

ABSTRACT

LOHMEYER, EDWIN LLOYD. *Unstable Aesthetics: The Game Engine and Art Modifications* (Under the direction of Dr. Andrew Johnston).

This dissertation examines episodes in the history of video game modding between 1995 and 2010, situated around the introduction of the game engine as a software framework for developing three-dimensional gamespaces. These modifications made to existing software and hardware were an aesthetic practice used by programmers and artists to explore the relationship between abstraction, the materiality of game systems, and our phenomenal engagement with digital media. The contemporary artists that I highlight—JODI, Cory Arcangel, Orhan Kipcak, Julian Oliver, and Tom Betts—gravitated toward modding because it allowed them to unveil the technical processes of the engine underneath layers of the game's familiar interface, in turn, recalibrating conventional play into sensual experiences of difference, uncertainty, and the new. From an engagement with abstract forms, they employed modding techniques to articulate new modes of aesthetic participation through an affective encounter with altered game systems. Furthermore, they used abstraction, the very *strangeness* of the mod's formal elements, to reveal our habitual interactions with video games by destabilizing conventional gamespaces through sensory modalities of apperception and proprioception. In considering the imbrication of technics and aesthetics in game engines, this work aims to resituate modding practices within a dynamic and more inclusive understanding of video game histories, digital media, and the avant-garde.

© Copyright 2018 by Edwin Lohmeyer

All Rights Reserved

Unstable Aesthetics: The Game Engine and Art Modifications

by
Edwin Lloyd Lohmeyer

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

Communication, Rhetoric, and Digital Media

Raleigh, North Carolina

2018

APPROVED BY:

Andrew Johnston
Committee Chair

Stephen Wiley

Nicholas Taylor

Chris Ingraham

DEDICATION

I dedicate this dissertation to my dog Bandit for his unwavering loyalty and support during the writing process.

BIOGRAPHY

Eddie Lohmeyer is a Ph.D. student and media artist in the Communication, Rhetoric, and Digital Media program at North Carolina State University. His research explores aesthetic and technical developments within histories of digital media, with an emphasis on video games and their relationship to theories of the avant-garde. Additionally, his art considers embodied experience through processes of play and defamiliarization. Using deconstructive approaches such as glitch, physical modifications to hardware, and assemblage, his installations stage bizarre encounters with nostalgic media objects to unveil our normal attitudes and perceptions toward technologies.

ACKNOWLEDGEMENTS

I want to thank the members of my committee including Andrew Johnston, Steve Wiley, Nick Taylor, and Chris Ingraham for their support and encouragement throughout the dissertation process. I especially want to thank my chair Andrew Johnston for his exceptional guidance and mentoring during the researching and writing of this project.

TABLE OF CONTENTS

LIST OF FIGURES	vi
Introduction: An Anatomy of Art Modding	1
Technicities of Modding	6
Aimlessness and the Formation of Sensory Experience	10
Theoretical Approaches to Video Game Art and the Avant-Garde	13
Methodologies.....	16
Chapter One: A (Scroll) Down Memory Lane: Non-Play, Commodity Aesthetics, and the Vitality of 8-bit Engines in Cory Arcangel’s <i>Super Mario Bros. Mods</i>	31
The Technics of the NES and Arcangel’s ROM Hacking	35
Speculative Ontologies of an 8-bit World	46
The Vitality of Glitch.....	51
The Démodé Object, Pop Art, and the Repetition of Smiling Clouds	60
Proprioceptive Non-Play and Engines of Difference.....	70
Chapter Two: Slaying Machines: Embodied Mimesis in <i>ArsDoom</i>	80
Developments in Game Engine Technology	82
<i>Doom</i> and Epistemologies of Modding	88
The Archive of the Engine.....	99
<i>ArsDoom</i> and Mimetic Vectors of Play	104
The Strangeness of Institutional Critique	115
Chapter Three: “Perspective Engines” and the Strangeness of 3D Space in JODI’s <i>Untitled Game</i>	122
The Work of Anamorphosis.....	124
A Grammatology of the <i>Quake</i> Engine	133
The Formal Materiality of <i>Untitled Game</i>	148
Flicker Mods and the Modulation of Bodily Space	156
Chapter Four: Generative Mods and the Violence of Sensation	172
The <i>Quake III: Arena</i> Engine	176
Volatile Machines and Physical Systems	185
The <i>Meat</i> of the Glitch.....	193
Conclusion: Intractable Spaces	205
References	212

LIST OF FIGURES

Figure 1.1	Horizontal scrolling thresholds in the SMB engine from Nathan Altice, <i>I am error: the Nintendo family computer/entertainment system platform</i> (Cambridge, MA: The MIT Press, 2017), 153	39
Figure 1.2	Hacked <i>Super Mario Bros.</i> cartridge with exposed PRG-ROM chip (Cory Arcangel, 2002). Photo courtesy of the artist at http://www.coryarcangel.com	42
Figure 1.3	Continually-scrolling clouds in <i>Super Mario Clouds</i> (Cory Arcangel, 2002)	44
Figure 1.4	Strobing patterns of glitch in the deteriorating gameworld of <i>Super Mario Movie</i> (Cory Arcangel and Paper Rad, 2005)	55
Figure 1.5	Puppet and dolls animated using stop-motion techniques in <i>Street of Crocodiles</i> (Quay Brothers, 1987).....	57
Figure 1.6	A sequence of glitched tile patterns resulting in Mario’s flight through the gameworld of <i>Super Mario Movie</i> (Cory Arcangel and Paper Rad, 2005).....	59
Figure 1.7	Cutout demode commodities animated through stop-motion techniques from <i>Pony Glass</i> (Lewis Klahr, 1998)	61
Figure 1.8	The traumatic image of Marilyn Monroe in <i>Marilyn Diptych</i> (Andy Warhol, 1962)	64
Figure 1.9	The scrolling repetition of game icons in <i>Super Mario Movie</i> (Cory Arcangel and Paper Rad, 2005)	67
Figure 1.10	Mario transforming into the energy of the system in <i>Super Mario Movie</i> (Cory Arcangel and Paper Rad, 2005).....	68
Figure 1.11	<i>Super Mario Clouds</i> exhibited at the Whitney Museum of American Art in 2002. Photo courtesy of the artist and Team Gallery, New York	69
Figure 1.12	<i>Super Mario Movie</i> exhibited at Deitch Projects in 2005. Photo courtesy of Deitch Projects	71
Figure 1.13	<i>Super Mario Clouds</i> exhibited at Arcangel’s <i>Masters</i> show at the Carnegie Museum of Art in 2012. Photo courtesy of Tom Little.....	74
Figure 2.1	Wireframe environments from <i>Akalabeth</i> (Richard Garriott, 1980)	83
Figure 2.2	The rendering of first-person perspective in <i>Catacomb 3-D</i> (id Software, 1991) ...	86

Figure 2.3	Fast-paced navigation through rendered hallways in <i>Wolfenstein 3D</i> (id Software, 1992).....	86
Figure 2.4	Run and gun tactics within the 3D gamespace of <i>Doom</i> (id Software, 1993)	90
Figure 2.5	Divided subsectors and their individual segs within the E1M1: Hangar map of <i>Doom</i> from “Doom rendering engine,” <i>Doom Wiki</i> , http://doom.wikia.com/wiki/Doom_rendering_engine	92
Figure 2.6	Subdivided planes via BSP ordering in the E1M1: Hangar map of <i>Doom</i> from Fabien Sanglard, “Doom Code Review,” Fabien Sanglard’s Website, January 13 th 2010, http://fabiansanglard.net/doomIphone/doomClassicRenderer.php	94
Figure 2.7	Thousands of monsters in the joke WAD <i>Nuts</i> (B.P.R.D., 2001).....	98
Figure 2.8	Gameplay from <i>Museum Meltdown II: The Vilnius Vengeance</i> (Palle Torsson and Tobias Bernstrup, 1997).	105
Figure 2.9	Architectural rendering of the Brucknerhaus exterior from <i>ArsDoom</i> (Orhan Kipcak and Reini Urban, 1995)	110
Figure 2.10	Firing at an enemy artist using a crucifix gun in <i>ArsDoom</i> (Orhan Kipcak and Reini Urban, 1995).	110
Figure 2.11	Shooting projectiles from a TV remote-launcher, a citation to the work of Nam June Paik from <i>ArsDoom</i> (Orhan Kipcak and Reini Urban, 1995).	113
Figure 2.12	Destroying an artwork by firing a crucifix gun and painting over its contents in the style of Arnulf Rainer from <i>ArsDoom</i> (Orhan Kipcak and Reini Urban, 1995)	115
Figure 2.13	Hans Haacke’s <i>Manet-PROJEKT’74</i> (1974) including ten panels of provenance documents under glass and color photo reproduction of Manet’s <i>Une botte d’asperges</i> (Bunch of Asparagus) at Paul Maenz Gallery, Cologne, 1974. Photo courtesy of Rolf Lillig.....	118
Figure 2.14	Splatters of blood-paint across an artwork in the simulated Brucknerhaus of <i>ArsDoom</i> (Orhan Kipcak and Reini Urban, 1995)	120
Figure 3.1	Swirling, psychedelic patterns in <i>Ctrl-Space</i> (JODI, 1996-2001).....	125
Figure 3.2	An anamorphic rendering of a skull in <i>The Ambassadors</i> (Hans Holbein the Younger, 1533).....	127
Figure 3.3	The distorted, anamorphic space in the installation <i>skulls</i> (Robert Lazzarini, 2000). Photo courtesy of the artist	129

Figure 3.4	JODI's <i>Untitled Game</i> mods projected onto large scrims in the 2012 <i>Street Digital</i> exhibition at the Museum of the Moving Image, New York. Photo courtesy of Daniel Love	132
Figure 3.5	Fighting monsters within the 3D environments of <i>Quake</i> (id Software, 1996)	134
Figure 3.6	BSP ordering in which polygons within the map are stored as leaves from Michael Abrash, "Inside Quake: Visible Surface Determination," <i>Ramblings in Realtime, Blue's News</i> , https://www.bluesnews.com/abrash/	138
Figure 3.7	Visible Surface Determination in which only visible sections of polygons are drawn within the frustum and all others are culled, from Michael Abrash, "Inside Quake: Visible Surface Determination," <i>Ramblings in Realtime, Blue's News</i> , https://www.bluesnews.com/abrash/	139
Figure 3.8	A PVS of visible polygons arranged within the player's frustum from <i>Quake</i> (id Software, 1996)	141
Figure 3.9	A pre-calculated lightmap for map E2M7 from <i>Quake</i> (id Software, 1996).	144
Figure 3.10	A surface rendered by illuminating texels in the pre-calculated lightmap from Michael Abrash, "Quake's Lighting Model: Surface Caching," <i>Ramblings in Realtime, Blue's News</i> , https://www.bluesnews.com/abrash/	145
Figure 3.11	Communication between the client and server in the <i>Quake</i> engine from Michael Abrash, "Quake's 3-D Engine: The Big Picture," <i>Ramblings in Realtime, Blue's News</i> , https://www.bluesnews.com/abrash/	147
Figure 3.12	JODI's use of aliasing effects in <i>Ctrl-9</i> (JODI, 1996-2001).	152
Figure 3.13	The rotating white line and flicker effect in <i>I-N</i> (JODI, 1996-2001).	154
Figure 3.14	The camera accelerating and summersaulting through walls in <i>Q-L</i> (JODI, 1996-2001)	155
Figure 3.15	Arranged rhythms of black and white frames in <i>The Flicker</i> (Tony Conrad, 1966)	159
Figure 3.16	Stroboscopic patterns generated from the rotating screen in <i>I-N</i> (JODI, 1996-2000)	162
Figure 3.17	Spinning and accelerating through the polygonal architecture of the map in <i>Q-L</i> (JODI, 1996-2001)	163
Figure 3.18	Abstract sensations of accelerating through space in <i>Koyaanisqatsi</i> (Godfrey Reggio, 1982)	165

Figure 3.19 Pulsating moiré effects caused when the player moves toward edges and corners of the map in <i>Crtl-9</i> (JODI, 1996-2001).....	167
Figure 3.20 Optical, illusory patterns in <i>Fall</i> (Bridget Riley, 1963)	168
Figure 3.21 The player’s wildly spinning viewpoint and an exploded piece of an enemy’s body (far right) in <i>Q-L</i> (JODI, 1996-2001).....	169
Figure 4.1 Fighting Bots in <i>Quake III Arena</i> (id Software, 1999).....	177
Figure 4.2 Top: <i>Q3A</i> level textures. Middle: Lightmap applied to arena. Bottom: multi-texturing combining both textures and lighting, from Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 2 of 5),” <i>Fabien Sanglard’s Website</i> , June 30 th 2012, http://fabiensanglard.net/quake3/	181
Figure 4.3 The <i>Q3A</i> engine architecture from Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 2 of 5),” <i>Fabien Sanglard’s Website</i> , June 30 th 2012, http://fabiensanglard.net/quake3/	185
Figure 4.4 <i>Méta-Matic No. 10</i> (Jean Tinguely, 1959).	186
Figure 4.5 <i>Homage to New York</i> (Jean Tinguely, 1960). Photo courtesy of The Museum of Modern Art Archives and David Gahr	189
Figure 4.6 <i>Condensation Cube</i> (Hans Haacke, 1965). A version of the cube at Museu d’Art Contemporani de Barcelona from 2006. Photo courtesy of the artist.....	191
Figure 4.7 <i>Grass Grows</i> (Hans Haacke, 1967-69). Photo courtesy of the artist	193
Figure 4.8 Superflat aesthetics in <i>In the Land of the Dead, Stepping on the Tail of a Rainbow</i> (2014) at Gagosian Gallery (Takashi Murakami, 2014). Photo courtesy of Ben Davis	194
Figure 4.9 <i>Figure with Meat</i> (Francis Bacon, 1954)	196
Figure 4.10 Abstractions of glitch in <i>QQQ</i> (Tom Betts, 2002).	198
Figure 4.11 A user interacting with <i>QQQ</i> at the 2012 Evolution Festival. Photo courtesy of the artist at http://www.nullpointer.co.uk/qqq/qqq5.htm	200
Figure 4.12 Frames from <i>ioq3apaint</i> (Julian Oliver, 2010).....	201
Figure 5.1 <i>San Andreas Streaming Deer Cam</i> (Brent Watanabe, 2015-2016).....	206

INTRODUCTION

An Anatomy of Art Modding

At the 1995 Ars Electronica Festival in Linz, Austrian artists Orhan Kipack and Reini Urban exhibited their work *ArsDoom*: an art modification (or mod) made to the game engine of *Doom*, the popular first-person shooter developed by id Software in 1993. Forgoing the use of *Doom*'s hell-like environments, the artists designed a custom map that painstakingly replicated the festival's Brucknerhaus exhibition hall using Autodesk AutoCAD and rendered the 3D space using the *Doom II* game engine. Within this reconstruction of the Brucknerhaus that employed video game technology contemporaneous with the festival, Kipack asked artists to submit work to be showcased within the gamespace. He then invited visitors, both remotely and at the festival, to log on to *Doom*'s online network to play as famous artists—Joseph Beuys, Jeff Koons, Nam June Paik among others—and destroy the artworks on display using custom weapons such as a paint brush gun and a bullet-firing cross.

Instead of advancing through hordes of demons or skirmishing in multiplayer deathmatches, participants within this *Doom* level-cum-museum took on the role of critic: they kept works intact that they liked, while desecrating the ones they didn't. This bizarre reconfiguration of a widely-known and commercially successful first-person shooter, marked not only an initial foray into the discipline of video game art, but introduced the public to a new mode of aesthetic participation with video games through art modding techniques. Players interacted with a popular medium in ways that were perceptually unfamiliar, prompting new sensory and affective engagements with video game technologies of the time.

This dissertation examines four unique episodes in the history of video game art modding situated around the introduction of the game engine as a software framework for the

development, rendering, and display of two and three-dimensional gamespaces. In particular, I look at art modding—the modification of an existing video game system’s software code or hardware components—as a unique, aesthetic practice that was used by gamers and artists between 1995 and 2010. Although the canon of video game art spans techniques beyond modding to include performance, in-game interventions, sculpture, and the appropriation of video game objects, I look at modifications made to game software and hardware in these four historical moments as ways in which artists explored the relationship between an aesthetic of abstraction, the technics of game systems, as well as our phenomenal engagement with the medium.

The contemporary artists that I highlight—Cory Arcangel, JODI, Julian Oliver, Orphan Kipack and Reini Urban, and Tom Betts—initially approached game modifications from other artistic fields—net art, experimental music, sculpture—and were experienced coders and programmers. In many ways, their attraction to modding arose out of previous artistic experimentation with digital media technologies situated within the hacker, DIY technoculture of the 1980s and 90s, and aesthetic strategies of appropriation more broadly within histories of the avant-garde. They were invested in concerns of both the video game’s materiality and the player’s sensory perceptual experience of a computational system as a cultural and aesthetic object.

Naturally, these artists gravitated toward game modding because, as I suggest, it allowed them to disclose the material operations of the game engine through which illusions of action and movement within space are produced. Modding allowed them to expose technical processes of the engine underneath layers of the game’s familiar interface, in turn, recalibrating conventional play into experiences of difference, of uncertainty, of the new. Most importantly, they were

drawn to modding practices because they provided a technical platform to generate novel sensory experiences with abstract forms, notably through aesthetics of world-building and glitch. Appropriating the mechanics of existing game systems created avenues for artists to engage the body in perceptual registers and intensities through the player's haptic input, in turn, generating an abstract, graphical output. They not only used modding to articulate new forms of aesthetic participation through bodily encounters with the altered game system, but used abstraction, the very *strangeness* of the mod's formal elements, to reveal our habitual perceptions of and interactions with video games.

This relationship among the technical conditions of art modding that allowed for artists to experiment with modes of abstraction, and the body's phenomenal engagement with its forms, has been largely overlooked in art history and game studies. My aim in this dissertation is to address this historical gap in which the introduction of game engine technologies beginning in the 1990s afforded artists the possibilities to modify, abstract, and *play with* the materiality of video game systems. In examining this period in the history of the avant-garde, my main outcome is not only to provide a theoretical understanding for art modding as an aesthetic practice, but to consider the player's sensual investment in abstract forms generated from game engines. To this end, I look at the unique *work of aesthetics* that unfolds among participant and the mod.

Here, I consider the ways in which novel sensory experiences are produced through our encounter with the mod that is fundamentally embodied and mimetic. I consider mimesis not as a product of verisimilitude as it is understood in the tradition of Western visual culture, but rather as an affective modality in which the movement of non-figurative forms are reciprocated intensively in the body. Thus, as a secondary outcome of this dissertation, my aim is to better

elucidate our phenomenal experience of the art mod and video games in general. I am interested in the medium's capacity to modulate the player's affective resonance of their own body and world.

Within the last two decades, scholars have begun to address the cultural significance of video games as an art form and particularly aesthetic strategies used in game modding. Yet, this scholarship largely focuses on the discursive construction of the video game medium as an artistic genre and its connections to previous art historical movements. Discussions of game art often assign modding aesthetics a representationalist label, with critics situating its formal abstraction in relationship to radical, subversive, or otherwise counterhegemonic interventions within the avant-garde. These considerations for the ways modding reworks or deconstructs modernist narratives are not without significance as this scholarship not only provides a means to properly contextualize game art and concepts of play within histories of the avant-garde, but also legitimizes video games as an artistic medium.

Yet, recent intellectual work in historicizing game art has largely overlooked what I see as two aspects critical to our understanding of art modding as an aesthetic practice: the complex materialities of game engine technologies that artists used to generate the illusory movement of abstract forms, and the vibrant possibilities for our embodied engagement with these forms. Scholars have approached recent histories of art modding only through a cursory examination of game technologies and the forms they produce, without reflecting upon how the technical affordances of modding and the interactivity inherent to the medium, both interrogate and *modify* our phenomenal immanence in the world. Thus, how do abstract forms generated from modding techniques engage the body in novel modes of affective and phenomenal experience? Why were artists and programmers particularly invested in using game engine technologies to question our

standard perceptions and operations of play with video games? In what ways did they alter the technics of game engines to produce new forms of aesthetic participation within the dominant framework of technoculture during the time?

In my aim to flesh out the vital relationship between technology, abstraction, and bodily experience in these historical episodes, I focus largely on the technical and often experimental methods artists used in the production of abstract gaming scenarios. These experimentations throughout arose during a unique period of video game history in which hegemonic forms of technicity—our cultural perceptions and attitudes toward the medium—developed through the introduction of the game engine and its ability to render 3D gameplay. As I will discuss, the game engine, notably id Software's *Doom* and *Quake* engines (also known as id tech 1 and 2) created by John Carmack and John Romero, organized a gaming subject that was phenomenally situated within a new perceptual regime of the first-person shooter genre, one in which the engine generated a seemingly palpable illusion: an aesthetic vector of movement through actionable space and highly-detailed 3D environments.

The pioneering game artists that I observe appropriated mainstream engine technologies and manipulated game mechanics using a range of techniques: altering commands and values within the game's source code; radically transforming the appearances of maps, characters, and weapons; breaking open game cartridges and rewriting the read-only memory (ROM); and writing generative algorithms that progressively abstract the gameworld through the use of its AI. I argue that they employed experimental uses of game engines from within a hegemonic field of popular gaming technologies as well as modding subcultures with a particular aesthetic outcome: to defamiliarize conventional attitudes toward play and produce new forms of subjectivity by making unstable the body's apperceptive and proprioceptive faculties. An

aesthetic of apperception is the body's affectivity as it relates to the act of perceiving the very strangeness of the mod's forms outside of standard gameplay, while proprioception is the body's interior sense of movement, balance, posture, and coordination within the gamespace. It is my argument that these two interrelated phenomenal modalities catalyze the player's bodily capacity to experience the video game medium and engage the possibilities for new forms of aesthetic participation. I consider the emergence of bodily sensation in relation to modding aesthetics as the ways in which video game technologies affirm a multiplicity of new modes of thought, sensation, and experience. In the remainder of this introduction, I will lay out in more detail the theoretical approaches I use to situate art modding within this historical period of video game development. Next, I include a review of literature on modding as an aesthetic practice inserted within theories and histories of the avant-garde. Lastly, I look at the methodologies I use and give a brief overview of each chapter included in the dissertation.

Technicities of Modding

The group of artists, hackers, and programmers that I engage throughout each chapter began experimenting with modding during a period that gave rise to new formations of both player and developer through technicities of video game development. By technicity, I refer to Jon Dovey and Helen Kennedy's use of the term as the imbrication of technologies, identity, and habitual uses of and attitudes toward game technologies. Technicities of cyberculture in the 1980s and 90s gave dominant meaning to computer game culture through the authority of game developers and consumer markets that circulated hegemonic truth about what constituted the social identity of a "gamer." The prevailing technicities of video games during the 1990s assumed that developer and player were technically proficient, white, heterosexual males and that gameplay was and still is predicated on mastery, productivity, pleasure, and competition

within capitalist regimes of the industry. Therefore, technicities are not inherent to game technologies themselves but are always the result of specific cultural processes brought into relationship with games, and our experience and knowledge of them are always in flux, contingent upon historical and social contexts and changing impositions of power. Dovey and Kennedy argue that when flows of power produce dominant technicities, they also invite subversive, counter identities and cultures to develop from within the hegemonic field. As an example, they look toward the ethos of the hacker and cyborg, two marginal forms of identity conceived during the 1980s that presented critical and often utopic means through which computer technologies might be re-appropriated.¹

These identities were embodied for instance, in “cracker” communities or rather, groups of coders who disabled the copy protection on games for early systems like the ZX Spectrum, Commodore 64, and Amiga to produce unique intro animation sequences that demonstrated the hacker’s technical prowess. In a similar way, John Carmack and John Romero, the creators and programmers of the popular first-person shooters *Doom* (1993) and *Quake* (1996) were often described as “social misfits” with cyborgian and automated work habits. They would stay up all night developing software with a machine-like efficiency.² Yet, by the early 1990s, these hacker/cyborg identities shifted from marginalized subcultures to central formations of technicity through the work of cultural capital that the video game industry produced. The game developer or expert player-coder who were once considered social outcasts, had by the early 1990s taken a central role in mainstream gaming culture.

¹ Jon Dovey and Helen W. Kennedy, *Game cultures: computer games as new media* (Maidenhead: Open Univ. Press, 2011), 63-66.

² See Dovey and Kennedy’s description of Romero and Carmack and “cyborgs” in Dovey and Kennedy, *Game cultures: computer games as new media*, 70-71.

It was also Romero and Carmack who anticipated the emergence of various modding subcultures through the open-source architecture of the engine. With the introduction of the *Doom* engine in 1993, players were given access to the game's source code, allowing them to record in-game speed runs—completing a certain map in the fastest time possible—and their skills in deathmatch competitions with other players. These early machinima demos, or short films generated in real-time using a game engine, were shared widely as .LMP files across the burgeoning Internet with networked communities of other skilled players in what was referred to as the demoscene. The open-source nature of the engine also afforded players the ability to design their own game maps and modify skins (the appearance of an avatar), weapons, and other tools for *Doom*. Romero and Carmack further solidified the demoscene and modding subculture with the release of *Quake* in 1996. With access to sections of the *Quake* source code, players produced a rich variety of user-generated mods within the complex, real-time 3D environments of the game. Like its predecessor *Doom*, these mods were distributed within a cultural economy of online *Quake* communities as downloadable .pak files for other players to use. Both the *Doom* and *Quake* engines facilitated the organization of new technicities we see in the game modder.

The *Quake* modder was not only an expert player, but also a proficient coder who was integral to the emerging demoscene. Groups of *Quake* players or clans, used their knowledge of the engine's source code to film and edit machinima footage of their gaming expertise—called .dem (demo) files—in multiplayer deathmatches as well as speed runs, such as with Anthony Bailey's famed *Quake Done Quick* series. Each .dem file recorded not only player input but the movement of all objects in the gameworld. In turn, this allowed *Quake* clans to experiment with producing narrative machinima recorded with a virtual camera, scripted dialogue, and choreography using avatars in real time. For instance, the famed Rangers Clan created their

animated film *Diary of a Camper* (1996), a short narrative about a player who is ambushed after strategically holding his position in an advantageous area of the map. Likewise, the ILL Clan created longer machinima films such as *Apartment Huntin'* (1998) which chronicled two blundering lumberjacks in search of a new place to rent. Other modifications included David "Zoid" Kirch's popular Capture the Flag game mode developed in late 1996 which included the Grappling Hook: a unique item allowing players to fast travel throughout a map by launching the hook and using its rope to swing and propel to various platforms. At roughly the same time, Robin Walker, John Cook and Ian Caughley created the influential Team Fortress which pitted specialized player classes (demolition, sniper, medic, heavy weapons expert, spy, etc.) against each other in team deathmatches or capture the flag.

The game modder and the range of innovations they produced using the *Doom* and *Quake* engines points to the fluid nature of technicity. Cultural attitudes developed toward modding and the early demoscene indicate the unique historical fabric from which artists experimented with manipulating game systems as an aesthetic technique. Here, I don't claim that game artists engaged strictly in counter-hegemonic practices to resist the commanding ideologies and capitalist regimes that undergirded the gaming industry. Rather, I believe they were drawn to repurposing the material affordances of engines within this dynamic era of game development to work through ways in which the medium could potentially reconfigure our phenomenological relationship to video game forms and bodily space within and of the world. As I argue throughout this project, artists experimented with engines to create new modes of bodily experience through the reconfiguration of play as an *aimless* encounter with the mod.

Aimlessness and the Formation of Sensory Experience

In using the term “aimlessness,” I refer not only to a disruption of a game’s flow through a type of sensual play lacking competitive reward or purpose, but also consider its double meaning as an intervention in the ability to aim a weapon within the simulated perspective of the first-person shooter genre which emerged from game engine technologies. Brian Schrank discusses this break in flow through what he calls “radical formal” strategies of the avant-garde in game art. He draws a relation between Renaissance painting, conventional gameplay, and our movement through a gamespace. Formal elements within Renaissance painting—anatomical precision, linear perspective, the reduction of scale as objects diminish toward a vanishing point—function to lead the eye throughout an illusory representation of depth within the picture plane. To Schrank, video games operate similarly in that they introduce a perceptual regime that guides our actions and optimizes the potential for reward, or the *flow* of the player’s experience.³

Dominant technicities within the gaming industry dictate that flow is an expectation for a game to appeal to consumer desires and result in a fun and pleasurable experience. Game developers mobilize flow by designing a game system to assess a player’s skills and adjust difficulty accordingly in response to their progress or establish given rules and mechanics to motivate them to keep playing. Therefore, a game is rewarding when flow falls between a player’s skill set and appropriate challenges that they can potentially overcome without getting too frustrated or conversely, too bored. Developers situate normative expectations of play in mainstream games around certain perceptual formulae through which flow becomes operational. In first-person shooters such as *Doom* and *Quake*, Cartesian perspective generated through the 3D space of the engine function as the coordinates through which the player’s experience is

³ Brian Schrank, *Avant-garde videogames: playing with technoculture* (Cambridge, MA: The MIT Press, 2014), 32.

optimized. Flow is realized in instances of movement within rendered 3D space—engaging in firefight, jumping, running, and solving puzzles to proceed to the next level or map—guided by the player’s point-of-view perspective. And in instances of optimal play, this 3D space technically and socially conditions the competitive player into what Emma Witkowski calls the “balanced body”: a body that adapts and responds to game scenarios through both its physical exteriority—pressing buttons to engage an in-game mechanic—and bodily composure when interacting with the affective forces of the game.⁴

Throughout these historical episodes, I argue that the type of sensory experiences revealed in the player’s strange interaction with the mod results from a disruption of flow. Artists intentionally practiced what Josephine Bosma refers to as “cargo cult coding” in her discussion of JODI’s game mods and works of net art. The “cargo cult” was a ritualistic belief system adopted by the people of the Melanesian islands in which they mimicked characteristics of Western colonists in hopes of bringing good fortune and wealth to their communities. In programming circles, the term indicates a type of “ritual” coding that does not have any functional outcome. In the case of JODI’s work, cargo cult coding is a way to appropriate specific computational effects without abiding by the standard logic and protocols of a game system.⁵ These formal interventions undertaken at the level of the engine’s code, worked to shift the player’s experience away from reward, pleasure, and mastery within a perspectively-coherent gamespace toward a type of aimless play: an aesthetic interaction with the system that forgoes optimal flow (and the ability to aim), re-channeling goal-driven play into sensory engagements with abstract forms.

⁴ Emma Witkowski, “On the Digital Playing Field: How we “do sport” with Networked Computer Games,” in *Games and Culture* 7, no.5 (2012): 359-360.

⁵ Josephine Bosma, “Jodi and the Cargo Cult,” in *INSTALL.EXE-JODI*, ed. Tilman Baumgaertel (Basel: Christoph Merian Verlag, 2002), 91.

The forms of bodily experience I articulate in these chapters that emerge from the aimlessness of play are aesthetically and politically constituted through what Jacques Ranciere calls the distribution of the sensible. It refers to the system of societal order, rules, and laws of sensory perception understood as self-evident that shape the conditions of possibility for what can be seen or said, or what one can do or make. It describes the circumstances for what can be sensed, divided into perceptual regimes that delineate who is included and excluded in forms of aesthetic experience within society. Ranciere argues that the distribution of the sensible produces the possibility for common experience, yet at the same time partitions experience through the circulation of temporalities, spaces, and movements, that arbitrate the ways in which groups or individuals participate in the aesthetic reception of a work. Within this political distribution, Ranciere relates aesthetics to a particular sensory regime for experiencing art, or rather the relationship among artistic production and the modes of visibility they generate as well as the possibilities for thinking through connections among making, doing, and seeing.⁶ Here, the articulation between politics and aesthetics does not describe a type of radical projection of politics by way of a given aesthetic practice, but instead it examines the participatory roles taken up within a work of art. I use Ranciere's relationship between politics and aesthetics to describe the type of sensory experience that is revealed through forms of aesthetic engagement within modding techniques of this period.

I want to suggest that the technical and material conditions that unfolded during this historical era produced what Ranciere refers to as the aesthetic regime of art. In Ranciere's understanding, the aesthetic regime does not operate through the canonization of art historical movements and theories, but rather indicates a particular mode of sensing for whatever objects in

⁶ Jacques Rancière, *The politics of aesthetics: The distribution of the sensible*, ed. and trans. Gabriel Rockhill (London: Bloomsbury, 2015), 4-7.

the world can be constituted within the realm of art. This mode of the sensible frees the art object from its relationship to canon, genre, and subject matter, its reliance on previous forms of cultural production and representation. As Ranciere tells us, art within the aesthetic regime is produced by a “heterogeneous power” as a singularity removed from previous designations of doing and making. It at once affirms the autonomy of aesthetic experience that is produced through the distribution of the sensible, yet simultaneously undoes any means of classifying its singular nature within art historical discourses.⁷ Here, I believe that Ranciere’s aesthetics allow us to think about techniques of modding as the product of a sensory regime through which new art forms, technicities, communities of practice, and modes of phenomenal engagement originate. The distribution of the sensible accounts for a multiplicity of subjectivities, identities, and sensual experiences in relation to art modding that extend from novice to technologically literate bodies. And, similar to the vibrant activity of both players and onlookers within an arcade, the public reception of art modding within the gallery implicates not only players but also spectators that view experiences of play. That is, the exhibition of mods I discuss within these chapters includes the phenomenal work of bodies that watch players interact with the strange spaces generated by the altered engine.

Theoretical Approaches to Video Game Art and the Avant-Garde

In this dissertation, I draw a relationship among video games and histories of the avant-garde because consumer technologies and experimental practices have held a close connection throughout art movements of the 20th and 21st centuries. Artists have often employed technologies contemporaneous with the era to critique mass commodification and conformity existing within capitalist structures. In this historical affiliation among technology and the avant-garde, I define “art modding” as a technical and phenomenal operation through what Matteo

⁷ Ranciere, *The politics of aesthetics*, 18-19.

Bittani suggests is the augmentation of existing software used to deconstruct the entertainment value of games and affirm non-normative modes of play. Thus, modding is often predicated on a type of unplayability—what I call aimlessness—that foregrounds abstract forms produced by disrupting the internal operations of the game system.⁸ To borrow Anne-Marie Schleiner’s term, mods are *parasitic* in nature: they re-contextualize popular game technologies to produce an aesthetic confrontation with participants.⁹

As I’ve mentioned, scholarship that addresses video games from the perspective of art history, defines modding largely through categories of representation and its relation to other movements of the avant-garde. John Sharp looks at modding practices through the ways artists conceptualize and produce works of game art based on their material affordances. Sharp situates modding as an aesthetic process of appropriation unfolding at the intersection of games and histories of art within the 20th century through the ways artists use video games “as tool sets and cultural tropes rather than as a medium or craft unto itself.”¹⁰ In Sharp’s view, a mod like Cory Arcangel’s *Super Mario Clouds* (2002) is a playful rearticulation of Duchamp’s *Fountain* (1917) or Robert Rauschenberg’s combines. Modding is a way of questioning the cultural value of the medium and what can ultimately be included as an aesthetic object. Although she doesn’t focus specifically on modding, Mary Flannigan’s work also positions video games within 20th century histories of the avant-garde yet largely through theories and concepts of play. In *Critical Play: Radical Game Design*, Flannigan introduces what she calls avant-garde game design, or when artists reuse mainstream gaming practices as a form of political and social subversion through

⁸ Matteo Bittani, “Game Art: (This) is not a Manifesto, (this) is a Disclaimer,” in *Gamescenes: Art in the Age of Videogames*, ed. Matteo Bittanti and Domenico Quaranta (Milano: Johan & Levi, 2006), 9.

⁹ Anne-Marie Schleiner quoted in Matteo Bittani, “Game Art: (This) is not a Manifesto, (this) is a Disclaimer,” 10.

¹⁰ John Sharp, *Works of Game: On the Aesthetics of Games and Art* (Cambridge, MA: MIT Press, 2015), 14.

playful engagements with the medium.¹¹ Flannigan considers radical appropriations of popular gaming technologies as a form of activism in relation to other counter-hegemonic approaches within the 20th century avant-garde.

Similarly, Brian Schrank looks at the ways in which video games shape the contemporary avant-garde and reciprocally how the avant-garde influences the production of games as artifacts within our current technoculture. He situates practices of modding within a multiplicity of aesthetic strategies that the technical conditions of video games make opportune. Games naturally reveal ways we interact with both technology and culture and their interactivity affords artists the possibilities to diverge from prior artistic mediums privileged in the historical and neo-avant-garde. Movements within the avant-garde have always found interest in using technologies contemporaneous with the times. Thus, instead of thinking about what defines the contemporary avant-garde and how games might fit into this definition, Schrank instead asks how do video games emerge from the avant-garde and allow for artists to deploy a range of formal and political strategies. Following Hal Foster and his historicizing of the neo-avant-garde, Schrank points out that there is no all-encompassing theory of the “avant-garde” but rather a dispersion of multiple avant-garde techniques and communities from which gaming practices emerge.¹²

Alexander Galloway proposes one such avenue for games within the contemporary avant-garde through what he calls counter-gaming: a strategy of subverting normal operations of play through modifications to existing software. He suggests that modding both disrupts the game’s flow while also foregrounding the materiality of the system and its unit operations.¹³

Although this scholarship does engage with issues of aesthetics that surround modding as a

¹¹ Mary Flanagan, *Critical play: radical game design* (Cambridge, MA: MIT Press, 2013), 1-2, 10.

¹² Brian Schrank, *Avant-garde videogames: playing with technoculture* (Cambridge, MA: The MIT Press, 2014), 3,

¹³ Alexander Galloway, “Counter-gaming,” *Gaming: Essays on Algorithmic Culture*. Minneapolis: University of Minnesota Press, 2006, 107-109.

specific trajectory of the avant-garde, it sidesteps what is at stake politically for the player's phenomenal relation to material forms generated by the game engine. In other words, current literature ignores the emergence of art modding as a singular type of aesthetic composed of linkages and connections within a heterogeneous terrain of popular gaming technologies, avant-garde communities, and contemporary attitudes toward technoculture. I expand upon this recent scholarship in game art and consider the ways modding generates sensory experience through what Gilles Deleuze calls "difference-in-itself," an affirmation of the unique singularities that compose a thing.¹⁴ Here, I want to validate modding beyond its taxonomical framing within the avant-garde, or rather what art modding *is*, to consider the possibilities for *what it does*; or, the intensities and perceptions produced in the body through our phenomenal relation to the game engine. Therefore, I define modding not in its categorical sameness to a specific strategy, tactic, or movement within iterations of the avant-garde, but rather through the ways in which its technical operations affirm a specific regime of sensible experience and new modes of aesthetic engagement with video games.

Methodologies

Throughout these episodes, I employ a range of methodologies to map the relationship between the materiality and technics of modding, the illusory, abstract forms they create, and the body's aesthetic encounter with those forms. Thus, my purpose here is to triangulate materiality, form, and phenomenology for each of the artists and works outlined in this dissertation. In order to examine the game engine as a software framework for modding practices, I draw upon recent methodological approaches grounded in media archaeology, what Jussi Parikka aptly describes as an "...analysis of the historical layers of media in their singularity—a conceptual and practical

¹⁴ Gilles Deleuze, *Difference and Repetition*, trans. Paul Patton (New York: Columbia University Press, 1994), 55-56

exercise in carving out the aesthetic, cultural, and political singularities of media.”¹⁵ I consider the emergence of art modding during the 1990s as a singular aesthetic practice and thus trace the technical operations of the game engine as a media technology that afforded artists unique possibilities to experiment with games.

To this end, I also look to histories and theories of animation within film and media studies. I believe that the inclusion of modding within processes of animation allows us to consider how mechanisms inherent to the game engine provide the technical generation of abstract forms in their vital movements. Therefore, I situate modding within recent conversations in animation studies to suggest that animation is the technical production of the mod’s strange formal articulations that are placed into movement by the artist’s manipulation and participant’s input. This approach also highlights the vitality of these moving forms and the minute ontologies that govern their computational processes. I believe these methods in media archeology and animation studies allow for a more inclusive definition of a “game engine.” I define an engine within these historical instantiations of modding as *a technical framework of hardware and software that generates video game forms through a range of computational mechanisms*. Therefore, each chapter focuses on specific historical game engine technologies with this definition in mind: the Nintendo Entertainment System, id Tech 1 (*Doom* engine), *Quake* engine, and id Tech 3 (*Quake III Arena* engine).

The work of Matthew Kirschenbaum provides a cornerstone to this media archeological method, specifically his concepts of forensic and formal materiality. Kirschenbaum states that forensics is the examination of the unique individuality of a media artifact and its interior processes at the level of code, digital inscription, and “where individual bit representations deposit discrete legible trails,” requiring technologies such as magnetic force microscopy for

¹⁵ Jussi Parikka and Garnet Hertz, "[CTheory Interview: Archaeologies of Media Art](#)", *ctheory.net*, April 1, 2010.

their analysis. It also includes the material surfaces, components, design, and labor that make up digital media. Forensic materiality is correlated to a media object's formal materiality, or the way in which interrelated layers of computer hardware and software act upon a user's experience. Kirchenbaum tells us that these computational layers, although individually self-supporting and complete in themselves, indicate "something of the procedural friction or perceived difference, the torque, as a user shifts from one set of software logics to another." It describes the push and pull of formal processes applied to an underlying structure of code, producing certain affordances and constraints for what software can do and through which user experiences become naturalized.¹⁶

By examining the fold between forensic and formal materialities, I trace the procedural behavior of game engine architectures to the formal parameters of their modifications. This close reading of the forensic and formal properties of engines is undertaken through archival work, which plays a significant role in my examination of art modding histories. Archival research of artists and programmers affords not only the study of technical innovations during the development of game engines, but also the techniques artists used to modify these engines. This thick description is guided by my own play of game engines as well as the mods produced from them. These play episodes have been attentive to the ways engine technologies construct dominant perspectival regimes. Through play, I also consider how these hegemonic modes of visuality are abstracted and as a result, question the body's phenomenal engagement with video games.

The next component of my methodology is an examination of the abstract forms produced by the game engine. Thus, I move from a thick technical description of these

¹⁶ Matthew Kirschenbaum, *Mechanisms: New Media and the Forensic Imagination* (Cambridge, MA: MIT Press, 2012), 10, 13, 20.

technologies to a formal analysis of the art mod informed by Kirchenbaum's tracing of a media artifact's forensic and formal materialities. This formal analysis is largely guided by aesthetic strategies of defamiliarization. Kirchenbaum suggests that the forensic investigation of a computational device resembles a process of defamiliarization through an unveiling of the mechanism's deep stratum of inscriptive operations. These computational processes are perceived as fundamentally alien in that they are removed from our experience of computers at the level of interfaces, software, and operating systems.¹⁷

In conjunction with the ways forensics construe media as strange, I draw upon aesthetics of defamiliarization from Russian formalist thought, especially the work of Viktor Shklovsky. In his essay "Art as Technique" from 1917, Shklovsky contends that defamiliarization is the underlying value of aesthetic practice. The purpose of art lies in its capacities to disrupt the habitual perception and conventions of a work, particularly in the way images produce an automated perception of an object that clouds the very means by which we come to know it. Shklovsky sees defamiliarization as an aesthetic strategy through which such perception is countered by imbuing the familiar conventions of an artwork with a sense of strangeness, or *ostranenie*. This strangeness Shklovsky describes as an act of "baring the device," is the way artistic techniques are self-reflexively foregrounded and made unfamiliar by an artist, calling out their artificial construction. Shklovsky tells us that defamiliarization expels "the automatism of perception" and instead produces "the vision which results from that deautomatised perception." Through defamiliarization, meaning and satisfaction elicited from a work are perceived through their lingering or *slowness*, or rather the particular continuity in space and time through which

¹⁷ Kirchenbaum, *Mechanisms*, 20, 133.

the unfamiliar becomes evident.¹⁸ I draw upon this formalist aesthetic to describe the means by which our attention is drawn to the mechanisms of the engine through a lingering contemplation of its minute operations.

Specific to the art mod, I believe that defamiliarization is ultimately expressed by the artists and programmers I highlight through aesthetic strategies of world-building and glitch. By “world-building,” I refer to the often surreal ways in which artists construct entities and environments that are seemingly unrelated to the diegetic space of the game, and employ the engine to render their appearances and movements within the gameworld. I think of world-building within these modifications through what Deleuze and Felix Guattari refer to as “disjunctive synthesis”: an affirmative relation between divergent parts, that although are separate in their qualities and modes of being, still interact through a *vital force of difference*.¹⁹ Here, the game engine acts as a force of difference through which the artist integrates custom weapons, characters, and maps into an existing game, and programs them to interact within the modified gameworld. Through this difference the game engine produces, world-building defamiliarizes the player’s experience by transmogrifying familiar in-game environments and objects into strange correlates. Kipcak and Urban’s use of the *Doom* engine to transform a gun into a cross, or Cory Arcangel’s technique of ROM hacking to include caricatures of rapper Flava Flav and The Pope in the NES game *Hogan’s Alley* for his work *I Shot Andy Warhol* (2002), take on an aesthetic of things *becoming something other than their familiar in-game status*.

¹⁸ Viktor Shklovsky, “Art as Technique,” in *Russian Formalist Criticism: Four Essays*, ed. Lee T. Lemon and Marion J. Reiss. (Lincoln, NE: University of Nebraska Press, 1965), 18, 22.

¹⁹ See Deleuze and Guattari’s concept of “disjunctive synthesis” in Gilles Deleuze and Félix Guattari, *Anti-Oedipus: capitalism and schizophrenia* (Minneapolis, MN: University of Minnesota Press, 1983), 76-77.

Along with concepts of world-building, I look to recent scholarship in glitch studies to consider computational glitch as a formal strategy used by artists to defamiliarize the player's standard perception of a game engine. I define glitch in relation to art modding through what artist Rosa Menkman calls the glitch's inherent *moment(um)*, or the possibilities for glitch to generate difference through the disruption or entropy of dominant technical frameworks, yet within the phenomenological moment through which such possibilities are initially experienced. She considers glitch as a rupture or unknown breach within the standard transmission of information in a system. It is a moment in which the computer or media device experiences a break in functionality, whether technical or social, that eludes codification within the norms of technoculture.²⁰ Menkman's assertion of glitch is particularly helpful in thinking about a formal analysis of art modding as it foregrounds the forensic materiality of the game engine. It unveils to us the alien processes of the system through an aesthetic of error that makes bizarre our standard experience of the video game.

Lastly, my methodology explores phenomenal experience of the art mod through an aesthetic of mimesis. I believe the forensic operations of code that generate abstract form, modulate the mimetic capacities of the player to *feel* the distortion of gamespaces—both two and three dimensional—as an intensive, bodily process. Thus, I define and situate aesthetics within theories of media phenomenology that explore the player's sensual encounter with the art mod. I believe that this consideration of art modding aesthetics through phenomenological traditions of mimesis allows us to rethink representation, figuration, and the illustrative through non-representational and embodied modes of experience. In using the term, I frame mimesis as an abstraction away from representational concepts of verisimilitude, naturalism, and imitation toward a non-representational theory grounded in our sensual engagement with form.

²⁰ Rosa Menkman, *The glitch moment(um)* (Amsterdam: Institute of Network Cultures, 2011), 8,9, 26,28.

Throughout these episodes, I look at aesthetics as a mimetic operation through which the player encounters the technical image of the mod and experiences its vital rhythms and movements as varying levels intensity in the body.

Here, I want to emphasize that modding aesthetics reflect a process of affective reciprocation between body and altered game engine. The unique *work* of aesthetics I discuss in relation to the specific mods throughout these chapters points to the valence among player and engine through which human (and non-human) phenomenologies emerge. I consider aesthetics through a neutral ontology: an exchange between the body's affectivity and the vital energies of form generated within the computational environment of the mod. Therefore, aesthetics becomes the technical capacity of the game engine to modulate and attune the senses through new and unexpected avenues of embodied experience. As I've mentioned, I believe that artists were invested in modding experimentations as a way to interrogate bodily experience through an aesthetic of mimesis that destabilizes the player's apperceptive and proprioceptive senses. In its processes of making strange familiar gamespaces, the mod engages the player in what I call *gamic apperception*, a bodily sensation of uncanniness or unfamiliarity in which our awareness is drawn to the very strangeness of the engine's formal disruptions. It is the body's affective response to the game system laying bare the material constructs and perceptual mechanisms through which we habitually play. It operates as a mode of intuiting the act of perception in itself. The other affective modality I situate in this aesthetic of mimesis is what I call *gamic proprioception*, or the player's affective attunement toward movement through and spatial grounding within actionable gamespaces. This describes the body's autonomic, intensive capacities to locate and orient itself within the gamespace through balance, muscle coordination,

and vestibular functions. By disordering the player's spatial grounding within the video game, modding aesthetics prompt an unraveling of the body's proprioceptive equilibrium.

Throughout this dissertation, I consider gamic apperception as closely related to proprioception in that both are constitutive of the body's affectivity to be aware of its own orientation within the gamespace, and I focus on the interrelatedness of these two terms as a means to articulate our mimetic engagement with the art mod. The player's experience of these disrupted gamespaces point to a politically-organized body that is apperceptively aware of the game engine's materiality and proprioceptively destabilized by its formal abstraction. I approach this aesthetic of mimesis through a methodological interweaving of several materialist and phenomenological traditions: theories of affect through the work of Henri Bergson and the aesthetics of Gilles Deleuze; Walter Benjamin's Marxist, materialist theories of media; the existential phenomenology of Maurice Merleau-Ponty; and the metaphysics of Alfred North Whitehead, among others.

Within these schools of thought, I look at mimesis as an affective resonance generated from an encounter with the art mod and the sensation of the familiar video game form transformed into *something-other*: an entity that exceeds the representational properties of what environments and objects appear to be to a player. I believe that these materialist traditions provide insight into our affective relation to art modding through interactivity with the medium and consider how expressions of game art modulate our bodily affectivity and immanence in the world as well as the embodied meaning we give to games as aesthetic objects. My understanding of mimesis within these phenomenological traditions is guided by the materialist thought of Henri Bergson in Mark Hansen's philosophy of digital media. Drawing from Bergson and his triangulation of perception, affection, and memory, and their role in embodied experience,

Hansen argues for the body's capacity to frame the perception of digital media through a process of diminution or reducing down the material images of a media work from other images in the universe.

Our perceptual selectivity of the qualities inherent to media in turn generate bodily affectivity, or to borrow from Bergson, affection as the body's "actual effort upon itself," a "zone of indetermination" or affective propensity to hesitate before a motor action is taken.²¹ Hansen proposes that this region of indeterminacy resulting from perception engages the body's creative capacities to "experience itself more than itself"; rather, its ability to produce embodied sensation as a process of instability, experimentation, or difference outside of rational cognition and representational frameworks. To Hansen, when the body perceptually frames digital information, it negotiates its own affectivity and "gives body" to the digital work by transforming disembodied information into something that possesses human meaning.²²

I believe that Hansen's reading of our affective embodiment in the presence of new media art, allows us to consider the ways in which art modding catalyzes bodily affectivity. Here, I refer to the virtuality of the body and the possibilities for it to feel itself in a surplus of experience—to intensively produce the new or unexpected—in our interactions with the game engine and its computational logic. Bergson informs us that perception is merely a process of the body translating other durations in the world so as to make them comprehensible within our reality, or our own embodied process of becoming. Our selective framing of material things and the rhythms, vibrations, and movements of matter that constitute their being, reflect the body's capacity to slow down and attune our perception to different durations within the world.²³ The

²¹ Henri Bergson, *Matter and Memory*. Trans. by Nancy Margaret Paul and William Scott Palmer. London: G. Allen &, 1912, 58-60, 21, 23, 40.

²² Mark B. N. Hansen, *New philosophy for new media* (Cambridge, MA: MIT Press, 2006), 6-7, 12-13.

²³ Henri Bergson, *Matter and Memory*, 186, 207.

body's phenomenal relation to the art mod involves a perceptual and sensory concretization of the material rhythms and temporalities of the game system. Therefore, I see the work of modding aesthetics as the body's engagement with the strange ontologies that compose the modified engine. To this end, my methods are informed by recent materialist thought in object-oriented ontology and speculative realism through the work of Ian Bogost and Jane Bennet among others, that inquire into the vital being of things and objects outside an anthropocentric framework. In considering the possibilities for how the player's affectivity is modulated by the game engine, we can also look to the unique agency of the art mod as a thing that has a neutral ontological status in its phenomenal encounter with the body. In Bergson's terms, an ontology of the art mod indicates its own alien perception of the body and computational abilities to reduce down and interpret human input into abstract forms. Thus, my methodology of aesthetics has a particular investment in the ways video game systems perceive us in a phenomenal relation among player, game engine, and world.

Chapter one "A (Scroll) Down Memory Lane: Non-Play, Commodity Aesthetics, and the Vitality of 8-Bit Engines in Cory Arcangel's *Super Mario Bros. Mods*," focuses on two works by artist Cory Arcangel: *Super Mario Clouds* (2002) and *Super Mario Movie* (2005). Arcangel's modding techniques consist of hacks to Nintendo Entertainment System game cartridges and their read-only memory (ROM). By rewriting the game's ROM chip, Arcangel alters the game code so when interpreted by the NES console, the game exhibits abstract, time-based animations. Here, I provide a recent archaeology of the game engine as a technical object through a close-reading of the NES system's split architecture and particularly the capacities for its 6502 processor and Picture Processing Unit to generate 8-bit gameworlds from the ROM chips embedded in each game cartridge. I further situate the NES within a more inclusive definition of

the engine as a technical platform for the generation of animated images beyond our discursive understanding of its development with games such as *Doom* and *Quake*. Through Arcangel's work, I demonstrate that an engine is fundamentally the code of a game system that coordinates complex operations among software and hardware such as graphics rendering and physics.

I argue that Arcangel's mods introduce an aesthetic of defamiliarization through ROM hacking techniques and glitch by disclosing the unique speculative phenomenologies of the Nintendo's 8-bit microprocessor and its units of operation. Arcangel uses the NES to generate abstract forms that engage the body's affective resonance with the system by making strange our perception of lateral movement through a scrolling of two-dimensional space. By taking away our ability to control *Super Mario Brothers* (1985), Arcangel recontextualizes the scrolling operations of the NES engine used for player navigation so that they engage the body in dynamic patterns of glitch. In disclosing the interobjectivity of the NES architecture through a glitch aesthetic, Arcangel's mods catalyze gamic apperceptivity while destabilizing our proprioceptive engagement with the two-dimensionality of the 8-bit platformer genre: what I call *embodied non-play*. In using this term, I refer to the player's inability to control the game through a disruption of our serial interfacing with the system. In this aesthetic of non-play, Arcangel's creative reuse of the NES as a *démodé* commodity, especially his integration of fragmented game icons into *Super Mario Clouds* and *Super Mario Movie*, modulates the body's proprioceptive sensitivity to the scrolling gameworld. As I argue, these animated sequences of the deconstructed pop icons, result in an affirmative sense of difference through the body's kinesthetic response to the scrolling and flickering 8-bit tiles that Arcangel programs into movement.

In chapter two "Slaying Machines: Embodied Mimesis in *ArsDoom*," I look at Orhan Kipcak and Reini Urban's *ArsDoom* to unveil the historical conditions from which the game

engine and modding techniques came into being beginning in the early 1990s. In performing an archaeological examination of John Romero and John Carmack's *Doom*, I consider the ways in which the engine proffered a new epistemology of modding practices for constructing and animating gameworlds. Romero and Carmack solidified our knowledge of an engine as a reusable platform for 3D content creation that is also separated from the content it renders.

As a modification derived from the *Doom* engine, *ArsDoom* operates through strategies of institutional critique in histories of the avant-garde, while evoking a mimetic response articulated as vectors of embodied vision within the 3D environments of the game. Here, I argue that modifying *Doom*'s familiar interface into uncanny sequences of play in *ArsDoom*, produces sensations of the mod's strange spaces mirrored intensively within the body. Through our encounter with *ArsDoom*'s surreal objects, spaces, and mechanics—blood-paint, crucifix-gun, gallery-deathmatch—mimesis is articulated as the sensation of things transforming into something-other than their gamic representation.

In *ArsDoom*, the player's experience of running and shooting across vectors of line within the simulated Brucknerhaus gallery, at once questions the body's proprioceptive receptivity through abstractions of the gamespace, while apperceptively engaging the senses in playful, mimetic associations among objects and environments. In the act of appropriating the *Doom* engine and reconfiguring the game into a bloody deathmatch of the Ars Electronica festival, Kipcak and Urban's mod reveals the engine's modularity while exposing institutional power through mimetic affinities among art world politics and first-person shooter tropes.

Chapter three, "'Perspective Engines" and the Strangeness of 3D Space in *JODI's Untitled Game*," examines the *Quake* engine as it was repurposed by the art duo JODI to create their work *Untitled Game*: a series of 14 mods developed between 1996 and 2001 that disrupt the

game's fully-actionable, real-time 3D space. By altering *Quake*'s source code, JODI reconfigure the game's 3D space as a series of bizarre psychedelic abstractions. Within this historical moment of modding, I look at the aesthetic work of *Untitled Game* through a triangulation of the technics of the *Quake* engine created by John Romero and John Carmack throughout 1995 and 1996 in its abilities to render fully actionable 3D space, to the formal experimentations of JODI's mods and the player's aesthetic investment in their strangeness.

In JODI's work, glitch reveals our phenomenal experience through the rendering of *anamorphic space* that impedes our normal perception of perspectival space common to the game engine. Through a mimetic encounter with anamorphosis, bodily experience emerges as both our apperceptive attunement to the glitch's formal disruption and a proprioceptive instability from its warped presence. Here, I situate the perceptual fluctuations generated from *Untitled Game* within avant-garde traditions of the structural flicker film, referring to JODI's work as a *flicker mod*. As with the flicker film, *Untitled Game* produces a threshold experience that simultaneously elicits perceptual awareness of the mods' effects while inciting sensual, autonomic responses from the body.

I focus on three of JODI's mods, *Ctrl-9*, *I-N*, and *Q-L* that demonstrate the body's affective engagement with anamorphic space through the flicker effect of glitch. First, I look at the historical emergence of the technics of the *Quake* engine through a close reading of what Matthew Kirchenbaum refers to as a grammatology of a computational artifact. I then trace the forensic environment of the engine to the surface of its spatial disruptions by examining the ways JODI have altered commands and variables within *Untitled Game*'s *autoexec.cfg* files. Finally, I look at the aesthetic work of *Untitled Game* upon the player through an embodied form of mimesis that generates anamorphosis as an abstract, intensive space reflected in the body. The

anamorphic nature of glitch in *Ctrl-9*, *I-N*, and *Q-L* engages a bodily mode of apperceptivity, revealing our automated perception of actionable, real-time 3D space. This apperceptivity co-extends to the player's proprioceptive capacities through kinesthetic illusions of movement—rotating, pulsating, and somersaulting. As I suggest, *Untitled Game* modulates our spatial resonance in the world just as it attunes the body to its own self-sensitivity toward navigating anamorphic space.

Finally, in chapter four “Generative Mods and the Violence of Sensation,” I look at two works by artists Julian Oliver and Tom Betts that utilize the *Quake III: Arena* engine to generate on-going glitch patterns through automatic processes. Specifically, I focus on Oliver's mod *ioq3apaint* created in different versions between 2002 and 2010, which uses the AI system and a “redraw” glitch in the engine to produce painterly abstractions of trailing frames and motion blurs. I also examine the work of Tom Betts who in 2002 used the engine to create his mod *QQQ*: a generative system that uses a multiplayer deathmatch server to render mutations of glitch in real-time.

In order to think beyond the representationalist labels of high modernism that Oliver and Betts' mods are often assigned, I frame the aesthetic importance of their work through the generative techniques that underlie their modes of production. Here, I situate their work within kinetic and conceptual art movements during the 1960s to understand how their mods articulate complex systems that generate aesthetic potential through their autonomous operations. In particular, I examine the generative strategies used by Jean Tinguely in his Meta-matic sculptures—machines that could draw abstract compositions and would occasionally self-destruct based upon chance and their own machinic autonomy. Likewise, I turn to the

installations of Hans Haacke that sought to produce generative outcomes outside the artist's control through natural and physical processes, namely temperature, humidity, air currents, etc.

Like Tinquely's unpredictable machines and Haacke's natural systems, our aesthetic engagement with *ioq3apaint* and *QQQ* is a result of the variations they generate within a dynamic and intricate system of human and technological bodies. Apperceptive-proprioceptive sensation arises from the flattening of 3D space in *Quake III: Arena*. Autonomous operations of glitch make strange our experience of the arena deathmatch by dispersing perception across a visual field of abstraction and at the same time, turning our gaze to the engine's repurposing as a generative system. Borrowing from Gilles Deleuze's work on the painter Francis Bacon, I suggest that the spatial disruption in *ioq3apaint* and *QQQ* produce a mimetic encounter in which the violence of the arena deathmatch is transposed within the body into a nonrepresentational *violence of sensation*.

CHAPTER 1

A (Scroll) Down Memory Lane: Non-Play, Commodity Aesthetics, and the Vitality of 8-bit Engines in Cory Arcangel's *Super Mario Bros.* Mods

I would love to say there was some contemporary artist who's work really got me thinking, but lately I have just been trying to sort out 20 years of garbage TV culture that is filling my brain.

My real experience with video games was watching other people play. That's why a lot of my work isn't really about playing. It's about watching video games.

-Cory Arcangel

Since the early 2000s, Cory Arcangel's work has investigated the relationship between the art world, digital media, and technicities of the Information Age. Working with video game modifications, installation, performance, and digital music, his projects range from hacking Nintendo cartridges to produce abstract animations, creating large color field C-prints using simple gradient tools in Photoshop, or arranging Arnold Schoenberg's infamously complex atonal *op. 11 Drei Klavierstücke* using notes hit by unsuspecting cats walking along piano keys in YouTube videos. Arcangel's practices often engage the public through a Warholian pop sensibility, at once appropriating popular forms of technology while playfully subverting our habitual perceptions of digital media. Like the other artists discussed within these historical episodes, Arcangel approached game modding through other creative disciplines, namely that of musical composition. In 2000, he graduated from the Oberlin College Conservatory of Music, where he studied classical guitar and majored in music technology. At Oberlin he met fellow collaborators Paul B. Davis, Joseph Beuckman, and Joe Bonn, forming the Beige Programming Ensemble. Working under the label BEIGE, together they released *The 8-Bit Construction Set* (2000), an album produced in its entirety using 8-bit sounds captured from older game consoles. Arcangel's Cagean approach to composition which foregrounded the use of readymade

technologies to generate musical soundscapes, would influence his critical exploration of coding and video games as an artistic medium.

With his game mods in particular, Arcangel attempts to defamiliarize expectations of normative play often by limiting the player's ability to control the game or disrupting the game's flow so as to compromise any chances of winning. For instance, his mod *Super Slow Tetris* (2004), a hacked NES cartridge of *Tetris*, subjects players to a version of the classic game in which it takes roughly eight hours for the blocks to fall and finish a round. Players still have the ability to move blocks right and left but the excruciating fall time makes the game nearly impossible to play. In some of his game mods, Arcangel removes player agency entirely, presenting his work as time-based animated sequences. As with his work *Super Mario Clouds* (2002) (abbrev. *SMC*), Arcangel re-wrote the source code to *Super Mario Bros.* (Nintendo, 1985) (abbrev. *SMB*) using a hex editor, removing the PRG-ROM from the game cartridge and burning new code onto the chip so that only blue sky with continually scrolling clouds remain and all other graphics are erased. Similarly, with his mod *Super Mario Movie* (2005) (abbrev. *SMM*), Arcangel in collaboration with *Paper Rad* collective used similar ROM hacking techniques to program an experimental animated film: a tragic machinima in which our protagonist Mario contemplates his existence among the pulsations of psychedelic glitches within a deteriorating gameworld.

Christiane Paul, the curator of Arcangel's 2011 *Pro Tools* exhibition at the Whitney Museum of American Art, argues that his works investigate the braiding among knowledge frameworks of tech industry expertise and the DIY ethos prevalent in punk subcultures of the 1980s.¹ This DIY ethos becomes especially evident in the documentation on his website as well as in his zine project *The Source*: a collection of instructions and code for his projects. These

¹ Christiane Paul, "Cory Arcangel: Pro Tools," *Whitney Museum of American Art*, 2011, 1.

projects allow the public to replicate *SMC* or software such as *Pizza Party* in which a user can order Domino's pizza at reoccurring times during the week using command line. Perhaps the most significant theme that runs throughout Arcangel's work is an affinity for repurposing démodé technologies within the recent past of the Information Age, in turn, eliciting our nostalgic remembrance for such technical artifacts. Arcangel grew up in the 1980s among the growing popularity of gaming consoles and personal computers, technologies that would come to exemplify dominant technoculture along with the development of the Internet in the early 1990s. In an interview with artist Mary Heilmann, Arcangel says that he grew up with an Apple IIGS and Commodore 64, however, was not an avid gamer. Instead his fascination with the cultural practice of gaming stemmed from *watching* a friend play games on an NES system and as he mentions, his game mods—particularly his *SMB* hacks—are largely about the aesthetic work of spectatorship. They refer to a mode of embodied perception that engages the viewer in a type of “non-play” instead of standard interactivity requiring input from peripherals.²

As the focus of this chapter, I look at Arcangel's unconventional repurposing of the NES as a platform for producing time-based animated sequences which question standard relationships to the Nintendo Entertainment System and two-dimensional play within the platforming genre. In particular, I examine Arcangel's *Super Mario Clouds* and *Super Mario Movie*, two modifications that exploit this affective modality of non-play by eliciting our nostalgic longing for the now obsolete game system. As I argue, Arcangel's game modifications using the NES point to a broader definition of the game engine beyond its conventional understanding with the development of Carmack and Romero's *Doom* and *Quake*. By examining a recent archaeology of the game engine beginning with the introduction of the NES platform

² Cory Arcangel, interview by Mary Heilmann, *Interview Magazine*, March 21· 2011, <https://www.interviewmagazine.com/art/cory-arcangel>.

during the mid-1980s, my aim is to assert that an engine is not just software for developing advanced graphics and novel forms of three-dimensional space. Rather at its essence, an engine is the underlying code of a videogame system that correlates and systematizes computational operations within a platform: rendering graphics, performing physics, and interpreting relationships between in-game objects and states. Arcangel's *SMB* mods indicate that games of the NES 8-bit era were in fact highly efficient and inventive engines that animated gameworlds through a unique coordination among software code and hardware components. In employing the NES as a generative framework for the vital movement of abstract forms, Arcangel's *SMB* mods elicit affectivity by removing our ability to control lateral movement, defamiliarizing our embodied perception of scrolling through two-dimensional space. Arcangel employs the scrolling capacities of the NES engine not for player navigation but rather to engage the body in animated patterns of graphical abstraction. Arcangel's ROM hacking techniques destabilize the play experiences of 2D platforming through a dismissal of a controller—what I refer to throughout this chapter as non-play—by prompting gamic apperceptivity and disclosing to us the speculative ontologies of the NES architecture. It is also in this aesthetic encounter of non-play with *SMC* and *SMM*, that Arcangel's reuse of the NES as a *démodé* commodity, particularly in the ways he integrates retro game icons and glitch aesthetics, modulates our proprioceptive attunement to movement in *SMB*'s two-dimensional gameworld. Through his inclusion of deconstructed pop image fragments, Arcangel's mods produce an affirmative sense of difference through a kinesthetic impression of scrolling and strobing graphic tiles choreographed on the screen.

The Technics of the NES and Arcangel's ROM Hacking

The Japanese video game company Nintendo achieved great success through their hand-held console *Game & Watch* in 1980, and several arcade games including the wildly popular *Donkey Kong* in 1981. During this time, Nintendo president Hiroshi Yamauchi along with engineer Masayuki Uemura, began executing plans for the Famicom, short for Family Computer: a cartridge-based console that sought to compliment the domestic space of the family room within Japanese households. Initially launched in Japan in 1983 with a colorful, toy-like appearance, the Famicom emerged among the U.S. videogame industry collapse with companies like Atari, Coleco, and Magnavox, largely a result of market saturation and competition from the home computer. In 1985, the red and white Famicom model was launched in North America and reworked into the aesthetically-restrained grey and black Nintendo Entertainment System. In his book *I AM ERROR*, Nathan Altice investigates the material affordances of the Famicom and its technical protégé the NES through a platform studies methodology, looking specifically at the hardware, software, and reception of those systems.³

Here, I largely draw upon Altice's work on the cultural production and reception of the Famicom/NES platform to articulate an archaeology of the game engine as a software framework for rendering different kinds of gamespaces, and to better explain the processes of Arcangel's ROM hacking. The Famicom/NES was a prominent console among the 8-bit generation, with "8-bit" designating the binary possibilities for values processed by the CPU. Because binary is a base 2 system (either 1 or 0) the NES can generate 2 (to the 8) or 256 possible values. Employing an MOS 6502 processor, the NES includes two types of ROM (Random Access Memory) with each game cartridge: the PRG-ROM that stores the game's program source code and the CHR-

³ Nathan Altice, *I am error: the Nintendo family computer/entertainment system platform* (Cambridge, MA: The MIT Press, 2017), 3, 6.

ROM which includes graphics data. The 6502 processor includes a 16-bit bus address that can process a total of 64KB of memory. The inclusion of two separate ROM chips for each game's source code and graphics palette was a design choice made by Uemura so that individual programmers could make decisions on how to allocate memory to either the CPU or the console's PPU: the Picture Processing Unit responsible for transferring graphics into video display.⁴ The 6502 processor includes an 8-bit data bus which sends data one byte at a time within what is called a memory map. The NES was designed to access only 2KB of RAM from the CPU while the remainder of the memory address is designated for the PPU, the APU (Audio Processing Unit), game cartridge, and serial information for peripherals. Because the NES employs cartridges with ROM chips in its interior to transfer game data, 16KB of this memory map is set aside for the PRG-ROM. Another 8KB within the CHR-ROM is used for processing pixel arrangements as graphic tiles and 8KB is used for what are called name tables that store pattern tiles for a game's background, and attribute tables that store palette information for background scenery.⁵

The architecture of the PPU includes a separate 14-bit address bus that accesses up to 16KB of memory allocated to graphical data in the form of palettes and tables. Within each game cartridge, 8KB of memory are stored in the CHR-ROM in the form of pattern tables comprised of both 8x8 and 8x16 pixel patterns made up of graphic tiles, while 8KB are coordinated to palette data, and attribute and name tables. As Altice explains, the "split architecture" at work in the NES mirrors the division of separate ROM chips within a cartridge when it is inserted. In other words, the CPU carries out the game's source code, while the PPU routes and displays pattern tables as graphics. The CPU can communicate with the PPU, telling it to arrange graphic

⁴ Altice, *I am error*, 27.

⁵ Altice, *I am error*, 27, 31.

tiles or swap out palettes, but it cannot retrieve these graphic tiles directly.⁶ The NES can only operate and render game data from the code stored within the PRG-ROM as it does not have a proprietary operating system. The source code for each NES game engine is compiled using assembly language which compared to languages such as C++, is considerably more incisive and similar to machine code. Written using a hexadecimal base 16 numerical system, assembly code includes lines of three-letter mnemonic along with a number value. The mnemonic portion of code indicates instructions for the CPU to carry out while the number represents a specific address or value. A base 16 numerical system means that each potential digit can be counted up to sixteen, meaning that it uses numerical values 0-9 as well as A through F. Altice suggests that because assembly language is merely a level of abstraction away from machine code and bears a close resemblance to the CPU operations of the NES, it allowed programmers to meticulously control hardware and subsequently generate new types of game physics and graphic worlds.⁷

Designed by Shigeru Miyamoto, Nintendo released *Super Mario Bros.* in 1985, which would become one of the most influential and well-designed video games in the medium's history. The success of Miyamoto's fantastical narrative, in which our plumber heroes Mario and Luigi travel throughout the Mushroom Kingdom battling surreal creatures to save Princess Toadstool from the evil King Koopa, is largely based on the game's side-scrolling "platformer" design. Miyamoto along with programmers Toshihiko Nakago and Kazuaki Morita created the *SMB* engine to utilize the full capacities of the NES split hardware architecture. It was a game meant to showcase the possibilities for what an 8-bit system can do with expressively rendering a two-dimensional gameworld. Nintendo began to explore the development of the platformer genre with their arcade cabinets *Donkey Kong* and *Mario Bros.* (1983). Both games required the player

⁶ Altice, *I am error*, 32.

⁷ Altice, *I am error*, 32, 50-51.

to jump up and across multiple tiers or levels but did not reveal a detailed world through a horizontally-scrolling screen. In essence, *SMB* positioned itself as the quintessential platformer because it introduced players to a series enemy encounters and obstacles that must be navigated across in order to reach the end of a level.

As Altice notes, Nakago and Morita were able to program the *SMB* engine with only 40 KB by using metatiles and compression. A metatile allows for more complex graphics by compressing combinations of multiple graphic tiles grouped as one object. As an example, Mario and Luigi are their own separate metatiles comprised of distinct tiles and palettes as are the Goombas, clouds, pipes, and so on. The *SMB* engine was designed to move graphic tiles based on grouping objects as types. Thus the source code can execute the movement of metatiles and tiles based on categories such as “enemies” or “player objects,” saving the engine additional RAM needed to execute numerous, complex subroutines.⁸ Compression also plays a fundamental role in how *SMB* utilizes repeatable sections of tiles to generate combinations of graphics for different levels. Varied worlds in the Mushroom Kingdom appear as such because palettes are swapped out for combinations of graphic tiles, producing the effect of levels having their own unique styles: underwater worlds, lava-filled dungeons, etc.

With NES engines, the PPU RAM cannot store multiple name tables of pattern tiles and thus compressing metatiles saves processing and memory requirements by grouping and handling larger, integrated graphic patterns which are unrolled throughout *SMB*'s longer two-dimensional levels. In this sense, an efficient NES engine is designed to group objects into specific types to which physics and rules can be assigned. For instance, Mario and Luigi both activate an extra life if they jump and hit the bottom of a box containing a 1up, or certain enemies will die in the same way if Mario jumps on their heads. Nagako and Morita developed

⁸ Altice, *I am error*, 123-124.

the *SMB* engine so that it renders each level based on parsing objects as single entities or rather, groups of graphic tiles into what Altice calls set decoration, area object data, and enemy object data.⁹ The set decoration component is the processes responsible for continually revealing the background and foreground terrain of each level: the clouds, sky, hills, buildings, and plants that comprise the Mushroom Kingdom. The area object data refers to the engine rendering pipes, bricks, and power-up blocks that Mario interacts with by either hitting or jumping over. These objects comprise the platforming elements of the game. Enemy object data refers to the

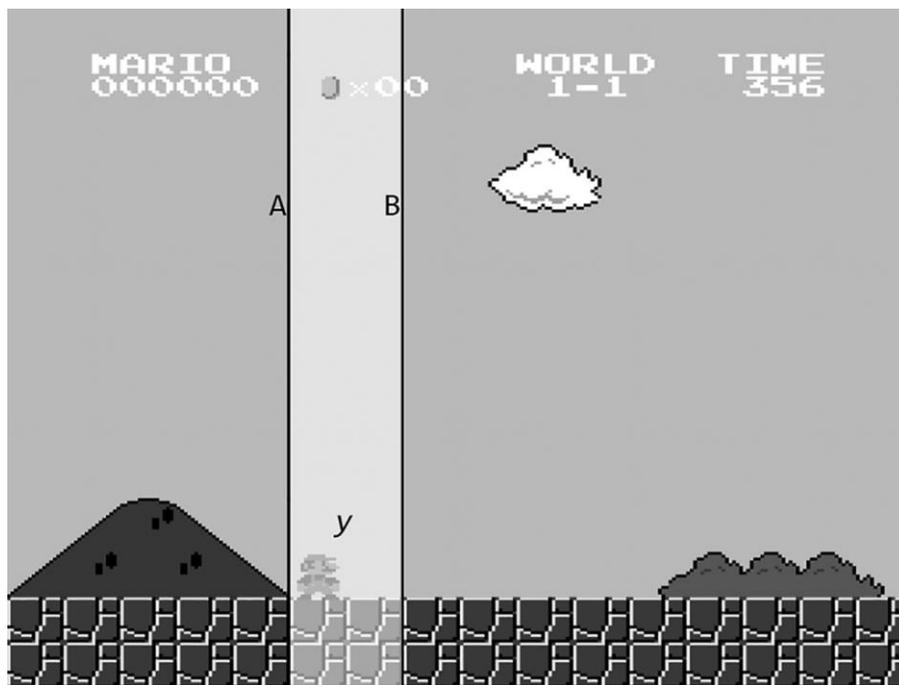


Figure 1.1: Horizontal scrolling thresholds in the *SMB* engine, Nathan Altice, *I am error: the Nintendo family computer/entertainment system platform* (Cambridge, MA: The MIT Press, 2017), 153.

operations that dictate enemies obstructing Mario's movement through scrolling 2D space. The engine's ability to scroll the in-game camera in coordination with the player's movement is based on algorithms that define two thresholds along the horizontal axis of the screen. The

⁹ Altice, *I am error*, 127.

primary threshold is located at 80 pixels from the left side of the screen, while the second threshold is at 112 pixels from the left side of the screen. Once the player moves the furthest pixel on Mario's left side past the first threshold, the engine begins to render the scrolling gameworld (Figure 1.1). As Mario accelerates, his metatiles remain within the 32-pixel area between both thresholds. Once Mario hits his top running speed, he eventually remains constant with the rightmost threshold.¹⁰ I now examine Arcangel's alterations to the *SMB* engine so as to understand how his written assembly code, when interacting with the NES split architecture, generates abstract forms in both his *SMC* and *SMM* mods.

Arcangel's *SMC* has become a canonical work within ROM hacking communities and the field of game and media art more broadly. It is included in the permanent collection at the Whitney Museum of American Art, and was exhibited at the Whitney Biennial in 2004, again in 2009, and at Arcangel's *Pro Tools* show in 2011. The majority of art historical narrative and criticism situate *SMC* within discourses of pop art and landscape painting through the appropriation of an entertainment commodity. In his DIY tutorial *The Making of Super Mario Clouds* written and updated in 2002-2003, Arcangel provides commentary, source code, and step by step instructions for individuals to reproduce their own version of the mod. As Arcangel describes, he "took an old Super Mario Brothers Nintendo video game and erased everything but the clouds."¹¹ In conjunction with the 2004 Whitney Biennial, Arcangel released a "documentary" video of the ROM hacking process also titled *The Making of Super Mario Clouds* which serves as a how-to video and documentation of his technical process. At 76 minutes, Arcangel films his *SMB* hack utilizing an intentionally shaky camera while framing certain

¹⁰ Altice, *I am error*, 153.

¹¹ Cory Arcangel, "The Making of Super Mario Clouds," in *Cory Arcangel: Beige*, ed. Raphael Gygax (JPR, 2006), 106.

minutiae of everyday life in his apartment: an electric guitar, a tartan-upholstered couch, wire cutters, a VCR, speakers, and a mess of old NES cartridges.

In the initial lines of Arcangel's code he includes various disses from pop culture references, in particular the quote "punks jump up to get beat down" by the hip hop group Grand Nubian: a reference to other artists who according to Arcangel, "think they can step up to my style."¹² Arcangel includes his reasoning behind these disses within his code. They are a citation to the cracking scene of the 1980s in which gamers would null the copyright protection to Commodore 64 games so as to produce elaborate animation sequences, whilst often insulting other cracking groups. As with the majority of programmers who developed NES engines, Arcangel mentions that his aesthetic choices for using assembly language are because the code is only a step away from the language of 1s and 0s, allowing for precise control over the NES hardware and its rendering capacities. Using assembly language, he establishes a sequence of code for the compiler, or the operations that convert the source code into machine level code so that it can be interpreted by the NES. The sequence also tells the compiler to work with the PRG-ROM chip at 32K and with a file titled "clouds.hex."

The next sequence of code informs the cartridge ROMs to operate once the NES reset button is pressed. Arcangel then provides a section of code that loads the colors used for the clouds and sky for the cartridge's PPU and allocates memory for the palettes blue, light blue, black, and white. As Arcangel mentions, he removes only the PRG-ROM chip and keeps the CHR-ROM intact, stating that "since I do not touch the graphics from the original cartridge, the clouds you see are the actual factory soldered clouds that come on the Mario cartridge. There is no generation loss, and no 'copying' because I did not even have to make a copy. Wasss up."¹³

¹² Arcangel, "The Making of Super Mario Clouds," 106-107.

¹³ Arcangel, "The Making of Super Mario Clouds," 109.

Arcangel then includes sections of code that load the name tables for the “cloud.hex” file mentioned earlier as well as portions that establish the NES graphic settings for the clouds’ scrolling movement when the screen is turned on. Lastly, Arcangel includes the sections of code that actually render the scrolling clouds. In “Section 2” Arcangel lays out the process of manipulating the cartridge and removing the PRG-ROM by clipping the legs from the chip and melting the connecting wires with a soldering iron. The next step requires soldering a low profile 28-pin socket so that a new PRG-ROM chip can be inserted and removed without additional soldering. In the final steps, Archangel burns the source code (the 32k file) onto a new 27C256 chip using an EEPROM burner, places the new chip into the cartridge, and cuts a whole in the front area of the plastic so that the new PRG-ROM can fit in the place of the original ROM

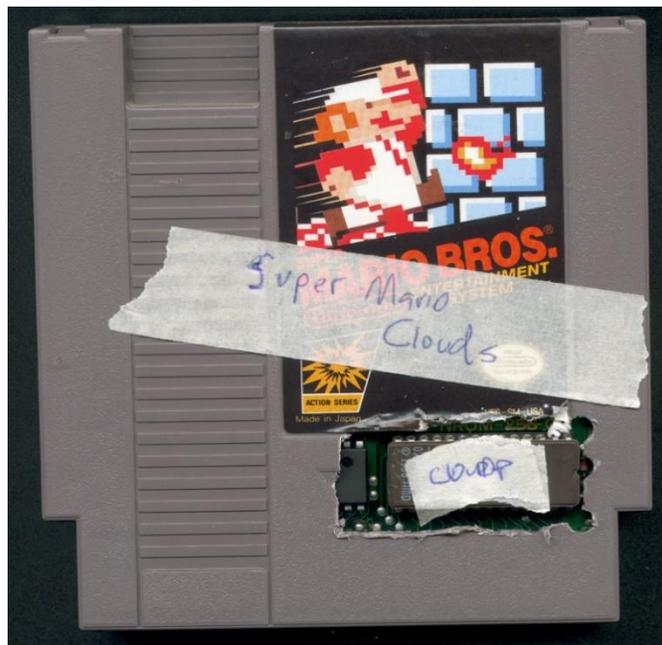


Figure 1.2: Hacked *Super Mario Bros.* cartridge with exposed PRG-ROM chip (Cory Arcangel, 2002). Photo courtesy of the artist’s website at <http://www.coryarcangel.com>.

chip (Figure 1.2).¹⁴ Yet, however much Arcangel's documentation appears to provide a DIY, open-source framework for rendering the sky and fluffy cartoon-like clouds sans the remainder of the Mushroom Kingdom, his tutorials for *SMC* ambiguously play with the fidelity of the clouds to their source material in *SMB*. In his analysis of *SMC* aptly titled *Everything but the Clouds*, Patrick LeMieux questions the fidelity of Arcangel's clouds and the artist's claims of appropriating their original graphic data. Looking at Arcangel's code and framing devices in the documentary tutorial of *SMC*, LeMieux argues that the artist's DIY ethos intentionally contradicts his artwork. *SMC* uses the NES engine architecture to repurpose the existing cartridge ROM, but rather to generate a type of "home brew game" through an asignifying rupture of the original cloud icons.¹⁵ LeMieux points out that Arcangel's *The Making of Super Mario Clouds* instructional video is meticulously staged with affectations that mirror the artist's persona.

In the video, glitch-like patterns of magnetic tape interference roll down the screen and in some sections, an intentionally blurry camera masks important steps in the artist's ROM hacking process. To LeMieux, Arcangel's documentation foregrounds conceptual techniques of erasure within histories of the avant-garde much like Robert Rauschenberg's erasure of a drawing by Willem de Kooning in 1953. Yet, LeMieux's examination of Arcangel's source code, when compared to the original *SMB* code, reveals that Arcangel's mod is an entirely different entity. It bears no relation to the game's original rendering, palettes, or tile patterns. LeMieux even goes as far as to brilliantly recreate *SMC* using Arcangel's tutorials to produce a mod in which everything but the clouds are actually erased, only to fall short with the inability to remove one

¹⁴ Arcangel, "The Making of Super Mario Clouds," 107,109, 114.

¹⁵ Patrick LeMieux. *Everything but the Clouds*. Directed by Patrick LeMieux. 2017. New York: Platform Games at Babycastles, Digital Video, <https://vimeo.com/241966869>.

small gold coin within the frame. As LeMieux points out, by hand assembling an entirely new series of clouds instead of appropriating the readymade status of the original tile patterns and palettes in *SMB*, Arcangel merely uses the representation of the *SMB* clouds as a pop icon within a mass entertainment, commodity aesthetic.¹⁶ Arcangel's clouds, through their continual scrolling, become analogous to Warhol's grids of Campbell soup cans in that repetition and seriality transform the familiar commodity icon into abstract forms. If the erasure of a work in essence always possesses some trace of the original, then Arcangel's scrolling clouds playfully question erasure and appropriation as conceptual practices by having us question if, in fact, a palimpsestual residue of the CHR-ROM remains on screen (Figure 1.3).

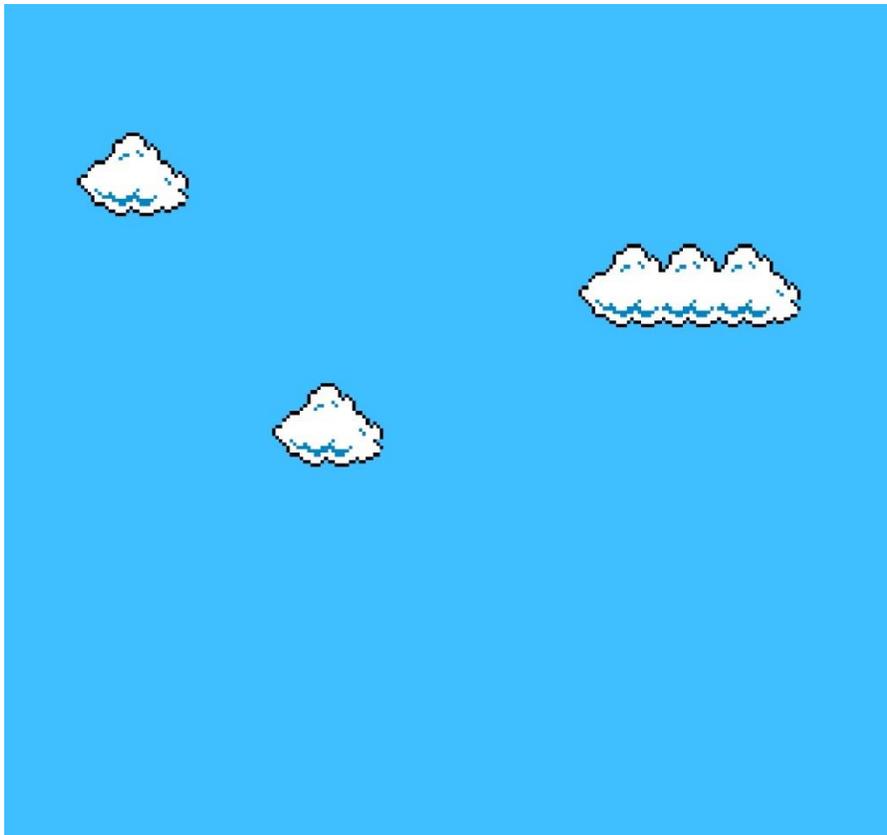


Figure 1.3: Continually-scrolling clouds in *Super Mario Clouds* (Cory Arcangel, 2002).

¹⁶ Patrick LeMieux. *Everything but the Clouds*.

LeMieux's analysis does significant work to question the status of *SMC* as a readymade object and the ethics and authenticity of Arcangel's ROM hacking process. My concern isn't so much about the fidelity of Arcangel's clouds, but rather the way in which he employs the generative properties of the *SMB* engine and NES hardware within an aesthetic economy of production. In an interview with John Bruneau, Arcangel discusses this economy of NES modding: "I like these systems not cause of nostalgia, but because they are cheap and easy to work. Also they are the perfect middle ground between analogue and digital video."¹⁷ Similarly, in an interview with curator Eryk Salvaggio, Arcangel elaborates on the aesthetic affordances of the *SMB* engine and 8-bit platforms:

...at this point most of my interest in the game is in the 16X16 sidescrolling graphical limitations...I don't love the games so much, but I really love the systems. I love the look that the old Systems have. For one, the NES directly accesses the TV's colors. So you tend to get these really bright colors on older TV's, much brighter than cable, or VHS tapes. This kinda thing also looks good even as a glow cast on a room. Like my cloud cartridge was really made for a TV I have in the corner of my apt, so at night it makes my living room glow that slurpy blue of the sky in the game. Second, I really like the idea of scrolling. Even more than movies, games tend to frame narratives in such a way that people really believe that the game world extends beyond the borders of the screen. All you have to do is move the joypad left or right to see it.¹⁸

Here, I would argue that Arcangel's hacking of the *SMB* cartridge points to this unique economy of modding predicated on NES engines and the system's split architecture. As LeMieux tells us, Arcangel's clouds *are not the SMB clouds*. They are something entirely different that relay to the viewer a pop superficiality reminiscent of the original. Instead, Arcangel employs the technics of the NES to question the body's normal relationship to the 8-bit scrolling platformer. The NES operates as a platform to generate new forms of sensual engagement by at once defamiliarizing our perception of in-game icons through an affirmation of

¹⁷ Cory Arcangel, interview by John Bruneau, "John Bruneau's Interview with Cory Arcangel vs The Stefanix of SWITCH," <http://artfail.com/arcangel/>.

¹⁸ Cory Arcangel, interview by Eryk Salvaggio, "Cory Arcangel Doesn't Even like Super Mario Brothers," April 2003, <http://turbulence.org/Works/curators/salvaggio/arcangel.html>.

their difference, while also destabilizing our movement through 2D space by abandoning a functional control interface. In the following section, I look at how the technical operations of Arcangel's *SMC* and *SMM* mods engage the body's sense of gamic apperception by revealing the speculative phenomenology of the NES architecture.

Speculative Ontologies of an 8-bit World

The body's aesthetic investment in Arcangel's *SMB* mods is a result of our interaction with the speculative reality of an 8-bit world, or what I refer to as the unit operations within the cartridge and system architecture which disclose the vital energies of the mod's abstract forms. To consider aesthetics as a process through which we come to interact with the non-anthropocentric experience of the NES system, I draw upon the metaphysics of early 20th century philosopher Alfred North Whitehead and recent conversations in object-oriented ontology and speculative realism stemming from his work. Whitehead's philosophy is grounded in lines of inquiry that push back against anthropocentrism as the assumed epistemological framework for knowing the world within contemporary traditions of Western thought. Whitehead suggests that we must exceed what he calls the bifurcation of nature: the sensory modalities through which we understand the phenomenal world revealed to us and the unseen material reality of atoms that undergird worldly appearances.

Instead, Whitehead abandons this binary in favor for a metaphysics based upon a neutral ontology among processes in the world. As Whitehead argues, we must consider a richer, metaphysical explanation of the world that privileges, for instance, "the red glow of the sunset" as well as the "molecules and electric waves" that underlie our experience of a sunset as possessing the same ontological framework.¹⁹ Thus, the material world is not constructed of "things" but rather of unique processes, differentiations, and variations. The *being* of an object or

¹⁹ Alfred North Whitehead, *The Concept of Nature* (Amherst, N.Y.: Prometheus Books, 1920/2004), 29-32.

entity is a product of its continual becoming. This mode of thought works across the bifurcation of nature in that the worldly sensations a body experiences exist along the same ontological plane as *what* we experience. Furthermore, the body is not reduced to a subject apprehending objects in the world separated from us, but rather subject and object hold the same ontological status in that both are articulated through their becoming; the processes that shape a thing—human, NES, silicon, scrolling cloud—into being.²⁰

Within this neutral ontology, Whitehead proposes that we must decenter human knowledge and experience things removed from a preconceived epistemology of the world. Instead, the processes of things experienced by us are taken up within a “common world” that although may incorporate epistemology, also transcends the boundaries of knowledge. As Whitehead tells us, the body does not come to apprehend things in the world separated from itself. Instead, the body senses and perceives an interrelatedness between things that moves outside human knowledge and our rational ability to understand those things.²¹ Our dynamic co-habitation with entities—atoms and molecules—consists of individual processes of becoming existing separately from others, yet are simultaneously composed of “drops of experience, complex and interdependent.”²² It is Whitehead’s speculative phenomenology that allows us to think of aesthetics as the interrelatedness among the body and technical components of modding; or rather, the ways in which interactions and “drops of experience” among body, pixel, 6502 processor, PPU, cartridge, and the hue of the blue sky, affirm new modes of human (and non-human) embodiment, difference, and the new.

Contemporary theorists of speculative realism such as Steven Shaviro, Jane Bennett, and Ian Bogost as well as Levi Bryant and Graham Harman (particularly with their development of

²⁰ Alfred North Whitehead, *Process and Reality* (New York: Free Press, 1929/1978), 22-24, 56-58

²¹ Alfred North Whitehead, *Science and the Modern World* (New York: Free Press, 1925/1967), 87-90.

²² Alfred North Whitehead, *Process and Reality*, 18.

object-oriented ontology) have revived Whitehead's phenomenology, especially in the wake of new materialism and the affective turn. Although these theorists may disagree with particular areas of Whitehead's work, they share with him a common philosophical inquiry into the metaphysical nature of objects and their ontological presence. Relating to the aesthetic work of Arcangel's *SMC* and *SMM*, I highlight Ian Bogost's speculative philosophy to consider the ways in which these games modulate the body's sense of gamic apperception, or how they draw our attention to the strange ontological immediacy of the *SMB* engine and NES split architecture. The manner in which Arcangel defamiliarizes our experience of movement and space within the 8-bit platformer, as I argue, is approached through an aesthetic of glitch that foregrounds a speculative phenomenology of the NES engine and its material operations. Arcangel's mods engage the body's sense of gamic apperception by laying bare the vitalist energies and relationality of the NES' working components, revealing to us modes of habitual interaction among body and the 8-bit side-scrolling video game.

Similar to Whitehead, Bogost's aim is to deterritorialize the long-held philosophical attention given to anthropocentrism and consider the vital being of everyday things—rock formations, computer hardware, chili peppers, a black hole—situated on an equal plane of material existence with humanity, or what he refers to as *alien phenomenology*. This speculative ontology points to the way things might cohabitate and experience each other within a vast ecosystem of being removed from human understanding.²³ Drawing from the work of Graham Harman and Levi Bryant, Bogost develops a phenomenology of things from the problem of *correlationism*, a term coined by Quentin Meillassoux indicating that things exist only in their

²³ Ian Bogost, *Alien phenomenology, or what it's like to be a thing* (Minneapolis, MN: University of Minnesota Press, 2012), 34.

relationship to humans.²⁴ Bogost’s alien phenomenology draws from an outright abandonment of correlationism. Instead, alien phenomenology does away with human access to things and unveils the ontological complexity among objects from which human epistemology becomes decentered. To Bogost, anticorrelationist thought can be approached through a metaphysics of *ontography*: a strategy of revealing the interobjectivity of things and their unit operations without classification or definition. Bogost tells us that, “Like a medieval bestiary, ontography can take the form of a compendium, a record of things juxtaposed to demonstrate their overlap and imply interaction through collocation.”²⁵

Within an alien phenomenology, the strangeness and incomprehensibility of objects is unveiled by their ontographic relationships, but also by what Bogost calls *metaphorism*, or the ways in which the vitality of objects is understood through analogies to other things. Thus, performing an alien phenomenology through anticorrelationist thought amounts to using anthropomorphism as a way to imagine an object’s material actions, offering a corrective to a human epistemology of how objects are perceived in the world.²⁶ Here, I want to frame aesthetic engagement with Arcangel’s *SMB* mods through an alien phenomenology by at once offering an ontographic mapping of the NES architecture as well as describing its system operations through the strangeness of metaphor. Aesthetics is not merely the ways in which the body experiences Arcangel’s floating clouds as a subject perceiving a console and screen, but rather it is the reciprocation of unit operations among things: PRG-ROM, eye, silicon, PPU, “cloud.hex,” 28-pin socket, 14-bit address bus, kilobyte, metatiles, palette, CHR-ROM, white, front-loading cartridge socket, sprite, pattern, control deck, reset button, 72-pin cartridge, gallery, pixel,

²⁴ Quentin Meillassoux, *After finitude: an essay on the necessity of contingency*, trans. Ray Brassier (London: Bloomsbury Academic, 2017), 5.

²⁵ Ian Bogost, *Alien phenomenology*, 38.

²⁶ Ian Bogost, *Alien phenomenology*, 64.

engine, slot, plastic, blue, cloud, and so on. Through an alien phenomenology we can also think of defamiliarization in the ways that Arcangel's aesthetic process and documentation show us the vital being of the NES through, as I argue, an anthropomorphic metaphorism. In other words, the SMB engine, MOS 6502 processor, and PPU all possess a unique mode of object perception that *sees* the gameworld from different vantage points.

When the body perceives Arcangel's scrolling clouds in the gallery, our affective process of apperception intuits the inner workings of the NES platform. We inquire into the potential ways an antiquated game system functions as an *engine*, and framework for developing abstract forms which Arcangel foregrounds through the inclusion of DIY tutorials. Yet, within the split architecture of the NES, its constituent parts see the moving clouds in multifarious ways.

Metaphorically, we can think of the modified PRG-ROM—the “cloud.hex” data—as a vessel which stores a series of commands that dictate the rules by which *SMC* operates. It is an inscribed text or tablet of an ancient language and rule of law delivered to the CPU from upon high. The CPU is the system's brain, interpreter, and high priestess, who quickly carries out this rule of law, sending serial data to the other hardware components, her various delegates.

Similarly, when the cartridge is inserted, the CHR-ROM operates as a storehouse filled with relics and offerings. These artifacts are sent to an entirely separate room of the console or temple, where the PPU chip, our resident shaman and conjurer, arranges and blesses the relics through various incantations to generate a video signal for display, or spirit visions. The CPU-high priestess can give orders to the PPU-shaman from his individual sanctum and demand that he arrange these artifacts in certain ways, however, both of our priests work together to summon a phantasmagoria of spirits: moving sights and sounds projected into the sky. Clearly, in this performance of an alien phenomenology, I analogize the NES to the organization within a

religious temple as a way to consider the various object perceptions of its parts. My aim here is to examine how Arcangel's use of the NES to generate forms separated from *SMB*'s source content, and his unveiling of the forensic environments within the system, open up our aesthetic experience to the speculative realities of its working components. That is to say, each part of the NES engine and hardware abides by its own technical duties yet exists within a web of complex and continuously changing interobjectivity that includes our embodied perception. I now look at how Arcangel's *SMM* mod articulates this alien phenomenology and makes strange our habitual interaction with 8-bit platforming through an aesthetic of glitch.

The Vitality of Glitch

In the context of Arcangel's *SMB* mods, I define glitch within a more inclusive media landscape as an aesthetic of error, or when a media artifact enters a non-normative state of disruption. Recent conversations in glitch art often revolve around issues of purity. Certain communities of media artists tend to see a glitch as an unpredictable rupture in a system's operating procedures which generate the novel and unexpected. Anyone who regularly uses computers or plays video games, has at one time or another witnessed the spontaneous generation of a glitch. For example, a glitch could describe when the system crashes or freezes, momentarily suspending functionality and presenting us with a series of formal abstractions hidden underneath the interface. Some may consider this encounter through what Curt Cloninger calls a "wild" glitch within its pure form, and artists attempt to capture a trace of these fleeting noise artifacts, as a particular technique within this broader aesthetic.²⁷ On the other hand, artists such as Arcangel, JODI, and Julian Oliver among others, produce glitches by establishing constraints in software code to execute a possible range of glitch patterns. This intentional form of glitch is what Rosa Menkman refers to as a "domesticated" glitch, or when a noise artifact is

²⁷ Curt Cloninger, *One Per Year* (Brescia: Link Editions, 2014), 77.

produced through a computational framework or tool aided by human intervention.²⁸ A “domesticated” glitch can be produced using certain presets and filters in Photoshop or in the case of *SMC* and *SMM*, by writing code that generates time-based glitch animations. Alexander Galloway reinforces this broader definition of glitch art in his discussion of material actions inherent to both system and player during gameplay. He calls one area of this player-system interaction “non-diegetic machine acts,” or operations carried out by the system that are essential to gameplay but not included within the narrative world of the game. These acts are composed of loading times, power-ups, statistics as well as events such as crashes and glitches. Galloway refers to the latter of these machine acts as “disabling acts”: an instance when the system ceases to function properly and affects the gameworld through destructive processes.²⁹ We can consider these glitches that occur spontaneously from within the system as “wild.” Yet, Galloway also discusses the recontextualization of disabling acts through avant-garde tactics of “counter-gaming” in which artists subvert the flow of gameplay through modifications that generate “domesticated” glitches.³⁰

Regardless of the purity of glitch art and how it comes into being, I consider glitch artifacts within a more inclusive aesthetic field in which we can conceive glitch through a multiplicity of techniques and practices. Menkman aptly reinforces this view of glitch art by describing it as “a procedural activity demonstrating against and within multiple technologies,” or what she calls *critical transmedia aesthetics*. This aesthetic unveils a medium in a state of decay and ruin, unfamiliar and neglected, whether accidental or intentional. It fundamentally disorders the way in which consumers normally experience media technologies, opening up our

²⁸ Rosa Menkman, “Glitch Studies Manifesto,” in *Video Vortex Reader II: Moving Images beyond YouTube*, ed. by Geert Lovink and Rachel Somers (Amsterdam: Institute of Network Cultures, 2011), 342.

²⁹ Alexander Galloway, “Gamic Action, Four Moments,” *Gaming: Essays on Algorithmic Culture* (Minneapolis: University of Minnesota Press, 2006), 28.

³⁰ Galloway, “Counter-gaming,” *Gaming: Essays on Algorithmic Culture*, 107-109.

phenomenal engagement to the medium as something-other than what it is. At the same time, this aesthetic pushes back critically against the ways a user perceives forms of genre, interface, and their assumptions about a media object's functionality.³¹ As Menkman tells us, when media enter this critical state as the result of errors in feedback or processes of encoding/decoding, it opens up the potential for what she calls a "destructive generativity" and the subsequent creation of unexpected media forms and new modes of sensory experience.³² An apt and concise understanding of glitch aesthetics is perhaps best summed up by Olga Goriunova and Alexei Shulgin in suggesting that it+ describes "when *something obviously goes wrong*." Glitch is a singular event that exposes the materiality of a medium and "the ghostly conventionality of the forms by which digital spaces are organized."³³

As I believe, within this broader aesthetic field, the forms of glitch that Arcangel produces using the NES engine and hardware reveal the strange phenomenology of the system's components through an embodied operation of mimesis. In Arcangel's *SMC*, glitch engages the body's affective capacities through apperception and the uncanniness that results when our perceptive faculties become aware of the game's non-normative state. By using the *SMB* engine and NES as a generative site for animated forms, Arcangel's scrolling clouds position the body at the phenomenological moment in which "something obviously goes wrong." The absence of Mario and the standard vista of the Mushroom Kingdom as well as the inability to control him is in itself a manner of glitch. *SMB* is placed into an unfamiliar state except for the iconic, albeit "home-brewed" (to borrow LeMieux's phrasing) clouds and blue sky. Ultimately, the graphic restraint of Arcangel's knockoff brand clouds forces an inquiry into the aesthetic process of

³¹ Rosa Menkman, "Glitch Studies Manifesto," 344.

³² Rosa Menkman, "Glitch Studies Manifesto," 339,341.

³³ Olga Goriunova and Alexei Shulgin, "Glitch," in *Software Studies: A Lexicon*, ed. by Matthew Fuller, (Cambridge, MA: MIT Press, 2008), 110, 114.

modding and the operations at work within the system as well as the vitality of the forms those operations produce. Arcangel's clouds and absence of the Mushroom Kingdom impel us to consider object relations at work, or rather the mechanisms within the NES that generate the scrolling clouds as unique and interrelated "drops of experience."

Likewise, Arcangel's *SMM* shows an alien phenomenology of the NES, *SMB* engine, and the abstract forms they produce, experienced intensively in the body. Arcangel produced *SMM* as a semi-narrative video work in 2005 with the art collaborative Paper Rad, known for their colorful, lo-fi zines, comics, and video art as well as their affiliation with punk and "retro-tech" subcultures of the 1990s. In addition to writing an animation engine using the same ROM hacking techniques in *SMC*, Arcangel programmed a music engine that employed the 230A APU (audio processing unit) to produce an original soundtrack accompanying the roughly 15:00 minutes of animated sequences in *SMM*. Using this music engine, Jacob Ciocci of Paper Rad composed an 8-bit techno "rave" soundtrack that utilized all of the notes from the original *SMB* engine. Arcangel programmed the stroboscopic visuals in *SMM* using the original palettes and tiles stored in the PPU (or so he claims).³⁴ Self-reflexive of the video game medium, the animation follows a story of the iconic Mario who in an existential moment of despair, embarks on an adventure across a rapidly deteriorating gameworld. As Mario's reality begins to fall apart, the internal operations of the system "bleed" through in the form of abstract glitch patterns that aggressively flicker and pulsate (Figure 1.4).

As I argue, this glitch aesthetic that Arcangel employs elicits a mode of bodily apperception by showing us the speculative world of the NES system. Through the stroboscopic flashes of colorful grid patterns and rows of alphanumeric values, *SMM* conveys not only the

³⁴ Cory Arcangel, *Artist Cory Arcangel discusses Super Mario Movie*, performed by Cory Arcangel (2009; Washington D.C.: Smithsonian American Art Museum, 2009), <https://www.youtube.com/watch?v=oDDCoa71-yY>.

figural disintegration of the diegetic space that comprises Mario's reality, but also the vitality of these glitch forms as Arcangel literally repurposes the normal operations of the system. In these phenomenological moments that equate to an 8-bit rave party, the viewer *feels* the minute ontologies of the system components at work.

In her book *The Quay Brothers: Into a Metaphysical Playroom*, Suzanne Buchan looks at the surreal and often grotesque stop-motion animations of the Quay Brothers, particularly in how

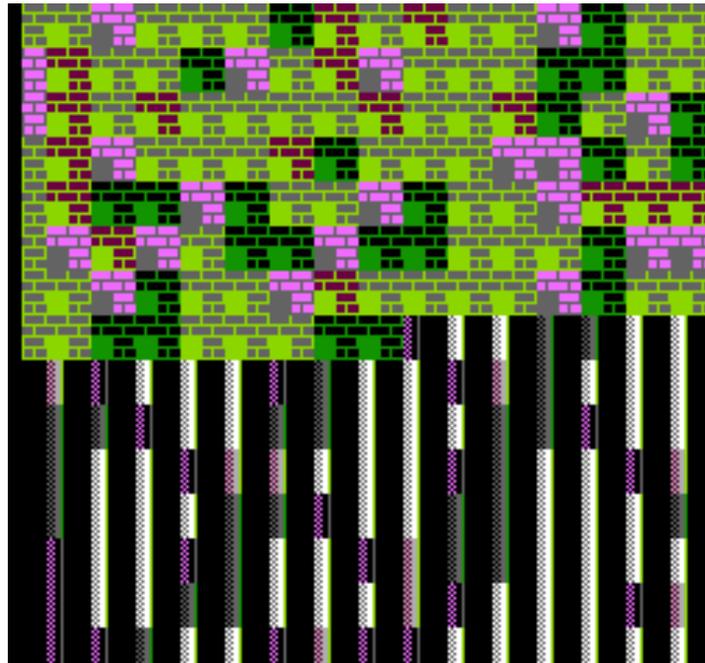


Figure 1.4: Strobing patterns of glitch in the deteriorating gameworld from *Super Mario Movie* (Cory Arcangel and Paper Rad, 2005).

their technical approaches elicit both affective pleasure and sensations of the uncanny.

Buchan argues that the animated compositions of the Quay's non-anthropomorphic things (insects, screws, meat), puppets, dolls, and fantastical machines disrupt physical laws of the natural world through what she calls a disjunctive synthesis, a concept she borrows from Deleuze and Guattari. The Quay's animations combine various inanimate materials that through their interactions demonstrate a powerful vitality. When these ordinary and extra-ordinary things—

puppets, automata, rube Goldberg-like mechanisms—both familiar and unfamiliar, are brought together, they form material instantiations of intensive pathological states. As a means to articulate the technical processes of stop-motion that bring life to these entities, Buchan employs the concept of vitalism to suggest that the spectator experiences the Quay's cinematic worlds through a force outside of animism. Their puppets, dolls, strings, scissors, and maps do not possess an anima, or soul but rather a nonhuman and non-anthropomorphic energy composed of organic and non-organic matter.³⁵

In the Quay's film *Street of Crocodiles*, a citation to a collection of short stories by Bruno Schulz of the same name, the artists use montage, lingering close-ups and macro lenses so that their puppets embody vague, metonymic fragments of literary and artistic references. The Quay's present to us sequences of visual portmanteaus that prompt a multiplicity of potential significations. Within the "psycho-topographic" spaces of the film, the miniaturized sets that possess the movements of the Quay's puppets, automata, and flesh-filled pocket watches, convey numerous paradoxes in spatial continuity. How do these pro-filmic objects relate to each other within the diegetic space of the film? How do these cluttered interiors co-extend into exterior spaces?

In *Street of Crocodiles*, the Quay's use these miniature sets to produce a spatial collage in which entities disrupt Cartesian perspective as they often do not fit a 1:1 scale. Composed of 29,500 single-frame shots and the live action sequence of the projectionist in the beginning of the film, *Street of Crocodiles* was created by exposing one still image at a time. In each new frame, the Quay's choreograph the puppet and the strange contents of their "metaphysical playrooms" at 1/24 of a second so that when played back, the carefully-arranged scene in each frame generates

³⁵ Suzanne Buchan, *The Quay Brothers: Into a Metaphysical Playroom* (Minneapolis, MN: University of Minnesota Press, 2011), xx, 33.

movement from these inanimate entities. The Quay's puppets and automata do not embody a soul or have an essence of character. Rather, through the Quay's choreography and stop-motion technique, they are placed into movement in ways that express a vitalist will (Figure 1.5).

As Buchan tells us, the viewer's aesthetic experience of this vitalist will describes a moment of apprehension, a term that parallels my understanding of gamic apperception, that being, a sense of grasping at something without "affirmation of that intellection" and that what follows is unknown. Apprehension is a dynamic of defamiliarization. It is the inability to comprehend the



Figure 1.5: Puppet and dolls animated using stop-motion techniques in *Street of Crocodiles* (Quay Brothers, 1987).

actions and consequences existing between the liminal spaces of the Quay's worlds, eliciting sensations of dissociation and the uncanny.³⁶

Just as the Quay's use stop-motion to animate a vitalist will within their puppets, Arcangel uses ROM hacking techniques to expose the viewer to the vibrancy of glitch forms generated by the NES architecture. In a particular sequence in *SMM*, Mario leaves with a small mushroom, jumping upon a floating "1UP" symbol and riding it across a violently lurching

³⁶ Buchan, *The Quay Brothers: Into a Metaphysical Playroom*, 68-70, 93.

glitch landscape. As the intertitle exclaims, Mario is joining the mushroom on a sort of vision quest, a “spirit rave” in which they travel to a skull-shaped castle housing a dance party. Before he takes off, we see the computational landscape begin to deteriorate through Arcangel’s





Figure 1.6: A sequence of glitched tile patterns resulting in Mario's flight through the deteriorating gameworld from *Super Mario Movie* (Cory Arcangel and Paper Rad, 2005).

meticulously programmed glitch patterns: flashing screens of the letters “J” “Y” and “O,” scrolling blocks of color amid a flashing background. As Mario’s “spirit rave” progresses, these scrolling visuals become increasingly chaotic and the flashing blocks begin to sporadically appear and reappear in different areas of the frame. Grids of 0s and fragments of spotted mushrooms are introduced into the pulsating tiles. These patterns eventually form large horizontal rows, resembling a tunnel, through which Mario begins to scroll through, giving the illusion of his body, the mushroom, and “1UP” platform accelerating through a kind of two-dimensional, psychedelic wormhole (Figure 1.6). His flight ends on the other side of this wormhole when he lands in front of the castle. It is this sequence in particular that illustrates the ways in which Arcangel’s orchestrated glitch patterns generate a moment of bodily apperception, revealing to us the alien phenomenology of the NES system. Through the vitalist energies of these erratic tile patterns, the body engages sensually with the system’s unit operations and abstract forms they engender: the game engine which instructs the CPU to move and scroll these palettes and titles; the PPU which displays patterns; the APU that plays the soundtrack in synchronization with the glitches. In our aesthetic engagement with *SMM*, Arcangel’s ROM hack not only gives a glimpse into the unique object perceptions of the system’s components but positions our bodily experience as a node within the interconnectedness of things in the world.

The Démodé Object, Pop Art, and the Repetition of Smiling Clouds

In addition to the ways glitch elicits a moment of the uncanny and opens up our experience to the speculative ontologies of the *NES*, Arcangel’s *SMC* and *SMM* likewise modulate the body’s proprioceptive capacities through non-play. Arcangel’s configuration of the *SMB* mods as time-based animations not played through a controller in the conventional sense attunes the body’s proprioceptive sensitivity to the side-scrolling mechanics of the 8-bit

platformer. In this strange instance of being unable to control Mario or prompt the CPU to scroll the screen on the horizontal axis, Arcangel's mods coordinate bodily affectivity with the démodé video game commodity. As a way to think about how the démodé commodity operates through a particular aesthetic vector in Arcangel's work, I draw a parallel to his own affinity with popular culture and the collage animations of Lewis Klahr. Klahr's films possess a certain dreaminess of a post-war consumer culture once lost, utilizing stop-motion animation and appropriating fragments of late 1940s Cosmopolitan magazine snippets, cut outs of old Jimmie Olsen comics, pornography, and ads for recipes, liquor, and perfumes. Through collage, Klahr's work not only deconstructs the dominant, utopic narrative of American consumerism but *reanimates*, as he refers to it, the material forms of the recent past. Like the gaming icons strewn throughout

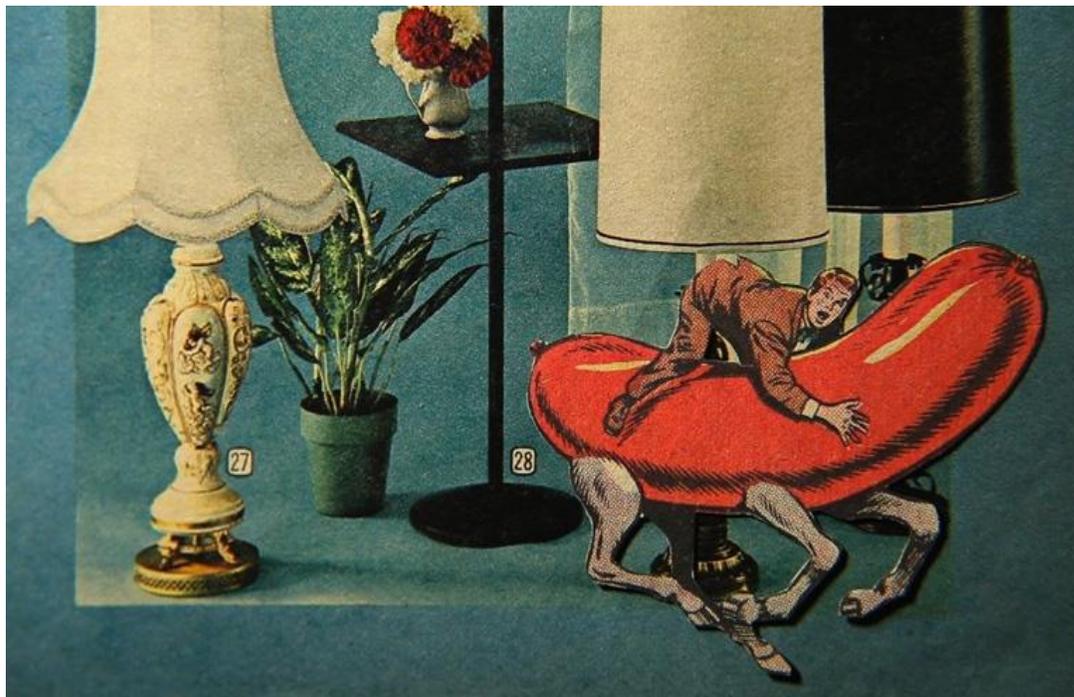


Figure 1.7: Cut out démodé commodities animated through stop-motion techniques from *Pony Glass* (Lewis Klahr, 1998).

Arcangel's mods, nostalgia associated with Klahr's remnants of pop culture open up the body to a sensuous experience of their forms: aesthetic vectors of line and shape by which we empathize

with the movement of Klahr's paper figures. In his own writings, Klahr refers to his films not as animation but as collage in which stop-motion produces a "kinetic interface between the personal and the cultural," and explores the representation of history and memory through the vantage point of the present.³⁷ Klahr's re-animations refer to a haunted return of the commodity object. Tom Gunning has discussed the "cultural interface" of Klahr's work through the early 20th century avant-garde's fascination with the *démodé* commodity, or a perception of the past that presents itself as an appropriation void of historical thickness. As Gunning describes, the *démodé* is the return of an object recently forgotten, retro and antiquated that produces a sense of the uncanny. As a once active commodity through which our desires are invested, yet now drained of its market value, the *démodé* object possesses a remainder of its libidinal energies through our nostalgia.³⁸

For instance, Klahr's *Pony Glass* (1997), a melodrama in three acts, shows Superman's friend Jimmie Olsen begin a love affair with the beautiful Lucy Lane, who breaks his heart when she leaves him for another man. Jimmy becomes overwhelmed with homosexual confusion and desire when he kisses another young man, leading to a series of erotic encounters. Jimmy begins dressing as a woman and having a new love affair with his boss and head editor at the Daily Planet Perry White, which culminates in Olsen's heightened exploration of his latent homosexuality with other male suitors. Klahr excavates the pages of old comic books, domestic interiors from home magazines, illustrations of classical statues, fragments of pornography, and food and vanity advertisements to produce Olsen's homosexual delirium (Figure 1.7). Klahr expresses the anxieties of Olsen's sexual crisis through these commodity artifacts and his material reanimations work to subvert the mass conformity of heterosexuality in the post war

³⁷ Lewis Klahr, "A Clarification," 235.

³⁸ Tom Gunning, "Lewis Klahr's *The Pharaoh's Belt* and the Amnesia of Modernity," (conference paper, *Der Blick der Moderne*, Vienna, June, 1996), 5, 7.

era. Arcangel's work operates in a similar manner to Klahr's collage films. Just as Klahr employs a commodity aesthetic of 1950s mass consumerism, Arcangel appropriates the NES as a *démodé* object of media's recent past. When Klahr's comic book figures reanimate and subsequently defamiliarize our nostalgia of 1950s culture, Arcangel uses the NES for its retro 8-bit aesthetic that taps into our libidinal desires *to play*, at once foregrounding the exterior design of the NES in the gallery, the pixelated forms generated by the system, as well as the familiarity of gaming icons that appear in his mods.

Arcangel's aesthetic of defamiliarization through the *démodé* NES is also indebted to a tradition of pop art within histories of the neo-avant-garde. There is no question that Arcangel was influenced by the work of Andy Warhol, especially in the way Warhol blurred the distinction between low-brow consumerist imagery and high art. Arcangel often positions himself as a type of "Warhol for the Information Age," especially in the way he employs repetition and the appropriation of popular forms of technoculture as modes of abstraction. His NES mod *I Shot Andy Warhol* (2002) pays homage to Warhol's fascination with the public's consumption of glamour and celebrity. Hacking an old cartridge of the NES game *Hogan's Alley* (1984), Arcangel modified the game so that players—using the NES Zapper light gun—shoot at caricatures of Warhol while avoiding those of Flavor Flav, the Pope, and Colonel Sanders. The mod itself is a direct citation to Valerie Solanas' attempted murder of Warhol in 1968, yet it also toys with the concept of fetishizing celebrity stardom, requiring the player to eliminate the Warhols—the "bad guys"—while saving the lives of Flavor Flav, the Pope, and Sanders. In his discussion of Warhol's contributions to the neo-avant-garde, Benjamin Buchloh argues that the artist's stylized paintings and serigraphs of Marilyn Monroe, Elvis Presley, and Elizabeth Taylor, retain their aesthetic appeal not through the enduring myth of those celebrities' stardom, but

rather through Warhol's construction of the star image as "the tragic condition of those who consume the stars' images in scopic cults."³⁹ As Warhol himself remarked about this tragic consumption of the celebrity image:

I made my earliest films using for several hours just one actor on the screen doing the same thing: eating or sleeping or smoking: I did this because people usually just go to the movies to see only the star, to eat him up, so here at last is a chance to look only at the star for as long as you like no matter what he does and to eat him up all you want to...⁴⁰

To Warhol, the subject's preoccupation with the celebrity image is a direct result of the repetition and seriality of the star icon. Repetition largely defined Warhol's aesthetic. The artist famously



Figure 1.8: The traumatic image of Marilyn Monroe in *Marilyn Diptych* (Andy Warhol, 1962).

described his persona and art by stating that: "I don't want it to be essentially the same—I want it to be *exactly* the same. Because the more you look at the same exact thing, the more the meaning

³⁹ Benjamin Buchloh, "Andy Warhol's One-Dimensional Art: 1956–1966," in *October Files: Andy Warhol*, ed. Annette Michelson (Cambridge: The MIT Press, 2001), 28.

⁴⁰ Andy Warhol, "Nothing to Lose," interview by Gretchen Berg, *Cahiers du Cinéma*, English ed. (May 1967): 39–43. Qtd. in Benjamin Buchloh, "Andy Warhol's One-Dimensional Art: 1956–1966", 28.

goes away, and the better and emptier you feel.”⁴¹ Here, Warhol doesn’t employ repetition solely as a means to overcome a traumatic occurrence and reintegrate it into a symbolic order as understood in Freudian psychoanalysis. Rather, as Hal Foster tells us, Warhol’s use of repetition indicates “an obsessive fixation on the object in melancholy.” To Foster, Warhol’s colored rows of Marilyn Monroe in his *Marilyn Diptych* (produced after her death in 1962) or grids of gruesome car crashes in his *Death and Disaster* series, simultaneously diminish the significance of traumatic affect while also generating an affective encounter with the traumatic image (Figure 1.8).⁴²

In his repurposing of the *NES* system and its 8-bit aesthetic as a *démodé* commodity, Arcangel employs the repetition of gaming icons in a way similar to Warhol’s grids of Marilyns and Taylors. In *SMC*, our embodied apperception of the continual scrolling of clouds points to an exhaustion of the clouds’ market significance. Yet, *these are not the Mario clouds*; they are a representational play on the original clouds that like Warhol’s Marilyns, elicit a melancholic longing for the recent past. A similar mode of repetition is at work in Arcangel and Paper Rad’s *SMM* especially in sequences where Mario falls from his “?” box and the engine scrolls patterns of smiling cloud faces on the vertical axis, or the horizontal scrolling of glitch patterns that include fragments of spotted mushrooms, cloud faces, and skulls appearing on the bottom of the “Bill Blaster” cannons from the original game (Figure 1.9). Arcangel and Paper Rad play with the obsolescence inherent to the *démodé* commodity through Warholian repetition in the narrative structure of *SMM*.

Throughout the film, Arcangel programs intertitles that narrate and provide dialogue for Mario throughout his psychedelic journey. Presented almost in the format of 8-bit haikus, the

⁴¹ Andy Warhol and Pat Hackett, *POPism: The Warhol '60s* (New York: Harcourt Brace Jovanovich, 1980), 50.

⁴² Hal Foster, *The Return of the Real: The Avant-Garde at the End of the Century* (Cambridge, MA: MIT Press, 1996), 132.

intertitles—which become progressively glitched as Mario advances through his crumbling reality—reflect upon the material ephemerality of the NES system. For example, the initial intertitle reminds us that:

AS A VIDEO GAME
GROWS OLD
ITS CONTENT AND
INTERNAL LOGIC
DETERIORATE

And later in the journey, as the gameworld falls apart even more, Mario’s spirit guide (in a garbled mess) explains that:

YOU’VE
BEENIN SOMEONES CLO-
SET FOR TWENTY YE RS
OF COURSE THE DATA
IS MELTI N GGG

In other sequences, Arcangel reflexively inserts melancholic reminders of obsolete media technologies from the 1980s. When he enters the rave party located within the skull castle, Mario bears witness to a short match of *Pong* (Atari, 1972) played between two flickering blocks composed of various tiles: 1-UP, in-game bricks, and numerical scores. Once one of the blocks misses the ball, Mario is projected on top of a giant flickering mosaic of his own face to which he exclaims “...this world is strobing me out” and to which his spirit guide replies, “...you are confronting your energy.” This strobing image of Mario’s face then begins to vertically scroll, similar to a rolling picture on an older CRT television. In this sequence, Arcangel develops perhaps the most aggressive instantiation of glitch in the film and Mario’s vision quest reaches a state of Nirvana, a moment of transcendence as his body transforms into a frenzied block of tiles. Mario becomes pure “energy”; his avatar along with the market value of “Mario” becomes



Figure 1.9: The scrolling repetition of game icons in *Super Mario Movie* (Cory Arcangel and Paper Rad, 2005).

immanent to the unit operations of the NES (Figure 1.10). As this kinetic block of tiles fades into a series of cascading letters along with bricks and flashes of color jumping wildly across the

screen, the system begins to repeat “THE DATA ITS...” followed by incomprehensible lines of letter values.

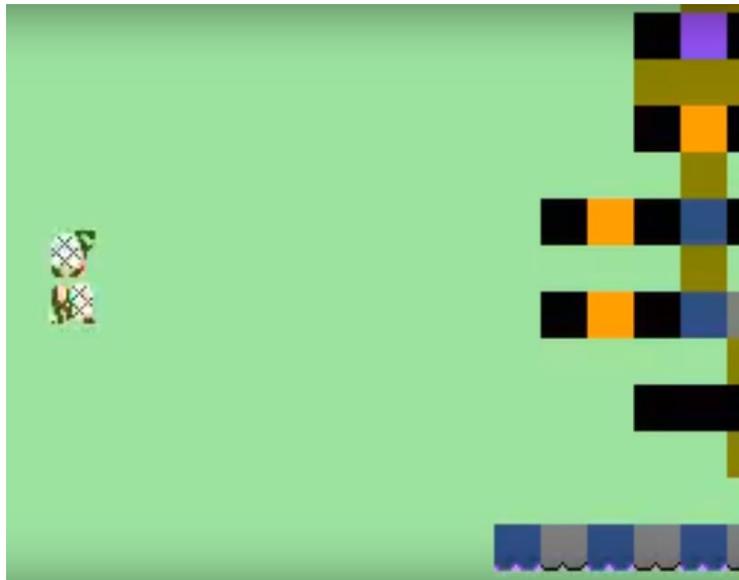


Figure 1.10: Mario transforming into the energy of the system in *Super Mario Movie* (Cory Arcangel and Paper Rad, 2005).

In this moment, Mario becomes pure data subsumed into the flows of serial information in the NES. The glitch patterns promptly transfer to a blank, white screen which gives way to the very initial frames of World 1-1 in *SMB* when the player first begins navigating Mario across the Mushroom Kingdom. Arcangel mentions that this white screen references the NES when a game crashes and that Mario’s journey in *SMM* operates as a prequel to the player booting up *SMB* indicated by the frames of World 1-1 that follow the white screen.⁴³

Yet, what do we make of this 8-bit acid trip that Arcangel and Paper Rad meticulously program? Here, I would argue that through *SMM*’s narrative, Arcangel is reflecting upon the vital materiality of the NES and its unique computational architecture that exists beyond representational modes of play. What Arcangel tell us is that although an NES may sit in someone’s closet for two decades and fall into obsolescence in the wake of more advanced

⁴³ Arcangel, *Artist Cory Arcangel discusses Super Mario Movie*.

gaming systems, it still possesses a residue of our wistful desires to reengage with the 8-bit aesthetic of *SMB*. Throughout *SMM*, Arcangel speculates as to what would occur within the gameworld after years without play. The internal rules and mechanics would dissipate, leaving our hero Mario lost to a gamestate without computational structure or constraints. Through his own playful and sardonic reflexivity, Arcangel suggests that without constraints that procedurally move him throughout the Mushroom Kingdom, Mario obtains enlightenment, becoming self-aware of his own materiality as data. Arcangel portrays the NES as a vital being with its own thoughts and memories—perhaps previous play sessions, resets, crashes—that form Mario’s dream-like journey. And just as quickly as Mario reaches a type of cosmic bliss and understanding of his own interobjectivity, the system crashes and its normal operations are restored. Perhaps the white screen isn’t so much a crash as it is a *power-on* or *reset*: a player dusting off the NES and returning Mario into the thick present of the system’s operations. When Arcangel first displayed *SMC* at the Whitney in 2002, he installed the work so that



Figure 1.11: *Super Mario Clouds* exhibited at the Whitney Museum of American Art in 2002. Photo courtesy of the artist and Team Gallery, New York.

viewers were subjected to the clouds in several modes of display: a large projection on the wall whose edges ran to the floor, a smaller projection with the same configuration, and *SMC* playing on an older CRT television played back through the NES console, haphazardly placed on the gallery floor among a maze of wires (Figure 1.11).

In 2005, Arcangel and Paper Rad showcased *SMM* at Deitch Projects in New York by projecting the film as a continuous loop within a darkened gallery. Next to the screening, they also included a series of alternate glitch patterns, or “secret scenes” projected onto a series of large blocks (Figure 1.12).⁴⁴ Here, I would argue that the perceptual conditions Arcangel creates through the exhibition of his mods, open up the viewer to affective sites through repetition, at once reproducing the traumatic affect of the NES system’s obsolescence, while at the same time generating novel affects through the body’s strange encounter with *démodé* objects. In other words, with each passing cloud or glitched Bill Blaster upon multiple screens, *SMC* and *SMM* not only articulate a bodily operation of mimesis—in which the game icon is experienced through non-play—but simultaneously drain the significance of the game icon to produce what Kirk Varnedoe has famously called of pop art: “pictures of nothing.”⁴⁵ In the following section, I examine this embodied mode of non-play as a particular vehicle through which the body’s proprioceptive capacities are instigated.

Proprioceptive Non-Play and Engines of Difference

As I have mentioned, the player’s inability to control *SMC* or *SMM*, modulates our proprioceptive sensitivity to the two-dimensional world of the NES. Arcangel’s insistence on the repetition of the deconstructed pop image leaves a kinesthetic impression upon the senses

⁴⁴ Arcangel, *Artist Cory Arcangel discusses Super Mario Movie*.

⁴⁵ Kirk Varnedoe, “Pictures of Nothing: Abstract Art since Pollock, Part 5: Satire, Irony, and Abstract Art” (lecture, National Gallery of Art, Washington D.C., May 4, 2003).



Figure 1.12: *Super Mario Movie* exhibited at Deitch Projects in 2005. Photo courtesy of Deitch Projects.

through the ways in which these abstract forms continually scroll across or down and flicker wildly upon the screen. Here, I frame non-play as an aesthetic operation that engages the body through Eugenie Shinkle’s understanding of proprioception in relation to video games, or as I refer to it, gamic proprioception. Shinkle draws from neurophysiologist Sir Charles Sherrington’s definition of the term introduced in 1906 as the body’s “six sense”: a concept taken from the Latin word *proprius* meaning “one’s own.” It refers to the sensory mechanisms dictating both conscious awareness of movement and presence, as well as unconscious processes of reflexivity, posture, motion, and balance.⁴⁶ Shinkle sees proprioception as connected to the

⁴⁶ D.I. McCloskey, “Proprioception and Related Somatosensory Mechanisms,” in *Proprioception, Posture, and Emotion*, ed. Garlick (University of New South Wales, Australia: Committee of Postgraduate Medical Education, 1982): 14-22.

affectivity of our perceptual engagement with video games, especially through the body's kinesthetic activity involved in play.⁴⁷

In Arcangel's mods, proprioceptive non-play stems from the capacities of the NES to modulate the body's perception of movement through two-dimensional space. Here, I would argue that the scrolling of abstract line and shape upon the television screen or gallery wall, mirrors kinesthetic sensations in the body generated from an inability to control the game and our perception of the system's technical generation of moving forms. According to James Ash, images possess a geographic status: they produce space and are composed of material space. An image possesses a unique spatiotemporality that exists outside the historical conditions in which it was engendered, yet in an ongoing process of becoming, the image rearticulates spatiality through the media objects that bear its existence. Especially evident in video games, our phenomenal experience of space produced through play constitutes what Ash calls an *affective materiality*, or the sensory-perceptual possibilities that emerge among player and image within the thick present, beyond discursive boundaries that dictate the game's production. Video games often emphasize the braiding of technics and bodily experience in that a player's kinesthetic actions via peripherals are reflected upon the space of the screen. Ash argues that the player's kinesthesia mapped to a controller and the negotiation of this movement upon the screen, produce multiple forms of spatiality. For one, an *existential space* emerges through bodily processes that attempt to enframe the video game and its display upon a screen, and secondly, an *ecological space* in which an "affective territory" is charted among the gamic image and exchange of bodily movement and activity on screen.⁴⁸

⁴⁷ Eugénie Shinkle, "Video games, emotion and the six senses," *Media, Culture & Society* 30, no. 6 (2008): 907, 911.

⁴⁸ James Ash, "Emerging Spatialities of the Screen: Video Games and the Reconfiguration of Spatial Awareness," *Environment and Planning A* 41, no. 9 (2009): 2106, 2108.

In the process of attuning the body's proprioceptive activity to the 8-bit platformer's side-scrolling mechanics, *SMC* and *SMM* generate the modes of spatiality mentioned by Ash. Yet what do we make of non-play in the experience of these mods and the player's inability to map their spatial movement onto image and screen? In the viewer's confrontation with Arcangel's animations, I suggest that existential and ecological spatialities emerge from an affective circuit among the unit operations of the NES, the abstract forms they produce, and the body's proprioceptive capacities to intuit lateral movement through a two-dimensional gameworld. In the time-based projections of *SMC* and *SMM*, it isn't a question of how a player's input from a controller might produce affective materiality, but rather in how our embodied perception enframes expressions of shape and line through the two-dimensional flatness of the screen. Furthermore, Arcangel has continuously experimented with the reception of these abstract forms upon different material instantiations of screens: CRT televisions atop roller carts, various-sized projections onto walls, and clusters of sculptural blocks. In his 2012 show *Masters* at the Carnegie Museum of Art, Arcangel simply displayed *SMC* on a 55" LED HDTV placed upon its original box and in a row of TVs showing other video projects, with a single chair for viewers to sit and view the passing clouds (Figure 1.13). With these forms of display, it is the *lack of* controlling Mario or prompting the CPU to scroll along the horizontal or vertical axis that generates spatiality and incites an intensive, kinesthetic feeling of space in a circuit among image, screen, and body. In this strange instance of non-play, proprioceptive sensitivity is attuned to the mod through an affective circuit among the material operations of the NES and the body. To return once more to Bergson's metaphysics and the relationship among image and embodied experience, he posits that perception via our recognition of an object creates a circuit among different circles of memory in states of expansion and contraction. Our perception of an



Figure 1.13: *Super Mario Clouds* exhibited at Arcangel's *Masters* show at the Carnegie Museum of Art in 2012. Photo courtesy of Tom Little.

image blends with memory when we mentally reach back into the past or forward into the future. On one end, this could be a superficial recollection acted out through stored motor habits, or in a deeper recollection in the form of dreams or fantasies, images relayed back and forth by the brain between perception and pure memory: the virtual construction of the past and future in the duration of the present as “a thousand repetitions of our psychical life.” Duration holds together the past and present, between memory and perception.⁴⁹ The process that occurs between perception and memory is merely a translation of material rhythms. In an ongoing process of becoming, it is the body’s durational rhythm contracting other vibrations of matter down to a

⁴⁹ Henri Bergson, *Matter and Memory*. Trans. by Nancy Margaret Paul and William Scott Palmer (London: G. Allen &, 1912), 127, 161, 186, 207.

concretized mode of perception: a hummingbird, a sunrise, the *SMB* engine scrolling along with a player's navigation.

In his philosophy of cinema, Gilles Deleuze borrows Bergson's concept of the image in relation to memory and duration to theorize cinema after World War II, particularly the ways in which it broke from conventional Hollywood editing and narrative. In his efforts to think cinema through philosophy, Deleuze sees our phenomenal engagement with film as a post-human construction, decentering the body as a privileged site of perception. Rather, he considers the body as an image among many on a plane of immanence. Deleuze sees the universe as a type of cinema: a continuous self-refracting aggregate of images that through their various assemblages, constitute the becoming and ultimately being of things in the world. In his *Cinema 2: The Time-Image*, Deleuze theorizes formal experimentations in post-war film as images that work beyond the sensory-motor schema associated with Hollywood narrative, presenting the viewer with memory images, sequences that like Bergson's circuits of memory, are virtual constructions of pure time actualized through perception in the thick present. For Deleuze, films of the Italian Neorealists or French New Wave are sequences of *time-images*, or images of pure duration such as a flashback that recalls a previous memory synthesized in the present through the medium of film.⁵⁰ Through time-images that move outside the causal, linear narratives of Classical Hollywood, film works aesthetically upon the senses through pure optical and sound sequences occurring before any action can signify them. Rather, time-images produce dream-like connections among body and image through the intermediary of the sense organs, generated from what Deleuze calls an any-space-whatever.⁵¹ In cinema, Deleuze conceives of an any-space-whatever as images of the virtual whose forces and qualities exist beyond the filmic milieu

⁵⁰ Gilles Deleuze, *Cinema 2: The Time-Image*, Trans. by Hugh Tomlinson and Robert Galeta (Minneapolis: University of Minnesota Press, 1989), 22.

⁵¹ Deleuze, *Cinema 2: The Time-Image*, 4-5.

in which they are actualized. It can be arrangements of light and shadows, colors, or a shot of an empty, demolished city street in post-war Europe.⁵²

Especially in the presence of time-images, Deleuze describes our embodied perception of film through what he calls spiritual automation, or rather the capacity for cinema to think and possess a spiritual existence beyond the thoughts humans bring to it. Just as cinema is a machine that thinks, a spiritual automaton, so is the brain and film forces thought upon the spectator at the same time that we possess the capacity of free thought ourselves. Here, Deleuze emphasizes that thought can exist independently from the human and our encounters with thought in the world, and in this instance, cinematic automata engage the body in expressions radical difference. This process of spiritual automation describes a circuit among director, film images, and viewer and through which cinema produces an affective *shock*. Images generate varying intensities which spur in us conscious or free thought in response to those images, before another sensory shock impinges upon the viewer. As Deleuze tells us, in the circuit among these coexisting automata—brain and film—both link “the highest degree of consciousness to the deepest level of the unconscious: this is the dialectical automaton.”⁵³

Through the use of *démodé* commodities, Arcangel’s mods produce sequences of time-images that engage the body’s proprioceptive sensitivity to two-dimensional movement—the scrolling and flickering of pattern tiles—through a circuit of spiritual automation. Deleuze states that:

It is only when movement becomes automatic that the artistic essence of the image is realized: *producing a shock to thought, communicating vibrations to the cortex, touching the nervous and cerebral system directly*. Because the cinematographic image itself 'makes' movement, because it makes what the other arts are restricted to demanding (or to saying), it brings together what is essential in the other arts; it inherits it, it is as it were

⁵² Gilles Deleuze, *Cinema 1: The Movement Image*, Trans. by Hugh Tomlinson and Barbara Habberjam (Minneapolis: University of Minnesota Press, 1986), 121.

⁵³ Deleuze, *Cinema 2: The Time-Image*, 161.

the directions for use of the other images, it converts into potential what was only possibility. *Automatic movement* gives rise to a *spiritual automaton* in us, which reacts in turn on movement.⁵⁴

Arcangel's homemade clouds scrolling continuously in *SMC*, project an any-space-whatever. They are an image of the virtual that takes up the viewer in an affective circuit among body and the *thinking* NES: the speculative ontologies of its working components. Our aesthetic reception of these clouds, which transform the body into a spiritual automaton, generates a sensory shock at the pre-conscious level through the two-dimensional movement of shape. Similarly, with Mario's psychedelic journey in *SMM*, the viewer experiences sequences of any-space-whatevers through abstract forms of glitch: stroboscopic patterns of shape, line, and color that give way to pure time and the machinic duration of the system. In these moments of non-play, we are subjected to the automatic movement of the NES, producing in us a felt, kinesthetic impression. In turn, our proprioceptive excitation gives rise to new thought—*our* thoughts and perceptions—which are once again taken up in a process of dialectic automation with the unit operations of the (thinking) NES.

In this dialectic exchange among body and flow of time-images, it is also the way in which our embodied perception of fragmented game icons catalyzes an affective shock when we reach back through various circuits of memory. Sifting through the cultural debris of Arcangel's mods, memory pulls from the virtual and our nostalgic remembrance of gaming culture's past, synthesizing with the perception of the *démodé* NES in the present. In this sense, the significance of Arcangel's repurposing of the NES platform to generate new modes of aesthetic experience is in the ways the iconic gaming system operates through *engines of difference*. Arcangel's reconfiguration of the NES finds resonance with the pop aesthetic of Robert Rauschenberg's combine paintings. Rauschenberg's "combines," composed of found commercial objects and

⁵⁴ Deleuze, *Cinema 2: The Time-Image*, 156.

seemingly random taches of paint, sought to challenge the viewer's reception of post-war America's commodified urban landscape through creative deconstruction. Branden Joseph argues that Rauschenberg's use of collage —appropriating commercial images from the spectacle of American consumerism and bringing them into composition with his painted surfaces—produces difference as a positive, affective force. Joseph defines the concept of difference through the philosophy of Gilles Deleuze and his criticism of representation.

For Deleuze, difference is understood in Western thought as a thing's identity, appearance, and sameness to similar things in the world. Instead, Deleuze asks us to consider difference through what he calls "difference-in-itself," an affirmation of a thing, event, or perception and the individual singularities that compose it.⁵⁵ Rauschenberg's combines open up the viewer to difference as an expression of singularities that dissociate the viewer from a concrete, habitual understanding of social reality predicated on mass commodification and conformity. As Joseph points out, Rauschenberg's work achieves this affirmative difference through the continuous multiplicities emerging from his seemingly random arrangements. In Rauschenberg's combines, difference emerges from the *instability* of collage that moves beyond any coherent, signifying complex among image and language, producing a sensory-perceptual disordering. The combines open signifying fragments to something beyond signification, difference outside of visual signifiers lacking a stable form.⁵⁶

Similarly, Arcangel employs the 8-bit engine to produce affirmative difference that exceeds representation and the NES' commercial signification as an entertainment platform. It is also these modding experiments utilizing the commercial economy of the NES that give us a

⁵⁵ See Gilles Deleuze, *Difference and Repetition*, trans. Paul Patton (New York: Columbia University Press, 1994), 55-56 in Branden Wayne Joseph, *Random Order: Robert Rauschenberg and the Neo-avant-garde* (Cambridge, MA: MIT Press, 2003), 68-69.

⁵⁶ Joseph, *Random Order: Robert Rauschenberg and the Neo-avant-garde*, 22, 157-158, 166.

more inclusive definition of what an engine is, outside of its discursive construction with the development of id Software's first-person shooters in the early 1990s. The modes of sensory engagement with Arcangel's mods at once generate the capacity for new thought, perceptions, and intensities through radical difference, while also imbricating the body in the phenomenal experience of the NES. What Arcangel shows us in *SMC* and *SMM*, is the démodé NES system—its unique components and the vital forms they produce—are *entities* in themselves. They are things that possess their own perception, existing not for our viewing, but coexisting with us in a shared experience of the world. And in this mimetic encounter with Arcangel's homemade clouds or hallucinogenic fragments of the Mushroom Kingdom, the affectivity of our nostalgic longing for a past now gone, transforms the NES and its 8-bit engines into something-other than what they are.

CHAPTER TWO

Slaying Machines: Embodied Mimesis in *ArsDoom*

In part, the discursive construction of a “game engine” emerged in the early 1990s with the programming innovations of John Carmack and the game designs of John Romero. While at id Software, their work on *Doom* (1993) fundamentally transformed the computer gaming industry by conceiving the engine as a software framework running essential operations of the game: physics, graphics rendering, AI, and audio output, among others. The *Doom* engine, also known as id tech 1, provided the technology to render novel forms of world-building and three-dimensional, algorithmic spaces through which players could openly explore. Yet perhaps more importantly, it established our knowledge of the engine as a *computational platform of content creation, yet entirelyly separate from its content*. In other words, the engine is a technology to be repurposed. The engine’s software architecture that renders 3D space can work with different assets, sprites, textures, and mechanics of play. As I have mentioned previously with Arcangel’s ROM hacking of the NES, an engine broadly describes the technical generation of moving forms, yet *Doom* introduced a singular, reusable engine running a multiplicity of games each with original content. This concept altered the epistemological and ideological underpinnings of the industry and remains embedded in the language of contemporary technoculture.

The *Doom* engine was the catalyst for new modes of embodied perception within actionable 3D environments, but also introduced novel forms of networked play, distribution via shareware, and invigorated modding communities who would alter its open-source architecture to create new gameworlds. Artists gravitated toward *Doom* because this modularity allowed for experimentation with the medium. They could replace the game’s textures, characters, and weapons with highly conceptual, abstract level designs. As I touched on in this project’s

introduction, at the 1995 Ars Electronica festival in Linz, Orhan Kipcak and Reini Urban exhibited their art modification *ArsDoom*. They used the *Doom* engine to create a simulation of the festival's Brucknerhaus exhibition hall, inviting festival artists to exhibit digital artworks and subsequently allowing patrons to interact by shooting images of the art and artists.

This chapter uses *ArsDoom* to examine the cultural fabric from which the game engine emerged and the modding practices it afforded in the early 1990s. The engine introduced a novel epistemology of aesthetic practices for world-building and animating digital forms. I argue that as an interactive form of institutional critique working within traditions of the avant-garde, *ArsDoom* produces a mimetic operation experienced as vectors of embodied vision within 3D space. By reconfiguring *Doom* as a series of bizarre expressions of play, the 3D environments of *ArsDoom* are mirrored within the body through the player's mimetic receptivity to their strangeness. Mimesis describes the feeling of *things becoming something-other than their representational status in the game*: a paintbrush-gun, splatters of blood-paint, a museum-battleground, and so on. In *ArsDoom*, running and shooting along vectors of line within the projected 3D space of the Brucknerhaus, interrogates our proprioceptive awareness by generating particular abstractions of the gamespace as varying intensities within the body. In appropriating *Doom* and transforming its familiar levels into a killing ground of the festival, Kipcak and Urban's work engages the body in gamic apperceptivity through mimetic plays of the senses. That is to say through embodied forms of mimesis, *ArsDoom* unveils the engine's modular architecture, or the ways in which it renders a bricolage of things, textures, environments, and mechanics. Yet, this modularity likewise exposes institutional power by generating playful affinities among the first-person shooter genre and the politics of the contemporary art world.

Developments in Game Engine Technology

In his discussion of game engines, Henry Lowood argues that the inception of *Doom* shouldn't be limited to a history of its invention with a specific origin as a discrete software artifact. Rather, Lowood looks at the engine through a genealogy of the software's organizational structure, one designed by Carmack so that its core operations are separate from the assets comprising the gameworld. This software architecture evolved from Carmack and Romero's work on previous 2D games at Softdisk beginning in 1990. At Softdisk, both programmers jointly developed titles such as the vertically-scrolling shooter *Slordax* (1991), utilizing the MS-DOS operating system. Improving upon this vertically-scrolling engine, Carmack and game designer Tom Hall, presented Romero with the 2D platformer *Dangerous Dave in Copyright Infringement*: a parodic knock off of *Super Mario Brothers* for the NES. With *Dangerous Dave*, Carmack proved that he could produce a horizontally-scrolling engine using MS-DOS to compete with popular 8-bit NES titles of the time. With this new engine, Carmack along with Romero, Hall, and artist Adrian Carmack left Softdisk and founded id Software in 1991, where they began developing titles for their previous employer as well as game publisher Apogee. Utilizing Carmack's innovations in seamless horizontal scrolling, id produced *Commander Keen in Invasion of the Vorticons* (1990) for MS-DOS, the first title in the *Commander Keen* series. As Carmack and Romero strategized additional *Keen* games, they decided to employ the same software code for the additional titles, which they called the *Keen* engine.

During this time, id also experimented with licensing out the *Keen* engine to other developers, allowing them to produce their own games. Although this model failed with the *Keen* engine (only Apogee bought the license to produce *Bio Menace* in 1993), the idea of selling engine architecture as a product separate from the game greatly influenced the development and

success of *Doom*. As Lowood tells us, the emergence of the game engine did not occur solely with Carmack's development of 3D graphics in *Doom*, but rather came into being through a field of technical practices surrounding the game's production. Decisions made by id to produce engine technology independent from the gameworld and with functionality that allowed developers to create multiple games sharing similar physics and graphics, ultimately helped to shape our current understanding of what an engine is.¹

As id Software continued developing more *Commander Keen* games using their reusable engine, Carmack was simultaneously exploring how to program a 3D graphics engine, one that would immerse the player into a gameworld through a point of view perspective. In *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*, David Kushner provides a rich historical account of id Software's technical developments, particularly Carmack's innovations leading up to the *Doom* graphics engine. During the 1980s, Carmack had

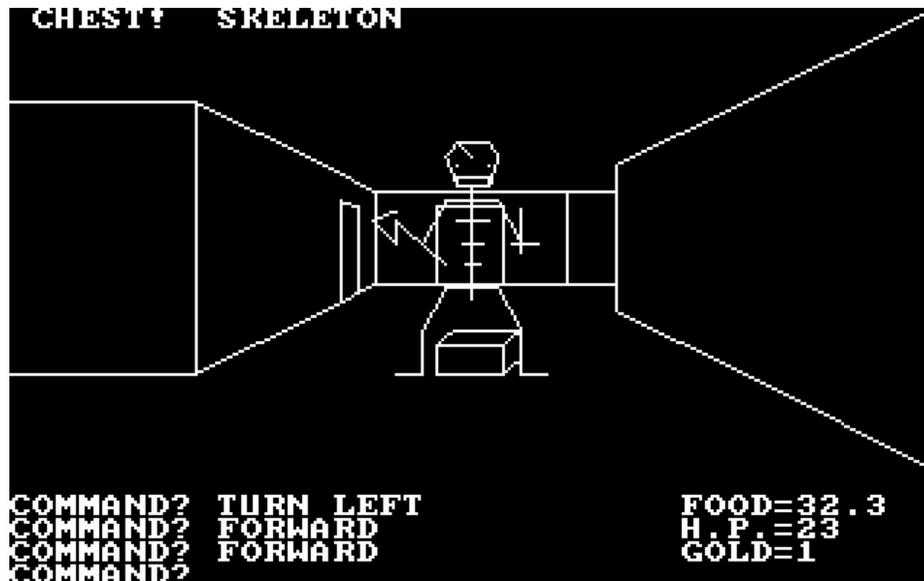


Figure 2.1: Wireframe environments from *Akalabeth* (Richard Garriott, 1980).

¹ Henry Lowood, "Game Engine," in *Debugging game history: a critical lexicon*, eds. Raiford Guins and Henry Lowood (Cambridge, MA: The MIT Press, 2016), 204-206.

experimented with computer-generated imagery by producing wireframe models of the MTV logo on an Apple II. Only a handful of computer games throughout the decade had attempted to render a 3D point of view, notably Richard Garriott's role-playing fantasy *Akalabeth* (1980) programmed on the Apple II which consisted of wireframe dungeon mazes (Figure 2.1).

In 1982, Paul Allen Edelstein at Sirius Software developed a similar first-person perspective maze game called *Wayout* for Atari's 8-bit home computers, the Apple II, and Commodore 64. Its graphics engine allowed for 360-degree player movement. There were also a number of flight simulators available that placed the viewer within the perspective of a plane's cockpit. For instance, Bruce Artwick developed a flight simulator at his software company subLOGIC and licensed it to Microsoft for IBM's PC in 1982 as *Microsoft Flight Simulator*. By 1990, Chris Roberts while at Richard Garriott's company Origin Systems programmed the wildly successful 3D space combat simulator *Wing Commander*. Carmack observed that maze games and flight simulators were slow and did not allow a player to move quickly though 3D space. He wanted to program an engine that could render polygons at a faster speed, in turn, allowing the player to navigate and engage with enemies at a pace equal to earlier 2D arcade games.² At the time, hardware limitations meant that rendering 3D graphics with numerous polygons would use considerable memory and slow down the framerate, especially in the case of overdraw, or when an engine renders the same pixels multiple times over a series of frames.

Carmack was dealing with a long-standing problem in computer graphics since the 1960s: hidden-surface algorithms, or rather the process that calculates which surfaces are occluded based on a given perspectival viewpoint. In his genealogy of computer graphics, Jacob Gaboury suggests that an ontology of simulated objects rendered by a computer is not limited to

² David S. Kushner, *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture* (New York: Random House, 2004), 68.

their representational appearance. In other words, a computer graphic is both a material image and object comprised of surface appearances as well as the unit operations which produce them. Instead, Gaboury proposes an alternate epistemology of the simulated image-object that focuses not solely on the mimetic qualities of computer graphics, but rather on visual occlusion, or those parts of the image that are not seen.³

Carmack first addressed the issue of hidden surfaces by programming an engine that would draw only a percentage of polygons within the frame, and specifically vertical trapezoids, or rather 3D spaces with walls but no ceiling or floor. He also experimented with a computational technique known as raycasting which reduced memory and processing requirements by tracing only “rays” of pixel data from the player’s perspective to surfaces in the gameworld. The engine draws individual vertical columns of pixels and applies textures based on the distance from player to the in-game surface. Essentially the process allowed for a 3D perspective to be applied to a series of 2D walls. To showcase the engine’s ray casting ability, Carmack and Hall designed *Hovortank* (1991): a fast-paced vehicle combat game in which a player gunned down monsters in the wake of a nuclear apocalypse. As Kushner argues, *Hovortank* was a landmark release in the computer gaming industry because it helped establish the first-person shooter (FPS) genre through a combination of quick gameplay and an immersive point of view camera.⁴ Carmack also experimented with a modified engine that could perform texture mapping or drawing pixel patterns on surfaces to produce detailed environments.

By the end of 1991, Carmack had developed an engine for id Software’s release of the fantasy first-person shooter *Catacomb 3-D* for MS-DOS, which utilized texture mapping for its slime-covered dungeon walls in a vibrant 16-bit EGA graphics display. As an important example

³ Jacob Gaboury, "Hidden Surface Problems: On the Digital Image as Material Object," *Journal of Visual Culture* 14, no. 1 (2015): 44-45.

⁴ Kushner, *Masters of Doom*, 68.

of a character-driven game in the genre, *Catacomb 3-D* featured the spell-casting hand of a wizard in first-person perspective, preceding the common trope of a character's gun-wielding hand which would appear later in *Wolfenstein* and *Doom* (Figure 2.2).



Figure 2.2: The rendering of first-person perspective in *Catacomb 3-D* (id Software, 1991).



Figure 2.3: Fast-paced navigation through rendered hallways in *Wolfenstein 3D* (id Software, 1992).

Carmack completed the *Catacomb* engine after witnessing texture mapping in Paul Neurath's 3D dungeon RPG *Ultima Underworld* developed at Blue Sky Productions and published the following year by Garriott's Origin Systems. He improved upon the *Catacomb* engine's existing code to produce *Wolfenstein 3D*, a game based upon a concept by Hall and Romero to reimagine the earlier Apple II title *Castle Wolfenstein* developed by Silas Warner in 1981.⁵ In 1992, id Software released *Wolfenstein 3D* and distributed its episodes as shareware. The controversial game allowed players to quickly move throughout the labyrinthine hallways of a Nazi castle, gunning down German soldiers and attack dogs while gathering spoils of war (Figure 2.3).

It immediately gained a cult following among gaming communities not only because it immersed players into a new kind of exploratory world—one in which they could open doors and discover hidden rooms—but also introduced an unprecedented level of violence and gore that had not yet been witnessed in a PC game. Carmack sped up the engine by rendering in-game objects as flat 2D sprites and upgraded the 256-color display to the newly developed VGA (Video Graphics Adapter).⁶ By 1993, Carmack had written a game engine and licensed the technology to Raven Software for the development of the first-person fantasy RPG *Shadowcaster*. The *Shadowcaster* engine included light values that would change dependent upon player perspective as well as textured ceilings, floors, and doors.

When *Doom* was first uploaded in December of 1993 by Jay Wilbur at id Software to an FTP (File Transfer Protocol) server at the University of Wisconsin-Madison, its cultural impact was unprecedented and its graphics engine, shareware distribution model, networked play, and open-source architecture would spawn multiple gaming subcultures. Initial efforts to upload the game caused the Wisconsin server to crash due to thousands of gamers attempting to download

⁵ Kushner, *Masters of Doom*, 75-79.

⁶ Kushner, *Masters of Doom*, 81-82.

simultaneously. Id released *Doom* for MS-DOS as shareware with the aim that gamers would openly distribute the game themselves. Shareware distribution avoided advertising and retail expenses and employed gamers to essentially market the game at a grassroots level by sharing it across multiple networks. The game sold roughly one million registered copies upon initial release and approximately nine million players downloaded the shareware version. Playing *Doom*, especially multiplayer deathmatch, became both an addictive cultural pastime and hindrance to workplace productivity. Employees at major tech companies like Intel and students at universities such as Carnegie Mellon slowed networks with increased traffic from deathmatch play, resulting in policies that banned *Doom*.⁷ At Microsoft, the game's popularity made such an impact that by 1995 the company developed a port of *Doom* known as *Doom95* to advertise its Windows 95 operating system as a legitimate PC gaming platform along with DirectX technology for rendering 3D graphics. Most importantly, the open-source format of the engine allowed players to create and share their own maps and in-game items using WAD files, a development I will touch on in the following section.

***Doom* and Epistemologies of Modding**

Throughout 1993, the development of *Doom*'s gameplay was largely situated around the technical capabilities and speed of Carmack's engine. In other words, id Software would need to structure an action-based narrative and quick gameplay around the engine's ability to efficiently draw multiple polygons for each frame. Carmack and Romero both held a particular fascination with demons, especially from late night sessions playing *Dungeons and Dragons* with Hall and Adrian Carmack. To mirror the advanced technics of Carmack's engine, they wanted a game that expressed a struggle of high technology against the supernatural, and thus decided loosely upon gameplay in which the player uses advanced weaponry to combat the forces of hell. The team

⁷ Kushner, *Masters of Doom*, 123-130.

conceptualized a story in which a space marine (often known as Doomguy) must destroy encroaching spawns of hell—cyberdemons, imps, zombies, and lost souls—within a series of Martian installations owned by the Union Aerospace Corporation. Naturally, the object of *Doom* would be to “run and gun”: eliminate all enemies using weapons such as a shotgun, chainsaw, rocket launcher, and the BFG (Big Fucking Gun), and escape each map by obtaining keycards to open locked areas.

As game developer Jean-Paul LeBreton notes, *Doom*'s “run and gun” design harks back to the mechanics of early 2D arcade shooters such as *Berzerk* (1980) and *Robotron: 2084* (1982). The modern FPS is predicated on the player receiving damage often from unavoidable enemy fire and then finding an area to heal via medical kits, armor, or recharging stations. Instead, *Doom*'s play operates through the mantra of “maneuverability as defense.” Because the engine allows the player to move swiftly and in multiple directions within 3D space, they can dodge hordes of enemies without losing damage (Figure 2.4). Agility often determines how skillful a player is.⁸ In *Doom*, these mechanics of play are foregrounded over narrative. As Carmack famously stated about id Software's first-person shooters...“Story in a game is like a story in a porn movie; it's expected to be there, but it's not that important.” Instead, Carmack and Romero were interested in employing the engine to *build a world* and from it, engage the player in a perceptual operation of progressing through “...one contiguous world, a seamless world” with massive angular spaces and walls with different heights.⁹

The engine architecture of *Doom* is comprised of both rendering algorithms that process maps, player and enemy movement, physics, collision detection, and effects as well as a networking element allowing for up to four players to engage in multiplayer deathmatch. As I

⁸ Jean-Paul LeBreton, “Coelacanth: Lessons from Doom,” *vector poem*, February 28th, 2010, <http://vectorpoem.com/news/?p=74>.

⁹ John Carmack qtd. in Kushner, *Masters of Doom*, 105-106.



Figure 2.4. Run and gun tactics within the 3D gamespace of *Doom* (id Software, 1993).

mentioned, the *Doom* engine refers to the game's reusable source code. The executable components of the game were used to generate and run other popular first-person shooters and RPGs such as *Doom II* (1994), *Heretic* (1994), *Hexen* (1995), and *Strife* (1996). This concept holds true with contemporary software platforms such as the Unreal Engine, developed by Epic to produce the popular deathmatch game *Unreal Tournament* in 1998 which remains an industry standard for developing games across multiple genres.

The engine was programmed by Carmack with additional code written by Romero, Dave Taylor, and Paul Radek at id Software on NeXT computer workstations. As much as it appears to be a 3D engine, *Doom* is more akin to what is often called "2.5D." In other words, the player

cannot direct the camera to look up or down and spaces (referred to as sectors) cannot be stacked upon each other to produce multi-level structures. The player can access the in-game map mode and see a 2D schematic of their position relative to the layout of the level. Still, *Doom* was remarkable in its abilities to render an illusory, texture-mapped 3D gamespace within a technical milieu dominated by computers running 386/486 Intel processors at 25-33Mhz.¹⁰

For each level in *Doom*, the engine renders a basic structure composed of objects: linedefs, sidedefs, sectors, and things. Objects that make up the architecture of the level are connected through vertices that when joined together, form lines or linedefs. Linedefs are defined by possessing either one or two sidedefs, or “sides” which store wall textures. Groups of sidedefs compose sectors, or the polygons that scaffold each level. Sectors appear as two-dimensional areas each defined by floor and ceiling dimensions, textures, and a light value. In some sectors, Carmack was able to program dynamic lighting in the form of strobes and flashing lights.¹¹ In-game objects that populate these sectors—players, demons, power-ups, and certain obstructions—are called “things” and are organized throughout each level according to a 2D coordinate and object type. Things are not rendered in true 3D and instead are conceived as sprites, or two-dimensional bitmap images. In the art direction for *Wolfenstein*, Adrian Carmack was limited to animating an enemy’s movement for each frame, yet with *Doom*, in-game things were produced by Carmack and graphic designer Kevin Cloud through digital compositing.¹² For instance, enemies were sculpted in clay in different action sequences and recorded as individual video frames. These frames were then scanned with applied color palettes and animated using a utility created by John Carmack known as the Fuzzy Pumper Palette Shop. The pixelated fidelity of these scanned images contributed to *Doom*’s iconic first-person shooter aesthetic.

¹⁰ “Doom rendering engine,” *Doomwiki*, https://doomwiki.org/wiki/Doom_rendering_engine.

¹¹ yKushner, *Masters of Doom*, 107.

¹² Kushner, *Masters of Doom*, 110.

Perhaps the most significant advancement in *Doom*'s engine architecture was Carmack's adoption of a programming technique called Binary Space Partitioning (BSP), a process used to greater effect in *Quake* a few years later. BSP describes when geometries in the gamespace are repeatedly partitioned into sets of smaller convex, polygonal hyperplanes which are stored as data in a BSP "tree." In 3D graphics, the engine retrieves this spatial data stored in the tree and is able to render the subdivided polygons in order of front-to-back within the view frustum, or player's viewpoint within a specific location of the game. This front-to-back ordering within the tree allows for the engine to quickly render visible polygons.

At the time, Carmack was aware that programmers at Bell Labs used BSP to draw an individual 3D model by subdividing its planes into smaller areas called leaves and rendering them based on which sections were currently visible. Carmack figured he could employ BSP to render more complex areas of 3D space. He used what is called a node builder to organize all of

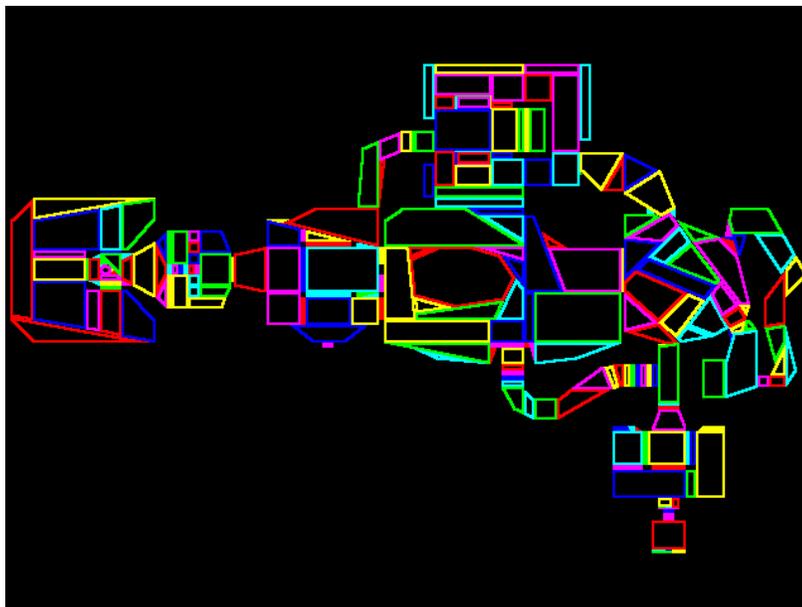


Figure 2.5: Divided subsectors and their individual segs within the E1M1: Hangar map of *Doom* from "Doom rendering engine," *Doom Wiki*, http://doom.wikia.com/wiki/Doom_rendering_engine.

the BSP data—the subdivided polygons or data leaves—for each level. The engine uses BSP to organize these data leaves into a binary tree in which each node within the tree’s structure stores a unique area of the level as individual leaves. A root node indicates polygonal data stored for the entire level and along the branches of the BSP tree, nodes are divided into smaller areas called child nodes. With the division of each node, data for linedefs are also further divided into what are called segs, or line segments. Within each leaf exists a series of convex polygons called subsectors which are assigned to a unique sector of the level. Subsectors also include a stored list of segs. At its core, the BSP algorithm places subsectors in the correct rendering order by beginning with the root node and then drawing child nodes starting with those closest to the player’s immediate point of view. The engine draws a new subsector and its segs when the player enters that area of the level. (Figure 2.5). To limit overdraw, each time segs within a subsector are drawn, they are stored in a list so that the engine does not have to redraw the same polygonal data.

Depending on the player’s camera position, the engine uses the BSP tree to draw polygonal nodes in the order of front-to-back from their line of sight based upon two distinct horizon lines.¹³ In his careful analysis of the *Doom* engine’s source code, software engineer Fabien Sanglard gives the following example of code for BSP ordering of a player observing the gamespace from a hidden balcony (Figure 2.6):

```
Player (green dot) watching from the secret balcony a point p=(5040, 2400 ):
```

```
// Player position = ( 5040, 2400 )
```

```
// R_RenderBSPNode run against AB splitter (-x + 3500 = 0):
```

```
-5040 + 3500 = -1540
```

```
Result is negative: Closest subspace is in the BACK of the splitting plane. (B is closer than A).
```

¹³ “Doom rendering engine,” *Doomwiki*, https://doomwiki.org/wiki/Doom_rendering_engine.

```
// R_RenderBSPNode now recursively run against the two child of the root node: A1/A2 splitter and B1/B2 splitter.
```

```
// R_RenderBSPNode run against B1/B2 splitter (-0.24x +0.97y - 650 = 0):
```

$$-0.24 * 5040 + 0.97 * 2400 - 650 = 468$$

Result is positive so the closest subspace is in the FRONT of the splitting plane. (B1 is closer than B2).

```
// R_RenderBSPNode run against A1/A2 splitter (x - 2500 = 0):
```

$$5040 - 2500 = 2540$$

Result is positive so the closest subspace is in the FRONT of the splitting plane. (A2 is closer than A1).

Result: Sorted zones, from near to far:

```
{ B1, B2, A2, A1 }14
```

The final result of this ordering are polygonal subspaces drawn in accordance with the player's point of view at any given time based upon the relationship between stored nodes in the BSP tree and the splitting planes within the map. As I discuss in the following chapter,

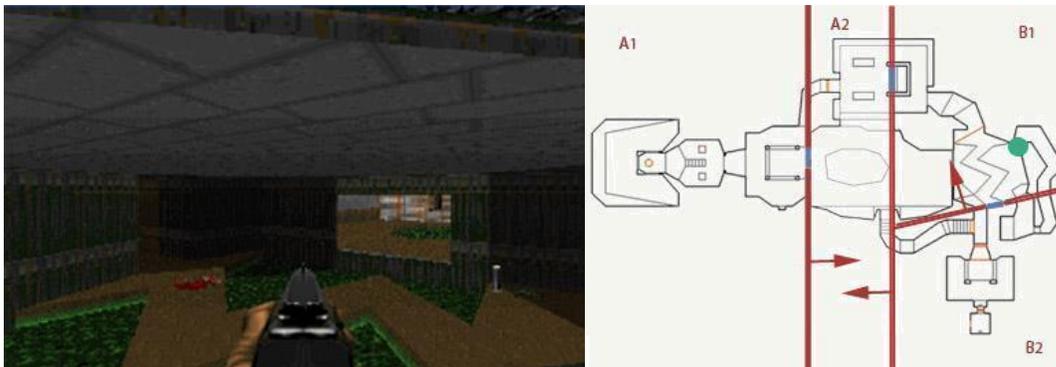


Figure 2.6: Subdivided planes via BSP ordering in the E1M1: Hangar map of *Doom* from Fabien Sanglard, “Doom Code Review,” Fabien Sanglard’s Website, January 13th 2010, <http://fabiansanglard.net/doomIphone/doomClassicRenderer.php>.

Carmack’s *Quake* engine would employ BSP to a more advanced degree to render real-time 3D space. Beyond these propriety functions of the graphics engine, Carmack designed parts of *Doom* to be modifiable by players. As Kushner suggests, by allowing loyal communities of gamers to customize and create their own weapons, skins (character appearances), and levels, id

¹⁴ Fabien Sanglard, “Doom Code Review,” Fabien Sanglard’s Website, January 13th 2010, <http://fabiansanglard.net/doomIphone/doomClassicRenderer.php>.

Software introduced a novel ideological framework; “a leftist gesture that empowered the people and, in turn, loosened the grip of corporations.”¹⁵ In other words, the *Doom* engine introduced a discursive field of practices situated around our understanding of a game modification. At the time, Carmack did not fully realize the social impact his engine would make on gaming culture. After seeing players hack the source code to produce mods for *Wolfenstein 3D*, he decided to create a file subsystem which stored customizable game data known as WADs, an acronym meaning “Where’s All the Data?” first outlined by Tom Hall in his design document the *Doom Bible*.¹⁶ When the engine boots up, it automatically searches for the WAD file and loads textures, weapons, skins, and sounds into the gameworld without affecting the proprietary source code. Carmack also shared the source code for his level-editor program, allowing amateur game designers to build their own gameworlds from the ground up.

Because each WAD file was unencrypted and easy to distribute, gamer-programmers not only wrote their own custom mods but produced a unique modding economy on the emerging Internet by sharing, uploading, and playing user-generated WADs. The creation of WADs also prompted grassroots programmers to develop new level creation and editing utilities such as Doom Editor Utility (DEU). DEU was created by University of Canterbury student Brendon Wyber and Raphael Quinet with the assistance of a crowd sourced community of programmers and gamers across the web. By early 1994, they had released a version of DEU that allowed players to modify nearly any element of a *Doom* level. Players could reposition and change the amount of in-game objects such as enemies or powerups, produce new wall textures, or add obstacles such as pools of acid.

¹⁵ Kushner, *Masters of Doom*, 134.

¹⁶ Tom Hall, “Doom Bible,” (PDF, id Software, 1992), Appendix A.

The modifiable components of *Doom* also contributed to a flourishing video game “demoscene” in the early 1990s, giving rise to the production of machinima, or utilizing an engine to record real-time animation sequences based upon player actions. The demoscene itself can be traced back to the early 1980s when communities of programmers and artists began to crack the copyright protection on games for personal computers such as the Commodore 64 and Amiga. Breaking copyright meant that hackers could generate “crack intros” also called “Cracktros.” As Michael Nitsche points out, crack intros or short animation sequences that would run before the start of a game, demonstrated the hacker’s artistic and technical prowess. Hackers would often compete to see who could program the most complex sounds and visual effects, in turn, establishing a close-knit network of digital artists.¹⁷ Obtaining access to the game’s source code also meant that hackers could directly change elements of play. These early modifications, referred to as “game trainers” allowed, for instance, players to apply cheats and jump to different levels in the game.

To Nitsche, the *Doom* engine introduced a new instantiation of the demoscene in that players could experiment with early forms of machinima production. Along with WAD files, the engine also afforded the ability to produce a demo recording as a log file, known as a lump file (.lmp) which captured all input data.¹⁸ Much like crack intros, the use of lump files prompted players to competitively produce deathmatch and speedrunning demos, or recorded sessions of how quickly an expert player, referred to as a “Doom God,” could complete a level. Players organized modes of competitive gameplay such as Frank Stajano’s *Doom* Honorific Titles (DHT) and Simon Widlake’s Compet-N: both started in 1994 as a way for players to showcase

¹⁷ Michael Nitsche, “Demo,” in *Debugging game history: a critical lexicon*, eds. Raiford Guins and Henry Lowood (Cambridge, MA: The MIT Press, 2016), 104.

¹⁸ Michael Nitsche, “Machinima as Media,” in *The Machinima Reader*, eds. Henry Lowood and Michael Nitsche (Cambridge: MIT Press, 2011), 118.

their skills. Famed speedrunners such as George Bell, Steffen Udluft, and Frank “Jesus” Siebers submitted their demo files to the Doom Honorary Titles to be credited with a status of expert play. A speedrunner could request to play a level from *Doom* or *Doom II* on a specific difficulty setting. If their submitted lump file was deemed authentic, they were rewarded titles such as “Doom Master,” “Doom Grand Master,” “Doom Tyson,” or the prestigious title of “Doom God.” Players introduced new modes of competition such as “Pacifist,” or speedrunning the “Ultra-Violence” difficulty setting without intentionally harming any monsters or “Reality,” a format invented by Benjamin “Cowboy” Lauterbach in which a player must speedrun without acquiring any damage or use the plasma rifle or BFG as these are not true to life.¹⁹

After Carmack released the proprietary *Doom* source code in December of 1997, modders were afforded the ability to create user-made port engines such as the popular *ZDoom* and Simon Howard’s *Chocolate Doom* (2005): a PC version which preserves the mechanics and aesthetics of the original while allowing open compatibility for sharing WADs. In his “people’s” history of the first-person shooter, Robert Yang explores the marginalization of modding practices throughout the 1990s in the wake of the commercial gaming industry. That is to say, gaming culture has until only recently, denied a minor history of independent, punk game design for the popular “manshooting” that has long represented the genre.²⁰ For example, in the nearly 25 years following *Doom*’s release, amateur designers have created 100s of WADs, particularly ones that infringe upon the copyright of popular Hollywood franchises such as *Batman Doom* (1999) and *Ghostbusters Doom* (2002), both total conversions (TC) of *Doom II*. Here, a total

¹⁹ Doomworld, “A Brief DOOM Demo History,” *Doomworld*, <https://www.doomworld.com/10years/demos/demos01.php>.

²⁰ Robert Yang, “A People’s History of the FPS, Part 1: The WAD,” *Rock Paper Shotgun*, September 19th, 2012, <https://www.rockpapershotgun.com/2012/09/19/a-peoples-history-of-the-fps-part-1-the-wad/>.

conversion refers to a WAD file in which a majority of the game's original environments, weapons, and assets have been replaced.

Yang points to Justin Fisher's *Aliens TC* WAD as central to this experimentation associated with early modding practices. Released in 1994, Fisher's *Aliens TC* was designed using a modding utility called DeHackEd which allowed users to access *Doom*'s executable code and overhaul the game's graphics and mechanics. Working from James Cameron's 1986 film of the same name, Fisher produced a campaign with proximity sensors, hatching alien pods, and acid blood. The initial level had no enemies or challenges and was designed solely for atmospheric effect. Another avant-garde approach to modding included the joke WAD: games that were intentionally buggy with unappealing textures and mismatched sound effects. They often introduced the player to absurd gaming scenarios or included level designs that sought to unveil *Doom*'s graphical limitations. For instance, the joke WAD *Nuts* created by B.P.R.D. in



Figure 2.7: Thousands of monsters in the joke WAD *Nuts* (B.P.R.D., 2001).

2001 consists of approximately 10,000 monsters crammed into the area of a giant courtyard. When the player attempts to kill these enemy hoards, the engine has difficulty rendering all of their movement, slowing the frame rate to a crawl and making the mod unplayable (Figure 2.7).

At a 2015 VideoBrains talk in London, modder Jazz Mickle aptly summarized the influence of the *Doom* engine in an open letter to John Romero titled “*Doom* is an Art Scene,” stating: “Do you know you created an art scene? Do you know you created an underground cultural timeline? Do you know you created an art form?”²¹ Specifically, Mickle was referring to the hand-drawn environments in the surreal fantasy WAD *Foreverhood* (2006) or the cel-animation aesthetic of *The Adventures of Square*: a standalone first-person shooter created by a twelve person team using *ZDoom*. That is to say, Mickle considers the expressive forms of animation afforded by the engine as a mode of cultural production akin to photography, television, or the printing press. In the following section, I argue that *Doom* introduced a new episteme of aesthetic techniques and modes of participation, a regime of experience in which a video game system could be used as a vehicle for worldbuilding and animating abstract forms.

The Archive of the Engine

To understand how the game engine was central to forming a new discourse of modding practices, we can look to Michel Foucault’s archaeology of painting in his writings on Edouard Manet. In *The Archaeology of Knowledge*, Foucault speculates on an archaeology of painting, stating that:

...it would try to discover whether space, distance, depth, color, light, proportions, volumes, and contours were not, at the periods in question, considered, named, enunciated, and conceptualized in a discursive practice; and whether the knowledge that this discursive practice gives rise to was not embodied perhaps in theories and

²¹ Jazz Mickle, *Doom is an Art Scene*, performed by Jazz Mickle (2015; London: VideoBrains, 2015), <https://www.youtube.com/watch?v=KxYND6K6u8w>.

speculations, in forms of teaching and codes of practices, but also in processes, techniques, and even in the very gesture of the painter.²²

Foucault's concern is not with an iconography of painting but rather with the ways it embodies certain techniques and effects within the historical conditions of its production. Joseph Tanke argues that Foucault's archaeology considers how a painting possesses a unique set of material operations. It conveys how the dispersal of aesthetic experience regularly arises within the historical fabric from which a work of art was created. Tanke asserts that the historical existence of an image lends itself to an examination of its fracture or division from previous aesthetic movements and canons, and the way it assigns relations of power among artist, institutions, and viewing audiences.²³ In Foucault's lectures given during his time in Tunis, he examines how Manet's painterly approach broke from Classical painting since the Renaissance. Foucault positions Manet as the artist by which the archive becomes apparent in that his paintings were the first to engender a regime of visibility removed from artistic representation of the Classical episteme.

In *Archaeology of Knowledge*, Foucault asserts that the archive exists as a "density of discursive practices," a system of interrelated statements that undergird the epistemological framework of a given historical order. In other words, an archive describes the material remnants of a historical period through which discourses become visible and statements transform, dissolve, or remain contingent.²⁴ In Foucault's lecture on Manet, he argues that as a collection of "painted statements" the artist's work forms a new archive. Through a profound rupture from the Classical tradition, Manet's work allows for the possibilities of painting to surpass naturalistic

²² Michel Foucault, *The Archaeology of Knowledge and the Discourse on Language*, trans. A.M. Sheridan Smith (New York: Pantheon Books, 1972), 193-194.

²³ Joseph J. Tanke, *Foucault's Philosophy of Art: A Genealogy of Modernity*, (London: Continuum International Publishing, 2009), 61-62.

²⁴ Foucault, *Archaeology of Knowledge*, 128.

representation, instead focusing on the material properties of the canvas itself. Since the Italian Renaissance, painting consisted of formal elements that sought to mask or “sidestep” the materiality of the painting and hand of the artist through the representation of an illusionistic, three-dimensional space configured onto a two-dimensional surface. The fact that a painting has a two-dimensional surface lit by real light from an outside source has been concealed by the conventions of Western painting in its effort to replicate reality onto a material surface. This illusion is typically achieved through linear perspective to construct naturalistic depth, the modeling of figures by a light source emanating from within the picture plane, and positioning the viewer so that depth unfolds from a centralized viewpoint.

For Foucault, Manet’s work foregrounds what is called the “picture-object.” This refers to the materiality of the painting that calls attention to the canvas’s limitations; or rather, color applied to a physical surface articulated by an external light and depth constructed symbolically through the relationships of objects on the canvas.²⁵ What is important for Foucault is that Manet’s formal experimentations implicate the viewer within novel discursive practices of visuality and looking as they are historically bound to modernity. They indicate the emergence of a 19th century viewing subject organized from the complex relationship among patron, artist, and intuitional power. Manet’s work is the foundation upon which viewing art within a museum rests.

My aim here is not to compare Manet’s paintings to the rendering of 3D space in a game engine as they resemble two vastly different forms of cultural production and reception. Rather, I would argue that just as Manet’s work formed the archive of modern viewing practices, Carmack’s *Doom* engine constructed the archive through which immersion within navigable 3D space and the capacities to modify such space enter into popular technoculture. Both articulate

²⁵ Michel Foucault, *Manet and the Object of Painting*, trans. Mathew Barr (London: Tate Publishing, 2009), 29, 31.

forms of visual media that function as an object of perception and organize novel ways of looking and making. By introducing the concept of a game engine as a reusable vessel for content creation, id Software was central to organizing a novel politics of aesthetic techniques, distribution, and experience. The engine afforded a new epistemological order through which a grassroots modding community flourished, producing a vibrant network of artists, programmers, and loyal gamers who experimented with new modes of art making.

The engine reshaped the technical landscape of the early 1990s, in turn, opening up new possibilities for constructing and animating vibrant 3D gameworlds through modding techniques. At the same time, the *Doom* engine played a crucial role in organizing a new mode of aesthetic reception toward digital media: a novel sensory regime in which players were immersed into detailed gameworlds through perspectively-rendered 3D space. It is helpful to think about this particular archive of the digital through what Robert Nideffer refers to as a database aesthetic. Nideffer explains that the game engine possesses a database interface through which a collection of procedural operations generate a gameworld for the player to interact with and move throughout. The engine is a cultural object used to organize game data—sprites, wall textures, maps, sound effects—for the computer to process in response to a player’s control input. Yet, because their activity is imbricated in the engine’s continual rendering of a gameworld, players and developers “inescapably function as indexical pointers to their respective cultural toolkits or databases at all phases of game-engine and game-application production, distribution, and consumption.”²⁶ Nideffer indicates that the player’s engagement with the engine’s database reciprocally produces a *database of the player*, one that organizes a novel epistemology of artistic techniques as well as perceptual and sensory operations. And, it is from this constructed

²⁶ Robert Nideffer, “Game Engines as Embedded Systems,” in *Database Aesthetics: Art in the Age of Information Overflow*, ed. Viktorija Vesna (Minneapolis: University of Minnesota Press, 2010), 219, 221.

archive of engine and player through which we can understand modding as a technique of world-building adopted by artists to defamiliarize our habitual encounters with the first-person shooter.

Carmack's engine held a particular attraction for artists because it allowed them to appropriate the look and interactivity of a popular first-person shooter as a way to subvert the game's logic. In other words, they employed a technical economy of the engine to build and animate abstract gameworlds. I would argue that one of the central affordances of the game engine's modular design is the ability to worldbuild: to render complexity and depth in a game environment while animating our perspectival movement within it. In relation to video games, worldbuilding often describes the mental processes by which a player infers the existence of a more expansive digital environment. Janet Murray describes the characteristics of interactive narrative within a constructed digital world as procedural, participatory, spatial, and encyclopedic. The modularity of the *Doom* engine allows for the construction of a gameworld that is *procedural* in that it organizes and renders data through a pipeline of complex rules and behaviors. Yet, the engine is also *participatory* in that it responds to the player's input by carrying out those procedures, or what Murray refers to as interactivity. Our interaction is also fundamentally *spatial*. It involves a perceptual process of moving through navigable space that extends into a larger gameworld and its mythos. Lastly, the engine is *encyclopedic* in that its modularity allows for the storage, retrieval, and distribution of game data through WAD files, producing a kind of digital repository for levels, textures, skins, weapons, and so on.²⁷ As I discuss in the following section, artists gravitated toward the worldbuilding capacities of the *Doom* engine to reconfigure a new politics of the body through mimetic operations of play by constructing abstract gameworlds and critiquing the hallowed ground of the art institution.

²⁷ Janet Murray, *Hamlet on the Holodeck: The Future of Narrative in Cyberspace* (Cambridge, MA: MIT Press, 1998), 71, 72, 74, 79.

***ArsDoom* and Mimetic Vectors of Play**

In the years following *Doom*'s release, artists experimented with the engine as form of institutional critique, resulting in the emergence of a subgenre within game art coined by Mathias Jansson as "First Museum Shooters."²⁸ As with Kipcak's *ArsDoom*, artists fabricated three-dimensional recreations of real exhibition spaces and used the mechanics of the engine to allow players to run throughout the museum, shoot at various monsters or other artists, whilst destroying the art. In 1995, Kipcak along with collaborators Reini Urban, Curd Duca, and XRay produced *ArsDoom* as arguably the first large-scale art mod, showcased within the Brucknerhaus at Ars Electronica in Linz, Austria. Kipcak's work opened a field of discursive practices in which game art held a political stake in attacking institutional structures of power, in turn, influencing others within the subgenre. In 1996, Swedish artists Palle Torsson and Tobias Bernstrup created Museum Meltdown, a mod utilizing the first-person shooter *Duke Nukem 3D* (1996) and its Build engine developed by Ken Silverman at 3D Realms. Torsson and Bernstrup modeled the layout of the Arken Museum of Modern Art in Copenhagen and exhibited the mod within the museum in conjunction with the *Borealis 8: The Scream* biennale. In *Museum Meltdown*, participants ridded Arken of invading aliens while desecrating the biennale's artworks using high-powered weaponry. In 1997, they produced a subsequent mod called *Museum Meltdown II: The Vilnius Vengeance* which pitted players against aliens within the reconstructed space of the Contemporary Art Centre in Vilnius, Lithuania (Figure 2.8). Torsson and Bernstrup completed the *Museum Meltdown* trilogy in 1999 with *Museum Meltdown: Moderna Museet*. Unlike its predecessors, *Moderna Museet* utilized the popular first-person shooter *Half-Life* and its Source engine to reproduce the interiors of the Museum of Modern Art in Stockholm.

²⁸ Mathias Jansson, "First Museum Shooters," in *The Pioneers of Game Art: From ArsDoom to SimBee*, ed. Mathias Jansson (Akarp, Sweden: Mathias Jansson, 2015), 8-9.

Participants were required to defend themselves against mutant creatures and opposing soldiers using the stock *Half-Life* crowbar and shotgun in which, as the artists' state, the game "...is reduced to a central perspective of violence - wherever your cross hair aims, the artworks and all living creatures are targets for your destructive gaze."²⁹ Similarly in 2005-2006, Chris Reilly modeled the gallery within the School of the Art Institute of Chicago to include detailed simulations of his fellow students' artworks for the school's 2006 Undergraduate Exhibition.



Figure 2.8: Gameplay from *Museum Meltdown II: The Vilnius Vengeance* (Palle Torsson and Tobias Bernstrup, 1997).

Using the Source engine and *Half-Life II* (1998), Reilly exhibited his mod *Everything I Do is Art, But Nothing I Do Makes Any Difference, Part II Or: How I Learned to Stop Worrying and Love the Gallery* as an installation in which participants defended themselves and in-game gallery-goers against hordes of zombies, robots, and alien creatures. Players could also earn additional

²⁹ Tobias Bernstrup and Palle Torsson, "Museum Meltdown: Moderna Museet," *bernstrup.com*, 1999, <http://www.bernstrup.com/meltdown/main.html>.

health and ammo by ruining the art using the game's gravity gun to launch paint cans at the works.

Jon Cates points out that the conceptual practices of first museum shooters are comprised of appropriating a game engine to simulate an existing museum space while recontextualizing gameplay by leaving the game's mechanics and rules intact and employing them within this constructed museum environment. Furthermore, the first museum shooter engages with issues of violence relevant to first-person shooters and reconstitutes such violence as a feature of institutional critique.³⁰ In other words, by allowing participants to destroy copies of artworks within a simulated museum, the genre pushes back against institutional power; namely, the ways in which the museum dictates aesthetic experience and knowledge in deciding what works and artists are exhibited. Through the interactivity of the genre, institutional power is decentered and rematerialized in the player as art critic who judges which works are salvaged and which get demolished among the carnage.

In order to understand the role of embodied mimesis in the player's apperceptive and proprioceptive response to *ArsDoom* and its institutional criticism, it is important to examine the media theory of Walter Benjamin. Benjamin's Marxist, materialist philosophy of media is ultimately concerned with the historically contingent manner in which technology conditions the senses. To Benjamin, aesthetics, and particularly mimesis, are tied to the reconfiguration of human perception and experience within the social and political conditions of modernity. Benjamin sees the role of technologies, namely cinema, to perceptually train the body in a

³⁰ Jon Cates, "Running and Gunning in the Gallery: Art Mods, Art Institutions and the Artists that Destroy Them," *artgamestudies.org*, <http://artgamestudies.org/RunningAndGunning.pdf>, 10.

mechanized world through what he refers to as innervation.³¹ Essentially cinema possesses the therapeutic potential to reverse the social alienation brought on by an industrial-capitalist world and a failed reception of technology. Within fascist society, technology isolates citizens from natural forms of communication and sentimentality through the mass consumption of images and an overstimulation of the senses. Instead, innervation suggests a neurophysiological channeling between the cinematic apparatus and the physic and perceptual mechanisms of the body. It describes an embodied, mimetic reception of the material world that possesses liberatory potential in which affect and desire are channeled toward utopic ends within a Marxist political ideology. As a transference of affective energies into somatic, motor responses, innervation can alter the relationship of the collective masses to the world through plays of the senses. Here, innervation is bound to Benjamin's concept of the mimetic faculty. Benjamin describes two different forms of mimesis: phylogenetic and ontogenetic. The former refers to a deep time of mimetic development through which representation and signification are organized. Benjamin relates phylogenetic mimesis to forms of language but also expressions of mysticism such as astrological signs and ritual dance. Benjamin's second, ontogenetic understanding of the term, and the tradition of mimesis I draw upon throughout this chapter, describes a modality through which innervation operates, a sensual interaction with things in the world that has the potential to disrupt conventional Bourgeois subject-object relationships. To Benjamin, this second form of mimesis is a relational process of generating similarities among humans and objects through expressions of play. It operates through sensuous qualities emerging among things and their characteristics, suggesting forms of interrelatedness.

³¹ Refer to Benjamin's use of innervation in Walter Benjamin, "The Work of Art in the Age of Its Technological Reproducibility, Second Version," in *The Work of Art in the Age of Its Technological Reproducibility and Other Writings on Media*, ed. Michael W. Jennings, Brigid Doherty, and Thomas Y. Levin. Trans. Edmund Jephcott, Rodney Livingstone, Howard Eiland, and Others. Cambridge: Belknap Press of Harvard University Press, 2008.

Benjamin ties mimesis to children and play through the ways they creatively engage with their environment. Ultimately, children's play is a mimetic behavior and not merely the imitation of particular subjectivities in the world as "the child plays at being not only a shopkeeper or teacher, but also a windmill and a train."³² The mimetic engagement of children is historically contingent and in constant variation with the transitory nature of an industrial capitalist economy through new technologies, commodities, and images. As Miriam Hansen notes, forms of play through Benjamin's mimetic faculty are a performance of an inventive, generative perception through relational similarities. Here, play resembles a type of productive "miscognition" resulting from a disjunctive synthesis among people and things.³³ For Benjamin, cinema possesses a unique aesthetic of innervation via the mimetic faculty that can be employed to utopic ends. As he says, "Just as a child who has learned to grasp stretches out its hand for the moon as it would for a ball, so humanity, in its efforts at innervation, sets its sights as much on currently utopian goals as on goals within reach."³⁴ Cinema holds the capacity to extend the human sensorium beyond the limits of Bourgeois subjectivity through the camera's mobility and otherness as well as editing together various shots of things in the material world. It shows the masses their own social and economic conditions by fragmenting the industrial world into an assemblage of disparate images, in turn, opening up a new type of collective, technically-mediated consciousness.

I propose that Benjamin's concept of mimesis lends itself particularly well to the study of video games as an interactive medium. Games afford certain material assemblages among players, controllers, systems, and gameworlds through which novel modes of playful

³² Walter Benjamin, "On the Mimetic Faculty," *Selected Writings, Volume 2: 1927–1934*. Ed. Michael W. Jennings, Howard Eiland, and Gary Smith. Trans. Rodney Livingstone and others. Cambridge: Belknap, 1999, 721-722.

³³ Miriam Hansen, *Cinema and Experience: Siegfried Kracauer, Walter Benjamin, and Theodor W. Adorno* (Berkeley, CA: University of California Press, 2012), 150.

³⁴ Walter Benjamin, "The Work of Art in the Age of Its Technological Reproducibility, Second Version," 45.

miscognition may be realized. However, my aim here is not to adopt Benjamin's perspective as a way to reform late capitalism through forms of interactive media. Benjamin's theories of technology are co-dependent upon the discursive conditions of modernity and thus there are inherent pitfalls in applying Benjamin's work to the political and economic complexities of contemporary technoculture. However, I do argue that we can use Benjamin's concepts of mimesis as a way to think about the novel modes of embodied play that video games afford. Mimesis operates as an imbrication of the phenomenal body and game system that attunes the sensorium to computational rhythms and temporalities.

Specifically, I focus on Kipcak and Urban's *ArsDoom* as a landmark development in game art and one that defamiliarizes dominant perceptual configurations related to navigating perspectival space within the first-person shooter through mimetic plays of the senses. In the process of replicating a destructible version of the Brucknerhaus, *ArsDoom* employs institutional critique through operations of mimesis that engage the body's apperceptive and proprioceptive senses. By abstracting *Doom*'s gameworld into a museum, Kipcak and Urban draw the act of embodied perception to the engine's modular capacities while engaging a kind of playful miscognition in which the game's familiar interface is made strange.

Kipcak mentions that his work on *ArsDoom* was influenced primarily by his skills as an architect and background in interaction design. In the early 90s, he began researching ways to construct virtual exhibitions as well as producing interactive web versions of his physical works for the Venice Biennale. At the time, Kipcak was creating architectural simulations using VRML (Virtual Reality Modeling Language), a file format used to render 3D vector-based graphics for web applications. In 1993, Kipcak adopted the *Doom* engine over VRML because its level editor was considerably more efficient and powerful for designing 3D interiors. He also gravitated to

the *Doom* engine because its ray-casting technology functioned as an “instrument of visualization,” allowing Kipcak to produce detailed surface textures and animate in-game objects



Figure 2.9: Architectural rendering of the Brucknerhaus exterior from *ArsDoom* (Orhan Kipcak and Reini Urban, 1995).



Figure 2.10: Firing at an enemy artist using a crucifix gun in *ArsDoom* (Orhan Kipcak and Reini Urban, 1995).

in real-time. He was interested in using *Doom* technology to produce a virtual environment that was at once visually complex yet held a “spatial haptic effect...a high degree of interactivity” for patrons.³⁵

In 1995, Ars Electronica urged Kipcak to create an interactive work for the festival. In response, he proposed *ArsDoom* as a combination of virtual exhibition and popular first-person shooter. Festival director Peter Weibel subsequently funded the project. Kipcak used the engine in conjunction with *Doom II* and Autodesk’s AutoCAD to reproduce the Brucknerhaus exhibition space, in turn, inviting festival artists—Ecke Bonk, Heimo Zobernig, Jörg Schlick and Peter Kogler, among others—to submit digital copies of their work to be displayed in the game map (Figure 2.9). All participating artists agreed with Kipcak’s terms that works would be reproduced in the game to be defiled by the player.³⁶ Kipcak made the goal of gameplay straightforward: players would use *Doom*’s iconic chainsaw and novel weapons such as crucifix and paintbrush guns to shoot at the artists and ruin their art (Figure 2.10).

Kipcak collaborated with eight studio assistants to create in-game assets and design characters and mechanics. In particular, Curd Duca, who would go on to become a prominent musician within the ambient scene, designed sound effects and composed the game’s jarring musical score. Reini Urban was responsible for level design and programming interactions within the game. He was also able to modify *Doom*’s level editor so that it could produce various angles to accommodate the geometric layout of the Brucknerhaus. Kipcak and Urban took the 3D data from AutoCAD and compiled the architectural designs using the AutoLISP programming language, exporting them to the *Doom* editor. In the editor, the artists programmed

³⁵ Orhan Kipcak, “ARSDOOM - art adventure,” *Ars Electronica Archive*, 1995, http://90.146.8.18/en/archives/festival_archive/festival_catalogs/festival_artikel.asp?iProjectID=8643.

³⁶ Orhan Kipcak, “Orhan Kipcak (ArsDoom, ArsDoom II) (1995-2005),” interview by *gamescenes.org*, November 4, 2009, <http://www.gamescenes.org/2009/11/interview-orphan-kipcak-arsdoom-arsdoom-ii-1995.html>.

interactions among player, artworks, and enemy artists, turning the digital exhibition space, as Kipcak mentions, “into a reserve of aesthetics with rapidly alternating focuses...In the agony of a hermetic world, the visitor, armed with color gun and water hose, slips on the character masks of Beuys, Rainer, Baselitz or Koons.”³⁷ Kipcak imagined that players would adopt various personas from the contemporary world when immersed into the game’s first-person perspective. At times, the player can channel Austrian artist Arnulf Rainer and his techniques of painting over existing photographs. Similarly, players embody multimedia artist Hermann Nitsch and splatter blood upon the artworks in reference to his brutal, ritualistic, paintings and performances. In other instances, participants fire projectiles from a TV remote, a citation to the legacy of Nam June Paik (Figure 2.11). While occasionally shooting at the paintings causes them to turn upside down, a reference to Georg Baselitz and his technique of inverting his works for display. Kipcak and Urban rendered characters and textures within the Brucknerhaus by scanning photographs and video footage for inclusion in the *ArsDoom* WAD file. Photographs and video were scanned in to create the game’s enemies: large floating heads of the artists participating in the festival. Similarly, footage of the urban landscapes surrounding the Brucknerhaus was scanned and imported into the map. Kipcak and Urban exhibited *ArsDoom* on a series of computers installed for public use but also created an installation of the game in stereoscopy. With this 3D rendering, they used an LCD shuttle screen as well as a prism and mirrored viewing device to produce a stereoscopic simulation of walk-throughs within the *ArsDoom* environment. Lastly, the artists exhibited *ArsDoom* through networked play. During Ars Electronica, players could directly download the mod’s client software and levels to access the game in four-person multiplayer. As Kipcak mentions, his aim was to extend the experience of the work into “telematic space.”³⁸

³⁷ Kipcak, “ARSDOOM - art adventure,” *Ars Electronica Archive*, 1995.

³⁸ Kipcak, “ARSDOOM - art adventure,” *Ars Electronica Archive*, 1995.



Figure 2.11: Shooting projectiles from a TV remote-launcher, a citation to the work of Nam June Paik from *ArsDoom* (Orhan Kipcak and Reini Urban, 1995).

The Brucknerhaus level, digital artworks, and AutoSLIP CAD tool were made downloadable as freeware throughout the duration of the festival.

This playful sense of miscognition afforded by the interactivity of *ArsDoom*, or rather trying on the masks of contemporary artists and employing their techniques as destructive forms of expression, instigates the body’s proprioceptive sense of moving and acting within *Doom*’s 3D gamespace. As Jacob Gaboury suggests, embodied perception within the first-person shooter indicates a simulation of perspective, reproducing the vision of an individual aiming a weapon within a confined projection of space. In the FPS, the player’s actions within space—their capacities to move and fire—are influenced by vectors of line projected from the center of the frame along their field of vision to visible points in the distance, often indicated by a cross-hair.³⁹ In *ArsDoom*, this embodied perspective of the FPS questions the body’s proprioception through operations of mimesis. Relational similarities among entities in the gameworld—museum and

³⁹ Jacob Gaboury, “Perspective,” in *Debugging game history: a critical lexicon*, eds. Raiford Guins and Henry Lowood (Cambridge, MA: The MIT Press, 2016), 364.

battlefield, paintbrush and gun, TV remote and rocket launcher, blood and paint, player and critic—produce an assemblage among body, system and engine in which perceptions of space are generated as varying degrees of intensification. When the player acts along the vector lines of this embodied vision, mimetic relations among objects rearticulate *Doom*'s gameworld as abstractions of space. Rapidly firing bullets from a crucifix at paintings across the gallery and witnessing them explode in blood, impinges upon the body's proprioceptive sensitivity in the way that vectors of firing and running through space are experienced as the push and pull of forces upon the player's affective capacities.

In his discussion of affect and gameplay drawing the from the work of Gilles Deleuze and Felix Guattari, Colin Cremin refers to the ebb and flow of in-game forces as rhizome-play, or when the player is co-constituted with other objects, characters, mechanics, and spaces, producing a field of intensities from their interactions within the system. This field of sensory experience emerges from the unfolding proximities and distances among bodies and forces of play at work. Within rhizome-play, the player collides with *friction-images*, or the invisible forces of game physics, gravity, tactility, slickness, and velocity through which intensities are reciprocated in the body.⁴⁰ In *ArsDoom*, the experience of friction-images—the forces at work when running through the Brucknerhaus, firing surreal weapons, or turning the paintings upside down—results in an affective experience of miscognition (Figure 2.12). This perceptual “grasping” as it were, at crucifix-guns, TV remote-rocket launchers and paint-blood, modulates the body's proprioceptive acuteness to perspectival space. The push and pull of the game's forces, its trajectories of slinging and firing strange objects along vectors of line, denaturalizes familiar modes of play within the FPS genre. Player actions in *Doom* transform into something-

⁴⁰ Colin Cremin, *Exploring Videogames with Deleuze and Guattari: Towards an Affective Theory of Form*. London, UK: Routledge, 2015, 61, 106.

other, producing within the body abstract sensations of movement along unfamiliar vectors of line. By articulating familiar actions and objects in *Doom* as seemingly unrelated expressions of play, 3D space is reciprocated intensively through the body's mimetic faculties.



Figure 2.12: Destroying an artwork by firing a crucifix gun and painting over its contents in the style of Arnulf Rainer from *ArsDoom* (Orhan Kipcak and Reini Urban, 1995).

In the following section, I examine the ways in which mimetic plays of the senses in *ArsDoom* are expressed through traditions of institutional critique within the avant-garde. As an art mod, *ArsDoom* defamiliarizes our expectations of navigable space in the first-person shooter by generating novel perceptions among relational similarities in *Doom*'s objects and environments and the Ars Electronica festival. By transforming *Doom* into a metaphorical killing field for the contemporary art world, Kipcak and Urban make apparent certain impositions of institutional power, while revealing the engine's modular architecture as an act of gamic apperception.

The Strangeness of Institutional Critique

For Kipcak, appropriating a first-person shooter and recontextualizing its violence as a strategy of institutional critique, presented "...obvious similarities between a battlefield and the Artworld." As Kipcak states, "The competitiveness of the artscene reflects the Darwinian nature

of videogame landscapes - in a sense, the videogame made the invisible conflicts of the artworld "real", explicit and visible, albeit in a playful way." Critics and the Linz media were unreceptive to *ArsDoom's* use of parody, going as far as to call it a "fascist orgy of violence." Other media outlets such as Der Spiegel and the BBC were more appreciative of a first-person shooter as an expressive art form, giving praise to *ArsDoom's* anarchy as something at once both subversive, yet affirmative in its critical position on the politics of contemporary art.⁴¹

Critical strategies of revealing the politics of collection and display emerged out of various directions of the avant-garde after World War II, predicated on what Arthur Danto has called the institutional theory of art. Danto holds that "To see something as art requires something the eye cannot decry—an atmosphere of artistic theory, a knowledge of the history of art: an artworld."⁴² That is to say, the viewer's reception of contemporary art asserts that an object transforms into art only within the "artworld" and the institutional frameworks that Danto describes. Outside the context of the white cube, Duchamp's infamous *Fountain* or Warhol's displays of Brillo boxes are recognized as commodities. Yet, when exhibited within the austere space of the gallery, they are taken into consideration as aesthetic objects and interrogate certain impositions of power within the museum. What does the museum consider art? Which artists and works are displayed? What works are chosen to represent an epistemology of the human condition?

Boris Groys refers to this dual nature of the Duchampian readymade—an everyday mass-produced commodity appropriated and designated as art—as a "paradox-object." This is something that possesses a dual ontology as both non-art outside of the museum and transformed into an art object when exhibited within an institution. As Groys mentions, the paradox-object

⁴¹ Kipcak, "Orhan Kipcak (*ArsDoom*, *ArsDoom II*) (1995-2005)," interview by *gamescenes.org*

⁴² Arthur Danto, "The Artworld," *The Journal of Philosophy* 61, no. 19 (1964): 580.

relates to practices of institutional critique in that the self-contradiction of such works aims to politically intervene in the institution's power to collect and display by "*transcending the borders of the art system – while at the same time remaining within these borders.*"⁴³ These subversive-affirmative approaches emerged in the first wave of institutional critique throughout the 1960s and 70s with works by Hans Haacke, Marcel Broodthaers, and Daniel Buren. A second movement of institutional critique emerged starting in the late 1980s and 90s with conceptual artists such as Fred Wilson, Andrea Fraser, and Mark Dion, who expanded the critique of their predecessors to investigate subject formation through collection and display. As Brian Holmes explains, this newer generation of artists... "*pursued the systematic exploration of museological representation, examining its links to economic power and its epistemological roots in a colonial science that treats the Other like an object to be shown in a vitrine.*"⁴⁴

As an example of institutional critique, in 1974 Hans Haacke exhibited his installation *Manet-PROJEKT '74* at the Wallraf-Richartz Museum in Cologne, which consisted of multiple documents unveiling the provenance and ownership of Eduard Manet's painting *Bunch of Asparagus* (1880), housed in the museum's collection (Figure 2.13). One particular document revealed that the painting was donated by Hermann Josef Abs in 1968, who at the time was chairman of the Friends of the Wallraf-Richartz Museum and had once served as financial advisor to Adolf Hitler. By situating the painting's provenance as a paradox-object, Haacke's conceptual work reveals the institution's abhorrent connections among capital and politics.

Following these critical movements, I would suggest that *ArsDoom* likewise employs sensual operations of mimesis to question institutional power through the game's paradoxical

⁴³ Boris Groys, *Art Power* (The MIT Press, London, 2008), 3.

⁴⁴ Brian Holmes, "Extradisciplinary Investigations. Towards a new critique of Institutions" (2007) in *Art and Contemporary Critical Practice: Reinventing Institutional Critique*, ed. by Gerald Raunig and Gene Ray, (MayFlyBooks, London, May 2009), 57.



Figure 2.13: Hans Haacke's *Manet-PROJEKT'74* (1974) including ten panels of provenance documents under glass and color photo reproduction of Manet's *Une botte d'asperges* (Bunch of Asparagus) at Paul Maenz Gallery, Cologne, 1974. Photo courtesy of Rolf Lillig.

status: an object that is simultaneously a popular entertainment commodity and artwork within the context of the Ars Electronica festival. In other words, *Doom* modified into a simulated museum through which works are exhibited and ultimately judged by patrons, reconfigures traditional modes of public participation within the institution through its interactivity, while remaining within the art world's organizational structure. As a video game, *ArsDoom* interrogates the borders of what the art world will accept as an aesthetic object within the wake of late capitalism. And, it is *ArsDoom's* contradictory status that engages the player in a mode of gamic apperception through its subversive expression of the art world's competitive nature. It discloses the engine's technical operations by transforming *Doom's* familiar levels into the strange gameworld of the festival.

In *ArsDoom*, gamic apperceptivity is made apparent in the way it introduces new modes of public participation. Within traditions of institutional critique in the historical and neo avant-

gardes, Frazer Ward posits that publicity, drawing from a Habermasian understanding of the term, expresses not only the artistic medium but the range of “practices of intervention in economies of cultural production and reception that go realize conceptions of the public sphere.” Ward contests that the works of an artist such as Haacke demonstrate a recognition of the contemporary museum as a space in which multiple and conflicting forms of publicity are braided together among historical bourgeois publicity and those publics of late capitalism. The institution becomes a site of “performing art’s function as publicity within a prescribed and always already compromised cultural space.”⁴⁵

In *ArsDoom*, apperception is realized in emergent forms of publicity organized around the interactivity of the mod (both within actual space and networked play) as a commodity of late capitalism and more traditional forms of publicity realized through the installation’s physical display within the Brucknerhaus. As Jon Cates points out, *ArsDoom* was a conceptual experiment in deliberately rupturing the values of Ars Electronica. Critics observed that participants were enthralled with the opportunity to shoot the image of Peter Weibel, director of the festival. To Cates, the publics’ desire to destroy Weibel’s image foregrounds “the carnivalesque inversion of the social order or hierarchical power structures of the festival.”⁴⁶ That is to say, allowing festival goers to destroy Weibel, the artworks, and caricatures of the other artists, gestures toward a unique re-materialization of institutional power onto the public through an interactive medium. Within these mimetic engagements of figurative violence, the player armed with bullet-firing cross, takes on the roles of critic, curator, and director. By vandalizing images and killing the artists, participants decide which artworks should remain exhibited and unscathed, or in the most anarchistic sense, they judge that nothing is of value and

⁴⁵ Frazer Ward, “The Haunted Museum: Institutional Critique and Publicity,” *October* 73 (1995): 72, 88-89.

⁴⁶ Jon Cates, “Running and Gunning in the Gallery: Art Mods, Art Institutions and the Artists that Destroy Them,” *artgamestudies.org*, <http://artgamestudies.org/RunningAndGunning.pdf>, 11.

the entirety of the festival should be razed. Within the strangeness of this mode of publicity—the game transformed into the simulated Brucknerhaus with splattered blood and paint across the artworks—the player is made aware of the engine’s modularity and its capacity to render gameworlds in a bricolage of things, environments, and mechanics (Figure 2.14).

As a pioneering work of game art, *ArsDoom* questions the ways in which a game engine can be repurposed toward expressive art forms. Within histories of the avant-garde, artists have always held a particular attraction to new technologies as a means to integrate their work into the

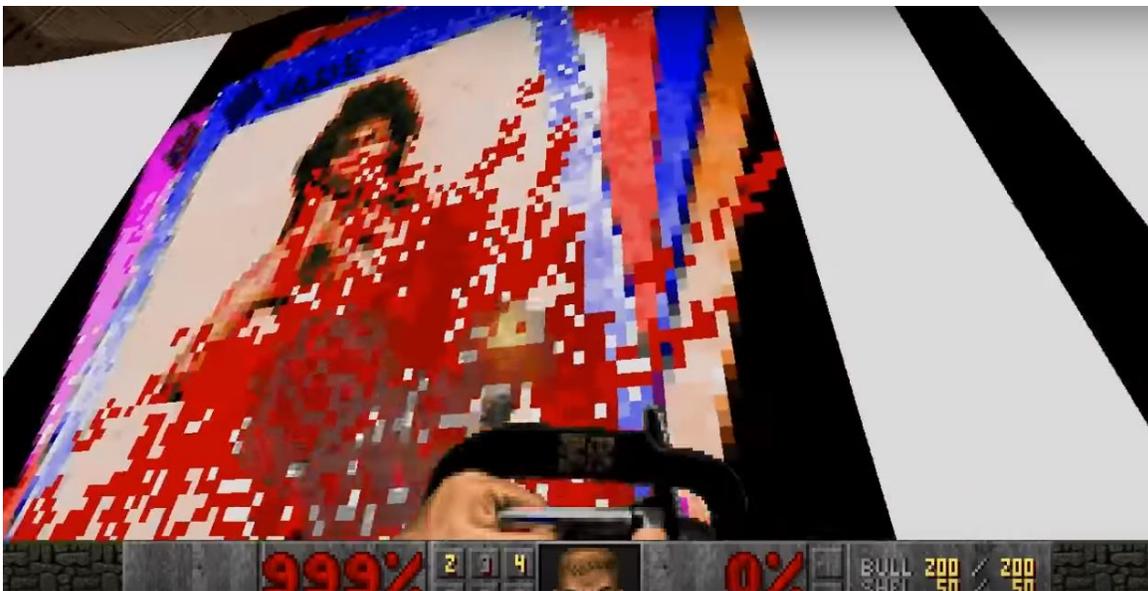


Figure 2.14: Splatters of blood-paint across an artwork in the simulated Brucknerhaus of *ArsDoom* (Orhan Kipcak and Reini Urban, 1995).

praxis of the public sphere; Duchamp and his cinematic experimentation with whirling, illusory rotoreliefs in *Anemic Cinema* (1926) or Rauschenberg’s interactive performance *Open Score* (1966) at *9 Evenings: Theatre and Engineering*, consisting of a tennis match rigged with a complex system of sound sensors and light control mechanisms. In the act of rearticulating a commercially recognizable video game technology into an aesthetic object, *ArsDoom* confronts institutional power by considering the ways in which a video game can express public forms of aesthetic participation. By employing the engine’s modular nature as a *medium of institutional*

criticism, *ArsDoom* produces novel forms of bodily experience through a parodic gamespace that invites us to consider the playful relationality among the violence of a first-person shooter and the combative ethos of the contemporary art world. As an affirmative projection of difference, *ArsDoom* territorializes new formations of aesthetic experience working from habitual perceptions of 3D space in the first-person shooter. And, in these familiar vectors of embodied vision, it is the uncanniness of *ArsDoom*'s mimetic operations —felt intensities of abstract gamespaces produced through a run n' gun in the museum—that potentially rematerializes the participant as critic, artist, curator, and founder of the Ars Electronica festival.

CHAPTER THREE

“Perspective Engines” and the Strangeness of 3D Space in *JODI’s Untitled Game*

That’s the whole trick of these games, that they are perspective engines. All the time they create tunnels and illusions of a 3-D space and that’s part of the “kick” you have as the user, that you think you explore and you enter and you move into. In fact, the only thing which is happening, is a perspective which just is drawn all the time – so it’s always about graphical tricks.

-JODI

At their 2012 exhibition *Street Digital* at the Museum of the Moving Image in New York, Netherlandish artist duo JODI (a portmanteau of the artists’ names: Joan Heemskerk and Dirk Paesmans) invited viewers to interact with their work *Untitled Game*, a series of 14 modifications made to id Software’s canonical first-person shooter *Quake* between 1996 and 2001. By altering the source code of the *Quake* engine, in turn, producing a series of glitches that disrupt the game’s graphical components, JODI’s *Untitled Game* rearticulates the player’s habitual encounter with fully-rendered, navigable 3D space as a series of bizarre and often psychedelic abstractions of *Quake*’s perspectively-rendered gameworld. Instead of fighting demons and interacting with the familiar polygons and textures of *Quake*’s hellish environments (although in several of the mods, JODI left *Quake*’s sound effects intact), participants at the *Street Digital* show were subjected to an abnormal encounter with four of the mods—*A-X*, *Ctrl-Space*, *Spawn*, and *Arena*—displayed on large hanging screens within the cinematic presence of a darkened gallery. Players could control the mods using joysticks and attempt to orient their viewpoint (often unsuccessfully) within the *Quake* gamespace transformed into a series of abstract spaces. For instance, in *Ctrl-Space* the game became a map of swirling black and white moiré patterns that appear to pulsate with each newly rendered frame. JODI’s exhibition at *Street Digital* not only introduced participants to a novel mode of embodied reception—one that was

disconcerting and at times assaultive to the senses—but also points to the ways in which *Untitled Game* stages a type of perceptual intervention into our normative play experience within the real-time 3D space rendered by a game engine. Within this historical moment of art modding, I look at the aesthetic work of *Untitled Game* through a triangulation of the technics of the *Quake* engine created by John Romero and John Carmack throughout 1995 and into 1996 in its abilities to render real-time 3D space, to the formal experimentations of JODI's mods and the player's aesthetic investment in their strangeness.

I propose that in JODI's work, it is this strangeness of *anamorphic space* that reveals our phenomenal experience and impedes our normal perception of perspectival space common to the *Quake* engine. Through a mimetic, embodied encounter with anamorphosis, the body becomes both apperceptively attuned to the glitch's formal disruption of space while proprioceptively destabilized by its warped presence. Specifically, I focus on three of JODI's mods: *Ctrl-9*, *I-N*, and *Q-L* that demonstrate our affective engagement with anamorphic space. First, I look at the historical emergence of the *Quake* engine through a media archaeological tracing its forensic environments to the formal disruptions of space in *Untitled Game*. Here, I examine the ways JODI have altered the game's source code, particularly in adjusting certain values of commands and variables within *Untitled Game's* *autoexec.cfg* files. Finally, I look at the ways *Ctrl-9*, *I-N*, and *Q-L* engage a bodily mode of apperceptivity that reveals our automated perception of real-time 3D space through the unpredictable generativity of glitch. Similar to the avant-garde techniques of the structural flicker film, apperception emerges from a threshold experience that at once draws awareness to the mods' perceptual effects while also eliciting a range of sensual responses from the body. Here, I frame the strange perceptual fluctuations generated from JODI's work as a *flicker mod*. This sensation of apperceptivity in the flicker mod co-extends to

the players proprioceptive capacities through kinesthetic illusions of movement that *Untitled Game* generates. JODI's work modulates our spatial resonance in the world just as it attunes the body's self-sensitivity to navigating anamorphic space.

The Work of Anamorphosis

JODI were among the first artists in the mid-1990s to experiment with net art. Forming their collaboration in 1994, they began to create technical interventions on the Internet to radically destabilize the intuitive interfaces that users had become familiar with when navigating the World Wide Web. In their aims to subvert our habitual experiences of computers, both artists were influenced by avant-garde traditions of video art that emerged during the late 1960s. Paesmans had even studied sculpture under the tutelage of Nam June Paik while a student at the Kunstakademie at Dusseldorf. In 1995, the duo accepted residency at the CADRE (Computers in Art, Design, Research, and Education) Institute at San Jose State University: a laboratory space supporting groups of artists who began using components of the Internet in their experimental works. They also proposed artistic uses of technologies within the dominant framework of Silicon Valley's computer culture, at the time a flourishing tech hub for IBM, Adobe, and Apple among others. While at CADRE, JODI worked with artist Anne-Marie Schleiner who in July of 1999 curated the online exhibition "Cracking the Maze: Game Plug-ins and Patches as Hacker Art": a seminal project in histories of game art featuring playable art modifications for popular commercial titles of the time. JODI featured their work *SOD* (1999), a modification that abstracted the gameworld of the popular first-person shooter *Wolfenstein* (1992) into a series of black and white geometric patterns.

Before moving to California and solidifying their position as artistic collaborators, Heemskerk had studied at the St. Joost Academie in Breda and was working in both photography and 3D modeling, while Paesmans graduated from St. Lukas in Brussels in the plastic arts and at the Royal Academy of Fine Arts at Ghent before working under Paik. While at CADRE, JODI explored software modifications made to video games through their work *Untitled Game*: a series of 14 modifications made to the *Quake* engine created between 1996-2001 by altering sections of the game's source code and transforming its 3D environments into abstract interfaces. These mods disrupt the intuitive flow of play by transforming Quake's perspectival gameworld into a series of anamorphic experimentations, subjecting the player to dizzying black and white moiré effects, flickering lines, monochrome static frames, as well as frustrating paradoxical movements (Figure 3.1).



Figure 3.1: Swirling, psychedelic patterns in *Ctrl-Space* (JODI, 1996-2001).

In an interview with conservator Lisa Adang, JODI describes that some of their earliest *Untitled Game* mods were created at CADRE using a Macintosh Quandra and the 1997 Mac Shareware of *Quake*, which only came with the initial episode of the game. Constrained to using only portions of the engine's source code (the proprietary code would not be available under general public license until 1999) Heemskerk designed the mod *Arena* by changing all of *Quake*'s colors to white, completely obscuring the player's navigation within the game. Yet with *Arena*, Heemskirk left the game's sound effects intact, providing basic auditory cues for attempting to move and fight off enemies within the static white frames of the interface.¹ As with *Arena*, JODI would opt to include the game's sound effects in the remainder of their *Untitled Game* mods, with some versions including glitched sounds coupled against an array of increasingly complex visual patterns.

In 2001, JODI circulated copies of *Untitled Game* on a CD-ROM in several net art magazines such as *Mute*, a leftist publication focused primarily on the intersections among art and cyberculture. The distribution of these mods on CD-ROM marked what Annette Schindler and Waling Boers see as a disruption within the isolated discourse of net art in that JODI released a work that could be played on a PC without any reliance on an Internet connection. Sharing *Untitled Game* on a disk indicated JODI's aim to reach a wider community of gamers and modders outside of Internet artists, while shifting focus away from the purity of net art and toward the capacity for video games to potentially deconstruct contemporary understandings of technoculture.² By altering the *Quake* engine at the level of code, JODI sought to shift the

¹ See Joan Heemskerk and Dirk Paesmans interviewed by the author, May 30, 2013, Dordrecht, Netherlands in Lisa Adang, "Untitled Project: A Cross-Disciplinary Investigation of JODI's *Untitled Game*," *Rhizome*, Version 2, 2014, <http://media.rhizome.org/artbase/documents/Untitled-Project:-A-Cross-Disciplinary-Investigation-of-JODI%E2%80%99s-Untitled-Game.pdf>, 12.

² Annette Schindler and Waling Boers, "The Likelihood of Disturbances," in *INSTALL.EXE-JODI*, ed. Tilman Baumgaertel (Basel: Christoph Merian Verlag, 2002), 55.

player's interaction away from optimizing a pleasurable experience within a perspectively-coherent gamespace toward a novel type of sensual engagement with anamorphic projection.

As I argue throughout this chapter, the experience of JODI's *Untitled Game* is largely attributed to the rendering of anamorphic spaces. Since the Renaissance, linear perspective has operated as the dominant mode of visuality to naturalistically reproduce depth within an image. In other words, it is a form of rendering illusory space that corresponds to human vision and our perception of objects as they diminish in scale and recede into the distance, converging at a given



Figure 3.2: An anamorphic rendering of a skull in *The Ambassadors* (Hans Holbein the Younger, 1533).

point along a horizon line. Alongside the historical development of linear perspective, artists since the Quattrocento have explored anamorphosis as an alternate means of rendering space, one that is stretched and warped yet retains geometric accuracy. Anamorphosis requires the viewer to actively interpret the vantage point of the image by viewing it diagonally, producing an

optical effect of the warped image appearing as foreshortened and thus geometrically precise. Perhaps the best-known example of anamorphic perspective is the mysterious skull in Hans Holbein's oil painting *The Ambassadors* from 1533 (Figure 3.2). In the painting, two well-dressed French statesmen stand amongst a collection of worldly possessions spread across a table including an astrolabe, books, and musical instruments. In the immediate foreground sits a conspicuously placed skull stretched across the canvas that when viewed from the correct angle reveals a proportioned skull, an icon of *vanitas* or *momento mori*: a visual reminder of death's inevitable arrival.

The operations of the code running JODI's *Untitled Game* modulate the player's mimetic capacities to feel the anamorphic distortion of 3D space as an intensive process. The player no longer perceives fully-rendered, actionable space through a visual regime of Cartesian perspective but rather considers the aesthetics of *Untitled Game* as an affective modality of anamorphic projection. When the player encounters the unique durations, temporalities, and processes that comprise the illusory and kaleidoscopic effects of *Crtl-9*, *I-N*, and *Q-L*, affectivity operates through an aesthetic vector of mimesis in which *anamorphic* space locates itself intensively within the body.

In his discussion of artist Robert Lazzarini's *skulls* (2000), an installation consisting of four warped skulls scanned from an actual human model, stretched and distorted using digital CAD software, and cast in bone, Mark Hansen proposes what he calls an affective topology of space. Lazzarini's warped skulls confront the viewer with anamorphic space that creates an affective "analogy" between the body's intensive response and the bewildering spatiality of the skulls in the gallery. As the viewer moves around the skulls, attempting to locate a perspectival vantage point that continually eludes them, the anamorphic rendering of the sculptures correlate

as an embodied “form-giving” of intensive space (Figure 3.3). Just as the viewer must work to decipher the anamorphic skull in Holbein’s painting, Hansen argues that Lazzarini’s distorted craniums mobilize affectivity as an intensive, bodily space that is analogous to the digitally-manipulated space of the skulls. Hansen calls this affective modality the digital any-space-whatever (ASW), a term he borrows from Gilles Deleuze’s philosophy of cinema to describe the virtuality of space, or the potential ontological registers produced among given objects, parts, and connections within the cinematic image. The digital ASW operates through a proprioceptive, non-optical mode of embodiment. The aesthetics of Lazzarini’s skulls points to the way the digitally-rendered sculptures generate an autonomous modulation of space as an intensive state reproduced within the body through anamorphosis.



Figure 3.3: The distorted, anamorphic space in the installation *skulls* (Robert Lazzarini, 2000). Photo courtesy of the artist.

As Hansen states, this digital ASW also generates a type of bodily anxiety over “just how radically alien the formal field of the computer is.” In this instance, anamorphic perspective actuates the virtuality of the body to create within itself a sensation of distorted space introduced from the exterior via the alien processes of the computer’s modulation of the skulls. Hansen calls

this process affective proprioception as the way in which the body decodes anamorphic space. This affective modality operates autonomous to perception and serves no relation to vision.³ Rather, it is a sensory mode of experiencing spatiality outside the figural, pictorial, or sculptural. Visual perception becomes impeded as the body works to decipher the distorted skulls, and in turn, space becomes intuited through an autopoietic response in which affective feedback from the body works to maintain its spatial equilibrium, producing an intensive analog to the warped, exterior space.⁴

In *Metagaming*, Stephanie Boluk and Patrick LeMieux bring Hansen's concept of the digital ASW into conversation with anamorphic video games. They suggest that when the player interacts with anamorphic rendering, it produces an experience of what they call *memento mortem mortis*. If the stretched skull that looms in front of Holbein's ambassadors operates as an emblem of *memento mori* and a reminder of death yet to come, then *momento mortem mortis* reflected in anamorphic games engages the viewer with the unique phenomenology of the gameworld's shifting two and three-dimensional perspectives. It operates as a reminder of the boundaries of an anthropocentric phenomenology. In games such as *Super Paper Mario* (2007) with its changing perspectives between side-scrolling and three-dimensional navigation, or the Escherian puzzle chambers one must teleport through in *Portal* (2007), the player is taken up in the alien rhythms of the game's technical operations that exceed human experience. Anamorphic games reflect a form of metagaming *about* graphical rendering, and more so, "about *aboutness* itself."⁵ As I demonstrate in this chapter, Hansen's digital ASW and Boluk and LeMieux's *memento mortem mortis* allow us to think about JODI's *Ctrl-9*, *I-N*, *Q-L* as anamorphic games

³ Mark B. N. Hansen, *New philosophy for new media* (Cambridge, MA: MIT Press, 2006), 203, 205-206.

⁴ Hansen, *New philosophy for new media*, 229.

⁵ Stephanie Boluk and Patrick LeMieux, *Metagaming: playing, competing, spectating, cheating, trading, making, and breaking videogames*, <http://staging.manifoldapp.org/read/c5926868-00c4-45f8-8e91-45cfd9140a87/section/fb4e77c9-5347-42a4-9149-dfd009bd864f#ch03>.

that operate through a proprioceptive-apperceptive aesthetic that works outside of visual perception. In disrupting the perspectival rendering of the game engine, anamorphosis in *Ctrl-9*, *I-N*, and *Q-L* propels the player toward an affective instability in which the body must work to ground itself in 3D space. And, this spatial grounding always eludes the player's ability to orient themselves. The alien processes of the mod exceed the perceptual capacities of the human and catalyze a bodily feeling of paradoxical space.

As I elaborate on in further detail, JODI employ the *Quake* graphics engine to render anamorphosis through what I argue are the perceptual mechanisms produced from glitch, and specifically the "flicker" effect. Here, I borrow the term from traditions of American avant-garde film and in particular, Tony Conrad's work *The Flicker* (1966), a film composed entirely of black and white frames that continually alternate in a system of patterns, generating a stroboscopic, hallucinogenic transmission of visual frequencies. Through their formal effects, we can think of JODI's *Ctrl-9*, *I-N*, and *Q-L* (among other mods in *Untitled Game*) as flicker mods, or what artist Axel Stockenburger describes as "kinetic screen sculptures" through the frenzied patterns of glitch they generate.⁶ JODI's first US solo exhibition *INSTALL.EXE* at Eyebeam in Brooklyn featured multiple *Untitled Game* mods situated at computer workstations for users to play. In 2007, the group exhibition *Show #9* at Pasadena-based gallery And/Or featured JODI's *Untitled Game* alongside fellow video game artist Archangel Constantini. At the show, audiences were invited to sit at a yellow table and play selected *Untitled Game* mods on desktop PCs contemporaneous with the *Quake* modding culture of the late 1990s. At the 2010 GLI.TC/H conference and festival at Roxaboxen Exhibitions in Chicago, JODI exhibited their mods *Ctrl-Space* and *EIMIAP* on dated laptops along a shelf for viewers to stand and play. Yet as I have

⁶ Axel Stockenburger, "From Appropriation to Approximation," in *Videogames and Art*, ed. Andy Clarke and Grethe Mitchell (Bristol, UK: Intellect Ltd., 2007), 33.

mentioned, at their *Street Digital* show in 2012 JODI transitioned away from displaying *Untitled Game* on outdated computers and instead projected them on large screens in a dark gallery (Figure 3.4). JODI's more recent exhibitions of these mods tend to mirror the sensory environment of Conrad's film played within an underground theater: the stroboscopic,

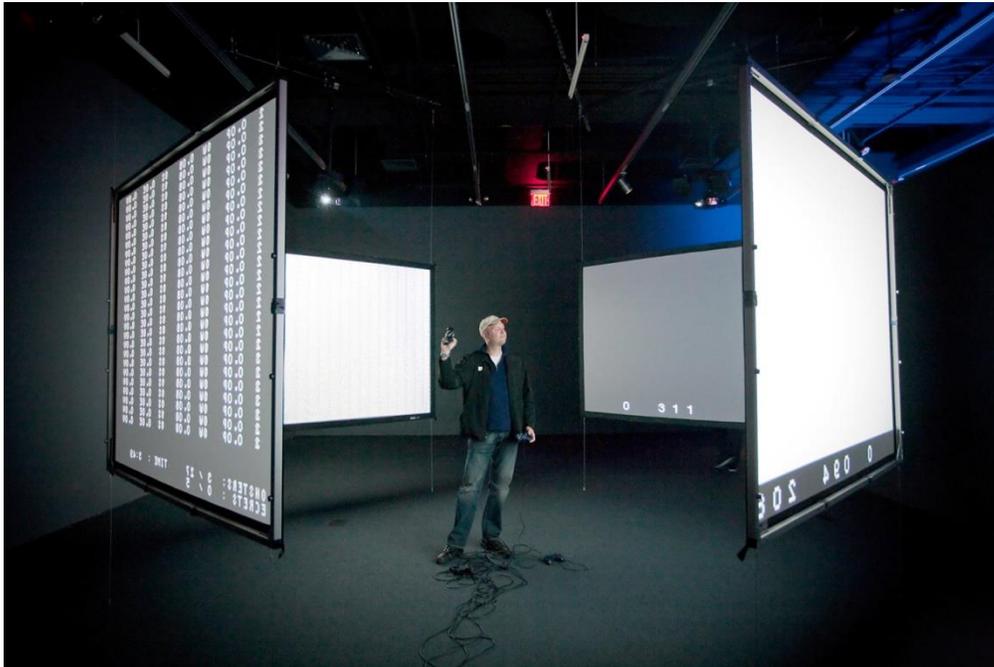


Figure 3.4: JODI's *Untitled Game* mods projected onto large scrims at in the 2012 *Street Digital* exhibition at the Museum of the Moving Image. Photo courtesy of Daniel Love.

accelerating effects of glitch patterns within the contrasting darkness. In this setting, the flicker mod opens up the body to sensory and perceptual effects much like Conrad's film, particularly through the immediacy of the large projections within the darkness of the gallery. JODI's use of defamiliarization as a formal strategy—altering the *Quake* graphics engine to produce these perceptual flickers—allow us to consider the aesthetic work of anamorphosis in *Untitled Game* and the bodily affectivity the mods generate, particularly through an aesthetic of glitch. Similar to Conrad's film, JODI's use of glitch modulates the body's apperceptive and proprioceptive faculties through a threshold experience, producing a strangeness in which the player is

perceptually aware of their spatial disequilibrium within 3D space, while opening up a range of affective responses in the body.

In the next section I propose a media archaeology of the *Quake* engine, that is to say, I uncover the game's graphics engine in its unique singularity, or rather the computational techniques that allowed for real-time 3D space in *Quake* to come into existence. Drawing from Matthew Kirschenbaum, I propose a forensic analysis of the engine through what he calls a grammatology—a machine reading of the working components of an inscription technology.⁷ I trace the mechanisms of the *Quake* engine to JODI's "glitching" its of source code at the forensic level of inquiry, to the formal materiality of *Untitled Game* and the player's sensual, mimetic engagement with the abstract space of the mods. By unveiling how a game engine works, its "engine-ness" we might say, and how JODI upend its standard operations through the rendering of anamorphosis, we can consider how art modding practices defamiliarize conventional modes of play.

A Grammatology of the *Quake* Engine

After the successful release of *Doom II* in 1994, John Romero and John Carmack began working on a 3D version of a previous 2D concept for a game called *Quake: The Fight for Justice*, inspired from their experiences playing *Dungeons and Dragons*. The game would be a medievaesque RPG with a warrior-like avatar named Quake equipped with a giant hammer. A comprehensive FAQ document titled QUAKETALK 95 created by Joost Shuur in October of 1995 provides a detailed archive of interviews, magazine articles, and programmer logs on the development of the *Quake* engine starting in the summer of 1994. In an interview from November of the same year, Romero spoke to the advanced nature of the 3D engine stating that:

⁷ Matthew G. Kirschenbaum, *Mechanisms: New Media and the Forensic Imagination* (Cambridge, MA: MIT Press, 2012), 88.

...the technology in quake will be much better than in doom...the graphics will use 3-D rendered models, unlike the 2-D images in *Doom*...so if you walk around a pillar, you'll see it in full 360 degrees, as opposed to only eight different rotations in *Doom*. You'll also be able to move in six different directions, and they'll [sic] be much better animation. We're also adding cool cinematic sequences that take place while you're playing the game. basically, doom will feel stiff compared to *Quake*.⁸

Toward the end of 1994, Romero also described in an interview that “the 3D [Quake] engine has gotten to the point where it’s going to be almost as much fun to experience the environment as it is to actually play the game...you should be walking around in quake just in awe, looking around and going, "this is awesome!"”⁹



Figure 3.5: Fighting monsters within the 3D environments of *Quake* (id Software, 1996).

John Carmack had already begun programming a potential 3D engine throughout 1995 but because of technical limitations with rendering and framerate, Romero and Carmack decided to change the direction of *Quake* from a slower, world-exploring RPG to a frenetic, fast-paced

⁸ John Romero, interview by Multimedia Magazine, in *QUAKETALK 95*, October 22nd 1995, <http://rome.ro/?offset=1481382456223&author=51a4f9c2e4b068f76a150e2a>.

⁹ John Romero, interview by Ed Dille, in *QUAKETALK 95*, October 22nd 1995, <http://rome.ro/?offset=1481382456223&author=51a4f9c2e4b068f76a150e2a>.

action shooter in the tradition of *Doom*. Developments on the reworked engine consisted of Carmack as main programmer, Michael Abrash assisting with assembly optimization and algorithms, and additional programming by John Cash. Game design was handled by Romero along with American McGee, Sandy Petersen, and Tim Willits, and art direction by Adrian Carmack, Kevin Cloud, and Paul Steed. The game even featured an aggressive industrial soundtrack produced by Trent Reznor of Nine Inch Nails to accompany the frenzied gameplay. Released on June 22, 1996, the final build of *Quake* immersed the player into a dark fantasy world as “Ranger,” a special forces op who must eliminate an enemy code-named “Quake” after being sent through an interdimensional portal using a government teleportation device known as the Slipgate. Using an armory of high-powered weapons—shotguns, nailguns, rocket launchers—Ranger must firefight his way through various demons and monsters to defeat the evil Shub-Niggurath (Figure 3.5).

It is important to note that Carmack developed the unique programming language QuakeC in 1996 for purposes of modification and expandability to the engine. According to modder David “DarkGrue” Hesprich in his *QuakeC Reference Manual*, the language has similarities to C, yet the QuakeC source code is compiled with the QuakeC compiler (QCC): a program that turns specific statements of a programming language into functioning machine code for a computer to interpret. The QCC compiles the portions of the engine’s QuakeC source code into what is called a progs.dat file. QuakeC is an interpreted language meaning that when the engine is running it continually reads and executes data for the progs.dat file, which is loaded when the game is started.¹⁰ Using QuakeC and the compiler, a player can modify the engine’s source code and reconfigure game physics, alter the appearance and abilities of weapons, change

¹⁰ For a detailed explanation of the QuakeC interpreted language see, David Hesprich, *QuakeC Reference Manual*, <http://pages.cs.wisc.edu/~jeremyp/quake/quakec/quakec.pdf>.

character skins, and control elements of AI and triggers. A user could not modify the proprietary code of the *Quake* engine; however, this was made available in 1999 by id Software through the GNU General Public License. This meant that programmers could openly modify and edit the engine, producing an entire family of derivative engines, perhaps most notably Goldsource used by Valve Corporation to develop the first-person shooter *Half-Life* (1998) and its successor, the Source engine, used for both Valve's *Counter Strike: Source* and *Half-Life 2* in 2004.

Unlike the 3D space projected onto two-dimensional planes (often referred to as 2.5 D space) that was seen in *Doom*, the efficiency of the final *Quake* engine build and its ability to render a real-time 3D world is predicated on two core components developed by Carmack: binary space partitioning (BSP) and pre-calculated lighting. The best technical documentation of the engine's development comes from Michael Abrash's "Ramblings in Realtime" that he wrote while working on *Quake* with Carmack between 1995 and 1997.¹¹ In November of 1995, both programmers considered simplifying the source code to speed up rendering and produce arbitrary, real-time 3D, or the avatar's ability to move and look in all directions within the game environment. The engine was designed to use "maps" in which each environment is pre-rendered as a way to lower the CPU requirements needed to run *Quake* on home computers running at an average of 50-75MHz processor speed. Abrash explains that for any engine to work properly, its baseline operations must rasterize the appropriate color at the level of individual pixels, while keeping a consistent framerate, which for *Quake* would be keeping a minimum of 10-15 frames per second.¹² In computer graphics, rasterization refers to the process of transferring vector-based graphics—polygons used to construct the game environment—to a raster-based image of pixels for video display.

¹¹ See Michael Abrash, *Ramblings in Realtime*, *Blue's News*, <https://www.bluesnews.com/abrash/>.

¹² Michael Abrash, "The Quake Graphics Engine," (presentation notes, *Computer Game Developers Conference* April 2, 1996).

The major problem for Carmack and Abrash was that in 1995-1996, rasterizing algorithms did not exist that could process the number of polygons used in *Quake's* level design fast enough, and if a program was capable of handling all of the game's vector graphics, this would only prompt developers to build complex 3D worlds with even more polygons. With more complex geometries in the gameworld comes the issue of the rasterizing program efficiently drawing only the polygons within the player's view frustum, or their perspectival angle of sight in the map. If a rasterizing program drew pixels for every polygon, visible and non-visible to the player, this would ultimately result in what is called overdraw, and diminish the ability to consistently draw the correct surface pixel for those polygons within the frustum, thus making the frame rate inconsistent. As Abrash explains, Carmack realized that overdrawing would occur in a typical *Quake* level with roughly 10,000 polygons, especially when the specific "painter's" algorithm used to draw each pixel is required for every polygon within the frustum. Because these polygons are often hidden or partially covered by other polygons, typical rasterizing algorithms might overdraw each pixel within a frame 10 times on average. The problem of overdraw led Carmack and Abrash to develop the final iteration of the engine with two principles in mind. First, they had to find a way to efficiently minimize the polygon set within the frustum to those that are needed for player navigation and two, getting the appropriate surface textures for each pixel for those polygons that do need to be drawn, which includes shading and perspective control.¹³

The *Quake* engine is essentially divided into two components: static 3D environments and the moving objects within it that include avatar, enemies, doors, platforms, etc. In the static world, Carmack addressed overdraw by rethinking a solution to visible surface determination (correctly drawing each pixel) and culling, or removing non-visible polygons from the view

¹³ Abrash, "The Quake Graphics Engine."

frustum during the rasterization process. The answer was to preprocess each *Quake* level into a tree structure using Binary Space Partitioning (BSP) that as I have previously discussed, was utilized in the development of the *Doom* engine. Carmack designed each map in *Quake* as a singular BSP tree in which spatial data is stored in the leaf structures which divide out from individual nodes (Figure 3.6). With each map's BSP tree in mind, correct visible surface determination would occur by culling any geometries entirely outside the frustum, and clipping off sections of polygons that are partially outside of the frustum. Rasterization algorithms then only draw pixels for polygons visible (whole and partial) within the player's view. Thus, culling unnecessary polygons outside of the view frustum and clipping irrelevant portions of others, would draw each pixel only once with no overdraw, allowing the engine to focus on rasterizing areas of the level with more 3D complexity and without redrawing polygons (Figure 3.7).¹⁴

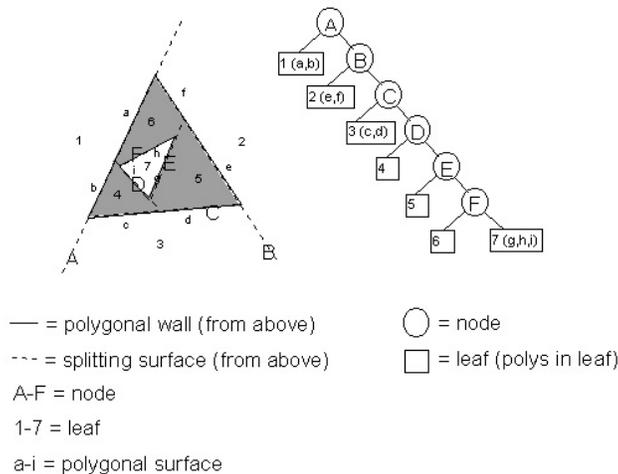


Figure 3.6: BSP ordering in which polygons within the map are stored as leaves from Michael Abrash, “Inside Quake: Visible Surface Determination,” *Ramblings in Realtime*, *Blue’s News*, <https://www.bluesnews.com/abrash/>.

¹⁴ Michael Abrash, “Inside Quake: Visible Surface Determination,” *Ramblings in Realtime*, *Blue’s News*, <https://www.bluesnews.com/abrash/>.

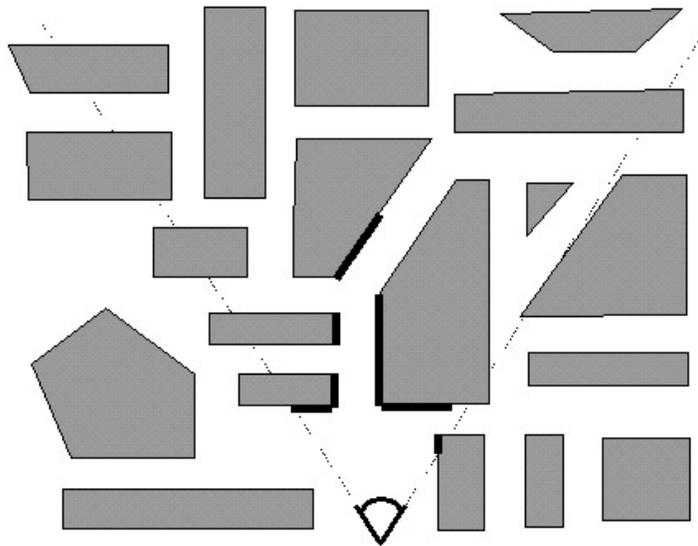


Figure 3.7: Visible Surface Determination in which only visible sections of polygons are drawn within the frustum and all others are culled, from Michael Abrash, “Inside Quake: Visible Surface Determination,” *Ramblings in Realtime*, *Blue’s News*, <https://www.bluesnews.com/abrash/>.

Carmack’s innovations in visible surface determination eventually lead him to use a potentially visible set (PVS) based on pre-calculated arrangements of polygons stored within the map BSP tree, which made for even less overdraw. With BSP ordering, Carmack placed the polygonal edges for a given map in a global list through which the engine can access and draw only those visible polygons within the viewpoint. The problem with using an edge list was that the engine could not retrieve and process thousands of polygonal edges within the frustum quickly enough, slowing frame rate to a crawl. Instead, using a pre-calculated PVS in which polygons are stored within BSP leaves derived from the global list would limit the number of edges that needed to be drawn for any given view of the gameworld. Within a PVS, the engine uses the global list to access convex portions of geometries subdivided out into the leaves. Carmack created a series of potentially visible sets for every possible viewpoint in the game stored in the leaves, meaning that the engine could easily access these polygons and draw them

using front-to-back ordering prearranged in the BSP tree, while clipping partially-obscured geometries and culling others entirely outside the frustum.

For a given PVS, the engine completes a visibility calculation for each leaf in the level's BSP tree, or area of subdivided, convex polygons. For each leaf, the engine determines what other subdivided geometries are in view and stores these polygons in the leaf. Abrash gives the analogy of standing in a house to explain how a PVS operates.¹⁵ If someone were standing in a downstairs kitchen, they would not be able to see an upstairs bedroom at the back of the house from their current vantage point. If this house were a *Quake* map, then the set of non-visible polygons that comprise the upstairs bedroom would be excluded from the spatial information stored in the leaf's PVS specific to the kitchen area. At the same time, if one were standing in the kitchen and could see parts of the living room from that viewpoint, then the visible polygons from the living room are stored in the kitchen leaf's potentially visible set. Thus, if the player is located in the corresponding leaf for an area within the map, the PVS for that area contains all of the polygons that might become visible depending on where they move or are looking within the gamespace. The engine only has to include the appropriate polygons in the edge list and render them based on what is visible from a given leaf's PVS, instead of processing thousands of polygons, visible and non-visible, within the map (Figure 3.8).¹⁶ Spatial information from the PVS is interpreted by the edge list as a series of "spans": sections of pixels on the screen that cover the areas of the polygons. Rasterizing algorithms are then able to apply the specific bitmap of color textures to the spans while simultaneously calculating for perspective upon surfaces.

¹⁵ Michael Abrash, "Quake's 3-D Engine: The Big Picture," *Ramblings in Realtime*, *Blue's News*, <https://www.bluesnews.com/abrash/>.

¹⁶ Abrash, "Quake's 3-D Engine: The Big Picture."



Figure 3.8: A PVS of visible polygons arranged within the player's frustum from *Quake* (id Software, 1996).

Yet, this still leaves the question as to how moving objects not included in the static world were drawn. Carmack and Abrash discovered that moving 3D models such as enemies, doors, and platforms, when animated, can encompass multiple leaves within the BSP tree. This was especially a concern because a given model includes hundreds of polygons, with a large portion of them simultaneously visible. The fact that say, a group of encroaching enemies could potentially span several leaves within a given PVS with those leaves changing at each of their movements, makes it more difficult for the engine to properly order polygons from front-to-back.¹⁷ If in *Doom*, enemies were constructed as a series of two-dimensional animated sprites, Carmack designed *Quake* to include 3D figures constructed of a series of vertices covered with a polygonal mesh. He approached this problem through z-buffering, a process by which the coordinates of moving models are measured by comparing the distance of each pixel that

¹⁷ Michael Abrash, "Consider the Alternatives: Quake's Hidden-Surface Removal," *Ramblings in Realtime*, *Blue's News*, <https://www.bluesnews.com/abrash/>.

requires drawing, called its z value, to the distance of pixels within the static world that the moving object would potentially overdraw. These z values for the static world and animated models are stored as units of memory referred to as the z-buffer. Rasterizing will only occur if the z values of the moving pixels are closer to the center of the frustum than those z values of the pixels they overlap in the static world. Z-buffering allows the engine to effectively order moving polygons among those that construct the map regardless of their trajectory or viewed angle.

As Abrash notes with z-buffering, the engine initially rasterizes polygons in the static world and fills only their values in the z-buffer memory without comparing them against the values of moving models. Then moving polygons can be quickly drawn when their z values are read against those already calculated within the static world. Carmack's adoption of a pre-calculated PVS for the static world benefited the z-buffering process in that moving entities were only required to be drawn if they were in a given leaf for the current PVS.¹⁸ Carmack and Abrash also employed z-buffering to approach visual effects like explosions and smoke. Instead of rendering weapon effects as sprites, they simply used large groupings of particles—small z-buffered squares—that would change in scale and color with increasing distance to produce smoke trails, explosions, and blood splatter.

The other technical breakthrough in the engine's development was Carmack's use of a pre-calculated lighting map. To produce detailed lighting and shadows, each polygon was assigned light values with a 16-pixel grid and pre-calculated before the polygon is rendered with bitmap texture. In computer graphics, conventional lighting methods often rely on Gouraud shading, a technique in which light values are assigned at each polygonal vertex in the gameworld. These values are then linearly interpolated or adjusted across polygonal edges to reflect an impression of depth. In this case, texture mapping is applied to each pixel first, and

¹⁸ Abrash, "Quake's 3-D Engine: The Big Picture."

then the interpolated light values are assigned to each texel, or the pixel's correlative point on the texture map. Carmack and Abrash did not use Gouraud shading to light the static world because coupling polygon vertices with light values can result in subdividing polygons into smaller geometries to produce more realistic lighting, subsequently increasing processing requirements for rasterization. Gouraud shading also creates lighting overlaps and discontinuities and requires perspective correction for lighting variations across multiple polygons. Carmack did use Gouraud shading for moving objects because most models are constructed of small triangles and their constant movements obscure any lighting glitches the player may notice.

To light the static world, programmers assigned preprocessed, surface-based light values to each polygonal vertex. A unique lightmap was produced for every polygon composed of a grid with a different light value for 16 x 16 grouping of texels (a textured pixel). For each polygon, lighting is generated by taking the values reflected from adjacent light sources in the map and calculating those values for each grid section on its surface (Figure 3.9). When the game is running, the texture for each polygon is temporarily stored in a buffer as a series of tiles and lighting is produced for each texel by averaging the values of the four closest lightmap areas, creating a tiled surface of lit texels (Figure 3.10).¹⁹ In order to speed up the lighting process, Carmack adopted surface caching so that a given surface could be repurposed in each subsequent frame without having to re-render. Surface caching is used to speed up rendering by reducing the amount of memory needed during runtime. Cache itself is typically a memory reserve comprised of a series of slots that in the case of *Quake's* 3D graphics, store surface textures and repurposes them in subsequent frames. However, Carmack realized that caching would require a considerable amount of memory—in the range of multiple megabytes—to reuse surfaces in

¹⁹ Michael Abrash, "Quake's Lighting Model: Surface Caching," *Ramblings in Realtime, Blue's News*, <https://www.bluesnews.com/abrash/>.

frames. To address the issue, he used mipmapping which lowers memory requirements by subdividing surfaces into four levels that each recede into the distance away from the player's viewpoint. Each mipmap surface possesses a lower resolution the further they recede into the background. Coupled with zero overdraw from BSP ordering, mipmapping allowed for the engine to use less memory during surface caching by sampling only a certain ratio of texels to pixels within different areas of depth in the image.



Figure 3.9: A pre-calculated lightmap for map E2M7 from *Quake* (id Software, 1996).

However, for the *Quake*'s dynamic lighting, or values reflected from moving entities, the engine must calculate light values for nearby polygons that are always changing. If the engine applies dynamic lighting, lightmapping is used prior to buffering and reflected surfaces are calculated accordingly before texture mapping is applied. With lightmapping and surface caching in mind, Carmack designed the engine to support 3D hardware acceleration. In December of

1996, he created a proprietary hardware port called *VQuake* to be used with the popular Rendition Vérité graphics card at the time.²⁰ Adoption of the Vérité chipset was also due its availability as there was not yet a standard 3D API (application programming interface) for computer games in conjunction with video cards. *VQuake*, however, did improve frames per second, supported higher resolutions, and bilinear filtering which reduced the visual effects of pixilation.

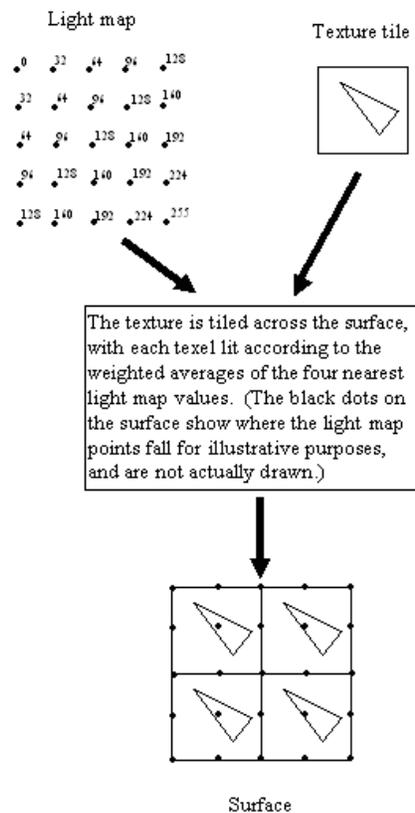


Figure 3.10: A surface rendered by illuminating texels from the pre-calculated lightmap from Michael Abrash, "Quake's Lighting Model: Surface Caching," *Ramblings in Realtime*, *Blue's News*, <https://www.bluesnews.com/abrash/>.

It also supported anti-aliasing, a process in which jagged areas of polygons are smoothed out and straightened, as well as 16-bit coloration, and more responsive dynamic lighting.

²⁰ Abrash, "Quake's 3-D Engine: The Big Picture."

Soon after the release of *VQuake* in January of 1997, Carmack decided to no longer code within the constraints of Rendition's proprietary API, and instead opted to develop the *GLQuake* port for use with the Open Graphics Library (OpenGL), a 3D API that could interpret 3D graphics hardware across multiple platforms and programming languages. The OpenGL API allowed for the graphics processing unit (GPU) on a graphics card to render the game instead of placing the work directly onto the computer's processor, resulting in higher resolutions and framerates, and options to adjust in-game reflections and shadows. Made to execute on Windows 95, *GLQuake* included a driver that allowed the OpenGL API to be used with the popular 3dfx Voodoo Graphics card of the time, of which *GLQuake* was able to demonstrate the chipset's graphics acceleration abilities. Of particular importance to a media archaeology of the *Quake* engine is its client-server architecture. When *Doom*'s multiplayer operated through peer-to-peer networking, Carmack designed *Quake* to operate on a server, with input and output functions occurring on each client computer. Abrash explains that individual client terminals send the player's input data via peripherals (keyboard, mouse, joystick, gamepad) for each frame to the designated server computer which interprets all client input and allows the engine to run the game for a set "timeslice." The timeslice is then sent back to the client as the game data displayed for the next frame after it is processed by the server. This same client-server model is at work in *Quake*'s single player operations, yet the client and server work as one process. The client input is sent to the server component which interprets play and allows the client output to display frames. Communication between client and server occurs through memory buffering. The *Quake* server is responsible for managing the gamestate, physics, and performs AI for enemies. Server data is stored for each map and retrieved from an external database located on the *Quake* disk. Within the server database, a player is able to program new behaviors for

enemies and objects as well as introduce new maps, weapons, and game rules using Quake-C. The client end of *Quake* includes the engine architecture. Once the player provides control input for the game, their gestures are interpreted by the server and sent back as a rendered frame. The same holds true for sounds and in-game menus (Figure 3.11).²¹

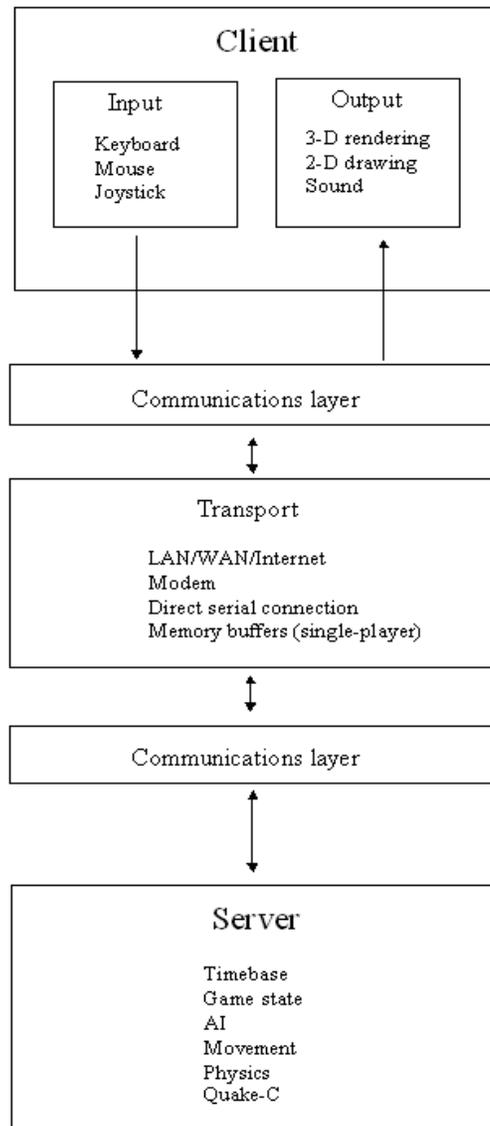


Figure 3.11: Communication between the client and server in the *Quake* engine from Michael Abrash, “Quake’s 3-D Engine: The Big Picture,” *Ramblings in Realtime*, *Blue’s News*, <https://www.bluesnews.com/abrash/>.

²¹ Abrash, “Quake’s 3-D Engine: The Big Picture.”

At this forensic level of investigation, the technical components of the engine’s architecture—BSP ordering, lightmapping, surface-caching, client-server operations—fold into the formal materiality of JODI’s *Untitled Game*. In Kirchenbaum’s terms, the formal materiality of JODI’s mods refer to the player’s experience of their formal processes, the “torque” or “friction” of the engine’s computational layers at work.²² In the following section, I trace these forensic operations to the formal qualities of JODI’s *Untitled Game*, focusing specifically on the mods *Ctrl-9*, *I-N*, and *Q-L*.

The Formal Materiality of *Untitled Game*

Before examining the forensic to formal transition in the code of JODI’s mods, I want to point out the significant work that conservator Lisa Adang has undertaken in providing technical documentation for *Untitled Game* during her tenure as Conservation Fellow for Rhizome in 2013. In *Untitled Project*, Adang examines *Untitled Game* by running the mods on an emulator using Mac OS 9 so as to play them within the context of their original production and exhibition. Specifically, she looks at the *autoexec.cfg* and *config.cfg* files within the *Untitled Game* root directory and compares their command values to the *Quake* command index produced by the Electrical and Computer Engineering/Computer and Information Sciences departments at the University of Delaware. She cross-references the *Quake* commands to JODI’s modified code to see how these altered values produce certain formal effects. Although Adang’s *Untitled Project* focuses on the conservation of *Untitled Game* as born-digital software, I use her strategy of cross-referencing the *Quake* command index with the code of each mod as a point of departure to consider the aesthetic work of JODI’s formal disruptions upon the body.

²² Matthew G. Kirchenbaum, *Mechanisms: New Media and the Forensic Imagination* (Cambridge, MA: MIT Press, 2012), 13.

Inside the *Untitled Game* root directory, there are 14 separate folders with the executable file and game data for each of the 14 mods. Adang divides these mods respectively into three categories based on the manner of abstraction each produces: Op mods (those that produce illusory optical effects through contrasting white and black lines), altered movement mods (those that disrupt the player's ability to move through 3D space), and color block mods (those that obscure the player's visibility with planes of color).²³ The mods I examine (*Ctrl-9*, *I-N*, and *Q-L*) are chosen based on the ways their unique glitches produce anamorphic space that works upon the apperceptive-proprioceptive capacities of the body.

The directory for each mod contains an executable file (.exe application) and a subdirectory titled "id1" which holds the pak0.pak, autoexec.cfg, and config.cfg files, the necessary code in QuakeC and game data interpreted by the engine. Adang draws from "The Quake Wiki," a reference website compiled by players and modders to explain the operations of these game files. The executable file for each mod functions as the *Quake* engine. It interprets the maps, models, objects, and player skins; renders 3D graphics, interprets player input from peripherals, and provides communication between networked server and client.²⁴ The id1 directory contains the pak0.pak file (package file) which is the required compressed file with the data for a given episode of *Quake* that the executable file loads and runs. The "id1" directory and pak0.pak file refers to "Episode 1: Dimension of the Doomed" in *Quake*, an indication that JODI only made modifications to the game's initial episode. Thus, each episode of *Quake* possesses its own package file subsequently named pak1.pak, pak2.pak and so on. The autoexec.cfg and config.cfg files also located in the id1 directory contain commands in QuakeC that are used to program in-game settings and player controls. The config.cfg file for each mod stores game

²³ Adang, "Untitled Project: A Cross-Disciplinary Investigation of JODI's *Untitled Game*", 20, 30, 32-33.

²⁴ "Engines," The Quake Wiki, accessed June 2013, <http://quakewiki.org/wiki/Engines> in Lisa Adang, "Untitled Project: A Cross-Disciplinary Investigation of JODI's *Untitled Game*", 19.

settings, controls, visual information about the player, and their name. The autoexec.cfg file inventories player commands and adjustments to *Quake*'s graphics that are executed automatically by the engine when the game starts.

Although the pak0.pak file in the mod subdirectories includes the sections of the game's code with JODI's modifications, it is partially illegible due to the text editor it is presented in. However, JODI has included sections of the source code on the *Untitled Game* website at www.untitled-game.org/progs/index.html. Here, the artists have displayed *Quake*'s programming as a series of links that describe various objects and entities as QuakeC .qc files (models.qc, world.qc, zombie.qc, etc.). Selecting any of these links displays fully-readable sections of the source code. Adang helpfully points out that one can identify the commands and lines of code that JODI have altered because they have created a file directory labeled "UNTITLED-GAME" with their own comments. For example:

```
/*UNTITLED-GAME/D func_illusionary (0 .5 .8) ?  
A simple entity that looks solid but lets you walk through it.  
*/  
void() func_illusionary =  
  
{  
self.angles = '0 0 0';  
self.movetype = MOVETYPE_NONE;  
self.solid = SOLID_NOT;  
setmodel (self, self.model);  
makestatic ();  
},25
```

Here, it becomes clear as to the ways JODI have altered code to disrupt the formal parameters of space within the game. From the portion of code above, we might guess that the illusionary function of being able to move through solid entities is what causes the player's viewpoint to accelerate and somersault violently through walls and enemies in the *Q-L* mod. More so than the

²⁵ Jodi, "progs", www.untitled-game.org, accessed June 2017, <http://www.untitled-game.org/progsx/miscqc.html>.

code's modified sections, which do not make clear what sections produce the formal disruptions for each individual mod, the code consisting of player commands and variables within the `autoexec.cfg` files interpreted by the source code on startup, tells us significantly more about the formal materiality of *Untitled Game*.

In *Ctrl-9*, or what Adang refers to as an “Op Mod,” the player attempts to move through a three-dimensional space constructed entirely of alternating black and white pixels. The rendered contrast between the black and white background produces an illusory moiré effect. As the player tries to move through the level, the spatial coordinates of the gameworld become indiscernible and instead they must attempt to navigate through the perceptual cues of pulsating checkboard patterns and lines (Figure 3.12). JODI have kept the in-game sound effects and control mechanics intact. The player can still look around, move directionally, jump, and fight in this illusory space, yet they are unable to make out objects or enemies among the psychedelic patterns that comprise the map. JODI have produced this psychedelic effect by taking advantage of the engine's aliasing qualities, which is enhanced by the artists' use of alternating black and white pixels. By aliasing, I refer to the problem in computer graphics when curves within a 3D gamespace appear “jagged” due the engine's inability to render certain areas of a map at a high enough resolution to interpret curved lines. Variations in these jagged lines between frames can produce an optical effect of wavy, oscillating polygonal edges. The perceptual phenomenon of aliasing is often lessened through anti-aliasing filters (Gaussian, box, etc.) that smooth out and/or blur jagged lines. In *Quake*, anti-aliasing filters are supported with the addition of specific 3D graphics acceleration.

From an interview with JODI, Adang explains that they modified the source code so that the map would include contrasting rows of pixels that are individually defined in black and

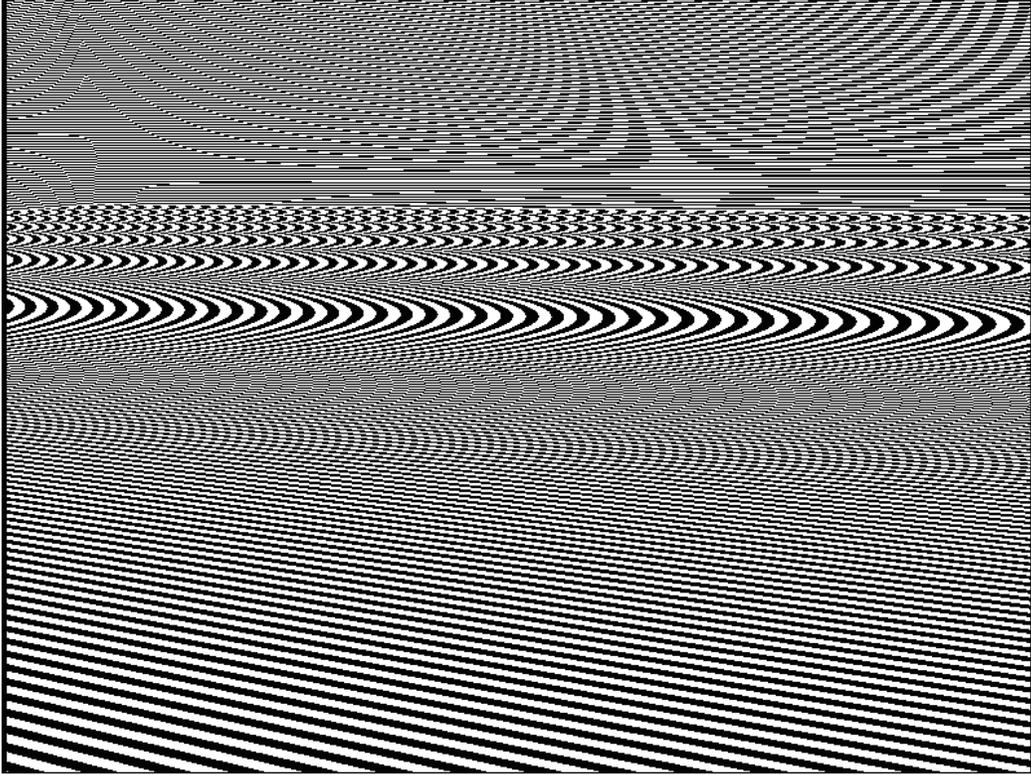


Figure 3.12: JODI's use of aliasing effects in *Ctrl-9* (JODI, 1996-2001).

white. The game engine attempts to draw these black and white pixels based on individual rows using scanline rendering, which in *Quake* is an algorithm used in visual surface determination. In conjunction with scanline rendering, z-buffering performed by the engine—the process through which moving objects and entities can be drawn by measuring the distances of moving pixels against pixels in the static map—also generates aliasing effects when the engine works to coordinate and draw the black and white textures to polygons.²⁶ As the player moves through the black and white areas, the continual adjustment of their frustum prompts the engine to draw each new PVS. When the engine draws and scans rows of black and white pixels for each PVS in addition to z values for moving entities, it continually generates undulating moiré patterns based on player navigation.

²⁶ Joan Heemskerk and Dirk Paesmans interviewed by the author, May 30, 2013, Dordrecht, Netherlands in Adang, “Untitled Project: A Cross-Disciplinary Investigation of JODI’s *Untitled Game*”, 30.

In the formal materiality of *I-N* and *Q-L*, what Adang calls “Altered Movement Mods,” the player is confronted by the perceptual disorientation of being unable to direct themselves effectively through the gamespace. In *I-N*, the player’s field of vision is obscured by a thick white line that runs horizontally across the length of the screen. When the mod starts, the player’s viewpoint rotates wildly, causing the white line to spin and generate an afterimage through its flicker. By adjusting the view angle or moving in a different direction, the player can momentarily stop the spinning camera only for it to erratically start again moments later (Figure 3.13). JODI have also altered sound effects to a series of static hums, blips, and the whirring of electrical frequencies. The spinning camera angle and afterimage of the flickering white line is generated by values that have been changed in the mod’s `autoexec.cfg` file:

```
sensitivity 12//Default: 3  
Indicates mouse sensitivity
```

```
r_clearcolor 256//Default: 2  
Sets colors for certain areas outside the map
```

```
cl_rollangle 444//Default: 2  
Indicates how much the screen will tilt if the player strafes.
```

As Adang already points out in her discussion of the “Altered Movement Mods,” the most significant change in JODI’s code is the increased value of the `cl_rollangle` command to 444.²⁷ Thus when the player strafes left or right, the screen tilts to such a degree that it frantically rotates the player’s entire viewpoint. The increased value of `cl_rollangle` coupled with the sensitivity command increased to 12 could attest to the erratic feedback of the player’s camera as they move through space. The formal construction of the thick, white line could be the result of increasing the `r_clearcolor` function to 256, in turn, altering areas of color outside the map yet still visible in the player’s immediate perspective.

²⁷ Adang, “Untitled Project: A Cross-Disciplinary Investigation of JODI’s *Untitled Game*,” 33.

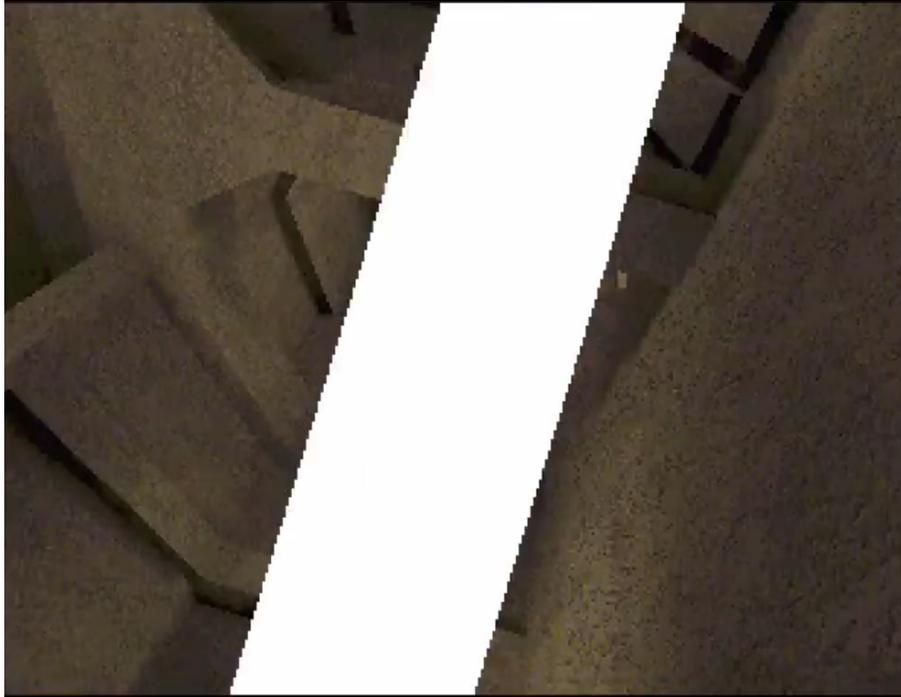


Figure 3.13: The rotating white line and flicker effect in *I-N* (JODI, 1996-2001).

Here, JODI have taken strafing—the player’s ability to move laterally through space—as a mechanic used to great effect within experienced *Quake* multiplayer communities and defamiliarized its strategic importance. Instead of expertly strafing or strafe jumping (jumping with simultaneous lateral movement) to dodge enemy fire, the player’s movement is transposed into the formal patterns of the flickering screen and the chaotic rotation of the white line.

More so than *I-N*, *Q-L* is perhaps the most spatially disorienting of the *Untitled Game* mods. In *Q-L*, spatial navigation through 3D space has completely eroded. The player’s viewpoint rapidly spins and flips in all directions, at times accelerating forward and backward through the map while moving through walls and enemies (Figure 3.14). Like *I-N*, this erratic movement generates an afterimage produced from the flickering between frames. In the upper portion of the screen, there are several lines of characters and numbers whose values change based on the player’s moving spatial coordinates.

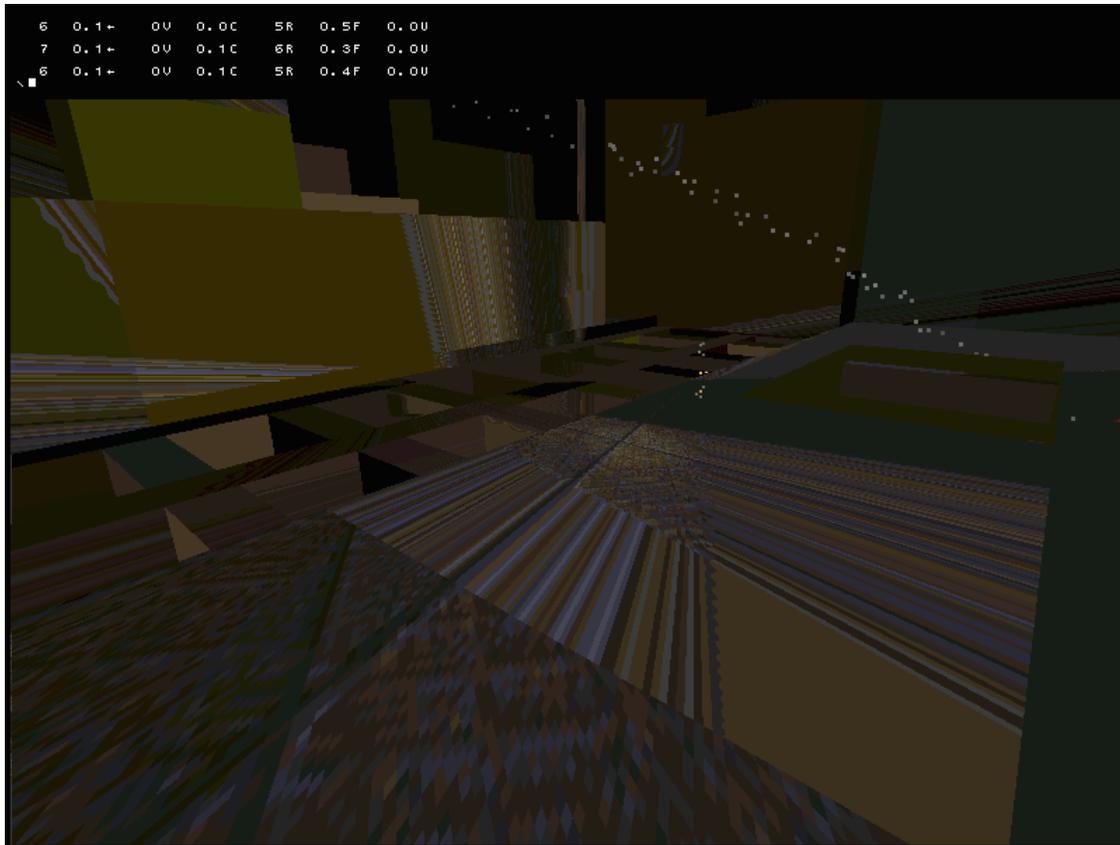


Figure 3.14: The camera accelerating and summersaulting through walls in *Q-L* (JODI, 1996-2001).

It is difficult to gauge if the player’s controls ground their movement within the space or if the player can control elements of the rotating and accelerating camera angle. The mod’s frenetic somersaulting can be explained as the altered values within the `autoexec.cfg` file documented by JODI on the *Untitled Game* website, which appears as the total list of commands for all of the *Untitled Game* mods. In this `autoexec.cfg` file, the `cl_rollangle` value is expressed at 844 which could, in part, explain the kaleidoscopic output produced by the code.²⁸ In what follows, I explore how the formal elements of glitch in *Ctrl-9*, *I-N*, *Q-L*—the way in which the altered code in *Untitled Game* outputs spatial disorientation via anamorphosis—works aesthetically upon the player through an apperceptive-proprioceptive resonance with *Untitled Game*.

²⁸ JODI, “autoexec.cfg”, [www.untitled-game.org](http://untitled-game.org), accessed June 2017, <http://untitled-game.org/autoexeccfg.html>.

Flicker Mods and the Modulation of Bodily Space

In the case of *Ctrl-9*, *I-N*, and *Q-L*, JODI use glitch to defamiliarize the player's conventional experience of gameplay. Daniel Morgan observes a similar effect of defamiliarization in avant-garde cinematic works, namely director Jean-Luc Godard's use of slow focus pulls. In Godard's *Soigne ta droite* (1987), he uses extended focus pulls to, for instance, bring the blurry shot of a post with barbed wire against the backdrop of the sky, slowly into focus. The abstraction of the blurry shot appearing over an extended period directs the viewer to the way in which the camera lens produces the formal dimensions of the film. The very strangeness of this blurred image that slowly becomes a recognizable shot provides a mode of apperceptive engagement that counters our standard habits of viewing film, revealing to us the very act of perception.²⁹

Similar to Godard's use of the focus pull, I see glitch in JODI's work as a formal technique that prompts gamic apperception through what I call the flicker mod. The flickering abstractions of glitch within the frames of *Ctrl-9*, *I-N*, and *Q-L* make strange our experience of gameplay and affirm the means by which our perceptual faculties operate within actionable 3D space. Here, glitch prompts the body to apprehend its own instability within 3D space through an alien landscape of data patterns that operate outside the player's capacity to frame the technical conditions of the modification. This is what Rosa Menkman has called the *moment(um)* of glitch: a moment that produces a feeling of the uncanny or unfamiliar, an inability to control or understand the computational processes at work. At the same time, this phenomenological encounter with the machine inherently carries a momentum in which a noise artifact "turns to glitch when it passes a momentary *tipping point*, at which it could tip away into a failure, or

²⁹ Daniel Morgan, *Late Godard and the possibilities of cinema* (Berkeley, CA: University of California Press, 2013), 45, 50.

instead force new knowledge about the glitch's techné, and actual and presumed media flows, onto the viewer."³⁰ In an interview with artist Francis Hunger, JODI have reflected upon this instance of the glitch's apperceptive revealing of *Untitled Game* through what they call a "perspective engine":

What our interest in modifying the game was, was that we always worked in the style of abstraction. Games in general are the total opposite from that. They have a narrative and they are very figurative, for example, Quake monsters they are like caricatures. They are so figurative to have some dramatic impact on the audience – let's say, a screaming monster with a lot of fire and shooting beasts or meat or blood. So they are very explicit graphically and what we wanted to do was in general to erase the story and the figurative site of these games. The starting idea was to find very basic forms like just a line or a square, just black and white, and attach these forms to the behaviour of the code so that we would have a better view on how such a game is driven, what are the dynamics of the game. So it's bringing those games back to the abstract dynamics of it and we were also trying to find out a little bit, how they do create the so-called 3-D space. That's the whole trick of these games, that they are perspective engines. All the time they create tunnels and illusions of a 3-D space and that's part of the "kick" you have as the user, that you think you explore and you enter and you move into. In fact, the only thing which is happening, is a perspective which just is drawn all the time – so it's always about graphical tricks.³¹

In *Ctrl-9*, *I-N*, and *Q-L* glitch reveals the perspective engine that grounds the player's experience of 3D space through a series of graphical abstractions. For instance, within the phenomenological moment of glitch that overwhelms the player in *Ctrl-9*, the body's apperceptive capacities intuit the undulating moiré patterns as something that at once proprioceptively unfixes the player from perspectival rendering and makes known the perceptual vectors through which vision operates within the *Quake* engine. JODI's code generates aliasing effects with these black and white lines that upend the functionality of the engine's visual surface determination through scanlines and z-buffering, producing an optical effect that contradicts habitual perception and reveals the bare-bone mechanisms which make the perspective engine

³⁰ Rosa Menkman, *The glitch moment(um)* (Amsterdam: Institute of Network Cultures, 2011), 30-31.

³¹ JODI quoted in Francis Hunger, "Perspective Engines: An Interview with Jodi," in *Videogames and Art*, ed. Andy Clarke and Grethe Mitchell (Bristol, UK: Intellect Ltd., 2007), 152, 156.

run. In JODI's work, glitch aesthetics are indebted to a tradition of 20th century American avant-garde cinema, particularly within discourses of minimalism and structural film. P. Adams Sitney discusses the concept of structural film in the work of experimental filmmakers such as Hollis Frampton, Michael Snow, and Tony Conrad throughout the 1960s. The "structural" element of their work relates to the viewer's perception of the totality of the film's form and structure, or its overall shape, which unfolds through its duration. Sitney mentions that these artists' experimentations in foregrounding the materiality and form of cinema, worked to more immediately arouse in the viewer deeper levels of consciousness and affective experiences of the brain-body. This becomes evident in the structural "flicker film," a radical instantiation of film form that attempts to produce a direct psycho-physiological response from the stroboscopic effect of frames projected at irregular intervals.³² This flicker process that defamiliarizes film's technical illusion of animating continuous movement is most evident in the work of Tony Conrad's *The Flicker* (1966). As I have mentioned, Conrad's film is composed of alternating black and white frames that produce a hallucinogenic effect. The film runs between 24 down to three transitions of black to white frames per second utilizing 47 different patterns. Conrad photographed these patterns multiple times, exposing clear frames to white paper and shot with the lens cap on for each black frame. He then meticulously spliced the patterns together so that when projected, the film would produce approximately 30 minutes of stroboscopic patterns (Figure 3.15).

Much in the way glitch mobilizes apperceptivity in the *Quake* engine, Sitney argues that the predetermined shape of Conrad's film is built around an apperceptive experience of its material conditions that conflict with the "illusionary participation" of narrative cinema. As he

³² P. Adams Sitney, *Visionary Film: The American Avant-garde, 1943-2000* (Oxford: Oxford University Press, 2002), 285, 305-306.

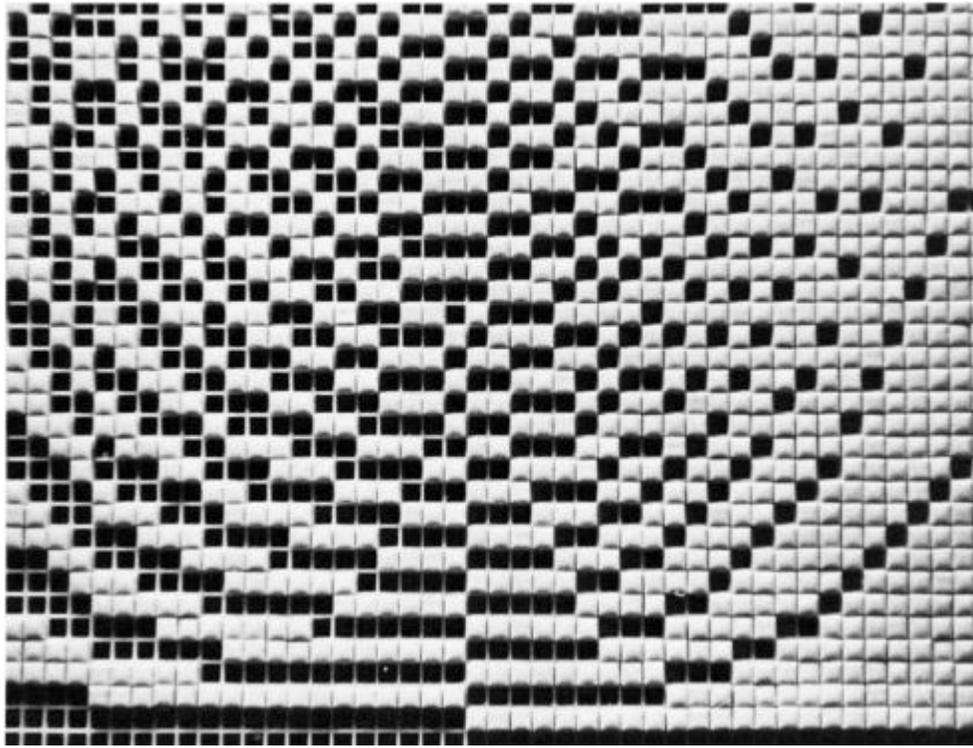


Figure 3.15: Arranged rhythms of black and white frames in *The Flicker* (Tony Conrad, 1966).

says, there is a sense of “transcendentalism that is always grounded in a dialogue between illusion and its unveiling,” or rather the hallucinogenic shock of flickering variations and the awareness of formal transitions between the projected arrangements of frames.³³ As a leading historian of American avant-garde cinema, Sitney situated his critique of Conrad’s flicker aesthetic appropriately within the defamiliarization inherent to Shklovsky’s Russian formalism. To Sitney, the viewer’s experience of film art relies on a process of perceptual disassociation or prolongation that interferes with and thus alters the visual perception of what we can consider an affective state of apperception.³⁴

³³ P. Adams Sitney quoted in Branden Wayne Joseph, *Beyond the Dream Syndicate: Tony Conrad and the arts after Cage* (New York: Zone Books, 2008), 297.

³⁴ Joseph, *Beyond the Dream Syndicate*, 297.

In his minor history of Conrad's work, Branden Joseph examines the phenomenological play of *The Flicker* as a film that produces a mental indeterminacy. It balances the viewer's engagement between an apperceptive awareness of the film's stroboscopic effects in which they remain in control of their perceptual responses, and affective operations through which they lose control to the body's autonomic activity. This indeterminacy interrogates our autonomy as a viewing subject while simultaneously affirming our subjectivity. Joseph explains that the viewer's experience of the flickering rhythms of black and white asserts a partial dissolving of subjectivity when taken up in the technics of the projector shutter and film. They are at once on a threshold of becoming subsumed into a circuit between the alternating flicker and the body's psycho-physiological feedback, yet they are also perceptually aware of the body's position at this threshold. That is to say, *The Flicker* counters the normal perception of film by engaging viewers in a technically-conditioned space of sensory overstimulation, what Joseph calls "a transgressive and indeterminate threshold experience" and "an instance of positive feedback that induces wild perceptual oscillations."³⁵

When *The Flicker* premiered at film festivals and underground theaters in 1965-66, the film was preceded by an on-screen disclaimer waiving liability for potentially ill side effects such as possible seizures and "symptoms of shock treatment."³⁶ During these initial screenings, the film subjected audiences to its stroboscopic effects with responses ranging from the witnessing of color arrangements, geometric patterns, amorphous floating forms, and complex hallucinations. In some instances, audiences found the flickering patterns to be calming and

³⁵ Branden Wayne Joseph, *Beyond the Dream Syndicate: Tony Conrad and the Arts after Cage*, 298-300, 340.

³⁶ See Figure 6.4 *Warning*, in Branden Wayne Joseph, *Beyond the Dream Syndicate: Tony Conrad and the arts after Cage*, 288.

relaxing, while others experienced heightened states of anxiety, particularly in response to being under control of the flicker through a hypnotizing effect.³⁷

I see JODI's use of glitch in *Untitled Game* as similar to Conrad's flickering frames in that both fundamentally explore the aesthetic capacities of a technical medium through the body's apperceptive response. JODI discusses their influence of experimental film stating that:

...And another film artist Joan and I recently looked at is Stan Brakhage, who seems to be a very important person in experimental film...Our sympathy goes to experimental film, because they managed to stay in their own medium. We use both these fields quite a lot to think about what we are doing or what would net.art or digital art be.³⁸

As a pioneer of American avant-garde cinema, Brakhage's work preceded the structural movement, yet similar to filmmakers like Conrad, his films explore ways in which the materiality of the film strip, projector and camera can produce novel modes of perception to contradict dominant film aesthetics and ideology. It becomes apparent as to why JODI would find resonance in the work of Brakhage and the tradition of experimental film and video art during the 1960s and 70s. To recall, Paesmans studied under Nam June Paik and both Heemskerk and Paesmans have previously expressed their admiration for the video works of Bruce Nauman.³⁹ Like the flicker film, JODI's use of glitch produces a threshold moment that propels the player into a state of apperceptive strangeness. They at once recognize their inability to control movement through 3D space and experience a dematerializing of the self within the circuit of the mod and body: a loss of control to the incomprehensibility of the code from which anamorphic space is produced.

³⁷ See Joseph's explanation of audience's various perceptual and affective responses of *The Flicker* in Branden Wayne. Joseph, *Beyond the Dream Syndicate*, 341.

³⁸ JODI quoted in Francis Hunger, "Perspective Engines: An Interview with Jodi," in *Videogames and Art*, 159.

³⁹ See JODI's discussion of Bruce Nauman in Francis Hunger, "Perspective Engines: An Interview with Jodi," in *Videogames and Art*, 158.

In *I-N*, this threshold is reached when the player is positioned at the boundary between an awareness of the glitch overcompensating for the strafing mechanic and the rotating vortex generated from the tilting screen and spinning white line. The rotating line against the textures of the map's background that scintillates with the frame rate, propels the body into a circuit among the output of stroboscopic patterns and the player's efforts to ground themselves within the dizzying gyrations of the interface (Figure 3.16). In this circuit, the player's bodily experience is modulated by the glitch and incomprehensibility of the game controls. As they feel out a possible control scheme to stabilize their vision within the flickering rotation of the screen, they output a continually evolving rhythm from the redrawing of each subsequent frame that, in turn, impresses upon their sensory faculties. Yet, it is JODI's *Q-L* mod that best demonstrates the apperceptive moment of the flicker effect. Upon start up, the player is immediately assaulted by the shifting, anamorphic landscape of the glitch. The body is launched backwards and forwards through walls and enemies while simultaneously orbiting out of control. The player finds



Figure 3.16: Stroboscopic patterns generated from the rotating screen in *I-N* (JODI, 1996-2001).

themselves frantically working against the control input, pressing keys and moving the mouse/joystick, hoping to discover some permutation of button mashing that might stabilize their somersaulting and acceleration across the map (Figure 3.17). It is difficult to say whether the player has any control over the outcome of the glitch at all. Like the strafing in *I-N*, what little if any control mechanics initiated by the player in *Q-L* are entirely oversensitive. Navigation is overcompensated as the player's viewpoint is hurled chaotically around the level, generating a violent kaleidoscope of patterns and further imbricating the player within the strange, agentic presence of the glitch. If the structural shape of Conrad's film operates through an orchestrated sequence of frames, then glitch in the flicker mod operates through the deliberate aimlessness of JODI's code. Glitch in the flicker mod produces an aesthetic environment that is at once a moment of sensory assault, yet through its technics, mobilizes the body's center of

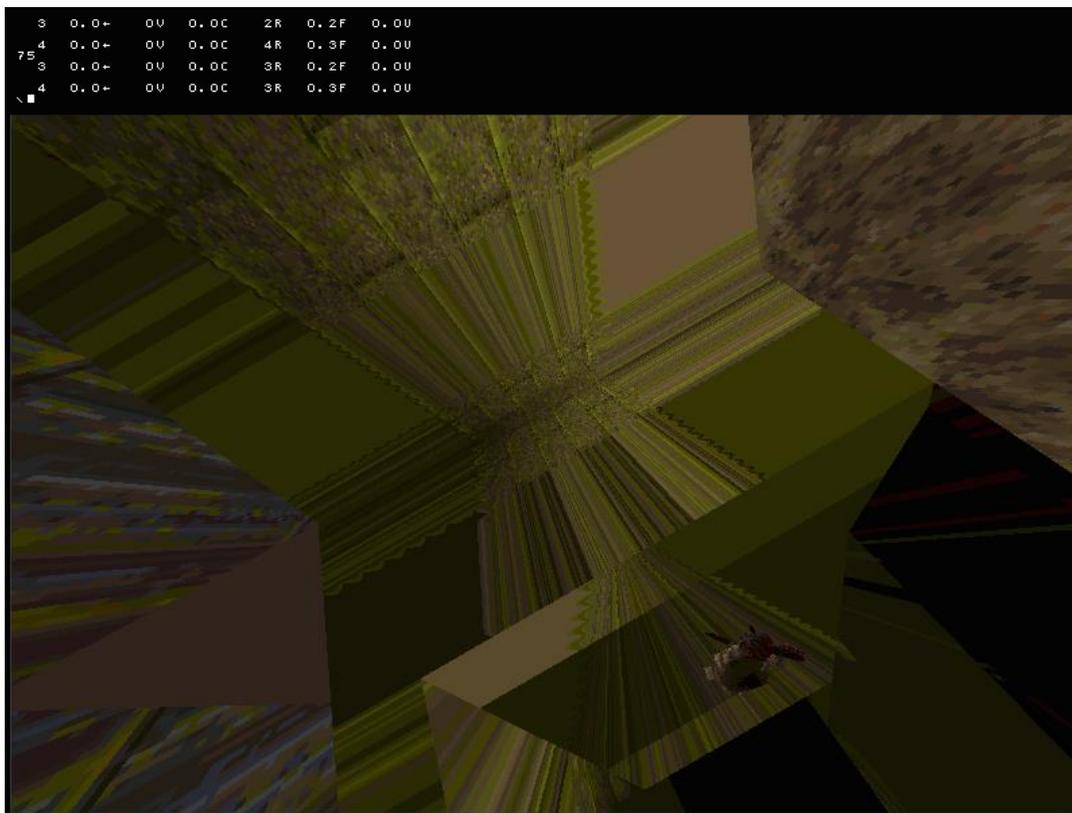


Figure 3.17: Spinning and accelerating through the polygonal architecture of the map in *Q-L* (JODI, 1996-2001).

indeterminacy. The glitch and its potential to generate the new through a ruined functionality is mirrored within the body as an *affective unpredictability* that at once counters our habitual perception of *Quake* and probes the body's proprioceptive capacities.

Within the player's threshold experience of *Ctrl-9*, *I-N*, and *Q-L* proprioception operates as a phenomenological modality of the strangeness of our apperceptivity. The defamiliarization of 3D space attunes perception, as intuiting the act of perception in itself, to the body's intensive capacities to orient itself within the gamespace and more broadly, the world. It relates to the body's apperceptive awareness of its own spatial bearings within the game engine and world through intensive operations that work beyond the exteroceptivity of vision. In his discussion of Godfrey Reggio's experimental film *Koyaanisqatsi* (1982), Scott Richmond explores how cinema, and specifically the cinematography of Reggio's film, opens up the viewer's proprioceptive relation with the world and yet reflexively points to the body's own spatial sensitivity as a subject constituted within the world.⁴⁰ The film implicates the body's dual capacities as a mutually-constitutive subject and object, what Maurice Merleau-Ponty calls the chiasmus of our embodied flesh in the world, through its ability to attenuate and magnify our proprioceptive experience. Merleau-Ponty tells us that the body is a subject that senses the surrounding world but is also an object that can be sensed. Our experience is merely the chiasm or our dual capacities as a sensing subject and sensible object.⁴¹ The film expresses our chiasmus of experience through the unique illusions of motion in Ron Fricke's brilliant cinematography that bears to us sequences of abstract space documented within both the natural world and the industrialized metropolises humankind has given rise to.

⁴⁰ Scott C. Richmond, *Cinemas bodily illusions: flying, floating, and hallucinating* (Minneapolis: University of Minnesota Press, 2016), 98.

⁴¹ Maurice Merleau-Ponty, "The Intertwining- The Chiasm," *The Merleau-Ponty Reader*. Edited by Ted Toadvine and Leonard Lawlor. Evanston, IL: Northwestern University Press, 2007, 395, 402.

Drawing from Merleau-Ponty, Richmond suggests the kinesthetic sensations that the film elicits affirms the body's phenomenal status through our proprioceptive sensitivity to the disorientation among subjective movement within the body and the objective motion of the film.⁴² Richmond discusses the chiasmus between objective motion and our proprioceptive registers through what Merleau-Ponty calls the gap or *ecart* of the flesh. The *ecart* demonstrates our phenomenal becoming as the unique separation among the body's capacities to sense the world around it without being entirely and objectively incorporated into the world.⁴³ When we encounter *Koyaanisqatsi* and are introduced to its abstract spaces, the film interrogates the *ecart* as the body's proprioceptive sensing of the world and sensation of itself. Our presence in the film's kinesthetic illusions, especially in sequences of extreme acceleration and motion blurring through city streets and desert landscapes, produces a pleasurable moment of exhilaration on the threshold of a nauseating transgression of the body's perceptual limits (Figure 3.18).



Figure 3.18: Abstract sensations of accelerating through space in *Koyaanisqatsi* (Godfrey Reggio, 1982).

⁴² Richmond, *Cinemas bodily illusions: flying, floating, and hallucinating*, 107.

⁴³ See Richmond's discussion of Merleau-Ponty's concept of the *ecart* in Maurice Merleau-Ponty, *The visible and the invisible: Followed by working notes*, ed. Claude Lefort, trans. Alphonso Lingis (Evanston: Northwestern University Press, 1968) in Scott C. Richmond, *Cinemas bodily illusions*, 110-111.

Yet, different from Richmond's assessment of *Koyaanisqatsi*, JODI's flicker mods *continually* attune the body's proprioceptive sensitivity to itself and world through the projection of abstract spaces. When *Koyaanisqatsi* grounds the viewer in cinematographic sequences of representational gestalt, before transitioning to the disorienting acceleration through abstract space, JODI's flicker mods never fully ground the player's spatial equilibrium. Much like Conrad's film, they interrogate our spatial awareness through an on-going resonance with the abstract forms as well as the body's immanence within worldly space. In the body's encounter with *Ctrl-9*, *I-N*, and *Q-L*, the ecart manifests as an embodied operation that is fundamentally mimetic. It is the reversibility of the viewer's on-going attunement toward the sensibility of worldly space through the objective motions of the flickering glitch and the sensing body's proprioceptive acclimation to anamorphosis. Here, the body's sensation of the world is in constant modulation as it attempts to work through an unintuitive control schema and navigate the strangeness of the gamespace. At the same time, the flicker of the glitch impinges on the player and continually works upon our own bodily affectivity and on-going self-reception to the disruption of perspectival rendering. Within the flicker of *Ctrl-9*, *I-N*, and *Q-L*, the player's movement through anamorphic space results in an unfolding, as Richmond suggests, of the body's continual orientation of its own schema in relation to the world. This describes the ways in which the body's physical arrangement in space is imbricated in its continual proprioceptive responsiveness.⁴⁴ Thus, *Ctrl-9*, *I-N*, and *Q-L* work upon the player's spatial equilibrium through the body's exteroceptive response to the erratic nature of the glitch—the fingers looking for working controls, minute twitches and jerks, the dilation of the pupils in response to the flickering frames—and upon the body's affective proprioception, or the kinesthetic sensations of rotating, accelerating, and undulating through the gamespace. Merleau-Ponty refers to this

⁴⁴ Richmond, *Cinemas bodily illusions: flying, floating, and hallucinating*, 116.

synthesis between the body's actual physical stature and the intensive measure of space as "bodily spatiality." This describes the player's embodied presence in the world as "the deployment of the body's bodily being, the manner in which it comes into being as a body." Merleau-Ponty examines the synthesis between our physical exteroceptive posturing and proprioceptive capacities, describing it as the way "I can feel my body's space as vast or minute despite the evidence of my senses, this is because there exists an affective presence and enlargement for which objective spatiality is not a sufficient."⁴⁵ In *Untitled Game*, bodily spatiality is the continual development of our body's composure that is at once our physical receptiveness to the game system and our affectivity as a proprioceptive response to the visual patterns of glitch.

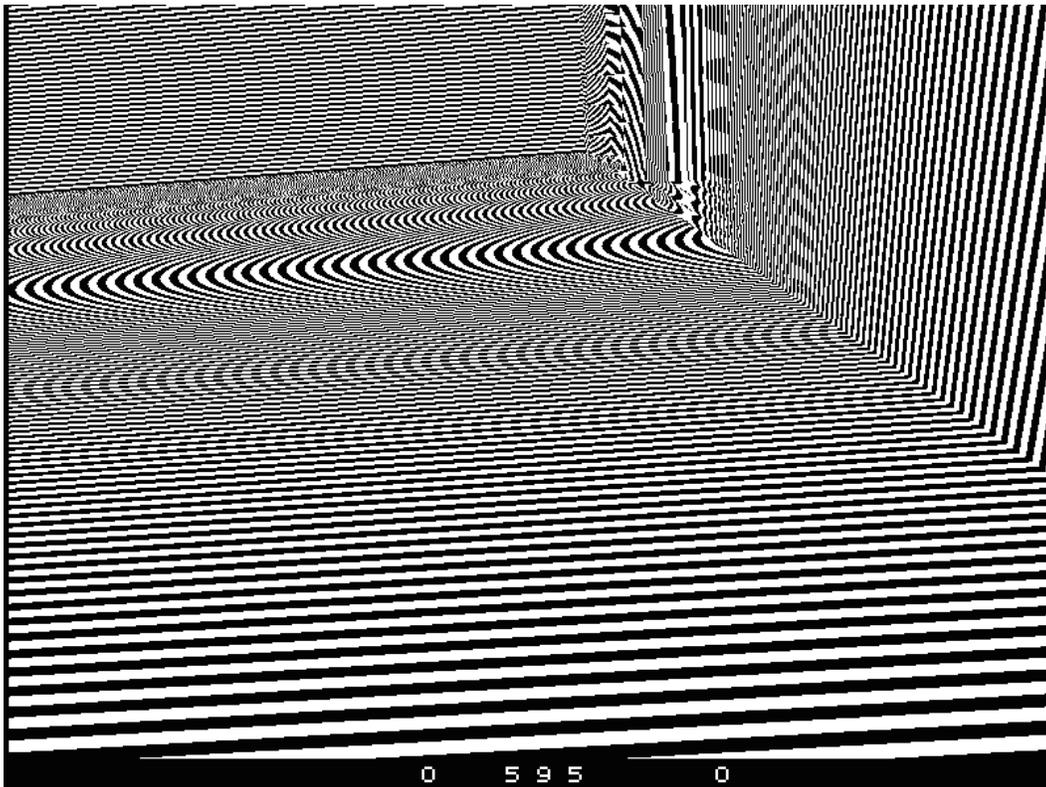


Figure 3.19: Pulsating moiré effects caused when the player moves toward edges and corners of the map in *Crtl-9* (JODI, 1996-2001).

⁴⁵ Maurice Merleau-Ponty, *Phenomenology of Perception*, trans. by Colin Smith (London: Routledge, 2005), 172.

JODI's *Ctrl-9* works upon the player's bodily spatiality through the instability of the aliasing effect that contracts and expands illusory space produced through the warping black and white lines. This is especially the case when the player attempts to locate themselves within the map by walking toward corners and edges of polygons (Figure 3.19). When a player moves too

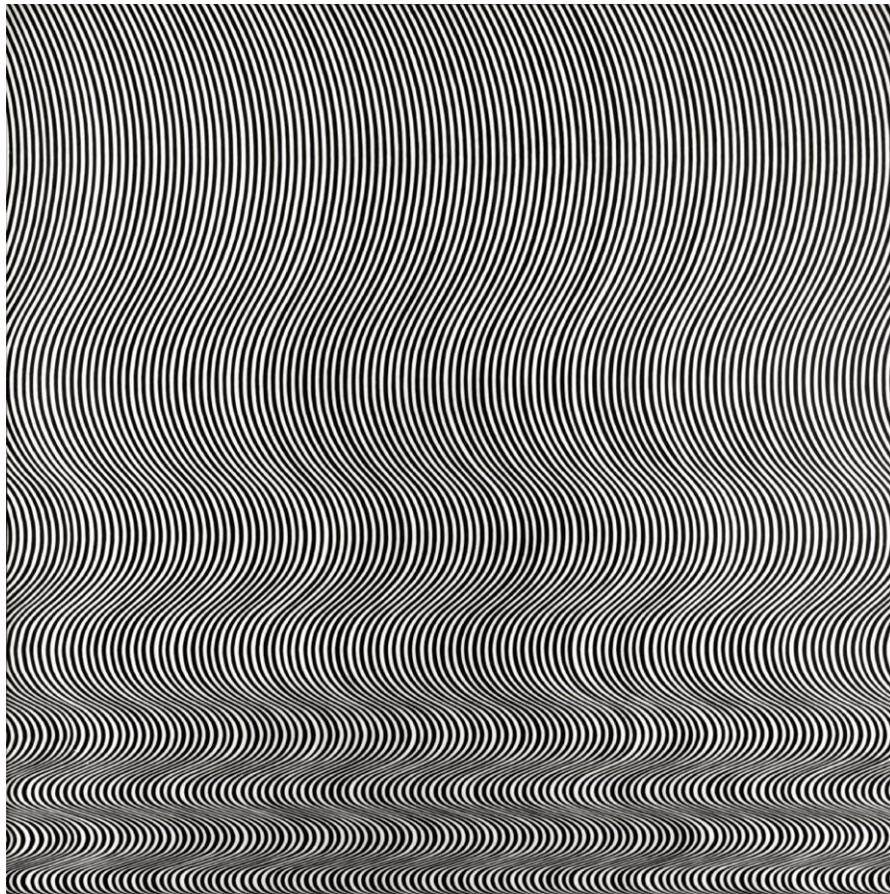


Figure 3.20: Optical, illusory patterns in *Fall* (Bridget Riley, 1963).

close to a surface, it causes black and white patterns to pulsate and warp the polygonal surface. When the player turns to move away from a wall in a different direction, these patterns transform into vectors of pulsating lines that produce an ongoing moiré effect. Here, glitch in *Ctrl-9* modulates the body's physical composure as it is positioned in worldly space at the same time it heightens our proprioceptive response to anamorphic forms: illusory contraction and expansion

of line and shape. This comes as a sensation of a *pull* upon the vestibular system that temporarily eschews our spatial equilibrium. In *Ctrl-9*, JODI borrow from an aesthetic previously explored in the Op Art movement of the 1960s and particularly the work of Bridget Riley. Riley’s paintings generate optical illusions through the use of contrasting black and white geometries that appear to vibrate and swell. Her technique of adjusting the scale of line and shape that we see, for instance, in her paintings *Movement in Squares* (1961) and *Fall* (1963) sought to interrogate a range of embodied responses through what Pamela M. Lee calls a “keenly felt physicality, even a dangerous physicality,” operating through the viewer’s kinesthetic response (Figure 3.20). The illusion of movement in Riley’s compositions, a sensation “falling or rushing or even

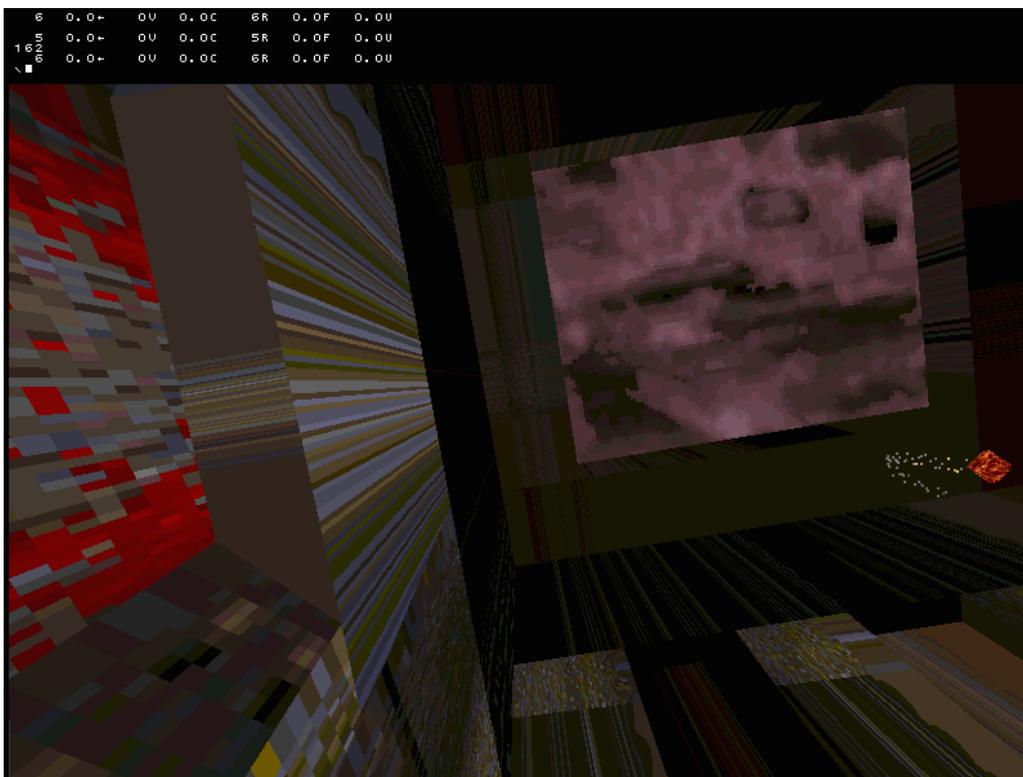


Figure 3.21: The player’s wildly spinning viewpoint and an exploded piece of an enemy’s body (far right) in Q-L (JODI, 1996-2001).

concentrated pacing,” situated viewers in a state of optical-bodily instability: a sensory stimulation that elicited vertigo and nausea in some while exciting a pleasurable “rush” in others.⁴⁶

Just as *Ctrl-9* modulates our proprioceptivity as a “keenly felt physicality,” *I-N* and *Q-L* through their more pronounced flicker effect, engage our bodily spatiality in the illusion of a violent acceleration through space. In *I-N*, the stopping and starting of the rotating line and its acceleration into a flicker is mirrored in the body as a churning or lurching to the left or right. It describes a kinesthetic sensation of becoming unhinged from our physical positioning in space, as if our actual posture was about to tilt with the screen. Similarly, *Q-L* takes up the body’s schema in a violent kinesthesia. As our viewpoint spins and accelerates backwards throughout the level, tearing through the flesh of an enemy and causing his body to explode, the player intuits anamorphosis as a violent intensification of space-gone-awry, of the engine imploding upon itself in its attempts to continually render the player’s constantly shifting frustum (Figure 3.21).

In deconstructing *Quake* and its “perspective engine,” JODI’s modifications guide our attention toward the means through which dominant forms of gaming technicities are historically organized around perceptual operations of navigating a three-dimensional Cartesian grid of rendered space. In other words, by rearticulating the perspective engine of *Quake* as a strange abstraction of space, JODI affirm our reliance on its technical operations to render perspectival space and its ideological framework for producing optimal flow through which player behavior is motivated to act, advance, and overcome obstacles within this space. With the defamiliarizing of real-time 3D space comes the severance of flow. Within this bizarre instance, a moment in which the perspective engine is everted, flow is channeled into an aimless encounter, producing

⁴⁶ Pamela M. Lee, *Chronophobia: on time in the art of the 1960s*(Cambridge, Mass: MIT Press, 2006), 164, 173.

an experience that is at once exhilarating and at times perhaps even masochistic. This phenomenal interaction with *Untitled Game* demonstrates the aesthetic possibilities for what art modding can do within this historical instantiation of the game engine. That is, to creatively engender a bodily experience that emerges from the strangeness of 3D space.

CHAPTER FOUR

Generative Mods and the Violence of Sensation

...the drive to hunt and kill is abstracted and used as control data to drive the painting process...to turn th[e] fight into something unrecognizable and perhaps even beautiful; a transduction of the hunt into a cumulative stream of graphic memory.

-Julian Oliver “Delire”

Julian Oliver known by his pseudonym Delire, created *ioq3apaint* as a generative “painting” program that utilizes the graphics renderer and bot AI of the *Quake III: Arena (Q3A)* engine to produce a continuous mutation of glitch effects. The first iteration of what Oliver refers to as a “Game-based Automatic Painting System” started in 2002-03 with *q3aPaint*, a work that used a modified bot AI and “redraw” glitch within the *Q3A* engine’s z-buffering. The result transformed the game’s recognizable arena into a continually-changing terrain of abstract patterns, painterly lines, and taches of color.¹ While artist-in-residence at Georgia Tech in 2007, Oliver developed an updated version of his painting mod called *ioq3aPaint*, which sought to use the improved features of the open-source *ioquake3* engine. The newer iteration extended the work’s generative possibilities by allowing the user to alter color palettes in real-time. Oliver’s completed project was shown in a 2010 solo exhibition at the Hannah Maclure Centre in Dundee Scotland, consisting of prints created from still frames of the modification as well as a real-time demonstration.

Like JODI’s *Untitled Game*, Oliver disrupted the engine’s graphics renderer using console commands and then configured the AI so that bots—enemies programmed to think and act autonomously—hunt each other without player intervention. Instead of fighting against or alongside the engine’s AI, the user simply moves from the current viewpoint of each bot during combat, employing their machinic perspective, as Oliver says, “to select an ocular host to assist

¹ Julian Oliver, “ioq3aPaint,” *julianoliver.com*, <https://julianoliver.com/output/ioq3aPaint>.

the painting process.” Each painting unfolds through the co-constituted involvement of the engine’s redraw operations and the user choosing the appropriate camera shot, or rather the bot’s vision of the gameworld. As Oliver states, each painting “is really a user’s record of single moment in a continuous stream of color and gestural information.”² Occasionally, a user will capture the simultaneous viewpoints of two bots in one frame, producing within the interval a bleeding of abstract forms. This strange ecology of human and bots, produces images based on a “hall of mirrors” effect common in glitch aesthetics. This typically describes when moving objects trail across the screen and draw over older frames that are not refreshed. Oliver suggests that the trace of moving bots throughout the arena is a transference of the game’s simulated killings into a pure abstraction of violence.

Just as Oliver’s *ioq3apaint* abstracts the *Q3A* deathmatch into graphic trails of line and shape, artist Tom Betts, known alternatively as Nullpointer, likewise altered the *Q3A* engine to produce his mod *QQQ* in the same year. Yet, different from Oliver’s re-articulation of *Q3A*, Betts produces a generative system utilizing a live multiplayer server and modified graphics engine that glitches networked play in real-time. Deathmatch bouts on a server created by the artist are unknowingly transferred and projected within an installation where viewers witness glitched sequences of motion trails left by online players and the deforming architecture of the arena they battle within. Using several keyboard controls positioned on a pedestal, the viewer assists in deconstructing the interface by frantically moving the camera, further generating patterns of glitch. Our experience of the abstract forms in *QQQ* describes a complex field of interaction among the game’s networked infrastructure, remote players, altered graphics engine, and viewer who bears witness to the palimpsestual frames that continually haunt the interface.

² Julian Oliver, interview by Domenico Quaranta, “Delire (Julian Oliver),” in *Gamescenes: Art in the Age of Videogames*, eds. Matteo Bittanti and Domenico Quaranta (Milan: Johan and Levi Editore, 2006), 131.

To Betts, this trace of player data transmitted from the 24/7 operations of the *Q3A* server produces a “...constant modulation of the digital landscape...an environment in flux, with no end or beginning.”³ In 2003, Bett’s *QQQ* was shown in the seminal exhibition *games. Computer games by artists* at Phoenix-West in Dortmund, Germany which sought to address the subversive and deconstructive potential of video games as an art form. Curated by Tilman Baumgartel, Hans D. Christ, and Iris Dressler, the exhibition showcased *QQQ* along with art mods by JODI, Cory Arcangel, Anne-Marie Schleiner, Brody Condon, Arcangel Constantini, and Joan Leandre, among others in a warehouse on the repurposed Phoenix-West blast furnace grounds. In this major exhibition of his work, Bett’s *QQQ* subjected audiences to fragmented objects and environments from the *Q3A* gameworld that as Baumgartel describes, “...look like they were formed in a broken kaleidoscope—sometimes in icy blue and white, sometimes in warm shades of brown.”⁴

The painterly forms in Oliver and Bett’s mods are often critically assessed by the way they provide a type of subversive commentary on traditions of modernist painting, replacing the expressive hand of the artist with—as organizers of the 2013 transmediale Festival observed of Oliver’s *ioq3apaint*—the “digital brushes” of the game’s bots.⁵ The images produced from Oliver’s mod are often labeled as Pollock-esque in appearance and process. John Sharp has gone as far as comparing the stills and recordings from *ioq3apaint* to Pollock’s gestural techniques in his action paintings.⁶ Here, I would argue that this representationalist thinking understates the aesthetic significance of Oliver and Betts’ work. It isn’t so much a question of how their painterly abstractions fit into a taxonomy of high modernism, but rather the *generative processes*

³ Tom Betts, “Init() > Startup code,” www.nullpointer.co.uk, <http://www.nullpointer.co.uk/qqq/qqq1.htm>.

⁴ Tilman Baumgartel, “On a Number of Aspects of Artistic Computer Games,” *medienkunstnetz*, http://www.medienkunstnetz.de/themes/generative-tools/computer_games/.

⁵ transmediale Festival, “ioq3apaint,” www.transmediale.de, <https://transmediale.de/content/ioq3apaint>.

⁶ Sharp, *Works of Game*, 24,29.

that undergird their production. In this chapter, I situate their art mods within a genealogy of generative techniques present throughout mid-century traditions of the neo-avant-garde. In the broadest sense, generative artworks do not refer solely to works produced via computational media as one might imagine, but rather how an artist employs a dynamic, complex system to produce aesthetic possibilities through its automatic processes. Throughout the 1960s, individuals within movements of conceptual and kinetic art, sought to reevaluate the Duchampian readymade by setting in motion complex systems abiding by their own natural or physical laws through which multiple iterations of an artwork might come into being. This push questioned the mere appropriation of a readymade as a commodity object whose status as art was pre-determined and unchanging. Generative techniques of the mid-century likewise questioned the logic of authorship in that the artist surrendered some or all their control to the autonomy of a system, countering for instance, the kind of intuitive, gestural command witnessed in Pollock's drip paintings.

To understand the generative processes inherent to Oliver and Betts's mods, I first undertake a thick description of the *Q3A* engine, particularly its use of a shader system to apply complex textures and a groundbreaking bot AI. I then examine the aesthetic techniques of two artists working throughout the 1960s—Jean Tinguely and Hans Haacke—to situate Oliver and Bett's work within kinetic and conceptual traditions of the neo-avant-garde. Here, I look at Tinguely's *Meta-matics*: machinic apparatuses designed to paint, draw, and in some cases, self-destruct based on chance and their own instability. In chapter two, I briefly discussed Haacke's practices in relation to institutional critique, but here I examine his earlier installations that sought to use natural and physical processes—condensation, ice, humidity—within generative systems that worked outside the artist's intent. Like Oliver and Betts' mods, Tinguely's *Meta-*

matics and Haacke's use of natural systems point to autonomous operations that engender a work of art through movement and variation. I argue that our aesthetic experience of Oliver and Betts' mods arise from their generative processes, or rather the way in which aesthetic potential is governed within a complex system of human and technological agents. Our apperceptive-proprioceptive engagement with Oliver and Betts' mods results from a *flattening* of the three-dimensional spaces and entities in *Q3A*. These autonomous mechanisms of glitch at once defamiliarize the game by distributing our perception of three-dimensionality across a visual field of abstraction, while drawing our gaze to the engine as an unpredictable, generative system. As I suggest, this formal disruption of the gamespace produces a mimetic operation in which the violence of the arena deathmatch is felt as a nonrepresentational, *violence of sensation* within the body, a concept I borrow from Gilles Deleuze's writings on the painter Francis Bacon.

The *Quake III: Arena* Engine

The generative techniques we see in Oliver's *ioq3aPaint* and Betts' *QQQ* are a result of disrupting the normative operations of id Tech 3: the engine used to power *Q3A*. The game was designed by Graeme Devine, Tim Willits, and Paul Jaquays. John Carmack along with Robert Duffy and Jim Dose programmed the engine while Adrian Carmack, Kevin Cloud, and Kenneth Scott headed its art direction. By the time *Q3A* was in development, Romero had already left id Software to start his company Ion Storm with previous id colleague Tom Hall. Unlike the previous *Quake* games, *Q3A* was designed without a single-player campaign narrative. Rather, single-player mode is predicated on the multiplayer deathmatch experience as the player battles AI-controlled "bots" in a series of arenas. The player progresses through gothic and futuristic environments, fighting increasingly harder bots in more complex arena spaces (Figure 4.1). The game's multiplayer was designed for networked play with up to 16 players including "Team

Deathmatch” in which two teams of four battle for the most kills; “Tournament” mode in which two players compete against only each other; and Capture the Flag in which teams attempt to steal the enemy flag while protecting their own.

The *Q3A* engine is a variant of the *Quake* and *Quake II* engines. Its rendering capacities are based on the hardware acceleration used in *Quake II*, while employing Binary Space Partitioning and the use of Potentially Visible Sets developed in the *Quake* engine.⁷ Unlike its predecessors, *Q3A* introduced a shader system designed in conjunction with OpenGL and

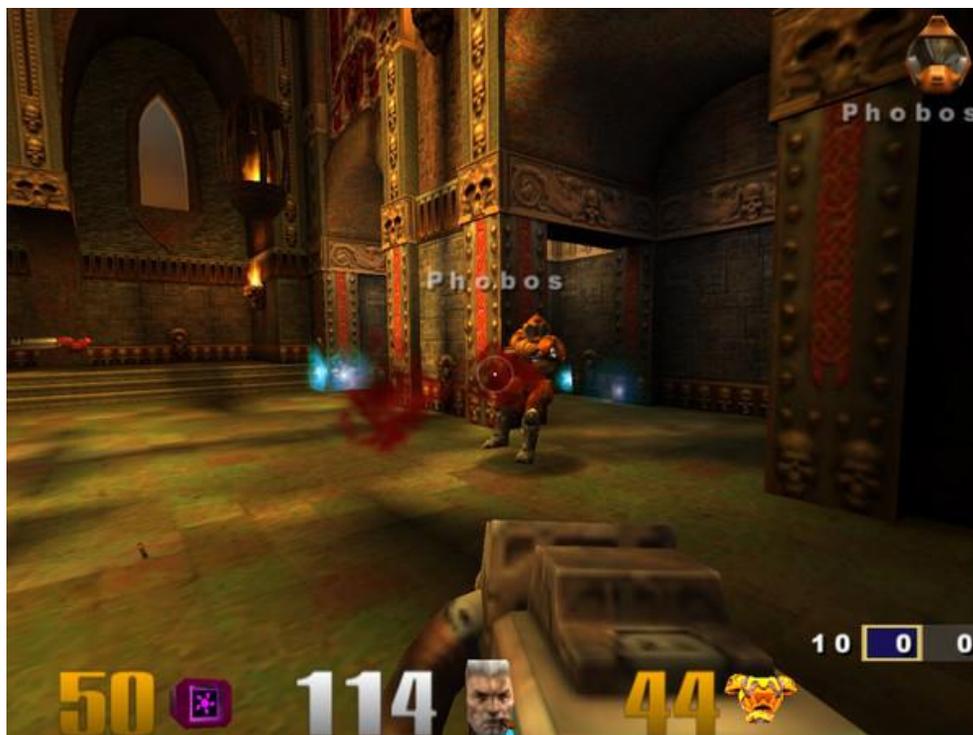


Figure 4.1: Fighting Bots in *Quake III Arena* (id Software, 1999).

specific graphics cards. Different from other engines of the time, particularly the contending Unreal Engine, *Q3A* does not include a native renderer and requires a graphics accelerator.

⁷ For a detailed description the *Quake III* engine architecture see Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 1 of 5)” *Fabien Sanglard’s Website*, June 30th 2012, <http://fabiansanglard.net/quake3/>.

According to the *Q3A Shader Manual* written by Paul Jaquays and Brian Hook, the shader system allows for rendering detailed curves and volumes using small text scripts that explain a given surface's properties and instructions for rendering its appearance in a map. Scripts are categorized based on which surface textures require rendering in each map: surfaces for base, castle, hell, and so on. Other scripts handle in-game special effects and environment textures such as the sky and various liquids.⁸

Each shader is composed of several texture layers for which the engine calculates the correct order to superimpose each one upon the other. A shader augments the visual appearance of textures on a brush, splined curves, or a mesh model. Here, a brush (often called a BSP brush) refers to specific geometric templates used to block out and give shape to a level. They can be standard shapes—a cube, sphere or cylinder—or pre-determined objects such as a spiral staircase. Brushes are the fundamental geometries of each map that are ordered by the engine during the rendering process. When a map is compiled, geometry brushes are converted into static meshes—models composed of a series of vertices to form polygons—and subsequently drawn by the engine. Because they can be cached in video memory and drawn by a graphics card, mesh models are efficiently rendered with considerably more detailed textures than BSP brushes. Yet, as Jaquays and Hook explain, in addition to surface effects, a shader file can also modify the “content” and perceived movement of in-game objects. The surface effect of a brush may contain, for instance, luminosity and transparency levels as well as vibrant RGB color values, yet a shader's content effects dictate how specific textures interact with a player in the gameworld. Content effects include texture layers for elements such as water, fog, mirror reflections, or non-solid objects that operate in different capacities when a player moves through

⁸ See introduction of Paul Jaquays and Brian Hook, “Q3Radiant Shader Manual, Revision #12,” icculus.org, https://icculus.org/gtkradiant/documentation/Q3AShader_Manual/ch01/pg1_1.htm#what .

them. A shader may also include deformation properties, giving the shape of an object the illusion of movement. For each shader, designers employ “blend” functions that direct the engine on how to correctly composite the texture layers.⁹

In the *Q3A* engine, shader scripts afforded the designers maximal accessibility and modularity over the look and actions of in-game surfaces, providing an unprecedented level of detail for each arena. However, the inclusion of multiple shader textures also constrains processing speed as each time a shader effect modifies a texture, the engine must re-render that section of the gameworld. For example, an illumination texture passing through a fog texture will require the engine to redraw those polygons affected by the shaders so that it can render volumetric fog in the game. To bypass issues of overdraw in the shader system, Carmack and others adopted a technique in computer graphics known as the fast inverse square root function which uses integer operations to efficiently calculate lighting, color values, and reflections.

In his production notes, game artist Paul Steed explains that models for players and bots in *Q3A* were rendered using MD3 files, a format that stores information for character movement and interactions with players and arena environments.¹⁰ Models were constructed using 3D Studio Max R2.5 by Kinetix who at the time was a division of the 3D modeling software company Autodesk and were animated using the “Physique and “Biped” functions in Kinetix’s Character Studio. For each model, animations were captured based on vertex movements that compose its structure and physicality, often referred to as mesh deformation in computer graphics. The MD3 format afforded developers the ability to modify the number of key frames per second as opposed to the fixed 10 frames per second used with the previous MD2 format in

⁹ Jaquays and Hook, “Q3Radiant Shader Manual.”

¹⁰ Jaquays and Hook, “Q3Radiant Shader Manual.”

Quake II. Increasing frames allows for more intricate animations, an improvement from the more unstable and jerky models in the previous *Quake* games.

As Steed mentions, the MD3 file allows for three areas of each character—the head, body, and legs—to be modified separately and blended together with independent textures. For instance, a bot may have sprinting movements for the legs and firing animations for the torso and arms.¹¹ As with the *Quake* engine, characters are lit by Gouraud shading while the maps are illuminated by a pre-calculated lightmap. Sanglard explains that at runtime the engine loads the map and divides it using BSP. It then calculates visible polygons for the current PVS and blends the lightmap with the appropriate shaders (Figure 4.2).¹² Within this multitexturing, the engine can also reflect light sources from moving models.

The *Q3A* engine utilizes a breakthrough AI system composed of bots or enemy characters programmed for combat awareness and to seek out and fight human players at different tiers of difficulty. In order to mimic human behavior, bots were designed with individual personalities articulated through scripted dialogue. When a human player engages in chat with enemy bots, the AI responds by cycling through various taunts and insults related to fragging or being fragged in the arena. The bot system was developed by industry pioneer Jan Paul van Waveren to be autonomous; or rather, to replicate believable human actions. In other words, the bots can make their own decisions predicated on goals they desire to achieve during play. Each bot employs what is called domain knowledge—an understanding of changes that occur within the gameworld—to reason and carry out a specific string of actions. Van Waveren mentions that bots are programmed to seek out goals within a hierarchy of importance, with winning as an individual or on a team—having the highest number of frags, capturing the enemy team’s flag,

¹¹ Paul Steed, “Putting New Models in Quake III Arena,” ed. by Paul Jaquays, icculus.org, https://icculus.org/projects/gtkradiant/documentation/Model_Manual/model_manual.htm.

¹² Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 2 of 5).”

and so on—as the most crucial outcome. Bots employ a path of sub-goals to win. These may include aiming and firing at an opposing human or bot, not losing a life by eluding enemy



Figure 4.2: Top: *Q3A* level textures. Middle: Lightmap applied to arena. Bottom: multi-texturing combining both textures and lighting, from Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 2 of 5)” Fabien Sanglard’s Website, June 30th 2012, <http://fabiansanglard.net/quake3/>.

projectiles, or to obtain better weapons, powerups, or health. Van Waveren explains that each bot makes reasonable decisions to accomplish goals and sub-goals using what is referred to in computer science as fuzzy logic, a system developed by Lotfi Zadeh at University of California, Berkeley during the mid-1960s that expands upon standard Boolean operations of true/false to include uncertain statements and partial truths. This logic system informs a bot's decision making through what are called "fuzzy relations" which calculate the amount it desires to use or obtain an object or achieve a particular goal within the gameworld. As an example, a value is assigned to a bot's desire to pick up a railgun weapon. Here, fuzzy relations are established as variables dictating what weapons/items the bot currently has and in what quantities, the bot's health and armor levels, and so on, to establish how much the bot wants the railgun based upon the gamestate and its current circumstances. Based on these variables, the bot calculates a decision to acquire the railgun only if it is advantageous toward the main goal of winning the match.¹³

A bot's decision-making processes are embedded among layers within a larger AI architecture. As van Waveren points out, a bot's capacity to make decisions at any given moment of play is based upon an awareness of the gameworld that increases with each layer of data. The initial layer describes the output of each bot or its basic actions—running, jumping, shooting, crouching—and van Waveren's Area Awareness System (AAS). The AI system was designed to give each bot data for all events currently happening in the gameworld. A bot "senses" the arena after receiving this information as a series of pre-processed variables pipelined directly from the engine. Each bot is provided with data about how to efficiently navigate each map and interact with other players and objects. As van Waveren explains, the AAS works similar to waypoints

¹³ J.M.P. van Waveren, "The Quake III Arena Bot," (thesis, Delft University of Technology, 2001), 4-5, 12, 65.

used in pathfinding for other forms of AI; or rather, a system that allows agents to quickly move from one area of the gameworld to another along designated “paths.” Essentially, a waypoint system operates through a series of linked coordinates, or nodes, along a map that allow agents to move between them using the shortest possible distances. In the *Q3A* engine, routes within arenas are pre-calculated and cached so that they may be easily retrieved for the bot to interpret.

In the AAS, van Waveren employed 3D “bounded hulls,” convex open areas (within an arena) that allow for a bot or player to move from one point to another along a minimal vector line. The convex areas of an arena have fewer obstacles to impede a bot from navigating to and from various points, what van Waveren refers to as “reachabilities.” A reachability is determined if a bot can move with little complexity between two areas of the map, for example, teleporting, riding an elevator, running along a vector, jumping onto an object, or propelling from a jump pad. Lastly, the AAS employs collision detection, or when an entity is able to discern boundaries and brushes in the game, by placing bots within bounding 3D boxes that interpret contact with solid objects.¹⁴ From the AAS, bots are given information about the 3D gameworld as it is sampled from the engine’s BSP tree. Data about each map is extracted by calculating the area coordinates for bots using the nodes of the BSP tree to narrow down the polygonal planes containing the bot’s bounding box and general location.

If the initial layer of the AI architecture is bot input and the AAS, then the second layer is comprised of the fuzzy logic a bot uses to achieve goals related to gameplay and make decisions about how to chat with a human player. A third layer is dedicated to a network of nodes for alternate AI scenarios and a module which affords bots the ability to interpret commands from their team leader and other bots/players. The module gives AI support for decisions related to navigation, fighting, and problem-solving for the alternate AI network. There are also a series of

¹⁴ J.M.P. van Waveren, “The Quake III Arena Bot, 24-25, 34.

conditional rules in the form of if-then-else statements that further assist the bot AI. The final layer is composed of the AI brain, or rather the team leader assigned to each bot platoon. Van Waveren explains that the team leader surveys the gamestate and its current situation at 10 Hz, meaning that at each tenth of a second it receives data from the AAS and decides upon the appropriate actions for the remainder of the team.¹⁵

In conjunction with van Waveren's AI system, Carmack updated the *Q3A* engine with new networking capabilities based on a snapshot architecture that transmits data for each frame of gameplay from the server to client computers using a UDP (User Datagram Protocol). The server receives the updated gamestate at a fixed rate asynchronous with its clients. It then sends the current frame of gameplay to the client using delta encoding to limit bandwidth, or rather it excludes information from each frame, sending only disparities from the last frame downloaded from the client. Sanglard explains that Carmack designed the engine to run a virtual machine (referred to as a QVM) which loads three separate machines at any given time among the client-server. Here, a virtual machine refers to the emulation of a computer's architecture. In the *Q3A* engine, two virtual machines operate on the client side of the network that communicate the current gamestate with each other. The "cgame" machine performs culling operations and runs the renderer library, while the "q3_ui" (user interface) operates menu phases in the game. The server side includes the "game" machine which triggers the bot library and run the game's AI component (Figure 4.3). Carmack used a virtual machine for client-server operations because it afforded a more stable format for programmers when introducing new modifications. In August of 2005, Carmack released the id Tech 3 source code under the General Public License version 2, allowing for modding communities to develop derivative engines.

¹⁵ J.M.P. van Waveren, "The Quake III Arena Bot, 22.

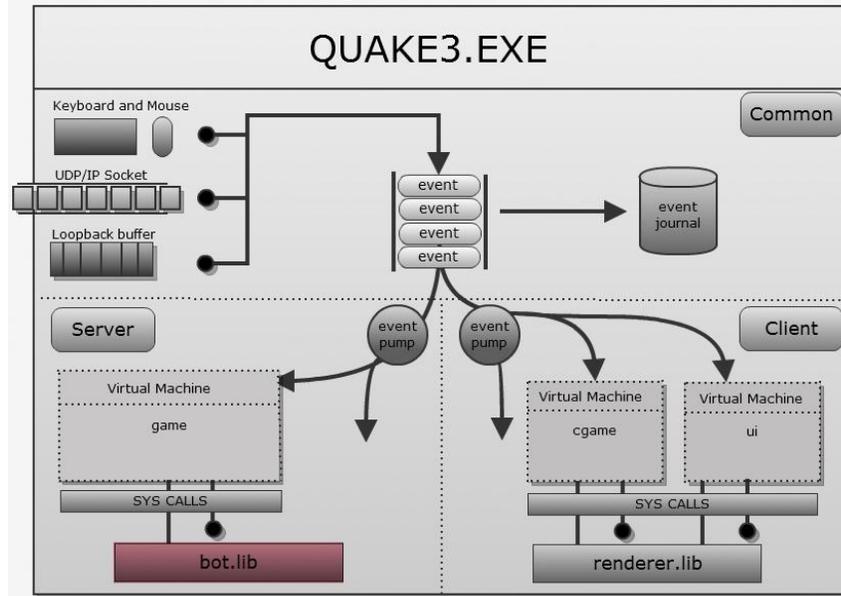


Figure 4.3: The *Q3A* engine architecture from Fabien Sanglard, “Quake 3 Source Code Review: Architecture (Part 2 of 5)” *Fabien Sanglard’s Website*, June 30th 2012, <http://fabiansanglard.net/quake3/>.

Soon after, the icculus.org software incubator created *Ioquake3*, an engine derived from the *Q3A* source code that sought to remove bugs, improve graphics, and sound through the addition of SDL (Simple DirectMedia Layer) and OpenAL: a cross-platform interface for 3D audio. As I expand upon in the following section, the generative systems of Tinguely and Haacke during the 1960s give precedent to the ways Oliver and Betts employ the engine’s bot AI, colors and textures within the shader system, and revamped network to produce abstractions of glitch.

Volatile Machines and Physical Systems

To understand generative processes that underlie Oliver and Betts’ modifications, I first turn to the Meta-matic sculptures of Swiss artist Jean Tinguely. Throughout 1955-1956, Tinguely began creating his Meta-matics as a series of automated machines that could paint or draw abstract compositions. Collecting materials from junkyards near his Paris studio, Tinguely constructed each Meta-matic as an assemblage of gears, belts, and electric motors that included a

pen or brush rigged to a mechanical arm. Positioned in a gallery, each Meta-matic required a



Figure 4.4: *Méta-Matic No. 10* (Jean Tinguely, 1959).

participant to insert a small piece of metal into a slot, causing the device to erratically jolt to life and move the pen spastically across a paper surface attached to the machine. The sculpture would produce a scribbled drawing in which no two were alike. Tinguely exercised certain constraints such as using a different colored pen or adjusting the machine's range of motion, and a participant in some cases could stop, start, or speed up its motor. Yet, once the sculpture was initiated, it generated a unique variation of scribbling through its own machinic logic (Figure 4.4). Pamela M. Lee observes that within discourses of kinetic art, Tinguely's Meta-matics unveil a temporal "worlding" in Heidegger's use of the term. The kinetic processes in Tinguely's work, particularly the explosive, self-destructing machines in his televised spectacle *Study for an End of the World No. 2* (1962), reveal the dynamic relationships among time and technology that emerged during the late 50s and early 60s. His sculptures resemble a microcosm of the hope and perils of computer automation for a traditional labor force of the machine age, as well as anxieties over geopolitical formations of power and the threat of global mass destruction. Lee

mentions that Tinguely's playful experiments with automation foreground a paradoxical stance in his artistic practice. His sculptures at once served to "facilitate artistic "expression," but by the same time they were necessarily *mechanical* objects, denying its user the freedom mythically alleged to guide the creative process."¹⁶

To Tinguely, this ceding of authorial control allowed his Meta-matics to be "free" through what he refers to as "the functional use of chance," or the unique variations and motions of the machine and its potential to break down, explode, catch fire, produce scribbled drawings, work perfectly, or not work at all.¹⁷ Calvin Tomkins argues that the Meta-matic and its capacities to generate potential outcomes—to never repeat the same scribble twice—stems from Tinguely's belief that the sculpture's freedom can only occur "from accepting the impermanence, the accidents, and the infinite transformations of life itself."¹⁸ As Tinguely observes of each machine's variations, "Life is movement...Everything transforms itself, everything modifies itself ceaselessly, and to try to stop it, to try to check life in mid-flight and recapture it in the form of a work of art, a sculpture or a painting, seems to me a mockery of the intensity of life...I want only to involve myself in the moving object that forever transforms itself."¹⁹

The bizarre activities of Tinguely's machines, the shuddering and quaking when an electric motor propelled their junk parts into motion, produced a sense of playful interactivity with spectators, a phenomenal engagement, as Lee argues, that was solidified through discourses of kinetic art during the mid-century. The playful kineticism of Tinguely's sculptures, the slug-operated Meta-matics or even his bizarre *Cyclo-Graveur* (1962)—a stationary drawing machine requiring participants to ride it like a bike—situated aesthetic experience as fundamentally

¹⁶ Pamela M. Lee, *Chronophobia: on time in the art of the 1960s* (Cambridge, Mass: MIT Press, 2006), 90-92, 111.

¹⁷ Jean Tinguely quoted in Calvin Tomkins, *The Bride and the Bachelors: Five Masters of the Avant-Garde* (New York: Penguin Books, 1976), 146-147.

¹⁸ Tomkins, *The Bride and the Bachelors*, 150.

¹⁹ Tinguely quoted in Tomkins, *The Bride and the Bachelors*, 150.

“material and embodied, as contingent and site-determined.” To Lee, kinetic art explicitly addressed the *timeliness* of the audience, whose encounter with the work mirrored its ever-fluctuating configurations.²⁰

The audience’s embodied encounter with such generative “configurations” was best demonstrated in Tinguely’s monumental, self-destructing machine *Homage to New York* that in 1960 within the sculpture garden of the Museum of Modern Art, was unable to kill itself among throngs of onlookers. Firefighters intervened to extinguish the fire bellowing from the machine before it destroyed itself entirely as Tinguely had hoped. Yet in its failed suicidal attempt, the sculpture produced a spectacular range of unanticipated effects once its complex system of motors and gears were actuated by Tinguely with assistance from experimental filmmaker Robert Breer and Billy Klüver, then an engineer at Bell Laboratories. The machine was constructed as a bricolage of junk parts that Tinguely, Breer, and Klüver collected from dumps and landfills. It consisted of electrical motors, bicycle and baby carriage wheels, levers, and cranks which supported a large sheet-metal container housing a roll of paper. Tinguely intended for mechanical arms with brushes to paint across the paper surface as the roll was continually blown out into the audience by a large fan. Yet when the machine was started, by its own volition, it rolled the paper in the wrong direction, spinning it wildly back into the belly of the apparatus after Tinguely and his assistants had installed a belt the wrong way. Operating on its own logic, the sculpture used its various mechanisms to produce a percussive cacophony. Notably, a large arm beat upon an old washing machine drum. Titanium tetrachloride was released from a child’s bassinet to produce thick plumes of smoke that wafted in the direction of the audience. A large meteorological balloon was filled with the aim that its eventual bursting would prompt the release of odorous chemicals, an event that never occurred. In another area of

²⁰ Lee, *Chronophobia*, 95.

the sculpture, Tinguely had installed an old piano whose keys were beat furiously with mechanical arms, producing only three notes. Attached to the piano was a small Meta-matic that would paint, unroll, and roll itself back up again. Inside the piano, a gas can was set to pour onto a candle, igniting the sculpture in flames (Figure 4.5). Tinguely's infamous *Homage to New York* enacted a generative process through the machine's freedom and its constant dynamics, implicating the audience's shock, awe, delight, and fear into the unfolding of the work.

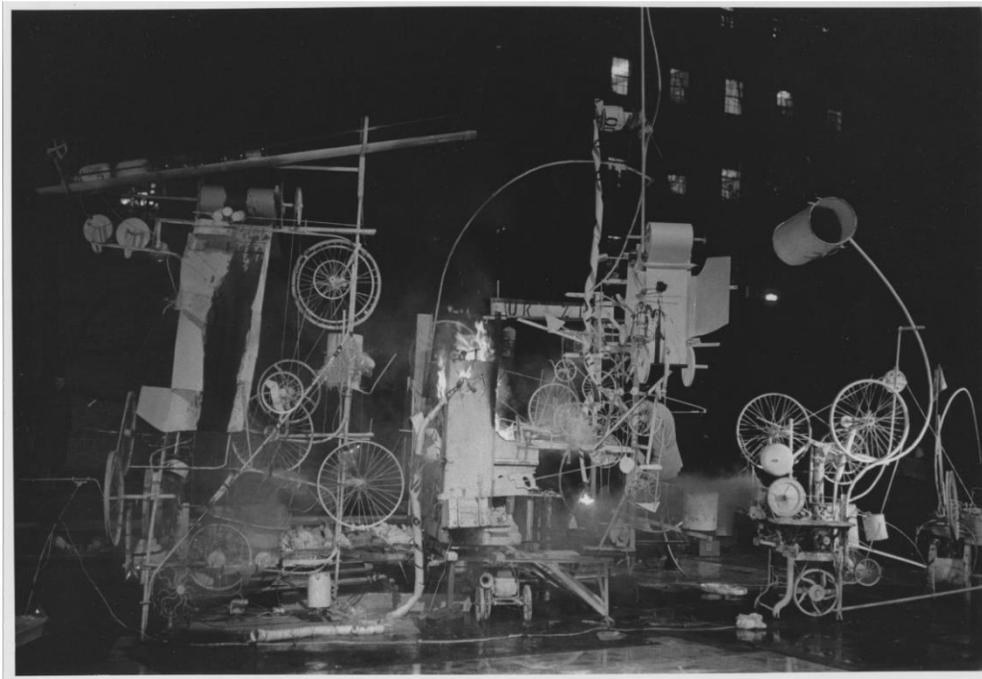


Figure 4.5: *Homage to New York* (Jean Tinguely, 1960). Photo courtesy of The Museum of Modern Art Archives and David Gahr.

If Tinguely sought to experiment with generative outcomes through the instability of his machines, then Hans Haacke's conceptual artworks explored variation within physical and natural systems. Haacke interrogated the status quo of the museum as a social system throughout the 1970s, yet his earlier projects during the 1960s examined how a work's composition could be derived from variations in its immediate environment: light, temperature, air, humidity, and the changing position of a spectator. Haacke discusses that the generative components of his

installations relate to how a sculpture ceases to be a fixed object and instead becomes constitutive of its environment to produce a larger, complex system.²¹ Each work operates as a system of interrelated physical processes that as Alexander Alberro says, possesses a life of its own much like an organism. Thus, just as Tinguely's machines functioned as a microcosm of the changing social and political conditions of the era, Haacke's installations serve as a correlate to natural processes within the world.²²

And like Tinguely's contraptions, Haacke's physical systems exist in their various permutations independent of the artist's intent or the viewer's interpretation. To Haacke, the system continues to operate outside human "knowledge, past experience, the mechanics of perceptual psychology...emotions, or any other involvement short of...running berserk and smashing it." Yet, the viewer's encounter with the system produces "a gulf of subjective projections," mental processes that are imbricated in the generative development of the work.²³ Haacke speaks to the viewer's interaction with a physical system in his early experimentations with mirrored stainless-steel objects:

There is neither a correct nor an incorrect point of view from which to look at them. Their environments—including the spectator—form an integral part of them. The environment is constantly participating in their creation. They are not fixed, their appearances are infinite. They are exceeding their material boundaries and are limited, respectively, by the boundaries of sight in the space in which they happen to be, the workshop, the exhibition room, and the stars.²⁴

Here, Haacke considers the body's incorporation into physical processes of light and reflection, yet during 1963-65 he would further experiment with the spectator's involvement (or lack

²¹ Hans Haacke, "Untitled Statement, "For some years . . . ," 1967," in Hans Haacke and Alexander Alberro, *Working Conditions: The Writings of Hans Haacke* (Cambridge, MA: MIT Press, 2016), 10.

²² Alberro, "Introduction: Hans Haacke and the Rules of the Game," in *Working Conditions*, xi.

²³ Haacke, "Untitled Talk at Annual Meeting of Intersocietal Color Council, New York, April 1968," in *Working Conditions*, 24.

²⁴ Haacke, "'Observing My Mirror Objects,' 1962," in *Working Conditions*, 2.

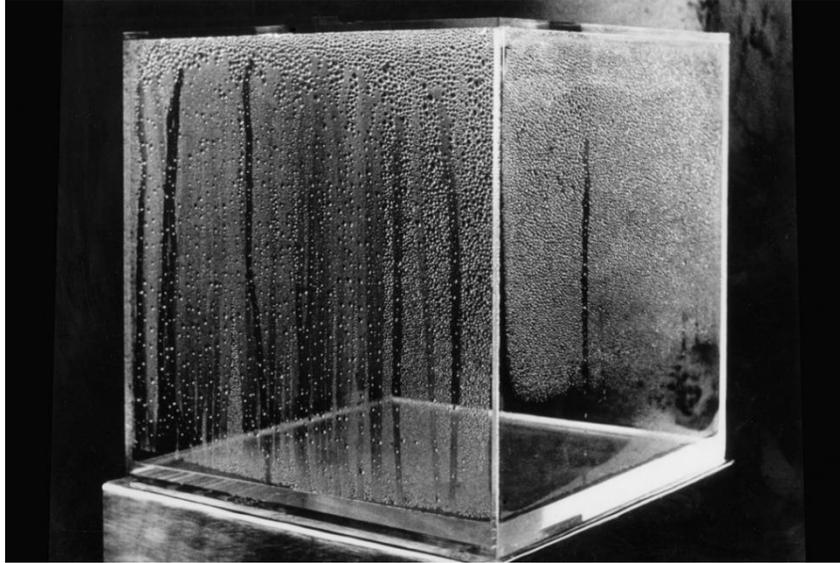


Figure 4.6: *Condensation Cube* (Hans Haacke, 1965). Version of the cube at Museu d'Art Contemporani de Barcelona from 2006. Photo courtesy of the artist.

thereof) in his hydrodynamic sculptures, namely his *Condensation Cube* which he would explore in various iterations during those years. Haacke constructed the sculpture as an enclosed cube of transparent plastic containing a small trace of water. Adhering to its own natural cycle, the sealed water evaporates and produces condensation droplets which eventually drip back down to the bottom of the cube. The cube indicates an on-going, autopoietic system composed of interdependent processes within the gallery including, for instance, temperature changes, air currents from an AC system, or varying directions and intensities of natural or artificial light. Even the traffic of spectators can potentially alter the rate at which the water performs its cycle, as the warmth of bodies slightly affects the surrounding room temperature, producing variations on a molecular scale (Figure 4.6).

In 1966, Haacke began experimenting with water systems in other physical states, witnessed in his installation *Ice Stick*: a refrigerated metal coil on which layers of moisture freeze and melt with changes in external temperature. Here, an unexpected change in temperature or humidity generates novel patterns of ice, introducing a degree of unpredictability

into the system. A year later, Haacke introduced a humidifying sculpture *Steam* that when exhibited with *Ice Stick* would provide a counterpart to the freezing water. He describes that water vapor from the humidifier would increase moisture in the gallery space, in turn, affecting the dehumidifying effect of the refrigerated stick, producing a “balanced humidity” through a “physical exchange” within a larger, intricate system.²⁵ By the end of the decade, Haacke would begin to interrogate the role of biological systems in works such as *Grass Grows* (1967-1969), a mound of grass centered within the gallery that carries out its photosynthetic processes and growth dependent upon changes in temperature and light (Figure 4.7). With these natural systems in mind, Haacke proposes several rules that guide his conceptual processes in a short manifesto from 1965, which parallels Tinguely’s *Meta-matics* and generative techniques in art modding more broadly:

...make something which experiences, reacts to its environment, changes, is nonstable...

...make something indeterminate, which always looks different, the shape of which cannot be predicted precisely...

...make something which cannot “perform” without the assistance of its environment...

...make something which the “spectator” handles, with which he plays and thus animates...

...make something which lives in time and makes the “spectator” experience time...²⁶

²⁵ Haacke, “Untitled Talk at Annual Meeting of Intersocietal Color Council, New York, April 1968,” in *Working Conditions*, 23.

²⁶ Haacke, “Untitled Statement, “. . . make something which experiences . . . ,” 1965,” in *Working Conditions*, 5.

As I now turn to, Oliver and Betts' modifications work within these traditions of generative art to engage the body's apperceptive-proprioceptive response. Namely, *ioq3apaint* and *QQQ* utilize an existing computational framework—the *Q3A* engine's bot AI, multiplayer network, and shaders—to produce a complex system of interrelated bodies and machines. In the presence of these generative mods, our aesthetic experience unfolds through the autonomy of the engine's logic—the disruption of glitch that produces abstract forms through a flattening of the



Figure 4.7: *Grass Grows* (Hans Haacke, 1967-69). Photo courtesy of the artist.

arena's depth. Once running, Oliver and Betts' mods operate through an on-going instability in response to their surrounding ecology, which as I suggest, rearticulates the game's representations of bloodshed as an intensive violence within the body.

The *Meat* of the Glitch

The ways in which *ioq3apaint* and *QQQ* continually transform conventional 3D space into patterns of glitch project what contemporary Japanese artist Takashi Murakami has termed

superflat aesthetics. Drawing from avant-garde traditions of the Japanese eccentric artists, Murakami articulates his own painterly approach through superflat—a style in which forms work to distribute perception across the visual field as opposed to perspectival rendering in which the eye penetrates the depth of the image. Within the Japanese superflat lineage, ranging from Edo period ink paintings of natural scenes to contemporary anime films, the viewer is prompted to trace and retrace paths of line, color, and shape throughout the image. To Murakami, this aesthetic “violently accelerates and decelerates the gaze as it swings it from side to side.” In superflat, the image possesses an equal balance of visual distribution that covers its surface (Figure 4.8).²⁷



Figure 4.8: Superflat aesthetics in *In the Land of the Dead, Stepping on the Tail of a Rainbow* (2014) at Gagosian Gallery (Takashi Murakami, 2014). Photo courtesy of Ben Davis.

²⁷ Takashi Murakami, “A Theory of Super Flat Japanese Art,” in *Super Flat*, ed. Takashi Murakami (Tokyo: MADRA Publishing, 2000), 193-194.

In Oliver and Betts' mods, the *Q3A* engine works to defamiliarize our experience of the arena deathmatch by flattening 3D space into a continually-shifting terrain of abstract forms—a result of the hall of mirrors effect that smears multiple frames across the interface. It works to elicit the body's apperceptive feedback to the engine's on-going processes as they function interdependently within a broader, generative system. Yet as I have mentioned, our mimetic encounter with the superflat forms of these glitched sequences, forces a proprioceptive strangeness by transposing the representational fragging in the arena into intensities of violence. In his writings on the painter Francis Bacon, Gilles Deleuze articulates how the figural is expressed through non-representational, bodily experience. Deleuze explores what he calls the *violence of sensation* inherent to Bacon's grotesque, abstracted images of popes, screaming mouths, and crucifixions as well as self-portraits and those of close friends, including painter Lucian Freud. Bacon often depicts his subjects in sequences across a diptych or triptych, isolating figures within parallelepiped spaces (Figure 4.9). Here, Deleuze is interested in the affective forces among materials, forms, and the body, especially how the image moves beyond figuration within our sensory experience of Bacon's work. Bacon's concept of the figure, through a deformation of its own body—through violent “spasms, vomiting, and excrement—is in constant escape of its “figuration,” hollowing itself of organizing bodily traits and returning to the materiality, the forms, the surface of the painting. These asignifying traits of the body produce a “zone of indiscernibility” between face and head, animal and human, that Deleuze describes as the *meat*. In his portraiture, Bacon does not portray the face—signifying expressions and features—but rather he paints the head without a face. Thus, through the head, Bacon's figures interrogate the strange indiscernibility among the flesh of the human and the meat of the animal, or rather other kinds of material bodies. Within each figure, the tension between the flesh

and bone—where the two meet upon the canvas—articulates the meat, which Bacon portrays through plays of color and line. In Bacon’s work, Deleuze introduces the concept of meat to interrupt the demarcation between human and animal as they become one in the same, a becoming-animal. Through the meat of the image and the blurring of what constitutes a *body*, Deleuze pushes us to think outside discursive constructions of the human form within histories of art.²⁸



Figure 4.9: *Figure with Meat* (Francis Bacon, 1954).

²⁸ Gilles Deleuze, *Francis Bacon: The Logic of Sensation*, Trans. by Daniel W. Smith (Minneapolis: University of Minnesota Press, 2004), 16, 19-22.

Through the meat, Bacon's paintings engage the body's affective faculties which for Deleuze, is to "experience the sensation only by entering the painting, by reaching the unity of the sensing and the sensed." This reciprocation of color, line, and shape in the viscera sustains sensation through a vital rhythm, or rather the violent spasms of the figure attempting to return to the visual field of the painting. It is the figure that constitutes this violence of sensation. As Deleuze states:

"The violence of sensation is opposed to the violence of the represented (the sensational, the cliché). The former is inseparable from its direct action on the nervous system, the levels through which it passes, the domains it traverses: being itself a Figure, it must have nothing of the nature of a represented object."²⁹

In the meat of the figure, the viewer experiences varying orders of the same violent sensation that refer to each other and as Deleuze says "constitute[s] the "pathic" (nonrepresentative) moment of *the* sensation." Here, the figure embodies a type of *hysteria*, referring to an imbrication of subject and object. Within these domains of sensation, the figure frees the subject's eye from its fixed status as an organ by engaging it in a haptic function that *sees-feels* the figure's rhythms as a pure, sensory presence. At the same time, the figure objectively portrays a body that has hollowed itself of naturalistic depiction. It is in this reciprocation among viewer and figure that Deleuze locates sensory experience.³⁰

In *ioq3apaint* and *QQQ*, the generative instability of glitch which continually flattens and deconstructs the gameworld, evokes the body's apperceptive-proprioceptive senses through a type of sensual violence (Figure 4.10). Betts has commented on this conflation of the game's deathmatches and the vibrancy of glitch in *QQQ*:

I wanted to question the traditional assumptions of video games (Quake3 in particular) as blood-filled violence simulators by creating an abstract, painterly re-rendering with an interface you couldn't use for killing. I was also interested in the endless recycling of

²⁹ Deleuze, *Francis Bacon: The Logic of Sensation*, 39.

³⁰ Deleuze, *Francis Bacon: The Logic of Sensation*, 37, 42-45.

data, the constant death/rebirth of players and the futile performance aspect of the game and the idea of data as a mutable form, with meaning constructed in the process of its rendering, either as a specific expression or as generative possibilities.³¹



Figure 4.10: Abstractions of glitch in *QQQ* (Tom Betts, 2002).

³¹ Tom Betts, interview by *Neural*, “Nonlinear Software: Tom Betts/Nullpointer Interview,” *Neural*, issue 23, 45.

This uncanny “re-rendering” of death and respawn was evident at Betts’ exhibit of *QQQ* at the 2002 Evolution Festival in Leeds where the mod was displayed from a scrim attached to metal scaffolding within the darkness of a theater. Before the screen, was a small, illuminated podium with several keyboard buttons protruding from its surface, inviting the viewer to interfere with the continuous mutation of forms outputted from the multiplayer server (Figure 4.11). As participants attempt to modulate the in-game camera, they are confronted by trailing effects that duplicate frames across the screen, whilst aggressively zooming in and out on colorful fragments of *Q3A*’s shader textures. With each death and respawn of live players, the observer witnesses the recycled pieces of old frames project wildly throughout the interface, prompting the eye to trace their circuitous paths laterally and medially across the screen. In perhaps a more pronounced superflat effect, Oliver’s stills of *io3qapaint* force the eye to follow calligraphic lines of trailing frames along the surface of the image (Figure 4.12). In these instantiations of 3D space gone awry, remnants of *Q3A*’s saturated textures and bot movements are dispersed with equal hierarchy along the screen’s surface. Here, the rhythms of the bots’ autonomous activity—killstreaks, headshots, frags—are reconfigured as abstracted forms of violence. Oliver states that *ioq3aPaint*:

...is closer to a graphic echo of an arena, an unforgetful [sic] transcript of the events therein. Here the drive to hunt and kill is abstracted and used as control data to drive the painting process; I have simply manipulated the representation in such a way as to turn this fight into something unrecognizable and perhaps even beautiful; a transduction of the hunt into a cumulative stream of graphic memory. The bots have no sense of this experimentation, they simply hunt each other in an endless cycle of simulated life and death.³²

I would argue that although static, Oliver’s stills portray a timeslice of a complex system in its dynamic unfolding. For instance, at the 2013 transmediale, *ioq3apaint* was projected in real-time

³² Oliver, interview by Domenico Quaranta, “Delire (Julian Oliver),” 132.

so that participants could play with its interface, yet was also shown with several prints captured from the system, allowing for a closer examination its material operations. Similarly, at Hannah Maclure Centre solo show, *ioq3apaint* self-generated 36 million images in a real-time installation during the two and half months of the exhibit, yet Oliver also displayed several still



Figure 4.11: A user interacting with *QQQ* at the 2012 Evolution Festival. Photo courtesy of the artist at <http://www.nullpointer.co.uk/qqq/qqq5.htm>.

images from the mod while freely distributing hundreds of prints and the system's source code to the public. Within the generative ecologies of *ioq3apaint* and *QQQ*, our spatial experience of the

arena deathmatch is fundamentally disrupted through the violence of sensation, or what I refer to as the *meat* of the glitch. I employ Deleuze's concept of meat to consider the strange

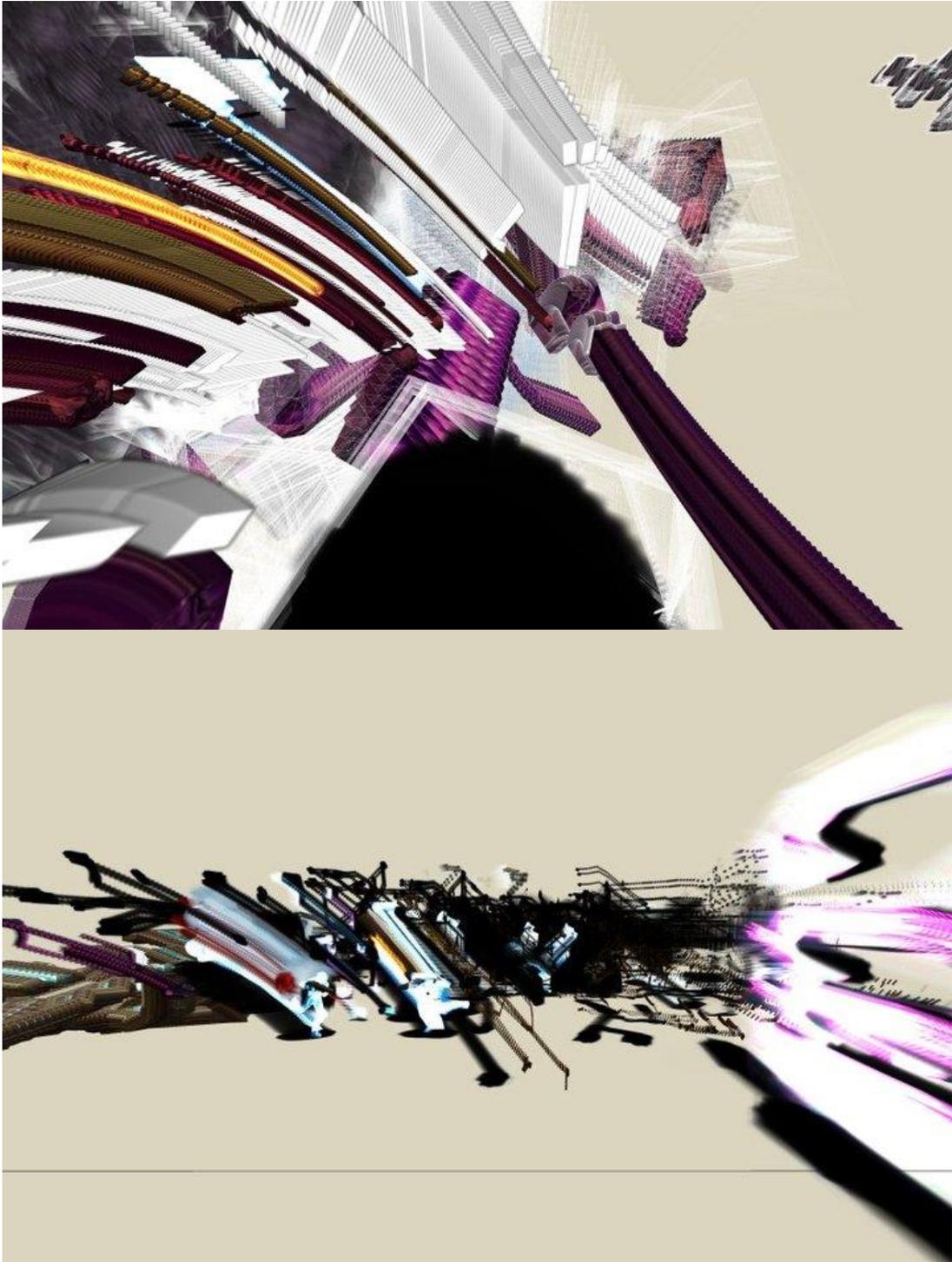


Figure 4.12: Frames from *ioq3apaint* (Julian Oliver, 2010).

indetermination among the representational properties of the gameworld—its polygonal architectures, figures, weapons, animated objects—and their re-territorialization into a continual field of abstraction. The meat emerges from the liminal space between the figurative remains of the arena and their on-going deconstruction into arrangements of glitch. It describes the bleed among the familiarity of the game into images of nothing, of pure form, of an intensive presence of violence.

In his discussion of glitch art, Tim Barker suggests that the generative characteristics of error are realized within the programming constraints set forth by the artist. Within histories of interactive art, glitch emerges from the virtual—in Deleuze’s use of the concept—as a field of aesthetic potential, shaped by both the logic of the software and interactions from artist and participants. Generative actualizations of error result from the interactivity among bodies within a complex system that probes this field of potential experience, in turn, generating difference by destabilizing the software and “shak[ing] the system free of its precise or pre-programmed functioning.”³³ Remarkably, *ioq3apaint* and *QQQ* do not require direct human intervention to generate aesthetic potential. The code executed by the altered mod is a singular process within a more extensive constellation of material operations. Like Haacke’s physical systems, *ioq3apaint* and *QQQ*, resemble self-sufficient organisms that respond to stimuli within their surrounding environment. Once booted up, *ioq3apaint* produces abstract patterns based solely upon the continual activity of the engine’s AI. When the user enters this technical milieu and moves the camera angle or captures a frame from the bot deathmatch, the system further generates potential meaning from a virtual field of interactions.

³³ Tim Barker, “Aesthetics of the Error: Media Art, the Machine, the Unforeseen, and the Errant,” in *Error: Glitch, Noise, and Jam in New Media Cultures*, ed. Mark Nunes (New York: Continuum, 2011), 52, 54, 57.

Similarly, Tilman Baumgärtel observed that at the *games. Computer games by artists* exhibition, *QQQ* operated through an inherent logic of its own that at times reacted to the unsuspecting human agents on the game's server. During the night or on weekends, when traffic on the game's server was slow, the installation would remain quiet, intermittently producing small deformations of the gamespace. Yet, if multiple players logged on, the mod would violently lurch into action and churn out kaleidoscopic forms. And just as suddenly, if players logged off, the mod would return to its graphically-retrained state.³⁴ As the altered *Q3A* engine exercises its freedom to generate errant forms within a complex ecology of machines and human users, it implicates the body within a phenomenal moment of the meat, a *hysteria*. The eye no longer operates as an organ by which to navigate and act within conventional 3D space, but instead adopts a type of sensual, haptic ability to explore the shape and contours of the game's dismantled interface. Yet, in this moment, we are also shown a gameworld removed from its codified, signifying structure as an arena of competition, of challenge, and reward.

I end here by proposing that within the broader, generative systems of *ioq3apaint* and *QQQ*, the violence of sensation articulates a sensory exchange among the body's apperceptive-proprioceptive activity and the strange agency of nonhuman matter, what Jane Bennett calls *thing-power*. In her speculative ontology, Bennett expresses the productive forces of things, specifically their capacities as actants freed from the discursive conditions that order how human subjects use and understand their existence in the world. To Bennett, things possess an ability to affect other things, limiting or empowering their capacities to act. She refers to the vital forces of entities as vibrant matter, "an earthy, not-quite-human capaciousness..." that seeks to dissociate nonhuman materials from their long-held dependency on automatistic and mechanistic theories. As Bennett asserts, by engaging a politics that recognizes the intricate connections among human

³⁴ Baumgärtel, "On a Number of Aspects of Artistic Computer Games."

and nonhuman things—stem cells, landfills, and fish oil—we are better equipped to manage ethical imperatives related to those natural ecologies.³⁵

Bennet's political theory of things does well to explain how we might conceive of a Rancierian politics of aesthetics related to generative modding. We can consider human users as things interacting with a nonhuman engine (and other material entities) that possesses a degree of autonomy and capacities to affect other bodies. Thus, the potential for sensory experience to emerge from Oliver and Betts' mods points to a braiding of actants—user, processor, server, screen, bot, shader, keyboard buttons, protocol—through which new modes of aesthetic participation are actualized among an assemblage of *both* human and nonhuman things. And herein lies the significance of the generative techniques foregrounded in *ioq3apaint* and *QQQ*. It is the violence of sensation that transpires from this ecology of things—software modification included—that demonstrates the vibrancy of our aesthetic involvement with the game engine.

³⁵ Jane Bennett, *Vibrant Matter: A Political Ecology of Things* (Durham, NC: Duke University Press, 2010), 1-2,4-5.

CONCLUSION

Intractable Spaces

A large, muscular stag runs through the streets of a low-income neighborhood. It startles an unsuspecting man walking toward the horned animal from the opposite direction. He yells, “What in the fucking world is that?” Frightened, the stag runs into several gang members standing on the corner, knocking them over. They draw handguns and open fire upon the creature, riddling its body with bullets, yet the stag continues to run unaffected. Suddenly, a Rottweiler attacks its flank, causing a taxi to careen off the road and smash into a tree.

The vignette I have illustrated here is not the product of a bizarre news story as one might imagine, but rather describes events that unfolded through the chance encounters of non-player characters in the AI of the popular game *Grand Theft Auto V* (2013). Created by Brent Watanabe throughout 2015-2016, *San Andreas Streaming Deer Cam* is a mod in which the artist programmed a deer to autonomously explore and interact with other humans and animals within the sprawling 100 square miles of the game’s fictional state of San Andreas and its major city, Los Santos (Figure 5.1). Using the RAGE (Rockstar Advanced Game Engine) engine, Watanabe programmed the deer’s actions based upon the AI of the game’s pedestrians, overriding certain functions within the Script Hook V library to generate random, unpredictable behaviors from the animal. Periodically, the deer will transport to another area of San Andreas, a failsafe designed by Watanabe to ensure the animal wouldn’t get stuck in any area of the gameworld, but also allowing it to engage with new characters. The deer is invincible yet cannot deliver any damage of its own. Watanabe exhibited the mod on multiple screens, each showing different episodes from the deer’s on-going exploration of its environment. In one instance, the deer unwittingly finds itself the subject of a police manhunt, of which it is able to escape from, while in the next,



Figure 5.1: *San Andreas Streaming Deer Cam* (Brent Watanabe, 2015-2016).

it breaks into a high-security military base. The work was also streamed live at sanandreasanimalcams.com and on twitch, with over 200,000 viewers joining in to watch the deer endlessly survey the game's boundaries.¹

Echoing Bennett, I would argue that our fascination with Watanabe's mod stems from the way in which it reconfigures a standard gamespace into one that portrays the complex fabric of nonhuman entities at work. The deer's erratic behaviors within the urban and natural ecosystems of San Andreas analogize the strange constellation of nonhuman actants that carry out the autopoietic operations of the engine's AI. The gameworld of *GTA V* is transformed into a space of *ecological surveillance*. The mod prompts us to closely observe the representational aspects of the deer's interactions within a fictional ecosystem at the same time it employs the deer as a visualization, an abstraction that shows us the AI's forensic procedures.

¹ Brent Watanabe, interview by Brian Feldman, "Hundreds of People Are Watching a Deer Run Around a Video-Game City, and It's Mesmerizing," *New York Magazine*, March 25, 2016, <http://nymag.com/selectall/2016/03/grand-theft-auto-deer-cam-is-mesmerizing.html>.

Beyond the works I have examined in the previous chapters, I use Watanabe's *San Andreas Streaming Deer Cam* to indicate that art modding still remains a vibrant aesthetic practice. As I have hoped to elucidate in this dissertation, the role of modding practices within histories of video games and digital media more broadly is mutually-constitutive with an understanding of game engine technologies. Throughout the 1990s and 2000s, artists and programmers dismantled the technical frameworks of game engines as a way to disrupt gamespaces, especially the perspectival rendering common to the first-person shooters that were synonymous with the engines that generated their gameworlds. The artists examined in these chapters engaged with both popular gaming technologies and strategies of the avant-garde within a heterogeneous site of experimentation. By altering the spaces of these industry-standard engines into fields of abstraction, they questioned common regimes of perception related to play, and as a result, worked to organize new modes of sensory engagement. By focusing on this singular period in video game history, my aim has been to bring to light the broader significance of modding as a cultural and technical movement. Histories of modding reveal a recent technical history of rendering space in computer graphics. By breaking open the computational black box of the game engine and producing new types of gamespaces, modders also disclosed the technical illusions through which industry gamespaces were constructed as immersive and palpable. Furthermore, modding practices and experimenting with game engines during this era, introduced new forms of digital media production motivated by affective labor and an eagerness to master coding techniques and elements of game design. That is to say, artists, gamers, as well as casual and expert programmers held a particular desire to playfully engage with game engines and explore their potential beyond pre-programmed industry operations.

Modding and its contingency within current media landscapes, by its nature, produces *intractable* spaces. Here, I use “intractable” to describe the disordering of space as difficult for the player to control or comprehend, a space that is uncooperative, awkward, and unruly. At the same time, I also evoke the term to indicate the vibrant materiality of the mod, or rather its ontological complexity, yet also uniqueness as a nonhuman being. That is to say, sequences of glitch, a surreal bricolage of game objects, or reconfiguring an AI system in novel ways, engenders a phenomenal instance of interobjectivity that surpasses our codified understanding of video games.² The mod’s intractability is a kind of anthropomorphic descriptor. To Bennett, providing metaphors for the way nonhuman things possess human characteristics “can catalyze a sensibility that finds a world filled not with ontologically distinct categories of beings (subjects and objects) but with variously composed materialities that form confederations.” She articulates that “revealing similarities across categorical divides and lighting up structural parallels between material forms in “nature” and those in “culture,”...can reveal isomorphisms,” or the unusual correspondences among things in the world.³ And, as Bennett asserts, nonhuman bodies find a place within a Rancierian conception of politics in their co-constitution with human actors. Through their interactions, the body, game engine, mod, and other nonhuman actants, possess the potential to repartition certain regimes of sensation. Like humans, nonhuman beings “...have the power to startle and provoke a gestalt shift in perception: what was trash becomes things, what was an instrument becomes a participant, what was foodstuff becomes agent, what was adamantine becomes intensity.”⁴

² In part, I borrow the term “intractable” as it is used by Bennet to describe the vibrant materialism witnessed in a pile of debris along a Baltimore street. Here, she quotes Stephen Jay Gould who articulates what he refers to as the “excruciating complexity and intractability” of nonhuman entities. See Stephen Jay Gould, *The Structure of Evolutionary Theory* (Cambridge: Belknap, 2002), 1338 qtd. in Bennett, *Vibrant Matter*, 4-5.

³ Bennett, *Vibrant Matter*, 99.

⁴ Bennett, *Vibrant Matter*, 107.

In the mod's capacity to act, intractability refers to its stubbornness, its refusal to render standard gamespaces or operations of play. It is the intractable "personality" of art modding—and particularly the works described in these historical episodes—that serves as an affective dynamic through which a type of aimless encounter with the video game emerges. By making space difficult or troublesome through various means of abstraction, the art mod disarms the aim of a weapon (leaving the player "aimless") within perspectival space and severs the player's optimal flow within the game. From this moment of aimlessness arises difference, unfamiliarity, and potential. In the intractable spaces of art modding, aesthetics is revealed as the affective interrelations among human and nonhuman bodies. Within this neutral ontology of beings, it describes the unfolding of experience among the body, engine, and the strange forms it generates. It is important to note that embodied meaning is a product of our encounter with the mod as a technological object *and* a form in itself. Eugenie Brinkema makes a similar argument in reference to cinema when she says that images carry affects—*anxiety, disgust, and grief*—within their form. As a textual artifact, film holds the capacity to express an "affective exteriority, an ectoaffect," through its formal composition whose meaning does not presuppose a spectator.⁵ Just as cinematic form can be read for its affects, I propose that the art mod possesses an ectoaffectivity in its formal disruption of space. When the mod interrupts the normal gamespace and subsequently the player's flow, it adheres to its own affective disposition of intractability. And just as it has the capacity to affect the body, its ontological status is not dependent upon a human participant. Rather it exists through its own autonomy, or the vital processes that occur among a constellation of technological components, images, and interface. Finally, I want to contend that modding has always been historically defined by a type of intractability, especially in the way that it deliberately works within capitalist regimes of the

⁵ Eugenie Brinkema, *The Forms of the Affects* (Durham, NC: Duke University Press, 2014), 4, 23.

industry to often subvert, deconstruct, or otherwise challenge our preconceptions of gameplay. With games such as *Doom* and *Quake*, players explored ways to alter environments and mechanics of play, working outside these engines' pre-calculated operations. Currently, communities of modders introduce new modes of play through Steam Workshop: a repository developed by Valve Corporation as part of their game distribution platform Steam, allowing for players to publish, share, and purchase modifications for popular titles such as *Skyrim* and *Team Fortress 2*. And, as Watanabe's *San Andreas Streaming Deer Cam* suggests, *GTA V* has likewise encouraged a vibrant modding community to create, for instance, a neon-saturated Los Santos reminiscent of *Blade Runner* or a vehicle cannon that uses cars as projectiles in lieu of bullets. This reterritorializing of play within dominant structures of the industry refers to what Anne-Marie Schleiner calls ludic mutation, or the player's capacity to upend the game's standard procedures. Through "a brief displacement of the dominant game or setting's normal procedures with a slightly different set of rules...", the player works to "...tak[e] back the authorial reins of game-making..." I agree with Schleiner when she says that ludic mutation, in part, resembles a "mutually beneficial, symbiotic evolution." From the open-source frameworks of *Doom* and *Quake*, to a distribution hub like Steam Workshop, players find novel ways to alter game engines, in turn, encouraging developers to adopt their innovative expressions of game design.⁶ Similarly, these histories of modding point to the ways in which artists worked within existing industry regimes to produce the new and unexpected. Early 20th century artist László Moholy-Nagy says that art emerges from the ways in which technical media—cinema and the phonograph—produce "unknown relations" instead of mere reproduction, or repeating existing

⁶ Anne-Marie Schleiner, *The Player's Power to Change the Game: Ludic Mutation* (Amsterdam: Amsterdam University Press, 2017), 11-12, 14.

modes of experience with technologies.⁷ We would do well to regard modding in a similar light, a technical practice that opens up potential through its intractability, its instability.

⁷ László Moholy-Nagy, “Production—Reproduction,” in *Moholy-Nagy*, ed. Krisztina Passuth (London: Thames and Hudson, 1987), 289.

REFERENCES

- Abrash, Michael. "Consider the Alternatives: Quake's Hidden-Surface Removal." *Ramblings in Realtime, Blue's News*, <https://www.bluesnews.com/abrash/>.
- Abrash, Michael. "Inside Quake: Visible Surface Determination." *Ramblings in Realtime, Blue's News*, <https://www.bluesnews.com/abrash/>.
- Abrash, Michael. "Quake's 3-D Engine: The Big Picture." *Ramblings in Realtime, Blue's News*, <https://www.bluesnews.com/abrash/>.
- Abrash, Michael. "The Quake Graphics Engine." Presentation notes for the Computer Game Developers Conference, April 2, 1996.
- Abrash, Michael. "Quake's Lighting Model: Surface Caching." *Ramblings in Realtime, Blue's News*, <https://www.bluesnews.com/abrash/>.
- Adang, Lisa. "Untitled Project: A Cross-Disciplinary Investigation of JODI's *Untitled Game*." *Rhizome*, Version 2, 2014, <http://media.rhizome.org/artbase/documents/Untitled-Project:-A-Cross-Disciplinary-Investigation-of-JODI%E2%80%99s-Untitled-Game.pdf>.
- Alberro, Alexander and Hans Haacke. *Working Conditions: The Writings of Hans Haacke*. Cambridge, MA: MIT Press, 2016.
- Altice, Nathan. *I am Error: The Nintendo Family Computer/Entertainment System Platform*. Cambridge, MA: The MIT Press, 2017.
- Arcangel, Cory. *Artist Cory Arcangel Discusses Super Mario Movie*. Performed by Cory Arcangel. 2009. Washington D.C.: Smithsonian American Art Museum. Digital Video. <https://www.youtube.com/watch?v=oDDCoa71-yY>.
- Arcangel, Cory. "Cory Arcangel Doesn't Even like Super Mario Brothers." *Turbulence.org*. By Eryk Salvaggio, April 2003, <http://turbulence.org/Works/curators/salvaggio/arcangel.html>.
- Arcangel, Cory. *Interview Magazine*. By Mary Heilmann, March 21, 2011, <https://www.interviewmagazine.com/art/cory-arcangel>.
- Arcangel, Cory. "John Bruneau's Interview with Cory Arcangel vs The Stefanix of _SWITCH." *Artfail.com*. By John Bruneau, <http://artfail.com/arcangel/>.
- Arcangel, Cory. "The Making of Super Mario Clouds." In *Cory Arcangel: Beige*, edited by Raphael Gyax, 106. JPR, 2006.
- Ash, James. "Emerging Spatialities of the Screen: Video Games and the Reconfiguration of Spatial Awareness." *Environment and Planning A* 41, no. 9 (2009): 2105-2124.

- Barker, Tim. "Aesthetics of the Error: Media Art, the Machine, the Unforeseen, and the Errant." In *Error: Glitch, Noise, and Jam in New Media Cultures*. Edited by Mark Nunes, 52-54, 57. New York: Continuum, 2011.
- Baumgärtel, Tilman. "On a Number of Aspects of Artistic Computer Games." *medienkunstnetz*, http://www.medienkunstnetz.de/themes/generative-tools/computer_games/.
- Benjamin, Walter. "On the Mimetic Faculty." In *Selected Writings, Volume 2: 1927–1934*. Edited by Michael W. Jennings, Howard Eiland, and Gary Smith. Translated by Rodney Livingstone and others, 721-722. Cambridge: Belknap, 1999.
- Benjamin, Walter. "The Work of Art in the Age of Its Technological Reproducibility, Second Version." In *The Work of Art in the Age of Its Technological Reproducibility and Other Writings on Media*. Edited by Michael W. Jennings, Brigid Doherty, and Thomas Y. Levin. Translated by Edmund Jephcott, Rodney Livingstone, Howard Eiland, and Others, 45. Cambridge: Belknap Press of Harvard University Press, 2008.
- Bennett, Jane. *Vibrant Matter: A Political Ecology of Things*. Durham, NC: Duke University Press, 2010.
- Bergson, Henri. *Matter and Memory*. Translated by Nancy Margaret Paul and William Scott Palmer. London: G. Allen & Co., 1912).
- Bernstrup, Tobias and Palle Torsson. "Museum Meltdown: Moderna Museet." *bernstrup.com*, 1999, <http://www.bernstrup.com/meltdown/main.html>.
- Betts, Tom. "Init() > Startup code." *www.nullpointer.co.uk*, <http://www.nullpointer.co.uk/qqq/qqq1.htm>.
- Betts, Tom. "Nonlinear Software: Tom Betts/Nullpointer Interview." *Neural*, issue 23.
- Bittani, Matteo "Game Art: (This) is not a Manifesto, (this) is a Disclaimer." In *Gamescenes: Art in the Age of Videogames*. Edited by Matteo Bittanti and Domenico Quaranta, 9. Milano: Johan & Levi, 2006.
- Boers, Waling and Annette Schindler. "The Likelihood of Disturbances." In *INSTALL.EXE-JODI*. Edited by Tilman Baumgärtel, 55. Basel: Christoph Merian Verlag, 2002.
- Bogost, Ian. *Alien Phenomenology, or what it's like to be a Thing*. Minneapolis, MN: University of Minnesota Press, 2012.
- Boluk, Stephanie and Patrick LeMieux. *Metagaming: playing, competing, spectating, cheating, trading, making, and breaking videogames*. Minneapolis, MN: University of Minnesota Press, 2017. <http://staging.manifoldapp.org/read/c5926868-00c4-45f8-8e91-45cfd9140a87/section/fb4e77c9-5347-42a4-9149-dfd009bd864f#ch03>.

- Bosma, Josephine. "Jodi and the Cargo Cult." In *INSTALL.EXE-JODI*. Edited by Tilman Baumgaertel, 91. Basel: Christoph Merian Verlag, 2002.
- Brinkema, Eugenie. *The Forms of the Affects*. Durham, NC: Duke University Press, 2014.
- Buchan, Suzanne. *The Quay Brothers: Into a Metaphysical Playroom*. Minneapolis, MN: University of Minnesota Press, 2011.
- Buchloh, Benjamin. "Andy Warhol's One-Dimensional Art: 1956–1966." In *October Files: Andy Warhol*, edited by Annette Michelson, 28. Cambridge: The MIT Press, 2001.
- Cates, Jon. "Running and Gunning in the Gallery: Art Mods, Art Institutions and the Artists that Destroy Them." *artgamestudies.org*, <http://artgamestudies.org/RunningAndGunning.pdf>.
- Cloninger, Curt. *One Per Year*. Brescia: Link Editions, 2014.
- Cremin, Colin. *Exploring Videogames with Deleuze and Guattari: Towards an Affective Theory of Form*. London, UK: Routledge, 2015.
- Danto, Arthur. "The Artworld." *The Journal of Philosophy* 61, no. 19 (1964): 571-584.
- Deleuze, Gilles and Félix Guattari. *Anti-Oedipus: capitalism and schizophrenia*. Minneapolis, MN: University of Minnesota Press, 1983.
- Deleuze, Gilles. *Cinema 1: The Movement Image*. Translated by Hugh Tomlinson and Barbara Habberjam. Minneapolis, MN: University of Minnesota Press, 1986.
- Deleuze, Gilles. *Cinema 2: The Time-Image*. Translated by Hugh Tomlinson and Robert Galeta. Minneapolis, MN: University of Minnesota Press, 1989.
- Deleuze, Gilles. *Difference and Repetition*. Translated by Paul Patton. New York: Columbia University Press, 1994.
- Deleuze, Gilles. *Francis Bacon: The Logic of Sensation*. Translated by Daniel W. Smith. Minneapolis: University of Minnesota Press, 2004.
- "Doom rendering engine." *Doomwiki*, https://doomwiki.org/wiki/Doom_rendering_engine.
- Doomworld, "A Brief DOOM Demo History," *Doomworld*, <https://www.doomworld.com/10years/demos/demos01.php>.
- Dovey, Jon and Helen W. Kennedy. *Game cultures: computer games as new media*. Maidenhead: Open Univ. Press, 2011.
- Flanagan, Mary. *Critical play: radical game design*. Cambridge, MA: MIT Press, 2013.

- Foster, Hal. *The Return of the Real: The Avant-Garde at the End of the Century*. Cambridge, MA: MIT Press, 1996.
- Foucault, Michel. *The Archaeology of Knowledge and the Discourse on Language*. Translated by A.M. Sheridan Smith. New York: Pantheon Books, 1972.
- Foucault, Michel. *Manet and the Object of Painting*. Translated by Mathew Barr. London: Tate Publishing, 2009.
- Gaboury, Jacob. "Hidden Surface Problems: On the Digital Image as Material Object." *Journal of Visual Culture* 14, no. 1 (2015): 40-60.
- Gaboury, Jacob. "Perspective." In *Debugging game history: a critical lexicon*. Edited by Raiford Guins and Henry Lowood, 364. Cambridge, MA: The MIT Press, 2016.
- Galloway, Alexander. *Gaming: Essays on Algorithmic Culture*. Minneapolis, MN: University of Minnesota Press, 2006.
- Goriunova, Olga and Alexei Shulgin. "Glitch." In *Software Studies: A Lexicon*, edited by Matthew Fuller, 110, 114. Cambridge, MA: MIT Press, 2008.
- Groys, Boris. *Art Power*. The MIT Press, London, 2008.
- Gunning, Tom. "Lewis Klahr's The Pharaoh's Belt and the Amnesia of Modernity." Presentation at the Der Blick der Moderne conference, Vienna, June 1996.
- Hackett, Pat and Andy Warhol. *POPism: The Warhol '60s*. New York: Harcourt Brace Jovanovich, 1980.
- Hall, Tom. "Doom Bible." (PDF, id Software, 1992).
- Hansen, Mark B. N. *New philosophy for new media*. Cambridge, MA: MIT Press, 2006.
- Hansen, Miriam. *Cinema and Experience: Siegfried Kracauer, Walter Benjamin, and Theodor W. Adorno*. Berkeley, CA: University of California Press, 2012.
- Hesprich, David. *QuakeC Reference Manual*, <http://pages.cs.wisc.edu/~jeremyp/quake/quakec/quakec.pdf>.
- Holmes, Brian. "Extradisciplinary Investigations. Towards a new critique of Institutions." In *Art and Contemporary Critical Practice: Reinventing Institutional Critique*. Edited by Gerald Raunig and Gene Ray, 57. London: MayFlyBooks, 2009.
- Hook, Brian and Paul Jaquays. "Q3Radiant Shader Manual, Revision #12." [icculus.org, https://icculus.org/gtkradiant/documentation/Q3AShader_Manual/ch01/pg1_1.htm#what](https://icculus.org/gtkradiant/documentation/Q3AShader_Manual/ch01/pg1_1.htm#what).

- Jansson, Mathias. "First Museum Shooters." In *The Pioneers of Game Art: From ArsDoom to SimBee*. Edited by Mathias Jansson, 8-9. Akarp, Sweden: Mathias Jansson, 2015.
- JODI. "autoexec.cfg." [www.untitled-game.org](http://www.untitled-game.org/autoexeccfg.html), accessed June 2017, <http://untitled-game.org/autoexeccfg.html>.
- JODI. "Perspective Engines: An Interview with Jodi." By Francis Hunger. In *Videogames and Art*. Edited by Andy Clarke and Grethe Mitchell. Bristol, UK: Intellect Ltd., 2007.
- JODI. "progs." [www.untitled-game.org](http://www.untitled-game.org/progsx/miscqc.html), accessed June 2017, <http://www.untitled-game.org/progsx/miscqc.html>.
- Joseph, Branden Wayne. *Beyond the Dream Syndicate: Tony Conrad and the arts after Cage*. New York: Zone Books, 2008.
- Joseph, Branden Wayne. *Random Order: Robert Rauschenberg and the Neo-avant-garde*. Cambridge, MA: The MIT Press, 2007.
- Kipcak, Orhan. (ArsDoom, ArsDoom II) (1995-2005)." *Gamescenes.org*. By *gamescenes.org*. November 4, 2009, <http://www.gamescenes.org/2009/11/interview-orphan-kipcak-arsdoom-arsdoom-ii-1995.html>.
- Kipcak, Orhan. "ARSDOOM - art adventure." *Ars Electronica Archive*, 1995. http://90.146.8.18/en/archives/festival_archive/festival_catalogs/festival_artikel.asp?iProjectID=8643.
- Kirschenbaum, Matthew. *Mechanisms: New Media and the Forensic Imagination*. Cambridge, MA: MIT Press, 2012.
- Klahr, Lewis. "A Clarification."
- Kushner, David S. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture*. New York: Random House, 2004.
- LeBreton, Jean-Paul. "Coelacanth: Lessons from *Doom*." *vector poem*, February 28th, 2010, <http://vectorpoem.com/news/?p=74>.
- Lee, Pamela M. *Chronophobia: on time in the art of the 1960s*. Cambridge, Mass: MIT Press, 2006.
- LeMieux, Patrick. *Everything but the Clouds*. Digital Video. Directed by Patrick LeMieux. 2017. New York: Platform Games at Babycastles, <https://vimeo.com/241966869>.
- Lowood, Henry. "Game Engine." In *Debugging game history: a critical lexicon*. Edited by Raiford Guins and Henry Lowood. Cambridge, MA: The MIT Press, 2016.

- McCloskey, D.I. "Proprioception and Related Somatosensory Mechanisms." in *Proprioception, Posture, and Emotion*, ed. David Garlick, 14-22. University of New South Wales, Australia: Committee of Postgraduate Medical Education, 1982.
- Meillassoux, Quentin. *After Finitude: An Essay on the Necessity of Contingency*. Translated by Ray Brassier. London: Bloomsbury Academic, 2017.
- Menkman, Rosa. *The glitch moment(um)*. Amsterdam: Institute of Network Cultures, 2011.
- Menkman, Rosa. "Glitch Studies Manifesto." In *Video Vortex Reader II: Moving Images beyond YouTube*, edited by Geert Lovink and Rachel Somers, 342. Amsterdam: Institute of Network Cultures, 2011.
- Merleau-Ponty, Maurice. "The Intertwining- The Chiasm." In *The Merleau-Ponty Reader*. Edited by Ted Toadvine and Leonard Lawlor. Evanston, IL: Northwestern University Press, 2007.
- Merleau-Ponty, Maurice. *Phenomenology of Perception*. Translated by Colin Smith. London: Routledge, 2005.
- Merleau-Ponty, Maurice. *The visible and the invisible: Followed by working notes*. Edited by Claude Lefort. Translated by Alphonso Lingis, 110-111. Evanston, IL: Northwestern University Press, 1968.
- Mickle, Jazz. *Doom is an Art Scene*. Performed by Jazz Mickle. London: VideoBrains, 2015. Digital video. <https://www.youtube.com/watch?v=KxYND6K6u8w>.
- Moholy-Nagy, László. "Production—Reproduction." In *Moholy-Nagy*. Edited by Krisztina Passuth, 289. London: Thames and Hudson, 1987.
- Morgan, Daniel. *Late Godard and the possibilities of cinema*. Berkeley, CA: University of California Press, 2013.
- Murakami, Takashi. "A Theory of Super Flat Japanese Art." In *Super Flat*. Edited by Takashi Murakami, 193-194. Tokyo: MADRA Publishing, 2000.
- Murray, Janet. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Cambridge, MA: MIT Press, 1998.
- Nideffer, Robert. "Game Engines as Embedded Systems." In *Database Aesthetics: Art in the Age of Information Overflow*. Edited by Viktorija Vesna, 219,221. Minneapolis, MN: University of Minnesota Press, 2010.
- Nitsche, Michael. "Demo." In *Debugging game history: a critical lexicon*. Edited by Raiford Guins and Henry Lowood, 104. Cambridge, MA: The MIT Press, 2016.

- Nitsche, Michael. "Machinima as Media." In *The Machinima Reader*. Edited by Henry Lowood and Michael Nitsche, 118. Cambridge: MIT Press, 2011.
- Oliver, Julian. "Delire (Julian Oliver)." By Domenico Quaranta. In *Gamescenes: Art in the Age of Videogames*. Edited by Matteo Bittanti and Domenico Quaranta, 131. Milan: Johan and Levi Editore, 2006.
- Oliver, Julian. "ioq3aPaint." *julianoliver.com*, <https://julianoliver.com/output/ioq3aPaint>.
- Parikka, Jussi. "CTheory Interview: Archaeologies of Media Art." By Garnet Hertz. *ctheory.net*, April 1, 2010.
- Paul, Christiane. "Cory Arcangel: Pro Tools." *Whitney Museum of American Art*, 2011.
- Rancière, Jacques. *The politics of aesthetics: The distribution of the sensible*. Edited and translated by Gabriel Rockhill. London: Bloomsbury, 2015.
- Richmond, Scott C. *Cinema's bodily illusions: flying, floating, and hallucinating*. Minneapolis, MN: University of Minnesota Press, 2016.
- Romero, John. *QUAKETALK 95*. By Ed Dille. October 22nd 1995, <http://rome.ro/?offset=1481382456223&author=51a4f9c2e4b068f76a150e2a>.
- Romero, John. *QUAKETALK 95*. By *Multimedia Magazine*. October 22nd 1995, <http://rome.ro/?offset=1481382456223&author=51a4f9c2e4b068f76a150e2a>.
- Sanglard, Fabien. "Doom Code Review." *Fabien Sanglard's Website*, January 13th 2010, <http://fabiensanglard.net/doomIphone/doomClassicRenderer.php>.
- Sanglard, Fabien. "Quake 3 Source Code Review: Architecture." *Fabien Sanglard's Website*, June 30th 2012, <http://fabiensanglard.net/quake3/>.
- Schleiner, Anne-Marie. *The Player's Power to Change the Game: Ludic Mutation*. Amsterdam: Amsterdam University Press, 2017.
- Sharp, John. *Works of Game: On the Aesthetics of Games and Art*. Cambridge, MA: MIT Press, 2015).
- Shinkle, Eugénie. "Video games, emotion and the six senses." *Media, Culture & Society* 30, no. 6 (2008): 907-915.
- Shklovsky, Viktor. "Art as Technique." in *Russian Formalist Criticism: Four Essays*. Edited by Lee T. Lemon and Marion J. Reiss, 18, 22. Lincoln, NE: University of Nebraska Press, 1965.

- Sitney, P. Adams. *Visionary Film: The American Avant-garde, 1943-2000*. Oxford: Oxford University Press, 2002.
- Schrank, Brian. *Avant-garde videogames: playing with technoculture*. Cambridge, MA: The MIT Press, 2014.
- Steed, Paul. "Putting New Models in Quake III Arena." Edited by Paul Jaquays, *icculus.org*, https://icculus.org/projects/gtkradiant/documentation/Model_Manual/model_manual.htm.
- Stockenburger, Axel. "From Appropriation to Approximation." In *Videogames and Art*. Edited by Andy Clarke and Grethe Mitchell, 33. Bristol, UK: Intellect Ltd., 2007.
- Tanke, Joseph J. *Foucault's Philosophy of Art: A Genealogy of Modernity*. London: Continuum International Publishing, 2009.
- Tomkins, Calvin. *The Bride and the Bachelors: Five Masters of the Avant-Garde*. New York: Penguin Books, 1976.
- Transmediale Festival, "ioq3apaint." www.transmediale.de, <https://transmediale.de/content/ioq3apaint>.
- Van Waveren, J.M.P. "The Quake III Arena Bot." Thesis, Delft University of Technology, 2001.
- Varnedoe, Kirk. "Pictures of Nothing: Abstract Art since Pollock, Part 5: Satire, Irony, and Abstract Art." Lecture at the National Gallery of Art, Washington D.C., May 4, 2003.
- Watanabe, Brent. "Hundreds of People Are Watching a Deer Run Around a Video-Game City, and It's Mesmerizing." By Brian Feldman. *New York Magazine*, March 25, 2016, <http://nymag.com/selectall/2016/03/grand-theft-auto-deer-cam-is-mesmerizing.html>.
- Ward, Frazer. "The Haunted Museum: Institutional Critique and Publicity." *October* 73 (1995): 71-89.
- Whitehead, Alfred North. *The Concept of Nature*. Amherst, NY: Prometheus Books, 1920/2004.
- Whitehead, Alfred North. *Process and Reality*. New York: Free Press, 1929/1978.
- Whitehead, Alfred North. *Science and the Modern World*. New York: Free Press, 1925/1967.
- Witkowski, Emma. "On the Digital Playing Field: How we "do sport" with Networked Computer Games." In *Games and Culture* 7, no.5 (2012): 349-374.
- Yang, Robert. "A People's History of the FPS, Part 1: The WAD." *Rock Paper Shotgun*, September 19th, 2012, <https://www.rockpapershotgun.com/2012/09/19/a-peoples-history-of-the-fps-part-1-the-wad/>.