by the programmer of immersion into the microworld. The monitor displays all the information pertaining to the environment. For example, creating a variable box will display the current value of the variable. Altering the value in the box will alter the variable value in the compiler. If the program alters the value in response to computing an algorithm, the text box displaying the variable's value also changes. The action of creating a variable also provides a spatial location for the state of the variable at any time.

Boxer is an offshoot of Logo, a programming language designed to challenge students' thinking about a wide range of phenomena.(Papert, 1980) The design of Logo mimics how people conversationally describe movement and thinking about the world. By using commands such as *forward* and *backward*, Papert hoped students would use their prior knowledge about movement to create a program to control the actions of an icon called a "turtle." (1980)

Although Logo was developed explicitly for educational purposes in the late 1960s, it pre-

[2]Not to be confused with the braille software to print diagrams and text by the Science Access Project at Oregon State University or the Windows text editing software available since 1991.

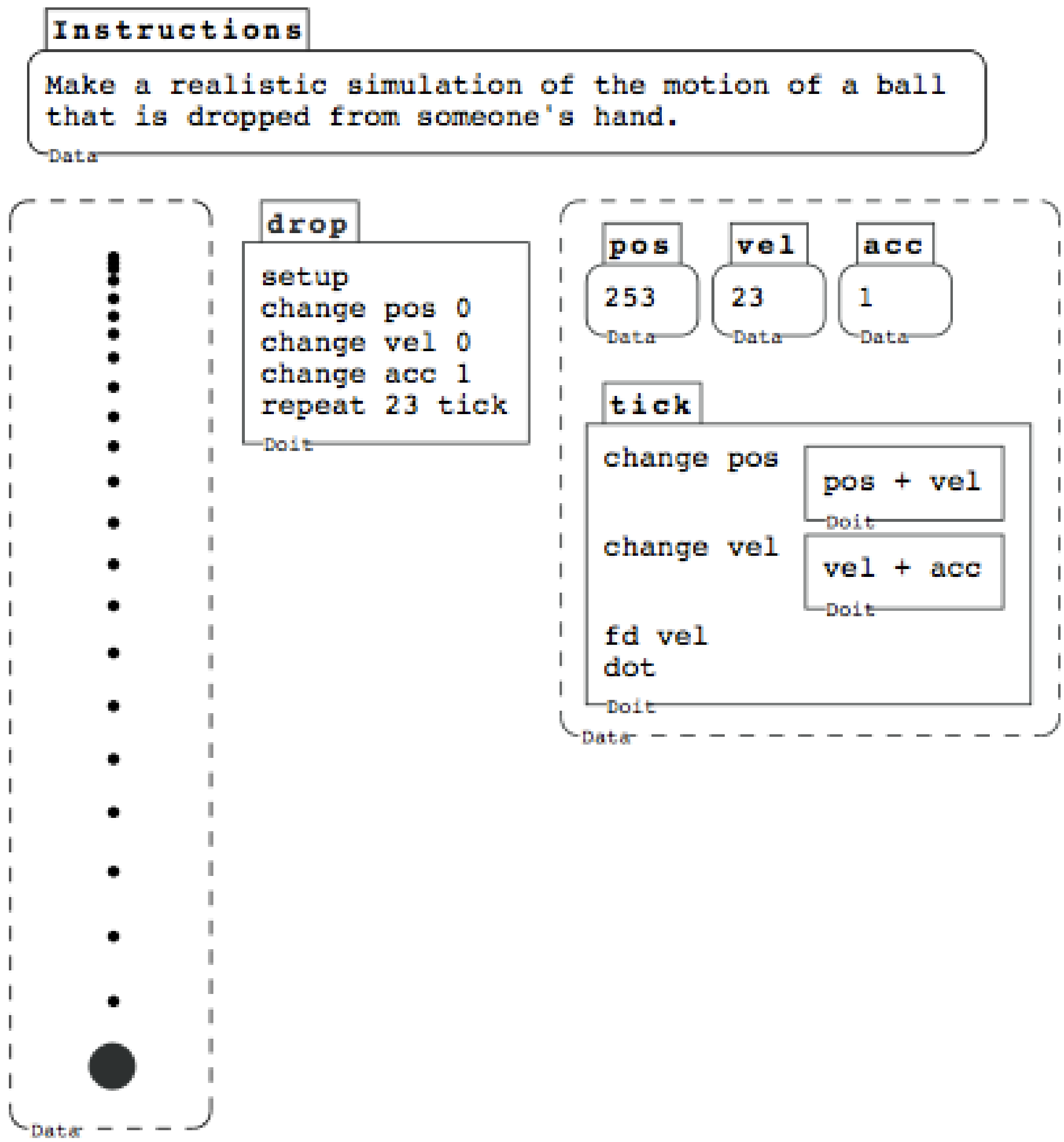


Figure 2.5: A Boxer screen showing the current values of 3 variables, the tick algorithm as nested boxes, and the initialization of the three variables.

ceded any pragmatic usefulness by 15 years. The cost of a computer to run Logo at the time of its development was well over a million dollars, and through the 1970s as the cost of computers came down, researchers refined the Logo language and ran extensive pilot studies with teachers and students. (diSessa, Abelson, & Ploger, 1991) Since the early 1980s educators have taken to the environment with research into its effectiveness as an introductory programming environment being investigated in over 300 instances since 1981[3]. Logo has recently been reincarnated (Mindstorms) as a language for robotics education in the LEGO community and has been used in summer camps to promote scientific inquiry skills and the development of conceptual physics knowledge through robotics.(Williams, Ma, Prejean, Ford, & Lai, 2008)

Penner (2000) notes in a review of research investigating the use of programmable media in education, that the focus of research involving students using Logo has mainly been within the domain of mathematics. Researchers have been interested in how students create procedures using operators to instruct the turtle to draw geometric shapes. And during the 80s, over 90% of the research focused on "how the use of Logo affected the development of general problem-solving skills, critical thinking, collaborative learning, or metacognitive awareness" (Penner, 2000) and not on modeling physical phenomena. Kynigos (1995) investigated a phenomena first observed by diSessa where 12 year-old students who have had experience drawing geometric shapes in Logo have a very difficult time transitioning to programming a turtle that is responsive to Newtonian physics. Kynigos attributes the difficulties to conceptions of processes of change of time, which is not relevant when drawing two adjacent sides of a square, but is relevant when programming a turtle to turn 90 degrees.

DiSessa(1982) reports in a study of elementary school students (11- 12 years old) and an undergraduate student (Jane) of uniform and consistent strategies to kick a moving Logo turtle in a direction towards the target, with the expectation that the turtle will move in the direction of the push. While using commands R, L to turn the direction a turtle faces by 30 degrees,

[3] A cursory search for "Logo" in ERIC for published papers specifically investigating the use of Logo in education, completed on May 15, 2009.

and K to give a turtle some kick/velocity along the direction of the facing turtle, students were asked to answer the question: "What kick (vector) should I provide to turn a corner." The program provided an initial position of the turtle and the target such that the turtle could not face the target. The turtle's initial velocity was zero. DiSessa provides the full 24-game results for Jane, an undergraduate student taking physics. Her Aristotelian strategy was only effective when she decided to provide an anti-kick to the turtle to bring it to rest followed by a rotation of the turtle and subsequent kick to reach the target. DiSessa notes this as a strategy related to that is commonly used in sports (soccer, hockey). Jane's strategies were remarkably similar to strategies used by elementary school students. Further, diSessa notes that Jane "did not, indeed for a time could not, relate the task to all the classroom physics she had had." (1982, p.59) DiSessa speculated that Jane has two types of representational schemes – one which is quite formal for deploying her physics knowledge (which she demonstrated having no fault in using vectors) and one intuitive, loosely organized schema motivated by a situational cue.

The Boxer Group at Berkeley investigated the use of Boxer in a high school physics course taught in 1991. (Sherin, diSessa, & Hammer, 1993) The 15 week long class was split into two parts: the first five weeks were devoted to Boxer programming and the remaining 10 were devoted to introductory physics. Activities during the Boxer portion of the class introduced students to the fundamentals of Boxer. The following 10 week physics course was designed around the creation of five core programs by student design: 1) A dropped ball from rest, 2) a ball dropped from a moving platform, 3) a puck sliding on frictionless surface, 4) a ball thrown in an arc, and 5) a chair pushed at constant speed across a floor. The design activity was facilitated by a teacher-led discussion including all eight students in the class. The teacher wrote syntax and asked follow-up questions of the students clarifying which elements were needed in the program for the turtle to complete a specific task.

Researchers videotaped student discussions involving the teacher and presented selected portions of interest, commenting on their interpretations of the episodes. Sherin commented specifically on an episode where students discussed how to write a modified procedure for the turtle graphic to turn right. The students discussion provided evidence of a pattern the researchers noticed as occurring often in these lab sessions: "students modified the program procedure, and then ran the program in their heads to see how it worked." (Sherin et al., 1993) Sherin attributed this behavior as a method for students within the group to clarify how Boxer commands were interpreted by the program, and if they match the actions required by the design task.

After analyzing the teacher-led design activities, Sherin explains how the structure of the tasks lead to a deeper understanding of physical phenomena without asking the students to analyze anything. Sherin argues that the task to analyze a physical phenomena by novice physics students is an ill-defined task, as it requires "a knowledge of particular reasoning strategies, appropriate approximations, and idealizations, as well as a knowledge of fruitful areas of focus." Rather, Sherin uses these programming episodes to justify a task modeled on how students learn as children – by actions that follow the design, implement, and modify sequence.(Sherin et al., 1993)

Sherin argues the design activity supported a necessary level of abstraction, which was naturally initiated by the students. For example, the students did not worry about air resistance when modeling a moving cart. The teacher had to press students on air resistance when challenging the accuracy of their working program. Sherin notes that idealizations such as ignoring air resistance is disingenuous when attempting to understand the real world; however such simplifications are adequate and perhaps necessary when trying to portray the real world.(Sherin et al., 1993)

Finally, the level of precision during the discussions is an artifact of the nature of programming. Students understood from their experience with previous Boxer programs that the language is precise and compact. While working on the program, students carefully analyzed the Boxer commands by unpacking their effects on the turtle by mentally running the command, and finally comparing this action to the desired goal. The execution of a command must be explicit and free of errors. Boxer contains a limited set of objects which serves to constrain the available set of programming options. Sherin argues that this limitation helps constrain student inquiry to how to use the available features in a physically meaningful way. Students benefit by focusing on the physics concepts rather than how different Boxer objects manipulate the turtle. (Sherin et al., 1993)

For his dissertation thesis, Sherin (1996) investigated how the nature of the Boxer programming environment affected the way in which students think about physics concepts. The Sapir-Whorf hypothesis proposes a framework for how learning is affected by the structural features of a language. Sherin extends this idea to learning physics in a computational environment and proposes that since computational languages have features which make them unique, then learning in the computational language should also be unique. Sherin compared students learning physics concepts in an algebraic representation and in Boxer. He established different interpretive devices and symbolic features between computational physics and algebraic physics, and used think-aloud protocols (Ericsson & Simon, 1993) to look for differences between students learning with each representation. Specifically students completing task using algebraic physics make use of an interpretive device, BALANCING, which is not used by any of the computational physics students. For example, the algebraic interpretation of "$F1 = F2$" represents equality of the values represented on both sides of the equation. The computational interpretation of the same equation is one of assignment. Specifically, the current value of $F2$ replaces the current value of $F1$. Sherin concluded that since the occurrence of behaviors are significantly different between students learning algebraic physics and students learning computational physics, then computational physics is a valid language according to the Sapir-Whorf hypothesis. Sherin acknowledged that there are some limitations to his findings, as computational physics is an offshoot of algebraic physics, and knowledge about algebraic physics is necessary in order to complete computational physics tasks. (Sherin, 2001)

One feature of Boxer's box structure that lends itself to investigations of how students perceive discrete motion is how Boxer represents processes with the tick model. The tick box repeats all lines of code for one unit of change in time (Δt). Boxer allows students to move within the tick procedure to view the effects of each line of code on the turtle and the values inside the variable boxes (Figure 2.5). All of the programs created by students in Sherin's thesis work used this tick box, which provided Sherin the opportunity to observe carefully how students interpret the tick procedure. He noted an exchange between two students who tried to determine if the order of the equations within the tick box mattered, with one student interpreting the tick box as evaluating each line separately occurring through time rather than a series of actions that occur once for each Δt. (Sherin, 2001)

There exists no evidence from literature or a public presence on the web of further development of Boxer[4]. The latest version of Boxer has not kept up with the popular operating systems and hardware of today's modern personal computers.

VPython

The development of a 3D graphics library for Python by the VPython development team[5] provided support for scientific visualization (Scherer, Dubois, & Sherwood, 2000) with a computer language that had already been argued as a suitable language for inexperienced programmers. (Conway, 1997) Ruth Chabay and Bruce Sherwood reported on their experiences introducing VPython to introductory calculus-based physics course at Carnegie Mellon University, offering an anecdote where 75% of students in a second-semester physics course were able to recall how to program a proton to move due to its interaction with an ambient electric field in approximately 15 minutes or less.(Chabay & Sherwood, 2000) Further development on computational activities continued and reported as a full-fledged component of the Modern Mechanics (Chabay & Sherwood, 2004) and Electric & Magnetic Interactions (Chabay & Sherwood, 2006) and fur-

[4]http://dewey.soe.berkeley.edu/boxer/papers.html

[5]David Scherer, Ari Heitner, Ian Peters, David Anderson, Ruth Chabay, and Bruce Sherwood

ther justified as a valid component of the introductory calculus-based physics course (Chabay & Sherwood, 2008).

The VPython programming environment is used to support the central goal of the Matter & Interactions curriculum: to develop a conceptual understanding of fundamental principles. One observable benchmark indicator of progress towards this central goal is for students to spontaneously use fundamental principles to analyze non-trivial systems. Research in physics problem solving reveals that novice students have difficulty thinking about physics in this fashion. A study by Chi et al.(1981) reveals differences between expert and novices during a sorting task for a collection of physics problem statements. Experts sort physics problems according to major fundamental principles that would be used in their solutions while novices sort physics problems according to the features of the system and the problem statement.

There exists very little education research investigating the computational activities that may help students develop this conceptual understanding. However, there are models of knowledge organization, beginning with the Chi study previously mentioned, for how to think about designing learning activities in a way that maximizes systematic processes. Johnson-Laird's theory of mental models argues that novices naturally reason through step-by-step thinking, rather than special-case closed-form solutions. (Johnson-Laird, 1986) Chabay and Sherwood (2008) therefore proposed that computational activities where fundamental principles are "deployed step-by-step to predict the motion of interacting objects" may aid in the conceptual development of the understanding and use of fundamental physics principles. In the M&I Modern Mechanics semester at North Carolina State University, students generate programs analyzing three dynamical systems: the fancart moving on a track, the gravitational interaction between a spacecraft and the Moon and Earth, and a hanging mass-spring. Students work towards completing these tasks in collaborative groups during a weekly studio-style laboratory. There exists no detailed empirical accounts of students completing these activities to assess how the activities promote thinking in a way that leads to conceptual understanding of fundamental principles. Further, empirical anecdotes from TAs indicate the state of students' epistemological focus is characterized by the intensive goal of generating a working program and not on building a greater conceptual understanding.

Buffler et. al.(2008) provide similar accounts of honors physics students' epistemological goals during computational activities. Buffler et al.(2008) detailed a model for designing computational tasks which closely models the methods used by computational physicists to build computational models of physical systems. The sequence of the activity begins with formulating a mathematical model using fundamental physics principles that constrains the time-evolution for a system, making predictions of the motion of objects based on the mathematical model, translating the mathematical model into a computational model, and reflecting on the numerical solution by comparing the visualization to the initial predictions of the mathematical model. This sequence, and the relationship between each component Buffler et al's(2008) conceptual model is shown in Figure 1.1. The model explicitly brings together all representations of a physical system. The activities assume students understand these representations serving as coherent and valid abstractions of the physical system and is cognizant of these limitations when applying to real-world phenomena. Further, Buffler's model does not include any reflection on relating the conceptual models, realized through the computational medium, to the real-world phenomena.

Matthew Kohlmyer's dissertation work investigated two different questions involving the use of computation by introductory physics students. In his first experiment, Kohlmyer provided students the option of solving physics problems by hand or by creating a computer program to complete the calculations. The physics problems were not closed-form, so creating a computer program was the most efficient solution. Students from both the traditional introductory calculus-based physics ($N = 6$) and the M&I introductory calculus-based physics ($N = 6$) courses were recruited for the study. Three difficult problems were given to students to solve, each requiring the iterative use of the momentum principle. Students were not given this information, however were told to expect the problems to be difficult.

None of the students from the traditional physics sections resorted to writing a computer program, or considered the use of a computer as a valid means to a solution. Kohlmyer responded by noting that this finding is expected, as traditional students are not taught iteration or use a programming tool during their formal instruction. Several traditional students attempted to develop integrals, recognizing the challenge of analytically solving equations of motion involving a time-dependent acceleration.

Only two of five M&I students chose to use the computer and generate programs to model the problem. Kohlmyer suggests the interaction of an unfamiliar, novel task may explain the actions of students rather than the ability of students to complete an iterative calculation. According to Kohlmyer, student apprehension of computer programing may be alleviated by the introduction of similar types of tasks in the M&I course. Providing an opportunity for students to make a decision on whether to complete a problem by analytical methods or computational modeling may improve student success on this sort of task.

Kohlmyer's second experiment investigated the difficulties students have with generating VPython programs and the effects of a tutorial-style intervention on these difficulties. Results reported student difficulties with physics concepts rather than with programming syntax. Further, the transcriptions detail how difficult the problem-solving task of generating a computer program for a simplistic two-body system can be for novice students. Kohlmyer adapted instruction to account for these difficulties by minimizing the amount of new information presented to students in any single computational activity, and reduce the number of complex tasks by providing some of the missing information, such as initial conditions.

Kohlmyer's instructional design used detailed evidence of students completing an activity to inform the revisions of the activity such that its instructional goals were met. The recorded data of students completing these activities shows where students run into roadbocks. Further, the recorded data detail the major interventions by Kohlmyer, which provide aid to the students.

Coding the major interventions by its conceptual purpose reveals the frequency with which these concepts impede completion of the programming task. Since some of these difficulties were not intended to be stumbling blocks in the activity, instruction was redesigned to provide additional help in completing the programming tasks. The modified instructional sequence includes an additional computational activity where students are guided through a sequence of calculations for the gravitational force of a static object at several locations in space. The next activity is designed to allow students to use the gravitational force calculations to model a moving and interacting spacecraft with the Earth. Due to the difficulties students were having choosing an appropriate initial velocity of the spacecraft to produce a circular orbit in Kohlmyer's second experiment, the modified instructional sequence provides the initial velocity and position of the spacecraft which will produce an elliptical orbit. The student task is to decide which of the gravitational force calculations are to go inside the loop structure, code these calculations into understandable computer syntax, and include calculations to update the momentum and position of the spacecraft. The influence of the spacecraft on the Earth is approximated as insignificant in this program.

2.2.4 Multiple Representations

Computational activities in M&I take advantage of the different types of representations afforded to VPython programs by importing the visual module. This section reviews how students generate, interpret, and use multiple representations while completing physics related tasks. For example, physics instructors have long taught the importance of generating free-body (formerly force-body) diagrams when demonstrating problem solving algorithms, and comment on students' reluctance to using these diagrams as a way to organize and evaluate information contained in problem statements. Larkin and Simon propose that students' unwillingness to utilize a free-body diagram in a problem solving task is related to the lack of understanding of how to make sense of the novel diagram, finding them useless.

Using VPython provides the opportunity to connect different types of representations as coherent and valid depictions of a computational model. Specifically, the visualization of abstract physics quantities using 3D objects in pseudo-time should help build an understanding of the fundamental principles of physics. In trying to understand how to make sense of each student's success in reaching the goals of the curriculum through activities designed to promote and evaluate these goals, researchers have found it helpful to think about how representations differ along cognitive dimensions.

Animations

Animations serving in a multiple representational regime is merely a collection of diagrams given a temporal sequence. Since the temporal nature of any two diagrams in the sequence is added information, one might hypothesize that this added information may be more useful to learners as it both increases "processability" as well as limits the need for the abstract transformations of elements in a static display. In a review of previous research investigating the perceived advantages of animated diagrams over multiple static diagrams Tversky (2002) criticizes conclusions that animations have a cognitive advantage over diagrams, due to hidden non-temporal features that make the animations superior. That is, the animations used in comparisons either contained extra information (e.g, Park & Gittelmann, 1992; Rieber, 1990), or different procedures such as added interactivity (Kieras, 1992), that were not included in the static diagrams. Further Byrne, Catrambone, and Stasko (1999) found no advantage of animations over providing static diagrams and asking for student predictions, in understanding of computer algorithms. And, conditions that compare the effects of both animation and prediction to conditions without either show no significant benefits of either treatment.(Hegarty, Quilici, Narayanan, Holmquist, & Moreno, 1999)

Zacks, Tversky, and Iyer (2001) investigated how subjects interpret mundane goal-oriented events completed by actors shown on videotape (such as making a bed or assembling a saxophone) by explaining and segmenting the event into units, making decisions when "one unit ends and one unit begins." Analysis reveals that the population simultaneously keep track of physical changes, goals and plans, causes and effects, and actions and objects. Segmenting includes a hierarchical structure of recursive activities (when valid) and verbal explanations of segments reflect the physical changes of objects towards the goal-state. This study is relevant to how one perceives animations, particularly for animations showing processes that can be described such as in circuits, pulley systems, or traveling across town. Zacks et. al. (2001) points to this research and others in claiming that even when motion is continuous, people conceive of it in discrete steps, similar to the key frames that would be present in multiple static diagrams of the event.

An open question is how animations that lack a goal-oriented focus are perceived into discrete events, the amount of abstraction inherent in these discrete events, and if these discrete events contain references to mechanism or merely descriptive of the motion characteristics of the animated objects.

Following Tversky's methodological analysis of research comparing the perceived learning gains attributed to animations over static diagrams, researchers have shifted their focus to evaluate how animations may benefit learning(e.g. (Hegarty, n.d.)) by taking advantage of their interactivity (e.g. (Lowe, 2004)), and studying cognitive strategies for extracting information (). Ainsworth(2008) reviewed the progress of research investigating how one learns with animation, identifying six "levels of explanation", or domains with which to make conclusions on an animation's effectiveness. Domains relevant to this review include the 1) cognitive and perceptual, 2) metacognitive, 3) rhetorical, and 4) strategic. This review has addressed each of these except for the rhetorical domain, concerning how social interactions influence learning. Since the research discussed by Ainsworth refers more appropriately to non-interactive animations, this review will discuss in the next section how representations as a whole may be used in social interactions as a device to instigate group mediation.

Different functions of multiple representations

In earlier work, Ainsworth(1999) synthesizes research in multiple representations to build a taxonomy of how multiple representations are used to "maximize learning outcomes." The taxonomy segments all uses of multiple representations into three separate functions: 1) to support complementary cognitive processes, 2) to constrain interpretations across multiple representations (where one representation acts to constrain the interpretation of another), and 3) to aid in constructing a deeper understanding.

This review has discussed how Larkin and Simon (1987) and Cox and Brna (1995) already support the idea that different representations of the same information support different inferences and cognitive strategies. Ainsworth's(1999) second function of multiple representations lies outside the scope of this review, referring specifically to graphs in physics as an example in which one representation may be used to constrain interpretations. The third function refers to the potential of multiple representations to support abstraction, relationships between several representations, and generalizations to novel situations. Abstraction refers to two different processes. First, as a term used by computer scientists (e.g. (Wing, 2006)), abstraction is similar to reduction with transformation - in line with model building. Second, abstraction refers to the representation of descriptive quantities that are used to characterize relationships between objects and their states. Multiple representations may also serve as a medium to facilitate bidirectional learning, where inferences and information presented in one representation may be used to learn from another. Finally, multiple representations may help students use their knowledge (existing or learned) in isomorphic yet novel situations. The ability of multiple representations to support transfer has not been explored in the literature.

Constructing multiple representations

Cox(1999) reviews the research literature investigating how constructing multiple representations may aid in problem solving. Cox characterizes the task of constructing a representation as a process that involves students "examining their own ideas, re-order information, translate information from one modality to another (re-represent), and keep track of their progress through the problem."(1999) Further, externalizing information into a representation may act to flesh out specific mental models used to visualize a particular problem, useful where problem solving is too difficult to complete internally. The underlying mechanism is the role of externalization serving as a go-between in the transfer and storage of information among cognitive sub-systems as described by dual-coding theory. Wilkin (1997) reports in a study on the links between students creating a diagram and their self-explanations, that diagrams contribute to inaccurate self-explanations and concludes that the self-explanation effect may only apply to strong students. However, in Wilkin's study, low performing students were encouraged to use a diagram system that they did not understand or find helpful in the same way the diagram was created and used during the self-explanations of high performing students.(Cox, 1999)

In his own work, Cox investigated any differences in the models of reasoning used by students who select, construct, evaluate one's own representations or use pre-fabricated representations from a textbook. Cox specifically investigated how one externalizes the representation, including any errors made in the representation's construction, and the interpretation of the representation.

Cox (1999) , in his summary, explores the different possible uses of constructing multiple representations as an aid in problem solving. Cox's list is filtered through the aims of this literature review, and include ways in which constructing multiple representations may be used via:

- translating information from one type of representation to another

- directing attention
- providing perceptual assistance
- the self-expanation effect
- facilitating the inference of motion (mental animation)
- refining and disambiguating mental images

Mechanistic reasoning and multiple representations

To achieve the conceptual goals of the M&I course, students must see physics as a collection of a limited set of fundamental principles, through which one builds an understanding of systems and the evolution of the dynamics through time. As characterized by Russ, (2008) this type of reasoning addresses mechanism as a basis for scientific explanations, which include causal relationships between principles and observation, rejecting descriptions of objects/processes by their function or intent (Carey, 1995), searching for an underlying relevant structure (Chin & Brown, 2000), all which is constrained by the framework of previous experiences in the real-world (diSessa, 1993).

Russ's (2008) dissertation work focused on developing a working hierarchical coding scheme through which one can determine the ability with which students use mechanistic reason, along with the transitions used when subjects move to, or out of this type of reasoning. The foundations for the coding scheme come from adapting Machamer, Darden and Craver's (2000) framework analyzing student and scientist thinking. The framework posits mechanical reasoning as a process involving mechanisms defined as "a series of activities of entities that bring about the finish or termination conditions [from the set-up conditions] in a regular way."(Machamer et al., 2000) The nine categories of codes include: 1) Describing the target system, 2) Identifying the setup conditions, 3) Identifying entities (things that play roles in producing phenomena), 4) Identifying activities (interactions), 5) Identifying properties of entities (such as mass), 6)

Identifying the organization of entities (spatial relations between entities), 7) Chaining: Backward and forward (claims about what might have brought about the entities current state, or its state in the future).

Russ (2008) applied this coding scheme as an analysis of student discourse for first-grade students to determine the quality of students' mechanistic reasoning engaged during the interactions between a teacher and her students. Russ provides episodes from a transcript and analyzes the quantity of each code, deeming episodes with prolonged statements at the top of the hierarchy as episodes with strong evidence of mechanistic reasoning. Further, Russ investigates the discriminatory power of the scheme to recognize transitions within an episode, noting that high-level codes tend to cluster. Transitions to higher-levels were either instigated by students or the teacher. Russ notes that transitions to level 7 all came from levels 3,4, and 5, suggesting that these levels provide the necessary building blocks to determine new mechanisms to consider.

Russ (2008) notes the potential of animated models to facilitate mechanistic reasoning, referencing work by Bechtel and Abrahamsen (2005), who posits that animations and diagrams help "supplement human abilities to imagine a system in action."

2.3 Program Comprehension

2.3.1 Introduction

Cognitive psychologists focused their research efforts of modeling the mind as metaphorical computer during the 1980s and well into the 90s. As such, much of the research by those who adopt this metaphor studied programmers in how they generate and comprehend computer programs. Cognitive psychologists explored the difference between expert and novice programmers in how they plan, generate, and debug programs in order to build models of problem solving and program comprehension. The first half of this section explores these findings in terms of how expert and novice programmers comprehend novel programs. The second half of this section investigates parallels between program comprehension and reading comprehension, specifically how one designs instructional tasks to improve reading comprehension and explores possible ways in which these tasks might improve program comprehension.

2.3.2 Cognition and Comprehension

Introduction

This section of the literature review details theoretical frameworks in comprehension, as supported by research by cognitive psychologists. The section begins with reviewing the interest in using programming tasks to understand how one makes sense of existing programs. This understanding leads to several models of program comprehension, which differ according to the programer's experience with the language. Next, the review discusses the reading comprehension framework as discussed in Palinscar & Brown and recently used by Richards (2008; 2009) to understand how students evaluate worked example physics problems as being correct or incorrect.

Program composition versus comprehension

Pennington, Nicolich, & Rahm(1995) investigated the use-specificity principle, derived from Anderson's ACT* theory of skill acquisition, as it applies to the learning of cognitive subskills involved in different programming tasks. The principle of the use specificity of knowledge claims that tasks that require the same abstract declarative knowledge will be compiled into different task-specific procedures (productions) for different uses. Or, any declarative knowledge learned for a specific task is encoded and in turn used for transfer tasks that share many of the same features of the learned task. The same declarative knowledge will take different forms, based on the way it is used. The principle predicts that two different tasks which rely on the same declarative knowledge for successful completion will produce little success if the tasks do not share the same productions as the task in which the declarative knowledge was learned. The principle has significant implications for learning of domains with multiple sub-skills.

Pennington et al(1995) investigated this claim specifically by comparing the transfer of knowledge between comprehension and composition subtasks based on overlapping declarative knowledge bases. The investigation compares two subtasks according to the productions used to complete each task, then develops tasks to control for the declarative knowledge necessary to complete the task. Any lack of transfer between the subtasks may be attributed to the use specificity claim which predicts that transfer will not occur between tasks which require different productions. Completing a GMOS task analysis of the productions necessary for each task reveals that 22% of the total comprehension task productions are also shared with the generation task. And, 12% of the total generation task productions are shared with the comprehension task. As a result, the use specificity principle predicts those learning the generation task first will produce greater transfer than those beginning with the comprehension task.

Pennington et al.(1995) reported empirical results which refute the predicted transfer outcomes of the use-specificity principle. The theoretical simulation based on the GMOS task analysis correctly predicted that the generation task would take longer than the comprehension task. However, the simulation "did not predict the performance difficulty based on percent correct (both tasks were equivalent) nor the relative learning difficulty of the two subtasks (equally difficult to learn)." In terms of transfer, greater transfer was observed going from comprehension tasks to generation tasks - a direct contradiction to the predicted transfer using the overlap of productions based from ACT*.

To account for the failure of the model to predict these outcomes, Pennington et al.(1995) further investigated the effect of descriptive declarative elaborations made by students completing both tasks to see if the elaborations can account for the discrepancy between the outcome and the prediction. The study used a think-aloud protocol on subjects randomly assigned to one of the two same conditions as in the first study. Elaborations were tracked and coded into four categories: those used to strengthen the correct definitions of functions, those used to elaborate feedback on the correct answer, those used during mistakes or incorrectly interpreting feedback, and those contrasting an incorrect elaboration with a correct elaboration. Comparing subject declarative elaborations and their associations with a concept across the two training treatments shows that there is substantial overlap in the declarative elaborations, regardless if the subject was trained in evaluating or generating programs.

Models of Program Comprehension

Brooks(1983) proposed the first model of program comprehension, presenting a set of mechanisms and relationships that helps explain variability among the ability of programmers to comprehend programs. The model explains why different programming languages are more comprehendible than others, why tasks such as debugging versus generating code affects comprehension, and differences in individuals completing the same comprehension task. Brooks's model defines comprehension as the mapping of information in the programming domain to the problem domain, possibly through several intermediate domains. Mathematical domains are used when interpreting the program code into recognizable equations, operational domains are called when objects are assigned attributes, and execution domains determine how the program will compile the program code. Successful comprehension includes reconstructing part or all of these mappings. Finally, the reconstruction process is expectation driven by the creation, confirmation, and refinement of hypotheses. (Brooks, 1983)

Shneiderman's model (Shneiderman & Mayer, 1979) incorporates chunking program code into a semantic hierarchical structure in working memory with high-level concepts at the top and low-level particulars at the bottom. The structure contains information that is generalizable across multiple programming languages and tools. As the structure takes shape in working memory, comparisons are made to similar semantic hierarchical structures in long-term memory and a syntactic database of language specific knowledge.

Soloway et al.(1988) presented a model which argues that the comprehension of a program does not lie in the line by line analysis of the program, instead one extracts a recognizable block of code, attributes a specific familiar function to the block of code and continue in a search pattern until enough information is collected to accurately comprehend the program. This strategy relies on programs that share similar structures and/or completes similar tasks. The model uses three types of plans: strategic, tactical, and implementation. Strategic plans contain no lower level detail or the constructs used for implementation. Tactical plans contain the strategies for solving a problem, including algorithms or abstract data-structures. However, they are not tied to any one language so they can not alone comprehend a program. Implementation plans contain the language specific instructions for how to represent the tactical plan as valid code. Comprehension includes creating subgoals to understand the higher-level function of the program and applying plans to satisfy the program goals. The model allows the creation of new plans which are stored in long-term memory.

Pennington's model (Pennington, 1987) describes how the programmer comprehends code that is completely new by potentially building two types of models of the program code. First, the programmer builds an abstraction of the program by first searching for elementary blocks of code in the program, called beacon lines. These beacon lines are used to inform the programmer of how data is controlled in the program. Second, the programmer builds a model of the program that is functional and relates the program code and function to the real-world.

All four major models of program comprehension address: 1) a mental representation of the code, 2) a knowledge base stored in long-term memory, and 3) a procedure for combining new information with this knowledge base to alter the mental representation of the program. The Brooks model only permits the modification of the mental representation through the use of hypothesis(3). Later models add additional detail to the form of the mental representation(1), without discussing the assimilation process(3). Shneiderman's model adds a hierarchical organization of knowledge and differentiates between semantic and syntactic knowledge. Soloway and Ehrlich adds detail to knowledge structures, categorized by strategic, tactical, or implementation plans. These plans are useful when engaged in a top-down decomposition of the program code by beginning with assumptions about the structure of the code or the functions it must contain. Pennington's model argues that programmers, when viewing completely new code, searches for beacons of code to develop a bottom-up model of the program. As the program model begins to form a coherent structure, the situation model is developed by combining the control-flow/functional abstraction with previous domain-specific knowledge.

Reading Comprehension

Palinscar & Brown(1984) identified six functions which lead to successful comprehension of reading passages. Using these six functions, Palinscar & Brown developed four tasks which require one or more of the six functions to complete. In a research setting, Palinscar & Brown used a reciprocal teaching pedagogy which emphasizes coaching to instruct elementary students on how to complete the tasks to satisfaction and evaluated the success of the pedagogy based on length of time to achieve improved comprehension.

Richards(2010) upon reviewing the worked example literature, identified how the functions involved in the comprehension of reading passages also correlate to student behaviors while reviewing worked examples. In his analysis of the literature, Richards identified that none of the worked-example studies identified students implementing internal consistency checks. Richards was able to show that students completing a problem selection task used internal consistency as a way to decide which problem solution was correct out of a set of four possible solutions.

2.4 Sensemaking in Physics Education Research

Overview of Major Perspectives in PER

During the 30 year history of scholarly research in the teaching and learning of physics, researchers have painted a picture of learning which is increasingly student-centered. It is easy to forget that this was not always the case. In her Millikan award lecture, McDermott argued that the introductory college course curriculum was designed by "faculty who think of students as very much like themselves." (McDermott, 1991) Further, the curriculum contained assumptions about student learning, including: (1) A functional view of physics is developed by solving standard quantitative problems and (2) qualitative reasoning skills are enhanced as a result of learning quantitative physics.(McDermott, 1993)

Guided by neo-Piagetian conflict theory, constructivism evolved and models such as conflict theory and reordering of schema were used to explain how students learn. Within the physics domain, researchers investigated how students complete qualitative physics tasks. Researchers noticed that students needed training in building qualitative models and drawing coherent inferences from those models. As a result, the community began documenting physics concepts with which students have difficulty learning and applying to new problems, and in many cases the prescribed instructional strategies/materials used to alleviate said difficulties.(McDermott & Redish, 1999)

On a parallel front, researchers began looking for commonalities among these initial misconceptions and saw consistencies in how students incorrectly complete physics tasks. Clement(1982) documented a collection of these alternate conceptions and DiSessa(1993) saw underlying phenomenological principles based on students' real-world experiences. This research led to the foundations of Resource Theory and justified how attention to a student's epistemology affects the learning process. Further, Gunstone(1992) argued to understand the learning process, one must understand the metacognitive strategies the student initiates during the problem solving process.

The perspectives of individual learning developed as researchers in science education focused their attention on the social dynamics of learning. Otero(2001) reviews research literature leading to multiple-perspective frameworks of student learning. Multiple-perspective frameworks allow the researcher to apply "lenses" to the data and either investigate the conceptual development of the individual or the conceptual development that emerges from the social interactions within the collaborative group setting. Otero critiques the body of literature for focusing on single-events, and not using the perspectives to explore transients in either the conceptual understanding over time or the changes to the group setting.

Finally, researchers across disciplines note the special role tools play in mediating awareness, problem solving, and communication. Historically, the PER community evaluates innovations in instructional tools according to conceptual learning gains extracted from valid and reliable test instruments using quantitative methods. Otero's(2001) dissertation reviews the literature which investigates the role of tools and tasks in how they facilitate learning. For example, Kelly and Crawford (1998) show how students use the computer representation of experimental data via real-time graphs in MBLs to direct sensemaking, in an ethnographic setting using discourse analysis. Otero's work analyzes ethnographic episodes while students use simulations of abstract static electric phenomena to build and maintain a conceptual model for static electricity during lab exercises. Otero identifies and compares sensemaking episodes in the transcript of student working on simulations and student working on experimental tasks to show how the computational simulation, as a learning tool, influences the development of conceptual models in a collaborative working environment.

Research in the PER community has expanded to include ethnographic accounts of how the curriculum, tools, and environment interact with students. The community continues to invest time in research-based instructional design to address practical concerns of educators who may adopt the instructional materials to guide the student learning of physics.

2.4.1 Sensemaking in Physics Education Research

Beginnings

Sensemaking has been identified as a necessary ability to practice science (Etkina & Heuvelen, 2007) and communicate information(Dervin 1992). In communications, researchers are interested in sensemaking as it is central to all communicating situations (Dervin, 1983). Communications researchers define sense-making as constructing sense of their world, constructing information needs and uses for information. Dervin identified three core conceptual premises of the sense-making approach: 1) reality is neither complete nor constant but filled with fundamental and pervasive discontinuities or gaps, 2) information is a product of human observations, and 3) information seeking activities are constructing activities where one creates a personal account of sense. To create this sense, which ultimately guides one's actions, sense-making research investigates how individuals use the observations of others as well as their own observations to construct and reconstruct their pictures of reality. Researchers investigate sense-making to anticipate the informational needs of those seeking sense, the perceived gap between the use of the information and the current conceptual analysis of the needs of the individuals, the strategies used by individuals to determine what information needs exist. (Dervin, 1992)

Schoenfeld(1992) agrees with Dervin and argues for a paradigm shift in mathematical pedagogy - where teaching mathematics should incorporate more than the "tools of the trade." Schoenfeld argues that learning mathematics should be empowering, and framed in classroom teaching as a living science where solutions are sought rather than memorizing procedures. Schoenfield argues that the mathematics classroom should be an environment where patterns are explored rather than memorizing formulas and conjectures are formulated rather than crunching through exercises. Schoenfeld agrees with Resnick's conclusions from the body of work on the nature of cognition which shows that "becoming a good thinker in any domain – may be as much a matter of acquiring the habits and dispositions of interpretation and sense-making as of acquiring any particular set of skills, strategies, or knowledge." (Resnick, 1989, p. 58)

During the late 80s through the mid 90s, psychologists began studying organizations to determine specifically how information is processed and created within organizations used for decision-making. Weick's(2005) review of the sensemaking literature by psychologists studying organizations identifies the following central questions for people interested in sensemaking: How do people construct sense, why, and with what effects does sense have on people's actions. Many investigators imply or explicitly state how sensemaking involves the placement of perceptual stimuli into a framework (Starbuck & Milliken, Louis 1980), which then enables them to "comprehend, understand, explain, attribute, extrapolate, and predict" (p.51). Louis views sensemaking as a retrospective occurrence motivated by discrepancies between expectations and reality:

> Sense making can be viewed as a recurring cycle comprised of a sequence of events occurring over time. The cycle begins as individuals form unconscious and conscious anticipations and assumptions, which share as predictions about future events. Subsequently, individuals experience events that may be discrepant from predictions. Discrepant events, or surprises, trigger a need for explanation, or post-diction, and correspondingly, for a process through which interpretations of discrepancies are developed. Interpretation, or meaning is attributed to surprises . . . It is crucial to

> note that meaning is assigned to surprise as an output of the sense-making process, rather than arising concurrently with the perception or detection of differences" (Louis, 1980, p.241)"

And, education researchers investigated how children studying a unit on wetlands use different sense-making heuristics to generate concepts.(Hulland & Mumby, 1994) Cognitive researchers have investigated how sense-making affects the development of situation awareness, structures and processes involved with sensemaking in organizations, and factors that enable or inhibit organizational sensemaking under abnormal conditions, in the context of military operations. (Leedom, 2001)

PER

In PER, sensemaking appears most often in the literature when researchers discuss student epistemologies or student engagement. Assessments of student attitudes and beliefs ask students to report how they view the enterprise of learning physics along the answermaking-sensemaking continuum. (e.g. E. F. Redish, Saul, & Steinberg, 1998; Adams et al., 2006) Or, as Redish differentiates the two,

> ...most professional scientists who teach at both the undergraduate and graduate levels will recognize a binary stage, in which students just want to be told the right answers, and a constructivist stage in which the student takes charge of building his or her own understanding. Consciously constructivist students carry out their own evaluation of an approach, equation, or result, and understand both the conditions of validity and the relation to fundamental physical principles. Students who want to become creative scientists will have to move from the binary to the constructivist stage. This is the transition that we want to explore.(Redish, 1998, p. 213)

Lising and Elby (2005) investigates this epistemology as it supports or inhibits reasoning further in an in-depth analysis of one student, Jan, as she progresses through an algebra-based physics course. Presenting evidence from weekly interviews, Lising and Elby argues that it is highly plausible that Jan's own epistemology provides a barrier between making coherent connections between the use of formal and intuitive reasoning. Jan resists reconciling the two types of reasoning throughout the course. Kanim(2001) developed instructional materials to help bridge this gap through analogies and the use of scaffolded questions to prompt considerations between these two types of reasoning. Lising and Elby note that Kanim's worksheets may show even more promise as an effective instructional tool if the worksheets focus on drawing out individual students' epistemologies. It is with that motivation which Elby proceeded to adapt the Washington Tutorial in Physics to confront the intuitive reasoning of novice students in an algebra-based physics course.(Scherr & Elby, 2006)

Researchers at the University of Maryland have been investigating how to improve the algebra-based physics course to modify curriculum to include a conceptual understanding of mathematical equations (Hull, Kuo, Gupta, & Elby, 2009). Scherr and Hammer (2009) investigates how students frame lab tutorials to understand how students answer the question "What am I supposed to be doing?" Scherr and Hammer identified student behaviors that suggest transitions in how a collaborative group switches between frames, whereas previously, only lingusitic markers were used to identify these transitions.

Beyond a students' view of how knowledge is constructed in a physics course, investigators use sense-making frameworks to identify the activity of students while completing instructional tasks. Otero's dissertation work analyzed how a computer simulation in the physics laboratory may be used to help build appropriate abstract conceptual models of static electricity when paired with laboratory activities. Otero was interested in two facets of the intervention: 1) How did student conceptual models develop over time, and 2) which student activities would be considered scientific. Otero compared the experimental group to a control who did not receive the simulation intervention the time spent making sense of data and building explanations.(Otero, 2001)

Otero characterized how students used the simulator as a scaffolding agent while creating explanatory models for laboratory results. Otero argues the simulator provides a shared representation for which individuals within a collaborative group may use to express their explanations of laboratory observations. Further, Otero detailed evidence of students modifying their conceptual models of electrostatics to explain observations of both the simulation and the experiment.

Kung (2007) used a sensemaking coding scheme to identify the metacognitive statements made by students completing physics lab activities and asks if it is pedagogically productive to do so. Kung completes two passes of the transcripts, assigning codes to student activity (sense-making, logistic, off-task) and student use of metacognition. One of these three codes identifies sense-making episodes, defined by Kung as "students responding to each other, making progress towards an answer, and holding a coherent conversation." The context of the sense-making episode may include having a "discussion about physics formulas, concepts, the experimental design, their data, the laboratory question, etc." Kung reports how metacognition statements may serve as transitions from student activity involving the logistics of the lab, to student activity involving how students engage in sense-making.

The Investigative Science Learning Environment (ISLE) curriculum (Etkina & Heuvelen, 2007) is designed to develop students abilities in the practice of science. After analyzing student lab reports, Etkina et al.(2006) determined that students significantly improved their abilities to "design an experiment to solve a problem, develop a mathematical procedure to analyze data, and to evaluate experimental uncertainties and assumptions." Karelina (Karelina & Etkina, 2007) analyzed observer field notes and transcriptions of recordings capturing student activity to determine if discussions between students also show characteristics found in the practice of science by professional scientists. Karelina developed a coding scheme based on Kung's codes for student activity (sense-making, logistics, off-task) and developed sub-codes which correspond to the context of the student discussion, described in Table 2.2.

For example, design, model, assumptions, uncertainties, minimizing and revising were all sub-codes used when sense-making discussions were identified from the transcript. Using a temporal map of the occurrence and duration of sense-making, logical, and off-task episodes, Karelina analyzed differences between cookbook + explanation labs and ISLE labs in the outcomes of each sense-making episode and the duration of the sense-making episodes using temporal plots as shown in Figure 2.6

2.5 Where does this leave us?

All of these investigations involved populations of physics students enrolled in an algebra-based physics course. There are no investigations of how students in a calculus-based introductory physics course decides to initiate activity which is characterized as sensemaking. Possible reasons for this selectivity has not been commented on in the literature since McDermott's

Table 2.2: Sense-making codes from Karelina&Etkina (2007).

D	Design	Discussing experimental design and setup, planning the experiment, etc.
M	Mathematical Model	Choosing the mathematical model and the physical quantities to be measured
A	Assumptions	Discussing the effects of assumptions inherent in the mathematical model and the model's limitations
U	Uncertainties	Discussing sources and calculating values of experimental uncertainties
Min	Minimizing	Discussing the ways to minimize uncertainties and the effects of the assumptions on the outcome
R	Revising	Discussing reasons for the discrepancy and the ways to improve the experimental design to get the discrepancy less than the uncertainty
O	Unexpected observations	Discussing the reasons for obtaining unexpected data
VA	Validating assumptions	Discussing how to justify assumptions

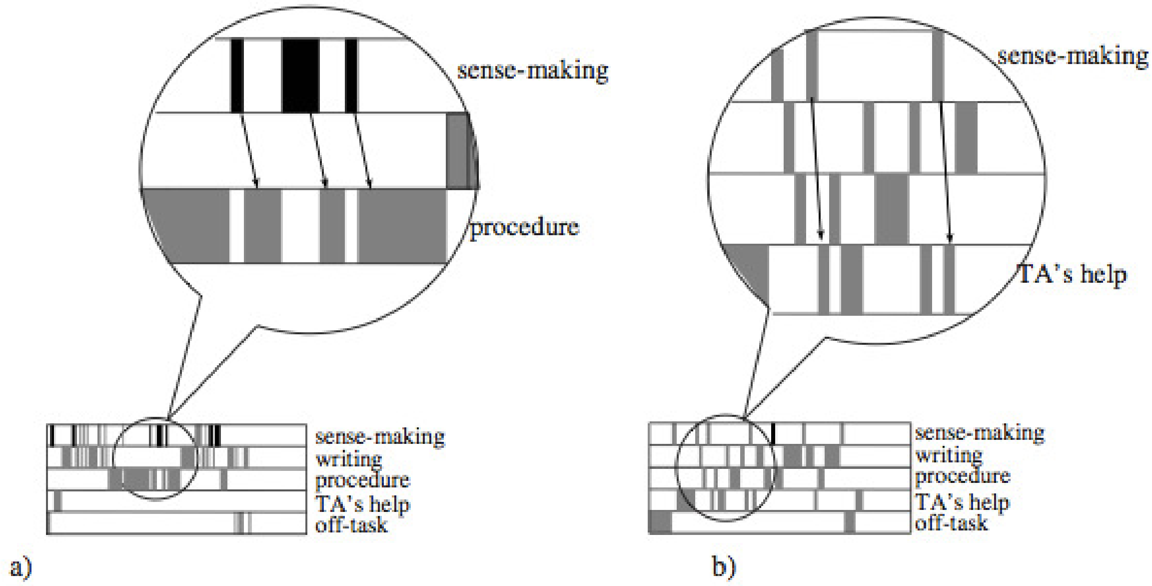


Figure 2.6: Temporal profile of sensemaking outcomes in a)ISLE labs where "a discussion leads to a discussion and a conclusion" and b) non-design labs where "sensemaking leads to TA's help with the TA answering and explaining."(Karelina et al, 2007)

editorial sixteen years ago. Researchers have used the sensemaking frameworks to identify when study participants are engaged in the tasks described by an instructional document and further look at the features of these sensemaking episodes. The models of programming comprehension developed through the 80s and early 90s identified how novices and experts read program code to characterize the strategies programmers used to inform an understanding of the function of the program. And, Palinscar and Brown identified a framework for how to design instructional tasks which improves one's ability to successfully comprehend reading passages within a reciprocal-teaching environment. Instructional tasks in physics which use computation as a way to have students practice utilizing their physics knowledge in a new environment ask students to build computer models through the task of programming.

There is a large hole in the research literature investigating how the comprehension of computer programs may help facilitate the conceptual development of physics and its fundamental principles. However, there is a large body of work which informs how one might go about remedying this omission. But, before building up a methodology for taking steps to investigate how the comprehension of a physics program might enhance one's understanding of physics we have to argue that the comprehension of physics program is not a trivial task. And we must argue with evidence from the student interpretation of physics programs that one must use one's physics knowledge to inform a coherent interpretation. From this there are several routes through which one can build a research study to investigate how the interpretation of physics quantities from the study of a computer program may evolve with practice and hopefully improve one's understanding of fundamental physics principles. But for first steps, this thesis will look at the use of a participant's physics knowledge through the lens of a task which asks students to make a prediction how a program modeling a physical system evolves the motion of the objects in a 3D visual scene.

Chapter 3

Pilot Study

A pilot study was conducted in May 2008, following the final exam period for the spring term. The goal of this study was to investigate how individual students modify VPython programs as they complete tasks to reach a specified goal state. This section details the participants in the study, the procedure followed to gather verbal data as students complete targeted tasks, and what was learned and used to design a larger study involving more students from the same population.

3.1 Participants

Participants were recruited from a population of students who were enrolled in the SCALE-UP section of the calculus-based introductory physics course using the Matter & Interactions: Modern Mechanics text (2nd edition) during the Spring 2008 term. SCALE-UP (Student Centered Active Learning Environment for Undergraduate Programs) is an environment designed for active learning in large enrollment courses. At NCSU, the physics SCALE-UP course replaces class time traditionally used for lecturing with student activities completed in collaborative groups. (Gaffney, Richards, Kustusch, Ding, & Beichner, 2008) Recruitment was limited to the SCALE-UP section for two reasons. First, the researcher served as the teaching assistant (TA) for the course and the personal connection to this student population may have helped students feel more comfortable volunteering for a research study by the TA. Second, students from SCALE-UP and non-SCALE-UP sections had different experiences with computational modeling activities due to features of the SCALE-UP learning environment, including instructor interactions, pacing, and scheduling. The differences between SCALE-UP and non-SCALE-UP courses was enough to justify controlling for the instructional environment. Recruitment began after final grades were determined for the course and submitted to the Registrar.

Six students volunteered, five males and one female. All volunteers completed exercises during the course using VPython to model physical systems. Upon reviewing course attendance records, it was determined that all students were present during the semester on days when VPython activities were completed in class. All participants were members of different collaborative groups for VPython activities.

One NCSU graduate student was recruited to complete the activities before the six undergraduate students were brought in for interviews. This student's verbal data was not used in the analysis; however, the data was used to identify errors in the interview script and offer suggestions for script clarity.

3.2 Environment

Each participant met with me individually for a 90-minute interview session in the Physics Education Research and Development group's laboratory space. Participants sat at a table with a computer workstation, monitor, keyboard, mouse and whiteboard. The interview room was equipped with an overhead ceiling mounted camera aimed towards a whiteboard lying on a table in front of the participant. The camera's position provided an unobstructed field-of-view to capture what the participant wrote on the whiteboard with dry-erase markers. Additionally, a second camera was positioned behind the participant aimed over the shoulder towards the computer screen to capture what the participant saw on the screen. Both video feeds were synchronized with a wireless microphone worn by the participant. Each camera recorded the video and audio feeds to tape.

3.3 Procedure

All participants heard identical instructions (except where noted) as read from an interview script. Participants were instructed to say aloud everything they were thinking, as described by the Think-Aloud Protocol (Ericsson & Simon, 1993) to capture both what participants did during the task as well as their thinking as reported by their verbalizations. Participants completed practice exercises to get used to the protocol.

The study consisted of four tasks of increasing difficulty. Participants worked on the tasks until either all tasks were complete or the interview had reached 90-minutes in length. At the end of the interview, participants were asked follow-up questions based on their work in the interview and paid for their time. Two of the six participants completed all tasks and finished early. The four tasks are summarized in Table 3.1 and discussed below.

Table 3.1: Pilot Study Tasks.

Task	Participants	Task Description
1a	6	Study `Spacecraft.py` program
1b	3	Draw a prediction of the visual output
1c	6	Run the program
1d	6	Determine when $\vec{p}$ and $\vec{F}_{grav}$ are perpendicular
2	6	Change program to produce a circular orbit
3	6	Add code to the shell program `Earth_Moon.py` to achieve a figure-8 orbit for the craft around the Earth and Moon
4	2	Add code to the shell program BinaryStar.py to achieve a closed orbit of a binary star system

First, all participants were asked to study the code in an example program and indicate when finished doing so. The example program code, shown in Figure 3.1, predicts the dynamics of a spacecraft orbiting the Earth (location fixed), presented in a 3D, navigable graphical output at runtime (Figure 3.1). Three of the six participants were asked to run the program and were reminded of the protocol's instructions to say everything one thinks as the program runs. Participants were then asked to determine if the momentum of the spacecraft was perpendicular to the net force at any time during the rendering of the visual output. The other three participants were asked to complete an additional task before running the program for the first time: to draw a prediction on the whiteboard of what one expects to see in the graphical output.

The second task asked all participants to alter the program such that the shape of the orbit changes from an ellipse to a circle. For the third task, participants opened a shell program, or a program that is missing several chunks of code, including any functional code used to create the dynamics of the graphical output. The shell program contained lines of code defining the same two objects as in the example program, the spacecraft and the Earth, with the addition of the Moon. The shell program did not have any of the calculations used to determine the gravitational force or update the momentum and position of the spacecraft. Participants were asked to add to the program lines of code necessary to get the spacecraft to complete a figure-8 orbit of the Earth and the Moon.

The fourth and final task asked participants to open a shell program containing two 3D objects representing two massive stars and modify the program to get the stars orbiting each other to model a binary star system. Only two of the six participants started the fourth task.

Spacecraft_test.py - /Users/Weatherford/Documents/Research/Pilot Study/Pilot Prog...

```
#from __future__ import *
from visual import *
#from __future__ import *
from visual.graph import *
scene.width =1024
scene.height = 760


#CONSTANTS
G = 6.7e-11
mCraft = 1.7e22
mEarth = 6e24
deltat = 10

#OBJECTS AND INITIAL VALUES
Earth = sphere(pos=vector(0.,0.,0.), radius=6.4e6, color=color.cyan)
Craft = sphere(pos=vector(-10*Earth.radius,0.,0.), radius=1.75e6, color=color.yellow)
trail = curve(color=Craft.color)
scene.autoscale = 1

vCraft = vector(0,1000,0)
pCraft = mCraft*vCraft
t = 0

#CALCULATIONS
while t < 60*365*24*60*60.:
    rate(2000)        ## slow down motion to make animation look nicer

    r=Craft.pos-Earth.pos
    rmag=sqrt(r.x**2+r.y**2+r.z**2)
    rhat = r/rmag

    Fgrav=-(G*mEarth*mCraft)/(rmag)**2
    Fnet = Fgrav * rhat
    pCraft = pCraft + Fnet * deltat
    Craft.pos = Craft.pos + (pCraft/mCraft)*deltat

    trail.append(pos=Craft.pos)

    t = t+deltat

print 'Calculations finished after ',t,'seconds'
```

Ln: 45 Col: 0

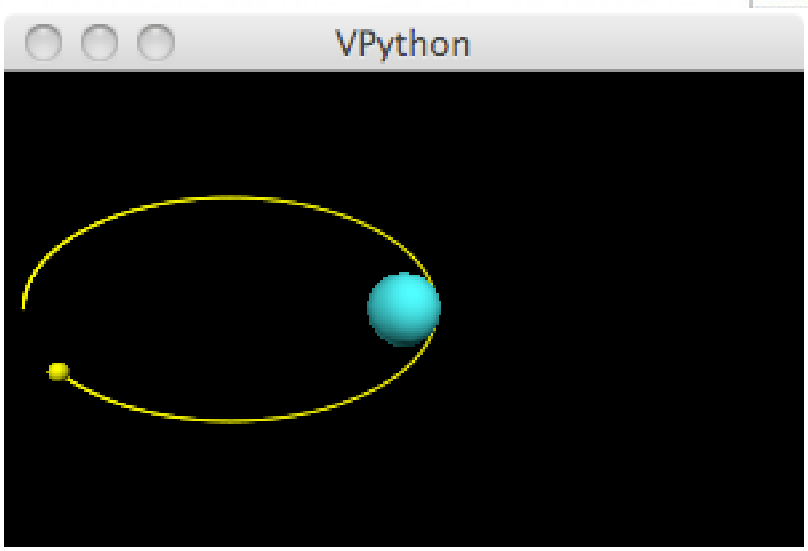


Figure 3.1: Program code and visual output for `Spacecraft.py`

3.4 Results

The original aim of the pilot study was to investigate how the participants modified the program code to achieve a goal state. The tasks were designed to observe how participants modify the initial conditions of the example program and evaluate the change based on the visual output (tasks 2,3,4), as well as adding calculations of forces and principles(tasks 3,4). The analysis of the pilot data revealed more promising avenues of research with regards to how participants completed tasks 1 (a,b,c, and d). That is, the tasks involving the example program and modifications of the example program. The results section reports differences among participants with respect to the completion of the first two tasks.

3.4.1 Task 1a - Study Program

The six participants studied the example program with different degrees of focus and interpretation. The focus of each participant is determined by mapping a participant's verbalization or gesture (with mouse pointer or finger) to a line in the program code. In this section, two participants Brian and Ned are compared with respect to how each participant completes the task of studying the program code.

Brian began the study task with the section of code used to define constants, skipping over the first five lines of code used to import the graphical module, and define the parameters of the window used to display the 3D visual output. The following segment of the transcript corresponds to Brian studying lines 3 - 11 of the program code, shown in Figure 3.2

B: I'm looking at here I see the constants where..

B: gravitational constant [pointing to line, G=]

B: the mass of the spacecraft [pointing to line, mCraft=]

B: mass of the Earth, [pointing to line, mEarth=]

B: are... fifteen times ten to the third,

```
3   #CONSTANTS
4   G = 6.7e-11
5   mCraft = 1.7e22
6   mEarth = 6e24
7   deltat = 10
8
9   #OBJECTS AND INITIAL VALUES
10  Earth = sphere(pos=vector(0.,0.,0.), radius=6.4e6, color=color.cyan)
11  Craft = sphere(pos=vector(-10*Earth.radius,0.,0.), radius=1.75e6,color=color.yellow)
12  trail = curve(color=Craft.color)
13  scene.autoscale = 1
14
15  vCraft = vector(0,1000,0)
16  pCraft = mCraft*vCraft
17  t = 0
```

Figure 3.2: An excerpt of lines 3-17 of the `Spacecraft.py` program

B: six times ten to the twenty-fourth.

B: Lets see a delta t [pointing to line, deltat=], a time step I guess which is ten, an interval of ten.

B: Um, let's see the objects that we've created here are the..there is the earth [pointing to line, Earth=]and its a sphere with this position vector [moving index finger across value]

B: and a radius of the six point four times ten to the six, I guess is in meters

B: and uh [pointing to color attribute] the color is cyan

B: okay, craft is a sphere,

B: position is equal to uh,

B: so its to the left [gestures along -x direction, repeating twice] ten times earth radius.

B: And zero, zero, in the y and z directions.

B: And its radius is[pointing to radius attribute for Craft] one point seven five times ten to the six meters,

B: and it is, I cant see that color, [highlights code at the end of Craft declaration line to reveal value for the color]yellow.

B: Okay.

Brian's use of his finger to point to the line of code he was reading or interpreting is very helpful in concluding the line and segment of code which was the subject of Brian's focus. From this data, its possible to reconstruct the parts of the code Brian was trying to understand. Not only does Brian read the code verbatim ("delta t") he also restates the variable names in terms of his interpretation of the quantities the variable names represent ("mass of Earth", "mass of craft", "this position vector"), he uses the values of vector quantities to determine the location of objects relative to an imaginary coordinate system ("so its to the left [gestures...]"), and he interprets the function of a line of code ("a time step", "the objects that we've created here are...").

Brian's interpretation of the program code continues in a similar manner through to the end of the program. Brian studies the program code starting from the top working his way down, adding meaning to what is there and inventing a world geometry in which the defined objects are located. Much of the program code to this point defines quantities which will be used in calculations defined in the iterative loop, or the attributes of the 3D visual objects plotted in the graphical output. Most of the physics in the example program is in the iterative loop. Here is Brian trying to determine which relative position vector is defined in the calculation loop:

> **B**: Let's see. r is the distance between the craft and the earth, and it is equal to the position of the craft minus the position of the earth.
>
> **B**: So that's a vector pointing towards the earth, I believe.
>
> **B**: Oh wait, no a vector point from the earth towards the craft.

Notice that Brian is now doing more than world geometry or identifying quantities based on variable names. First, Brian identifies the function of the line, then identifies the variables used to determine the calculation of r. Finally, Brian attempts to determine if the calculation represents a relative position vector which points from the craft to the Earth, or the Earth to the craft. To make this distinction, Brian recalls some operational procedure for determining which vector is defined. Unfortunately, there's not enough evidence from the transcript to evaluate how Brian came to this conclusion, either by a heuristic, or the use of the definition of relative position vectors.

It's also useful to analyze these three segments in terms of what Brian did not do. Brian did not make a drawing of the system, which would have been helpful in determining the direction of $\vec{r}$. He didn't calculate the value for $\vec{r}$. Brian, like the rest of the participants, studied the code to interpret the identity of defined quantities and the function of the lines of code.

Ned, on the other hand, took a different approach. Here's the transcript of Ned studying the program code:

N: so, import visual data file,

N: import the visual graphs,

N: we set the screen, set the height

N: define the constants

N: gravity, mass of craft, mass of Earth, and change in time.

N: We create four objects, no three objects, the Craft the Earth and a trail.

N: We set the scene,

N: define the velocity of the craft,

N: define, mathematically, the momentum of the craft.

N: Set time equal to zero.

N: Iterate through the loop to find how fast the rate goes.

N: To find the position of the craft.

N: To find the magnitude of the distance between the craft and the earth.

N: Define a unit vector, rhat.

N: To find the force of gravity, find the net force, find the momentum, finds the crafts final position, add a trail, increased time.

N: And, print calculations after t seconds, cool.

Ned's approach is to primarily determine the function of each line of code. Ned interprets lines of code using the assigned variable name ("gravity"), identifies the 3D objects ("the Craft the Earth and a trail"), and sections of program code ("We create four objects..", "Iterate through the loop"). Ned generally stops interpreting at the equal sign in the program code, except for two instances ("Set time equal to zero", "To find the magnitude of the distance between the craft and the earth.")

So what can one make of this data and the differences between Ned and Brian? Does Ned not find it necessary to interpret the entire line of code? Why is Ned invoking a comprehension strategy to identify program function on such a small grain size? The task asked the participant to study the program code. The action of studying a program is ill-defined and open to interpretation for each participant. To emphasize this point, here's Jason, a participant who made several errors during the later tasks, studying the program code:

J: So I'm looking at the program and I see that there some commented out from future import, from visual import.

J: Um, yeah that stuff confused me.

J: See the constants, objects, and initial values, and the calculations.

J: All looks pretty good.

Jason *only* pays attention to the lines which are commented out (ignored by the compiler). Jason had considerable difficulty extending the program code when asked to do so in tasks 2 and 3. The pilot study task was motivated by a similar task asked during the semester during any computational activity when students receive a shell program (a program without calculations in the loop). The variance among these three individuals in how they study the program suggests further research is needed to determine whether these differences are associated with program comprehension or the nature of the ill-defined task of studying program code.

3.4.2 Task 1b - Predict Visual Output

Three of the six participants (Brian, Ana, and Alex) were asked to draw a prediction of the visual output after studying the program code. The task was described by the interviewer in the following way: "Alright, now I like for you do is to use the whiteboard and draw what you would expect to see when you run this program." It is important to note that the introduction to task 1a (Study Program), the interviewer described the program as "a program that produces an animation of a spacecraft around the earth." This section will first interpret how Brian and Alex complete the prediction task, and finally make comparisons between the two participants.

Brian, having studied the program code in the previous task, returns to the creation of the Earth and Craft as spheres to draw the location of these two objects:

> **B**: Alright so I think this is what I will see I don't think this word right here will appear on VPython program but figure this is the earth and it has a radius, um of whatever we said here.
>
> **B**: Uh, I guess this is the earth dot radius. [labels radius of the Earth]
>
> **B**: Okay and I expect to see the spacecraft initially somewhere out here with a smaller radius, of radius Craft dot radius. [Draws spacecraft to the left of the Earth, labels the radius]
>
> **B**: And it's distance away from Earth is -10 times Earth radius and its zero in the, in the z and y direction so is purely on the x although it doesn't look like it here on this picture here, is purely in a negative x direction
>
> **B**: and then as the objects moves this object has initially a velocity of, lets see. 1000 m/s as it moves I expect to see a trail following it the same color as this [pointing to spacecraft].
>
> **B**: Uh, What else do I expect to see? I expect it to see to follow a curve an orbiting an elliptical shape of some kind so it would look something like that that's kind

of what I would see but all right.

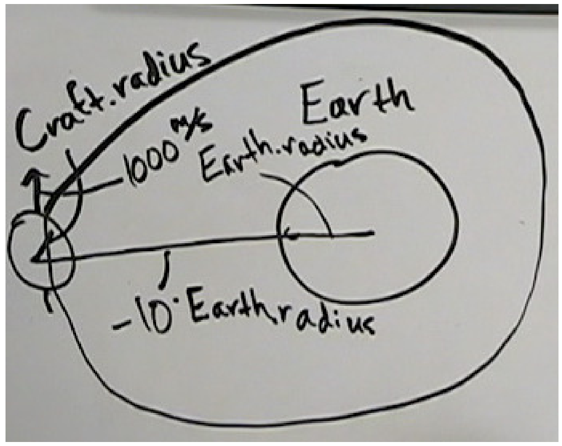


Figure 3.3: Brian's whiteboard prediction.

Brian's whiteboard prediction is shown in Figure 3.3. Brian's prediction focused on the initial conditions of the output such as the location of the spacecraft and the Earth, the values for the size and distance between the two objects, and the initial velocity of the craft (direction and magnitude). The prediction began by using the interpretation of the function of object attributes in the program code, applied to a world geometry that was invented by Brian. His prior experience writing and evaluating VPython programs during the semester was useful in creating this world geometry, which was implicitly defined in his drawing of both the initial location of the spacecraft and the direction of the initial velocity.

There are two ways to interpret Brian's description of the orbit: "I expect it to see to follow a curve an orbiting an elliptical shape of some kind..." Brian's choice of using the word "curve" leaves open the possibility that Brian is interpreting the line:

```
trail=curve(color=Craft.color)
```

There is further evidence in the full study, to be discussed in Chapter 5, that students interpret the 3D object “curve” as a representation of a command for the 3D object, in this case the craft, to follow a curve. There is not enough information in this transcript to determine if Brian was of this mindset, or coincidentally used the same word in his description of a trajectory, uninformed by the name of the 3D object.

The shape of the orbit, however, is not an interpretation from the program code. There are four possible trajectories for all values of an initial speed in the $+y$-direction: escape, circular, and two types of ellipses. Brian only considers an elliptical path, without completing any calculations to check to see if his guess is correct. Brian does not specifically justify the mechanism for the ellipse, the gravitational attraction between the spacecraft and the Earth. But, this justification is not necessary to complete the task.

Alex and Brian’s interpretation of the program code during the study task was very similar, with the exception that Alex did not read through the lines of code used to construct the two 3D objects (lines 12-22 in Figure 3.2). Alex interpreted the lines of code in the `#CONSTANTS` and `#CALCULATIONS` sections. Here’s the transcript of Alex completing the prediction task:

A: Okay lets see, some trailing, so let me see weve got.

A: So youre calculating. Okay, scene width, Earth, position vector, radius okay.

A: Well, Im thinking you’re going to have a.., the Earth is going to be right here [draws circle at center of board]. Label that E [writes “E” inside of circle].

A: And then youre going to have a spacecraft [draws a circle in direction $< -1, 1, 0 >$ relative to Earth] which is going to... go around the Earth [draws a clockwise path, nearly circular and closed, stopping when returning to spacecraft’s position].

A: And I’m assuming, because youre going to have 6.7 e to the -11 is going to slowly move in closer and closer towards the Earth.[gestures with pen starting from craft, moving in straight line towards Earth]

A: So, maybe kind of like that [draws successive orbits, spiraling to form tighter orbits] , at the end of time. (see Figure 3.4)

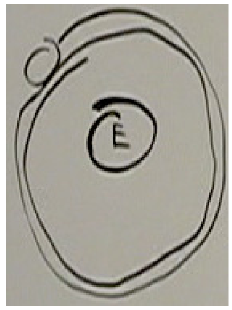


Figure 3.4: Alex's whiteboard prediction.

Since Alex did not read through the section of the program code defining the properties of the two 3D objects, he began the prediction task by returning to the program code. He began searching through the program looking for information, landing on the line used to define the attributes of the Earth. It's not clear what information Alex used in his prediction, as he does not say aloud any of the values used to define the properties of the Earth or spacecraft. The drawing of the spacecraft's location off-axis suggests that Alex was either not careful in drawing the location of the spacecraft such that it lies on the x-axis, or that he simply did not consider the position attribute for the craft. Alex had two opportunities to interpret the craft's position. The evidence from this transcript suggests that Alex interpreted lines 12 and 13 according to its function in creating two 3D objects representing the Earth and the spacecraft, rather than as instructions on how to create the two 3D objects.

Alex's prediction of the dynamics of the spacecraft is quite unique. Again, remember that the instruction to all participants included information that the spacecraft was going to travel around the Earth. Alex's prediction that the spacecraft will go around the Earth does not contradict what he was told; however, Alex added an additional piece, that the orbit is circular. Further, he explicitly entertained what the orbit would look like as time elapsed, suggesting that the spacecraft "is going to slowly move in closer and closer towards the Earth." Alex justifies this prediction by referring to the gravitational constant (which he identified as such during the study task) 6.7e-11. First, notice that Alex justifies the change in motion of an object with a constant, rather than fundamental principles. This error is a major conceptual failure in connecting object dynamics with the momentum principle. Putting the failure in principle-based reasoning aside, notice that there is a second error in how Alex interprets the role of the constant in the computational model. Alex justifies the change in the shape of the spacecraft's orbit with a line of code that comes before the calculation loop rather than inside the loop. The data suggests that Alex's interpretation of the program code, although correct on a line-by-line interpretation, does not extend to a functional view in how the calculation loop updates the position of the spacecraft.

The data shows that the prediction task can reveal appropriate and inappropriate interpretations of the program code and how the interpretation of the code is used to create a prediction of the evolution of the system. With Alex, the prediction task revealed conceptual errors in physics reasoning and the interpretation of the function of the calculation loop. Brian's prediction demonstrated a correct interpretation of the lines of code used to display the spacecraft and Earth, and define the initial conditions of the program. It's not clear, however, that Brian used his interpretation of the calculation loop to predict the dynamics of the system. It is conceivable that Brian either had an expectation of an orbit, or used the information stated in the introduction of the example program to draw his prediction.

3.4.3 Task 1c - Run the Program

All six participants were asked to run the program after studying the program (study group, N=3) or predicting the visual output (study + predict group, N=3). Comparing the two groups of participants reveals noteworthy differences in how the participants responded to the visual output. Each of the participants in the study group described what they saw when viewing the visual output. Ned simply stated that "N: the craft in yellow orbiting the Earth." Joseph commented on the change in speed of the spacecraft, "Joseph: Speeds up when it comes close to the planet." Jason, unlike the other two participants in the study group, searches for an explanation of what he is seeing in the visual output by returning to the code. Jason is bothered by the trajectory of the orbit, as it appears as though the spacecraft is moving through the Earth, as shown in Figure 3.1:

Jason: Alright, I see the spacecraft modeling earth.

Jason: Looks like its almost hitting the earth and then moving out.

Jason: I wonder why that is. Hmm.[Returns to program code, hiding visual output]

Jason: Lets see maybe there is something to do with the um, mass of earth, no.

Jason: Hmm. Looking for whats making it do that elliptical orbit, but can't seem to find it.

Jason: Perhaps it has something to do with one of my constants.

Jason: No, probably from, possibly from its start point or its rate.

Jason: Its going down, Its going much faster. Its going pretty fast.

Jason, as discussed earlier, did not attempt to interpret the lines of code during the study task. Jason commented on the different sections of code in the program (constants, calculations) and indicated he was done studying the program. His return to the program to search for an explanation of the visual output gets him nowhere. The data suggests that Jason is searching for a single line of code to justify what he is seeing. His search for cause is influencing his interpretation of lines of code, such as the rate statement ("It's going down, its going much faster. It's going pretty fast."). Eventually, Jason shifts his interpretation of the program to function:

Jason: Vector, the velocity is , looks like its set there.

Jason: But for some reason its changing as it gets closer,

Jason: so, that must change down here.

Jason: But I dont see it changing.

Jason: Unless its momentum.

Jason: Oh, because its the force thats changing.

Jason: And thats whats causing it to change.

Jason: I dont know, thats kinda weird.

Jason was trying to align his interpretation of the program code at the same time as explaining the visual output. Notice what happens when Jason interprets the initial value for velocity: He describes that the velocity is changing in the visual output, which leads him to look in the calculation loop, makes a connection between velocity and momentum, and finally an erroneous connection between force and momentum. It is promising that Jason, despite his incorrect conclusions, is looking to the program code to uncover justification for the visual output. This is the sort of thinking envisioned by the instructional objectives of the computational activities: to connect events in the visual output to the program code.

The study + predict group participants created a whiteboard prediction of the visual output, so it was possible to compare their prediction to the output. Each of the participants described the output. Ana focused on the initial velocity of the orbit and its shape, "..its going fast enough initially so it doesnt smack into the Earth." Brian described how the speed of the spacecraft was changing, "Um, I notice that, uh, it speeds up right all its speeds up whole lot as a gets closer to the Earth. Initially it starts off with a pretty slow momentum or velocity and it speeds up on its path towards the Earth and the path is elliptical." And Alex found justification for the speed of the spacecraft in his understanding of the gravitational constant, "...so as it approaches the earth is going to speed up because the gravitational constant is going to increase."

Additionally, each participant in the study + predict group commented on how closely their prediction matches the visual output. The participants commented on differences between the prediction and the output and monitored their comprehension of the program code by reflecting on how a faulty understanding of the program code led to an incorrect prediction. For example, Alex said, "I guess I must not of when I try to predict what the what its going to do. I didnt really pay much attention to the radius. Or the original position or it started at." Brian, whose prediction was very close to the visual output said, "..thinking I'm seeing it start off, initially [the visual output] looks a lot like my picture here." And Ana commented on her prediction and her reasoning for why it missed the mark, "It started out like I though but it is in a circular orbit. Probably because, because the position, the distance changes. So, its not like a circular orbit. Its more like this [redraws the prediction to look like the visual output]."

The participants were not asked to evaluate their whiteboard prediction. However, each participant who created a prediction commented on its accuracy and possible reasons for inaccuracy. Alex related the prediction and the visual output back to his understanding of the program code. Ana determined that her prediction's initial state is fine, however the evolution of the system can be justified by considering the possibility that the distance between the spacecraft and the Earth may change. And Brian, simply noted that he was right, which allowed him to focus on other salient features of the visual output which were not salient in his prediction: the speed of the spacecraft during the elliptical orbit. The whiteboard prediction artifact, which represented the participant's understanding of how the program code produced the visual output, served as a tool to check one's understanding quickly by matching features between two graphical representations. The participants spent more time on looking at the differences between the two representations than on describing the visual output or what was correct about the prediction.

3.4.4 Task 2 - Change the program to produce a circular orbit

All participants were asked to change the example program such that the spacecraft moves in a circular orbit, rather than an elliptical orbit. Participants Ana and Brian, who seemed to be proficient by the interviewer at the prior tasks were asked the more open ended question to change *something* in the program to achieve a circular orbit. Participants Joseph, and Alex, who struggled with the study and prediction tasks were asked a more direct question, to change *the initial velocity* to achieve a circular orbit. Jason was asked to complete the more open task, although it would have been wiser to ask Jason to complete the direct task. Ned was not asked to complete this task due to an error by the interviewer.

Ana's solution took an intuitive approach, predicting how a larger value for the initial speed of the craft would affect its trajectory, using the whiteboard drawing of her prediction juxtaposed with the visual output in her explanation, (see Figure 3.5):

> **A**: Okay. My guess would be that it wasnt moving initially fast enough to do a circular orbit?
>
> **A**: Because by the time it got to say here [pointing on whiteboard at], it wasnt it didnt have enough upward momentum to combat the force of gravity so it got pulled inwards, so it was accelerated.
>
> **A**: So maybe change the initial velocity of the craft?

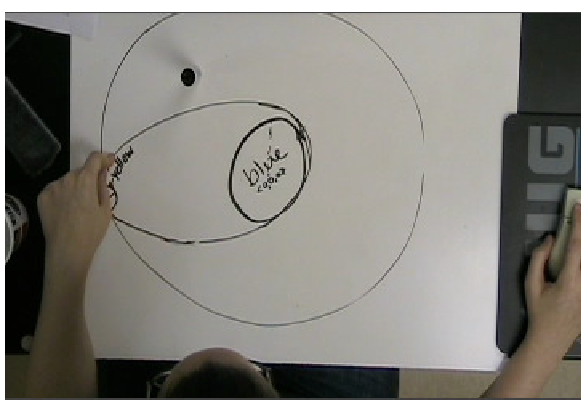


Figure 3.5: Ana describing the effect of the force on the spacecraft with a momentum smaller than the momentum required to maintain a circular orbit.

Ana's justification to change the initial velocity comes from reasoning that the upward momentum must be large enough to prevent the spacecraft from being pulled inward. The reasoning, as described by Ana, is a combative play between momentum and force. Joseph, a participant who was asked to change the velocity, followed a very similar line of reasoning:

Joseph: So it needs to be equal to hm.

Joseph: Having a hard time visualizing what to change.

Joseph: How the velocity affects the make it higher it goes more circular.

Joseph: Cause, markers dying but.

Joseph: Earth [redraws Earth] spaceship [redraws craft].

Joseph: With a little bit of velocity it goes like that [draws visual output]

Joseph: With more it should go around.[draws a circular orbit]

Joseph: But, not sure what it must be equal to.

Joseph: It needs to be pulled this way [draws arrow to the right] and pulled that way [draws arrow up] at the same speed.

Joseph: So force net has to be the same as that.

Joseph: I'm not sure if velocity is equal to force.

Joseph: Momentum principle is...[looks to equation sheet provided] I never liked this sheet, too much stuff on it..

Joseph: Oh right cause the momentum principle is not on it. [moves sheet to the side]

Joseph: force kinda equals mass times[writes F = m].. momentum equals mass times velocity [changes p to F, adds v], approximately.

Joseph: But the pull is force, not momentum.

Joseph: Fnet times change in time [writes $p = F_{net}\Delta t$]

Joseph began down a path describing a goal to determine the velocity such that the outward pull matches an inward pull. Joseph questioned the suggestion that velocity is equal to force, and turns towards the definition of momentum. Next, Joseph wrote what he believed was the Momentum Principle after deciding that pull is determined by force, not momentum. Notice that Joseph said that pull is a force, not momentum, but then immediately equated the two with the incorrect form of the Momentum Principle. Joseph makes a transition from an intuitive based reasoning to principle based reasoning, although the principle is incorrect.

Moving on to Brian, who described the goal state as one where the relative position vector changes but its magnitude does not. His intuition suggests an action to perform:

> **B**: Well for sure I know for sure the velocity is going to have to increase.
>
> **B**: Am I okay to run a program just to try it out?
>
> **I**: Whatever you want to do.
>
> **B**: Lets try this out. If we just try to increase the velocity lets see what that does.
>
> **B**: [changes 1 to 2 in, vCraft = vector(0, 2000, 0)]
>
> **B**: Its closer to a circle than the last run was. So why is that? Why does that work?

Brian did not find a satisfactory answer to his question. Brian continues along this line of questioning, eventually completing the task while searching for an explanation. Brian eventually turned to principle-based reasoning but has difficulty connecting the iteration of the Momentum Principle to the special initial velocity which produced a circular orbit. Brian understood that the relative position vector and its unit vector changed during the course of a circular orbit, however the magnitude of the relative position vector did not change. He could not find the connection between how a certain value of the initial velocity had this effect. Here we see an episode where a participant was asking questions that one would not expect to hear asked while solving the two-body central force problem for uniform acceleration. Changing the initial condition, by a small percentage of the original value, produces results which are visibly different in the visual output. The reasoning that Brian was searching for begins by considering how the net force changes the momentum of an object with different initial speeds. If Brian completed this exercise, he would have seen that at a very special speed, the change in momentum of the spacecraft only changes the direction of the initial momentum, without affecting the speed of the spacecraft.

3.5 Discussion

Notice that the participants have written zero lines of code in the three tasks described thus far in the pilot study. The study task revealed that all students can interpret program code with respect to the code's function, and the physics quantity represented by variables in the lines of code. It would be incorrect to conclude that the failure of some of the participants to dig deeper into the program's meaning is some indication of the ability of those participants to comprehend programs. It is more likely that the incomplete performance of students during the task is due to the interpretation of the study task by the individual. Participants who studied the deep structure of the program code recognized representations of physics quantities and principles, and identified the purpose of physics equations in creating the visual output.

Participants who were asked to create a whiteboard prediction of the visual output returned to the code to use the attributes of the 3D objects to create the geometry of the virtual world, position the 3D objects within that world, and compare objects with each other (Brian and Ana). Ana and Brian drew a vector representing the initial velocity of the spacecraft, and predicted the motion of that spacecraft through the space. Neither Brian or Ana justified the trajectory with some reference to the interaction between the objects. Alex only drew the trajectory of the spacecraft; however, he elaborated on his justification for his unique choice of a degrading orbit. Alex's reasoning referred to the definition of a constant, rather than the use of a fundamental principle.

Drawing the prediction changed how the participants interpreted the visual output. Those who drew the prediction evaluated one's success, and further discussed the differences between the prediction and output. Brian, the only prediction participant who drew an elliptical orbit, did not focus on how the speed of the craft will change along the trajectory in his prediction. Brian did note this feature of the dynamics during his evaluation of the prediction, and further justified the reason for the change in speed using fundamental principles.

Finally, asking participants to make a minor change to the program code with the think-aloud protocol shows how powerful this task is in revealing how students apply their physics knowledge in this programming environment. Ana relied on an intuitive justification for why increasing the initial speed would work. Joseph, who must have ignored the direction to change the velocity, followed a similar reasoning path to determine what to change in the program code, but later shifted to principle-based reasoning. And Brian, who figured out the puzzle that he needed to change the initial speed, was unsatisfied with completing the task without understanding why this change would produce a different trajectory. Brian spent a significant amount of time searching through the program code and his prior physics knowledge for an explanation.

3.5.1 Link to Reading Comprehension

The initial focus of the pilot was to investigate how students *modify* example and shell programs. However, the actions of the participants prompted a reassessment of the following assumption: students who have created VPython programs can explain the workings of similar VPython programs. The data from participants completing the sequence of tasks in the pilot shows that there are errors in the comprehension of a program that was previously created by students in a classroom setting. Therefore there is evidence that the assumption should be examined closer to understand the types of errors students were making and how this might affect the battery of instructional materials used in computational activities using VPython. The errors have been traced back to the participants' prior physics knowledge, the interpretation of the function of lines and sections of program code, and a failure to monitor one's understanding while completing these tasks. These errors can be correlated to the six functions necessary for successful comprehension of reading passages identified by Palinscar and Brown (1984), as discussed in Chapter 2. Further, the tasks the participants completed are closely related to the tasks Palinscar and Brown settled on as those which promote the development of these six functions. Table 3.2 compiles the actions of the pilot study participants according to the task they were asked to complete.

Table 3.2: Student Actions taken with each pilot study task

Task	**Function**
Study	Activate relevant background knowledge in physics and programming
	Interpret function and purpose of code
	Monitor understanding of program code
Predict	Use interpretation of code as evidence to identify the presence and behavior of 3D object
Run	Evaluate understanding of program code
	Connect the interpretation of the display to the code
Modify	Activate relevant background knowledge in physics and programming
	Allocate attention on the calculations used to create the visual output
	Draw and test inferences (implicit or explicit) on which code to modify
	Monitor comprehension using the effects of code modifications on the visual output

For example, in order to interpret the program function or the representation of physics quantities, the participant must call on prior knowledge in physics and programming. A participant who studied each line and determined its function in the program code must monitor one's understanding of the purpose of each line of code by some check method for understanding. Evidence that the participants are following through on these actions have been discussed in section 3.4.

This data reveals some differences between reading comprehension and the comprehension of the example program. The modification task is perhaps the most clear place to start, since modifying the reading passage is not considered by Palinscar and Brown as a task which fosters reading comprehension. And nor should it, since modifying a reading passage requires additional cognitive skills beyond those identified as relevant to reading comprehension. However, the task of modifying an initial condition's value to change the dynamics of a 3D object is a relatively small change that has the potential to have a drastic effect on the outcome of the visual output. Students struggling to understand why this is so is another form of comprehension involving the understanding of the program algorithm and physics principles. At this point, Ana and Joseph reveal an incomplete understanding of the relationship between change in momentum and net force.

A second point of divergence from Palinscar and Brown is the prediction task. In reading comprehension, the prediction task is used to elicit the possible future content from inferences drawn from the passages read. The prediction task of the pilot study asked participants to determine the result of the visual output. Therefore the whiteboard prediction is an artifact of the participants interpretation of the program code in a graphical representational space. There are other choices for elaborating and storing a prediction of the visual output, such as a verbal or written representation; however, it would be more difficult to capture the evaluation of the participant's prediction at program runtime. Further, it would not be clear that the participant is evaluating the previously stated prediction, or some other unknown internal representation of the visual output. The 2D whiteboard drawing of a predicted visual output provides a quick and honest evaluation between what one sees in the visual output and what one drew on the whiteboard.

Finally, while running the program and watching the visual output, the participant interprets the animation in context of the program code or some other prior knowledge. All the participants commented spontaneously on the speed of the spacecraft increasing as it moved closer to the Earth as a valid behavior of the spacecraft. Earlier, Jason was disturbed by the trajectory of the spacecraft and returned to the program code to search for a justification. Alex justified the speed of the spacecraft by referring to the gravitational constant. Other participants are not clear as to why they buy into the validity of the simulation. There's not enough information here to determine if the visual output meets some expectation based on prior experience with the space voyage task or a recognition that the program animation represents the physical behavior of the system in the real-world.

Differences exist between reading comprehension and program comprehension, although it appears as though the differences can be attributed to the context rather than cognitive processes which occur in comprehension. This data shows the tasks associated with program comprehension reveals errors in participants physics knowledge when they try to apply it to explain the motion of interacting objects within a system. The predictions are well informed by the program code, up to the actual description of the trajectory, where it is not explicitly clear the thinking behind the choice of motion. Is this motion determined by attention to the fundamental principles in the program code, or some other reasoning which is not apparent from this data? The next chapter describes a new type of program which will be deployed to determine how students are using physics to create predictions of the visual output which model systems that lack a complete description of the interactions between objects.

Chapter 4

Methodology

This chapter describes the formal study in detail, as well as the research methods used to interpret the data generated from the formal study so that the research questions can be addressed. This chapter begins with a discussion of the development of a new instructional tool, the Minimal Working Program (MWP), the empirical results of its implementation in a trial lab section, the deployment of the MWP in a research lab section, and the research methods used to analyze the data generated as students complete activities starting from the MWP.

4.1 Motivating a new instructional tool

Participants in the pilot study completed an entire course which involved students working through computational assignments with VPython. The data from the pilot is a collection of six different approaches to the tasks by participants who completed an entire semester of computational activities. After analyzing the pilot study, it is clear that asking students to read and discuss a program and predict what the visual output will show fosters the sense making actions of interpreting the program code using one's prior physics and computational knowledge. However, the pilot data does not provide convincing evidence that students are

```
from __future__ import division
from visual import *
scene.width =800
scene.height = 800

#CONSTANTS
G = 6.7e-11
mEarth = 6e24
mcraft = 15e3
deltat = 60

#OBJECTS AND INITIAL VALUES
Earth = sphere(pos=vector(0,0,0), radius=6.4e6, color=color.cyan)
craft = sphere(pos=vector(-10*Earth.radius, 0,0), radius=1e6, color=color.yellow)
vcraft = vector(0,1200,0)
pcraft = mcraft*vcraft
trail = curve(color=craft.color)    ## produces a trail behind the craft

t = 0

#CALCULATION LOOP: ALL REPEATED CALCULATIONS GO INSIDE THE LOOP
while t < 10*365*24*60*60:
    rate(100)                 ## slow down the number of loop calculations per second
    craft.pos = craft.pos + (pcraft/mcraft)*deltat
    trail.append(pos=craft.pos) ## this adds the new position of the spacecraft to the trail
    t = t+deltat
```

Figure 4.1: The MWP code for the Earth-Spacecraft system.

using their interpretation of the physics in the calculation loop to inform their predictions of the system's dynamics. Would the participants in the pilot study have interpreted the calculation loop and commented on the lack of interactions or lack of physics principles in the program? Would this omission inform their prediction of the visual output? It's difficult to say one way or the other. One of the major instructional goals of including computational activities in the Matter & Interactions curriculum is to help students understand the dynamic nature of the momentum principle (Chabay & Sherwood, 2008). Since the participants were using their physics knowledge to identify the momentum principle in the program when studying the code, part of the objective is met. A solution is required to focus students on the dynamical nature of the momentum principle in addition to the identification of the principle. That is, it's not clear that the participants in the pilot study who correctly predicted the motion of the spacecraft used an operational definition of the momentum principle to inform their prediction.

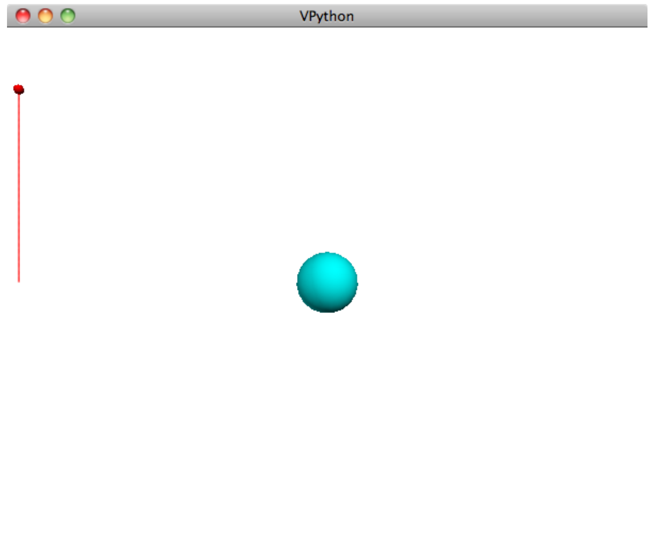


Figure 4.2: The visual output for the Earth-Spacecraft MWP

To investigate how students use the Momentum Principle in drawing whiteboard predictions of the visual output, we took the example program and removed the lines of code used to update the momentum of the craft and the lines of code used to calculate the gravitational force. Figure 4.1 shows the program with these omissions. What's left is a program which, just as before, will produce a visual output and an animation. However, since the program does not update the momentum of the spacecraft, the spacecraft moves in a straight path at a constant speed as shown in Figure 4.2. The animation does not predict the time evolution of an analogous physical system consisting of a spacecraft and an Earth. This program creates a representation of the physical system in a 3D virtual world, defines the physical properties of these objects (mass, radius, initial values), and animates the objects using only the initial value of the object's momentum. Since the computational model is an incomplete representation of a physical system, we're calling it a Minimally Working Program, or MWP. Again, the program will work in the sense that the program will display a time-varying animation. The program is minimally working in the sense that the time-varying animation is not representative of all or any of the interactions which would exist between the objects in a physical system.

4.1.1 A Minimally Working Program

Three MWPs were developed for use by students while completing computational activities in lab. The differences between the three MWPs go beyond the modeling of different physical systems on varying time scales (time steps) or spatial dimensions. Table 4.1 details the differences between the three physical systems and the features of the computational model in the MWP. And Figure 4.3 displays the MWP program code, a capture of the visual output to give a sense of the time-varying simulation, and the code students are expected to generate such that the visual output predicts the time-varying behavior of the physical systems the program intends to model.

Table 4.1: Comparison of the physical system to the features of the computational model which are included and omitted from the MWP.

Physical System	Computational Model omits:	Computational Model includes:
Spacecraft interacts with Earth	Gravitational force acting on spacecraft	Update of the spacecraft position using a constant velocity
Ball attached to an oscillating vertical spring	The spring interaction	Gravitational force acting on the ball due to the Earth; Momentum principle; Update of the ball's position from new momentum values
Scattering of an alpha particle off a gold nucleus	Electrostatic force acting on the alpha particle due to the gold nucleus; Electrostatic force acting on the gold nucleus due to the alpha particle; Momentum principle for each particle; Update of the gold nucleus position	Update of the alpha particle position using a constant velocity

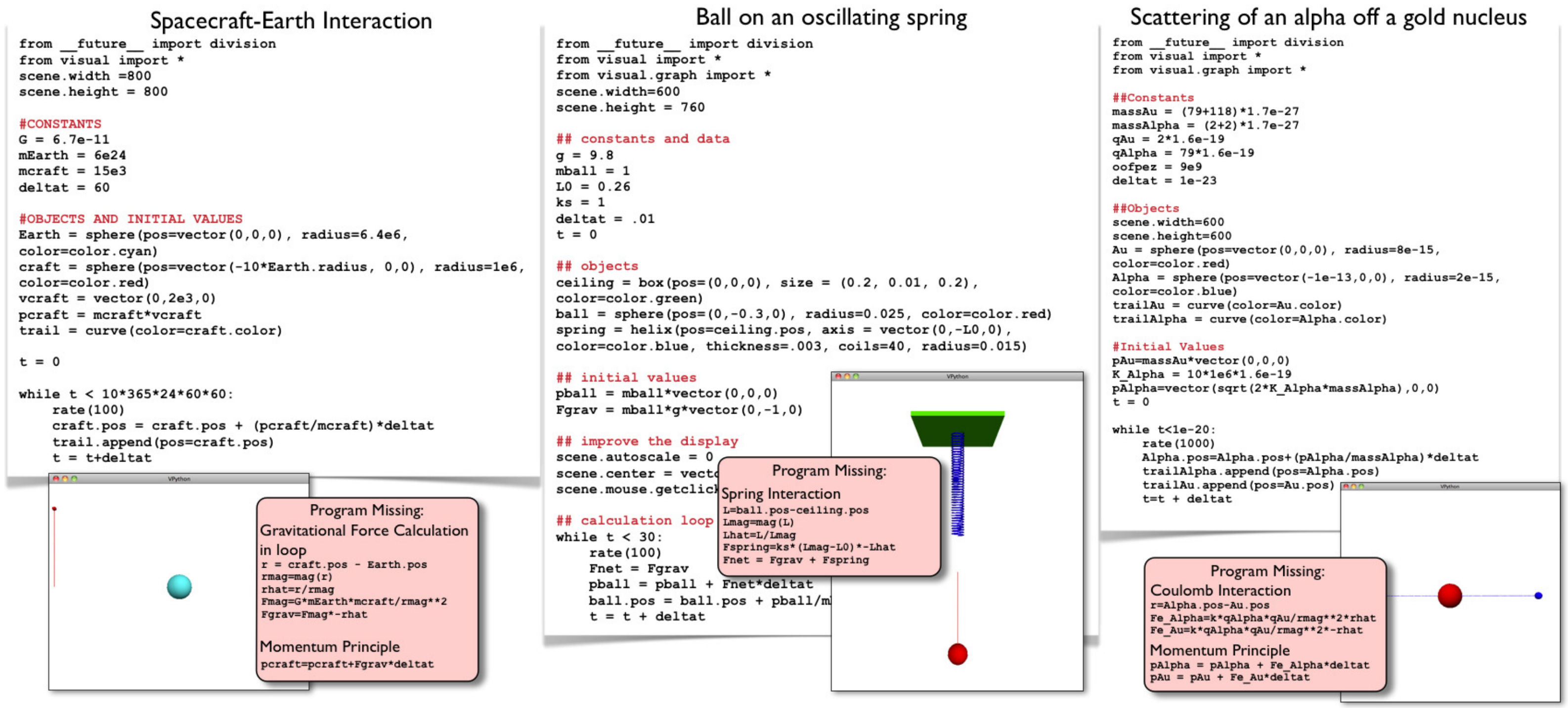


Figure 4.3: The program code and visual output for the three MWPs developed to be included in computational activities. The pink box lists the new lines of code students need to generate and add to the program such that the MWP predicts an appropriate behavior of the corresponding physical system.

The features of the Spacecraft-Earth MWP was discussed in the last section. The Vertical Spring MWP omits the lines of code used to give the helix object drawn in the visual output its macroscopic behavior characterized by Hooke's Law,

$$\vec{F}_{spring} = -k_s(|\vec{L}| - L_0)\hat{L}$$

where $\vec{L}$ is defined as the relative position vector pointing from the fixed end of the spring to the free end of the spring, $\hat{L}$ is the unit vector pointing along $\vec{L}$, L_0 is the length of the spring without any attached mass to its free end, and k_s is the spring constant. The spring is modeled to be *massless*. For simplification, students are allowed to attach the spring to the center of the ball, as opposed to the surface of the ball.

The Vertical Spring MWP is different from the Spacecraft-Earth MWP such that the Vertical Spring includes a calculation of the gravitational force acting on the ball due to the Earth, and uses this calculation in the computational model to update the momentum of the ball. With the update of the ball's position, the visual output will display a ball which falls from rest, accelerating due to its interaction with the Earth. The initial position of the ball is below the end of the spring, so the ball does not have the appearance of being attached to the end of the spring and the free end of the spring remains at the same location during the entire length of the simulation.

The Scattering MWP omits the lines of code used to calculate the Electrostatic interaction between the two charged nuclei, the relative position vector used to calculate the direction and magnitude of this interaction, the updated momentum values for the nuclei due to the interaction over a short time interval, and the updated position of the gold nucleus due to the interaction. In the Spacecraft-Earth system, students didn't consider the effect of the pairwise interaction on the motion of the Earth; however, in the Scattering MWP, students are expected to generate code which calculates and displays the motion of the gold nucleus.

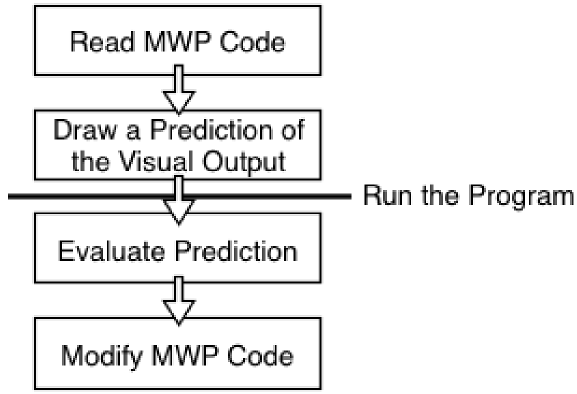


Figure 4.4: Instructional tasks listed in sequence using a MWP.

4.1.2 Instructional tasks using the MWP

Asking students to read and comprehend a program in the pilot study revealed differences among students in how they apply their physics knowledge to understand how the program uses the code to create the visual output. Following the instructional model of Palinscar & Brown (1984), we developed tasks which first focuses students' attention on the existing MWP code. Figure 4.4 shows the sequence of instructional tasks and student actions involving the MWP. Students are asked to read the existing code for understanding and use their understanding to generate a prediction of what students will see in the visual output once the program runs. Students run the program and use the displayed visual output to evaluate their prediction. Next, students are asked to modify the MWP such that it calculates the missing interactions and/or physics principles and predicts the behavior of the physical system as observed in the natural world.

Additional instructional tasks are added after students generate new code to create a complete and error-free program. These additional tasks vary according to the system being modeled in the program. For example, in the Spacecraft-Earth system, students are asked to change the initial speed of the spacecraft to predict different types of orbits (closed-circular orbits or open orbits) and investigate how increasing the time-step of the iteration loop affects the accuracy of the predicted orbit. In the Alpha-Gold system, students are asked to increase the impact parameter of the alpha particle and plot the momentum of the system as a function of time.

4.1.3 Empirical observations of a Spacecraft-Earth MWP test-run

During the Spring 2008 semester we developed the Spacecraft-Earth MWP and the instructional tasks described above. I was assigned to serve as a teaching assistant for two sections of the lab course paired with the Matter & Interactions lecture course. The activity for each week is chosen by the lecture/lab course coordinator, in consultation with the lecture instructors for the course. After receiving approval from the course coordinator, I switched the traditional Space Voyage activity with the activity which uses the modified instructional task sequence described in Section 4.3 and the Spacecraft-Earth MWP. The Appendix contains the instructional documents for both the traditional and modified activity. The intent was to monitor how students completed the instructional tasks and provide an opportunity for the teaching assistant to have conversations with students to probe how they interpret the MWP and modify the MWP code to achieve the complete and valid model of the Spacecraft-Earth system.

These observations were recorded by the TA, from memory, after the end of a lab section. Since the teaching assistant was involved with duties in addition to observing how students completed the modified instructional activity, the reporting only captures the salient events relating to the perceived success or failure of the activity. With these confounds understood, the reporting of the student activity while engaged in the instructional tasks is recognized as incomplete and piecemeal. However, this data is useful as a zeroth order evaluation of the instructional activity in noticing which tasks were confusing, skipped, or too difficult to complete without substantial intervention by the teaching assistant.

The modified instructional activity with the MWP took approximately the same amount of time as the traditional Space Voyage activity. Students starting from the MWP did not have to write as many lines of code, but they were asked to spend some additional time explicitly reading and predicting the visual output of the MWP. Therefore any additional time spent on new tasks were offset by the reduced time spent on debugging erroneous lines of code. Student predictions of the MWP were a mix of both systems where the spacecraft interacted with the Earth and the spacecraft traveling in a straight line. All of the student groups created predictions. A minority of groups were confused about the task sequence and why they were asked to read the MWP program code.

After generating the missing gravitational force calculation and interactions, the teaching assistant asked some groups to explain why changing the initial velocity has an effect on the orbital trajectory. One group's discussion was especially memorable, as students were using their knowledge about the rate of change of the spacecraft's momentum in context of the iterative loop and the spacecraft interaction with the Earth to explain the differences observed in the visual output on the trajectory of the spacecraft. That is, the students were using the computational model along with their physics knowledge to explain the program's output. After the group discussion ended, the group was asked to complete a written explanation to the same question asked by the teaching assistant. It was expected that the written explanation would match the discussion involving the teaching assistant; however, the written discussion contained very few of the conceptual elements which were part of the verbal discussion involving the teaching assistant. This group's written discussion consisted of an analysis void of references to the computational model. Instead, the group tracked the energy of the spacecraft-Earth system, commenting on the exchange between the kinetic energy of the spacecraft and the gravitational potential energy in the system as the spacecraft followed an elliptical path. This shift from using computational thinking to an analysis based on energetics was unexpected.

Capturing the written explanations and drawing conclusions about the types of discussions students are having based on the written response would have been an inaccurate representation of what occurred. To capture the discussions of students as they are completing these tasks, one has to video record the discussion. It's not clear how much of the teaching assistant guided discussion was controlled by the teaching assistant leading and prompting the students to consider how the computational model produces the visual output. If the teaching assistant intervened considerably, then the students may not have been able to recreate the reasoning based on the computational model when formulating a written response to the prompt. It was decided upon reviewing the empirical data that a more rigorous, student-centered investigation was needed to track teaching assistant interventions and the reasoning chains used by students when completing instructional tasks.

4.2 Design of the Experimental Study

Observations of students completing the modified instructional tasks using a MWP revealed that the modifications didn't add to the amount of time required for completion or outright fail as an instructional activity. To answer the research questions examining how students use their physics knowledge to comprehend the computational model in an incomplete but functional program, we needed to analyze audiovisual recordings of students working on the activities. Use of the word "Participants" refers to those students who were part of the general student population who were recruited for this study, who volunteered to participate, and who were ultimately video recorded while working on the instructional tasks.

4.2.1 Qualitative Education Research Lab Environment

In 2008, The Science Technology Engineering and Math Initiative at North Carolina State University funded the purchase and installation of audiovisual equipment to record, compile, and archive data of participants working in a space large enough to gather four groups of students at a time. Figure 4.5 shows a rendering of the largest of three rooms in the lab

Figure 4.5: Computer model of the QERL lab environment. The lab furniture includes four tables, each accommodating three participants per table, a workstation and a whiteboard.

space, depicting four mobile square tables with tabletop space for a computer workstation and a portable whiteboard. Each table has enough room for three participants to sit comfortably, in rolling chairs, which allows students to gather around the computer screen or whiteboard. There is enough room in the aisle for the teaching assistant to move through the lab space, and a rolling whiteboard for use by the teaching assistant to aide in communicating with the participants.

The room modeled in Figure 4.5 is equipped with two-way mirrors along one interior wall, permitting an observer on the other side of the partition to watch participants completing the activities. This observation room houses the audiovisual controllers as well as the hard drives used to record and store all audiovisual data. While in the observation control center, a single researcher can monitor the field-of-view of each camera, the audio feed from each microphone, and control and merge all incoming audiovisual feeds to send to hard drives for recording.

4.2.2 Guiding Principles for the Design of the Study

Utilizing the tools provided to capture audiovisual recordings of students working in Qualitative Education Research Lab at North Carolina State University, the design of the larger experimental study was motivated by the following design goals:

- The results from an analysis are to be valid in a controlled environment as well as the lab sections.
- Participants work in cooperative learning groups while engaged in the instructional activities.
- Participating students receive the same instruction as those who choose not to participate.
- Students deciding to volunteer consider a balanced risk/benefit ratio for participation in the research study.
- Participant identities are protected and anonymity is preserved.
- Reduce the personnel resources required for capturing and storing audiovisual data.
- The experimental environment matches as closely as possible to the lab course environment.
- Audiovisual devices capture all actions, discussions, computer inputs and outputs, and are unobtrusive.
- A flexible study design can accommodate on-the-fly alterations to modified instructional activities based on how participants respond.
- Absent participants or technical glitches with equipment do not prevent the instructional activity from proceeding forward.

- Teaching assistant running the experimental section has the freedom to make managerial decisions based on student progress and unforeseen complications.
- The course coordinator has overriding control of the experimental lab calendar and oversees the student progress in the experimental lab sections and traditional lab sections.
- Participants are provided the opportunity to transfer to a traditional lab section at any point in the semester if the participant decided to withdraw consent.

4.2.3 Participants

During the 2008 spring semester, two experimental lab sections of the Matter & Interaction: Modern Mechanics course were added as available lab sections for Fall 2009. These experimental lab sections were limited to 12 students per section, with the location assigned to the QERL space described in section 4.2.1. Participants for these two experimental lab sections were not recruited. Students taking the Matter & Interactions: Modern Mechanics course were required to register for one of 18 available lab sections. There were no indications at the time of registration as to why the enrollment cap was set to 12 for two of the available lab sections, or as to why two lab sections were assigned to a different location from the 16 traditional lab sections. When a student registered for an experimental lab section, the student received an email notifying the special status of the lab section with a copy of a consent form detailing the benefits and risks of participating in the research study (**See the Appendix for these documents**). Registered students were asked to switch to one of the 16 available traditional lab sections if they didn't want to volunteer to participate in the lab section. This procedure continued through spring registration and the summer semester, until the beginning of the first day of lab. On the first day, students were presented with a consent form and asked to sign, if willing to participate. One of the conditions for participation is consenting to allow researchers within the Physics Education Research and Development group at NC State to use videos containing their likeness at presentations and publications such as this one.

Table 4.2: Demographics data of participant's gender for the experimental lab sections.

Gender	Section 240	Section 241	Total
Female	8	6	14
Male	4	5	9

The gender demographics of the participants are not entirely reflective of the student population taking the Matter & Interaction: Modern Mechanics course. Table 5.1 shows the number of males and females in each experimental lab section. The female/male ratio is in line with the broader student population at NC State.

Participants were evaluated based on their performance in the experimental lab, consistent with the evaluation of their peers in traditional lab sections. Students did not receive a separate grade for lab and lecture, rather scores on all lab activities were used in the calculation of the student's course grade. Participants were not paid for their participation since the work the participants complete is factored into the performance based score for the course.

4.2.4 Integrating the MWP activities into the lab calendar

The lab sections met weekly, for 110 minutes, to complete activities which provided practice on problem solving, analyzing data using physics principles, or writing computer programs that reflect the topics discussed in the previous week's lectures or homework assignments. Typically, two lab activities were completed during the lab. Participants were free to leave the lab after the activities were completed and approved by the TA. Table 4.3 lists the schedule of lab activities for each weekly lab meeting for both the experimental and traditional lab sections. Any activities in the experimental lab section where the instructional documents are significantly modified from the traditional lab section are indicated in the table. Not all modifications were informed by the pilot study or part of the data set intended to address the current research questions. For example, the Intro to VPython document included instructions

Table 4.3: Lab activity schedule for experimental and traditional lab sections during the Fall 2009 semester.

Lab Week	Experimental Lab	Traditional Lab
1	Group Roles Intro to VPython*	Group Roles Intro to VPython
2	Fancart Experiment	Fancart Experiment
3	Impulse Experiment Fancart Motion*	Impulse Experiment Fancart Motion
4	Macroscopic Springs Gravitational Force VPython*	Macroscopic Springs Gravitational Force VPython
5	Young's Modulus Experiment Spacecraft-Earth MWP	Young's Modulus Experiment Spacecraft-Earth VPython
6	Spacecraft-Moon Interaction Projectile Motion Lab	Spacecraft-Moon Interaction Projectile Motion Lab
7	Adding Energy Graphs in VPython Fission Whiteboard Problem	Adding Energy Graphs in VPython Fission Whiteboard Problem
8	Spring-Mass MWP Spectra Lab	Air Resistance Experiment Spectra Lab
9	Jump-Up Lab Multiparticle Whiteboard Problem	Jump-Up Lab Multiparticle Whiteboard Problem
10	Rutherford Scattering MWP	Spring-Mass MWP
11	Angular Momentum Lab	Angular Momentum Lab
12	Statistical Mechanics VPython	Statistical Mechanics VPython

* - Instructional document is significantly modified to include directions for watching YouTube videos introducing new programming concepts.

to view YouTube Videos introducing the visual module and presented a challenge for students to complete. The traditional Intro to VPython document provided written explanations of lines of code and directed students to type in the program what was needed to complete the activity.

Due to time constraints, the course coordinator initially decided to skip the traditional Spring-Mass activity for the traditional lab sections during Week 8. In previous offerings of the course, the academic calendar provided additional weeks for lab meetings so it was possible to complete the Spring-Mass VPython activity in addition to the Air Resistance activity. For the Fall 2008 term, the schedule only allowed for one of these activities to occur. The course coordinator decided that during Lab Week 8, the traditional lab sections would work on an activity analyzing air resistance rather than the traditional spring-mass VPython activity. He agreed to allowing the experimental lab to work on the Spring-Mass MWP activity. The course coordinator is responsible for writing test questions which may draw from students' experience completing lab activities. As a compromise, the course coordinator agreed not to write a test question which may put participants in the research study at a disadvantage for not having completed the Air Resistance activity.

Subsequently, the course coordinator was forced to make a mid-course correction to the lab schedule due to the pacing of the lecture courses. The Angular Momentum lab was originally scheduled for Week 10, but due to the pacing of the lecture course, the lab was moved to Week 11. As a result, a hole appeared and a new MWP was developed for the experimental lab. The course coordinator decided to use the Spring-Mass MWP activity for the traditional lab. The MWP program given to the traditional lab students in Week 10 was modified from its original version as assigned in Week 8 to the experimental lab to take advantage of what was learned from its implementation. The Week 10 Spring-Mass MWP activity is included in the Appendix.

Like the traditional lab activities, all modified activities ask students to form collaborative groups. In a collaborative group, each team member is responsible for implementing the duties which define their assigned role to ensure that important cognitive tasks required for successful problem solving are addressed.(Heller & Hollabaugh, 1992) These roles (Manager, Recorder, Skeptic, Summarizer) were modeled and practiced during the first lab and were used for each lab activity. The teaching assistant reminded students to switch roles from week to week; however, the participants decided which role each team member played. Collaborative group membership was assigned by the teaching assistant and participants were assigned to different groups every three weeks. The Appendix contains diagrams of the location of each participant in the classroom by table/group number for each experimental lab section for weeks 5, 8, & 10 during which participants were asked to complete the MWP activities. Notice that the participants in a group changed for each one of the MWP activities. For the purposes of this study, this variance in group membership may confound conclusions drawn from an analysis of how different groups complete the programs over the course of the semester. This possible confound in the study design will be address further in Chapter 6.

4.2.5 Texture of the Data

A tabletop conference microphone was placed at each table to capture all conversations in the immediate vicinity. A webcam was mounted on a vertical pole attached to the table, positioned behind the monitor facing towards the participants to capture body language, gestures, teaching assistant interactions with each group of students, and a limited omnidirectional sound field (Figure 4.6(a)). A ceiling-mounted camera directly over the tabletop whiteboard with pan-tilt-zoom functionality recorded video of participants pointing on the board, to the screen, and the writings or drawings on whiteboards (Figure 4.6(b)). Screen-capture software on each workstation recorded the monitor output and the audio captured by the webcam (Figure 4.6(c)). These three audiovisual feeds constitute the raw data available for further analysis. The data set consists of three video feeds per student group per MWP activity (weeks 5, 8, & 10).

After the end of the semester, the three video feeds recorded per participant group were combined into one video file, making use of cuts between video feeds based on the visual information presented on a video frame. Figure 4.7 shows an example of one of the video feeds

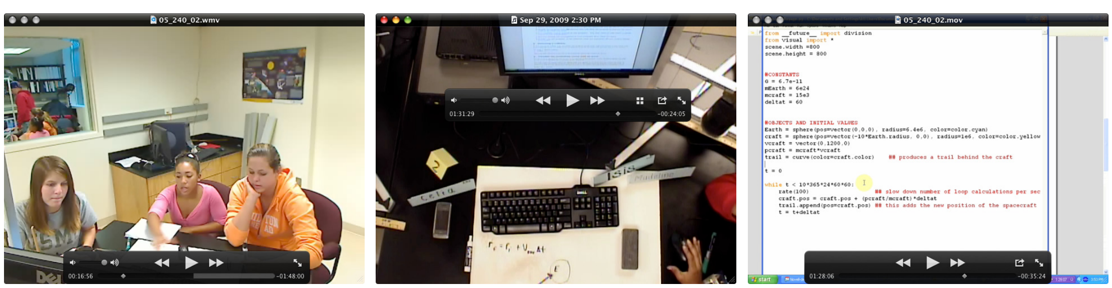


Figure 4.6: Screenshots of the video captured by the (a) webcam, (b) over the head, pan-tilt-zoom camera, and (c) the software used to record the monitor output.

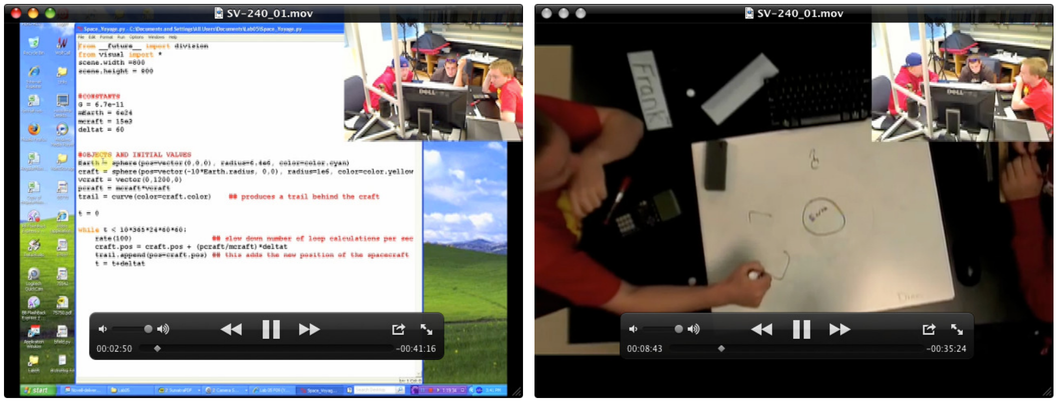


Figure 4.7: Screenshot of two frames from the same video file for a group working on the Spacecraft-Earth MWP activity. The webcam video feed is inlayed into an alternating feed between the over-the-head video and the recording of the monitor output.

from the Spacecraft-Earth MWP activity. Each video file was given a unique identifier and transferred to a secure hard drive for storage until analysis could begin.

4.3 Methodology for Data Analysis

During analysis, the primary task is to identify what participants are doing at any one moment in time and gather evidence of how participants use the program code to inform an understanding, a prerequisite for students completing the tasks. Comparing what participants say about the program to the literal substance of the program code provides data of what extra information the participants must provide to develop an understanding of what's before them. The primary challenge is to identify which of these statements correspond to physics knowledge and compare differences among participant groups and among MWP activities. Searching for these statements assumes that participants are actively engaged in the task and are interpreting the program code for comprehension. Each pass of the video data focused on extracting different types of information from a participant's actions or speech. For example, a single pass may identify the instructional task participants were completing. A subsequent pass may only focus on identifying the context of off-task discussions. Before this analysis can occur, one must extract information from the video data.

Due to the rich nature of the video data, it was necessary to transcribe all videos before proceeding with the first pass. Each video file was transcribed to capture dialogue between all persons appearing in the video frame as well as indications of gestures where the body acts as a communication tool to represent manifestations of objects, or the predicted motion of objects through space. The transcriptions include references to gestures which aid in communicating one's focus, such as pointing to a line of code or a feature of a whiteboard drawing. Transcriptions also include the Recorder's additions or modifications to the MWP program code, initiating commands to the computer to run or kill the program, and any mouse interactions used to zoom or rotate the visual scene.

Each line in the transcription was time-stamped and segmented. A single segment refers to a line in the transcription which contains the audible speech, gestures, typed statements, and/or computer actions of a participant. While transcribing, new lines on the transcription document (and thus new timestamps) were created if a speaker paused, or another speaker interjected. If gestures augment, clarify, or are attached to a speaker's statement, the description of the gesture is included within the segment. If the gesture doesn't match the previous speaker's statement, or precedes a verbose description of the speaker's thinking, then the gesture begins the next new segment.

This transcription procedure generated segments differentiated by speaker and by statements focusing on a single idea. There no instances where similar ideas by the same speaker spanned several segments. For an idea statement to be similar, two statements must offer identical information. Any elaborations which add additional information to the same line are preceded by pauses and therefore coded as a new segment. However, it is possible for a speaker to switch between ideas between pauses, while reading a single line of code. For example, there are instances when a speaker, while focusing on the line of code used to update the position of the spacecraft, interpreted certain, but not all of the physics quantities. Therefore, as participants study and interpret a single line of program code, the grain size of an idea statement is more difficult to define. As a participant reads a line of programming code, does an idea refer to the interpretation of a single variable, or the interpretation of the entire line of code? Reviewing the transcript for these segments which mix interpretation of physics quantities with verbalizations of variable names reveals that there are three occurrences where a participant may begin reading a line of code verbatim and then switch to interpreting physics quantities. Due to the small number of instances where mixing occurs, there was no modification of the segmenting procedure. The coding scheme will detail how to handle these three instances of mixing when deciding how to assign a code to these segments.

The procedure used to segment the transcript followed Chi's approach in identifying an appropriate grain size for further analysis. Segmenting by pauses and turn taking produced segments which naturally contain single ideas, with one exception, the interpretation of multiple computer variables contained in a single line of programming code. Participants use these segments to communicate one's ideas with each other, and using the idea as the unit of analysis adds context validity to the intent of the verbalization.

The remaining sections of this chapter will define the focus of each coding pass of the transcripts. Each coding pass will assign a single specific code to each segment, based on the evidence presented in a single segment. Coding passes are differentiated according to the information a segment contains. For example a reading of a line of programming code includes evidence that the speaker is attempting to make sense of the program, the line of programming code the speaker is paying attention to, and how the speaker is making sense of the program code. To extract this evidence, at least three passes of the transcripts are needed.

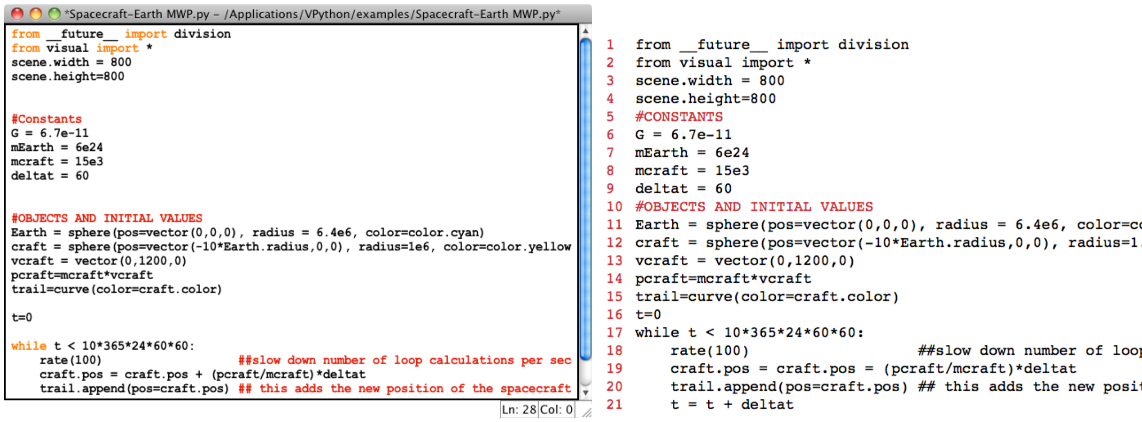


Figure 4.8: On the left, the Spacecraft-Earth MWP program provided to the participants. On the right, the same program with all blank lines of code removed and each line of code assigned a line number.

4.3.1 Line-Number Coding

The first pass of the transcripts uses evidence in the segment to identify the line of program code to which the speaker is referring. It's possible that a speaker is referring to multiple lines of code within a single segment. For example, a speaker may say, "The loop, this is where everything is happening." The segment refers to the loop as a structure, and all lines of code contained in the loop structure are coded for this segment. There's no evidence that the speaker has reviewed all of these lines of code. The multiple line numbers offer evidence that the speaker is talking about a section of code as opposed to a single line of code.

During analysis, each MWP program was modified from its original state as presented to the speaker, mainly to remove any blank lines of code which separated different coding sections. Figure 4.8 compares the Spacecraft-Earth MWP as it appeared to the participants, and the modified Spacecraft-Earth MWP after assigning line numbers to each line of code.

4.3.2 Sensemaking Coding Scheme

The sensemaking coding scheme identifies periods of time during the task where participants are either trying to understand what they're supposed to be doing, intentionally taking a break from the task, or having conversations with each other in an attempt to successfully complete the task. This coding scheme also tracks the involvement of the teaching assistant. Groups request the assistance of the teaching assistant to clarify the task, the instructional documents, or provide hints to help the group complete the task. Conversations with the teaching assistant involve requests and negotiations of information in a back-and-forth exchange. This exchange may include information about the task logistics, scaffolding questions which ask participants for information using their physics knowledge, or unnecessary banter. Any conversations with a teaching assistant receives a separate code. Therefore, the four sensemaking codes in the scheme are: Logistics, Off-task, TA, and Sensemaking.

The Logistics code applies to a speaker's statement or actions which have the purpose of identifying the information in the instructions to inform the group of a task. The code also applies to participants making the materials accessible on the monitor or asking questions to each other about the task's objectives to understand when the task is finished.

The Off-task code applies to a speaker's statements or actions which are unrelated to completing the instructional task. For example, Dora, Norma, and Lidia carried on a three minute conversation about where each person sat in lecture, not having realized that they all were in the same lecture section.

The TA code applies to all conversations, regardless of substance, that involve the teaching assistant. The research questions are not primarily focused on analyzing teaching assistant interactions, therefore it's not necessary to dig deeper into how the teaching assistant interacts with the group. All statements by speakers starting from the time the teaching assistant speaks until the time when he walks away are coded as TA.

The Sensemaking code applies to a speaker's statements or actions which have the purpose of competing the task through reading or interpretations of code, drawing predictions on whiteboards, asking a team member's ideas through questions, etc. The Sensemaking code identifies periods of time during the activity when a speaker is working towards the completion of the task.

4.3.3 Action Coding

The action coding scheme identifies the aim of a speaker's statements or actions, relevant to completing the instructional tasks. That is, statements serve a purpose through their verbalization. The underlying action of a statement may transmit information through conversation about how to say a variable name aloud (such as "oofpez"), the representation of the variable name as a physics quantity ("one over four pi epsilon zero"), or identifying the purpose of the line of code ("needed to calculate the electrostatic force acting on an alpha particle").

The action codes were generated out of the data, following a methodology described by Otero (2001) as emergent coding. That is, the codes were developed while reading the transcripts. When a new type of action appeared that hadn't been previously identified, a new action code category was created. This continued until all segments were represented by an action code. Action codes where the frequency was less than 1% were merged into the catch-all category "Other". Seven categories remained, including: Say, Interpret, Predict, Justify, Question, Evaluate, and Goal. Segments that contained no information pertaining to how speakers are completing the instructional tasks were also coded in the Other category.

The Say code applies to a speaker reading and saying-aloud the program code verbatim. For example, during the Spacecraft- Earth MWP activity, Zeke, while looking at the monitor, said "*craft dot position craft plus p-craft m-craft times delta t.*" There is evidence in this segment that Zeke was simply sounding out the variables and sounding out the names of the mathematical symbols shown. However, Zeke added extra information while saying the line of code aloud.

Zeke didn't say "craft dot pos" while reading `craft.pos`, he said "*craft dot position*". Any information added to the program code is defined as an interpretation; however, only one of the four variables was interpreted verbally. Therefore, the segment received the action code "Say."

The Interpret code applied to a speaker who added additional information to a line of code. Rather than sounding out the variable name "`pcraft`", the speaker said "*the momentum.*" While reading `craft = sphere(` Zeke didn't say "craft equals sphere," instead Zeke said "*the craft is a sphere.*" Notice the difference: in this example, the speaker assigns identity to the variable name `craft` by using the verb "is". The interpret code captures participants using the vector quantities to identify locations in space and directions of motion through space, the association of initial values to physics quantities according to the name given to the variable, and thinking about the function of an entire line of code.

The Predict code applies to a speaker stating how the time-varying visual output will appear when the program is instructed to run. Participants are instructed to use the whiteboard on the table in front of them to draw a representation of the visual output before running the program. Any drawings of the location of objects on this board is a segment in the transcript. Drawings of 3D objects on the whiteboard or how these objects move through space receive the Predict code. For example, Paine saying, "*it's the Moon dude, going around the Earth*" is a prediction that the program will either display the Moon traveling around the Earth, or will resemble the Moon-Earth system. Earlier in the transcript, Paine drew a circle on the whiteboard and labeled the circle "*Earth.*" Both of these examples received Predict codes. The Predict code is independent of an interpretation of the line of code which might validate the prediction. Participants can (and often do) make predictions of the visual output in addition to interpreting the lines of code used by the program to display these objects.

The Justify code applies to a speaker's statement which used information in the MWP code or the visual output as evidence for one's interpretation or reasoning. For example, after running the program and interpreting the visual output, Norma said the following while pointing at the Earth in the visual output: "*I guess there's nothing in the program about the force from the Earth.*" Norma watched the visual output, and validated the visual output based on the omission of code, rather than the existence of code. She argued that the visual output was reasonable for programs where the gravitational force was excluded. Another example of a justification comes during the prediction task, where Norma, after agreeing with Dora on her prediction of the spacecraft orbiting the Earth, said , "*because of the gravitational constant and stuff.*" Norma was clearly using a line of code as evidence of support for Dora's prediction.

The Evaluate code applies to statements which judged a prediction, visual output, or modification of the program code. Most Evaluate codes occur after the program runs for the first time. Participants compared the visual output to their whiteboard prediction and quickly stated if the prediction is correct. Roslyn, for example, said "*Yeah, we were right,*" immediately upon seeing the visual output although the group's whiteboard prediction did not match what she saw. Roslyn quickly revoked the evaluation after she recognized that she spoke too soon. Isis took her evaluation of the group's Mass-Spring prediction one step further: "*okay...well that's where we said it would be, but it's not moving...*"

The Question code applies to statements which asked for clarification, agreement of one's ideas, or help from a team member. A statement which ends with an inflection of pitch doesn't necessarily indicate that the speaker asked a question. For example, while thinking about what the spacecraft would do in the prediction task, Zavier said, "*It would be, craft would be up, wouldn't it?*" Zavier added new information to the conversation that had not previously been suggested as a possible prediction. The inflection in the end of the statement might suggest that the statement was a request for agreement. Segments like this that follow the formula: " new information + request for agreement" were coded according to the new bit of information provided, not the request for agreement. Therefore, in this example, Zavier's statement was assigned a Predict code. One example of a segment coded as Question comes from Ramon, who said, "*p craft, what's p craft?*"

The Goal code applies to statements which summarized the changes to the program or visual output to match the desired behavior of the system. These statements were also planning statements, where a speaker suggested a possible route to follow to complete the task. The speaker may have suggested adding physics quantities or calculations, or using previously defined quantities to achieve the goal. The goal category includes statements about what participants wanted to see happen in the visual output. For example, Goal statements included segments such as "*Okay, so we need to change it so that it goes in a curve around the Earth,*" or "*okay so the position of the ball would have to be the same position...*".

Statements that did not fall in any of the above categories were coded as Other. These statements included off-task conversations, directions to complete some action, metacognitive statements about one's understanding, reading instructions, incomplete and/or hanging statements void of information, and confirmations of another speaker's idea. Examples include, "*it would be off to the...*", "*Yeah, so there you go,*" and "*I guess since its got gravity we want to make it.*"

4.3.4 Focus Coding

The focus coding scheme adds a finer resolution to the action codes. The focus pass assigns codes identifying the attention of the speaker's action as it relates to the MWP code or visual output. The focus code is identified by asking, "What is the speaker talking about?" Participant predictions are either focused on the location of objects in space or the dynamics of objects. Therefore, the focus coding categories for the Predict Action Code are Objects or Dynamics. The focus coding categories for Interpret Action Codes are different than the focus coding categories for the Predict Action Codes. An interpretation can focus on the representation of a physics quantity, the function of a line of code, the attributes of 3D objects, and more. This section will define Focus codes developed for the Interpret, Predict, and Goal Action Codes.

Focus: Interpret Action Codes

The Focus pass on Interpret action codes identifies what the speaker is interpreting while reading a line or segment of program code. The speaker might simply identify the variable name as representing a physics quantity, or may interpret the function of the line of code when the program runs. For example, participant interpretations were determined to fall into six distinguishable categories referring to either the program code or the visual output. While interpreting the program code, speakers focused on the physics QUANTITY represented, the OMISSION of lines of code, the ATTRIBUTE of 3D objects, the DIRECTION in which vector quantities point, or the FUNCTION of a line of code The final Focus code refers to interpretations about the visual OUTPUT.

The QUANTITY code applies to interpretation statements where the speaker is interpreting the variable name or the value assigned to the variable as the representation of a physical quantity. This code also applies if the speaker is identifying the quantity as a positive or negative value, without using the information to indicate direction. And, the code applies to the identification of a line of code as a fundamental physics principle, without regards to its function in the program code. For example segments coded as participants focusing on the interpretation of a physics quantity include "*[Where's gravity?] Yeah, [pointing to screen] it's right there [points to* `G = 6.7e-11`*] idn't it?*" or "*"okay so that's the position update formula for alpha dot pos"*.

The OMISSION code applies to interpretation statements where the speaker is noticing that the MWP is missing some line of code that was expected to be included in the program. The speaker may indicate that the program doesn't have a line of code which uses a previously defined variable name. For example, the statement "*it doesn't ever calculate a spring constant*" focuses on the omission of a line of code which is expected to appear in the program.

The ATTRIBUTE code applies to the interpretation statements where the speaker was discussing the physical dimensions of the 3D objects or their locations in space. For example, statements like "*position vector for the craft is the same as Earth vector*" or "*the spring's initially compressed*" are focused on interpreting the attributes of the 3D objects. Notice that this is different than representing a 3D object's position on a whiteboard. It is often the case that an interpretation of a 3D object's attributes follows with a whiteboard prediction of where the 3D object appears in the visual scene.

The DIRECTION code applies to the interpretation statements where the speaker is explicitly using the vector nature of physical quantities to map to a coordinate system. For example, statements such as "*and the velocity is going striaight up*" or "*so the velocity is in the y direction*" indicate that the speaker is using the x, y, and z coordinates of the vector quantities to indicate the direction of the physical quantity.

The FUNCTION code applies to the interpretation of a line of the MWP with respect to its purpose. Generally, lines of code define constants, initial values for physical quantities, assign names to 3D objects (or produce 3D objects associated with variable names), or update the values of variables and 3D object attributes. For example, statements which associate a variable name representing an object, whether realized as a specific 3D shape or not, such as "*so you have Earth*" or "*[You have a craft] which is also a circle*" are statements focused on the function of the line of code. Participants also focused on the function of lines of code in defining calculations, such as "*so you're defining K alpha and you're plugging it in there..*" and "*so..the momentum is updating.*"

The OUTPUT code applies to all interpretations about the visual output. When the program runs to produce a time-varying visual scene, participants are tasked with determining what information is in the scene, then compare the output to their predictions. Any statements about the information presented in the scene are interpretations of the visual output. Due to the instructional task asking participants to evaluate their prediction based on the visual output, there are few statements where participants interpret the visual output, unless something happens that is unexpected or unclear. For example, the statement "*looks like the Earth is moving away and getting smaller...okay*" the speaker is trying to interpret what's happening as the camera begins zooming out. In another example, the speaker shows surprise with "*whoah...it didn't do anything*" expecting the spring to interact with the ball. The speaker is only stating surprise by describing the visual output, not offering any additional information on justifying the output from the existing code, or evaluating the prediction.

Focus: Predict Action Codes

The Focus pass on Predict action codes identifies if the speaker is predicting the DYNAMICS of objects in the visual output, or the location of the visual OBJECTS in the visual scene by drawing their locations on a whiteboard. The DYNAMICS code applies to prediction statements, whiteboard drawings, or gestures that describe how one thinks the objects will move in the time-varying visual output. The OBJECTS code only applies to the drawing of the location of 3D objects on the whiteboard. Verbal descriptions of the locations of 3D objects reflect the interpretations of the attributes of 3D objects, and are therefore not predictions. However, when a speaker draws the location of a 3D object on a plane representing the 3D visual space, the speaker is developing a prediction of where the objects in the visual output should appear.

Focus: Goal Action Codes

The Focus pass on Goal action codes identifies if the speaker is stating how the objects presented in the visual output should move through the scene (BEHAVIOR), or if the group should focus their efforts to ADD CODE or MODIFY CODE in the existing MWP. The BEHAVIOR code, for lack of a better word, applies to statements about how objects in the time-varying visual output should evolve when the program includes all the valid interactions between the objects. For example, in the segment "*Alright, so we, [moves pointer to screen] I guess are going to have to [motions with hand] make it rotate... around it,*" the speaker is identifying the current task through the actions of the spacecraft in the visual output.

The ADD CODE or MODIFY CODE codes apply to statements where a speaker begins planning out the changes to the MWP code. Participants may think about using existing variable names or creating new variables to develop new lines of code (ADD CODE), or decide to alter existing lines of code by changing values or adding variables to what is already there (MODIFY CODE), such as adding an additional force to the line of code responsible for calculating the net force acting on an object. Planning statements such as "*We've gotta make something that's like Fnet.*" and "*write all those equations first like r equals whatever...*" indicate the speaker is focusing on how to complete this task.

4.3.5 Reliability and Validity Testing

Following the development of the coding schemes, an independent rater was asked to read through a section of the transcript and code individual segments according to the definitions provided for each scheme. Comparing the independent rater's coding of the transcript segments to this researcher's coding of the transcript segments offers an evaluation of the reliability of the coding scheme to generate the same coding result among different researchers. Two independent researchers were used to determine a measure of reliability for the coding schemes.

For the Line-Numbering Coding Scheme, the rater's task was to read each transcription segment provided and identify if the speaker was referring to a line of code in the MWP. If so, the segment was assigned one or more line number codes to identify which line the speaker was referring to in conversation. If the application of the coding scheme is consistent among raters, as measured by agreement between raters, then the coding scheme is clear enough to be used by research colleagues who are not involved with the research project. On the first attempt, the agreement between raters was 80% on all segments, and after conferencing on coding differences, agreement increased to 100%. On the second attempt, agreement between raters was 92% for all segments, and after conferencing on coding differences, agreement increased to 100%.

Table 4.4: Reliability Measures for Coding Schemes

Coding Scheme	Cohen's Kappa
Sensemaking	1.00
Action	0.86
Focus:Interpret	0.89
Focus:Predict	1.00
Focus:Goal	0.84

The Sensemaking, Action, and Focus Coding Schemes were evaluated on their reliability using a measure that accounts for the possibility that two rater agree on the code applied to a segment due to chance, rather than the substance of the segment. The Cohen's Kappa (Cohen, 1960) statistic compares agreement taking into account the number of categories in a coding scheme. The interpretation of Cohen's Kappa values greater than 80% are considered measurements of acceptable agreement between raters (Geisler, 2003). Table 4.4 reports the results for reliability measurements for each coding scheme. The Action coding scheme required two revisions to clarify differences between the Interpret, Predict, and Other categories. The reported Cohen's Kappa value in Table 4.4 measures the reliability between two raters applying the revised coding categories for the Action coding scheme.

Following the reliability tests, all segments were coded using the revised coding scheme. Segments were sorted into coding categories for each scheme. Two segments were randomly selected from each coding category to serve as examples for the coding category. An independent researcher examined the example segments for each coding category and evaluated if the coding definition is a valid representation of the information in the segment to measure the validity of the coding schemes. The independent researcher reported that for each coding category, the examples provided captured an accurate depiction of the information in the segment.

Chapter 5

Results

This chapter reports the results from the analysis performed on the data, following the methods described in Chapter 4. Each transcript of the participants working through the three activities involving MWPs was coded according to the revised coding schemes. Table 5.1 lists the alias names of the participants for the three activities and grouped according to the assigned Group Number. The Appendix contains diagrams showing the location of each participant in the lab section. As discussed in Chapter 4, the membership of these groups changed between activities.

5.0.1 Data Plots

Using a time code generated for each transcription segment, it is possible to generate a time-series plot for each group, indicating when the segments occur during the course of the activity. Combining the time codes for each segment with the speaker's alias name, the Line-Number codes, and the codes from the Action and Focus coding schemes produces enough data to generate a plot that shows who is speaking, which line the speaker is referring to, and if the speaker is adding additional meaning to the line of code or stating a prediction about the location of 3D objects or how the objects move through the visual scene. These line-number

Table 5.1: Group Membership by MWP Activity. Boldface denotes the Recorder

Group	Spacecraft-Earth MWP	Mass-Spring MWP	Scattering MWP
G1	**Paine**, Zeke, Frank	Roslyn, **Howard**, Beatriz	**Zeke**, Isis, Roslyn
G2	**Isis**, Celia, Madeline	**Celia**, Madeline	**Frank**, Madeline, Estelle
G3	**Roslyn**, Yolanda, Estelle	**Yolanda**, Zeke, Xavier	**Xavier**, Tina, Howard
G4	**Tina**, Howard, Xavier	**Isis**, Estelle, Frank	**Paine**, Yolanda, Celia
G5	**Otis**, Ramon, Greg, Fern	**Ramon**, Selma, Dora	**Eugene**, Dora
G6	**Dora**, Lidia, Norma	**Lidia**, Otis, Selma	**Beatriz**, Lidia, Max
G7	N/A	**Norma**, Fernanda	**Greg**, Otis, Selma
G8	**Eugene**, Selma, Beatriz	**Eugene**, Max	**Norma**, Ramon

plots will inform how the group is moving through the program code and differences between lines of program code in how they are interpreted by different groups of participants. Figure 5.1 shows the Line Number plot for Group G1. To make it easier to follow the progress of the group through the graph, the points on the plot are connected with a line to serve as a visual aide. This line is not part of the data nor should there be any conclusions drawn from interpreting this line.

The vertical axis of this plot corresponds to the line number codes for the Spacecraft-Earth MWP. For reference, a picture of the Spacecraft-Earth MWP code is provided to match the line-number to the program. Recall, the MWP provided to the participants does not contain these line numbers and also contains blank lines, which have been removed from this plot.

The horizontal axis of this plot corresponds to the lab time, measured in seconds, where zero seconds corresponds to the group beginning the MWP activity when the activity's instructions appear on the monitor. For example, in Figure 5.1, Group 1 begins studying the MWP code 155 seconds into the activity. During the first 155 seconds, the group was engaged in actions to inform them of the initial task to study the program code. Group 1 also arranged windows on the screen and opened the MWP program code during this time.

The vertical lines on the plot indicate points in time where a Prediction segment occurs in the transcript. The capital letter at the top of the graph indicates which participant in the

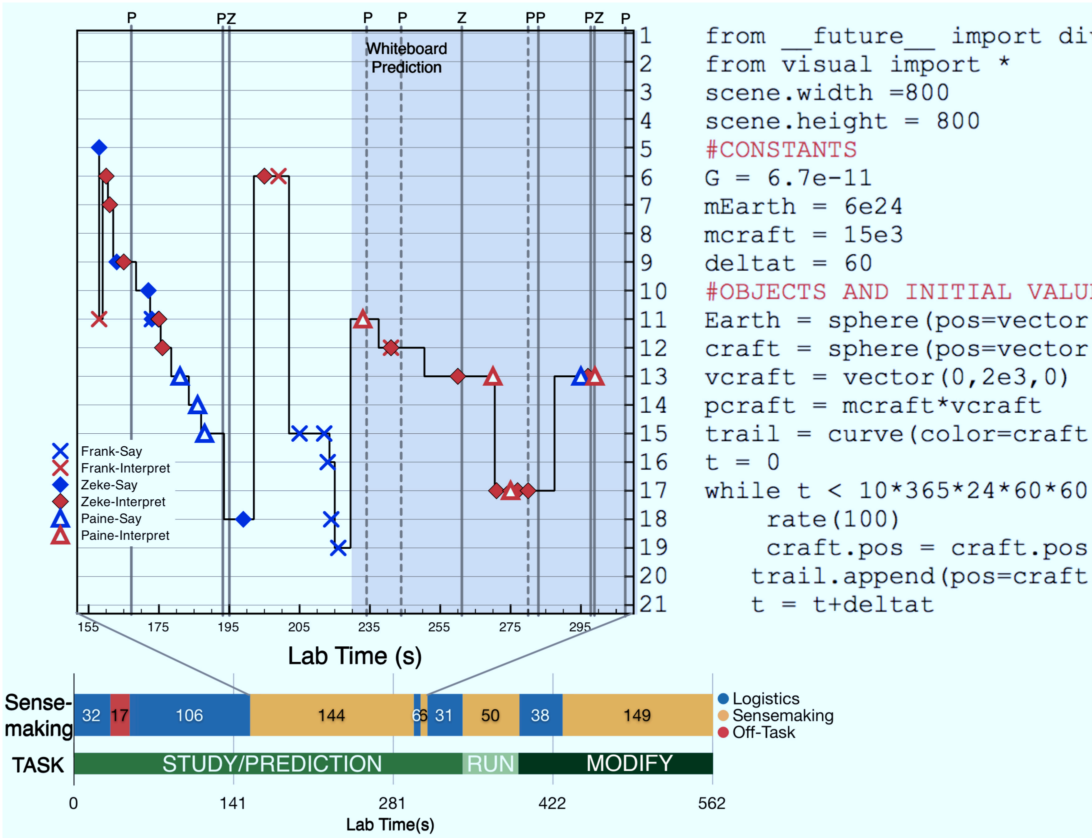


Figure 5.1: A Line-Number plot for Group G1 as the participants work on the Spacecraft-Earth MWP activity.

group makes the prediction. A dashed vertical line represents a prediction about the location of objects with a whiteboard drawing. A solid vertical line represents a prediction statement about the dynamics of the visual scene. Finally, when the group begins to use the whiteboard to focus their attention on creating a prediction, the background of the plot changes from white to gray.

Underneath the Line-Number plot are two horizontal bar graphs which share a common horizontal axis. The top bar graph identifies periods of time during the activity when the participants are engaged in actions corresponding to the Sensemaking coding scheme (Logistics, Sensemaking, Off-Task, TA). The bottom bar graph identifies the task the participants are working to complete (Study/Prediction, Run Program, Modify Program). Most groups combined the Study and Prediction tasks together, creating or discussing their predictions as the group studied the program code, therefore it is not useful to distinguish between these two activities on a bar graph. The Line-Number plot will identify when predictions overlap with the interpretation of the program code. The horizontal axis of the bar graphs correspond to the same lab time defined above, beginning the lab clock when the group opens the instructional document. The activity continues beyond what is shown here. However, the actions of students to modify the program code are outside the scope of the research questions.

This chapter looks at each MWP activity separately, reporting the results which will be used to draw conclusions to answer the research question: How do students use their physics knowledge to make sense of incomplete, but functional programs? The sequence of the analysis follows the sequence of the MWP activities from lab, starting with the Spacecraft-Earth MWP, then the Mass-Spring MWP, and finally the Rutherford Scattering MWP. The final section of the chapter will analyze the data across the three MWP activities to see if there is any significant differences among the activities with how the participants completed the MWP tasks.

5.1 Spacecraft-Earth MWP

Seven groups of participants completed the Spacecraft-Earth MWP; however, one group did not follow the instructions in the activity, directing the group to first read through the program and make a whiteboard prediction. Instead the group decided to first run the program. Since the group did not initially generate an informed prediction of the MWP before running the program, their actions are removed from the data set, leaving six groups. It's not appropriate to compare this group's actions to those participants who followed the instructions. However, it will be interesting to see how a group recovers if the group follows some other task sequence other than the one provided in the instructional document.

This section uses the Line-Number plots to determine how the participants in each group use physics to make sense of the Spacecraft-Earth MWP code, generate a prediction of the visual output, and ultimately make decisions on how to modify the program code to achieve the goal of the activity. Each subsection reports features of the data which will be used to draw conclusions to answer the research question.

The six groups who followed the directions to complete the Study & Predict tasks before running the program followed different strategies for generating a prediction. Groups either created a whiteboard prediction at the same time as reading the program code, read the program code first and returned to the lines of code which create 3D objects and constrain motion to inform a prediction of the visual output. Four of the six groups created predictions on the fly, meaning as the group interpret a line of code which might have some function to produce a 3D object, or some constraint on the motion of 3D objects in the visual scene, the group used this information to generate a whiteboard prediction. One group read the program code first, for interpretation purposes, and returned to the beginning of the program to gather information to inform a whiteboard prediction.

5.1.1 Study/Prediction Task

First interpret code, then create prediction: Eugene, Selma, Beatriz

Eugene, Selma, and Beatriz, participants in Group G8 begin the MWP activity by first reading and interpreting the program code, then returning to specific lines of code to use as a basis for their whiteboard prediction. Figure 5.2 shows that the interpretation of the program code follows a top-down sequence, beginning with Line 6, the declaration of the gravitational constant, and ending with Line 21, the last line of the program. Eugene dominates the discussion until Line 15, which creates the `trail` object. The group ponders the interpretation and function of this line of code:

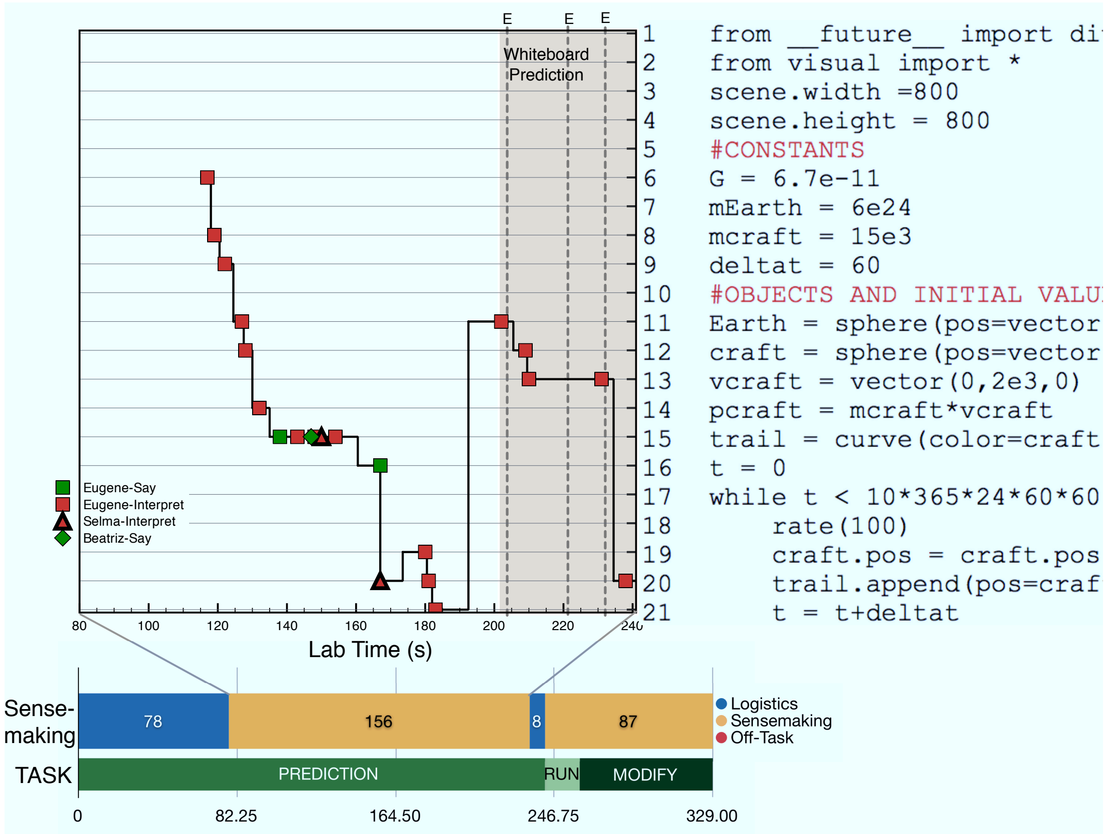


Figure 5.2: Line Number plot for Eugene, Selma, and Beatriz as they study and predict the MWP

Eugene: wait, going to... trail..

Eugene: trail must be a curve behind.

Beatriz: craft.

Beatriz: yeah, i dunno

Eugene: that doesn't do anything..

Selma: I think it is just like where its going...just so you can see.

Eugene: but the curve is only dependent on the color of the craft... so its not actually going to do anything yet.. I don't think..

Although Beatriz participates in the exchange, she doesn't offer any constructive help. Selma, on the other hand, interprets the possible function of Line 15 in terms of the visual output, adding an expectation to how the `trail` object might be used in the visual scene to aid in tracking the motion of the spacecraft. It is peculiar that Selma used the words "I think it is just like where its going." Interpreting Selma's word choice, it appears that Selma expected the trail to offer evidence to extrapolate a future position, rather than a history of the location of the spacecraft. Eugene's interpretation stuck with the function of this single line of code, interpreting that the color attribute of the `trail` matched the color attribute of the spacecraft. As Eugene continued on reading through the program code, Selma returned to her prediction, offering some evidence from her interpretation of Line 20 to support her claim:

Eugene: t...

Selma: cause it says trail like, in the while loop..

Eugene: Ah..okay gotcha.. yeah that must be what its doing..

Eugene is the only participant out of the six groups who interprets Line 19, the line which updates the spacecraft's position attribute. However, it's not clear that Eugene is associating this function to the line of code. Eugene's exact phrasing is "*position update*." Using the Focus coding scheme definitions, this segment fits the QUANTITY definition, where the speaker identifies a fundamental physics principle without specifically referring to its function in the program code. Note that one of the limitations of this coding scheme is that it only seeks to identify what is said by participants, not the intent of the speech. There's not enough evidence in this segment to call this an interpretation of the FUNCTION of this line of code, although that might have been the intent of Eugene's statement.

After reading through the program code, Eugene says, "Okay, so we're supposed to draw this," and returns the focus back to Line 11 to read the position of the sphere representing the Earth. Eugene then does the same for the spacecraft, moves on to Line 13 to use the direction of the velocity, and finally back to Line 20:

Eugene:so Earth starts at...zero, zero, zero.

Eugene:[draws earth at center of board]

Eugene: the craft...is.. over here?

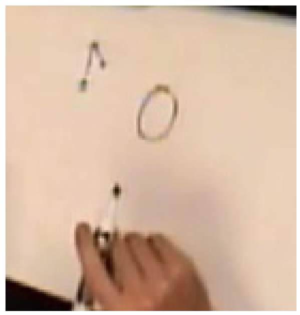
Eugene:[draws craft to left of earth]

Eugene: yeah..

Eugene: and velocity ... is ...*inaudible*

Eugene:so its going that way....[motions in +y axis on whiteboard frame]

[draws velocity vector]



Eugene:and will be a random trail that will be behind it..and that's it....

Eugene returns to the Objects section of the code to inform his whiteboard prediction, to draw the locations of the spacecraft and the Earth. Notice that Eugene doesn't take his prediction of the spacecraft's motion any further than the direction of the initial velocity. When Eugene returns to the loop, he adds a comment about the `trail.append` line, and ends his prediction with acknowledging that there's nothing else the program will do.

Eugene's approach to the `trail` object is not consistent with his treatment of the initial velocity. In his interpretation, Eugene only comments on the function of the line of code of the trail object without imagining it's role in the output. The same is true for the initial value for the velocity vector. Before the while loop, the declaration of the initial value has no functional purpose other than some variable stored in system memory. It's not until Line 19 in the while loop is this velocity actually "attached" to the spacecraft object to update the spacecraft's position. Eugene doesn't break down Line 19 to interpret the functionality of the line of code, as far as one can tell from the transcript. There are two possibilities, first that Eugene associates the spacecraft's motion with the initial velocity value, or second, Eugene was informed about the spacecraft's movement in the visual scene during his interpretation of Line 19, when he said "position update." This data doesn't have enough resolution to identify how Eugene is thinking about the interplay between initial values and the while loop; however, it does reveal that Eugene can recognize physics principles and quantities represented in VPython syntax, even when presented in terms of the attributes of 3D Objects rather than algebraic symbolic forms.

Making predictions while interpreting code: Roslyn, Estelle, and Yolanda

Roslyn, Estelle, and Yolanda, participants in Group G3 approach the study/predict task differently from Eugene, Selma, and Beatriz. Instead of first reading the program code for understanding, and returning to specific lines of the code to inform a prediction, Roslyn, Estelle, and Yolanda are creating a prediction as they move through the program code, starting from the beginning, as shown in Figure 5.3

Roslyn begins the task slightly confused about the prediction task, thinking that simply identifying the objects in the visual output satisfies the goal of the prediction task:

Yolanda: okay so.. lets see what we got.

Roslyn: so you have earth [cursor over Earth=],

Roslyn: [*so you have*] a craft [cursor over craft=],

Roslyn: [*so you have*] a vector.

Roslyn: So earth and a craft.

Yolanda: Erases whiteboard.

Roslyn: Let's run it [laughs]

Yolanda: [laughs] well that's cheating.

Estelle began interpreting the program in the similar fashion to Roslyn; however, Yolanda

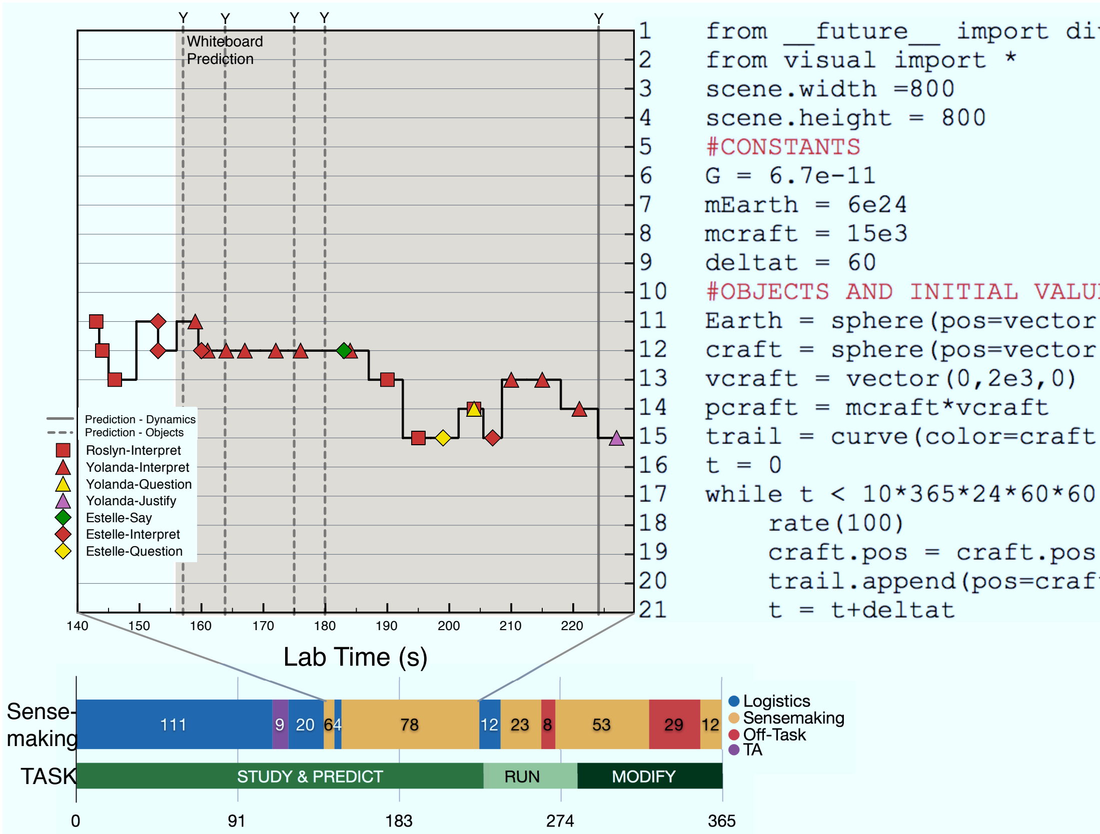


Figure 5.3: Line Number Plot for Roslyn, Estelle, and Roslyn as they study and predict the MWP

took over leading the group through the program, with the goal of generating a whiteboard prediction while studying the program code. Yolanda spent much of the time between 160 and 185 seconds on interpreting the attributes of the spacecraft to draw the location of the spacecraft relative to the Earth. Yolanda generated a whiteboard prediction for the location of the two 3D objects which is somewhat interesting. Yolanda's coordinate system for the prediction lied along the bottom edge of the whiteboard, shown in Figure 5.4.

Further, Figure 5.4 shows the labeling of an arrow drawn on the spacecraft object representing the velocity vector, and a dashed line representing the trajectory of the spacecraft. This images was rotated clockwise 90 degrees. Yolanda drew the prediction relative to her seating

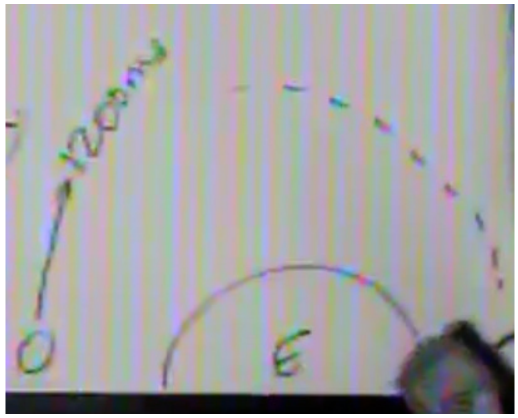


Figure 5.4: Yolanda's whiteboard prediction.

position in the group although there was ample space in front of Roslyn. Roslyn and Estelle discussed the function of Line 15, getting ahead of Yolanda:

Roslyn: And you have a curve,

Estelle: I dunno where the.. what what trail? Trail...

Yolanda: pcraft, what's pcraft?

Roslyn: Momentum

Yolanda: Yeah, so there you go.

Estelle: Okay so there's a trail behind the craft.

Roslyn identified the `curve` object as some object, with Estelle adding that Line 15 produces a trail behind the spacecraft. However, it's Yolanda who, a few sentences later, used the `curve` object as justification for the spacecraft *curving* around the Earth:

Yolanda: So then its going to travel in a curve,[drawing a dashed line around Earth]

Yolanda: it says around the Earth.

Estelle: Okay

Yolanda: Right, ain't that what it says?

Roslyn: Yep...run.

Yolanda changed the interpretation of Line 15. Yolanda associated the curve as a constraint on the motion of the spacecraft, that the spacecraft is following a curved path. This interpretation of the `curve` object is the first time a 3D object interfered with, and misguided the group's prediction. The participants interchange the usage of curve to reprsent a thing with the usage of curve to represent an action. Having enough information to draw a prediction of the spacecraft's initial location and its trajectory, the group decided to run the program. There was no further investigation of the program code, including the while loop.

Making predictions while interpreting code: Xavier, Howard, and Tina

Xavier, Howard, and Tina, participants in Group G4 also blended their prediction with the interpretation of the program code; however the group never drew a whiteboard prediction. The group communicated their predictions and interpretations of the program code with gestures and speech. Also, the group arrived at two separate predictions at runtime. Figure 5.5 shows the Line Number plot generated from the transcript.

Tina began by saying aloud the name of the constants in Lines 5-7. Immediately, Howard suggested that the spacecraft moves around the Earth, while his finger traces out on the whiteboard, a single circular orbit approximately 4 inches in diameter. As Tina continued silently looking at the code, Howard and Xavier began an off-task conversation about throwing a whiteboard marker. Tina ended the off-task conversation by asking the guys about the reference in Line 13 to `Earth.radius`.

There was very little interpretation going on in the group. However, the episode becomes interesting as Xavier focuses the group on making a prediction of what the spacecraft will do in the visual output:

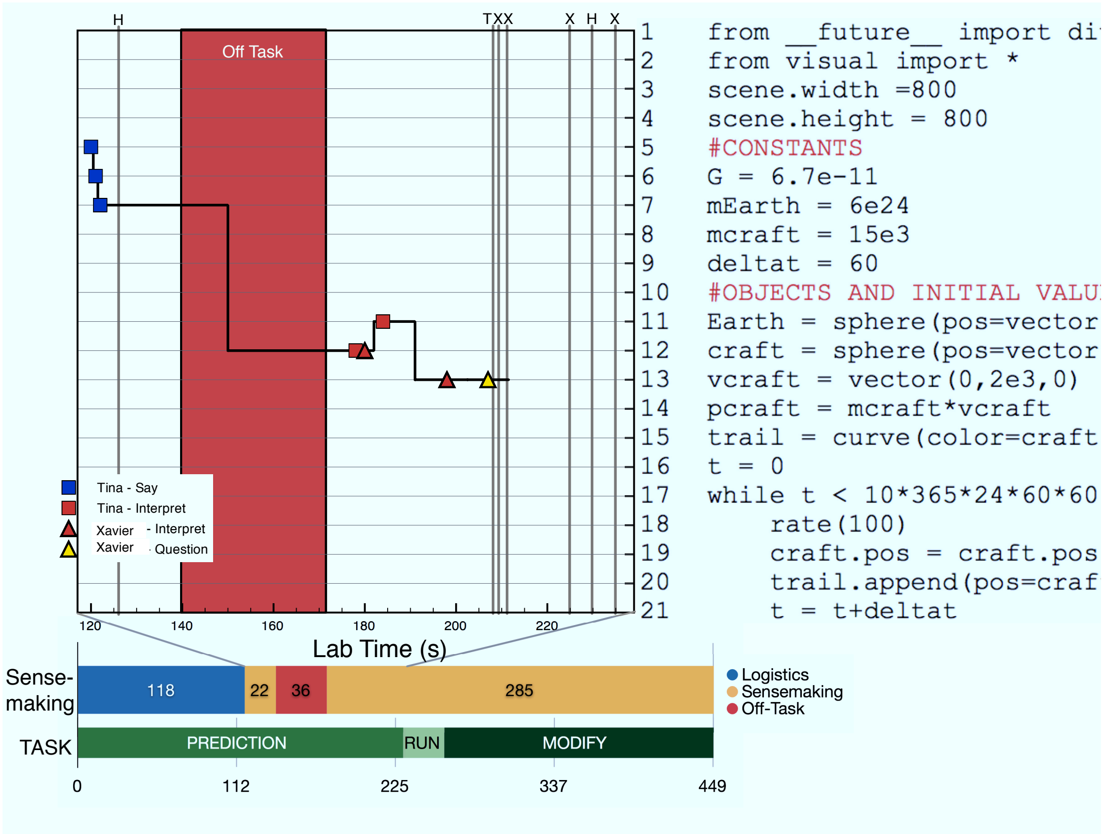


Figure 5.5: Line Number Plot for Xavier, Howard, and Tina as they study and predict the MWP

Xavier: So... alright the... velocity of the craft,...

Xavier:Is that what vcraft is?

Tina: It's going up..up.[gestures with finger twice]

Xavier: It's going up. [gestures up away from table]

Xavier: So its going to go up..once it breaks through that..i dunno..hit F5 and find out.

Tina: No we're supposed to make a guess before we do it...

Howard: [laughs]

Xavier: I think its going to start going up and then [right hand moves up and drifts sideways slowly, away from his body]

Howard: It's going to keep going up ...

Xavier: Yeah...

Tina: [runs program]

Xavier: [while the program is taking some time to compile, but before the visual output is displayed] Then start curving a little bit...

Everyone was on the same page, agreeing that the velocity of the spacecraft was up, and that the spacecraft will travel up. Xavier states that he is considering that the spacecraft might do something in addition to traveling up by the motion of his right hand, drifting away from his body. Six seconds later, Xavier's speech catches up with his gesture, and suggests that the spacecraft will "*start curving a little bit.*" The initial gesture didn't contradict Xavier's speech, but preceded it. The gesture replaced the need for a verbal description, however six seconds later, Xavier found the word he needed to describe his thought, and it is quite interesting that he chose *curving*. Xavier was the first person to use this word in the group, and he didn't use the word in interpreting Line 15. There's no evidence that Xavier read or interpreted Line 15. But, he chose *curving* to describe the motion of the spacecraft and there's just not enough evidence to determine if Line 15 had some influence on Xavier's prediction statement.

At the end of the prediction task there are two predictions: Howard believed the spacecraft will travel up and keep going up, Xavier believed the spacecraft will go up and start curving, and Tina remains silently neutral.

Making predictions while interpreting code: Dora, Lidia, and Norma

Dora, Lidia, and Norma, participants of Group G6 also focused on creating a prediction as they read and interpret the program code, as shown in Figure 5.6. This group differentiates their interpretation of the program from other groups by the difficulty they have with Lines 11 and 12, the lines of the code which create and place the 3D objects representing the Earth (Line 11) and the spacecraft (Line 12). Particularly, Norma had difficulty interpreting the position

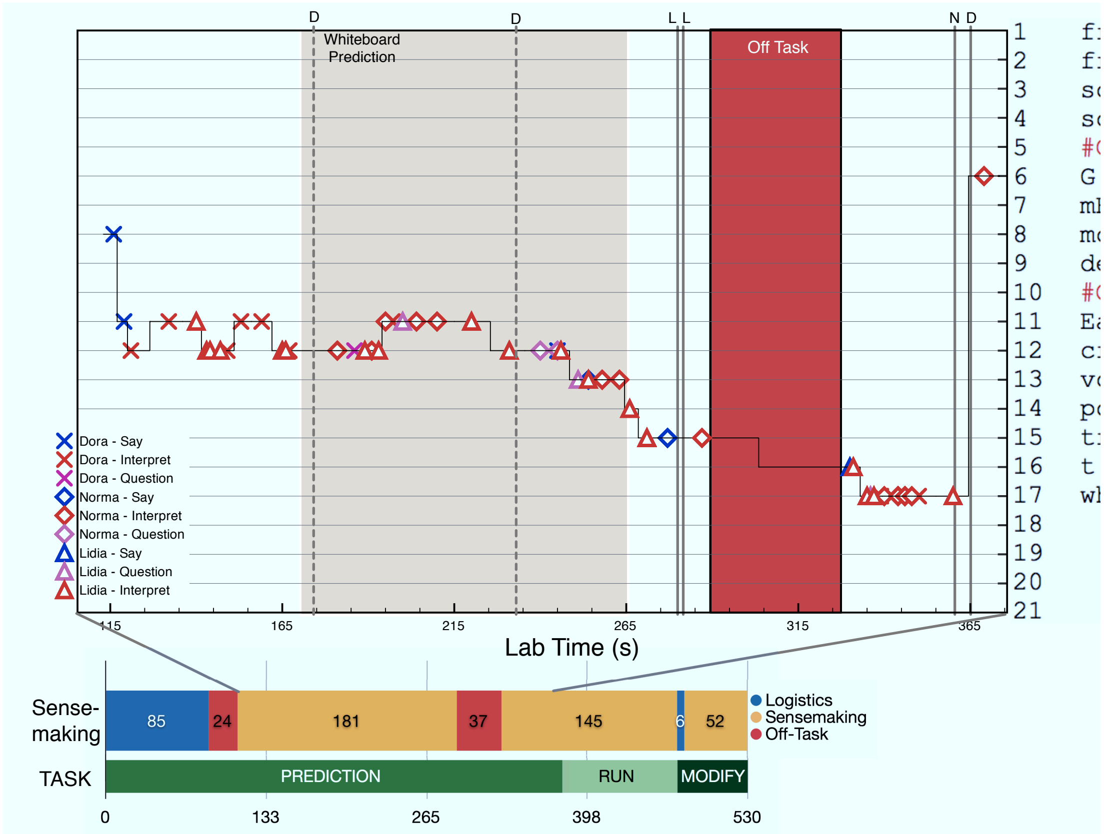


Figure 5.6: Line Number Plot for Dora, Lidia, and Norma as they study and predict the MWP

of the spacecraft based on the value set for `Earth.radius`:

Dora: okay, two sphere..

Lidia: so we have an earth and a craft..

Dora: are they insane?

Lidia: and a vector connecting them..

Lidia: or coming at the craft?

Lidia: A position vector maybe?

Dora: Like position vector for craft is the same as earth vector?

Dora: Is like zero zero zero times negative 10 which is scalar...

Dora: It's like same as zero zero zero.

Lidia: No its the radius..

Dora: oh..

Lidia: ..its the radius

Dora: no radius, radius

Dora: So okay okay way over this side, like [left hand on left edge of the whiteboard]

Dora: [uses marker to draw earth near center of board]..

Dora: [while labeling the circle with an "E" inside, says] Okay its Earth

Dora:[moves marker to the left of the Earth on the x axis]

Lidia specifically interpreted the `pos` attribute for craft to be a position vector, since the vector is "*coming at the craft?*" However, Dora was confused about the value for the position vector, which uses a value stored as an attribute for the radius of the Earth, and confused the reference to `Earth.radius` with `Earth.pos`, which had a stored value of $\langle 0, 0, 0 \rangle$. In the next section, Norma was confused about the location of the spacecraft, and it's not exactly clear what information she was using in her interpretation of the spacecraft's position. However, Lidia and Dora were able to convince Norma that they were correct with their interpretation that the spacecraft is 6.4e6 to the left of the Earth:

Norma: And then the craft is above it [points up in air, then points above earth] above the...

Dora: Well its x axis isn't it?

Norma: Yeah, so

Lidia:It should be just right beside it...

Norma: Oh it should be over here shouldn't it [points in the middle of Quad I on the cartesian plane defined by the x and y axis on the board], this..

Lidia: It's going negative ten..

Norma: This is negative, the earth is negative...

Norma: is this earth?

Dora: Yeah, this is position zero, zero, zero.

Norma: Okay

Dora: [writes on board $\langle 0, 0, 0 \rangle$]

Lidia: so negative would be this way, shouldn't it?

Norma: Negative earth...

Lidia: or that way right?

Norma: This isn't... this isn't zero, zero, zero...

Norma: oh, I'm looking at the wrong thing, I'm sorry. I was looking at this[points to screen, not clear which line]

The discussion involved the coordinate system as presented on the whiteboard, and plotting the locations of the 3D objects using the attributes of the two objects. The group explicitly decided that the pos attribute referred to the position vector of both the Earth and the craft. This identification of the position attributes as a physically relevant quantity is juxtaposed to ignoring the other two attributes of each object, the size of the objects and its color. However the only physics knowledge the group used was an identification of an attribute representing a physics quantity. The rest of the interpretation involved defining and using a coordinate plane. The coding scheme distinguishes the multiple interpretations of Lines 11 and 12, noting that using physics knowledge is necessary to identify the use of a position vector, but not in decoding how the values for the position vector map to a coordinate plane.

Norma and Lidia move on to Lines 13, 14, and 15 to determine the motion of the craft and identify the function of the `trail` object:

Lidia: okay, and there is a vector coming out of it, right?

Norma: no...

Lidia: What's vcraft, then?

Norma: vcraft.

Lidia: velocity maybe?

Norma: Oh yeah, let's draw that vector...

Norma:it's the velocity,

Norma: so its going up [gestures out of board]

Lidia: right

Norma: That's the direction of its velocity, so just draw an arrow going up...[draws upward arrow]

Norma: yeah..

Lidia: and the momentum... is the mass times the velocity, obviously...

Lidia: trail,.. makes a curve?

Norma: curve..

Lidia: So obviously it's going in a curve

Norma: you wanna..

Lidia: maybe its gonna curve around the earth...[gestures in circle]

Norma and Lidia discussed the initial velocity of the spacecraft, and represented this initial velocity as a vector on the whiteboard. Lidia began interpreting the `curve` object as a description of the behavior of the spacecraft, associating the spacecraft's dynamics with the constraint on its motion that the spacecraft will follow a curved path around the Earth. Norma remained silent, and immediately after this conversation, Norma recalled a program that was shown in lecture. Norma didn't have access to the program code during the lecture demonstration, but she did see the visual output, which displayed a trail object as a sphere moved around on the screen:

Norma: I think its gonna like ...when she did those programs in class its

Lidia: yeah

Norma: gonna show that

Lidia: yeah

Lidia: gonna..

Norma: [gonna show that] where its been[gestures in circle]

The group began an off-task conversation about where each person sits in the lecture hall.

Dora focused the attention of the group back on the program code to interpret how long the loop will perform iterative calculations (Line 17), converting the number of seconds into years. Lidia reported that this amount of time was equal to "*315 million 360 thousand..yeah that's about a year or two...*". At this moment, the group finalized their prediction of the motion of the spacecraft around the Earth and justified the prediction based on a line in the program code:

Norma: I'm assuming... what's gonna happen, is its gonna go around [points on whiteboard]

Lidia: [So its like] ... 315 million 360 thousand..yeah that's about a year or two...

Dora: like [gestures in a circle with pen, doesn't draw it.]

Norma: yeah...

Norma: Cause we've got gravitational ..constant and stuff...

Dora: um hum um hum..

Norma: Okay I think that's a good ..

Dora: Run the program...[runs]

The group never discussed the lines of code in the while loop. And the group agreed with Norma's justification that the spacecraft travels around the Earth due to the inclusion of the gravitational constant on Line 6. Although Norma related the function of the `trail` object with its function in another program, it's not clear that the group has dismissed the previous conjecture that the spacecraft travels in a curve. However, the justification of the predicted dynamics of the spacecraft, based on the gravitational constant, provides some evidence that the prediction is somewhat based on the program code and a physics quantity, as opposed to a constraint based command, such as `curve`.

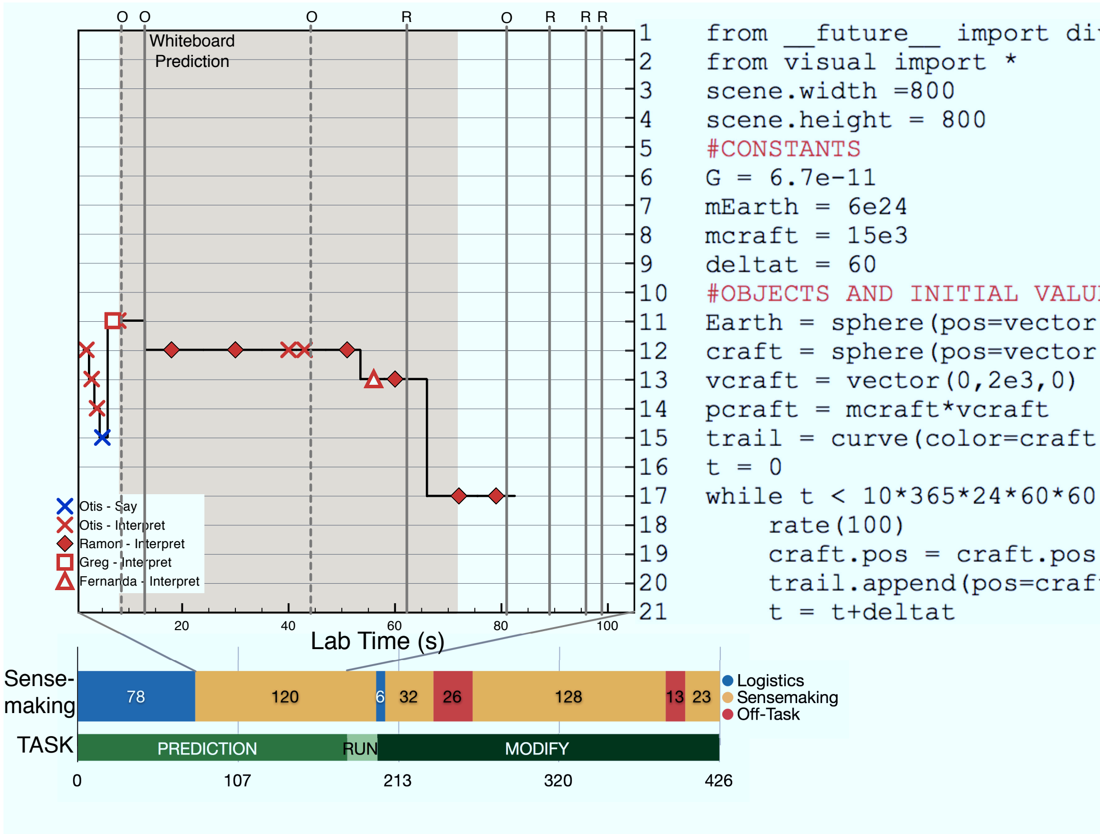


Figure 5.7: Line Number Plot for Ramon, Otis, Greg, and Fernanda as they study and predict the MWP

Making predictions while interpreting code: Ramon, Otis, Greg, and Fernanda

Ramon, Otis, Greg, and Fernanda form an unusually large Group G5. Due to a computer malfunction, Greg and Fernanda joined Ramon and Otis for this assignment. The group combined the study and prediction tasks, to interpret the program code while the group generated a whiteboard prediction. Otis is the Recorder for the group. Figure 5.7 shows the progress of the group as they read through the program code and formed a prediction.

Otis began the study task identifying the function of Line 12, "*Alright, so we have a spacecraft*" and identified the physics quantities represented in Lines 13 and 14 "*there's a velocity,* *and a momentum.*" The rest of the group joined in the interpretation of the code, bringing Otis back to Line 11, which he didn't explicitly mention on his first pass. Otis drew the location of the Earth on the whiteboard and immediately followed the drawing of the Earth on the whiteboard with a prediction statement, "*what is it something traveling around earth or something..*" This prediction appears to serve to suggest a possible outcome in the visual output, rather than a prediction based on the program code. The basis of Otis's prediction is not coming from any attention to mechanism in the program code, but it is possible that Otis was stating an expectation based on his knowledge about the dynamics of an analogous spacecraft-Earth system in the natural world. There's no evidence that Line 6 informed him of the potential for the use of a mechanism to constrain the motion of the spacecraft to travel around the Earth. Ramon dismissed Otis's suggestion and focused the group on the interpretation of the location of the spacecraft. After working on the location of the spacecraft and mapping the spacecraft to the coordinate system on the whiteboard, Fernanda moved on to interpreting Line 13.

Fernanda: It would be, craft would be up, wouldn't it?

Ramon: Yeah...

Ramon: and its velocity is going straight up,

Ramon:so its going up....

Otis: [Draws up arrow at location of spacecraft]

Otis: I'm going to run the program...[laughs]

Fernanda: yeah.[laughs]

Ramon: Ah then it's multiplying by days,

Ramon: hours..

Otis: So yeah it's got to be going around... right?

Otis: or... I dunno.

Ramon: I thinks its gonna... I don't think its not curving around it, it's just shooting straight up...

Ramon: so I think it's going to keep going up..

Otis: okay

Ramon: But you think its gonna curve, it might curve...I dunno

During the exchange following Fernanda's interpretation of lIne 13, Ramon suggested that the spacecraft is going to go up. Notice that Otis was okay with this statement at first, drawing the velocity vector on the whiteboard, and suggested that the group run the program. Ramon continued on with the interpretation of Line 17 and Otis interrupts him, offering his prediction again that the spacecraft moves around the Earth, with more emphasis in his choice of words. Rather than using evidence from the program code to justify Ramon's prediction, Ramon left both predictions on the table. Ramon even restated both predictions before conceded that he couldn't tell which person was correct. Notice the word choice in Ramon's restatement of Otis's prediction. Each time Ramon restated Otis's prediction, he used a word that Otis didn't use: *curve*. Ramon awkwardly stated that "*I don't think its not curving around it, it's just shooting straight up*." Ramon acknowledged the function of Line 13 and rephrased Otis's prediction in terms of evidence in the program code for Otis's suggested prediction. The meaning behind Ramon's statement is not exactly clear, however we have seen that other students interpret the curve 3D object as a constraint on the motion of the spacecraft. One possibility is that Ramon's own internal conflict stems from his interpretation of Line 15 and he considered that the line may serve the purpose of constraining the motion of the spacecraft and therefore may validate Otis's prediction. There's not enough evidence in this conversation to confidently say this is Ramon's reasoning behind the way he acknowledged the two conflicting predictions. However, it is clear that Ramon can't rule out Otis's prediction based on the program code. And, that the group does not interpret the while loop.

Blending the two approaches: Zeke, Paine, and Frank

Zeke, Paine, and Frank form Group G1. The Line Number plot for this group, shown in Figure 5.8, reveals a very different progression through the program code to complete the study/predict tasks. The group first interpreted the program code and generated ideas on what the visual

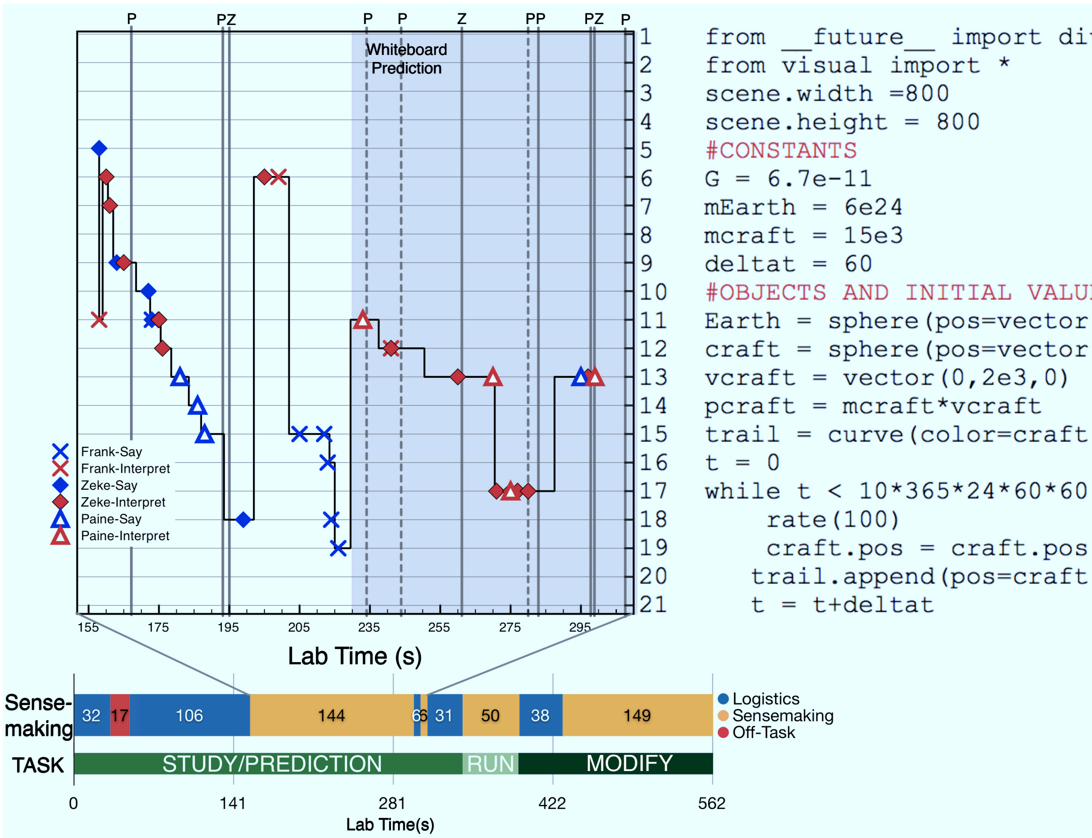


Figure 5.8: Line Number Plot for Paine, Zeke, and Frank as they study and predict the MWP

output will show, then returned to the beginning of the program to formalize their whiteboard prediction.

Zeke is the second participant to begin the interpretation of the program code from the declaration of the physical constants in the `Constants` section. Zeke's interpretation of Line 6 immediately began a conversation about the visual output.

Zeke: You've got gravity,

Zeke: and you've got two masses...

Zeke: and a delta t,

Zeke: change in time.

Paine: So obvious something's falling, ...apparently.

Paine suggested a prediction for what the group would see before leaving the `Constants` section of the program code. Paine used Zeke's interpretation of Lines 6-9, and the assumption that these lines are relevant in displaying the visual output as a basis for his initial prediction. Notice that he formed this prediction based on the program code *and* a connection between the behavior of objects in the real world under the influence of a gravitational interaction.

The group continued on, interpreting that the function of Lines 11 and 12 are to display two spherical objects representing the Earth and craft. Line 15 confused the group, and in the interpretation of Line 15, both Zeke and Paine offered a revised prediction:

Paine:trail curve

Zeke: I think that, what is that?

Paine: Going to be a craft going around the Earth

Zeke: [Gestures on whiteboard with rt. hand] craft going to be going around it or something

Paine: [chuckles] That's what I was thinking.

Paine and Zeke both talk over each other when offering their predictions, which followed a question about the function of Line 15. This is the first time anyone suggested that the spacecraft might follow a path which takes it around the Earth. As the group moved on, Paine raised a question that no one had yet considered in any of the groups:

Paine: What, is gravity in here whatsoever?

Zeke: Yeah, [points to screen] it's right there [points to G = 6.7e-11] idn't it?

Paine: Yeah .. [interrupted]

Frank: It's that gravitational constant.

Paine: I really don't, ... I don't see .. it in any formula on here.

Frank: [picks up pointer and begins pointing to lines of code on screen, saying] alright... trail, [inaudible] [interrupted]

Zeke: Yeah, I don't see it in any .. anything.

Figure 5.9: Paine's position for the spacecraft relative to Earth.

Both Zeke and Frank responded that gravity is in the program, on Line 6. However, Paine indicated that he was looking for where gravity is used in the program to calculate a quantity. Frank didn't acknowledge any issue with this and moved on; however, Zeke verified that the gravitational constant doesn't appear again in the program code. Paine was taking another step towards mechanistic reasoning, to connect the definition of the gravitational constant with its use in the program code. At this point, the group stalled and returned to the prediction task. Paine returned to Line 11 and drew a circle on the whiteboard, without noting any of the attributes of the sphere. Next, Paine drew a spacecraft located above the Earth on the +y axis, taking the location of the Earth as the origin, as shown in Figure 5.9. The group then turned their attention to Line 13:

Paine: which way do you think its going?

Zeke: Looks like its positive?

Zeke: So be going clockwise?

Paine: What's positive?

Zeke: [Points to screen] what's this? This is [inaudible]

Paine: velo..

Since the group decided that the spacecraft was above the Earth on the +y axis, they needed to determine the direction of the velocity by interpreting Line 13. Zeke, who previously agreed that the spacecraft was going to go around the Earth, interpreted the positive y component of the velocity as indication that the spacecraft will follow a clockwise path around the Earth. Zeke didn't convince Paine that the positive value of the y component determines that the spacecraft will travel in a clockwise path, and Paine didn't appear to know where Zeke was getting this information. Zeke didn't stop to explain and moved on to interpreting the series of numbers multiplied together in Line 17. Paine interrupted this interpretation and suggested that the craft-Earth system is analogous to the Moon-Earth system:

Paine: its going...its a moon,

Paine:it's probably the moon dude..

Paine:going around the earth.

Zeke: Yeah, I think that's what it is

Paine: [laughs].

Zeke: A moon going around the Earth.

Paine: Which way is it going?

Paine: It's uh, its, what's its velocity?

Paine: vcraft, 1200 so its just,

Zeke:positive, so it'd be.

Paine: **its going up [points with marker]**

Zeke:[**positive so it'd be**] **going this way right [gestures in clockwise path around Earth**]

Paine: It's positive in the y direction.

Paine was still not comfortable with the direction of the motion of the craft/Moon. Paine and Zeke were talking over each other when each suggested a different initial velocity of the object. Frank suggested that the group make a "*random prediction*" and then run the program. Paine drew the final prediction, shown in Figure 5.10, then preceded to run the program. Paine did not explicitly discuss how the object's motion will evolve. Paine only interpreted that the positive y-component of the velocity indicates that the spacecraft "*is going up.*" It's not clear that he thought the spacecraft will continue to go up or that only the initial velocity is up, without specifying if this velocity will change. Paine relented to Zeke's prediction; however, the stage is set for a discussion about each person's prediction when the program displays the

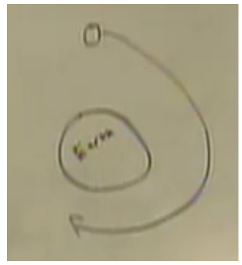


Figure 5.10: Final prediction for the motion of the spacecraft traveling around the Earth.

visual output.

Comparisons of interpretations among groups

Comparing the Line Number plots among the six groups shows that there are differences between which sections of code were interpreted. Table 5.2 summarizes how the groups interpreted the lines of code by looking at the results of the Focus coding pass.

This table identifies how different groups used their physics knowledge to interpret the MWP code. The Quantity coding category was designed to capture when a speaker identified code as

Table 5.2: Interpreting sections of code across all groups

Code Section	G1	G3	G4	G5	G6	G8
`Constants`	Q(All)	–	–	–	–	Q(All)
`Objects`						
`Earth`	F	F,A	F	F,A	F,A,Q	F,A
`craft`	F	F,A	F	F,A	F,A	F,A
`trail`	–	F	–	–	F	F,A
`Initial Values`						
`vcraft`	Q,D	Q,D	Q,D	Q,D	Q,D	Q,D
`pcraft`	–	Q	Q	Q	Q	–
`Loop Condition`	Q	–	–	Q	Q	–
`Loop`	–	–	–	–	–	Q:Line 19

Q: physics quantity, **F**: function, **A**: 3D object attribute, **D**:direction of vector.

representing a physics quantity. Groups G1 and G8 were the only groups who interpreted the lines of code defining the mass of the two objects and the gravitational constant (or "gravity"). All of the groups interpreted that `vcraft` represented the velocity of the spacecraft and its direction. All groups interpreted the function of Lines 12 and 13 to generate two objects in the visual output; however, not all of the groups used the attributes to identify the location of the two objects (G1 and G3). Three of the six groups discussed the function of the `curve` object, with G8 correctly identifying the function of the trail as an object which tracks the spacecraft. The other two groups incorrectly interpret the function of the curve object as a command for the spacecraft to follow a curved trajectory. Groups G4 and G5 do not discuss an interpretation of the function of the curve object, however the choice of using the word "curve" in each group's description of the predicted motion of the spacecraft may suggest that a silent interpretation of this object influenced each group's prediction. Group G1 stated confusion about this line of code. After a brief pause, two of the three group members offered the same prediction, that the craft will go around the Earth.

Only group G8 interpreted lines of code in the loop structure, with Eugene identifying the line of code representing the position update formula. He did not elaborate on the line's function in generating the output. None of the other groups discussed any of the lines of code following the loop conditional (Line 17). If the lines of code past Line 17 were omitted, the visual output would only display the locations of the craft and Earth objects. Participants were not waiting for lines of code to direct the spacecraft to update its position to predict that the spacecraft moves through the visual output. Once groups discussed lines of code which defines the initial values for velocity and momentum, the group discussed how the spacecraft moves through the visual output. Since only one group interpreted a physics quantity in the loop, it will be of interest to track how this changes as groups discuss the other two MWPs.

Prediction statements on the dynamics of the spacecraft

None of the participants specified in their prediction that the spacecraft will travel in a certain direction at a constant speed. Four participants predicted that the spacecraft would travel in a certain direction without mentioning any description of the spacecraft's speed. Of these four participants, Paine was the only one to yield to the competing prediction from Zeke that the spacecraft will travel around the Earth. Howard and Ramon both offered their predictions as possible candidates for the outcome, competing with predictions by others in the group that spacecraft would follow an orbital path. And Eugene offered the only prediction of his group, that the spacecraft would just "*go up.*" Groups were very comfortable with running the program with more than one prediction on the table. This is somewhat curious, since there is only one exact outcome of the program code. Groups were not willing to investigate the program further to rule out one of the two conflicting predictions.

The process through which groups refined their predictions about the dynamics of the objects in the visual output was based on reading and discussing the interpretations of program code, with a few exceptions. Group G1 drew on their knowledge about the dynamics of objects in the natural world when Paine offered a prediction that this MWP models the Moon-Earth interaction. It's not exactly clear if Paine was offering that the program represents this real system or that the dynamics is analogous to this real system. Otis, in Group G5, may have had the same approach, expecting that the program will produce a visual output which resembles the interactions of objects in the natural world, when he says, "*So, yeah it's got to be going around...right?*" There isn't as much evidence to support this possible interpretation of the basis for Otis's statement, since it is possible that the prediction reflects a silent interpretation of the `curve` object.

The `curve` object presents an unforeseen challenge during the prediction task. Either through explicitly justifying the "curving" motion of the spacecraft with Line 15, or using the choice of the word "curve" in the statement of the prediction, nine participants interpret one of the functions of the line of code to constrain the motion of the craft to either go around the Earth or follow a curved path.

Running the program before interpreting the code and predicting the output: Celia, Isis, and Madeline

As alluded to earlier, one group of three participants did not follow the instructions and preceded to run the program before giving any effort at interpreting the program code or predicting the visual output. Although this was greatly disappointing in that it is not possible to compare this group's interpretation of the MWP with that of the other six groups, all is not lost. This group's confusion after running the program code provides some data to what happens to instruction if the group didn't follow the instructional task sequence. And, it's possible to compare the differences in the interpretation of the visual output by a group that skipped the interpretation of the program code with groups who followed the instructional tasks as directed. Upon running the program, the group commented on what they saw in the visual output:

Celia: Alright we gotta run...

Madeline: What is that?

Celia: [whispers] spacecraft.. [laughs]

Celia: Uh.. obviously we need to modify ...

Isis: [closes visual window]

Celia: Okay, what a random diagram...alright what do we have to do?

Isis: Okay, [reading] first draw on the whiteboard how you want the space-craft to interact with the Earth.

Celia: Can we do whatever we want?

Isis: I dunno?

The group was puzzled about what they were looking at and referred to the visual output as a "diagram." Celia suggested that the group's next goal was to modify the program. This idea came from simply viewing the output, not from instructions. Noticing that there was no prediction drawn on the whiteboard, the TA interrupted the group to ask, " *Did y'all predict it?*" He discovered that the group ran the program code without making a prediction. The TA suggested the group move on, following the instructions to modify the program code:

Isis: Okay.. is .. okay. [reading] first draw on the whiteboard how you want the spacecraft to interact with the Earth.

Isis: Do we want it to collide with earth?

Madeline: No.. okay yeah, okay wha?

Celia: We can just have it orbiting..

Madeline: ...Yeah I was going to say orbiting but I don't know if it would be harder to get it to do that.

Celia: I wonder if it would be easy.

Isis: I really don't even know how to even...

Celia: [raises hand as TA approaches] Would it be hard like to make it orbit Earth, would it be really hard [inaudible]

TA: I dunno..

Madeline:Or would it be [loudly]easier to make it collide into the Earth?

Celia: [interrupting, underneath **Madeline**] Is it going to be too hard to modify?

The group was having an extraordinarily difficult time coming to an agreement on how the spacecraft should move through the scene. The group suggested two possible outcomes of the modification task. The group used the perceived difficulty of writing program code to achieve an orbit or collide with the Earth as the criterion for determining which route to take. The group, not sure of the difficulty involved in either exercise, asked for help from the TA. The TA recognized that the groups were lost and attempted to get the group to draw what they wanted to happen and review the program code to identify what was already included and what they might need to add. The details of this interaction involved a back and forth exchange between the TA and Celia, lasting several minutes, and are not germane to this investigation. THe TA attempted to focus the group on the missing interaction and Celia was focused on the behavior of the spacecraft. Due to the significant amount of TA intervention to get the group back on track, it is most desirable to build in additional safeguards to prevent students from running the program before studying and predicting the visual output. The group decided to pursue the goal of getting the spacecraft to crash into the Earth, without investigating the location of the spacecraft or its initial velocity.

5.1.2 Running the program: Interpreting the visual output and evaluating predictions

After participants complete the prediction task, the instructional document asked groups to run the program and evaluate the prediction based on what the participants saw in the visual output. To complete this evaluation task, participants have to negotiate an understanding of what they observe in the visual output. Figure 5.11 shows screen captures of the visual output taken at three different times. The first image was captured shortly after the visual output displays the two objects, with the spacecraft, to the left of the Earth, moving in the positive y direction leaving a trail. The second image was captured as the visual output camera begins to zoom out so that both the spacecraft and the Earth stay in the field-of-view. The third image was captured after a few seconds later with the spacecraft continuing to move in the $+y$ direction with the camera continuing to zoom out. Participants had a difficult time interpreting that the camera was zooming out. Four of the six groups had a discussion about the camera zooming out with three participants in different groups asking if the spacecraft is moving towards the Earth. For example, Group G6 has the following conversation once the visual output appears on the screen:

Dora: Yeah...

Norma: Is it gonna curve?

Dora: No curve? okay go.. [laughter]

Dora: whoa, it curves... oh its ..

Norma: It's not getting closer to it, we're just getting..

Lidia: yeah..its just getting yeah

Norma: [we're just getting] further away...

Dora, expecting to see the spacecraft "curve" around the Earth, which was consistent with

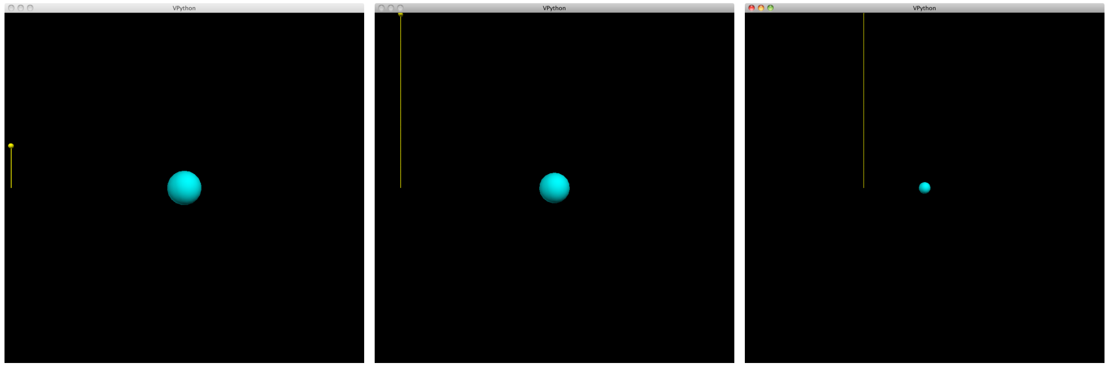


Figure 5.11: Screen shots of the visual output running the Space Voyage MWP code (a)when the visual output first becomes visible, (b) as the camera begins to zoom out to keep the objects in the field-of-view, and (c) some time later.

the group's prediction, initially interpreted the zooming camera as the spacecraft curving around the Earth. This interpretation was quickly overruled by the two group members, and it appears as though Dora might have been rethinking the interpretation after stating it. The other two discussions about the spacecraft appearing to move toward the Earth are very similar to what happened here. And, the interpretation that the spacecraft is moving toward the Earth only occurred within groups who predicted that the spacecraft would orbit the Earth. Three of the groups resolved their confusion and interpreted that the spacecraft moved in the +y-direction. The fourth group required an intervention by the TA to clarify what was occurring in the visual output.

The instructional task was to evaluate the prediction; however, groups took the task further than simply determining if the prediction was correct. While participants were interpreting the program code, three groups discussed conflicting predictions on the motion of the spacecraft before running the program. For these groups, the visual output served as a referee, showing which prediction was correct. For example, in Group G1, Paine predicted that the spacecraft would move up and Zeke predicted that the spacecraft would travel in a clockwise orbit around the Earth. Here's the transcript of what the group said as the program displayed the spacecraft traveling in a straight path:

Frank: ...um... oh no its just going away [laughs]

Paine: It's just going straight up [laughs].

Zeke: It doesn't have gravity,

Zeke: you were right.

Paine: What?

Zeke: Cause it doesn't have the gravity in it.

Zeke stated that he understood what Paine said earlier about using "gravity" in the program. Zeke used evidence from the program code to justify the visual output. Norma, in Group G6, also justifies the motion of the spacecraft based on the omission of "the force from the Earth." Recall, Norma's prediction of the spacecraft traveling around the Earth was based on Line 6, the declaration of the gravitational constant. Howard, in Group G4, also justified the motion of the spacecraft in the visual output based on the program code, saying "cause we didn't tell it to turn or nothing." Taking Howard's statement literally, he expected there to be a command which turned the craft. There are other issues with the MWP code, such as the omission of the update form of the Momentum Principle. None of the participants discussed the Momentum Principle during the prediction and evaluation tasks.

The other three groups indicated if the group prediction was correct and moved on to the next task: modifying the program. Paine, Zeke, and Frank had a discussion after the one appearing above where the group generated the goal of modifying the MWP code before being instructed to do so:

Frank: Alright, so we, [moves pointer to screen] I guess are going to have to [motions with hand] make it rotate... around it.

Zeke: Probably so.

Paine: [laughs, erases clockwise orbit and draws an upward arrow at location of craft.]

[everyone laughs]

Zeke: alright, so what is it say, what is it....

Paine: the only reason you thought it was going around was it said G equals.

Zeke: Yeah, even though

Paine: even though it wasn't in uh.

Zeke: it wasn't in the thing anywhere though.

In addition to Frank predicting the next task, Paine diagnosed Zeke's faulty reasoning which resulted in the group's incorrect prediction. The group began the next task understanding why the prediction was wrong and thinking about which physics concepts might help in achieving an orbit.

5.1.3 Defining the modification goal

The Goal action codes were defined to capture instances when a speaker identified how 3D objects should move through the visual output and instances when the speaker planned modifications to the MWP code. These planning statements come from knowledge about what is already present in the program, the desired behavior of the 3D objects in the output, and the physics principles required to constrain the motion of the objects in a path that is a valid prediction of natural physical phenomena. This section reports how each group navigated from what they saw in the visual output to a statement of the action items required to achieve the desirable result. This section will not focus on the actions taken towards completing the task of modifying the MWP to produce a complete and valid program. Exploring how physics students modify the MWP program and evaluate their modifications using the visual output is important work which needs to be done; however, it is not the focus of the research questions. This section reports the evolution of the thinking of the seven groups (six groups who generated a prediction and the one group who cheated) and identifies if the groups are focused on creating a model which issues instructions to "move in a circle" or creating a model where the mechanism for the desired visual output behavior comes from physics principles.

To complete the modification task, the instructions ask the groups to consider how the objects in the visual output should behave and then identify the nature of the interaction between the two 3D objects. All of the student groups who made predictions before running the program followed a very similar progression from first identifying that the spacecraft should orbit around the Earth, to the physics quantities which are necessary in calculating the gravitational force. Only two of the six groups mention using this gravitational force to update the velocity.

Paine, Zeke, and Frank: Considering where to calculate a force.

Paine, Zeke, and Frank already attributed the error in the interpretation of the program code to the omission of gravity in the program. The conversation picks up from there after the group reads the instructional document aloud:

0:07:00 **Zeke**: [reads] Modify the program with calculations that will allow the program to instruct the spacecraft to do what you want.

0:07:10 **Paine**: You want it to interact with the earth.

0:07:11 **Paine**: You want the gravity to pull it in.

0:07:13 **Zeke**: Yeah, I guess the gravity.

0:07:14 **Paine**: I don't know how...how do you do that now?

0:07:25 **Zeke**: cause tha.. are we supposed to do... wait .. i think ... go back to the code.

0:07:32 **Zeke**: [go back to the code] when you put G in there somewhere.

0:07:35 **Paine**: G is over here...

0:07:36 **Paine**: [G] It's gotta be in here [puts mouse cursor in loop]

0:07:38 **Zeke**: rate

0:07:39 **Paine**: I mean the vector, it's not even going to stop moving. It's going to be moving the same speed right?

0:07:45 **Paine**: Or is it?

0:07:47 **Zeke**: It would make it ...

0:07:52 **Paine**: What's that formula for gravitational force?

0:07:59 **Zeke**: We've gotta make something that's like F-net.

0:08:03 **Paine**: So what we did last week.

At 7:36, Paine suggested that G needs to be used in the loop, that the spacecraft traveled at a constant speed and asked for the formula for the gravitational force. Zeke took this a step further, suggesting that the program needs a line which calculates Fnet and this suggestion triggered Paine to recall the VPython program the group completed last week in lab where they were asked to write the lines of code necessary to calculate the gravitational force acting on a spacecraft at five separate locations in space. This group did not use a loop structure to complete the gravitational force calculation last week. Paine's suggestion that the calculation should go inside the loop is new. However, Paine also suggested that the speed will remain constant and returned to this idea when Zeke suggested to change `vcraft`:

Frank: [points in initial values section] vcraft, vcraft, we have to change vcraft.

Zeke: nuh uh,

Paine: vcraft stays the same doesn't it?

Zeke: That's the initial...

Frank: okay... alright yeah.

Paine has now extended his reasoning that the speed will remain constant to *vcraft*, which defines the initial value of the velocity vector. Zeke skirts the issue by suggesting that they are both wrong since the *vcraft* line defines the initial value of the vector. What to do about the velocity vector remained an open question for the group. The group continued on to add in lines of code before the loop to calculate the gravitational force vector. This analysis will stop here, since the group begins to move beyond what needs to occur to focus on the mechanics of adding code to the program.

Tina, Howard, and Xavier:"What are we changing?"

Tina, Howard, and Xavier began the modification task answering the scaffolding questions in the instructional document. This group did not spend much time during the prediction task reading through the lines of code in the MWP. Both Xavier and Howard returned to the program to identify what was in the program to inform the decision of what to change and what to add to the code. The interesting conversations occurred when Tina asked the group "What are we changing?" This question spurred a conversation between Xavier and Howard which revealed confusions about some fundamental physics concepts.

0:06:22 **Tina**: what are we, what are we changing?

0:06:24 **Xavier**: We need to add...fgrav

0:06:34 **Xavier**: cause we don't have that..

0:06:36 **Howard**: And then, we have to somehow manipulate the velocity of the craft...somehow we have to add it in there right? ... that would be in the x direction?

0:06:46 **Xavier**: we'll the velocity is not going to change, its just going to be affected by ... what it will change, but the velocity is constant, the only reason it's changing is because of the force on it...

0:06:56 **Howard**: Yeah, so speed will be the same but we got to change the x um com-ponent?

0:07:03 **Xavier**: Of the.....

0:07:04 **Howard**: [change the x component] of the velocity...of the craft...right?

0:07:08 **Howard**: Well, cause we're trying to make it go around [gestures]

0:07:10 **Xavier**: I thin,...yeah...

0:07:15 **Howard**: So velocity is the ...direction also as well as the speed...

0:07:24 **Xavier**: Velocity is .... what did you say?

0:07:25 **Howard**: Velocity determines the direction of the...

0:07:31 **Xavier**: Well the think is.. the reason to do ... the reason it just went straight up is there is ... there's no force acting on it.

0:07:35 **Howard**: Yeah.. that's what I'm saying we need to add the force of gravity somehow into the x component of ...cause

0:07:41 **Tina**: and you think..

0:07:41 **Xavier**: It will be ... I'm saying once you find the [points to Fgrav equation]..

0:07:44 **Howard**: yeah yeah yeah yeah I got it, I got it.

Xavier suggested adding the gravitational force and Howard followed up with recognizing that the velocity will change. It's not clear what "it" is referring to in the segment "somehow we have to add it in there right?" If Howard was referring to the gravitational force, then he was trying to build a mechanism for the gravitational force to "manipulate" the velocity of the spacecraft, noting that the gravitational force is in the x direction. There is evidence later at time 7:35 that Howard wanted to use the gravitational force calculation to determine the x component of the velocity vector. Xavier was focused on the vector's magnitude. Xavier was very confused at 6:46 on if the velocity vector changes. The conversation Xavier has aloud to himself is similar to Brian's think-aloud conversation from the pilot data. Both Brian and Xavier struggled with how a velocity vector must change while its magnitude remains constant. Brian's conversation was more sophisticated since he was interpreting the lines of code calculating the position vector and its magnitude. Here, Xavier only had the initial value for *vcraft* and understanding that the addition of the gravitational force was required to achieve the orbit. More important to the progress of the task, Xavier resisted Howard's suggestion of changing the x component of vcraft and offered the alternative action item of calculating the gravitational force since "the reason it just went straight up is there is ...there's no force acting on it." At 7:41, Xavier waved off Howard's idea of modifying the x component of the velocity with the gravitational force, and suggested instead to simply calculate the gravitational force and the calculation will take care of the direction. Following Xavier's strong stand, the group moved forward with adding lines of code to build a gravitational force calculation.

Norma, Lidia, and Dora: Who needs gravity?

Norma and Lidia had a conversation about the application of gravity in this model. After reading the question asking the participants to think of which calculations are required to get the spacecraft and Earth to interact, Norma suggested her answer:

Norma: F gravitation?

Dora: Yeah

Lidia: no, because its not.. gravity wouldn't be applied to it...cause its in space..

Norma: no it would be... the earth.. that's how its going to go around the earth cause of the gravity of the earth..

Lidia: but wouldn't...wouldn't we do Fg is the gravitational would be the big G times mass one

Norma: we need..

Lidia: [continuing] and mass 2.. right..

Norma: we need one of these [draws a vector towards earth from craft]

Lidia: right.

This group's conversation shows that the task of asking what types of interactions are required in the program to model a physically valid prediction of how these two objects move through space get at the underlying physics concepts the computational activities are attempting to address. Lidia raised a concern that gravity doesn't exist in space. In the previous episode, Xavier struggled with how a velocity vector can change while its speed remains constant. Paine and Zeke consider calculating the net forces acting on the spacecraft. None of these participants were thinking of defining a command to "curve" although two of the groups used this word when making a prediction of the visual output.

Selma, Eugene, and Beatriz: Adding the Momentum Principle

Eugene had a clear idea of the sequence of calculations required to achieve the desired goal and appears to think through the necessary additions in a valid algorithmic procedure:

> **Eugene**: okay...wait, draw how we want it to interact... so what do we want it to do?
>
> **Beatriz**: um..
>
> **Selma**: be a circle, i mean
>
> **Eugene**: orbit?
>
> **Beatriz**: yeah...
>
> **Eugene**: [draws a dotted circle around Earth..] [laughs]
>
> **Selma**: Okay so what calculations do we have to do?
>
> **Eugene**: So we need gravitational... force [draws] in and magnitude of that will always be the same but the direction will be changing...but we can just do the momentum update with this force [taps on board to grav vector], right?
>
> **Selma**: the speed might not,... I mean if... well, if we make it circular, it should be the same...
>
> **Eugene**: same magnitude of the speed..but the direction will be changing...
>
> **Selma**: right, the magnitude, okay.
>
> **Eugene**: so, we need f_G [draws on whiteboard], p v, r...so that means we need...r

Selma suggested a circular orbit, prompting Eugene to think about which additional quantities were needed to achieve a circular orbit. He bypassed the concerns raised by participants in other groups about the velocity's magnitude remaining constant with the direction changing by allowing the momentum principle to take care of those calculations. Eugene was the only participant to invoke the need for the update form of the Momentum Principle which, with the gravitational force calculation, is all that was required to change the MWP into a program which models and predicts a physically valid system. Eugene listed on the whiteboard the additional lines of code which needed to go into the loop: the gravitational force calculation, the momentum, the velocity, and the position vector. The group focused on constructing a definition for the position vector in the next phase of the task.

Summary

All six groups who completed a prediction of the visual output identified the gravitational interaction as the mechanism for predicting the orbital motion of the spacecraft. Only two of these six groups considered how to update or change the velocity vector so that its direction changes, with only one group mentioning the update form of the Momentum Principle. It is important to note that all of these groups finished the assignment and turned in a complete and valid program which calculates the gravitational force, updates the initial value of the momentum, and updates the position of the spacecraft using these updated values of the momentum, in an iterative procedure using appropriate time steps to advance the simulation. The procedure commonly used was to add some code and run the program to evaluate the effectiveness of the additions or modifications to the program, based on the visual output. And, three of the six groups who were successful at correctly adding lines of code which calculated the net gravitational force were able to recognize that the spacecraft continued to follow a straight path due to the omission of the momentum principle.

It is interesting that the groups had such a difficult time with calculating the gravitational force and deciding where these calculations belong in the MWP. As mentioned earlier, understanding the difficulties students have with loop structures, keeping track of vector and float quantities and the rules for algebraically manipulating these quantities is an area of active research.

The performance of these participants working in the collaborative groups following a task progression of studying and predicting the MWP, evaluating the prediction, and modifying the program code to achieve a physically valid prediction provides opportunities for participants to apply their physics knowledge. Participants used their physics knowledge to identify physics quantities defined in the MWP, locate the positions of Objects on a 2D projection of a 3D virtual environment, interpret the initial velocity vector's direction and predict how the craft moves through the visual scene. Participants pointed to lines of code to justify their predictions and falsely interpreted the curve 3D Object as a command to the craft to "curve" around the Earth. This roadblock was unfortunate and not predicted to have an effect on participant predictions. Only one group interpreted the lines of code in the loop structure. One feature of the MWPs is that there is very little physics in the program. As a result, the program predicts trivial solutions. However, participants who are not experienced at interpreting physics principles make fanciful predictions motivated somewhat by analogous systems in the real-world (Paine, Zeke, and Frank).

5.2 Mass-Spring MWP

Participants completed the Mass-Spring MWP activity during Lab Week 8, three weeks after the Spacecraft-Earth MWP activity. During weeks 6 and 7, students continued working on the Spacecraft-Earth program to add additional interactions and calculate the energetics of the system. Recall that the MWP was defined as a program that was missing the required interactions or physics principles to predict the dynamics of the modeled physical system. The Mass-Spring MWP fits this definition even though all of the physics principles are included in the computational model. Figure 5.12 shows the MWP, modified to insert line numbers and remove any blank lines.

The Mass-Spring MWP contains the update form of the Momentum Principle (Line 28) and the approximation for the gravitational interaction for a ball near the Earth's surface. As a result, the visual output will show that the ball accelerates in the -y direction. The computational model does not calculate an interaction between the ball and the spring. The

```
1  from __future__ import division
2  from visual import *
3  from visual.graph import *
4  scene.width=600
5  scene.height = 760
6  ## constants and data
7  g = 9.8
8  mball = 1        ## change this to Mass 2 (in kg) for your spring color.
9  L0 = 0.26        ## this is an approximate relaxed length of your spring in meters calculated in lab.
10 ks = 1           ## change this to the spring constant you measured (in N/m)
11 deltat = .01
12 t = 0            ## start counting time at zero
13 ## objects
14 ceiling = box(pos=(0,0,0), size = (0.2, 0.01, 0.2))            ## origin is at ceiling
15 ball = sphere(pos=(0,-0.3,0), radius=0.025, color=color.orange) ## note: spring initially compressed
16 spring = helix(pos=ceiling.pos, axis = vector(0,-L0,0), color=color.cyan, thickness=.003, coils=40, radius=0.015)
17 ## initial values
18 pball = mball*vector(0,0,0)
19 Fgrav = mball*g*vector(0,-1,0)
20 ## improve the display
21 scene.autoscale = 0              ## don't let camera zoom in and out as ball moves
22 scene.center = vector(0,-L0,0)   ## move camera down to improve display visibility
23 scene.mouse.getclick()           ## Simulation doesn't start when you Run Program. Must also click the simulation
24 ## calculation loop
25 while t < 30:
26     rate(100)
27     Fnet = Fgrav
28     pball = pball + Fnet*deltat
29     ball.pos = ball.pos + pball/mball*deltat
30     t = t + deltat
```

Figure 5.12: The Mass-Spring MWP code with line numbers added and blank lines removed.

Mass-Spring MWP is also considerably different from the Spacecraft-Earth MWP in that the modification goal for the Mass-Spring MWP includes more than adding the additional spring interaction. Lines 14-18 place the ball 0.04 m below the end of the spring with an initial speed of 0 m/s. Also, the computational model does not update the axis of the spring and therefore displays a helix with a fixed end "attached" to the ceiling with a dangling free end, as shown in the left figure in Figure 5.13. To prevent the simulation from running before the participants had time to focus on the visual output, a mouse-click command was added to require the participant to click anywhere on the visual output before running through the while loop. Once the participant clicked on the screen, the visual output modeled a ball falling due to the gravitational interaction with the Earth. A snapshot of this simulation is shown in the right figure of Figure 5.13. Notice that there is no *curve* 3D object to track the history of the positions of the ball as it falls.

One motivation for including the Mass-Spring program as a VPython activity is that students measure the spring stiffness and period for the spring they model in the program with

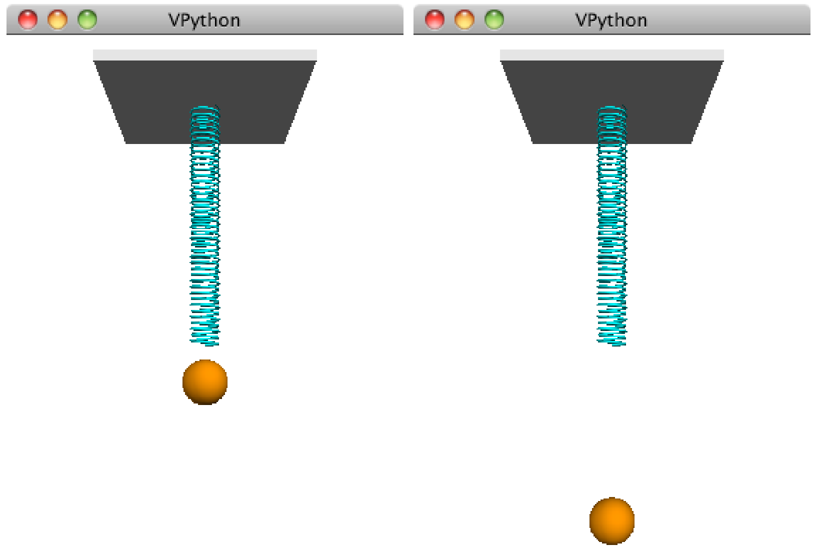


Figure 5.13: Visual output for the Mass-Spring MWP program before the mouse-click (left picture) and after the mouse-click (right picture).

apparatus during a lab activity in weeks preceding this VPython activity. In a best case scenario, students connect the motion predicted by the computational model to the motion of real physical springs. Using a graphing module, students plot the vertical positions of the mass as it oscillates in the computer model to determine the period of the system and compare this value to the experimental value obtained weeks earlier.

The sequence of instructional tasks using the Mass-Spring MWP were identical to those followed by participants using the Spacecraft-Earth MWP. The following sections will report how the participants complete the tasks. Seven groups completed the instructional sequence as directed in the instructions. The eighth group ran the program and then drew "the prediction." Again, even though this group did not study the MWP code and draw a prediction of the visual output before running the program, it is of interest to look at how this group completed subsequent tasks.

5.2.1 Study/Prediction Task

Groups followed the same strategies while completing the study and predict tasks. Four of the seven groups created predictions as they interpreted program code, one group read through the program code first and then returned to the beginning of the program code to draw a whiteboard prediction. However, one of the seven groups began the study task interpreting the computational loop as opposed to the beginning of the program. And, disappointingly, one of the seven groups simply drew a prediction with very little evidence that the group used the program code to inform the prediction. This section reports how each group studied and predicted the visual output, sorted by the four strategies used during the task.

Interpret code to refine a disingenuous whiteboard prediction: Celia and Madeline

Celia and Madeline were two members of the group who ran the Spacecraft-Earth MWP before following the instructions to study the program code and create a whiteboard prediction. These two participants were paired together with a new group member, Tina, who was absent for the lab session. Celia began the task pressuring Madeline to cheat by first running the program and then drawing what they see on the whiteboard.

Celia: Okay copy that into VPython..run it and then predict what its going to do cause then we already know what its going to do..[laughs]

Madeline: [copies code into program]

Madeline: He's going to yell at us again...

Celia: He didn't yell at us last time...

Celia: He didn't yell, do it really quick...

Madeline: [runs program]

The group didn't click the visual output to start the simulation and was puzzled as to what they were seeing. Celia hid the visual output and drew the ceiling as the TA came by. Celia asked for clarification of the task and the TA restated the purpose of studying the program

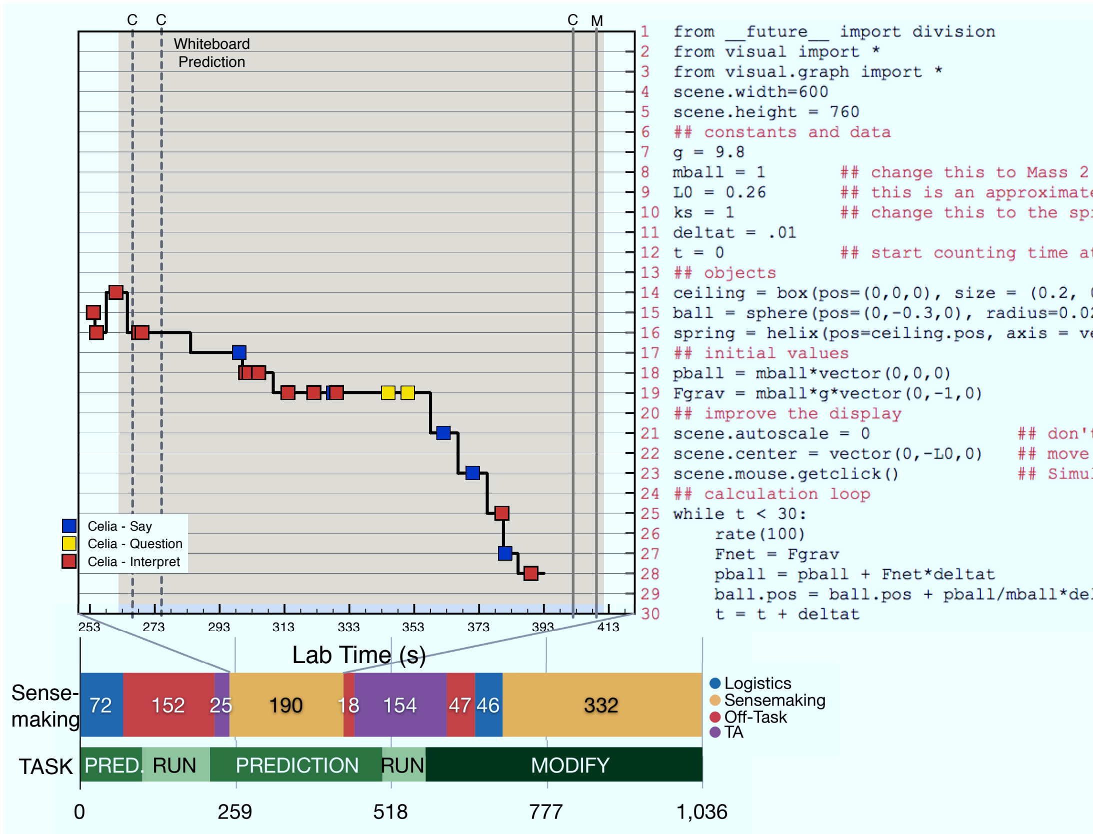


Figure 5.14: Line Number plot for Celia and Madeline as they study the program code and draw a "prediction" on the whiteboard.

code. Celia then completely shifted her focus to the program code and began reading the code to identify features of the 3D objects that she saw on the visual output. Figure 5.14 shows the line number plot as Celia read through the program code while Madeline began drawing the 3D objects on the whiteboard.

The plot shows that Celia and Madeline stuck with the task as clarified by the TA to read through the code and "*try to understand what this program is going to do*." Only Celia attempts to interpret and read the program code for understanding. At first, Celia interpreted the Spring attributes, noting that the spring was compressed. It's not clear what evidence Celia used to state that the spring was compressed. The magnitude of the axis attribute for the spring was smaller in value to the magnitude of the position of the ball, so it's likely that Celia used what she saw in the visual output, that the ball was disconnected from the spring with the spring's free end above the location of the ball:

Celia: The spring goes down from the ceiling.

Celia: The spring is initially compressed..

Madeline: [draws a ball at the end of the spring, attached.]

Celia: So it starts short and then it goes long...

Celia then infers some motion about the spring based on some knowledge about how real springs operate. Celia implies that the spring will eventually move down which was a sensible suggestion, that the virtual spring would behave similar to a physical spring. Celia began interpreting the initial values section of code and ran into difficulty parsing the representation of the gravitational force:

Celia: momentum of ball..

Celia: ball is not moving...

Celia: so the ball is not moving.

Madeline: [writes on board: not moving]

Celia: um...gravitational....force

Madeline: [draws arrow down, writes g]

Celia: Its mass times a g vector,

Celia: so...that's like m g but its like.. or negative, wait.. so negative m g

Madeline: [writes g=-mg] uh..

Celia: Where, so how does negative...where does negative one come from?

Celia: I guess we have a mass...do we have a mass?.

Madeline: Yeah...um..well.. I don't understand how its not.. how its just negative one....

Celia and Madeline had a difficult time with the representation of the gravitational force as the magnitude of the force times its unit vector. Celia was able to identify the symbolic representation of physics quantities, but not able to identify the form of the gravitational

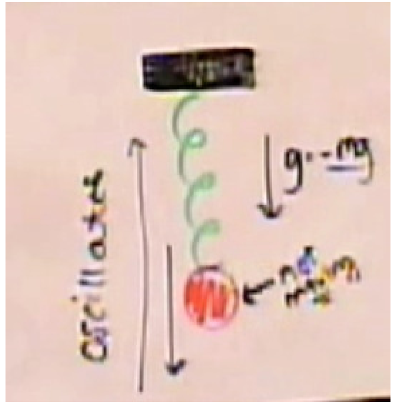


Figure 5.15: Madeline's whiteboard prediction before running the program for a second time.

force. Celia interpreted the initial condition for the momentum of the ball, so thus far, Celia interpreted the location of the end of the spring, the initial speed of the ball, and recognized that Fgrav represents the gravitational force even if she was not able to completely understand how the force was calculated. Next, Celia kept going and found the while loop:

Celia: the net..the fnet is equal to Fgrav...

Celia: so we do momentum.. fi..nal momentum...

Celia: It's probably going to just move up and down...maybe [draws on whiteboard]

Madeline: It's going to oscillate...[writes oscillate]

Although Celia began interpreting the update form of the Momentum Principle, she stopped in the middle of the line of code and didn't try to understand the role of the line of code in the loop. Instead, Celia suggested a prediction of what the visual output might do. Without justifying the prediction with earlier statements, it's not clear if the prediction is based on an understanding of the code or the oscillatory motion of physical springs. Madeline, while drawing the initial locations of the spring, ball, and ceiling, attached the ball at the end of the spring, as shown in Figure 5.15. This is rather curious, since this group already saw the static visual output. Since the whiteboard prediction was drawn connecting the ball to the spring, Celia might have also informed her prediction based on the whiteboard drawing which placed the ball at the end of the spring. Again, there's just not enough evidence to conclusively say exactly what evidence informs Celia's prediction of an up and down motion of the ball. Nonetheless, there are multiple errors in the interpretation of physics quantities (the gravitational force), the interpretation of the momentum principle in the program and it's role in updating the motion of the mass, and an expectation that objects given the name spring will behave like physical springs. Celia is proficient at recognizing the representation of physics quantities in the program code and using the initial values for the attributes of the 3D objects to determine the location of the end of the spring. Although Celia attempts to interpret the loop, the interpretation is one of identification of physics quantities, not putting the physics quantities to use to run through a mental calculation of the function of the lines of code in the computational model.

Interpret code to inform an immediate whiteboard prediction: Isis, Estelle, and Frank

Isis, Estelle, and Frank began the study/prediction task by first stating that the program code seemed overwhelming; however, the team quickly dived into the program, interpreting the program line by line to produce a whiteboard prediction. Figure 5.16 shows the progress of the group as the participants read through the program code. After first interpreting the representation of the momentum of the ball, Isis decided to focus her attention on interpreting the attributes of the 3D objects to inform a prediction:

Estelle: yeah.. okay the ceiling is at the origin..

Isis: Okay so let's say this is the origin..so would

Isis: [draws dot, then line through it]

Estelle: yeah..the ceiling

Isis: okay ceiling, okay

Estelle: The ball is a sphere...

Isis: zero, negative point 3, zero so that's like..like if this is the origin that means the ball is like..

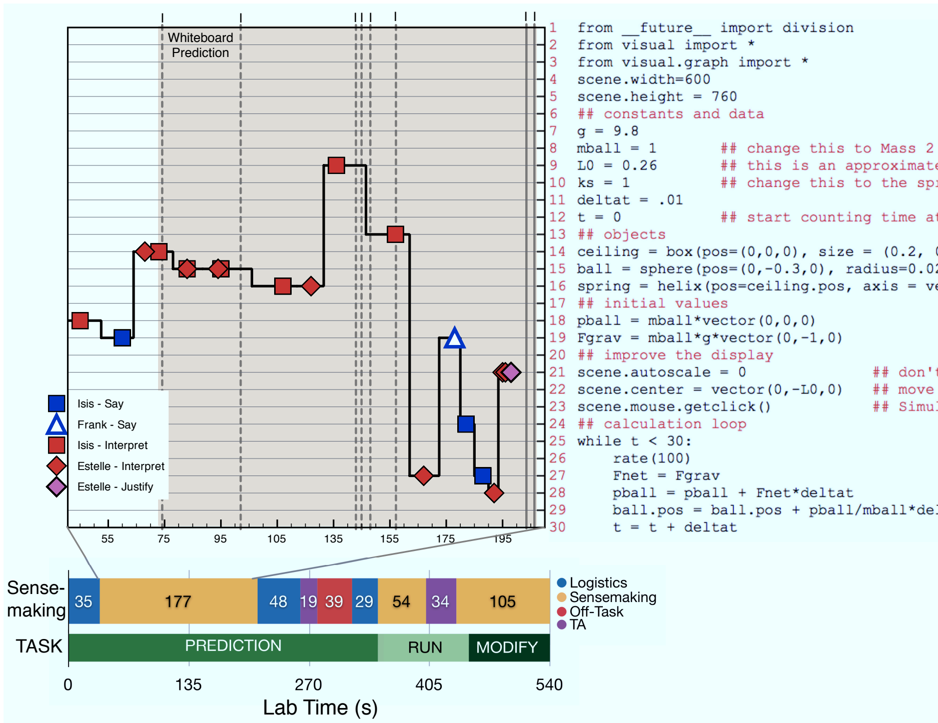


Figure 5.16: Line Number plot for Isis, Estelle, and Frank as they study the program code and draw a whiteboard prediction

Estelle: negative..so its going to be like the y...

Isis: oh negative y you're right.. so its going to be like...here

Isis: [draws position of ball]

Isis and Estelle focus attention on the coordinate system and the location of the ball on the coordinate system. The previous group did not specify the location of the ball and assumed the ball would be located and attached to the free end of the dangling spring. Figure 5.17 shows Isis's drawing of the location of the ball. Isis and Estelle are thus far completely correct on the interpretation of the location of 3D objects, however in the next brief episode, Estelle asks Isis to redraw the prediction much larger. Isis redrew the ceiling, but rather than drawing the location of the ball, she began interpreting the attributes of the spring, which offered some difficulty in determining the orientation of the spring:

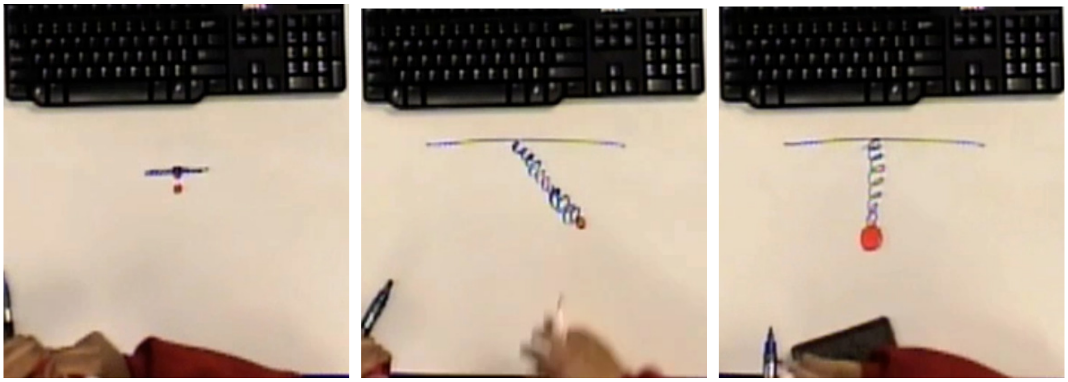


Figure 5.17: Isis drew predictions for the location of the ball (left), a larger prediction for the spring at an angle (middle), and a vertical spring with the ball attached to its free end(right).

Isis: oh yeah..okay so....0,0,0, the spring is the same position as the ceiling..

Estelle: yeah, that's right where it starts I guess..the length is just its original length..not stretched...

Isis: Okay we have that..like .26 so...[laughs]

Isis: I'll just....[draws spring at angle]

Estelle: yeah.. its all good

Isis: And then...the ball [draws a ball at the end of the spring]

Estelle: Do we know that its going sideways though?

Isis: no..we don't [erases spring] so lets just..[laughs]...straight down [draws vertical spring]

Frank: where are we?

Isis: The ball..[draws a ball at end of the spring]

Isis and Estelle used the references to the relaxed length of the spring in determining the length of the spring in the prediction. Isis again had difficulty with the orientation of the spring, drawing the spring hanging from the ceiling at some nonzero angle from the -y axis, as shown in the center image of Figure 5.17. Estelle questioned the interpretation, asking for evidence from the program code that the spring would be positioned as drawn on the whiteboard. Isis corrected the drawing by aligning the axis of the spring along the y-axis (right image of Figure 5.17), and added the ball to the end of the spring. The group was very close to recognizing that the position of the ball is not the same as the position of the end of the spring, and therefore not attached to the spring. The group moved on to briefly discuss an initial value and then interpret the function of some lines in the while loop:

Isis:let's look at the calculation loop....

Isis: Fnet equals fgrav..so

Estelle: so...momentum is updating,

Estelle: so its moving,

Estelle:velocity or whatever

Estelle:..see its says don't let...don't let camera zoom in or out as ball moves..

Isis: I think its just going down and up...[gestures with marker on board] and like..

Estelle: yeah..

Isis: down..oscillating..

Estelle: yeah, yeah, I think so too. I don't know if there is more to this than that.

While Isis focused on the net force, Estelle interpreted the function of the update form of the Momentum Principle as updating the ball's momentum. Estelle interpreted the function of the Momentum Principle, which is more than recognizing and identifying its presence in the program. It's not exactly clear that Estelle's interpretation of the Momentum Principle or the update of the ball's position on the next line informed her next statement, "so its moving." Since Estelle justified the statement two segments later with text that appeared in a comment at the end of Line 21, there's no direct evidence that the motion that Estelle predicted was based on the computational loop. Isis proposed her prediction that the ball will oscillate without offering any evidence from the program code to support her prediction, and Estelle also agreed.

This group was very close several times during the interpretation of the code at developing a prediction of the correct locations of 3D objects and their motion. The group never discussed the missing interaction of the spring or that the gravitational force was the only force acting on the ball, although there's evidence that Isis read the line of code and was about to say something before being interrupted by Estelle. Although we don't know Isis or Estelle's train of thought or have data from a think-aloud protocol, there is data on how the two students refine each other's interpretation of the program code and prediction of the visual output. Estelle offered many comments to question Isis's interpretation of the program code or prediction. Estelle and Isis worked together, both reading through lines of code offering their interpretations as opposed to two group members interpreting different sections of code at the same time. Thus far, this is the first instance where a participant interpreted the purpose of the Momentum Principle, although Estelle didn't complete the calculation loop by commenting on how the updated momentum informs the new position of the ball. Similar to the Spacecraft-Earth program, there's an artifact of the program code which unintentionally offered information that affected the interpretation of the program: the comment explaining the purpose of the `scene.autoscale` line of code.

Interpret code to inform an immediate whiteboard prediction: Ramon, Selma, and Dora

Ramon, Selma, and Dora began the study/prediction task by interpreting the representation of "*the gravitational constant, the mass of the ball, the length of a relaxed spring,* [and] *spring stiffness.*" Ramon identified these quantities without mentioning the assigned values until interpreting the initial value of time, where he identified that time begins at zero. Figure 5.18 shows the rapid pace of interpretations through the Constants section with a shift in focus as Ramon approached the Objects section of code. Ramon turned his focus to creating a prediction in a manner very similar to Zeke, Paine, and Frank from the Spacecraft-Earth program, where it's not clear that Ramon interpreted the attributes of the 3D objects, only the function of the

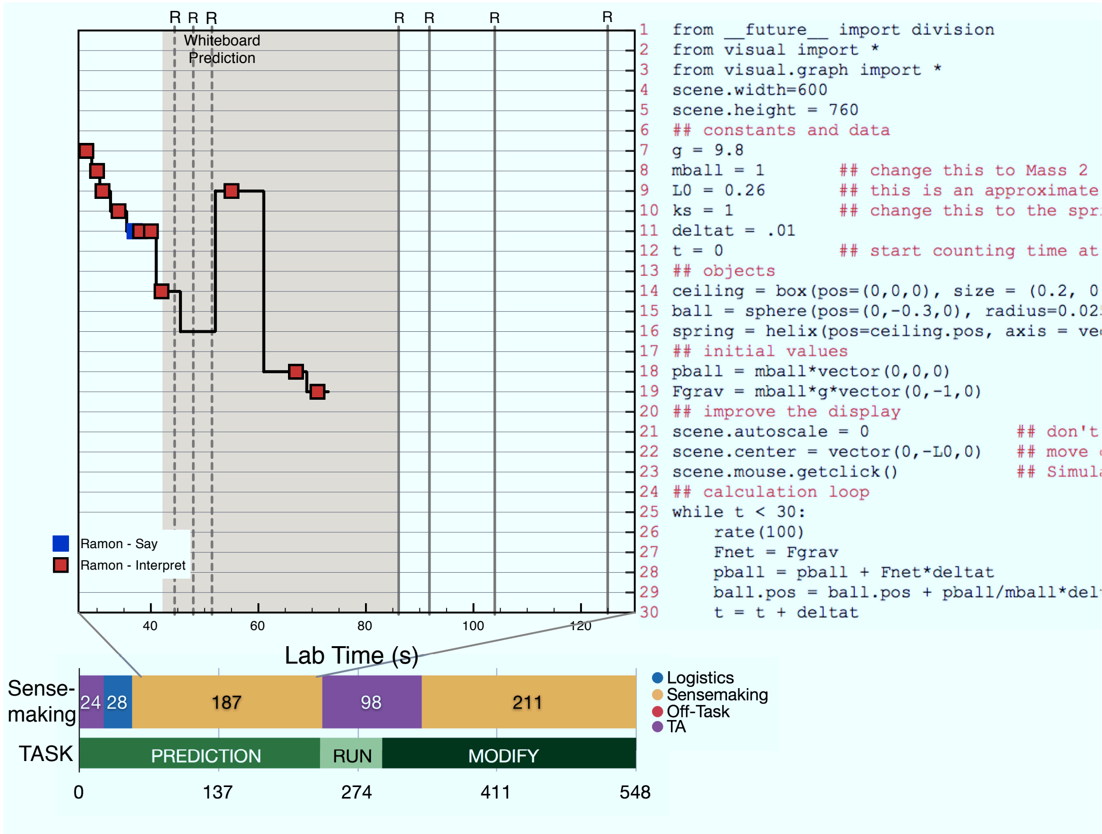


Figure 5.18: Line Number plot for Ramon, Selma, and Dora as they study the program code and draw a whiteboard prediction.

line of code to display the object on the visual output:

Ramon: so...we've got...ball.. or uh ceiling an origin,

Ramon: [draws on whiteboard a line representing the ceiling]

Ramon: spring coming down to a ball

Ramon: [draws a spring starting from the ceiling, draws a ball at the end of the spring]

Ramon: and this length is... point 2-6 [writes .26 on whiteboard besides the spring]

There's no evidence that Ramon interpreted the ball being attached to the end of the spring

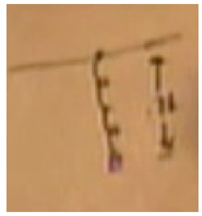


Figure 5.19: Ramon's prediction for the locations of the ceiling, ball, and spring. The writing off to the right of the helix is ".26"

based on a false interpretation of the program code, however he did note that the spring is .26 in length. Ramon may have correctly identified the length of the spring based on the relaxed length value, not the attribute for the axis of the spring. The only direct evidence of Ramon interpreting the spring's length occurred in the constants section where the relaxed length of the spring is defined. Lacking any specific evidence from Ramon that he was interpreting an object attribute, this segment was coded as an interpretation of a physics quantity, the relaxed length of the spring, as opposed to the interpretation of the helix object's axis attribute having a magnitude of 0.26. Ramon's prediction for the locations of the ceiling, ball, and spring are shown in Figure 5.19.

Ramon moved on to the initial values section of the code and developed his prediction for the motion of the ball, neglecting any mention of lines of code in the computational loop:

Ramon: the momentum of the ball is zero initially

Ramon: and the force of gravity is going down...

Ramon: [draws arrow down, writes Fgrav]

Ramon: so then..its going to go down..

Ramon: it should just go down....

Ramon: it doesn't ever calculate a spring constant...

Ramon: so I think its just going to go down..so..

Dora: um hum

After identifying the calculation of the momentum of the ball and the direction of the gravitational force, Ramon offered his prediction that the ball falls. His prediction didn't specify how the motion of the ball might change, for example that the ball's speed increased. He justified his prediction based on the omission of the calculation of a spring constant. It's not clear what he means by this, since he identified `ks` in the constants section as the "*stiffness*." It may be that Ramon was thinking of a spring force, not a spring constant, but there's no evidence for that hypothesis, until Selma asked the following question:

Selma: it's not going to come back up?

Ramon: well I don't see why it would....cause then they would have to put in a potential...and this is ..

Ramon: I just think it is going to keep going down...

Ramon: yeah, cause it doesn't ever..doesn't have a force to pull it back up..

Selma asked a great question, one that was on target with her role as the Skeptic for this assignment. The question asked Ramon if he had considered that the ball might come back up. For data purposes, the answer to the question reveals what Ramon had not considered or looked for in the program code until Selma called for evidence for Ramon's reasoning. Ramon formulated a concept of a required potential that would have been necessary in order to have the ball move upward. It's not clear what type of potential to which Ramon was referring. If he was using physics knowledge consistent with the textbook and lecture, he may have been thinking of a gravitational, electrostatic, or spring potential energy. Notice that Ramon paused, restated his prediction, and then determined that the program omitted the calculation required to pull the ball upward. Comparing to Ramon's justification for his prediction of the spacecraft traveling in a straight line, this justification identified a missing physics quantity that he associated as a necessary calculation for the behavior that Selma suggested. With the Spacecraft-Earth MWP, Ramon couldn't take this step. He couldn't look at the program code to identify why the spacecraft would not orbit the Earth.

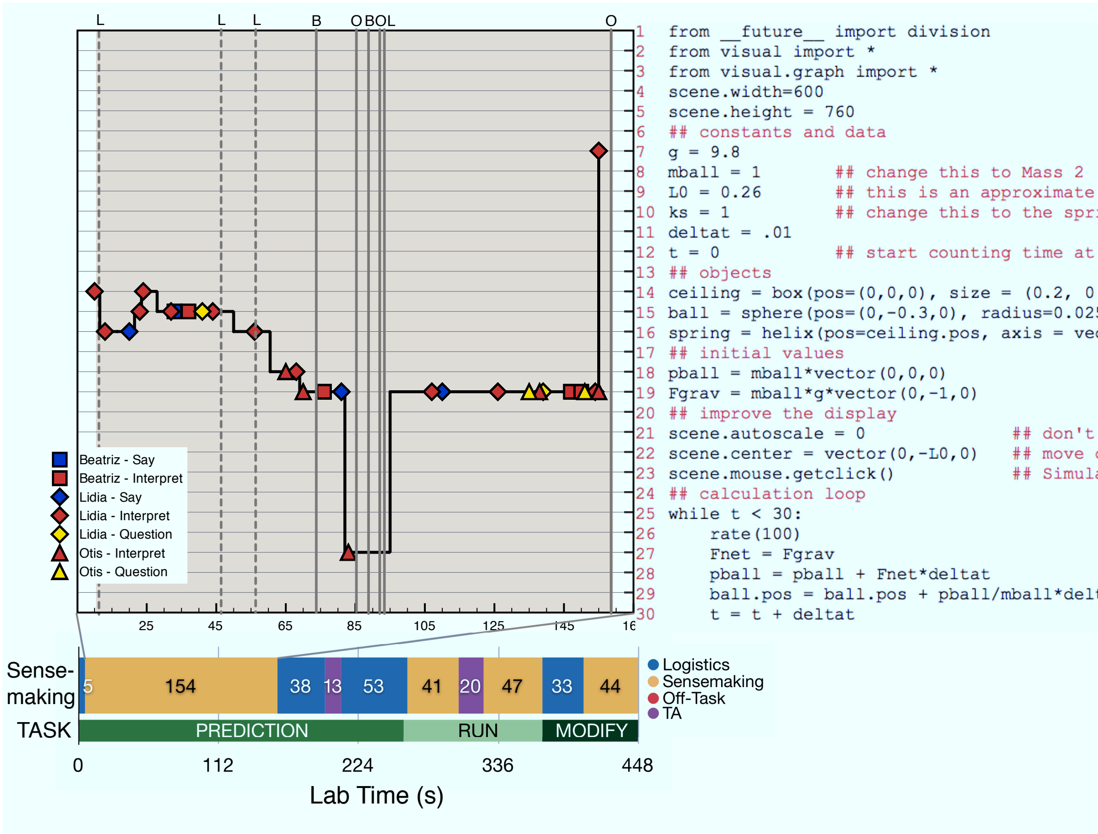


Figure 5.20: Line Number plot for Lidia, Beatriz, and Otis as they study the program code and draw a whiteboard prediction.

Interpret code to inform an immediate whiteboard prediction: Lidia, Beatriz, and Otis

Lidia, Beatriz, and Otis began the study/predict task with the objects section of the program code with the goal of using the information about the location of 3D objects to inform a whiteboard prediction. Figure 5.20 shows the progression of the group as the three students interpreted and added elements to the whiteboard prediction.

Lidia began the task focused on the location of the ceiling, ball, and spring, not the constants. Therefore it's not surprising that the drawing for the predicted locations of the end of the spring connected the spring to the ball. The group took turns discussing the location of the ball on the coordinate system and came to the conclusion that the ball hung below the center of the ceiling. This discussion, as well as a similar type discussion from the previous group, reveals that mapping the location of 3D objects is not a trivial task for participants. In both instances, the error in the geometric interpretation was due to flipping the x and y components of the attribute for the ball's position.

The group moved on to the initial values section of the program code and used this information to generate their prediction of the visual output:

Otis: momentum is zero..

Lidia: so its not moving...

Otis: gravity... course....um..

Beatriz: so is it going to go down or something,

Beatriz: cause of the force of gravity.

Lidia: is this *inaudible*..this is negative one..

Otis: total force is just gravity...

Otis: yeah...I think its going to ..[gestures towards his body with index finger on the table] its just going to go inaudible..

Beatriz: So maybe it's going to go down and come back up and then go down[gestures on whiteboard]

Otis: yeah..its like [gestures oscillation with hand on table]

Lidia: oscillate?...[draws double headed arrow on whiteboard]

The group interpreted the code with the explicit goal to generate a whiteboard prediction. The group doesn't attempt to understand every line of code, instead the group discussed the functional lines of code that they believed would have an effect on the dynamics of the ball in the visual output. Beatriz suggested that the ball would "*go down*," which was a sensible idea with the amount of code the group had interpreted. However, the group did not interpret the position update formula in the loop and were inferring motion based on the initial value of the

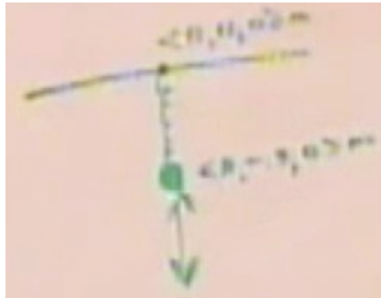


Figure 5.21: Whiteboard prediction for Lidia, Beatriz, and Otis. The text above the line representing the ceiling is "$\langle 0, 0, 0 \rangle$", and the text besides the helix is "$\langle 0, -.3, 0 \rangle$"

gravitational force. The prediction worked since their assumption that the force would interact with the ball to update it's motion was correct. Notice that Beatriz suggested a completely different motion of the ball: that the ball would fall, rise, and fall again. Beatriz's prediction completely shifted the group's focus from creating a prediction based on the program code to a prediction based on a physical mass-spring system. The addition of the arrow indicating the group prediction of an oscillating ball is shown in Figure 5.21. In the final episode prior to asking the TA to evaluate the prediction, the group discussed the representation of the gravitational force:

Lidia: Where is the vector...that vector come in though..where is that?

Beatriz: for what?

Lidia: this is like..the mass of the ball times g times the vector zero negative one zero

Lidia: ..what is that...

Lidia Is that ks, maybe?

Otis: oh that's just....are you looking at Fgrav[points to screen]?

Lidia: yeah

Otis: Initialized? That's just how you get it...mass times ...gravity..

Lidia: Yeah but what's the vector?

Beatriz: Like, which way its going to go.

Otis: yeah yeah,...

Beatriz: Cause its going to go down...

Otis: isn't that..isn't that what it is ..I mean

Lidia: negative g...

Otis: neg..negative m g

Lidia: yeah neg g yeah g's up there..its 9.8...

Lidia: okay..Yeah..thats okay

Lidia had a difficult time understanding the purpose of the unit vector and asked a question similar to Celia's from another group. Again, there exists evidence that the representation of the gravitational force as a product of its magnitude and unit vector caused confusion about the function of the unit vector in the calculation. Since Lidia spent some time talking about the negative g quantity, she may have been confused about whether the value of g was negative. Lidia insisted on thinking about the negative as something that was associated with the acceleration due to gravity rather than a direction of a unit vector and reported that she was okay after she saw that the constant g was defined as 9.8.

The group's final prediction before the group ran the program consisted of a ball connected to a spring, hanging vertically 0.3 meters from the center of the ceiling and oscillating after first falling. The only indication that the group interpreted the while loop was when Otis said the total force was "*just gravity*" yet he, and his group inferred that the ball would oscillate.

Study loop first to inform prediction: Norma and Fernanda

Norma and Fernanda began the study/predict task inside the while loop. Figure 5.22 shows the progression through the program code and the construction of a whiteboard prediction. This is the first group that began interpretation inside the loop. Norma justifies beginning in the loop by saying:

Norma: so this is where its doing stuff[cursor moving back and forth over while loop]

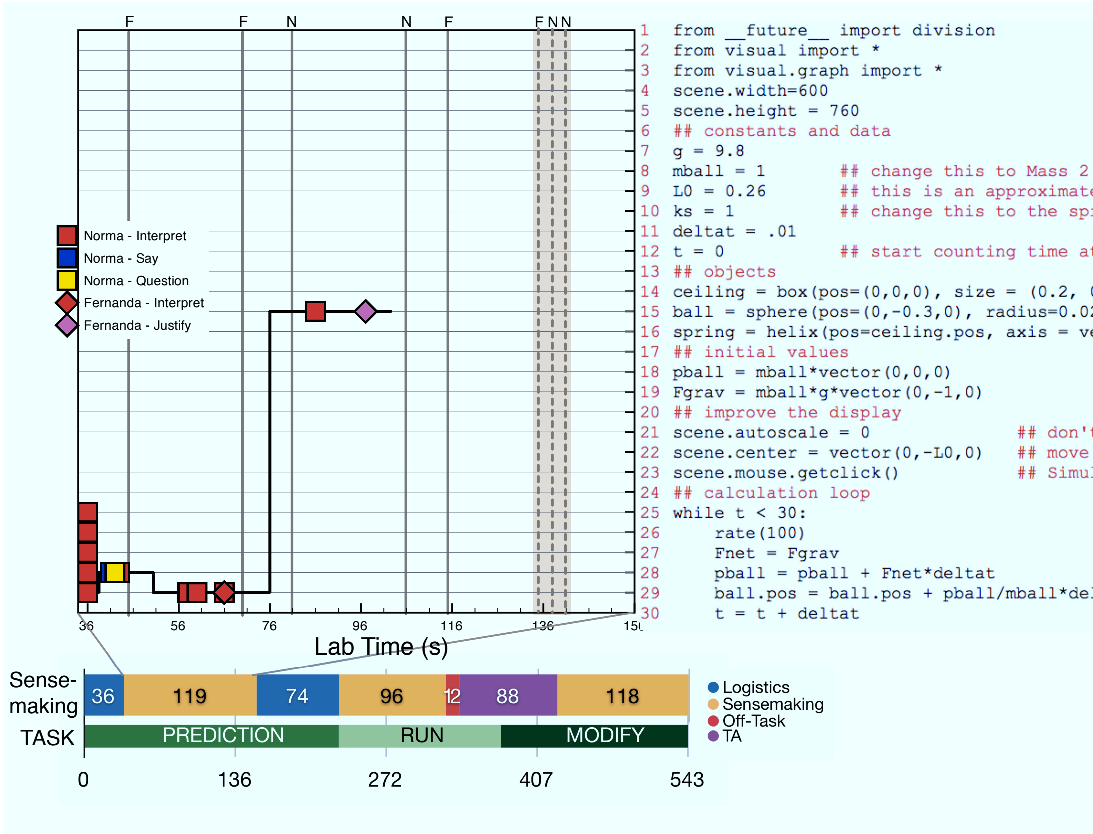


Figure 5.22: Line Number plot for Norma and Fernanda as they study the program code and draw a whiteboard prediction.

Fernanda: it's going to...

Norma: pball.....

Norma:what's pball..

Fernanda: its going to..

Norma: momentum of the ball?

Fernanda: It's going to go back and forth and when it gets lower its going to stretch....that's my guess...

Norma: p...m...

Norma: okay this is finding velocity [looks at position update]

Norma: so..the ball..

Fernanda: that means its got to be moving, right?

Norma: velocity times time...

Norma: yeah so it will move..

Norma began interpreting the loop, attributing the loop as the part of the program that controls the visual output. Notice Norma didn't interpret the net force calculation, rather moved to interpret the Momentum Principle. As Norma interpreted the physics quantities in the line of code, Fernanda began suggesting possible outcomes of running the code. Fernanda's suggestions are specifically not based on program code nor did Fernanda have any interest in reading the code to make sense of what was happening. Norma determined from the position update that the ball had to be moving in the output from the calculation of the velocity of the ball. In the next segment, Fernanda asked a question about whether the motion of the ball would change:

Fernanda: so like it will be swinging back and forth? you think?

Norma: um..it will swing?

Fernanda: i dunno...

Norma: or would it just stretch.

Fernanda: it could just stretch...

Norma: ball is just in the negative y....

Norma: does ball have a velocity?

Fernanda: yeah, its got to stretch..

Fernanda:it doesn't go back and forth...because it doesn't have the..x....no...if it just has a y...then..

Norma: it will just go down...you think it just stretches down?

Fernanda: yeah, let's just guess that one..

Fernanda corrected her earlier guess that the spring would swing back and forth, but it's not clear if she corrected the guess from interpreting the code herself, or using Norma's comment that the ball "*just in the negative y.*" Norma later added her prediction that the spring would stretch downward; however, there's no indication that Norma interpreted the gravitational force. The conversation between these two participants is scattered and unfocused. Norma didn't talk to Fernanda about what she understood from the program code, only the parts that confused her. As a result, it's not clear what features of the program code informed the prediction. During this last episode, Fernanda added an additional feature to the prediction due to Norma's question:

Norma: do you think it will go up and down?

Fernanda: Well its got to go back up...

Norma: yeah..its got to go back up... okay..

Norma: [draws] so you want to say it will initially go ..

Fernanda: down..

Norma: down...

Norma: [draws ball at end of line]

Norma: [draws arrow pointing down]

Norma: inaudible ball...up

Norma: [draws second ball to right of first, with up arrow]...[laughs]

Norma: nice..

Although the group began the interpretation task in the computational loop and thinking about the functions of the calculations in the loop, the task evolved into a description of how a physical ball and spring would interact. Fernanda didn't consider that the program might not model a nonphysical system. Much was missed in the interpretation of this program; however, Norma correctly identified the role of the computational loop and the orientation of the spring.

Giving up on interpreting the program: Yolanda, Xavier, and Zeke

Yolanda, Xavier, and Zeke began the study/predict task modifying the values for the quantities defined in the constants section of the code. This was not the appropriate time to modify these values and the instructions did not mention modifying these values until after completing the study, predict, and evaluate tasks. Figure 5.23 reports the progress of the participants through

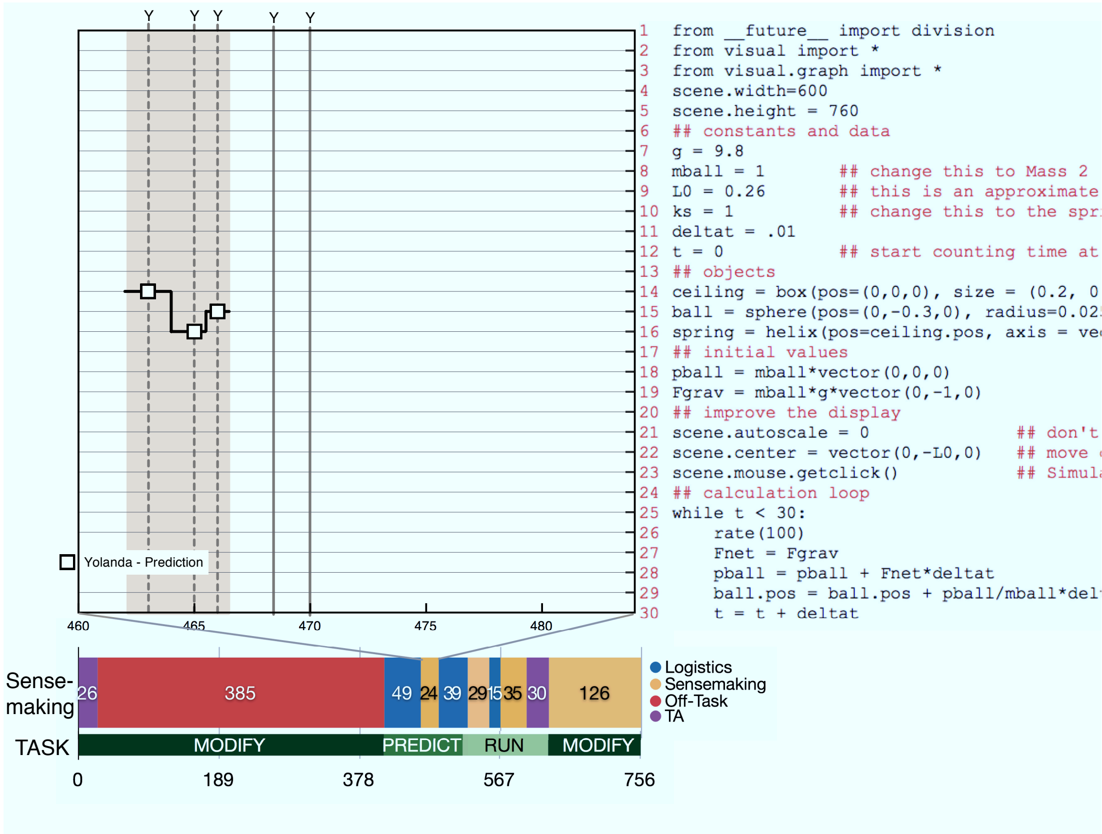


Figure 5.23: Line Number plot for Yolanda, Xavier, and Zeke as they study the program code and draw a whiteboard prediction.

the activity. After the group changed the values for the mass, spring stiffness, and relaxed length of the spring to match physical springs and masses, the group moved on to the study/predict task. Yolanda was quite confused, thinking that the group had already completed the tasks asked before running the program for the first time, and it took direction from the TA to get the group back on track:

Yolanda: Okay predict what you expect to see when you run the program...

Yolanda: I mean I think we're supposed to run the program right now...

Zeke: Oh, we should run the program?

Yolanda: yeah..

TA: Did y'all do this stuff here?

TA: did you explain it?

Xavier: We we were just talking about it..

TA: Make sure y'all do this...and then run it...and then we've got to do this thing here too.

Zeke: We've already changed the program..

Xavier: It's gonna..[gestures oscillation with apparatus]

Zeke: Does it matter that we've already changed..

TA: [To **Xavier**]Is that based on what you saw in the program? Or is that based on how a spring works?

Yolanda: removes additions to code...

This group was the only group who modified the program code before attempting to form a prediction of the visual output. The instructions explicitly laid out the sequence of tasks to complete; however, the program code does offer hints in the form of comments to help students identify what values to change during a later task. Confusion may have come from reading these comments too literally as the participants studied the constants section of the code. The TA caught the problem after he noticed that the group was talking about running the program code although there was no prediction drawn on the whiteboard. Xavier suggested a verbal prediction and the TA challenged the prediction's evidence by questioning the source of Xavier's prediction. This question was not enough to redirect the group's focus back to the program code:

Yolanda: okay, so, what do you predict to see?

Xavier: What you think?

Yolanda: I think there's going to be a ceiling,[draws a line]

Yolanda: [there's going to be] a spring,[draws a spring starting from the ceiling extending down the y axis]

Yolanda: there's going to be a ball [draws a circle at the end of the spring]

Yolanda: and its going to oscillate

Xavier: just one?

Yolanda: yeah, just one..just one ball.. and its going to oscillate up and down..

Xavier: Alright, cool..

The line number plot marks the location of Yolanda's predictions for the position of the 3D objects and the motion of the mass-spring as corresponding to focusing on Lines 14, 15, and 16. This is not an accurate interpretation of what happened, since it is not clear that Yolanda used the information from these lines of code to inform her prediction. The data points on Lines 14,15,16 indicate elements of the prediction which correspond to the function of these three lines of code, which is why the data series is called Yolanda - Prediction. This group only discussed the values of the mass of the ball, the spring constant, and the relaxed length of the spring while modifying those values to match the experimental values determined in a previous lab. It is possible that thinking back to the experimental lab framed an expectation that the program would model the behavior of this physical mass-spring system and therefore the prediction was based on what the group observed during that lab. Again, there is no evidence to back this claim other than an observation of what the activity of the group before stating a prediction. The group analyzed data from a previous lab where they measured or calculated values for the physical properties of a mass-spring system and inserted these values into the program code. As an effect of this modification, it was reasonable for the group to expect the program code to model the same situation if they thought the task was to modify the program with values so that the program and physical system match in behavior.

Read the program for comprehension, then create a whiteboard prediction, almost: Max and Eugene

Eugene began the study/predict task using the same strategy he deployed when he completed the Spacecraft-Earth MWP activity: read the program first for understanding then return to lines of code that had a direct effect on creating and updating the visual output. Max was absent

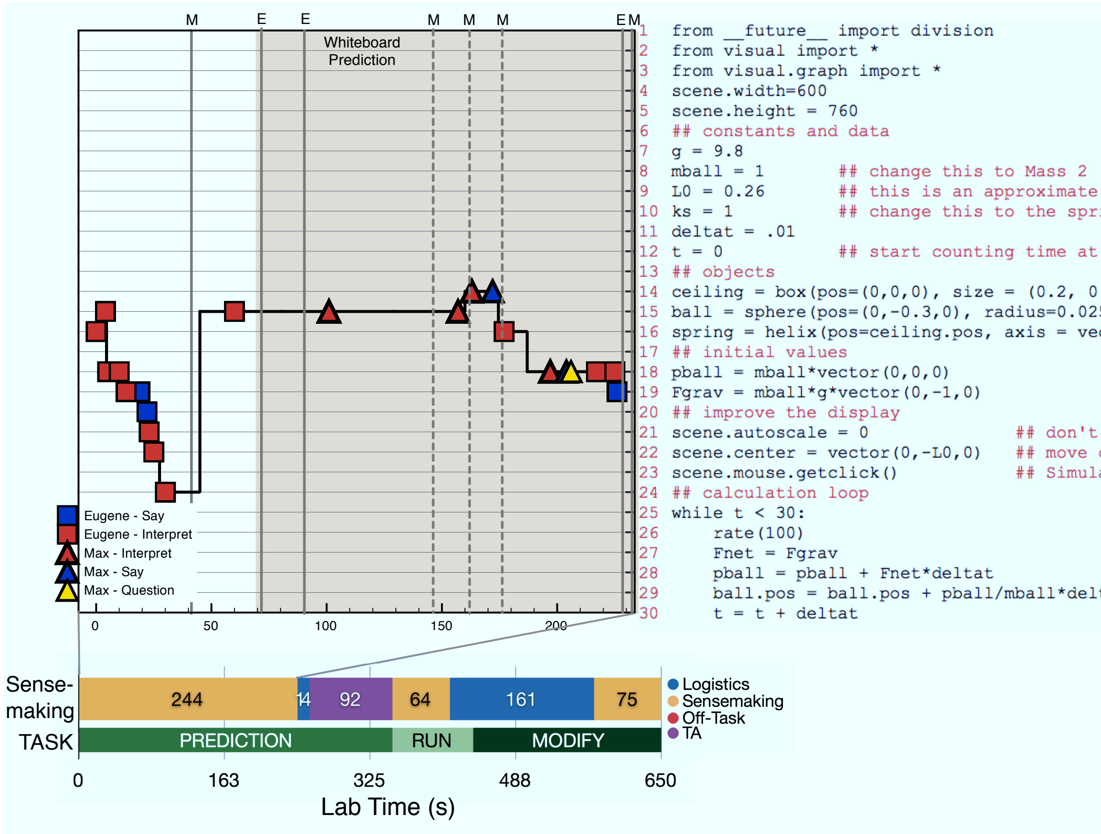


Figure 5.24: Line Number plot for Max and Eugene as they study the program code and draw a whiteboard prediction.

during the Spacecraft-Earth MWP activity, so this was his first attempt at the instructional sequence. Max didn't shy away from his inexperience or the ample experience demonstrated by Eugene at interpreting program code. Figure 5.24 shows the progress of the group interpreting the program and drawing a prediction of the visual output. Eugene began the interpretation but stopped due to Max's question about what would they see when they ran the program code. In the following episode, Eugene read through lines of code interpreting the function of each line with respect to the objects in the visual output:

Eugene: okay ..so we've got spring attached to the ceiling..

Eugene: a ball at the end..

Eugene: they got masses...

Eugene: the ball has an initial momentum..

Eugene: it has gravity on it...wait..

Eugene: g times the vector..okay um..

Eugene: improve the display

Eugene: so it autoscales..

Eugene: [so it] centers..

Eugene: [so it] and waits...oh you start it by clicking..start the simulation by clicking...that's cool.

While Eugene studied this code, he used language different from the previous MWP. Each of these segments add information about the 3D objects. Eugene didn't identify the gravitational force calculation, he specifically said "*it has gravity on it.*" Eugene paused after interpreting the calculation of the gravitational force. Assuming that Eugene was reading the line from left to right, the pause occurred at the end of the line of code, where the unit vector defined the direction of the gravitational force. He didn't comment any further on the line and moved on to the section of code used to control the camera's field-of-view and the beginning of the simulation. Eugene's interpretation of the physics quantities consisted of identifying the purpose of the line of code as adding a quantity to the 3D objects. He didn't begin the interpretation in the constants section. Eugene made a major error in the interpretation of the position of the ball by placing it at the end of the spring. It's not clear if Eugene used an incorrect interpretation of the program code or an expectation that balls are attached to springs to inform this statement. At this point in the study task, Eugene entered the loop to interpret the program further; however, Max interrupted and asked Eugene a question about the dynamics of the ball when he presented a scenario that was observed in the experimental lab:

Eugene: okay in the loop...

Max: what are we supposed to predict? that its going to start by moving and then its going to circle around like..inaudible [picks up pen and gestures motion]

Eugene: yeah that's probably like..

Eugene: [reading] inaudible draw a digram of your prediction..

Eugene: so it starts at..zero negative point three zero..

Eugene: its going to yeah bounce around like...

Max: [draws coordinate axis on whiteboard]

Eugene: its going to go kinda circular and going up and down, right?

It's peculiar that Eugene, who had up to this point in this activity and the prior activity based his prediction on the program code, agreed with a prediction that was loosely based on the MWP code. Eugene didn't hold this standard to Max and didn't state (nor did he need to) evidence for why he agreed with Max's prediction of an oscillating and circular motion. Max's prediction was creative in the sense that it added a circular motion component to the ball's motion. Eugene and Max shifted their focus from thinking about this possible motion to drawing the prediction the agreed upon on their whiteboard. While discussing the location of the ball, the group had a lengthy conversation about the differences between coordinate systems used in calculus and physics. Once Max decided that the y axis was the vertical axis, he drew the ball on the whiteboard and commented that the ball was 0.3 below the ceiling:

Max: [draws a ball along -y axis]

Max: so it starts there..then...origin is the ceiling..okay that makes sense

Max: ball sphere...pos radius color..

Max: spring's going to look like this...

Max: [draws spring starting at ball ending at origin]

Eugene: the spring's initially compressed...

Max: so the spring is going to look like..

Eugene: then around...

Max: cheating as always [talking to Fernanda]

Max: mass of the ball times the vector...do they define vector?

Max: says...oh..vector zero, zero, zero.

Max: um...does that mean a vector pointing to zero, zero, zero?

Eugene: no its just starting out with an initial momentum of zero...

Max: oh..okay...

Eugene: gravity is down,

Eugene: so fnet equals just F grav.

Eugene:..so it goes down...

Max: so it shouldn't be moving around at all...it just goes straight up and down..

Eugene: yeah I guess...

This is the second time that a participant interpreted that the spring was compressed. There's no evidence that Eugene was interpreting a line of code since what he said doesn't correspond with a correct interpretation of the value of the axis of the spring. Since the statement corresponds with an incorrect interpretation of Line 16, this statement was represented on the Line Number plot. Again, a Line Number plot shows when a statement corresponds with the function or interpretation of a line of code, even though the speaker may not have been looking or interpreting that line of code. If Eugene was interpreting Line 15, he may have come to this conclusion comparing the values for L0 to the position of the ball, without interpreting L0 as a physics quantity representing the relaxed length of the spring.

Max also had difficulty interpreting the vector for the initial value of the momentum of the ball. Curiously, Max interpreted the vector as a vector that pointed in the direction of the indicated coordinate in space, not the vector providing information about the initial momentum of the ball. Eugene and Max worked out that since the initial momentum of the ball was zero and that the net force was the gravitational force, the ball "*goes down.*" Max, insisting that the ball must come back up, agreed that the ball was not going to move around. Eugene reluctantly agreed with Max.

The TA was walking near the table at that moment and Max called the TA over to explain his prediction to check for completeness and said, "*it starts here...it's compressed..gravity is the only force on it..its going to go down..spring is going to pull it back up..its going to oscillate.*" The entire discussion was summarized in this explanation of the group's prediction. The initial location of the ball, Eugene's interpretation that the spring was compressed, Eugene's interpretation that the only force acting on it was gravity, the initial motion of the mass falling and that *the spring pulls the ball up.* Max doesn't definitively equate the interaction with the ball and the spring to one characterized by a well defined force. Max clearly stated the expectation that the spring behaved like a real physical spring.

Comparisons of interpretations among groups

Comparing the Line Number plots among the six groups shows that there are differences between which sections of code were interpreted. Recall, two groups did not interpret the program code. Group 240_G1 drew a prediction based on what they saw in the visual output and group 240_G3 didn't provide any evidence of interpreting a line of code. Table 5.3 summarizes how the groups interpreted the lines of code by looking at the results of the Focus coding pass.

Table 5.3 shows that most groups interpreted the function and attributes of the 3D objects. Groups were using evidence from the code to generate predictions about the objects and their locations in the visual output. However, the interpretations were not necessarily correct or complete. Although groups G4 and G8 interpreted the function and attributes for all of the 3D objects, they did not correctly identify that the ball was at a different location than the end of the spring.

The table also shows a successful interpretation of the physics quantities defined in the initial values section of the program including the gravitational force and its direction. There was no consistency in the interpretation of the calculation loop. Only one group discussed the interpretation of the position update formula for the ball. One group interpreted the function of the update form of the Momentum Principle to update the ball's momentum. And two of the groups identified that the only force acting on the ball was a gravitational force. Group G8 did say that "*so fnet equals just f-grav*," but this was coded as a verbalization of the line of code and not an interpretation. There's simply not enough evidence in this statement of an interpretation consistent with how other groups discussed the function of this line of code.

5.2.2 Running the program: Interpreting the visual output and evaluating predictions

Once participants were pleased with their predictions, the instructions asked the groups to run the program code for the first time and evaluate the prediction, just as before with the Spacecraft-Earth MWP. However, after the groups were asked to evaluate the prediction, they were asked to interact with the TA and explain how to recreate what they saw in the visual

Table 5.3: Interpreting sections of code across all groups

Code Section	G2	G4	G5	G6	G7	G8
`Constants`	–	Q(`L0`)	Q(All)	Q(`g`)	–	Q(`mball`)
`Objects`						
`ceiling`	F,A	F,A	F,A	F,A	–	F,A
`ball`	F	F,A	F	F,A	A	F,A
`spring`	F,A	F,A	F,A	F	–	F,A
`Initial Values`						
`pball`	Q	Q	Q	Q	Q	Q
`Fgrav`	Q,D	Q,D	Q,D	Q	Q	Q,D
`Loop Condition`	F	–	–	–	–	–
`Loop`						
`Fnet`	–	F	–	F	–	–
`pball`	Q	F	–	–	Q	–
`ball.pos`	–	–	–	–	Q,F	–

Q: physics quantity, **F**: function, **A**: 3D object attribute, **D**:direction of vector.

output in the real-world. The goal of this instructional task was to ask the participants to relate what they saw happen in the virtual world to the behavior of objects in the natural world and, with the apparatus at the table beside them, recreate the visual output with the materials on hand. Groups had mainly ignored the apparatus until asked to use it in reproducing the output. Since the ball's location was below the spring in the visual output, the correct solution was to place the ball below the free end of the spring and simply release the ball. Not all of the groups completed this task and many times the TA interrupted the group to ask for their answer to the question. This section will also report participant's answers to this question, as well as the interpretation of the visual output and the evaluation of the whiteboard prediction.

Interpreting the visual output: Celia and Madeline

The TA was present while Celia and Madeline clicked on the screen to begin the simulation and asked some followup questions asking the group to relate the visual output to a similar situation in the real-world:

Madeline: [initiates mouseclick]

Celia: ..the spring didn't go with it..

TA: The spring didn't go with it

TA:..the ball just..

Celia: fell off..

TA: So how would you do that in lab? With this equipment I have conveniently set up..how would you recreate that

Madeline: I would... not connect it.[releases mass from rest below spring] [laughs]

TA: Exactly..good, good good job okay so we're at 100%, good job y'all.

TA: okay so guess what we've got to do now...

Celia: Make it work.

TA: make it work..yes..

Celia: of course

This episode is similar to the interpretation of the visual output by three other groups who, like Celia and Madeline, did not comment on their prediction after viewing the visual output. All four groups mentioned that the ball was not attached to the spring and that the ball only fell. Madeline had no problem envisioning a method for recreating the visual output with the apparatus at the desk. The group ended the evaluation task with an understanding that the next task was to modify the program to "*make it work.*" The remainder of this section takes a closer look at how the remaining three groups interpret the visual output and evaluate the prediction.

Evaluating the output: Isis, Estelle, and Frank

Isis, Estelle, and Frank predicted that the visual output would display an attached ball at the end of the spring which would oscillate up and down. Upon running the program, Isis made a false comparison between what the participants saw in the visual output and the group prediction: '*okay...well that's what we said it would be but its not moving...*" The group didn't interpret the `scene.mouse.getclick()` command in the program and were deeply puzzled as to why the program wasn't producing a time-varying output. Isis didn't comment that the ball's position was below the free end of the spring. Frank asked if the program included a variable for time and when Isis clicked the screen, the simulation ran, startling the group. Isis justified the event as an issue with the compiler running slow and decided to run the program code once more. On the second run, the group had the following conversation:

0:16:36 **Frank**: It's not even connected..it's floating there...

0:16:46 **Estelle**: hm...it doesn't just fall..its weird

0:16:53 **Isis**: it should be moving..

0:16:56 **Isis**: especially cause it has position update and everything..

On the second run, Frank identified that the position of the ball is not at the end of the spring with Estelle still puzzled about why the ball didn't fall. Isis went as far as confidently stating that the visual output is incorrect due to the presence of code in the program. The interpretation of the function of the position update formula was not a conversation topic while the group was interpreting the program code during the prediction task. This was the first occurrence of a participant questioning the computer generated output based on her understanding of the function of the program code. Up to this point, participants typically questioned their understanding of the program code based on what they saw in the visual output. The group called the TA over for help, which took the TA to direct the group to the interpretation of the getclick command in the program code.

Evaluating the prediction: Max and Eugene

Max and Eugene were quick to interpret the prediction and evaluate their prediction after the TA explicitly asked for justification for what happened in the visual output:

0:24:11 **Max**: our ball just fell..not even attached to the spring..

0:24:22 **TA**: so you saw what happened...why?

0:24:26 **Eugene**: the position update's just based on gravity

0:24:32 **Max**: and it also is not attached to the spring when it starts...

0:24:36 **Max**: that's not good..

0:24:40 **Eugene**: fix. it...

0:24:43 **Max**: how would you do this in lab?

0:24:46 **Max**: glue...

0:24:49 **Max**: we don't have to write a program in lab to make a spring work...

0:24:55 **Eugene**: drop it...let go...

0:25:02 **Max**: you really wouldn't like have to write a program to be like..force spring..it kind of happens..you need glue for the lab...

Eugene justified the output from evidence in the program code that the update of the position of the ball is only due to the gravitational force. However, Max stated that the ball was not attached to the spring. It's possible that Max thought that by simply attaching the ball to the spring the program would display an oscillating ball. Later in the episode, Max restated this idea by comparing the differences in a virtual environment and the natural world. Max explicitly stated that he would have to add in the interaction of the spring to the program whereas the natural world has these interactions are built in. This is the first piece of evidence that Max considered that a spring in the program doesn't inherit the properties of physical springs in the natural world. Up to this point, it wasn't clear that Max made this distinction between the two environments. And, it's entirely possible that Max didn't consider the distinction until this point in the lab. Again, this is Max's first attempt at interpreting a MWP program.

Evaluating an understanding of the program code: Fernanda and Norma

Fernanda and Norma were deeply concerned about why they didn't predict what occurred in the visual output. Norma returned to the program code to search for an explanation for why the ball and the spring were not connected:

0:20:08 **Norma**: it doesn't look good...

0:20:10 **Fernanda**: that's different..

0:20:11**Norma**: The ball just falls

0:20:15 **Fernanda**: what? [laughs]

0:20:20 **Norma**: how would we know that?

0:20:33 **Norma**: Why is this so..

0:20:35 **Fernanda**: how did it just fall...It doesn't make any sense..

Norma returned to the objects section of the program code to search for why the end of the spring and the ball are not at the same location:

0:21:23**Norma**: Ball....

0:21:25 **Norma**: ceiling....okay ...

0:21:29 **Fernanda**: Where's the end of the spring?

0:21:30 **Norma**: end of the spring.[looks to the end of the spring object line]

0:21:32 **Fernanda**: what is....

0:21:33 **Norma**:...radius .015...

0:21:36 **Norma**: but also it would be..down [gestures]...

Only two groups of the eight returned to the program code to identify lines of code which might have explained the visual output. Norma couldn't figure out the discrepancy between the fixed end of the spring and the position of the ball. Norma also had difficulty with the location of the spacecraft while interpreting the Spacecraft-Earth MWP. This was as far as the two got before the TA came over to offer assistance. The discussion between the group and the TA focused the group on explaining why their prediction didn't match the output with the group reporting the positions of the two objects were not the same.

5.2.3 Defining the modification goal

The first priority of the modification goal by all groups was to move the ball or the end of the spring such that they were connected in the visual output. In order to keep the ball and the spring connected during the simulation, the groups had to add an additional line of code to the while loop which recalculated the 3D spring's axis to stretch to the location of the ball. In order to have the 3D helix object respond to changes in its length according to Hooke's law, participants had to add lines of code in the while loop to calculate the force acting on the ball due to the spring's stretch. The instructional document also asked participants to gather the data for the physical spring they used in a previous lab (L0, mball, ks) and replace the default values in the constants section of the program code with these experimentally determined values. This section reports the initial focus of each group as they begin the modification task. Unlike the Spacecraft-Earth MWP, where the modification task was well defined, the number

Table 5.4: Initial focus of groups beginning the modification task

Group	Constants	Ball/Spring Position	Attach Spring to Ball	Spring Force
G1*	X		X	X
G2		X	X	
G3				X
G4	X	X		
G5*	X	X	X	
G6	X	X	X	
G7		X		
G8	X			X

* Considerable TA intervention at the beginning of the modification task

of alterations to both initial conditions, the values for physics quantities, the attributes of the spring, and defining the force due to the spring was a lot for the groups to juggle. Table 5.4 shows the initial focus of the groups after evaluating the whiteboard predictions.

Groups G3 and G8 modification plan only included updating the force acting on the ball due to the spring. Both of these groups stated the need for this calculation and identified that the calculation should go in the while loop. Both groups also stated during the conversation that although the calculation should go in the while loop, there needed to be an initial value for the calculation. For example, here's G3 talking about where to put a calculation called "Fspring":

0:23:35 **Zeke**: So we need to put the force of the spring somewhere in there..

0:23:38 **Zeke**: It would be underneath the while..

0:23:40 **Xavier**: Put it in the loop

0:23:40 **Zeke**: Yeah.

0:23:42 **Yolanda**: yeah

0:23:43 **Xavier**: but you have to define it first...

0:23:44 **Yolanda**: Ah...*inaudible*

0:23:45 **Xavier**: Ah...

0:23:47 **Yolanda**: k sub s is right here though [points to ks in constants section]

0:23:49 **Xavier**: well

0:23:50 **Zeke**: So then we just do the..

0:23:53 **Xavier**: do what Fspring is..

0:23:54 **Zeke**: We need to put it with Fnet..don't we...

0:23:57 **Xavier**: Yeah, put it with Fnet...

0:23:59 **Zeke**: We need to put, we need to define..something up here the Fspring..

0:24:01 **Xavier**: Right there, initial values...Fspring, right there...

0:24:07 **Yolanda**: [Types in initial values section Fspring=]

Not only did the group have a conversation about the location of the Fspring calculation, the group also inferred some relationship between Fspring and Fnet. This exchange noting the conflict in where to define Fspring was very similar to the conversation between Max and Eugene in group G8 with the exception that Eugene was comfortable with putting the calculation in the initial values section *and* the while loop. And, Max added the variable name Fspring to the Fgrav in the line of code representing the net force on the ball.

Group G1 initially focused on thinking about how to connect the ball to the end of the spring, not specifying that the ball should be at the same location of the end of the spring, but the ball and spring should be in contact. That is, the group wanted to fix a problem with the visual output, not the physics in the program code, until Howard suggested otherwise:

0:14:53 **Roslyn**: How do you connect it?

0:15:04 **Roslyn**: [reading in section two] when you connect the ball..

0:15:11 **Howard**: [points to instructions]change... wait

0:15:18 **Paine**: [reads from section 3] Change the initial position of the ball to zero negative point 5 zero..

0:15:26 **Roslyn**: Well it says you've got to connect the ball..to it...do you want me to check it out?

0:15:30 **Howard**: Yeah...

0:15:36 **Roslyn**: So, how do we connect it?

0:15:48 **Howard**: I don't know...

0:15:55 **Howard**: We've got to think of uh..vector force don't we? for the spring? cause the only force[points on screen] working on it now is gravity...on the ball...

0:16:08 **Roslyn**: So you have to put spring constant?..

0:16:13 **Howard**: I think.. I dunno cause spring isn't doing anything right now...

0:16:16 **Howard**: I don't think you can just say ...connect ball to spring....

0:16:22 **Roslyn**: How come it doesn't tell you how to do that?

Howard completely shifted the group's focus from connecting the ball to the end of the spring to thinking about the interaction of the spring on the ball. The group took this suggestion further and began formulating a calculation for the spring force in the while loop. The other five groups who did not initially mention the missing spring interaction were focused on moving the position of the ball or the free end of the spring. This is not an appropriate use of physics knowledge to modify the program to achieve the desired goal. All of these groups eventually finished the program which produced a simulation of a ball oscillating at the end of a spring. However, the initial modification focus was to fix a flaw in the initial setup of the simulation, not the addition of physics calculations.

Summary

All groups predicted that the ball was connected to the end of the spring. When the groups ran the program and saw that the ball was below the spring, the groups shifted their focus to fixing this error rather than using their physics knowledge to modify the program to include the spring interaction. As a result of this data, a new version of the MWP was created for future use, where the MWP code was altered to move the spring to the location of the ball and include a line of code in the while loop to keep the free end of the ball at the location of the ball during the simulation. As a result of these modifications to the MWP code, TAs report that students are focused on adding the interaction of the spring on the ball to the program. We've learned from this activity that a MWP should only omit the physics calculations we want students to focus on during the modification task. Giving students a program that would require modifying the initial conditions in addition to the computational loop distracted the students from the instructional goals of the activity.

There was a substantial shift to interpreting lines of code in the computational loop to either inform a prediction or evaluate the visual output. And, although a group may spend most of the time interpreting the computational loop, the focus didn't necessarily produce a prediction based on the function of those lines of code. All but one group predicted that the ball and spring would oscillate. There are two explanations for this: 1) participants were predicting the behavior of a physical mass-spring system in addition to interpreting the initial locations of the 3D objects and 2) participants associated physical spring-like properties to a 3D object called "`spring`," meaning that not only was the program going to automatically create a helix, but it was also going to provide attributes for the 3D object to interact with stuff connected to it. Curiously, no group mentioned anything about the ceiling's mass, the ceiling falling with the ball, the spring's interaction with the ceiling, etc. Participants seemed to be comfortable with the ceiling having nothing to do with the simulation, which was confirmed in the visual output.

5.3 Rutherford Scattering MWP

Participants completed the Rutherford Scattering MWP activity during Lab Week 10, two weeks after the Mass-Spring MWP activity. The name given to the instructional document was "Modeling a Collision" and not Rutherford Scattering, so that the name of the activity would not cue students to think about the Rutherford Scattering experiment before attempting to interpret the program code. However, there is evidence, presented later in this section, that by calling the activity "Modeling a Collision" participant predictions included features of a collision between two hard spheres. The Rutherford Scattering MWP was very similar to the Spacecraft-Earth program such that both programs omit the calculation of any interactions and the update form of the Momentum Principle. Both programs also calculate the and update the position of one of the objects based on an initial value of the velocity of a single moving object. These two programs differ in how the initial velocity was calculated. In the Spacecraft-Earth MWP, the initial velocity was simply stated as a vector with numerical values for the x, y, and z components. In the Rutherford Scattering MWP, the kinetic energy of the Alpha particle was provided and the velocity of the Alpha particle was calculated from this quantity using the mass of the Alpha particle, and assigning the initial velocity to the +x - direction. Figure 5.25 shows the MWP, modified to insert line numbers and remove any blank lines that appeared in the version provided to participants.

The alpha particle begins to the left of the gold nucleus which appears in the center of the visual output. The initial velocity, as mentioned above, points in the +x direction, which will take the alpha particle through the gold nucleus. Since no interaction between the alpha particle and gold nucleus is calculated in the computational loop, the alpha particle continues through the gold particle and stops on the other side as the computational loop completes its last iterative calculation. Figure 5.26 shows the progression of the time-varying visual output for the MWP provided to participants. The left image shows the location of the alpha particle (red) and gold particle (yellow). The center image shows the position of the alpha particle before it reaches the gold nucleus. And the right image shows the position of the alpha particle at the end of the simulation. The year 2011 is the 100th anniversary celebration of the discovery of the atomic nucleus credited to Ernest Rutherford. James Marsden, a student of Rutherford and Geiger, was asked to run an experiment to detect the scattering angles of charged particles deflected by a gold foil using a scintillating screen coated with zinc sulfide. The visual output of the MWP, as given to participants, shows the result which corresponds to the predicted outcome of the standard model for matter at the time of what is now referred to as the gold-foil experiment: that most particles would travel straight through the gold foil and emerge on the

```
from __future__ import division
from visual import *
from visual.graph import *
##Constants
massAu = (79+118)*1.7e-27
massAlpha = (2+2)*1.7e-27
qAu = 2*1.6e-19
qAlpha = 79*1.6e-19
oofpez = 9e9
deltat = 1e-23
##Objects
scene.width=600
scene.height=600
Au = sphere(pos=vector(0,0,0), radius=8e-15, color=color.yellow)
Alpha = sphere(pos=vector(-1e-13,0,0), radius=2e-15, color=color.blue)
trailAu = curve(color=Au.color)
trailAlpha = curve(color=Alpha.color)
#Initial Values
pAu=massAu*vector(0,0,0)
K_Alpha = 10*1e6*1.6e-19   #10 MeV in Joules
pAlpha=vector(sqrt(2*K_Alpha*massAlpha),0,0)
t = 0
while t<1e-20:
    rate(1000)
    Alpha.pos=Alpha.pos+(pAlpha/massAlpha)*deltat
    trailAlpha.append(pos=Alpha.pos)
    trailAu.append(pos=Au.pos)
    t=t + deltat
```

Figure 5.25: Rutherford Scattering MWP code provided for participants with line numbers added and blank lines of code removed.

other side, with little deflection. The path of this alpha particle would correspond to a particle traveling through the nucleus with zero net effect on the trajectory of the particle..

The sequence of instructional tasks for the Rutherford Scattering MWP were identical to those followed by participants using the Spacecraft-Earth MWP. All eight groups completed the instructional sequence as directed on the instructional document. None of the groups cheated.

5.3.1 Study/Prediction Task

The Line Number plots for the Rutherford Scattering MWP activity shows groups interpreting lines of code following two strategies: 1) creating whiteboard predictions while reading the

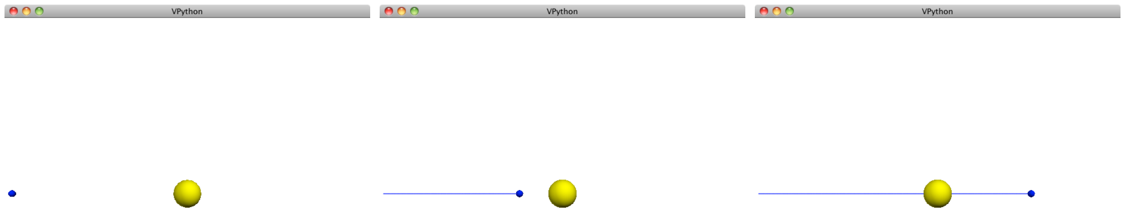


Figure 5.26: Visual output for the Rutherford Scattering MWP program at the beginning of the simulation (left picture), as the alpha particle approached the gold nucleus traveling at a constant speed (middle), and after the alpha particle passes through the gold nucleus(right).

program code and 2) reading a few line of code for understanding then return to the whiteboard prediction task. Fewer groups read the program code in a top-down fashion, meaning fewer groups began interpreting the beginning of the program and worked their way down line-by-line. Groups often jumped between sections of code in their discussion providing evidence that participants were interpreting lines of code that either provided the most difficult lines to interpret or the code that directly informed their prediction. It's also possible that participants were chunking lines of code together according to their function and jumping between these sections. There is not enough evidence to justify this explanation without the participants explicitly talking about grouping lines of code together by function.

Interpretation of the position update formula: Zeke, Roslyn, and Isis

Zeke, Roslyn, and Isis discussed very few lines of the MWP code. Figure 5.27 shows that not only did the group interpret very little, the group focused their discussion on the line used to update the position of the alpha particle. The group began noting that masses of the two objects were defined and corresponded to an "*alpha particle and gold atom.*" Zeke began the next discussion with an assumption about the prediction based on the name of the program

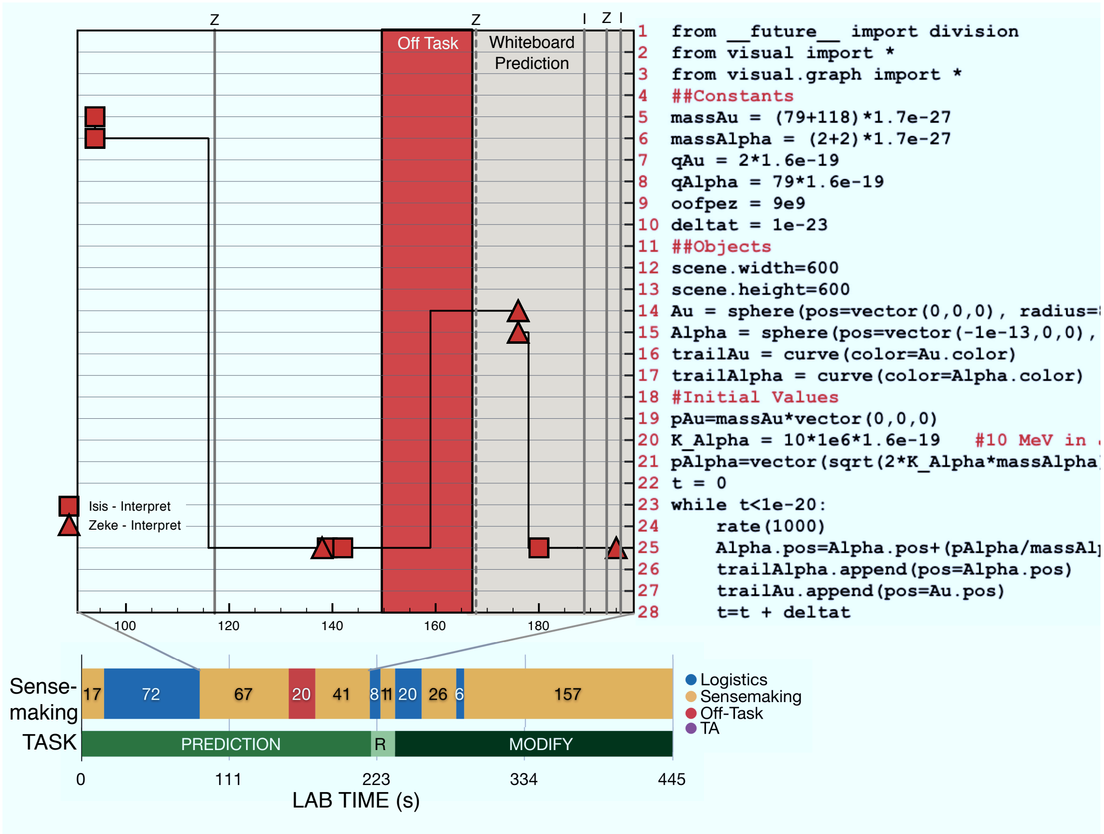


Figure 5.27: Line Number plot for Zeke, Isis, and Roslyn as they study the program code and draw a whiteboard prediction.

and the title of the activity:

Zeke: I'm guessing we're going to collide it eventually.....

Isis: yeah..

Zeke: I don't see where....[cursor in loop pointing to blank space after Alpha position update]

Isis: okay..that's uh..what that um....

Zeke: that's the momentum [cursor over pAlpha]

Isis: position...

Isis: no that's not momentum...that's uh...the position of it..no but it comes from the momentum principle...yeah...

Immediately after Zeke suggested that there would be a collision, he searched in the computational loop for evidence from the program code for this prediction. It's not clear what Zeke was looking for when he moved his cursor to the blank lines between the position update for the alpha particle and the update of the trail following the alpha particle. Isis began interpreting the previous line, the position update formula and indicated that the `pcraft` variable "*comes from the momentum principle*." Isis didn't attempt to identify the momentum principle in the program code or note its omission from the computational loop. So far, the discussion had not discussed the interaction between the two particles, which must have bothered Roslyn. Immediately after Isis's statement about the Momentum Principle, Roslyn asked, "Which one are we...are we doing like a gravitational problem?" This statement provides evidence that Roslyn was disengaged from the task of interpretation. It may have been an coincidence that Roslyn asked a question about an interaction at that moment, or the question may have been triggered by the group's discussion of the computational loop. Isis answered Roslyn by stating that this was a collision problem. Roslyn continued, outlining the analytical solution for the final momentum of the collision of two hard spheres. The group did not pay attention to Roslyn, and immediately after Roslyn's discussion about the analytical solution, Zeke began drawing the whiteboard prediction:

Zeke:So we got like two things..

Zeke: [draws two spheres on whiteboard]

Isis: yep

Zeke: and then..

Isis: the postion...the final..this is the final position of the alpha particle....its

going to equal to blah blah blah blah..plus the average velocity..I mean...

Zeke: it looks like the other one doesn't even move...

Isis: yeah..

Zeke: I guess just one of them is going to move..

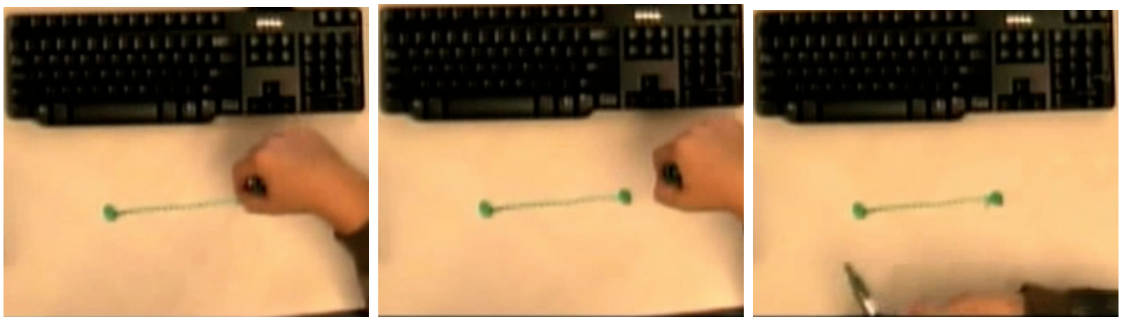


Figure 5.28: Screenshot of Zeke drawing the path of the alpha particle moving towards the gold particle(left), picks up marker and continues tip through the gold (middle), and returns to the trail to add an arrow at the location of the gold (right).

Isis: so the alpha particle collides into the..gold..

Zeke: [draws a path from left sphere towards right sphere, when marker arrives at right sphere, picks up marker but continues to move marker point through and to the right of the sphere, stopping three inches past the sphere. Adds an arrow head at the end of the path on the right sphere.]

Isis: the gold is at rest, that's what I think....I think the..yeah I think the gold is at rest..

Zeke's prediction used the omission of the position update for the gold particle to conclude that the gold particle didn't move. He was greatly bothered with what would happen to the alpha particle once it approached the gold's position. Isis suggested that the alpha particle collides into the gold, but Zeke had to draw what that would look like on the whiteboard. As Zeke did this, he was visibly conflicted with what happens. The whiteboard marker tip drew a path from the left of the board to the location of the gold nucleus. Then, Zeke raised the marker from the whiteboard, but continued the motion of his wrist to take the tip of the marker through the gold to the other side. There is evidence through this gesture that Zeke was considering that the alpha particle would travel through the gold. He didn't discuss this option with the group or say aloud any thinking on this possibility, but the gesture is clearly defined on the video tape, as shown in Figure 5.28.

Although the group did not discuss the attributes of the alpha and gold objects to identify the locations of these objects, Zeke drew the locations of the two objects in the correct place. Either the group didn't need to have a conversation about the location of the two objects or Zeke coincidentally drew the correct locations of the two objects. There's no evidence providing justification for Zeke's decision to place the alpha object to the left of the gold object.

Analyzing interactions in their absence: Frank, Madeline, and Estelle

Frank, Madeline, and Estelle interpreted the program from line 5 through line 21, the end of the initial values section. Estelle's contribution to the discussion was infrequent, however when she offered an interpretation, it focused on the computational loop and it drastically modified the group's thinking about the prediction of the visual output. The progression of the group through the interpret and prediction task is shown with a Line Number plot in Figure 5.29.

Frank and Madeline read through the program code and started with the values for the mass of the gold and alpha. Frank was confused about the type of atom represented by the chemical symbol Au, and interpreted the mass as the value for a Silver atom. Madeline and Frank also had difficulty interpreting what quantity the variable `q` represented, but eventually settled on interpreting `q` as representing the charge of the object. The participants didn't recognize the variable name `oofpez` as representing the acronym for the value of `o`ne-`o`ver-`f`our-`p`i-`e`psilon-`z`ero, but Madeline identified the quantity as a constant and eventually identified the variable name based on recognizing the value associated with the variable. Madeline recognized that 9e9 is the value for the constant the participants had been using to represent $1/4\pi\epsilon_0$. While Frank and Madeline interpreted `oofpez`, Estelle spoke and interpreted the position update of the alpha particle, on line 25, and said, "it's the alpha position is going to be updated and all of that." Frank and Madeline didn't acknowledge this statement and moved on and interpreted the attributes of the two 3D objects which led to correct predictions of the locations of the alpha particle and the gold nucleus. Frank identified the 3D object curve as a command to

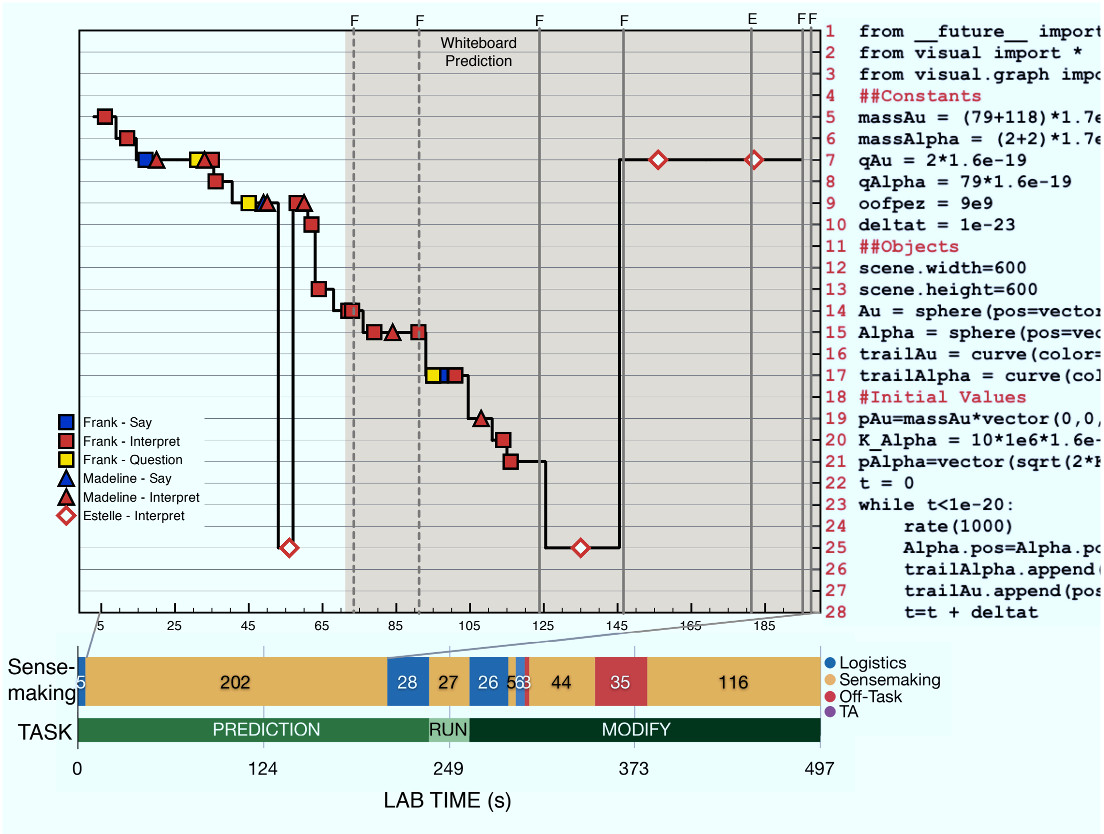


Figure 5.29: Line Number plot for Frank, Madeline, and Estelle as they study the program code and draw a whiteboard prediction.

follow this type of path through space:

Frank: curve.

Frank:..is that's how its moving..

Frank: trail Au, trail alpha..

Frank: that's what its going to follow I think, alright..

Frank explicitly stated his justification for his interpretation that something (unclear as to what) was going to move in a curve, and follow a curve due to this line of code. Frank's group during the Spacecraft-Earth MWP activity did not clearly justify their prediction of a spacecraft going around the Earth based on the curve object, although it was suspicious that the first time a group member mentioned an orbit was after saying the word "curve" aloud. Here, Frank provided enough evidence to link the interpretation of the curve object as a description of the path a particle is "*going to follow.*"

The group interpreted the representation of kinetic energy and momentum of the alpha particle without discussing the calculations used to define these quantities. Having read the program up to the calculation loop, Frank began a lengthy conversation about the prediction of the visual output:

Frank: so they're going to...if they're going to be attracted to each other magnetically....

Estelle: It looks like the loop is updating the position of the alpha particle but not the other one..

Frank: okay..so..I guess so we're trying to predict what they're going to do..

Frank: they're going to collide..

Estelle: I guess they're positively charged so...

Frank: run the program, okay never mind we still need to make a prediction..

Estelle: I guess they would collide....or no...would they both repel

Estelle: because they're both positively charged?

Madeline: Ah, that makes sense...

Frank: positively charged so that would make sense that they repel each other.

Estelle: yeah

Frank: alright so...let's say they repel each other..

Frank: [draws on whiteboard two arrows indicating movement in opposite directions]

Estelle: it seems like we should have a collision since that's what the lesson is about...*inaudible*

Frank began by suggesting an interaction between the two objects. Estelle stated that only one of the two object's position was updated in the computational loop. Using evidence that the two particles were positively charged, Estelle suggested that the two particles would repel. Everyone agreed with the prediction and moved on to drawing arrows representing the direction of the motion of each object moving away from each other. Estelle called everyone's attention to the inconsistency between their prediction and the name of the activity. Twice Estelle interpreted the position update of the alpha particle and she just mentioned that the gold nucleus position did not update in the loop. Estelle was either not making the connection to the consequence of this omission or distracted by Frank's suggestion of a magnetic interaction between the two objects. The participants invented an undefined interaction due to only the quantities defining the amount of charge on the two objects. The line of reasoning follows the form: if two objects are positive, they repel. This is nearly acceptable in the analysis of physical systems. So this group's prediction of an interaction is not a prediction entirely based on the program code, but a blending of the interpretation of the objects in the program, the quantities used to describe the objects and how two similar objects would interact if they were real.

Predicting the dynamics of a physical system: Howard, Tina, and Xavier

Xavier wanted to run the program to inform his prediction; however, Tina was quick to keep Xavier from doing so by restating and clarifying the study/prediction task. The group began interpreting the program code and almost immediately began making statements about the visual output and drawing analogies to physical systems, as shown on the line number plot in Figure 5.30. Xavier interpreted the mass of the alpha particle stating "*the alpha particle is a lot smaller than the other one.*" Tina interpreted this comparison as a relationship between the size of the two objects, not the mass, and immediately thought, "*that's like the bowling ball and the ping pong ball thing*?" Xavier asked the same question a few seconds later:

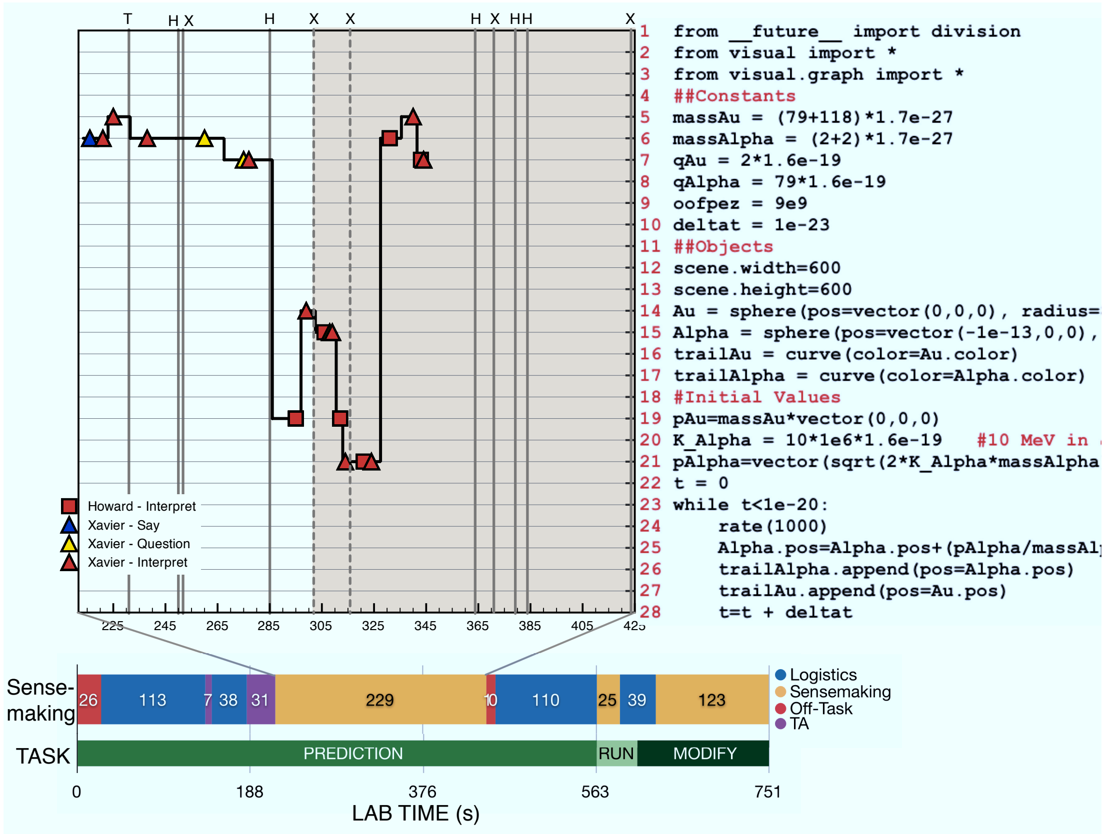


Figure 5.30: Line Number plot for Howard, Tina, and Xavier as they study the program code and draw a whiteboard prediction.

Howard: its going to be a collision

Xavier: its going to be change in energy's going to be zero..its just going to be....you know delta e equals zero...its just..the mass of the..

Xavier: is the mass of the gold sitting still?

Xavier: Is it like the bowling ball where like in the wasn't the mass of the big one ....

Howard: it will tell you...

Xavier: uh...

Howard: I still don't get how to read this...

The group was trying to guess what the program will show based on a lecture demo where a lecturer tossed a ping-pong ball at a stationary bowling ball. Howard tried to tell Xavier that the program code will illuminate what the initial state of the two objects, but they would have to interpret the code to figure it out. Xavier tried to interpret Line 7, the charge of the gold nucleus:

Xavier: what does the q mean?

Xavier: are you talking about the charge?

Xavier: Oh is this electric? Is this force electric? Is this force electric?

Howard: alright so its going to happen when the get closer..they repel each other...

Xavier: oh yeah...

Howard: So it...the gold is sitting still..[points to screen]

Xavier: yeah.

Xavier: so we have these two...we have one, alright so this is our gold right...

Xavier:[draws red circle on whiteboard]

Howard: then the....alpha particle..

Xavier: is it moving or is it staying still?

Xavier: and then we have our little alpha particle...

Howard: gold is staying still [points to whiteboard]

Xavier: gold is still...and we got, we're shooting an alpha particle at it..

Xavier:[draws sphere to the left of the sphere at the center of the board]

Howard: yes

Xavier: and its going...

Howard:[and its going] in the x direction..

Xavier: slike [sound effect] yeah in the positive x direction...okay...so..

The group, for the second time while interpreting the constants section of code, discuss the interaction between the two objects. The group was very uncomfortable moving forward in the interpretation of the program without considering how the two objects interact. In terms of the overall goals of including computational activities in the course, this was not evidence of a failure of instruction. The expectation that the program should model a physical system and therefore relate the program to their knowledge about those interactions is a promising connection to make.

Xavier in the episode above was gathering the information interpreted by Howard and recording it in the form of a whiteboard prediction. Xavier would ask a question, Howard would search and report an answer, and Xavier would modify the whiteboard prediction. This exchange continued until both the alpha and gold particle were drawn on the whiteboard and Howard interpreted the initial direction of motion for the alpha particle. The group returned to the previous conversation discussing what would happen when the two interact:

Howard: Well our prediction is, that its going to bounce off?

Xavier: well, when you start ..so its going to be like...delta E....[writes E=0]

Howard: we just got to predict what its going to look like...

Howard: so its just..going to bounce off right,

Howard: [going to bounce off right] since this is more...

Xavier: yeah once it gets closer..

Howard: this is going to move a little bit..[points to Au, moves finger in +x direction]

Xavier, like Roslyn in the previous group, began an analytical approach to explaining the motion of the system. Xavier began by writing one of the fundamental principles in the course: The Energy Principle. Again, Xavier tried to think from fundamentals about the interactions between the objects and the energetics of the system to predict the visual output. The prediction task must be ill-defined for the group. With this much focus on the physical system, the group must have interpreted the task differently than intended. Howard had to bring Xavier back to the prediction task, arguing that they didn't have to explain the motion, just predict it. However, Xavier kept going with his analytical analysis, defining the potential energy between the two objects:

Xavier: Potential electric...

Xavier:its just r..rmag?...r....[erases magnitude around r vec]

Xavier:so

Howard:probably r mag...

Howard:you can't divide by a vector...

Xavier:yeah...

Xavier:[replaces mag around rvec] yeah so once it gets close enough, its gonna have [adds some trail to the Au]...push away..

This episode was not predicted as actions students would take interpreting the program code. Yet these participants were using their physics knowledge to predict the outcome of an interaction and doing so appropriately from fundamental principles. This type of activity should be encouraged and this episode provides justification for instructional modification to the MWP activities that will be discussed further in Chapter 6.

Interpret program code, add hard-sphere interaction: Celia, Paine, and Yolanda

Celia, Paine,and Yolanda began the study + predict task reading through the program on their own. A period of 34 seconds passed while each participant silently looked at the computer screen. Celia broke the silence with the prediction, "*so basically the alpha particle is moving*," and Paine immediately followed with, "*and uh..the other one is not.*" Having completed a prediction, Celia asked the nearby TA, "Can we run it now?" After the TA clarified that the group should create a whiteboard prediction, the group shifted their focus back to the program code and read lines of code that inform the spatial locations of the 3D objects and how they would move through the visual output. Figure 5.31 reports the line number plot for this group.

Yolanda began the conversation about the location of the two 3D objects and interpreted the differences between the color of the alpha particle and the gold nucleus. Paine drew the alpha particle first by placing it at the center of the whiteboard, followed by placing the second

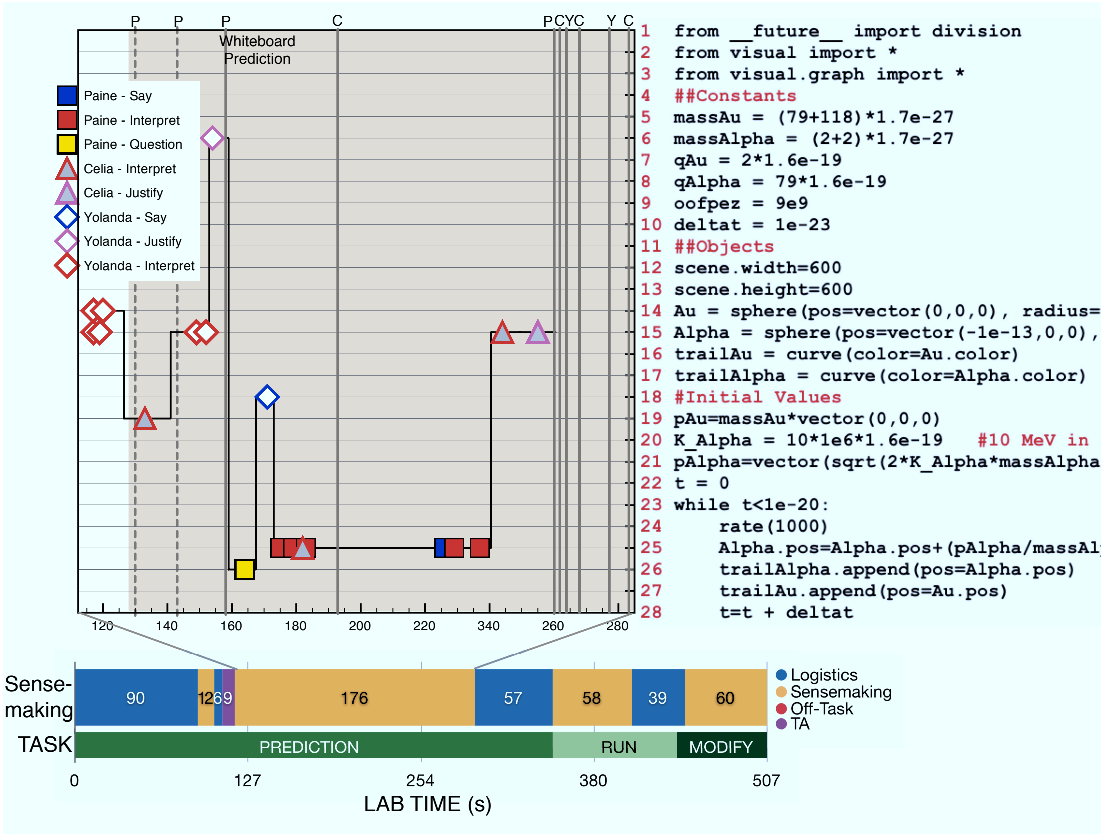


Figure 5.31: Line Number plot for Celia, Paine, and Yolanda as they study the program code and draw a whiteboard prediction.

sphere to the right of the first.

Yolanda: one's a lot bigger than the other...

Paine: which one's bigger?

Yolanda: the one at rest...

Yolanda: so its like...cause it has more mass, I assume ..

Celia: I don't know if its like moving towards it or..

Paine: is it moving around it?

Paine: What does append mean?.

Paine: I don't know what that means.

This is the second group who commented on the size or mass of the two objects. Interpreting

Yolanda's statement literally, Yolanda compared the radius attributes of the two spheres and commented that the gold sphere is bigger than the alpha sphere. Standing alone, this statement is vague to declare if Yolanda was interpreting either the radius attribute of the two spheres or the masses of the two objects. Yolanda provided more information; however, justifying the size difference due to the difference in mass values, so she must have interpreted the radius attributes of the two spheres in her initial interpretation. Otherwise, her statements would have been circular.

Paine suggested that one of the spheres was going to move around the other, most likely due to his attempt at interpreting the function of the `curve` object. Evidence of this interpretation followed in the next line, where Paine asked others about the `append` attribute. Paine did not seriously consider that one of the particles would orbit another, as he did not bring up this possible trajectory again during the rest of the prediction task. Paine was still quite confused as to what would happen in the visual output having interpreted the initial values for the momentum of the two objects, the positions of the two objects, and still confused over the purpose of the trail object. The only thing left for him to consider was the rest of the computational loop:

Paine: alpha position equals alpha position plus momentum of alpha over mass of alpha ...

Paine:so its velocity...

Celia: and momentum equals...[pause]

Paine: so its alpha position plus velocity...doesn't make any sense....

Celia: um..I think its like...it might be moving away. Alpha. I dunno.

Paine: I don't have a clue what this means dude..

Celia: It doesn't really tell you, I feel like you can't really see..

Paine: it says p alpha over mass times delta t

Paine:so its moving, velocity times the change in time..

Yolanda: yeah..

Paine: position plus velocity...

Paine struggled with interpreting the position update formula. Paine successfully interpreted `pcraft/mcraft` as representing the velocity of one of the objects. He recognized that the right side of the equation represented the position of the object plus its velocity, which didn't make sense to him, since you can't add a velocity quantity to a position quantity. Once Paine interpreted the `deltat` quantity as representing the change in time, he seemed more comfortable with the function of the line of code to move one of the objects. Celia took the implications of the interpretation one step further, and considered the direction of the motion. She eventually returned back to the interpretation of Line 15 to extract the information about the direction of the momentum based on the position of the alpha and gold spheres:

Celia: Do you think its moving towards it?

Yolanda: um hum...

Celia: its position vector is negative..so I think its on this side of the....[points on board to empty space to the left of the red sphere]...I think this [points to blue dot] is on the other side [points to left of the red sphere]...

Yolanda: yeah, you think the blue is over here and its moving towards it, right?

Celia: cause it has a negative x value...

Yolanda: yeah that's probably true..

Paine: alright...

Paine:[draws blue to left of AU, draws arrow to right]

Yolanda: okay

Celia: and its probably moving towards it..

Yolanda: yeah...and then when it collides...cause this is so much bigger

Celia: its smaller..this will bare.. [points to red sphere in center of board representing AU]...[interrupts herself]

Celia: its like the thing in class...if it barely moves then this one bounces off in some weird direction...

Yolanda: yeah...that's what I would think..

Celia: okay

Yolanda: it depends on where it hits it..okay..so let's see

Celia: so just...like...go like this [picks up marker and draws a trajectory path after collision]

Yolanda: and then the red will go slowly..

Celia: put like a couple dashes and then..

Paine: [draws a small red arrow on right side of red sphere pointing to the right]

Celia based the direction motion on which sphere was on the left. Celia justified her idea from the following logic: since the blue sphere was on the left, it must travel towards the larger red sphere on the right, therefore the blue sphere travels to the right. Yolanda thought this sounded fine and took the prediction a step further. Up to this point, the prediction involved all aspects of the program code and the group was completely correct. Yolanda wanted to determine what would happen when the alpha particle hit the gold. At this point, Celia drew on her recollection of the ping-pong and bowling ball demonstration from lecture and what she learned from her observations. Yolanda added an additional element to the prediction, that the larger more massive sphere would move slowly and indicated this motion on the whiteboard. The group predicted interactions which were not present in the program, nor referred to the program when thinking about the change in the motion of the two objects. The group extended the program beyond its design to include the interactions of a ping-pong ball and a bowling ball after the two physically collide.

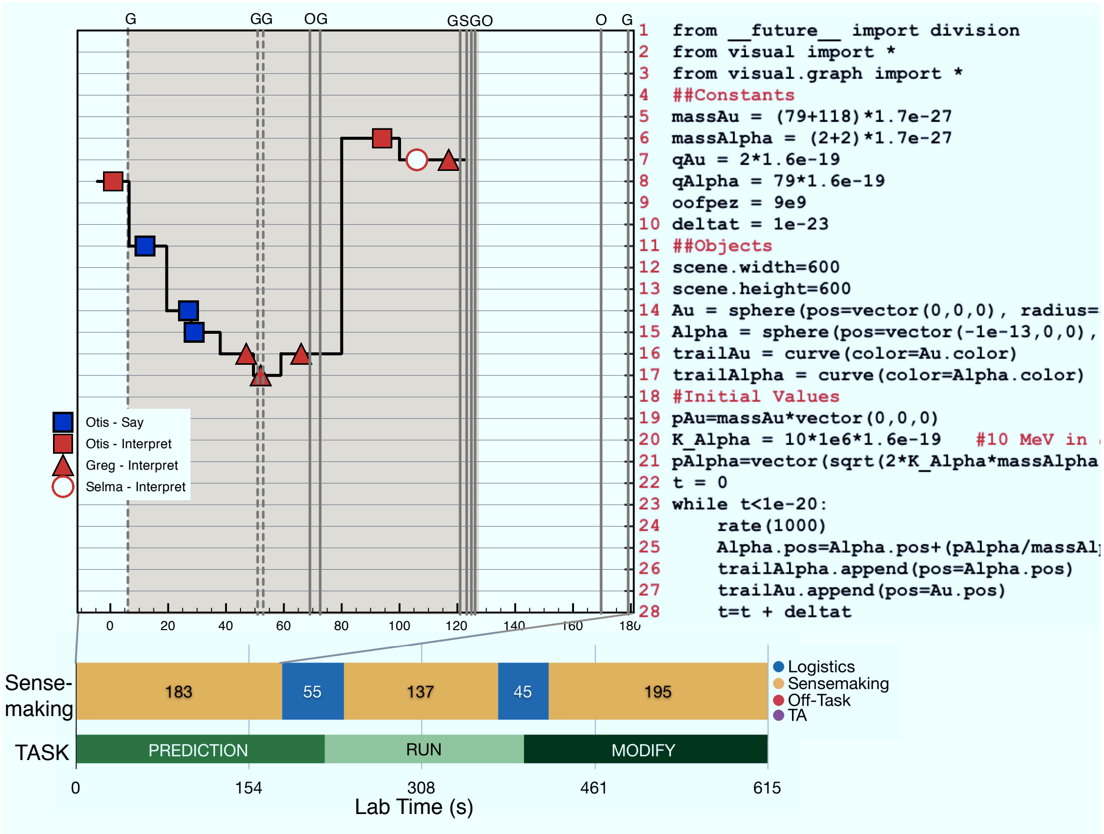


Figure 5.32: Line Number plot for Otis, Selma, and Greg as they study the program code and draw a whiteboard prediction.

Interpretation based on trails: Otis, Selma, and Greg

Otis, Selma, and Greg primarily used their interpretation of the trail object to inform their prediction. Figure 5.32 shows the progress of the group as they interpret the program code and create a whiteboard prediction.

At first glance of the line number plot for this group, there's very little interpretation of the program code. The group interpreted the mass of the alpha particle, the charge of each object, and the trail objects. The prediction this group created was not based on discussions about the interpretation of the location of objects, the initial momentum of the two objects, or the computational loop. The group discussed how to draw the trail to indicate the direction of motion of each object with a challenge from Otis:

Greg: we've got a trail going from this one [points to right sphere] following this one

Greg: [draws horizontal line towards right from right sphere],

Greg: and following this one

Greg: [draws horizontal line towards left from left sphere]

Otis: how do you..why do you think that?

Greg: [pointing to screen] trail curve, color alpha.. I guess the color..

Otis: how do you know they're going this way though [marker down on sphere, drags toward right]

Greg: I don't..

Although there was no discussion of physics in this episode, it is relevant that Otis asked for a justification for why Greg drew the trails the way he did. Otis therefore recognized that there were at least two possible interpretations of the motion of the two objects. In the next episode, Greg identified an interaction which would determine how the two objects would move, not the initial conditions for each sphere:

Greg: very true..[erases trail]..there's some trail somewhere..

Otis: yeah so...either they could like crash into each other or they could..go away from each other..they could..

Greg: I guess it depends whether they're...attracting each other or repelling each other..

Otis: ah..

Greg: Is it...no wait no.

Greg: which is q..is that the..

Otis: oh..um...[pointing to screen] 1.6...they're both...two..ah..one's 79 times as big..what? And one is two times that...and 79 times that?

Selma: that's the..is that the charge?

Otis: yeah

Greg: oh oh oh,..

Selma: cause like in the formula, q1 over q2 or whatever..

Otis: yeah q1 q1 over...r..

The variable q didn't completely make sense until the group considered its context in a formula used to calculate a potential. Now that the group identified an interaction, they moved on to determine the nature of the interaction and how it might determine the motion of the two objects:

Greg: alright..so..would that mean they're both positive...

Greg:so they'll be going opposite

Selma: repelling..

Greg: [drawing trails on other side of each sphere]

Otis: yeah I think it will crash into each other..let's see what it does..

Greg: is that all..wait what about all this stuff [points to while loop]..

Greg: do you think that maybe they were starting going towards each other and then the..[gestures] like repel off, or something?

Otis: no cause we've...we've been talking about uh...this is related to uh..what we've been doing with..not angular momentum..

Greg: Yeah he did say this has nothing to do with angular momentum

Otis: so...I'm thinking back to the lecture..whatever we talked about before that we had two particles...collisions [gestures] yeah we were talking about collisions...

Greg: yeah

Otis: right, so yeah..

Greg: I think its also collisions dot py...

Otis: alright

Otis: yeah so..I think they're just going to collide

Greg's interpretation of the representation of the trail and its function was different from

Greg's interpretation. Greg interpreted the trail as showing where the particle had been while Otis interpreted the trail as showing where the particle will go. Greg stated the two objects repel with Otis interpreting Greg's drawing of trail lines between the two objects as moving towards each other. This miscommunication completely changed the prediction and when Greg asked Otis to reconsider that the two objects wouldn't attract, Otis used evidence for what they had been doing in lecture, not evidence from the program code. Greg was okay with this interpretation, which was consistent with the name of the program.

The group interpreted very little program code, didn't consider the computational loop (although Greg mentioned that maybe they should talk about it), and didn't communicate clearly the evidence each person used to suggest a prediction. There was plenty of miscommunication between Greg and Otis about the trail and what its function was in the output. The entire group discussed the type of interaction that would occur in the real world between these to types of objects.

First interpret code, then focus on a prediction: Eugene and Dora

Eugene took the lead interpreting the MWP code and then deciding on a prediction. What's interesting about this group studying the code is seeing what wasn't interpreted by Eugene, a participant who has an expert-like understanding of how programs compile to produce the visual output. Eugene was very comfortable with interpreting the MWPs and often used his understanding of the object-oriented programming language to help inform him of how the lines of code of the MWP function to define and update the values of variables. Figure 5.33 reports the progress of Eugene and Dora through the program during the study task and the lines of code they returned to in order to inform their whiteboard prediction.

The group did not discuss the mass of the two objects or the values for their charge, only briefly commenting on what `oofpez` could represent, leaving the interpretation unresolved. As Eugene read through the objects, he only commented on the function of the lines of code,

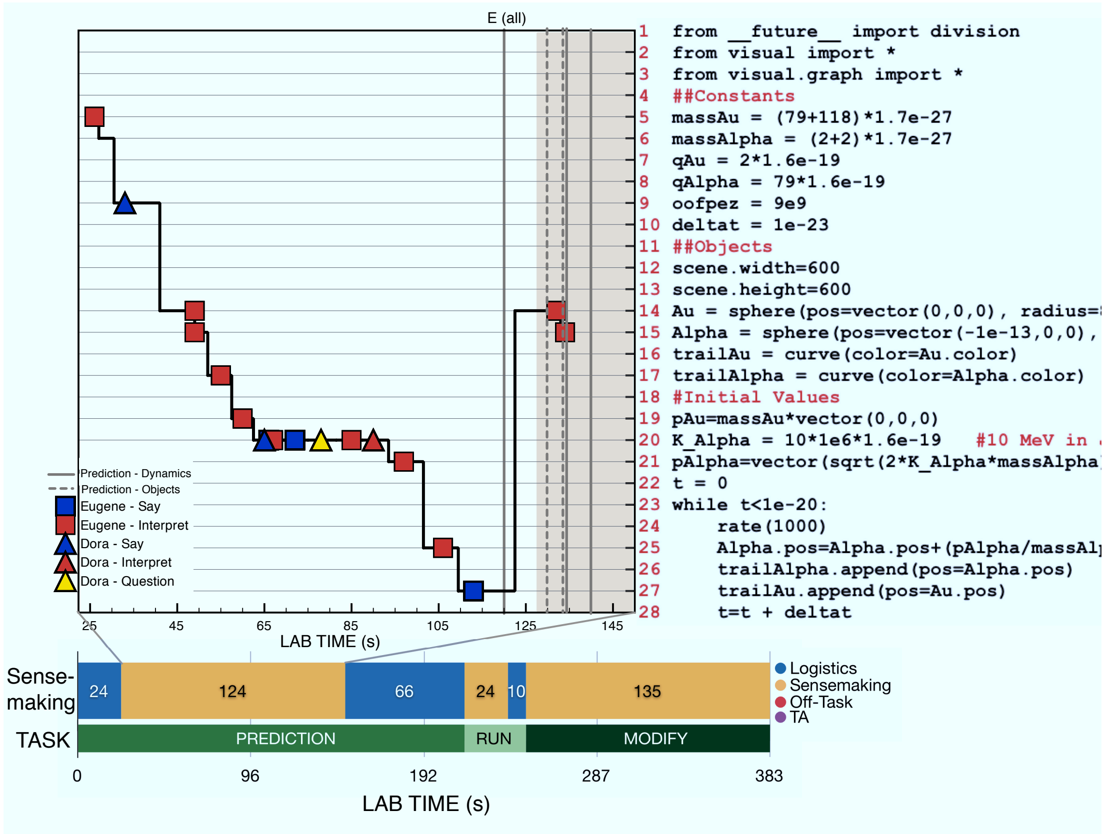


Figure 5.33: Line Number plot for Eugene and Dora as they study the program code and draw a whiteboard prediction.

ignoring any of the values associated with attributes for the alpha or gold spheres. Dora stopped Eugene to ask about the calculation of the kinetic energy:

Eugene: and that's

Eugene: ....what does that mean

Dora: what does that space mean? What does that mean like..

Eugene: I dunno.

Eugene: 10 times..electrons..cause this is 10 times something..

Dora: oh...its like 10 this unit and then multiplied by this number to get joules..

With Eugene's help interpreting the charge on an electron, Dora noted that the line of code simply converted the value into joules. This was the only line of code which gave the group significant difficulty, but only briefly interrupted Eugene's task of identifying the function of each line of code. Eugene continued until the end of the program, noting the appearance of the position update, and then indicated:

Eugene: its just going to be alpha particle moving..that's all it does now.

Eugene: so we're supposed to draw this or something?

Eugene: we predict a gold

Eugene:[drawn at center, diameter approximately 2 inches].

Eugene:..and we have an alpha particle

Eugene: [drawn smaller and to the left of Au on the x axis]

Eugene: draws a trajectory in a straight line that does not go through the AU, but beneath it.] It just keeps going...

Eugene: laughs

Eugene: it goes through this...[erases AU circle, moves it so that its center is on the trajectory of the alpha]

Eugene was hesitant to draw the trajectory of the alpha particle through the gold nucleus, but replaced his initial trajectory drawing with one that clearly showed the alpha particle traveling through the gold. The group ran the program and was pleased with the results.

Interpreting initial conditions: Max, Beatriz, and Lidia

Max, Beatriz, and Lidia began the study + predict task interpreting the initial conditions of the program before drawing a whiteboard prediction or talking about what they would see in the visual output. Once the group began the discussion about what the prediction would look like, Max avoided the conversation about how the two objects would interact as the two get closer. Figure 5.34 shows the line number plot for the group as they complete the first two tasks. Max began the discussion about the interpretation of the program code with Line 19,

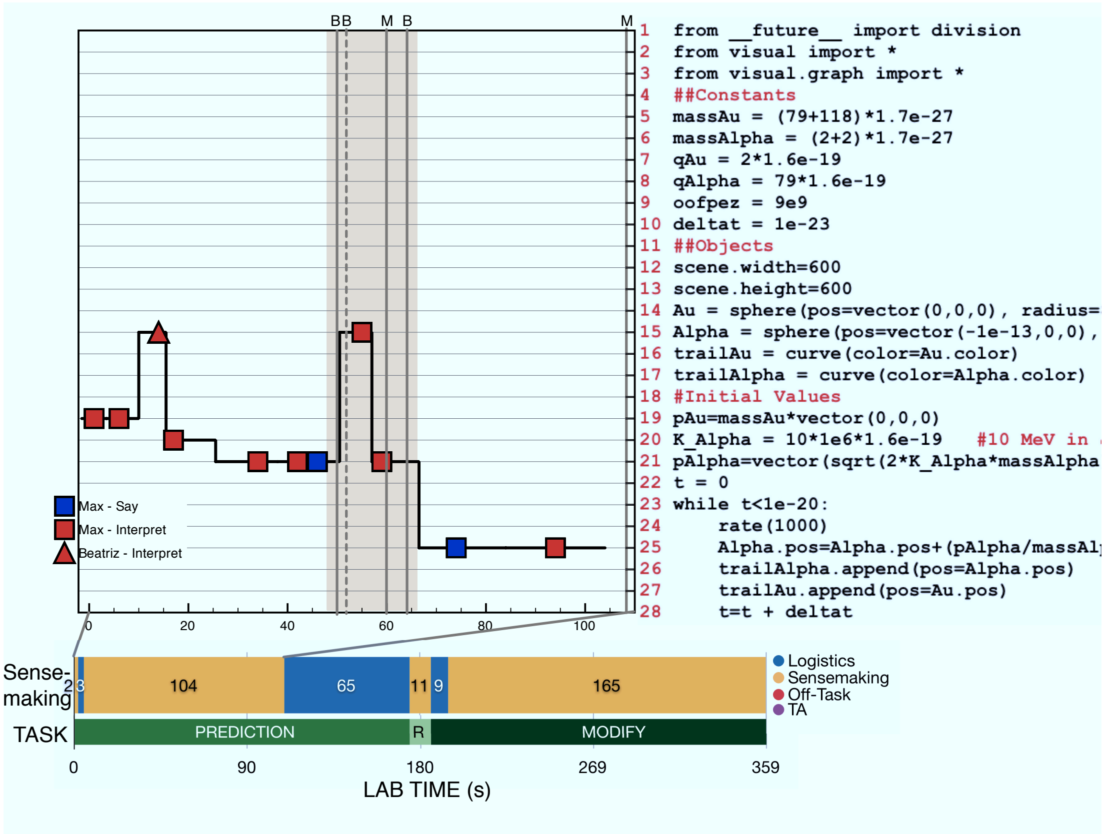


Figure 5.34: Line Number plot for Max, Beatriz, and Lidia as they study the program code and draw a whiteboard prediction.

defining the initial momentum of the gold nucleus:

Max: initially the au is not moving

Beatriz: okay

Beatriz: we have an alpha particle

Lidia: um hum..

Max: kinetic energy alpha..

Max: alpha particle has some momentum..and it is.

Max:..no idea which way its moving..

Max: oh its moving in the..positive x direction..

Max's interpretation of Line 19 may appear as a prediction statement about the dynamics of the gold nucleus, however it added information to Line 19 in the form of identifying the quantity and its value and how this value translates to motion. The interpretation was isolated to a single line of code. And, Max was right. Line 19 sets the initial value of the quantity representing the momentum of the gold nucleus to zero. This line has no function in the MWP code since this variable is not used anywhere else in the program. We know Max was looking at Line 19 as opposed to recognizing an omission for the position update for the gold nucleus because he used the word "*initially*." Max was not precluding that the gold nucleus would stay motionless, only that the simulation began with the gold nucleus motionless.

Max had difficulty interpreting Line 21 and stopped interpreting the quantities used to define this initial momentum of the alpha particle. Max decided that the only information he needed from this line was the initial direction, which he could determine, assuming the x component was a positive number. Max didn't explicitly state this assumption nor was there evidence that Max considered that the quantity could be negative (which it can't). Beatriz noted that if the alpha particle moved to the right, it would hit the gold particle, which began the whiteboard prediction task:

Beatriz: [erases alpha drawing..] okay so its going to hit the gold particle..or

Beatriz:[Draws alpha particle to the left of the first object.]

Beatriz:[writes "Alpha" under left object]

Max: yes..yeah..it starts to the left

Max: and its moving to the right...

Max: I dunno if it is going to hit but its going to go towards it

Beatriz: [draws arrow to right ending to the right of the first object]

The whiteboard prediction contained all the information that the group agreed upon up to this moment in the interpretation of the program. Max hesitated in agreeing with Beatriz that the alpha particle would hit the gold nucleus. Max instead restated the interpretation that the alpha particle moved towards the gold. Max took the group inside of the loop to interpret the code further:

Max: where's the while loop...Alpha dot... position of alpha particle..is gonna..

Max: [picks up marker, writes p=mv , v on board]

Max: okay so that's the position update formula..for the alpha dot pos..

Lidia: okay..

Max: yeah I think its just going to go

Max began reading off the code on Line 25, and needed to write the algebraic definition for momentum at low speeds to interpret the entire line of code. Max stopped short in his final prediction, of saying that the alpha particle travels through the gold nucleus and left his prediction open to interpretation by his group. He may have been specifically indicating that the alpha particle will only go to the right or that the alpha particle will initially move to the right and not consider what would happen when it moves to the same position of the gold nucleus.

Ramon and Norma

Ramon and Norma spent the entire time completing the study + predict task focused on the objects and initial values sections of the MWP code. Figure 5.35 shows that the group interpreted the locations and initial values of the two objects, then returned to the objects section of the code to inform their whiteboard prediction.

Ramon: do you think the two spheres are going to hit each other?

Norma: yeah..

Ramon: an alpha particle and..

Norma: gold..

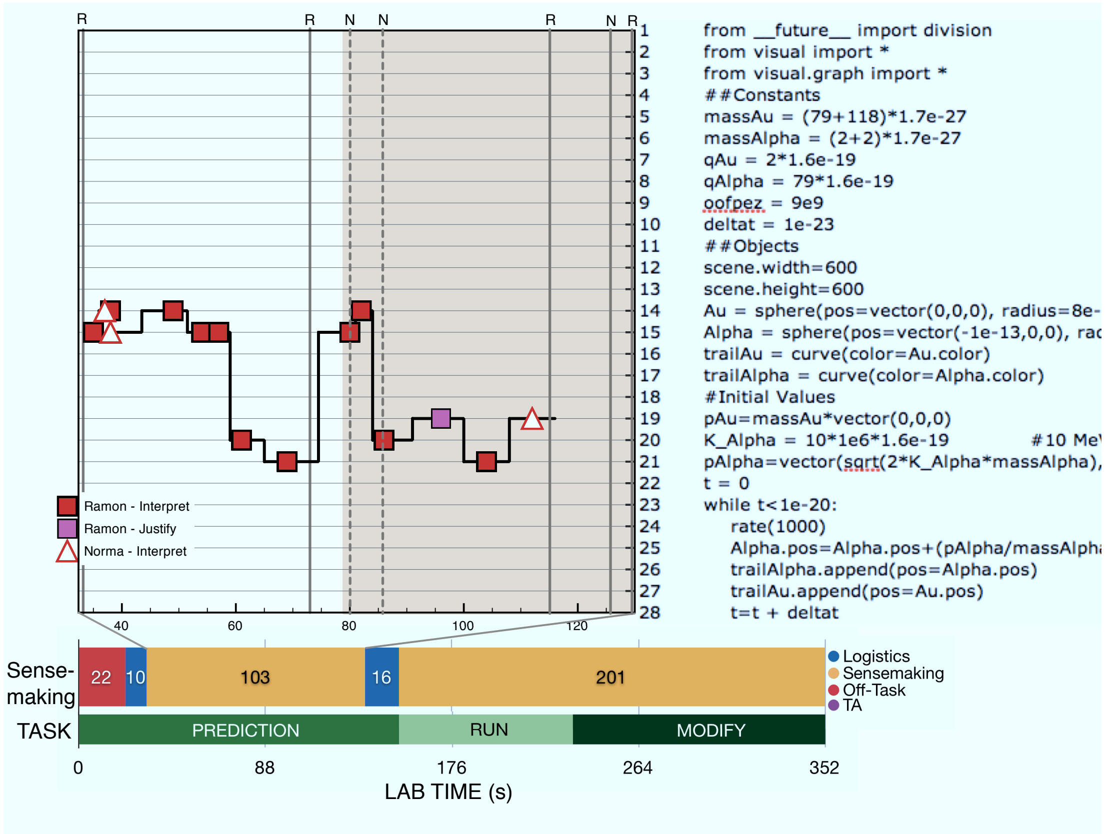


Figure 5.35: Line Number plot for Ramon and Norma as they study the program code and draw a whiteboard prediction.

Ramon: gold..

Norma: gold nucleus..and an alpha particle. they're spheres..and we're going to see [inaudible]..

Ramon: [inaudible] the gold is in the middle..

Ramon: sphere is to the left..

Ramon: .the alpha particle is to the left..

Ramon: .and the alpha particle has kinetic energy

Ramon: so it will move in the positive x direction

Ramon: so it will come..[gestures] over here..and hit

So far, everything in the episode is correct, including Ramon's initial prediction, although it's not clear that the prediction was based on the program code. The prediction emerged ten seconds after Norma read the instructions directing the group to make a prediction. Norma immediately agreed with Ramon's prediction and this idea of the two objects hitting carried through the task. However, Ramon didn't imagine what would occur after the "hit." The group then discussed the gold nucleus motion, identified that the gold would not move based on Line 19. Norma wanted to complete the prediction to describe what would happen after the hit:

Ramon: um hum..so we'll see it hit it..and..I guess it will come over and hit it..and then..

Norma: and this one should go off?And with this..

Ramon: uh..um..yeah, then this one [gold, gesturing] will probably go off..

Norma was ready to think about what would happen in a physical collision between two objects and used that knowledge to predict the motion of the two objects. Now that Norma included an effect of the "hit," there's a greater error that the group didn't consider an electrostatic interaction between the two objects. The group didn't interpret the values representing the charge on the two objects so they might not have considered that the two objects would interact in this manner.

Comparisons of interpretations among groups

Comparing the Line Number plots among the eight groups shows that there are differences between which sections of code were interpreted Table 5.5 summarizes how the groups interpreted the lines of code by looking at the results of the Focus coding pass.

Two groups explicitly stated that the function of the position update formula was to update the position of the alpha particle. One group identified the physics quantities in the position update formula but stopped short of interpreting the line of code as updating the position of the alpha particle. Two additional groups identified the line as the position update line, without explicitly interpreting its role in creating the visual output. Five of the eight groups interpreted at least one line in the computational loop and three of the five groups mentioned the `alpha.pos` or the omission of the position update for the gold sphere while interpreting the visual output or evaluating their prediction.

The `curve` object continued to offer some confusion, although this confusion was very brief and corrected by either the speaker or another group member. Two groups mentioned that the alpha particle might travel around the gold after interpreting the trail line, and one group began their prediction by considering how the trails would appear in the visual output. Two members of this group, Greg and Otis, interpreted the trail object very differently. Greg used the trail to discuss with the group where the two objects had been in the past whereas Otis interpreted the trail as a prediction of where the two objects had yet to go. The different interpretations of the whiteboard drawing of the trail objects led to a discussion about the possible types of interaction between the two objects.

Participant groups interpreted the variable `q` representing the physical quantity charge also

Table 5.5: Interpreting sections of code across all groups for the Rutherford Scattering MWP

Code Section	G1	G2	G3	G4	G5	G6	G7	G8
`Constants`								
`m`	Q,F	Q	Q		F			
`q`		Q	Q				Q	
`oofpez`		Q						
`Objects`								
`Alpha`	F	A,F	F	A,F	F	A,F	F	A,F
`Au`	F	A,F	F	A,F	F		F	A,F
`trails`		F			F		F	
`Initial Values`								
`pAlpha`		Q	F	F	Q	Q,F		Q
`KAlpha`		Q			Q	Q		Q
`pAu`		Q	F,D		Q	Q,D		D
`Loop Condition`	–	–	–	–	–	–	–	–
`Loop`								
`alpha.pos`	Q	F		Q,F	Q	Q		

Q: physics quantity, **F**: function, **A**: 3D object attribute, **D**:direction of vector.

discussed how charged objects interact. Three groups interpreted at least one charged sphere. Each of these three groups talked about the electrostatic interaction between the alpha particle and the gold sphere. Predictions noting an electrostatic interaction did not necessarily come immediately after interpreting q. None of the five groups who skipped the lines defining the charge of the two objects spoke of an electrostatic interaction.

5.3.2 Running the program: Interpreting the visual output and evaluating predictions

All of the participants ran the program code after agreeing upon a whiteboard prediction of the visual output. Just like previous MWP activities, participants commented on the prediction being correct or incorrect, the features of the code that were missing or would need to be added, and described the motion of the objects in the visual output. Two of the groups merely commented on their prediction being correct and moved on to the modification task. Zeke, Roslyn, and Isis merely described the motion of the alpha particle, "*it just went right through it.*" Celia, Yolanda, and Paine decided that their prediction was wrong and noted that "*it [gold] had no effect on it...it went through.*" Celia's statement recognized the lack of an interaction, which was important for the modification task. Howard, Xavier, and Tina also noted that their prediction was wrong. Xavier said, "*I don't know why we didn't predict that,*" and Howard responded, "*cause we haven't added the force in yet, right?*" Howard related the visual output back to a missing feature of the program code, the calculation of some force.

The other three groups, all of whom incorrectly predicted the visual output, spent more time with the evaluate task. Two of these three groups returned to the program code to justify the motion of the objects in the visual output. The other group spent more time than others interpreting the motion of the visual output and discussing what should happen to the dynamics of the two objects if they were interacting. This section reports the actions of these three groups.

Interpreting the visual output: Ramon and Norma

Ramon and Norma predicted that the two objects would interact and focused their interpretation of the program code to the objects and initial values section. While interpreting the visual output, Norma wanted to determine visually if the particle's speed changed after it hit the gold nucleus:

Norma: it went through

Ramon: yeah, so it hit and pass through it alright..

Norma: I don't think it changed the..when it passed through..or did it? Let's look.

Ramon: yeah

Norma: cause it could be like the electron thing. I don't think it changed.

Ramon: no, didn't really look like it.

Norma: okay.

Ramon: so it..when it hit, went through it, and kept going...basically the same..what looks like the same velocity..

Based on the Matter & Interactions curriculum, Norma may be referring to her physics knowledge of two possible explanations. Norma may recall the VPython demonstration of the Franck-Hertz experiment, where an electron accelerates towards a Mercury atom and excites the atom into a higher electronic energy state. The quantized electronic energy states for atoms and the mechanism for excitation was a topic discussed in detail weeks before this activity. Or, Norma may recall a more recent discussion from lecture on the Rutherford experiment where a VPython program modeling the outcome of the experiment derived from the "plum pudding" model of a gold atom where positive and negative charges are smeared throughout the volume occupied by a single atom. In the plum-pudding model, the alpha particle shoots straight through with little deflection and at times, no detectable deflection. Norma recalled something. It's too much of a stretch to suggest that Norma thought of either one of these experiments with as much detail described here, however the Franck-Hertz visual output does resemble the visual features of this MWP visual output.

Return to the code: Greg, Selma, and Otis

Greg, Selma, and Otis spent more than three minutes interpreting the program code and talking about the group prediction before running the program. However, the group only interpreted the trail objects, the mass of the alpha particle, and the charge. The group's prediction was focused on how the two objects should behave, not what the program would produce. After viewing the output, Otis led the group back to the program code for additional interpretation. He returned to the objects section noting the colors of the two spheres and matching these colors to the locations of the alpha particle and gold nucleus (he called it silver). Otis then moved on to the initial values section and the calculation of the kinetic energy:

Otis: equals mass..one half m v...

Otis: oh I guess cause...momentum of gold is zero..

Otis: there's the kinetic energy so it was..

Otis: .okay I think i see it now why one went through and the other one just stayed there cause you've got the momentum of gold..its not going anywhere..and then

Otis: ..momentum of the alpha particle has got an x..uh value..positive x value

Otis began his interpretation with his knowledge of the equation for kinetic energy, $K = 1/2mv^2$, and attempted to map the multiplication of the values on Line 20 with what he understood to be the equation for kinetic energy. He didn't pursue this task any further and only noted that the alpha particle had kinetic energy. Otis returned to the initial value for the momentum of the gold nucleus and noted that the alpha particle traveled to the right due to its positive x value. He didn't look to the computational loop and his group pressured him to move on to the modification task. Otis was the only member of the group seriously attempting to determine how the program code could produce the visual output.

Justification of output from the code and an attractive/repulsive interaction: Frank, Estelle, and Madeline

Frank introduced the idea of an attractive interaction during the prediction task, but the group ultimately settled on two objects moving in the opposite direction due to a repulsive interaction. Frank's rejected prediction returned while he attempted to explain the motions of the alpha particle. Estelle, on the other hand, correctly related the motion of the alpha particle and the lack of motion of the gold nucleus to the computational loop. Estelle recalled that she interpreted the positon update line for the alpha particle and didn't see a position update for the gold:

Frank: alright so, we were wrong..

Frank: that one doesn't move....[points to AU on whiteboard]

Estelle: uh so the alpha particle passes through.

Estelle:.okay that makes more sense because the alpha particle is the only one that position's updated..

Estelle: I forgot about that..the one that's going through it..

Frank: and then this one [points to alpha on whiteboard]is going to be attracted to it..

Frank:..its going to go through it...[gestures on whiteboard]..

Frank justified the motion of the alpha particle due to its attraction to the gold. He then further explained that the reason the alpha goes through it is due to this initial attractive force. He didn't consider what happened after going though it, or the flaws in his logic. Frank simply justified his incorrect prediction of an attractive interaction between two charged particles with the motion of a sphere traveling at a constant speed through a second object. His group didn't object, with Estelle simply conceding with, "*I guess so.*"

5.3.3 Defining the modification goal

At the end of the evaluation task, participants were asked by the instructional document to "modify this program to simulate what should happen to the alpha particle and gold nucleus as the alpha particle approaches." Participants started the task in one of two different ways. Participants either identified physics quantities that needed to be modified or added to the program or identified how the two particles should interact. There was a lot of discussion about each of these features of the program code or the visual output. Much of the discussion focused on the interaction and how the interaction should be described and represented in the program. Participants either focused on the changing dynamics parameters, such as the momentum quantity, or focused on inventing a potential energy term for the system of both the alpha particle and the gold nucleus and updating this quantity in the computational loop. None of the participants started with defining the interaction in terms of a vector force. None of the participants first thought to use the Momentum Principle. By the end of the activity, all students turned in a complete and valid program modeling the interaction of the gold on the alpha and the alpha on the gold, with both object's positions changing as a result of this interaction.

As with previous MWPs, this section will only focus on the stated modification goals after the groups completed the evaluation task. These goals changed during the course of the modification task due to TA interactions with the group, participants raising questions about alternative routes to completing the task, and information from the visual output or shell window on the results of executing the added or modified code. This work is left for future analysis.

The section is organized according to the type of interaction the students discussed while determining what should happen in the visual output. Five groups decided on adding quantities to show a physical collision where the alpha particle hits the gold nucleus and bounces off or travels together. The remaining three groups added quantities to show the interaction affecting the dynamics of one or more objects while the alpha particle was traveling towards the gold (action at a distance).

The bouncing formula: Zeke, Isis, and Roslyn

Zeke, Isis, and Roslyn created a prediction that was vague about the dynamics of the alpha particle after it reached the position of the gold nucleus. After viewing and interpreting the visual output, Roslyn began the conversation about what should happen in the visual output:

Roslyn: Are we just trying to make it bounce off?

Isis: yeah....I think so..that's what happens when an alpha particle hits a gold nucleus right? it bounces off? Yeah...

Zeke: I guess..[laughter]

Isis: I think so...

Zeke: either it bounces off or it sticks to it...its the only two options...

Isis: I'm pretty sure it bounces

Isis: because the alpha particle has positive charge?....that's...yeah.

Roslyn: okay so what are we having to do now?

Zeke: so we need to write the formula for what would make it hit it and bounce off of it..

Isis: yeah..okay so the initial momentum of the ....of the alpha particle would equal the final momentum of the alpha particle..

Zeke provided two possible outcomes of the physical collision: bounce off or stick together. Isis used evidence from the charge of the alpha particle to justify her preference for the alpha particle to bounce off the gold nucleus. Now that the group identified how the two should interact, the group focused on developing the mathematical model they should use to add to the program code to achieve this result. Isis began her analysis with the momentum principle for the system containing both the alpha particle and the gold nucleus. This analysis is typical of an analytic approach to determining the final momentum of the interacting objects, but will not help the group build a computational model to show the predicted motions of the two objects.

The group began with a goal about the dynamics of the two objects and transitioned to thinking about how "*to write a formula for what would make it hit and bounce off of it.*" Choosing both particles as the system eliminated the immediate need to consider the nature of the interaction between the two. Although Isis eliminated the "stick together" scenario based on the nature of the alpha particle being positively charged, her analysis did not take into account how this interaction constrains the motion of the two objects.

A completely inelastic collision: Estelle, Frank, and Madeline

Estelle, Frank, and Madeline predicted that both objects would repel each other and move away. During the evaluation of the prediction, Estelle noted that only the alpha particle's position was updated in the loop. Frank justified the visual output by inventing an attractive interaction between the two and in his mind, explained the motion of the alpha particle traveling through the gold nucleus. During the modification task, Madeline was first to suggest the quantities needed to calculate the interaction between the alpha particle and the gold nucleus:

0:08:44 **Madeline**: delta E equals.....what? I dunno...

0:08:49 **Estelle**: inaudible [reading] to calculate the interactions...

0:08:52 **Frank**: that'd be the...

0:08:55 **Estelle**: making me worry more about the test, okay...

0:08:56 **Madeline**: I know I know that's definitely what's running through my head.

0:08:58 **Frank**: would that be that whole uh...[writes 1/(4E...)] or whatever that was and then the q1 q2 over r [writes this on board]?

...and after a short time later...

0:10:17 **Madeline**: so what should happen?

0:10:15 **Madeline**: should it not go through it?

0:10:16 **Madeline**: should it hit it and they both go together?

0:10:20 **Frank**: its...yeah I think they should collide and become one particle..like it shouldn't go through it, it should stick together, I think...

Madeline suggested thinking about the motion of the objects in terms of how the energy changes in an unidentified system. The group wasn't sure how to complete the equation until Frank identified an electrostatic potential energy term. The group continued to identify which quantities were already defined in the program code (omitted from this episode) and Madeline shifted the focus of the group to ask for agreement on how the two objects should interact. Frank suggested the idea of the two particles physically colliding and moving off as one object. The group agreed and were stuck. The group did not know how to proceed further in the modification task and requested some help from the TA. The TA began a sequence of questions to understand how the group wanted the two to interact, what quantities describing the motion of the objects would change due to the interaction, and relate this back to the momentum principle.

The bouncing formula: Ramon and Norma

Ramon and Norma began the modification task after having a discussion about how the dynamics of the alpha particle should change if it interacted with the gold atom. Both agreed that if the gold atom's electrons interacted with the passing alpha particle, then the alpha particle should have decreased its velocity. Ramon and Norma shift their thinking towards a physical collision where the alpha particle should bounce off of the gold nucleus.

Norma: so what would take..do we need a thermal? Do you think we need a thermal or something?

Ramon: um..[gets notebook]

Ramon: so its probably going to uh..its probably going to be elastic so...I'm going to say..

Norma: so its going to bounce off of it? Like didn't that gold foil experiment didn't they...bounce off of the gold nucleus..

Ramon: In the which experiment?

Norma: that gold foil experiment when they found the positive..

Ramon: I have no idea..if you think so, that's cool with me.

Norma: yeah I think that's what happens..

Ramon: alright so you think it will bounce off

Norma: cause the alpha particle was positively charged so it should bounce off of the ..

Ramon: okay, alright so...it will bounce back and the gold will move with some momentum

Norma: I don't know if the gold will move because it may be too massive to really...I guess it will move some...

Ramon: a little..a little bit

Norma: but it might not move very much..it might..yeah it may not move very much..so we could put like..

Norma implied that her interpretation of the gold foil experiment that the alpha particle physically bounced off of the gold nucleus due to its positive charge. Ramon suggested that the other object's momentum should change and Norma cautioned that the new momentum might be nearly undetectable. The group discussed the demonstration from class of the pingpong ball bouncing off the bowling ball and that the bowling ball "*got a little bit, but not a whole lot*." It's not clear if Norma was thinking of momentum or velocity, but this recollection of the demonstration set the group up to think of the analytical solution of the two particle system:

Norma: what quantities do we need?

Ramon: um..well final energy will equal initial energy.

Norma: [writes $p_f = p_i$]

Ramon: and the intial energy is...just the kinetic of the alpha and the final is going to be the new kinetic of the alpha and the kinetic of the gold.

Norma: [writes $k_{f,a} + k_{f,au} = K_{i,a}$]okay

Interaction through potential energy: Howard, Xavier, and Tina

Howard, Xavier, and Tina predicted that the alpha particle would bounce off of the gold nucleus, but were also focused on the type of interaction between the two particles during the prediction task. Xavier, at several points during the interpretation of the program code, insisted on setting up the Energy Principle to begin an analytical approach to determine the types of interactions and how that would change the dynamics of the alpha particle. Therefore, it's not surprising that Xavier picked back up on this train of thought when he considered what quantities the group would need to add to build in an interaction. After Xavier read the question asking for the quantities needed to calculate the interaction, Howard said "*would it be mass and velocity I guess.*" This prompted the group to return to the code to see if these quantities were defined and how they were used in the program. The group returned to interpret the `Kalpha` line of code and Howard asked:

Howard: but how do we get it to interact?

Xavier: We have to have a uh....we have to enter in a potential...energy...

Xavier: so...where does that go?

Howard: in the loop

Xavier: down here?

Howard: yeah..

Not only was the group comfortable with adding in a potential energy calculation, they also decided that the calculation should go in the computational loop, without offering an explanation. The conversation during the prediction task carried over to the modification task and little further conversation occurred to justify why a potential energy was needed in the program code.

Celia, Paine, and Yolanda

Celia, Paine, and Yolanda predicted that the alpha particle would bounce off of the gold nucleus and referred to the demonstration from lecture as a justification that the prediction was plausible. During the modification task, the group began a discussion about the how the two objects should move in the output:

Celia: so like.. I dunno...kinetic energy, potential energy..

Paine: what's it supposed to do?

Yolanda: well it should bounce off...like it should have..like if the ball had mass it wouldn't plow straight through it..

Celia: yeah..I think it was supposed to apply to what we predicted it to...

Yolanda: probably

Celia: in the end...

Celia: okay so I basically we're adding new stuff..

Yolanda: so..the collision...

Celia: potential and kinetic energy..maybe?

Yolanda: yeah

Celia: as it gets closer the kinetic energy....increases...the potential ...decreases....

Yolanda suggested that the program should show a physical collision. Celia loosely suggested that the program needed to include kinetic and potential energy terms to produce the desired interaction. Then Celia justified the energy approach and noted that the kinetic energy increases as the potential decreases when the alpha particle gets closer to the gold nucleus. She just implied an interaction between the two particles over a distance to counter the physical collision claim by Yolanda. It's not clear if Celia knew that she transitioned the group from thinking of a physical collision model to an action at a distance model. However, this statement completely changed the focus of the group with Paine stating, " *potential....its one over..isn't it the one where q1 q2...over....*" The group returned to the program code and Yolanda found evidence that they were on the right track based on the lines of code defining the amount of charge on each of the spheres.

Physical collision analysis not right: Eugene and Dora

Eugene began the modification task using the momentum principle to analyze the two particle system analytically, but then identified that the analysis was appropriate for physical collisions. Eugene took a step towards considering how the physical quantities for the kinetic and potential energy change during the interaction:

Eugene: [gestures] this is ... inaudible the charge potential..does that work directly like that?

Eugene: [points to board] this will still work if its a physical collision right cause..if its like all the kinetic is change into potential and back into kinetic because its a inelastic collision.

Dora: um hum

Eugene: can we just go directly to the momentum update and consider the potential

Eugene: because all the kinetic goes to potential and back to kinetic..its an..

Eugene: or is it an inelastic collision?

Eugene was not clear how to develop the algorithm for updating the momentum of the alpha particle. He considered somehow using the potential calculation to update the momentum but he wasn't sure if the approach was valid for inelastic collisions. He previously stated that the kinetic transfers to potential and all back to kinetic only if the collision was elastic, which was correct. The TA came over and answered Eugene's question and pushed him to consider the problems with developing a way to calculate the momentum of the particle from an energy analysis based on the definition of momentum as a vector quantity in the program.

Action at a distance: Max, Beatriz, and Lidia

Max was the first participant to recognize that during the interaction the alpha particle would not physically touch the gold nucleus due to the electrostatic interaction. However, he had to convince Beatriz to leave behind her idea that the alpha particle would bounce off of the gold nucleus:

Beatriz: it should...is it supposed to bounce off of it and like go

Lidia: yeah

Beatriz: in a different direction?

Max: well...an alpha particle is ...I think two protons and two neutrons..

Beatriz: um hum

Max: is that right?

Beatriz: um hum...I think..right?

Lidia: I have no idea

Max: and the gold nucleus...gold nucleus is protons and neutrons, so its positive and positive so it should [gestures]

Beatriz: bounce off of them

Max: not actually hit but the potential energy will push them away...do you know what I mean?

Max: we're going to use the potential energy formula for...

Beatriz: so wait its supposed to like go around it? [gestures on board]

Max: I think...if its heading straight at it its kinda going to go like...[draws] and back that way..like it will get close..

Beatriz: um hum

Lidia: okay.

Max: we need a formula for potential energy though I think..

Max convinced Beatriz to consider the nature of the electrostatic interaction by considering the charge of each object and how the potential energy was calculated. Beatriz asked an odd question and there's no evidence to why Beatriz thought the alpha particle would travel around the gold nucleus.

Otis, Selma, and Greg

This group has nearly the same discussion as Max, Beatriz, and Linda. First Greg suggested that the two objects should interact and collide, but then quickly corrected himself:

Greg: I think this is where we try to make the...instead of make that pass through it..

Otis: um hum..

Greg: make it interact and collide off of it..

Otis: okay..

Greg: or, not necessarily collide with it..but interact with it cause its alpha particle ...

Otis was not yet on board with Greg's suggestion that the alpha particle would not touch the gold nucleus and it took Greg to point out that each particle was charged:

Greg: got to have like the uh..the energy to make it deflect right? Right? uh electric energy..is that it? The...

Otis: well is there..I mean, it's just..you have some particle..you've got gold and some particle...those aren't electrically charged are they?

Greg: yeah, they're both positive charge..cause the alpha particle is positive and it interacts with the...nucleus in the gold atom when it hits the..

Otis: how do you know both of them are positively charged?Uh from up here?

Greg: Cause the gold's not positively charged but there's protons in the nucleus that it will interact it to..and whenever it hits the nucleus...

Otis: so...they're going to basically move apart then..

Greg: yeah.. like I think what will happen is it will be like..

Selma: come in and then bounce off..

Greg: it will go ..and go..and it..basically travels off..[draws out of frame]..do like that..

Otis: will it bounce off or will it go together and then this one will [gestures]...

Greg: I don't know if it will really bounce off but it will be deflected even..[draws on whiteboard]..like that..

Otis: wouldn't it just...go backwards though? Like..once it gets close enough its just going to move back...

Greg: it wouldn't..

Otis: like why would it make it go like that...[gestures a scatter]

Greg: cause like if both of them are positively charged I don't think that they could ever actually touch..

Otis: right..I'm just saying like why doesn't the particle go backwards...

Greg: I think that's true..it could...

Then Otis pointed out that it didn't make sense to him why the alpha particle wouldn't just travel straight back, a feature of Greg's predicted motion that wasn't quite relevant to the task at hand. Greg yielded that it was possible that the alpha particle would travel backward. The group proceeded to define a quantity representing the final kinetic energy of the alpha particle. Although the group had a clear idea of how the two particles would move due to the interaction, the group was not clear as to how to calculate the interaction in an iterative way.

5.4 Comparisons across all MWPs

The last section of the Results chapter will look at how the interpretations, predictions, evaluation, and modification of the MWP evolved over the three MWP activities.

5.4.1 Interpreting and Predicting a MWP

Looking at the line number plots for the three different activities, or comparing Tables 5.2, 5.3, and 5.5 reveals that groups interpreted programs differently at the end of the semester compared to the first time they saw a MWP. This section reports how the class, as a whole, interpreted the Spacecraft MWP, the Mass-Spring MWP, and the Rutherford Scattering MWP.

Representations of physics quantities

Participants had no difficulty interpreting variable names as representing physics quantities, with a few exceptions. During the Spacecraft-Earth MWP, two groups interpreted the amount of time represented by the product in the loop condition statement. Two groups called the variable `g` the gravitational constant, which is not the common name for the quantity with a value of 9.8 m/s^2. During the interpretation of the Spring-Mass MWP, two groups had difficulty interpreting why the gravitational force calculation included a unit vector. These two groups moved on once another group member determined that the unit vector indicated the direction of the gravitational force. Perhaps the participants didn't first think of the gravitational force as a vector. Distinguishing vector quantities from scalar quantities is a challenging task for students programming in a mathematical model into the computational loop. If these participants don't identify the gravitational force as a vector, then it would be increasingly difficult to expect the participants to interpret errors between invalid algebraic operations when attempting to add vectors and scalars or vise versa. It would be interesting to investigate relationships between participants initial interpretation of vector quantities and errors these participants make when algebraically manipulating these quantities.

Along similar lines, one group interpreted the initial momentum of the ball as a vector *pointing to* the coordinate $< 0, 0, 0 >$ rather than having a value of 0 in all directions. Again, this confusion worked itself out by a group member correcting the error in the interpretation of the value. During the Rutherford Scattering MWP, participants had difficulty interpreting the position update, the initial kinetic energy for the alpha particle, the acronym representing one over four pi epsilon zero, and the initial value for the momentum of the alpha particle. Two groups paused to interpret `oofpez` with both participants eventually figuring out that the quantity, `9e9`, was the value for the constant term used to calculate the electrostatic force acting on an object.

3D objects

Participants had very little difficulty interpreting the location of 3D objects on a coordinate system, with a few exceptions. During the interpretation of the Mass-Spring program, Ramon and Eugene had to resolve a discrepancy between the orientation of the right-handed cartesian coordinate system used in their calculus class and the version used in physics. Isis drew the location of the ball away from the $-y$ axis; however it's not clear if this was an error of interpretation or a failure of interpretation the ball/spring's attributes. Lidia did not interpret the attribute of the ball correctly and predicted that the ball would have both negative x and y coordinates before being corrected by a group member. During the interpretation of the Rutherford Scattering MWP, Paine switched the locations of the alpha particle and gold nucleus; however, later Celia corrected the drawing using information about the direction of the momentum vector for the alpha particle and the negative value for the position attribute for the alpha particle. Any other errors in the interpretation of the actual locations of 3D objects were not due to mistakes in geometry.

Errors in the predicted location of 3D objects were also due to assumptions participants made about the system or simply not interpreting the attributes of the 3D objects. During the Spacecraft-Earth MWP, Zeke, Paine, and Frank did not attempt to interpret the attributes for the spacecraft or the Earth and therefore placed the spacecraft in the wrong initial location in their prediction. During the Mass-Spring MWP activity, all groups connected the ball to the end of the spring. Groups either interpreted the position of the ball and drew a spring from the ceiling to the ball's location (3 of 6) or interpreted the location of the end of the spring and drew the ball at the end of the spring (3 of 6). Isis, Frank, and Estelle interpreted the initial location of the ball *and* the location of the free end of the spring, but did not catch the discrepancy that the two were not referring to the same location in space. The group predicted that the ball was ultimately connected to the end of the spring at $y = -0.26$m. Celia was the only participant who interpreted that the free end of the spring was compressed and it's not clear how she arrived at this conclusion.

This thesis has already commented on the troubling interpretation of the `curve` 3D object as a description of how *other* 3D objects move through the visual output. The Spacecraft-Earth MWP and the Rutherford Scattering MWP contain curve objects. During the Spacecraft-Earth MWP, half groups interpreted the trail object with 2 of the 3 interpreting that the curve object had an effect on the motion of the spacecraft. There was additional evidence in the choice of words used to describe the motion of the spacecraft that two additional groups were conflicted about the function of the curve object and its effect on the spacecraft. These two groups did not explicitly discuss the curve object, but again their use of the word "curve" is suggestive that the word choice was informed by the 3D object itself. During the Rutherford Scattering MWP activity, 2 groups briefly interpreted the trail object as a command to travel in a curve; however, these interpretations were quickly overruled. And, the interpretation of the function of a trail differed within a group to the point where Otis and Greg were talking about two different predictions and they didn't realize it. Although the trail object had a significant effect on the prediction of the motion of the spacecraft in the MWP, it had a very minor effect on the interpretation of the motion of the alpha particle.

Computational loop

Participants increasingly looked to the computational loop to inform a whiteboard prediction throughout the MWP sequence. Beginning with the Spacecraft-Earth MWP, only one group (Selma, Eugene, and Beatriz) specifically mentioned physics quantities that appeared in the loop. Paine mentioned the omission of the acceleration due to gravity anywhere in the program code and used this omission to inform his prediction about the motion of the spacecraft, but didn't interpret the lines of code in the computational loop.

During the Mass-Spring MWP interpretation, 6 of the 8 groups discussed their interpretation of the program code (one group didn't interpret the code and one group ran the program before interpreting the program code). Half of the groups interpreted the MWP and suggested that only the gravitational force was acting on the ball (3 of 6) but only 2 of these 3 groups determined this by looking at the calculation of the net force in the computational loop. The other group noted that the only initial value calculated was the one corresponding to the gravitational force and that the spring force did not have an initial value and therefore didn't need to search for an Fspring in the computational loop. Two groups interpreted the momentum variable in the Momentum Principle without talking about the function of the line of code. One of the groups identified the function of the Momentum Principle to update the momentum of the ball during the simulation. However, this identification of the function of the momentum principle was not used to inform the group's whiteboard prediction. And, two of the six groups interpreted the function of the position update formula in the loop to move the ball. In all, 4 of 6 groups interpreted the function of at least one line of code in the computational loop; however only one group interpreted the function of both the net force and the Momentum Principle and none of the groups interpreted the net force, Momentum Principle, and Position Update formulas together.

During the Rutherford Scattering MWP interpretation, there's evidence that 4 of the 7 groups interpreted the position update formula for the alpha particle. Two of these 5 groups only identified the line as the Position Update formula and the other 2 groups talked about the function of the Position Update formula to move the alpha particle. One group simply identified the physics quantities in the Position Update formula. Additionally, one of the 5 groups who interpreted the Position Update formula also noted specifically that the Position Update formula for the gold nucleus was missing.

Across the three MWP activities, participants increasingly identified the physics quantities in the computational loop and interpreted the function of the lines of code in the loop. Table 5.6 reports the number of groups interpreting either a physics quantity or the function of the line of code in the computational loop across the three MWP activities. Looking at the interpretation of the position update line across the three activities, 5 of 8 groups identified either the line as the position update or the physics quantities in the calculation to update the position of the alpha particle which is more than the number of groups who identified the position update in either the Spacecraft-Earth MWP (1) or the Mass-Spring MWP (2).

Interpreting from beginning to end versus searching for beacon statements

While reporting the results of groups interpreting the MWP code, I combined groups according to the strategy in which they deployed to interpret a program and create a prediction. There

Table 5.6: Number of groups interpreting the computational loop across all MWP activities

Function of Line of Code	Spacecraft-Earth N=6	Mass-Spring N=6	Rutherford Scattering N=8
Net Force		0 (Q) 2 (F)	
Momentum Principle		2 (Q) 1 (F)	
Position Update	1 (Q) 0 (F)	1 (Q) 1 (F)	4 (Q) 2 (F)

(**Q**): physics quantity, (**F**): function were three primary approaches to this task: 1) read the program from the beginning (after the standard module statements and visual scene) and work their way down, creating a whiteboard prediction as information from the code could be translated into a visual representation, 2) read the program code from beginning to end to understand the function of each line of code and then return to make a second pass to inform a whiteboard prediction, or 3)search for, and interpret beacon lines of code which had the role of creating a visual output. Each strategy is easily identifiable by looking at the line number plots produced from each transcript. Table 5.7 reports the number of groups deploying each strategy.

Groups who deployed a beacon statement approach attempted to identify and interpret key lines of code to inform their prediction of the visual output as opposed to reading the MWP code from the beginning to end. In Pennington's(1987) model of program comprehension, experts at interpreting program code search for lines of code which perform the task the code was designed to handle whereas novices read the program code from the beginning to the end. We don't know if individual students followed this approach since this protocol can only comment on group discussions about the program code. However, as a group, the focus jumped around and every group deploying the beacon strategy interpreted the position update for a 3D object. These groups were not always successful in correctly predicting the visual output mainly due to using extra information not in the program code to generate a whiteboard prediction, as discussed in the next section.

Table 5.7: Interpretation strategies across all MWP activities

Strategy	Spacecraft-Earth N=6	Mass-Spring N=6	Rutherford Scattering N=8
Interpret then Predict	2	1	3
Interpret and Predict	4	4	3
Beacon Statements	0	1	2

Predictions about the dynamics of 3D objects

The information participants used in generating a whiteboard prediction did not vary from MWP to MWP as much as the interpretations of the program code. Students continued to use information from their interpretation of the program code to draw a whiteboard prediction. However, participants didn't stop there. Participants also used their own knowledge of the interaction of physical systems, demos from lecture, or previous VPython programs to inform what they would see when they ran the MWP code. Table 5.8 shows the number of statements that refer to each of these references to information that was introduced by analogy to inform and understand what the visual output would produce for each participant group.

The MWP activity can distinguish predictions about the dynamics of the system which use information from the program code or something else, such as interactions in the physical system or knowledge about the real-world behavior of the objects. There were relatively fewer predictions of the Spacecraft-Earth visual output that used knowledge about the physical system, however most groups predicted that the spacecraft would travel around the Earth. This may seem like a contradiction; however, the basis for the predictions that the spacecraft would orbit came from a false interpretation of the program code. Specifically, the false interpretation that the curve object would constrain the motion of the spacecraft to travel around the Earth. There were two groups where one or more participants discussed the motion of the spacecraft traveling around the Earth either before interpreting the curve object or by drawing an analogy

Table 5.8: Source for information informing a whiteboard prediction across all MWP activities

Source	Spacecraft-Earth N=6	Mass-Spring N=8	Rutherford Scattering N=8
Program Code	6	7	8
Physical System	2	6	5
Lecture/Demo	1	0	3
Other Program	1	0	0
MWP Visual Output	0	2	0

to the Moon-Earth system.

While creating a whiteboard prediction for the Mass-Spring MWP, all but one group (the group that cheated) used information from the program code to inform a whiteboard prediction. Six of the eight groups used extra information extraneous to the program code to inform their prediction. All of the groups attached the ball to the end of the spring and 6 of the 7 groups making a prediction gathered that the ball would "*oscillate*" or "*go up and down*". None of these groups predicting an oscillation attempted to search for a line of code that would serve this function. It is likely that the participants assigned real-world properties of physical springs to the helix object called "spring." The single group that predicted the ball "*would just fall*" noted the omission of a calculation of "*the spring constant*."

While creating a whiteboard prediction for the Rutherford Scattering MWP, all groups used the program code to inform their whiteboard prediction with 5 of the 8 groups drawing on knowledge about how two objects with mass would interact in a collision or how two charged objects would interact when traveling closer together. Interestingly, three of the eight groups referred to a demonstration from lecture where a ping-pong ball was thrown at a bowling ball to inform their prediction of the motion of an alpha particle and gold nucleus. All of the groups interpreting a physical collision involving contact between the alpha and gold skipped the interpretation of the quantity representing the charge of each object. Those group interpreting the two objects as having some charge also predicted at some point during the task that the two would interact due to both objects being charged. These groups didn't necessarily settle on a final prediction where the two would repel or attract. There were two groups who considered the objects would interact in a physical collision and an electrostatic interaction. Analysis of the predictions for the Rutherford Scattering MWP shows participants inferring electric interactions based on whether or not they interpret the charge of the 3D objects.

During the prediction of the Spacecraft-Earth MWP, participants were fine with having more than one prediction at runtime rather than attempting to resolve the contradicting predictions by using the program code. Three of the six groups creating predictions involved two predictions at runtime where one participant predicted that the spacecraft would travel "up" and another participant predicted the spacecraft would travel around the Earth. Dual predictions at runtime only occurred during the first MWP activity and didn't appear again in the predictions for other MWPs. Instead, the group resolved contradictions between predictions. Each resolution brought the prediction closer towards containing the same dynamical features of the interactions of the physical system, and not the accurate prediction of the MWP.

There's also evidence of conflict within a participant as one considered different predictions for the motion of the system. The evidence comes from indications that the gestures representing the motion of 3D objects contradicts or precedes the verbalization of the prediction and represents a shift from one's prior thinking about the motion of the objects. Xavier's actions follow this sequence when considering that the spacecraft might just travel around the Earth right at runtime before the visual output was displayed. Zeke's actions with the whiteboard marker show that he was silently considering a possible trajectory of the alpha particle that took it through the gold nucleus but never mentioned this consideration to the group and ended his prediction just before the alpha particle reached the location of the gold nucleus.

5.4.2 Evaluating the Prediction

Participants were quick to identify if their prediction was correct or incorrect and identify visual discrepancies between the two. There were a few instances where the visual output was difficult to interpret, such as instances when the camera zooms away and gives the appearance that the spacecraft moved to the right. More interestingly, a few participants interpreted the visual output to salvage their incorrect interpretation of the program code. While viewing the visual output for the Rutherford Scattering MWP, Norma wanted to determine if the alpha particle slowed down after traveling through the gold nucleus to conclude that the alpha particle may have transferred some kinetic energy to the gold via an electron excitation mechanism. In a different group, Frank interpreted the alpha particle moving towards the gold particle due to an attractive interaction which, to him, explained why the alpha particle traveled through the gold and emerge on the other side. During the Spacecraft-Earth MWP, Roslyn immediately confirmed that their prediction of the spacecraft traveling around the Earth was correct as soon as they saw the spacecraft and Earth pop up on the visual output.

During the evaluation of the Mass-Spring MWP predictions, all participants were very quick to note that the ball was not "attached" to the spring. Unfortunately, this created problems later during the modification task as participants believed their goal was to somehow attach the ball to the spring and then the ball would oscillate.

Increasingly over the MWP sequence, participants returned to the program code after evaluating the whiteboard prediction. These participants were usually looking for an explanation for what they observed or why their prediction was incorrect. However, a few groups returned to the program code to interpret additional lines of program code that was passed over during the first attempt at making a prediction. During the Spacecraft-Earth MWP, only one group returned to the program code to justify the output, but didn't interpret any additional lines of code. During the Mass-Spring MWP activity, three of six groups returned to the code and interpreted more quantities after viewing the visual output to search for why the ball was not attached to the spring. During the Rutherford-Scattering MWP activity, one of the eight groups returned to the program code for further interpretation to understand why only the alpha particle moved in the visual output.

Increasingly over the MWP sequence, participants justified what they saw in the visual output based on the interpretation of the function of program code. Participants said things like "*it should be moving, especially since it has the position update and everything*" and "*the position update's just based on gravity.*" These justifications are always about the position update or the lack of forces acting on the objects. Justifications are only made by groups who incorrectly predicted the visual output. During the evaluation of the Spacecraft-Earth

MWP, 3 of 5 groups who incorrectly predicted the visual output justified the visual output with interpretation of what was missing from the program code. During the evaluation of the Mass-Spring MWP, 4 of 6 groups who incorrectly predicted the visual output justified the visual output from the program code. And similarly, 4 of the 6 groups whose predictions were incorrect for the Rutherford Scattering MWP justified the visual output based on missing position update formulas or missing calculations for the interaction between the two objects. These justifications occurred spontaneously out of comparing the group's prediction and thus, their understanding of the program code with what occurred in the visual output. The instructions did not ask groups to justify the visual output based on the features of the program code.

5.4.3 Modification Goals

The modification goals were heavily influenced by the features of the MWP code that were omitted or the initial values of the locations of 3D objects. During the Spacecraft-Earth MWP activity, groups thought about how the spacecraft should move through the visual output due to an interaction with the Earth through the gravitational force. The modification task was heavily scaffolded in the instructional document to ask the groups to consider the type of interaction, how the interaction should effect motion in the visual output, and the calculations required to define the interaction. All of the groups identified the correct interaction and the algebraic formula representing the magnitude of the gravitational force between the two 3D objects. Only one of the groups additionally noted the requirement of the Momentum Principle to update the momentum of the spacecraft due to this interaction.

During the Mass-Spring MWP activity, nearly all of the groups were initially focused on attaching the spring to the ball or moving the ball to the end of the spring. This feature of the visual output was distracting for the groups and fixing this issue took priority over any other modifications to the program. And, it appears that 3 of the 7 groups expected or hoped that moving the ball to the end of the spring would then attach the ball to the end of the spring and have the spring interact with the ball. The other four groups mentioned adding in a spring force to the program. The TA unintentionally gave away the answer to one of the eight groups modifying the program.

During the Rutherford Scattering MWP activity, groups were focused on how the two objects should interact in the natural world, more so than with any prior MWP activity. Groups either focused on modeling a physical contact collision or an interaction at a distance. Participants wanted to deploy the analytical machinery of the Energy Principle or the Momentum Principle taking both objects as the system of analysis. The energy approach would be successful in predicting the speed of the particles at some distance from one another, but would not be able to calculate the specific vector quantities required to update the position of the alpha particle or the gold nucleus. Taking both particles as the system and deploying the momentum principle simply gives a trivial result, that the momentum of the system doesn't change. Groups had a difficult time thinking of an iterative procedure for calculating an interaction, updating the momentum of the two objects due to the interaction, and updating the position of the two objects with the new momentum values all over short time steps where the interaction is approximated as a constant value.

5.4.4 Logistics of completing the tasks

Time completing tasks

Participants continued to make sense of the MWP by staying on task towards interpreting and predicting the program code, with a few exceptions. Table 5.9 reports the amount of time each group spent on the study+predict tasks engaged in making sense of the program code and generating a whiteboard prediction. There is no statistical difference among the means for the time to complete the study+predict tasks across the three MWP activities (Whitney-Mann, $p < .25$) however there are positive correlations between the number of interpretations of MWP

Table 5.9: Time spent per group interpreting and creating a whiteboard prediction across all MWP activities

Group	Time On-Task (s)		
	Spacecraft-Earth	Mass-Spring	Scattering
G1	144		108
G2		190	202
G3	78	24*	229
G4	79	177	176
G5	109	187	124
G6	219	154	104
G7		119	103
G8	156	244	183
Mean	131	179	154
St.Dev.	54	42	50

*Outlier removed from statistics

code and the time taken to complete the study+predict task.

TA interventions

The teaching assistant intervened during the activity to clarify the interpretation of the visual output or to provide additional help as participants navigated through the activity. During the Spacecraft-Earth MWP, the TA only intervened to clarify the interpretation of the camera zooming out of the scene. During the Mass-Spring MWP, the TA intervened more than necessary and even gave away too much information to Ramon and Dora's group when asking the group how they might go about modifying the program code. The conversation prompting the TA to say too much was due in part to the question asked in the instructional document for the group to describe to the TA how to recreate the visual output with lab apparatus. The TA's interaction wasn't always distracting to the groups working on the activity and at times the interaction helped clarify the task or challenge the evidence the group used to generate a prediction of the visual output. For example, after overhearing Xavier predict that the ball would move up and down at the end of the spring, the TA asked Xavier if his prediction was for how a real spring works or if the prediction was about the program code. The group ignored the TA and continued on with their prediction of an oscillating spring.

Cheating

There were three instances of cheating across the MWP activities. The first case, during the Spacecraft-Earth MWP activity, occurred due to the group being distracted by an off-task conversation and simply ran the program before identifying the tasks associated with the MWP. The group struggled with getting back on track with the remainder of the activity. Since the group didn't interpret the program code, they were very confused as to what they saw in the visual output and what they were supposed to do with the program. The modification task was very difficult for them since they didn't form a whiteboard prediction. Celia's initial goal was to change the velocity of the spacecraft so that it travels into the Earth rather than calculating an interaction between the two objects. The TA eventually told the group what other participants predicted before running the program and their approach at completing the modification task. With this knowledge, the group move on and worked towards modifying the program to include the calculation of the gravitational force.

The second and third instances of cheating were on purpose, no mistake about it. During the Mass-Spring MWP, two groups ran the program with the purpose of informing their whiteboard prediction. One of the groups was nervously rushing to copy the program code into an editor window to save and run before the teaching assistant came around. The other group began looking at the program code in the editor window when Roslyn suggested running the code instead of interpreting. The other two participants stated that they didn't care and the group recorder ran the program. After running the program code and drawing the prediction, Celia's group asked the TA what they were supposed to be doing and the TA instructed them to read the code for understanding. Interestingly, Celia and Madeline did just that. They began reading the program code identifying the function of individual lines of code and identifying physics quantities. Roslyn's group did not return to the program code to read for understanding before the TA came over and questioned the group why the simulation was showing a static picture. Howard answered that the net force was just gravity, so there was some silent interpretation by Howard even though he did not include the group in on his understanding of at least one line from the program code. During the modification task, Howard spoke up again to suggest that the group needed to calculate a spring force which might be why the ball is not interacting with the spring.

Since the incomplete computational model of the MWP produces a visually striking output, viewing the simulation before interpreting the program code only added confusion in the interpretation of the visual output. These episodes of cheating, although discouraging, provided evidence for the route these groups must take to continue on with the activity. Celia's second cheating episode showed that returning to the program code and reading for understanding negated the negative effects of viewing the visual output before reading the program for comprehension.

Chapter 6

Conclusions

I've presented data and analysis of how participants completed instructional tasks that ask the participants to make sense of a minimally working program. These tasks were motivated based on prior research in fostering reading comprehension among novice readers and some unexpected results from a pilot study conducted in May 2008. To investigate how participants use their physics knowledge to interpret VPython programs, we provided participants with example programs that were missing lines of code responsible for calculating the interactions between 3D objects or lines of code responsible for using the interaction to effect the motion of the 3D objects. This chapter will summarize the findings from this study to answer the research question: How do students make sense of incomplete, but functioning VPython programs? This chapter will present the current state of computational activities that make use of minimal working programs including activities which are already using the results from this research project. And, this chapter will also present current and future avenues of research which may build upon these findings.

6.1 Using Physics Knowledge

Participants had very little difficulty interpreting physics quantities according to their variable name, which verified a finding out of Sherin's(1996) work. However, participants had a difficult time interpreting a value for physics quantities that consisted of the multiplication of a string of numbers, such as the initial value for the kinetic energy of the alpha particle in the Rutherford Scattering MWP, or how long the simulation would last when interpreting the loop conditional statement in the Spacecraft-Earth MWP. Participants increasingly identified physics principles throughout the semester, or noted the omission of physics principles which should have appeared in the computational loop. There were few errors in the interpretation of the location of 3D objects or the initial directions of motion.

The line number plots showed that when participants informed their whiteboard predictions from the program code, the `Objects` and `Initial Values` sections of code provided the crux of evidence for the predictions. If participants interpreted physical constants or quantities associated with the calculation of the interaction between two 3D objects, (such as g, qAlpha, ks) participants used the inclusion of these constants as justification that the animation would produce objects which would interact accordingly. Since participants were inferring interactions from the inclusion of constants or the initial values of forces (as in the Mass-Spring MWP), participants didn't gain any information from interpreting the computational loop. Exceptions to this generalization occurred when participants noted the omission of lines of code in the loop which would use the constants to define a force or update the position of 3D objects.

Participants became increasingly proficient at debugging the animation during the evaluation task. Participant solutions were not always appropriate for generating code to modify the MWP (energy approach is not effective) however they focused more on how to add physics principles and define forces between 3D objects rather than attempting to generate a programming command to direct motion as in a Pac-Man® game.

6.2 Real-World Expectations

Participants infused expectations for interactions between 3D objects by drawing upon analogies to real-world physical systems. The best example comes from the prediction of the Spacecraft-Earth MWP where Paine makes an analogy between the Moon-Earth system and this Spacecraft-Earth system. Other analogies occur which greatly influence the prediction of the MWP visual output including: helixes behave like physical springs and oscillate and spheres with mass will bounce off other spheres with mass. These expectations, which come from participant's knowledge about real world systems served to fill in missing lines of code. The Moon-Earth analogy was mentioned when the group was trying to predict how the spacecraft would move through the space. The spring-like helix analogy was mentioned after participants recognized that the gravitational force would pull the mass downwards. The bouncing sphere analogy was suggested after participants recognized that the two spheres would otherwise occupy the same location in space. Although these analogies served as evidence for the incorrect predictions of the visual output, they were productive in determining how to modify the program such that the objects would interact and model the dynamics of real-world physical systems.

Having analyzed how participants made sense of the MWP through the study + prediction tasks, we can revisit the model for instruction introduced as a modified version of Buffler's model for computational activities and say something more about how participants use information to generate a conceptualization of the system described in the MWP code. Since participants introduced knowledge about the real-world dynamics of analogous physical systems to create predictions, the predictions are not entirely based on the program code. As such, the model introduced in Chapter 1 needs some revision in light of these findings. Figure 6.1 shows a new arrow which represents the possibility that students completing these computational activities may interject their knowledge about analogous physical systems. Students also interpret the 3D Objects in the program code as real objects, inheriting the qualities of real-world materials.

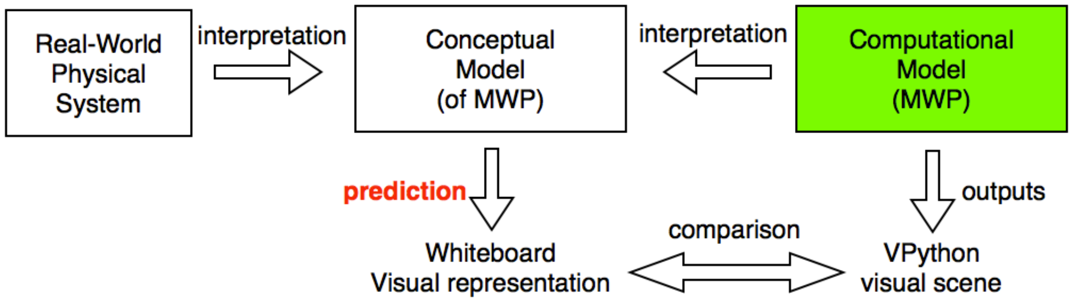


Figure 6.1: Revised model for computational activities using a MWP as a result of the findings of this research study.

Therefore, students inform their conceptual model of the MWP using their interpretation of the evolution of real-world physical systems.

6.3 Effect of research findings on the design of instructional activities

Creating predictions of the MWP program code that used information from knowledge about real-world physical systems is not detrimental to the instructional sequence and appears to have helped participants define their modification goals. However, it's not entirely desirable to have students make predictions where they are drawing from two competing sources of information: the program code and their previous knowledge of the interactions of real-world systems. One solution that now appears in the current state of computational activities which use MWPs is to ask students to create two side-by-side whiteboard predictions. One prediction should only contain information that was interpreted from the program code and the second separate prediction should contain information about how an analogous system with the same objects and initial conditions should evolve in the real-world. Figure 6.2 shows a representation of this side-by-side model. After students evaluate the whiteboard prediction of the computational model, the modification task asks the students to think of the MWP prediction as the initial

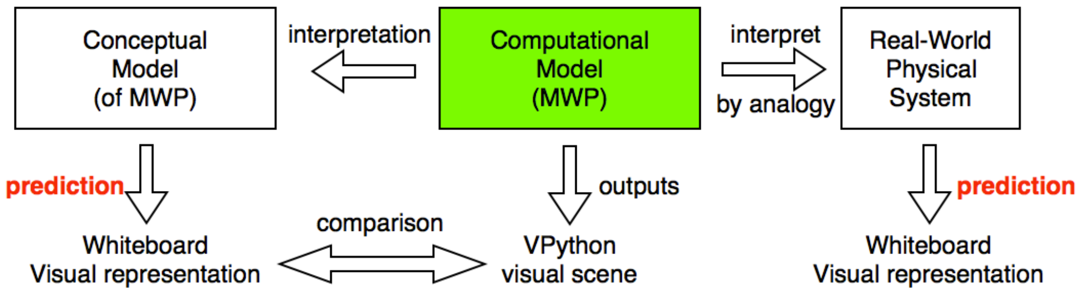


Figure 6.2: Side-by-side predictions model for the interpretation and prediction of the visual output for a MWP. Students predict the features of the visual output as well as the time evolution of an analogous physical system in the real-world.

state of the program and the real-world prediction as the goal state for the visual output.

Additional instructions are now part of the modification task which were added to remind students of the sequence of calculations in the iterative loop which update the momentum and positions of 3D objects. This addition was also a result of reviewing the transcripts of students struggling during the modification task and it was determined that students needed additional scaffolding during the modification task. Students modifying the MWP was beyond the scope of this research question and is left for future work. The versions of the computational activities using MWP included with the instructor resources disk for the first semester of Matter & Interactions are included in the Appendix section of this dissertation.

6.3.1 Effect of findings on the development of VPython

This thesis reports findings which are applicable to programming languages beyond VPython and primarily shows how the interpretation and prediction tasks prompt participants to use their knowledge about real-world systems and the use of physics principles to constrain the motion of objects. However, since participants were using VPython to complete the instructional tasks, there is additional information in the data on how the programming language may interfere with the interpretation tasks. For example, this thesis reported how the `curve` 3D object was interpreted by students as a command for the spacecraft to follow a curved path through space. Due to the preponderance of evidence for this unforeseen interpretation of the `curve` object, the developers of VPython took it upon themselves to add an additional method for tracking the path of 3D objects through the visual scene. VPython 5.70 and beyond now includes a new attribute, make_trail, which serves the same function as the curve object and now appears in the same line of code which creates the 3D object. Therefore it is now possible to add a trail to a 3D object by specifying which object you want the trail to follow without using the word "curve." See `vpython.org` for more information about VPython and documentation about the new `make_trail` object.

6.4 Future Work

These findings and the implications for the design of new instructional activities raise additional questions ripe for future work. Asking students to complete side-by-side predictions may affect how students make sense of the MWP code. If the second whiteboard prediction reduces the competition in only using information from the MWP code to generate a prediction of the visual output, then there should be more evidence of students justifying the whiteboard predictions using the functions of lines of program code. It would be interesting to know if this is indeed the case. Instructional design is a cyclical process with instructional materials adapting to incorporate how students complete tasks and achieve the goals of the instruction.(Reif, 2008) There are open questions with the initial trial of the MWP activities to determine how these participants modified the MWP code and identify the difficulties the participants had with the modification task. Are these difficulties related to how the physics quantities were initially interpreted?

The MWP activities are a solution to the instructional problem of trying to bring novice programmers up to speed in programming knowledge by only asking them to create programs during computational activities at most once per week. It's not clear when the sequence of tasks that use a MWP are no longer challenging as students become more proficient at program comprehension and modification. We saw in this research that participants were more willing to interpret features of the computational loop but we didn't see the participants reach a comprehension of the loop that led to accurate predictions. Therefore there's evidence that this sequence of MWP activities doesn't quite raise the program comprehension to the level where no mistakes are made in interpretation, and further research may shed light as to when students reach an acceptable level of proficiency of few interpretation errors.

References

Abelson, H., & Sussman, G. J. (1984). *Structure and interpretation of computer programs* (Vol. 1). MIT Press.

Adams, W. K., Perkins, K. K., Podolefsky, N. S., Dubson, M., Finkelstein, N. D., & Wieman, C. E. (2006, Jan). New instrument for measuring student beliefs about physics and learning physics: The colorado learning attitudes about science survey. *Phys. Rev. ST Phys. Educ. Res.*, *2*(1), 010101.

Ainsworth, S. (1999). The functions of multiple representations. *Computers & Education*, *33*, 131-152.

Ainsworth, S. (2008). Current perspectives on cognition, learning, and instruction: Recent innovations in educational technology that facilitate student learning. In D. Robinson & G. Schraw (Eds.), (p. 37-67). Information Age Publishing.

Bechtel, W., & Abrahamsen, A. (2005). Explanation: A mechanist alternative. *Mechanisms in Biology: Studies in History and Philosophy of Biological and Biomedical Sciences*, *36*, 421-441.

Brooks, R. (1983). Toward a theory of the comprehension of computer programs. *International Journal of Man-Machine Studies*, *18*, 543-554.

Buffler, A., Pillay, S., Lubben, F., & Fearick, R. (2008, April/May). A model-based view of physics for computational activities in the introductory physics course. *Am. J. Phys.*, *76*(4&5), 431-437.

Byrne, M., Catrambone, R., & Stasko, J. (1999). Evaluating animations as student aids in learning computer algorithms. *Computers & Education*, *33*, 253-278.

Carey, S. (1995). Causal cognition: A multidisciplinary debate. In D. Sperber, D. Premack, & A. Premack (Eds.), (chap. On the origins of causal understanding). New York: Clarendon Press.

Chabay, R. W., & Sherwood, B. A. (2000, Sept./Oct.). Computer modeling in a modern introductory physics course. *Computing in Science & Engineering*, *2*(5), 58-59.

Chabay, R. W., & Sherwood, B. A. (2004, April). Modern mechanics. *Am. J. Phys.*, *72*(4), 439-445.

Chabay, R. W., & Sherwood, B. A. (2006, April). Restructuring the introductory electricity and magnetism course. *Am. J. Phys.*, *74*(4), 329-336.

Chabay, R. W., & Sherwood, B. A. (2008, April/May). Computational physics in the introductory calculus-based course. *Am. J. Phys.*, *76*(4&5), 307-313.

Chi, M., Feltovich, P., & Glaser, R. (1981). Categorization and representation of physics problems by experts and novices. *Cognitive Science*, *5*, 121-152.

Chin, C., & Brown, D. (2000). Learning in science: A comparison of deep and surface approaches. *Journal of Research in Science Teaching*, *37*(2), 109-138.

Christian, W., & Belloni, M. (2000). *Physlets: Teaching physics with interactive curricular materials*. Prentice Hall.

Clement, J. (1982, January). Students' preconceptions in introductory mechanics. *Am. J. Phys.*, *50*(1), 66-71.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, *XX*(1), 37-46.

Conway, M. J. (1997). *Alice: Easy-to-learn 3d scripting for novices.* Unpublished doctoral dissertation, University of Virginia.

Costanza, R. (1987, Feb.). Simulation modeling on the macintosh using stella. *BioScience*, *37*(2), 129-132.

Cox, R. (1999). Representation construction, externalized cognition, and individual differences. *Learning and Instruction*, *9*, 343-363.

Cox, R., & Brna, P. (1995). Supporting the use of external representations in problem solving: The need for flexible learning environments. *Journal of Artificial Intelligence in Education*, *6*(2/3), 239-302.

deJong, T. (2006). Computer simulations: Technology advances in inquiry learning. *Science*, *312*, 532-533.

Dervin, B. (1983, May). *An overview of sense-making research: concepts, methods, and results to date.* International Communication Association annual meeting, Dallas.

Dervin, B. (1992). Qualitative research in information management. In J. D. Glazier & R. R. Powell (Eds.), (chap. From the Mind's Eye of the User: The Sense-Making Qualitative-Quantitative Methodology). Englewood: Libraries Unlimited, INC.

diSessa, A. A. (1982). Unlearning aristotelian physics: A study of knowledge-based learning. *Cognitive Science*, *6*, 37-75.

diSessa, A. A. (1993). Toward an epistomology of physics. *Cognition and Instruction*, *10*(2&3), 105-225.

diSessa, A. A., & Abelson, H. (1986). Boxer: a reconstructible computational medium. *Communications of the ACM*, *29*(9), 859-868.

diSessa, A. A., Abelson, H., & Ploger, D. (1991). An overview of boxer. *Journal of Mathematical Behavior*, *10*, 3-15.

Ericsson, K., & Simon, H. (1993). *Protocol analysis* (2nd ed.). Cambridge, MA: The MIT Press.

Etkina, E., & Heuvelen, A. V. (2007). Investigative science learning environment - a science process approach to learning physics. College Park, MD: American Association of Physics Teachers. Available from `http://www.per-central.org/document/ServeFile.cfm?ID=4988`

Etkina, E., Heuvelen, A. V., White-Brahmia, S., Brookes, D. T., Gentile, M., Murthy, S., et al. (2006). Scientific abilities and their assessment. *Phys. Rev. ST Phys. Educ. Res.*, *2*(020103).

Gaffney, J., Richards, E., Kustusch, M., Ding, L., & Beichner, R. (2008). Scaling up education reform. *Journal of College Science Teaching*, *37*(5).

Geisler, C. (2003). *Analyzing streams of language: Twelve steps to the systematic coding of text, talk, and other verbal data.* New York, NY: Pearson/Longman.

Gunstone, R. (1992). Constructivism and megacognition: Theoretical issues and classroom studies. In R. Duit, F. Goldberg, & H. Niedderer (Eds.), *Research in physics learning: Theoretical issues and empirical studies* (p. 129-140).

Hegarty, M. (n.d.). Dynamic visualizations and learning: getting to the difficult questions. *Learning and Instruction*, *14*, 343-351.

Hegarty, M., Quilici, J., Narayanan, N., Holmquist, S., & Moreno, R. (1999). Designing multimedia manuals that explain how machines work: Lessons from evaluation of a theory-based design. *Journal of Educational Multimedia and Hypermedia*, *8*, 119-150.

Heller, P., & Hollabaugh, M. (1992). Teaching problem solving through cooperative grouping. part 2: Designing problems and structuring groups. *Am. J. Phys.*, *60*(7), 637-644.

Hogan, K., & Thomas, D. (2001). Cognitive comparisons of students' systems modeling in ecology. *Journal of Science Education and Technology*, *10*(4), 319-345.

Hull, M., Kuo, E., Gupta, A., & Elby, A. (2009). Undergraduate engineers' sense-making of math'.

Hulland, C., & Mumby, H. (1994). Science, stories, and sense-making: A comparison of qualitative data from a wetlands unit. *Science Education*, *78*(2), 117-136.

Johnson, I. D., & McPhedran, R. C. (1993). Computational physics in the introductory curriculum. *The Australian and New Zealand Physicist*, *4*(2), 67-73.

Johnson-Laird, P. N. (1986). *Mental models: Towards a cognitive science of language, inference, and consciousness.* Harvard U. P.

Kanim, S. (2001). Connecting concepts to problem-solving. In *Proceedings of the 2001 physics education research conference: Education at the interface* (p. 37-40). Rochester, NY, July 25-26, 2001.

Karelina, A., & Etkina, E. (2007). Acting like a physicist: Student approach study to experimental design. *Phys. Rev. ST Phys. Educ. Res.*, *3*(020106).

Kelly, G., & Crawford, T. (1998). Student's interaction with computer representations: Analysis of discourse in laboratory groups. *Journal of Research in Science Teaching*, *33*(7), 693-707.

Kieras, D. (1992). Diagrammatic displays for engineered systems: effects on human performace in interacting with malfunctioning systems. *International Journal of Man-Machine Studies*, *36*(861-895).

Kung, R. L. (2007). Metacognitive activity in the physics student laboratory: is increased metacognition necessarily better? *Metacognition Learning*, *2*, 41-56.

Kynigos, C. (1995). Computers and exploratory learning. In A. A. diSessa, C. Hoyles, & R. Noss (Eds.), (p. 399-419). Springer.

Landau, R. (2006, September). Computational physics: A better model for physics education? *Computing in Science & Engineering*, *8*, 22-30.

Larkin, J. H., & Simon, H. A. (1987). Why a diagram is (sometimes) worth ten thousand words. *Cognitive Science*, *11*, 65-99.

Leedom, D. (2001). *Sensemaking symposium, final report to the command and control research program* (Tech. Rep.). Office of the Assistant Secretary of Devense for Command, Contro, Comunications and Intelligence, US Dept. of Defense.

Linn, M., Lee, H., Tinker, R., Husic, R., & Chiu, J. (2006). Teaching and assessing knowledge integration in science. *Science*, *313*, 1049-1050.

Lising, L., & Elby, A. (2005, April). The impact of epistemology on learning: A case study from introductory physics. *Am. J. Phys.*, *73*(4), 372 - 382.

Lowe, R. (2004). Interrogation of a dynamic visualization during learning. *Learning and Instruction*, *14*, 257-274.

MacDonald, W. M., Redish, E. F., & Wilson, J. M. (1988, July/Aug). The m.u.p.p.e.t. manifesto. *Computers in Physics*, *2*(4), 23-30.

Machamer, P., Darden, D., & Craver, C. (2000). Thinking about mechanisms. *Philosophy of Science*, *67*(1), 1-25.

McDermott, L. (1991, April). Millikan lecture 1990: What we teach and what is learned - closing the gap. *American Journal of Physics*, *59*(4), 301-315.

McDermott, L. (1993, April). Guest comment: How we teach and how students learn - a mismatch? *American Journal of Physics*, *61*(4), 295-298.

McDermott, L., & Redish, E. F. (1999, September). Resource letter: Per-1: Physics education research. *Am. J. Phys.*, *67*(9), 755-767.

McIntyre, D. H., Tate, J., & Manogue, C. A. (2008, April/May). Integrating computational activities into the upper-level paradigms in physics curriculum at oregon state university. *Am. J. Phys.*, *76*(4&5), 340-346.

M.Kohlmyer. (2005). *Student performance in computer modeling and problem solving in a modern introductory physics course.* Unpublished doctoral dissertation, Carnegie Mellon University.

Otero, V. (2001). *The process of learning about static electricity and the role of the computer simulator.* Unpublished doctoral dissertation, San Diego State University.

Palinscar, A. S., & Brown, A. L. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, *1*(2), 117-175.

Papert, S. (1980). *Mindstorms: Children, coputers, and powerful ideas.* New York: Basic Books.

Park, O., & Gittelmann, S. (1992). Selective use of animation and feedback in computer-based instruction. *Educational Technology, Research, and Development*, *12*, 27-38.

Penner, D. E. (2000). Chapter 1: Cognition, computers, and synthetic science: Building knowledge and meaning through modeling. *Review of Research in Education*, *25*(1), 1-35.

Pennington, N. (1987). Comprehension strategies in programming. In G. M. Olson, S. Sheppard,

& E. Soloway (Eds.), *Empirical studies of programmers: Second workshop* (p. 100-113). Ablex Publishing Corporation.

Pennington, N., Nicolich, R., & Rahm, J. (1995). Transfer of training between cognitive subskills: Is knowledge use specific? *Cognitive Psychology*, *28*, 175-224.

Perkins, K. K., Adams, W., Dubson, M., Finkelstein, N., Reid, S., Wieman, C., et al. (2006). Phet: Interactive simulatins for teaching and learning physics. *Physics Teacher*, *44*, 18-23.

Redish, E., & Wilson, J. (1993, March). Student programming in the introductory physics course: M.u.p.p.e.t. *Am. J. Phys.*, *61*(3), 222-232.

Redish, E. F., Saul, J., & Steinberg, R. (1998). Student expectations in introductory physics. *American Journal of Physics*, *66*(3), 212-224. Available from http://dx.doi.org/10.1119/1.18847

Reif, F. (2008). *Applying cognitive science to education*. Cambridge, MA: The MIT Press.

Resnick, L. (1989). The teaching and assessing of mathematical problem solving. In R. Charles & E. Silver (Eds.), (p. 32-60). Hillsdale, NJ: Erlbaum.

Richards, E. (2010). *Selecting the correct solution to a physics problem when given several possibilities*. Unpublished doctoral dissertation, North Carolina State University.

Rieber, L. (1990). The effects of computer animated graphics with science instruction with children. *Journal of Educational Computing Research*, *82*, 135-140.

Roos, K. R. (2006). An incremental approach to computational physics education. *Computing in Science & Engineering*, *8*, 44-50.

Russ, R., Scherr, R., Hammer, D., & Mikeska, J. (2008). Recognizing mechanistic reasoning in student scientific inquiry: A framework for discourse analysis developed from philosophy of science. *Science Education*, *92*(3), 499-426.

Schecker, H. (1993). Learning physics by making models. *Physics Education*, *28*, 102-106.

Scherer, D., Dubois, P., & Sherwood, B. A. (2000, Sept./Oct.). Vpython: 3d interactive scientific graphics for students. *Computing in Science & Engineering*, *2*(5), 56-62.

Scherr, R., & Elby, A. (2006). Enabling informed adaptation of reformed instructional materials. In L. McCullough, L. Hsu, & P. Heron (Eds.), *2006 physics education research conference* (p. 46-49).

Scherr, R., & Hammer, D. (2009, April). Student behavior and epistemological framing: Examples from collaborative active-learning activities in physics. *Cognition and Instruction*, *27*(2), 147-174.

Schoenfeld, A. (1992). Learning to think mathematically: Problem solving, metacognition, and sense-making in mathematics. In D. Grouws (Ed.), (p. 334-370). New York:MacMillan.

Sherin, B. L. (1996). *The symbolic basis of physical intuition: A study of two symbol systems in physics instruction.* Unpublished doctoral dissertation, University of California, Berkeley.

Sherin, B. L. (2001). A comparison of programming languages and algebraic notation as expressive languages for physics. *Int. J. Comp. Math Learn.*, *6*, 1-61.

Sherin, B. L., diSessa, A. A., & Hammer, D. (1993). Dynaturtle revisited; learning physics through collaborative design of a computer model. *Interactive learning environments*, *3*(2), 91-118.

Shneiderman, B., & Mayer, R. (1979). Syntactic/semantic interactions in programmer behavior: A model and experimental results. *International Journal of Computer and Information Sciences*, *8*, 219-238.

Singh, C., Belloni, M., & Christian, W. (2006). Improving students' understanding of quantum mechanics. *Physics Today*, *59*(8), 43-49. Available from `http://link.aip.org/link/?PTO/59/43/1`

Soloway, E., B.Adelson, & Ehrlich, K. (1988). Knowledge and processes in the comprehension of computer programs. In M. Chi, R. Glaser, & M. Farr (Eds.), *The nature of expertise* (p. 129-152). Hillsdale, NJ: Lawrence Erlbaum.

Syslo, M. M., & Kwiatkowska, A. B. (2008). The challenging face of informatics education in poland. In R. Mittermeir & M. Syslo (Eds.), (Vol. Informatics Education-Supporting

Computational Thinking, p. 1-18). Springer.

Taylor, J. R., & King, B. A. (2006, September). Using computational methods to reinvigorate an undergraduate physics curriculum. *Computing in Science & Engineering*, *8*, 38-43.

Timberlake, T., & Hasbun, J. (2008, April/May). Computation in classical mechanics. *Am. J. Phys.*, *76*(4&5).

Tversky, B., Bauer Morrison julie, & Betrancourt, M. (2002). Animation: can it facilitate. *Int. J. Human-Computer Studies*, *57*, 247-262.

Weick, K., Sutcliffe, K., & Obstfeld, D. (2005). Organizing and the process of sensemaking. *Organization Science*, *16*(4), 409-421.

Wilkin, B. (1997). *Learning from explanations: diagrams can "inhibit" the self-explanation effect.* Menlo Park, CA: AAAI Press.

Williams, D. C., Ma, Y., Prejean, L., Ford, M. J., & Lai, G. (2008). Acquisition of physics content knowledge and scientific inquiry skills in a robotics summer camp. *Journal of Research on Technology in Education*, *40*(2), 201-216.

Wing, J. M. (2006, March). Computational thinking. *Communications of the ACM*, *49*(3), 33-35.

Zacks, J. M., Tversky, B., & Iyer, G. (2001). Perceiving, remembering, and communicating structure in events. *Journal of Experimental Psychology: General*, *130*(1), 29-58.

APPENDICES

Pilot Script - 205

Welcome to my research study, and thank you for participating! I greatly appreciate your help.

[Ensure that the first part of the consent form is signed and dated]

If you have a cell phone here, please turn it off (it can interfere with the wireless mic).

Here is a reference sheet that you can use at any time [gesture to the reference sheet]. You can also use this calculator at any time [gesture to the calculator]. If you want to write anything, please use the whiteboard [gesture to the whiteboard and marker]. Please don't erase anything; just cross out your work. Above and behind you is a video camera that will record the session. Only NC State researchers will see the videotape unless you give me permission to show the tape to others. Once the session is done, you can fill out the last part of the consent form specifying how the video can be used.

<u>Administer ROT</u>

First, I'll ask you to take this test. Do the best you can. You may have as many as 10 minutes to complete all 20 questions.

<u>Warm-Up Exercises – 8 second rule for silence</u>

Now, I'll ask you to complete three warm-up exercises to practice saying your thoughts while you work on different types of tasks.

[Give Warmup Task 1]

[Give Warmup Task 2] Comment on participant's speaking

[Give Warmup Task 3]

[Extra Tasks as Needed]

<u>Study Spacecraft.py Program</u>

Next, study this program that will produce an animation of the Spacecraft around the Earth. If you have any questions about syntax, please let me know as I may be able to answer them. As you study, remember to say aloud everything you read and think, it doesn't matter if you say something several times.

After you feel you are comfortable with the program, let me know and I will ask you to complete a task.

[Open Spacecraft.py program]

[Wait for questions and indication that participant is ready to move on]

Change the velocity of the craft such that it is moving in an elliptical orbit rather than a circular orbit.

[Let student complete task]

Is there a point in the orbit where the spacecraft's momentum is perpendicular to the net force? Where?

Build Moon-Earth program

Now, open the program MoonEarth.py. What do you notice is different in this program from the Spacecraft.py program?

[Let student complete task]

The next task is to make it possible for the spacecraft to make a figure-8 orbit around both the Earth and the Moon.

[Let student complete task. If student gets stuck, suggest allowing the student to review the Spacecraft.py program]

OR

Figure-8 Moon-Earth program

Next, study this program that will produce an animation of the Spacecraft around the Earth with the Moon in the distance. If you have any questions about syntax, please let me know as I may be able to answer them. As you study, remember to say aloud everything you read and think, it doesn't matter if you say something several times.

After you feel you are comfortable with the program, let me know and I will ask you to complete a task.

[Open the program MoonEarthII.py.]

[Wait for questions and indication that participant is ready to move on]

Your job is to give the spacecraft enough speed in the y-direction such that it makes a figure-8 orbit around both the Moon and the Earth.

[Let student complete task]

Construct BinaryStar.py program

Close the MoonEarthII.py program and open the program called BinaryStar.py. Your task is to create a program that allows two stars to orbit each other beginning from a shell that creates two spheres. Remember to say aloud everything you read and think, it doesn't matter if you say something several times. Work as far as you can.

[Allow participant to write code. If student struggles, let student refer back to the MoonEarthII.py program.]

[]

Interview Questions

1. Specific questions pertaining to participant actions.
2. VPython experiences:
    a. Do you enjoy programming?
    b. What is your programming experience?
    c. How comfortable are you with programming?

```
#Spacecraft.py program from pilot study. Participants used this program to complete Task 1 from the study.

#from __future__ import *
from visual import *
#from __future__ import *
from visual.graph import *
scene.width =545
scene.height = 545


#CONSTANTS
G = 6.7e-11
mCraft = 1.7e22
mEarth = 6e24
deltat = 10

#OBJECTS AND INITIAL VALUES
Earth = sphere(pos=vector(0.,0.,0.), radius=6.4e6, color=color.cyan)
Craft = sphere(pos=vector(-10*Earth.radius,0.,0.), radius=1.75e6, color=color.yellow)
trail = curve(color=Craft.color)
scene.autoscale = 1

vCraft = vector(0,1000,0)
pCraft = mCraft*vCraft
t = 0

#CALCULATIONS
while t < 60*365*24*60*60.:
    rate(2000)       ## slow down motion to make animation look nicer

    r=Craft.pos-Earth.pos
    rmag=sqrt(r.x**2+r.y**2+r.z**2)
    rhat = r/rmag

    Fgrav=-(G*mEarth*mCraft)/(rmag)**2
    Fnet = Fgrav * rhat
    pCraft = pCraft + Fnet * deltat
    Craft.pos = Craft.pos + (pCraft/mCraft)*deltat

    trail.append(pos=Craft.pos)

    t = t+deltat

print 'Calculations finished after ',t,'seconds'
```

A Spacecraft Voyage

Part 1: Spacecraft and Earth

45 min

OBJECTIVES

In this VPython program you will model the motion of a spacecraft traveling around the Earth. To do this you will iteratively (repeatedly) :

- calculate gravitational force on the spacecraft by the Earth
- apply the momentum principle to update the momentum of the spacecraft
- update the position of the spacecraft

You will use your working program to explore the effect of the spacecraft's initial velocity on its trajectory.

GROUP ROLES

Before you begin, you should agree on the responsibilities of the Manager, the Recorder, and the Skeptic for this activity.

Data for this and subsequent programs:

Mass of Earth: 6e24 kg	Radius of Earth: 6.4e6 m
Mass of spacecraft: 15e3 kg	Radius of spacecraft: very small (exaggerated in program)
Mass of Moon: 7e22 kg	Radius of Moon: 1.75e6 m
Distance from Earth to Moon: 4e8 m	G = 6.7e-11 N m**2/kg**2

You may find it useful to refer to previous programs in which you calculated gravitational forces, and predicted motion iteratively.

EXAMPLE PROGRAM

An example program has been provided in WebAssign, and you can copy this code and paste it into an IDLE window, and save it in My Documents. Remember to give it the extension ".py".

1. Study the Example Program

Read through the program shell together. Make sure everyone in the group understands the function of each VPython instruction. Answer the following question before going on:

Q1: Draw on the whiteboard what you would expect to see once you run the program.
Q2: Which line(s) are responsible for allowing the spacecraft to move through space?

2. Modify the Example Program

- **What is needed in order to get the spacecraft to orbit the Earth? First, draw the orbit you want the visualization to display on your whiteboard.**
- **Modify the example program with calculations that will allow the program to instruct the spacecraft to orbit the Earth.**
- **Use the symbolic names defined in the program shell. You may need to add other symbolic names in your calculations.**
- **Are you making any assumptions in your program? If so, what are they?**
- **Do this, and run your program. Is the behavior what you expected?**

3. Detecting a collision

If your spacecraft collides with the Earth, the program should stop. Add code similar to this inside your loop (using the name you defined for the distance between the spacecraft and the center of the Earth):

```
if rmag < Earth.radius:
     break
```

This code tells VPython to get out of the loop if the spacecraft touches the Earth.

4. Answer these questions about the effect of initial velocity on the motion on your whiteboard

- **Approximately, what initial speed is required to make a circular orbit around the Earth? You may wish to zoom out to examine the orbit more closely.**
  - **WHY DOES CHANGING THE INITIAL VELOCITY HAVE AN EFFECT ON THE ORBIT?**
- **Approximately, what minimum initial speed is required so that the spacecraft "escapes" and never comes back? You may have to zoom out to see whether the spacecraft shows signs of coming back. You may also have to extend the time in the while statement.**

Run the program with an initial velocity that produces an elliptical orbit.

5. Answer these questions about the changes in the spacecraft's momentum on your whiteboard.

- **For this elliptical orbit, what is the direction of the spacecraft's momentum vector? Tangential? Radial? Something else?**
- **What happens to the momentum as the spacecraft moves away from the Earth?**
- **As it moves toward the Earth?**
- **Why? Explain these changes in momentum in terms of the Momentum Principle. Be careful: The Momentum Principle does NOT say "big force -> big momentum".**
- **Compare your answers to those of another group.**

Turn in your program. Make sure your program file name ends in ".py"

A spacecraft voyage, part 1: Spacecraft and Earth

OBJECTIVES
In this program you will model the motion of a spacecraft.

You will use your working program to explore the effect of the spacecraft's initial velocity on its trajectory.

TIME
You should finish this activity in 45 minutes or less.

GROUP ROLES
Before you begin, you should agree on the responsibilities of the Manager, the Recorder, and the Skeptic for this part of the lab.
Record your answers to the QUESTIONS sections; you will need them later.

A WORKING EXAMPLE PROGRAM

An example program has been provided in WebAssign. Right click on the linked file and save it to your Desktop. Remember to give it the extension ".py".

1 Study the Program - DO NOT Run It Yet (That's Cheating!)

Read through the program together. Make sure everyone in the group understands the function of each VPython instruction.

Draw on the whiteboard what you expect to see once you run the program.

Whiteboard Prediction

2 Run the Program

- Is your prediction okay? Did something happen that you didn't expect to happen?

3 Modify the Program

- **First, draw on the whiteboard how you want the spacecraft to interact with the Earth.**
- **What calculations are needed in order to get the spacecraft and the Earth to interact?**
- **Modify the program with these calculations. Where should you place these new lines of code in the program?**
- **Use symbolic names defined in the program. You may need to add other symbolic names in your calculations.**
- **Are you making any assumptions in your program? If so, what are they?**
- **Finally, run the program. Is the behavior what you expected? Do you need to change your code?**

4 Detecting a collision

If your spacecraft collides with the Earth, the program should stop. Add code similar to this inside your loop (using the name you defined for the distance between the spacecraft and the center of the Earth):

```
if rmag < Earth.radius:
   break
```

This code tells VPython to get out of the loop if the spacecraft touches the Earth.

5 Visualize the momentum vector with an arrow

In a previous program, you used arrows to visualize gravitational force vectors. In this program, you will use an arrow to visualize the momentum of the spacecraft.

You will create an `arrow` before the `while` loop and update its position and axis inside the loop, as the spacecraft's momentum changes. You need to know the approximate magnitude of the momentum in order to be able to scale the arrow to fit into the scene, so just before the loop, print the momentum vector:

```
print 'p=', pcraft
```

In the **#CONSTANTS** section of your program, add this scale factor

```
pscale = 0.1
```

Also insert the following statement **in the #OBJECTS section of your program (NOT inside the loop):**

```
parr = arrow(color=color.green)
```

This statement creates an arrow object with default position and axis attributes. You will set these attributes inside the loop.

- **Comment out any print statements which are inside your loop because they slow down plotting.**
- **Inside your loop, update the pos attribute of the parr arrow object to be at the center of the spacecraft, and on a separate line update its axis attribute to represent the current vector value of the momentum of the spacecraft (multiplied by pscale).**
```
parr.pos = ?
parr.axis = ?
```
- **You may have to adjust the scale factor once you have seen the full orbit.**

6 Answer these questions about the changes in the spacecraft's momentum

- **For this elliptical orbit, what is the direction of the spacecraft's momentum vector? Tangential? Radial? Something else?**
- **What happens to the momentum as the spacecraft moves away from the Earth?**
- **As it moves toward the Earth?**
- **Why? Explain these changes in momentum in terms of the Momentum Principle.**
- **Compare your answers to those of another group.**

7 Answer these questions about the effect of initial velocity on the motion

- **Approximately, what initial speed is required to make a circular orbit around the Earth? You may wish to zoom out to examine the orbit more closely.**
- **Why does changing the initial velocity have an effect on the orbit?**
- **Approximately, what minimum initial speed is required so that the spacecraft "escapes" and never comes back? You may have to zoom out to see whether the spacecraft shows signs of coming back. You may also have to extend the time in the while statement.**

```
from __visual__ import division
from visual import *
scene.width =800
scene.height = 800

#CONSTANTS
G = 6.7e-11
mEarth = 6e24
mcraft = 15e3
deltat = 60

#OBJECTS AND INITIAL VALUES
Earth = sphere(pos=vector(0,0,0), radius=6.4e6, color=color.cyan)
craft = sphere(pos=vector(-10*Earth.radius, 0,0), radius=1e6, color=color.yellow)
vcraft = vector(0,2e3,0)
pcraft = mcraft*vcraft
trail = curve(color=craft.color)        ## produces a trail behind the craft

t = 0

while t < 10*365*24*60*60:
        rate(100)                    ## slow down motion to make animation look nicer
        craft.pos = craft.pos + (pcraft/mcraft)*deltat
        trail.append(pos=craft.pos)  ##this adds the new position of the spacecraft to the trail
        t = t + deltat
```

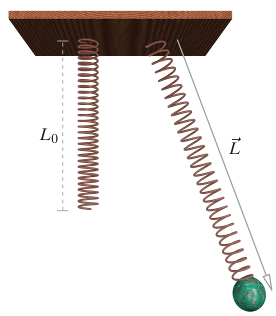
Computer Model of Spring-Mass System

Questions:
You should be able to answer these questions by the end of this lab.

1. Using your measured data on the mass and spring you used in a previous experiment, will VPython accurately predict the oscillation period you measured in your experiment?
2. What initial conditions must you specify in your program in order to get your virtual mass-spring system to oscillate in all three dimensions, instead of just staying in one plane?
3. How does energy flow within the system consisting of a mass and spring?

1 The Spring Force as a Vector

To write an equation for the spring force as a vector, combining magnitude and direction into one expression, we need to consider the general case, in which the spring may be stretched both vertically and horizontally. Since the force exerted by the spring will be directed along the axis of the spring, we need a unit vector in that direction. We will get this unit vector from the vector $\vec{L}$, which is a relative position vector: it specifies the position of the movable end of the spring relative to the fixed end of the spring. $\vec{L}$ extends from the point at which the spring is attached to a support to the mass at the other end of the spring, as shown in the diagram below. L_0 is the length of the relaxed spring.



We can factor the relative position vector $\vec{L}$ into the product of a magnitude (the length of the spring) and a unit vector:

$$\vec{L} = |\vec{L}|\hat{L}$$

The stretch of the spring can now be written as:

$$s = |\vec{L}| - L_0$$

- When the spring is longer than its relaxed length, the value of s will be positive, but the direction of the force will be given by $-\hat{L}$.
- When the spring is shorter than its relaxed length, the value of s will be negative, but the direction of the force will be given by $+\hat{L}$.

The general expression for the vector spring force is therefore:

THE VECTOR SPRING FORCE

$$\vec{F} = -k_s s \hat{L}$$

where
L_0 is the length of the relaxed spring
The vector $\vec{L}$ extends from the point of attachment of the spring to the mass at the other end
The stretch s may be positive or negative

1.1 Example

Suppose the stiffness of the rightmost spring shown in the diagram above is $9\,\mathrm{N/m}$, and its relaxed length is 21 cm. At the instant shown, the location of the green mass is $\langle .07, -0.33, 0\rangle$ m relative to an origin at the point of attachment of the spring. What is the force exerted by the spring on the green mass at this instant?

1.2 Solution

$$\begin{aligned}
\vec{L} &= \langle 0.07, -0.33, 0.0\rangle\,\mathrm{m} - \langle 0, 0, 0\rangle\,\mathrm{m} = \langle 0.07, -0.33, 0.0\rangle\,\mathrm{m} \\
|\vec{L}| &= \sqrt{(.07\,\mathrm{m})^2 + (-.33\,\mathrm{m})^2} = 0.337\,\mathrm{m} \\
\hat{L} &= \frac{\langle 0.07, -0.33, 0.0\rangle\,\mathrm{m}}{0.337\,\mathrm{m}} = \langle 0.208, -.979, 0\rangle \\
s &= 0.337\,\mathrm{m} - 0.21\,\mathrm{m} = 0.127\,\mathrm{m} \\
\vec{F} &= -(9\,\mathrm{N/m})(0.127\,\mathrm{m})\langle 0.208, -.979, 0\rangle = \langle -.238, 1.12, 0\rangle\,\mathrm{N/m}
\end{aligned}$$

Check: The x component of the force is negative, and the y component is positive, as they should be.

To check your understanding of how to calculate the vector spring force, complete the WebAssign problem before moving on in the instructions.

On Your Whiteboard

1. Explain & Predict Program Visualization

- Work with your group to **explain** the workings of the program.
- Make a **prediction** on a whiteboard of what you will see in the visualization when you run it.
- Run your program after everyone agrees on the prediction. Discuss how closely your prediction matches what you see in the visualization.

Checkpoint: Describe to your TA how you can recreate this simulation with lab equipment.

2. Extend Program to Include Additional Interactions

Change the values in the program for the ball's mass, spring constant, and relaxed length, using the recorder's data from Lab 04.

Modify this program to simulate what happens when you connect the ball to the spring and release it from rest. It may help you to plan what types of additional calculations are needed in the program to achieve this goal. Try the same strategy you used to develop the gravitational force calculation in Lab 04:

- List the quantities needed to calculate the total interaction.
- Determine where each new calculation should go in the program.
- Define the quantities in terms of the attributes of 3D objects and other variables or constants that already exist.
- Predict what you would expect to see when you run the program.
- Run the program and make any necessary changes to the code to achieve your goal.

3. Change the Initial Position of the Ball.

Change the initial position of your ball to $\langle 0, -0.5, 0\rangle$. Run the program and pay attention to the amplitude of the oscillation. Next, determine two ways in which you can modify the program's initial conditions such that the amplitude is less:

TA Checkpoint: Explain why the amplitude changed by thinking about what happened during the first two iterations of the `while` loop.

4. Using a graph to answer Question 1 (page 1):

Have your program produce a graph of the y-coordinate of the ball's position vs. time. Use this graph to determine the period of the oscillating system in your computer model. Here are some reminders on how to do this:

In the `##objects` section of your program, create a gcurve object, for plotting the position of your ball:

```
ygraph = gcurve(color=color.yellow)
```

Inside the loop, after updating the momentum and position of the ball, and the time, add the following statement:

```
ygraph.plot(pos=(t, ball.pos.y))
```

Answer the following questions (related to Question 1 on page 1):

- Why doesn't the graph cross zero?
- What is the period of the oscillations shown on the graph? (You will find it easier to read the period off the graph if you change the while $t < 60$: statement to make the program quit after a small number of oscillations.)
- How does the period of your model system compare with the period you measured for your real system?
- What does the analytical solution for a spring-mass oscillator predict for the period?
- Should these numbers be the same? If they are not, why not?
- Make the mass 4 times bigger. What is the new period? Does this agree with theory? (Afterwards, reset the mass to its original value.)
- Make the spring stiffness 4 times bigger. What is the new period? Does this agree with theory? (Afterwards, reset the spring stiffness to its original value.)
- Make the amplitude twice as big. (How?) What is the new period? Does this agree with theory? (Afterwards, reset the amplitude to its original value.)
- Write down the answers to these questions, since you will need them later.

5. 3D Motion

a. Comment out the y vs. time graph defined in the ##objects section and in the loop.

b. Add the following lines of code to produce a trail tracking the motion of the ball:

```
trail = curve(color=ball.color) ## in ##objects section
trail.append(pos=ball.pos) ## inside the loop, after the position update formula
```

c. Find initial conditions that produce oscillations not confined to a single plane. Zoom and rotate to make sure the oscillations are not planar.

```
###############################################################################
####                                                                       ####
####         DO NOT RUN THIS PROGRAM UNTIL YOU ARE ASKED TO DO SO          ####
####                          IN THE INSTRUCTIONS                          ####
####                                                                       ####
###############################################################################
from __future__ import division
from visual import *
from visual.graph import *
scene.width=600
scene.height = 760

## constants and data
g = 9.8
mball = 1       ## change this to Mass 2 (in kg) for your spring color.
L0 = 0.26        ## this is an approximate relaxed length of your spring in meters calculated in lab.
ks = 1          ## change this to the spring constant you measured (in N/m)
deltat = .01

t = 0           ## start counting time at zero

## objects
ceiling = box(pos=(0,0,0), size = (0.2, 0.01, 0.2))            ## origin is at ceiling
ball = sphere(pos=(0,-0.3,0), radius=0.025, color=color.orange) ## note: spring initially compressed
spring = helix(pos=ceiling.pos, axis = vector(0,-L0,0), color=color.cyan, thickness=.003, coils=40, radius=0.015) ## change the color to
be your spring color


## initial values
pball = mball*vector(0,0,0)
Fgrav = mball*g*vector(0,-1,0)

## improve the display
scene.autoscale = 0              ## don't let camera zoom in and out as ball moves
scene.center = vector(0,-L0,0)   ## move camera down to improve display visibility
scene.mouse.getclick()           ## Simulation doesn't start when you Run Program. Must also click the simulation with mouse to start.

## calculation loop

while t < 30:
    rate(100)
    Fnet = Fgrav
    pball = pball + Fnet*deltat
    ball.pos = ball.pos + pball/mball*deltat
    t = t + deltat
```

Computer Model of a Collision

On Your Whiteboard

1. Explain & Predict Program Visualization

- Work with your group to **explain** the workings of the program.
- Make a **prediction** on a whiteboard of what you will see in the visualization when you run it.
- Run your program after everyone agrees on the prediction. Discuss how closely your prediction matches what you see in the visualization.

2. Extend Program to Include Additional Interactions

Modify this program to simulate what should happen to the alpha particle and gold nucleus as the alpha particle approaches.

- List the quantities needed to calculate the interactions.
- Determine where each new calculation should go in the program.
- Define the quantities in terms of the attributes of 3D objects and other variables or constants that already exist.
- Predict what you would expect to see when you run the program.
- Run the program and make any necessary changes to the code to achieve your goal.

3. Momentum Plots

On Your Whiteboard

- Draw a plot of the x component of momentum for the alpha particle, gold nucleus, and total momentum of the system vs. time.
- Draw a plot of the y component of momentum for the alpha particle, gold nucleus, and total momentum of the system vs. time.

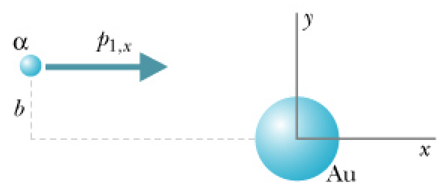
TA Checkpoint: Show your plots to the TA. Afterward, add the code on the next page to your program above the while loop to display these plots.

```
#Display
gdisplay(xtitle='Time', ytitle='Px', x=0, y=scene.height, width=scene.width/2, height=200)
Alphapx=gcurve(color=color.blue)
Aupx=gcurve(color=color.yellow)
totalpx=gcurve(color=color.green)

gdisplay(xtitle='Time', ytitle='Py', x=scene.width/2, y=scene.height, width=scene.width/2, height=200)
Alphapy=gcurve(color=color.blue)
Aupy=gcurve(color=color.yellow)
totalpy=gcurve(color=color.green)

#Add following in the loop after time update
Alphapx.plot(pos=(t,pAlpha.x))
Alphapy.plot(pos=(t,pAlpha.y))
Aupx.plot(pos=(t,pAu.x))
Aupy.plot(pos=(t,pAu.y))
totalpx.plot(pos=(t,pAu.x+pAlpha.x))
totalpy.plot(pos=(t,pAu.y+pAlpha.y))
```

- Compare the plots generated by the program with your predictions.

4. Impact Parameter

The impact parameter b is the distance between centers of the alpha particle and the gold nucleus, perpendicular to the incoming momentum, as shown below.



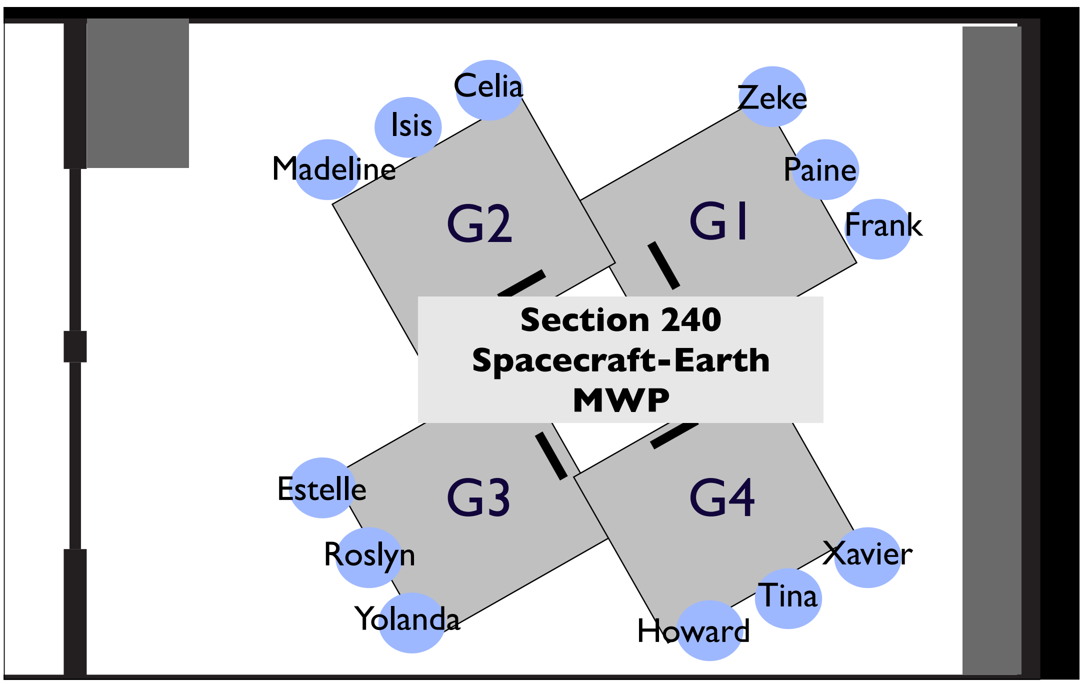
- Set $b = 5{\times}10^{-15}$ m in the initial values section of the program.
- Use b to change the initial position of the Alpha particle.
- How will your p_y graph change due to the new impact parameter?
- Run your program.

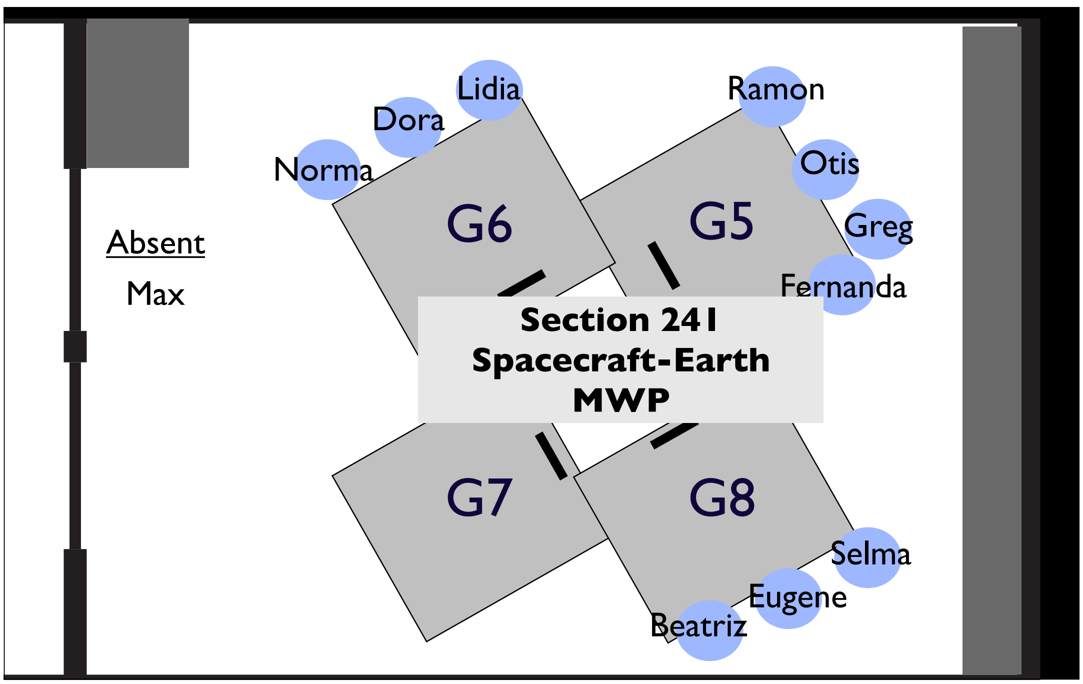
5. Scattering Angle

- The scattering angle is the angle measured between the $+x$ axis and the momentum vector of the scattered alpha particle. In VPython, the scattering angle in degrees is `atan2(py,px)*180/pi`.
- Determine the scattering angle for the alpha particle for the following impact parameters:

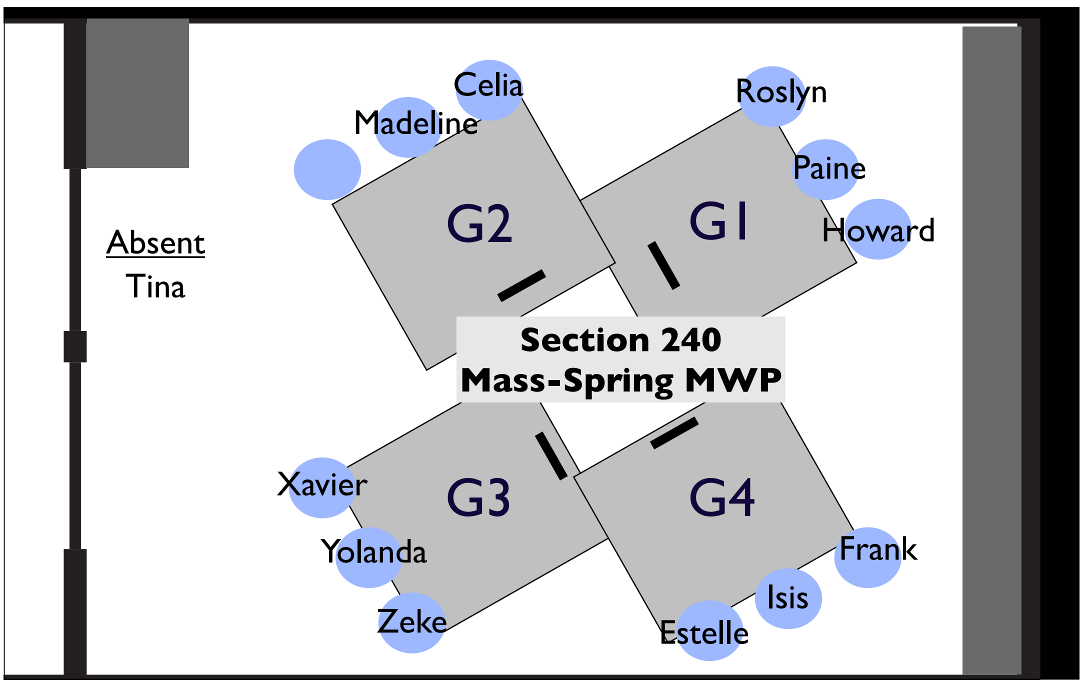
$b_1 = 5{\times}10^{-15}$ m
$b_2 = 10{\times}10^{-15}$ m
$b_3 = 100{\times}10^{-15}$ m

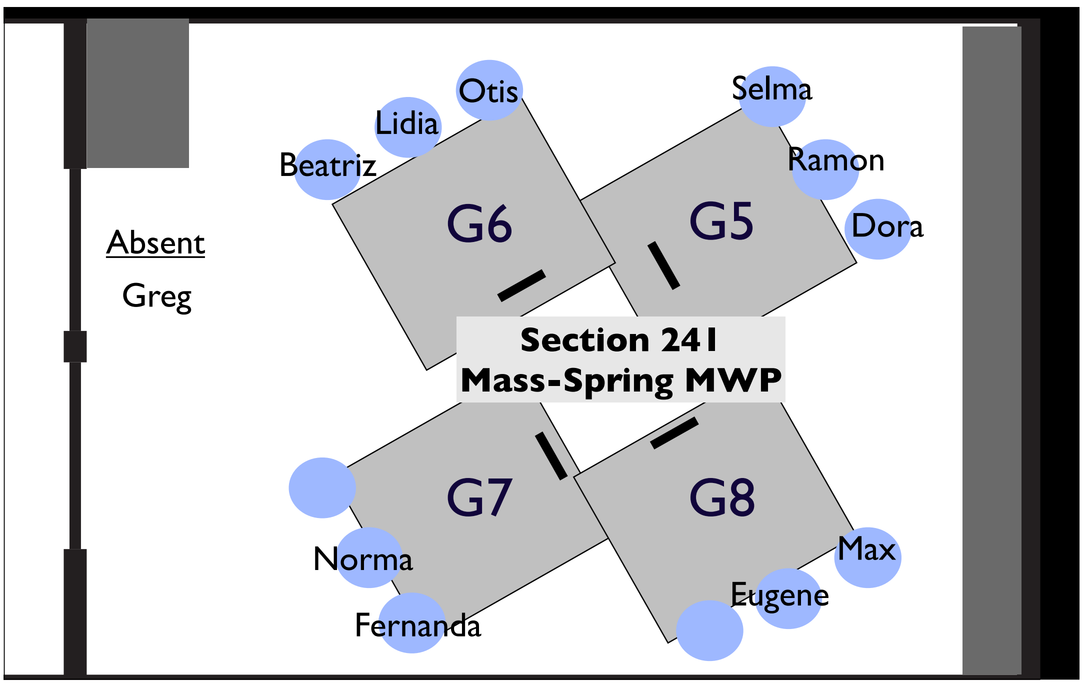
```
################################################################################
####                                                                        ####
####         DO NOT RUN THIS PROGRAM UNTIL YOU ARE ASKED TO DO SO           ####
####                            IN THE INSTRUCTIONS                         ####
####                                                                        ####
################################################################################
from __future__ import division
from visual import *
from visual.graph import *

##Constants
massAu = (79+118)*1.7e-27
massAlpha = (2+2)*1.7e-27
qAu = 2*1.6e-19
qAlpha = 79*1.6e-19
oofpez = 9e9
deltat = 1e-23

##Objects
scene.width=600
scene.height=600
Au = sphere(pos=vector(0,0,0), radius=8e-15, color=color.yellow)
Alpha = sphere(pos=vector(-1e-13,0,0), radius=2e-15, color=color.blue)
trailAu = curve(color=Au.color)
trailAlpha = curve(color=Alpha.color)

#Initial Values
pAu=massAu*vector(0,0,0)
K_Alpha = 10*1e6*1.6e-19            #10 MeV in Joules
pAlpha=vector(sqrt(2*K_Alpha*massAlpha),0,0)
t = 0


while t<1e-20:
    rate(1000)
    Alpha.pos=Alpha.pos+(pAlpha/massAlpha)*deltat

    trailAlpha.append(pos=Alpha.pos)
    trailAu.append(pos=Au.pos)

    t=t + deltat
```

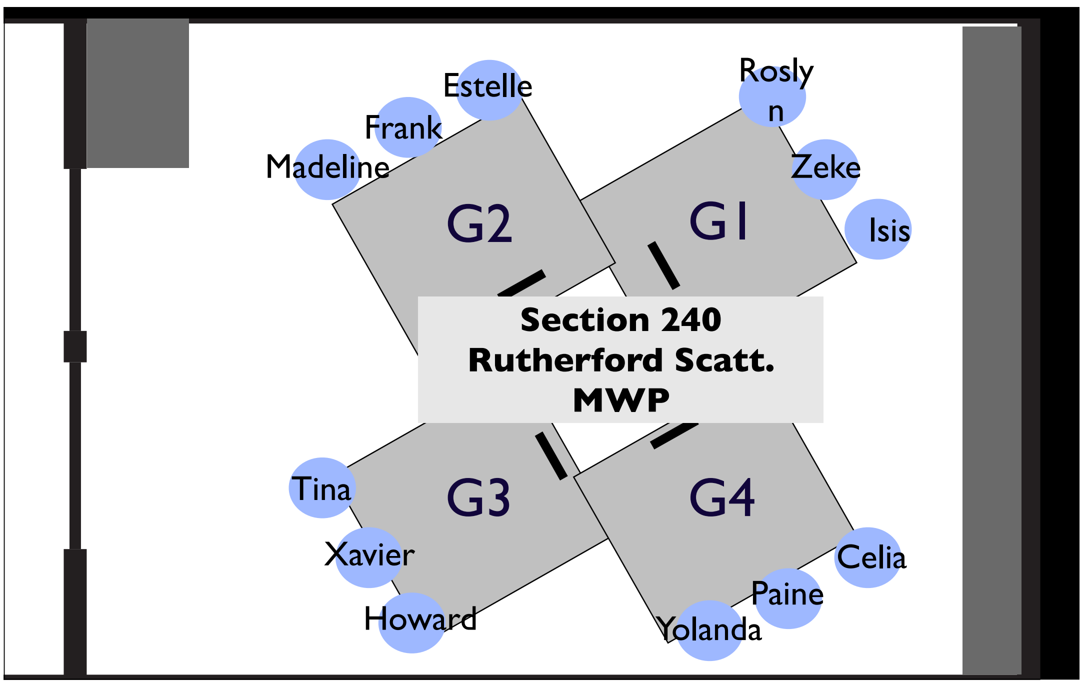

Celia
Isis
Madeline
Zeke
Paine
Frank
G2
G1
Section 240
Spacecraft-Earth
MWP
Estelle
G3
G4
Roslyn
Yolanda
Xavier
Tina
Howard

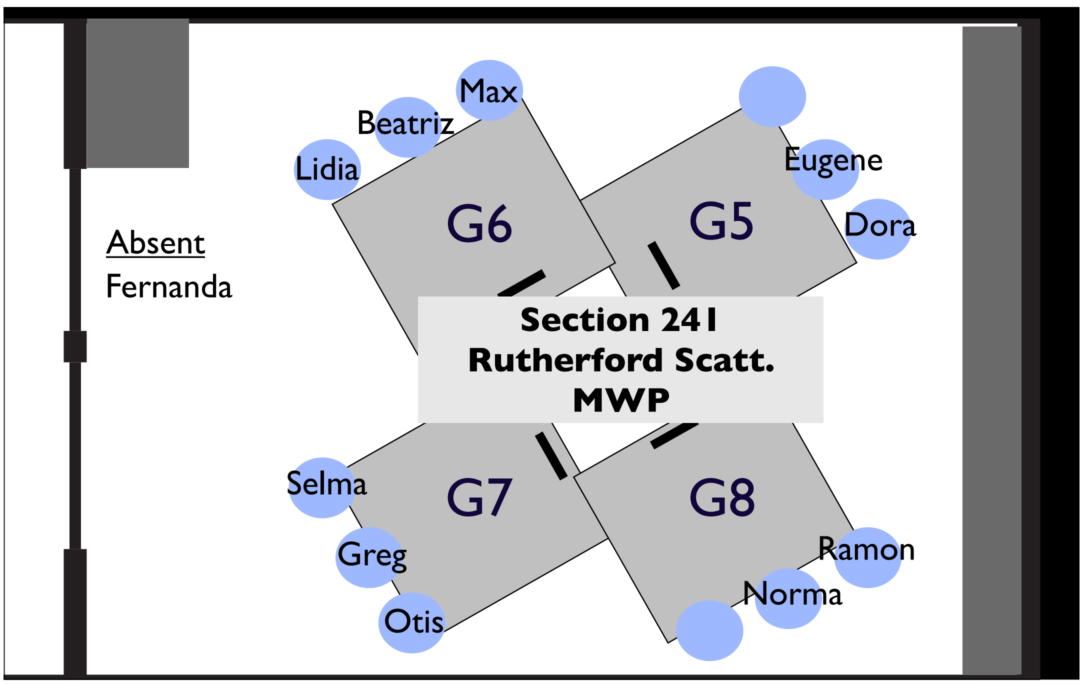


Absent
Max
Lidia
Dora
Norma
G6
Ramon
Otis
Greg
Fernanda
G5
Section 241
Spacecraft-Earth
MWP
G7
G8
Selma
Eugene
Beatriz


Celia
Madeline
Roslyn
Paine
Howard
G2
G1
Absent
Tina
Section 240
Mass-Spring MWP
G3
G4
Xavier
Yolanda
Zeke
Frank
Isis
Estelle


Otis
Lidia
Beatriz
G6
Selma
Ramon
Dora
G5
Absent
Greg
Section 241
Mass-Spring MWP
G7
Norma
Fernanda
G8
Max
Eugene


Estelle
Rosly
n
Frank
Madeline
Zeke
G2
G1
Isis
Section 240
Rutherford Scatt.
MWP
Tina
G3
G4
Xavier
Celia
Paine
Howard
Yolanda


Max
Beatriz
Lidia
G6
G5
Eugene
Dora
Absent
Fernanda
Section 241
Rutherford Scatt.
MWP
Selma
G7
G8
Greg
Otis
Ramon
Norma

Revised February 3, 2009

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
SUBMISSION FOR NEW STUDIES

GENERAL INFORMATION

1. **Date Submitted:** *April 9, 2009* 1a. **Revised Date:** _____	
2. **Title of Project:** *Student Learning in a Modified Introductory Physics Laboratory*	
3. **Principal Investigator:** *Shawn Weatherford*	
4. **Department:** *Physics*	
5. **Campus Box Number:** *8202*	
6. **Email:** *saweathe@ncsu.edu*	
7. **Phone Number:** *513-7214*	
8. **Fax Number:** (919) 515-6538	
9. **Faculty Sponsor Name and Email Address if Student Submission:** *Ruth Chabay*	
10. **Source of Funding? (required information):** *NSF Award DUE-0618504*	
11. **Is this research receiving federal funding?:** *Yes*	
12. **If Externally funded, include sponsor name and university account number:** *Ruth Chabay Account Number 526878-1*	
13. **RANK:** ☐ Faculty ☐ Student: ☐Undergraduate; ☐ Masters; or **X** PhD ☐ Other (specify): _____	

As the principal investigator, my signature testifies that I have read and understood the University Policy and Procedures for the Use of Human Subjects in Research. I assure the Committee that all procedures performed under this project will be conducted exactly as outlined in the Proposal Narrative and that any modification to this protocol will be submitted to the Committee in the form of an amendment for its approval prior to implementation.

Principal Investigator:

Shawn Weatherford ______________________ *April 8, 2009*
*
(typed/printed name) (signature) (date)

*As the faculty sponsor, my signature testifies that I have reviewed this application thoroughly and will oversee the research in its entirety. I hereby acknowledge my role as the **principal investigator of record**.*

Faculty Sponsor:

Ruth Chabay ______________________ *April 8, 2009*
*
(typed/printed name) (signature) (date)

***Electronic submissions to the IRB are considered signed via an electronic signature. For student submissions this means that the faculty sponsor has reviewed the proposal prior to it being submitted and is copied on the submission.**

Please complete this application and email as an attachment to: joe_rabiega@ncsu.edu or send by mail to: Institutional Review Board, Box 7514, NCSU Campus (Administrative Services III).

Please include consent forms and other study documents with your application and submit as one document.
****

For SPARCS office use only

Reviewer Decision (Expedited or Exempt Review)

☐ Exempt ☐ Approved ☐ Approved pending modifications ☐ Table

Expedited Review Category: ☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8a ☐ 8b ☐ 8c ☐ 9

Reviewer Name Signature Date

North Carolina State University
Institutional Review Board for the Use of Human Subjects in Research
GUIDELINES FOR A PROPOSAL NARRATIVE

In your narrative, address each of the topics outlined below. Every application for IRB review must contain a proposal narrative, and failure to follow these directions will result in delays in reviewing/processing the protocol.

A. INTRODUCTION

1. Briefly describe in lay language the purpose of the proposed research and why it is important.

This research project will video record physics students completing modified laboratory activities in two experimental lab sections for the Fall semester. The goal of the project is to study, in detail, how students approach and complete lab activities. The kinds of lab activities we will ask students to complete are the same as in the non-experimental lab sections. These activities include working physics problems on whiteboards, completing hands-on experiments, and programming. As in the non-experimental lab sections, students will work in collaborative groups, have access to a TA for support, and receive credit for the laboratory requirement for PY205M.

Participation in the experimental lab section is voluntary. Students who decide to register or opt out of the experimental lab section will receive help in switching to a non-experimental lab section.

These modified laboratory activities are designed to improve the student learning of physics. All modifications will be cleared with the course coordinator ahead of the lab.

The purpose of this study is to fine-tune the lab activities for the PY205M course, study discourse while students work in groups to complete these activities, investigate how to design computational activities to improve conceptual understanding of physics, and capture TA-student interactions to use for the departmental TA training workshop.

2. If student research, indicate whether for a course, thesis, dissertation, or independent research.

Data on students working on computational tasks will be used in the PI's dissertation.

B. SUBJECT POPULATION

1. How many subjects will be involved in the research?

24 students involved (2 experimental lab sections with 12 students per experimental lab

section).

2. Describe how subjects will be recruited. Please provide the IRB with any recruitment materials that will be used.

Students are not initially recruited. Students register for Fall 2009 labs according to the registrar's registration calendar. Students who register for the PY205M lecture must register for one laboratory section. Sixteen of the 18 laboratory sections offered are for the non-experimental lab where no data is collected.

Students who choose to register for one of the two experimental lab sections where data recording will occur will be notified via their NCSU email address of the purpose of the lab section and mailed a consent form detailing risks to the student (notification and consent form attached). The student will have the opportunity to decide during the Spring registration period to stay in the course or choose one of the 16 non-experimental lab sections.

If the enrollment of each lab section is below 12 students during the first week of classes in August, the PI will actively recruit students by making a brief two-minute presentation at the beginning of the lecture course, if the lecture instructor agrees to yield time for this presentation. The handout for this recruitment (if necessary) is attached.

3. List specific eligibility requirements for subjects (or describe screening procedures), including those criteria that would exclude otherwise acceptable subjects.

Any student taking the PY205M lecture may register for the lab as long as there are seats open. These student willing to participate in the study having read the consent form, must also elect to allow researchers to use all video data for the purposes of 1) analysis by members of the NCSU PERD group, 2) evidence in presentations at professional conferences where their likeness may be shown, and 3) example case studies in departmental TA training workshops. If the subject is a minor, the legal parent/guardian must additionally agree to provide permission for the same purpose.

These short segments of video will be presented as evidence to the claims or counterclaims made by the researcher, only if the video segment captures additional resolution that the audible transcript does not.

4. Explain any sampling procedure that might exclude specific populations.

None.

5. Disclose any relationship between researcher and subjects - such as, teacher/student; employer/employee.

The PI will serve as the assigned TA for the two experimental lab sections. TA responsibilities include evaluating student assignments completed during the lab with the same grading rubric used in the regular labs. The TA will only evaluate students in these experimental lab sections, thus eliminating any possible bias or judgments about non-experimental lab students. The TA is the only person allowed to evaluate student performance in 205M lab sections. Lecture instructors have no say in the laboratory component of the course grade.

The faculty sponsor is a lecture instructor for PY205M and assigns a final score encompassing both lecture and lab as detailed in the course syllabus. All assessments that are part of the lecture section are objective and include homework scores calculated by WebAssign, participation in lecture with clickers using Student Response Software, and the grading of exams with rubrics developed by the PY205M course coordinator, who is not directly involved with this project. Some of the 24 students involved in the study may also be students in the faculty sponsor's lecture section.

Students from either the experimental or non-experimental lab sections with issues about grades may appeal to the Course Coordinator.

Additional measures to protect student identity are discussed below.

6. Check any vulnerable populations included in study:

   X minors (under age 18) - if so, have you included a line on the consent form for the parent/guardian signature
   - ☐ fetuses
   - ☐ pregnant women
   - ☐ persons with mental, psychiatric or emotional disabilities
   - ☐ persons with physical disabilities
   - ☐ economically or educationally disadvantaged
   - ☐ prisoners
   - ☐ elderly
   - **X** students from a class taught by principal investigator
   - ☐ other vulnerable population.

7. If any of the above are used, state the necessity for doing so. Please indicate the approximate age range of the minors to be involved.

Since 1st semester freshmen are part of the population eligible to participate in the study, the anticipated range of minors is 17-18 years old. Participants will be asked to volunteer their date of birth on the consent form to determine if a parent/guardian signature is required, in addition to the minor's assent to participate.

C. PROCEDURES TO BE FOLLOWED

1. In lay language, describe completely all procedures to be followed during the course of the experimentation. Provide sufficient detail so that the Committee is able to assess potential risks to human subjects. In order for the IRB to completely understand the experience of the subjects in your project, please provide a detailed outline of everything subjects will experience as a result of participating in your project. Please be specific and include information on all aspects of the research, through subject recruitment and ending when the subject's role in the project is complete. All descriptions should include the informed consent process, interactions between the subjects and the researcher, and any tasks, tests, etc. that involve subjects. If the project involves more than one group of subjects (e.g. teachers and students, employees and supervisors), please make sure to provide descriptions for each subject group.

Potential subjects will register for one of the lab sections during the registration period beginning Spring 2009. Subjects who choose one of the experimental labs will be notified of the purpose of the lab section as well as its risks and benefits as indicated on a consent form sent to the student's NCSU email address.

If the two sections of experimental lab have not filled by the first week of classes during the Fall 2009 semester, the PI will ask permission from the PY205M lecture instructors to make a short two-minute presentation about the experimental lab in an attempt to explain its purpose to a larger audience. Students interested in more information will be directed to email the PI who will at that time send the interested student a consent form to read over. At that time, it is up to the student to make the decision whether or not to switch through normal registration procedures.

On the first day of the lab, before any data capture begins, the PI will again go over the consent form and make sure all the subjects are clear about the experience and have signed the form (as well as any parent/guardian signatures for participants who are under 18). The TA will describe the types of data we are collecting and recording, including the use of video/audio devices to capture discussions, work written on whiteboards, computer models displayed on screens, all gestures, interactions with the TA, activities performed as a group, and off-task behaviors. Additionally, students are reminded that they may transfer to a non-experimental lab section during the semester.

Subjects are instructed about working in collaborative groups including the role of manager, recorder, skeptic, and summarizer. The subjects are also informed that in order to help maintain their anonymity, the PI will refer to individuals in the group by their role and not their given name. All student names uttered aloud will be deleted from the audio record.

Students complete the same activities as regular lab sections. Modifications to the activities will be variations of the standard tasks, intended to improve students' learning and understanding. All modifications will be submitted to the PY205M course coordinator for approval before they are implemented in the experimental sections. This alternate method may include altering the tasks to complete or questions to answer pertaining to an activity.

For example, programming activities used in regular lab sections involve students generating a program by supplying missing lines of code. We would like to shift the student focus from developing a working program to thinking about how the program works. A proposed modification involves providing groups with a complete example program where students study the program code and use a whiteboard to draw a prediction of what happens when the program runs.

In addition, students may be asked to submit an alternate type of report. Students in non-experimental lab sections use WebAssign to answer essay questions pertaining to lab activities. The experimental lab section may ask students to record a short 90-second audio/visual clip explanation instead.

All modifications are designed to improve student learning.

During a lab, students are given adequate time to complete all tasks as well as answer related questions delivered by WebAssign or the TA. Students will receive a score in lab based on the correctness of these answers. Subjects who miss a lab will receive zero points for the lab session, however subjects may be given permission to attend a makeup lab in a non-experimental section if the absence meets criterion defined by the lecture instructors for all PY205M students. Experimental-lab scores are calculated the same way as non-experimental lab scores. This lab score is then combined with other objective work related to the lecture section (tests, homework, participation) to calculate a final course grade.

At the conclusion of each week's lab, the TA will collect all data, monitor the audio for any use of a subject's given name and then deleting or beeping out that audio identification, and transfer it over to a secure server. Some analysis by the TA and other PERD research group members may occur before the end of the semester.

After all final grades are uploaded to the registrar's office, the PI will destroy the record used to track student attendance which includes subjects' identities and relates those identities to group membership during any single lab week.

2. How much time will be required of each subject?

The subject would spend the same amount of time in lab as a student enrolled in the non-experimental lab.

D. POTENTIAL RISKS

1. State the potential risks (physical, psychological, financial, social, legal or other) connected with the proposed procedures and explain the steps taken to minimize these risks.

Although careful consideration was given to protecting student anonymity, there is the unforeseen possibility of a researcher who is permitted to view the video data to recognize an individual participating in the study.

No persons who are currently, or in the future may be in a position of influence or power over the subject will learn the identity of the subjects involved in the study as all data revealing the names of students will be destroyed.

Other risks to subject anonymity that may be involved in processing data (and their countermeasures) are discussed in section 3.

2. Will there be a request for information that subjects might consider to be personal or sensitive (e.g. private behavior, economic status, sexual issues, religious beliefs, or other matters that if made public might impair their self-esteem or reputation or could reasonably place the subjects at risk of criminal or civil liability)?

No.

a. If yes, please describe and explain the steps taken to minimize these risks.

N/A.

b. Could any of the study procedures produce stress or anxiety, or be considered offensive, threatening, or degrading? If yes, please describe why they are important and what arrangements have been made for handling an emotional reaction from the subject.

The procedures in the study are no different from the non-experimental lab sections. Students will complete tasks in groups directly related to the goals of the course and correlate to the topics covered in lecture. Students may feel anxious or stressed while attempting a difficult physics task, however this is no different from peers participating in the non-experimental lab sections. The TA's primary role is to help students successfully complete all assigned tasks, therefore any anxiety or stress will be short lived. Students may initially be self-conscious about being video recorded but this goes away as students get used to the lab room.

3. How will data be recorded and stored?

The following data will be collected(method of storage).

1. WebAssign is used to deliver all lab assignments and handle records of student submissions and scores. The course coordinator, who is not associated with the study, and the graduate student TA will have access to the this data which contains a list of the subject's names which is stored on WebAssign's server. Only the TA will have the information necessary to connect the subject's name to their likeness captured by the video cameras using a log that correlates the assigned cooperative group number to the group's location in the room, to the subject's identifying name in WebAssign. This "Rosetta stone" is a paper document and its function is only to match the credit each group receives for tasks completed to a WebAssign score for an individual. This document serves an administrative purpose only, to record lab attendance, and will be destroyed at the end of the Fall 2009 semester. None of the information in this document is used for research purposes while analyzing or reporting data.
2. Assignment Submission Data. Students offer submissions to targeted questions in WebAssign as evidence of completing lab activities. The subject's responses are valid data for this study. This data is stored electronically on WebAssign's server and is only accessible by the TA and course coordinator.

3. Overhead Video Camera and Audio. This data records any work written on the whiteboard, individuals pointing at a computer monitor, and gestures (stored electronically on a PERD secure server).
4. Computer Screen Recording and Webcam. The computer screen recording software will capture all activity on the computer screen as well as any audio signal intercepted by an external microphone. The webcam attaches to the computer monitor and is directed towards the student group. Its camera will record gesture communication between the students in the group and the interactions between the group and the TA. The webcam will also capture audio using the same external microphone as above (stored electronically on a PERD secure server).

a. How will identifiers be used in study notes and other materials?

The four tables in the lab are numbered and groups each week are rotated randomly from one table to another. Any identifiers created during data analysis and used in reporting will be randomly assigned to distinguish one group from another during a lab. Note: These identifiers have no association with the "Rosetta stone" mentioned above and serve a research purpose rather than an administrative purpose, therefore there are no risks to compromising subject anonymity.

b. How will reports will be written, in aggregate terms, or will individual responses be described?

This study's unit of analysis is each group, consisting from two to four subjects. When reporting an episode of interest that occurs within the group, individuals are distinguished from one another based on the role they assume for the activity (Manager, Recorder, Skeptic, Summarizer, TA).

4. If audio or videotaping is done how will the tapes be stored and how/when will the tapes be destroyed at the conclusion of the study.

All audio/visual data exists in electronic form, stored on a secure PERD group server with access restricted to members of the PERD group. As the amount of data accumulates on the server, data may be transferred to external hard drives secured in the PERD observation room by lock and key. Additionally, the drive itself will be password encrypted to secure the data against loss and theft. PERD group members are allowed to use the data and its analysis in future research projects. All data may be kept indefinitely.

5. Is there any deception of the human subjects involved in this study? If yes, please describe why it is necessary and describe the debriefing procedures that have been arranged.

No deception, whatsoever.

E. POTENTIAL BENEFITS

This does not include any form of compensation for participation.

1. What, if any, direct benefit is to be gained by the subject? If no direct benefit is expected, but indirect benefit may be expected (knowledge may be gained that could help others), please explain.

Students participate in a laboratory with a lower teacher-student ratio as compared to non-experimental labs. The lower student-teacher ratio is dictated by the physical size of the experimental lab rooms. A lower student-teacher ratio is important because newly modified instructional materials may be confusing to students in unanticipated ways, and the TA must be able to detect and remedy such confusions quickly. Students will spend only a small fraction of their time interacting with the TA, since they work in formal cooperative groups, and the student groups bear the primary responsibility of planning and completing lab work.

In PY205m, students' lab grades contribute 13% of their overall course grade. Students in the experimental sections will not have significantly higher lab grades, because in the regular sections almost all students who attend every lab earn lab scores of 95-100%. Any score improvement would be nearly insignificant: a 2% increase in lab score would yield only a 0.3% increase in the student's overall course grade.

The end product of all laboratory exercises will be the same for students in both the standard and experimental lab sections (experimental data and analysis, problem solution, or computational simulation with visual output). Grades will be based on these end products. Modified instructional materials may vary the starting point, questions to be answered during the activity, or explanations requested while students are working. The experimental activities will not be easier, and are not intended to be more difficult.

To help illustrate the types of modifications we are seeking to test, the application packet contains supporting material that compares an activity that has been modified to its original form as well as the rubric used to grade the programs. The rubric is the same for both experimental and non-experimental sections. The specific goal of the modification is to see how alternate tasks affect students' ability to relate the visualization of a computer program to the algorithm the computer uses to create it. The rubric for the experimental and non-experimental lab sections do not evaluate the ability of students to accomplish this goal. Therefore, there is minimal risk that the modifications to the instructions will have an effect on course grades.

The experimental labs are offered during the early afternoon at popular times. This is determined by how quickly lab courses fill up at this time of the day during previous semesters. The TA has ample experience teaching introductory physics courses and labs and English is his native language.

F. COMPENSATION

Please keep in mind that the logistics of providing compensation to your subjects (e.g., if your business office requires names of subjects who received compensation) may compromise anonymity or complicate confidentiality protections. If, while arranging for subject compensation, you must make changes to the anonymity or confidentiality provisions for your research, you must contact the IRB office prior to implementing those changes.

1. Explain compensation provisions if the subject withdraws prior to completion of the study.

None.

2. If class credit will be given, list the amount and alternative ways to earn the same amount of credit.

Students may alternatively transfer to a non-experimental lab at any point during the semester. There are **many** non-experimental lab sections.

Lab scores for both experimental and non-experimental sections contribute towards 13% of the overall PY205M course grade.

G COLLABORATORS

1. If you anticipate that additional investigators (other than those named on **Cover Page**) may be involved in this research, list them here indicating their institution, department and phone number.

Brandon Lunk, a NCSU physics graduate student, will control the ceiling-mounted cameras from an observation room. Brandon Lunk will not have access to any of the subject's names, however he will see the likeness of subjects through monitors in our observation room. As a member of the PERD research group, Brandon will also have access to the data for research purposes.

2. Will anyone besides the PI or the research team have access to the data (including completed surveys) from the moment they are collected until they are destroyed.

Yes, all NCSU PERD group members present and future will have access to the audio/video data. To participate, subjects must elect to provide permission to the NCSU PERD research group to present short video clips, which may or may not contain the subject's likeness to other researchers in the field at professional conferences. If the subject is a minor, the legal parent/guardian must additionally agree to provide permission for the same purpose. These short segments of video will be presented as evidence to the claims or counterclaims made by the researcher, only if the video segment captures additional resolution that the audible transcript does not.

Students who do not elect to provide permission to the NCSU PERD research group to record their likeness or present video data at research conferences will be asked to switch to a non-experimental lab section.

H. CONFLICT OF INTEREST

1. Do you have a significant financial interest or other conflict of interest in the sponsor of this project? _No_

2. Does your current conflicts of interest management plan include this relationship and is it being properly followed? _N/A_

I. ADDITIONAL INFORMATION

1. If a questionnaire, survey or interview instrument is to be used, attach a copy to this proposal.

2. Attach a copy of the informed consent form to this proposal.

3. Please provide any additional materials that may aid the IRB in making its decision.

J. HUMAN SUBJECT ETHICS TRAINING

***Please consider taking the Collaborative Institutional Training Initiative (CITI), a free, comprehensive ethics training program for researchers conducting research with human subjects. Just click on the underlined link.**

Revised 03/2009

North Carolina State University
Institutional Review Board For The Use of Human Subjects in Research

GUIDELINES FOR PREPARATION OF INFORMED CONSENT FORM

PLEASE READ ALL OF THIS INFORMATION CAREFULLY PRIOR TO COMPLETING THE CONSENT FORM

An **Informed Consent Statement** has two purposes: (1) to provide adequate information to potential research subjects to make an informed choice as to their participation in a study, and (2) to document their decision to participate. In order to make an informed choice, potential subjects must understand the study, how they are involved in the study, what sort of risks it poses to them and who they can contact if a problem arises (see informed consent checklist for a full listing of required elements of consent). Please note that **the language used to describe these factors must be understandable to all potential subjects, which typically means an eighth grade reading level.** The informed consent form is to be read and signed by each subject who participates in the study **before** they begin participation in the study. A duplicate copy is to be provided to each subject.

If subjects are **minors (i.e. any subject under the age of 18)** use the following guidelines for obtaining consent:

0-5 years old – requires signature of parent(s)/guardian/legal representative
6 – 10 years old - requires signature of parent(s)/guardian/legal representative and verbal assent from the minor. In this case a minor assent script should be prepared and submitted along with a parental consent form.
11 - 17 years old - requires signature of both minor and parent/guardian/legal representative

If the subject or legal representative is *unable to read and/or understand the written consent form*, it must be verbally presented in an understandable manner and witnessed (with signature of witness). If there is a good chance that your intended subjects will not be able to read and/or understand a written consent form, please contact the IRB office (919-515-7515 or 919-515-4514) for further instructions.

***For your convenience, attached find a sample consent form template that contains necessary information. In generating a form for a specific project, the principal investigator should complete the underlined areas of the form and replicate all of the text that is not underlined, except for the compensation section where you should select the appropriate text to be used out of several different scenarios.**
***This consent form template can also be adapted and used as an information sheet for subjects when signed informed consent is waived by the IRB. An information sheet is usually required even when signed informed consent is waived. The information sheet should typically include all of the elements included below minus the subject signature line; however it may be modified in consultation with the IRB.**

North Carolina State University
INFORMED CONSENT FORM for RESEARCH

Student Learning in a Modified Introductory Physics Laboratory

Principle Investigator: Shawn Weatherford   Faculty Sponsor: Ruth Chabay

What are some general things you should know about research studies?

You are being asked to take part in a research study. Your participation in this study is voluntary. You have the right to be a part of this study, to choose not to participate or to stop participating at any time without penalty. The purpose of research studies is to gain a better understanding of a certain topic or issue. You are not guaranteed any personal benefits from being in a study. Research studies also may pose risks to those that participate. In this consent form you will find specific details about the research in which you are being asked to participate. If you do not understand something in this form it is your right to ask the researcher for clarification or more information. A copy of this consent form will be provided to you. If at any time you have questions about your participation, do not hesitate to contact the researcher(s) named above.

What is the purpose of this study?

This research project will record physics students completing some modified laboratory activities while working in cooperative groups once a week per lab section. The research data consists of submissions to lab assignments, video and audio from the taped lab sessions, computer screen capture and webcam images. This data may be kept indefinitely and may be used for other research projects. These activities are designed to improve the learning of physics. The purpose of this study is to fine-tune the lab activities for the PY205M course, study discourse while students work in groups to complete these activities, and capture TA-student interactions to use for the departmental TA training workshop.

What will happen if you take part in the study?

If you agree to participate in this study, you will be asked to come to lab each week and agree to be video recorded while completing lab activities with your lab group. These activities may be slightly modified from other lab sections. Also, some of your coursework will be used as data. You will receive credit for the lab component of PY205M.

Risks

You may temporarily feel frustrated if you are unable to complete a laboratory task, however an experienced TA will offer hints and suggestions to help you complete all tasks successfully. You may feel self-conscious about being video recorded, however this feeling is temporary and will pass. Although careful consideration was given to protecting student anonymity, there is the unforeseen possibility that a researcher who is permitted to view the video data may recognize you and form an opinion about you.

Benefits

Students participate in a laboratory with a lower student-teacher ratio(12:1) as compared to non-research labs(24:1). The two research labs are offered during the early afternoon at popular times. The TA has ample experience teaching introductory physics courses and labs. English is his native language. Participation in this lab will lead to knowledge gained that will help instructors design effective activities for future lab students. We do not anticipate any student’s course grade to improve or decline by participating in the experimental section.

Confidentiality

Your identity in the study records will be kept strictly confidential. All audio/visual data exists in electronic form, stored on a secure Physics Education Research and Development (PERD) group server with access restricted to members of the PERD group. As the amount of data accumulates on the server, data may be transferred to external hard drives secured in the PERD observation room by lock and key. Additionally, the drive will be password protected and encrypted to secure the data against loss and theft. No reference will be made in oral or written reports, which could link you to the study. The researcher will delete any identifying information from all study materials taken from WebAssign. All verbal utterances of your name will be deleted from the audio record.

Compensation
You will not receive any monetary reward for participating. Attending this lab section will fulfill the course requirement for PY205M lab. Grading in the experimental lab is the same as grading in non-experimental labs.

What if you are a NCSU student?
Participation in this study is not a course requirement and your participation or lack thereof, will not affect your class standing or grades at NC State. To participate on the first day of lab, you must sign this consent form acknowledging the proposed benefits and risks of this study. You may elect to attend one of the non-research lab sections available if you decide at any time during the semester to terminate your participation in this study, at no penalty to you.

What if you have questions about this study?
If you have questions at any time about the study or the procedures, you may contact the faculty sponsor, Ruth Chabay, in 248 Riddick Hall, via phone at 513-4826, or via email at rwchabay@ncsu.edu. Or, you may contact the research lab TA, Shawn Weatherford in 224A Riddick Hall, via phone at 513-7214, or via email at saweathe@ncsu.edu

What if you have questions about your rights as a research participant?
If you feel you have not been treated according to the descriptions in this form, or your rights as a participant in research have been violated during the course of this project, you may contact Deb Paxton, Regulatory Compliance Administrator, Box 7514, NCSU Campus (919/515-4514).

Consent To Participate
"I have read and understand all the above information. I have received a copy of this form. I agree to participate in this study with the understanding that I may choose not to participate or to stop participating at any time without penalty or loss of benefits to which I am otherwise entitled. I agree to have my likeness captured by video equipment. I agree to allow the researchers to use videos that contain my likeness in presentations at research conferences with the purpose of enhancing those presentations. I agree and allow the researchers to use videos that contain my likeness in materials designed to train future TAs in departmental TA workshops, or for other research purposes. I understand that the researchers will never use my real name, even though I allow the use of the video recordings in public presentations."

Participants under 18 years old must also obtain permission from a parent or guardian to participate in this study.

____ ______________ **Date** ___________
Subject's signature **Date of Birth**

____ **Date**________________
Parent/Legal Guardian (if Subject is under 18)

____ **Date** _______________
Investigator's signature

North Carolina State University is a land-grant university and a constituent institution of The University of North Carolina

Office of Research and Graduate Studies

NC STATE UNIVERSITY

Sponsored Programs and Regulatory Compliance
Campus Box 7514
2701 Sullivan Drive
Raleigh, NC 27695-7514

919.515.2444
919.515.7721 (fax)

From: Debra Paxton, IRB Administrator
North Carolina State University
Institutional Review Board

Date: April 29, 2009

Project Title: Student Learning in a Modified Introductory Physics Laboratory

IRB#: 931-09-4

Dear Mr. Weatherford:

The research proposal named above has received administrative review and has been approved as exempt from the policy as outlined in the Code of Federal Regulations (Exemption: 46.101.b.1). Provided that the only participation of the subjects is as described in the proposal narrative, this project is exempt from further review.

NOTE:

1. This committee complies with requirements found in Title 45 part 46 of The Code of Federal Regulations. For NCSU projects, the Assurance Number is: FWA00003429.
2. Any changes to the research must be submitted and approved by the IRB prior to implementation.
3. If any unanticipated problems occur, they must be reported to the IRB office within 5 business days.

Sincerely,

Deb Paxton
NCSU IRB

A Spacecraft Voyage, Part 1: Spacecraft and Earth

OBJECTIVES
In this program you will model the motion of a spacecraft. You will use your working program to explore the effect of the spacecraft's initial velocity on its trajectory.

TIME
You should finish this activity in 45 minutes or less.

GROUP ROLES
Before you begin, you should agree on the responsibilities of the Manager, the Recorder, and the Skeptic for this part of the lab.
Record your answers to the QUESTIONS sections; you will need them later.

1 Explain and Predict Program Visualization

A minimal working program has been provided in WebAssign. Right click on the linked file and save it. Remember to give it the extension ".py".

DO NOT RUN THE PROGRAM YET. Read through the program together. Make sure everyone in the group understands the function of *each* program statement. Reading and explaining program code is an important part of learning to create and modify computational models.

After reading *every* line of the program, answer the following questions:

- What is the physical system being modeled? In the real world, how should this system behave? On the left side of the whiteboard, draw a sketch showing how you think the objects should move in the real world.
- Will the program *as it is now written* accurately model the real system? **DO NOT RUN THE PROGRAM YET.** Study the program again. On the right side of the whiteboard, draw a sketch of how the objects created in the program will move on the screen, based on your interpretation of the code.

Predictions	
Real World	Visual Output

- Run your program *after* everyone agrees on both predictions. Discuss how closely your prediction of what the program would do matches what you see in the visual output.

2 Run the program

- How did your prediction compare to what you saw? Did something happen that you didn't expect to happen?

3 Modify the program

- **What calculations are needed to express the interactions between the spacecraft and the Earth?** Recall the discussion of iterative prediction of motion in the textbook (Chapter 2). All repeated calculations go inside the loop in the program.

  Calculate the (vector) forces acting on the system and their sum, $\vec{F}_{\text{net}}$.
  Update the momentum of the system: $\vec{p}_f = \vec{p}_i + \vec{F}_{\text{net}}\Delta t$.
  Update the position: $\vec{r}_f = \vec{r}_i + \vec{v}\Delta t$.
  Repeat.

- **Modify the program with these calculations. Where should you place these new lines of code in the program?** The previous two programs you have written may be helpful resources.

- **Use symbolic names defined in the program. You may need to add other symbolic names in your calculations.**

- **Finally, run the program. Is the behavior what you expected? If not, check that your code correctly expresses the relevant physics, and that you have placed repeated calculations inside the `while loop`.**

4 Visualize the momentum vector with an arrow

In a previous program, you used arrows to visualize gravitational force vectors. In this program, you will use an arrow to visualize the momentum of the spacecraft.

You will create an `arrow` before the `while` loop and update its position and axis inside the loop, as the spacecraft's momentum changes. You need to know the approximate magnitude of the momentum in order to be able to scale the arrow to fit into the scene, so just before the loop, print the momentum vector:

```
print 'p=', pcraft
```

In the #CONSTANTS section of your program, add this scale factor

```
pscale = ??
```

Also insert the following statement **in the #OBJECTS section of your program, NOT inside the loop**, because that would create a very large number of arrows:

```
parr = arrow(color=color.green)
```

This statement creates an arrow object with default pos and axis attributes.

- Comment out any print statements that are inside your loop, because they slow the program down.
- Inside your loop, update the pos attribute of the parr arrow object to be at the center of the spacecraft, and on a separate line update its axis attribute to represent the current vector value of the momentum of the spacecraft (multiplied by pscale).

```
parr.pos = ?
parr.axis = ?
```

- You may have to adjust the scale factor once you have seen the full orbit.

5 Answer questions about changes in the spacecraft's momentum

- **For this elliptical orbit, what is the direction of the spacecraft's momentum vector? Tangential? Radial? Something else?**
- **What happens to the momentum as the spacecraft moves away from the Earth?**
- **As it moves toward the Earth?**
- **Why? Explain these changes in momentum in terms of the Momentum Principle.**
- **Compare your answers to those of another group.**

6 Answer questions about the effect of initial velocity on the motion

- **Approximately, what minimum initial speed is required so that the spacecraft "escapes" and never comes back? You may have to zoom out to see whether the spacecraft shows signs of coming back. You may also have to extend the time in the while statement.**
- **What initial speed is required to make a nearly circular orbit around the Earth? You may wish to zoom out to examine the orbit more closely.**
- **Why does changing the initial velocity have an effect on the orbit?**

7 Optional: Detecting a collision

If your spacecraft collides with the Earth, the program should stop. Add code similar to the following inside your loop (using the name you defined for the distance between the spacecraft and the center of the Earth):

```
if rmag < Earth.radius:
   break
```

This code tells VPython to get out of the loop if the spacecraft touches the Earth.

8 Turn in your program to WebAssign

Make sure that everyone in the group agrees that the program is correct, is producing an elliptical orbit, and is displaying at all times an arrow representing the momentum. Check with a neighboring group.

The Recorder should turn in the program to WebAssign, and make sure everyone has a copy of the program.

When you turn in a program to WebAssign, be sure to follow the instructions given there, which may sometimes ask you to change some of the parameters in your program. The Recorder should email copies to everyone before leaving the lab.