2021

Critical Modding: A Design Framework for Exploring Representation in Games

Kenton Howard
University of Central Florida

CRITICAL MODDING: A DESIGN FRAMEWORK FOR EXPLORING REPRESENTATION IN GAMES

by

KENTON TAYLOR HOWARD
B.A. Flagler College, 2009
M.A. Florida Atlantic University, 2012

A dissertation submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
in the College of Arts and Humanities
at the University of Central Florida
Orlando, Florida

Summer Term
2021

Major Professor: Anastasia Salter

ABSTRACT

In this project, I created a framework for exploring problems with representations of marginalized characters in video games called “critical modding.” The main goal of this project was to provide a method for addressing issues with portrayals of queer characters in video games through modification of a game’s narrative and gameplay systems. I also created a video game prototype called *Life in the Megapocalypse* as a digital tool for engaging in critical modding. In addition, I created classroom assignments based on the game prototype aimed at helping people learn more about problems with portrayals of queer characters in games through critical modding that can be found in Appendix B. I created the critical modding framework based on established research on education, queer representation, and narrative design related to video games. The video game prototype is built in a text-based interactive fiction scripting engine called Inky and is stored in a web-based version on my website; I also provided selected source code from the game in Appendix C of this project. The game focuses on representations of queer characters in a post-apocalyptic world and asks the player to make choices by offering the characters guidance, as well as talking with them to learn more about them. This project is valuable in that it provides a research-based framework for exploring problems with portrayals of queer characters in video games through modification, an idea that has not been explored in established video game research. It also provides a tool for doing so in the form of the video game prototype, offering a unique approach that blends traditional academic research with digital design.

ACKNOWLEDGMENTS

I offer my sincerest thanks to Anastasia Salter, Mel Stanfill, Rudy McDaniel, and Lisa Swanstrom for offering me feedback and advice throughout this project, as their suggestions provided me with invaluable help. I also thank everyone else who gave me feedback and advice at different stages of my writing and game prototyping process. Finally, I thank my friends and family whose emotional support was critical to the success of this project.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ACRONYMS

Life in the Megapocalypse	LitM

CHAPTER 1:INTRODUCTION: THEORETICAL FOUNDATIONS FOR RADICAL GAME DESIGN

In this project, I lay out an approach for modifying video games that focuses on education, representation, and radical game design. In terms of education, video games are often seen as a new form of media that can be used to replace or supplement traditional educational media like textbooks as well as tools that can teach a particular topic and assess a student's mastery of that topic at the same time, while less attention is paid to the potential of game design itself as a learning method. Such potential could be harnessed to help people learn more about representation in games, but using video games for educational purposes could be problematic because mainstream games have often either underrepresented marginalized groups of people or represented them very poorly. Smaller and independent game developers have made games such as *Dream Daddy (*2017) that attempt to address those issues by creating games that focus on queer characters, but many video game designers aim at creating large mainstream games in an industry setting that typically ignore such issues and often use representation of marginalized groups of people as a selling tool rather than striving for actual inclusion. To address these issues, in this project I lay out an approach that focuses on how video game education, representation, and design research can be used to create video game modifications that are aimed at addressing critical issues in video games. I call this method "Critical Modding:" overall, I argue for a modification-based approach as method for exploring problems with representation in video games.

I focus on a modification-based approach in this project because games are an effective medium for exploring representation as most of them feature some kind of story and because game modification allows players to directly change that story. Games also inherently encourage players to experiment with different approaches to scenarios and to solve problems, naturally suggesting a problem-solving oriented approach to issues with how marginalized groups are represented in video games. As such, critical modding is an approach that is aimed at exploring solutions to problematic issues of representation, such as those related how queer characters are portrayed in video games, could be an effective vehicle for modeling ways to address those issues. Games struggle with many kinds of representation and often reinforce negative stereotypes about marginalized groups, however, and there is no guarantee that modders will necessarily address issues with how such groups are represented in a positive way, which means caution must be used when suggesting such an approach. I therefore suggest that the best approach is to create a game that focuses on a particular kind of representation with critical modding in mind, as well as some examples of the kinds of changes that might be made to the game. With this approach in mind, I have built a modifiable game prototype as part of this project that is intended to model more positive forms of queer representation and serve as a starting point for people who are interested in exploring that concept further through game design. This game, *Life in the Megapocalypse (LitM),* is a choice-based interactive fiction game that features a narrative focused on queer characters surviving in a hostile setting. *LitM* is therefore an example of a game that is intended for critical modding: the prototype puts the ideas I lay out here into practice through the design of a digital object that others can modify and expand upon. I have also created sample classroom assignments based on critical modding that could be used with *LitM* in a classroom setting.

Research Question

While many authors have examined games and learning in different contexts, there is less work on using modification of video games as a method for exploring critical concepts like queer characters in video games. I believe that modification of video games can be an extremely effective method for exploring representation in video games, and therefore this dissertation focuses on expanding the field of research on learning and games by answering the following main research question:

RQ1: How do educational, critical, and design concepts help people explore problems with representation in games and how do those concepts inform the design of digital tools for exploring those problems?

In addition, I explore these secondary research questions in individual chapters of this project:

SRQ2: How does a modification-based method for exploratory learning through interactive fiction help people learning the skills needed to explore problems with representation in video games?

SRQ3: How do indie video games and video game modifications address problems with how queer characters are represented in mainstream video games and what design strategies can be drawn from them?

SRQ4: What design elements are required for a modifiable digital game informed by theories about narrative design and queer game studies?

I answer these questions in this dissertation by laying out a method for using video game modification to help players explore portrayals of queer characters in video games that is based on established research related to education, representation, and design. First, I created a written project that lays out a pedagogical, critical, and design framework for using modification of video games to explore issues with representation in gaming that I call "critical modding." Second, I developed an easily modifiable video game prototype, *LitM,* that is built in the Ink scripting language and that provides a framework that users can modify to explore portrayals of queer characters in video games through critical modding; I also provided the full source code for the game on my website and offered code samples in Appendix C. The two parts also build upon each other in that the written dissertation addresses theoretical considerations related to this approach that would be difficult to address directly in a game, while the game prototype is an example of a tool that can be used to address these considerations in the form of an easily expandable digital object. The main objectives of these two objects are therefore related: one of my goals is to create a written framework for educational game design that is focused on using modification for exploring portrayals of queer characters in video games, and my other goal is to create a playable and modifiable example of what a game based on that framework might look like that others can build upon. I have provided a design document for the prototype in Chapter 4 and have included sample source code from the game in Appendix C

Theoretical Frameworks and Methodology

Education and Games

This dissertation builds upon research related to video games and education and some key themes from research in that field informs the methodology that I lay out in my dissertation. In James Paul Gee's *Good Video Games and Good Learning* (2006) and *What Video Games Have to Teach Us About Learning and Literacy* (2007) the author argued for one of the primary reasons that video games might be useful as learning tools in that they are more engaging than traditional classroom instruction, an idea echoed by many scholars who argue for using games in education. Building upon that notion, I suggest that the engaging quality of games makes them a useful medium for exploring representations of marginalized characters because games might help people engage with narratives they might otherwise not explore. Similarly, in Katie Salen's (2008) *The Ecology of Games: Connecting Youth, Games, and Learning,* the author argued for another key theme in research on video games and education: the idea that active creation of digital objects is a useful approach to learning. Salen (2008) suggested that "contemporary research influencing the games and learning space has started to move beyond an analysis of media as a way to become literate about it, and towards an emphasis on creative production and the principles of design" (p. 6); as such, I argue that the creative production involved in modifying a game could provide players with a way of finding solutions to the kinds of problematic representations of marginalized characters that are often seen in video games. Finally, Jennifer DeWinter et. al.'s (2010) "Computer Games Across the Curriculum: A Critical Review of an Emerging Techno-Pedagogy" discussed the flexibility of games, suggesting that they are useful because of the variety of concepts they can engage with and the numerous different they can do so. In a summary of the situation, the authors argued that "the medium teaches multiple things in multiple ways" (2010). This flexibility is particularly useful for the critical modding strategy I propose here because it enables players to enact their own perspectives on queer characters in video games rather than being limited to only the portrayals that I built into the base game. Additionally, that flexibility suggests that other issues with representation, such as the problematic ways in which video games depict race, might also be meaningfully addressed through video game modification, though I primarily explore queer representation in this project and do not suggest that such issues could be simply addressed in exactly the same way that I describe here. Overall, these major themes in education research suggest that critical modding could enable players to engage with representation and create their own narratives that can help contextualize the different kinds of queer characters encountered in video games. As such, in this project I demonstrate how critical modding can be used to address to problems with representation in video games and can provide mechanisms for meaningful play and design.

I also suggest that the kinds of interaction that people have with games and the importance that people place on games as a form of media are both useful for learning about issues with representation in games. In Tom Apperly and Catherine Beavis' "A Model for Critical Games Literacy," the authors suggested that "while in many respects the literacy practices developed through digital games are similar to those required for any other digital media, we argue that digital games are different because they are enacted by the player" (2013, p. 3).This kind of enacted literacy is particularly valuable to my critical modding approach because it enables people to actively explore how queer characters are represented in video games: players can put their understandings of game stories into practice through modification of gameplay. I also rely on such an approach in *LitM* itself by presenting narrative and gameplay scenarios that focus on queer characters. I further suggest ways that players might change those scenarios and characters in comments within the game's code and through the sample assignments found in Appendix B, encouraging them to act on those suggestions. Catherine Beavis et. al.'s "Turning Around to the Affordances of Digital Games: English Curriculum and

Students' Lifeworlds" also established a framework for how video games can be used for learning about critical concepts in educational settings by validating students' experiences with digital texts outside of the classroom. The authors highlighted various educational projects that helped to demonstrate the "critical social and cultural significance of engagement with digital games in young people's lives" (Beavis et. al. 2015). While I discussed the importance of engagement above, such research also shows that games can be a powerful medium for considering the stories of queer characters because of their critical and cultural significance: put simply, games have become an important form of media, especially for younger students. I argue that critical modding is similarly useful because of this importance: it gives people the ability to critically engage with a medium that now informs many cultural considerations about representation.

Transference – the idea that skills learned in an educational environment can be applied in other environments – is also a key consideration when working with any kind of game-based learning projects. Though some scholars are skeptical about whether the skills learned in games are transferrable, in "Epistemic Games" David Schaffer argued that almost any kind of skill learned in gameplay could potentially be transferrable because "in play, we participate in a simulation of a world we want to inhabit" (2005, p.2). As an example of this idea, he describes a game that could be used to simulate the real-life tasks of an urban planner, allowing players to learn the same skills that one would need in that career. By extension, I suggest that a game with a narrative and gameplay that focuses on the stories of queer characters allows players to participate in a world in which queer representation is particularly important: furthermore, by modifying *LitM*, people can enact the kinds of representation they would like to see in the game's story themselves much in the same way that actual game designers do. Through this process, I suggest that critical modding offers an approach that can address problematic issues with representation in games while also learning game design skills, and I discuss how those game design skills can be learned in Chapter 2 of this project.

It is also worth noting that there has been skepticism surrounding the notion of modding as a learning practice. In *Video Games and Learning: Teaching and Participatory Culture in The Digital Age,* Kurt Squire (2011) suggested that the skills learned by creating content for games through modification are quite useful, even if they are not always seen as such: "the most advanced, sophisticated forms of participation in games culture – designing levels, mods, fictions, or guilds – are largely unacknowledged, if not derided, by mainstream educational institutions" (p. 59). As the author notes, learning through creative practices such as modification is not common in traditional educational institutions even though modding teaches many transferrable skills about programming and game design, so one goal of this project is to legitimize such forms of creative practice as valuable learning experiences that can teach such skills in an effective way. While modification has been explored in the context of learning in the past, few projects have asked learners to engage in the practice directly, which is also one primary contribution of the approach I propose in this dissertation. Squire (2011) also argued that "video game players develop a feel or intuition of how systems work" (p.5) and suggested that "this *systemic* thinking is valuable because it helps people solve problems holistically, rather than focusing on single-cause solutions" (p. 5). As such, I suggest that critical modding of a game focused on the stories of queer characters can be a useful learning practice in that it allows people to explore the ways in which problematic kinds of representation are encoded into the systems of video games and to focus more on changing the systems as a whole rather than single instances of problematic representation.

As noted earlier, *LitM* is designed to be easy to modify so that users can change the game's content or add new content to the game. This design is possible because the Inky engine is available for free and relatively easy to learn due to the extensive documentation provided on Inkle's website, which is one reason for why I choice the engine as opposed to a much more difficult to learn engine like Unity or Unreal. I also provide sample source code of the game in Appendix C of this project along with extensive code comments that describe how the game works and how users can make their own changes to it, as my aim is to create a tool that others can build upon. Designing a game so that it is easy for users to create content for it could be powerful as a learning method because that design makes digital content creation more accessible. When discussing user-created content in text-based multi-user dungeon games of the 1980s, Squire (2011) argued that such games were useful for learning because "any player – even kids – could invent new creatures and author new areas or suggest changes to the underlying rules structure" (p. 43). One similar advantage of a game like *LitM* is that it makes content creation more accessible because it is designed to be modified by users, allowing players to create content more easily than they could if the game was developed in a more complex game engine. This kind of gameplay modification is also the primary framework for learning that I argue for in this project, as it is well established that players can learn through playing games. I argue that critical modding of a game that focuses on queer characters and their stories can further help users explore that kind of representation in games through their own creative practices. In addition, Squire (2011) suggested that people who created content for games demonstrated “deep knowledge” and “systemic understandings” (p. 44), such that users learned about the how the games in question worked to effect the changes they made. Critical modding of *LitM* therefore serves a similar purpose, as the game is a code-based learning tool that focuses on game design and queer representation and can give players a systemic understanding of how those issues come together in a digital object.

Queer Game Design and Theories About Representation in Games

I relied on critical sources related to queer game design to inform the critical modding approach that I took in this project. Ruberg (2019) described the concept of queer game design as “the practice of deliberately using game mechanics and other design elements to challenge normative expectations around gender, sexuality, and the established logics of digital gameplay” (p. 111). I employed queer game design in *LitM* by primarily relying on design choices that challenge the norms of post-apocalyptic game stories, such as focusing on queer characters in a genre that typically includes few or no queer characters and often assumes that most of their players are straight. *LitM* also challenges these norms on a mechanical level: the game’s only straight character, Guy McMannis, is the worst choice for resolving in-game challenges because using his abilities repeatedly causes the player to receive one of the worst endings for the game. Furthermore, my suggestions for modifying the game described in Chapter 5 focus on changing elements related to the game’s queer characters and gameplay systems, effectively allowing people to queer the logic of the game by changing the norms built into its code. I suggest that embedding queer game design throughout smaller design choices such as the ones described above is useful because such elements contribute to the atmosphere of a game and impact the player’s overall gameplay experience. I also relied on queer game design advice from sources such as *Queerly Represent Me* (2019), who provided a flowchart model titled “So, you’ve decided you want to make a diverse game… now what?” to answer common questions that designers have about creating a game narrative that focuses on queer characters. Their advice that designers avoid stories of trauma and focus on empowering portrayals of queerness (Queerly Represent Me, 2019) informs the design of the narrative and gameplay in *LitM.* Another common problem in games is a lack of agency for queer characters, so the focus of the narrative and gameplay in *LitM* is on using the strengths of the game’s queer characters to solve problems and not relying on the typical violence-driven straight male characters found in many post-apocalyptic games. The assignments I discuss in Chapter 5 further encourage people to explore those concepts by providing suggestions for how modders might change the game’s code to further expand on the stories of the game’s queer characters, providing a model for putting the critical modding approach I describe here into practice.

Since the narrative of *LitM* focuses on queer characters and their stories, some of my methodology focuses on establishing a theoretical framework for addressing queer stories through modification. Games are powerful tools for exploring representation because, as Squire (2011) suggested, “Games are both media *and* play. They allow us to play with representations… and to test the rule systems of those representations” (p. 30). Despite this fact, video games have commonly struggled to represent queer characters effectively: for example, in *Video Games Have Always Been Queer* (2019), Bonnie Ruberg argued that “video games have notoriously underrepresented LGBTQ people” (p. 3), suggesting that games and queerness have typically engaged in problematic ways, often simply because of their lack of queer characters. This lack of engagement is important because, as Adrienne Shaw argued in *Gaming at the Edge* (2014),

> to treat representation in games as being just about games… fails to account for the ways in which violence against queers (homo- or bisexual or not), women (cisgendered or queer or not), and people of color (queer or not, cisgendered women or not) exists everywhere, in all media, and in all institutions of power. (p.2)

Shaw suggests here that the way marginalized people are represented – or, in this case, are not represented – in media like games is important because those problems reflect larger issues with a lack of representation of LGBTQ people in media and culture. It is important to note, however, that the inclusion of queer characters in a game like *LitM* is not an automatic solution to problems with a lack of representation: Warner (2017) points out that "strategies that attempt to overlap negative images with positive ones in a quest to present a respectability of social imagery do not eradicate either" (p. 34). In fact, many mainstream games that do include queer characters portray them particularly poorly, an issue I explore in more detail in Chapter 3. Instead, the inclusion of those characters in *LitM* is meant to provide a framework for critical modding that focuses on queer characters in games, which is the primary form of representation that this project engages with.

Games are also a particularly useful medium for exploring problems with queer characters in games because of the ability to actively model solutions through critical modding. Shaw's (2017) article "Diversity Without Defense: Reframing Arguments for Diversity in Games" featured discussions with queer gamers, many of whom "rejected direct media effects theories, embracing a more active audience description of their own relationship to the media they consumed" (p. 55) in that many queer gamers construct their own narratives about their in-game characters that are not always supported by the game's actual story. A good example of this kind of construction would be Harper's (2017) description of "ClosetShep," an approach to playing the *Mass Effect* games that treats its main character, Commander Shepard, as a closeted gay man (p. 127). I argue that this active audience effect can be harnessed for exploring queer representation through critical modding in that it offers a method for expanding the stories of queer characters in games. Rather than only analyzing problematic representation in games or constructing narratives that sit outside the game's story, critical modding of a game like *LitM* allows people to easily change the game's narrative and gameplay as well, such that they can engage with queer representation in a way that more closely represents their own relationship to video games. In this sense, critical modding is less about simply including queer characters or more positive forms of queer representation and more about enabling people to explore queer representation in an open-ended way that suits their own interests.

Queer thinking and critique also informs my critical modding approach, as I suggest that modification itself can be a way of queering a game's narrative and mechanics by changing its norms. In Shaw's collection with Bonnie Ruberg, *Queer Game Studies,* the authors suggested that "queer thinking has the potential to simultaneously destabilize and reimagine video games themselves" (Shaw and Ruberg, 2017, p. ix). One of the goals of this project is to lay out the ways in which critical modding can address problematic issues surrounding representation in games by destabilizing common gameplay narrative norms, so this kind of queer thinking forms an underlying theoretical perspective for both my written dissertation and the narrative design of the queer characters in *LitM.* In addition, in *Undoing Gender,* Judith Butler argued that "critique is understood as an interrogation of the terms by which life is constrained in order to open up the possibility of different modes of living" (2004, p. 4) and suggested that much of queer critique is focused on considering ways of life that would reduce the marginalization of queer people. I argue that critical modding is one way to put this kind of queer criticism into practice in that it enables video game players to directly change narrative and gameplay to represent queer characters in games in ways that are less problematic than current models. Butler (2004) also raised an important question when considering complex situations like the DSM-IV diagnosis that both enables insurance for sex reassignment surgery and simultaneously pathologizes trans people: “the critical question thus becomes, how might the world be reorganized so that this conflict can be ameliorated?” (p. 5). Critical modding is a powerful framework for learning because it allows for this kind of reorganization: in effect, players can queer a game to include queer characters or expand upon the portrayals of queer characters that it already contains. Modification is also a useful framework for queer theory because of its focus on continuous revision, which Eng, Halberstam, and Muñoz suggest is a necessity with respect to queer theory: “the operations of queer critique… can neither be decided on in advance nor be depended on in the future. The reinvention of the term is contingent on its potential obsolescence, one necessarily at odds with any fortification of its critical reach in advance” (2005, p. 3). From this perspective, critical modding could be an ideal vehicle for exploring about queerness in video games because it allows players to directly critique the games that they are playing by reinventing elements of the game and building their critique into the design of the game itself, revising that critique whenever necessary to address new queer issues. Similarly, this factor is why I intend to continue expanding *LitM* once this project is complete: I envision the prototype as a project that evolves over time and one that can be expanded upon by others as well.

Narrative and Critical Theories for Radical Game Design

Since *LitM* is an interactive fiction game that relies on randomized events and text to build an overall story, theoretical frameworks related to procedural narrative also inform the game's design. In Ian Bogost's *Unit Operations: An Approach to Video Games Criticism,* the author argued that "any medium… can be read as a configurative system, an arrangement of discrete, interlocking units of meaning" (2006, p. ix), an effect that more obvious in interactive fiction games, as such games typically rely on chunks of text that are configured into interlocking units of meaning that create a story for that particular playthrough of the game. As such, I suggest that critical modding allows players to gain a deeper understanding of how systems in games like *LitM* construct meaning because they can modify those systems to create new ways that the game's narrative can portray queer characters. Critical modding could therefore be valuable as an educational approach because, according to Squire (2011), "games are 'ideological worlds' in that they instantiate ideas through implicit rule sets and systems" (p. 28). As such, modification of the rule systems of *LitM* allows players to change the ideological world of the game directly and explore different understandings of how queer characters are portrayed in games through changes to the game's rules. Bogost's (2007) *Persuasive Games* also supports this approach, as he develops the notion of procedural rhetoric in that text by suggesting that videogames exercise "the art of persuasion through rule-based representations and interactions" (p. ix), such that modifying a game could be seen as its own act of persuasion. Taken in conjunction with Squire's notion of games as ideological worlds, I frame critical modding as a method that can reinvent the rule systems that govern representation in video games. The ability of games to create a procedurally generated narrative is key to my approach, as it allows people to modify the game's story and see the results of those changes reflected in the game's systems. As such, I did not create lengthy, in-depth narratives for the queer characters in my prototype: instead, *LitM* uses procedural story generation to create a narrative that encourage players to explore the stories of the game's queer characters, and critical modding offers a method that allows people to expand those stories if they wish.

From a design perspective, narrative is particularly useful for learning when compared to other elements of games, suggesting the potential benefits of an educational approach that is focused on modification of narrative as a way for people to explore problems with representation in games. Martey et al. (2017) conducted a study called "Testing the Power of Game Lessons: The Effects of Art Style and Narrative Complexity on Reducing Cognitive Bias" that demonstrated the effectiveness of video game narratives at teaching cognitive bias when compared to video-based lessons aimed at accomplishing the same goal (p. 1652). While the authors could not make an overall determination as to whether art style or narrative complexity had a more significant impact on student learning (Martey et. al., 2017, p. 1652), they did note that "light narrative" games were particularly effective at helping students retain information about cognitive bias (CB): "average CB mitigation was actually *better* at the eight-week retention test than it was at the immediate posttest" (Martey et. al., 2017, p. 1653). This data suggests that people might retain the information they learn through game narratives better than the information they learn in other contexts, which is a powerful effect to harness for learning about problems with how queer characters are represented in games through modification. The authors also suggested that a game with a "rich narrative" was very effective at reducing cognitive bias overall (Martey et. al., 2017, p. 1647), so it seems that increasing the amount of narrative in a game does have a noticeable impact on a game's learning outcomes. In these terms, I would define *LitM* as a game with "light narrative:" the game focuses primarily on telling the story of the game's queer characters through a text-based narrative that provides a setting focused on queer survival in a hostile world, a narrative that both mirrors the stories often seen in post-apocalyptic video games as well as the actual lived experience of many queer people. I avoided defining too many elements of the characters' backstories other than their sexualities to give players room to expand the prototype into a "rich narrative" game through modification and to allow for exploration of other elements of a character's story through modification, such as their race. Of course, narrative in games can appear through design elements like visuals or audio as well, but I avoided such elements because they are more difficult to modify, and one main contribution of this research is expanding on design research into the kinds of game narratives found in text-based games by demonstrating critical modding of interactive fiction games can be an effective way to explore the stories of queer characters in video games. As such, I provide sample assignment prompts based on these concepts in Appendix B and discuss those prompts in more detail in Chapter 5.

The critical modding approach I propose here also relies on theories of critical game design. When designing a game, it is important to remember that games encourage players to optimize when making decisions: for example, in Greg Costikyan's "I Have No Words and I Must Design," the author argued that "if a game involves any kind of decision making, or trade-offs between different kinds of resources, people will treat these as 'puzzle elements,' trying to devise optimal solutions" (Costikyan, 2002, p.10). As such, players will often try to "solve" the stories that they encounter in games, and I take advantage of this in *LitM* by giving the player gameplay benefits for exploring the narratives of the game's queer characters. Kurt Squire (2011) also suggested that a game does not have to be conceived specifically as an educational game to support learning, as games have an inherent focus on learning through problem solving: "games *already* operate as a medium for learning, whether or not we design educational games… Video games are all about problem solving" (p. 4). Because of this factor, I developed *LitM* with a focus on the stories of queer characters to emphasize the importance of their stories in the game; using educational game design strategies was instead a secondary focus. In addition, Marie-Laure Ryan's "From Narrative Games to Playable Stories" distinguished between two different approaches to narrative design in games: "narrative games," which have a story but are mostly played "to win" (Ryan, 2009 p.46) and "playable stories" (p. 46), in which "the purpose of the player is not to beat the game, but to observe the evolution of the storyworld" (p. 46). *LitM* relies on both of these approaches: it might be described as a playable story because the focus is on a narrative that emerges over the course of a playthrough, but since the game also has clearly defined victory and loss states that center on the game's queer characters, it could be considered a narrative game as well. Both of these design structures are therefore useful to the critical modding approach that I propose here since they encourage people to think about the role of story when designing a modification, and I applied both of them to the design of *LitM.*

Finally, I created this critical modding approach as a model for using modification as a method for learning in my dissertation. While limiting game modification to a classroom educational context is not the sole goal of this project, in Chapter 5 I offer some suggestions about how to implement it effectively in a game design class, mostly through the framework of assignments based on modifying the game. This approach is based on the idea that games can be a meaningful form of criticism: for example, in Patrick Lemeiux and Stephanie Boluk's (2017) *Metagaming: Playing, Competing, Spectating, Cheating, Trading, Making, and Breaking Video Games*, the authors suggested "a critical practice in which playing, making, and thinking about videogames occur within the same act" (p. 2). I apply this notion to educational settings by suggesting that one use for my game in a digital design classroom could involve playing, discussing, and modifying *LitM* as a series of assignments focused on portrayals of queer characters in video games. Eric Champion's (2012) collection *Game Mods: Design, Theory, and Criticism* also examines the notion of modifying games as a form of critical practice, and in his essay "Mod Mod Glorious Mod," he argued that "designing a game mod is one of the few fields of criticism where one can design and test an alternative to the current offering" (p. 15). My critical modding approach therefore demonstrates how game modification can be a form of critical practice that could also serve as a learning activity, and to that end, I provide examples of game modification assignments for classroom settings in Appendix B based on *LitM* as a model that other educators can use to incorporate the game into their own classes.

Summary of Future Chapters

Chapter 2: Exploring Problems with Queer Narratives Through Critical Modding

Chapter 2 lays out the pedagogical framework informing my dissertation and game, building a theoretical justification for using video game modification as an educational tool for exploring problems with representation. Video games have been used in education for quite some time, but in Chapter 2 I expand upon their ability to deal with critical issues like problematic portrayals of queer characters in games. I also analyze the usefulness of interactive fiction as a genre for educational games, especially since such games have less gameplay complexity than other genres and often deal with topics like queer representation. Finally, I look at the ways in which modification is useful as an educational strategy and the ways in which it has been used in the classroom in the past. The focus of Chapter 2 is therefore on contextualizing the value of video games as educational tools and demonstrating that the critical modding approach I lay out in this dissertation could be useful for dealing with concepts like queer representation. While my dissertation does not focus exclusively on the use of games in classroom settings, I discuss that topic in Chapter 2 because most educational methods are developed for classroom contexts; as such, framing my approach in such a manner provides a justification for its use in other learning situations.

Chapter 2 also provides a framework for using modification of video games to teach critical game design concepts that is grounded in established pedagogical methods related to gameplay modification, both in the classroom and outside of it. I suggest that video games allow users to put their critical ideas into practice in an interactive environment through modification, an effect that can be particularly valuable in educational contexts related to representation. I establish this framework specifically in the context of using modification of video games as a method of exploring portrayals of queer characters in games, but I also suggest that it could applied to different critical contexts so that the usefulness of modifying video games as an educational method can be leveraged to teach people about other problematic issues in video games as well, such as the ways in which video games depict race. The ability of video games to address critical concepts such as queer representation is a notion I explore more fully in Chapter

3, and the use of these pedagogical and critical frameworks as a method for creating and modifying a game is a concept that I discuss in Chapter 4.

Chapter 3: Modification and Portrayals of Queer Characters in Mainstream and Indie Games

In Chapter 3 I examine queer representation in video games to demonstrate examples of problematic portrayals of queer characters in mainstream games and more thoughtful queer portrayals in indie games. I also discuss how those examples could be applied to the design of an easily modifiable game that helps players explore those concepts. I examine the way that interaction has been framed in specific games to suggest that certain problems related to how queer characters have been depicted, such as a lack of agency, are particularly common in mainstream games. I also lay out ways to address problematic issues related to queer representation in video games through the more effective portrayals of queerness found in indie games. As such, I analyze examples from specific mainstream video games such as Bethesda Game Studios' *Fallout 4* (2015) and Bioware's *Dragon Age: Inquisition* (2014) to lay out common problems with portrayals of queerness in mainstream games. I also discuss more thoughtful depictions of queer chararacters in indie games such as Technocrat Game's *Technobabylon* (2015), Squinky's *36 Questions* (2015), and Dreamfeel's *Curtain* (2014) to provide examples of ways that games can deal with queer representation in a serious manner.

In Chapter 3 I also discuss how fan created modifications of popular mainstream games are often aimed at addressing problems with the way queer characters are depicted in those games. I look at mods like "Simply Gay Letters" (boringvlln, 2019) that add small amounts of queer content to Bethesda Softworks' *The Elder Scrolls V: Skyrim* (2011), mods like "Boibomb – A Femboi CBBE Preset" (bxbblegumbxtch, 2016) that alter the starting sequence of *Fallout 4*

(Bethesda Softworks, 2015) to include queer characters, and major overhauls like Robert Yang's *Radiator 1* (2015), a queer "total conversion mod" that completely alters Valve's (2006) *Half Life 2: Episode 2* into a story about queer characters. In many ways, Chapter 3 is a kind of critical companion to Chapter 2: in Chapter 2 I suggest that video games are powerful learning tools, and in Chapter 3 I argue that that video games can provide a useful way to explore different critical perspectives. These notions are then fully explored in Chapter 4 through discussion of an example game that I built myself, *LitM.*

Chapter 4: Narrative, Design, and Modification

Chapter 4 applies the ideas developed throughout the previous chapters of this dissertation to game design. In this chapter I discuss the narrative design framework informing *LitM*, focusing on how ideas developed in works such as Ian Bogost's (2007) *Persuasive Games* and Bonnie Ruberg's (2019) *Video Games Have Always Been Queer* helped me construct the game's story and queer characters. I also present my game design document to demonstrate the design work that went into *LitM*, offering it as a model for other scholars who are interested in creating an interactive fiction game that deals with representation in video games. Finally, I argue for the value of design documentation as a form of scholarly practice that highlights the actual design work that goes into the creation of a digital artifact.

The focus of Chapter 4 is on putting many of the theoretical concepts described in previous chapters into practice, so it also demonstrates how many of the ideas explored in Chapters 2 and 3 can be built into the design of a digital object. I argue that educators and critics can put theoretical ideas into practice directly through the design of digital objects that support those ideas; I also suggest that an easily modifiable game built in a free, simple engine like Inky is valuable because it enables people to explore such concepts through game design even if they do not have the skills to create their own game from scratch. My goal in Chapter 4 is to support the arguments I made in previous chapters with a very specific example: the game prototype that I created. Finally, Chapter 4 demonstrates how I put those arguments into practice through narrative design and radical game design in *LitM.* The prototype is also designed to support those arguments in the sense that it serves as an example of a game that can challenge common problematic tropes related to queer characters in video games, but since the game does not contain direct critical commentary or game design discussions, Chapter 4 serves as an important meta-discussion surrounding the digital object itself.

Chapter 5: Life in the Megapocalypse and Critical Modding: Design Principles and Educational Practices

The conclusion of this project both reviews the key design principles I discovered during this project and suggests further directions for research based on the concepts I outlined in previous chapters. I first offer a post-mortem discussion surrounding the creation of *LitM* for those considering taking a similar type of approach in the future. I focus on the creation process of the game itself and the key design principles that I learned about creating a game about queer characters. My aim with this post-mortem is to provide lessons that can aid other scholars in the creation of interactive content that has a radical goal. I also offer a brief model for using critical modding in the classroom through modification-related assignments, along with sample assignments that can be found in Appendix B of this dissertation. I then propose a framework for building an entire class around the concepts outlined in the dissertation and game, looking at how such a class might fight into the large scope of a humanities or game-design program; I also discuss how this approach might be adapted to deal with other kinds of representation, such as race or gender. Overall, in the conclusion I argue that video game modification can be a useful approach for education, criticism, and representation in almost any context and challenge teachers and scholars to find new ways to employ the new digital tools available to us.

Appendices

Appendix A contains instructions for downloading Ink and *LitM*, as well as other technical documentation related to the game prototype. Appendix B provides some sample assignments for classroom applications of the game that are discussed in Chapter 5. Appendix C provides selected source code for *LitM*, which can also be downloaded as an Ink file for use in the Inky editor from my website.

Conclusion: Critical Modding and Life in the Megapocalypse

I have named the approach I describe in this dissertation critical modding because it is a modification-based game design strategy that aims to help people explore problems with representation of marginalized groups through playing and modifying games. The future chapters of this project that I outlined above are designed to both describe my approach and to justify the methods that I suggest here. This approach is pedagogical in that people will learn more about the problems with representation that they see in games through modding, though I suggest that this approach could be embedding in a number of different disciplinary contexts such that it does not need to be restricted to a particular discipline. I use queer representation as a basis for this exploration in this dissertation because games have often struggled to portrayal queer characters effectively, though I argue that this approach could be extended to other marginalized groups that have been portrayed poorly in video games. I suggest that a digital object that is designed both for modification and for exploration of a particular kind of representation is an effective tool for facilitating the methods that I outline here.

As I mentioned earlier, *LitM* is an attempt to build these concepts into an easily modifiable digital object that others can expand upon. The game prototype can function as a standalone piece on its own, but since it is built with my critical modding framework in mind, it also serves as a kind of "digital chapter" or companion piece to this project, though the code and narrative structure makes it significantly longer than a traditional dissertation chapter. That being said, there is not necessarily a specific point during the written project at which readers should stop reading and play the game instead: it is possible to play it before reading, between chapters, or after reading the entire written project. Since much of Chapter 4 focuses on documenting the design of the game prototype, however, I often recommend playing *LitM* before reading that chapter since it offers some additional context for the chapter; similarly, I also suggest reading that chapter before modifying the game to get an understanding of the design considerations that went into it. Overall, though, the game prototype serves as both an example of my approach and as a tool that others can build upon to explore queer representation through modification.

CHAPTER 2: EXPLORING PROBLEMS WITH QUEER NARRATIVES IN GAMES THROUGH CRITICAL MODDING

In the previous chapter, I laid out the overall goals of this project: in the written portion I provide a method for using video game modification to explore problems with representation in games that I call critical modding, and the digital game prototype I created, *Life in the Megapocalypse (LitM),* is an interactive fiction game prototype that people can modify to address issues with queer representation in video games with also building their digital design skills. Many mainstream games either exclude marginalized voices entirely or portray them very poorly, and portrayals of queer characters in games have often been particularly problematic; in addition, while video games have gained acceptance as tools that people can use to learn various skills, there are still concerns about whether games should be used to address certain topics. Because of these issues, it might seem counterintuitive to use the approach I proposed in Chapter 1: if games struggle to portray marginalized identities well and there is skepticism about their use as learning tools, then why use video game modification to learn the skills needed to explore issues related to queer representation in games? I will discuss the problematic ways in which games have portrayed queer characters and how critical modding might be a potential solution to those narrative issues in Chapter 3, but in this chapter I examine the ways in which video game modification can help people learn the digital design skills that are necessary for addressing problems with queer narratives in video games at the level of code.

A cursory review of literature surrounding learning and games reveals countless ways that games have been used at almost every education level and in almost every discipline. Some educators go further than just using games in the classroom, creating modified versions of popular games to ensure that they meet specific learning goals. Modification is a particularly useful approach because it allows people to tailor an existing game to address issues that the game originally did not deal with; furthermore, I focus on modification in this chapter to build upon Chang's (2017) notion that "the challenge of new, different, alternative, even radical games requires a thoughtful reconfiguring of more than just screen, pixel, interface, content, and controller" (p. 17). I argue that critical modding is an approach that allows people to learn how to address narrative issues in games at the level of code, one that is important given the problems with queer characters that are often found in mainstream games. To this end, I look at modification-based approaches to learning that ask people to create their own video game mods, a method that allows them to learn computer programming and digital design skills while creating their own narrative content for a game. Modification based approaches are less common than other approaches to games and learning, however, and usually focus more on learning digital design skills and less on narrative issues in games: my critical modding approach therefore aims to help establish a framework for using modification as a method that allows people to build the computational and programming literacies needed to address problems with representation in video games at the level of code while also being aware of the narrative issues that need to be addressed as well, a concept which I expand on further in the next chapter of this project.

My main goal in this chapter is to address these major threads of research on education and games in the context of critical modding to argue that learning to create video game mods allows people to address issues with how queer characters are depicted in video games directly through queer game design. I first review concerns about game-based learning and highlight the effectiveness of games for exploring problematic narratives in video games, such as those that often arise when games include queer content. After that, I examine how game genres impact learning and suggest that choice-based interactive fiction games are ideal for exploring narrative issues such as problems with depictions of queer characters in games. I then analyze how people use video game modification for learning, noting that one of the most common approaches is educators modifying popular games for use in specific courses or for teaching interdisciplinary skills. More importantly, I also look at how people can learn by modifying games themselves, an approach that helps people build the game design and coding skills that are necessary for critical modding. Connecting these threads of research, I argue for developing specialized, easily modifiable games that address certain topics, such as *LitM*, as tools for learning; I also argue for asking people to build mods of such games themselves to build digital and narrative design skills, suggesting that the combination of approaches gives people both a framework for exploring narrative issues in video games and the skills necessary to do so. I close the chapter with a short discussion about using this strategy to address issues with portrayals of queer characters in games. Overall, in this chapter I argue that interactive fiction works are very well-suited to dealing with problematic narratives in games. I also suggest that modification of interactive fiction games is an effective way to learn the digital design skills necessary to explore narrative issues in games, arguing that such games are a particularly useful medium for critical modding because of their focus on representation at both a textual and programmatic level.

Considerations About Using Games as Learning Tools: Transference, Accessibility, and Gameplay Complexity

Several important considerations arise when using games as learning tools. One main concern about video game-based learning is whether games can achieve transference, which is the ability for people to apply skills they learned in an educational setting within another environment. This skepticism arises from the idea that that the skills learned from playing a video game might not be transferable in the way that skills learned through traditional classroom instruction are: a person who plays an video game might only be learning how to play that game and not learning skills they could use more broadly. This concern is why many games that are explicitly educational, such as the “Traffic Mania” and “Parking Mania” minigames found on the *Cool Math Games* (1997) website for children, rely on real world settings: these games make the application of the skills learned through the game obvious to aid in transference. Such games do not actually teach children how to drive a car, but the real-world setting contextualizes what the students learn from the game so that they can see how the information could be used to complete an everyday task. Learning games that are aimed at simulating such tasks are not as useful for exploring narrative issues such as problems with queerness in mainstream video games, however, and limiting games to real world settings to aid in transference is not always necessary. Hopkins and Roberts (2015) argued that “[basing] the value of educational games on the basis of ‘real-world’ connections is… unduly restrictive” (p. 227) because that outlook ignores the many other potential ways that games can be used for learning. While recreating real world tasks in the educational games described above is a useful approach for aiding in transference of math skills, one of the most powerful elements of games is that they allow players to act on hypothetical settings that they cannot otherwise experience in real life: as Halberstam (2017) argued, “play, in many gaming contexts, both releases players from the restrictions of everyday reality but also encourages them to think in terms of parallel worlds [and] extended realities” (p. 187). This ability to escape the restrictions of the real-world and to explore other potential realities is especially useful in queer games because, as Chang (2017) suggested, such games are about “heterogeneity of play, imagining different, even radical game narratives” (p. 15). Queer games often focus on different modes of gameplay and narrative than those found in mainstream games, and therefore queer games often model quite different kinds of realities than those found in other games. These qualities mean that games can allow players to explore what things would be like in a world where queer people receive the narrative focus rather than straight ones, the scenario I developed in *LitM.* I also suggest that critical modding of such a game allows for transference in that modification helps people learn digital design skills that could be useful in many scenarios, such as the skills needed to modify games with problematic narratives.

When games are used for learning, another major consideration is accessibility: older, low-cost games without significant system requirements, such as the web-based *Cool Math Games* (1997) described above, are often used to the concern that students might not have access to up-to-date devices that are needed to run newer games. An example of this consideration would be when Gee (2004) suggested that “poor children and teenagers do play video games, even if they have to find a computer or game console at school, in a library or community center, or at a friend’s house” (p. 10). Gee was arguing that accessibility might not be an issue for people without computers at home since those people might be able to access such devices elsewhere; however, his argument assumes that all games take the same amount of time to learn and that people without access to devices at home still have the necessary time to learn how to play them. Considerations about gameplay complexity are therefore often not accounted for in discussions about accessibility: the expectation is that young people will be able to play quite complex games games, which ignores the fact some games can require a considerable amount of time to learn. This assumption is a problem because games that can be quite time-consuming to learn, such as those from the turn-based strategy genre, are often used for learning with the expectation that game complexity will not be an issue for students: for example, both Pagnotti and Russell (2012) and Wainwright (2014) used *Civilization IV* (Firaxis Games, 2005) in two different studies to create lessons for college world history courses. In both cases, *Civilization IV* (Firaxis Games, 2005) was used even though *Civilization V* (Firaxis Games, 2010) had already been released, the idea being that the older game was cheaper and easier to run on a variety of computers and would therefore be more accessible for students: however, the difficulty of learning the game was not a significant consideration. The approaches taken in each study were solid, but these successes may also have been because the students involved had regular computer access at their colleges and plenty of time to learn the systems of the game. In contrast, someone who only has access to a computer for short periods of time at a friend's house might struggle to learn *Civilization IV* (Firaxis Games, 2005), as a typical game scenario can take dozens of hours to complete, creating a significant barrier to learning. Using complex game genres as learning tools might therefore cause people to spend much of their time trying to learn how to play the game, suggesting that concerns about complexity are important to any approach that uses games as educational tools. Such considerations also raise the question of what game genres might be appropriate for dealing with problematic narratives in gaming, as a time-consuming, challenging genre like turn-based strategy might not be well suited to addressing such issues.

Given the considerations outlined above, I suggest that interactive fiction games are an especially useful tool for exploring problematic narratives in games. Interactive fiction games have quite a bit of flexibility in terms of subject matter because they are frequently text-based, allowing them to portray both real-world and highly fictionalized situations without considerations about graphical fidelity, gameplay mechanics, and other elements that come in to play with more complex genres, such as turn-based strategy games. Such games are also easy to learn even for people with less experience playing games: they typically distill gameplay down to entering simple text commands or selecting choices from a menu of options, which are both systems that are straightforward in terms of gameplay. Interactive fiction games are also ideal for learning from a cost and system compatibility perspective: many interactive fiction games are cheap or free and they usually run in a web browser, eliminating some of the accessibility concerns mentioned above. Most importantly, however, interactive fiction allows players to focus on making choices and considering potential narrative outcomes those choices might have in the context of the game's story, such that much of the gameplay in interactive fiction games centers on narrative issues. I therefore argue that a simple, text-based design is useful for a game that centers on queer characters, which is why I chose the interactive fiction genre for *LitM* and used it for the critical modding approach I developed here. In the next section, I examine some key elements of interactive fiction games to outline their effectiveness for exploring problematic narratives in games.

Interactive Fiction and Queer Narratives

Like the use of games for learning overall, interactive fiction has been used for learning a variety of different topics. Writing and language-learning focused learning are common ways that interactive fiction is used: for example, Lundberg and Lyons (2018) used it to create a Twine game that “served both as an alternative to a traditional grammar assignment and as a pre-writing activity and prompt for an informal, creative writing assignment” (p. 101) in an English as a Second Language class. The usage of interactive fiction in such courses is likely due to it being well suited to writing assignments, as most interactive fiction is text-based, and it usually requires a good deal of writing to accommodate the multiple story paths and outcomes that are typical in such games. Improving literacy is another common educational use for interactive fiction games: Pereria (2013) argued that it can be effectively used for language learning (p. 33 - 34) and Hovious (2014) suggested that it is a useful tool for encouraging literacy in younger students, as interactive fiction usually involves reading a great deal of text as a user plays through a story. Interactive fiction has also been used to bridge writing and game design in college courses: Skains (2019) used Twine to create a class called “Playable Fiction” in a game design program, describing it as “a 12-week taught undergraduate module… designed as an exercise in experimental writing” (p. 5). In addition, Donley and I (2019) used Inky, the same interactive fiction engine I built *LitM* in, to introduce students to basic game programming concepts while also helping them build their digital design and narrative construction skills in an introductory level interactive media course. These examples suggest that interactive fiction games are flexible as educational tools and point to their ability to allow people to explore a variety of concepts, and Lester (2018) summed up their educational benefits by arguing “IF games jibe well with what we hope our learners will discover in any case: more-frequent, longer-lasting, world-forgetting, playful immersion in texts" (p. 263). This kind of flexibility and immersion is particularly useful when addressing problematic issues in video games through critical modding: learners can explore a fictional world and adapt its narrative to address those issues. Such factors are also why I built *LitM* as a framework that others can modify and expand upon: the game focuses on queer characters and I provide sample assignments in Chapter 5 that encourages people to expand the stories of those characters through modding and queer game design. These elements make interactive fiction an especially suitable genre for the critical modding approach I lay out in this project, though it could certainly be used for other game genres as well.

A key point to consider when using interactive fiction for learning is that the genre encompasses a variety of forms that feature different modes of user interaction, story structure, and game mechanics. While it is difficult to classify all the different forms that interactive fiction can take, the two most well-known styles are parser-based interaction fiction and choice-based interactive fiction. Parser-based interactive fiction involves a user typing commands into the game to indicate their actions: "NORTH" might move the player in that direction, or "GET KEY" might allow the player to pick up a key. An example of such a game is the classic text adventure game *Zork* (Infocom, 1980), and many early interactive fiction games took this form. While this gameplay structure is not too difficult to learn, one challenge with parser-based interactive fiction is the design of the parser systems themselves, as they may require users to type commands using a specific syntax to achieve in-game tasks: for example, "GET KEY" might be a valid command in one game, while another game might only accept "TAKE KEY," with “GET KEY” producing an error message instead. While there are often similarities in syntax between different parser systems and some are set up to accept variations on commands, this design means that players must learn the syntax that works for a particular parser system to play games made in that system. Choice-based interactive fiction, on the other hand, requires users to make choices from a menu of options as they play: for example, users may be offered choices like “GO NORTH” or “GET THE KEY” and interact by clicking on the option they desire. This is the structure I used in *LitM*, and many other modern interactive fiction games, such as Inkle’s *80 Days* (2014), use this approach as well. The benefit of such a design is that the options available to the player are usually unambiguous: the player can simply select the options they desire during gameplay when prompted. In addition, choice-based interactive fiction games are often built using hypertext systems, which makes them familiar to anyone who has used the Internet. While parser-based interactive fiction has useful educational applications, I suggest that choice-based interactive fiction is especially useful for critical modding because it allows learners to focus on how their decisions affect the narrative and game world without having to learn the syntax of a parser system. This design quality is also why I built *LitM* using Inky, a choice-based interactive fiction engine.

Interactive fiction is also well suited to dealing with narrative problems related to marginalized groups in games since such issues are frequently addressed by interactive fiction works. Many interactive fiction works were designed to raise awareness and help people learn more about a particular underrepresented group through gameplay: *Depression Quest* (Quinn, 2013), for example, was described by the creator as an attempt to “to show other sufferers of depression that they are not alone in their feelings, and to illustrate to people who may not understand the illness the depths of what it can do to people" (depressionquest.com), which indicates that the game is aimed at portraying a character who has depression in a way that helps people learn more about the subject through a participatory experience. The gameplay mechanics of *Depression Quest* (Quinn, 2013) also require the player to interact with the concept of depression directly, as various user interface elements indicate the main character's current mental health status and much of the game's story revolves around changing that status for the better or worse. This gameplay structure does not necessarily mean that mental health and queer identity in video games should be handled the same way, but like mental health, queer identity has significant impact on people's lives: as such, Keeling (2014) suggests considering the "historical, sociocultural, conceptual phenomena that currently shape our realities in deep and profound ways, such as race, gender, class, citizenship, and ability… to be mutually constitutive with sexuality and with media and information technologies" (p. 153). Such an argument does not mean that issues like race, mental health, and queer identity in games should simply be conflated, but the kind of enacted representation found in an interactive fiction game like *Depression Quest* (Quinn, 2013) offers a useful model for exploring almost any narrative issue related a marginalized group in video games because it suggests a participatory approach to play that is informed by the actual lived experience of that identity. Such an approach therefore invites players to consider not only how a character in a game is depicted, but how the player's choices contribute to the portrayal of the character, a notion I explore further in Chapter 3 in relationship to how games often assume a heteronormative identity for players.

If interactive fiction lends itself well to dealing with narrative issues related to marginalized identities, it follows that the genre could also be useful for learning about problems with depictions of queer characters in games. Just as *Depression Quest* (Quinn, 2013) lets people have an enacted experience of mental health issues through gameplay, interactive fiction works that focus on issues with queer representation could allow people to engage more directly with queer experience by enacting it in a participatory manner through play. That being said, Nakamura's (1995) concept of identity tourism, in which people take on other identities in online spaces, is important to consider in relationship to the participatory elements of games: while discussing people who roleplay as Asian male characters in online games, the author points out that "the choice to enact oneself as a samurai warrior… constitutes a form of identity tourism which allows a player to appropriate an Asian racial identity without any of the risks associated with being a racial minority in real life." This factor is why I did not have the player of *LitM* take on the role of one of the game's queer characters directly: instead, the player takes on advisory role in the game's narrative and interacts with the queer characters mainly through conversation. Though identity tourism is certainly a concern with regards to people taking on the roles of queer characters as well and should be considered in any discussion of participatory learning that involves depictions of people with marginalized identities, I argue that interactive fiction as a genre serves well a starting point for learning the digital design skills needed for critical modding, as the genre inherently encourages active participation in narrative construction, such that modification of interactive fiction games encourages modders to address such issues in a participatory way as well. Furthermore, I provide sample assignments that focus on modding *LitM* to make changes to the game's queer characters in Appendix B as a framework to specifically encourage deeper engagement with queer representation through critical modding and to help reduce the potential problem of identity tourism.

Another useful quality of interactive fiction for critical modding is that, as a genre, interactive fiction games often have players participate in the construction of queer identity either through their own choices or through incidental narrative elements, making them ideal for the critical modding approach that I argue for in this project. There have been many well-known interactive fiction authors whose works require the player to engage with queerness directly: for example, Shelley Jackson's *Patchwork Girl* (1995), one of the most famous works of interactive fiction, has the player experience a romantic relationship between a fictionalized version of Mary Shelley and the female monster character from the original novel. Anna Anthropy's *Dys4ia* (2012) is another good example of such a work, as the game was created as an expression of the creator's experiences with gender dysphoria, transitioning, and taking estrogen: the game is designed such that the player engages with and enacts that experience through gameplay. Pieces like Aaron Reed's (2009) *Blue Lacuna* also tie engagement with queerness to the player's choices, with a particular character's queerness being dependent on decisions the player made throughout the game. Even when a game does not ask players to engage with queerness directly, however, incidental queer presence is common in the narratives of interactive fiction works as well: pieces like Reed's (2013) *18 Cadence* and Technocrat Games' (2015) *Technobablylon* feature queer characters throughout their stories, and Inkle's *The Intercept* (2018) features a gay protagonist in World War II whose sexuality simply serves as a characterization detail. I mostly relied on this kind of incidental queer presence in *LitM* in the sense that most of the characters have relatively brief backstories, though the assignments I discuss in Chapter 5 encourage players to expand upon those narratives through critical modding, encouraging a more enacted form of engagement that relies on a player's own experience.

Finally, one useful quality of queer interactive fiction for critical modding is that the community surrounding it has seen involvement from people with all kinds of identities, such as nonbinary nonwhite creators like Squinky, which helps to encourage people of all kinds to engage with narrative issues in games through queer game design. There are countless queer interactive fiction games that are based on the life experiences of their creators that use novel approaches to gameplay intended to queer normative gaming conventions, include queer characters and stories, or that simply embrace queerness in some way: in fact, the authors of *Rainbow Arcade* used the term "the queer games movement" (Ruberg et al., 2019, p. 40) to describe the prevalence of such interactive fiction games throughout the 2010s. Notably, many of these games used "tools like Twine… that [allow] people to make interactive stories without needing any knowledge of computer code" (Ruberg et al., 2019, p. 40), making it easier for creators to learn the skills necessary for getting involved in queer game design. In fact, such notions are why I built my game in an easy-to-use engine like Inky: however, creating an Ink game does involve more coding than some other interactive fiction tools, making it more useful for helping people learn the programming skills necessary to create video game modifications. In fact, the ways in which people can expand queer narratives in interactive fiction works by making code changes to the game is central to my critical modding approach. As such, building a user's programming skills though modding is important, a notion which I explore in the next section of this chapter.

Building Digital Design Skills Through Critical Modding

As I outlined in the previous section, choice-based interactive fiction games are useful for learning because they are much easier to learn how to play in comparison to other games. This quality is also particularly useful for the critical modding approach I lay out in this project: it is easier to learn to modify choice-based interactive fiction games than other kinds of games since modifying them typically does not require advanced programming skills, allowing people to learn such skills in a simpler environment. Learning to modify games has rarely been considered as a method to address problematic narrative issues in gaming, however, even though it is a method that bridges game design and narrative design, as well as many other skills, into an approach that would be quite appropriate for exploring such issues. Modified games have been used for learning in the past in general, though, and one of the most common approaches is an educator modifying an existing game for use in the classroom; similar approaches have also been used outside the classroom as well to teach interdisciplinary skills. Situations in which students have been asked to modify games themselves as a learning experience are rarer, though research on such approaches and on self-driven learning in modding communities suggests that modding could be useful as a way to address problems in games. Overall, in this section I suggest that modifying a game to address a specific issue is a good starting point; more importantly, having people modify games themselves should be employed as a learning strategy more frequently, as such an approach allows players to learn the digital design skills needed to change a game's narrative and gameplay. I also argue that both approaches could be particularly useful for addressing problematic narratives in video games and demonstrate how both inform the critical modding method that I lay out in this project.

When discussing potential uses of modified games for learning, many argue that people should modify games for use in a particular discipline, and this approach seems to be one of the most common ways that modification of games is used in educational settings. As an example of this idea, Brown et al. (2018) argued that people could “modify or adapt an existing game or software and incorporate educational content to achieve science learning” (p. 7) and they highlighted several successful modified games that were used for such a purpose in science higher education. An example of such a mod is Smaldone et al.’s (2016) *Polycraft World*, a *Minecraft* (Mojang, 2011) mod that was created to teach chemistry at the college level. The creators designed the mod to link the chemistry concepts they wanted students to learn to the game’s mechanics, constructing the game’s systems such that “learning more science makes the game more fun and the player more powerful in the game” (Smaldone et al, 2016, p. 99). The primary benefit of this approach is therefore that one can modify a game so that learning the game’s mechanics is synonymous with learning about a particular concept. This factor is also why I built *LitM* myself: creating it myself allowed me to design the game prototype such that it functions as both a framework for exploring the stories of queer characters and as a starting point for learning the digital design skills needed to modify games. This structure encourages players to explore the game’s queer narratives so that players naturally engage with queerness as they play, and the game’s easily modifiable nature allows people to expand the game’s narrative in their own ways if they wish through critical modding, a strategy I explore more fully in Chapter 4. As such, I suggest that building easily expandable games to address particular topics is a good way to create tools for critical modding and provide *LitM* as an example of what such a game would look like.

While I argue for designing games and mods to address specific issues, it is important to acknowledge a key issue related to that approach: digital content creation often requires a level of procedural literacy and programming knowledge that not all people have. When discussing the idea of educators making games for use in the classroom, Lester (2018) argued that coming up with ideas for a learning game is not the issue, but that “translating these ideas into digital format is, for nearly all of us, the stuff of magic” (p. 263): the challenge is that most people are not trained in the skills necessary to create or modify games. As such, a key element to consider when designing mods for learning purposes is choosing games that have accessible engines and that are easy to modify. This factor is why *Minecraft* (Mojang, 2011) is one of the most frequently chosen games for making educational mods. Modding tools for *Minecraft* (Mojang, 2011) are available through a free software development kit and the tools are well-documented on the game’s official wiki, making mod creation for the game relatively easy in comparison to other games. Because of its ease of use, *Minecraft* (Mojang, 2011) mods have been created for use in many disciplines: the *Polycraft World* (Smaldone et al., 2016) mod described earlier, for example, is a mod of the game for science higher education. As another example, Al-Washmi et al. (2014) created a mod for *Minecraft* (Mojang, 2011) that was “developed specifically to get children to play collaboratively while engaging in mathematics” (p. 11): they chose the game because of the “ease with which it can be used to allow investigation of educational ideas” (p. 11). While mods for *Minecraft* (Mojang, 2011) are common because it is easy to modify, text-based interactive fiction games might be even better suited for such purposes: as Lester (2018) argued, “text-based interactive fiction… may offer a medium in which [educators] can not only play or modify, but create” (p. 263) because it is relatively easy to make content for interactive fiction games. Such games also place more of a focus on narrative elements than on gameplay mechanics and level design, making them useful for exploring concepts like problems with queer representation in games that are often most apparent in a game’s narrative elements. These factors are why I built *LitM* in the Inky scripting engine: Inky is designed for creating interactive fiction and naturally emphasizes storytelling due to its text-heavy presentation. The engine also has many qualities in common with *Minecraft* (Mojang, 2011) that make it useful: Inky is freely available, has a great deal of documentation, and the developers provide a sample game to help users get started with building content.

While creating mods that focus on specific topics as described above is a promising strategy, I suggest that learning to modify games is also useful because it is an interdisciplinary practice, making it appropriate for addressing problematic narratives in games of all kinds. The value of interdisciplinary work and collaboration through modding is often emphasized when people discuss mods: Elkins (2015) argued that games naturally encourage people with different skills sets to collaborate, and "this collaboration also extends outside of the game environment as gamers work together to create mods" (p. 59). Beyond just offering opportunities for such work, mods have also been analyzed for their ability to teach people to collaborate more effectively: for example, Wendel et al. (2013) created a "collaborative multiplayer game specifically for enhancing teamwork and collaboration" (p. 571) and tested it with 28 participants in groups of 4 (p. 576), suggesting that it taught team-building skills very well. The usefulness of mods in teaching interdisciplinary skills like information literacy is another commonly discussed element: for example, Clyde and Thomas (2008) created a mod called *Benevolent Blue,* an "information literacy focused modification of the video game Half-life 2" (p. 367) for use as a learning tool that was intended for school libraries. While digital design skills are not learned the same way as those skills and problems with depictions of queer characters in games is a different topic than those mentioned above, the interdisciplinarity inherent to modding is similarly useful for exploring issues like queer representation in games because so many other topics intersect with those issues. A game designed to help people explore queerness therefore should accommodate many different kinds of content and must be useful in many different scenarios, and this factor is why I designed *LitM* as an interdisciplinary learning tool that can be used in a variety of situations, as it can be used in almost any educational context that deals with queerness.

Most of the research on using game modding for learning relates to having educators create game mods for use in their classes, but there is much less discussion about having people modify games themselves as a way to learn. That being said, the research that has been conducted on that approach seems promising, especially with regards to learning digital design skills: El-Nasr and Smith (2006) conducted a study with students in high school and college level game design courses that used modding as an instructional strategy and suggested that their approach helped the students learn "lighting, texturing, and limitations of real-time rendering and effects" (p. 14) as well as "project management, scheduling, and iterative development" (p. 14), all of which are particularly important skills that are important for creating digital content like video game modifications. Robertson and Good (2005) also looked at how students used game creation toolkit for *Neverwinter Nights* (Obsidian Entertainment, 2002) to create mods, suggesting that modding "[allows] children to experiment with a type of fiction with which they may have little experience, namely interactive fiction" (p. 62). They conducted a workshop in which 10 students created mods using the toolset, finding that "in spite of the complexity of the games-creation task, all of the young people managed to create a game to their satisfaction" (Robertson and Good, 2005, p. 65), which suggests that interactive fiction is a useful starting point for introducing people to modding. In addition, Yucel et al. (2006) held a "Gaming for

Girls" workshop aimed at teaching middle and high school girls IT skills through modding; they found that the workshop was particularly effective at increasing students' skills at sustained reasoning, managing complexity, and testing solutions (p. 153) after conducting pre and post surveys with the participants. The workshop addressed narrative issues in gaming as well, as the creators made "specific considerations unique to the audience, including gender imbalance in the video game industry, hyper-sexualized character design, and social perceptions of women and games" (Yucel et al., 2006, p. 147), suggesting that engagement with problematic narratives in games through modification in an educational environment could be effective. Given such examples, I argue that a simpler set of tools, such as those offered by the Ink scripting engine, might accomplish such goals even more effectively. That being said, these examples focus on having students learn computer science and game design skills, and there is little research about how students might use such skills to engage with a more narrative-focused concept such as depictions of queer characters in games; the studies also involved contexts in which the students were motivated to create modifications because of the classes or workshops they were taking. As such, I provide an outline for a digital design class focused on programming and representation in games in Chapter 5, as well as assignments in Appendix B that focus on queer representation and could be used in such a class.

As noted above, there is some research on the kinds of skills people learn through modding, but many of these discussions focus on examining how mods could be used in formal educational settings, while most people learn to mod through participation in more informal settings. People have explored how people learn through modding in such settings to some extent, however: for example, Ryu and Jeong (2019) conducted a study that "[explored] game players' participation in learning of fan programming or *modding* in an online community" (p. 1353). In a review of various other sources that discuss modding from an educational perspective, they argued that "modding has been placed within the contexts of participatory learning" (Ryu and Jeong, 2019, p. 1355) in that it requires users to actively create new content for games rather than simply playing them. Ryu and Jeong (2018) also conducted a survey of modders from the CivFanatics fan site, a popular modding site for the *Civilization* series: they found that participants "spontaneously tried to learn how to mod in the online community" (p. 1357) even though many of them came from educational and professional backgrounds that gave them little prior experience with programming. This study highlights an important quality about participatory learning through modding that can be useful for exploring problematic narrative issues in video games: modding provides a useful introduction to the kinds of programming skills which are necessary for addressing problems such as issues with queer narratives in games at the level of code. While the examples above obviously involve learners who were motivated to create content for their favorite game franchises, creating mods for a queer game might help people build their programming skills while also encouraging them explore issues related to queerness: I explore that approach more fully in the fourth chapter of this project with examples based on the game prototype I created, *LitM*, but one of my overall goals with this project is to expand on modification-based approaches by creating a critical modding and game design framework that people can use to address narrative issues like problems with queer representation in video games.

It is also worth considering why people engage in modding in these informal settings, as those who would be interested in creating queer mods for a game like *LitM* might be driven by the same kinds of interests as other modders. Subramanian (2012) conducted dissertation research on the CivFanatics modding community and found three distinct “participatory identities” (p. 116) for people who used the site: “mod-savvy” players who “[like] to design funny, creative missions in game” (p. 118), “improv gamer” players who “used the editor in-game to improve [their] single playing strategies” (p. 118), and “hobbyist programmer” players who are “modding to learn to program” (p. 118). Importantly, two of the three identities, the “improv gamer” and “hobbyist programmer” identities, were using modding to learn, whether it was learning to play the game better or learning to code. This study suggests that a desire to learn is a key element of why many people begin modding, though people who mod history-focused strategy games like the *Civilization* games might be interested in learning different things than someone who mods a queer interactive fiction game and are likely motivated to mod for very different reasons. It is more likely that people who create mods for a game like *LitM* would fall into the “mod savvy” or “hobbyist programmer” categories as such people might be interested in creating new queer content for the game or in improving their programming skills by creating a mod for an interactive fiction game. Overall, considering the motivations of people who engage in modding is critical, as those motivations will inform how people create modifications for games. This factor why I provided extensive code comments aimed at “mod savvy” and “hobbyist programmer” modders in my game prototype: the comments are designed to accommodate both people who want to expand the game’s queer narratives and people who want to hone their coding skills through critical modding.

Finally, it is worth considering the factors that make choice-based interactive fiction a particularly useful kind of game design for critical modding. I outlined the benefits of using choice-based interactive fiction instead of parser-based interactive fiction for learning earlier in this chapter, and many of those benefits also apply here as well: modifying parser-based interactive fiction is more challenging than modifying choice-based interactive fiction because it requires learning both the syntax of the parser that the game uses as well as the syntax of the programming language used to create the game, which poses a dual challenge for those who are unfamiliar with such games or languages. Because of such issues, creating a mod for a game built in a parser-based system like Inform 7, such as *Bronze* (Short, 2006), could be daunting for an inexperienced creator. When discussing the idea of digital content creation in a blog post, Emily Short (2017), the creator of *Bronze* (2006) and a well-known author of interactive fiction, summed up the challenges of writing parser-based interactive fiction: “If you want to write something quickly and easily, you should not write parser IF.” In contrast, modifying a choice-based interactive fiction game built in an engine like Inky, such as *The Intercept* (Inkle, 2018), only requires learning the programming language that the game is based on, which is not too challenging as long as the language itself is simple. Inky is also a relatively easy engine to learn: Donley and I (2019) successfully introduced the Inky engine to students over an eight-week period in an introductory digital design course with few issues. This factor is why I specifically why I built *LitM* as a choice-based interactive fiction game: users who are interested in critical modding can quickly understand how gameplay mechanics work in the game’s code. Those users can then learn how to easily expand the game’s portrayal of queerness by modifying its narrative and gameplay systems, both of which are built around problems with queerness that are common in mainstream video games.

Conclusion: Addressing Problems with Queer Representation in Games Through Critical Modding

In this chapter, I argued that choice-based interactive fiction games are ideal tools for exploring problematic narratives in video games because such games are easy to learn and place a focus on narrative. I have also argued modification of games is an effective method for addressing such concepts: people can create game mods that are specifically tailored to addressing issues with a game's narrative, such as problems with its depiction of queer characters, because interactive fiction games are not too difficult to modify, allowing people to build their narrative and digital design skills at the same time. Such skills are valuable because of the issues with portrayals of queer characters found in almost any form of media: as Halberstam (2017) argues, "since the world as we know it was not designed for queer subjects, then queer subjects have to hack straight narratives and insert their own algorithms" (p. 187). Games, especially interactive fiction games, are ideal for this kind of exploration because of this ability to hack and modify them, allowing users to insert their own queer narrative content into them. I do believe that such an approach could be applied to a variety of issues related to representation, however, and games are notorious for having many of those: problematic portrayals of race, gender, and sexuality have plagued gaming culture for a long time, and while games as a whole have gotten better at handling those issues, many mainstream franchises still have problems with inclusion. That being said, I do not aim to survey all of these issues in this project or to suggest that the particulars of this approach could meaningfully address all such issues surrounding games: issues with queer representation are not the same as problems with race or gender, and those problems are complex themselves, requiring solutions that are tailored to the particulars of each issue. I do think that this approach is adaptable enough in general to address many kinds of problematic narrative issues in games, however, and would encourage people to consider creating game like the one I have created to address other issues as well; to that end, I expand upon the design of my game prototype in Chapter 4 and discuss how critical modding could be applied to other issues in Chapter 5.

While I have discussed some of the general narrative issues that games have with queerness and how code and modification could address them in this chapter, in the next chapter I turn to problematic portrayals of queer characters in mainstream game more specifically to provide examples of the kinds of narrative issues that the approach I proposed in this chapter could help people explore. I also discuss the more positive ways that indie games have engaged with queerness, as these games offer suggestions for how game designers can create narrative and gameplay that portrays queer characters more effectively. In addition, I analyze the ways in which modders have addressed issues with queer characters in mainstream games by creating mods that add queer content to such games or by modifying the ways in which a game's narrative and gameplay depict such characters. Such changes are sometimes small, but queer modders have also created mods that entirely change the way that a game's story or narrative work, representing the amount of effort that modders are willing to put in include queer content in their favorite games. Overall, I suggest that the critical modding approach I outline here is a useful method for learning the coding skills necessary to explore such issues, and I argue for the potential of that approach for addressing problems with representation in games in the next chapter of this project.

CHAPTER 3: MODIFICATION AND PORTRAYALS OF QUEER CHARACTERS IN MAINSTREAM AND INDIE GAMES

In the previous chapter, I argued for using modification of choice-based interactive fiction games to learn the digital design skills necessary to modify games, suggesting that the critical modding approach I lay out in this project allows players to explore problematic narratives in games by reorganizing a game's narrative and gameplay elements. In that chapter I suggested that the method I laid out could be adapted to explore problems related to queerness in video games, but primarily focused on how critical modding allows people achieve the programming literacy necessary to address those issues at the level of code. In this chapter, I turn to a problem that modding has often engaged with: the way queer characters are portrayed in video games. Throughout this chapter I examine how mainstream games, indie games, and modification relate to issues with how queer characters are depicted in video games. It seems obvious to note that queerness and games have interacted in controversial ways for quite some time, so I do not intend here to provide a historical overview of issues that have arisen in the past: sources such as the *Queer Game Studies* (Shaw and Ruberg, 2017) collection and *Video Games Have Always Been Queer* (Ruberg, 2019) have covered many issues related to queerness and games and establish that games have many problems with portraying queer characters effectively. Instead, my goal here is to analyze the ways in which modification can help players explore some of the problems surrounding portrayals of queer characters and relationships in mainstream games. I also focus on the lessons that game designers and modders can learn from more effective portrayals of queer characters in indie games.

The relationship between mainstream and indie games is critical to my analysis of queerness and modification in this chapter. I first examine how many of the problems related to portrayals of queer characters in games have been stratified along the lines of popular mainstream games and less popular indie games to demonstrate that mainstream games have typically handled queer characters much more poorly than indie games. Second, I look at more thoughtful examples of queer characters in indie games and how those games are often in dialogue with the problematic portrayals of queer characters found in mainstream games. Finally, I look at how problematic portrayals of queer characters and relationships in mainstream games have been addressed by fan communities through gameplay modifications. I use these examples to suggest that modding is one potential solution to those problems by looking at how queer modders work within existing narrative and gameplay frameworks. After exploring these issues, I briefly discuss a tool that can help to avoid those problems with queerness in games: an easily modifiable game I created, *Life in the Megapocalypse (LitM)*, that is designed for exploring portrayals of queer characters in games through critical modding. Overall, I suggest that considerations about a game's mainstream or indie status influences how both developers and players engage with queerness in games and argue that critical modding is a powerful way for players to engage with and explore issues with queerness found in mainstream games.

Queer Characters in Mainstream Games

While the importance of representing marginalized forms of identity in popular visual media is relatively uncontroversial at this point, portrayals of marginalized identities in mainstream games have frequently been met with resistance, or in some cases, outright hostility. As an example, reactions to characters like Tracer in *Overwatch* (Blizzard Entertainment, 2016), a prominent lesbian character who serves as the one of game's mascots, were generally divided into two groups: "those who applauded the inclusion of a prominent gay character, and those who [were] mad about 'diversity' and 'political correctness' being shoved down their throats" (Feldman, 2016). Inclusion of queer content in games is important despite this resistance, though games have problems related to the structure of the gaming industry that are worth considering when discussing queerness. In this section I analyze portrayals of queer characters in games through the lens of the mainstream / indie divide to examine how those problems relate to a game's mainstream or indie status. I suggest that even though depictions of queer characters in games have become more common in mainstream games, indie games still handle queer characters more effectively because indie games do not engage with queerness simply as a vehicle for profit and popularity as mainstream games often do. To address the issues surrounding queerness in gaming culture, I argue that game critics should highlight problems with the ways that mainstream games portray queer characters and provide examples of such problematic portrayals. In addition, I claim that game scholars should highlight effective depictions of queer characters in indie games to help combat those problematic narratives and provide examples of such games as well.

It is important to acknowledge the position of many games as commercial products when considering how they depict queer characters: mainstream games have become a big business, and many of the considerations that surround such games are driven by profit-based motives. Clark (2017) argued that "anyone who has engaged with video games in the last three decades is familiar with the primary institution that games *have* been deployed successfully in service of: the consumer marketplace" (p. 12). The author pointed out that the categorization of games is often based on elements that relate to the game's status as a commercial product, and I suggest that the terminology surrounding games has likewise mirrored this focus: mainstream and indie have become common descriptors used by fans, journalists, and scholars to refer to different kinds of games, even if those groups use such terms in varying ways. While both terms are relatively broad qualifiers for describing games, mainstream games are almost always described as being mass-market products: for example, Lipkin (2013) argued that "mainstream game development is corporate in nature and capitalist in ethos" (p. 9) and suggested that "mainstream is characterized as emphasizing profit and popularity over creativity and artistry" (p. 9). As such, mainstream games operate much like other forms of corporate media in terms of how they depict queer characters: such games usually only do so if it can bring in more money for the game's creators. It can be even harder to determine what qualifies as an indie game as "indie media is defined by what is not mainstream" (Lipkin, 2013, p. 10): such media is often identified by the qualities it does not have when compared to mainstream media. If this is the case, indie games are those that are driven by artistic motives rather than profit and popularity and are not aimed at mass-market appeal: indie games are therefore more likely to engage with queerness from an aesthetic perspective than other games. In fact, the authors of *Rainbow Arcade* (2019) included a section titled "2010s Queer Indie Game Boom," noting that when several queer indie games "all came out in quick succession, they ushered in the queer games movement" (p. 41), and that such games "were often critiqued as not being real games or 'just' being art games" (*Rainbow Arcade*, 2019, p. 41). Those games were also "markedly personal in nature [and] communicated the designers' own lived experiences" (*Rainbow Arcade*, 2019, p. 41): these queer indie games were developed with intentions that were very different from the design goals that drive mainstream games, and queer characters in such games were often informed by the life experiences of the designers. Overall, I suggest that this difference in design goals is one of main reasons that portrayals of queer characters in mainstream and indie games differ so widely.

Because of the issues noted above, it is important to acknowledge that the definitions that underpin terms like mainstream and indie are nebulous at best. Despite how vague the terms are, they are still useful descriptors for games in terms of their engagement with money and artistry, and as Lipkin (2013) suggested, "given the dominant economics, development and publishing standards that account for the large percentages of attention and sales in contemporary game markets, it is worth providing some stability to the terminology" (p. 9). This factor means that when analyzing how mainstream and indie games interact with each other in terms of portrayals of queer characters, considering whether a particular game is a mainstream or indie game is useful because that consideration could have an impact on the kinds of queer content a player might expect to see in each type of game. The terms mainstream and indie are therefore useful in that they allow for some assumptions about the kinds of queer characters that each type of game might contain: in general, mainstream games are less likely to contain queer characters and indie games are more likely to contain such characters, and mainstream games are also more likely to engage with queerness in a shallow, profit-driven way than indie games. While such an assumption is not a guarantee and cannot describe all the kinds of queer content found in mainstream and indie games, the terms allow for an examination of how depictions of queer characters in games are often related to a game's commercial status. The terms also allow for an analysis of the kinds of problems with queerness that mainstream games often run into and the ways that indie games have avoided them.

A lack of queer characters and a focus on straight male players is one of the most common issues that comes up with mainstream games. Ruberg (2019) argued that “video games have notoriously underrepresented LGBTQ people” (p. 3), and in many popular game franchises, the issues is simply that there are simply no queer characters at all. This lack of queer content in many mainstream games suggests that the developers assume that their players are straight, or at least that those players do not want to encounter queer characters. Clark (2017) also noted that “the status quo, to nobody’s surprise, is that games have seldom been made by or for queers, or even with queers in mind” (p. 2): instead, mainstream games are typically aimed at the straight male players who are seen as the target audience for such games and queer players are ignored altogether. Portrayals of queer characters have therefore been rare in mainstream games, and if queer characters are included such games, such content is an afterthought at best. A good example of such an approach is found in *The Elder Scrolls* series: most games in the series rarely engage with sexuality at all. Any relationships the games do include are almost exclusively heterosexual, are almost never involve the player, and tend to not have any particular relevance to the overall story. The most recent game in the series, *The Elder Scrolls V: Skyrim* (Bethesda Softworks, 2011) goes a bit farther than other games in the series, however, in that it allows the player character to marry some of the game’s NPC characters. More interestingly, the game’s NPC characters will marry the player’s character regardless of either character’s gender, which allows for queer relationships in the game. Most of the narrative elements of those marriages are interchangeable, however, and most of the marriable NPC characters use the same lines of dialogue no matter what the player’s gender is with no acknowledgement of any difference between a straight and gay marriage. This approach is certainly flexible, but (Lo, 2016) criticized approaches like this for being shallow as they tend to flatten any potential diversity in queer relationships: in fact, there is very little narrative difference between the game's queer relationships and straight ones at all. While approaches like the one used in *The Elder Scrolls V: Skyrim* (Bethesda Softworks, 2011) are becoming more common, most mainstream games simply choose not to include queer characters at all. Because so much of the popular discourse surrounding games focuses on mainstream games, this issue means that queer players are often excluded from that discourse in the sense that they are unlikely to find queer characters and narratives that they can relate to in mainstream games, and when they do, those narratives are often very shallow or brief.

While a lack of queer characters in mainstream games has been one of the most common problems with them throughout the history of gaming, it is worth noting that queer content in games has become more common over the years, especially in roleplaying games or other story-driven genres that allow for significant character interaction. As such, queer content that was once only the purview of indie games has started to be included in mainstream games, particularly if romance between the player and another character is a possibility within the game. Even when queer romance is included, however, mainstream games still struggle to portray it effectively: for example, Bagnall (2017) suggested that mainstream games are "bound up with naturalized, patriarchal constructions of gender and sexuality" (p. 135) in the sense that they take a heteronormative stance toward those elements and portray queer characters as outside of those norms. Sundén (2009) also pointed out what often happens when queer players play mainstream games: "for queer gamers, sexuality comes into play in ways that make visible the cultural norms of the 'ideal' player – *a player who is at least symbolically male and straight"* (p. 3). This comment highlights the fact that queer players of mainstream games are usually thought of as falling outside of the game's "default" (i.e. straight male) player identity, a fact that queer players are also constantly reminded of as they play many popular games. This issue means that when queer characters and romance elements are included in mainstream games, such games treat "queerness [as] just a different twist on non-queer heterosexuality" (Lauteria, 2012, p. 2): queer romance options are treated as just another selection from the menu of choice offered by the game rather than a meaningful story element that is handled differently than the other romances. Overall, when mainstream games include queer characters, frequently their "solution to difference and diversity… is token inclusion or flattened representation, and what it means to be inclusive is still narrow, often stereotyping, and problematic" (Chang, 2017, p. 18): as such, the inclusion of queer characters in such games is not always positive. That being said, I do not suggest that mainstream games simply avoid queer representation altogether: as noted earlier, that approach causes its own problems, and those problems have likely lead to many of the issues with depictions of queer characters in mainstream games in the first place. Because of that problem, however, it is worth examining some specific examples of mainstream game franchises that have included queer characters, but have struggled to portray queerness well.

A mainstream game series that has frequently included queer characters is the *Dragon Age* series, which were developed by Bioware and published by Electronic Arts, one of the largest publishers in the gaming industry. The franchise was very commercially successful: for example, *Dragon Age: Inquisition* (Bioware, 2014), the final game in the *Dragon Age* series, was Bioware's largest selling launch in history as of 2015 (Electronic Arts Inc, 2015), an impressive feat considering the popularity of their games in general. The series has also included quite a bit of queer content: for example, Østby (2017) wrote a PhD dissertation discussing queerness in the *Dragon Age* games, demonstrating the breadth of queer content in the series. It is important to note, however, that the developers of the series have publicly framed the inclusion of queer characters in the games in terms of consumer considerations, an approach which resembles the descriptions of the profit-driven motives of mainstream game developers discussed earlier. One of the developers of the series, David Gaider (2013), asked the following question with regards to queer players at a GDC talk about sex and representation in Bioware games: “is there an untapped market out there full of gamers waiting to be appealed to?” One of his goals with the talk was to highlight the potential of queer players as this untapped market, an approach which suggests that gestures toward inclusion in the *Dragon Age* series were motivated at least partly by the potential profit and popularity to be gained by including queer characters. In this sense, then, queer content in these games is meant to draw queer players in and get them to buy the game, and such goals might be more important than portraying queer characters effectively.

As noted earlier, romance plots are one of the most common methods that mainstream games use to include queer content: for example, in the *Dragon Age* games, queer content mostly comes in the form of LGBTQ identifying characters that the player can form relationships with. These romantic portrayals are also where issues with queer characters in the games become very visible: as noted earlier, many mainstream games present queerness as simply an alternative to straightness. The way the *Dragon Age* games handle player sexuality is a good example of this: Greer (2013) argued that in most of the games “the potential for non-heterosexual identification or role play is primarily constructed and validated on the same terms as playing straight” (p. 5) because a queer player’s sexual identity is expressed through romantic interactions that are very similar to the kinds of interactions that a straight player engages in. Choosing a romantic partner in the *Dragon Age* games is therefore the player's primary means for expressing a particular sexual identity, with the games typically including an LGBTQ character or two as gesture toward inclusion: in *Dragon Age: Origins* (Bioware, 2009), for example, most of the player's potential romantic partners are straight, while two, Leilana and Zevran, are bisexual and react the same way to romantic advances of any gender. These issues suggest that mainstream games have an issue with queer romance at the very least, if not with other kinds of queer content: providing a variety of romance options is not necessarily a bad thing, but reducing queer identification down to the selection of one of the game's few LGBTQ characters as a romantic partner is a particularly reductive choice in terms of narrative and game design. This design choice is problematic in that portrays queerness as being only about the selection of a partner, ignoring potential identities such as demisexuality or asexuality as well as the larger complexities of queer identity. This approach is not often used in indie games, which tend to build queerness into the game's narrative and mechanics in a variety of ways and typically avoid having the player choose a romantic partner in favor of including queer characters and relationships that have more depth, such as the way *Curtain* (Dreamfeel, 2014) explores the complexities of a lesbian relationship by having the player take on the role of one of the two characters.

Finally, a common problem in mainstream games is the inclusion of visual elements that undermine the game's engagement with queerness. I previously discussed this problem in relationship to the *Fallout* series (Howard 2019), particularly in regards to how *Fallout 4* (Bethesda Softworks, 2015) offers opportunities for same-sex relationships, but undermines them by forcing the player to start the game in a heterosexual marriage. Visual elements throughout the game’s introduction, such as the main character’s spouse holding their child in a 1950’s style home, continually reinforce the importance of the player’s straight marriage to the game’s overall story: even though the player might not want their character to be in such a relationship, the game’s visuals serve as frequent reminders of their previous marriage. These elements could simply be ignored and written off as poor visual and storytelling choices by the developers, but Lauteria (2018) argued that “to be a ‘player’ is to hold a subject position in the gaming experience, bound up in symbolic and representational processes” (p. 38): players are involved in making meaning when they play a game, and that factor might contribute to the discomfort felt by queer players when a game forces them to define their character’s sexuality in a particular way. Visuals that reinforce that straightness serve to intensify the problem because video games are a visual medium: a few lines of dialogue might be ignored or easily removed by modders, but visuals that continually suggest straightness, such as the repeated visual depictions of the main character’s wedding ring in *Fallout 4* (Bethesda Softworks, 2015) that serve as a reminder of straight marriage, are harder to ignore and are not easily modified without more advanced video game development skills. This factor might also be why queer indie games such as *Curtain* (Dreamfeel, 2015) make a conscious effort to include visual representations of queerness, such as markers of the main character’s lesbian relationship: such visuals can reinforce in-game dialogue or story elements to create a more queer experience overall.

Queer Characters in Indie Games

While categorizing the kinds of problematic portrayals of queer characters found in mainstream games is not too challenging because those portrayals often suffer from similar issues, indie games create a wide variety of queer experiences that I could not hope to fully encapsulate here. Despite that diversity, some queer indie games seem to be in dialogue with the approaches that mainstream games take in that they directly address some of the problems with portrayals of queer characters that I described above. That being said, I would not suggest that these games represent particular archetypes of queerness or narrative benchmarks for queer characters in games since that suggestion would undercut the very diversity of approaches that queer games take: while mainstream games often suffer from specific types of problems in terms of how they portray queer characters, indie games represent queerness in a much broader variety of ways. This variety is another reason why I have focused on specific indie games here instead of large groups of games: the games I highlight here provide examples of the kinds of design strategies I employ in my own game, *LitM*. While I will discuss my own game in Chapter 4, I will highlight some of those design strategies here in addition to the ways that these games address common problems with depictions of queer characters in mainstream games.

One way that indie games handle queer characters more effectively than mainstream games is simply by including queer characters and narratives more frequently throughout a game's narrative instead of avoiding them entirely. *Technobablyon* (Technocrat Games, 2015) provides a good example of this approach, as the game presents a fair number of queer relationships throughout the game world. The game's story focuses on several queer characters, but their queerness is not necessarily a central element of the game's story or mechanics; it is simply a part of each character's overall identity. In terms of design, the game is a point-and-click adventure game in which the player controls various protagonists navigating a futuristic cyberpunk style city while trying to solve a mystery; one of those protagonists, Dr. Max Lao, is trans, which is a significant part of the character's overall identity and backstory in the game.

The game presents the character's queer identity as being fully accepted in the futuristic setting, however, and the other queer characters who appear are likewise accepted within the game world as well: their queerness is important in terms of who they are, and the characters are not marginalized within the narrative or treated as simply alternatives to straight characters. Sinclair (2017) argues that "*Technobablyon* presents a world where gay relationships and queer identities are normative:" by using this design strategy, the game presents queer characters as simply part of the setting, an approach I use in my own game as well. Like many other indie adventure games, *Technobablylon* (Technocrat Games, 2015) is designed with Adventure Game Studio, an open source game engine that is commonly used to create point and click games that resemble "old school" LucasArts adventure games. Using an engine specifically created for adventure games allowed the developers to include a significant number of audiovisual assets and to make the game relatively lengthy compared to other indie games since the engine saved them significant amounts of time during development. Such an approach is common within indie game development, and reliance on an existing game engine is an approach that I also employed when creating my own game for very similar reasons: I used the Inky engine to create *LitM* because it is specifically designed for creating choice-based interactive fiction games, a factor I discussed in more detail in Chapter 2.

Another approach to queer indie game design that demonstrates the diversity found within that space is basing much of the game on the creator's own personal experience, allowing such portrayals to provide a depth that goes beyond straightforward romance plots or simply the inclusion of characters who identify as queer. *36 Questions* (Squinky, 2015) is a good demonstration of that approach: the game is clearly referencing the kinds of romance plots that often appear in mainstream games, but the developer primarily focuses on portraying queerness by describing events from their own life throughout the game's story. The game focuses on a discussion between the player and Squinky that is based on a scientific study that outlines 36 questions that are aimed at finding out whether or not two people are romantically compatible (Aron et al, 1997). The game takes place during the apocalypse – certainly something that the developer has not experienced themselves – but most of the dialogue is based on Squinky's in-game persona relating their own experiences with being a nonbinary queer game developer and the difficulties that came along with that identity. At one point, for example, the character commented:

> My goal was to have 1000 true fans — that, they said, was about how many dedicated supporters an artist needed to be able to make a sustainable living off of their work. I didn't want anything more than that, because as we all saw, if a queer non-white non-guy like myself gets *too* popular, well, the trolls come out and harass the living daylights out of you. (Squnky, 2015).

Such comments in the game feel quite authentic, as they should: narratives are often at their strongest when they are based on lived experience that is only slightly fictionalized, and the above dialogue is clearly a description of the developer's own experiences with creating games. While this kind of reliance on personal experience might be considered a common quality of queer indie games, queer indie games that rely on such an approach are always quite different from one another because the experiences of queer developers themselves are likewise quite different from one another's. In terms of game design, the game does not use a preexisting game engine, but is instead delivered through a text-based web interface, placing a focus on the dialogue between the player and Squinky rather than on the game's audiovisual elements. This element highlights another design strategy often used in queer games: because audiovisual elements take a significant amount of time to create, especially if the visuals rely on 3D assets, queer indie developers often focus their development efforts on other areas such as the game's narrative. While I did use a preexisting game engine to create my game, I used a web-based interface that is similar to the one used in *36 Questions* because it made including minor visual elements in the game relatively simple and allowed me to focus more on creating systems that procedurally generate a story. That being said, audiovisual elements are found in queer games as well, and they are often used to portray elements of queer experience that are difficult to get at via a textual narrative or dialogue system.

When queer games include audiovisual elements as prominent parts of game design, they often employ those elements in a way that reflects queer experience metaphorically rather than placing a focus on more literal elements like the selection of an in-game romantic partner with a particular gender identity as found in games like the *Dragon Age* series. An example of this approach can be found in *Curtain* (Dreamfeel, 2015), which is a first-person narrative account of two queer women in a destructive, abusive relationship. The player plays as one of the women and does not get the opportunity to choose between other romantic partners or relationships at any point in the game. The game features a 3D environment that the player can move about in, and interacting with objects provides bits of narrative that expand on the relationship between the two characters: for example, touching a telephone presents a voicemail from the narrator's family while her partner comments: "Oh My God, delete that. You're not really going to get in touch with them are you?" (Dreamfeel, 2015). While the game takes place in a 3D world, its visuals are blurry and indistinct: it is possible to make out the environment and the objects within it, but they are often out of focus. The visuals depict the apartment that the two characters live in, and various objects demonstrate that the women are in a punk band together. The narrative advances as the player steps into the shower at various points in the story, with the shower itself functioning as a kind of portal that transports the player between two nearly identical versions of the apartment that represent different chronological points in the relationship between the characters. Color shifts in the environment between each apartment indicate changes in the emotional relationship between the two women, often moving between hot colors that indicate emotions like anger or jealousy and cold colors that indicate anxiety or depression. The visuals are much simpler than what might be found in a mainstream game, and they are focused on giving impressions of emotional states rather than offering exacting visual fidelity, providing an example of how queer indie games often employ audiovisual elements to serve less as selling points for a game and more as aesthetic devices that enhance the game's portrayal of queerness. Again, such a design strategy is not necessarily archetypal since queer games use a broad variety of visuals: *Curtain* (Dreamfeel, 2015) is quite visually distinct from most other queer games, and in fact is distinct from most 3D games in general. Instead, the game represents a common approach to visuals in queer games while also demonstrating the diversity of visual aesthetics found in those games. While I did not on implement 3D environments and visuals into my game during this project, I built the game in the Inky scripting engine because it can be integrated with the popular Unity game engine, offering me the option to create 3D environments and gameplay elements in the future.

Overall, it is clear that indie games tend to handle queer characters better than mainstream games, and it is likely that they can do so because they are free of the pressures of producing games that are commercially successful. That being said, I do not suggest here that it is impossible for mainstream games to include queer characters in more effective ways, nor would I argue that all indie games represent queer characters well; in addition, I do not argue that mainstream games should simply abandon the portrayal of queer characters altogether or that such content is antithetical to mainstream success. Instead, I suggest that the creators of mainstream games should be acutely aware of the impact of queer content in their games given the position of mainstream games in the overall gaming market and that designers can look to queer indie games as examples of how to meaningfully include queer characters in all kinds of games. More importantly, as Allen (2014) argued, "we have to employ a plurality of tactics to bring down the monolith:" pushing for better queer representation in mainstream games is one approach to doing so, and highlighting good queer representation in indie games is another. Since mainstream games will likely continue to dominate overall discussions surround games, however, a third approach could involve critical modding: players could modify games themselves to include queer content and gameplay elements. I explore that approach more fully in the next section.

Queer Mods for Mainstream Games

Since much of the discussion about problematic depictions of queer characters in games centers on mainstream games due to their market share, one solution to the problems with queer characters in those games could involve designers making their games more easily modifiable so that fans can create better queer content themselves. On the surface, this solution seems to have a lot of potential, as it offers fans a way to comment on problematic elements in their favorite games even more directly than through critical discourse: critical modding allows fans to actually change the game itself so that it portrays queerness more effectively. Modding is a solution I have explored elsewhere with regards to the *Fallout* franchise (Howard, 2019), and the notion of adding queer content to mainstream games through mods works within a tradition of queer modding that is quite longstanding: for example, Lauteria (2012) highlighted queer mods for the *Fallout* series as well and suggested that they "yield anti-normative play spaces through an orientation to queer sexualities and genders," acting as a form of resistance to the portrayals of queer characters in the games. Queer modding is therefore a tradition worth examining as it relates to mainstream games, as many of the issues that I raised in regards to mainstream games are addressed directly by such mods, and the queer modding tradition offers one of the most prominent examples of what critical modding looks like. I argue that the issues revealed by my analysis in this section demonstrate a need for an indie game with a narrative focused on queer characters that can be used for critical modding, such as the game prototype I created as part of this project, *LitM*.

As I discussed in the previous section, portrayals of queer characters in mainstream games are often either nonexistent or problematic, so it is worth considering why are players interested in modifying such games: why not simply play and modify some of the indie games mentioned earlier that handle queer characters in a more thoughtful way? It would be difficult to find out exactly what the motives of individual modders are without directly surveying the people who play and modify such games, but since most queer indie games do not offer modding tools directly to players, a lack of access might be one answer: there are simply not many modding tools available for indue games, which makes them much more difficult to modify. On the other hand, many mainstream game franchises have a long history of offering modding tools to their fans, which encourages queer modders to work with such games. There are also longstanding and very active modding communities surrounding popular mainstream game franchises that have consistently offered mod support, offering a larger audience for mods that are designed for those games. In addition, since modders are necessarily a subset of the entire player base for a particular game, there will typically be more mods for mainstream games with mass-market appeal than for indie games. Modding communities for mainstream games can therefore grow so large that some people build upon the creations of other modders, a process that Morshirnia and Walker (2007) called “reciprocal innovation” (p. 6), which expands the modding community for such games even further. These factors encourage queer modders to create content for mainstream franchises despite their problematic portrayals of queer characters: both the tools for mod creation and an audience for those mods are available for such games. Frameworks such as Steam Workshop have begun to change this dynamic by offering a space for modding communities for a wider range of games, but the lack of easily available modding tools for indie games and the smaller player base for such games means that modding communities for such games will always be smaller than those for larger mainstream games. *Dream Daddy* (Game Grumps, 2017), a fairly well-known indie game that focuses on queer relationships, provides a good example of this issue, as a search for mods for the game on both the Steam Workshop and on the popular modding website *NexusMods* returns no results despite the fact that the game is made in Unity, a very popular and free to use game engine. Overall, since queer modders tend to focus primarily on mainstream games, my analysis in this section will therefore primarily focus on mods for such games as well.

While there are many kinds of mods for mainstream games, queer mods occupy a specific space in relationship to mainstream games when compared to other kinds of mods. Since mainstream games do not typically focus on portrayals of queer characters or depict them poorly, queer mods for those games often operate as a kind of resistance to a particular game: they might add queer content where none existed before, or they might modify existing portrayals of queer characters to make them less problematic. Queer mods for mainstream games could therefore be considered a form of "Queergaming," a practice Chang (2017) described as:

> borrowing, appropriating, and repurposing… players take existing game titles, characters, stories, and worlds and queer them, remediate them to refashion and reimagine not only content and play but their very own relationship to ostensibly non-queer games and communities. Queergaming then is a response to the lack of queer representation, narratives, game play, games worlds, and gaming communities. (p. 20)

The concept of queergaming in relationship to modding suggests that players have a desire to add queer characters to games that do not have them or to modify games that have problematic depictions of queer characters. Queer mods for mainstream games also fit into the definition of queergaming because Chang (2017) argued that "queergaming expands what it means to create, consume, and play to include game mods" (p. 20): as such, creating queer mods for games could be described as "queermodding." Despite the problems with queer characters often found in popular mainstream games, there is a great deal of queer mod content for such games and looking at queer mods for such games can provide insight into why they chose these particular games in the first place. More importantly, looking at the mods created by those communities helps to demonstrate the kinds of content that queer modders find problematic since that content is what is most often addressed by queer mods.

Queer mods for mainstream games are quite diverse, but while these mods take a variety of approaches to altering queer portrayals in mainstream games, they often address similar kinds of content. In fact, some authors have even tried to categorize queer mods in terms of the types of game content that they modify: for example, Welch (2018) suggested distinguishing between queer mods that are "cosmetic" mods and queer mods that focus more on "gameplay:"

> We might imagine video game modifications on a coordinate plane of queer possibility. One axis represents cosmetic enhancements, or the swapping and appropriation of assets in the game in order to evoke or augment the player's relationship to that game… The other axis is mechanical alterations. These mods change the rules and systems of gameplay to allow for emergent gameplay possibilities.

This distinction is particularly useful in that it provides a framework for discussing the different ways that queer mods engage with games, but more importantly, it suggests that analysis of queer mods should go further than simply pointing out that queer mods exist for certain popular games: that analysis should also consider what kinds of content the mod engages with and the ways in which it does so. While I will not necessarily consider the distinction between cosmetic mods and gameplay mods in this analysis, I do pay attention to the kinds of content that queer mods engage with, as doing so demonstrates the kinds of critical modding strategies that queer modders use to address issues with portrayals of queer characters in mainstream games. I also analyze how much content the mods alter, as some of them only change small elements of their respective games, while others represent large overhauls or even approach what could essentially be considered almost entirely new games.

Many queer mods engage with the various problems with queer portrayals in mainstream games that I discussed in the previous section. Queer mods often focus on simply adding queer content to games that have little such content already: some queer mods for Bethesda's *The Elder Scrolls* franchise, for example, exist mostly to address the series' lack of engagement with queer sexuality, which is one of the most common problems with portrayals of queer characters in mainstream games. Bethesda has made modding tools available for almost every game in *The Elder Scrolls* series, however, and many queer mods for *The Elder Scrolls V: Skryim* (Bethesda Softworks, 2011) aim at adding more narrative significance to queerness in the game world by adding elements other than romanceable NPC characters. Mods like "Simply Gay Letters" (boringvlln, 2020) are a good example of this approach: it is a relatively small mod in terms of content, as it simply adds letters to the game world. Those letters discuss queer relationships between characters, and the mod's creator says the letters are intended to "simply [bring] a bit of immersion here and there" (boringvlln, 2020), making queer characters feel like a more significant part of the game world but not making any drastic changes to the game's mechanical or visual feel. The approach also resembles the way that games like *Technobablylon* (Technocrat Games, 2015) handle queer characters by including them throughout the game's narrative rather than as a part of a specific plotline or as a romance mechanic. Mods like "Simply Gay Letters" therefore address a common problem in mainstream games that I discussed earlier: such games rarely depict queer characters at all, so these mods fix that by adding in small amounts of queer content throughout the game. As such, one approach to critical modding could simply be using it for inclusion by creating mods that add new content to a game.

A lack of engagement with queer characters is a problem in mainstream games, but as noted in earlier, some mainstream game franchises have depicted queer characters more frequently over the years. Their portrayals of queer characters are not always positive, however, leading modders to address that issue as well by creating mods that change the way a game handles those characters. The *Fallout* franchise is a good example of this situation: the games are frequently modded because the developers have offered a set of modding tools for many games in the series shortly after their release, the games engage with sexuality fairly often for a mainstream gaming franchise, and they typically do so from a very heteronormative and binary perspective despite their apparent flexibility. *Fallout 4* (Bethesda Softworks, 2015) is a good example of this problem: as mentioned earlier, the game allows the player to be in queer relationships, but the game's visuals continually reinforce the player's previous straight marriage. These visuals cannot be avoided without the use of mods, and queer mods for the *Fallout* series therefore usually focus on modifying the game's visuals to place more of an emphasis on queerness. I previously highlighted examples of queer mods for *Fallout 4* (Bethesda Softworks, 2015) that alter the game's introduction so its visuals can depict queer characters in an attempt to address this issue (Howard, 2019). These mods often make significant visual changes to the game: for example, one of these mods, "Same Sex Couples and LGBT Families" (Sensia, 2020), alters the game so the player can be in a same-sex marriage throughout the introduction, which changes the game's visuals a fair amount, and another one, "Boibomb – A

Femboi CBBE Preset" (bxbblegumbxtch, 2016) has a dramatic impact on the game's visuals by enabling a "femboy" appearance for the game's male characters, which changes the entire visual presentation of the game. These mods are clear examples of queer modding as a form of resistance to the visual heteronormativity found in mainstream games, and many queer mods for the *Fallout* games exist as a direct response to the problems with the ways in which the games visually depict queer characters and relationships. While I focus less on creating visual overhaul mods in this project because those mods are often difficult to create, they do represent another approach to critical modding that could be employed by creators with 3D modeling and video game programming skills.

Queer mods for Bioware's *Dragon Age* series take yet another approach to modifying the portrayals of queer characters in the games that highlight an issue mentioned earlier: romance plots in those games often portray queer romance and relationships in almost the exact same way as straight ones. Both *Dragon Age: Origins* (Bioware, 2009) and *Dragon Age: Inquisition* (Bioware, 2014) featured romanceable characters who have specific preferences in terms of gender, as they will only form a relationship with the player character if the player character's gender identity matches the interests of the NPC. While this model was an effective way of giving those characters a particular sexual identity, the difference between straight and queer relationships was not heavily acknowledged in the game world of either game. In addition, some players found that they were unable to romance a particular NPC in *Dragon Age: Origins* (Bioware, 2009) because the player chose the "wrong" gender and that they were restricted to only romancing the game's programmatically defined queer characters. As such, mods that alter the NPCs preferences so that they are interested in a variety of player genders and that expand the narratives of queer relationships are extremely common and often change the romance mechanics of the games entirely. An example of such a mod is GoesOnGhost's (2018) "Complete Bi Overhaul," a mod for *Dragon Age: Origins* (Bioware, 2009) that alters all the NPCs so that they can be romanced regardless of the player's gender. On the surface, the mod essentially used the same approach to romance that *Dragon Age II* (Bioware, 2011) and *The Elder Scrolls V: Skyrim* (Bethesda Softworks, 2011) used, which could potentially recapitulate the problem of straight and queer relationships being treated the same way; however, the mod creator also engaged in a significant amount of work to alter the narrative of *Dragon Age: Origins* (Bioware, 2009) so that queer relationships were not portrayed the same way as straight ones. The creator created new "banter" dialogue for other NPC characters so that they acknowledge the new queer relationships created by the mod (GoesOnGhost, 2018), and even went so far as to alter dialogue for the romanceable NPCs so that "characters won't misgender the [player]" (GoesOnGhost, 2018). The created noted the dialogue changes enabled by the mod "[apply] to every time the [player's] relationship… is acknowledged" (GoesOnGhost, 2018): all of the NPC characters in the narrative use the appropriate pronouns and gender terminology for the player character's queer relationship even though the dialogue references for those pronouns and story elements did not originally exist. These kinds of queer mods illustrate the lengths to which players will go to alter a mainstream game: queer modders are often willing to create a significant amount of new narrative and dialogue content for games to address problems with the way that mainstream games present queer characters. They also represent an approach to critical modding that is well-supported by the game prototype I created, *LitM:* the game's narrative is designed such that modifying a character's gender and pronouns is easy because those references are stored in variables that can be changed, and such changes are automatically reflected throughout the game's narrative and dialogue because the appropriate variables are used throughout the game.

Finally, some queer modders engage in a complete reconfiguration of a game, altering the original in such totality that the mod itself could essentially be considered an entirely different game. Such mods fall into a larger category of mods referred to as "total conversion mods," which are built with many of the same assets as the original game but represent completely new gameplay experiences from start to finish. The most well-known queer total conversion mod is likely Robert Yang's *Radiator 1* (2015), an episodic series that is based on *Half Life 2: Episode 2* (Valve, 2006). None of the original narrative elements of *Half Life 2: Episode 2* (Valve, 2006) are present in Yang's mod: unlike other mods that alter parts of a game but leave the overall story relatively intact, *Radiator 1* (Yang, 2015) is a totally separate story that does not intersect with the story of the original game at all. The mod completely reimagines the gameplay of the popular first-person shooter as well, transforming it from science fiction action game about a male hero fighting off an alien invasion to a game about "stargazing, gay divorce, and Emily Dickinson" (Yang, 2015). Overall, *Radiator 1* (Yang, 2015) is clearly a significant departure from the original game in terms of gameplay, narrative, characters, and many other elements. Yang continued working on the project for quite a bit of time and eventually remastered the mod in 2015 using an open-source software development kit created by Valve to make it freely available without needing a copy of the original base game, further blurring the line between its status as a *Half Life 2: Episode 2* (Valve, 2006) mod and a completely standalone game. Total conversion mods like *Radiator 1* (Yang, 2015) therefore represent perhaps the most radical form of queer resistance to mainstream games: they require significant development skills and a lasting time investment in order to create, but they so thoroughly change the game that in some cases they become completely divorced from the original game they came from. They also represent the most radical form of critical modding, though I do not explore such an approach too much in this project because such mods require a great deal of video game design skill and I created *LitM* as a tool that can be used to introduce novices to such skills.

Of course, there are many other problematic portrayals of queer characters in mainstream games as well as many other franchises with modding tools: the examples in this section are not meant to be an exhaustive or categorical list of issues and solutions. They are instead meant to illustrate the relationship between depictions of queer characters in mainstream games and modification of those games, a relationship in which modders both work within a hegemony while simultaneously resisting it. In particular, modding problematic portrayals of queer characters in mainstream games is an approach that is double-edged: it allows modders to address the issues in their favorite games, but also enables mainstream game developers to avoid doing the work of addressing those problems themselves, an issue I have discussed in the past (Howard, 2019). This issue is particularly complex, as modders should not stop making queer mods, nor should mainstream games stop portraying queer characters, as both of those approaches would simply make matters worse. It might instead be better to continue using this approach while also addressing the issue from a different angle entirely by designing an indie game from the ground up to focus on portrayals of queer characters and a modifiable structure that can be easily learned. I briefly sketch out that approach to close this chapter by discussing an example game prototype I built with those goals, *LitM.*

Conclusion: The Benefits of a Modifiable Queer Indie Game

The issues I discussed in previous sections raise clear problems: mainstream games do not portray queer characters well, and while smaller studios and indies portray them more effectively, their games are not as popular. Additionally, modification of mainstream games can effectively add queer content to them, but that approach allows mainstream game publishers to avoid the work of making their own portrayals of queer charcaters better, placing such work on communities who are already being marginalized by those portrayals. These are complex problems and addressing them is not simply a matter of a single, one-size-fits-all approach: that being said, exploring potential solutions to these problems is still important, especially since it will likely require multiple approaches to address them all. Documenting the design work that goes into exploring these solutions is therefore critical, and in the next chapter I propose a solution that I expand upon with an example: creating an easily modifiable game with an eye toward the problems with portrayals of queer characters found in mainstream games and making it available for people to freely use for critical modding. This approach is effective because it avoids many of the problems mentioned above: it stays away from the problematic portrayals of queer chracters often found in mainstream games; it does not require financial support to be successful; and it removes the exploitative elements that can come up when mainstream companies leave queer modders to fix problems with representation in games. More importantly, it allows people explore depictions of queer characters through a framework in which the narrative and gameplay norms of the source material have been specifically designed for that purpose and inherently encourages people to experiment with the norms of the game.

As an overall approach, building an easily modifiable queer game also has many significant benefits: first, modification inherently suggests reinterpretation, much in the way that fanfiction, role playing, and other similar fan creation practices do. In fact, Harper (2017) argued that “queer readings, and fan (re)interpretation are nothing new in mass media, and not even really new in video games, particularly in gaming fan culture” (p. 132). Mainstream games are obviously a form of mass media, and I suggest that critical modding is an extension of those practices as they relate to games: modding almost always involves some form of reinterpretation of the inherent norms of the narrative or gameplay rules found in a particular game. This factor also means that modifications that add queer content to mainstream games could be seen as similar to queer readings of mainstream media. Second, a game that deliberately focuses on queer content to begin with would also likely encourage queer mods since modders typically expand on the existing narrative and gameplay content of the game when they make mods. While it would certainly not guarantee that someone will create queer content for the game, building many of a game’s frameworks around queerness encourages any mod for that game to engage with queerness. Third, a game that is deliberately designed for critical modding would encourage players to expand on queer characters in the game since, as mentioned above, modding is already a practice with inherently queer elements in that in encourages reinterpretation of the norms of a particular form of media. Building the game so that players can easily interact with its system is, by extension, a practice that already has inherently queer elements as well. Such a game invites players to change its norms rather than abide by them and even offers players a framework for doing so if they wish.

A major portion of this project involved building a digital prototype of a game based on the design strategies, modifications, and concepts described above called *LitM.* That game is available on my website and sample source code for the game is also provided in Appendix C. The next chapter describes the theoretical approaches informing the narrative and game design of that prototype before presenting the game design document for the game itself. While I do not suggest that *LitM* addresses all the problems outlined in this chapter, I argue that considering issues surrounding portrayals of queer characters in mainstream and indie games was critical to its design, as it allowed me to build the narrative and gameplay mechanics with an eye toward the problems mentioned in this chapter. I also argue that examining that actual design work that went into the digital object is critical, which I will turn to in the next chapter.

CHAPTER 4: NARRATIVE, DESIGN, AND MODIFICATION

This chapter discusses the choice-based interactive fiction game prototype that I designed for this project, *Life in the Megapocalypse (LitM);* as such, it is structured differently than the previous chapters in that it relies less on theoretical discussions and focuses primarily on game design. Because much of the chapter directly discusses the prototype, I recommend playing through it before reading this chapter; sample code from the project can also be found in Appendix C as well. Though this chapter has less discussion of critical scholarship than the last few chapters, some of that discussion is still relevant to the game's design, so I first review the narrative design framework that informs the game prototype to provide context for how it is built, especially those sources related to the theoretical approaches that informed *LitM.* I then present the game design document for *LitM;* it provides a discussion of the game's mechanics, narrative, and systems, as well as an overview of some of its technical elements, its user interface, and other relevant materials. Finally, I briefly discuss the value of providing design documentation as part of digital projects, especially those that focus on modding, as those who are interested in doing so can benefit greatly from an overview of the digital artifact itself. Overall, in this chapter I suggest that documenting the design work that goes into creating a game is important when building a game for critical modding, as a game's design elements have a significant impact on how players engage with it. I also demonstrate the arguments I made in previous chapters through a discussion of the design of *LitM.* Most importantly, I argue that a modification-based framework built around a game designed for critical modding is the most effective method for integrating the concepts discussed in this project into a classroom education setting.

Narrative Design Framework

Before discussing the narrative design framework informing *LitM,* a brief review of the game's story structure is useful. Much like the narratives of many text-based video games, *LitM* relies on individual units of narrative in the form of short chunks of text that the player encounters as they play the game and make decisions, and those units connect to create an overall story for each playthrough of the game. The game takes the form of a choice-driven interactive fiction story, with players selecting choices from a list of options in response to game events. Those choices lead to different outcomes, many of which are driven by chance or determined by previous choices that the player made. As in many such games, players typically need to play *LitM* multiple times to access all the narrative units available in the game since many of the game's story elements are mutually exclusive and cannot be encountered during a single playthrough. This is also true of many of the game's queer narratives, as those stories are tied to in-game characters who many not always survive a particular playthrough of the game. Because of these factors, I designed the game's story around theoretical frameworks that focus on interactivity, procedural rhetoric, and queer game design. As such, I first review the theoretical sources informing *LitM's* narrative design that I relied on when building the game to provide context for the design choices that I discuss in the game design document later in this chapter. Before moving into that framework, however, it is worth noting that not all who play *LitM* will necessarily create modifications for the game; that being said, one primary way I envision the game being used is for modification-based classroom assignments that I discuss in Chapter 5 and provide prompts for in Appendix B. In addition, I also provide much of the source code for the game in Appendix C, as well as a full version of the game on my website. Because of those factors, I designed the game's narrative and other systems with the consideration that a larger portion of the potential player base might modify it in comparison to other indie games, as it is more common for modders to work with mainstream games and indie games usually do not receive as much attention from modding communities.

The relationship between interactivity and game narratives is a common discussion in game design, as players can almost always change a game's story through gameplay interactions. An example of such a discussion is when Ryan (2009) suggested that "the combination of narrativity and interactivity oscillates between two forms: the *narrative game,* in which narrative meaning is subordinated to the player's actions, and the *playable story,* in which the player's actions are subordinated to narrative meaning" (p. 45). She portrayed the two forms as similar, but with differing goals: she argued that "in a narrativized game the player purses the kind of goal that people may form in everyday life or in their fantasies: goals such as saving the world from invaders and rescuing people from danger" (Ryan, 2009 p. 46). In comparison, a playable story is a form in which "the purpose of the player is not to beat the game, but to observe the evolution of the storyworld (Ryan, 2009, p. 46). Ryan's position suggests that these forms are mutually exclusive, but I suggest that interactive fiction games like *LitM* can also bring the two together: it is a narrative game in the sense that the game's goal is saving as many characters on the team as possible, but it is a playable story in the sense that doing so provides a more complete narrative experience of the game's story. More importantly, because the game is designed to be easily modifiable, *LitM* encourages modders who change the game to potentially emphasize one of those two goals more significantly. *LitM* demonstrates that some game narratives can operate within a spectrum between the narrative game and the playable story, which allows them to take advantage of elements of both approaches to achieve particular needs.

LitM also takes advantage of structural affordances of games that encourage players to experiment with narrative and gameplay modification. Bogost (2006) noted that "common gameplay in works of the same genre makes it possible to develop new games based on the code written for existing games" (p. 55), highlighting early examples of fan produced content for text-based games like *Zork* (p. 55). He suggested that this practice is common in gaming culture, so much so that "developers often create tools to allow nonprogrammers to build game components" (Bogost, 2006, p. 55 – 56) with the expectation that players will want access to such tools. While discussing such tools, however, he argued that "the capabilities of game engines have been limited to visual and physical experience, rather than emotional and interpersonal experience" (Bogost, 2006, p. 64): in particular, many video game creation tools such as the popular Unity engine focus on enabling the kind of visually fantastic, violence-oriented gameplay common in mainstream games. More recent tools like Ink, Twine, and Ren'Py have focused on providing story creation functionality to people interested in creating games that emphasize narrative content, however, and Bogost's attention to the importance of the material functions of a game engine is one reason why I chose to develop *LitM* in the Inky engine. The engine affords both story and gameplay creation, which I emphasize in my game, while placing less focus on visual and tactile elements. This aspect of Ink means that *LitM's* story is easy to modify for those interested in doing so.

LitM similarly relies on the natural tendency of video games to teach players about how the game operates as they play. While I discuss educational theories and games more fully in

Chapter 2, it is worth briefly summarizing the situation here as well. Bogost (2007) argued that "current theories of videogames as educational tools mirror our views on classrooms… we can roughly split our perspectives on videogame learning into behaviorist and constructivist modalities" (p. 236). Bogost (2007) here references the common argument that most educational theories can be mapped to one of those two traditional approaches: behaviorist approaches that focus on learning through reinforcement of information or constructivist approaches that focus on learning through interaction with an environment (p. 235). More importantly, however, Bogost rejected those notions and suggested that they do not adequately describe the kinds of learning that takes place in games. Instead, he argued that "videogames do not just offer situated meaning and embodied experiences of real and imagined worlds and relationships; they offer meaning and experiences of *particular* worlds and *particular* relationships" (Bogost, 2007, p. 241). This kind of specificity is important to the way that people engage with *LitM* through critical modding: altering the game's story and gameplay by changing its code offers modders ability to not only explore portrayals of queer characters in games in games but to model the particular kinds of queer characters they want to see in *LitM* itself. Engaging with the specifics of a game world is important to critical modding because "rhetorical positions are always particular positions: one does not argue or express in the abstract" (Bogost, 2007, p. 241). As such, *LitM* allows modders to make arguments about queerness in games by providing a specific set of narrative and gameplay elements that are designed for expression about that topic, and critical modding offers a framework for making such arguments. Modifying the game therefore enables people to explore queerness in ways that are more naturally suited to video game-based learning approaches instead of simply applying the behaviorist and constructivist educational approaches to video games.

Of course, like any game, *LitM* contains rhetorical elements, but it is worth noting that any rhetorical argument that a game's narrative makes is at least somewhat dependent on the decisions of players, since they are always interacting with the story and constructing it during gameplay. Games are different from other media, however, in that they encourage players to optimize those decisions as well: Costikyan (2002) argued that "if a game involves any kind of decision making, or trade-offs between different kinds of resources, people will treat these as 'puzzle elements,' trying to devise optimal solutions" (p.10). In this case, then, *LitM's* story itself becomes a puzzle: interacting with the narrative elements in the game can lead to better outcomes during the game's encounter sections since those elements often give the player information about which choices are better in the game's conflict situations. More importantly, Costikyan (2002) suggested that all kinds of games have such 'puzzle elements,' claiming that "you can't extract puzzle from game entirely" (p. 10). This factor is why the encounters in *LitM* are designed with better and worse options: each encounter has both an optimal solution and a choice that is significantly worse than the others, and players will naturally try to figure out those solutions, since games inherently encourage players to do so. The distinction between puzzles and games has been contested, however, and Costikyan (2002) argued that "a puzzle is static. A game is interactive" (p. 11): a game must change state in response to a player's actions in addition to having puzzles to be solved. As such, *LitM* can change significantly based on a player's actions, and the optimal solution to an encounter can be quite different when the player reaches certain points in the game: if a character is dead, for example, players may not be able to choose the best solution to an in-game problem and must instead rely on the options they have left. On the other hand, if a player has had conversations with many of the game's characters, the player's chances are significantly better on certain encounters, so one character might be much better at resolving encounters than the others. This link between narrative and gameplay ensures that the game's encounters are not simply static puzzles to be solved, since the resolution of these gameplay elements is directly connected to the game's story and the ideal solution for each encounter changes over time.

Since the story of *LitM* itself focuses primarily on queer characters, the design of the game's narrative also relies on the concept of queer game design. Ruberg (2019) described queer game design as "the practice of deliberately using game mechanics and other design elements to challenge normative expectations around gender, sexuality, and the established logics of digital gameplay" (p. 111). *LitM* employs this approach by placing an emphasis on empowering the game's queer characters over the game's sole straight character in that the majority of the game's narratives, mechanics, and systems are all build around the game's queer characters. That being said, Ruberg (2019) also "[complicates] the idea that the goal of queer game design is simply to present alternatives to the status quo" (p. 114): queer game design is about more than just including queer content, so simply including positive portrayals of queer characters in the game and emphasizing them over the portrayals of straight characters is not enough. This factor is why I designed *LitM* as a modifiable gameplay and narrative generation system: instead of only presenting queer content, the game's code itself encourages people to change the norms of the game modifying its narrative and gameplay logic. I included both descriptions of how the code works and suggestions for how it could be modified throughout the game's code; in addition, in the Chapter 5 I discuss some sample assignments that focus portrayals of queer characters in games. I argue that this approach is an effective way to explore queer representation in games because “game design is itself a critical practice and mode of cultural analysis” (Ruberg, 2019, p. 114): as such, critical modding of *LitM* allows players to engage with a narrative and gameplay system that explicitly places an emphasis on the stories of queer characters in video games so that they can critically engage with that concept. Since “game design has the potential to lay bare and invite reflection on the workings of larger structures of power” (Ruberg, 2019, p. 114), critical modding of *LitM* can explore the way queer characters have been portrayed in games and to create new ways of portraying such characters. Overall, modification allows players to explore queerness in games in a way that goes beyond play: Ruberg (2019) suggested that “queer play resists normative expectations both in and out of the game and establishes its own, emergent rules for how the game should be played” (p. 136), and I argue that the notion can be extended to the practice of critical modding as well, which allows players to establish their own rules for how a game should be played directly in a game’s code.

In addition, I relied on other resources about queer game design to inform the narrative of *LitM.* A good example of such a resource is Queerly Represent Me (2019), a nonprofit consultancy group from Australia, who provided a significant number of resources for developers interested in queer representation in games. In particular, they suggested that developers focus on narratives that empower queer characters (Queerly Represent Me, 2019), an approach I put into practice in *LitM.* This concept does not necessarily mean that queer games must always be positive or uplifting, but it does suggest that queer characters in such games should have significant agency. While the player’s actions ultimately decide the outcome of *LitM* as they do in most games, the queer characters in the game have a significant amount of narrative self-determination: they can decide they are not interested in talking with the player, they determine what the conversation topic is when they do talk to the player, they can avoid revealing more personal information about themselves until they know the player better, and they can even decide not to enter the player's community at the end of the game if they have not interacted with the player much. The game's queer characters can die, but only during the game's encounter sequences, and character death only occurs when a queer character acts to protect the team by trying to resolve an encounter without violence. By contrast, the game's only straight, violent character has little narrative self-determination: if the player relies on his abilities, the other characters quickly abandon him, and using him too often results in the worst ending for the game. This inversion of the typical approach to post-apocalyptic games in which violence is the most common solution to problems is intended to encourage the player to use the other characters' abilities instead: in fact, using violence is actually the least effective approach to the game. Overall, this structure maintains an element of risk for the game's queer characters so that their acts still have an impact within the narrative of the apocalyptic game world, but also empowers them by placing much of the narrative and gameplay focus on what those characters do, rather than solely placing the focus on the player's decisions and story.

While quite a few theoretical approaches inform the narrative design of *LitM*, one key element of this project is that the game is designed to be modifiable, allowing users to take their own approaches to the game's content that are in conversation with the concepts outlined above. As in most games, many of the gameplay elements in *LitM* are based on random chance or are determined by the player's previous actions, so the critical modding approach I outlined in previous chapters of this project encourages people to engage with the game’s mechanics through modification so that they can resdesign these elements as they like. More importantly, much of the game’s narrative is tied to these randomized elements as well, and the game’s characters have relatively brief stories that are intended for expansion by people interested in critical modding, especially those who interested in exploring depictions of queer characters in games through that approach. Of course, this approach also means that the backstories of each character could have more depth and likewise means that there is room to “go queerer” with the story of each character. It is also possible that people could potentially modify the game in problematic ways, and while I cannot prevent all such possible uses, I provide an outline for a class based on critical modding and queer representation in Chapter 5 to offer a model for how this approach could be implemented in a way that focuses on positive examples of queer representation. I have also documented my design process in the next section of this chapter and offered extensive comments in the game’s code for those who are interested in modifying it so that the design of the game itself is intelligible.

Game Design Document

Game Overview

LitM is a choice-based interactive fiction game prototype developed in Ink, a scripting language designed to build text-based branching digital stories. The game depicts five characters in a “megapocalypse” setting: a world in which many of the common tropes related to the end of the world in apocalypse fiction have all played out simultaneously, leaving few survivors or safe places. The characters are trying to reach a nearby refuge after the previous one they were living in was destroyed, and the player, an inhabitant of that new safe haven, follows the team's journey through text communication and gives the team advice about how to resolve the dangerous situations they encounter. The goal of the game is for the player to help as many characters survive as possible, and it is designed to be short and replayable so that the player can get many different experiences over the course of multiple playthroughs.

LitM's narrative portrays the game's characters from a critical perspective that focuses on queer representation in video games. The game's narrative and gameplay both focus on empowering the game's queer characters and placing less of an emphasis on the violent straight male characters typically found in apocalypse fiction. As such, most of the game's main characters are characters with queer identities that are not common in post-apocalyptic video games; similarly, much of gameplay centers around conversations with the characters and finding ways to use their skills and resolve situations through methods that are not violent. Like much of apocalypse fiction, however, there is a violence-oriented, masculine straight character in the game: this commonly trope is represented by a character named "Guy McMannis." Character interaction is therefore one of the main components of the game, and many of the game's systems center around creating procedurally generated dialogue as the player advances through the game.

Main Gameplay Mechanics and Progression

LitM relies on two main sequences of gameplay that the player alternates between. Gameplay challenges are created through "encounters" that present the team with randomized conflict scenarios based on various apocalypse tropes: for example, the team might have to get past a horde of zombies, an alien invasion team, or a robot extermination squad. In each encounter, the player advises the team by choosing which character should lead the team from a text-based list of options. Each character is an expert at resolving dangerous situations based on diplomacy, stealth, or other methods that avoid combat. Each team member is good at resolving a different hostile situation while being less useful in others, encouraging the player to try to find the best solutions to encounters with the aim of keeping more of the team's characters alive. During an encounter the game randomly determines the team's success or failure based on the character the player chose to lead the team: choosing a character who is well-suited to the task gives the team a much better chance of succeeding, while choosing the wrong character gives a higher chance of failure. Success means that the team escapes the encounter unharmed, but failing an encounter means one of the team members dies, determined at random – except for Guy McMannis, who is immune to harm to represent the privileged status of violent straight male heroes in most video games. If everyone but Guy McMannis dies, the game ends, but the player is given the option to start over and the game suggests that the player try again. If at least one character other than Guy McMannis survives until the end, however, the player wins, and the ending the player receives is based on which characters survived the journey, which characters the player talked to, and many other narrative and gameplay elements.

The player also has a second option for approaching each of the game's encounters: in each situation, the player has the choice of letting Guy McMannis lead the team instead of one of the game's more peaceful queer characters. No matter what situation the team encounters he always takes an aggressive, combat-oriented approach, and having him lead the team always ends with another randomly chosen team member abandoning the team due to his actions. If the player makes this choice repeatedly, the game's narrative becomes more and more antagonistic toward the player, but the player can eventually reach the end of the game by having Guy McMannis take the lead in every encounter. In many ways, this play style represents siding with the violent male character that is so often cast as the hero in many video games: for example, Guy McMannis shares many traits with Cecil, the protagonist of Final Fantasy IV (Square, 1991), a character whose abilities harm himself, such that his "role in battle is reduced to a self-destructive aggression" (p. 125). Guy McMannis shares such a role when chosen to lead the team in the game's encounter situations, as using him frequently is similarly self-destructive: playing the game this way results in Guy McMannis surviving to reach the safe haven, only to find that the other characters have already been there first and warned the community about his violent ways, with the player being exiled from the community alongside Guy McMannis for giving the team such poor advice. The game's narrative very explicitly comments on the player's self-destructive choices throughout this ending, pointing out that the player advised the team to resort to violence as much as possible and ended up focusing on no one other than the game's sole hyper violent straight male character. The goal of the ending is to encourage the player to replay the game again and to choose less violent methods of resolving the game's encounters the next time around as well as to focus more on talking to and interacting with the game's queer characters.

The other main sequence of gameplay in *LitM* takes place during the "chat" sections of the game. Between each encounter, the player can attempt to talk with each character on the team, learning more about that character. These conversations give clues as to which character might be a good choice for a particular encounter in the future or what types of situations the character might not be good at resolving. Each character has their own unique conversation dialogue related to their strengths, weaknesses, and sexuality; these topics are randomly chosen when the player speaks with a particular character to encourage the player to chat with all of the characters throughout the game, though the characters will not talk about their sexuality until the player has already engaged them in conversation about other topics first. This structure also creates a more organic conversational experience in comparison to the approach taken in post-apocalyptic games like *Fallout 4* (2015) or *The Walking Dead: Season One* (2012), in which the player controls the topic and flow of conversation and the game's characters simply respond when prompted. In *LitM*, queer characters decide what they are interested in talking about when the player initiates conversation by selecting a topic from a semi-random set of options. Those characters can also reject the player's attempts at conversation if they are not in the mood to talk: this design strategy is intended to portray a degree of self-determination for the characters and also encourages the player to try to keep the characters alive and explore each character's dialogue multiple times to hear all they have to say. These conversations impact the "encounters" potion of gameplay as well: talking to the characters can improve their chances of success in certain encounters and learning more about the team makes some of the game's more difficult encounters much easier, suggesting that the player should learn more about the backgrounds of these characters to get gameplay advantages. The only character for whom these things are not the case is Guy McMannis, as he presents the same dialogue each time the player chats with him and simply encourages the player to take a more violent approach if the player has not done so yet. Overall, the goal of these chat sections is for gameplay to encourage the player to avoid the story of hyper violent straight male character and to explore other narrative options that are less common in video games.

Game Setting and Background

LitM takes place in a "megapocalypse" setting: the game takes place in a near future world in which in many different apocalypse tropes have played out, leaving few survivors and even fewer safe havens for people to escape the dangers of the wasteland. The player inhabits one of the few secure communities left in the world, and the player's community has set up small bunkers throughout the wasteland to assist other survivors in the hopes of finding people with valuable skills who can contribute to their safe haven. Those outside of such safe havens are constantly beset by both hostile enemies and environmental hazards, and the team of characters that the game focuses on are trying to survive in this hostile environment in the hopes of reaching the player's community and potentially joining it. Some computer networks and text messaging survived the apocalypse, so the player's character relies on both extensive knowledge of the world's dangers and text conversations with the characters themselves to try to guide them through the dangers they face to the safety of the community.

The game's main enemies are drawn from common tropes found in apocalypse fiction. Bandits are groups of people who have been driven to attack others as a means to survive, and they are best dealt with through discussion and diplomatic means. Zombies are flesh-eating undead who relentlessly come after travelers and are most easily handled through stealth and avoidance. Aliens are creatures from another world who have come to destroy humanity; luckily an element of their biology allows them to be harmlessly rendered unconscious by someone with the right medical and scientific knowledge. Robots are deadly machines programmed to never stop until they have killed all humans, but they can be disabled by someone with the proper technical skills and computer expertise. There are also gigantic, deadly monsters with no known weaknesses: the only way to overcome and evade them is with proper teamwork, and even that is a risky proposition. Finally, there are devastating ice storms that ravage the planet due to climate change: only the most prepared teams can survive them, and only someone with a great deal of knowledge about the team's skills can guide them through an ice storm safely.

The choice of a post-apocalyptic setting for *LitM* was made for many reasons including my own enjoyment of games with such settings, but one of those reasons was very straightforward from a design perspective: post-apocalyptic settings are very common in video games as such settings allow for the inclusion of life and death conflict scenarios that can be tied to the kinds of victory and defeat structures that form the backbone of gameplay. Such a setting also allows *LitM* to be in dialogue with many gameplay franchises, such as the *Fallout* series, simply because of their shared narrative contexts. While an emphasis on survival in a hostile world is useful from a game design perspective, it also works well with a game that focuses on depictions of queer characters, as it reflects the hostility that many queer people face in their own daily lives. *LitM's* setting therefore provides a context in which placing the characters in conflict scenarios makes sense and in which the setting can reflect the lived experience of queerness.

Game Characters

The player plays as "The Gatekeeper," a member of one of the few safe communities left in the post-apocalyptic world of the game. The player's role is to communicate with people outside of the community and to guide them to safety if they have useful skills that they can contribute to the group. Details about the player's character are deliberately nebulous: the character has no defined age, gender, orientation, or any other specific characteristics other than being "The Gatekeeper." This approach allows players to place themselves in the role of the character, an effect that is enhanced by the game's usage of the term "you" for the player, an approach commonly used in interactive fiction works. Overall, the main goal of this character is to enable a sense of presence for the player in the game world since the player's choices determine a significant amount of the game's narrative content while still placing the focus of the game's story on the queer characters themselves.

Mike is a member of the team who specializes in talking with people and is therefore especially good at dealing with bandits. He learned his negotiation skills while working in sales, and while he was quite good at it, he eventually decided to pursue a life of academia instead. He does not like working with machines, however: his dad tried to force him to work on cars when he was younger, and he grew to hate any kind of work that involved machinery, so he prefers to avoid dealing with the deadly robots that roam the wasteland whenever possible. He is demisexual, and romantic relationships have always been challenging for him. In some ways the apocalypse made his life easier since people seem less concerned with romance in such a dangerous world. He can sometimes be a bit quiet, but when he is discussing a subject he is passionate about he suddenly opens up and becomes very talkative.

Jean is a member of the team who specializes in stealth: she is particularly good at helping the team avoid zombie attacks. She learned her stealth skills while hunting with her dad at a young age, but later decided hunting was not for her. She pursued a career in filming nature documentaries instead and became quite proficient at keeping a group of people quiet and sneaking around with lots of supplies. She always hated books, movies, and video games about aliens, though, and when an extraterrestrial invasion actually happened, she felt like her worst fears finally came true. She is pansexual and has always struggled a bit with dating because she never understood the idea of being attracted to a particular gender, but she has also had some successful relationships – though the apocalypse has certainly made dating harder for her. She likes to talk and tends to be the most conversational of the group.

Dana's strengths lie in medicine: they are the team's doctor and they typically handle encounters with the aliens, whose biology is weak to certain chemical compounds. They wanted to be a doctor from a young age and pursued a medical career all throughout skills, excelling at it in almost every case. Their only real weakness was in their bedside manner: they were never very good at negotiating with patients and always seemed to come off as if they were giving orders, which makes them not particularly good at dealing with bandits that often need to be talked down from violence. Dana is nonbinary, and the idea of conforming to a particular set of gender standards never made sense to them. Dana is also polyamorous: monogamy never worked for them either and only being able to love one person seemed silly, but once they discovered polyamory they had several fulfilling relationships. Dana does not mind talking but tends to pause a lot to consider their words, since sometimes they can come off as overbearing or as if they are giving commands when they are in a leadership position.

Alex is the team's computer technician and is an expert at dealing with the deadly robots that roam the wasteland. She grew up working on machines with her parents, who were both computer programmers and who liked to build computers for fun in their spare time. Alex kept up with building computers until college, where she went to school to learn video game design. Alex has never really liked zombies: the idea of the undead has always scared her, and when they came to life for real she was terrified and she avoids them at all costs. Alex is transgender and her family and friends have always been very supporting of her. She is one of the least talkative members of the group, but she does not mind talking with people over the computer, since it makes her more comfortable.

Guy McMannis is the team's most violent member. He prefers to resolve situations through hostility whenever possible, and if given the chance, he will charge into nearly any encounter with guns blazing, regardless of the consequences to the rest of the team. He does not have any significant dislikes in terms of the dealing with conflict situations in the apocalypse other than not being chosen to lead the team: he feels that his violent approach is the best way to handle any scenario. Guy McMannis is straight, but he does not talk about his relationships much, and seems uninterested in sharing facts about himself or his backstory. In fact, he does not talk much at all, unless it is to encourage the team to take a more violent approach to solving the conflicts they are involved in.

It is also worth discussing how each character's story is also encoded in the game's variables and how characterization is tied to the game's code. Each character's name and pronouns are stored in text-based variables, allowing them to be quickly changed by someone who is interested in modifying the game to create a new character with a different gender identity. However, I avoided storing each character's sexuality in these kinds of variables and instead focused on providing those narrative elements about the characters through the game's dialogue sections, as I felt that such characterization elements could not be handled with appropriate depth by encoding them in a simple design element such as a variable. I also did not assign specific racial characteristics to any of the characters through variables, much like I avoided creating code variables specifically tied to sexuality, as both race and sexuality are identity elements that are too complex to be stored in simple code structures like variables. The characters also do not bring up the topic of their own race during the dialogue of *LitM.* This approach is not meant to undermine the importance of how racial identity intersects with queerness or to ignore the significant contributions of nonwhite people to queer history; instead, I wanted to avoid suggesting that racial issues in games should be addressed in the same way as queer issues or that that *LitM* could fully address racial issues simply by adding text variables for each character's race. I also avoided such elements because depictions of race in games are often heavily tied to a game's visuals and *LitM* does not contain images; while each character could discuss their race as part of their backstory at a characterization element, I do think that race could not be effectively engaged with through critical modding without the inclusion of images. That being said, I do believe that elements of racial identity could be added to the stories of each character, especially if a modder was interested in included images for the characters as well.

Game Story and Dialogue

While *LitM* has a main story that focuses on the team's travels through the post-apocalyptic landscape, much of the game's narrative content is randomly generated from code that controls the structure of the game. The game's story is told through both the encounters that the team must overcome and optional conversations with the team members that the player can engage in. The player takes the role of The Gatekeeper, a person responsible for guiding survivors to a safe haven and deciding who will be allowed to enter. After a short introduction, the player receives a text communication from the team indicating that they have found a series of bunkers along a road the leads to the safe haven that the player lives in. The player discusses the situation with the team and decides to offer them guidance with the hopes that the team can survive the journey to the player's community. The team must then get through eight encounters with hostile opponents before reaching the safe haven, where the player must make a final choice about who to invite into the community.

For much of the game, the player communicates with the team via text messages as they travel to the refuge and offers advice during each encounter as to which team member should resolve the situation. Each one of the team's members has a specialization that increases the team's likelihood to survive an encounter without losing anyone on the team, and each is also particularly bad at resolving a certain type of encounter. Talking with the characters during the game's chat sequences can reveals what their strengths and weaknesses are, encouraging the player to engage in the optional chat sequences that expand the game's story. If the correct character is chosen to lead the team in an appropriate encounter, the game highlights the idea that that character was a good choice; the game also provides text that notes the character was not the best choice if that is the case. Randomized conversation snippets from one of the team's characters are also presented between many of these encounters, so the game's story varies each time depending on who survives throughout the various encounters. Because of the way the encounters are designed, this ensures that multiple playthroughs are necessary to see all of the game's endings and other narrative content.

If the player instead chooses the aggressive, combat-oriented approach to the game that focuses on Guy McMannis, the violent straight male character, the game's story is very blunt and short, with little variation. His dialogue options and abilities display the same text every time, deliberately creating a stale, repetitive experience. The game's narrative becomes increasingly antagonistic toward the player each time they choose this option, slowly breaking the fourth wall and eventually directly criticizing the player for their choices. The ending achieved for continuously choosing this option over the others is always the same, highlighting the fact that the rest of the team abandoned the character and that the player got kicked out of the community for siding with the violent straight male character. The idea behind this ending is to criticize the common focus on violent straight male characters in post-apocalyptic video games. This approach is the quickest way to reach the ending of the game, but the game's narrative deliberately and aggressively undercuts the player's choices along the way to that ending, encouraging the player to explore other options for resolving the game's encounters that will provide a richer narrative experience.

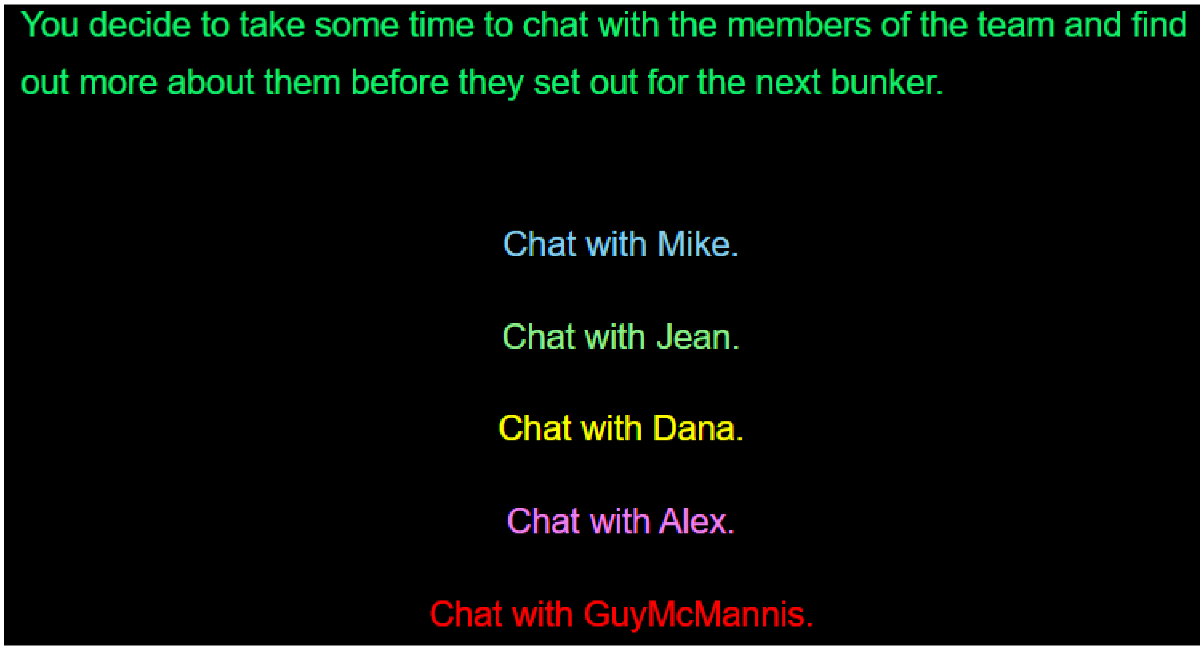
In addition, much of the game's story is revealed through the chat sections of gameplay discussed earlier. The player can access the chat portion of gameplay before each of the game's encounters and can talk with any of the team's characters who are still alive at the time. These chat sections reveal a character's backstory, strengths, and weaknesses, so exploring them is valuable since they also give the player gameplay information that can be useful for resolving the encounters in the game. Fully exploring these stories typically requires multiple playthroughs because conversation topics are randomly chosen each time the player talks to a character and because the characters can randomly turn down conversation if they are not in the mood to speak with the player. These conversations also impact the game's ending: if the player has not fully explored a particular character's dialogue, that character will not be interested in joining the player's community at the end of the game.

The game's endings are generated in response to the player's choices throughout the game but are also based on a few "final" choices that the player makes at the end of the game. When the team reaches the player's safe haven, the player has the option to invite any of the characters who are still alive into the community that they have finally reached. If the player has explored all of the conversation options with a particular character, that character will join the community and the player will be able to experience that character's "best" ending. On the other hand, if the player has not conversed with the character enough, that character will instead leave to roam the wasteland alone. The player can also invite Guy McMannis into the safe haven, a choice that instead results in the player being exiled from the community alongside him for making such a bad decision.

Overall, the game's design supports telling stories in a procedurally generated manner because the Ink scripting language is designed for creating code-based interactive fiction. The game's code creates randomized narrative scenarios that change each time someone plays the game, ensuring a different gameplay experience each time. Players can also explore the design of the game itself through its code because I have shared the game's code on my website, and I have also included samples of the game's code in Appendix C of this project. I encourage modification of the game's narrative and gameplay systems, particularly critical modding approaches that focus on exploring queer representation in video games.

User Interface

The user interface (Fig. 1) for *LitM* is web-based: Ink can create web content via its "export for web" option that can be easily uploaded to a website. Most of the player's interaction with the game comes in the form of "choices," which are clickable text-based options created within Ink code that lead the player to different parts of the game's story. I have also color coded many of these choices using inline CSS to make it easier to differentiate choices that are related to one character from one another. The game's user interface is intended to be simple and easily modifiable, which is why I chose a web-based interface for the game.


You decide to take some time to chat with the members of the team and find out more about them before they set out for the next bunker.

Chat with Mike.

Chat with Jean.

Chat with Dana.

Chat with Alex.

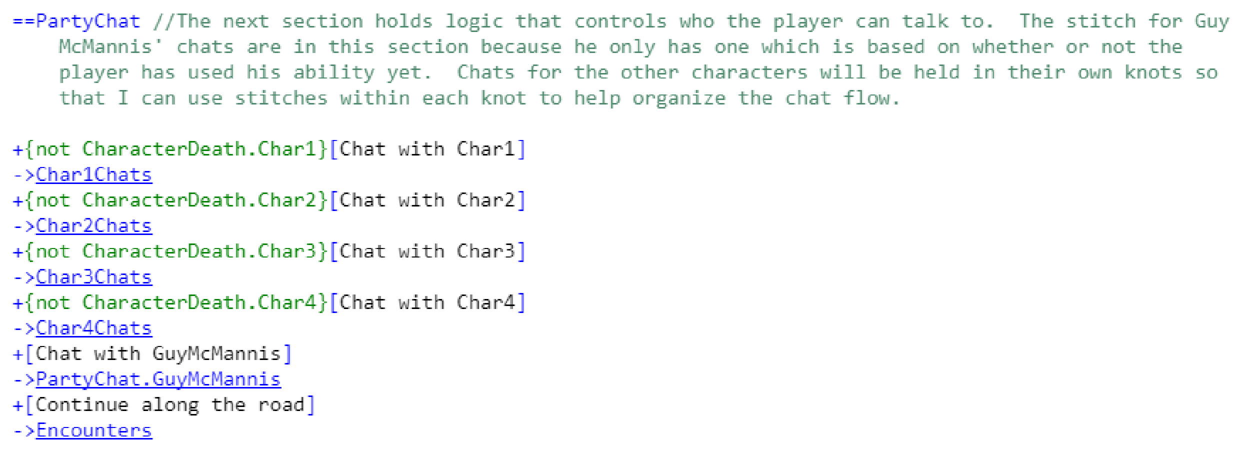
Chat with GuyMcMannis.

Figure 1: A screenshot of the web-based user interface for LitM during one of the game's "chat" sections. Source: Life in the Megapocalypse.

Art Style

Since Ink is designed as a narrative middleware engine that can incorporated into other game engines, the platform allows creators a lot of freedom in how they incorporate visuals into their games, though it does not provide a significant amount of in-engine support for art assets. For the game's visuals, I used the "export for web" function built into Ink itself; this option produces an HTML file, a CSS stylesheet, and Javascript files that can all be edited directly and uploaded to the Internet with minimal effort. Images can also easily be incorporated into the game in this fashion as well, though I did not include any in the game. As far as the game's visuals themselves go, I rely only on colored fonts and HTML / CSS styling so that the game's visual assets are simple, making them easy to modify and expand upon. I created the game using this artistic approach so that I could easily put it on my own website to share with other users; in addition, since the primary focus of my project is on narrative and gameplay modification, expanding on the game's visuals was not a major goal of mine.

Prototyping and Development Process



```
==PartyChat //The next section holds logic that controls who the player can talk to.  The stitch for Guy
    McMannis' chats are in this section because he only has one which is based on whether or not the
    player has used his ability yet.  Chats for the other characters will be held in their own knots so
    that I can use stitches within each knot to help organize the chat flow.

+{not CharacterDeath.Char1}[Chat with Char1]
->Char1Chats
+{not CharacterDeath.Char2}[Chat with Char2]
->Char2Chats
+{not CharacterDeath.Char3}[Chat with Char3]
->Char3Chats
+{not CharacterDeath.Char4}[Chat with Char4]
->Char4Chats
+[Chat with GuyMcMannis]
->PartyChat.GuyMcMannis
+[Continue along the road]
->Encounters
```

Figure 2: *A screenshot of code from one of the earliest prototypes of LitM during one of the game's "chat" sections. Source: Life in the Megapocalypse*

While I provide a post-mortem in that describes the design process of the game prototype in more detail along with some design principles that I discovered during the creation of the game, an overview of the game's prototyping process is useful here as well, as prototyping was a significant portion of the design process, and the game currently takes the form of an advanced, more fully developed prototype. I developed numerous prototypes of the game in Ink over the course of this project. I began with a mechanical proof-of-concept prototype that demonstrated some basic core gameplay, such as the game's encounter system, random death mechanics, and a placeholder chat system that did not contain narrative content (Fig. 2). From there, I focused on developing the gameplay and fleshing out the game's mechanics while introducing systems to link the game's chat system to the encounters so that the player gained advantages in the encounters for engaging with the game's narrative. The game's next prototype therefore encompassed most of the current structure of the game, though most of the game's narrative content was not in place at that point. I then began adding narrative elements to the game, such as character backstories, strengths and weaknesses, endings, and random dialogue quips that show up during gameplay. I also added variables for character names and pronouns so that they could be quickly modified in the future. I focused on creating this narrative content later because I was able to create informal character sketches and other content for them in word processing programs that were better suited to story writing while using placeholder content in the game's code that allowed for easy testing of mechanical functionality. Once this narrative content was imported into the game itself, I created a web prototype of the game to provide to peer reviewers and my dissertation committee and to get feedback on it at doctoral consortiums. I then revised the prototype based on feedback from these reviews while also redesigning the game's code to be more easily legible to both programmers and nonprogrammers alike. The total development process for the game prototype took roughly 24 months.

One reason I used Inky was because of its usefulness for prototyping: the engine allowed me to flesh the game's narrative and gameplay out at a relatively early stage so that I could begin testing it and getting feedback on it. Making significant structural changes to the game is also relatively easy in Ink, as it is designed to allowed developers to make major alterations to the flow of a story without too much difficulty. Ink is also easy to learn, which makes it ideal for use in a dissertation project focused on making gameplay modification more accessible as a critical approach. I constructed multiple prototypes of the game throughout the writing of this dissertation to allow the two projects to inform one another and connect them together. I focused on building the early Ink prototypes first because it was easier to finalize the game's narrative and gameplay logic before writing about it in the dissertation: once I moved to that stage, I wanted to avoid major changes to the game's structure and to focus only on improving the game's visuals using Ink's web integration and fleshing out the game's story. This approach also made it easier for me to make the game available on my website so that users can easily download it and modify it if they wish.

Technical Design

The game described above is built in Ink, the main scripting language used in the Inky editor that was created by Ink Studios, a small game design company from Cambridge, England. The language is designed for creating interactive fiction with a significant amount of gameplay logic and allows for both easy writing of text and quick modification of story structure. It also supports limited HTML and CSS, allowing games created within Ink to be exported for the web and uploaded to the Internet. The Inky editor runs on both Windows and Mac systems and has been used to create professionally released games such as *Heaven's Vault* (Inkle Studios, 2019) and *80 Days* (Inkle Studios, 2014) in conjunction with the popular Unity game engine. It is free, relatively minimal in terms of system requirements, and can run on most computers, making it an ideal platform for making a game accessible to critical modding by a wide audience.

Life in the Megapocalypse's code is also designed to be legible to people with a variety of coding backgrounds and experience levels so that users can easily modify it in many different contexts. A major design goal behind the game was to keep it very easy for users with a small amount of experience to understand the code that makes the game work. This goal was informed by queer game design, which often turns to software tools "which make game development accessible for creators without extensive backgrounds in coding" (Ruberg, 2019, p. 221). To facilitate that goal, I have provided a commented version of the game's Ink code on my website, and offer samples of the game's code in Appendix C of this project. This is also one reason that Ink was an ideal choice for the project: the Ink version of the game on my website only uses Ink, HTML, and CSS, which is much easier for coders with less experience to modify.

Ink's code structure is also unique and has some clear distinctions from other programming languages, so it is worth discussing here. Most of the *Life in the Megapocalypse's* code is stored in various "knots," a term Ink uses to describe specifically named blocks of text and code that are used construct Ink games. Knots are similar to functions or methods in object-oriented programming languages in that they can be used to execute several code commands at once, so I separated the gameplay mechanics into individual knots that contain major functionality of the game itself: random character deaths, characters abandoning the team, the chat systems, the game's encounters, endings, etc. These knots also contain "stitches," a term used by Ink for smaller subdivisions of text and code within the larger knots, which can be used for organization. While Ink supports creating games out of multiple linked Ink files, which is more similar to the approach used in other kinds of game programming, the technical design I used in *Life in the Megapocalypse* is aimed at allowing a player who is interested in modifying the game to quickly find the section of Ink code that deals with a particular system without needing to search through multiple individual files. Each of the game's systems also has comments that explain how it works and what might need to be changed inside other systems if certain kinds of content are modified within it. These factors make the *Life in the Megapocalypse's* code easy to understand to people who are new to programming, but someone who has more coding experience can also quickly grasp how the game's knots and stitches can be manipulated to create even more complex narrative and gameplay states. Traditional programming concepts such as variables also work in Ink as well: for example, all the game's character names and pronouns are stored in text-based string variables that can be quickly changed to swap one of the characters to a different gender, so a user with only a small amount of programming experience might be able to focus on changing these variables and then rewriting some of the narrative elements of the game. Finally, while Ink supports common elements of computer programming such as functions that can create complex gameplay states, it also has built-in code structures for elements like randomness, sequencing, and cycling that are designed to be both easy to understand and very flexible so that users with less experience can quickly create deterministic gameplay. I take advantage of these elements in *LitM's* encounter system, chat system, and in many other places in gameplay rather than building those systems through more traditional programming functions so that players can quickly modify things like the game's random death system. These elements are also much easier to learn than traditional computer programing concepts, which makes them ideal for use in educational settings in which learners might not be as familiar with those concepts. I discuss some sample assignments that center on such concepts in Chapter 5 of this project.

Conclusion: The Value of Documenting Digital Design Work

In this chapter I focused on the narrative and game design elements of *LitM* to make the prototype's overall structure visible to those who are interested in modifying it. While this chapter was primarily aimed at those who want to engage in critical modding using *LitM* as a framework, one goal of this discussion was also simply to document the digital design work that went into this project: such work is significant, but often goes unrecognized in hybrid digital projects in favor of a heavier focus on scholarship and theoretical elements. Such elements are important and theoretical considerations certainly informed the design of *LitM*, but I suggest that critical modding helps to place digital design work at the center of an approach that can meaningfully address problems in video games as a medium, and as such I placed that work at the center of this chapter as well. I argue that this kind of discussion should become a larger part of digital scholarship in the future: design documentation should be valued as meaningful scholarship in its own right, as such documentation offers insights into the actual process involved in creating the digital elements of hybrid projects such as this one. I would therefore encourage others who do such work to document their design processes as well, especially if such work is being conducted in an academic environment, as legitimizing this kind of work as a scholarly practice will require the efforts of numerous designers. Doing so is also especially important because digital design work is often unrecognized by promotion committees in academic environments, highlighting a need for structural changes that help to value the considerable efforts that go into such projects; a push for recognition of game design documentation like concept documents, art style guides, post-mortems, and other conceptual materials is therefore necessary for those who do such work. As such, I hope this chapter offers a useful model for presenting game design documents as scholarship to others who also aim toward recognition of such works.

Digital design work can also produce meaningful scholarship of other kinds in that it often reveals useful information about design principles that others can use to guide their work as well. In Chapter 5 I continue my discussion of narrative and game design, but from a different perspective and through a different kind of design document: I offer a post-mortem on the development process of *Life in the Megapocalypse* to lay out some key design principles that I discovered while creating it. These design principles are aimed at those doing queer work, modification-based work, or any kind of radical work that intersects with games. In addition, I offer suggestions for potential classroom implementations of my critical modding approach based on some sample assignments that I provide in Appendix B. While I do not think this approach should be limited strictly to classroom settings, I offer these suggestions with the aim of providing a practical application and context for learning the kinds of skills discussed earlier in this project. Finally, I continue my discussion of potential educational uses for critical modding by proposing the idea of an entire class based on the approach I outlined in this chapter; I also look at ways that critical modding could be expanded to explore other kinds of representation through modification. Overall, the final chapter of this project focuses on the lessons I learned as part of this project and how this kind of work can be expanded and continued into the future by both myself and other scholars.

CHAPTER 5: LIFE IN THE MEGAPOCALYPSE AND CRITICAL MODDING: DESIGN PRINCIPLES AND EDUCATIONAL PRACTICES

In the previous chapters of this project, I addressed issues related to learning, queerness, and game design that apply to the critical modding approach I have outlined in this dissertation. In Chapter 1 I provided a framework for using video game modification for exploring queer representation: I discussed why games are useful for learning, why queer theory is an ideal framework for considering modification, and why certain design strategies are valuable when creating a game focused on portrayals of queer characters. I also offered a brief overview of the game prototype I made as part of the project, *Life in the Megapocalypse (LitM).* Chapter 2 is where I addressed the educational elements of my approach, analyzing the ways in which games, especially interactive fiction games, are useful for learning, as well as the ways in which video game modification intersects with game-based educational approaches. Chapter 3 focused on portrayals of queer characters in mainstream and indie games as well as queer modifications of mainstream games: I argued that mainstream games handle queer characters poorly and that indie games do a significantly better job with them, and I also suggested that queer modifications of mainstream games are often designed to address problems with portrayals of queer characters in mainstream games that are common in many different franchises. In Chapter 4 I provided a narrative design framework and a game design document for *LitM* to outline the design considerations that went into the game and to document that game's overall design process, as well as to argue for the value of design documentation as a form of scholarship.

In this chapter, I offer design principles that other designers can apply to their own work on representation in video games that are based on the game prototype I created. I also address the ways in which this approach can be used in the future by looking at how these ideas could fit into a larger game design curriculum. First, I provide a post-mortem about the game prototype I made as part of this project, *LitM.* My goal with this post-mortem is to provide a narrative about the creation process of the game so that other designers can envision the kind of work this project entailed; more importantly, I offer some key design principles that I discovered while working on the prototype that are applicable to designers who are interested in creating games that are based on representation of characters with marginalized identities. After that, I discuss class assignments based on queer representation, *LitM* and critical modding that are designed to be integrated into a game design class to provide an example of how this approach could practically applied as an educational strategy; the assignment prompts themselves can be found in Appendix B. Finally, I close the chapter with some notions about how a whole class could be created based on this approach, including a discussion of some readings and games that could be used in such a class, as well as how this approach could be adapted to deal with other issues related to representation of characters with marginalized identities in video game culture. Overall, in this chapter I argue that this project offers useful lessons that other designers, scholars, and educators can draw upon when working with interactive fiction games that focus on representation. I also suggest that the critical modding approach I outlined in project can be used both in and out of the classroom and provides a useful framework for research about representation in video games.

Lessons from Life in the Megapocalypse: A Post-Apocalyptic Post-Mortem

Overall, one primary contribution that this project makes to research on game design is demonstrating some key design principles for other creators. Before reviewing those design principles, which are particularly useful for people interested in creating games that focus on representation of marginalized groups, I first provide a narrative about the creation process of *LitM* here to help other designers conceptualize the scope of work that a project like this entails. I began creating content for the game in Fall 2018: I wrote a short game design document that eventually became Chapter 4 of this dissertation and designed a bare-bones playable prototype of the game built in the Inky engine. I created the game design document first because that is a common practice in game production process: such a document is typically created during the preproduction process to assist designers in communicating ideas to other parties (Chandler, 2014, p. 166). I also created that document out of necessity: I needed to provide some basic information about the game as part of my dissertation prospectus so that my committee had an idea of the kinds of content I was creating and what my overall goals for the game were, much as a designer might do when initially pitching a game to a studio in an industry setting. Similarly, I created the first prototype of *LitM* at this point because building a playable prototype is typically the first major milestone in a game project (Chandler, 2014, p. 159); however, I also needed that prototype to give my committee with an idea of how the game would actually work. The first demonstration of the prototype occurred when I presented my dissertation prospectus and game prototype to my committee in Spring 2019. After receiving feedback on both, I continued work on the game over the Summer of 2019 and created a complete alpha prototype of the game's mechanics and narrative by the beginning of Fall 2019, which served as the first full version of the game. During the beta prototyping phase I made smaller revisions to the game's narrative and code, which essentially consisted of the rest of Fall 2019. The final revisions to the game came throughout 2020 and took place while I was writing the rest of the content for this project. Most of the revisions during that phase focused on refining the narratives of individual characters to address issues pointed out by people who gave me feedback on the game and refactoring the game's code to be more efficient, readable, and easy to modify. Overall, work on the current fully released prototype of *LitM* that accompanies this dissertation took well over two years, though I anticipate continual work and changes to the game in the future as well. This time frame was significantly longer than I originally anticipated, and while such a lesson might seem obvious, this experience demonstrates that designers should prepare form such challenges, especially when doing design work that intersects with academic work.

The game also went through several planned design iterations during this prototyping process, and those iterations highlight the value of using a simple, text-based game design for projects like this. Initially, I planned to use both the text-based Inky editor and the popular Unity game engine to create *LitM*. I considered this approach because Inkle, the creators of the Inky editor, have made a plugin that allows Ink stories to be edited inside of Unity, and that approach is the way they created most of their games. As such, I planned on creating the game's narrative and mechanics in Inky first, then using Unity to add physics, visuals, and audio assets to make the game more interesting, an approach very similar to the one used by Inkle in games like *Heaven's Vault* (Inkle, 2019). I was dissuaded from using this approach based on feedback from colleagues who played *LitM*, many of whom pointed out that the audiovisual and physics elements provided by Unity were not necessary given the goals I had for the project. Additionally, many of those who tested the game pointed out that it would be much more difficult to modify it if the game was built in Unity because the engine requires a significant amount of digital design skill to use, undermining my intention to make the game easily modifiable by others. Similarly, I originally considered creating images to include in the game since Inky's "export for Web" option can create a web browser version of the game and allows for easy inclusion of images and HTML / CSS content. I moved away from this approach for three reasons: first, I am not skilled at the creation of visual art and would have likely needed to ask someone else to create images for my characters. Second, I felt that not including images allowed a player to imagine the characters' visual appearance however they wished: this is a common approach that is used in many interactive fiction games, and I wanted to take advantage of that genre affordance to give the game more flexibility. Finally, including images would make the game slightly more difficult to modify, as anyone who wanted to modify the composition of the team of in-game characters would have to either remove the image references from the code, create new images to use in place of the originals, or face the problem of their modified character not having any visual references while other characters do. In the end, I decided to restrict the project's scope by using simple HTML and CSS code to create small visual flourishes in the game, such as color-coded fonts for each character's dialogue, that are relatively easy to modify with basic web design techniques and code. Overall, I argue that this experience demonstrates the value of using simpler, minimalistic tools like Ink and focusing on text-based representation over heavily visual tools like Unity that are more difficult to learn and work with, especially when working on a project alone.

During the roughly two years that I spent creating *LitM*, I discovered several key design principles that both helped me revise the game and that provide lessons for other game designers who are interested in creating games that focus on representation of marginalized characters. I will note, however, that I focus here on design principles that are relatively unique to this project and that have not been heavily discussed in game design literature: while I learned a lot about game design by making *LitM*, I do not aim here to reiterate lessons about the topic that are found in texts like *The Game Production Handbook* (Chandler, 2014)**.** I do not mean to suggest that such lessons are unimportant either, as many of them were things I also learned while creating my game, and all game designers should learn about the challenges of managing scope creep, the value of creating game design documentation, the importance of using a milestone system to track progress, and many other things about how to successfully create a game that are often covered in general game design texts. My goal here is to instead highlight key design principles that apply to creating games that focus on representation of marginalized characters: I argue that creating a game with such a narrative offers some unique challenges that are not faced by other designers, and the principles I lay out below are aimed at providing advice to designers who are working with similar topics.

One of the first key design principles I discovered while creating *Life in the Megapocalpyse* was that building game mechanics and major gameplay systems first can be useful even when designing a narrative game that focuses on representations of marginalized characters. While many game design texts suggest that designers should focus on mechanics first, such as the well-known Mechanics, Dynamics, and Aesthetics framework that describes mechanics as the primary "lens" that designers use when making games (Hunicke, Leblance, and Zubek, 2013), such texts typically temper that advice with statements like "game design and authorship happen at many levels" (Hunicke, Leblanc, and Zubek, 2013). Such caveats are usually made because game design is an interdisciplinary process that can begin from many perspectives: starting with game mechanics is not a strict rule, and when designing a game that focuses on portrayals of queer characters where the narrative is obviously a critical element of the game, it might make more sense to start with the story instead. Surprisingly, however, I still found it more effective to create mechanics for *LitM* first: I designed most of the game's systems early on in the prototyping process and then added narrative content to the game after that. Such an approach may seem counterintuitive when building a narrative game like *LitM,* but I suggest that this design principle reflects a commonly referenced adage from Mark Rosewater (2016)**,** a long-time designer for *Magic: The Gathering,* who claimed that "restrictions breed creativity" in a GDC talk reflecting on 20 years of working on the game. His argument is that restrictions on how a narrative can be constructed, such as major mechanics and gameplay systems, can help designers create a better narrative because they provide an existing structure that might lead to a blend of story elements that a creator might not otherwise consider (Rosewater, 2016). This design principle does not mean that creating a game's story first is always bad idea, but I argue that adding narrative elements to a game with existing mechanics can help a designer realize when those mechanics need to be redesigned to suit that narrative, as story elements that do not mesh well with the game's mechanics become more obvious when a system acts on them. Those mechanics can also create narrative structures that a designer may not have originally considered, which could open interesting story avenues or potentially create narrative problems. As an example of how this principle can be useful, during the design of *LitM* I received feedback that suggested that the characters were discussing very personal elements of their backstory too early, which felt jarring to some players. This problem arose from the fact that I wrote the character backstories in a word processing program to keep them separate from the game's code, which was useful from a writing perspective and is a common practice when writing dialogue for a game. When I added those backstories to the game, however, I found that because the game's systems were designed to allow the characters to choose a conversation topic randomly, the characters could potentially talk about elements of their sexuality and other personal topics in their first conversations with the player, a problem that was difficult to notice until those backstories were implemented into an existing dialogue system. I subsequently revised the structure of the game's dialogue system so that the game's characters do not delve into those topics as quickly during conversations with the player: instead, they will not do so until all other conversation options have been exhausted. Overall, this design principle illustrates that designing gameplay systems first can help ensure that those systems and the game's narrative will function in tandem: in a well-designed game, the game's mechanics reinforce the narrative, and the game's narrative supplements the mechanics by providing a story that reinforces the procedural elements of the game.

Another design principle I discovered while building *LitM* was the importance of doing research on other games that feature representations of marginalized characters when creating a game that deals with similar issues. This design principle was particularly relevant to the design of my game because of the problems with portrayals of queer characters in video games that I outlined in Chapter 3: I argue that this research is especially important when creating games with characters from marginalized groups, such as queer characters, because games have often struggled to portray such groups well. As such, one important element to consider is how games have dealt with similar marginalized groups to the one a creator is representing: as I outlined in Chapter 3, mainstream games typically offer examples of problematic ways that games have portrayed queer characters, while indie games almost always provide examples of ways to handle such portrayals better. Such research should take the form of looking at specific depictions in specific games, especially those with aesthetics or gameplay systems that are similar to those of the game the designer is making. During this project, I researched both mainstream games that portray queer characters and indie interactive fiction games that focus on queer representation; much of that research is outlined in Chapter 3. That research helped me improve the narrative of the *LitM's* queer characters by comparing their stories to examples of queerness found in other video games and revising the stories in my game to be more empowering: for example, I looked at problematic narratives in post-apocalyptic games like *The Walking Dead: Michonne* (Telltale Games, 2016), where the queer characters have little agency and are killed off during dramatic moments in an attempt to have an emotional impact on the player. Analyzing issues with portrayals of queer characters in other games with a similar aesthetic led me to redesign elements of *LitM* story to focus less on the kinds of traumatic narratives for queer characters seen in such games: for example, I changed the way the game's death mechanics worked so that talking to the game's queer characters made success more likely in dangerous situations, reducing the chances for those characters to die without removing that danger altogether. Overall, this kind of research on games can provide a wealth of information about how representation of a particular group has been handled in the past, and this factor is also why such research was so important to this project.

Similarly, another design principle I discovered while creating this game prototype is that it is critical to get feedback from members of a marginalized group about how they are being represented in a game. Such feedback is some of the most important information a creator can gather, as it allows a creator to gather the opinions of those who are being represented directly.

People who have life experience as a member of the group being represented as well as experience with fiction about such groups, often referred to as sensitivity readers, can be incredibly helpful to gather this kind of feedback: however, they must be compensated them in some way, as not doing so creates the problem of asking people who have been marginalized by a narrative to do the work of fixing it. Getting feedback from sensitivity readers is especially important if a creator represents characters with identities that are outside of their own life experience, but such feedback is still important even when creating content that is based on one's own personal experience, as such feedback can help to expand the narrative beyond one's own experiences. Getting additional feedback from others with similar experiences therefore helps to make a game's narrative feel more realistic, and at several points, I revised the stories of characters in *LitM* based on feedback from colleagues who played the game: for example, I rewrote the dialogue of the game's pansexual character based on the notes of a colleague with similar life experiences, changing some small details about how the character discusses relationships so that they more closely reflected that colleague's experiences. Such feedback helped me make many improvements to the game's overall story and was especially effective for helping me write the dialogue of the game's characters. Overall, I suggest that getting feedback from sensitivity readers with the similar kinds of identities that are being represented is valuable to anyone creating a game that focuses on portrayals of marginalized groups.

Finally, a design principle that I learned when building this game is that it can be helpful to focus on empowerment and self-determination when working with representation of marginalized group: in particular, it is important to avoid narratives of trauma that remove the agency of queer characters, as that is a particularly common problem in representations of marginalized groups. An example of why this lesson is important is illustrated by a common trope in fiction that deals with queer representation: the so-called “bury your gays” (Hulan, 2017) trope, in which queer characters in stories are killed or experience some kind of severe trauma that permanently removes them from the action so that the narrative can focus on straight characters. The previously discussed example from *The Walking Dead: Michonne* (Telltale Games, 2016) highlights this trope well, especially since the deaths of queer characters in the game are often unavoidable. To address this issue, one element I removed from *LitM* was the ability for Guy McMannis, the game’s straight, male, violence-oriented character to get the party’s queer characters killed during random encounters. In the game’s original iteration, choosing Guy McMannis to lead the party always ended up with one of the other queer party members getting killed due to his violent approach. I intended this gameplay element to be a satirical comment on how the “bury your gays” trope frequently shows up in video games in the form of queer characters dying for the game’s straight hero; however, I realized that the approach undermined the game’s overall goal of providing an empowering narrative for the queer characters, making their narrative feel more traumatic and less important. I subsequently redesigned the game such that the queer characters instead abandon the party if the player keeps asking GuyMcMannis to lead the team: doing so now causes the player to get the game’s “bad” ending in which the player’s character is exiled from their own home community for favoring violent solutions instead of asking the other characters to use their skills. In this ending, the game’s queer characters are instead welcomed into that safe haven, making their endings less traumatic and more empowering, and the game itself suggests that the player should play again and focus more on those characters. Overall, I kept a goal of creating an empowering narrative for the game's queer characters in mind while working on *LitM*, and I argue that creators should employ a similar perspective when creating narratives that focus on marginalized groups. While I would not suggest that all queer narratives must be positive or uplifting – games that focus on traumatic queer relationships like Curtain (Dreamfeel, 2014) certainly illustrate that – I would remind designers that games have struggled to portray marginalized groups well in the past and that they consider such issues when designing their narratives. I also believe that critical modding could provide a framework through which people can engage with such issues directly, and in the next section I discuss how critical modding could be implemented in a game design class through classroom assignments that address queer representation through both code and narrative.

Classroom Assignments Based on Life in the Megapocalypse

This section focuses on how *LitM* and critical modding could be implemented in game design classroom settings. Before moving on to that discussion, I will first note that I believe *LitM* could be used in many kinds of learning environments, so classroom settings are not the only way this game and framework could be employed; likewise, critical modding could be used in disciplines other than game design. As I discussed in Chapter 2, modding skills are also often learned in informal environments, and as such I suggest that critical modding could likewise be employed in such a way. Examples from drawn educational environments will likely be useful to many who have experience in that field and classroom assignments offer a clear practical application for this critical modding framework, however, so I focus on such examples so that readers who have worked in such contexts can apply the suggested assignments to their own experience. In this section I discuss several sample assignments which can be found in Appendix

B of this project; these assignments are designed for the context of a game design class, but could also be modified for use in other disciplines, especially those that focus on narrative or digital design. I also discuss some learning objectives for these assignments and describe ways that these assignments could be integrated into a game design class.

Critical modding-based approaches to exploring portrayals of queer characters in video games are an obvious way to implement this game and framework in an education environment since my approach was designed with that as one potential use. As such, I have developed this approach with some learning objectives that could be learned by modifying *LitM* in mind as well. While these objectives could be changed to make them applicable to other classroom contexts, they offer an illustration of how critical modding could be used for exploring representation in video games in an educational environment in the context of a game design class. They also offer clear goals for the kinds of learning that students would engage in by working with this approach and provide outcomes that can be assessed by the sample assignments that I present in Appendix B. To be clear, I do not suggest that these learning objectives are inherent to *LitM* or that they can be learned simply by playing it: they are intended to be used in conjunction with a critical modding-based learning approach focused on exploring queer representation in video games through modification.

Two main learning objectives for using *LitM* in a game design classroom setting are:

1. Students should understand how to modify the game’s code to change narrative elements related to one of the game’s queer characters
2. Students should understand how to modify the game’s code to add a new character to the game that focuses on empowering queer representation

These objectives are designed for use in a game design classroom because they can be assessed directly through the kinds of academic measures often used in such classes. Game design classes often rely on assignments in which students create or modify games as assessment mechanisms: for example, Salen (2007) had students create games to teach them critical literacy and game design concepts. As such, these assignments would be used in courses that place a focus on both programming techniques and the aesthetic elements that can be created through those techniques. In essence, students would play through *LitM* and then be given assignments that require them to create new content for *LitM*. Those assignments would then be assessed through rubrics or other mechanisms and linked to specific course outcomes; to that end, I offer sample rubrics along with the assignments in Appendix B as well. Since *LitM* focuses on queer portrayals of queer characters, the assignments place an emphasis on both narrative and game design, especially because the game's code encourages exploring those concepts through critical modding. I offer a brief summary of the assignments and their intended goals below, but the assignment prompts themselves can be found in Appendix B.

This first assignment in the sequence focuses on introducing students to the basic programming concept of variables: it asks them to modify one of the characters in the game by changing the character's name and pronouns. These bits of narrative information are stored within string variables inside *LitM's* code, which means that a student with little coding experience can learn the basics about how variables work by changing them and can then quickly see the results of their changes throughout the entire game's story by playing through it again, providing them with immediate context for how their changes work within the game's narrative. In addition, the assignment asks students to make minor narrative changes to account for the modified character’s name and gender so that the character’s story makes sense: for example, if a student modifies one of the game’s cisgender characters and chooses nonbinary gender pronouns for that character instead, the student would also be expected to change other elements of that character’s story to account for the fact that the character now identifies as nonbinary. This assignment structure therefore both has the student practice the student basic programming techniques while also connecting those techniques to specific narrative elements related to gender identity, encouraging the student to explore that concept through both game and story design. It also helps introduce students to some of the basics of using the Inky editor that they will need to know for the next assignment.

The next assignment in the sequence asks students to make much more significant changes to the game: students are expected to modify narrative elements of one of the game’s characters entirely. This could include changing variables related to the character’s name and pronouns as in the previous assignment, but students must also make changes to that character’s dialogue, endings, and other significant narrative elements. Since all of these elements are also connected to gameplay systems within the game, students also necessarily have to alter elements of gameplay to accomplish these tasks: for example, if a student changes a character’s strengths so that the character is now skilled in making traps instead of being an expert in stealth, the student will also need to change the gameplay systems so that the character’s trapmaking skills are now relevant. Much of the game’s narrative also focuses on how the characters identify in terms of gender and sexuality and the assignment encourages students to continue writing stories that are related to those elements, though the assignment does not explicitly ask them to change such details, allowing students to explore them however they wish. This second assignment is therefore primarily intended to expand a student's narrative and design skills by having them use critical modding to make significant alterations to the game. Since students modify many of the game's systems during this assignment, it also helps prepare students for the final assignment, which asks them to interact with almost all elements of the game's story and gameplay mechanics.

The final assignment in this sequence asks students to add an entirely new character to the game. In this assignment, students cannot simply change the existing code in the game that relate one of the characters that is already there: they instead must create their own character from scratch and integrate that character into the game's story and mechanics. This assignment involves a significant amount of narrative design: students have to come up with an appropriate story for the character, decide on that character's strengths and weaknesses, figure out how the character's story will end, and structure all these narrative elements in a way that fits with the game's focus on queer representation. Since the game systems include variables that specifically outline the character's gender and dialogue systems that heavily discuss each character's sexuality students will also have to develop those elements of the new character. The assignment also involves a significant amount of game design in that it requires modification of the overall gameplay design of *LitM* because the character has to be added to all the game's systems: most of those systems are based on the game having five characters, which means many of them need significant revision in order to work with six instead. Adding another character means that the code for the entire death system has to be rewritten, for example: the game determines when a loss state occurs by using variables to keep track of how many team members have died, so modifying the game to include additional character requires subsequent modification of all of those systems. This final assignment serves as a test of students' narrative and game design skills: it requires quite a bit of work to accomplish and asks them to make use of all the elements they learned about in previous assignments.

Overall, I suggest that this assignment sequence would result in students gaining a significant understanding of both game design and narrative design with a focus on critical representation in both areas. These assignments would also necessarily be paired with in-class discussion, readings, and other content, and in the next section I discuss some potential readings that could be used in a class based on critical modding and queer representation; those readings also might pair well with these assignments, and likewise these assignments could be used in such a class, though I have placed them in separate sections because the assignments and readings could also be used separately. That being said, such content should be considered in the context of issues in representation in video games that the instructor feels are valuable to address in the class, as critical modding is an approach that could potentially be adapted to address other issues, such as race or mental health. The assignments proposed above would also accomplish the learning objectives laid out above as they place a focus on adding and changing queer content within the game, though both the assignments and objectives could also be changed to address other kinds of representation in video games. While I have not tested these assignments in a classroom setting, I plan on conducting a study into the effectiveness of this approach in the future and outline the basic structure of a class focused on critical modding below to illustrate what such a class might look like.

Conclusion: Future Directions for Critical Modding

In the previous section of this chapter I described assignments based on the critical modding approach I outlined in this project. Those assignments are aimed at game design classes but are adaptable enough to be used in a variety of educational contexts; however, an even more effective method could be using them as a basis for creating an entire class based around addressing the dual contexts of programming and problems with representation of marginalized groups in video game culture through critical modding. Such a class would be part of a program focused on game design, where topics like coding and narrative are usually covered in different classes: therefore, the class I propose here is aimed at filling a gap in most game design programs. The current curriculum structure used in many game design programs makes sense because coding classes are typically very challenging, but that structure also means that discussions about topics like how code can reinforce problematic representations of marginalized groups are often had in classes where code itself is not being discussed. The ideas I propose in this conclusion are therefore aimed at creating a class that focuses on the connection between game code and game narratives; in particular, the class would look at issues like problematic representations of marginalized groups and how those representations are reinforced in a game's narrative through the game's code. I do not propose specific assignments in this section, however, as I have already provided examples of what those might look like in the previous section, and I do not aim to describe all the details of how such a class might be conducted here; that being said, those assignments could certainly be used in such a class. Instead, I focus on the context surrounding the class by offering key readings that the course could use as texts and games that students could play as examples of more positive representations of marginalized groups in game, along with some related course concepts that those readings and games would help introduce students to. Finally, I also consider what further research based on critical modding might look like and suggest some potential directions that such research might take.

One goal of this project has been to create an approach that is flexible enough to be adapted to a variety of educational contexts: while the reading list and games list I provide here are aimed at the game design class I am proposing and focus primarily on queer representation in video games, many of the same readings and games I suggest here are chosen because they could also be useful in other disciplines or classes that focus on a different kind of representation as well. Bonnie Ruberg's *Videogames Have Always Been Queer* (2019) would serve well as an overarching "textbook" for a class focused on representation in games as it addresses the way queerness has been portrayed in games from a variety of perspectives: in particular, Ruberg's notion that portrayals of queerness should go beyond simply having queer characters provides a useful framework in that the class I propose would consider how many kinds of representation can be handled through gameplay elements other than narrative or visuals. Similarly, Clark's (2017) "What is queerness in games, anyway?" provides an overview of the ways that queerness has been handled in games that goes beyond just looking at queer narratives; as such, I suggest that this class would analyze how gameplay elements intersect with representation as well. I argue that these texts should give students an introduction to the diversity of ways in which queerness has been portrayed in video games and how representation can be embedded in both story and game mechanics, which would be a key focus of this class. Clark's work is drawn from *Queer Game Studies* (Shaw and Ruberg, 2017), which would also work well as a collection of texts that a class based on representation in games could be built around in that it provides an example of how a particular kind of representation can be examined in the context of games.Lauteria's (2018) "Envisioning queer game studies: Ludology and the study of queer game content" would also be useful when introducing students to such works, as the author discusses queer game studies from a large-scale perspective. Examining a larger perspective on queer representation would help to provide context for the more focused discussions drawn from texts like *Queer Game Studies*, allowing the class to look at how more specific issues addressed in that text are situated in the wider discourse surrounding queer games. Finally, at least some content that focuses directly on advice for designers is useful in a class that looks so heavily at queer game design: Queerly Represent Me's (2019) "So you've decided you want to make a diverse game… now what?" is a great starting point for such discussions as it provides numerous tips for designers who are interested in creating queer games, as well as creating more diverse games in general. Since this class could also involve creating the kinds of game design assignments I laid out in the previous section, conversations about design strategies that focus on representation would necessarily be an important part of the course. Overall, the main theme of the readings I suggest here is the way in which queer representation and portrayals of queer characters are encoded in games through design. While almost all of these readings address queer stories and visual representations of queer characters, they all highlight the fact that those superficial elements are reinforced by design choices and enacted in game code, which would be a core concept of the class I am proposing. This element also means that looking directly at queer games in such a class would also be useful; as such, I suggest some games that might be good choices for this purpose below.

As I discussed in Chapter 3, there are numerous ways in which queer characters have been portrayed in both mainstream and indie games; however, I argued there that indie games have typically handled queerness much better than mainstream games. As such, my suggestions here focus on indie games that have handled queerness well rather than mainstream games that have portrayed queer characters poorly: while discussions of mainstream games would be part of this course because it looks at representation in games, such games are already popular enough that students may have played them already, and students could easily familiarize themselves with those games through things like video recorded playthroughs if they have not played the games in question. A main goal of this focus on queer indie games is therefore to provide students with examples of how representation of marginalized groups in games can be handled well, as those examples would be valuable to students as they create their own digital works as part of the class. As I also noted in Chapter 3, text-based interactive fiction games have been a major part of the queer games movement as a whole, and as such, Squinky's (2015) *36 Questions* would serve as a starting point for discussions about portrayals of queer characters in games. The gameplay is easy to learn since it is a text-based game, and the game directly engages with race and queer game design as the game's creator comments on their experiences and struggles with being a nonwhite queer game designer throughout the story; I argue that such a game is therefore valuable in that it provides students with a lens into how queer designers themselves see their own work, which would be an important part of course discussions. I then suggest moving to more visually focused games, such a Dreamfeel's (2014) *Curtain:* the game's story is about a queer relationship, but the game itself does not portray the characters visually and instead uses stylized, impressionistic visuals that create a narrative focused more on mood and emotional tone than visual fidelity. As such, *Curtain* (Dreamfeel, 2014) would give students with an example of how queerness can be portrayed through visual elements as well as dialogue and text and the class could expand upon that example with discussions about other ways to do so as well. Similarly, Andrews and Schmidt's (2014) *Realistic Kissing Simulator* suggests that portrayals of queer characters can use visual elements that do not reinforce stereotypes about gender or sexual identity, as the characters in the game do not have any recognizable gender characteristics; I claim that such a game would help to expand students' notions of how queerness in games might be visually depicted. After looking at such games, I suggest moving on to a slightly more mainstream portrayal of queer characters, such as Game Grump's (2017) *Dream Daddy*: the game's visual representation of queer characters and narratives that focus on queer relationships could then be discussed in the context of the more stylized representations found in the other games that the class has already examined, which would give students examples of the broad range of ways that representation can be handled in games. Overall, these games all portray queer characters in positive or thoughtful ways, all of them with the exception of *Dream Daddy* (Game Grumps, 2017) are available for free, and none of them have particularly restrictive system requirements; therefore, I argue that they would make good choices for use as examples in a class that focuses on representation of marginalized groups in video games.

While I have focused on queer representation both throughout this chapter and in this project as a whole, I have also claimed that critical modding could be used to look at other issues with representation in video games, and I believe that this approach can be extended to such issues as well: for example, games like Zoe Quinn's (2013) *Depression Quest* suggest that games can address a variety of issues related to representation, such as representations of mental health issues, and Squinky's (2015) *36 Questions* also engages with race by discussing the creator's experiences of being a nonwhite game developer. This is one reason why I have suggested modification as a larger context for this project: critical modding is an approach with a great degree of flexibility that can be adapted to a number of different purposes, as its main goal is simply to address problems in games through modification. Obviously, issues like mental health, race or disability in games could not be addressed in exactly the same way as queerness and attempting to do so would be inappropriate, but I argue that critical modding is an important contribution to game design discourse because it is flexible enough to provide a framework that allows players to address the issues that they find in games in a way that they feel is appropriate to those issues. As such, digital content creation is a skill that can be useful to almost any scholar of games who is interested in representations of marginalized groups in video games, and tools like *LitM* that are designed to be easy to learn, easy to modify, and that have a clear focus on a specific kind of representation can provide a framework that such scholars can use to begin exploring that idea. This project therefore provides both theoretical and practical examples of how such a tool can be constructed as well as a method for how it could be used and that other scholars can expand this approach to the issues of representation in games that matter most to them.

Finally, it is worth summarizing what critical modding is as a whole to close this project. While I have discussed critical modding in detail throughout the previous chapters, a short summary of the idea might be the best way of considering how the concept contributes to gaming culture in a broader context. In essence, critical modding is an approach to modifying games that aims at addressing problems with representation in games through game design: this method is in contrast to other approaches to modifying games, which usually aim at using modification to learn coding and game design, to create interesting in-game scenarios, or just to simply make the game more fun. Critical modding can therefore be distinguished from other forms of modding by its intentions: just as critical theory differs from other kinds of theory in that it is aimed at analyzing power structures to reveal the ways in which they marginalize and reify existing social dynamics, critical modding is aimed at analyzing games to reveal the ways in which they recapitulate power dynamics that marginalize people, such as queer players. Furthermore, critical modding offers an approach that goes beyond the theoretical in that it provides a way for people to put their critiques into practice through digital design by directly addressing those problems. This means that any form of modding that aims at addressing such problems could be considered critical modding, and in fact, many fan mods, such as the ones I discussed in Chapter 3, could be considered examples of the practice. Critical modding does not require that a modder engage with any particular kind of content, however, as specifics of what kinds of content might be modified rely on many contextual elements, such as the kinds of representation a game presents and how well it handles it. That being said, critical modding also requires learning a set of game design skills, using a research-based approach to analyzing problems with representation in games, exploring the narrative design changes necessary to address those problems, and applying a set of design principles that are informed by problems with representation of marginalized groups in games. Those elements can be learned and practiced in both informal and formal educational contexts in the ways that I described throughout this project. As such, this project laid out a framework for learning such skills and exploring such concepts, and I hope that others can expand on that approach to address many different issues with representation in games.

APPENDIX A
INSTRUCTIONS FOR DOWNLOADING INK AND LIFE IN THE MEGAPOCALYPSE

Downloading the Inky Editor

To modify *Life in the Megapocalypse,* users will first need to download the Inky editor. The editor is available for Windows, Mac, and Linux systems and all of *Life in the Megapocalypse's* functionality should work properly on any platform, though I created and tested it on a 64-bit Windows system using the 0.11.0 version of the editor. As of this writing, the Inky editor can be downloaded from Inkle's GitHub account at the following link:
https://github.com/inkle/inky/releases/tag/0.11.0

Note that the version listed above is the version I used to create *Life in the Megapocalypse* for this dissertation project, so I cannot guarantee that future versions of the Inky editor will work properly with the current version of the game; however, since the game prototype is an ongoing project of mine, I will try to ensure that it functions in the most recent release of the Inky editor. My website will indicate the version of Inky I use to create any updates for the game, and in the event that I no longer continue to update the game, I will provide a notice as to the appropriate version of Inky to use for creating modifications of the game.

Instructions for Downloading and Modifying Life in the Megapocalypse

IMPORTANT: Before downloading *Life in the Megapocalypse,* make sure that you have downloaded the Inky editor as described in Appendix A. Once you have downloaded the Inky editor, you can access the most recent version of *Life in the Megapocalypse* at my website, kentonhoward.com

You should download the linked .ink file from my website and open it on your system. I have also included a .zip file on that page that contains all of the files for a Web version of the game, though you do not need to download that version unless you are interested in creating your own Web version of the game. You can find an example of the Web version of the game on my website as well.

Instructions for how to modify the game's system can be found in the Ink file. These instructions primarily come in the form of code comments: these are short notes I have added to the game's code about how each system works along with examples of what you might need to change to accomplish certain modifications. Note that these comments do not necessarily provide a full explanation of how the Inky engine works, so it may also be useful to review Inkle's documentation on Inky. At this time of this writing, the introductory documentation that can be found here is the best place to start: https://www.inklestudios.com/ink/web-tutorial/

APPENDIX B
SAMPLE ASSIGNMENTS

Sample Assignment 1: Introductory Assignment: Change a Character's Name, Gender and Pronouns

For this assignment, you will modify a few elements of a character in *Life in the Megapocalypse.* These changes will help introduce you to the basics of how variables work in both general computer programming and within the Inky Editor itself.

For this assignment, you should do the following:

1. Change a character's name (2 points)
2. Change a character's singular pronoun (2 points)
3. Change a character's possessive pronouns (2 points)
4. Change a character's object pronoun (2 points)
5. Create a playable game with a consistent story (2 points)

Each of these elements will be assessed independently, which means if you do not attempt to do a particular element, you will not get points for it.

In addition, consider the effect your changes will have on the game's narrative: for example, how will changing character's name and pronouns affect that character's story in the game? You could also think about making changes to elements of the character's backstory, such as their sexuality. No matter what, you should make small changes to the narrative of the game to account for the programming changes you made. You do not have to completely change a character's story, but you should make sure it is still consistent and that it fits with the other changes you made so that a player of your modification would not be able to tell that you made those changes.

Grading

This assignment is assessed on a 10 point scale. Each of the modifications above are worth 2 points each: if your changes to any of those elements lead to inconsistent, offensive, or otherwise problematic narrative content, you only receive 1 point for that modification. Finally, the overall playability and story of your modified game are worth 2 points: if there are problems with either element you only receive 1 point, and if there are problems with both, you receive 0 points.

Tips

Most of the content that you need to change for this assignment is stored in variables that hold information about the characters: for example, each character has a name variable that stores the character's name. That variable is then used throughout the story whenever the character's name shows up, so if you change the variable, you will change the character's name throughout the whole story!

Changing names and pronouns should not be too tricky, but you will need to read through the story afterward to check it for consistency. In most cases you should not need to change much, but sometimes a change in pronouns might cause a sentence to read incorrectly. As an example, consider a sentence like "She has a dog" - if you changed the pronoun variable from "she" to "they," you end up with "They has a dog," which is not correct. In this example, you would need to also edit the game's story text so it reads "They have a dog" instead.

Sample Assignment 2: Advanced Assignment: Modify a Character Entirely

For this assignment, you will modify a character entirely in *Life in the Megapocalypse.* These changes will help introduce you how many systems with the Ink language work and how they can be modified or changed to create new narrative and gameplay.
For this assignment, you should do the following:

1. Rewrite the character's chats in the game's "chat" system, including the character's backstory (2 points)
2. Rewrite the character's "random chats" system in the "game structure" system (2 points)
3. Modify the character's abilities in the game's "abilities" system (2 points)
4. Modify the character's reactions in the game's "results" system (2 points)
5. Rewrite the character's endings in the game's "endings" system (2 points)
6. Create a playable game with a consistent story (5 points)

Each of these elements will be assessed independently, which means if you do not attempt to do a particular element, you will not get points for it.

Note those while you are expected to change all of these elements, you do not have to completely change everything about those things – if there are parts of the original character's story that might be applicable to your new character as well, you can tweak that content instead of completely writing it. That being said, you might consider rewriting content to account for the changes you made to the character's gender and sexuality, as those might be some elements of the narrative that might need the significant modification in order to create a coherent experience for your player. Each character's backstory in particular focuses on their sexuality, so you might want to specifically focus on that portion of each character's story.

Grading

This assignment is assessed on a 15 point scale. Each of the modifications above are worth 2 points each: if your changes to any of those elements lead to inconsistent, offensive, or otherwise problematic narrative content, you only receive 1 point for that modification. The overall playability and story of your modified game is worth 5 points: if there are problems with either element you only receive 2.5 points, and if there are problems with both, you receive 0 points.

Tips

You might want to use the same character that you modified in the previous assignment for this assignment as well: that will save you the time of having to change variables for another character and then check the text for consistency again.

Most of this assignment asks you to change narrative elements about the character rather than making major code changes, so this time around you can focus more on coming up with a new story for the character you are modifying. You should pay attention to the code for the systems you are changing, however, since the next assignment will ask you to do things that will involve using those systems.

Sample Assignment 3: Mastery Assignment: Create a Fully New Character

For this assignment, you will create a new character for *Life in the Megapocalypse* using the Inky editor. You will need to modify numerous components of the game's code to do so:

1. Add the character to the game’s “chat” system (2 points)
2. Add the character to the “random chats” system in the “game structure” section (2 points)
3. Add the character to the game’s “encounter” system (2 points)
4. Add the character to the game’s “abilities” system (2 points)
5. Add the character to the game’s “results” system (2 points)
6. Add the character to the game’s “abandonment” system (2 points)
7. Add the character to the game’s “deaths” system (2 points)
8. Add the character to the game’s “endings” system (2 points)
9. Create a short, 250-word document detailing the changes you made (4 points)
10. Create a playable game with a consistent story (5 points)

Each of these elements will be assessed independently, which means if you do not attempt to do a particular element, you will not get points for it.

Note that while you modified a character in the previous assignment, you need to create a new character for this assignment: you cannot use the character you modified in the last assignment as your “new character” for this one.

In addition, you should consider how your new character’s gender and sexuality might interact with the other characters in the game when adding narrative content for the character, as those are some of the major elements of each character’s story. You might want to play through the game again and consider how the characters are currently represented when thinking about what to add: for example, you could add another character who identifies in similar ways to one of the game’s current characters, or you could instead focus on a kind of identity that is not represented in the game at all!

Grading

This assignment is assessed on a 25 point scale. Each of the modifications above are worth 2 points each: if your changes to any of those elements lead to inconsistent, offensive, or otherwise problematic narrative content, you only receive 1 point for that modification. The written rationale for your changes is worth 4 points: if it is unclear or has significant writing errors, you will only receive 2 points. The overall playability and story of your modified game is worth 5 points: if there are problems with either element you only receive 2.5 points, and if there are problems with both, you receive 0 points.

Tips

This assignment mostly asks you to build on things you learned in the previous two assignments. Making a new character will probably first involve creating new variables for the character’s names and pronouns, which you did in the first assignment. You will need to use those variables throughout the story, so if you are writing your story text out in advance, you might want to use those variables in the appropriate places so you do not have to rewrite the story text later to include your variables.

Once you have created variables and story content, you will need to add those things to each of the game’s systems. You saw all of those systems in the last assignment as you were modifying an existing character, so you already have examples of how each of those systems work.

Finally, you might want to think about how your character will fit into the game's mechanics and overall design. Right now, the game has five characters: four of them have strengths that each make them good against a different encounter and bad at another one. There are six kinds of encounters total, and two of those six are more difficult in that no particular character is good at them. The game's fifth character is there to give the player a way to reach the game's "worst" ending: using that character in an encounter ends with another team member abandoning the party. This structure gives you a lot of options: for example, you could add a character in that is good at the more difficult encounters but worse at the easier ones. Another option could be to add a character that has no particular strengths or weaknesses, though in that case you would need to come up with something for that character to say in those parts of the game's chat system. You could even create a character as a "foil" for the game's fifth character: perhaps your new character could might have a special ability that allows them to find characters who have previously abandoned the party.

APPENDIX C
SELECTED CODE FROM LIFE IN THE MEGAPOCALYPSE

```
# author: Kenton Taylor Howard

# title: Life In The Megapocalypse

# theme: dark

//Welcome to Life in the Megapocalypse!  If you are reading this comment, you are viewing the game’s code, probably because you are interested in how the game works or might even want to modify it for your own purposes.  If so, you will find comments like this throughout the code to help you!

->setup

==setup //This is a "knot," which allows you to mark a location in your story that you can send your player to. You can also keep track of whether or not the player has been to a certain knot and do things based on that.  You can create one using "==knotName" and you can put anything you want inside of them - text, code, choices (explained below), and more.  This knot, for example, holds various "setup" options; it allows the user to get more info about my dissertation and Ink or start the game itself.

VAR charOneChatted = false //This is a variable, which are much like variables in other kinds of programming. They are used throughout the story to keep track of various elements of the game. You create your own variables using "VAR variableName = value" and you can change the value of a variable by using "variableName = newValue."
```

VAR charTwoChatted = false //Variables can have a variety of values, but one of the most basic ones is a true/false variable - a "boolean." These are useful for narrative and gameplay because you can check whether they are true or false and do things based on that - for example, if "Char1Chatted" is false, you can talk to Char1, but if it is true, you can't. If you are interested in modding the game, you can change these variables, but keep in mind that most of them are related to talking to the game's various characters and changing them will likely break the conversation system.

VAR charOneStoryHeard = false // In many cases throughout this code, I provide only one example of a variable for a given character; there are equivalent variables and other content for each of the game’s characters in the full source code, but such examples are mostly removed to shorten this sample code. I have provided comments in cases where that material has been removed.

VAR charOneStrongHeard = false

VAR charOneWeakHeard = false

// and so on for the other three characters…

VAR totalChats = 0 //Variables can also have integer values - 0, 1, 2, etc. - which can be used to keep track of things. These two variables keep track of how many times you have chatted with the characters and how many characters you have lost.

VAR charsLost = 0

VAR charOneName = "Mike" //Variables can have string values too, which allows you to store text inside of a variable. All of these variables are used to set the names of the characters in the player's team, which allows you to easily change the character's names if needed.

//VARS for other character names here

VAR charOneSingular = "he" //These variables store the gender pronouns for each character so they can be easily modified; for example, if you change "she" to "they," "they" will be used throughout the story in appropriate places when referring to the character's gender.

VAR charOnePossessivePro = "his"

VAR charOnePossessiveAdj = "his"

VAR charOneObject = "him"

//VARS for other character pronouns here

->start //This is a "divert," which allows you to jump to a new location in your story. You can creat one using "->knotName," where knotName is the name of a knot you have already created in your story. In this example, the divert happens automatically without any user input - the player jumps to the "Start" knot without clicking anything.

==start //This knot starts the story. It is useful to have a knot like this to keep your game organized. If you are interested in modding the game and want to add some information explaining what you changed this is a good place to do it.

Welcome to <i>Life in the Megapocalypse</i> (the prototype version, at least)! From here, you can get a bit more information about the project or start the game.

+[Get more information about this project.] // This is a "choice" - text that the user can click on. You create one using a "+" symbol and then placing text after it. Note that this is also a "bracketed" choice - it has [] brackets around it. Bracketed choices are not shown in the game window on the right if a player clicks on them.

->projectInfo //Anything you place inside of a choice will happen when your player clicks on it; since this divert is placed after or "inside" a choice, it only happens when the user clicks on the choice. This means that unlike the previous divert, this one does not happen until the player clicks on the "Get more info about this project" text.

[Start the game.] # CLEAR //Note that this is also a choice, but it is created with the "" symbol instead of the + symbol. Choices with a * get "used up" when the player clicks on them - you can't click on them again. Basically, you use + choices if you want the player to be able to make that choice more than once, and * choices if the player should only be able to make that choice once.

->gameStart

==gameStart //This knot starts the game itself! It is separate from the "Setup" and "Start" knots to keep things organized. If you want to know more about how the game works or how to modify it, start here!

*[I have never played before or I want a refresher on the game's introduction.]

->longIntro

*[I have played before and want to get right to the game!]

->gameStructure

=longIntro //This is a "stitch," which works exactly the same way as a knot, except that stitches are nested "inside" knots. You can create one using "=stitchName" and they are primarily used for story organization. In this example, the stitch called "LongIntro" is inside the knot called "GameStart" and contains some lengthy introductory text for the game. Structuring things this way allows the player to skip the intro if they have played the game before.

You are sitting in your office when a chime suddenly plays on the computer - a notification of an incoming message. That surprises you because things out there have gotten even worse than usual lately. When climate change started bringing giant ice storms that covered the earth everyone thought the world was ending, and you couldn't go anywhere without getting ambushed by desperate people who turned to banditry. That was before the dead came back to life and started eating people... and THAT was before the extraterrestrial invasion... and THEN came the killer robots. By the time the giant monsters began showing up you started to lose track of the apocalypses. # CLASS: playernotes

+Read the message. # CLASS: playernotes

"Hey, is anyone there? We found a note in this bunker telling us to send a message if we needed help - and we definitely do!" # CLASS: teamnotes

At this point, you can't believe anyone is still out there in the megapocalypse - the only term appropriate enough to describe the seemingly endless series of catastrophes that have struck the Earth. But the blinking amber text of the message on your screen is there, all the same. # CLASS: playernotes

++Answer the call. # CLASS: playernotes

You respond and ask them for a bit more information about who they are. Your job as The Gatekeeper has been quiet lately, but you are the person who guides people to the small safe haven you and your friends call home in the hopes that they can contribute valuable skills to the community. # CLASS: playernotes

- "We are a team of five survivors. Our last safe haven fell, and we've been trying to survive out here. We thought we were doomed until we found one of your bunkers and a note that told us to call. Are you somewhere safe?" # CLASS: teamnotes

+++Respond to their message. # CLASS: playernotes

- You chat with them a bit more, telling them about your community and finding out more about them. It seems like they have some useful talents, so after a short discussion, you offer to guide them to your community. With your advice, hopefully they can make it to the next 8 bunkers and then to your safe haven. # CLASS: playernotes

*Start guiding them along the road.->gameStructure # CLASS: playernotes # CLEAR

==gameStructure //This knot allows the player to either go talk to the party or to take on an encounter; it is essentially the "main" knot of the game in that it is used to track the player's progress and advance the story. It is useful to have a knot like this to keep track of the overall game progress, and if you are interested in modding the game, you can change lots of elements of the game by modifying things here: for example, there is lots of "story" content in this knot that shows up at different points in the game, so if you want to add new story content this is a good place to do so.

{charsLost == 4: ->endings} //This is a conditional: remember those variables from earlier? A conditional is one way you can "test" them and do things with them. You write one by using "{variableName: some text you want to show or logic you want to do.}" In this code, we check to see if the "CharsLost" variable is equal to 4, and if it is, we divert to a knot called "Endings."

{encounters == 8:->endings}

The team is ready to set out for {the first|the second|the third|the fourth|the fifth|the sixth|the seventh|the final} bunker - but maybe you should talk to the team first. # CLASS: playernotes //This is a "sequence," a special kind of text that varies each time the user sees it. You can create one by using "{someText|secondText|thirdText|...}" and when a player sees the sequence, the text the player sees will be based on how many times the player has visited the sequence before. In this example, the user sees "the first" the first time, "the second" the second time, "the third" the third time, etc. A sequence "stops" on its last element - so in this example, the user will see "the final" each time the user visits the sequence after the 8th time.

{abilities.guyMcMannis == 1:It looks like you have decided that violence is the best way to reach the safe haven. It doesn't look like it turned out too well, though - maybe you should try another approach?} # CLASS: fourthwall //Note that this is also a conditional, but here we just used the name of a knot and stich instead of checking a variable. Conditionals can check to see if a user has been to a certain knot or stitch by just using the name of the knot / stitch - you can write something like "{knotName.stichName:some text.}" This conditional checks to see if your player went to the "Abilities.GuyMcMannis" knot, and if so, it warns them not to take the violent approach that leads to the "bad" ending.

{abilities.guyMcMannis == 2:...well it looks like you are going to keep trying the violent route. Even though so far it has just caused the other characters to get separated from the team. You know, the ones that you can talk to. The ones who can do something other than just shoot at everything. Maybe you should try using one of them this time?} # CLASS: fourthwall //You can also check to see if someone visited a knot more than one time - knots and stiches are basically variables and work just like them - so here the game checks to see if the player has used the "bad" ability more than once and keeps warning them not to do it.

{abilities.guyMcMannis == 3:Hmm. It looks like you are really committed to the violent route, even though it has caused nothing but problems for your team. Maybe just this once you should try talking to your teammates or using someone else?} # CLASS: fourthwall {encounters == 7 and totalChats < 11:Reports indicate that storm activity has greatly increased, and you haven't learned enough about the team to form a decent plan. You definitely need to start learning more about them!} # CLASS: playernotes

{encounters == 7 and totalChats > 10:Reports indicate that storm activity has greatly increased, but you have learned enough about the team to form a decent plan!} # CLASS: playernotes

{encounters == 6: As the team travels, you strike up a conversation with the remaining team members about what their futures might hold.} # CLASS: playernotes

{~{encounters == 6 and not charDeath.charOne and not charAbandon.charOne:{charOneName}:I wonder if they are going to need history teachers? I guess someone has to remember what the world was like before everything fell apart. Though... there are probably some things that it might be worth leaving out.}|{encounters == 6 and not charDeath.charTwo and not charAbandon.charTwo:{charTwoName}:You know, I'm thinking once we are safe, I might try to start shooting nature documentaries all over again - they would certainly be a lot more exciting than the ones I used to do, and they might even keep someone alive out there.}|{encounters == 6 and not charDeath.charThree and not charAbandon.charThree:{charThreeName}:I wonder if they have any kind of training program set up for new doctors there? I'd love to help people out, but I'd love even more to make sure that we are teaching new people! If there is one thing we need after what I have seen out here, it's more doctors!}|{encounters == 6 and not charDeath.charFour and not charAbandon.charFour:{charFourName}:Considering that we have been communicating via text messaging with these folks, I can definitely tell I will be able to help out once we get to this place. They need to update their systems for sure!}} # CLASS: teamnotes

{encounters == 5 and totalChats < 6:Reports indicate that monster activity has greatly increased, and you haven't learned enough about the team to form a decent plan. You definitely need to start learning more about them!} # CLASS: playernotes

{encounters == 5 and totalChats > 5:Reports indicate that monster activity has greatly increased, but you have learned enough about the team to form a decent plan!} # CLASS: playernotes

{encounters == 4: As the team travels, they begin talking about dating and relationships.} # CLASS: teamnotes

{~{encounters == 4 and not charDeath.charOne and not charAbandon.charOne:{charOneName}:You know, it was always hard for me to figure out the whole dating thing, but I guess the one nice thing about the apocalypse is that it doesn't really matter anymore... unless your idea of a nice first date is a near-death bonding experience.}|{encounters == 4 and not charDeath.charTwo and not charAbandon.charTwo:{charTwoName}:I've seen a few cute people out here on the road - but at this point, I think I'm most interested in a date that doesn't involve keeping watch for all sorts of things that want to kill me and whoever I am with.}|{encounters == 4 and not charDeath.charThree and not charAbandon.charThree:{charThreeName}:One fun thing about the apocalypse is not having to figure out being poly. People don't ask too much about dating when you stitching up their wounds!"}|{encounters == 4 and not charDeath.charFour and not charAbandon.charFour:{charFourName}:You know, it's kind of cool that nobody worries about dating in the apocalypse - though it would be nice to be worried abotu a date for a change instead of what we will meet on the road.}} # CLASS: teamnotes

{~{encounters == 3 and not charDeath.charOne and not charAbandon.charOne:{charOneName}:It's funny... whenever I played video games about the apocalypse, I always liked the ones with zombies and aliens the most. Like, they always seemed the most fun and over the top, and I could never take them seriously, which is why I enjoyed then. Now I wish I had been paying more attention!}|{encounters == 3 and not charDeath.charTwo and not charAbandon.charTwo:{charTwoName}:One thing that amazes me is that people still find the time to try to rob one another even with all this destruction around - I mean, there are literally robot death squads roaming the earth and groups of bandits still get together and try to steal from us. And everyone wonders why I liked getting away from all the people and machines.}|{encounters == 3 and not charDeath.charThree and not charAbandon.charThree:{charThreeName}:I can't believe there are robots AND zombies out here trying to kill us! One tireless, unstoppable, murderous horde of enemies wasn't enough for this apocalypse?!?}|{encounters == 3 and not charDeath.charFour and not charAbandon.charFour:{charFourName}:I wonder why the bandits are always trying to rob us? You would think that they might try to steal some laser rifles or spaceships from the aliens or something. Then again, it might be tough to rob extraterrestrial beings with a shotgun.}} # CLASS: teamnotes

{encounters == 2:You have gotten reports that a deadly storm could be in the area. You should chat with the team to be sure you have a good plan in place, since these storms can be deadly.} #CLASS: playernotes {encounters == 1:You have gotten reports that a giant monster could be in the area. You might want to chat with the team and be sure you know enough about them to form a decent plan.} #CLASS: playernotes

+Talk to the members of the party. # CLASS: playernotes //This choice goes to the "chat" section of the game, so if you are interested in modifying the game's dialogue and story, you will want to look at that section.

->partyChat

+Give the party directions to the next landmark on the road! # CLASS: playernotes # CLEAR // This choice goes to the "encounters" section of the game, so if you are interested in modifying the gameplay of the game, you will want to look at that section.

->encounters

==charDeath //This knot contains most of the functionality necessary for handling and keeping track of character deaths. If you want to add a new character to the game or change the way character death works you will need to edit the content here.

=randCharDeath //This stitch is used to produce a random character death when neccessary.

{~->charOneCheck|->charTwoCheck|->charThreeCheck|->charFourCheck} //This is a "shuffle." You can create one using "{~someText|someMoreText|evenMoreText|...}" and they allow for randomization - a shuffle will randomly pick one of the options inside it. In this example there is a shuffle with four diverts inside it - so this shuffle will randomly divert to either the Char1Check, Char2Check, Char3Check, or Char4Check stitch.

=charOneCheck //These stitches are used to check if the relevant character is already gone or dead before moving on to the death sequence.

{not charAbandon.charOne and not charOne:->charOne|->randCharDeath} //You can use "not" in your code to check if a certain condition has NOT been met and then do things based on that: for example, here the game checks to see if the character is still alive before the game kills that character.

//Other stiches for abandon checks for other characters go here

=charOne //These stitches kill various characters and also keep track of the fact that they are dead.

~charsLost = charsLost + 1 //You can change the value of a variable you have already created using the "~" symbol. An "=" sign will allow you to tell the game what the new value should be. In this example, the code sets the value of the CharsLost variable to a new value that is 1 higher than the previous value so the game can track how many characters have been lost or killed.

Unfortunately things did not go well and {charOneName} has been killed in the encounter with the {TURNS_SINCE(->encounters.bandits) == 1:bandits}{TURNS_SINCE(->encounters.robots) == 1:robots}{TURNS_SINCE(->encounters.zombies) == 1:zombies}{TURNS_SINCE(->encounters.aliens) == 1:aliens}{TURNS_SINCE(-

>encounters.monster) == 1:monster}{TURNS_SINCE(->encounters.climate) == 1:ice storm}. # CLASS: blue //This is a "TURNS_SINCE," a special kind of conditional. They can be created in a few ways, but the basic format is something like "{TURNS_SINCE(-> knotName.StichName) == X}" and they allow you to check how many choices the user has made since they have been to a particular knot or stitch. In this example, the game displays the text only if the player has made exactly one choice since visiting the encounters section of the game. This allows the game to track which encounter just happened and pull up the appropriate text, so if you add new encounters you will need to modify this code.

->gameStructure

//Other stiches for randomly killing other characters here

==charAbandon //This knot contains most of the functionality necessary for handling and keeping track of character abandons. It works basically the same way as the random death system, so if you add another character to the game you will need to modify this code as well.

=randCharAbandon //This stitch is used to produce a random character abandon when neccessary.

{~->charOneCheck|->charTwoCheck|->charThreeCheck|->charFourCheck}

=charOneCheck

{not charDeath.charOne and not charOne:->charOne|->randCharAbandon}

```
//Stiches to check for other character deaths go here

    =charOne //These stitches have various characters abandon the group and also keep track of
the fact that they did so.


    ~charsLost = charsLost + 1


    {charOneName} got separated from the group. # CLASS: blue


    ->gameStructure
//Stitches to have the other characters abandon the group go here
==abilities //This knot is used to keep track of character abilities - i.e. how good or bad at a
particular scenario the character is.  The stitches inside are accesed during encounters when the
player chooses to use a certain character.  Individual success chances, text, and other stuff can be
quickly edited here and new encounters can be easily added if you are interested in modifying
that kind of content You can see examples of how to create new character abilities here as well.
If you make a new character or a new encounter you will need to edit this content.


//Note that I built the game so that talking to the characters can improve success chances on the
encounters: for example, learning a character's backstory gives a slightly better chance on certain
```

encounters, and learning more about the team in general can improve success chances on the "monster" and "climate" encounters, which no individual team member is good at. This design encourages players to explore the narrative and talk to the characters instead of simply replaying the encounters over and over to learn the team's strengths and weaknesses. If you modify the character abilities, you might also want to modify how the chat system works as well.

```
=charOne

{TURNS_SINCE(-> encounters.bandits) == 1:{charOneName}'s diplomacy skills could be especially helpful in talking down the bandits.->results.good} # CLASS: blue

{charOneStoryHeard and TURNS_SINCE(->encounters.zombies) == 1:Knowing {charOneName}'s story might help you lead the group past the zombies.->results.story} # CLASS: blue

{TURNS_SINCE(->encounters.zombies) == 1:{charOneName} tries to lead the group past the zombies.->results.moderate} # CLASS: blue

{charOneStoryHeard and TURNS_SINCE(->encounters.aliens) == 1:Knowing {charOneName}'s backstory might help you lead the group past the aliens.->results.story} # CLASS: blue

{TURNS_SINCE(->encounters.aliens) == 1:{charOneName} tries to lead the group past the aliens.->results.moderate} # CLASS: blue

{TURNS_SINCE(->encounters.robots) == 1:{charOneName} has never been a big fan of working with machines.->results.bad} # CLASS: blue
```

{totalChats > 5 and charsLost < 3 and TURNS_SINCE(->encounters.monster) == 1:Knowing more about the team might help you use {charOneName}'s strengths along with those of the other teammates to get past the monster.->results.story} # CLASS: blue

{TURNS_SINCE(->encounters.monster) == 1:{charOneName} tries to lead the group past the monster, but unfortunately, there are no great choices for dealing with monsters on one's own.->results.moderate} # CLASS: blue

{TURNS_SINCE(->encounters.climate) == 1:{charOneName} tries to lead the group past the giant storm, but there is no good way of dealing with the problem.->results.bad} # CLASS: blue

{totalChats > 11 and charsLost < 3 and TURNS_SINCE(->encounters.climate) == 1:The storm is devastating, but knowing so much about the team might help you and {charOneName} come up with a decent plan for getting past it using teamwork.->results.moderate} # CLASS: blue

{TURNS_SINCE(->encounters.climate) == 1:{charOneName} tries to lead the group past the giant storm, but there is no good way of dealing with the problem.->results.bad} # CLASS: blue

//Stiches for the other character’s abilities go here

=guyMcMannis //Using Guy McMannis' ability always causes another character to abandon the team. It works the same way every time to encourage the player not to use it and to instead explore nonviolent ways of resolving the game's encounters that the other characters are good at. Using it repeatedly leads to the "bad" ending.

```
    "I will save us all!" shouts Guy McMannis, drawing his guns and charging into the fray.  In the ensuing chaos... # CLASS: red
    ->charAbandon.randCharAbandon
==results  //This knot is used to generate the results using various character abilities.  Good results give a 3/4 chance of surviving with no character death; Moderate give a 50% chance; Bad give a 25% chance.  Chatting with the characters gives the player hints at which choices are good vs. bad and improves their chances of success in  many situations. The "backstory" result is used in special situations and rewards the player for learning more about the team.  You can easily tweak success chances or add new categories here.
    =good
    Good choice! # CLASS: good
    {~->noDeaths|->noDeaths|->noDeaths|->charDeath.randCharDeath}
    =story
    Knowing more about the team made this a better than average choice! # CLASS: backstory
    {~->noDeaths|->noDeaths|->noDeaths|->charDeath.randCharDeath|->charDeath.randCharDeath}
    =moderate
    An okay choice. # CLASS: okay
    {~->noDeaths|->noDeaths|->charDeath.randCharDeath|->charDeath.randCharDeath}
    =bad

    A bad choice. # CLASS: bad

    {~->noDeaths|->charDeath.randCharDeath|->charDeath.randCharDeath|-
>charDeath.randCharDeath}

    =noDeaths

    No one died! # CLASS: playernotes

    ->gameStructure
```

==encounters //This knot is used to create and keep track of encounters. Most of the logic for the encounters is handled inside of stitches inside of this knot, but having this knot allows the game to track how many encounters have taken place. It also allows for the easy addition of more random encounters or even different random encounter generators if you want to make those kinds of modifications.

~charOneChatted = false //These reset the "chat" variables so that the player can talk to the party again. The player can talk to each character once but then the player has to do an encounter before chatting with the characters again.

//Resets for other chat variables go here

->randEncounter

    =randEncounter //This code produces random encounters. I nested this code inside of a stich so that it would be easy to add new random encounter generators or modify this one. You can create a new randEncounter stitch, copy this code, modify it, and then send the player to the new stitch you created to create an entirely different encounter system.

{~You receive a message from the group!|A notification from the team has just come in!|The survivors are trying to contact you!} # CLASS: teamnotes

Hey, we have run into some trouble on the road and need your advice! <> # CLASS: teamnotes

{encounters == 8:{~->bandits|->zombies|->aliens|->robots|->climate|->climate|->climate|->climate}} //This code keeps track of which encounter the player is on and then produces an appropriate random encounter based on that. It shows off one important function of Ink - you can "nest" almost anything inside of anything else! So in this example, I have a conditional with a shuffle nested inside of it, which means the shuffle won't happen unless the condition is met. If you want to mod the game, you can use this functionality to create all sorts of different gameplay and narrative content!

{encounters == 7:{~->bandits|->zombies|->aliens|->robots|->monster|->monster|->climate|->climate}}

{encounters == 6:{~->bandits|->zombies|->aliens|->robots|->monster|->monster|->monster|->monster}}

{encounters == 5:{~->bandits|->zombies|->aliens|->robots|->monster|->monster}}

{encounters == 4:{~->bandits|->zombies|->aliens|->robots|->climate}}

{encounters == 3:{~->bandits|->zombies|->aliens|->robots|->monster|->climate}} {encounters == 2:{~->bandits|->zombies|->aliens|->robots|->monster}}

{encounters == 1:{~->bandits|->zombies|->aliens|->robots}}

=bandits //These stitches are the actual individual encounters, so if you want to add new encounters you can add new stitches here and then send the player to those stitches.

We have been held up by a group of well-armed bandits who are trying to rob the team! # CLASS: teamnotes

->encounterController

//Stitches for the other encounters go here

=encounterController //This is where you choose which character you want to use in the encounter. If you want to add new characters or abilities you will need to modify this code; you also need to modify the abilities section of the code.

What advice do you have for the team? # CLASS: playernotes

+{not charDeath.charOne and not charAbandon.charOne}<p style = "color:skyblue;">Advise {charOneName} to use {charOnePossessiveAdj} conversation skills to help deal with the situation.</p> # CLASS: blue

->abilities.charOne//This is a "conditional choice" - the user can only make the choice if the user meets the appropriate condition. In this example, the user must NOT have been to the

CharacterDeath.Char1 or CharacterAbandoned.Char1 stitches - if the user has been to those stitches, the choice is unavailable because that character is dead.

```
//Choices for other the other characters go here

    +<p style = "color:red;">Advise Guy McMannis' to use his superior firepower to handle things, which is clearly the right choice!</p> # CLASS: red

    ->abilities.guyMcMannis

==partyChat //The next section holds logic that controls who the player can talk to.  The stitch for Guy McMannis' chats are in this section because he only has one which is based on whether or not the player has used his ability yet.  Chats for the other characters are held in their own knots so that stitches within each knot can help to organize the chat flow. If you want to modify the game's dialogue, this is where you should be looking! If you add another character you can copy this code to create that character's dialogue.

    You decide to take some time to chat with the members of the team and find out  more about them before they set out for the next bunker. # CLASS: playernotes

    +{not charDeath.charOne and not charAbandon.charOne}{not charOneChatted}<p style = "color:skyblue;">Chat with {charOneName}.</p>  # CLASS: blue

    ->charOneChats

//Choices for chatting with the other characters go here

    +<p style = "color:red;">Chat with GuyMcMannis.</p> # CLASS: red
    ->partyChat.guyMcMannis

    +Continue along the road! # CLASS: playernotes # CLEAR

    ->encounters

    =guyMcMannis

    {abilities.guyMcMannis:"Good to see you made the right choice in letting me handle things!"}
#CLASS: red

    {not abilities.guyMcMannis:"Hmm, seems like you are a bit of a coward... maybe you should
try having me lead the group next time!"} # CLASS: red

    ->partyChat

==charOneChats //Knots, stitches, and logic for each party member's chats can be found below.
Chats reveal backstory about the character as well as have give some hints about the encounters
they might be good or bad at. If you want to add additional characters or chat content you can
modify the code below.

~charOneChatted = true

  {charOneStrongHeard:{charOneWeakHeard:{charOneStoryHeard:You have finished
{charOneName}'s chats!->partyChat}}}

+Message {charOneName} and chat. # CLASS: playernotes

->charOneRandChat

+Talk to someone else. # CLASS: playernotes
```

->partyChat

=charOneRandChat //This stitch determines whether or not the character is in the mood to chat. It uses a shuffle that gives the player a 75% chance to talk to the character and a 25% chance the the character will not be interested at the time.

{~->randYes|->randYes|->randYes|->randNo}

=randNo

{charOneName} is not in the mood for chatting right now. # CLASS: blue

->partyChat

=randYes //This logic produces a random chat that the player has not seen yet for that character. If you add a new character you can copy this code to easily do the same for that character, but if you add new chats for the characters you will have to modify how this logic works.

{charOneStrongHeard and charOneWeakHeard and not charOneStoryHeard: ->charOneStory}

{charOneStrongHeard:->charOneWeak}

{charOneWeakHeard:->charOneStrong}

{not charOneStrongHeard and not charOneWeakHeard:{~->charOneStrong|->charOneWeak|}}

```
=charOneStrong //The "strong" chat for each character tells you what they are good at.

~charOneStrongHeard = true

~totalChats = totalChats + 1

You start chatting with {charOneName}. After a short conversation, you ask {charOneObject} to tell you a bit more about {charOnePossessiveAdj} strengths to get a sense for how {charOneSingular} might lead the team best in a dangerous situation. # CLASS: playernotes

{charOneName}:# CLASS: blue

"Hmm... well I haven't always liked talking to people, but I'd say that is my biggest strength. I found out I was good at it when I took a insurance sales job right out of college... my History degree wasn't bringing in any paychecks.  I was actually pretty talented at selling things to people even though I hated doing it."  #CLASS: blue

    +You ask {charOneObject} what happened when the apocalypse hit. # CLASS: playernotes

    {charOneName}:# CLASS: blue

    "Well, I had quit my job and gone back to grad school just a year before the apocalypse hit, and after going to a few academic conferences, I found I actually liked talking to people when it was about something I was passionate about." # CLASS: blue

    ->partyChat

=charOneWeak //The "weak" chat for each character tells you what they are bad at.

~charOneWeakHeard = true
```

~totalChats = totalChats + 1

You start chatting with {charOneName}. After a short conversation, you ask {charOneObject} to tell you a bit more about {charOnePossessiveAdj} weaknesses so that you can try to keep the team safe on the road by avoiding them. # CLASS: playernotes

{charOneName}:# CLASS: blue

"I have always hated working with machines... even talking to you on this thing isnt really my favorite. I'm pretty sure it came from growing up... my dad loved to work on cars and always tried to get me interested, but after several failed attempts to learn - and lots of yelling matches with him - I decided I was better off just taking my car to a mechanic and being done with it." #CLASS: blue

+You ask {charOneObject} about the robot extermination squads. # CLASS: playernotes

{charOneName}:# CLASS: blue

"Yeah, I don't really like working with any kind of machines... I don't even particularly like working with computers, though I eventually learned my way around them out of necessity. So let's keep me away from the killer robots." # CLASS: blue

->partyChat

=charOneStory //The "backstory" chat for each character gives some background about each character - it usually focuses on how they grew up and their views on relationships, gender, and sexuality. This chat won't show up until the player has already heard the character's strengths and weaknesses- the idea is that the characters don't share these more intimate details until they trust the player more.

~charOneStoryHeard = true

~totalChats = totalChats + 1

You start chatting with {charOneName}. After a short conversation, you ask {charOneObject} to tell you a bit more about {charOnePossessiveAdj} life to get a sense for how {charOneSingular} might fit into the community. Since you have already chatted quite a bit with {charOneName}, {charOneSingular} is willing to talk more about {charOnePossessiveAdj} personal life. # CLASS: playernotes

{charOneName}:# CLASS: blue

"I grew up pretty happy... I got along with my parents, did well in school, and had plenty of hobbies. I didn't necessarily have a lot of friends, but I was very close with the ones that I did have. People were sometimes surprised that I was kind of shy and quiet when I was younger because talking to people was such a big part of my job." #CLASS: blue

+You ask {charOneObject} about their relationships.

"Relationships were the one thing I always had trouble figuring out. I never felt like I was attracted to anyone until I got to know them closely, and never felt like I looked at women the same way as other guys. I struggled with things like dating and thought it was just anxiety or being shy, and it took me awhile to figure out that there was a word for what I was, mostly

```
because things like "asexual" and "demisexual" weren't talked about much until recently."
#CLASS: blue

        ++You ask {charOneObject} how things are now. # CLASS: playernotes

        {charOneName}:# CLASS: blue

        "Well, when the apocalypse happened it actually made that kind of stuff easier for me...
people were less worried about dating and sex when death was around every corner. It's kind of
nice not having to worry about those things as much anymore, though it would still be better if
everything wasn't trying to kill us." # CLASS: blue

            ->partyChat

//Chats for all the other characters go here.  While I have removed the redundant sections of code
throughout most of this code example, I’ve left in the sections that provide narrative content for
charFour as well as an example of what another character’s story looks like.

  =charFourStrong

  ~charFourStrongHeard = true

  ~totalChats = totalChats + 1

  You start chatting with {charFourName}. After a short conversation, you ask
{charFourObject} to tell you a bit more about {charFourPossessiveAdj} strengths to get a sense
for how {charFourSingular} might lead the team best in a dangerous situation. # CLASS:
playernotes
```

{charFourName}:# CLASS: purple

"I started working with machines at a young age since mom and dad were both computer programmers and liked to work on machines and build computers for fun in their spare time. I always loved helping them put stuff together and figuring out how things worked and I was fascinated by software. I was coding by 8 years old and built my first computer by the time I was 10, and I kept up with it as a hobby all the way until college when I decided to go to school for video game design." #CLASS: purple

+You ask {charFourObject} how things went after the apocalypse. # CLASS: playernotes

{charFourName}: # CLASS: purple

"I had actually gone to work for a pretty big company and was working on their next big release when, all of the sudden, the end of the world happened. At first I figured a computer nerd like me wouldn't make it, but it turned out working with machines was pretty useful when killer robots suddely started roaming the earth." # CLASS: purple

->partyChat

=charFourWeak

~charFourWeakHeard = true

~totalChats = totalChats + 1

You start chatting with {charFourName}. After a short conversation, you ask {charFourObject} to tell you a bit more about {charFourPossessiveAdj} weaknesses so that you can try to keep the team safe on the road by avoiding them. # CLASS: playernotes

{charFourName}:# CLASS: purple

"I never got everyone's fascination with the undead - it was always weird to me. Frankly, the idea of things like zombies had always scared me a little bit, and I stayed away from a lot of video games because of it. The idea of being eaten alive always freaked me out too, and I never liked dead bodies, much less ones that walked around, so the idea of them getting up, shambling about, and eating people really bothered me." #CLASS: purple

+You ask if the team should keep {charFourObject} away from any zombies. # CLASS: playernotes

{charFourName}: #CLASS: purple

"Definitely. Unluckily for me, zombies aren't fictional anymore - the apocalypse happened and they started walking around the earth for real. Maybe if I had known that was going to happen I'd have paid more attention to all those movies and shows and games. But it would be awesome if someone else could handle the flesh-eating corpses, if that's alright with you." # CLASS: purple

->partyChat

=charFourStory

~charFourStoryHeard = true

~totalChats = totalChats + 1

You start chatting with {charFourName}. After a short conversation, you ask {charFourObject} to tell you a bit more about {charFourPossessiveAdj} life to get a sense for how {charFourSingular} might fit into the community. Since you have already chatted quite a bit with {charFourName}, {charFourSingular} is willing to talk more about {charFourPossessiveAdj} personal life. # CLASS: playernotes # CLASS: playernotes

{charFourName}:# CLASS: purple

"I was always close with my parents while I was growing up. I was close with my wider family too, and some of my best friends were cousins. None of them seemed to mind when I began transitioning in high school - in fact, they were some of my strongest supporters! The only problems I had ever faced came from other people, but my family always had my back." #CLASS: purple

+You ask about {charThreePossessiveAdj} experience with relationships. # CLASS: playernotes

{charFourName}: # CLASS: purple

"Relationships were tricky for me at first, but a lot of that was my fault: I can be a bit standoffish in person, though talking on the computer like this makes it easier for me. Online dating helped me a lot with that since I was more comfortable talking with people that way, and I had a few steady boyfriends before the world fell apart." # CLASS: purple

++You ask if things have been easier or harder since the apocalypse hit. # CLASS: playernotes

{charFourName}: # CLASS: purple

"Honestly, missing out on my love life hasn't been much of a big deal to me - losing my family during the apocalypse hit me much harder. At least they were spared the hell of trying to survive now. That being said, it seems like people care so much about survival that they aren't worried about anything like dating, so I guess it's gotten easier in some ways." # CLASS: purple

->partyChat

==endings //The game uses one big knot to keep track of endings with lots of stitches and conditionals inside. This structure allows the game to keep track of the game state and what the player has done and then to show the player ending content based on those factors. You can make lots of modifications to the way the game ends by editing the various stitches inside this knot, so if you want to add a new ending or more ending content this is where you should start!

The team makes it to the gates of the safe zone. You can take a moment to review your notes about their journey or simply start inviting people into the safe haven. # CLASS: playernotes

+Review your notes about which encounters the team survived. # CLASS: playernotes

->encounterEndings

+Decide who to invite in. # CLASS: playernotes

->finalChoice

```
=encounterEndings //This stitch tracks how far the player got. You do not need to modify it unless you want to add more than 8 encounters.

The team survived the following encounters: # CLASS: playernotes

{encounters > 0:Encounter 1} # CLASS: playernotes

{encounters > 1:Encounter 2} # CLASS: playernotes

{encounters > 2:Encounter 3} # CLASS: playernotes

{encounters > 3:Encounter 4} # CLASS: playernotes

{encounters > 4:Encounter 5} # CLASS: playernotes

{encounters > 5:Encounter 6} # CLASS: playernotes

{encounters > 6:Encounter 7} # CLASS: playernotes

{encounters == 8:Encounter 8} # CLASS: playernotes

+{charDeath.charOne or charDeath.charTwo or charDeath.charThree or charDeath.charFour}Review your notes about which members of the team died. # CLASS: playernotes

->deaths

+{abilities.guyMcMannis}Review your notes about who abandoned the team. # CLASS: playernotes

->abandons
```

```
+{not charDeath.charOne and not charDeath.charTwo and not charDeath.charThree and not charDeath.charFour and not abilities.guyMcMannis}Review your notes about who you chatted with. # CLASS: playernotes

->chats

+Decide who to invite in. # CLASS: playernotes

->finalChoice

=deaths //This stitch tracks character deaths. You do not need to modify it unless you add a new character or change the death system.

The following characters died: # CLASS: playernotes

{charDeath.charOne:{charOneName}} # CLASS: blue

{charDeath.charTwo:{charTwoName}} # CLASS: green

{charDeath.charThree:{charThreeName}} # CLASS: orange

{charDeath.charFour:{charFourName}} # CLASS: purple

+{abilities.guyMcMannis}Review your notes about who abandoned the team. # CLASS: playernotes

->abandons

+{not abilities.guyMcMannis} Review your notes about who you chatted with. # CLASS: playernotes
```

```
->chats

+Decide who to invite in. # CLASS: playernotes

->finalChoice

=abandons //This stitch tracks character abandons. You do not need to modify it unless you add a new character or change the abandon system.

{abilities.guyMcMannis:You got {abilities.guyMcMannis==4:4}{abilities.guyMcMannis == 3:3}{abilities.guyMcMannis == 2:2}{abilities.guyMcMannis == 1:1} people to abandon the team by advising the team to take the violent route. Maybe it would be better to try another way next time?} # CLASS: red

but.. # CLASS: playernotes

{charAbandon.charOne:Luckily, {charOneName} made it to the safe haven alone!} # CLASS: blue

{charAbandon.charTwo:Luckily, {charTwoName} made it to the safe haven alone!} # CLASS: green

{charAbandon.charThree:Luckily, {charThreeName} made it to the safe haven alone!} # CLASS: orange

{charAbandon.charFour:Luckily, {charFourName} made it to the safe haven alone!} # CLASS: purple

+Review your notes about who you chatted with.
```

->chats

+Decide who to invite in. # CLASS: playernotes

->finalChoice

=chats //This stitch tracks what conversations the player had. You do not need to modify it unles you change the dialogue system; if you add a new character, you can simply copy the code here. Note that the current tracking system will need to be changed heavily if you add new chats other than the "strong," "weak," and "backstory" chats because this code is designed to check for any potential combination of those three.

These were the results of your conversations: # CLASS: playernotes

//Char1 tracking

{charOneStrongHeard:{charOneWeakHeard:{charOneStoryHeard:You heard all of {charOneName}'s chats!}}} # CLASS: blue

{not charOneStrongHeard:{not charOneWeakHeard:{not charOneStoryHeard:You didn't hear any of {charOneName}'s chats.}}} # CLASS: blue

{charOneStrongHeard:{not charOneWeakHeard:{not charOneStoryHeard:You found out about {charOneName}'s strengths, but not {charOnePossessiveAdj} weaknesses or backstory.}}} # CLASS: blue

{charOneWeakHeard:{not charOneStrongHeard:{not charOneStoryHeard:You found out about {charOneName}'s weaknesses, but not {charOnePossessiveAdj} strengths or backstory.}}} # CLASS: blue {charOneStrongHeard:{charOneWeakHeard:{not charOneStoryHeard:You heard about {charOneName}'s strengths and weaknesses, but not {charOnePossessiveAdj} backstory.}}} # CLASS: blue

//Stitches for tracking the other characters go here

+Decide who to invite in. # CLASS: playernotes

->finalChoice

=finalChoice //At this point, the game branches slightly - if the player has only chosen the violent route, the final choice does not happen and the player gets the "worst" ending. If not, the player gets the chance to invite people in to the safe haven. If you add a new character you will need to modify this section accordingly.

{abilities.guyMcMannis == 4:You directed Guy McMannis to lead the team over and over again, leading everyone else on the team to abandon him to the wasteland because of his violent approach. Luckily, everyone else on the team made it to the safe haven on their own. Eventually GuyMcMannis shoots his way to the safe haven; when he arrives, everyone else is bewildered, as none of them can understand why you thought he would be a useful contribution to the community. His former comrades tell their stories, reminding everyone that they abandoned him because of his senseless, violent approach. The community decides on a simple solution: since you seem to prefer the violent lifestyle of people like GuyMcMannis, you are exiled from the safe haven along with him, and the two of you can wander the megapocalypse for the rest of your short lives.->finalEnd} # CLASS: red

You finish reviewing your notes about the journey as the team arrives at the gate. Now you have one final job - offering these hardened survivors refuge in the hopes that they will lend their valuable skills into the community. # CLASS: playernotes

*{not charDeath.charOne and not charAbandon.charOne}<p style = "color:skyblue;">Offer {charOneName} a place in the safe haven.</p> # CLASS: blue

->charOneFinal

//Choices for all the other characters go here

*<p style = "color:red;">Offer Guy McMannis a place in the safe haven.</p> # CLASS: red

->mcMannisFinal

*I am done offering people a place in the safe haven. # CLASS: playernotes

->storyEnd

=charOneFinal //This section checks to see if the player fully explored each character's chats. If the player did, the character will join the safe haven; if the player did not, the character will refuse.

{charOneWeakHeard == true and charOneStrongHeard == true and charOneStoryHeard == true: ->charOneSafe}

{charOneName} considers his options but is not interested in entering the safe haven because {charOneSingular} hasn't heard much about it. # CLASS: blue

->finalChoice

=charOneSafe

{charOneName} considers {charOnePossessiveAdj} options and since {charOneSingular} has communicated with you a lot {charOneSingular} decides to enter the safe haven. # CLASS: blue

->finalChoice

//Stitches for all the other characters go here

=mcMannisFinal //This section checks to see if the player offers GuyMcMannis a spot in the safe haven and immediately ends the game if the player does.

->storyEnd

=storyEnd //This stitch ends the story after the final choice and provides the final story content of the game. You can modify this content to change the game's endings, and if you add a new character you can use this content as a guide for what you need to include.

You finished deciding who to invite in to the safe haven. These were the results of your choices: # CLASS: playernotes

{mcMannisFinal:You tried to allow Guy McMannis into the safe haven, but everyone else voted you down and decided that you should be exiled alongside him instead.->finalEnd} # CLASS: red

{charDeath.charOne:{charOneName} died before reaching the safe haven.} # CLASS: blue

//Conditionals for all other characters go here

```
    {charAbandon.charOne:{charOneName} got separated from the team but made it to the safe
haven on {charOnePossessiveAdj} own.} # CLASS: blue

//Conditionals for all other characters go here

    {not charDeath.charOne and not charAbandon.charOne and not charOneFinal:You did not
invite {charOneName} into the safe haven.} # CLASS: blue

//Conditionals for all other characters go here


    {charOneFinal and not charOneSafe:What {charOneName} did after deciding not to enter the
safe haven is unknown, but you have to wonder if {charOneSingular} could actually survive out
there.} # CLASS: blue

//Conditionals for all other characters go here

    {charOneSafe:{charOneName} fit in well - it turns out people actually wanted someone who
knew history and could teach it to the young people of the community. Once {charOneSingular}
became an established speaker, {charOneSingular} ended up as one of the foremost experts on
the "prepocalypse," and the study of prepocalypse history became one of the most popular
subjects in school.} # CLASS: blue

//Endings for all other characters go here

    ->finalEnd
```

```
=finalEnd //This stich ends the story; it is separate from all the rest so that the game can easily create an ending whenever necessary and to allow for restarts.

*[Restart the game?] (NOTE - ONLY WORKS IN WEB VERSION)

# RESTART

->END

*[Nah I am good!]

->END
```

LIST OF REFERENCES

Al-Washmi, R. et al. (2014). Design of a math learning game using a Minecraft mod. In *Proceedings of the 8th European Conference on Games Based Learning (ECGBL 2014)*, 1. DOI: 10.13140/2.1.4660.4809

Allen, S. (2014). Closing the gap between queer and mainstream games. *Polygon.* Retrieved from https://www.polygon.com/2014/4/2/5549878/closing-the-gap-between-queer-and-mainstream-games

Andrews, J. and Schmidt, L. (2014). Realistic kissing simulator. Jimmylands.com

Anthropy, A. (2012). Dys4ia. Newgrounds.

Apperly, T. and Beavis, C. (2013). A model for critical games literacy. *E-Learning and Digital Media 10*(1), pp. 1 – 12. Retrieved from https://research-repository.griffith.edu.au/bitstream/handle/10072/61222/88437_1.pdf

Aron, A., Melinat, E., Aron, E. N., Vallone, R. D., and Bator, R. J. (1997). The experimental generation of interpersonal closeness: A procedure and some preliminary findings. *Personality and Social Psychology Bulletin 23*(4), pp. 363 – 377.

Bagnall, G. (2017). Queer(ing) gaming technologies: Thinking on constructions of normativity inscribed in digital gaming hardware. *Queer game studies,* eds. B.Ruberg A. Shaw. Minneapolis, MN: University of Minnesota Press. pp 135– 144.

Beavis, C. et.al. (2015). Turning around to the affordances of digital games: English curriculum and students' lifeworlds. *English in Australia 50*(2), pp 30 – 40. Retrieved from http://dro.deakin.edu.au/eserv/DU:30080156/bradford-turningaround-2015.pdf

Bethesda Game Studios (2015). Fallout 4. Bethesda Softworks.

Bethesda Game Studios (2011). The Elder Scrolls V: Skyrim. Bethesda Softworks.

Bioware (2011). Dragon age II. Electronic Arts.

Bioware (2014). Dragon age: Inquisition. Electronic Arts.

Bioware (2009) Dragon age: Origins. Electronic Arts.

Blizzard Entertainment (2016). Overwatch. Blizzard Entertainment.

Bogost, I. (2006). *Unit operations: An approach to videogame criticism*. Cambridge, MA: The MIT Press.

Bogost, I. (2007). *Persuasive games: The expressive power of videogames.* Cambridge, MA: The MIT Press.

Boluk, S. and Lemieux, P. (2017) *Metagaming: Playing, competing, spectating, cheating, trading, making, and breaking videogames*. Minneapolis, MN: University of Minnesota Press.

boringvlln (2019). Simply gay letters [computer game modification for The Elder Scrolls V: Skyrim]. *Nexusmods.* Retrieved from http://www.nexusmods.com/skyrimspecialedition/mods/26423

Butler, J. (2004). *Undoing gender.* New York, NY: Routledge.

Brown, C.L., Comunale, M.A., Wigdahl, B., and Urdaneta-Hartmann, S. (2018). Current climate for digital game-based learning of science in further and higher education. *FEMS Microbiology Letters* 365, 21. DOI: 10.1093/femsle/fny237

bxbblegumbxtch (2016). Boibomb – A femboy CBBE preset [computer game modification for Fallout 4. *Loverslab.* Retrieved from https://www.loverslab.com/files/file/3556-boibomb-a-femboy-cbbe-preset/

Champion, E. (2012). *Game mods: Design, theory, and criticism.* Pittsburgh, PA: ETC Press.

Chang, E. (2017). Queergaming. *Queer game studies,* eds. B.Ruberg A. Shaw. Minneapolis, MN: University of Minnesota Press. pp 15 – 24.

Clark, N. (2017). What is queerness in games, anyway? *Queer game studies,* eds. B.Ruberg A. Shaw. Minneapolis, MN: University of Minnesota Press. pp 3 – 14.

Clyde, J. and Thomas, C. (2008). Building an information literacy first-person shooter. *Reference Services Review* 36, 4, pp. 366-380.

Cool Math Games (1997). Parking mania. Cool Math Games.

Cool Math Games (1997). Traffic mania. Cool Math Games.

Costikyan, G. (2002). I have no words and I must design. *Computer games and digital cultures conference proceedings 1*(1). Tampere, Finland: Tampere University Press. Retrieved from http://www.digra.org/wp-content/uploads/digital-library/05164.51146.pdf

DeWinter, J. et. al. (2010). Computer games across the curriculum: A critical review of an emerging techno-pedagogy. Retrieved from https://currents.dwrl.utexas.edu/2010/dewinter_et_al_computer-games-across-the-curriculum.html

Dreamfeel (2014). Curtain. Dreamfeel.

Eng, D.L, Halberstam, J., and Muñoz, J.E. (2005). "What's queer about queer studies now?" *Social Text 3-4* (84 – 85), pp. 1 – 18.

Electronic Arts Inc. (2015). Electronic Arts reports Q4 FY15 and full year FY15 financial results. *Electronic Arts Financial Information.* Retrieved from https://s22.q4cdn.com/894350492/files/doc_financials/2015/q4/Q4_FY15_Earnings_Release.pdf

Elkins, A.J. (2015). Let's play: Why school librarians should embrace gaming in the library. *Knowledge Quest* 43, 5, pp. 58 – 63.

El-Nasr, M. S. and Smith, B.K. (2006). Learning through game modding. ACM *Computers in Entertainment*, 4, 1.

Feldman, B. (2016). Overwatch has a gay character and horny games are melting down. *NYMag.* Retrieved from https://nymag.com/intelligencer/2016/12/horny-gamers-are-melting-down-over-overwatchs-gay-character.html

Firaxis Games (2005). Civilization IV. 2K Games.

Firaxis Games (2010). Civilization V. 2K Games.

Filipowich, M. (2018). Maidens and muscleheads, white mages and wimps, from the light warriors to Lightning Returns. *Queerness in Play*, eds. T. Harper, M. B. Adams, and N. Taylor. pp. 115 – 131.

Gaider, D. (2013). Sex in video games. *GDC vault.* Retrieved from https://www.gdcvault.com/m/play/1017796/Sex-in-Video

Game Grumps (2017). Dream daddy. Game Grumps.

Gee, J.P. (2003). *What video games have to teach us about learning and literacy.* Palgrave Macmillan, Hampshire, England.

GoesOnGhost (2018). Complete Bi Overhaul [computer game modification for Dragon age: Origins]. *NexusMods.* Retrieved from https://www.nexusmods.com/dragonage/mods/4659/

Greer, S. (2013). Playing queer: Affordances for sexuality in Fable and Dragon Age. *Journal of Gaming and Virtual Worlds 5*(1): pp 3 – 21.

Halberstam, J. (2017). Queer gaming: Gaming, hacking, and going turbo. *Queer game studies,* eds. B.Ruberg A. Shaw. Minneapolis, MN: University of Minnesota Press. pp 187 – 200.

Harper, T (2017). Role-play as queer lens: How "ClosetShep" changed my vision of Mass Effect. *Queer game studies,* eds. B.Ruberg A. Shaw. Minneapolis, MN: University of Minnesota Press. pp 125–134.

Hopkins, I. and Roberts, D. (2015). 'Chocolate covered broccoli'? Games and the teaching of literature. *Changing English: Studies in Culture and Education* 22, 2, pp. 222-236. DOI: https://doi.org/10.1080/1358684X.2015.1022508

Hovious, A. 2014. Inanimate Alice. *Teacher Librarian* 42, 2.

Howard, K. (2019). Romance never changes… or does it: *Fallout,* queerness, and mods. *DiGRa '19 – Proceedings of the 2019 International Conference.* Retrieved from http://www.digra.org/wp-content/uploads/digital-library/DiGRA_2019_paper_90.pdf

Howard, K. and Donley, R. (2019). Using Ink and interactive fiction to teach interactive design. *12th International Conference on Interactive Digital Storytelling, ICIDS 2019, Salt Lake City, Utah November 19 - 23, proceedings,* pp. 68 - 72.

Hulan, Haley (2017). Bury your gays: History, usage, context. *McNair Scholars Journal, (21)*1. Retrieved from https://scholarworks.gvsu.edu/cgi/viewcontent.cgi?article=1579&context=mcnair

Hunicke, R., LeBlanc, M. and Zubek, R. (2013). MDA: A formal approach to game design and game research. *Northwestern.edu.* Retrieved from https://users.cs.northwestern.edu/~hunicke/MDA.pdf

Infocom (1980). Zork: The great underground empire – Part I (Zork 1). Infocom.

Inkle (2014). 80 days. Inkle.

Inkle (2018). The intercept. Inkle.

Inkle (2019). Heaven's vault. Inkle.

Jackson, S. (1995). Patchwork girl. Retrieved from http://www.eastgate.com/catalog/PatchworkGirl.html

Keeling, K. (2014) Queer OS. *Cinema Journal* 53, 2, pp. 152 – 157.

Lauteria, E. (2012). Ga(y)mer theory: Queer modding as an act of resistance. *Reconstruction: Studies in Contemporary Culture 12*(2).

Lauteria, E. (2018). Envisioning queer game studies: Ludology and the study of queer game content. *Queerness in play,* eds. T. Harper, M. Blythe Adams and N. Taylor. Palgrave-McMillian. pp. 35 – 54.

Lester, G. B. (2018). What IF? Building interactive fiction for teaching and learning religious studies. *Teaching Theology and Religion,* 21, 4, pp. 260 – 273.

Lipkin, N. (2013). Examining indie's independence: The meaning of "inlidie" games, the politics of production, and mainstream cooptation. *Loading…: The Journal of the Canadian Game Studies Association 3*(11): pp. 8 – 24.

Lo, C. (2016). Flirt, flirt, romance: Fallout 4's problem ghtnwith queer relationships. *The Daily Gazette.* Retrieved from http://daily.swarthmore.edu/2016/01/22/flirt-flirt-romance-fallout-4s-problems-with-queer-relationships/

Lundberg, K. and Lyons, C. (2018). Using twine to deliver a grammar-linked creative writing assignment in a hybrid ESL course. *HETS Online Journal* 9, 1.

Martey, R.M. et. al. (2017). Testing the power of game lessons: The effects of art style and narrative complexity on reducing cognitive bias. *International Journal of Communication 11,* pp. 1635 – 1660. Retrieved from http://ijoc.org/index.php/ijoc/article/view/5032/1997

Maxwell, H. C. (2014). *The Game Production Handbook,* 3rd Edition. Burlington, MA: Jones and Bartlett Learning.

Mojang Studios (2011). Minecraft. Mojang Studios.

Morshirnia, A. V. and Walker, A.C. (2007). Reciprocal innovation in modding communities as a means of increasing cultural diversity and historical accuracy in video games. *DiGRa 07' – Proceedings of the 2007 International Conference: Situated Play*.

Nakamura, L. (1995). Race in/for cyberspace: Identity tourism and racial passing on the internet. *Work and Days* 13, pp. 181 – 193.

Obsidian Entertainment (2010). Fallout: New Vegas. Bethesda Softworks.

Østby, K. J. (2017). From embracing eternity to riding the bull: Representations of homosexuality and gender in the video game series Mass Effect and Dragon Age. *University of Oslo DUO Research Archive.* Retrieved from http://urn.nb.no/URN:NBN:no-58062

Pagnotti, J. and Russell, W.B. (2012). Using Civilization IV to engage students in world history content. *The Social Studies* 103, 1, pp. 39-48. DOI: 10.1080/00377996.2011.558940

Pereira, J. (2013). Video game meets literature: Language learning with interactive fiction. *E-Teals: An E-Journal of Teacher Education and Applied Language Studies*, 4, pp. 19-45. DOI: http://ojs.letras.up.pt/index.php/et/article/view/4043

Presada, D. (2015). Teaching literature by means of games in higher education. *Journal Plus Education, 12*(2), 113 – 120.

Queerly Represent Me. (2019). "So, you've decided you want to make a diverse game… now what?" *Queerly Represent Me.* Retrieved from https://queerlyrepresent.me/resources/articles/making-diverse-games

Quinn, Z. (2013). Depression quest: An interactive (non)fiction about living with depression. *DepressionQuest.com.* Retrieved from http://www.depressionquest.com

Reed, A. (2009). Blue lacuna. Retrieved from http://aaronareed.net/blue-lacuna/

Reed, A. (2013). 18 cadence. Retrieved from https://aaronareed.net/18-cadence/

Robertson, J. and Good, J. (2005). Children's narrative development through computer game authoring. *TechTrends: Linking Research and Practice to Improve Learning 49*(5), pp. 43 – 59.

Rosewater, M. (2016). Magic the Gathering: Twenty years, twenty lessons learned. *Game developers conference 2016.* Retrieved from https://www.youtube.com/watch?v=QHHg99hwQGY

Ruberg, B. (2019). *Video games have always been queer.* New York, NY: New York

University Press.

Ruberg, B. and Shaw, A. (2017). *Queer game studies.* Minneapolis, MN: University of Minnesota Press.

Ryan, M. (2009). From narrative games to playable stories: Towards a poetics of interactive narrative. *Storyworlds: A Journal of Narrative Studies 1*(1), pp. 433 – 60. Retrieved from http://www.marilaur.info/poetics.pdf

Ryu, D. and Jeong, J. (2019). Two faces of today's learners: Multiple identity formation. *Journal of Educational Computing Research,* 57, 6, pp. 1351-1375.

Salen, K. (2007). *The ecology of games: Connecting youth, games, and learning.* Boston, MA: MIT Press.

Salen, K. (2007). Gaming literacies: A game design study in action. *Journal of Educational Multimedia and Hypermedia 16*(3), pp. 301 – 322. Retrieved from https://llk.media.mit.edu/courses/readings/salen-gaming-literacies.pdf

Sensia (2020). Same-sex couples and LGBT families [computer game modification for Fallout 4]. *Nexusmods.* Retrieved from https://www.nexusmods.com/fallout4/mods/25254/

Shaffer, D. (2005.) Epistemic games. *Innovate: Journal of Online Education 1*(6). Retrieved from http://nsuworks.nova.edu/cgi/viewcontent.cgi?article=1165&context=innovate

Shaw, A. (2014). *Gaming at the edge: Sexuality and gender at the margins of gamer culture.* Minneapolis, MN: Minnesota University Press.

Shaw, A. (2017). Diversity without defense – Reframing arguments for diversity in games."

Kinephanos: Special Issue on Gender in Games, pp. 54 – 76. Retrieved from http://www.kinephanos.ca/Revue_files/2017_Shaw.pdf

Shaw, A., Rudolph, S. and Schnorrenberg, J. (2019). *Rainbow arcade: Over 30 years of queer video game history.* Schwules Museum.

Short, E. (2006). Bronze. Retrieved from https://ifdb.tads.org/viewgame?id=9p8kh3im2j9h2881

Short, E. (2017). The unique selling points of parser IF. Retrieved from https://emshort.blog/2017/05/04/mailbag-the-endangered-art-of-parser-if/#more-32431

Sinclair, C. (2017). Technobabylon's dystopian queer utopian vision. *Qleer Qritics.* Retrieved from https://qleerqritics.com/index.php/2017/09/13/technobabylons-dystopian-queer-utopian-vision/

Skains, L. (2019). Teaching digital fiction: Integrating experimental writing and current technologies. *Palgrave Communications* 5, 1, pp. 1 - 10. DOI: 10.1057/s41599-019-0223-z

Smaldone, R.A. Thompson, C.M., Evans, M. and Voit, W. (2017). Teaching science through video games. *Nature Chemistry,* 9, pp. 97 – 102. DOI: 10.1038/nchem.2694

Squinky (2015). 36 questions. Squnky.me

Square (1991). Final fantasy IV. Square.

Squire, K. (2011). *Video games and learning: Teaching and participatory culture in the digital* *age*. New York, NY: Teacher’s College Press

Sundén, J. (2009). Play as transgression: An ethnographic approach to queer game cultures. *DiGRA ’09 – Proceedings of the 2009 International Conference: Breaking New Ground: Innovation in Games, Play, Practice, and Theory*. DiGRA. London, United Kingdom, pp. 1 – 7. Retrieved from http://www.digra.org/wp-content/uploads/digital-library/09287.40551.pdf

Technocrat Games (2015). Technobablylon. Wadjet Eye Games.

Telltale Games (2012). The walking dead: Season One. Telltale Games.

Telltale Games (2016). The walking dead: Michonne. Telltale Games.

Valve (2006). Half-Life 2: Episode two. Valve.

Wainwright, M.A. (2014). Teaching historical theory through video games. *The History Teacher* 47, 4, pp. 579-612. DOI: https://www.jstor.org/stable/43264355

Warner, K. J. (2017). In the time of plastic representation. *Film Quarterly 71*(2), pp. 32-37.

Welch, T. (2018). The affectively necessary labor of queer mods. *Game Studies 18* (3). Retrieved from http://gamestudies.org/1803/articles/welch

Wendel, V. et al. (2013). Designing a collaborative serious game for team building using Minecraft. In *Proceedings of the 7th European Conference on Games Based Learning (ECGBL 2013)*, 2. DOI: http://tubiblio.ulb.tu-darmstadt.de/63618/

Yang, R. (2015). Radiator 1 [computer game modification for Half Life 2: Episode Two]. *Itch.IO.* Retrieved from https://radiatoryang.itch.io/radiator1

Yucel, I., Zupko, J. and El-Nasr, M.S. (2006). IT education, girls and game modding. *Interactive Technology and Smart Education* 3, 2, pp. 143-156.