

Gay Engines:

Imagining a Queer Development Platform for Video Games

Cass Zegura

MCM 1990: Honors Thesis/Project in Modern Culture and Media

April 5, 2020

Preface

The story of this project begins in the fall of 2017, at a crowded tapas bar in downtown Providence. My aunt and uncle, who live in Boston, had come down to visit and take me out to dinner. It was a perfectly ordinary night except in one regard: when I told my aunt and uncle about the classes I was taking that semester, a strange look crossed their faces. I thought the source of their confusion might be the computer science course I had mentioned, an upper-level design class called Independent Study in 2D Game Engines. I assumed that they, like I, had never heard of the term “game engine” before and thus did not know what it was. And as I launched into my explanation (which I won’t repeat here—I’ve already written an entire chapter dedicated to this question), the confusion diminished, but only somewhat, and we soon changed subjects.

On a phone call with my mom a few days later, I learned the true source of my aunt and uncle’s befuddlement. In that noisy restaurant, the word “game” had transformed into the word “gay.” They thought that I was taking a class on 2D *gay* engines. No wonder they were confused! I would’ve been too. What the heck is a gay engine anyway?

Of course, I didn’t know then that I was going to spend my senior year trying to answer that question. In fact, it probably never would have occurred to me again were it not for a class I took a year after that dinner. In the fall of 2018, I took *Queerness & Games* with Dr. Teddy Pozo. This class, quite simply, changed my life forever. I already loved video games. Though I had only briefly been exposed to queer theory before, I was fascinated by it and eager to learn more. And *Queerness & Games* not only gave me more exposure to both, but it paired them together into a new and fascinating field of academic inquiry: queer games studies.

But I wouldn't reach my thesis topic for another semester. I knew, broadly, that I wanted to do something in queer games, but my specific point of interest—gay engines—wouldn't come until I took Introduction to Video Games Studies, also with Dr. Pozo. One of the weeks of the class was dedicated to platform studies, at the time an unfamiliar discipline but one which I also took to quickly. I was struck by how platform studies professed to be about studying the technological alongside the cultural. As someone who wanted to scream “WHY NOT BOTH?” each time they learned about the ludology vs. narratology debate, platform studies seemed like the perfect framework, not only in how it approached the objects of its study but also those objects themselves. See, the game engine had lingered in my mind since the fall of 2017, and as I began to use software like Unity, Twine, and Bitsy for my own creative practice, I became all the more interested in studying the game engine as a technical object *and* piece of cultural software. Platform studies could allow me to do just that.

And if there were queer games, why couldn't there be queer game engines?

Thus I had my thesis topic, a profoundly ambitious project to establish, defend, and contribute to a new subset of queer games studies.

This project has changed a lot since I started it. It was initially even grander in scope. My plan was to spend Chapter 3 examining six components commonly found in game engines and analyzing how and to what extent they could be queered. I only ended up doing two of the original six, not because I had nothing to say about the other four (which were UI, scripting, AI, and input event handling) but simply because I did not have time. I also intended to talk more about the game engine's role in game development, closely analyzing how the queer game engine could potentially mediate a more ethical game design process and asking about its

relationship to production given the prominence of theories which locate queerness in a failure to (re)produce and succeed within capitalism.¹ I wasn't able to get around to that either.

The project of the queer game engine remains terribly incomplete, but that isn't a bad thing. It means that I can return to it later in my academic career. And, perhaps more importantly, it means that I can open the project up to others. Nothing would excite me more than to see someone else continue this work and take it to places I could never have before imagined.

I owe what imagining I've been able to do thus far to a number of people. I'd like to acknowledge first my family, both the aunt and uncle who inadvertently inspired the title of this piece, and my mom, dad and sister, who I'm pretty sure don't entirely understand what I've been working on this year but have nonetheless always supported me.

I'd also like to thank the various mentors I've had over the years, including both my wonderful college professors and my mom's Georgia Tech faculty friends, who have given me a lot of advice, particularly about grad school. I want to specifically express gratitude toward Arnav Jhala, who was my faculty mentor when I did summer research at North Carolina State University, and James Tompkin, who was kind enough to be my secondary reader for this thesis and has since put up with the utter lack of communication on my end. So very sorry about that!

A very special and heartfelt thanks goes to King House. I would not be who I am today if I had not found such a loving community on campus. You all have always supported me in both my academic and literary passions and also my emotional and mental trials and tribulations. I

¹ Jack Halberstam, *The Queer Art of Failure* (Durham: Duke University Press, 2011).

love you all so very very much, and it saddens me greatly that I will not be able to present this thesis for you all—at least, not any time soon.

Two members of King House deserve a special shoutout: Mary Woodard, who has always inspired me to write and love better and who told me about the Queerness & Games class in the first place, and Ari Kintisch, my best friend and currently the only person besides myself and my thesis advisor who has read any of this work.

Finally, I have to, of course, acknowledge that thesis advisor. Teddy Pozo, I've told you this before, but you truly changed my life and my academic trajectory for the better. I couldn't have asked for a kinder and more supportive mentor. You've always seen the potential of my work and known how to push me to more fully realize that potential. I'm not sure I would've ever had the idea for this thesis were it not for Queerness & Games, and I certainly wouldn't have been able to develop a paper this strong without your guidance.

This thesis is the culmination of a lot of long hours and a *lot* of love, and that makes all those hours, all that work and all that effort, worth it. And it is all the more worth it to think of someone else reading this and being inspired by it as I have been inspired by countless queer games academics.

And to that future reader, I say: welcome to my passion project!

Chapter 1: Representation, Rules, and Rupture

In the forward to *Gaming Representation* (2017), Anna Everett warns of the “neo-formalist tech turn to platform, software, and code” within games studies.¹ This shift is marked by a preoccupation with the technical above all else, while questions of identity, culture, and politics are ignored or isolated out of fear that they “might prevent one from accessing the technical nature of the computer.”² Yet as Jennifer Malkowski and TreaAndrea M. Russworm point out in the book’s introduction, “representation is not fully separate from the implicitly hard-core elements of games: it is achieved through and dependent on player and machine actions, on code, and on hardware, not just on surface-level images and sounds.”³ In other words, one can analyze platform, software, and code through frameworks of representation and cultural theory. There are numerous examples of such an approach, like Teddy Pozo’s analysis of how a change in hardware changes the experience of playing Mattie Brice’s *Mainichi* (2012)⁴ or Jordan Wood’s commentary on how a procedural generation algorithm imbues *The Binding of Isaac* (Edmund McMillen, 2011) with queer affects of fluidity and unknowability, even if the game lacks queer characters.⁵

Despite these valuable contributions, a number of the more technical aspects of gaming and game design remain undertheorized from a digital humanities perspective. One such aspect, and the focus of this work, is the game engine: a software platform which lays the foundation for

¹ Anna Everett, forward to *Gaming Representation: Race, Gender, and Sexuality in Video Games* (Bloomington: Indiana University Press, 2017), ix.

² Tara McPherson, “Designing for Difference,” *differences* 25, no. 1 (January 2014) doi: 10.1215/10407391-2420039, 179.

³ Malkowski and Russworm, introduction to *Gaming Representation*, 3.

⁴ Teddy Pozo, “Queer Games After Empathy: Feminism and Haptic Game Design Aesthetics from Consent to Cuteness to the Radically Soft,” *Games Studies* 18, no. 3 (2018): <http://gamestudies.org/1803/articles/pozo>.

⁵ Jordan Wood, “Romancing an Empire, Becoming Isaac: The Queer Possibilities of *Jade Empire* and *The Binding of Isaac*,” in *Gaming Representation*, ed. Jennifer Malkowski and TreaAndrea M. Russworm (Indiana University Press, 2017), 224.

game development by providing some initial functionality. Essentially, by using a game engine, designers do not need to code their games from scratch. This makes the game engine the core of most modern games, and positions it as a site of control, as it is able to shape and limit a game's development.⁶ Thus, studying the game engine critically and culturally is not merely a question of personal interest or curiosity, but an imperative in order to expose its mechanisms of power which remain largely invisible. This necessity is being realized, as is evidenced by works such as *The Unity Game Engine and the Circuits of Cultural Software* (Brendan Keogh and Benjamin Nicoll, 2019) and Eric Freedman's "Engineering Queerness in the Game Development Pipeline." But the game engine is a large, complex piece of software, and its complete analysis from a digital humanities perspective will require far more work.

I view my own work here as a small contribution to understanding and reframing the game engine, specifically within the context of queer theory, which I am drawn to as a tool of intervention for three reasons. First, there is a fundamental synergy between queerness and digital technologies. This connection is perhaps best elucidated by Zach Blas and Micha Cárdenas, who explain that queerness "[is] invested in creating logics and codes" much like digital technologies are built upon and propagate logics and codes.⁷ Secondly, because of this affinity between queerness and the digital, queer theory has already made significant interventions into video games in the form of queer games studies. This "nascent yet thriving academic area" is concerned with both the representation of LGBTQ characters in games and queerness beyond representation—as "a way of designing a game, interpreting it, or playing it."⁸

⁶ Eric Freedman, "Engineering Queerness in the Game Development Pipeline," *Game Studies* 18, no. 3 (2018): <http://gamestudies.org/1803/articles/ericfreedman>, n.p.

⁷ Zach Blas and Micha Cárdenas, "Imaginary computational systems: queer technologies and transreal aesthetics," *AI and Society* 28 (2013), 63.

⁸ Bo Ruberg, *Video Games Have Always Been Queer* (New York: New York University Press, 2019), 7, 15.

Such analyses offer a more specific starting point for considering queerness alongside the game engine. Finally, by using queerness as my guiding principle, I am allowed, indeed required, to go beyond analysis of existing game engines and propose new, alternative ones. As Sara Ahmed says, “to queer phenomenology is also to offer a queer phenomenology ... queer does not have a relation of exteriority to that with which it comes into contact.”⁹ Thus, to bring queerness into conversation with the game engine is to propose that, just as there are queer games, there are—or could be—queer game engines.

The intention of this work is to begin to imagine the queer game engine by revealing the queerness already present, but perhaps hidden, within existing game engine architecture. In order to approach this goal, however, it is first necessary to study its terms. Queer, game, and engine are, after all, nebulous words, and a wealth of literature has already been dedicated to clarifying them. I do not intend to do away with this vagueness altogether; there is value in slippage between meanings, in instability and porous boundaries. Yet in the interest of making my writing legible to queer theorists and computer scientists alike, I would like to begin by establishing, if not a universal definition, at least a general understanding of how I am using these fuzzy words. Finally, rather than considering each term separately, I will be examining them in pairs: queer+game, game+engine, and, eventually, queer+game+engine. In doing so, I will have a point of focus within the broad associations of meanings attached to each of these words.

Consider, for example, queer+game. It would be impossible to talk comprehensively and briefly about either term on its own, yet, when coupled together, an organization suggests itself. Specifically, I will be borrowing from the two understandings of queerness which Ahmed

⁹ Sara Ahmed, “Orientations: Toward a Queer Phenomenology,” *GLQ: A Journal of Lesbian and Gay Studies* 12, no. 4 (2006): 544.

presents later in “Orientations: Toward a Queer Phenomenology” in order to study both different types of queerness and different expressions of queerness within games. Rather than presenting a universal definition of either queer or game, this approach reveals through specific examples and analyses how these terms are understood throughout the rest of this work.

So, what are Ahmed’s two definitions of queerness? First, she uses it “to describe what is oblique or off-line or even just plain wonky,” and second, “to describe nonstraight sexual practices ... as a form of social and sexual contact.”¹⁰ This is not to suggest that these two understandings of queerness are neatly separated. As Ahmed points out, the two exist in constant dialogue, a slippery relationship of signification wherein, for example, certain sexualities came to be described as queer, a word which already meant odd, because they were at odds with the established norm of heterosexuality.¹¹ Furthermore, these terms do not form a polarizing binary between which all other inflections of queer must be located, nor do they encompass all of its permutations and manifestations. Queerness can, for example, also describe a personal relationship to gender, as in the inherent “wonkiness” of the transgender identity in relation to cisnormativity.

Nonetheless, Ahmed’s two general meanings of queerness—the oblique and anti-normative, and the identity—offer a strong model for structuring my own conversations about queerness and games. This chapter is organized around three points of interaction between these categories. The first two relate to the understanding of queerness as an identity, while the last is rooted in queerness as the crosswise, adverse, or disruptive. Within each of these points of intersection, I will also examine the relevance to the queer game engine. Here, relevance refers to

¹⁰ Ahmed, “Orientations,” 565.

¹¹ Ahmed, 565.

whether that specific expression of queerness can be located at the level of the game engine as well. Can the questions it asks of games and the gaming industry also be asked of the game engine? Can it be produced by the game engine? The answers to these questions will offer an initial possibility space, as well as some loose limitations, when it comes time to finally consider what the queer game engine might be.

Queerness and Representation

Within the realm of queerness as an orientation of desire and gender, perhaps the most obvious connection between it and games has to do with the representation of characters who are skewed in their orientations. There is a long, if undertheorized and scarcely acknowledged, history of LGBT characters in games which runs counter to the prevalent notion that LGBT people, women, and people of color are absent from every part of video game history. For example, the alleged first game to have a lesbian character was a 1986 Infocom interactive fiction detective game called *Moonmist*.¹²

The more modern examples of LGBT representation largely fall into two categories with very different tendencies in how that representation is handled: that in mainstream gaming, and that in queer indie gaming. Queer indie gaming is usefully distinguished from broader indie gaming by both the sexuality and gender identity of the creator and the scale of the production, with the assumption being that queer indie games are small and personal, and thus made, primarily, by one person whose personal experiences are reflected in the game's content.¹³

¹² "Vivien in Moonmist," *LGBTQ Game Archive*, accessed December 19, 2019, <https://lgbtqgamearchive.com/2015/09/11/vivien-in-moonmist/>.

¹³ Brendan Keogh, "Just making things and being alive about it: The queer games scene," *Polygon*, May 24, 2013, <https://www.polygon.com/features/2013/5/24/4341042/the-queer-games-scene>.

Fundamentally, queer indie games draw directly and primarily from the lived experiences of their LGBT creators, while indie games may feature LGBT characters in narratives that are not about identity,¹⁴ or present queerness as a prominent plot point yet be made by straight creators.¹⁵ This distinction reveals the difficulty in making generalized statements about queer representation in (not-queer) indie games, which is highly variable in both extent and quality. For this reason, rather than attempting to making a broad claim about queer representation in indie games, I will maintain the mainstream/queer indie gaming dichotomy, with the understanding that a given not-queer indie game can likely be located somewhere between the two in evaluating how it handles queerness.

Within mainstream game development, examples of queer representation are steadily increasing at least in number.¹⁶ Unfortunately, the quality of that representation may not be advancing at a rate to justify praising these games for their supposedly revolutionary nature. When Blizzard Entertainment revealed that one of its characters from *Overwatch* (2016), Tracer, was a lesbian, it was hailed as historic.¹⁷ Blizzard may deserve some praise given that Tracer is heavily featured in the game's marketing, but whether this action was historic is questionable. After all, being a multiplayer first-person shooter, *Overwatch* does not focus much on the interiority of its characters—indeed, Tracer's sexuality was confirmed not within the game itself but within a promotional comic.¹⁸ Besides, there have been lesbian characters in AAA games

¹⁴ See *Night in the Woods* (Infinite Fall, 2017).

¹⁵ See *Dream Daddy: A Dad Dating Simulator* (Game Grumps, 2017).

¹⁶ Ruberg, *Video Games Have Always Been Queer*, 4.

¹⁷ AJ Moser, "Overwatch's historic openly gay character draws (mostly) praise from gamers," *The Daily Dot*, December 21, 2016, <https://www.dailydot.com/parsec/overwatch-tracer-reactions/>.

¹⁸ Michael Chu and Miki Montlló, "Reflections," 2016, https://assets.blz-contentstack.com/v3/assets/blt2477dc4f4ebd440c/blt26451d9115028b05/5d02e33f7b48be290a7f964d/comic-overwatch-reflections_en-us.pdf.

since at least 2003 with Juhani from *Star Wars: Knights of the Old Republic* (BioWare), and Juhani's sexuality is part of the game play. She is a lesbian in the sense that a female player character can enter into a relationship with her, and a male player character cannot.

Still, even this moment should be contextualized within the broader narrative of LGBT representation in Western role-playing games. It is, unfortunately, difficult to say that BioWare's LGBT representation has advanced substantially since then. Certainly, several LGBT characters, both romanceable and non, have made appearances in several of their titles such as in the *Dragon Age* (2009-2014) and *Mass Effect* (2007-2017) series. However, few speak to the lived experiences of queer people because the character's identities are presented in a sort of apolitical vacuum, a homophobia- and transphobia-free utopia.

For these characters, sexuality is merely a matter of coding—literally. When the game reaches a point where it could give the player the option to flirt with a character, it checks the player character's sex. This means looking at one of two Booleans (a data type which can only ever be true or false): Player is Male or Player is Female. This Boolean check is tucked into an if-else statement, where if the player has the correct sex (male for a gay male NPC, female for a straight male NPC, etc.), the flirtatious dialogue option appears. If the player is not correctly aligned, generally speaking, that option does not show up. For bisexual NPCs, this check usually does not occur, since a player of either sex is a potential partner.¹⁹

A notable exception to the formula described above is Dorian Pavus, from *Dragon Age: Inquisition* (BioWare, 2014). For one, Dorian's personal quest is about his identity as a gay man.

¹⁹ However, even for bisexual characters, flirtatious dialogue options can be restricted by gender. In an early conversation with Leliana, from *Dragon Age Origins*, for example, a particular flirty line ("Those initiates can't have been more lovely than you.") is only available to male player characters, despite Leliana being bisexual: <https://dragonage.fandom.com/wiki/Leliana/Approval>.

Specifically, the player character must help him navigate his relationship with his homophobic father, who attempted to use a kind of magical, fantasy conversion therapy on him. Unlike that of most other queer characters in the series, Dorian's sexual orientation has a political dimension and is not completely normalized.

His sexuality also works differently at a mechanical level compared to the other non-bisexual characters in the franchise. Unlike the first game in the series, *Dragon Age: Origins* (BioWare, 2009),²⁰ rather than simply not giving them the flirtatious dialogue option, *Inquisition* gives its non-bisexual characters the chance to turn down player characters who are not the gender they are attracted to. For example, if the player, as a woman, flirt with one of the game's straight female characters, Cassandra Pentaghast, she will reject the player character by reasserting her heterosexuality, and the flirtatious dialogue options will never appear again.

With Dorian, however, his writer David Gaider decided that female player characters as well as male should be able to see flirtatious dialogue. Not only that, they should be able to select it, flirt with Dorian, and then have him flirt back. After Dorian's personal quest, he explains that he viewed their interactions as playful rather than serious expressions of interest. Beyond the questionable optics of allowing a woman to be angry or upset with a gay man for his sexuality,²¹ Gaider's decision has problematic consequences at the level of the game's code. Because female player characters can flirt with Dorian—meaning that the sex-specific Boolean checks discussed earlier do not occur—it becomes that much easier for fans to change his sexuality within the

²⁰ All romance options in *Dragon Age 2* (BioWare, 2011) are bisexual and thus do not pose the kind of restrictions being discussed here.

²¹ In so far as a player of any gender, playing as a female player character, can select angry or upset dialogue options in response to learning about Dorian's sexuality—although the presence of both mods (note 22) and fanfiction created by female fans in order to make Dorian available to female player characters suggests these dynamics extend beyond the screen.

code. The relative weakness of how sexuality generally is encoded allowed fans to queer straight characters in the first game by changing Boolean values or circumventing the if-else statements,²² yet in *Inquisition*, it also lets them obliterate Dorian's queerness.²³

Worth finally mentioning is that BioWare's representation of transgender characters is woefully lacking. *Inquisition* introduced the series' first transgender character,²⁴ a transgender man named Cremisius Aclassi, or Krem. While a notable step in the right direction, the size of that step is diminished by the fact that his voice actress is a cisgender woman, his gender identity is only confirmed if the player has a sufficiently good relationship with another character, and a number of dialogue options around Krem's identity are, if not blatantly transphobic, messily worded.²⁵

However, far more successful representations of queerness in video games abound—representations which reject the kind of logic put forth by developers like David Gaider who worry about “[raising] the specter of ‘the gay thing’ if [they] talk about [LGBT issues] too much, as if it were the only thing with which [they’re] concerned.”²⁶ These are queer indie games, and many of them are *only* concerned with representing gender and sexuality. A powerful counterpoint to the representations of Krem and Dorian, for example, is Anna Anthropy's *dys4ia*

²² <https://www.nexusmods.com/dragonage/mods/4659/> (GoesOnGhost, 2017).

²³ <https://www.nexusmods.com/dragonageinquisition/mods/616> (Atherisz, 2015) Of note is how (because of the Frostbite engine) having this mod installed makes it impossible to romance Dorian with a male Inquisitor.

²⁴ An expansion pack for *Dragon Age 2*, “Mark of the Assassin,” includes a transmisogynistic portrayal of a drag queen sex worker (<https://dragonage.fandom.com/wiki/Serendipity>), yet her writers have specified that they did not intend for her to be read as a transgender woman

(<https://the-gaider-archives.tumblr.com/post/108930086798/on-transgender-characters>).

²⁵ Krem comes out to the main character in a scene where he talks about binding. The player character's response options are “Why pass as a man?”, “Are you a woman?”, “When did you know?”, and “And the rest of your crew?”. The game invites the player to doubt and question Krem's identity in two of these options instead of affirming it or, better yet, respecting Krem's privacy and not needling him with unprompted questions.

²⁶ David Gaider, “A Character Like Me: the lead writer of Dragon Age on inclusive games,” *Polygon*, February 18, 2014, <https://www.polygon.com/2014/2/18/5422570/the-lead-writer-of-dragon-age-on-the-first-steps-towards-inclusive>.

(2012). This autobiographical Adobe Flash game narrates Anthropy's real experiences with hormone replacement therapy, from the initial motivation and medical hurdles to the slow morphing of her body. It is divided into a series of minigames simulating everything from navigating a woman's bathroom to the abstract sensation of having a body that does not "fit". In *dys4ia*, there is not a single game mechanic which is not used to represent Anthropy's experiences as a trans woman. Not only can the queer representational content of *dys4ia* not be destroyed, it also cannot be ignored. While a player in *Dragon Age: Inquisition* has the option not to recruit Dorian or to ignore his personal quest, *dys4ia* does not give its players such a choice. Finally, *dys4ia* points to a possibility for representing queerness not through identities superficially applied to non-player characters and easily modded over, but rather as a deep, integral part of a game's mechanics.

There are a plethora of other mechanical representations of queerness-as-identity, such as the more abstract *Lim* by meritt kopas (2012). In *Lim*, the player controls a block which can either flash rainbow or adopt the color of the block nearest it. Such an adoption forces the player into an uncomfortable zoom-in where the game slows down and vibrates as disquieting sound effects play. However, the single-colored blocks will push the player's block around if it is not the same color as them. The player must endure extreme discomfort if they are to make progress—but that progress is ultimately rewarded by encountering an entity like the player's block. Together, these two blocks, violently harassed by the rest of the world, flash in queer rainbow glory.

It must be noted that *Lim* and *dys4ia* together have been made to represent a sort of queer video game "canon", a select number of games which, around 2012, "gained unprecedented

critical acclaim in the independent games industry.”²⁷ As Teddy Pozo warns, reading these games “through dreams of diversity [risks] emphasizing the games’ potential to teach cisgender straight gamers about queer and trans lives, over the artistry of their creators and their appeal to queer audiences.” To talk about them here requires engaging with this history, as well as the fact that both games have been removed from the Internet and are retrospectively viewed by their creators with a degree of frustration.²⁸ In mentioning them, my intention is not to further reify this incredibly narrow canon or to talk about their potential for fostering empathy in cisgender straight gamers. Rather, what I would like to point to *is* their artistry and how that artistry affects how these games represent queer lives.

Discussing *Lim* and *dys4ia* together reveals a distinction in representational queerness which is pertinent to the question of the queer game engine. One might say that *dys4ia* is explicit in its representational queerness—no work is required to explain why *dys4ia* is a queer game because it is self-evident in its narrative. By contrast, it is difficult to claim that rectangles have sexual orientations or gender identities. Calling *Lim* a queer game thus points in a different direction for queerness-as-identity, one predicated less on the traditional definition of representation and more on a natural affinity between sexuality and games as being rule-based systems. Representations of queerness which are not narrative but mechanical.

The value of such a distinction lies in the fact that the traditional understanding of representational queerness as queer characters is a limited entry point for considering the queer game engine. Fundamental to the definition of the game engine (which will be expanded upon in Chapter 2) is that it specifies the separation between itself and the game. On one side, there is

²⁷ Pozo, “Queer Games After Empathy.”

²⁸ Pozo, “Queer Games After Empathy.”

what can be general to many different games, and on the other, that which makes each game unique, including, for example, “the art assets, game worlds, [the] rules of play that [comprise] the player’s gaming experience,” and much more.²⁹ The former is the game engine, the latter the game. Generally speaking, representational queerness—defined by clearly identifiable sexualities and/or gender identities—are only found within the game, not the engine.

This is not to say that the game engine is completely devoid of representational content. Being a piece of software, the game engine is a system of signs, and all signs are representational. Not only is the sign “the associative total of [the signified and the signifier],”³⁰ but “[l]anguage is a system of interdependent terms in which the value of each term results solely from the simultaneous presence of others.”³¹ Subsequently, when the game engine Unity, for example, uses the terms “parent” and “child” to describe the hierarchical organization of game objects, it simultaneously evokes a spatial relationship³² and an interpersonal one as well. This would suggest that traditional representational queerness might be located in the engine too—but to what end? If “parent” and “child” were changed to “daddy dom” and “little boy”, would it undermine the underlying encoded spatial hierarchy? Unlikely. Would it disturb or challenge the assumed Truths of game design, like that games can only be made by working oneself to death? Definitely not. Ultimately, then, queerness as a clearly delineated representation of identities, while undeniably valuable and a crucial part of queer games studies,³³ is not the prerogative of the engine. Another form is needed.

²⁹ Jason Gregory, *Game Engine Architecture* (Boca Raton: Tayler and Francis Group, LLC, 2009), 11.

³⁰ Roland Barthes, *Mythologies* (Les Lettres nouvelles, 1972), 113.

³¹ Ferdinand de Saussure, *Course in General Linguistics* (1916), 114.

³² A child object’s position in the game world (as well as scale and rotation) is expressed in relation to its parent’s. For example, if the parent was at (-10, 0, 0), and I wanted the child to be located at the origin, I would express its position as (10, 0, 0), not (0, 0, 0).

³³ Malkowski and Russworm, introduction to *Gaming Representation*, 3.

Systems of Rules

In Chapter 1 of *Video Games Have Always Been Queer*, Bo Ruberg analyzes *Pong* (Atari, 1972), a classic arcade game and seminal work within video game history, alongside Eve Sedgwick's equally foundational book *Between Men: English Literature and Male Homosocial Desire*. Near the end of the chapter, Ruberg advocates for "thinking about the formation of queer intimacy as an interactive system" structured, like games, around "unspoken rules."³⁴ For games, this is a relatively uncontroversial claim. In *Rules of Play: Game Design Fundamentals*, for example, Katie Salen and Eric Zimmerman argue that rules are "the logical underbelly beneath the experiential surface of any game."³⁵ They go on to distinguish rules from aesthetic qualities and representational identities, specifying that rules are "formal structures that constitute the game."³⁶ Implicit in this statement, of course, is that without rules, there is no game. It is no wonder that, when Salen and Zimmerman perform a quasi-statistical analysis of eight definitions of "game", 7 of the 8 definitions examined name "proceeding according to rules that limit players" as an essential quality.³⁷ It also figures prominently into their own definition of a game, which is: "[a] game is a system in which players engage in an artificial conflict, defined by rules, that results in a quantifiable outcome."³⁸ Yet it is an even more crucial part of a definition like Anna Anthropy's. In fact, "rule-based" encompasses nearly all of the definition she gives in *Rise of the*

³⁴ Ruberg, *Video Games Have Always Been Queer*, 54.

³⁵ Katie Salen and Eric Zimmerman, *The Rules of Play: Game Design Fundamentals* (Cambridge: MIT Press, 2004), 123.

³⁶ Salen and Zimmerman, 124.

³⁷ Salen and Zimmerman, 79.

³⁸ Salen and Zimmerman, 80.

Videogame Zinesters (2012), where she states that “[a] game is an experience created by rules.”³⁹

Rules, then, are crucial to even the broadest possible definition of games.

How, though, do rules relate to queerness? The answer comes from Michel Foucault and *The History of Sexuality* Volume I, where he argues that the roots of what we know today to be modern sexuality began within a system bound tightly by rules: the confessional. He explains that sex first entered the confession in the Middle Ages, when it was required to include the most intimate details of sex.⁴⁰ The rules began to change, however, in the 17th century, and new ones took their place: those of self-examination, discrete language, and demanding penance not only for the acts themselves but the thoughts that were associated with them.⁴¹ These rules would soon expand beyond the confines of the confessional booth, prompting “the multiplication of discourses concerning sex in the field of exercise of power itself: an institutional incitement to speak about it.”⁴² Such discourses began to coalesce in specific fields such as medicine, criminal justice, and education, which all added their own rules to the set.

And from them came homosexuality:

There is no question that the appearance in nineteenth-century psychiatry, jurisprudence, and literature of a whole series of discoveries on the species and subspecies of homosexuality, inversion, pederasty, and “psychic hermaphroditism” made possible a strong advance of social controls into this area of “perversity”; but it also made possible the formation of a “reverse” discourse: homosexuality began to speak on its own behalf, to demand that its legitimacy or “naturalness” be acknowledged, often in the same vocabulary, using the same categories by which it was medically disqualified.⁴³

³⁹ Anna Anthropy, *Rise of the Videogame Zinesters: How Freaks, Normals, Amateurs, Artists, Dreamers, Dropouts, Queers, Housewives, and People Like You Are Taking Back an Art Form* (New York: Seven Stories Press, 2012): 43.

⁴⁰ Michel Foucault, *The History of Sexuality Volume I* (1978), 19.

⁴¹ Foucault, 19.

⁴² Foucault, 18.

⁴³ Foucault, 101.

Sexuality here is controlled by the rules which determine what is perverse and what is not—what is normal. Yet it also establishes the groundwork for breaking those rules—not for completely rejecting them or the categories they create, but rather twisting and flipping them. Queering them.

Rules thus form the underpinnings of both games and queerness. From there, new understandings of and possibilities for the intersection of queerness and games arise, founded upon games' proclivity for representing, above all else, systems of interaction.⁴⁴ Through play, design, or interpretation, these systems can be made to represent at a mechanical level queer experiences. I call this mechanical queerness—formal systems and mechanics in video games which represent or speak to queer experiences. While *Pong*, for example, has little by way of representation, Ruberg deftly argues for understanding it as a queer game through its mechanics, such as seeing the ping-pong ball as an erotically charged mediator of touch between the two paddles, stuck at either side of the screen.⁴⁵

Before moving on to another example, it is worth briefly making explicit an implicit argument in the previous paragraph—that the mechanics of a game exist even when the game is not being played. It is worth lingering on this claim because, in addition to being rule-based, many definitions of games associate them with play and, by extension, a player. After all, Salen and Zimmerman call their book *Rules of Play*, explaining there are two possible relationships between play and games: games might be a type of play, or play might be an essential element of games.⁴⁶ Within queer games studies, Ruberg has posited play as a crucial site for finding or creating queerness in games, using the term playing queer to refer to “a mode of nearly infinite

⁴⁴ Ian Bogost, *Persuasive Games: The Expressive Power of Videogames* (Cambridge: MIT Press, 2007), 46.

⁴⁵ Ruberg, *Video Games Have Always Been Queer*, 47.

⁴⁶ Salen and Zimmerman, *Rules of Play*, 303.

[queer] possibility, brought to games largely through their players rather than systems structured by (mainstream) developers.”⁴⁷ Yet the critical nature of play in games is not uncontested, as can be seen in Mattie Brice’s blog entry “Death of the Player”. Responding to the claim that games are only ever complete with the addition of the player, Brice points to her own game called EAT, which requires the player to take on the kinds of harsh restrictions faced daily by impoverished people. EAT does not require play or players in order to communicate and be understood—“[i]t’s an experience that might just need a mental understanding to succeed” because simply looking at the rules and interpreting them is sufficient.⁴⁸ In emphasizing rules over play, my intention is not to dismiss the importance of play, particularly queer play. Rather, it is a question of explicitly including in my understanding of games those that are unplayable or go unplayed, those games which are already a step removed from what games are traditionally assumed to be because of their refusal to cater to players.

Turning to rules also allows me to consider broad patterns within games, rather than focusing only on individual instances of play. For example, a broad affinity between game rules and queerness can be found in how almost any game can simulate the process of queer identity formation. Sara Ahmed describes the normative—in this context, heterosexuality—as “an effect of repeating bodily actions over time, which produces ... a space for action, which puts some objects and not others in reach.”⁴⁹ When what is in reach, what is put in reach of children before they can conceive of wanting proximity to anything else, are straight objects, one becomes straight as a result of “having tended toward what is within reach ... bodies become straight by

⁴⁷ Ruberg, *Video Games Have Always Been Queer*, 17.

⁴⁸ Mattie Brice, “Death of the Player,” last modified October 29, 2013, <http://www.mattiebrice.com/death-of-the-player/>.

⁴⁹ Ahmed, “Orientations,” 561-2.

tending toward straight objects, such that they acquire their tendencies, as an effect of this tending toward.”⁵⁰

This normative process of repetition and tending can be seen in games as well— not only those with explicit queer representation, but practically all games. James Paul Gee, for example, has argued that “[t]he virtual character (in terms of the character’s skills and attributes) and the virtual world are built to go together such that the character’s goals are easier to reach in some ways than in others.”⁵¹ In a game, being in reach is not a question of spatial distance, but rather temporal. Playing the game ineffectively, if not outright incorrectly, costs a player the time it takes to fight against a system which is so adamant about prescribing a particular way to move through it. Gee’s example is trying to play *Thief: Deadly Shadows* (Eidos Interactive, 2004) as a brawler, a style of play made incredibly difficult because the main character “can find no guns or weapons much beyond a small dagger, and the spaces that would allow outright fights with multiple guards are pretty cramped, allowing guards easily to surround [the player].”⁵² In other words, the game acts as a corrective force, disciplining those who play it differently or even wrongly by making progress easier if they repeatedly perform certain bodily actions and not others.

The connections to be drawn to the lived queer experience are undeniable. Ahmed describes the process of becoming a lesbian in terms surprisingly similar to choosing to play a game incorrectly: “it takes time and work to inhabit a lesbian body, the act of tending toward other women has to be repeated, often in the face of hostility and discrimination, to gather such

⁵⁰ Ahmed, 557.

⁵¹ James Paul Gee, “Video Games and Embodiment,” *Games and Culture* 3, no. 3-4 (2008): doi: 10.1177/1555412008317309, 258.

⁵² Gee, 259.

tendencies into a sustainable form.”⁵³ As it is easier to be straight and cisgender, it is easier to play a game correctly, to succumb to normalizing forces in whatever space, digital or real, that they appear.

The next question, then, is to what extent this affinity between rules and queerness can be seen in the game engine. Although game rules and mechanics, like dialogue and assets as discussed in the conversation about queer characters, are game-specific, and thus not within the engine’s domain, it is nonetheless true that the game engine can influence a game’s rule set in obscure and unpredictable ways. Consider, for example, *Trespasser* (DreamWorks Interactive, 1998), a video game based on *The Lost World: Jurassic Park* (Universal Pictures, 1997). The engine underneath *Trespasser*, also called Trespasser, was intended to be a significant step forward in the, at the time, still nascent field of 3D game engine design, particularly due to its new real-time physics simulations.⁵⁴ However, fundamental flaws in that lauded physics system meant that the game built on top of it was glitchy and frustrating. For example, the engine used a physics model which allowed objects “to intersect with each other (mathematically) [before pushing] each other apart until they are no longer intersecting.”⁵⁵ While suitable for resolving collisions between two objects, as soon as there are more, a problem called interpenetration arises, wherein one object pushes another into a third object, which then pushes the second object back into the first, which then pushes the second object back into the third, et cetera. The issue is that because the objects are allowed to intersect before the resolution is calculated, the second object can end up stuck inside objects one and three, potentially making the game

⁵³ Ahmed, “Orientations,” 564.

⁵⁴ Richard Wyckoff, “Postmortem: DreamWorks Interactive’s *Trespasser*,” *Gamasutra*, last modified May 14, 1999, https://www.gamasutra.com/view/feature/131746/postmortem_dreamworks_.php, 1.

⁵⁵ Wyckoff, 3.

unplayable.⁵⁶ Another problem was the inefficiency of the physics code, which could not quickly detect intersections particularly given the hardware limitations of the time and thus could run quite slowly when multiple objects were on screen.⁵⁷ Because of the problems mentioned here and many more, *Trespasser* not only contained flawed mechanics, such as dropping your gun whenever you so much as brushed against another object,⁵⁸ but was also a commercial failure.⁵⁹ A failure of a game and a failure of an engine.

What parallels, if any, exist between the *Trespasser* engine and the lives of people in the LGBT community? What queer experiences do its systems and rules represent? One could, for example, read the interpenetration issue described above in a manner similar to Ruberg's analysis of *Pong*, with an additional layer of sexual metaphor thanks to the fact that the objects are not merely touching each other but entering each other. That the engine allows its subjects to occupy the same space in its mathematical calculations produces a sense of intimacy between game objects which largely no longer exists, as developers have tried their best to scrub interpenetration from their games.

This is an example of mechanical queerness within a game engine. Though a game's specific rules may be outside of the game engine's purview, the rules it does provide as the general basis for game development can be analyzed in much the same way as mechanical queerness in games themselves. As you will see in Chapter 3, this is the form of queerness that I have chosen to focus on in my own initial attempts to build the queer game engine. The reasons

⁵⁶ Wyckoff, 3.

⁵⁷ Shawn Sackenheim, "Trespasser-Review," *AllGame*, accessed December 19, 2019, <https://web.archive.org/web/20141115043401/http://www.allgame.com/game.php?id=14402&tab=review>.

⁵⁸ Kim Randell, "Jurassic Bark – this one's a dog," *Computer And Video Games*, last modified August 15, 2001, <https://web.archive.org/web/20070126112724/http://www.computerandvideogames.com/article.php?id=8366>.

⁵⁹ Wyckoff, "Postmortem," 2.

for choosing this form are described there, along with more thorough analyses. But, for now, there is one more understanding of queerness left to discuss in relation to both games and the game engine...

Queerness Beyond Identity

Sara Ahmed offers two understandings of the word queer. One is based in a particular sexual orientation (and, I would argue, gender identity) which is at odds with the hegemonic norms of a heterosexual and cisgender society. The other describes a general oddness and at-odds-ness not tied to a specific identity category, instead encompassing any oblique or adverse orientation toward a normative structure. It is within this understanding of queerness that queer theorist Riki Wilchins, for example, concludes that Woman is the queer gender because it is the Other to Man's Normal.⁶⁰ This is queerness not only as Otherness, itself another identity category, but as broadly non- or anti-normative.

There is a danger in such broadness, however, one which Ahmed acknowledges when she warns that her oblique and nonnormative definition of queerness "[risks] losing the specificity of queer as a commitment to a life of sexual deviation."⁶¹ Similarly, in their theory of playing queer, Ruberg explains: "I ... recognize that the vision of queer play that I am offering runs the risk of becoming diffuse and all-encompassing, as if all types of engagements with video games that are non-normative, in one way or another, now have grounds to call themselves queer."⁶² Ruberg's solution is to insist that their general notion of queer play "belongs, first and foremost, to those who live the joys and the pains of their queer lives each day in the world beyond gaming as well

⁶⁰ Riki Wilchins, *Queer Theory, Gender Theory: An Instant Primer* (New York: Alyson Books, 2004), 46.

⁶¹ Ahmed, "Orientations," 565.

⁶² Ruberg, *Video Games Have Always Been Queer*, 18-19.

as within them.”⁶³ While this is a serviceable distinction for play, understanding the hypothetical queer game engine exclusively in terms of the queer lives it touches would be severely limiting. For Ahmed, the advantage of such a broad understanding of queerness is that it “[sustains] the significance of deviation in what makes queer lives queer,”⁶⁴ and there is a wealth of unexplored opportunities for deviation and oddness to be found within the game engine in addition to how it interfaces with users.

To ensure, then, that such deviations are themselves queer requires a different distinguishing framework, like Halberstam’s understanding of queer versus unqueer failure. For Halberstam, this difference comes down to the question of whether an instance of failure is productive in the sense that it offers up alternative possibilities for life. He explains this distinction through an analysis of *Trainspotting* (1996), a novel about the hardships of the Scottish working class during Margaret Thatcher’s tenure as Great Britain’s Prime Minister. As Halberstam notes, the book is drenched in failure, particularly as a “critique of the liberal rhetoric of choice and ... rejection of hetero-domesticity.”⁶⁵ This failure is ultimately severely limited, however, because it “cannot imagine the downfall of the white male as part of the emergence of a new order.” Rather than proposing alternative ways of existing that do not hinge on sexist and racist hierarchies, *Trainspotting* “collapses into the angry and seething language of the male punk from whom a legacy of patriarchal and racial privilege has been withheld.”⁶⁶ Simply put, unqueer failure reaffirms the status quo; queer failure not only challenges it but suggests new ways of life that could replace it.

⁶³ Ruberg, *Video Games Have Always Been Queer*, 19.

⁶⁴ Ahmed, “Orientations,” 565.

⁶⁵ Jack Halberstam, *The Queer Art of Failure* (Durham: Duke University Press, 2011), 91.

⁶⁶ Halberstam, 92.

This is the understanding of queerness-as-antinormative which I would like to put forward here: not simply an orientation toward something, but a vector. A vector consists of a direction (orientation) and a magnitude, and it is this magnitude which carries the initial moment of anti-normativity forward. It is what turns a frictional point of contact into a point of penetration.

Penetration threatens the structural integrity of the entered object. To occupy an anti-normative subject position is to destabilize the norms one interrupts. Riki Wilchins outlines this process in relation to universal Truths. Such Truths can be understood as that which is allegedly transcendent and can be “declared to be universally true—for all people, in all societies, and at all times.”⁶⁷ They are created from repetition—by examining that which occurs over and over again and is the same in every iteration and declaring it to hold true, in every case.⁶⁸ Thus, in addition to large Truths like “the One God of the Catholic Church,”⁶⁹ there are smaller ones as well. For example, the universal binary of sex could be considered a Truth, in that it is constructed through the repeated categorization of bodies, human or otherwise, into one of two slots. That which does not fit is discounted “as a noise in the system, a problem to be resolved.”⁷⁰ Gay kids suffer through conversion therapy. Doctors coercively assign intersex infants to one sex or another. Subsequently, when a problem refuses to be resolved, the system of Truth within which it is contained is threatened. Queer desire and transgender bodies jeopardize the paradigms which insist on their non-existence.

⁶⁷ Wilchins, *Queer Theory, Gender Theory*, 47-8.

⁶⁸ Wilchins, 49.

⁶⁹ Wilchins, 47.

⁷⁰ Wilchins, 49.

Within this understanding of queerness as a vector of anti-normativity, queer indie game development takes on a new dimension of queerness, based not on the identities of its creators but on the work done by their games. Consider, for example, the structure of *dys4ia*. There is no real narrative choice in the game, yet this is essential to the game, even if such stringent limitations on player agency disturb “the convention that games must have choices that let a player alter the state of the system.”⁷¹ As Naomi Clark states, “[i]f Anthropy had followed the conventional rubric of games and allowed player actions to determine the course of the game, the autobiographical fidelity of *dys4ia* would have become mutable and subject to player whim.”⁷² Although *dys4ia* is a representationally queer game, through this analytical lens, its queerness lies also in how it destabilizes the assumptions about what a game is.

Anthropy has troubled several other game norms, particularly those related to industry design practices. For example, she wrote her game design book in part as an intervention into these practices. *Rise of the Videogame Zinesters* provides an alternative to game creation which is accessible, in that it does not require a technical background nor demand its creators labor endlessly and arduously in what the industry has termed crunch time.⁷³ This is an orientation toward games which is at odds with mainstream and independent game development alike,⁷⁴ which is adverse and wonky. It is destabilizing in how it undermines “the industry’s claim to

⁷¹ Naomi Clark, “What Is Queerness in Games, Anyway?” in *Queer Games Studies*, ed. Bo Ruberg and Adrienne Shaw (University of Minnesota Press, 2017), 6.

⁷² Clark, 6.

⁷³ Anthropy, *Rise of the Videogame Zinesters*, 18.

⁷⁴ Although the mainstream-indie binary is illustrative in some cases like discussing production scale, it also sometimes implicitly argues that indie games are somehow quarantined from the toxic nature of game development. *Night in the Woods* (Infinite Fall, 2017) offers an unfortunate example of why this is not the case. One of the game’s creators, Scott Benson, has talked about crunch time near the end of the game’s development, and the severely adverse health effects it had on him (<https://www.youtube.com/watch?v=Xzhe45Q8780>). The other creator, Alec Holowka, has been accused of sexual abuse by a former partner (<https://twitter.com/UnburntWitch/status/1166212005629325313>).

being the only route to game creation,”⁷⁵ yet constructive in how it offers a way out. By promoting tools and design processes that open game creation up to everyone, and not only those with technical expertise or a place in the industry, Anthropy sketches a kind of line of flight—a potential means of escape. Queerness here is thus the act of making an oblique intervention into universalizing and oppressive logics, not in any one game in particular, but rather in the industry as a whole, and game design as a generalized, recurring process.

This is a type of intervention which the game engine is particularly well-equipped to carry out. Whether they are made in AAA studios or indie ones, whether they are queer in their representational content, formal properties, or not at all, games run the risk of being developed in fundamentally unqueer spaces characterized by crunch time, unstable employment, and misogyny. Through Ruberg’s analysis of the precarity of queer indie game making, it has become painfully obvious that these risks are omnipresent, yet I want to linger on one final piece of evidence: the MidBoss scandal.

At first blush, the independent games studio MidBoss seems like a promising step toward queering the game design process. Its founder described its goal as “making the gaming space better for queer people,”⁷⁶ and, at least nowadays, its mission statement lists queerness explicitly:

MidBoss seeks to make the world better, safer and more inclusive to marginalized people, especially those in the gender and sexuality spectrum.

- **Creating fun and cool experiences** through games and other forms of media.
- **Create more representation** through creating worlds that include rich, meaningful characters of a wide variety of race, gender, sexuality and identity.
- **Becoming a platform for queer content around the world** through seeking out and shining a bright spotlight on other content creators who are making a positive mark in the world.⁷⁷

⁷⁵ Anthropy, *Rise of the Videogame Zinesters*, 18.

⁷⁶ Matt Conn, interview by the staff of Queerly Represent Me, *Queerly Represent Me*, accessed October 15, 2019, <https://queerlyrepresent.me/projects/interviews/matt-conn>.

⁷⁷ “MidBoss Mission Statement,” accessed October 9, 2019, <https://midboss.com/about.html>.

This queerness is not only a question of representation but building a utopian queer future through game design.

However, this promise did not shield employees from the forms of abuse more commonly associated with the mainstream game industry. Katherine Cross paints a damning picture of MidBoss, describing it circa 2018 as

a company where dual passions for game design and queer representation within those games were exploited to ensure a mostly young, queer and trans group of artists and developers could put long hours for little pay into a game that was capriciously managed.⁷⁸

Most of the problems raised by employees were linked to the company's then-CEO and the aforementioned founder, Matt Conn. Conn both severely underpaid workers (in some cases less than minimum wage for full-time game design work) and did not provide his employees with routes to address this exploitation, such as through an HR department or contracts which would guarantee salary. This unprofessionalism was even used by Conn as a rebuttal against sexual harassment claims, dismissing them "in multiple comments as him being 'weird' or 'sloppy about the way [he] communicated'."⁷⁹ Conn is now gone from MidBoss, as well as the woman who was initially set to replace him, after allegations were raised against her as well.⁸⁰ It is clear, though, that neither having a queer staff nor professing queer values are sufficient to guarantee that a game studio be queer in another sense, in that sense of destabilizing such game industry

⁷⁸ Katherine Cross, "Analysis: A closer look at MidBoss' chaotic, hostile work environment," *Gamasutra*, last modified March 28, 2018, https://www.gamasutra.com/view/news/315910/Analysis_A_closer_look_at_MidBoss_chaotic_hostile_work_environment.php.

⁷⁹ Cross, "Analysis: A closer look at MidBoss' chaotic, hostile work environment."

⁸⁰ @jessfromonline, "midboss has named toni rocca to take matt conn's place...", Twitter, March 26, 2018, <https://twitter.com/jessfromonline/status/978416750806040576>.

Truths as “games can only be made in toxic environments.” MidBoss in fact did little more than perpetuate that particular Truth.

I mention MidBoss not to lament the impossibility of queer game development, but rather to lend credence to the idea that the game engine—the queer game engine—is uniquely well situated to intervene in game development at a level which individuals or small studios cannot. After all, all the queerness which might be found in the (queer) game engine or its algorithms would not serve for anything if it could also be used as a tool to further someone’s exploitation, as was the case when MidBoss used Unity to make their game *2064: Read Only Memories* (2015) and underpaid and overworked their employees in the process. The queer game engine would thus have to have mechanisms by which to ensure that this is not the case. One could imagine it functioning like *Hurt Me Plenty* (Robert Yang, 2014), where the game acts as the mediator of consent and enforcer of boundaries between a digital body and a non-digital one. In the case of *Hurt Me Plenty*, if the player violates the boundaries of the digital character, it refuses to open for a certain amount of time depending on the severity of the violation. Maybe the queer game engine would function similarly, trying to facilitate ethical relations between users working on the same project.

Ahmed says that “[t]o make things queer is certainly to disturb the order of things.”⁸¹ What does the game engine disturb when it becomes queer? It might point outward to its place within the fundamentally unqueer process of game design as it currently exists. It might do so at twisted and disruptive angles, and it might then draw out the lines made by those angles, extend them through and beyond their origins, and see what alternatives might be waiting at the end. It

⁸¹ Ahmed, “Orientations,” 565.

might, in its most utopic form, be capable of disturbing and queering the entirety of game development.

Next Steps

Having discussed the relationship between queerness and games at length, as well as having begun to consider what these relationships imply for the queer game engine, it is worth pausing, briefly, to look at the road ahead. With this foundational understanding of queer theory, games studies, and queer games studies, Chapter 2 will dive into the game engine as it currently exists: its history, its definition, its contours and limits and how to study them. It will argue for understanding the term ‘game engine’ as expansively as possible so as to consider if the queer game engine already exists in the form of the platform Twine, a game development software tool deeply associated with the queer games movement. After arguing that a game design platform developed from the get-go with queerness in mind is still necessary, I will turn in Chapter 3 to this platform. As I have done throughout this chapter, I will continue to sketch out possibilities for the queer game engine’s design before ending with a close reading of two existing game engine components and their potential within the queer game engine. Finally, the conclusion will discuss the future of this project, as best I can currently envision it. The pages left here are not enough to consider the queer game engine in all of its facets and permutations, but they will offer a place to begin for continuing to flesh out this project in the years to come.

Chapter 2: Dreams, Doom, and Democratization

The queer game engine has thus far been defined by its relationship to some prominent ideas in queer games studies, organized along a handful of underdeveloped connections and brief points of synergy where the concepts of queer games studies might also be relevant to the queer game engine. Yet this approach to the queer game engine has largely ignored any technical considerations—considerations which need to be taken into account in order to account for differences in how queerness operates at the level of the game versus the level of the engine. It is, however, far easier to propose a technical analysis of the queer game engine than it is to carry it out. After all, how does one study imaginary software? An unreal development platform?

This exact incongruence between existing methodologies of software studies and the speculative is what motivated Thomas Apperley and Jussi Parikka's insightful critique of platform studies:

In terms of existing platform studies methodology ... the investigation of a platform is relatively straightforward. A scholar in the present examines a platform that exists. The evidence of its existence must be unambiguous. To achieve this recognition, the platform requires a degree of *stability and consistency* as a technical object.¹

Were it not for the queer game engine's nonexistence (the same nonexistence which has motivated this entire project), platform studies would be a natural framework to pick for approaching the game engine. As is, however, some adjustments are required, not only to accommodate the queer game engine's unreality but also to better study its relationship with game engines that do exist. These connections are crucial for moving toward the queer game engine because they offer the technological context for its interventions, yet other limitations of

¹ Thomas Apperley and Jussi Parikka, "Platform Studies' Epistemic Threshold," *Games and Culture* 13, no. 4 (December 2015): 354, doi:10.1177/1555412015616509.

platform studies—such as its insistence on the platform as a discrete entity—make these connections difficult to draw. At least, without some other frameworks to supplement it.

Queering Platform Studies

But before amending platform studies, it is important to understand what it is. The framework was first laid out by Nick Montfort and Ian Bogost in their 2009 book, *Racing the Beam: The Atari Video Computer System*. The first of several entries in MIT Press' *Platform Studies* series, *Racing the Beam* offers an intensive technical investigation of the Atari VCS. The book also outlines a brand-new form of technical and cultural inquiry into the “low level of digital media,”² referring, of course, to the platform. For Montfort and Bogost, the platform is “a computing system of any sort upon which further computing development can be done.”³ Put in simpler terms, a platform is programmable, and although most of the books within the *Platform Studies* series discuss hardware,⁴ the term encapsulates software—like the game engine—as well.

The limitations of platform studies have already been alluded to, but it is worth noting what the approach does well, and where it is useful to my own project. Specifically, Montfort and Bogost suggest that part of the work of platform studies is revealing that which “the programmer takes for granted when developing,”⁵ and this is work well worth doing in the context of the game engine. As a class of software, the engine remains relatively untouched by

² Nick Montfort and Ian Bogost, *Racing the Beam* (Cambridge, MA: MIT Press, 2009), 2.

³ Ian Bogost and Nick Montfort, “Platform Studies: Frequently Questioned Answers,” *Proceedings of Digital Arts and Culture 2009*, University of California, Irvine.

⁴ Of the books currently published within the series, eight out of the ten are concerned with pieces of hardware like the Atari VCS, the Wii, and the Nintendo Entertainment System. The two exceptions are a book on Adobe Flash and a French computer network called Minitel.

⁵ Montfort and Bogost, *Racing the Beam*, 2.

technical and cultural analysis alike. Discussions around game engines are largely limited to which one to choose for a particular game development project—and even this choice is often obscured by other factors, be they technical, economic, or the sheer pervasiveness of a particular engine. A study of independent game developers in Australia, for example, revealed that they “often use [the game engine] Unity because they already know how to use Unity, and developers already know how to use Unity because Unity has successfully ensconced itself within local communities as the default option.”⁶ Analyzing game engines with platform studies thus converts them from a ubiquitous, necessary, and invisible part of the game design process into explicit technical objects which can be studied and critiqued.

Where platform studies falters, however, is first in requiring that these objects have a stable, unambiguous existence, and second in suggesting that they are wholly separate entities from the systems that surround them. This second perspective is clearly displayed in Montfort and Bogost’s model of the five levels of digital media (fig. 1), which was originally released in *Racing the Beam* without the enclosing box labeled “Culture and Context.” In justifying their new academic inquiry, Montfort and Bogost privileged the platform as the base level of all digital media and outright refused to include cultural considerations alongside the technical until later.

⁶ Benjamin Nicoll and Brendan Keogh, *The Unity Game Engine and the Circuits of Cultural Software* (Cham, Switzerland: Springer Nature Switzerland, 2019), 89.

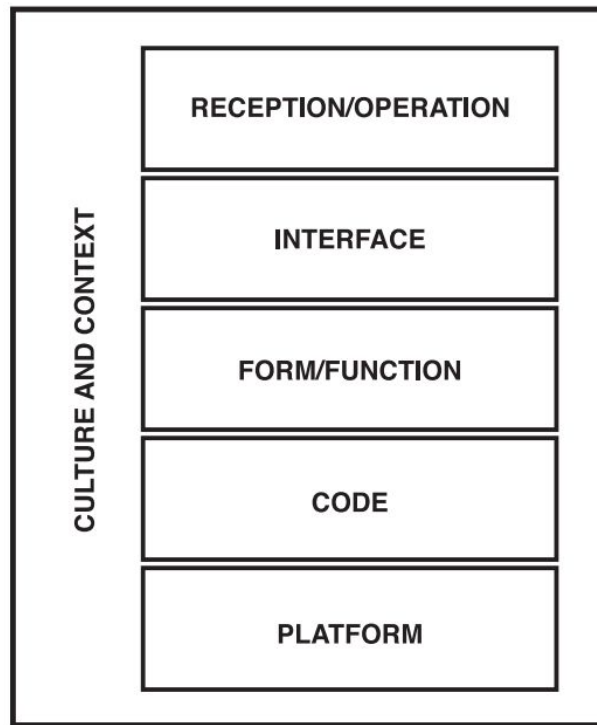


Figure 1. The five levels of digital media, from Montfort and Bogost's *Racing the Beam*.
 Source: (Montfort and Bogost, *Racing the Beam*, 146).

Furthermore, as Aubrey Anable points out in her feminist critique of platform studies, “[r]ather than being discrete objects, platforms, and the ways they connect us technologically and socially to others are porous, penetrating, and penetrable.”⁷ Considering queerness alongside platform studies could allow us to point to and revel in exactly this kind of slippage—after all, Sara Ahmed suggests that the word queer “might allow us to “twist” between sexual and social registers, without flattening them or reducing them to a single line.”⁸ Maybe it can do the same for the registers of digital media, permitting an intertangling between the engine and how it is

⁷ Aubrey Anable, “Platform Studies,” *Feminist Media Histories* 4, no. 2 (2018): <https://fmh.ucpress.edu/content/4/2/135>, 137.

⁸ Sara Ahmed, “Orientations: Toward a Queer Phenomenology,” *GLQ: A Journal of Lesbian and Gay Studies* 12, no. 4 (Duke University Press: 2006), 565.

received, or how it interfaces with the body of the user—a subject currently overlooked by platform studies, as both Apperley and Parikka,⁹ and Anable¹⁰ note.

A queering of platform studies thus helps to overcome some of the limitations posed by platform studies, such as the notion of hierarchical and rigidly separated layers and the absence of the body. What, though, is to be made of the problem of the queer game engine's nonexistence? Simply augmenting platform studies with queer theory seems insufficient for overcoming such a fundamental premise within the methodology—that of a stable existence.

Luckily, Apperley and Parikka offer a solution here. As Anable suggests feminist theory as the antidote to her critiques of platform studies, Apperley and Parikka suggest media archaeology as a tool through which “platform studies can develop a more sustained, long-term, historical engagement with media cultures ... and more nonlinear connections between past and future platforms and the conditions of the present.”¹¹ Media archaeology itself is defined rather vaguely. In the introduction to *Media Archaeology: Approaches, Applications, and Implications*, Parikka himself and Erkki Huhtamo describe media archaeology as an *approach* more than a discipline. Indeed, they go to great lengths to avoid making any effort “to nail down ‘correct’ principles or methodological guidelines or to mark fixed boundaries for a new discipline.”¹² They further configure media archaeology as “a practice of resistance [against] the increasing uniformity of mainstream media culture, but also against media archaeology itself, or rather its assimilation and hardening into the normalcy of contemporary media studies.”¹³ In these ways,

⁹ Apperley and Parikka, “Platform Studies’ Epistemic Threshold,” 355.

¹⁰ Anable, “Platform Studies,” 136.

¹¹ Apperley and Parikka, “Platform Studies’ Epistemic Threshold,” 359-60.

¹² Erkki Huhtamo and Jussi Parikka, “Introduction: An Archaeology of Media Archaeology,” in *Media Archaeology: Approaches, Applications, and Implications*, ed. Erkki Huhtamo and Jussi Parikka (Oakland, CA: University of California Press, 2011), 2.

¹³ Huhtamo and Parikka, 10.

media archaeology bears a striking similarity to queer theory, itself oriented against assimilation and resistant to concrete definition and the harsh boundaries of a discipline.

The closest that Huhtamo and Parikka get to a definition of media archaeology is that it “rummages textual, visual, and auditory archives as well as collections of artifacts, emphasizing both the discursive and material manifestations of culture.”¹⁴ From this initial orientation, two broad tendencies have arisen within the field. One trends so strongly toward materialism as to veer into technological determinism or move into hardware studies, as is the implication when Michael Goddard claims the likes of Friedrich Kittler and Marshall McLuhan as the field’s ideological predecessors.¹⁵ However, a materialistic perspective has little to offer in studying the queer game engine, as it is not a piece of hardware, and its claims of being apolitical¹⁶ are incompatible with queerness.

The other trend is, fortunately, more compatible with the queer game engine. This other articulation of media archaeology is best seen in the works of Siegfried Zielinski, who coined the term *anarchaeology*. Zielinski advocates for “a history that entails envisioning, listening, and the art of combining using technical devices, which privileges a sense of their multifarious possibilities over their realities in the form of products.”¹⁷ This focus on plurality and orientation against the singular narrative dictated by commercial success is undeniably queer. Indeed, Goddard summarizes media anarchaeology as “a non-linear history, one that is as, if not more, interested in the ‘losers’, or inventors and inventions that remained potential and imaginary

¹⁴ Huhtamo and Parikka, 3.

¹⁵ Michael Goddard, “Opening up the Black Boxes: Media Archaeology, ‘Anarchaeology’ and Media Materiality,” *New Media & Society* 17, no. 11 (2015): doi: 10.1177/1461444814532193, 1767.

¹⁶ Goddard, “Opening up the Black Boxes,” 1774.

¹⁷ Siegfried Zielinski, *Deep Time of the Media* trans. Gloria Custance (Cambridge, MA: MIT Press, 2006), 27.

rather than actual and implemented.”¹⁸ Given the association between queerness and failure elucidated by Jack Halberstam,¹⁹ an approach that lingers on the history of failure and embraces that failure might be said to be queer itself.

Above all else, this is what makes media archaeology such a useful tool. It permits the construction of “alternate histories of suppressed, neglected, and forgotten media that do not point teleologically to the present media-cultural condition as their ‘perfection’.”²⁰ Would it be such a leap to call this media queer? After all, as Eric Kluitenberg points out in his survey of this exact type of unrealized and unreal media, “[i]maginary media mediate impossible desires.”²¹ Is there any desire more impossible than that which requires turning away from what one is compelled to desire “as a condition of familial as well as social love?”²² Which resists the obsessive correcting and straightening of heteronormativity?

The final reason to turn to media archaeology is perhaps the most obvious one—history is important. Such is the essential thesis of all of media archaeology, with “[d]iscontent with ‘canonized’ narratives of new media culture and history [being its] clearest common driving force.”²³ Understanding the history of the game engine, of the people who made it and why they gave it the name that they did, is central to my argument about what a queer game engine could be and that it is possible for it to exist—if only in my imagination.

¹⁸ Goddard, “Opening up the Black Boxes,” 7-8.

¹⁹ Jack Halberstam, *The Queer Art of Failure* (Durham: Duke University Press, 2011), 89.

²⁰ Huhtamo and Parikka, introduction to *Media Archaeology: Approaches, Applications, and Implications*, 3.

²¹ Eric Kluitenberg, “On the Archaeology of Imaginary Media,” in *Media Archaeology: Approaches, Applications, and Implications* ed. Erkki Huhtamo and Jussi Parikka (University of California Press, 2011), 48.

²² Ahmed, “Orientations,” 557.

²³ Huhtamo and Parikka, introduction to *Media Archaeology: Approaches, Applications, and Implications*, 3.

From Doom to Unity: A Chronology

There is an official history to the game engine, one which begins with two much revered figures in game history: John Carmack and John Romero, known together as the Two Johns or the “Masters of Doom,” per the title of David Kushner’s book extolling their contributions to video games as an industry and creative medium.²⁴ And, indeed, the significance of these contributions is undeniable, ranging from iconic video games like *Doom* (idSoftware, 1993) and *Quake* (idSoftware, 1996) to the piece of software we know today as the game engine. Such is the chronology canonized by the likes of Jason Gregory, author of the definitive game engine design book *Game Engine Architecture*,²⁵ and Henry Lowood, the Curator for History of Science & Technology Collections at Stanford University who has written extensively about 90’s game history.²⁶

But this is not my history of the game engine. My interest is not in retracing yet again the established, normative path through video game history with its hegemonic prioritization of the technical, the programmed, the executable, the successful. While the Two Johns will certainly have their time and place in my history, if my ultimate goal is to argue for the possibility of a queer game engine, the two of them will not serve as a starting place. I would instead like to begin instead, in the spirit of media archaeology, with the impossible, the imaginary. The first section of *Media Archaeology: Approaches, Applications, and Implications* is, after all, called “Engines of/in the Imaginary”, and it is this nonlinear lineage of imaginary engines that I would

²⁴ David Kushner, *Masters of Doom: How Two Guys Created an Empire and Transformed Pop Culture* (London: Piatkus, 2013).

²⁵ Jason Gregory, *Game Engine Architecture* (Boca Raton: Tayler and Francis Group, LLC, 2009), 3.

²⁶ Henry Lowood, “Game Engines and Game History” (conference presentation, History of Games International Conference, 2014).

like to construct, beginning with the first computational engine of them all—the first two, actually.

I am referring, of course, to Charles Babbage and his imaginary calculating engines. In 1812, Babbage had a dream of the possibility of a machine that could calculate logarithmic tables. Over the succeeding decades, this possibility would begin to solidify into two designs, each imaginary and impossible to varying degrees. The first, the more feasible of the two, was called the Difference Engine, “a calculator which had a single, specific purpose: the mechanization of the production of mathematical tables.”²⁷ The other was the Analytical Engine, “a calculator [but also] a general-purpose machine which was automatic and programmable.”²⁸ In other words, it was a platform, and a clear predecessor to the modern computer, at least in its conception, if not in practice because it was never put into practice. Though construction began in 1859, it was never finished,²⁹ and it is doubtful that it could have been built given the technology of the time, slotting the Analytical Engine definitively into Zielinski’s category of the “untimely media/apparatus/machine.”³⁰

As mentioned, the Difference Engine came closer to realization. Construction went from 1823 to 1834 and produced enough of a machine for it to be displayed in the International Exhibition of 1862.³¹ However, the semi-finished product lacked a crucial part. It could perform

²⁷ Louise Purbrick, “The Dream Machine: Charles Babbage and His Imaginary Computers,” *Journal of Design History* 6, no.1 (1993), 10.

²⁸ Purbrick, 10.

²⁹ Purbrick, 10.

³⁰ Siegfried Zielinski, “Modelling Media for Ignatius Loyola: A Case Study on Athanasius Kircher’s World of Apparatus between the Imaginary and the Real,” in *The Book of Imaginary Media: Excavating the Dream of the Ultimate Communication Medium*, ed. Eric Kluitenberg (Rotterdam, Netherlands: nai010 PUBLISHERS: 2007), 30.

³¹ Purbrick, “The Dream Machine,” 10.

the specified computations, but it could not print its results.³² It was work without production, input and process without output.

By no stretch of the imagination am I claiming Babbage's calculating engines as a direct technological predecessor to the game engine, but then, the point is not to follow the linear paths of least resistance. The intention is to zero in less on the technology itself and more the words used to describe it. The term engine here gains its association with computation, which can still be seen in modern software like the browser and search engine. Yet it also takes on the implication of the elusive, the suggestion of something which does not yet and possibly can never exist. Something which, subsequently, remains somewhat volatile. The thread of the imaginary may be lost within the modern game engine, more concerned with presenting its users with possibility spaces than existing within one itself, but my ultimate goal is to find the impossible engine again.

And, of course, the game engine did once exist in the realm of dreams and imagination, in the years before it was licensed and came to dominate the video game industry as the easiest, and only, way to make games. Here is where the Two Johns reenter my history. John Carmack and John Romero first used the word engine in conjunction with the Commander Keen series (id Software, 1990-1991). Their idea was that certain functionality, like how to draw something on the screen or tell whether two objects are touching, was common to all games in the series. It did not matter whether Commander Keen was fighting Martians or potato men. Some things, like the *Super Mario Brothers*-style side scrolling the game was known for,³³ were universal, and that meant they could be abstracted out. This was the first game engine: a "single piece of software

³² Purbrick, 11.

³³ Lowood, "Game Engines and Game History," 184-185.

that produced common functionality for multiple games,” first within the Commander Keen series, and then Carmack and Romero’s later games, like *Wolfenstein 3D* (id Software, 1992) and *DOOM* (id Software, 1993).

In the case of *DOOM*, the game engine became more than a tool for developers, who no longer had to build each new game from the bottom-up, as was the practice before Carmack and Romero.³⁴ The engine also became an intimate point of connection between the player, game, and developer. Crucially, the engine enforces a strict separation between code that is general and reusable across different games, and that which is specific to each game. Subsequently, if players have access to what is specific and can change it, the game engine provides a stable mechanism to execute those changes. A player can tinker with assets and maps and, without having to touch underlying systems like the renderer, turn *DOOM* into *Hotline Miami* (Dennaton Games, 2012).

³⁵ The game engine, paired with id Software’s specific commitment to making modifications easy, thus enabled what Lev Manovich termed “a new cultural economy that transcended the usual relationship between producers and consumers [where] hacking and adding to the game became an essential part of the game.”³⁶ The engine turned the games built on top of it into platforms in and of themselves.

Another example of engine-game separation enabling fan practices is *Quake* movies, a predecessor to machinima made specifically with another of Romero and Carmack’s games, *Quake* (id Software, 1996). As Henry Lowood explains, “[d]emo files were a particular kind of asset file in both *DOOM* and *Quake*. A few players figured out how to change these files and

³⁴ Gregory, *Game Engine Architecture*, 3.

³⁵ <https://www.moddb.com/mods/hotline-uac> (Angel-Neko_X, 2019).

³⁶ Lev Manovich, *The Language of New Media* (Cambridge, MA: MIT Press, 2001), 245.

produce player-created movies that the game engine could then play back.”³⁷ One of the earliest such productions, called *Diary of a Camper* and made only two months after the game’s release, is generally regarded as the first machinima,³⁸ and it was first distributed online not as a screen-captured video but as a demo file to be played back by the Quake engine.³⁹

While Carmack and Romero developed a valuable tool for expediting the game design process and fostering fan practice, it took time for it to evolve into a valuable commercial endeavor. Lowood mentions that id Software only received one order for the original Keen engine, from a company which used it to make one game and then designed their own engine based on id’s code.⁴⁰ Indeed, in the 90s, derivatives of Carmack and Romero’s engines were common, like Valve’s GoldSrc engine, which was used to make the *Half-Life* series (Valve Corporation, 1998). Such games obviously could not make money for id Software, but the company eventually did learn how to monetize their engine. It did this by making older ones open source while simultaneously licensing the most up to date iteration for half a million dollars.⁴¹ Studios who wanted the best technology paid for it, and the game engine became an industry.

And with being an industry came competition, primarily from the engine which Epic Games used to build their hit first-person shooter *Unreal* (Epic Games, 1998). This is, of course, the Unreal Engine, which upon its initial release was heralded as beginning the 3D engine wars.⁴²

³⁷ Lowood, “Game Engines and Game History,” 187.

³⁸ Henry Lowood, “High-Performance Play: The Making of Machinima,” *Journal of Media Practice*, 7, no. 1 (2006): doi: 10.1386/jmpr.7.1.25/1, 32.

³⁹ United Rangers Films, “Diary of a Camper,” Archive, 1996, 1:21, <https://archive.org/details/DiaryOfACamper>.

⁴⁰ Lowood, “Game Engines and Game History,” 185-86.

⁴¹ J.C. Herz, “GAME THEORY; For Game Maker, There’s Gold in the Code,” *The New York Times*, December 2, 1999, <https://www.nytimes.com/1999/12/02/technology/game-theory-for-game-maker-there-s-gold-in-the-code.html>.

⁴² “3D Gaming Technology,” *Maximum PC*, November 2004, 59.

The Unreal Engine's advantages as a competitor were obvious even at the outset. Its editor was easier to use than that of the Quake engine, it had a dedicated third-party support staff, and it was designed from the outset to be "extended and improved throughout many generations of games."

⁴³ By contrast, rather than being successive iterations, each software in the id Tech series is actually a separate engine, each initially designed in order to build one specific game, with other production as a secondary concern.

The outcome of the 3D engine wars is conclusive. id Software stopped licensing its engines in 2010 with id Tech 5, which could only be used by studios that were a part of ZeniMax Media Inc.⁴⁴ In this way, id Software's engines nowadays have far more in common with, say, Frostbite, the in-house proprietary engine of Electronic Arts which can only be used by it and its subsidiaries such as BioWare and Ghost Games.

The Unreal Engine's position in the modern-day game development industry is quite different. It still licenses its latest iteration, Unreal Engine 4 (2014), to third-party developers, but it now does so with a much smaller price tag than the original Unreal Engine with its \$1 million license.⁴⁵ In fact, for many users, there is no price tag at all: in 2015, Unreal Engine 4 became free—in the sense that it no longer required a paid subscription to use.⁴⁶ Epic Games' CEO and president described the move in optimistic terms with regards to the bright, democratic future for game development, saying that "[t]he more barriers we take away for more people to use the

⁴³ "The 3D Engine Wars," *Maximum PC*, November 1998, 43.

⁴⁴ Kris Graft, "id Tech 5 Rage Engine No Longer Up For External Licensing," *Gamasutra*, last modified August 13, 2010.

⁴⁵ "The 3D Engine Wars," *Maximum PC*, 43.

⁴⁶ Paid licenses did not, of course, disappear entirely. Negotiating custom licenses beyond Unreal's default End User License Agreement is necessary, for example, to avoid paying royalties to Unreal for money made on the game, which is required for those who use Unreal for free.

engine, the brighter the future is for Unreal and for our community of developers.”⁴⁷

Furthermore, for Unreal, accessibility is not only a question of economic barriers—the entirety of its source code is available on GitHub once you have downloaded the engine.

Yet Unreal was not the first game engine to mobilize the rhetoric of democratization. To hear them tell it, that honor would likely go to Unity, a 2D and 3D game engine made by Unity Technologies and first released in 2005. Unity entered the game development scene in close proximity to id Tech 4 (2004) and Unreal 3 (2006), with the latter reviving claims about Unreal’s ability to seriously challenge or even displace id Tech’s monopoly.⁴⁸ Whereas id Tech and Unreal were developing for the latest generation of computer graphics, however, Unity had a different target in mind: games for the iPhone⁴⁹ and Nintendo’s Wii, which was released a year after Unity. With specific regards to the Wii, material promoting this new license, Unity Wii, made specific appeals to the shared goals and values of the two platforms: “The Wii console is all about new types of games and new ways to play. Unity embodies that philosophy with an Editor that’s built around creating game play.”⁵⁰ As one of the founders and former CEO of Unity Technologies, David Helgason, explains, from the outset, Unity was designed “as a tool to help developers build the games that they wanted to build,” such that its developers did not actually initially refer to it as an engine.⁵¹ Given that both mobile gaming and the Wii console have been

⁴⁷ John Gaudiosi, “Why Epic Games is giving away its game technology,” *Fortune*, last modified March 3, 2015, <https://fortune.com/2015/03/03/epic-games-unreal-tech-free/>.

⁴⁸ “3D Gaming Technology,” *Maximum PC*, 59.

⁴⁹ Samuel Axon, “Unity at 10: For better—or worse—game development has never been easier,” *ars technica*, last modified September 27, 2016,

<https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>.

⁵⁰ “UNITY: Features - Wii Publishing,” *Unity*, <https://web.archive.org/web/20081220051710/http://unity3d.com/unity/features/wii-publishing>.

⁵¹ “A Conversation with David Helgason,” *DICE Europe*, 2015, https://www.diceeurope.org/news/conversation_with_david_helgason.asp.

associated with the rise of casual gaming,^{52 53} itself a kind of democratization not of game development but of game *play*, Unity's early licenses lend credence to Helgason's frequent assertion that the democratization of game development was always the engine's goal, rather than a rhetorical position later adopted for marketing purposes.⁵⁴

Furthermore, in 2009, six years before Unreal, Unity became free to use through its personal license.⁵⁵ Yet the exact definition of free here is debatable. Although users do not have to pay anything to download the software, they must abide by stipulations like paying royalties to Unity Technologies "if the project surpasses an annual revenue threshold of USD\$100,000" and allowing Unity to collect user data.⁵⁶ Additionally, as Bo Ruberg has pointed out, just because the game development software is free does not mean that the entire process is free, with expenses often manifesting in other parts of the developer's life, particularly if they exist in a position of economic and social precarity.⁵⁷ These limitations are crucial to consider alongside claims made about Unity's democratic, accessible, or revolutionary nature. Unity has had real successes in both making game design more economically accessible and "levelling out ... work role hierarchies in studio environments—hierarchies that, historically, have delegated power to programmers and software engineers, as opposed to artists and designers."⁵⁸ At the time of its release, particularly within the milieu of id Software and Epic Games' competition, focused on

⁵² Omer Kaplan, "Mobile gaming is a \$68.5 billion global business, and investors are buying in," *Tech Crunch*, August 22, 2019, <https://techcrunch.com/2019/08/22/mobile-gaming-mints-money/>.

⁵³ John Vanderhoef, "Casual Threats: The Feminization of Casual Video Games," *Ada: A Journal of Gender, New Media, and Technology* no. 2 (2013): doi:10.7264/N3V40S4D, n.p.

⁵⁴ David Helgason, "Leading Unity into the Future," *Unity Blogs*, October 22, 2014, <https://blogs.unity3d.com/2014/10/22/leading-unity-into-the-future/>.

⁵⁵ David Helgason, "A free Unity?" *Unity Blogs*, October 29, 2009, <https://blogs.unity3d.com/2009/10/29/a-free-unity/>.

⁵⁶ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 37.

⁵⁷ Bo Ruberg, "The Precarious Labor of Queer Indie Game-making: Who Benefits from Making Video Games 'Better'?" *Television & New Media* 20, no. 8 (2019): 778-788.

⁵⁸ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 3.

larger studios, more expensive licenses, and top-of-the-line graphics, Unity was a step in the right direction.

However, Unreal eventually took that step as well with Unreal Engine 4, and both Unreal and Unity have

not democratized employment opportunities, nor [have they] addressed tendencies within studio environments to turn a blind eye towards issues of precariousness, toxicity, burnout, and exploitation; the same goes for the industry's long-standing ambivalence towards collective organization.⁵⁹

This is particularly dangerous within Unity's (and to a lesser extent Unreal's) growing monopoly on game development. For example, Unity's CEO John Riccitiello has claimed that Unity is used in the production of half of all games.⁶⁰ According to a survey conducted in Australia in 2018, roughly 75% of video game companies surveyed were using Unity.⁶¹ Finally, because of its prevalence in the industry—in a 2017 LinkedIn report, Unity Developer was ranked 7th out of 20 top emerging jobs in the United States—⁶² Unity has also begun to monopolize game design programs in higher education.⁶³

This is the current state of the game development industry. Unity is no longer the outsider it originally conceived itself to be, as, alongside Unreal Engine 4, it dominates the independent game development scene. Meanwhile AAA studios like id Software, Electronic Arts, and Bethesda Game Studios use in-house engines which are not available for licensed development outside of the studio.

⁵⁹ Nicoll and Keogh, 113.

⁶⁰ Romain Dillet, "Unity CEO says half of all games are built on Unity," *Tech Crunch*, September 5, 2018, <https://techcrunch.com/2018/09/05/unity-ceo-says-half-of-all-games-are-built-on-unity/>.

⁶¹ Game Developers' Association of Australia, "What Game Engines Are Companies Using?" *Facebook* (2018): <https://www.facebook.com/GameDevAssocAus/photos/a.1943622552593569/2062766267345863/?type=3&theater>.

⁶² "LinkedIn's 2017 U.S. Emerging Jobs Report," *LinkedIn*, December 7, 2017, <https://economicgraph.linkedin.com/research/LinkedIns-2017-US-Emerging-Jobs-Report>.

⁶³ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 93.

Yet my overview of the game engine's chronology—what might better be termed the canonical or industry-certified chronology—has at least two glaring omissions. The first are 2D game engines such as GameMaker (YoYo Games, 1999) and RPG Maker (ASCII, 1992). These platforms have been integral to 2D indie and queer game development. Toby Fox's enormously popular *Undertale* (2015) and *Deltarune* (2018), for example, were developed with GameMaker, as well as a number of Anna Anthropy's games like *Calamity Annie* (2008). Unproductive Fun Time made *Off* (2008) in RPG Maker 2003, and Mattie Brice used it to build *Mainichi* (2012). They may not call themselves game engines—YoYo Games calls GameMaker “game development technology” instead—⁶⁴ but they have been legitimated as such through their popularity. Nowadays, in guides which help beginning game developers choose the right software, GameMaker and RPG Maker are frequently listed as 2D game engines.⁶⁵

Equally absent from this chronology are those more unruly game development platforms which have not been absorbed into the game engine legacy. These platforms might be new and small, having not yet been discovered by those within the industry with the power to legitimize them as game engines. Or they may be older and well-known but find themselves harshly excluded from the category because of the threat seemingly posed by including them. Some might argue that they are absent from the history I have sketched here because they do not belong in it, yet I disagree. I believe that a queer intervention into the game engine is not a mere academic exercise in being theoretically and needlessly destabilizing, but rather an imperative in order to disrupt the monopoly which Unity and Unreal have on indie game development. Furthermore, I believe that moving these marginalized platforms from the category of

⁶⁴ “About YoYo Games,” *YoYo Games*, accessed April 11, 2020, <https://www.yoyogames.com/about>.

⁶⁵ Florian, “The Best Game Engines for Beginners: 14 Alternatives,” *Website Tool Testers*, November 29, 2018, <https://www.websitetooltester.com/en/blog/best-game-engine/>.

not-engines to game engines is the first step. As long as they remain relegated to the fringes of game development, their ability to make insightful and incisive critiques of industry-sanctioned engines, especially those which are not proprietary like Unreal and Unity, is undermined.

If there is queer potential to be found in the game engine, it is not within Unity's democratization *dispositif*⁶⁶ or the fact that *Life is Strange* (Donnod Entertainment, 2015) or *Mass Effect* (BioWare, 2007-2017) were made in Unreal. It lies in troubling and expanding the very category of the game engine, and so I must rewind the clock on my game engine history. I must return to 2009, the year when Unity became "free" to use under its personal license and another piece of software, free from the very beginning, was released. My chronology of the game engine ends with Twine.

Twine

Twine is an "an open-source tool for telling interactive, nonlinear stories"⁶⁷ that was made in 2009 by Chris Klimas. It was born out of Klimas' personal desire for a software that would allow him to more easily write hypertext stories—stories that unfold across discrete chunks of text (called passages or lexia) which are connected by links.⁶⁸ Klimas had begun writing hypertext with TiddlyWiki, "a web page you could download and then edit directly in your web browser." When he grew frustrated with the software, he wrote a markup language for it called Twee, which allowed him "to create TiddlyWiki files using a text editor." Finally, he designed a

⁶⁶ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 101.

⁶⁷ "Twine," accessed April 11, 2020, <https://twinery.org/>.

⁶⁸ meritt kopas, introduction to *Videogames for Humans: Twine Authors in Conversation*, ed. meritt kopas (New York: Instar Books, 2015), 9.

graphical user interface for authoring Twine files and then converting them into HTML files. This was Twine.⁶⁹

Although Twine was not initially designed with game development in mind, game designers, and specifically queer ones like Anna Anthropy, Porpentine, and merritt kopas, began to use it for that purpose. The sudden and rapid proliferation of Twine games, particularly after the release of Anna Anthropy's *Rise of the Videogame Zinesters*, gave rise to claims of a revolution.⁷⁰ The Twine Revolution asserted that the video game was "a medium for everyday creative expression, as opposed to a highly technical and institutionalized craft."⁷¹ Quite simply, the Twine Revolution argued, much like Anthropy herself, that anyone could make video games.

There are a number of reasons for Twine's popularity among this particular set of game makers, people holding marginalized identities working at the margins of indie game development, itself already offset from mainstream development. For example, Twine is technologically accessible for both developers and players. At a bare minimum, making something in Twine only requires the ability to write and create links, itself a trivial act of putting one or more words between double brackets, so while Unity claims to displace the programmer, Twine dispenses with them entirely. Furthermore, because of the ubiquity of link-clicking in an online world, playing a Twine game requires skills already familiar to most users, as opposed to, say, a fighting game which requires the perfect execution of a series of rapid button presses.

⁶⁹ Chris Klimas, "Twine: Past Present, Future," *Chris Klimas* (blog), June 2019, <https://chrisklimas.com/twine-past-present-future/>.

⁷⁰ Cara Ellison, "Hypersexed Hypertext: Porpentine and the Twine text game revolution," *PC Gamer*, last modified April 3, 2013, <https://www.pcgamer.com/hypersexed-hypertext-porpentine-and-the-twine-text-game-revolution/>.

⁷¹ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 161.

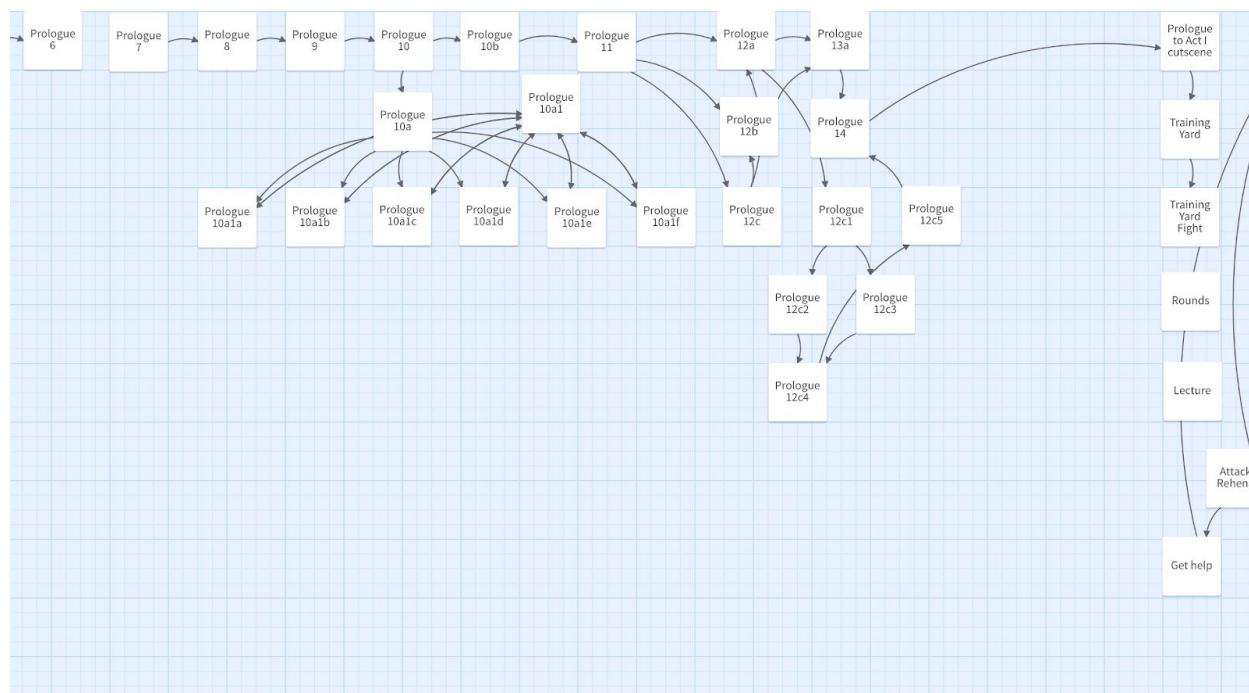


Figure 2. Twine 2 interface. The white blocks are passages, with the black arrows representing links between them.

Source: (Screenshot of a personal Twine project)

Porpentine lists several more attractive qualities of Twine, such as being free and having a spatial interface, rather than being text- or code-based (fig 3).⁷² Many other Twine authors have commented on the editor's visual nature as being a boon, with meritt kopas explaining that "[the] passages can be dragged and shifted around the screen, with links between passages represented by arrows, creating a kind of flowchart. Thus it's easy to track branching paths as you build by following the trail of arrows through the narrative."⁷³ She also suggests that there is something welcoming about this interface, that "[i]t didn't demand ... that [she] tell a linear story, one that was neat and made sense and contained some kind of resolution ... The

⁷² Porpentine, "Creation Under Capitalism," *The Gamer Trust*, last modified November 25, 2012, <https://nightmaremode.thegamertrust.com/2012/11/25/creation-under-capitalism/>.

⁷³ kopas, introduction to *Videogames for Humans*, 9.

connections could come later.”⁷⁴ Porpentine connects this layout to the reality of trying to make games while sick, saying that “[she] can work with Twine when [she is] too tired to deal with anything else. You don’t have to wrestle with anything between the emotion and the page, your fragile thoughts survive.”⁷⁵

Of course, while Twine has many features which made it attractive to the queer game designers who adopted it, it is crucial to avoid an overly deterministic approach to the platform, particularly within the context of its revolution. Twine became revolutionary not because of its technical affordances, but rather because of the people who used those affordances in unintended ways. It could not have democratized the game industry if a group of queer trans women had not built games with it, thus bringing it into contact with that industry.⁷⁶ Furthermore, these characteristics which made Twine so enticing as a game development tool are not static, stable things. Indeed, Twine has evolved since *Rise of the Videogame Zinesters* allegedly began its revolution in 2012, most significantly in 2014 when a new version, Twine 2, was released.

Twine 1 and 2 are not fundamentally different pieces of software. Still, it is worth lingering for a moment on their distinctions. After all, the shift from Twine 1 to 2 has received little attention in other analyses of Twine, even though it renders much of the literature written during or immediately after the revolution, while not obsolete, less relevant than it was at the time of writing.

Perhaps most notable of the differences between the two is that Twine 2 can be used in a web browser, which enables production on a wider range of devices like tablets. Working in

⁷⁴ kopas, 7.

⁷⁵ Porpentine, interview by Emily Short, *Emily Short's Interactive Storytelling*, November 23, 2012, <https://emshort.blog/2012/11/23/interview-with-porpentine-author-of-howling-dogs/>.

⁷⁶ Benjamin Nicoll, *Minor Platforms in Videogame History* (Amsterdam University Press, 2019) 185.

Twine 2 in a browser is also unique compared to other similar services because it does not require an account to use. Rather than storing each user's work on a remote server, Twine circumvents the problematics of cloud storage like environmental impact⁷⁷ by saving files to a user's web browser's local storage.⁷⁸ This means that, although Twine 2 is open to different kinds of accidental data loss, it is ultimately even more accessible than its predecessor. It can be used without any kind of installation on the part of the user.

Further changes between Twine 1 and 2 demonstrate that the platform's ease of use is not a singular, constant thing. Twine 2 implemented a "Debug" mode which made it easier and faster to test games. The Debug menu lists all the current variables in one's Twine piece as well as their values, so users can see when those values change—or don't. Imagine, for example, if you want to unlock a door if the player has a key. If the door does not open, it may be because the player did not pick up the key, or it may be because they picked it up, but the key variable was not properly set. Such problems become rapidly evident in Twine 2's Debug mode but could be harder to locate in Twine 1.

⁷⁷ AJ Dellinger, "The environmental impact of data storage is more than you think -- and it's only getting worse," *Mic*, last modified June 19, 2019, <https://www.mic.com/p/the-environmental-impact-of-data-storage-is-more-than-you-think-its-only-getting-worse-18017662>.

⁷⁸ "Where Your Stories Are Saved," *Twinery*, last modified October 9, 2017, http://twinery.org/wiki/twine2:where_your_stories_are_saved.

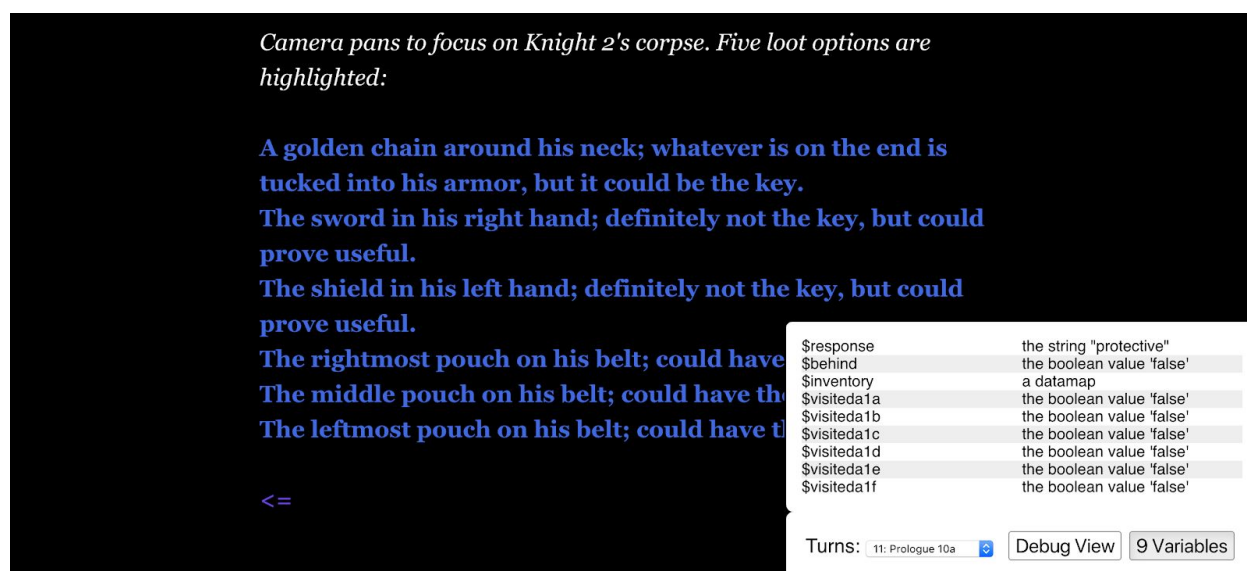


Figure 3. Twine 2's Debug mode. The game uses \$visiteda1a, \$visiteda1b, etc. to only show loot options which the player has not yet explored; this interface is helpful for checking, for example, that these values are actually changing when the player clicks the corresponding loot option.
Source: (Screenshot of a personal Twine project)

Twine 2's system for saving files is simpler as well. Twine 1 works with three different file types depending on the step in the process: Twine Story files (.tws) which store the data for displaying a story in the GUI, Tweek files (.tweek) which allow a user to send their Twine story from one computer to another but not run the game itself, and finally HTML files (.html) which represent the finished work. Twine Story files are generated when the user hits "Save", Tweek files when they hit "Export", and HTML files when they hit "Build Story". By contrast, Twine 2 only works with HTML files: one type which is stored automatically when a change to the story is made and only stores passage information, and one which is generated when the user hits "Publish to File", which contains information for rendering the game in a browser.

The language used to produce the final HTML file in Twine 1 and 2 is particularly interesting. As noted, Twine 1 uses a button called "Build Story", which is at once putting forth a claim about the types of works made in Twine and evoking an idiom found in both higher-order

programming languages like C++ and industry game engines like Unity, where building is a sort of preparatory step before running the finished product. In a programming language, the build stage catches errors of syntax or type but not problems that only become evident when the program is executed, like a loop whose exit conditions cannot be met and thus repeats endlessly, forever. In Unity, what building does varies depending on which system the game is supposed to run on—building for Mac, for example, produces an Application, while building for Windows produces a folder with, among other things, a .exe file. Still, the broad idea is the same: making something that can be run “on its own,” to the extent that it is not dependent on the software in which it was made or, say, local file paths for assets. If I just send the Application to someone, they have all they need to play my game.

In Twine 1, this same idea is present in calling this process “Build”. Yet in Twine 2, this was replaced with a button called “Publish to File,” which evokes authorship far more than it does a development environment. To someone with little programming experience, this terminology is likely easier to understand than the abstract and technical concept of building. It is thus clear that Twine’s much-lauded accessibility is not static but has rather continued to mutate and develop since its launch.

Twine and Games

The ultimate question is: is Twine a queer game engine? And in order to answer it, or to even be able to justify proposing it, it must first be established that Twine is a game engine, and in order to do that, the question must be asked: are works made in Twine games? Obviously, the artists discussed above consider their works to be games, given that they promote them as such. On

Porpentine's website, for example, she lists all of her Twine works on her "games" page, and merritt kopas edited a Twine anthology called *Videogames for Humans*, explicitly arguing that all the works found within the book are games. Personally, I am comfortable taking these designers at their word, and uninterested in enforcing the exclusionary boundaries which have historically denied queer people, women, and people of color from the game industry. Yet to pretend that this is a universally accepted position is to efface the tremendous harm done by those who do not hold it.

Twine games—if they are games—are easy to make and easy to play, which contradicts and threatens supposed fundamental tenets of game design and the gamer identity. The response to the tool is the (re)valorization of the idea that video games are hard to make.^{79 80} The response to the game is to treat it as illegitimate. Its illegitimacy is then used as grounds either to ignore it, to declare it irrelevant to gaming as a culture and an industry,⁸¹ or to treat it as a threat which warrants violence to neutralize it. The most organized of such responses came in the form of GamerGate, an online harassment campaign from 2014 which targeted, among others, Zoë Quinn and her Twine game *Depression Quest* (2013). As Benjamin Nicoll explains, "[GamerGate's] supporters took issue with the fact that a Twine videogame, written and developed by a woman, could receive such widespread critical acclaim."⁸² He goes on to suggest that "[the] movement was, perhaps, a response to the disintegration of the gamer identity and the

⁷⁹ Aleksander Adamkiewicz, "No, You Can't Make Video Games," *Gamasutra*, last modified January 8, 2012, https://www.gamasutra.com/blogs/AleksanderAdamkiewicz/20120703/173489/No_You_Cant_Make_Video_Games.php.

⁸⁰ Sophie Houlden, "Stop being such an asshole to people who don't start from scratch," *Sophie's Blog*, last modified June 15, 2012, <http://www.sophiehoulden.com/stop-being-such-an-asshole-to-people-who-dont-start-from-scratch/>.

⁸¹ Vanderhoef, "Casual Threats: The Feminization of Casual Video Games," n.p.

⁸² Nicoll, *Minor Platforms in Videogame History*, 168.

structures of power that have traditionally supported it,”⁸³ with this identity specifically being derived from an exclusive and masculine conception of the “hardcore”. Both game design and game play, so these particular gamers believe, have to be difficult.

The question of whether or not Twine games are real, legitimate games is one with enduring ramifications, particularly considering the off-line manifestations of Quinn’s harassment after she was doxed.⁸⁴ It is also a historical one, given Twine’s established and enduring relationship to hypertext, a medium which has itself been historically marginalized. Games researcher Espen Aarseth, for example, has repeatedly insisted on relegating hypertext fiction to the category of novel⁸⁵ or literature.⁸⁶ In his mind, its electronic or digital components are not worth studying, as he says that “[l]iterary experiments are either interesting or they are not. What medium they take place in should have little or nothing to do with it.” For him, that medium certainly is not and cannot be “video games” because hypertext fiction fundamentally lacks rules, instead creating a mere illusion of play.⁸⁷ This relates to the idea that hypertext offers a “low grade” of interactivity “[c]ompared to other electronic applications such as computer games, MOOs and MUDs, and especially ‘goggle-and-glove’ VR.”⁸⁸ Low grade here specifically refers to an ability to make superficial choices that move players along predetermined, pre-scripted paths but not construct alternate routes through the material, or author new material

⁸³ Nicoll, *Minor Platforms in Videogame History*, 168.

⁸⁴ Zoe Quinn, “All Gamergate Has done is ruin people's lives,” interview with Keith Stuart, *Guardian*, December 3, 2014, <https://www.theguardian.com/technology/2014/dec/03/zoe-quinn-gamergate-interview>.

⁸⁵ Espen Aarseth, Technoculture discussion list, September 23, 1993.

⁸⁶ Espen Aarseth, “Genre Trouble: Narrativism and the Art of Simulation,” in *First Person: New Media as Story, Performance, and Game* (2004), 53.

⁸⁷ Aarseth, 53.

⁸⁸ Marie-Laure Ryan, “Interactive Drama: Narrativity in a Highly Interactive Environment.” *MFS Modern Fiction Studies* 43, no. 3 (1997): 677-707, doi:10.1353/mfs.1997.0065.

themselves. Play requires interaction, and in a definition of games based on play, such a seemingly low-interactivity system like hypertext does not fall within its bounds.

Indeed, this characterization of hypertext as not truly facilitating interaction has led to it being defined out of other categories as well, like “interactive fiction”. In the first chapter of *Twisty Little Passages*, for example, Nick Montfort offers a radically narrow understanding of interactive fiction based on “[providing] the computer with the means to parse or understand natural language [which is] used to react to typed text from the interactor.”⁸⁹ He further insists on “an intermediate, programmatic representation of the narrated world” as another essential element of interactive fiction which hypertext lacks,⁹⁰ a claim with which Aarseth would surely be sympathetic given his criteria for the elements of a game. “Any game consists of three aspects: (1) rules, (2) a material/semiotic system (a gameworld), (3) gameplay (the events resulting from application of the rules to the gameworld).”⁹¹ It is doubtful that Aarseth would consider clicking a link to move to a new passage to be a true application of a rule to the gameworld, given that his framework implies a change of state. To what extent can the state of a piece of hypertext fiction truly change if each passage and link is static, already definitively written by the author?

Yet my definition of game, the one borrowed from Anna Anthropy which posits games as “an experience created by rules,”⁹² is not concerned with interactivity or dynamic representations of a mutable game state. It is simply concerned with rules. Aarseth argues that hypertext fiction

⁸⁹ Nick Montfort, *Twisty Little Passages: an Approach to Interactive Fiction* (Cambridge, MA: MIT Press, 2005), 12.

⁹⁰ Montfort, 13.

⁹¹ Espen Aarseth, “Genre Trouble: Narrativism and the Art of Simulation,” 47-48.

⁹² Anna Anthropy, *Rise of the Videogame Zinesters: How Freaks, Normals, Amateurs, Artists, Dreamers, Drop-Outs, Queers, Housewives, and People like You Are Taking Back an Art Form* (New York, NY: Seven stories Press, 2012), 43.

lacks recognizable rules. A response might be that some Twine authors have expanded upon the platform through JavaScript and Cascading Style Sheets, using coding to add more rules to their works, and thus making them more like games. For example, Anthropy puts a timer in her game *Queers in love at the end of the world* (2013), which adds a kind of intensity and pressure not often found in the otherwise languid medium. The developer of *Even Cowgirls Bleed* (2013), Christine Love, uses JavaScript to turn the mouse into a wild and unpredictable gun, firing off if the player so much as grazes one of the links; stripped of the agency conveyed by a mouse click, the player finds themselves out of control, another rare sensation for a Twine game. Finally, *Horse Master* (Tom McHenry, 2013) styles itself after a resource-management game where the player builds the statistics of their “horse”. Complete with a small pictorial representation of the game world, *Horse Master* seems to most loudly assert its gameness, even if this is ultimately a ruse, and player choice truly does not matter.⁹³

However, to assert these works and the coding behind them as the proof that Twine games are real games would be to simultaneously assert that any Twine games without coding are *not* games. This is a counterintuitive gesture given that my goal, like Anthropy’s, is to be as inclusive as possible, and to consider Twine itself, and not its interfacing with JavaScript and CSS, as a game engine. Furthermore, any Twine work contains rules. Those rules may be more simplistic than those found in hardcore games, but “click link to progress” and “you cannot interact with text that is not a link” are nonetheless rules because they limit the possibility space for player action. Finally, as Anthropy herself points out, “rule-based” is the less important part of her understanding of games. As she rightfully acknowledges, her definition invites calling any

⁹³ Naomi Clark, “*Horse Master* by Tom McHenry,” in *Videogames for Humans: Twine Authors in Conversation*, ed. merrikk kopas (New York: Instar Books, 2015), 132-133.

number of activities “games”, like filling out a tax form—but a key distinction is that this process does not create a strong experience.⁹⁴ As Motherboard writer Joe Bernardi points out when favorably comparing the Twine game *Brooklyn Trash King* (Ben Esposito, 2012) to *Call of Duty: Black Ops II* (Treyarch, 2012), even Twine games that only use the rules that result from using the default, built-in link structure can create experiences more memorable than those found in “real” games.⁹⁵

Twine games are games. Next question.

Twine and the Game Engine

A number of definitions of the term game engine have been given over the years, and Twine fits a surprising number of them. Rather than just picking one that proves my point, however, I would like to consider a select few and separately evaluate to what extent Twine matches the definition. I want to make this move in order to point toward a fundamental incongruence between the more vague, expansive, and inclusive definitions of a game engine that have been given since its conception, and the extremely limited range of software which actually constitutes the category today. For example, if I were to draw up a new definition of a game engine based solely on those platforms which are called engines, my conclusion would likely be that a game engine *has* to be 3D because, generally speaking, only 3D game engines like Unity and Unreal have branded themselves as such.⁹⁶ Yet obviously, as discussed earlier, there are 2D game

⁹⁴ Anthropy, *Rise of the Videogame Zinesters*, 44.

⁹⁵ Joe Bernardi, “Choose Your Own Adventure-Maker: Twine and the Art of Personal Games,” *Vice*, last modified February 19, 2013, https://www.vice.com/en_us/article/xyyp9a/twine-and-the-art-of-personal-games.

⁹⁶ Within the list of game engines listed on the game engine Wikipedia article (https://en.wikipedia.org/wiki/List_of_game_engines), of the 35 with “Engine” in the title, only 4 are not 3D engines.

engines as well, like Gamemaker and RPG Maker. Finally, by grounding my analysis in the words of those who helped define the game engine industry in the first place, like John Romero and Jason Gregory, my argument is no longer that the term game engine should be expanded to include Twine and other small platforms like it. I do not need to justify calling them game engines instead of using broader terms like “game development platform” and “middleware”. Instead, my argument is that these platforms have *always* belonged in the category of game engine.

The first definition to be considered is one which delimits its use. Benjamin Nicoll and Brendan Keogh argue that “a game engine is a software tool that enables real-time interactive digital content to be created, and a code framework that enables that content to run on different platforms.”⁹⁷ Rather than specifically restricting the game engine to being a tool for creating a game, Nicoll and Keogh point out that these pieces of software “can be (and increasingly are) used to make non-videogame content and applications such as 3D animations, architectural models, complex training simulations, and interactive novels.”⁹⁸ Twine is certainly a software tool that can be used to develop a variety of real-time interactive digital content, including, as established earlier, games and hypertext fiction or poems. Somewhat less obvious is whether that content can run on different platforms, given that the output of Twine is always an HTML file, as opposed to an engine like Unity, which can be configured for making games for Windows, macOS, iOS, Android, Linux, WebGL, PlayStation 4, Oculus Rift, Steam VR, and Nintendo Switch, among many others. Twine is certainly smaller in publishing scope, but the advantage of the HTML file, and one of Twine’s major appeals, is its ubiquity: it can be opened, in essence

⁹⁷ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 9.

⁹⁸ Nicoll and Keogh, 10.

run, in any web browser. Although Twine games likely will not be ported to consoles any time soon, they can certainly run on many different (web browsing) platforms.

The next definition delimits the game engine based on design. It requires that a game engine have

a reasonably well-defined separation between ... core software components (such as the three-dimensional graphics rendering system, the collision detection system, or the audio system) and the art assets, game worlds, and rules of play that [comprise] the player's gaming experience.⁹⁹

The idea is that the game engine does not contain any code that is game-specific, which might be refined as saying that a game engine must be extensible and “can be used as the foundation for making many different games without major modification.”¹⁰⁰ Though Twine lacks some of the core software components Gregory identifies, like a 3D rendering system and collision detection system, it nonetheless contains several others. For example, it provides the functionality for linking passages together or checking the number of times a player has visited a passage.

Although Twine is less extensible and imposes more restrictions on its users in terms of what they can ultimately create when compared to a general-purpose engine like Unity or Unreal,¹⁰¹ the games that can be made with it are nonetheless quite diverse. The works contained, for example, in *Videogames for Humans* are described as “interactive poetry, traditional choose-your-own-adventure games, therapeutic experiences, personal essays, and elaborate jokes.”¹⁰² Some Twine games closely resemble games in other genres, like *Horse Master* or *The*

⁹⁹ Gregory, *Game Engine Architecture*, 11.

¹⁰⁰ Gregory, *Game Engine Architecture*, 11.

¹⁰¹ It is important to note that the extent of Twine's extensibility is largely a question not of the platform itself but the story format someone decides to use. Twine 2, for example, has three official story formats, which each give the user a different level of freedom when making their own modifications. Harlowe, the default format, is the least extensible, while Snowman provides very little built-in functionality but is the most customizable of the official story formats. SugarCube is between the two, being more limited than Snowman but still allowing users to define their own repeatable code pieces called “macros”.

¹⁰² merrikk kopas, introduction to *Videogames for Humans*, 13.

pipes are my solitude (James Shasha, 2014) which is a Twine-version of *Flappy Bird* (dotGears, 2013). Others unwind more like stories, like *Even Cowgirls Bleed*, or poetry, like *Anhedonia* (Maddox Pratt, 2013). The world of Twine games is broad—even when they may be technically similar, all employing the same basic hypertext principles of passages and links, their collective content and style is far-reaching in scope.

The final definition under consideration is the broadest of them all, and it comes from the man who helped to create the category in the first place. According to Henry Lowood, the men who coined the term game engine, John Carmack and John Romero, “were both automobile enthusiasts and, as Romero explained, the engine ‘is the heart of the car, this is the heart of the game; it’s the thing that powers it’.”¹⁰³ Such a definition emphasizes not a specific design or use but rather the essential nature of that use as key to the game design process. Not only that, but it evokes a sense of affection, something which is intimate not only with the game built on top of it but also the people who use it. Based on this definition, a game engine is merely that which lies underneath a game’s hood, integral not only to the process of its design but its functionality after it is completed. Twine games could not run without the code Twine provides¹⁰⁴ to parse and render the HTML when it is opened in a web browser. Furthermore, as opposed to more generalized game engines, because of the narrow limitations it imposes on its users, Twine’s presence can be felt strongly in the games created with it, be it for the aesthetic defaults which the game maintains (black background, white text, blue links) or the links which structure the game space and interactivity, which are also afforded by the platform. In short, Twine is

¹⁰³ Lowood, “Game Engines and Game Histories,” 186.

¹⁰⁴ This is another situation where it is worthwhile to distinguish between Twine and the story formats built to develop in it. Each story format has a JavaScript file called `engine.js` which provides unique functionality depending on the specifics of the story format.

undeniably the heart of any Twine game, providing both a game's core aesthetics and mechanics. It is thus, per the definition of Romero, undeniably a game engine.

Twine and the Queer Game Engine

Now comes the final question, harder to answer than any of the others because it combines three different nebulous terms: queer, game, and engine. As the definition of game and game engine have already been discussed this chapter, before finally considering whether Twine is a queer game engine, it is worthwhile to return briefly to *queer*. What kind of queerness is being discussed going forward? As mentioned in the last chapter, a representational queerness, defined by the fictional representations of queer characters within the games themselves, has little relevance to a conversation about the possibility of a queer game engine. After all, game engines are defined by a separation between game and engine code, so anything game-specific like characters is not the prerogative of the engine. Thus, while Twine has been used to make a number of queer games, perhaps more so than other engines, this is not sufficient to make Twine a queer game engine.

Another kind of queerness that is commonly evoked specifically in conversations about Twine focuses on the communities that use it. As Alison Harvey argues:

[Twine's queerness] stems from [its] accessibility, and its resulting use by a wide range of people, including women, genderqueer, and trans* people, poor people, older people, younger people, people of color and first-time game-makers ... Twine queers game design through its inclusion of those typically excluded from the traditional training and education of gamemakers.¹⁰⁵

¹⁰⁵ Alison Harvey, "Twine's revolution: Democratizing, depoliticization, and the queering of game design," *GAME 3* (2014): 99.

Yet the extent to which this form of queerness is an inherent part of the platform is debatable.

Benjamin Nicoll explains that:

[although] Twine is ... a free-to-use and intuitive piece of software that encourages a highly specialized and authorial approach to videogame development ... it attained its status [as a game engine] because it was taken up and developed by people whose voices had, historically, been marginalized by the dominant cultures of computing and videogaming.¹⁰⁶

Taken together, Harvey and Nicoll suggest that Twine's queerness is, if not solely, then mostly located not in the platform itself—its design and affordances—but in the marginalized status of the community that has used it.

However, use by queer people is not sufficient to make a game engine queer. The reasoning for this is plainly evident in normative, industry-approved game engines like Unity which can and are used by LGBT game designers who work in fundamentally oppressive development environments. When clarifying the limits of Unity's democratizing effect, Nicoll and Keogh argue that Unity has not “addressed tendencies within studio environments to turn a blind eye towards issues of precariousness, toxicity, burnout, and exploitation,”¹⁰⁷ and this can clearly be seen in a game like MidBoss' *2064: Read Only Memories* (2015), which was developed in Unity. The sexuality of the studio's then-CEO, Matt Conn, and many of his employees did not prevent the development of a highly toxic work environment, which was brought to light thanks to trans women like Katherine Cross.¹⁰⁸ To suggest that Unity is queer simply because it could be used by them would be irresponsible as it would overlook the harm

¹⁰⁶ Nicoll, *Minor Platforms in Videogame History*, 185.

¹⁰⁷ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 113.

¹⁰⁸ Katherine Cross, “Analysis: A closer look at MidBoss' chaotic, hostile work environment,” *Gamasutra*, last modified March 28, 2018, https://www.gamasutra.com/view/news/315910/Analysis_A_closer_look_at_MidBoss_chaotic_hostile_work_environment.php.

done while the platform was being used. A queer game engine should be deeply preoccupied with being used in an ethical manner, and thus its relationship to usage and production is more complicated than simply asking who uses it.

The difficulty, not only for analyzing Twine but conceptualizing the queer game engine in general, is trying to find a form of queerness that resides within the platform itself without eliding the precarious and marginalized communities who use it. My solution to this is to focus on developer-centric queerness. This is as opposed to, for example, Bo Ruberg's framework of playing queerly, which they describe as “a mode of nearly infinite possibility, brought to games largely through their players rather than systems structured by (mainstream) developers.”¹⁰⁹ Ruberg concentrates on players as the site of uncovering or encountering queerness in games—by, for example, “playing to lose ... playing to hurt ... playing too fast or too slow.”¹¹⁰ The arguments made by Ruberg are undeniably valuable, particularly in asserting that by “identifying how video games, even those that appear most simplistic or heteronormative, can resonate powerfully with queer experiences, queerness emerges as a form of potential that lies in all games.”¹¹¹ Still, with regards to analyzing Twine specifically and discussing the queer game engine generally, such a perspective is less useful than looking for those moments of queerness which arise at the point of interaction between a developer and the game engine, where queerness might be described as revealing unspoken and oppressive norms, disturbing those norms, and then suggesting alternatives which move beyond them.¹¹²

¹⁰⁹ Bo Ruberg, *Video Games Have Always Been Queer* (New York: New York University Press, 2019), 17.

¹¹⁰ Ruberg, *Video Games Have Always Been Queer*, 17-8.

¹¹¹ Ruberg, 17.

¹¹² Naomi Clark, “What is Queerness in Games Anyway?” in *Queer Game Studies* ed. Bo Ruberg and Adrienne Shaw (Minneapolis: University of Minnesota Press, 2017), 4.

One such point of interaction is the relationship between the medium of hypertext and authorial control, a relationship which is sustained within Twine as well because of its origins as a tool for authoring hypertext. This relationship is explained by Lev Manovich as follows:

Before, we would read a sentence of a story or a line of a poem and think of other lines, images, memories. Now interactive media asks us to click on a highlighted sentence to go to another sentence. In short, we are asked to follow pre-programmed, objectively existing associations. Put differently ... we are asked to mistake the structure of somebody else's mind for our own.¹¹³

He argues, in short, that hypertext restricts its readers by forcing them to occupy the mind of the author. Yet this positioning, where the author/developer can “exert control over others’ movements through a virtual or symbolic space [and] exploit a power gradient within the textual construct”¹¹⁴ — takes on a different connotation when it is in the context of a *game*. This is because, as Matte Brice explains:

Within conventional wisdom of designing and critiquing games lies the assumption that the player is paramount. Much of criticism writes to inform players, games are designed with the player in the forefront of the developer's mind. There is even the idea that games are completed by players, that without players, there is no game.¹¹⁵

To privilege the developer, then, in moments where “[t]he program does not answer our expectations [and unsettles] an assumption that the story [game] really does exist to please us”¹¹⁶ is to offer a queer disruption of game design norms which privilege the player.

This is a disruption which Brice in particular has championed in her assertions that “[p]layers are overrated” and that “[i]f your game leaves out traditional qualities and emphasizes

¹¹³ Manovich, *The Language of New Media*, 61.

¹¹⁴ Stuart Moulthrop, “Traveling in the Breakdown Lane: A Principle of Resistance for Hypertext,” *Mosaic: An Interdisciplinary Critical Journal* 28, no. 4 (1995): 66.

¹¹⁵ Mattie Brice, “Death of the Player”, last edited October 29, 2013, <http://www.mattiebrice.com/death-of-the-player/>.

¹¹⁶ Moulthrop, “Traveling in the Breakdown Lane,” 73.

voice, then player-centric design is a useless paradigm for you.”¹¹⁷ Building off of kink terminology like *domme/sub* and the consent-scene-aftercare framework, Brice further develops these critiques into a new framework for reconceptualizing the relationship between the developer, their game, and the player. In this new model, “dommes can stand in for the game-design role as the person who is crafting an experience for the submissive, and the submissive relinquishes control after negotiating with the *domme* the rules of the play session.”

¹¹⁸ Though not so severe as to call for ignoring the player entirely, this framework does dislocate them as the most essential part of the game, as well as offers a way to conceptualize the power of the game developer within ethical and consensual terms.

By virtue of being initially created as a tool for authoring hypertext, and the idioms of that medium continuing to be present in its games, Twine is uniquely equipped to engender this kind of queer developer-centric design. This line of reasoning is further supported by analyses of the link structure which define the rule systems of both hypertext and Twine. One such analysis avers that:

the process of linking itself implies a rhetoric of repeated disorientation and reorientation. That is, a reader departs from a familiar node and arrives, in an instant, at an unfamiliar one without any immediate understanding of its relationship to what came before or its bearings on what comes next.¹¹⁹

This question of disorientation figures prominently into Elise Vist’s essay “Cyborg Games: Videogame Blasphemy and Disorientation,” where she connects the power of her new category, cyborg games, to the creation of “a space where normative bodies and normative *gamers* are not

¹¹⁷ Brice, “Death of the Player”.

¹¹⁸ Mattie Brice, “Play and Be Real about It,” in *Queer Game Studies* ed. Bo Ruberg and Adrienne Shaw (Minneapolis: University of Minnesota Press, 2017), 79.

¹¹⁹ David Ciccoricco, *Reading Network Fiction* (Tuscaloosa: University of Alabama Press, 2007), 80.

immediately oriented.”¹²⁰ In other words, cyborg games (a category which overlaps with queer games to such an extent that Vist’s arguments apply to them as well) are destabilizing because they are disorienting to a group which otherwise enjoys the incredible privilege of always being correctly oriented in a video game.

In this sense, then, Twine is the perfect tool for queer game design because hypertext is predicated on moments of disorientation. The tendency amongst queer game developers to reject the category of “empathy game” further strengthens Twine’s position because the notion of reorientation is subsequently complicated. Empathy can be understood as a kind of reorientation; by making the unfamiliar familiar and known, one finds oneself on common ground once more. This framework presents disorientation as something which can and should be recovered from, corrected. However, a cisgender person playing *dys4ia* (Anna Anthropy, 2012) cannot be oriented correctly, and any pretensions of using empathy to end the disorientation overstate empathy’s capacity to let someone know, really, what an experience is like.¹²¹

Thus, even given the current state of game design, where free and accessible game development tools are more common and Twine is not unique in its ability to promote the creation of small, personal, and queer games,¹²² it is nonetheless distinct. Having been adopted rather than designed for game development, and having instead been designed for the developer-focused medium of hypertext, Twine is particularly well-suited for making games that

¹²⁰ Elise Vist, “Cyborg Games: Videogame Blasphemy and Disorientation.” *Loading... The Journal of the Canadian Game Studies Association* 9, no. 14 (2015): 56.

¹²¹ Cecilia D’Anastasio, “Why Video Games Can’t Teach You Empathy,” *Motherboard*, last modified May 15, 2015, https://www.vice.com/en_us/article/mgbwpv/empathy-games-dont-exist.

¹²² Nicoll, *Minor Platforms in Videogame History*, 159.

are, above all else, personal, a queer gesture in an industry which insists that games cannot even exist without a player.¹²³

A sticking point in this argument is that it comes down to the same question asked earlier of the technological and economic affordances which encouraged queer game designers to use Twine in the first place. It is not clear whether its aptitude for creating disorienting, highly authorial works is queer in and of itself as a queer quality inherited from hypertext, or whether it is a consequence of how Twine has been adopted and applied by a particularly queer community of game developers. After all, the traits discussed above only challenge and destabilize established game design norms when Twine is actually used to make games. Rather than coming down on one side of the question or the other, I would like to suggest an alternative explanation which does not deny the advantages of Twine's affordances but also locates the medium's queerness somewhere else.

In a presentation at MIT's *Transformations of the Book* Conference, famous hypertext author Shelley Jackson spoke of hypertext as being inherently feminine. She was not asserting hypertext as a female-exclusive medium, but rather that the medium encouraged a style of writing which was "amorphous, indirect, impure, diffuse, multiple, evasive" in contrast to "good" writing which "is direct, effective, clean as a bleached bone" and, for Jackson, coded masculine. As she explains, bad or feminine writing "is all flesh, and dirty flesh at that: clogged with a build-up of clutter and crud, knick-knacks and fripperies encrusted on every surface, a kind of gluey scum gathering in the chinks."¹²⁴ This feminine aesthetic of excess, of accumulation and trash, continues to be championed in the twenty-first century by Twine

¹²³ Brice, "Death of the Player".

¹²⁴ Shelly Jackson, "Stitch Bitch: the patchwork girl," presented at Transformations of the Book Conference, Cambridge, Massachusetts, October 1998, n.p.

developers, particularly Porpentine. Her prose is rich and confusing, forcing her readers to work in order to make sense of the sexualized and slimy futuristic worlds she imagines. Consider this excerpt from her game *Vesp: A History of Sapphic Scaphism* (figure 4).

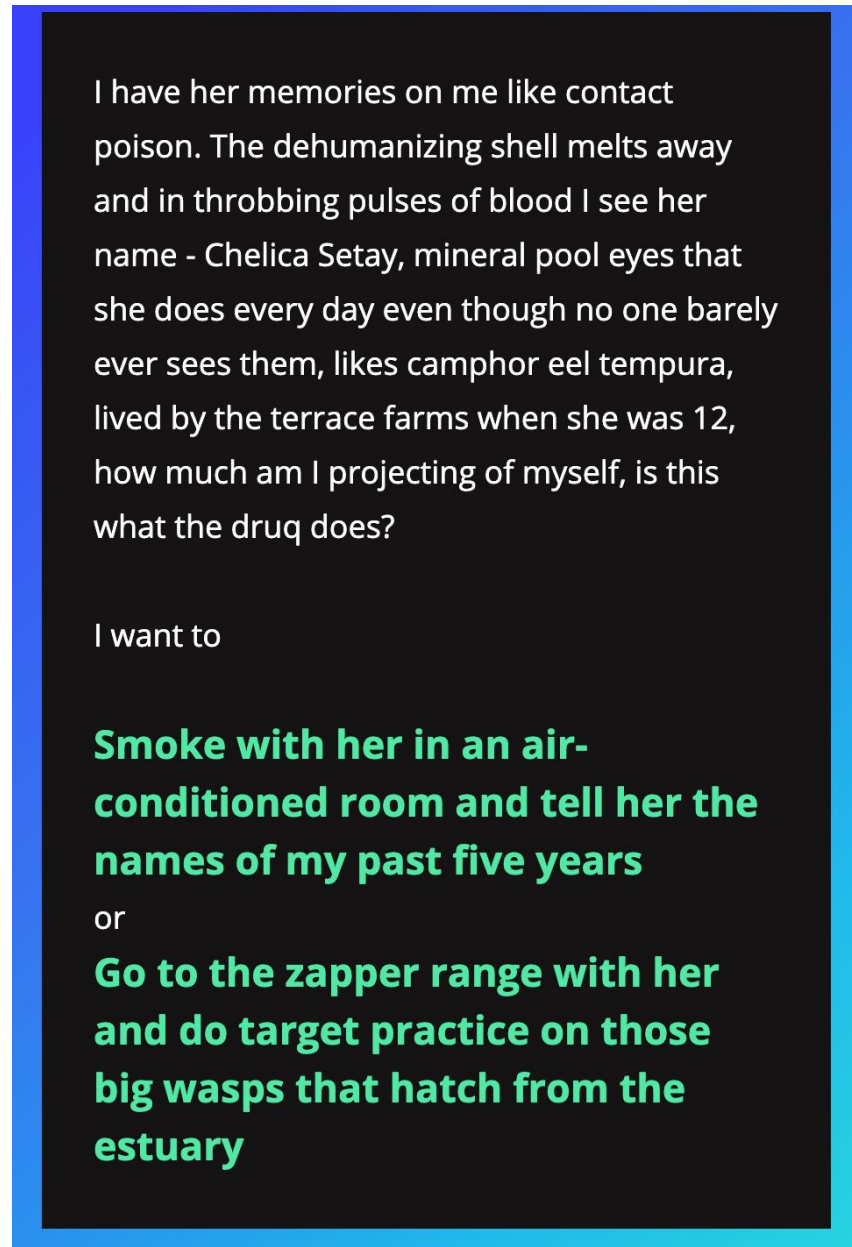


Figure 4. A passage from Porpentine's *Vesp: A History of Sapphic Scaphism*, highlighting the colored background (as opposed to the default black Twine background), the green links (as opposed to the default blue), and Porpentine's literary style.

Source: (Screenshot from <http://slimedaughter.com/games/twine/vesp/>)

The reader has no reference point for what “camphor eels” or terrace farms are or why there are big wasps being grown in estuaries, and Porpentine provides few answers. Her prose style constantly frustrates reorientation by relying on the fantastic and nonexistent. In the absence of specific clarity, Porpentine’s games traffic general affect above all else. Rather than being direct or clean, they operate within that feminine stylistic mode which Jackson associates with hypertext.

However, Porpentine does not call her works hypertext fiction; she calls them games, and in doing so, she performs a curious type of work. To assert that videogames are the complementary masculine medium to hypertext’s femininity is to repeat an all-too-frequent erasure of the women and LGBT people who make games or identify as gamers. For example, despite the stereotypes of the hypermasculine male gamer, research has shown that nearly half of gamers are women.¹²⁵ However, as Bo Ruberg notes, it is nonetheless also true that “[g]ames culture ... continues to be dominated largely by the voices of straight, white, male consumers,” who have both the loudest voices and the greatest social capital.¹²⁶ Furthermore, many of the established game norms challenged by queer games are in and of themselves masculine—the valorization of competition and violence, for example, with violence being in a way endemic to the game engine itself, given the technology’s historical association with first-person shooters. What I mean to suggest here is that when Porpentine, or other Twine developers, call their works games, they are bringing two mediums into contact: one historically feminine, the other historically masculine. Rather than read this as a moment of heterosexual touch, I’d instead like

¹²⁵ “2019 Essential Facts About the Computer and Video Game Industry,” Entertainment Software Association, accessed November 29, 2019, <https://www.thesa.com/esa-research/2019-essential-facts-about-the-computer-and-video-game-industry/>.

¹²⁶ Ruberg, *Video Games Have Always Been Queer*, 58-9.

to suggest that Twine games demand that these two gendered mediums exist in the same work. In the same body. Thus, they embody transness. Just as transmasculinity and transmasculine aesthetics such as radical softness¹²⁷ destabilize assumptions of what men should be, just as they expand conceptions of what masculinity should look like, Twine games trouble the medium they transition to from hypertext.

Yet still this transition is only possible because of the queer game designers who brought the two mediums into contact in the first place. It would thus seem then that I am back in the same place as Benjamin Nicoll when he warned that Twine only democratizes game development insofar as it is *used* for game development.

Before writing off the possibility that Twine is a queer game engine, however, I would like to discuss one more aspect of the platform: its relationship to labor. Bo Ruberg has argued that the prevalent narrative around queer games, of “queer subjects operating at the margins ... bringing much-needed change to video games” can have problematic consequences by “obfuscating both the challenges that many queer indie game-makers face and the tensions between marginalized game-makers and the mainstream game industry.”¹²⁸ Out of a deep respect for these game makers who paved the way for my own involvement with the queer games scene, I do not want to perpetuate this same obfuscation. Rather than asserting Twine’s usage as a queer utopian ideal, I want to acknowledge its limits and the harm it can do to those using it.

Because Twine was not designed to make games but rather hypertext fiction, which itself exists within a historical context of individualized labor, it lacks certain technical features industry game engines use to organize and distribute work. For example, game engines have

¹²⁷ Teddy Pozo, “Queer Games After Empathy: Feminism and Haptic Game Design Aesthetics from Consent to Cuteness to the Radically Soft,” *Game Studies* 18 no. 3 (2018), <http://gamestudies.org/1803/articles/pozo>.

¹²⁸ Bo Ruberg, “The Precarious Labor of Queer Indie Game-making,” 779.

historically been built to engender certain formations of labor, not unlike the divisions imagined by Charles Babbage himself with his calculating engines. Like the modular and hierarchical architecture of the game engine, the Difference Engine was built on top of “a descending separation between the production of the formula ... the conversion into numbers ... and calculation.”¹²⁹ Both are designed to enable a division of labor. The team designing the combat system can work separately from the artists, who work separately from the sound designers, who work separately from the level designers. Likewise, in the Difference Engine, “the conceptual work of producing formulas [is separate from] the mechanical work of addition,”¹³⁰ with the mental work being more highly valued than the physical.

To a certain extent, this is beginning to change in indie game development. Unity, for example, “[foregoes] the hierarchical production pipelines that characterized video game development in the 2000s and early 2010s [by making] it easier for artists, designers, and audio engineers to directly participate in the collaborative process.”¹³¹ These changes are slow, however, and small when compared to a piece of software like Twine which was designed for individual use, and thus lacks many of the structures found in Unity or other industry game engines which would suggest or require a hierarchical division of labor. Indeed, it is hard to split labor at all because Twine lacks mechanisms for even equitable collaboration. This is because all Twine files are stored locally, and there is no built-in infrastructure for sending them to other users or uploading them to a shared remote repository like GitHub or Bitbucket. As a result, collaboratively working on a project in Twine requires sending the project file back and forth every time a change is made. To the extent that Unity claims to create new values by allegedly

¹²⁹ Purbrick, “The Dream Machine,” 21.

¹³⁰ Purbrick, 21.

¹³¹ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 54.

refusing to prioritize one set of skills over another, Twine mandates that its singular user have all the skills required to realize their vision, necessarily restricting any works made in Twine in terms of scope and scale.

While highly individualized labor can be empowering—for example, in enabling developers to create deeply personal games—it also comes with perils that Twine is not well-equipped to counter. This is true of other game engines like Unity, where the claim of democratizing game development “contributes to a romanticized and individualized notion of creative work under neoliberalism that commonly sugar coats precarious, contingent work, or a lack of employment opportunities under the utopic language of ‘entrepreneurism’” in which one’s failure to succeed is always and only one’s own fault.¹³² Twine certainly does not deploy the language of entrepreneurship—its own creator does not work on it full time,¹³³ yet avoiding one problem creates another. As Nicoll and Keogh note, even within the romanticized individual labor model put forth by Unity, its “commitment to accessibility, low-cost software, and affective community creates an affective space where developers are granted a slight degree of social security to explore possibilities for self-entrepreneurship in what would otherwise be a career path fraught with risk and uncertainty.”¹³⁴ Twine lacks even this slightest degree of security.

For example, by virtue of being built in Unity, Unity games are granted a degree of legitimacy which allows them to be sold through formal and monetized distribution platforms like Steam. Although Twine games can be found on Steam as well—like, most notably,

¹³² Nicoll and Keogh, 58.

¹³³ Klimas, “Twine: Past, Present, and Future.”

¹³⁴ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 109.

Depression Quest—it is far more common to encounter them for free on personal sites, like Porpentine’s games, or on a site like itch.io, like Anna Anthropy’s games.¹³⁵

Unity also offers its users opportunities to monetize their own games beyond the initial purchase via Unity Ads. This is a Unity-specific ad network which enables particularly developers of free mobile games to make money by putting ads in their games.¹³⁶ Twine has no such mechanisms, providing no native ways for its users to be compensated for their labor.

Labor beyond game-making is equally undervalued within the Twine community. For example, Unity users can upload custom content like scripts, shaders, and sprites to the Asset Store where they can be distributed freely or sold for a one-time fee. While “[t]here is a strong incentive among Asset Store developers to provide one’s plugins or assets for free [because of] an open-source mentality and a desire to ‘give back’ to the community,”¹³⁷ the fact that Unity developers even have the option to be compensated, if not completely equitably, for this labor is notable compared to Twine. All Twine story formats, for example, are made and maintained without generating revenue for their creator. Or there are Twine Instructable guides, which are more or less equivalent to a Unity asset in that they provide a bundle of functionality—like adding a combination lock to the story¹³⁸ — yet are distributed freely, without the opportunity for compensation. The assumption is that, if one is doing work on or with Twine at any step in the process, one is doing it for free. To champion this free labor is to ignore the other costs of game development, such as “[t]he emotional cost of making video games [which] is considerably higher for those creators who find themselves the targets of homophobia, transphobia, misogyny,

¹³⁵ See <http://slimedaughter.com/> and <https://w.itch.io/>

¹³⁶ Nicoll and Keogh, *The Unity Game Engine and the Circuits of Cultural Software*, 39.

¹³⁷ Nicoll and Keogh, 40.

¹³⁸ ddall1, “Building a Combination Lock in Twine 2, Harlowe 2,” *Instructables*, accessed November 29, 2019, <https://www.instructables.com/id/Building-a-Combination-Lock-in-Twine-2-Harlowe-2/>.

racism, and ableism” or “the cost of life itself”— putting food in one’s stomach, keeping the lights on, supporting oneself financially.¹³⁹

Ultimately, then, because of the number of queer people among Twine’s user base, this inability to be justly compensated for one’s work specifically harms the group of people Twine has been championed for uplifting and empowering, because “queer creators often come to the work of indie game-making from places of financial precarity.”¹⁴⁰ The question of labor is a tricky one. As Alison Harvey points out in her analysis of Twine, “the emancipatory possibilities of queer gamemaking do not provide an adequate answer when the sustainability of this life for many of these game-makers is a constant source of fear.”¹⁴¹ Twine does not provide an adequate answer either, and it is difficult to imagine that it could do so without violating the principles which led to its “revolution” in the first place. Unity has the money to, say, invest in Unity Ads. Twine does not, because, unlike Unity which makes revenue off of its paid licenses, Twine is completely free.

Twine is not a queer game engine. At best, it has a sort of situational queerness/transness granted to it through the (undercompensated) labor of queer game developers who first brought the platform into contact with the gaming industry, where it was destabilizing because it had been built differently, with different technical affordances and different understandings of labor. But it was not built with the intention of being a different game development tool. Rather than being inherent, its queerness came later.

Finally, the scope of that queerness must be called into question. Even if it was something that resided within the platform itself, Twine would still unfortunately lack the power

¹³⁹ Ruberg, “The Precarious Labor of Queer Indie Game-making,” 783.

¹⁴⁰ Ruberg, 784.

¹⁴¹ Harvey, “Twine’s revolution,” 105.

to make a truly incisive critique of and intervention into the world of 3D game design, where the game engine is most present and most problematic. In fact, most tools championed for making games that are small, personal, and queer are, to a certain extent, similarly limited because they are largely used for two-dimensional production. There is of course value in this, particularly in the context of an industry where the games with the largest budgets and profits are 3D games, where AAA studios have all but abandoned making 2D games. This work is also far from done, yet to challenge directly Unity, Unreal, Frostbite, id Software 6, and any other 3D game engine which can uncontroversially trace its legacy through the history of pursuing photo- and hyperrealistic graphics and first-person shooters, a different tool is required. The solution is not just a queer game engine. It is the imaginary, the impossible: a queer 3D game engine.

Chapter 3: Queering the Engine's Bodies

Tara McPherson once asserted that “[i]f the core activity of the digital humanities has been the building of tools, we should design our tools differently, in a mode that explicitly engages power and difference from the get-go, laying bare our theoretical allegiances and exploring the intra-actions of culture and matter.”¹ The preceding chapters have been concerned with establishing the cultural and theoretical context of the queer game engine's intervention, as well as exploring power, particularly that of normativity and ubiquity. Per McPherson's mandate to design in such a way that takes difference into account, these conversations are not over, but rather will continue to occur even as the focus of this work shifts to that core activity of the digital humanities: building tools. Building the queer game engine.

Yet within the digital humanities, there have been numerous different approaches to building differently. Some involve fully realizing a hardware or software project, such as McPherson and the Alliance for Networking Visual Culture's platform for digital scholarship, Scalar.² Others result in digital objects which approximate existing forms of technology yet are, for some reason or another, nonfunctional. The .txt files which make up Zach Blas' *transCoder*, for example, combine a queer/trans poetics with a coding one in order to produce a “queer programming anti-language [which] provides new programmatic and linguistic possibilities for the queer sociality.”³ Finally, some reject “the requirement that the digital humanities create working technologies. In lieu of tools [they] offer up theoretical vaporware, speculative potential

¹ Tara McPherson, “Designing for Difference,” *Differences* 25, no. 1 (January 2014): doi: 10.1215/10407391-2420039, 182.

² McPherson, 183.

³ Zach Blas and Micha Cárdenas, “Imaginary Computational Systems: Queer Technologies and Transreal Aesthetics,” *AI & Society* 28, no. 4 (October 2013): doi: 10.1007/s00146-013-0502-y, 561

ware, ephemeral praxis.”⁴ “QueerOS: A User’s Manual” best represents this last form of building, presenting “an accessible introduction ... to a new operating system, with each component given a poetic description of its features and limitations.”⁵ Thus, classical components of OS software like Interface and Memory are described in fantastical, speculative terms; extrapolated from their context in existing technological systems, these terms are put to work imagining a social operating system responsible for organizing ethical relationships on a global scale.

What does building the queer game engine look like? This is a question without a definitive answer, and I eagerly look forward to seeing other contributions toward this project which take different approaches. For my part, I draw inspiration from the introduction to “QueerOS: A User’s Manual” and particularly the work’s self-categorization as “a speculative proposition for a technical project that does not yet exist and may never come to exist, a project that does not yet function and may never function.”⁶ My intention is not to produce a working piece of software, or to propose its design in great technical detail.

It is the interest, however, in *some* technical detail which aligns my approach with works such as Wendy Chun’s “Habits of Leaking: Of Sluts And Network Cards” or “On ‘Sourcery,’ or Code as Fetish.” In both works, Chun considers a software issue that is both technical and cultural. In the case of the former, she and Sarah Friedland discuss how the lax security of networked systems interfaces with misogyny to create a culture of victim blaming where women

⁴ Fiona Barnett, Zach Blas, Micha Cárdenas, Jacob Gaboury, et. al, “QueerOS: A User’s Manual,” in *Debates in the Digital Humanities 2016*, ed. Matthew Gold and Lauren Klein (University of Minnesota, 2016), <https://dhdebates.gc.cuny.edu/read/untitled/section/e246e073-9e27-4bb2-88b2-af1676cb4a94>, n.p.

⁵ Barnett et. al, “QueerOS: A User’s Manual.”

⁶ Barnett et. al, “QueerOS: A User’s Manual.”

are deemed responsible for the nonconsensual proliferation of their digital image.⁷ In the case of the latter, she analyzes the cultural impulse to fetishize source code “as a magical entity—as a source of causality—when in truth the power lies elsewhere, most importantly in social and machinic relations.”⁸ Both works also end by briefly proposing an alternative cultural perspective which would engender more ethical relations with the technology discussed. For “Habits of Leaking,” Chun proposes a “right to loiter” which would embrace the leaky interconnectedness of the digital network.⁹ For “On ‘Sourcery,’” the solution is conceiving of code as a re-source, as this framework does not conflate source with execution.¹⁰

My approach to building the queer game engine is similar to Chun’s approach to the network or to source code. I will look at a few components of the game engine as they currently exist and discuss their limited compatibility with queerness. Whereas Chun’s solutions are cultural, however, mine are technical—suggesting alternative systems and components that would be implemented in the queer game engine were it to exist.

This is not, of course, the only way to tackle the queer game engine. One could imagine, for example, an approach far more similar to Chun’s, where one analyzes the game engine as a piece of cultural software, advocating for a queerer understanding of it given its problematic positioning in game development. My decision to organize my work around components is motivated first by the fact that they offer a clear point of focus. Rather than being concerned with the whole of the game engine, I can pick a few subsystems on which to concentrate.

⁷ Wendy Hui Kyong Chun and Sarah Friedland, “Habits of Leaking: Of Sluts and Network Cards,” *Differences* 26, no. 2 (2015): doi: 10.1215/10407391-3145937.

⁸ Wendy Hui Kyong Chun, “On ‘Sourcery,’ or Code as Fetish,” *Configurations* 16, no. 3 (2008): doi: 10.1353/con.0.0064, 311.

⁹ Chun and Friedland, “Habits of Leaking,” 19.

¹⁰ Chun, “On ‘Sourcery,’” 321.

Furthermore, this technique aligns well with my skill set, first developed when I built a 2D game engine from scratch in the fall of 2018. My approach to the queer game engine now reflects the approach used then to design a game engine by implementing one component at a time.

Which components, then, have I chosen to prioritize now? Answering this question is a non-trivial task given the complexity of the game engine. As Jason Gregory captures in his diagram of the modern 3D Game engine (fig. 1), this is a complicated piece of software composed of dozens of different parts. Each of these deceptively neat boxes dividing up the game engine's systems is a potential site for queer intervention, yet a comprehensive queering of each box would require far more time and space than I have here. Furthermore, I have no pretensions of writing a definitive manifesto or having a complete idea of what all, or even most, of these components should look like once queered. I chose the components I have because their reconfigurations came naturally to me, yet I would be at a loss for queering many of these boxes, such as resource management, skeletal animations, or audio. I also recognize that, despite critiquing the diagram of layered, independent systems, I treat my two components of interest separately from each other, rather than troubling their boundaries or focusing on boxes that already disturb this neat separation, such as Ragdoll Physics. I hope that, once the queer game engine becomes a collaborative project and not merely the result of my own musings, others will be able to push it in directions that I myself have never before imagined.

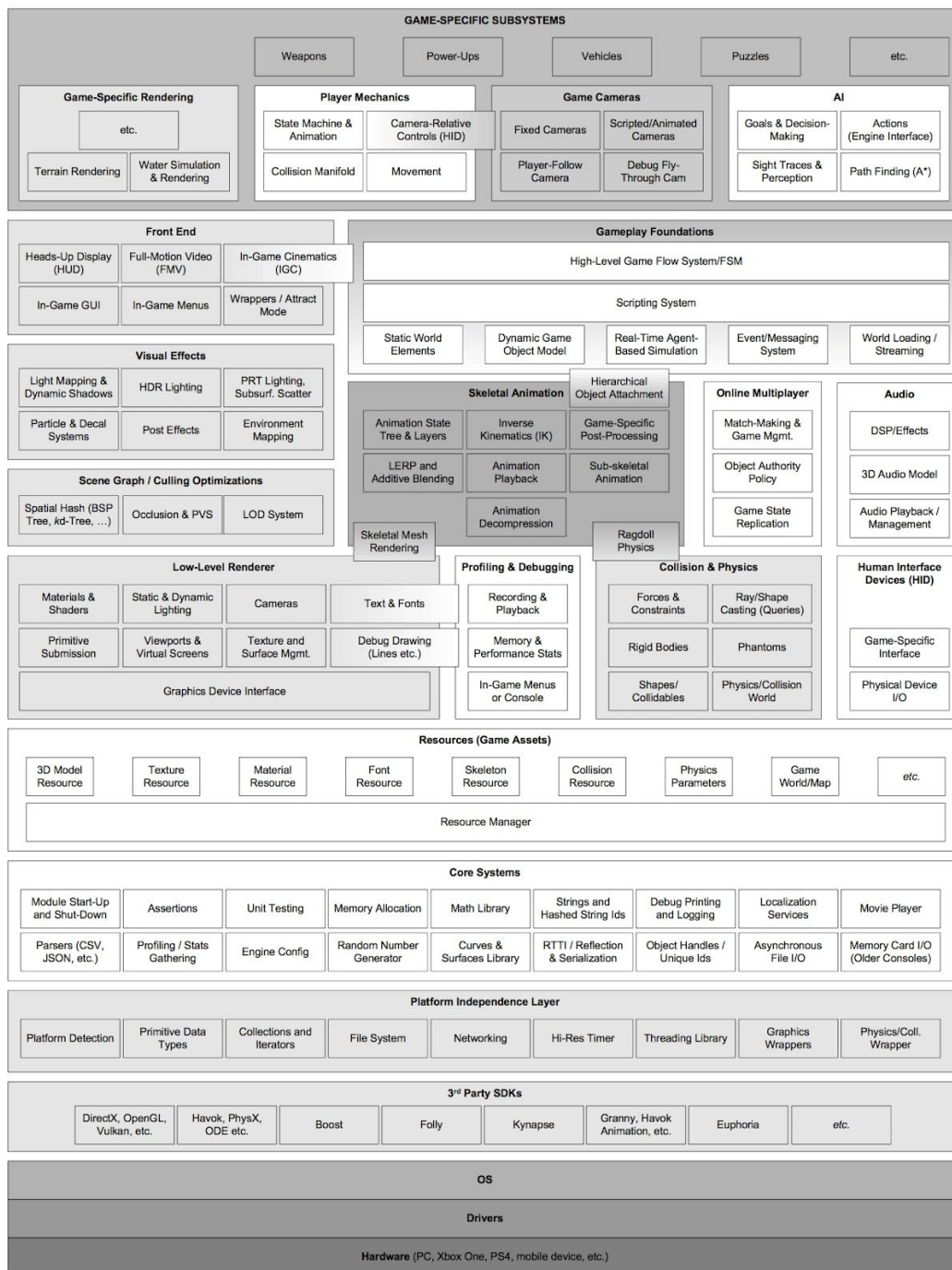


Figure 1. Runtime game engine architecture.
 Source: (Jason Gregory's *Game Engine Architecture*, 29)

The systems that I have chosen are the graphics and physics engine. I was drawn to these components for two reasons. First, they both have a proximity to the digital body which I have found amenable to queerness, as graphics determines how a body appears, and physics determines how it moves. Second, both have already received considerable attention within queer or queer-adjacent analyses of games. Such works offer a jumping-off point for considering how these systems could work queerly within the engine itself, as arguments are extended to encompass not only the games they were written about but the engines that underlie them.

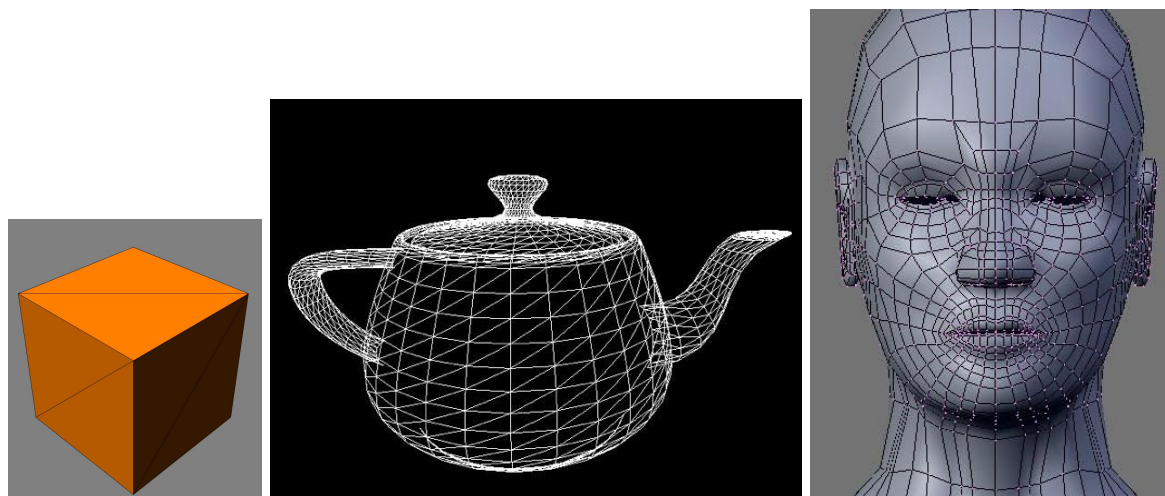
Again, I must stress that, even after restricting my focus to these two components, this work remains far from comprehensive. The graphics and physics engines are themselves complicated systems with profound bodies of literature, both technical and cultural, dedicated to their intricacies. As such, I have limited myself to critiquing one small part of each and proposing one small remedy which the queer game engine could provide. In doing so, I hope to leave the door open not only for more conversations about systems I do not discuss, but also more profound analyses of those I do. This is a preliminary and partial pass at the queer game engine, and it is my most sincere hope that myself and others will continue contributing to its impossible existence.

Graphics

Digital three-dimensional objects are visually represented by something called a mesh. Gregory describes a mesh as “a complex shape composed of triangles [or quadrilaterals] and vertices,”¹¹ and it can be used to model any solid object, from simple cubes (fig. 2) to teapots (fig. 3) to

¹¹ Jason Gregory, *Game Engine Architecture* (Boca Raton: Tayler and Francis Group, LLC, 2009), 51-52.

human faces (fig. 4). Comparing the three example meshes, one might notice that the more complex the shape, the more triangles or quadrilaterals needed to represent it. The cube, for example, only needs two triangles per each face, while approximating the round body of the teapot requires dozens more, and the highly detailed, precise curves of the human nose and mouth need the most of all.



Figures. 2, 3, & 4. Images of meshes for a cube, teapot, and human face.

Sources: (Screenshot from a personal project; <https://teajourney.pub/utah-teapot/> accessed April 13, 2020; <https://blenderartists.org/t/human-face-which-mesh-to-use/43040>, accessed April 13, 2020).

The relationship between mesh detail (or polygon/poly count) and the supposed quality of a digital image is perhaps even clearer when comparing stills from older video games with modern ones. Consider the differences between the original model of Lara Croft from *Tomb Raider* (Core Design, 1996) and the most recent rendition of her body from *Shadow of the Tomb Raider* (Square Enix, 2018) (fig. 5). To refer to 1996 Lara Croft’s body as “low poly” is to point to the fact that her mesh consists of far fewer triangles than that of 2018 Lara Croft, and as a result, her body lacks curves, instead consisting of sharp edges and visible disconnects at the joints. Of course, the number of polygons in Lara Croft’s mesh is not the only thing that has

changed about her visual representation. Other computer graphics advances like improvements in modeling skin, hair, and clothes have also helped 2018 Lara Croft's approximation of realism.



Figure 5. An image contrasting Lara Croft from 1996 (left) with Lara Croft from 2018 (right).
Source: (https://www.youtube.com/watch?v=xUraty_OaGI, accessed April 13, 2020)

Yet the mesh is nonetheless bound up in computer graphics' tireless pursuit of photorealism—what Stephanie Boluk and Patrick LeMieux called the “graphical arms race: a dogged pursuit of ocularcentric spectacle culminating in the hypertrophy of the visual economy” of video games.¹² The products of this arm race include techniques applied *to* the mesh¹³ and techniques for creating the mesh itself. In the case of the latter, this involves complex 3D and 4D scanning with software such as Infinite Realities¹⁴ in order to port real objects—most often

¹² Stephanie Boluk and Patrick LeMieux, *Metagaming: Playing, Competing, Spectating, Cheating, Trading, Making, and Breaking Videogames* (Minneapolis: University of Minnesota Press, 2017), <https://manifold.umn.edu/read/metagaming/section/84dabaa3-647e-4b18-8c8a-ba61cbf48fe3#evi>, n.p.

¹³ For example, 2018 Lara Croft is rendered using a process called dynamic lighting, which is what gives certain parts of her skin a shine. This shine, called the specular highlight, can only be calculated using dynamic lighting as it is based on an object's relationship to a source of light, which changes during gameplay.

¹⁴ Infinite-Realities, accessed April 12, 2020, <https://ir-ltd.net/>.

human bodies—into the digital space. The resulting meshes are expensive, both in terms of their production cost¹⁵ and the time it takes to render them. Photorealism is reserved for those platforms with the hardware capabilities to handle it, as more detail demands more work from a computer or console’s Graphical Processing Unit (GPU). And the systems that can output high-resolution graphics while maintaining a game’s framerate are more expensive—just consider the difference in price between the PlayStation 4 (\$300) and the PlayStation 4 Pro (\$400),¹⁶ which is specifically advertised for its “4K-TV Gaming” technology.¹⁷

I am particularly concerned by sites like Triplegangers,¹⁸ where one can browse and purchase scanned meshes of human faces with names attached. These hollowed subjects have been purged of identity in order to make them vendible, and though only a little over 500 models have been uploaded to Triplegangers, the biases of consumption have already been made clear. The site’s chart of models by age and sex demonstrate an explicit valorization of youth in women not seen to nearly the same extent in men (fig. 6).

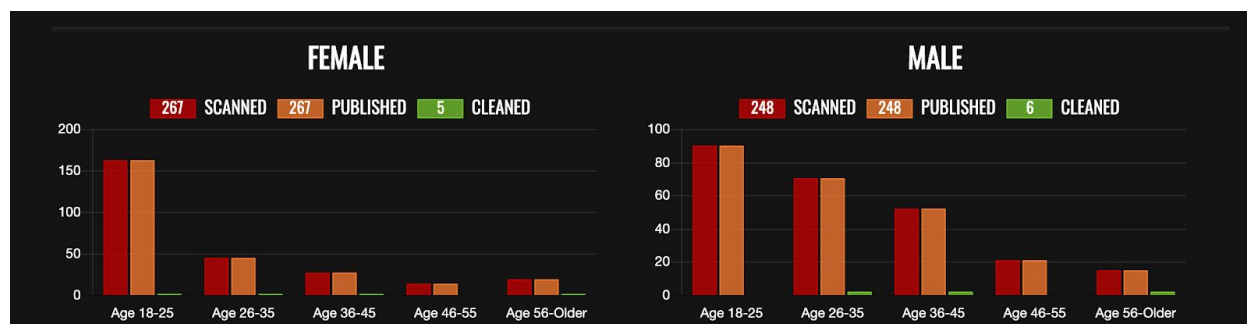


Figure 6. Two graphs displaying the number of meshes available on Triplegangers by sex and age.

Source: (<https://triplegangers.com/>, accessed April 13, 2020)

¹⁵ Licenses for software like Russian 3D Scanner costs upwards of \$100 a year (https://www.russian3dscanner.com/download_and_buy/, accessed April 12, 2020).

¹⁶ Estimates are from prices listed on Best Buy as of March 29, 2020.

¹⁷ “PS4 Pro Console – PlayStation 4 Pro Console: PS4™ Pro Features, Games & Videos,” PlayStation, accessed April 12, 2020, <https://www.playstation.com/en-us/explore/ps4-pro/>.

¹⁸ “Home,” triplegangers, accessed April 12, 2020, <https://triplegangers.com/>.

I am worried about what these regimes, currently still emerging and developing but already organizing themselves along the lines of binary sex and gender, will make of the transgender body. Boluk and LeMieux suggest that the photorealistic rendering process is a form of control, necessitating mastery of a real body in order to ensure its graphical fidelity when it is imported into the digital,¹⁹ and I am worried about bringing the transgender body, already regimented by medical, governmental, and cultural establishments, into contact with another system of control.

Of course, there are techniques for modeling transgender bodies that does not rely on such invasive practices. Consider Krem, from *Dragon Age: Inquisition* (BioWare, 2014). Rather than intensively sculpting a unique mesh based off of a real trans man's body, Krem shares a mesh with the game's other cisgender human men. His transgender identity is revealed only through dialogue. While preferable to the dangerous trope of trans characters being outed via the forcible revealing of their body, such a representation nonetheless presents the trans and cis body as identical, trafficking a dream of passing perfectly, transitioning seamlessly—a dream which cannot be attained by all or even most trans people. Indeed, this dream is not even shared by all of us, as Sandy Stone noted in *The Empire Strikes Back: A Posttranssexual Manifesto*:

the transsexuals for whom gender identity is something different from *and perhaps irrelevant to* physical genitalia are occulted by those for whom the power of the medical/psychological establishments, and their ability to act as gatekeepers for cultural norms, is the final authority for what counts as a culturally intelligible body.²⁰

Though Stone was writing before the proliferation of nonbinary identities in the 21st century, there is a resonance between her writing and the experiences of those transgender people who

¹⁹ Boluk and LeMieux, *Metagaming*, n.p.

²⁰ Sandy Stone, "The *Empire Strikes Back*: A Posttranssexual Manifesto," in *The Transgender Studies Reader* ed. Susan Stryker and Stephen Whittle (New York: Routledge, 2006), 232.

have no “opposite” gender to transition toward. For whom the perfectly transitioned body does not reflect their desires.

Thus, *Inquisition*’s attempt to incorporate the transgender body is a mixed success at best, like many other examples of transgender people in AAA games which do not concern themselves with the question of body. However, to bring those bodies into games via the techniques currently used to render photorealistic humans would be to subject them to the forces of mastery and control which plague photorealistic 3D representations.

A response might be that the solution is to look outside of photorealism, a category which Boluk and LeMieux overlook in their critique. There are a number of prominent examples of non-photorealistic AAA games, many of them made by Nintendo, yet here it is important to distinguish between aesthetic and technique. While Nintendo does not make use of the aesthetics of photorealism, its games nonetheless frequently feature techniques which computer graphics has developed over the decades to make 3D digital images more realistic. Consider the difference in how shadows appear in *Animal Crossing: New Leaf* (2012) versus *Animal Crossing: New Horizons* (2020). In *New Leaf*, shadows are either static for non-moving objects or simple circles for moving ones (fig. 7). By contrast, the shadows in *New Horizons* are generated in real time as the game is playing. As a tree blows in the wind, its shadow moves as well. As the characters stroll about the town, their shadow is a perfect two-dimensional projection of their form (fig. 8). This change in technique speaks to Nintendo’s participation in the graphical arms race, as the tools of photorealism such as shadow mapping are adopted and used even if the aesthetics of it are not.



Figures 7 & 8. Images from *Animal Crossing: New Leaf* and *Animal Crossing: New Horizons*, demonstrating the difference in graphics between the two games.

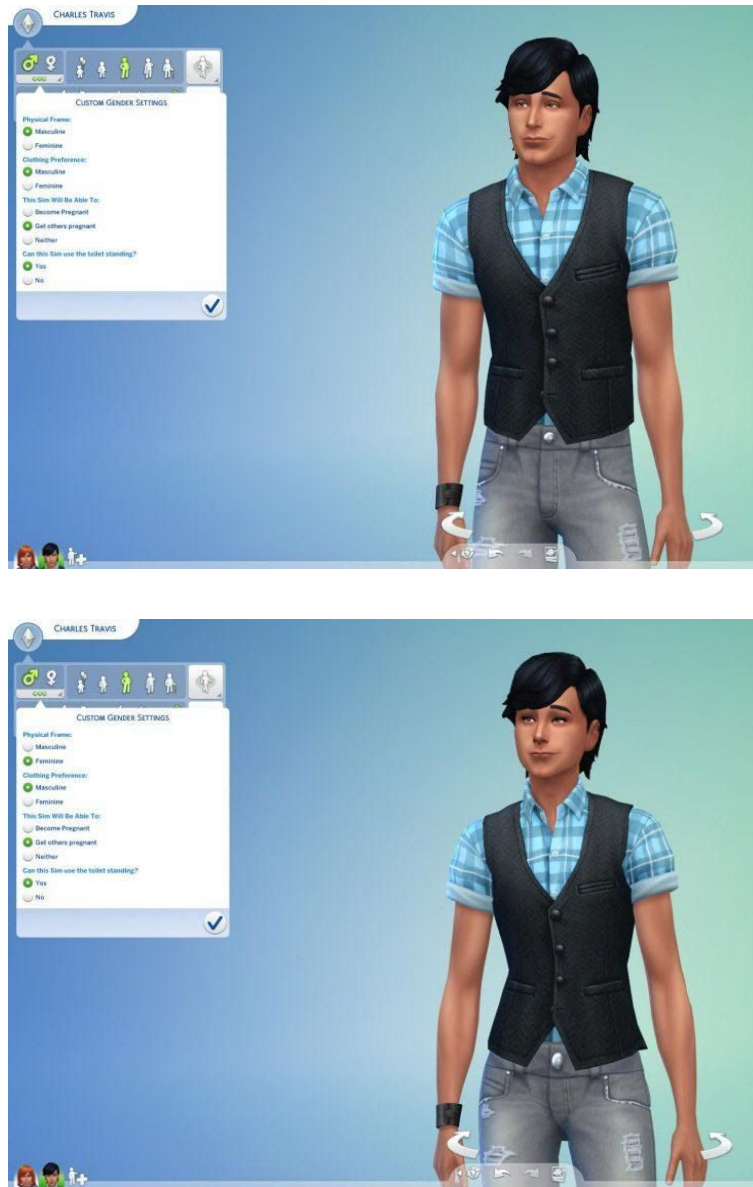
Sources:

(https://youtu.be/Qb2zfWnMxdM?list=PL_sYhAj0WXROF7vtyEO7WscywfAWn0Xbs accessed April 13, 2020; https://youtu.be/_chRj2C_jlM accessed April 13, 2020)

Nonetheless, I do not think that it is a coincidence that one of the strongest examples of transgender representation in AAA gaming comes from another series known for its non-realistic graphics: *The Sims*, and particularly the latest entry into the franchise, *The Sims 4* (Maxis, 2014). The human beings who populate the world of the Sims are not trying to pass as real. They speak in gibberish, their naked forms are hidden underneath pixels, and their faces, while having increased in detail throughout the franchise's history, lack the hyper-detailed skin of most other digital humans. By sacrificing graphical realism, *The Sims* games can instead put their development resources toward social and emotional realism—simulating interactions, orchestrating relationships, managing and maintaining emotional wellbeing.

Another key part of *The Sims* gameplay is creating these virtual humans. Players can shape their bodies and faces, pick their clothes and hair, and determine their personality and goals. And in 2016, players became able to customize their genders with unprecedented nuance (fig. 9 & 10). Though the default male/female choice remained, players could now inflect their

character's gender within four predetermined (mostly) binary choices: masculine/feminine frame, masculine/feminine clothing preference, this sim can become pregnant/get others pregnant/neither, and this sims pees standing up/sitting down.



Figures 9 & 10. Images of *The Sims 4* character creator, showing the gender customization settings and the effect that changing “physical frame” has on a character.

Source:

(<https://simscommunity.info/2016/06/02/the-sims-4-june-2016-update-gender-customization-overview/> accessed April 13, 2020)

The effectiveness of such a representational mode is complicated. Whatever transness arises from these choices, for example, is mechanical rather than textual. They are concerned with which body mesh to use or what animation to play when the character goes to the bathroom, but the lives of these “trans” sims are identical to the lives of their “cis” counterparts. One could easily frame this as utopic, removing the painful realities of the trans experience like being misgendered, experiencing dysphoria, or overcoming medical, legal, social, and financial hurdles in order to transition. However, besides also inadvertently removing the joys of being transgender like gender euphoria and trans community building, *The Sims 4* ultimately offers a limited model for imagining alternative realities for trans bodies. Its character creation is still fundamentally predicated on binaries which are more rigidly enforced than the game’s gestures toward inclusivity may suggest. Characters with masculine frames cannot have breasts, those with feminine frames cannot have facial hair, and, no matter what, players must ultimately elect one of the two binary genders for their character, even if they customize within it. The world of transness conceived of by *The Sims 4* has no room for the intersex, the androgynous, the genderless, genderqueer, and genderfucked.

The Sims 4 also reveals a fundamental problem with the mesh and, specifically, how it is stored by the game. Recall that a mesh is a set of vertices and triangles/quadrilaterals. The cube, as mentioned earlier, can be represented by eight vertices and twelve triangles, two for each face—but where are those vertices positioned? The cube in figure 2 is the unit cube, meaning that it is centered at the origin and has a length of 1, but what if we wanted the cube to be somewhere else? What if we wanted it to be larger, or smaller, or not a cube at all but a rectangular prism? You could make another set of vertices and triangles to represent this new

cube, but this means you are storing twice as much data when the second cube could instead be expressed as a transformation of the unit cube. What is important here is not the math which performs these transformations but rather the implications they have for the mesh. Significantly, there is only ever one copy of a given mesh stored in the data of a game or game engine. Copies of a mesh are represented not by a concrete set of vertices but by a transformation matrix. When it comes time to draw the variant mesh, it is the original/normal/unit mesh which is drawn, but with the transformations applied to move the vertices to the correct location. However, these altered vertices only exist as they are drawn by the computer. Visible to the human user but not the code.

One could imagine a version of *The Sims 4* character creator where problematic constraints are not imposed on the meshes. The game would present a single androgynous frame to the user for them to toy with at will, yet the continuing use of the mesh would limit this freedom of digital gender expression. The mesh is bound deeply to normalcy, defined by the false notion of a default state. It is a highly technical expression of Riki Wilchins' mapping of man/woman, white/black, straight/gay and other binaries of power onto another dichotomy: that of normal/derivative.²¹

I have other structural issues with the mesh. It is also a representational shortcut which assumes that interiority is unnecessary. The objects rendered with a mesh are superficial, represented only by the hollowed-out husk of their shape. The justification for this is technical, an appeal to optimization of resources and efficiency of process. There is, supposedly, no need to represent the insides of objects because they are not visible to players. Yet without interiority,

²¹ Riki Anne Wilchins, *Queer Theory, Gender Theory: an Instant Primer*, (Bronx, NY: Magnus Books, 2014), 46-47.

the mesh cannot feel things deeply within itself as transness is felt deeply, profoundly. A mesh is also finite, closed, not rigid but far from possessing the fluidity that marks many trans people's experiences with gender and presentation. Finally, a mesh cooperates within a game engine's larger systems. It behaves. It is far from an unruly, troublesome, or queer subject. It will not do for a queer game engine.

As I was considering this chapter, contemplating what alternatives I would propose in order to make the engine's 3D graphics queer, I found one (of likely infinite) solutions in the most unexpected of places: the Unity engine. On January 22, 2020, Unity released a short film called *The Heretic*, which was made with Unity 2019.3. The video's description explains that "[t]he film uses every aspect of Unity's High Definition Rendering Pipeline, features advanced effects created with the VFX Graph, and showcases Demo Team's first realistic digital human,"²² and it certainly lives up to its advertising as yet another entry in the graphical arms race. The visuals are stunning, rendering both a realistic (male) human and more magical figures and forms, such as the human's animal-like companion made out of shifting and rearranging black cords.

On my first watch of the film, I was prepared to resign it to another example of Boluk and LeMieux's hypertrophy and not engage with it further, but as I read more about it, I came across the most puzzling line from the Unity website.

At Unite Copenhagen, Technical Artist Adrian Lazar and Unity Evangelist Andy Touch explained how the team used the Visual Effect Graph to bring the Morgan character to life. By creating the simulation with GPU particles, the artists can

²² "The Heretic: a Short Film Created with Unity 2019.3," Unity, January 22, 2020, video, 7:43, <https://youtu.be/iQZobAhgayA>.

change the shape, gender, appearance, and behavior of the character and see their adjustments in real-time.²³

Though my first reaction to the quote was to protest that changing the physical appearance of a character did not amount to changing their gender, I was nonetheless intrigued by what those fluid particles had to offer by way of a trans aesthetic.

Morgan in the film is a godlike entity of immense stature and unknown power who morphs between two forms: one described by the production team as female and one described by them as male. The entity's two personas have different voices and, seemingly, personalities. In a presentation at a 2019 Unity conference, the demo team describes changes in Morgan's posture as they shift between personas, specifically signaling the aggressive stance of the male form.²⁴ The demo also illustrates exactly how this change between personas occurs. There are some changes in the mesh itself, particularly in the chest, hips, and shoulders, but, as the quote from the Unity website says, the most dramatic change in presentation occurs in the particles which compose Morgan's form (fig. 11 & 12). The change is even clearer in the conference video as the demo team slowly slides through the morph, highlighting the flicker of the scale-like particles as they flip from one form to another (fig. 13 & 14).

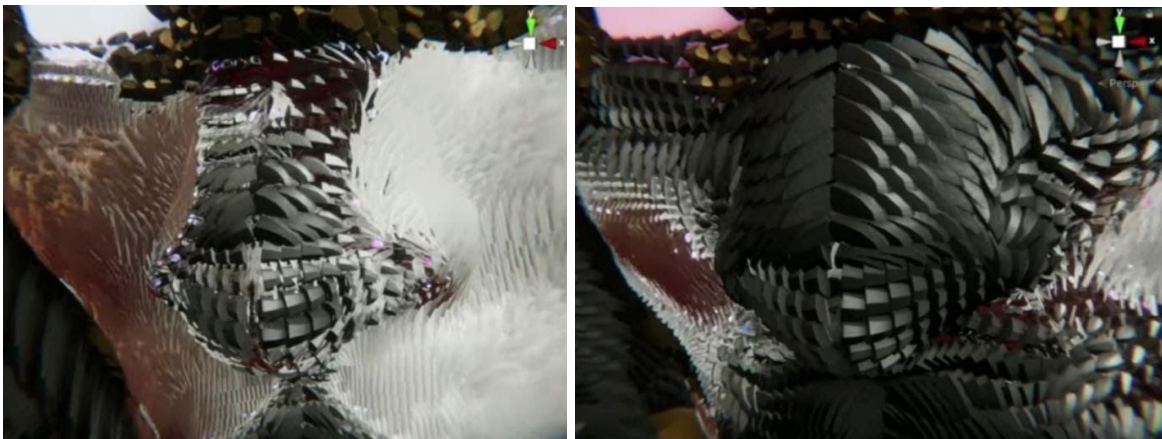
²³ "The Heretic, a Cinematic Real-Time Demo, Made in Unity," Unity, accessed April 12, 2020, https://unity3d.com/the-heretic?utm_source=youtube&utm_medium=social&utm_campaign=engine_global_general_promo_2020-01-22_heretic&utm_content=video.

²⁴ "Real-Time VFX Workflow in The Heretic - Unite Copenhagen," Unity, September 26, 2019, video, 44:33, <https://www.youtube.com/watch?v=uHD8OfSzCsg>.



Figures 11 & 12: Images of Morgan from *The Heretic*, in both their lighter/feminine form and darker/masculine form.

Source: (*The Heretic*)



Figures 13 & 14: Close-ups of Morgan's face during the transformation, clearly demonstrating the difference between their two forms.

Source: (<https://www.youtube.com/watch?v=uHD8OfSzCsg> accessed April 13, 2020)

So little context is given for Morgan's gender identity either within the film or in material produced by the demo team that, beyond analyzing these slips in presentation, it is difficult to

label their gender or, indeed, parse it through modern transgender terminology. They likely were not created to be read as a well-executed representation of the genderfluid identity, with this fluidity instead marking simultaneously their power and their non-humanness, in addition to being founded on fluctuating between gender essentialist understandings of male & female. By locating such fluidity in a malevolent entity, the film's creators have also associated gender fluidity and transness with evil.

Despite the problematics of Morgan, their shimmering, shifting body proposes a uniquely and undeniably trans application of particle systems. As a technique in computer graphics, particle systems were first introduced in a 1983 paper by William Reeves of Lucasfilm Ltd. In this paper, Reeves' proposes particle systems as a solution for rendering what he terms "fuzzy" objects—"phenomena such as clouds, smoke, water, and fire [which] are irregular, complex, and ill-defined."²⁵ In other words, particle systems were created in order to deal with those unruly subjects which resisted being understood through "classical surface-based representations" embodied nowadays by meshes. It is worth drawing specific attention to Reeves' use of the term *fuzzy* to describe such subjects. Far from being a technical word, fuzzy is both connotative and affective in that it speaks to a capacity "[t]o touch and to be touched"²⁶ as well as a particular quality of that touch. An object can be fuzzy as in blurry but also as in soft: a fuzzy blanket or a fuzzy animal. That Reeves reaches for such a word when describing the work of particle systems speak to their nature as fundamentally emotive and representational subjects, slippery beings

²⁵ William T. Reeves, "Particle Systems—A Technique for Modeling a Class of Fuzzy Objects," *ACM Transactions on Graphics* 2, no. 2 (April 1983): doi: 0730-1301/83/0400-0091, 91.

²⁶ Aubrey Anable, *Playing with Feelings: Video Games and Affect*, (Minneapolis: University of Minnesota Press, 2018), 38.

who are not so easily touched but nonetheless have a capacity to be touching in how they represent queer lives.

Reeves explains that “[a] particle system is a collection of many minute particles that together represent a fuzzy object,”²⁷ where particles are thought of as “a point in three-dimensional space.” He describes the operations of a particle system in five clearly delineated steps:

(1) new particles are generated into the system, (2) each new particle is assigned its individual attributes, (3) any particles that have existed within the system past their prescribed lifetime are extinguished, (4) the remaining particles are moved and transformed according to their attributes,²⁸ and finally (5) an image of the living particles is rendered in a frame buffer.²⁸

The clear separation of steps allows for easy customization and flexibility within a given step. For example, the particle system described by Reeves generates their particles, gives them a velocity, and updates their position as they fly away from their point of generation. However, the particle system used by Morgan does not generate new particles, and their movements are more tightly controlled. While Reeves only envisioned particles as being colored points of light, Morgan’s particles have shape and dimension. As Reeves acknowledges, “[t]he number of possible attribute control parameters and their variants is endless.”²⁹ Much like the plurality of queer and trans bodies, particles have an infinite capacity for variation.

Additionally, the objects simulated or modeled with particle systems are not static, but rather shift, ebb, and flow in form because their particles are constantly moving. Particle systems are volatile, and the fact that they are frequently predicated on randomness contributes to this inherent instability. Consider, as a simple example, a particle system that generates new particles

²⁷ Reeves, “Particle Systems,” 92.

²⁸ Reeves, 93.

²⁹ Reeves, 95.

every second. It could do so at a fixed rate, or, more commonly, at a rate that varies according to chance. Not only is a particle system's object constantly slipping between forms, displaying the same fluidity as Morgan's gender presentation, but it is doing so in a way that is unpredictable, structured by randomness and chaos rather than neatly ordered rules. The particle system is an unruly, queer subject.

Of course, such disobedience invites attempts at control, and this is the final resonance I read between particle systems and the queer/trans experience. Even in their earliest iterations, Reeves describes wrangling with these misbehaving particles, countering their random nature. He mentions, for example, controlling particle spawning based on how much space an object takes up on screen. As he says, "there is no need to generate 100,000 particles in an object that covers 4 pixels,"³⁰ and this controlling of excess represents an attempt to regulate the chaotic, odd nature of the particle system in the name of efficiency.

These attempts have continued long past 1983. Morgan is a particularly carefully controlled implementation of a particle system, with certain shapes and colors of shards needing to appear at specific places on their body and stay there. These particles do not fly around, nor do they spawn, live, and die at unexpected intervals. Indeed, the issue of the mesh returns with Morgan's body, as it is used here specifically as a mechanism for controlling the particle system. Morgan's mesh (fig. 15) is not rendered in the final scene, invisible to the cameras which output the finished video. Instead, the mesh is rendered to a separate camera specifically for Morgan, which stares down at them from above their head and records information about their body, like the positions and normals of its vertices. This data is then passed into Unity's Visual Effects

³⁰ Reeves, 94.

Graph, which uses it to render particles in their appropriate place. As a simple example, when the mesh moves, its new position is recorded by Morgan's special camera, which sends this information to the VFX graph, which changes the particles' positions accordingly. A more complicated example would be how, in the colored version of Morgan's mesh, their hands, eyes, nose, and mouth are bright red (fig. 16) in order to indicate that those areas are more detailed, and thus should have smaller particles.³¹



Figures 15 & 16. Two images of Morgan's mesh, one with vertex colors (right) and one without (left).

Source: (<https://www.youtube.com/watch?v=uHD8OfSzCsg> accessed April 13, 2020)

³¹ "Real-Time VFX Workflow in The Heretic - Unite Copenhagen," <https://www.youtube.com/watch?v=uHD8OfSzCsg>.

Yet amongst all these techniques, the ones I would like to focus on—the ones that have the most consequences for reading particles as queer/trans subjects—are those that are concerned with the production and termination of particles. With controlling their life and death. Such techniques include killing particles when they have floated off-screen or their transparency has dropped below a certain threshold,³² or simply giving particles an expiration date so that they do not endlessly accumulate on screen. These particles have lifetimes, and it is curious to read Reeves' description of that lifetime with control and power in mind: "[o]ver a period of time, particles are generated into a system, move and change from within that system, and die from that system."³³ One can easily draw parallels between this and Foucault's articulations of how systems of medical and legal power produced the homosexual identity in the first place.³⁴ As the particle system makes new particles, medical and legal systems make new homosexuals through their meticulous regimenting of sexuality, the careful tracking of "perversions," and the ascribing of labels and treatments to those they find lacking. As the particle system gives meaning to its particles through the rules defining their color, shape, movement, and other attributes, these other systems shape, move, and change the bodies of those caught within them, with examples including Alan Turing's chemical castration in order to physically correct his homosexual behavior. And as the particle system eventually extinguishes its subjects, so too must Foucault's homosexuals navigate the threat of death at the hands of the very system which created their identity in the first place, whether from direct homophobic violence or circumstances compounded over a lifetime of oppression. In short, particles express the process of being born, living, and dying within and because of a larger system, a system which gives life and shapes its

³² Reeves, "Particle Systems," 96.

³³ Reeves, "Particle Systems," 92.

³⁴ Michel Foucault, *The History of Sexuality Volume I* (1978), 101.

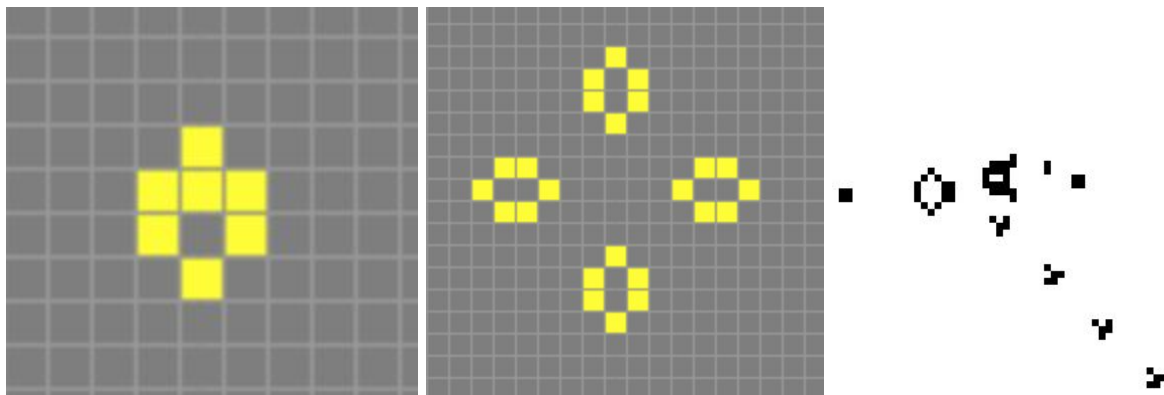
meaning as easily as it takes it away. To die from the system that creates your identity is a persistent threat in queer, trans, disabled, and racialized lives.

This more somber turn in representation significantly reconfigures the earlier discussion of the customizability of particles in the pursuit of pluralistic trans representation. If particle systems can also be read as representing not only the process of identity production but all the dangers and threats associated with it, advocating using such systems as a designer seems distinctly antithetical to the queer game engine's project. I do not want to suggest that this control should be seen as "worth it" because it could serve the needs of queer and trans designers.

The solution I have come to lies a little way back in videogame history. Thirteen years before Reeves first proposed 3D particles, John Horton Conway built a game predicated exclusively on 2D "particles" which were also entangled in the processes of living and dying—so much so that Conway called his game the Game of Life (1970). In the Game of Life, the player begins by filling in some cells in a grid. Then, each turn, they populate and depopulate cells based on a set of rules that very roughly approximate the social conditions of life. If a full cell is overcrowded or lonely, it dies, and empty cells become populated by the full cells near them. Although Conway's particles do not represent fuzzy objects like Reeves, they share a visual resemblance and certainly fit the description of spawning and dying according to the system within which they exist.

The value in turning to Conway lies in how the Game of Life positions its user—no longer a developer implementing particle systems in a game engine, but a player. In the absence of a mandated objective, players of the Game of Life have to impose their own goals in order to

motivate their play. One such goal is to populate the initial grid in such a way that particles live longer, rather than eventually dying and ending the game. And, because this is a fairly easy goal to achieve, other users have turned to looking for initial configurations which produce aesthetically and mathematically pleasing results. Some, like the Small Exploder (fig. 17 & 18) end up producing perfectly stable patterns. Another interesting configuration is known as the Gosper glider gun (fig. 19), discovered by Bill Gosper. This pattern has some unique properties, including that it is not only generating infinitely but also requires that the grid on which it was initially built infinitely expand.³⁵ Those smaller cells trailing off to the right are called “gliders,” and they are always being produced by the interaction of the two larger shapes at the top (the “gun”). Those gliders are continuing on into infinity, always being made and always moving forward.



Figures 15, 16, & 17. Various patterns from *The Game of Life*. Figure 15 is the starting Small Exploder pattern, and Figure 16 is the stable pattern that arises many iterations after Figure 15. Figure 17 is one iteration of the Gosper glider gun, after it has begun to produce infinite gliders.

Sources: (<https://bitstorm.org/gameoflife/> accessed April 13, 2020;
https://en.wikipedia.org/wiki/Conway%27s_Game_of_Life accessed April 13, 2020)

³⁵ “Does John Conway hate his Game of Life?” Numberphile, March 3, 2014, video, 6:53,
<https://www.youtube.com/watch?v=E8kUJL04ELA>.

What relevance does Conway's Game of Life have for particle systems and the queer game engine? I believe that it points to a different way to conceptualize the relationship the user of the queer game engine might have with particles—a relationship founded not on control but on play and care. Both these values can be seen in the Game of Life, as players are invited to play within Conway's systems and rules but also charged with the responsibility of making particles live not only longer but more beautifully. What if the queer game engine gave its designer the same task? What if it emphasized interactions with the particle system, like establishing its rules and boundaries, as part of a relationship which the designer is responsible for negotiating, rather than as an instance of power and demanding submission from the computer? What if the designer saw themselves as responsible for the care and longevity of the particle system? What if, as is the case with Conway, this responsibility was mediated via play? There is, indeed, something already playful about building particle systems in an engine like Unity. While possibly initially daunting, once a user is familiar with the various parameters they can change, they are free to experiment wildly. And if the technical dauntingness was removed? If these editable parameters were understood not as points of control but of playful interaction? If those parameters were designed around a commitment to respecting the inherent randomness of the particle system rather than reigning it in?

To offer a specific instance of what this more consensual relationship with particles could look like, one might imagine that the queer game engine would do away with the mesh entirely, replacing it instead not with particles as exterior decoration but rather as the contents of a volume. This might look like the designer specifying the shape they want the particles to fill but not how to fill it, leaving issues of density, color, size, velocity, lifetime, and other parameters in

different hands. This proposal is also in keeping with the original conception of particle systems laid out by Reeves, in which “an object is represented not by a set of primitive surface elements ... that define its boundary, but as clouds of primitive particles that define its volume.”³⁶ Finally, representing bodies not as tightly regulated boundaries but rather porous volumes with interiorities opens up new possibilities for queer expression within 3D digital space.

However, this implementation would also cause performance problems. For their representational flaws illustrated thus far, well-made meshes remain efficient. A cube could, for example, be represented by its 8 vertices and 6 quadrilaterals drawn between them. Drawing said quadrilaterals is a trivial task and has been for computers since at least the 1970s.³⁷ By contrast, depending on their size, millions of particles could be required to form the same cube, necessitating exponentially more draw calls which may not be individually computationally expensive but stack to cause severe lags between frames. This exact concern with performance is why Unity’s Visual Effects Graph can only be built to target certain machines— specifically, platforms that support Compute Shaders, graphics programs which run on the graphics card rather than the CPU.³⁸ Computations done on the GPU are faster than those done on the CPU, and it is this speed which allows the VFX graph to support millions of particles, compared to Unity’s standard particle system, which can only handle thousands.³⁹ Were a queer game engine to be similarly restrictive in the platforms which can use it and which it can produce for, it would

³⁶ Reeves, “Particle Systems,” 91.

³⁷ Jesse Dunietz, “The Most Important Object In Computer Graphics History Is This Teapot - Facts So Romantic,” *Nautilus*, last modified February 29, 2016, <http://nautil.us/blog/the-most-important-object-in-computer-graphics-history-is-this-teapot>.

³⁸ Unity Technologies, “Compute Shaders,” *Unity*, accessed April 12, 2020, <https://docs.unity3d.com/Manual/class-ComputeShader.html>.

³⁹ “Fast particle rendering with VFX Graph - Unity at GDC 2019,” Unity, April 11, 2019, video, 23:39, <https://youtu.be/oWVHc0M2RjI>.

be antithetical to its larger goal of deconstructing the technological exclusivity of game design. Yet were it to try to render every object with particles, it would be inefficient—likely to the point of being difficult, or impossible, to use.

As I mentioned in the introduction to this chapter, at this moment in its life, the queer game engine does not yet need to contend with the reality of hardware, or with the demands it makes for the cleanest, most efficient solutions. Yet rather than use this as an excuse to not grapple with the possible exclusivity of the queer game engine as I have proposed it thus far, I would like to contend, if briefly, with the issue of implementation.

And, again, I am going to draw from Conway to do it. In an interview with the YouTube channel Numberphile, Conway demonstrates the Game of Life with a piece of paper, a Sharpie, and almonds. One of the lowest possible levels of game design. I would like to propose that the queer game engine should also be able to work on this level. In fact, it should be able to work at whatever level its user is at. The trick, of course, is being able to scale the queer game engine without losing functionality but then, maybe, the queer game engine is not one singular engine, but a collection—a plurality—of them, each designed with different technological and economic needs in mind. Or maybe it is a collection of parts which its user can arrange in whatever fashion they desire, pulling from pencil and paper graphics but high-tech physics, mixing and matching component parts and reconfiguring their relationships until the user has the best tool for their personal expression.

I do not know what a concrete implementation of my queer particle system would look like, whether that implementation is for the most powerful hardware that currently exists or for the sheet of paper on the desk next to me where I sketched the outline of this chapter. But, as a

proposal like the QueerOS, as an imagined engine like Babbage's Analytical or Difference Engines, the queer game engine is allowed to be messy, impractical, and not yet fully formed. So, in suggesting an engine whose graphics are entirely particle- rather than mesh-based, my intention is not to also put forth a detailed plan for implementation, but rather to prod, however gently, at the normative structures which the queer game engine may one day be able to more strongly challenge.

Physics

The movement of virtual bodies in the game space is generally controlled by one of two systems: animation or physics. In the case of animation, “[a] character is imbued with the illusion of motion by arranging the character’s body into a sequence of discrete, still *poses* and then displaying those poses in rapid succession.”⁴⁰ Because the character’s animator creates each of these poses, they exercise tight control over what the final animation sequence will look like. While potentially advantageous for this predictability, this system is more labor-intensive than using a physics system. By using a game engine’s native physics system, developers can quickly simulate physically correct movement within their games. Unity’s physics system, for example, handles issues of gravity, velocity, acceleration, applying forces, resolving collisions, and much more without any effort on the part of the user.⁴¹

This is accomplished through adding Unity’s Rigidbody component to a game object. Components in Unity are small, discrete bits of functionality which users can easily attach to game objects in order to change their behaviors. The MeshFilter component, for example, gives

⁴⁰ Gregory, *Game Engine Architecture*, 499.

⁴¹ Unity Technologies, “Rigidbody,” *Unity*, accessed April 12, 2020, <https://docs.unity3d.com/ScriptReference/Rigidbody.html>.

an object the ability to be represented by a mesh, and the AudioSource component lets game objects make noise. The Rigidbody component specifies that the object it is associated with should obey the laws of rigid body dynamics. Dynamics here is a subset of mechanics which refers to “how [objects] move over time,”⁴² with rigid body specifying the type of objects—namely, those which “are perfectly solid and cannot be deformed.”⁴³ In other words, rigid body dynamics is used to represent and simulate hardness within moving digital objects.

These hard bodies have been and continue to be ubiquitous within 3D games because assuming an object is rigid makes it easier to perform physics calculations.⁴⁴ However, a number of queer and queer-adjacent analyses have reckoned with the problems posed by this normativity, and they frequently center around the ragdoll as an alternative to the hardness which rigid body dynamics simulate. The ragdoll “is a physical simulation of the way a human body might move when it is dead or unconscious [and it is] created by linking together a collection of rigid bodies, one for each semi-rigid part of the body,”⁴⁵ and many analyses imbue it with queer potential. Amanda Phillips, for example, understands the ragdoll as the “unmaking of the hard, controlled masculine body” both through its association with death and its “unpredictable, accidental [sexually suggestive] bodily positions.”⁴⁶ For them, ragdolls and floppy dicks “offer us another way to think about the erotic potential of soft masculinities.”⁴⁷

Yet at the level of the game engine, this analysis is complicated by the fact that rag dolls are still composed of rigid bodies. I do not point this out to challenge or dismiss the work of

⁴² Gregory, *Game Engine Architecture*, 630.

⁴³ Gregory, 631.

⁴⁴ Gregory, 631.

⁴⁵ Gregory, 660.

⁴⁶ Amanda Phillips, “dicks dicks dicks: Hardness and Flaccidity in (Virtual) Masculinity,” *Flow Journal*, November 27, 2017, <https://www.flowjournal.org/2017/11/dicks-dicks-dicks/>.

⁴⁷ Phillips, <https://www.flowjournal.org/2017/11/dicks-dicks-dicks/>.

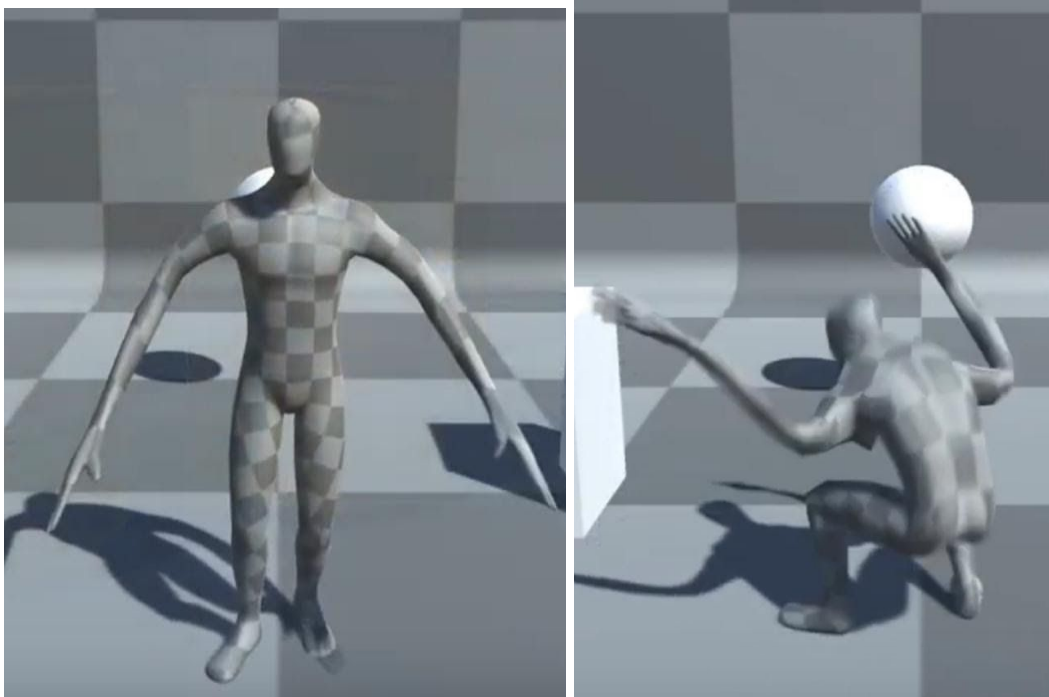
Phillips or others in offering insights into how different, queer bodies move through the game space. Indeed, I also read ragdolls through softness— specifically, a kind of “soft” control which they enable. As a matter of player-directed movement, ragdolls, in those games which divorce them from their historical association with death, make for notoriously unruly bodies. For example, in *Octodad: Dadliest Catch* (Young Horses, 2014),⁴⁸ players must pilot the slippery body of a humanoid octopus, but they must “move each of Octodad’s arms and legs separately, making his body difficult to direct with precision and impossible to fully master.”⁴⁹ As a result, actions which would be effortless on the part of the player in another game, like walking forward or picking up an object, become tedious feats which mirror the very awkwardness of the body which the player is inhabiting. Players are granted a partial mastery over Octodad, an incomplete, selective control of his tentacled ragdoll form. Softness here refers not only to how Octodad’s body flops but to the lax power which the player exercises over that flopping.

Unfortunately, as a possibility for the queer game engine, the ragdoll does not have much to offer because it is a specific implementation of rigid bodies, rather than an alternative to them. A ragdoll is a collection of rigid bodies which the player has lost control of. This points to another problem with ragdolls and the game engine, namely that they are highly contextual. Their presence in a game engine as the primary, or only, physics system necessarily assumes that the bodies being represented in a game are humanoid, and either killable or controlled via a schema that is hyper specific to the game itself. Such specificity would severely limit a queer game engine’s ability to serve as a platform for generalized game creation.

⁴⁸ Ian Bryce Jones, “Do the Locomotion: Obstinate Avatars, Dehiscent Performances, and the Rise of the Comedic Video Game,” *The Velvet Light Trap* 77 (2016): doi: 10.7560/vlt7706, 87.

⁴⁹ Bo Ruberg, *Video Games Have Always Been Queer* (New York: New York University Press, 2019), 91.

However, ragdolls are not the only possibility for representing softness in games or the game engine. Another option is soft body dynamics. Dynamics here continues to refer to how bodies move, but with said bodies being “soft” instead of “rigid”. There is not a precise definition of what makes a body soft in a physics engine— like Man/Woman in the analysis of Riki Wilchins, “one term is always the defining one while the other is derivative,”⁵⁰ and soft is the derivative here. Thus, the term includes a number of disparate representations of real world phenomena, such as liquids, gases, clothes, hair, rope, yarn, plastic, and things that, broadly speaking, jiggle, from depictions of jellos to the “softbodyfied” arms represented in a promotional video for the Obi Softbody asset for Unity (fig. 18 & 19).



Figures 18 & 19. Images of a humanoid model. As the model jumps, their arms bend, flop, and fold because they use soft body dynamics instead of rigid.

Source: (<https://youtu.be/-qaQETsXeWA> accessed April 13, 2020)

⁵⁰ Wilchins, *Queer Theory, Gender Theory*, 46.

What all of the above examples have in common, and what could be said to define soft bodies as a category, is that, unlike the rigid body models described earlier, each vertex of the model can move independently in response to an applied force.⁵¹ However, it is difficult to generalize about how these vertices move because it is highly dependent on the object being modeled. Fluids cannot use the same system as ropes which cannot use the same system as hair which cannot use the same system as jiggly objects. By contrast, any hard, solid object can be reasonably well-represented by a rigid body, with no distinction made as to *what* object is being modeled.

Considering soft bodies at the level of the game engine, this specificity causes problems. A game engine cannot offer a singular, general algorithm for soft body dynamics the way that it can for rigid bodies, yet, as mentioned with regards to the ragdoll, the promise of the game engine is the general. The game engine is founded on the separation between what is common to all games and what is specific to a developer's implementation. It makes (or claims to make) no assumptions about use, and yet to implement soft bodies within the game engine would require making assumptions about the world being built within it. A world of fluids, fabrics, squishy things.

Yet does the presence of rigid bodies within game engines not make similar assumptions? Implicit in their inclusion are a wealth of presumptions about the design goals of the engine's user, even if such presumptions are hidden by rigid body dynamics' status as the default for physics simulations. Through their ubiquity, the rigid body is rendered neutral, seeming to have no consequences on the game engine's usage. The soft body is cast as the "other" which must be

⁵¹ Jaruwan Mesit, "Modeling and Simulation of Soft Bodies" (doctoral thesis, University of Central Florida, 2010), 1, http://etd.fcla.edu/CF/CFE0003477/Jaruwan_Mesit_201012_PhD.pdf.

added to the existing system on an individual basis. However, when thought of as a technique that could also be included by default, the soft body reveals that the rigid body similarly breaches the engine/game divide by presenting hardness as the only way to exist in the world. The soft body undoes the naturalization of rigidity.

In just this brief consideration of putting soft body dynamics into an engine, the soft body has already begun to do queer work. It may simply be enough to suggest that the queer game engine include soft body dynamics natively, rather than requiring users to depend on third party software, such as Blender, Houdini, or Maya, or paying for assets developed by other users. In the latter case, implementation within the engine would also greatly help with financial accessibility, as, for example, Obi's Fluid and Softbody assets each cost upwards of \$50.⁵² Yet, having the space to do so, I would like to linger for a moment on the queer potential specifically of jiggly objects. Of the types of soft bodies discussed so far, jiggly objects are uniquely well-positioned for a queer analysis, because rather than referring to a type of object, they reference a *behavior*. This behavior can be applied to any mesh, regardless of the subject it represents—once an engine has support for jiggling, any object can become soft and squishy. As queer itself is an adjective which can be applied to a wide variety of subjects, it is easy to explore affinities between the two and between the subjects which they produce.

And these affinities are numerous. Jiggly objects must, for example, contend with charges of lewdness, inappropriateness, and sexual excess arising from their association with the modeling of breasts. Soft body dynamics are, after all, integral to “breast/boob physics,” the term for producing breasts in a video game that move rather than being another stiff part of a

⁵² “Obi,” *Virtual Method Studios*, accessed April 12, 2020, <http://obi.virtualmethodstudio.com/index.html>.

character's body.⁵³ Fighting games are particularly notorious for using breast physics, with perhaps the most (in)famous of them all being the *Dead or Alive* series (Team Ninja, 1996-2019). *Dead or Alive 5: Last Round* (2015) was even developed on an engine designed with jiggling in mind—hence the name, Yawaraka Engine (やわらかエンジン) or *Softness* Engine.⁵⁴ As queer people must navigate accusations of perversion and sexual deviancy, so too must these particular soft bodies contend with their association with a particularly misogynistic construction of desirability.

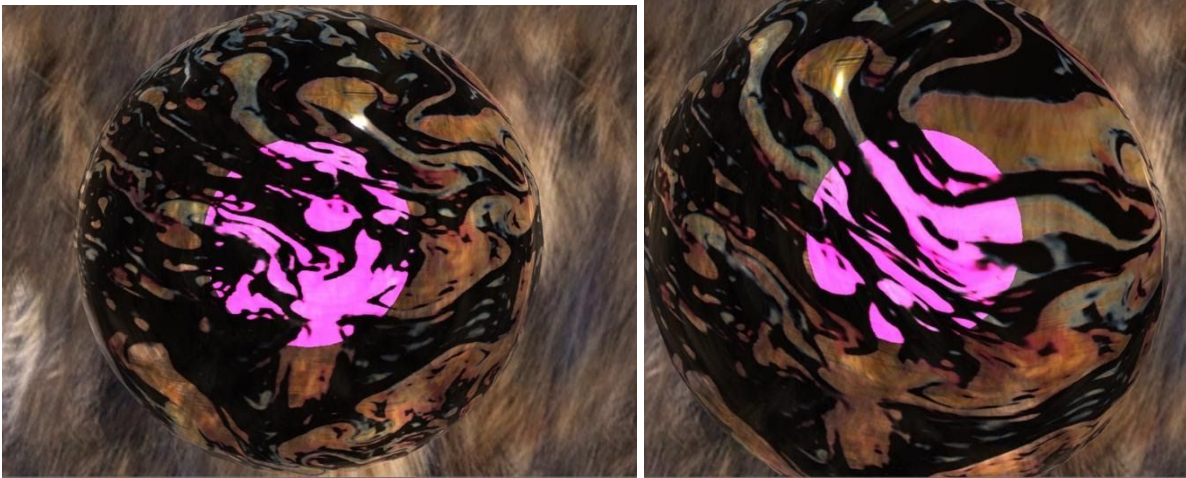
Yet I do not believe that the solution is to excise the sexiness of the soft, jiggly body, a move which would be disconcertingly similar to the demand that queer people purify their sexuality in order to be accepted by and assimilate into hetero- and cisnormative society. Indeed, this association can be quite productive. For example, soft body dynamics are also (though far more rarely) used to model penises, most famously displayed in *Conan Exiles* (Funcom, 2018). The game's character creator comes with an endowment slider, allowing players to adjust the size of their virtual penis which flops between their legs as they run, squat, jump, and move throughout the game world.⁵⁵ Soft body dynamics thus create the possibility for alternative constructions of digital masculinities, in contrast to the ubiquity of hardness and rigidity which Amanda Phillips criticizes.⁵⁶

⁵³ Patricia Hernandez, "How Video Game Breasts Are Made (And Why They Can Go Wrong)," *Kotaku*, last modified February 24, 2015, <https://kotaku.com/how-video-game-breasts-are-made-and-why-they-can-go-so-1687753475>.

⁵⁴ Toshi Nakamura, "You Can Adjust 'Bounce' in the New Dead or Alive," *Kotaku*, last modified October 23, 2014, <https://kotaku.com/you-can-adjust-bounce-in-the-new-dead-or-alive-1649708514>.

⁵⁵ Christopher Livingston, "Conan Exiles Has an 'Endowment' Slider and Genital Physics," *PC Gamer*, January 31, 2017, <https://www.pcgamer.com/conan-exiles-dick-slider/>.

⁵⁶ Phillips, <https://www.flowjournal.org/2017/11/dicks-dicks-dicks/>.



Figures 20 & 21. Two images of a sphere, one before being poked, and one after, demonstrating the effects of the soft body dynamics.

Source: (Screenshots of a personal Unity project)

Other relationships between the soft body and queerness include the way in which jiggly objects could be said to pass as rigid bodies (fig. 20). It is only when a user or another object gets closer to it, intimate even, that its soft nature is revealed, as its mesh yields to touch (fig. 21).

One could read this as a moment of being outed as the body reveals its incongruities. This could turn this touch into a violent and invasive probing—or it could reconfigure it into the complex negotiations between those parts of the body which can be made to pass and those which cannot. Rae Spoon, for example, explains that “[their voice] gave [them] away immediately if [they were] being read as a man at all in a situation,”⁵⁷ but they could not change it (or any other part of their body, for that matter) via hormone replacement therapy because it would damage their career as a singer.⁵⁸ Similarly, the jiggly object might be said to pass in some ways (appearance) but not others (behavior).

Yet while trans people may choose to consciously change their behavior in order to pass better, be these modifications innocuous (crossing one’s legs differently) or dangerous

⁵⁷ Rae Spoon and Ivan E. Coyote, *Gender Failure* (Vancouver, BC: Arsenal Pulp Press, 2014), EPUB, 69.

⁵⁸ Spoon and Coyote, *Gender Failure*, 68.

(replicating misogyny in order to “be a man”⁵⁹), this intentionality, for me, distinguishes the type of passing which the soft body performs, in that I do not think that it is deliberate. Jiggly bodies, rather than striving toward the platonic ideal of another form, pass effortlessly, almost accidentally. In this way, the soft body can more closely be identified with the femme lesbian identity, or queer femininity more broadly. The phenomenon of femme invisibility has been contended with by a number of authors in academia⁶⁰ and online.⁶¹ The term refers to the fact that a femme’s queerness is not immediately evident in their presentation, particularly in contrast to the visible gender nonconformity (and thus queerness) of butches. This leads to an assumption of heterosexuality from both straight and queer people, and though it affords a certain degree of privilege, it is also a source of frustration for most femmes.⁶² Being read as straight is undesirable.

I believe that the jiggly object is imbued with a certain degree of queer femininity for the way in which it passes, but not of its own volition. The soft body wants to be read as soft, hence its yielding nature, its welcoming response to penetration—itsself a contrast (or perhaps complement) to the stone butch identity. As stone butchness is constructed around an aversion to penetration during sex, mentioning it here reveals some of the limits of the jiggly object’s queerness. As Heather Love has noted, “[u]ntouchability runs deep in queer experience,”⁶³ yet this is a form of queerness which the soft body does not contain. I do not want to suggest that

⁵⁹ Spoon and Coyote, *Gender Failure*, 45.

⁶⁰ Jack Halberstam “Queer Art of Failure,” Kara Keeling “The Witch’s Flight”

⁶¹ The first few pages that showed up for me when I googled “femme invisibility”:

https://www.huffpost.com/entry/femme-lesbians_b_1237648,

<https://thelesbianblog.com/blog-posts/femme-invisibility>,

<https://www.dailydot.com/irl/femme-invisibility-queer-community/>,

<https://www.urbandictionary.com/define.php?term=femme%20invisibility>.

⁶² Melissa A. Fabello, “Femme Invisibility: On Passing Right by Your People and Not Being Recognized,” *Everyday Feminism*, October 16, 2014, <https://everydayfeminism.com/2014/03/femme-invisibility/>.

⁶³ Heather Love, *Feeling Backward: Loss and the Politics of Queer History*, (Harvard University Press, 2007), 40.

these objects are universally representative, nor do I want to read the butch identity as associated with the rigid body dynamics I have spent so long critiquing. If the rigid body is masculine and the soft body feminine, then the butch, that figure which performs masculinity but also reveals its impossibility,⁶⁴ has yet to appear in the game engine.

Still, I would like to briefly touch on the jiggly object's relation to masculinity in the form of fatphobia. There is a gendering to fat, already alluded to in Amanda Phillips' arguments—as they say, “[t]oo much jiggle spoils the man” because it flies in defiance of the demand that masculine bodies be hard, toned, ripped, muscular.⁶⁵ Anything but soft and fat. Fatphobia against masculine bodies has expressed itself in video games through a particularly concerning trope involving fat male villains such as Wario from the *Mario* franchise (Nintendo, 1981-present), Doctor Robotnik from the *Sonic the Hedgehog* franchise (Sega, 1991-present), Pokey Minch from the *Mother* series (Nintendo, 1989-2015), King Dedede from the *Kirby* franchise (Nintendo, 1992-present), Scott Shelby from *Heavy Rain* (Quantic Dream, 2010), and many more.⁶⁶ It is important to note that soft body dynamics are not necessarily used for these bodies, or those of any other fat video game character, evil or otherwise. Body fat in games is typically only represented visually, through a larger mesh. However, soft body dynamics do offer a possibility for these masculine bodies, a pathway toward the softer masculinities which Phillips imagines without simultaneously being rendered as evil and/or undesirable.

Indeed, the softness found in these jiggly objects is quite pleasurable. There is something immensely gratifying about prodding these objects. There is joy in pushing their vertices as far

⁶⁴ Jack Halberstam, *The Queer Art of Failure*, (Durham NC: Duke University Press, 2011), 100.

⁶⁵ Phillips, “dicks, dicks dicks.”

⁶⁶ Benny Disco, “Let's Look at Tropes Vs. Fat Guys in Videogames,” *Destructoid*, October 28, 2013, <https://www.destructoid.com/let-s-look-at-tropes-vs-fat-guys-in-videogames-264375.phtml>.

back as they will go, in holding them at their tensest points, watching the rest of the form tremble with every slight movement of the mouse. And it is just as satisfying to release them and watch the shape bounce back, elastic, rippling until it settles into its original form. This pleasure and desirability which the soft body evokes here suggests a resonance with cuteness within queer theory/analysis as “a productive category for expressing queer sexual value.”⁶⁷ While I think “cuteness” as a descriptor is too dependent on aesthetic qualities to suggest that using soft body dynamics inherently makes an object cute, the attractiveness of said body is similar to the attractiveness of the cute body. In much the same way that cuteness can be “a way of identifying someone as an erotic commodity without gendering them and without making reference to existing beauty standards,”⁶⁸ the jiggly object is erotically charged and actively challenges existing beauty standards, particularly for masculine bodies, by embracing not only softness but penetration as its main mode of intimacy.

Not all intimacy is queer, of course, and, as noted in the stone butch aside, queer intimacy can reject being touched, but this kind of intimacy is integral to queer pleasure. Sara Ahmed argues that “pleasure orients bodies toward other bodies in a way that impresses upon the surface, and creates surface tensions.”⁶⁹ In the case of the jiggly object, these surface tensions are quite literal. Of course, as Ahmed notes, “pleasure is not simply about any body opening up to any body ... When bodies touch and give pleasure to bodies that have been barred from contact, then those bodies are reshaped.”⁷⁰ One could then read soft body dynamics as opening the rigid body up to the body of the user, permitting a kind of contact which was once denied.

⁶⁷ Teddy Pozo, “Queer Games After Empathy: Feminism and Haptic Game Design Aesthetics from Consent to Cuteness to the Radically Soft,” *Games Studies* 18, no. 3 (2018): <http://gamestudies.org/1803/articles/pozo>, n.p.

⁶⁸ Pozo, n.p.

⁶⁹ Sara Ahmed, *The Cultural Politics of Emotion*, (Edinburgh: Edinburgh University Press, 2015), 165.

⁷⁰ Ahmed, 165.

Ahmed's use of the term "reshaping" leads me to the final expression of queerness within jiggly objects. These bodies are indestructible. The technical term for their yielding is 'mesh deformation', but this represents a temporary state, a momentary response to touches both violent and loving, a commentary on the queer body's resilience. Even in the case of a deformable mesh, which remains permanently reshaped by contact, the soft body cannot be destroyed. Push it in on itself, and it may become smaller, its vertices crowding together, triangles and quad narrowing, but they do not vanish. Its shape changes, but its existence is not erased.

These jiggly objects thus navigate with surprising deftness the tension between softness as a queer affective mode and hardness as a means of survival. micha cárdenas, for example, has proposed an app called #stronger where trans and gender nonconforming people can document their "affective experience of gaining strength" and network with other trans exercisers.⁷¹ cárdenas' project comes out of a legitimate concern with survivability and "encouraging more trans people to prioritize practices that increase their own fitness and wellness,"⁷² yet she does not take the time to comment upon those for whom a hard, strong body is unachievable or undesirable, such as for fat and disabled members of the community.

Nor does she mention how exercise can be an unhealthy behavior if it is motivated not by self-love but self-hatred, as Rae Spoon notes: "When it came to my gender and body ... I had a block. I thought the way my body had arranged itself was not the way a man's body looks. Even if I could be positive about fat, I thought that my fat was in the wrong spots, and there was little I

⁷¹ Mica Cárdenas, "Dark Shimmers: The Rhythm of Necropolitical Affect in Digital Media," in *Trap Door: Trans Cultural Production and the Politics of Visibility* ed. Reina Gossett, Eric A. Stanley, and Johanna Burton (Cambridge, MA: MIT Press, 2017), 175.

⁷² Cárdenas, 175.

could do about it.”⁷³ They go on to describe their exercise as an attempt to gain control over their body before eventually accepting, later in life, that their body was not the problem: “I refuse to accept that the way my body looks should determine my gender. I refuse to accept that fat is gendered in any arrangement on my body or anyone else’s. I refuse to think of fat as unhealthy.”

⁷⁴ For Spoon, and for many other trans people, a stronger, harder body is not only an undesirable goal but an unhealthy one as well.

However, in the jiggly object, we can locate the indestructability for which cárdenas and others yearn alongside a rejection of hardness and a commitment to softness. Attractiveness alongside fatness. Invisible queer feminities and soft queer masculinities. A plethora of representational resonances with queerness and transness.

Like replacing every mesh with particles, replacing every rigid body with a soft one is easier to propose than it is to implement. Unlike in rigid body dynamics, where the entire object moves in response to a force, soft body dynamics must contend with how a force affects individual surface points. This complicates physics calculations, collision detection, and runtime performance.⁷⁵ The queer game engine is also revealing itself to be riddled with contradictions, as soft body dynamics depend on the mesh, which has already been substituted for the particle system.

Yet I am content to let this contradiction and the impossibility it suggests be. The queer game engine remains unbound by the practical considerations of implementation. It does not have to be possible, or cohere into a single entity free of friction. Indeed, this friction is valuable.

⁷³ Spoon and Coyote, *Gender Failure*, 69.

⁷⁴ Spoon and Coyote, *Gender Failure*, 70.

⁷⁵ For more information about the math: http://etd.fcla.edu/CF/CFE0003477/Jaruwan_Mesit_201012_PhD.pdf.

It is evidence of the engine's interconnectedness, of the contact between those neat, tidy boxes.

What other productive tensions might the queer game engine reveal in the years to come?

Conclusion

So, that's it. The first pass at the queer game engine. As I mentioned in the introduction to Chapter 3, there are a lot of directions to go from here, and I would like to briefly acknowledge two such directions which I wanted to take up in this paper but couldn't because of the limited time I had to work on it.

The first is the interface. The interface seems so crucial to any software analysis that I do regret not being able to include it here. It is such a complex and rich site for queer analysis—for talking about control and intimacy, and for addressing issues which this paper circles around but ultimately keeps in the periphery, such as the blackbox, accessibility, and use. In order to talk about the queer game engine's potential to reconfigure the ethics of the game design process (a potential I believe in strongly), we have to talk about interface. We also can't talk about how the queer game engine could empower queer and trans designers without talking about it first. Nor can we talk about the tensions between the game engine's positioning as a tool of (re)production and queerness' positioning as antithetical to (re)production in works like *The Queer Art of Failure* (Halberstam, 2011). The interface mediates the relationship between the user and the queer game engine, and without it, it is far harder to talk about those users.

The second is scripting. As one of the main technological barriers to game development, scripting is another crucial site for issues of accessibility and control. It also has a particularly privileged role in the design process, a revelation I had during TreaAndrea Russworm's guest lecture for the second offering of MCM 1204F: Critical Video Game Studies. As Russworm was describing the central role of rules in distinguishing games from other mediums and conveying their meaning, I realized, for the first time, that scripting is how designers author new rules for

their game, how they implement new mechanics. Given how important scripting is in the development process, it now seems like another glaring omission, especially given how easily it is taken for granted by people who already know how to code, which I am absolutely guilty of doing. I want to know what happens when scripting is dislodged from this privileged position. I want to make its limitations obvious, especially to those of us who have never grappled with those limitations before because we have been conditioned by our coding backgrounds to play within them. To not question them.

These are just two of the possibly infinite queer interventions waiting to be made into the game engine, but I mention these specifically because their omissions seem particularly glaring, and because, more so than other engine components, I already have things I want to say about them. Not fully formed critiques, but the first sketches of a plan for future work.

And I fully intend to continue working on the queer game engine. There is so much more to be said about it, and so much more that I want to contribute to it, especially now that the hardest part is over. I've proposed the queer game engine. I've explained why I think it's necessary and established its theoretical allegiances. Now, I get to watch it grow. I can explore whatever offshoot or point of intersection interests me, trusting that this initial defense of its existence will offer a strong enough foundation for whatever I want to do next.

And I hope that that "next" is collaborative. I do not yet have specific plans for the future of the queer game engine, but it would mean the absolute world to me to take up this project alongside others, or to see others take it up on their own. I have no pretensions of being able to design the queer game engine by myself, and I would love to see others contribute to it. I hope

that this text can offer them enough of a jumping-off point to begin their own work. I hope some of them will let me work with them.

I have some more specific desires for the queer game engine as well. This fall, I will begin a masters program in informatics at the University of California, Irvine. The program is structured around the study of the relationship between people and computing—a relationship which is currently unexplored within the queer game engine. I currently plan on using my thesis there to discuss that relationship and develop the queer game engine in terms of interface and interaction. Of course, I leave myself open to whatever changes may occur in my interests as I transition to grad school.

Above all else, I am excited to see what the future holds, both for myself and this project. As it currently exists, the queer game engine, like myself, is the culmination of the last four years at Brown, and I hope that it will leave Brown with me. I hope that, side-by-side, we will be exposed to the wider world of academia, and all the new voices, ideas, and perspectives that will come with it.

Quite simply, I hope we will continue to grow, together.

Bibliography

“2019 Essential Facts About the Computer and Video Game Industry.” Entertainment Software Association. Accessed November 29, 2019.

<https://www.theesa.com/esa-research/2019-essential-facts-about-the-computer-and-video-game-industry/>.

“The 3D Engine Wars.” *Maximum PC*. November 1998. 43.

“3D Gaming Technology.” *Maximum PC*. November 2004. 49-66.

Aarseth, Espen. “Genre Trouble: Narrativism and the Art of Simulation.” *First Person: New Media as Story, Performance, and Game* (2004): 45-55.

Aarseth, Espen. Technoculture discussion list. September 23, 1993.

“A Conversation with David Helgason.” *DICE Europe*. 2015.

https://www.diceeurope.org/news/conversation_with_david_helgason.asp.

Adamkiewicz, Aleksander. “No, You Can't Make Video Games.” *Gamasutra*. Last modified January 8, 2012.

https://www.gamasutra.com/blogs/AleksanderAdamkiewicz/20120703/173489/No_You_Cant_Make_Video_Games.php.

Ahmed, Sarah. *The Cultural Politics of Emotion*. Edinburgh: Edinburgh University Press, 2015.

Ahmed, Sarah. “Orientations: Toward a Queer Phenomenology.” *GLQ: A Journal of Lesbian and Gay Studies* 12, no. 4 (2006): 543-547.

Anable, Aubrey. “Platform Studies.” *Feminist Media Histories* 4, no. 2 (2018): 135-140,

<https://fmh.ucpress.edu/content/4/2/135>.

Anable, Aubrey. *Playing with Feelings: Video Games and Affect*. Minneapolis: University of Minnesota Press, 2018.

Anthropy, Anna. *Rise of the Videogame Zinesters: How Freaks, Normals, Amateurs, Artists, Dreamers, Dropouts, Queers, Housewives, and People Like You Are Taking Back an Art Form*. New York: Seven Stories Press, 2012.

Apperley, Thomas and Jussi Parikka. "Platform Studies' Epistemic Threshold." *Games and Culture* 13, no. 4 (December 2015): 349-369. doi:10.1177/1555412015616509.

Axon, Samuel. "Unity at 10: For better—or worse—game development has never been easier." *ars technica*. Last modified September 27, 2016.
<https://arstechnica.com/gaming/2016/09/unity-at-10-for-better-or-worse-game-development-has-never-been-easier/>.

Barnett, Fiona, Zach Blas, Micha Cárdenas, Jacob Gaboury, Jessica Marie Johnson, and Margaret Rhee. "QueerOS: A User's Manual," in *Debates in the Digital Humanities 2016*, ed. Matthew Gold and Lauren Klein (University of Minnesota, 2016),
<https://dhdebates.gc.cuny.edu/read/untitled/section/e246e073-9e27-4bb2-88b2-af1676cb4a94>.

Barthes, Roland. *Mythologies*. Les Lettres nouvelles, 1972.

Bernardi, Joe. "Choose Your Own Adventure-Maker: Twine and the Art of Personal Games." *Vice*. Last modified February 19, 2013.
https://www.vice.com/en_us/article/xyyp9a/twine-and-the-art-of-personal-games.

Blas, Zach and Micha Cárdenas. "Imaginary computational systems: queer technologies and transreal aesthetics." *AI and Society* 28 (2013).

- Bogost, Ian. *Persuasive Games: The Expressive Power of Videogames*. Cambridge, MA: MIT Press, 2007.
- Bogost, Ian and Nick Montfort. "Platform Studies: Frequently Questioned Answers." *Proceedings of Digital Arts and Culture 2009*. University of California, Irvine.
- Boluk, Stephanie and Patrick LeMieux. *Metagaming: Playing, Competing, Spectating, Cheating, Trading, Making, and Breaking Videogames*. Minneapolis: University of Minnesota Press, 2017.
- <https://manifold.umn.edu/read/metagaming/section/84dabaa3-647e-4b18-8c8a-ba61cbf48fe3#cv1>.
- Brice, Mattie. "Death of the Player." Last Modified October 29, 2013.
- <http://www.mattiebrice.com/death-of-the-player/>.
- Brice, Mattie. "Play and Be Real about It." In *Queer Game Studies*, edited by Bo Ruberg and Adrienne Shaw, 77-82. Minneapolis: University of Minnesota Press, 2017.
- Cárdenas, Micha. "Dark Shimmers: The Rhythm of Necropolitical Affect in Digital Media," in *Trap Door: Trans Cultural Production and the Politics of Visibility* edited by Reina Gossett, Eric A. Stanley, and Johanna Burton, 161-181. Cambridge, MA: MIT Press, 2017.
- Ciccoricco, David. *Reading Network Fiction*. Tuscaloosa: University of Alabama Press, 2007.
- Clark, Naomi. "Horse Master by Tom McHenry." In *Videogames for Humans: Twine Authors in Conversation*, edited by merriitt kopas, 85-134. New York: Instar Books, 2015.
- Clark, Naomi. "What Is Queerness in Games, Anyway?" In *Queer Games Studies*, edited by Bo Ruberg and Adrienne Shaw, 3-14. Minneapolis: University of Minnesota Press, 2017.

Chun, Wendy Hui Kyong. "On 'Sourcery,' or Code as Fetish," *Configurations* 16, no. 3 (2008): 299-324. doi: 10.1353/con.0.0064.

Chun, Wendy Hui Kyong and Sarah Friedland. "Habits of Leaking: Of Sluts and Network Cards." *Differences* 26, no. 2 (2015): 1-28. doi: 10.1215/10407391-3145937.

Cross, Katherine. "Analysis: A closer look at MidBoss' chaotic, hostile work environment."

Gamasutra. Last modified March 28, 2018,

https://www.gamasutra.com/view/news/315910/Analysis_A_closer_look_at_MidBoss_chaotic_hostile_work_environment.php.

Conn, Matt. *Queerly Represent Me*. Interview by the staff of Queerly Represent Me. Accessed

October 15, 2019. <https://queerlyrepresent.me/projects/interviews/matt-conn>.

D'Anastasio, Cecilia. "Why Video Games Can't Teach You Empathy." *Motherboard*. Last modified May 15, 2015.

https://www.vice.com/en_us/article/mgbwpv/empathy-games-dont-exist.

ddall1. "Building a Combination Lock in Twine 2, Harlowe 2." *Instructables*. Accessed

November 29, 2019.

<https://www.instructables.com/id/Building-a-Combination-Lock-in-Twine-2-Harlowe-2/>.

Dellinger, AJ. "The environmental impact of data storage is more than you think -- and it's only getting worse." *Mic*. Last modified June 19, 2019.

<https://www.mic.com/p/the-environmental-impact-of-data-storage-is-more-than-you-think-its-only-getting-worse-18017662>.

Disco, Benny. "Let's Look at Tropes Vs. Fat Guys in Videogames." *Destructoid*. October 28, 2013.

<https://www.destructoid.com/let-s-look-at-tropes-vs-fat-guys-in-videogames-264375.html>.

Dunietz, Jesse. "The Most Important Object In Computer Graphics History Is This Teapot - Facts So Romantic." *Nautilus*. Last modified February 29, 2016, <http://nautil.us/blog/the-most-important-object-in-computer-graphics-history-is-this-teapot>.

Ellison, Cara. "Hypersexed Hypertext: Porpentine and the Twine text game revolution." *PC Gamer*. last modified April 3, 2013. <https://www.pcgamer.com/hypersexed-hypertext-porpenine-and-the-twine-text-game-revolution/>.

Everett, Anna. Forward to *Gaming Representation: Race, Gender, and Sexuality in Video Games*. ix-xviii. Edited by Jennifer Malkowski and TreaAndrea M. Russworm. Bloomington: Indiana University Press, 2017.

Fabello, Melissa A. "Femme Invisibility: On Passing Right by Your People and Not Being Recognized." *Everyday Feminism*. October 16, 2014. <https://everydayfeminism.com/2014/03/femme-invisibility/>.

Florian. "The Best Game Engines for Beginners: 14 Alternatives." *Website Tool Testers*. November 29, 2018. <https://www.websitetooltester.com/en/blog/best-game-engine/>.

Foucault, Michel. *The History of Sexuality Volume I*. 1978.

Freedman, Eric. "Engineering Queerness in the Game Development Pipeline." *Game Studies* 18, no. 3 (2018): <http://gamestudies.org/1803/articles/ericfreedman>.

Gaider, David. "A Character Like Me: the lead writer of Dragon Age on inclusive games."

Polygon. February 18, 2014.

<https://www.polygon.com/2014/2/18/5422570/the-lead-writer-of-dragon-age-on-the-first-steps-towards-inclusive>.

Game Developers' Association of Australia. "What Game Engines Are Companies Using?"

Facebook (2018):

<https://www.facebook.com/GameDevAssocAus/photos/a.1943622552593569/2062766267345863/?type=3&theater>.

Gaudiosi, John. "Why Epic Games is giving away its game technology." *Fortune*. Last modified

March 3, 2015. <https://fortune.com/2015/03/03/epic-games-unreal-tech-free/>.

Gee, James Paul. "Video Games and Embodiment." *Games and Culture* 3, no. 3-4 (2008):

253-263, doi: 10.1177/1555412008317309.

Goddard, Michael. "Opening up the Black Boxes: Media Archaeology, 'Anarchaeology' and Media Materiality." *New Media & Society* 17, no. 11 (December 2015): 1761-76.

doi:10.1177/1461444814532193.

Graft, Kris. "id Tech 5 Rage Engine No Longer Up For External Licensing." *Gamasutra*. Last modified August 13, 2010.

Gregory, Jason. *Game Engine Architecture*. Boca Raton: Tayler and Francis Group, LLC, 2009.

Halberstam, Jack. *The Queer Art of Failure*. Durham: Duke University Press, 2011.

Harvey, Alison. "Twine's revolution: Democratizing, depoliticization, and the queering of game design," *GAME* 3 (2014): 95-107.

Helgason, David. "Leading Unity into the Future." *Unity Blogs*. October 22, 2014.

<https://blogs.unity3d.com/2014/10/22/leading-unity-into-the-future/>.

Helgason, David. "A free Unity?" *Unity Blogs*. October 29, 2009.

<https://blogs.unity3d.com/2009/10/29/a-free-unity/>.

Hernandez, Patricia. "How Video Game Breasts Are Made (And Why They Can Go Wrong)."

Kotaku. Last modified February 24, 2015.

<https://kotaku.com/how-video-game-breasts-are-made-and-why-they-can-go-so-1687753475>.

Herz, J.C. "GAME THEORY; For Game Maker, There's Gold in the Code." *The New York*

Times. December 2, 1999.

<https://www.nytimes.com/1999/12/02/technology/game-theory-for-game-maker-there-s-gold-in-the-code.html>.

Houlden, Sophie. "Stop being such an asshole to people who don't start from scratch." *Sophie's*

Blog. Last modified June 15, 2012.

<http://www.sophiehoulden.com/stop-being-such-an-asshole-to-people-who-dont-start-from-scratch/>.

Huhtamo, Erkki and Jussi Parikka. "Introduction: An Archaeology of Media Archaeology." In

Media Archaeology: Approaches, Applications, and Implications, edited by Erkki

Huhtamo and Jussi Parikka, 1-21. Oakland, CA: University of California Press, 2011.

Jackson, Shelley. "Stitch Bitch: the patchwork girl." Presented at Transformations of the Book

Conference. Cambridge, Massachusetts, October 1998.

Jones, Ian Bryce. "Do the Locomotion: Obstinate Avatars, Dehiscent Performances, and the Rise of the Comedic Video Game." *The Velvet Light Trap* 77 (2016): 86-99. doi:

10.7560/vlt7706.

Kaplan, Omer. "Mobile gaming is a \$68.5 billion global business, and investors are buying in."

Tech Crunch. August 22, 2019.

<https://techcrunch.com/2019/08/22/mobile-gaming-mints-money/>.

Keogh, Brendan. "Just making things and being alive about it: The queer games scene." *Polygon*.

May 24, 2013,

<https://www.polygon.com/features/2013/5/24/4341042/the-queer-games-scene>.

Klimas, Chris. "Twine: Past Present, Future." *Chris Klimas* (blog). June 2019.

<https://chrisklimas.com/twine-past-present-future/>.

Kluitenberg, Eric. "On the Archaeology of Imaginary Media." In *Media Archaeology:*

Approaches, Applications, and Implications, edited by Erkki Huhtamo and Jussi Parikka,

48-69. Oakland, CA: University of California Press, 2011.

kopas, merriitt. Introduction to *Videogames for Humans: Twine Authors in Conversation*, edited

by merriitt kopas, 5-19. New York: Instar Books, 2015.

Kushner, David. *Masters of Doom: How Two Guys Created an Empire and Transformed Pop*

Culture. London: Piatkus, 2013.

"LinkedIn's 2017 U.S. Emerging Jobs Report." *LinkedIn*. December 7, 2017.

<https://economicgraph.linkedin.com/research/LinkedIns-2017-US-Emerging-Jobs-Report>

- Livingston, Christopher. "Conan Exiles Has an 'Endowment' Slider and Genital Physics." *PC Gamer*. January 31, 2017. <https://www.pcgamer.com/conan-exiles-dick-slider/>.
- Love, Heather. *Feeling Backward: Loss and the Politics of Queer History*. Harvard University Press, 2007.
- Lowood, Henry. "Game Engines and Game History." Conference presentation at the History of Games International Conference, 2014.
- Lowood, Henry. "High-Performance Play: The Making of Machinima." *Journal of Media Practice*, 7, no. 1 (2006): 25-42. doi: 10.1386/jmpr.7.1.25/1.
- Malkowski, Jennifer and TreaAndrea M. Russworm. Introduction to *Gaming Representation: Race, Gender, and Sexuality in Video Games*. 1-16. Edited by Jennifer Malkowski and TreaAndrea M. Russworm. Bloomington: Indiana University Press, 2017.
- Manovich, Lev. *The Language of New Media*. Cambridge, MA: MIT Press, 2001.
- McPherson, Tara. "Designing for Difference," *Differences* 25, no. 1 (January 2014): 177-188. doi: 10.1215/10407391-2420039.
- Mesit, Jaruwan. "Modeling and Simulation of Soft Bodies." Doctoral thesis, University of Central Florida, 2010, http://etd.fcla.edu/CF/CFE0003477/Jaruwan_Mesit_201012_PhD.pdf.
- Montfort, Nick and Ian Bogost. *Racing the Beam*. Cambridge, MA: MIT Press, 2009.
- Montfort, Nick. *Twisty Little Passages: an Approach to Interactive Fiction*. Cambridge, MA: MIT Press, 2005.

Moser, AJ. "Overwatch's historic openly gay character draws (mostly) praise from gamers." *The Daily Dot*. December 21, 2016.

<https://www.dailydot.com/parsec/overwatch-tracer-reactions/>.

Moulthrop, Stuart. "Traveling in the Breakdown Lane: A Principle of Resistance for Hypertext." *Mosaic: An Interdisciplinary Critical Journal* 28, no. 4 (1995): 55-77.

Nakamura, Toshi. "You Can Adjust 'Bounce' in the New Dead or Alive." *Kotaku*. Last modified October 23, 2014.

<https://kotaku.com/you-can-adjust-bounce-in-the-new-dead-or-alive-1649708514>.

Nicoll, Benjamin and Brendan Keogh. *The Unity Game Engine and the Circuits of Cultural Software*. Cham, Switzerland: Springer Nature Switzerland, 2019.

Nicoll, Benjamin. *Minor Platforms in Videogame History*. Amsterdam University Press, 2019.

Phillips, Amanda. "dicks dicks dicks: Hardness and Flaccidity in (Virtual) Masculinity." *Flow Journal*. November 27, 2017. <https://www.flowjournal.org/2017/11/dicks-dicks-dicks/>.

Porpentine. "Creation Under Capitalism." *The Gamer Trust*. Last modified November 25, 2012. <https://nightmaremode.thegamerstrust.com/2012/11/25/creation-under-capitalism/>.

Porpentine. *Emily Short's Interactive Storytelling*. Interview by Emily Short. November 23, 2012.

<https://emshort.blog/2012/11/23/interview-with-porpenine-author-of-howling-dogs/>.

Pozo, Teddy. "Queer Games After Empathy: Feminism and Haptic Game Design Aesthetics from Consent to Cuteness to the Radically Soft." *Games Studies* 18, no. 3 (2018):

<http://gamestudies.org/1803/articles/pozo>.

Purbrick, Louise. "The Dream Machine: Charles Babbage and His Imaginary Computers."

Journal of Design History 6, no.1 (1993): 9-23.

Quinn, Zoe. "All Gamergate Has done is ruin people's lives." Interview with Keith Stuart.

Guardian. December 3, 2014.

<https://www.theguardian.com/technology/2014/dec/03/zoe-quinn-gamergate-interview>.

Randell, Kim. "Jurassic Bark – this one's a dog." *Computer And Video Games*. Last modified

August 15, 2001.

<https://web.archive.org/web/20070126112724/http://www.computerandvideogames.com/article.php?id=8366>.

Reeves, William T. "Particle Systems—A Technique for Modeling a Class of Fuzzy Objects."

ACM Transactions on Graphics 2, no. 2 (April 1983): 91-108. doi:

0730-1301/83/0400-0091.

Ruberg, Bo. "The Precarious Labor of Queer Indie Game-making: Who Benefits from Making

Video Games 'Better'?" *Television & New Media* 20, no. 8 (2019): 778-788.

<https://www.facebook.com/GameDevAssocAus/photos/>

Ruberg, Bo. *Video Games Have Always Been Queer*. New York: New York University Press,

2019.

Ryan, Marie-Laurie. "Interactive Drama: Narrativity in a Highly Interactive Environment." *MFS*

Modern Fiction Studies 43, no. 3 (1997): 677-707, doi:10.1353/mfs.1997.0065.

Sackenheim, Shawn. "Trespasser-Review." *AllGame*. Accessed December 19, 2019.

<https://web.archive.org/web/20141115043401/http://www.allgame.com/game.php?id=14402&tab=review>.

Salen, Katie and Eric Zimmerman. *Rules of Play: Game Design Fundamentals*. Cambridge: MIT Press, 2004.

Saussure, Ferdinand de. *Course in General Linguistics*. 1916.

Spoon, Rae and Ivan E. Coyote. *Gender Failure*. Vancouver, BC: Arsenal Pulp Press, 2014. EPUB.

Stone, Sandy. "The *Empire* Strikes Back: A Posttranssexual Manifesto." In *The Transgender Studies Reader*, edited by Susan Stryker and Stephen Whittle, 221-235. New York: Routledge, 2006.

"UNITY: Features - Wii Publishing." *Unity*.

<https://web.archive.org/web/20081220051710/http://unity3d.com/unity/features/wii-publishing>.

Vanderhoef, John. "Casual Threats: The Feminization of Casual Video Games." *Ada: A Journal of Gender, New Media, and Technology* no. 2 (2013): doi:10.7264/N3V40S4D.

Vist, Elise. "Cyborg Games: Videogame Blasphemy and Disorientation." *Loading... The Journal of the Canadian Game Studies Association* 9, no. 14 (2015): 55-69.

Wilchins, Riki. *Queer Theory, Gender Theory: An Instant Primer*. New York: Alyson Books, 2004.

Wood, Jordan. "Romancing an Empire, Becoming Isaac: The Queer Possibilities of *Jade Empire* and *The Binding of Isaac*." In *Gaming Representation: Race, Gender, and Sexuality in Video Games*. 212-226. Edited by Jennifer Malkowski and TreaAndrea M. Russworm. Bloomington: Indiana University Press, 2017.

Wycoff, Richard. "Postmortem: DreamWorks Interactive's *Trespasser*." *Gamasutra*. Last modified May 14, 1999.

https://www.gamasutra.com/view/feature/131746/postmortem_dreamworks_.php.

Zielinski, Siegfried. *Deep Time of the Media*. Translated by Gloria Custance. Cambridge, MA: MIT Press, 2006.

Zielinski, Siegfried. "Modelling Media for Ignatius Loyola: A Case Study on Athanasius Kircher's World of Apparatus between the Imaginary and the Real." In *The Book of Imaginary Media: Excavating the Dream of the Ultimate Communication Medium*, edited by Eric Kluitenberg, 28-55. Rotterdam, Netherlands: nai010 PUBLISHERS: 2007.