UNIVERSITY OF CALIFORNIA AT SANTA CRUZ

INVESTIGATING PROCEDURAL EXPRESSION AND INTERPRETATION IN VIDEOGAMES

A dissertation submitted in partial satisfaction
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

COMPUTER SCIENCE

by

Mike Treanor

June 2013

The Dissertation of Mike Treanor is
approved:

Professor Michael Mateas, Chair

Professor Noah Wardrip-Fruin

Professor Ian Bogost

Rod Humble
(CEO Linden Lab)

Tyrus Miller
Vice Provost and Dean of Graduate Studies

Table of Contents

List of Figures

Abstract

Mike Treanor

Investigating Procedural Expression and Interpretation in Videogames

What a videogame is about is not easy to say. While existing theoretical approaches can help us understand games as narratives or generally cultural artifacts, it can be argued that the unique aspect of games is that they are comprised of processes and new theoretical tools and design approaches are needed if we want to utilize this feature of the medium. This dissertation develops and applies the claims of those who espouse the virtues of a procedurally oriented approach toward design and interpretation.

The contributions of this dissertation take the form of theoretical investigations and media artifacts that explore how videogames are both expressive and representational. The first investigation claims that what a game represents is grounded in patterns of abstract game rules and a player's beliefs about a game's visuals. This theoretical framework informed the creation of *Game-O-Matic*, a game generation tool that is able to generate simple games about subjects. The second investigation claims that more complex systems of rules are best understood through experimentation leading to an understanding of the game's representational principles. This approach informs a discussion of the creation of the social simulation game *Prom Week*, which uses simulation to represent a theory of social interaction in

which the character's social actions are determined by a myriad of varied and complex reasons. The third investigation concludes that accounting for an individual player's subjectivity is essential when discussing what a game is about. A "proceduralist" position is then defined as someone who prioritizes a comprehensive account of a game's processes which can aid in the discovery of new representational affordances for games.

The insights and conclusions of this dissertation resulted from a methodology that embraces both humanistic investigation and technical research (critical technical practice). By developing theories and having these theories drive technical practice that result in art works, insights are presented about the relationship between instantial assets and rules, complex simulations and representation, and why players understand games differently.

Acknowledgements

I have been very fortunate to spend these five years with so many generous and intelligent people. First of all, I want to thank my advisor Michael Mateas. When I started working with him, I was admittedly naive, and his mentorship made me an undoubtedly better theorist, engineer and artist. I would also like to thank the rest my committee: Noah Wardrip-Fruin, for his ideas and helping me to present my own ideas effectively, Ian Bogost, for his friendship and for instilling in me a sense of criticality, and Rod Humble for being the type of artist I have always wanted to be.

Also, thanks to my collaborators on *Prom Week* and *Game-O-Matic*. We have been through a lot and they have been great colleagues and friends.

My family and friends are what kept me from falling off the deep end through the weird process of getting a PhD. I want to particularly thank my mom Shirley Treanor for her encouragement and inspiration, and Chris Thomas for his friendship and discussions through the years that have been a major influence to my thinking.

I couldn't have created this work without my amazing wife Brianna. She unwaveringly supported me, listened to me, gave me ideas and advice, and most of all, filled my days with companionship and joy.

Chapter 1. Introduction

Over thirty years ago, game designer and theorist Chris Crawford wrote that "games constitute a new and as yet poorly developed art form that holds great promise for both designers and players" (Crawford 1982). Since then, videogames have become a significant commercial force and more people are playing videogames than ever. Even more, both creators and players are finding new uses for videogames: artists are trying to express themselves, activists are trying to spread awareness, businesses are trying to encourage customers to use their products, educators are trying to teach, governments are trying to train, etc. Considering this, it might be concluded that the *new* and unique art form of games is now better understood and the *great promise* for designers and players has been realized.

However, this optimistic conclusion doesn't fully stand up to scrutiny. For example, *BioShock* is a first person shooter set in a failed utopia that is often praised for its *deep* story which critiques the individualist philosophy of Ayn Rand. The problem with this praise is that it only addresses part of the player's experience: the story as presented through animated cut scenes. The majority of the player's experience involves frenzied navigation through corridors while shooting guns at the city's mutant population. Therefore, the gameplay does not meaningfully contribute to the critique that the game is praised for. Game designer Clint Hocking describes this as *ludonarrative dissonance* (Hocking 2007) – the gameplay and the story are not in alignment. Furthermore, the commentary that is actually provided about Rand's philosophy is achieved only through the traditional media of cinema. Because of this, it can be concluded that *BioShock*, a critically acclaimed and commercially successful game, is not good evidence that videogames are delivering the supposed *great promise* that was enabled by this *new* art form as touted by Crawford.

Of course, this problem is not unique to *BioShock*. Many other games that purport to be about certain subjects fail to achieve what game studies scholar Ian Bogost calls "tight coupling" (Bogost 2007) – where gameplay and story are in alignment. As another example, playing the serious game *Global Conflicts* involves taking on the role of an investigative journalist who travels the world collecting information through interviews. These interviews are presented to the player as text. Again, the text itself is successful in communicating concepts about the regions that the player is exploring, but the gameplay, which involves clicking on characters to get them to reveal static chunks of text, doesn't contribute the game's subject. In other words, *Global Conflicts* manages to be educational and informative, but it does so by treating the gameplay as merely the vehicle to deliver information in the form of borrowed techniques from established media like newsprint and televised journalism.

A reason why ludonarrative dissonance is a reoccurring problem is that it isn't clear what it even means for gameplay to be meaningful. For example, if the power pellets of *Pac Man* were replaced with hearts, would that be enough to say that the game is about love? Or if two guild members meet and then get married in *World of Warcraft*, does that make the game about love? Videogames are difficult to understand because they confront players with all the varied forms of visual, aural and textual media, in addition to an abundance of unique characteristics such as rules and goals. The degree to which a player finds meaning in the visuals, social relations between players, strategies for playing, etc. vary from person to person and it isn't clear what the important elements to consider are. In addition, the dynamic nature of videogames makes it unlikely that two players will have the same experience. Currently, we lack the theoretical perspectives to understand how these dynamic and interactive aspects of videogames can be expressive or interpretable to designers and players.

The goal of this dissertation is to better understand how these unique aspects of videogames can deliver new expressive and interpretive opportunities for game developers and players. Rather than try to address the ambiguous concept of how a game can be meaningful in general, this dissertation explores the specific question of how it is that players are able to recognize a game's processes as *intentional*, or *about* a subject. Intentionality is a word used in philosophy to refer to something's ability to refer to something outside of itself. For example, while some might say that *Pac-Man* is meaningful because it is *historically meaningful* for appealing to female players in a time when male players dominated the arcades (Goldberg 2006), this dissertation specifically investigates questions about what *Pac-Man* might be said to be represent, or stand for. For instance, game studies scholar Steven Poole claims that *Pac-Man* is a representation of rampant consumerism because the game involves the relentless consumption of power pellets (Poole 2000). The words *about* and *representational* will be used to refer to this concept of intentionality.

Procedural Rhetoric

Throughout this dissertation, the concept that games are dynamic and interactive will be expressed as the game being comprised of *processes*. The goal is to better understand how a game's processes (rules, goals, player choices) can be treated as an essential aspect of a game, rather than an incidental aspect as with *BioShock* and *Global Conflicts*. Below is a brief discussion of other designers and theorists that have explored the procedural nature of games.

An early example of this can be seen in Chris Crawford's argument that since information processing is unique to computer-based media, and that interactivity is of primary significance to games, a videogame should strive to maximize the ratio between a game's processes (code, algorithms) and its instantial assets (hand crafted video, text, etc.) (Crawford 1982). In more recent years, game designers like Rod Humble and Brenda Romero have argued that "a game needs nothing else apart from its rules to succeed as a work of art" (Humble 2006).

Frasca describes videogames as dynamic simulations that produce particular static narrations through interaction (Frasca 2001). In other words, understanding a videogame involves having a conception of how and why a simulation generates particular narratives. Unlike films and works of literature, which can be approached as being single static narratives, the notion that games are generators of static representations emphasizes the importance of the role of a game's rules and goals, and the resulting processes, in what it might be argued as being about.

Bogost states that "Video games are models of real and imagined systems... when we play, we explore the possibility space of a set of rules—we learn to understand and evaluate a game's meaning" (Bogost 2008). Inside a game like *SimCity*, the set of rules largely determines what the game is able to represent about how public policy decisions determine the health and population of a city. These rules may have been developed to create fun and engaging gameplay, rather than to provide an accurate model of urban planning, but nonetheless it can be argued that the model can still be persuasive and shape the way players understand the world. Frasca highlights that simulations are hardly neutral playing grounds where any outcome is possible and that they necessarily privilege ways of playing by design (Frasca 2001).

Bogost further describes this as a game's *procedural rhetoric* – "the practice of effective persuasion and expression using processes." He continues, "Since assembling rules together to describe the function of systems produces procedural representation, assembling particular rules that suggest a particular function of a particular system characterizes procedural rhetoric" (Bogost 2008). In discussing the procedural rhetoric of his *Take Back Illinois*, a game created about a political race, Bogost argues, "In playing the game, the player is not 'brainwashed' or otherwise fooled into adopting the candidates' policy position. Rather he is afforded an understanding of that position for further inquiry, agreement, or disapproval" (Bogost 2008). Bogost describes the *simulation gap* as the conceptual space between the player's existing mental model about what a game is about and the player's interpretation of how the game itself operates. He argues that the act of performing this comparison, exploring the simulation gap, can be educational and is one of the greatest strengths of the medium of videogames (Bogost 2007).

Wardrip-Fruin describes how players build mental models of the systems that governs the processes of a game: "Successful play requires understanding how initial expectation differs from system operation, incrementally building a model of the system's internal processes based on experimentation" (Wardrip-Fruin 2009). Elsewhere, he and Mateas, describe that the computational models that underlie games largely determine what subjects designers are able to make games about. These *operational logics* shape both what the game designer is able to conceive of creating as well as how players are able to understand them (Mateas and Wardrip-Fruin 2009). This idea places processes as central to what a game can be about.

Even outside of the field of game studies, in artificial intelligence some describe that software can be about subjects by arguing that *computer programs are theories* of what they are about. Simon and Newell write "Programs can be regarded as theories, in a completely literal sense, of the corresponding human processes" (Simon and Newell 1962). Johnson-Laird elaborate on this idea in saying "There is a well established list of advantages that programs bring to the theorist: they concentrate the mind marvelously; they transform mysticism into information processing, forcing the theorist to make intuitions explicit and to translate vague terminology into concrete proposals..." (Johnson-Laird 1981). As computer programs themselves, videogames and their underlying operational logics can be understood to be theories of their subject matter.

Critical Technical Practice

As shown above, there is a tradition of theorists discussing the relationship between the procedural aspects of games and what they are about. This dissertation continues in this tradition, but takes a different approach. Rather than attempting to achieve a broad account of how processes are meaningful, the following pages will strive to describe how procedural representation operates at a very fine level of detail. For example, when Bogost discusses his *Take Back Illinois* and claims that players "afforded an understanding" of a candidate's position (Bogost 2007), he does not precisely explain the details of how this political position is represented. What is it about the processes that represent the political position? This dissertation strives to understand specifically how low level game mechanics and the resulting processes contribute to representation.

To achieve this kind of detailed understanding, the work presented in this dissertation is both theory and practice based. Like the early film theorists who discovered the affordances of the medium of film through experimentation, this dissertation explores how videogames can be about subjects through experiments.

The work presented in this dissertation can be understood as a form of what Agre calls critical technical practice (CTP) (Agre 1997). CTP strives to reveal the conceptual instability of the metaphors we use to describe processes and then to apply the lessons learned to future work. "...the whole point of a critical technical practice is to work from the inside, driving the customary premises and procedures of technical work toward their logical conclusions" (Agre 1997). When applied to videogames, the methodology of this approach first involves performing humanistic investigation to develop a representational theory that can serve as a lens for understanding and creating videogames. Next, the theory becomes the basis from which a system and artwork is created. Both the completed artwork and technical work that went into creating it are then critiqued to reveal areas of the theory that are undeveloped.

Agre's own work developing a new theory to drive the practice of creating artificial intelligence provides a good introduction to CTP. Agre began by looking at the state of artificial intelligence (AI), and recognized that many systems ran into the same persistent problems. In investigating the state of planning architectures, systems that given a goal are able to produce a sequence of actions that will arrive at that goal, he concluded that they all suffered from the failings of what he called a *mentalist* theoretical approach that drove the practitioner's work. A mentalist perspective is defined as believing that there is a strict separation between the mind and body. The implication of this is that there is a world that the brain reasons over with limited information that is presented in the form of sensed data. The reasoning happens in a vacuum separate from the world. One problem of this approach is that the mentalist is inclined to create systems that are less reactive, as any newly sensed data will likely invalidate a plan, and thus the planning system will need to begin again.

In identifying this mentalist approach, Agre made explicit the theoretical assumptions that were implicit in the planning architectures. From there, he developed his own theoretical approach through the *reversal* of the assumptions of the mentalist approach. Where the mentalist separates the mind from the world, Agre defines the *interactionist* to assume that reasoning happens from an embodied perspective that is entrenched in the world. From there, he set out to create an AI architecture that enacted the interactionist approach, as the planner could be said to have enacted the mentalist approach. The system he created, called *Life*, constantly decided what to do by running rule based arguments for actions and maintaining a model of the dependencies between actions. When the world changed, inappropriate actions would no longer be possible, without having to re-plan. Next, he made use of *Life* to create two systems: *RA* and *Pengi*. In attempting to use the system that was created from the interactionist approach, he discovered strengths and weaknesses of the approach.

While Agre's impressive work on this subject ended after discussing *RA* and *Pengi*, he could have continued on to create a new theoretical approach that resulted from analysis of what he learned about the interactionist perspective, and then continued further to create architectures that enacted this new theory, and so on. CTP does not arrive at definitive conclusions, but instead is a theoretical exercise intended to reveal the problematic assumptions that underlie practice as well as suggest new directions. This dissertation presents three separate applications of CTP that explore the theoretical foundations of how videogames can be about subjects. Each will be introduced below.

Research Contributions

In the traditions of the theorists exploring procedural rhetoric and those who perform critical technical practice, this dissertation sets out to understand how the procedural aspects of games contribute to what a game is about. With this understanding, designers would be able to better utilize the expressive affordances of games, and players and scholars will be able to better understand how games can be about subjects. Specifically, there are three primary research questions that will be explored in this dissertation:

1. How do instantial assets interact with game rules to be about subjects?
2. How can complex systems of rules be about subjects?
3. Why do players understand games in different ways?

Each theory presented in this dissertation strives to avoid the pitfall of defining videogames by reductive essential features, while also being precise enough to be useful by designers, players and critics. Central to my argument is the idea that a videogame is a set of processes whose future is not predetermined but is still constrained by a structure. In videogames, it is code that strictly governs what actions and responses are possible. Of course, any particular player's experience with a game is also reliant on all of the social, historical and cultural systems that already influence the player's life. Thus, to make a claim about a videogame is to interface with all of these systems. This is no easy task, and in fact, this dissertation will not arrive at any definitive conclusions. Even so, the investigations presented throughout this dissertation do allow us to understand how processes can be about subjects *more* than we were able in the past.Each of the three sections of this dissertation will explore a humanistic theory, then describe artificial intelligence architectures and art work that explores the theory and conclude with a discussion of the lessons learned about the theory in the practice of creating the architecture and art work.

Mechanics and Instantial Assets

Chapter 2 explores how a game's visual static assets relate to the game's mechanics to create procedural metaphor. Beginning by looking at a subset of games that attempt to employ procedural rhetoric, those that use simple game mechanics metaphorically, a theory of *micro-rhetorics* is presented. Micro-rhetorics are patterns of game mechanics and the beliefs that a player must have about the instantial assets in order for a segment of gameplay to about a concept. This theory makes explicit the theories that are arguably implicit in the design of these games. Next, this theory is implemented in *Game-O-Matic*, a videogame generator that takes a network of nouns with verb relationships as input and generates videogames that represent it using simple arcade game mechanics. Next, through discussing both the games that *Game-O-Matic* generates and the process of creating *Game-O-Matic* itself, the strengths and weakness of the micro-rhetoric theory is presented.

Simulation Representation

Chapter 3 begins by carefully describing two different modes of engagement, or semiotic registers, that players employ when understanding games: the instantial register, where existing beliefs about the static assets take precedence, and the simulative register, where an understanding of the processes takes precedence. Next, the chapter explores how games with complicated mechanics and a high degree of dynamism can be interpreted as embodying sets of *principles*. Principles are defined to be the player's beliefs about how the game generates particular instances of gameplay, or narrations, as Frasca would call them. Next, the challenging process of trying to create *Prom Week* is presented *Prom Week* is a narrative simulation game where players control the social actions of high school students in the week before their prom, that was in part created to represent a complicated theory of social interaction. A discussion of the persistent problems encountered when trying to create a game that primarily represents through the complex system of rules reveals that the simulation theory alone is insufficient to describe why players will understand games in different ways.

Players and Processes

Chapter 4 presents a theory of videogame meaning that accounts for both a comprehensive account of the game's processes and how and why players might arrive at different interpretations of the same game. This theory of *proceduralist readings* describes the context from which an individual player will believe that a game is about a subject. This section also responds to critics of this proceduralist approach and provides an account of how players independently create meaning apart from authorial intent. In Chapter 5, a theoretical exercise is presented that can be used to discover new ways that games be meaningful. Referred to as *alien readings*, this exercise involves imagining someone who prioritizes a comprehensive and consistent account of a game's processes when interpreting it. An extended example of interpreting *BurgerTime* from this alien perspective demonstrates how the high level gameplay dynamics of expert players can be a potential expressive and interpretive affordance for designers and players. Next, a discussion of how alien readings could be implemented as a system and artwork is presented as future work. This system and art work would perform alien readings of gameplay as the player enacts it.

Game-O-Matic, *Prom Week* and the proposed alien readings system, exemplify different understandings of how videogames are about subjects. By implementing the theories in playable systems, each theory is taken to its logical conclusion, revealing flaws and gaps in the theory. This self destruction, or auto-critique, is part of the motivation for creating the playable system in the first place as it points to future research directions. The goal of each chapter is not to achieve one all encompassing theory of videogame representation, but to thoroughly understand the foundation, strengths and weaknesses of different approaches.

The contribution and evaluation of my work is both the playable systems as artistic artifacts, and also the philosophical progress made toward understanding how videogames are meaningful artifacts and activities in the first place. Overall, the aim of this work is to broaden audience's awareness of what there is to understand about a videogame as well as to demonstrate that creators have more authorial materials than they may have realized.

Chapter 2. Mechanics and Instantial Assets

Videogames are clearly representational. From a player's understanding that his avatar is standing on the *ground* in *Super Mario Brothers*, to the emotional investment in *Final Fantasy VII's* love story, players make judgments and interpretations about games. This section discusses videogames whose creators set out specifically to represent a message and how they were able to achieve this.

As a painter intentionally arranges pigment on canvas to have the desired result, videogame developers arrange the authorial materials of videogames to represent ideas. The following investigation attempts to better understand what materials game developers manipulate in order to represent ideas. In the end, *Game-O-Matic* - a videogame generator and tool that takes a message as input and generates simple arcade games that attempt to represent it - will implement and automate the presented theory.

The goal is to understand how videogames represent ideas differently than other forms of media. Thus, the starting point is the observation that a videogame requires the interactive participation of the player in order to have material to interpret at all. Aarseth describes this by identifying that in non-digital media like literature and film, aporia (an interpreter's state of puzzlement) is resolved through introspection or reflection (epiphany), while in computational media, resolution is impossible without taking action (Aarseth 1997). Players don't just observe and contemplate, they shape the object of contemplation itself through gameplay. Thus we begin by studying a game's interactivity and the rules that determine its structure.

Rod Humble's *The Marriage* has been much discussed since its release in 2007 as a prime example of a work that has as its "primary medium of expression something unique to games" – the rules (Humble 2007). Humble wanted to express his ideas and feelings about marriage without relying on story, imagery, sound, etc. Instead, he wanted to convey a message through only the game's rules. For Humble, "a game needs nothing else apart from its rules to succeed as a work of art" (Humble 2006). This approach helped to inspire a *proceduralist* movement of game designers who create games where the "expression is found primarily in the player's experience as it results from interaction with the game's mechanics and dynamics…" (Bogost 2009a).

In *The Marriage*, players exert a minimal amount of control over the movement of two blue and pink squares by positioning the cursor over them in order to command them to move toward one another. Based on the passage of time and collisions with various circles and each other, the square's scale and transparency are modified. Different rules govern what happens to the blue and pink square. The following game rules, described by the game's creator, give an idea of the gameplay:

> *When the edge of the blue square collides… with the edge of the pink square (but not when they overlap): the blue square shrinks slightly and becomes more transparent. The pink square grows slightly and becomes less transparent.*

> *When the blue square touches any coloured circle but black then the blue square becomes less transparent and grows in size to a significant degree. When the pink square touches any coloured circle but black then the pink square grows in size slightly* (Humble 2007).

The first rule indicates that the blue square must not collide with the pink square too much, or it will disappear. Furthermore, the blue square can gain back some of its size if it collides with the green circles. Just from these few rules, gameplay can be imagined where the player is balancing having the blue and pink squares collide and getting the blue square to collide with green circles. From this, a bare bones interpretation is possible: in a relationship, a female needs more attention than a male does who is in need of some contact with entities outside the relationship. Clearly, this interpretation involves more than is encoded in the rules, as it relies heavily on the game's visual theme. Even when interpreting this highly abstract game, an interpreter cannot rely only on the rules of the game but must also draw upon cultural gender connotations of the colors of the blue and pink squares.

The Marriage is one of the most ambitious attempts of a game whose meaning is represented solely through its rules and gameplay, but it also indicates that rules are only meaningful when applied to some theme or domain.

Other videogames have been created that attempt to be representational through gameplay without self imposing the strict constraint that the visuals should strive to be as abstract as possible. In recent years, many developers have been rapidly creating and releasing games in response to current events. Game scholar

Gonzalo Frasca described these games with the phrase “simulation meets political cartoons” and labeled them as “newsgames.” Illustrating this comparison, Frasca created a game that plays very much like Activision’s *Kaboom!* (Activision 1981) yet was intended to carry a biting critique of the foreign policy of the United States dropping humanitarian aid and bombs on the same regions following the attacks of September 11th.

In *Kabul Kaboom*, the player’s goal is to collect food falling from the sky while avoiding falling bombs. The game always ends in failure. After playing for a short while it becomes clear that there is no way of winning. This guaranteed failure, initially unknown to the player, allows the player to discover and experience firsthand the author’s opinion about the event it simulates. This authorial device has previously been referred to as the rhetoric of failure (Lee 2003). With *Kabul Kaboom*, Frasca demonstrates how authoring game rules in this way can serve as an additional rhetorical outlet, and in this case is how this game provides even deeper commentary than the related political cartoons (Figure 1). This game serves as an example of how newsgames have the potential to communicate and persuade players in ways that political cartoons are unable to do.

Bogost describes the term procedural rhetoric as “persuading through processes” (Bogost 2007) or “the way that a videogame embodies ideology in its computational structure” (Bogost 2006). This provides a lens for analyzing how games can persuade through rules and player goals. In *Kabul Kaboom* (Figure 1) the game rules guarantee that the player will lose if they are hit with a bomb. Frasca’s

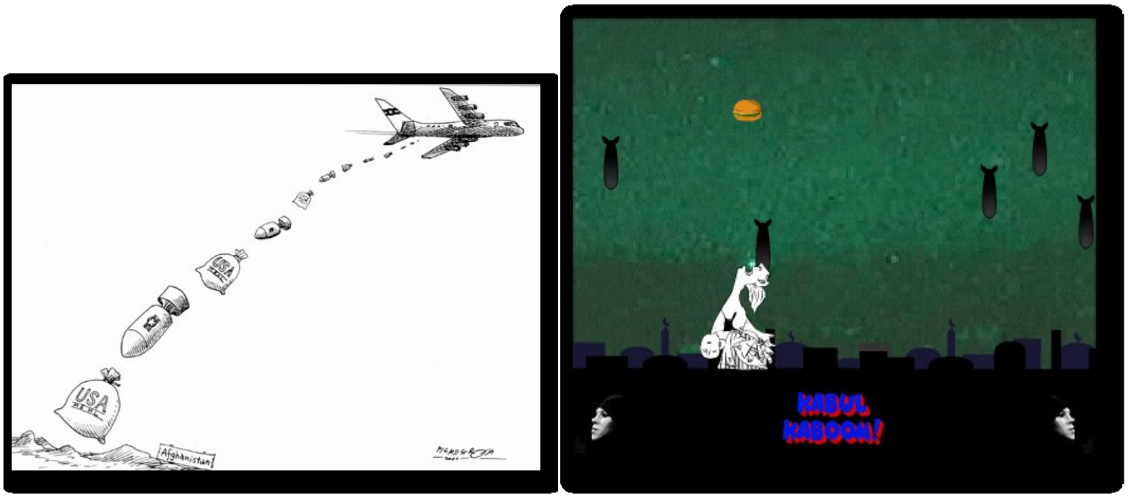

Figure 1 - An editorial cartoon (left) that expresses its critique visually and a videogame (right) that expresses a critique through its processes.

intended message is understood as the player enacts the rule system, communicating how a foreign policy that involves dropping food and bombs on the same country will ultimately end in failure by hurting the same people the food is supposed to help.

Elsewhere in his writings, Frasca explains how games can be best understood as simulations rather than narratives (Frasca 2001). Simulations can provide thousands of narrations depending upon the interaction of the player (i.e. a narrative is a particular run through a simulation). However, a simulation is not a neutral playing ground where any outcome is possible. It is the job of the game designer to imagine the possible interpretations of gameplay as he creates the rules that govern it. *Kabul Kaboom* clearly demonstrates this idea in that the rules of the game always produce the same outcome and influence the player to enact the supporting arguments for the game's editorial assertion.

Crafting the game in this manner effectively proves Frasca's editorial statement within the micro world of the game by having the player enact the undesirable outcome with some degree of agency. A high degree of agency can be an important rhetorical device because of how it naturalizes an editorial opinion. In this case, agency is achieved by giving the player very little interactivity as well as very little motivation or opportunity to imagine performing any action other than what is available to the player (Mateas 2006). *Kabul Kaboom* is persuasive, and a great early example of a game that purposefully limits interactivity, while maintaining agency, in order to limit the outcomes of its simulation to communicate a political opinion. *Kabul Kaboom* illustrates how newsgames can function as strong editorial tools and it leads the way for future development of newsgames and videogames as expressive media.

Theorizing how Rules and Theme are Representational

The above discussions of *Kabul Kaboom* and *The Marriage* show how videogames are able to represent an idea through the design of its rules and goals. However, both heavily rely on visuals to give context to a player's interpretation.

Semiotic analysis can help us understand the relationships between a videogame's visuals and rules. Semiotics is the study of how objects are able to represent, or signify, other concepts. A videogame can be regarded as being a system of signs: signifying elements which bring to mind signified concepts. As a simple example, the player's avatar in *Kabul Kaboom* is a small segment of Picasso's antiwar painting Guernica and is understood by most as representing the civilians of the country that food and bombs are being dropped on. However, this may not be the case for all people as a sign may signify any number of signifieds. The topic of why an interpreter might associate a particular signifier with a particular signified is a much discussed and theorized subject and will be returned to throughout this dissertation.

Semioticians call systems of signs texts, and often use these two forms of analysis to understand them: syntagmatic and paradigmatic. A syntagmatic analysis identifies sequences of signs, syntagms, in texts and attempts to understand what they represent as a whole. 'What does A *and* B *and* C signify?' A paradigmatic analysis attempts to understand a series of signs by imagining what *was not* selected by the author to be part of the sequence. A paradigmatic analysis involves *substituting* one sign for another in a syntagm. 'What would A and B and C signify if it were instead A and B *and D*?' The goal of a paradigmatic analysis is to understand "the influence of the substitution on the meaning and also to identify syntagmatic units" (Chandler 2007).

Toward understanding exactly how theme relates to rules in videogame representation, below is a paradigmatic analysis of the visuals of a simple arcade game.

Analyzing *Kaboom*!

The following analyses explain how Activision's *Kaboom!* (Figure 2) can be about different subjects when different visuals are substituted and the rules are left the same. The analysis grew out of the initial interpretation of the game below. It should

Figure 2 - A screenshot from Activision's *Kaboom!*

be noted that initial interpretations are easy to generate, but the goal of this chapter is not just to arrive at interpretations, but to understand why we believe them.

On the top of *Kaboom!*'s game screen a "Mad Bomber" moves in unpredictable patterns from left to right dropping bombs that fall straight down the screen. At the bottom of the screen is the logo for Activision, the company that released the game. The player controls a set of three "buckets" that can move from left to right. A score counter is incremented for each bomb the player is able to "catch" and is given three chances to miss the bombs before the game ends.

The animation of the Mad Bomber as he drops the bombs make it clear that the bombs are being dropped intentionally (as opposed to accidentally dropping them). Also, it can be inferred by the fact that as the game progresses it becomes increasingly difficult for the player to catch bombs that the Mad Bomber is not intentionally dropping bombs into the buckets. Once three bombs get past the player, the damage done by the Mad Bomber—to a largely-unseen fictional world—is too great for the scenario to continue. Because *Kaboom!* is unwinnable (like many early arcade-style games, the difficulty just keeps progressing until inevitable defeat), this quest to protect the world from damage is ultimately hopeless.

Given the appearance of the Activision logo at the bottom of the screen, one might make the interpretation that *Kaboom!* is about a helpless quest to *protect* Activision from an insane criminal. This interpretation might be discarded, as it treats a piece of visual representation that is clearly paratext (in this case, a brand identifier), with visual representations that are part of the represented game world. However, for the purposes of our analysis below, the presence of the Activision logo points to the possibility of placing a visual representation at the bottom of the screen that is intended to be in the game world. In this way, the unseen fictional world of the original *Kaboom!* can be replaced with an explicitly represented world that the player is trying to protect from the bombs. Considering this additional entity allows for many more potential messages for *Kaboom!*'s mechanics to represent with different thematic mappings (as demonstrated below).

As an example of *Kaboom!* representing a different protection message, consider modifying the visuals by substituting the sprite of the Mad Bomber with a sprite of a Republican elephant, the buckets with a Democratic donkey, the bombs with images of guns and the Activision logo with an image that represent citizens. In concert with *Kaboom!*'s rules, this theme now makes an editorial cartoon-like statement about the gun control debate. In this case, Democrats would be protecting

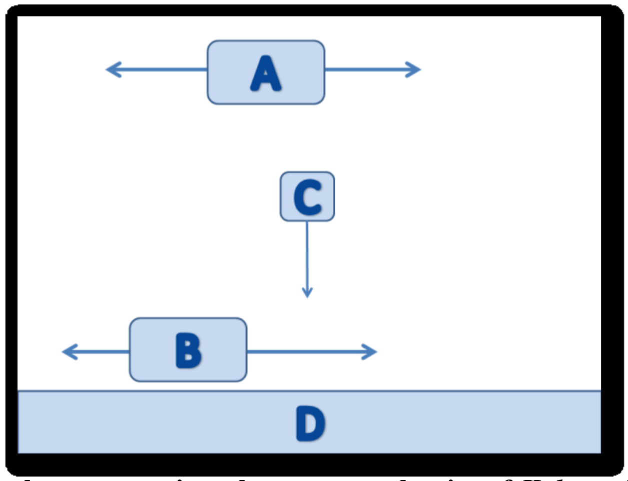

Figure 3 - An image that summarizes the game mechanics of *Kaboom!* without taking into account the game's visuals.

the citizens from the guns that the Republicans are making plentiful (top left of Figure 4). The investigation that follows seeks to systematically explore the constraints and relationships between visual elements of *Kaboom!* to characterize a space of meanings conveyable by the mechanics of *Kaboom!*. This paradigmatic analysis of *Kaboom!* is intended to help isolate the role of game rules in procedural representation.

The first step of this analysis involves abstracting the pre-existing beliefs about the visually represented entities (the Mad Bomber, bombs, etc.) from game tokens controlled by game mechanics. This separation of the theme from the game mechanics allows us to consider the relationships between them and how they are used to form an interpretation. All references to the game mechanic's tokens will be referred to by arbitrary labels as follows (Figure 3): A = Mad Bomber, B = the buckets, C = the bombs and D = the area below the buckets (including the Activision logo).

With this, the mechanics of *Kaboom!* can be described as follows:

- C is generated at A's location and falls at a constant rate
- When C collides with D, the number of remaining attempts to play the game is decremented
- When C collides with B, C is removed and the score is incremented
- A moves left and right
- B is controlled by the player
- B can move left and right

Given this separation between theme and rules, the specific details of why *Kaboom!* can be said to be about protection can be explored. The concept of protection can be said to imply that the entity doing the protection is preventing harm to the object of protection. This can be seen in our initial interpretation. A bomb is harmful to an unseen world and the buckets are preventing the bombs from reaching the unseen world. Generalizing this, the idea that *Kaboom!* is about protection relies on whether it is plausible that an interpreter might believe that the game entity C would be harmful to D in some way. From here, we can represent this concept as simple thematic considerations that characterize the relationships between the represented entities that the rules of the game put into contact with one another. In the case of *Kaboom!* there are only two collisions to evaluate: D and C, and B and C. Because there are cases where two thematic concepts might not carry reciprocal evaluations of the other, it makes sense to consider how each thematic entity would evaluate a collision with the other (i.e. D may evaluate a collision with C as a bad encounter, where C might consider it a good one). This adds two more evaluative considerations: C and D, and C and B. In addition to these evaluative considerations about collisions, it makes sense to consider whether the thematic representations of the game elements that perform actions would be interpreted as taking them with intention or not. In *Kaboom!* the only NPC entity that takes action is A dropping C, so we add a fifth consideration of whether the interpreter believes that A would drop C on purpose or not. Below are all five interpretive considerations and a set of possible responses:

- From the perspective of B, evaluate a collision with C. Responses: [Good, Bad or N/A]
- From the perspective of C, evaluate a collision with B. Responses: [Good, Bad or N/A]
- From the perspective of C, evaluate a collision with D. Responses: [Good, Bad or N/A]
- From the perspective of D, evaluate a collision with C. Responses: [Good, Bad or N/A]
- Does A intentionally drop C? Responses: [Yes, No, or N/A]

We can now understand the original interpretation of *Kaboom!* being about B protecting D from A's attack of C in terms of our thematic interpretive considerations (presented in an abbreviated formal notation):

- CollisionEval(Buckets, Bombs) → N/A
- CollisionEval(Bombs, Buckets) → N/A
- CollisionEval(Bombs, Activision) → N/A
- CollisionEval(Activision, Bombs) → Bad
- Volition(Mad Bomber, Drops(Bombs)) → Yes

Here we see that an anthropomorphized Activision considers a collision with bombs "bad" and the Mad Bomber drops the bombs on purpose. For this interpretation, the label of "N/A" for CollisionEval(Buckets, Bombs) states that it doesn't make sense to consider what Buckets think about a collision with Bombs.

This pattern of responses to the considerations can now be used to apply the game mechanics of *Kaboom!* in order to represent a game about the player protecting

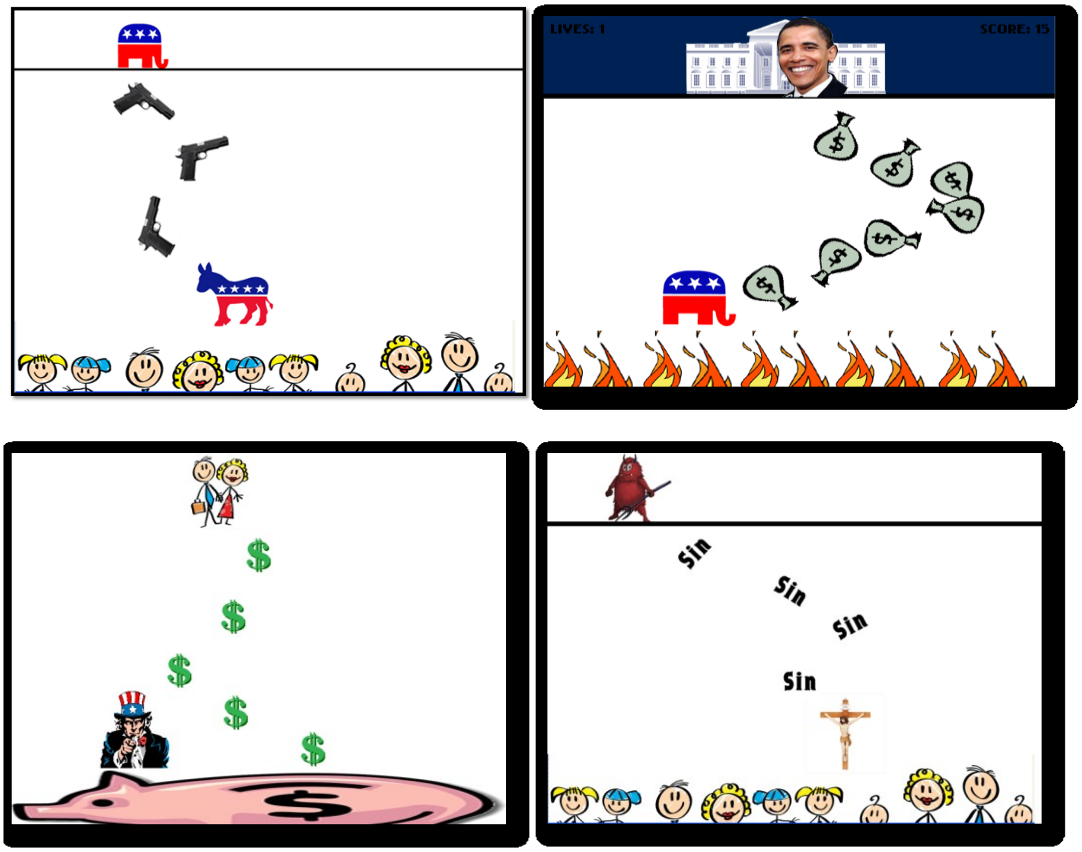

Figure 4 - Themings of *Kaboom!* where democrats protect the citizens from guns (top left), Republicans rescue money from being wasted by President Obama (top right), Uncle Sam steals the citizen's money (bottom left) and Jesus sacrifices himself to protect the people from the sins of the devil.

anything, so long as the interpreter's understanding of the relationships between A, B, C and D conform to the pattern.

Through enumerating the possible combinations of responses to the five interpretive considerations and interpreting them, we are now able to find classes of meanings for *Kaboom!* besides protection. An asterisks (*) will be used where the response of an interpretive consideration is not absolutely necessary for a game's inclusion in a class of meanings.

For example, the following responses to considerations would make the game about B rescuing C from being destroyed by D:

- CollisionEval(B, C) → *
- CollisionEval(C, B) → Good
- CollisionEval(C, D) → Bad
- CollisionEval(D, C) → *
- Volition(A, Drops(C)) → *

In this case, anytime when the interpreter would believe that the entity being dropped (C) would rather collide with the player controlled entity (B) than the entity at the bottom of the screen (D), an interpreter might think that the game is representing B rescuing C.

For *Kaboom!* to be about being self destructive just this one thematic consideration must hold:

- CollisionEval(B, C) → Bad
- CollisionEval(C, B) → *
- CollisionEval(C, D) → *
- CollisionEval(D, C) → *
- Volition(A, Drops(C)) → *

These classes of meaning can be combined as long as the responses to the considerations do not conflict. For example, *Kaboom!* can be a game about B being a "self-sacrificing hero" by combining the protect and self-destructive patterns (Figure 4).

A conclusion of this analysis of *Kaboom!* is that the meaning of a game is only partly related to its mechanics and that a group of mechanics can be said to have *rhetorical affordances* that are actualized by the player's pre-existing beliefs about the thematic assets of a game. Rhetorical affordances are defined to be the opportunities for representation made available by the rules that govern the relationship between objects and processes in a system. The representation that is selected from a set of possible representations afforded by a set of game rules is a product of its relationship with other dynamics in the system and the interpreter's beliefs about the instantial assets that specify its domain. Thus *Kaboom!'s* game mechanics have rhetorical affordances for protection, rescuing, being self destructive and more.

Discovering the meaning classes of *Kaboom!* involved laboring over both discovering an appropriate set of interpretive considerations, and then enumerating and interpreting what the games would mean for different combinations of answers to the interpretive considerations. When the same process was attempted on Taito's *Space Invaders* (1978) it was found to be intractable with *Space Invaders'* greater number of game entities and mechanics. Because *Space Invaders* has many more entities, collisions and potential actions of volition to consider (whether or not entities are taking an action on purpose or not), enumerating over possible responses to the interpretive considerations becomes very cumbersome. Below, the concept of micro-rhetorics is introduced to give an intermediate grouping mechanism that makes the interpretation of more complicated games than *Kaboom!* tractable.

Micro-Rhetorics

Building from the insights learned from the analysis of *Kaboom!* the following section describes an approach for interpreting a game by analyzing the lower level meanings that support higher level interpretations. For example, implicit in the interpretation that the player of *Kaboom!* is protecting the world are the interpretations that the Mad Bomber is *attacking* the world, and that the Bombs *destroy* the World. These lower level meanings are referred to as *micro-rhetorics*.

Micro-rhetorics are combined to form the complete rhetoric of a game. For example, the overall rhetoric of the newsgame *September 12th* —in which players target a Middle Eastern city with missiles intended to kill terrorists—is that the United States' policy of smart-bombing serves only to kill civilians and produce more terrorists. The component micro-rhetorics of this piece are the collateral damage that results from the imprecise targeting control the game gives players and the process of mourning civilians turning into angry terrorists.

We can use this concept to revisit *Space Invaders* and see how smaller units of gameplay come together to make the game represent aliens invading. Beyond the title alone, we can explain why the aliens appear to be invading by describing the rules of the system, assumptions about player behavior and the game's visuals. Antagonism is established because the aliens spawn bullets in the direction of the player's ship and the ship is removed from play when the bullets collide with it. The aliens' horizontal arrangement and their movement—slowly descending upon the player, side-to-side then down a row—is perceived as an invading march. And, because the player's movement is limited to a horizontal line at the bottom of the screen, the outcome is either destroying all the invaders or being overrun.

Similar to the meaning classes discovered in *Kaboom!*, micro-rhetorics can be generalized into patterns of game mechanics and beliefs about the instantial assets (visual and sounds) that can reasonably be said to represent an idea. For example, consider a collision between two sprites, A and B, after which B is removed from the screen (denoted by collision(A,B) →remove(B)). Watching this abstract rule execute would not represent a concrete idea in itself. For example, the collision(A,B) →remove(B) mechanic, could be used to represent A harms B, A makes B invisible, A eats B, A catches B and many more ideas.

The interpreter's beliefs about instantial assets determine how a set of abstract mechanics are understood. For example, if A was a picture of a shoe, and B was a picture of an ant, it is likely that an interpreter would understand collision(shoe,ant) →remove(ant) as the shoe killing the ant. Whereas if A was a man and B was a burger, the interpreter would likely understand collision(man,burger)→remove(burger) as the man eating the burger. The different common-sense beliefs about humans, burgers, shoes and ants can completely change what this abstract game rule can be said to represent.

By clearly noting what beliefs about the instantial assets are assumed in order to arrive at an interpretation of a game mechanic, like collision(A,B)→remove(B), we can now define micro-rhetoric patterns that can be employed by designers to reliably represent ideas.

The following examples and discussions will further illustrate the concept of micro-rhetoric patterns.

A destroys B:

Mechanics:

- When A collides with B, B is removed from the screen

Thematic Requirements:

- B must be vulnerable in some way
- A must be capable of causing harm

This is one of the simplest micro-rhetoric patterns, and one that is common to most classic arcade games. Simply, A collides with B and B disappears. As noted before, this abstract description doesn't complete the micro-rhetoric's definition as it also requires thematic constraints about what the visuals of A and B are. In this case, A must be understood as being destructive to B. Or even, more generally, B must be vulnerable in some way, and A must have the ability to cause harm.

With this micro-rhetoric, any two images that satisfy the thematic requirements can be applied to a game with the micro-rhetoric's mechanics and it is reasonable to say that the game represents that A is destroying B. For example, if A is an axe, and B is a tree, this instantiated micro-rhetoric would represent that an axe destroyed a tree. This particular example highlights how gameplay mechanics function metaphorically. While one could state that an axe destroys a tree, what really happens is that an axe is wielded to chip away at a tree's supporting structure until it is overcome by gravity. The game mechanics as described do not simulate this representation. The tree's removal from the screen is understood to be a simplification, or metaphor, for being chopped down. To whatever extent a player interprets the game as being about chopping down trees, it is happening as a result of the interpreter's preexisting beliefs about the visuals interacting with mechanics that afford that interpretation.

The point here is that we cannot say that this micro-rhetoric, as defined, represents something more specific like A chopping down B, as the game mechanics do not fully support this interpretation and the thematic considerations do not specify that A has the characteristics of an axe, and B has the characteristics of a tree. All valid assignments to a micro-rhetoric's entities that satisfy all of the constraints must be consistent with the desired representation. In this case, it is possible to make assignments to A and B such that it would not represent A chopping down B; however all valid assignments to A and B do result in the reasonable interpretation that A destroys B.

A attacks B by shooting at it:

Mechanics:

- A spawns C
- C moves in a straight line along the vector that A is facing when it is spawned
- When C collides with B, B is removed from the screen

Theme Requirements:

- C is a generic shape

In this example, C does not resemble any particular object, and as a result, the videogame-literate player assumes C to be a generic projectile. Of course, one could define a micro-rhetoric with the thematic constraint that C is understood to be harmful to A, but this example demonstrates how the tropes of classic arcade games, such as the aliens in *Space Invaders* shooting small white lines toward the player, have predisposed players to attribute meaning to themeless entities.

A protects B from C:

Mechanics:

- When C collides with B, B is removed from the screen
- When A is overlapping with B, a collision between C and B does not remove B from the screen

Theme:

- B must be vulnerable in some way
- C must be capable of causing harm
- A is not harmful to B

This micro-rhetoric was generalized from the game *Yars' Revenge* (1982), where a region on the screen protects the player's ships from the enemy's laser. This micro-rhetoric contains the "destroys" micro-rhetoric pattern from above. The additional entity, mechanic and thematic consideration are what create a representation of another entity protecting the entity being attacked. It seems as though this representation of protection could be rhetorically broken down into C destroys B and A helps B. However, because we cannot represent that A is helping B, without the representation of C destroying B, the micro-rhetoric for protection cannot be factored further and must contain the micro-rhetoric details for C destroys B.

These three examples hardly break the surface of what is possible to represent in 2D arcade-style games. Micro-rhetoric patterns with varying levels of generality can be defined for many more concepts (as will be shown below).

Game-O-Matic

According to the representational theory presented above, given a set of micro-rhetoric patterns, it would seem possible to design games by coming up with a message to represent, and then assembling the corresponding micro-rhetoric's game mechanics and choosing sprites that conformed to the thematic requirements. *Game-O-Matic* is an artificial intelligence driven tool designed to do exactly that.

Game-O-Matic generates simple arcade-style videogames from input that lists objects, actors and their relationships. *Game-O-Matic* also addresses a problem facing newsgames: journalism has been hesitant to adopt the form because news organizations don't have the resources to train or hire game designers and integrate game development into their workflow. The difficult processes of game design and programming are automated in *Game-O-Matic* so that the journalist need only conceive of their stories in a way that can be expressed through a concept map diagramming the relationships between entities in the story.

The following is a detailed system description of *Game-O-Matic*. In creating a system that enacts the theory of micro-rhetorics and metaphorical representation described above, many hurdles were encountered where a game's intended message would not be realized by the system. In fact, much of the system description below can be understood as “workarounds” or “patches” to get the system to generate games that were legible at all. Agre claims that these kinds of problems are the necessary result of unstable metaphors and theories driving the production of a technical artifact (in this case the theory of micro-rhetorics) (Agre 1997). Following the system description will be a discussion of the system’s (and thus the theory’s) weaknesses and where it fails to embody a complete theory of videogame representation.

From Concept Map to Micro-Rhetorics

Concept Map Input

Users of *Game-O-Matic* input their desired stories in the form of a concept map: networks of nodes and arrows where the nodes contain actors in the story (nouns) and the arrows are labeled with their relationships (verbs). A set of verbs that *Game-O-Matic* supports was determined by analyzing newspaper articles and attempting to summarize the stories using a few noun-verb-noun phrases.

Accepting input in this form implies no chronology and any that does appear in the generated games arises from the dynamics of the micro-rhetorics simulated. Also, all relationships are transitive and only involve two nouns. This limitation was introduced to maximize accessibility for non-technical users and is not a limitation of the approach.

The first stage of generation involves *interpreting* the concept map input. In this way, *Game-O-Matic* can often provide its own editorial spin on the user’s editorialized input.

In a process called augmentation, the system infers new verbs based on patterns appearing in the concept map using a library of interpretation rules. As a result of this process, we are able to apply micro-rhetorics from verbs that the user never input into the concept map, but likely intended. For example, if a user inputs "A protects B" and "C attacks B," we can, through augmentation, infer that A is protecting B from C. With this reasoned assumption, *Game-O-Matic* can apply the micro-rhetorics for the micro-rhetoric "A protects B from C" and will generate games that contain mechanics relating entities A and C (which wouldn't have been possible with the two entity relationship "A protects B").

Other replacements can be authored in the system. During synthesis, the two verbs that connect three entities are analyzed to see if they could form an entirely new verb. If a protester 'informs' a citizen and the citizen 'joins' the protester, the protester 'mobilizes' the citizen and a micro-rhetoric written for mobilize is used instead of the two mechanics for the original verbs. During decomposition, one verb is split into two. 'Repair' could be broken down into 'Touch' and 'Heal.' *Game-O-Matic* may also selectively omit verbs in very complicated concept maps.

Micro-Rhetorics

Every valid verb that the user enters into a concept map has a corresponding set of micro-rhetorics that can be selected to represent two entities with that relationship. Game-O-Matic makes use of over thirty micro-rhetoric patterns.

As shown above, purely abstract game mechanics cannot be said to represent concretely. How a set of abstract mechanics are understood is determined by the interpreter's beliefs about the depicted objects participating in the mechanics. In order for *Game-O-Matic* to reliably generate games that represent the input verbs, the user must input actors that can be reasonably understood to be related by the verb on the arrow between them. For example, the input of "man eats concrete" will not likely produce a game that will be reliably interpreted as representing that message as the thematic requirement that "concrete is edible" is not satisfied.

Game-O-Matic represents micro-rhetorics as sets of abstract entities and game mechanics that should involve them. As a high level example, a micro-rhetoric for "harms" could be that A spawns a shape that moves toward B, and when that shape collides with B, B shrinks.

*Game-O-Matic*s's system of mechanics is primarily based on the highly modular component-based framework of the PushButton Engine (PBE), a Flash game engine (PushButton 2011). PushButton is able to add a behavior to an entity by simply declaring that the entity should use a component with various parameters. Example components include RemoveOnCollideComponent, DestroyIfOffScreen, FollowBehind, and MouseController. This modularity matches the conceptual theory of micro-rhetorics very well and, as will be explained below, enables us to query the state of a game in the generation process.

This representation of micro-rhetorics also make use of a simple grammar structure to support specifying sets of possible components than can be added to a component. These are useful for gameplay patterns where the particulars of a component are not rhetorically important, and several component assignments could

Figure 5 - The structure of *Game-O-Matic's* micro-rhetorics.

be used to represent the verb. We call these assignments *non-terminals* and they are denoted by an underscore as their first character. For example, a non-terminal of "_isVulnerable to target B" can be added to an entity A and any component that could be understood as making A vulnerable to B could be selected by the generator. Every PBE component is given a set of tags that are used when the non-terminals are resolved (described below). Examples of PBE components that are tagged with _isVulnerable are RemoveOnCollideComponent, ShrinkOnCollideComponent and StopOnCollideComponent.

Figure 5 shows the structure of *Game-O-Matic's* micro-rhetorics. Micro-rhetorics are defined by a verb that it can represent, a specific id (to distinguish between the multiple micro-rhetorics that represent the same verb in the micro-rhetoric library), and a set of component assignments. Component assignments specify which entity should be assigned the component (the owner), any other entity involved (the target) and the specific parameters that the particular component should be assigned in order behave as desired by the author of the micro-rhetoric. The owner and target values are assigned either the subject or predicate from the "subject-verb-predicate" concept map structure.

For example, consider a micro-rhetoric for the input "A avoids B." In this case A is the subject, B is the object and avoids is the verb. This particular micro-rhetoric will describe a set of game rules that will have A striving to avoid collision with B at risk of being harmed in some way. The first component assignments are to make sure that both the subject and object have the non-terminal _moveInAnyWay. This non-terminal makes sure that entities have some sort of movement behavior. Next, ChaseDownComponent is assigned to the object with a parameter of evaderName being set to the subject. This demonstrates how micro-rhetoric can set variables that are specific to particular PBE components – evaderName in this case. Finally, a non-terminal component of _isVulnerable is assigned to the subject with a target of the object. Because this micro-rhetoric is defined with the non-terminals of _movesInAnyWay and _isVulnerable, it can be realized in many different ways. For example, A could be moving erratically while B moves directly toward it, and A would shrink it upon collision, or the player could control A with the mouse while being chased by B which would remove A upon collision. How non-terminals are resolved will be explained below.

With the above understanding of micro-rhetorics, we can explain the first phase of *Game-O-Matic's* generation process. For each node in the concept map, an Entity data structure is created. These structures closely mirror the structures PBE uses to run games, but we wait until the game is completely generated before we "render" our internal data structures into a form that PBE will accept. This is done to separate the generation code from the workarounds we had to introduce because of the particulars of PBE's implementation.

Next, one micro-rhetoric is selected for each verb and its parameterized components are added to Entity structures that map the nouns connected by the verb. The resulting combination of game mechanics is referred to as the *partial game description*. Because there are several micro-rhetorics for any verb, *Game-O-Matic* builds a partial game description for every combination of micro-rhetoric. The decision process for selecting the specific game that is to be presented to the player will be discussed below.

Recipes: Partial Game Description Made Complete

At this point in the generation process, we have generated many partial game descriptions that are made up of a list of entities and parameterized, or non-terminal, behavior components that each entity should have to represent the verbs in the concept map. It should be assumed that the process described below is applied to each partial game description.

Because micro-rhetorics are authored to be as abstract as possible (to maximize the system's generativity and component compatibility) and there has not been any consideration given to the overall shape of the game, there is no promise that the partially formed game description will even have such necessary features like win or lose conditions, an avatar to control, or logically placed entities. The next phase of generation looks at the partial game description, and selectively applies modifications to add structure to the abstract rules generated by the application of micro-rhetorics.

Each set of modifications to the partial game description is what is called a *recipe*. They are called recipes because the set of modifications can be understood as instructions for how to make the game become more like the gameplay pattern that the recipe was modeled after. For example, there could be a recipe that specified that the game could be won once all of one entity type has been removed from the screen. Note how this recipe would not make sense to apply if there was no behavior component that removed that entity. To avoid this sort of situation, each recipe has a set of preconditions which query the current partial game, and add or subtract from that recipe's salience, or appropriateness, score for the current game. The highest scoring recipe is chosen and the selected recipe's modifications are applied.

Precondition Predicates

A recipe's precondition is comprised of a set of *predicates*. Predicates are queries about the current game description that can be evaluated for truth. Each predicate can be a strict precondition (if it doesn't evaluate to true the recipe cannot be applied), or can have independent true or false weights that are added to the recipe's overall score.

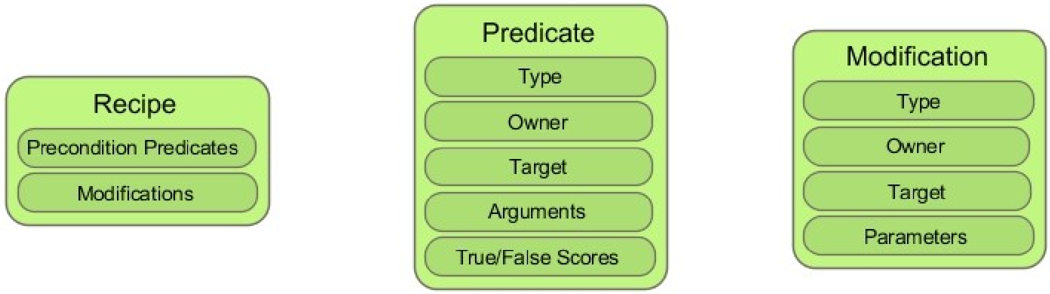

Figure 6 - Recipes are selected based on precondition predicates, which query the working game description, change the working game description to make the games more sensible.

Predicates can make queries about an arbitrary number of entities in the working game description and are authored using logical variables. For example, one predicate could query whether entity X is controlled with the mouse, and another could ask if X is spawned by Y. When evaluated, all possible combinations of entity bindings to predicate roles are considered. If all strict precondition predicates evaluate to true, and that recipe has gotten the highest score, its score and entity/variable bindings (the assignments to the variables that produced the score) are stored and later its modifications applied (see below).

Predicates can check if an entity has a component (explicit or non-terminal) or if the entity is being controlled by the player. They can also query the original concept map to see if the entity was the subject or predicate connected by a verb in the concept map input.

Modifications

Every recipe has a set of modifications that are applied if the recipe is selected. A modification changes the working game description to give it sensible gameplay and structure. Modifications can be made that add or remove components to an entity, give the entity player control and set the scale, rotation and placement of an entity. In a modification, entities are denoted by logical variables that are resolved by the same variable bindings that created the highest score for that recipe during precondition evaluation. For example, one modification would be to add a component to entity X that makes it follow Y closely.

Recipes also have access to a shared blackboard that recipe modifications can write to and predicates can later query. This is used to allow recipes to communicate with one another about things that aren't easily represented in entities and components. For example, one recipe can note on the blackboard that an entity is intended to be the primary antagonist to the player and a later recipe can use this when setting entity positions.

Types of Recipes

Three types of recipes are scored and then applied to a game in sequence: win, lose and structure. *Game-O-Matic* makes use of 15 win recipes, 6 lose recipes and 11 structure recipes.

Win recipes determine the player's goal. Examples include removing all of one type of entity, having the player move to the right side of the screen, and surviving for a specified amount of time. Of course, not all partial game descriptions support all win conditions. For example, if the player has no way to remove an entity, it doesn't make sense to have the goal be to remove all of them from the screen. Preconditions and modifications enable us understand the current state of the generated game and modify it to be a more coherent game that players will be more likely to understand how to interact with.

Lose recipes determine what causes the player to lose the game. Examples include running out of lives, failing to protect one entity from another, and not getting a high score in a specified amount of time.

Both win and lose recipes contain templates that are used on the final game's title screen to tell the player what he or she should try to accomplish and try to avoid. For example, "Make %X% huge to win" and "Lose if %Y% removes %Z%." Also, in the final game, upon triggering a win or lose condition a screen will pop up that can hold custom text that the user can author in *Game-O-Matic's* interface.

At this point in the generation process, entities are specified, they each have mechanics that represent the micro-rhetoric and there are appropriate ways to win or lose the game. However, the game lacks sensible structure. Where are the entities placed on the screen? How big are they? Should specific entities be limited to specific regions of the screen? This sort of information is what structure recipes are meant to provide.

Structure recipes are roughly modeled after classic arcade games. For example, the structure of *Frogger* (Konami 1981) could be appropriate to impose on a game where an entity A strives to collide with entity B, but something negative happens to A when it collides with C. Mirroring *Frogger*, where the player controls a frog trying to get from one side of the screen to the other while avoiding obstacles, the structure recipe would put A and B on opposite sides of the screen and put C between them moving erratically in order to create a challenge for the player. In terms of *Frogger*, A is roughly the frog, B is the goal area (lily pad) and C acts like the cars. In the current version of *Game-O-Matic*, we have defined structure recipes based on *Frogger*, *Space Invaders*, *Kaboom*, *Asteroids* and several other custom patterns.

Note that applying the structure of a classic arcade game does not mean that the generated game will be a skinned clone of the arcade title. Not only will the player have different win and lose conditions, but the mechanics of the game will be completely different. For example, a game with the structure recipe inspired by *Space Invaders* could have the player controlling the bullets of the invaders trying to avoid colliding with the ship at the bottom of the screen. Also, as structure recipes don't modify the movement behavior components established by the micro-rhetorics, entities move in ways that can make the original game inspiring the structure recipe unrecognizable. The purpose of structure recipes is to ensure entities are spaced sensibly, in reasonably familiar patterns, such that movement around the screen maximizes the entity interactions specified by the micro-rhetorics, such as the *Frogger* recipe maximizing the negative interactions between entities A and C above.

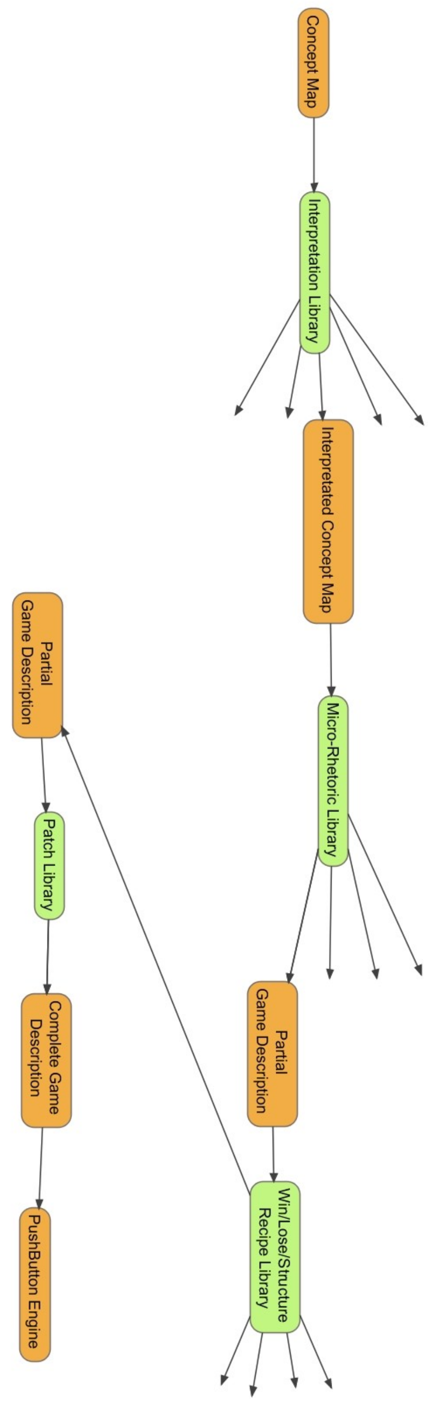

Figure 7 A diagram that shows the generation process for *Game-O-Matic*. The green nodes show the choice points in the generation process. First the concept map is interpreted, next different combinations of micro-rhetorics are chosen and finally different recipes are applied.

Finalizing the Game

At this point, any non-terminal components still remaining in the entities after the recipe modifications have been applied are resolved to specific PBE components by randomly choosing a specific PBE component that is associated with that non-terminal.

Finally, a set of patches are applied to fix any unexpected problems that arise from combining all of these independently authored structures. These have the same form as the recipes, except all preconditions are strict and all relevant patches are applied (rather than just one). Patch preconditions often make use of the blackboard. For example, if the win condition is to make it to the right side of the screen, and a structure recipe has moved the player's starting location away from the left side, a patch would recognize this and move it back. The patch phase allows an easy to author, case specific final check to make sure the generated game is as good as it can be.

At this point, the system has generated a complete game for all combinations of micro-rhetorics. The system chooses which game by adding the scores of each recipe (win, lose and structure) to form at *total score* for that game. The game with the highest total score will be selected to be presented to the player. At this point, the complete game structure, made up of entity and components specifications, is written out to Pushbutton Engine's XML level file format and the game can be played.

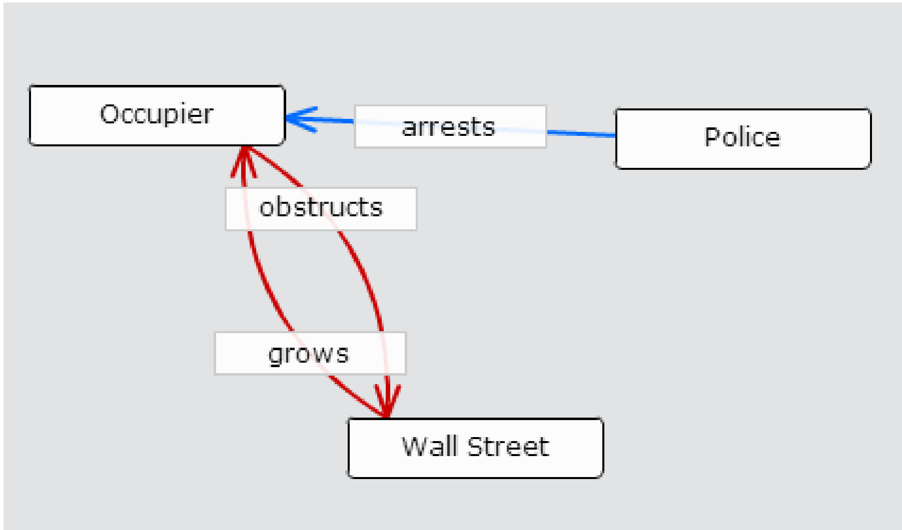

Figure 8 - An example concept map created to represent a newspaper article

Example

The following section explains *Game-O-Matic's* processes using a specific example.

Concept Map

On the six month anniversary of the Occupy Wall Street movement, protesters returned to New York City's Zuccotti Park and several were arrested. Figure 8 shows a simple concept map meant to capture a high level description of this story. From the diagram we can see that the occupiers are obstructing Wall Street and are being arrested by police, but Wall Street is also growing the occupy movement. The concept map represents three relationships between Wall Street, the occupiers and the police.

Micro-Rhetorics

For clarity, the following will describe the generation process for one game, as opposed to the generation of all combinations of micro-rhetorics. For each verb, *Game-O-Matic* selects one micro-rhetoric that is tagged as representing it. For "arrests" in "police arrests occupier," it selects "take custody" micro-rhetoric. This gives the police and occupier entities a _movesInAnyWay component, and the occupier is given a StopOnCollideComponent which is activated upon a collision with police. _movesInAnyWay is a non-terminal which can be applied to either the player or NPC and will be converted into a specific PBE component later in the generation process.

For "occupier obstructs wall street," an *obstructs* micro-rhetoric is selected where one entity stops the movement of the other. This gives the StopOnCollideComponent to Wall Street, thus preventing Wall Street's movement while colliding with an occupier. Other possible micro-rhetorics for obstructs could have been "redirect," which would have given Wall Street a ReflectOnCollideComponent with a target of the occupiers, which would cause Wall Street to bounce off of the occupiers.

Next, for "Wall Street grows Occupiers," a grow micro-rhetoric is selected that gives the GrowOnCollideComponent to the Occupiers with a target of Wall Street.

Choosing Recipes

At this point, the system knows all of the entities, and the micro-rhetorics have given them a small set of components. Next, the win, lose and structure recipes are scored based on the partial game description and one of each type is applied. First, the win recipes are scored. The first win recipe sets the win condition to be for the player to "score 100 points." This recipe has the precondition:

Y has a component of type _isVulnerable to X. [True: +4/False: -0]

The Y and X in the preconditions are entity bindings, each recipe will be scored for each possible combination. In the case of Y=Wall Street, X=Occupier; the recipe has a score of +4, because StopOnCollideComponent is tagged as being type "_isVulnerable." For Y=Occupier, X=Wall Street, the score would be 0 because GrowOnCollideComponent is not tagged in this way. If a precondition such as "_isCollidable" (which both the Stop and Grow components are tagged with) were used instead of "_isVulnerable" the two bindings would have equal scores. The highest the top scoring recipes are chosen at random.

Applying Recipe Modifications

After selecting the winning win recipe, its modifications are applied. Assuming it was "score 100 points" with Y=Wall Street, X=Occupier; this recipe would make the following modifications:

1. *Write to blackboard: "removeToWin Y"*
2. *Remove component: Y _isVulnerable with target X (this will remove any components Y has that are tagged as _isVulnerable)*
3. *Add component to Y: _isRemovedBy with target X*

4. *Add component to Y: ScoreRemovalOfComponent with parameters: winScore=100, scoreEachRemoval=10*
5. *Add component to Y: RespawnOnRemoveComponent*
6. *Make X the player*

As recipes are applied, the variables are substituted for their bound entity. Modification 1 write on the blackboard “Wall Street is being removed to win,” which could be checked in the preconditions of a later recipe.

Modification 2 removes the “_isVulnerable” tagged component StopOnCollideComponent from Wall Street and modification 3 replaces the removed component with a stricter “_isRemovedBy” component, such as RemoveOnCollideComponent in order to guarantee that the player will be able to remove Wall Street and win the game. Replacing a StopOnCollide from “obstructs” with RemoveOnCollide constitutes a change to the micro-rhetorics first built from the concept map. Occupier removing Wall Street as a form of obstruction seems reasonable. This rhetorical leap enables *Game-O-Matic* to give novel interpretations of the system represented in the concept map, but can sometime significantly change the message being expressed. For example, in this case, the idea of the occupier obstructing Wall Street is replaced with something that might be more interpretable as the occupier destroying Wall Street.

Modification 4 adds a ScoreRemovalOfComponent to Wall Street, so that each time it is removed, 10 points are added to the score, and if the score is 100, the game is won. Modification 5 makes Wall Street respawn each time it is removed, so that the win score can be reached. Finally, modification 6 gives the player control over the Occupier.

Next, the lose and structure recipes are scored and of the highest scoring recipes for each type, one is randomly selected and its modifications are applied. For the example below, the "run out of time" lose recipe is selected, which adds a MeterComponent to the World. The World is an entity which holds global components, such as UI elements and components which instantiate entities into the game.

Next assume the *Frogger* structure recipe is selected, which places the player on the top, an entity which collides with the player on the bottom, and several of another entity in the middle, in this case the Police. The middle entities are limited to only allow horizontal movement. The bottom entity, Wall Street, is set to double size.

Finalizing the Game

If a recipe hasn't already set the player, one is selected randomly from the nouns on the concept map. Non-terminal components are resolved to terminal components, and then patch recipes are applied. Patch recipes aren't scored; they are all applied if their preconditions are met. The "everything moves" patch gives a movement component to every entity which does not have one. After all patches are applied, any remaining non-terminal components are resolved, and the instruction text for the start screen is generated.

At this point, all entity components are in place and we can start generating the XML which will be read into the PushButton engine. As we generate, some component parameters will have a single value, but others will have a range of values, which we select randomly. For example, if a micro-rhetoric or recipe modification has not set a components parameter, such as movement speed, the value is randomly selected from a defined range defined per parameter.

With the XML generated, we can load it into the game engine. Components like MeterComponent and ScoreRemovalOfComponent will report to the UI to get their elements on the screen.

The top of Figure 9 shows the start of the game which tells the player that he controls the Occupier with the arrow keys, and will need to collect 100 points worth of "Wall Streets" before the timer runs out to win. When the game starts (bottom of Figure 9), the occupier dashes past the police until the player manages to make it run into the Wall Street. At this point, Wall Street begins to shrink (the system chose ShrinkComponent when it resolved the non-terminal "_isRemovedBy"). Wall Street shrinks until it bleeps out of existence, the player gains 10 points, and a new one is spawned to take its place. All this time, the occupier is growing and will soon be stopped by police as the player moves the occupier to collide with the next Wall Street. As the occupy movement grows to fill the screen, overwhelming the police forces, removing Wall Street happens without any actions from the player. And this is just the first game generated! With the press of a button the user can generate other games that carry different interpretation of the user's input and different gameplay.

Discussion of Micro-Rhetorics and *Game-O-Matic*

Game-O-Matic takes a network of nouns and verbs as input and constructs games that represent the network using micro-rhetoric patterns of gameplay. It is the only extant videogame generator that is able to create games *about* concepts and arguably generates videogames of higher complexity than the other videogame generation systems (see appendix). However, it can hardly be said that the messages it is able to express are very profound, or that the games it makes have lasting entertainment value. This section suggests reasons why this is the case and critiques the theory of micro-rhetorics.

First, the practical concerns. *Game-O-Matic* makes use of several hand authored libraries: concept map interpretation rules, micro-rhetoric patterns, win, lose and structure recipes and patches. Because a goal of the system is to give the user the open ended ability to input any noun connected by almost any verb, it is very difficult to author for these libraries while maintaining coverage of both the conceptual and game mechanics space. Furthermore, when various elements of the library are added to a game, they interact with each other, making it hard to predict how a authoring choice might be modified by a later part of the generation process. Also, to maximize the generativity of the system, the libraries were authored to make the fewest commitments possible. As examples, where non-terminals could be used in micro-rhetorics, they were. Or when an aspect of gameplay meant to be created by a recipe wasn't essential to the desired gameplay, it was not included as a modification. While this succeeds in ensuring a wide range of variability, authoring the libraries in this

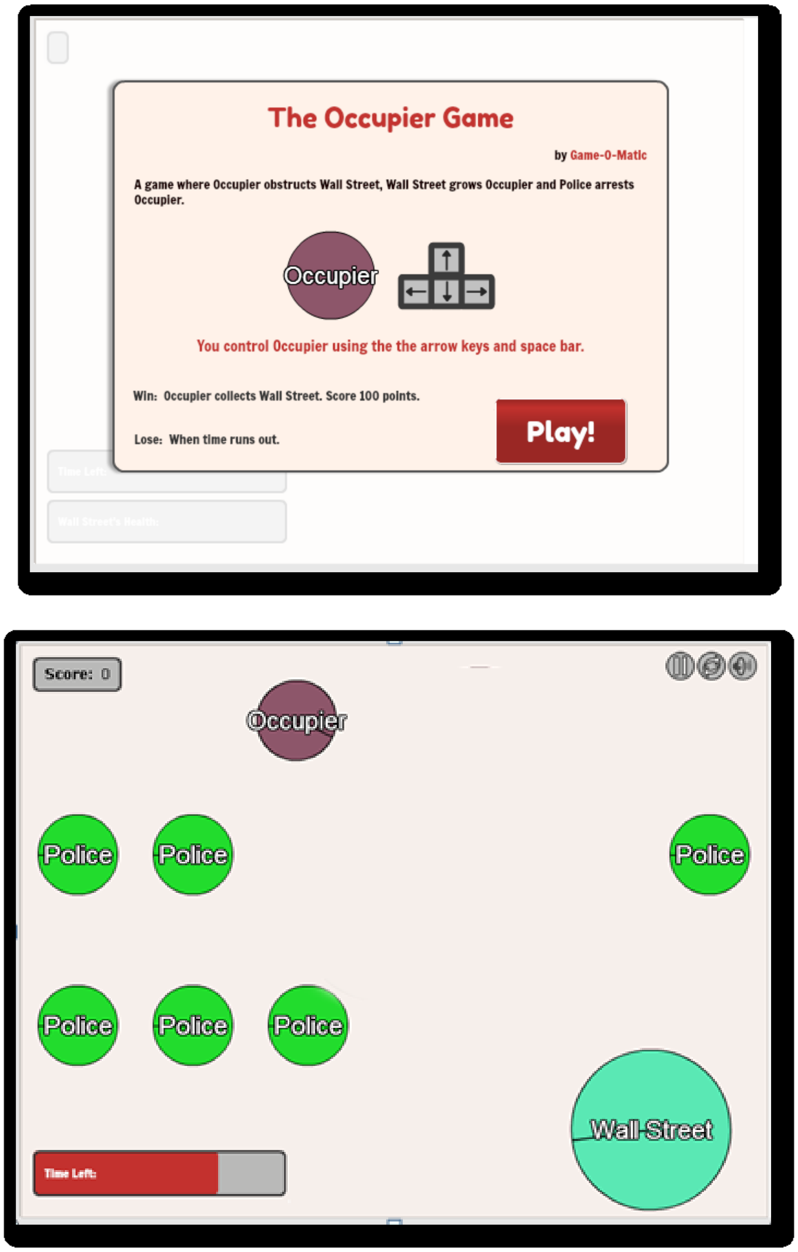

Figure 9 - The instruction screen and a screenshot of the game generated from the concept map in Figure 8.

manner requires a good deal of error prone abstract thinking. Furthermore, with so many phases of generation, tuning and debugging *Game-O-Matic* is a very difficult process and requires a comprehensive and thorough understanding of the inner workings.

Game-O-Matic's inability to create games of lasting significance can also be understood by explaining this observation as the product of the full implementation of an incomplete theory. Agre notes that "the limit pushing of technical work will reveal its margins" (Agre 1997). By margins, Agre is referring to Derrida's project of deconstruction which strives to show that any theory's *center* (the applications which confirm its correctness) also have *margins* (applications or points of view where the theory is found to be incomplete or inapplicable). Derrida wanted to show that margins can be as important as centers, and should not be ignored. Agre's notion of critical technical practice (of which this document is written in the tradition of), describes a process where an artist/engineer creates technical work to reveal the margins of theories that were used to create it in order to create better theories. The following is a discussion of how the system reveals the weaknesses of the micro-rhetoric theory.

A good deal of *Game-O-Matic's* processes are not about crafting the meaning of the generated game, but rather are there to give the games enough structure for a player to have something to do and be able to reasonably engage with them. The recipe system described above is how *Game-O-Matic's* games go from being a jumbled assortment of abstract rules and randomly placed entities to games with goals and reasonably paced gameplay that players are able to engage with.

The assumption that individual units of representational meaning can be separated from the overall structure of a game is made explicit in the micro-rhetoric theory and *Game-O-Matic's* implementation of it. *Game-O-Matic's* understanding of where meaning in games comes from imposes a hierarchy between *essential* and *nonessential* mechanics and groupings of mechanics that players are supposed to find meaningful. Generation decisions that are deemed nonessential are made by areas of the system that do not consider the meaning it is attempting to generate. Specifically, recipes add to and make modifications to the mechanics of the micro-rhetoric patterns as if the micro-rhetoric patterns will maintain their rhetorical significance and representational priority. In other words, it is assumed that a collision between A and B, which causes the removal of B, has *more rhetorical significance* than choices made by the recipe system, such as the way in which A moves, or where B is placed on the screen. *Game-O-Matic's* design assumes that players will *uncover* the relevant mechanics from the micro-rhetorics and note them as the primary features of their gameplay experience to consider for interpretation. This is arguably a reductive view of player experience.

Another assumption *Game-O-Matic* makes is that micro-rhetoric patterns are additive. The concept of the micro-rhetoric involves conceiving of an isolated segment of gameplay and bracketing the rest of the game with a tacit assumption that any changes that are introduced will only have local significance. However, once a

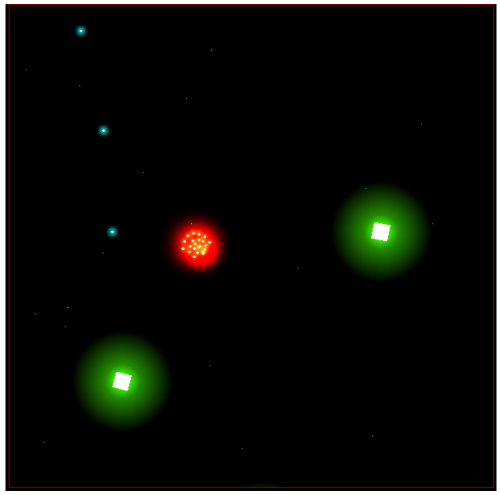

Figure 10 - A screenshot from *Kosmosis*. The green war machines are overcome by the red space prolets.

game is given to players, it can hardly be said that they will recognize the same micro-rhetoric segments that the author intends for them to consider, or that elements of micro-rhetorics will not combine to produce unintended micro-rhetorics that are at odds with the desired representation. *Game-O-Matic* especially suffers from this because recipe modifications are made independently of the micro-rhetoric choices.

This complication is evident in Molleindustria's *Kosmosis*, a short form game created to be a "procedural representation of collectivist/revolutionary statements..." (Molleindustria 2009) (Figure 10). In this "shoot 'em up" style game, the player uses the arrow keys to control a small shape, labeled the "vanguard" in the introductory text. When the vanguard collides with inactive small white shapes, labeled the "space prolets," they begin to swarm around the vanguard. Also on the screen are inactive green dots which are identified as the "war machines." If the player collides with the war machines, all the space prolets are dispersed and cease to follow the vanguard.

When enough space prolets surround the vanguard, the player can press the spacebar to transform itself and the space prolets into a large yellow shape that can push the war machines off the screen upon collision.

The collision between the yellow shape—the vanguard and space prolets transformed into a greater force—and the war machines represents an attack. It implies antagonism between the concept of a war machine and the proletarians, and after a collision, the war machine no longer exists on the screen. A critical mass of proletariats can dismantle the war machine.

Also, the collision between the space prolets and the vanguard represents the vanguard mobilizing the prolets as the concept of a vanguard is exactly that they lead proletarians in revolution – the space prolets become active upon collision with the vanguard. Individually, each micro-rhetoric is consistent and convincing. However, the relative behavior of the war machines and the space prolets comes into question when both micro-rhetorics are combined into the same game. The inactivity of the war machine and prolets, when compared with the busy and aggressive activity of the vanguard, give the sense that the vanguard is the only active agent of change in the microworld. The war machines are not much of a threat, highlighted by the fact that the only active entity is the vanguard. The vanguard's attacks appear unmotivated and aggressive, undermining the game's authorial intention of creating a game where the "non-degenerated socialist values are hegemonic." This problem could have been remedied by adding a micro-rhetoric of the war machines attacking the space prolets.

However, this was only made necessary because of the interaction between micro-rhetorics – a phenomena not explored in the theory as presented.

The last critique I will offer of this theory of meaningful videogame design is about the complexity and quality of the games that can be created using it. Even when *Game-O-Matic* succeeds in representing the user's input, it is not clear that the resulting game is any more impactful, relevant or expressive than a short paragraph of text describing the same situation. The same critique could be applied to *The Marriage* and the other subset of games that employ procedural rhetoric in this fashion. A micro-rhetoric pattern relies on players to enact specific mechanics, or sequences of actions, in order to carry its claimed meaning. In this way, it has been argued that games like those that *Game-O-Matic* generates and newsgames and artgames leave little role for the player and demands that players enact specific sequences of action (Sicart 2011) thus making the gameplay experience arguably like that of static media like film or literature.

The themes of great works of literature and film are surely not reducible to a few sentences. And when they are, it is understood that these descriptions are summaries that leave out any number of nuanced themes, many of which are intangible. While arguments can be made about the value of promoting system level analysis, as a design philosophy, *Game-O-Matic's* simple and narrowly interpretable games are evidence that micro-rhetorics are not the ultimate unit of analysis for games. This isn't to say that a game's complexity necessarily correlates with the complexity of what it can represent. An interesting juxtaposition of images alone could make for a very expressively deep or interpretable game, but these games hardly utilize the interactivity that is inherent to videogames. *Game-O-Matic's* operation enacts a criticism of this sort of game mechanic essentialism and thus itself. That being said, the theory led to the creation of *Game-O-Matic*, thought provoking and entertaining artifact in its own right.

In summary, this section began by analyzing a subset of games that attempt to employ procedural rhetoric through using simple rules metaphorically. Next, the theory of micro-rhetorics was developed that made explicit a design theory that was implicit in the analyzed games. Next, an AI architecture and media artifact, *Game-O-Matic*, was presented which enacted the micro-rhetoric theory. Finally, by reflecting upon the process of creating and using *Game-O-Matic*, the margins of the theory that drove its creation were revealed, along with the implicit design theories of existing games such as *The Marriage* and *Kabul Kaboom*.

In conclusion, while this chapter began by attempting to understand how a game's rules alone can be representational, the investigations and systems built ended up showing how only through the interaction of instantial assets and rules are games representational. The following section will strive to create a theory of game representation that accounts for more complicated rule systems where player action is not prescribed.

Chapter 3. Simulation Representation

On the user generated news and entertainment website Reddit, a user uploaded a story titled "I've been playing the same game of *Civilization II* for almost 10 years. This is the result." In describing the state of his unusual experience with the game he said "The world is a hellish nightmare of suffering and devastation." The article goes on to describe how the game's world is comprised of three warring factions fighting over the resources, which they only use to destroy each other in a perpetual war. In short, the player had discovered a state where *Civilization II* embodied a dark sci-fi dystopia from which, only with the help of Reddit community, the player was able to escape (Lycerius 2012). Could it then be said that *Civilization II* is *about* a dismal future? While this might be a reasonable interpretation based on this particular player's experience, this "hellish nightmare" world is hardly representative of the majority of experiences with *Civilization II*, which often describe the rise to a technological utopia or a happy kingdom.

With such a breadth of possible outcomes, it would seem that in order to speak about *Civilization II*, one would need to account for all of these gameplay experiences. Game scholar Kurt Squire, in his work studying the use of *Civilization III* as a teaching aid in schools, provides such an interpretation by saying that *Civilization II's* sequel has "unique affordances as a world history simulation." He sings the praises of the game by saying "Civilization III represents world history not as a story of colonial domination or western expansion, but as an emergent process arising from overlapping, interrelated factors" (Squire 2004). Squire's claim about what *Civilization* accounts for and represents both the dystopian and utopian gameplay experiences by providing an account of the system that generated the experiences as opposed to any specific representation. Also, it is interesting to note that it would be difficult to use the concept of micro-rhetorics as an analytical tool to arrive at an interpretation like Squire's that is able to describe many types of gameplay experiences. Micro-rhetorics describe specific representations, where what Squire is describing is representative of many representations.

As another example, *SimCity* was one of the first games to foreground the idea that videogames can be arguments about "how the world *functions*" and is often considered the canonical simulation game. Playing *SimCity* involves taking actions such as raising taxes, building roads, or choosing where to place a power plant, in order to influence the growth of a simulated city. Over time players develop a sense of which choices create different outcomes and eventually they are able to shape cities with intention. Wardrip-Fruin describes this process where players learn a mental model of the system that governs the outcomes of interaction as the *SimCity Effect* (Wardrip-Fruin 2009).

SimCity was able to achieve a level of believability about its subject matter by having complex and deep gameplay, as well as being marketed as a simulation (a term typically reserved for science and public policy). This led to concern that players might accept the claims of the simulation as fact without recognizing the biases and assumptions that underlie the simulation (Starr 2001). It has been claimed by those on both the political right and left that *SimCity* ideologically biased (Friedman 1996). Different understandings of the same media artifact are common, however adding to the interpretive ambiguities that arise with other forms, such as film and literature, any particular player's experience of a simulation is bound to be significantly different than another's. Videogames produce a space of possibilities, and for most, interpretations or understandings of a game will involve particulars of their gameplay session rather than a comprehensive view of what the game affords.

Because the processes of a game like *Kaboom!* are evident by observing the movement of the game entities, it was easy to describe the observations about the games in terms of simple rules (e.g. *when A collides with B, B will disappear*). However, in a game like *SimCity*, at any given time, there are many interacting variables that might be argued to have produced an effect. For example, when a building becomes abandoned, a player might believe that it is because there wasn't a park nearby, or that taxes were too high, or any other number of reasons. Making matters worse, in most cases, changes to the game state are influenced by many factors at once. This makes it difficult, if not impossible, to confidently describe the processes of *SimCity* using simple rules about the discrete entities that appear on the screen. Instead, *SimCity* makes use of complex resource management rules where quantities of money, population, pollution interact to produce results. In other words, the rules of the game are not directly evident to the player.

The following section explores how simulation games can be interpreted as systems that generate representations, rather than particular representations. Unlike games built by configuring micro-rhetorics, a kind of game and interpretation mode will be identified where players find a game meaningful through an understanding of how its system operates and interpreting understanding it as representing a set of general principles about its thematic domain. Beginning by teasing out a distinction between games that primarily represent through instantial assets and those that represent through simulation*,* I will present a theory of simulation representation and apply it to the early Apple II game *Lemonade Stand.* Next, a videogame that represents a theory of social interaction, *Prom Week*, will be presented in full detail. Finally, this design and interpretation approach will be discussed through an analysis of *Prom Week* and its interface. The analysis will be discussed from my perspective as being both a lead designer and a core technology programmer for the game. As an instance of critical technical practice, the difficulties of getting *Prom Week* to represent a set of ideas about social reality are evidence that the theory that drove the practice was unstable. The chapter concludes with a discussion of the weaknesses of the simulation theory and how these weaknesses point toward theoretical solutions.

Identifying Simulative Representation

In the language of semiotics, the distinction between the games like *Kaboom!* and games like *SimCity* is a matter of which semiotic *codes* an interpreter employs to understand the game's signs. Given the indeterminate nature of a sign, it cannot be said that a signified *must* follow from any signifier. A code is a rule, or convention, that generally tells the interpreter which of the signifieds to select from the many that are possible. For example, the appearance of a green circle on the screen of an abstract videogame could represent a green apple or a non-diegetic interface element to indicate that the player should start moving (like a form of traffic control) depending on which semiotic codes the interpreter utilizes. Codes are culturally determined and their application is a product of the individual and their beliefs about the rest of the semiotic system. If the player believed they were playing an abstract farming game about collecting food, he might understand the green circle as an apple to collect. But if he believed he was playing a racing game, was not currently moving and carried the cultural association of green with "go," he might understand it as a "green light" that indicates the beginning of a race. Codes determine which of the many possible interpretations to believe.

Groups of semiotic codes will often be employed in the same contexts and groupings of these codes have been referred to as *semiotic registers* (Huber 2012). Genre conventions often establish which semiotic register an interpreter adopts when encountering a system of signs. Applying this concept to games, Huber describes semiotic registers as "a conceptual entity produced by the player's attempt to understand and successfully play a game by organizing the signs he/she encounters" (Huber 2012).

The following discussion of two hypothetical segments of gameplay will conceptually develop two registers of videogame signification. The first segment uses a thematically consistent set of visuals but seemingly incoherent mechanics. It will be claimed that it successfully signifies via an *instantial register* which relies on codes that emphasize beliefs about a game's instantial assets. The other will use an arguably incoherent set of visuals and mechanics that are able to represent *despite* the visuals. It will be argued that this segment is able to represent because of a *simulative register* that emphasizes codes about the game's processes.

The Instantial and Simulative Registers

Most classic arcade games can be said to signify in what could be called an *instantial* semiotic register. Understanding a game of the instantial register primarily involves beliefs about instantial assets (visuals, sound, cut scenes, etc.). Often, these games could be understood by just watching gameplay traces, and thus it can be said that the forms of representation in these games bears resemblance to those of static media.

Consider the following description of a gameplay segment:

> *A picture of a screwdriver moves around the screen and periodically white circles appear near it and move away along the angle that the screwdriver was facing at the time of the circle's spawning. Also on the screen are pictures of screws which are removed upon collision with a white circle (Figure 11).*

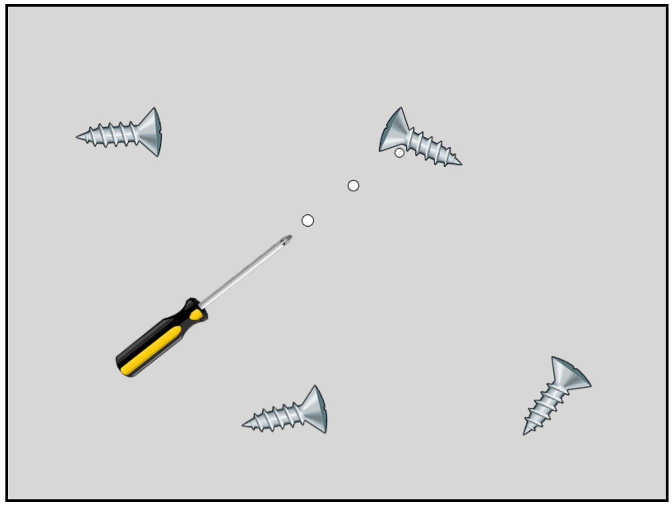

Figure 11 - A screenshot of a screwdriver shooting white dots at screws might metaphorically represent a screwdriver screwing in screws.

Given the history of space "shoot 'em up" style videogames, this might be said to represent a screwdriver *shooting at* and destroying screws. When taken as a literal visual representation, this gameplay segment is nonsensical. However, if select parts of the elements are emphasized, the visuals, and others deemphasized, the mechanics, a more sensible interpretation is possible. I propose that the segment could be a highly abstract, metaphorical, representation of a screwdriver *screwing in* screws.

This claim is not hard to accept if the game mechanics aren't considered. Screwdrivers are designed to screw in screws and those two entities are the only entities in the game world. Even a still picture of a screwdriver and screw might be said to represent, or at least imply, this concept. However, it can also be argued that the mechanics support this interpretation through metaphor. In the details of how this metaphor functions is the distinction between the instantial and simulative registers.

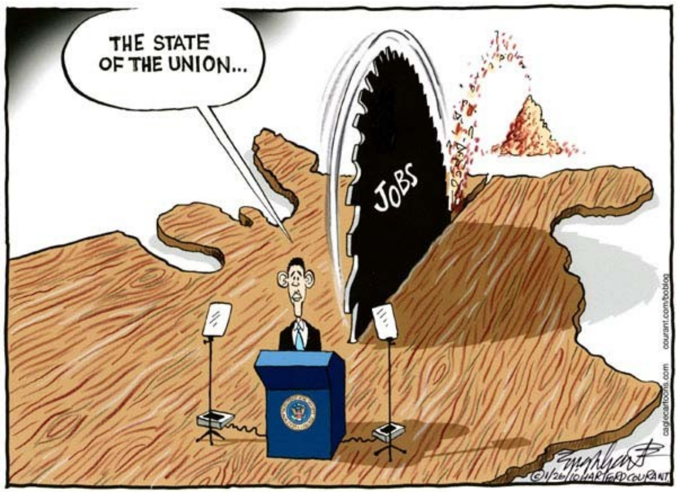

Figure 12 - An editorial cartoon that makes use of multiple visual tropes.

Lakoff and Johnson write "the essence of metaphor is understanding and experiencing one kind of thing in terms of another" (Lakoff and Johnson 2003). Metaphors allow interpreters to transfer the qualities of one concept and apply them to another. As an example, Figure 12 shows an editorial cartoon that makes use of caricature, physical and spatial metaphor, literal signifying images and text to communicate a message about an encroaching and dividing problem of unemployment in the United States. The impending disaster is communicated through the use of a destructive entity, the saw, which is labeled "jobs." This labeling of objects to make a metaphor explicit is an established convention of the editorial cartoon and indirectly communicates using both visual and written signs. A further explanation of how this metaphor functions sheds light on how metaphor applies to the screwdriver and screw gameplay segment.

Peirce's semiotics describes three kinds of signs: iconic, indexical and symbolic. Iconic signs represent because they physically resemble what they stand for. The image of the saw blade is an example of an iconic sign. Indexical signs are those that reliably correlate with what it stands for. Understanding that the saw being behind President Obama represents that he does not anticipate the problem of jobs involves an indexical sign. The saw's placement does not directly visually resemble the image that it signifies, yet it still brings to mind concepts that are associated with saws colliding with people because those concepts are directly correlated with saws moving toward people. Symbolic signs are culturally determined signs that map from signifier to signified via convention. Most words are symbolic signs.

Operating as iconic signs, a picture of a saw cutting through a picture of a board will not likely cause any difficulty for an interpreter as most believe that saws can cut through wooden boards, and the image they are considering denotes this belief through visual similarity to their existing beliefs about how the world functions. Without being attached to the saw, the word "jobs" would be shown to be dividing the board, and an interpreter would be forced to confront the unbelievable implication that jobs cut through wood, which would no longer be able to be justified through visual similarity. The interpreter would not likely interpret it as a symbolic sign as the convention of understanding the words "jobs" in this context is not established.

The labeling of the saw with the "jobs" allows us to understand jobs in terms of a saw. Jobs are now understood as being able to cut, destroy and divide. Also through metaphor, the wooden board shaped like the United States that President Obama is standing on, is able to be cut, destroyed and divided. This symmetry between metaphors allows this editorial cartoon to convey a lot of information about abstract concepts in a single image. Metaphors function by attaching one concept, the *tenor*, to another, the *vehicle*. The tenor is the concept which features are ascribed to and the vehicle is the concept from which the features are inherited from (Richards and Constable 1936). The interpretation of the editorial cartoon described above treats the saw and board as the vehicles, and jobs and the United States as the tenors.

Returning to the screwdriver and the screw, we can use the metaphorical and semiotic language to describe why we might interpret the "screwdriver shoots at screws" mechanics as a representation of the screwdriver screwing in screws. Unlike the editorial cartoon, where both the tenor and vehicle were visually represented, the gameplay segment as described is made up of visuals and game processes. The claimed interpretation of the screwdriver screwing in the screws becomes possible only by ascribing the metaphorical roles of tenor/vehicle to each of these qualities of the text, despite processes not being *visual*. In this case, the gameplay segment's tenor, the screwdriver and screws, can be said to inherit from a *procedural* vehicle.

A procedural vehicle gets its meaning from the way in which the abstract operation of the processes can be narrated. In describing the meaning of abstract machinery, Agre argues that machines have *narrative affordances* (Agre 1997). The ways in which we can sensibly, or convincingly, narrate the operation of game mechanics is constrained by the shape of the computational material and cultural context of the interpreters. In other words, if there is a game mechanic where when A collides with B and B is then removed from the screen, it will be unconvincing to narrate the enactment of these mechanics as something that contradicts it. For example, if A was believed to be a human head, and B looked like a hamburger, and we attempted to narrate the enactment of the mechanics as the hamburger eating the man, it would be unconvincing as that interpretation would seem to imply that A, the head, is removed from the screen upon collision with B, the burger. This is explicitly not the case and can never be the case because code runs on the machine outside of the player's control.

When we abstract away the visual representations from the screwdriver gameplay segment, we have mechanics that can be described by the following game rules: *A spawns C, C moves in a straight line along the vector that A is facing when it is spawned* and *when C collides with B, B is removed from the screen.* Because it visually appears as though A is producing C, and some event occurs when C collides with B, solely based on this simple narration of the abstract processes, an interpreter might consider the general question "for what purpose might A be producing C in order to act upon B?"

When considered at this level of abstraction, we can now consider how the visuals relate to the mechanics. Given the thematic assignments of A as a screwdriver, B as screws and C as an abstract shape, we can reconsider the question as "for what purpose might a screwdriver act upon a screw?" To this question there is the obvious response of our claimed interpretation that a screwdriver *screws in* a screw. In this interpretation, the role C, the small white circles, plays is to establish through metaphor that A is acting upon B. For this metaphor, the interpretation that the procedural mechanics represent that A is acting upon B serves as the procedural vehicle, and the belief that A and B are screwdrivers and screws serves as the instantial tenor. Thus, screwdrivers metaphorically inherit the quality of acting upon screws.

Admittedly, this metaphor provides a weak, or possibly unconvincing, interpretation. Furthermore, the procedural vehicle that the tenor of the metaphor relies on only relates to the subject matter of screws and screwdrivers in the most general of ways (most things can be said to *act upon* others in some capacity).

Now consider a videogame that represents part of the editorial cartoon described above:

> *The player controls a saw labeled with the word "jobs" and is given a goal to collide with wooden boards shaped like the United States. Upon collision, the boards split in half and fade away.*

Like the screwdriver example, this segment of gameplay metaphorically can be understood using a procedural vehicle and a visual tenor. Even in denotatively describing the visuals that result from the mechanics being enacted, *A collides with B and B is split in half*, it fairly accurately describes what happens when a saw is put to wood. When images that are consistent with the narration of the abstract processes are applied, a stronger coherence is achieved and it seems more likely that an interpreter will arrive at a similar conclusion: jobs are dividing the United States. Bogost refers to this alignment between processes and instantial assets as a "tight coupling" (Bogost 2007). This segment of gameplay is able to represent just as clearly as the editorial cartoon because "the elements presented on the surface have analogues within the internal processes and data" (Wardrip-Fruin 2009).

However, even when nonsensical instantial assets are put into the game, this particular set of abstract mechanics still might be argued to represent using a similar metaphor. Consider the same gameplay segment with seemingly nonsensical visuals:

> *The player controls a cupcake and is given a goal to collide with manhole covers. Upon collision, the manhole covers split in half and fade away.*

This game appears to represent that cupcakes divide manhole covers. This statement may be nonsensical, but we still are able to grasp it because of the processes and despite the visuals.

The screwdriver and the screws, and the cupcake and the manhole represent two separate semiotic registers: the instantial and the simulative. In the screwdriver segment, an interpretation was arrived at mostly through preexisting beliefs about the instantial assets rather than through consideration of the mechanics. While it was able to be argued that the mechanics supported the interpretation, it was only after abstracting the narration of them to a point that almost any two objects could be related by them. In other words, the mechanics were not simulating any particular aspects of screwdrivers screwing in screws. The defining characteristics of the instantial register are that interpretations are greatly determined by existing beliefs about a game's instantial assets and mechanics only relate at high levels of abstraction.

In contrast, the cupcake and the manhole cover is able to represent despite the nonsensical relationship between beliefs about the instantial assets and the processes. Different than the screwdriver segment, the mechanics lend this segment a more coherent explanation of what it might represent than the instantial assets do. The codes of the simulative register involve taking the operation of the mechanics as particular representations, and beliefs about instantial assets are less essential than, or ornamental to, the processes. Unlike the cupcake example, but demonstrated by the "jobs dividing the United States" gameplay segment, which simultaneously employed both visual and the procedural style metaphors, the simulative register can offer potent representations because the visuals *and* the processes can be understood as iconic signs.

These registers are not exclusive categories and operate simultaneously and to different extents for any particular interpretation and segment of gameplay. However, each register demonstrates how types of observations can have varying degrees of significance when interpreting different types of games. Differentiating these two registers helps us pinpoint exactly how a game like *Kaboom!* can be about subjects differently (i.e. rely on different semiotic codes) than a game like *SimCity*. With *Kaboom!* we employ the instantial register, and with *SimCity* we employ the simulative register.

Of course, the hypothetical examples above offer fringe cases of each of these registers for the purpose of precisely differentiating them. Most games that operate in the instantial register are more sensible than the screwdriver and the screw, and most simulation games are more sensible than the cupcake and the manhole cover. These simple examples can hardly be said to be simulations at all. In fact, the cupcake and the manhole cover might be thought of as the "0^{th} order" simulation game as the process that is being narrated occurs directly in front of the player. Most simulation games have systems of rules that happen outside of the player's view (e.g. the precise effect of pollution in *SimCity* is not directly evident to the player). The cupcake and manhole cover example shows that the line between simulative and instantial representation is blurry.

The following will use these insights about simulation representation to build up to a theory that accounts for how simulations with more complex rules can be said to be about subjects.

Interpreting Simulation Games

As discussed above, regarding a videogame using the simulative register will involve paying attention to the operations of its processes and narrating, or telling stories about how they work. However, most simulation games are very complex and the operation of their processes cannot be described simply. For example, depending how someone plays a game like *SimCity,* one might walk away believing that the game was advocating for or against mass public transit if it played or didn't play a significant role in the city's success. Simulation games are able to produce a vast space of possible outcomes, and describing the game as a representation of just one possibility would be to neglect a great deal of what the game can represent. Frasca points out that "…for an external observer, the outcome of a simulation is a narration. But the simulation itself is something bigger than narrative" (Frasca 2001). However, as of now, the analytical tools for discussing a simulation as opposed to particular narratives are underdeveloped. One of the many problems is not clear at what level of abstraction an interpreter should attempt to describe what the game is about. *How do you talk about what a space of possibilities is about?*

One way to better understand how the simulation operates would be to look to the code itself. While this would be the best way to ensure that statements about a game's processes are accurate, accuracy is not the goal in humanistic interpretation but rather to understand a player's experience. Furthermore, the running code of a videogame creates a dynamic environment where a space of possible operations is possible. By looking at code, an interpreter is not able to foresee all possible operations in that space. As an extreme example, much of the code base might be devoted to an aspect of the game that the player never sees as they never create the conditions such that the code executes. In addition, code describes a level of detail that can be argued to be *inaccessible* to players. Much like how we do not describe what we do in everyday life in terms of chemical reactions and physics, it doesn't necessarily make sense to describe gameplay in terms of code.

To comprehensively speak about simulations games, we must have a language that accounts for the experience of the player, while not omitting from consideration that the experience could have been different had other choices been made. We need a framework that can describe a game as a space of possibilities.

Instances and Principles

Borrowing from philosophy, and emphasizing the concrete nature of a game's processes, the concept of *principles* can describe what shapes a game's theory (or procedural rhetoric). Principles are the player's perception of the general truths that serve as the foundations for how a videogame generates representations of how things function. A principle of a videogame involves both its computational structure and commitments about what those processes are about. What a game is generally about is mutually determined by both a game's instantial assets and the narrative affordances of its processes as illustrated in the analyses above.

Discovering principles of a videogame necessitates the consideration and interpretation of different aspects of the artifact than when considering static media, and the previously discussed simple games of the instantial register. Where most other media can be understood as single static narratives, videogames *generate* many narratives, or representations (Frasca 2001). Understanding a game's principles does not only involve understanding what happened in a play session but also what *could* have happened. Discovering a game's principles can be difficult as players do not have direct access to the structure of the simulation. Access is limited to the individual generated narratives, or *instances*, that the simulation affords. An instance is defined as a static gameplay experience, or in Frasca's language, a particular narrative about gameplay. Interpreters construct principles through the interpretive generalization of instances.

Crawford argues that “the best measure of the success of a game is that the player learns the principles behind the game” (Crawford 1986). Elsewhere, he describes a framework for how interactive systems operate and players earn this understanding: the listen/think/speak loop (Crawford 2003). A system *listens* to player input, *thinks* about how this input changes its state, and then *speaks* about the new state to the player. Players also enact a listen/think/speak loop when engaging with interactive systems. A player observes the system’s output (listens), considers the space of possible actions (thinks), and then chooses and performs some action (speaks). Note how a principle is not solely the system’s *think* part of the loop. A principle involves commitments about what is being represented (which is not present in abstract code) and its recognition is influenced by an individual’s personal history and cultural context.

An instance encompasses an interpretation resulting from one pass through this loop that begins with the system’s expression of its state and ends with the player’s consideration of the new state. Many instances can be in process at any point in time, and it is not the case that every pass through the loop comprises an instance. Only those cycles of interaction that are found to be significant are interpreted and considered. In other words, an instance is a moment of reflection about a segment of gameplay. By iteratively considering the relationship between the player’s action, motivations, and how the system responded to the action, the player begins to construct interpretations of the videogame's principles. An instance can either support or distract from a principle. Wardrip-Fruin's concept of the *SimCity Effect* relates: "Successful play requires understanding how initial expectation differs from system operation, incrementally building a model of the system's internal processes based on experimentation" (Wardrip-Fruin 2009). Of course, an instance that distracts an interpretive hypothesis doesn't necessarily invalidate it. With humanistic interpretation, an interpreter may or may not be wholly consistent.

Conclusions

Given a set of principles, the interpreter will make *conclusions* about what a game is representing. In this context, a conclusion is a generalization about what the system tends to represent. For example, after playing *September 12th*, a game in which players target a Middle Eastern city with missiles intended to kill terrorists, for some time the city will most likely be destroyed and filled with a much higher number of terrorists and a much higher terrorist to civilian ratio. An example of a principle of *September 12th* is that terrorists are born from civilian deaths and a conclusion could be that *September 12th* represents a critique of the bombing of foreign nations and implies a simple solution - to stop.

If an interpreter is dissatisfied with a conclusion, his critique may take to task a principle that led to it. For example, one might argue that the terrorist to civilian ratio in *September 12th* does not accurately represent how actual people become terrorists. This is an example of the simulation gap that exists between player's beliefs about the real world and what they perceive the game as representing.

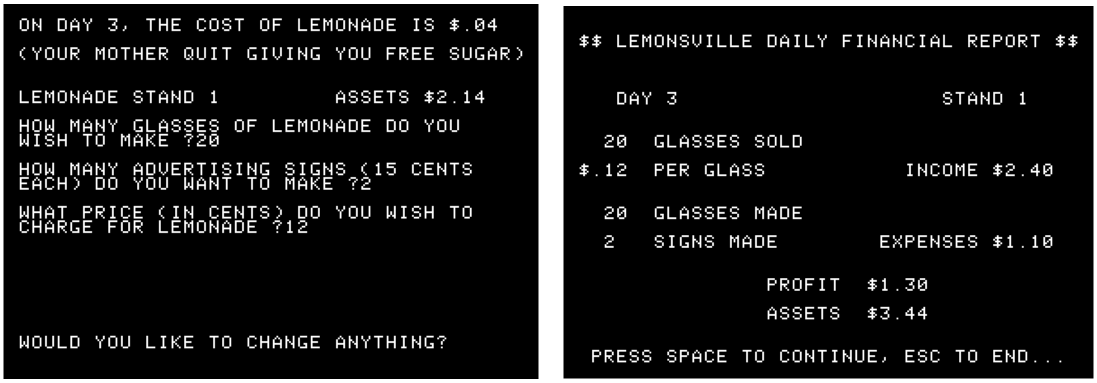

Figure 13 - In *Lemonade Stand*, players choose how much lemonade to make, how much advertising to buy and how much to charge (left) and then see how the day's sales went.

In summary, the simulative register regards the ways that a game's operation can be narrated as salient material for interpretation. Through experimentation (experimenting gameplay instances), the player constructs generalities about how the system operates, and these are called the principles of the game. These principles become the vehicles of metaphors that drive the interpretation, and ultimately provide the material that constitutes an interpretation. Finally, conclusions about what the system tends to output/represent can be asserted.

Interpreting Lemonade Stand

Using the above insight that simulation games are best understood by narrating their processes and regarding these as metaphorically representational, and the framework of understanding a game in terms of its principles, this section will analyze the classic computer game *Lemonade Stand* (1979).

Gameplay in *Lemonade Stand* involves running a small business while trying to maximize profits. In the early versions of the game, as discussed here, given a weather forecast, the player chooses how many glasses of lemonade to make, how much to charge for the beverage and how many signs to buy for advertising. After these choices are made, the player is presented with a report of how many sales were made and how this affected the stand's financial standing. After this, the player is given the next day's weather forecast, and chooses again how many lemons and signs to buy and how much to charge for each glass of lemonade (Figure 13).

In every simulated day, players of *Lemonade Stand* are confronted with choices about how much of their assets to invest into that day and in what way. Each day can be considered an instance of the simulation. For example, on a sunny day, the player might choose to charge a very low price, with no signs for advertising and all of the glasses might sell. Note how *Lemonade Stand* provides a particularly clean example of instances where other games, such as *SimCity*, do not have such clearly delineated instances.

With *Lemonade Stand*, a day, or instance, where the weather was cloudy, given the same price and number of glasses, none of the glasses would sell. Based solely on these two instances, the player could interpret a principle of *Lemonade Stand* through a generalization of these observations: *ceteris paribus*, sunny weather encourages and cloudy weather discourages the sale of lemonade. In terms of the simulative register, this belief stands on a metaphor that people and lemonade, the instantial tenors, operate according to the processes that would have people buy more lemonade when the weather is sunny and less when it is cloudy, the procedural vehicle.

Note how as described, this principle is too general to describe *how much* more or less people want to purchase lemonade based on the weather. For instance, where on a sunny day a whole stock of ten glasses of lemonade might sell and on a cloudy day only three would, given a different stock size of twenty glasses, twelve might still on the sunny day, where four might sell on the cloudy day. The exact relationship is hidden in the code, to which the player does not have direct access. To reiterate an earlier point, the reason for speaking about the representations of a simulation game at the level of generalized principles, as opposed to particular descriptions, or instances, is that the relationship between weather and how much lemonade might sell is dynamic. An assumption of this interpretation framework is to speak about the system as a whole as much as possible.

Lemonade Stand is an interesting example as most of the simulation's variables appear to be directly visible to the player: the number of glasses made, the price, the number of advertising signs made and the weather. *Ceteris paribus* claims, like that about the influence of weather, can be made about each of these variables after performing simple experiments. For example, through comparing two instances where the player charged a high price for the beverage and another where a low price was charged, it will be found that more glasses of lemonade will tend to be sold as the price decreases.

While these sorts of claims help understand specifically what the simulation is claiming about people and how they purchase lemonade, simulation games can also be claimed to represent more general principles. For instance, *Lemonade Stand* involves constantly trying to maximize profits and minimize costs. The player doesn't want to make more glasses of lemonade than will sell, and doesn't want to buy more advertising signs than are necessary to attract customers. Furthermore, the player wants to sell all the glasses made, but wants to do so at the highest price possible. This relationship describes almost exactly a core introductory concept in economics: the laws of supply and demand. Thus, the game can be said to be representations of those ideas. For this reason, *Lemonade Stand* is commonly used in classrooms to introduce economics. Whatever the exact processes that compute the results of the day, adopting the general principles of supply and demand as the way to understand, or narrate, the simulation will consistently match the output of the simulation.

By identifying these principles of the simulation, other concepts from economics become relevant to the discussion of what the game represents. It can be said that what the player is constantly trying to do is achieve *equilibrium* – the state where both supply and demand are equal. Demand in *Lemonade Stand* is influenced by the weather, price and amount of advertising and supply is specified by the player in terms of the number of glasses of lemonade that are made. The player must try to avoid excess supply – where price is set too high and not all glasses of lemonade are sold – and excess demand – where price is set too low and all glasses are sold. Because demand is a function of three variables, two of which the player is in control of, another task is to determine the most efficient way to create demand. More specifically, the player needs to determine whether advertising or lower prices will attract more customers in various weather conditions. In this way, *Lemonade Stand* is a game about learning to understand the principles of supply and demand. These principles exist through the narration of the processes that *Lemonade Stand* creates through the structure of its simulation.

After identifying the economic principles at play in Lemonade Stand, an interpreter can draw various conclusions based on whether the representations match preexisting ideas about the subject matter, or simply to draw conclusions about what situations the generative principles will tend to create. For example, one could claim that *Lemonade Stand*, by reducing customer's willingness to purchase lemonade to weather, price and advertising, oversimplifies how market choices are made. Specifically, one might argue that reputation plays a key role that is not addressed in the simulation (later versions of *Lemonade Stand* do incorporate this dimension). One might also conclude, after being unsuccessful at creating profit, that *Lemonade Stand* is a cautionary message to avoid the stress of entrepreneurship in a cut throat world. On the flip side, one might critique *Lemonade Stand* for neglecting the subtle political and socio-cultural factors that influence the success or failure of a business, and claim that *Lemonade Stand* offers an overly optimistic view of how one might become successful ("Just make a quality product and you will be successful!"). The conclusions that can be drawn from a simulation game are many and varied.

Summary

Understanding what a simulation game represents involves understanding how the game's processes function. Different than other media, interpreting a simulation game requires regarding the game's processes as first class representational entities.

Through metaphor, the processes act as vehicles from which attributes, or narrations of the processes, are ascribed to the subject matter (typically represented by instantial assets).

Because a simulation generates many narrations, or instances, interpretive generalizations about the processes are required. These generalizations can be described as the game's principles. The principles of a simulation can take varying forms of generality, and can range from very specific descriptions of how the processes function, to general tendencies that arise from the specifics. However, principles cannot describe exactly how a simulation operates, as the code of the system is never revealed to the player through gameplay. Based on principles, an interpreter can make evaluations, or draw conclusions about what the principles tend to represent.

Using this theory of simulation representation, we can now return to Squire's claim that *Civilization III* represents world history as "an emergent process arising from overlapping, interrelated factors" (Squire 2004). Rather than considering just the Reddit user's dark world plagued by endless war as the natural outcome of the game, we now treat that as one possible outcome of many. Given that most gameplay sessions end quite differently, we can treat each outcome as an instance from which we can make generalizations about the representation formed by the simulation as a whole. Squire's claim is easy to justify by simply observing that it would be difficult to point to exactly one factor that would create divergent outcomes. It is complicated. More specific claims could be made about what *Civilization's* simulation can represent to a player, and it is exactly these interpretations, negotiated between a student and game, that Squire's research demonstrates as powerful for education.

Like the theory of micro-rhetorics, this discussion of simulation representation is meant to set the tone for an interpretive point of view. The claims made about *Lemonade Stand* and *Civilization III* were intended to demonstrate how this point of view can generate understandings of a game that are grounded in a solid foundation. The following section will demonstrate how this point of view can be used to create a game that is interpretable as carrying specific themes.

***Prom Week*: A Game about Social Interaction**

This section describes *Prom Week* - a simulation game about the social lives of a group of high school students. *Prom Week* represents an advance in the state of the art of both social simulation and interactive narrative, but this section will focus on the process of creating *Prom Week* as a representation of a theory of social interaction. As discussed above, being able to represent this theory through simulation involves crafting the game such that the processes can be narrated in terms of principles that support the intended representation. This section will begin by giving a high level description of how *Prom Week's* simulation technically operates as well as a discussion of how the game is structured to support a believable social world for the characters to act in. Following this will be a discussion of the design process in which the interaction, interface and game design were iterated upon. Finally, the latest version of the game will be evaluated and critiqued in terms of how well the gameplay represents a set of principles according to the previously presented simulation theory.

The Game

Gameplay in *Prom Week* revolves around the social lives of eighteen characters. In any given "Story," or campaign, the player is given a set of goals to complete during the week before the prom. For example, in Zack's Story, one goal is to get him a date for the prom. Goals can be satisfied through an open-ended set of solutions discovered through interaction with the characters and social state. For example, the player could have Zack form a friendship with a popular character over a shared interest, or exploit another character's "competitive" trait to make an enemy when Zack flirts with someone the competitive character has a crush on.

The player works toward goals by choosing *social exchanges* for each character to initiate (Mccoy, Mateas, and Wardrip-fruin 2009). Social exchanges are multi-character social interactions that modify the social state connected to the participants. Which social exchanges are available and how each changes the social state is managed by the game's AI system, *Comme il Faut* (*CiF*) (McCoy et al. 2011). The player chooses from the top social exchanges that each character desires to play with each other character. *CiF* provides this ordered list based on its character models and the current social state.

In addition to determining what exchanges characters want to perform with each other, the system also determines whether a responding character will *accept* or *reject* a proposed social change, and selects a scene to best perform that decision from a large library of alternatives. Figure 14 shows an excerpt from a social exchange where Zack asks Monica on a date and Monica rejects him because he isn't popular. Each factor in this scene (Zack's desire to ask Monica out, her decision to reject him, and her reasons for doing so) are all part of the underlying social simulation rather than pre-decided or static story content.

While goals usually pertain to specific characters, players take on the role of an external observer and manipulator who can select a social action for any character to initiate. For example, to remedy the situation in Figure 14, the player might try to make Zack popular, by getting him more friends, performing actions categorized as *cool*, etc. Or the player can make it so Monica is no longer popular, by having her do *embarrassing* things, cut ties with her popular friends, etc.

Because the gameplay of *Prom Week* involves manipulating the social space, which is often what stories about high school are about, the *gameplay is the story*. Every action the player takes advances the game's narrative and sends ripples

Figure 14 An excerpt from a social exchange where Zack tries to ask out someone out who is out of his league. Her rejection reflects her cold and honest personality.

throughout the internal social state, which in turn affects which actions are available in subsequent turns. The system is then a partner of the player, giving the narrative meaning and shape. This is in contrast to a sandbox game in which gameplay may be the story, but the story is represented only in the mind of the player. While *CiF*-enabled stories are authored in the sense that the designers create the initial situation, define the goals for each scenario, and create a pool of templated scenes for characters to perform, since *CiF* enables emergent solutions to each social puzzle the resulting story space is also highly dynamic and responsive to player actions.

Stories

Prom Week provided unique opportunities to innovate in the design of emergent story-based puzzles. The complicated social puzzles could easily have conflicted with the goal of telling coherent and satisfying stories. The structure of the game's final levels and goals was designed to address these potential conflicts.

A player of *Prom Week* begins by selecting a *story*. A story is a collection of levels, each representing a specific time and place in the week before the prom, where the player can take social actions involving a particular subset of the characters in the story. In addition to getting Zack a date, some other example goals include ending Zack's war against a popular bully, or specifically getting Zack into a relationship with someone "popular." Goals in a story are sometimes designed to be complementary, as in this example where patching up Zack's war with a popular bully will likely improve his relations in general with the popular crowd. As mentioned above, objectives can be met in a variety of ways; the player could forge a friendship between Zack and the bully, or perhaps make the bully lose his social standing, which might change his antagonism towards Zack.

Every story's last level takes place at the prom. After the player runs out of turns, or decides to skip to the end of the night, a customized ending is presented that reflects the combination of goals achieved. For example, Zack's story might happily end with him becoming the prom king if the player was able to get him to date a popular person. Or, if in order to make this happen the player had him abandon his unpopular friends (to appeal to the popular crowd), the player might get a bittersweet ending where he still becomes the prom king, but also is confronted by his old friends. Every story has many possible endings for various combinations of goals the player might have completed. As the player finds more endings, additional stories are unlocked. In addition to the explicit rewards of endings and new stories, players are free to define their own criteria for play and success, such as creating particularly awkward or humorous situations, recreating events from their own lives, or trying to solve each level's social puzzles in as few moves as possible.

Prom Week's Simulation

Prom Week allows players to solve goals flexibly, while maintaining consistent and believable characters. *CiF* enables a style of gameplay we call *social physics*. While video games have achieved a high level of playability in physical spaces, with activities like combat, movement, and physics-based environmental manipulation all well-explored, *Prom Week* set out to make social spaces as playable as physical spaces currently are. The goal was not to recreate the everyday social world, but to create social dynamics specifically crafted for a targeted experience — just as platforming games don't reproduce the physics of the everyday world, but rather an enjoyable simplification tuned for gameplay, and fiction writers portray behavior and dialogue in stylized fashions that differ markedly from the average conversation.

Without a system like *CiF*, representing social interactions between any two characters in our story that takes into account cultural context, personal history, and current relationships would be impractical, or perhaps impossible. The space of contexts (states of the virtual world) and social interactions (player interactions) is prohibitively large and not amenable to brute-force authoring. *CiF* provides knowledge representation and processes that model social interactions to make this ambitious goal tractable to implement.

Prom Week social physics is based on a set of over 5,000 sociocultural considerations. These considerations are the rules that influence the characters' desires, each adding either a positive or negative numerical weight to the desirability of each potential social exchange. One example rule is: a character who is *vengeful* (a static character trait) will be more likely to do something mean to someone who has recently done something mean to them. A more complex example: a character might be more likely to do something romantic with someone who was recently mean to the person who was mean to them ("the knight in shining armor"). These rules encode a notion of "social common sense" which is what the player will reason over while striving to satisfy each level's goal.

The following list shows the social dimensions of the game world:

Relationships: binary, reciprocal and public connections between characters. The three relationships in *Prom Week* are: friends, dating and enemies.

Social Networks: scalar, non-reciprocal and private feelings from one character toward another. The three networks are: buddy, romance and cool.

Statuses: temporary feelings, either unitary or directional, that are often consequences of social interactions. Some statuses, such as *embarrassed*, are internal feelings. Other statuses and represent social standing, for example, being *popular*.

Traits: permanent attributes of a character's personality. Most traits are private, such as being *competitive*, while others are public knowledge, such as being a *sex magnet*.

Social Fact Database: the social history of interactions between characters. All entries in the social fact database are public knowledge and thus comprise the characters' collective social history.

Cultural Knowledge Base: the objects of the social world, a zeitgeist of popular opinion about each object, and each character's personal relationship to that object, which can be *likes*, *dislikes*, *wants*, or *has*. For example, Zack may *like* and *want* a scientific calculator even though they are generally considered *lame*.

The following example illustrates how the structures described above constitute a social state.

Simon is a character with the traits *helpful* and *witty*. Naomi is a character with the trait *attractive*. Simon has the status of *has a crush on* Naomi, and Naomi has the status of *popular*. Naomi and Simon have the relationship of being *friends*. Simon has a high *romance network* value toward Naomi but she has a very low *romance network* value towards him. Naomi also has a low *cool network* value toward Simon. All other network values are neutral. The cultural knowledge base states that both Simon and Naomi like scientific calculators, which are generally labeled as *lame*, and footballs, which are considered *cool*. In the social fact database is a past action Simon took towards Naomi marked as *embarrassing*, labeled as ”Simon misunderstood Naomi asking for help on homework as a romantic advance.”

Given a social state, *CiF* operates by looping through a set of processes to determine what characters are interested in doing, and how they might respond to the other characters taking these social actions with them. The first process is desire formation. This process determines a character’s volition (or will) to play a social exchange with other characters. Every time desire formation is executed, every character determines their volition to play every social exchange with every other character. Volition is scored by accumulating the numerical values of all the individual rules that evaluated to true between each character. After this process, all characters in the cast have a volition value for every social exchange with regards to every other character.

Next, the player selects a social exchange for one character to perform with a second. Social exchanges have an initiator intent (the initiating character’s desired social change, such as to start dating) and three roles: an initiator, a responder, and a possible third party. When the player selects a social exchange, basic information about how the initiator and responder relate to one another is displayed. If a third party is involved, *CiF* selects the character for whom the most influence rules pertaining to a third party were true. For instance, in the *Spread Rumors* social exchange, a third party who both characters have a low buddy network value toward will be selected.

Once an exchange is chosen, *CiF* determines how the responder reacts based on the social context. This process is very similar to scoring volition for initiators: a sum is calculated for true social considerations rules that pertain to responding to the social exchange. If the sum is zero or greater, the game responder *accepts* the intent of the game. Otherwise it is *rejected*. The concept of accept and reject refer to whether the exchange will go according to the initiator's intent or not. For example, if an initiating character intends to make the responding character like him more, and the exchange is accepted, the responder will in fact like the initiator more. If the responder rejected the attempt, this would not happen.

While each social exchange has a primary result for success (such as changing the dating relationship to true for an accepted Ask Out between two characters), the system includes a large number of scenes that comprise different ways an exchange can play out based on the social state of the participants and whether the exchange was accepted or rejected. These are called *effects*. For example, if a character plays *Share Interest* with another character, and the exchange is accepted, there could be an effect specific to situations in which the two characters both like a *cool* object in the cultural knowledge base, or another in which they bond over a *lame* object, celebrating their deviation from the will of the zeitgeist.

Each effect is associated with a performance realization instantiation. An instantiation is a set of template-based dialogue acts and associated animations. After the instantiation is realized, the social state change associated with the chosen effect is applied. This includes placing an entry into the social facts database to account for the exchange, to be referenced and considered in all future social exchanges.

The last step is running a set of "trigger rules" over the new social state. Trigger rules account for social changes that result from multiple social exchanges and other elements of the social state. For example, a character will receive the status of "cheating" after starting a dating relationship with one character when they are already dating someone else.

The Evolution of *Prom Week* and AI-Based Design

The impetus for creating *Prom Week* was to create a compelling game experience around the social AI system *CiF*, and the entire game was designed with this system in mind. This methodology, called AI-based game design (Eladhari and Mateas 2008) or expressive AI (Mateas 2001), fundamentally changes the concerns of typical design: instead of thinking of design choices and game mechanics in terms of what existing conventional systems can do, the primary criteria for design becomes creating a game that best leverages the power of a novel system. In this case, *CiF* is

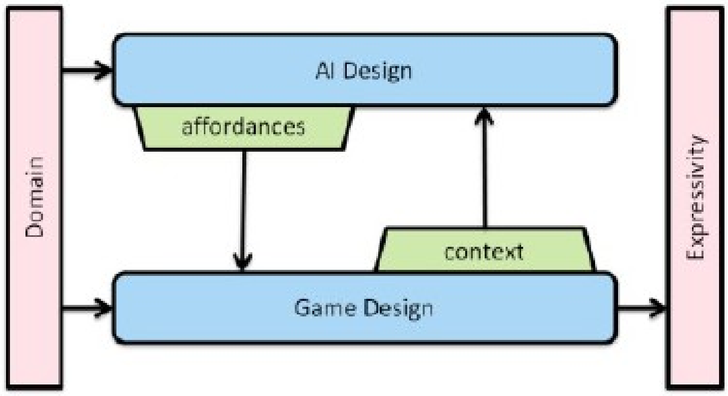

Figure 15 The AI-based game design process. Creating new AI systems, such as CiF, provide new affordances in the space of Game Design, while implementing AI in a game, such as *Prom Week*, offers new context for expansion on the AI itself.

the AI system around which the design was centered, so the changing social situations of virtual characters brought about through game play were the primary concern.

As AI-based game design is distinctly different from other game design methodologies, it has the potential to create new types of video games. The space of all possible video game designs is considerably larger than the fraction which has been explored to date. A research-centered approach has the potential to lead to unexplored design spaces. AI-based design raises the priority of technological innovation to the same level of the game design itself. In other words, with new technological abilities, new types of games can be imagined.

A benefit of AI-based game design is that the processes of designing of game, authoring content for it, and refining the AI system each inform one another (Figure 15) (G. Smith et al. 2012). The act of designing game mechanics to be used in conjunction with an AI system tests the system. By exploring and determining the affordances the AI provides (or fails to provide) for gameplay, the designer exposes the weaknesses and strengths of the AI in modeling its domain, which can be used to further improve it. As the AI continues to evolve, it in turn suggests different game design possibilities. This cycle of iterative refinement of both AI system and game improves the design and functionality of both systems: the AI becomes better at modeling its domain while the game becomes both a better gameplay experience and better at providing play in its domain. This process of creating a fully-playable game based on an AI system is potentially very beneficial to developers of AI in areas such as story generation, natural language generation, and social or psychological modeling. Creating a full game with this methodology allows for many more cycles of iteration and refinement on the underlying systems, enabling a richer final product than a system developed in isolation or with only a system demo as a demonstrator.

Furthermore, basing the design of a game on an AI system can be said to specify what the game is *about*. Simon and Newell wrote “Programs can be regarded as theories, in a completely literal sense, of the corresponding human processes” (Simon and Newell 1962). According to the theory of simulation representation, this theory then governs the processes that the simulation affords, and thus the ways that interpreter’s can narrate the game’s operation.

With this in mind, the following will be a discussion of how the AI architecture and game design of *Prom Week* evolved over its two and a half year development toward being able to represent a theory of social interaction.

Figure 16 The computationally assisted paper prototype for *Prom Week*.

Assisted Paper Prototype of Prom Week 1.0

Prom Week was first implemented as a paper prototype with a computational assistant (Figure 16). This goal of this version of *Prom Week* was to represent and reason over compelling social situations along with the variations of the resultant behavior that arise from different personalities being placed in similar roles. The prototype had the player choose to side with one of two high school factions (Goths or Emos) and help that faction win the favor of the student in charge of the audio equipment at the prom, Milton. Characters with personality descriptions taken from Reiss' motivational analysis (Reiss) were present as stand-up models and character sheets. The player was dealt a hand of cards (each listed with basic needs effects) and was able to play them on the characters. After cards were played on characters, the game master would enter the cards' effects in the computational assistant, which would then determine which social exchanges would be initiated by the characters. If the social exchange resulted in behavior in line with Milton's personality, a token would go to the faction of the character who initiated the social exchange. After 10 rounds of game play, the faction with the most tokens would gain Milton's favor and control of the playlist for the evening.

As this prototype was the first incarnation of a playable form of *CiF*, the design space was highly malleable and resulted in sweeping changes to our preconceptions of what a game in the space of social play could be. Through the development process and playtesting, we discovered that social exchanges solely driven by psychological needs were unintuitive and hard to communicate or justify to players. Particularly, the abstracted social exchanges performed by the characters did not match the exchanges that were anticipated by the play testers given the characters' basic needs. For example, having a need to be embarrassed resulting in a desire to engage in a flirt action was unintuitive. Motivated by this, the next iteration of *Prom Week* shifted its focus to the logic of social statuses and relationships between characters.

Realizing that creating an AI system to be used as the core of a video game requires a different frame of thinking than implementing a model "correctly" was an important step in our process. Our direct, straightforward implementations of complex topics, such as our basic needs modeling from motivational analysis, did not capture the depth of social play we were hoping to capture; a few vectors of scalar values with a small amount of conditional logic did not provide a compelling game experience that players wanted to understand. It also proved to neglect the aspects of social state that players tended to reason over, the social context. Instead of exposing the engineering choices as game mechanics, we decided to base the affordances given to the player on what our players were thinking when they played the game.

Promacolypse Demo

The version of *Prom Week* presented at *Game Developers' Conference* 2010 (Lowensohn 2010), titled *Promacolypse* (Figure 17), comprises *Prom Week's* second iteration. The version of *CiF* used in *Promacolypse* was a redesign focused much more on the social space around the entire cast of characters and not focused on individual characters and their psychological needs. We also abandoned the idea of *antithetical* ways to play social exchanges stemming from Berne's transactional analysis (Berne 1964). These antithetical ways of playing social games flipped the intent of the social exchange on its head which resulted in unpredictable agent behavior and ultimately confused players. For example, a character asking another out on a date could turn into the character just telling a joke, with no visible explanation for why the character took that action.

This demo was completely computational and consisted of many of the same processes and data structures described in the preceding sections: social networks, statuses, CKB, SFDB, and triggers were all added to *CiF* to support the new design decisions as well as facilitate making the previously-paper parts of the game computational.

With such large changes made to the AI system, many new options of game design presented themselves. While the game was still character-based, the goal of the game became to reach certain social states through making the characters play social exchanges with one another (as opposed to the previous paradigm of playing cards consistent with characters' basic psychological needs). With the exclusion of antithetical social exchange outcomes, characters needed a way to respond to the intent of an exchange; if the initiator started a flirtatious exchange with someone who had low amount of romance with them, the system needed a way to factor the responder's social situation into the outcome of the exchange. To achieve this, antithetical social game outcomes were replaced with accept/reject logic that is deeply tied to the social state existing among the characters.

While authoring content for the *Promacolypse* demo, consisting of social exchanges and their instantiations, we found that we were building a lot of common sense about social behavior into the rules for each social exchange. This repetition of rule writing revealed the need for constructing a mechanism of general social reasoning that would encompass the concerns of many social exchanges. To address this, we developed the structures we call *microtheories* to capture the social knowledge of how to act within the context of a particular social framework (such as a friendship, or towards someone you think is cool).

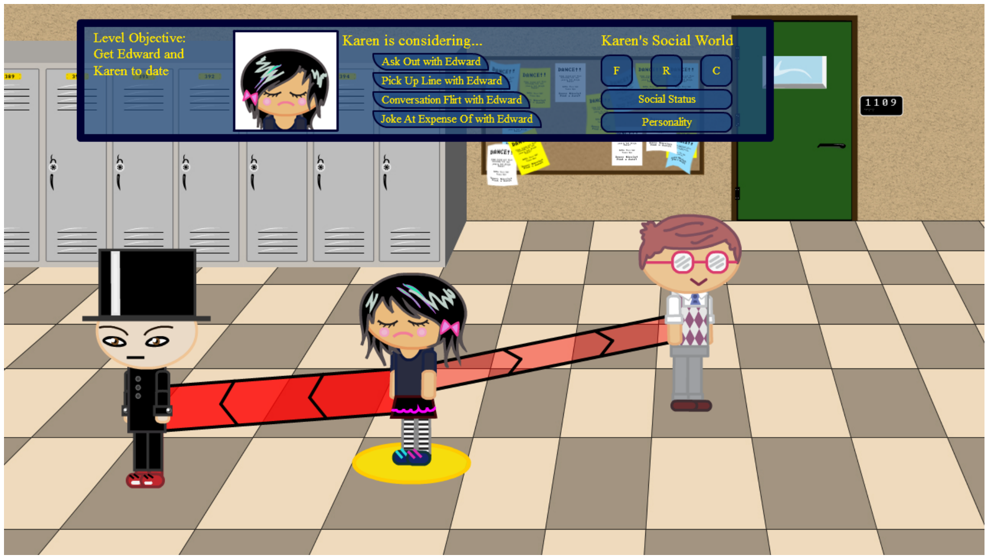

Figure 17 A screenshot of *Promacolypse* demo that demonstrated the early "social context" based version of *Prom Week*

With social exchanges each having an intent, a large set of microtheories with rules to influence a character's desire to perform that exchange, and the ability of characters to accept or reject a proposed exchange, social exchange authoring could focus on what makes a particular exchange a unique act within the system. For example, the exchanges *Share Interest* and *Reminisce* both have the intent to raise another character's buddy network value toward the initiator. In general, if two characters are friends, the *friend* microtheory will increase a character's desire to play exchanges with this friendly intent. However, if the two characters share an interest in an item from the cultural knowledge base, for example, we can write rules making the initiating character more likely to play *Share Interest*. Likewise, if the two characters have a positive history of social interaction, the initiator might instead want to play *Reminisce*. Such exchange-specific rules are now the only ones embedded in each social game, leaving general rules to the domain of the microtheories. These changes dramatically reduced authoring time.

Beta Version

Playtesting the *Promacolypse* demo brought to light several necessary improvements to the game's design and AI system based on data from players. The first was that the game needed more narrative structure. This led to the addition of the story progression, level, and ending structure described above. Another change came from players' frequent desires to solve problems using a third character in addition to the initiator and responder. To support this, the beta version added third party social exchanges that can be initiated by the player such as s*pread rumors.*

The players were often confused by the outcomes of social exchanges played in the demo. They asked questions like "why did that happen?," "why did the initiator want to do that to that person?," or "why did the responder act that way?" We needed to expose the reasoning done by *CiF* in a way that added to the game experience. We decided to present this information in an abstracted form, and erred on the side of providing too much detail, giving the player the ability to dig into the interface to learn the details of what was happening within *CiF* (Figure 18).

Figure 18 A screenshot that shows the beta version of *Prom Week*. The yellow bar on the left shows the social influence points.

As refinement and playtesting continued, another concern became evident: the game was too hard. With such a complex simulation, the results of any given social exchange, while believable, were often unpredictable. For example, while the interface might have indicated that two characters liked each other, an attempt to make them become friends with the *Make Plans* exchange might fail, perhaps because of the responder's trait of *shy,* or a long-ago event in the social facts database where a friend of the initiator's did something mean to the responder. While these cases demonstrate exactly the sort of complex social intelligence we wanted to give to characters, they were not always apparent (or fun) for players. Because of this, we introduced a new game play mechanic called *social influence points* (SIP). SIP allows players to know more about and change how characters will respond to a social exchange before it is played. SIP is a limited resource that is increased when an unmodified social exchange is used, and decreased when the player either reveals if a character will accept or reject a social exchange, changes a reject into an accept or vice versa, reveals all of the motives for why a character will respond, or forces an initiating character to select a social exchange that is not one of his top five priorities. With SIP, players can complete goals much more easily because they can carefully choose which social exchanges really must succeed to make progress towards a particular goal, and players can "nudge" the fictional world in directions they find more interesting without turning the characters into puppets. Making SIP a limited resource ensured that the majority of player choices were still governed primarily by *CiF's* simulation.

The primary problem with this version is that players were uninterested in understanding why characters did what they did. SIP and level redesign attempted to make the game easier, but only helped retain the player's interest and didn't help make the operation of *CiF* more apparent to players. In other words, the principles that governed the processes were not apparent to players.

Final Version

The most common feedback from the beta was that the game was still too hard and that the interface was too complicated. To address the first problem, we had to reevaluate the sort of goals we were asking players to complete. The level of detail of the simulation, together with the difficulty of clearly communicating the many parts of the system and their effect on characters' actions, was still making it hard for players to reliably get the characters into a desired social state (even with SIP). For example, some goals were stated in the form of information easily visible to the player (such as making two characters enemies), while others were based on information not exposed by the interface (such as becoming embarrassed, or acquiring a history of a certain type of behavior). To fix this, we modified all goals to match closely to the most obvious social exchange intent types, so the player could more easily see the relevant parts of the social state and determine what actions to take to change it.

To address the second concern, the final version of *Prom Week* had a completely redesigned interface that used interface metaphors to fictionalize the social state data. Rather than presenting most social state information in menus or abstract information bars, these details were presented as if they were the thoughts of the characters. We have found that by tapping into concepts that players are familiar with (such as media conventions and their own thinking about their social world) the game play experience feels less technical and thus easier for most to digest.

Below is an interpretation and evaluation of this final version of *Prom Week*.

Evaluating *Prom Week* as Simulation Representation

Using the simulation theory presented at the start of this chapter, the following section will analyze and make arguments for how *CiF* and the the final version of *Prom Week* work together to represent a set of social principles. *Prom Week* was developed in the tradition of what Michael Mateas calls Expressive AI (Mateas 2001). The practice of Expressive AI is guided by a metaphor that considers an artifact's meaning as the result of a negotiation between an audience and an artist with the artifact under consideration acting as a mediator. This practice places the actual functioning of the artifact as secondary to how an audience might be able to understand it. These following discussions will not assume that the interpreter understands the above system description of *Prom Week* and will only refer to observations that could be made by a player (rather than a developer).

Rather than reporting the findings of actual players, I will be discussing hypothetical play instances. Recall that instances are segments of gameplay that a player recognizes as evidence toward concluding that a game represents an operational generalization about a domain (a principle). The goal of this section is to describe *Prom Week* in terms of the instances, principles and conclusions it affords. This analysis is intended to exemplify an approach that can be used for future theoretical and design work. With the simulation representation theory and its application to *Prom Week* in hand, future empirical studies to confirm the analysis can be imagined.

Theme

The setting of *Prom Week* is a place of high drama: high school. Established through background images and the distinct fashion styles of the characters, the game taps into western popular media representations of high schools. This setting affords an expectation of dramatically volatile characters partly because adolescents tend to have emotional mood swings, and partly because popular western media has represented them that way. The setting involving a high school dance, where attendees are culturally expected to partner up into relationships, thematically motivates the socially oriented gameplay. Further establishing that this game is about social reality are iconic representations of common social relationships. These visuals can be found throughout the title screen and in the game interfaces (smiley faces, hearts, etc.).

Characters in *Prom Week* are represented using an abstract cartoon style. It can be argued that the reason for this is to permit players to project expectations about the traits of these characters and how they might behave. For example, the character Buzz is wearing a football jersey and has a buzz haircut. In the setting of a high school, these are iconic tropes that point to him being a jock or bully. Again, these codes are established through representations of high schools in popular culture. These associations are confirmed by the introductory text that accompanies the stories and levels.

Figure 19 A screenshot of *Prom Week* that shows Zack's three story goals.

The visual expectation of high drama is confirmed by the game's goals and level introductions. For instance, when the player starts playing Zack's story, he is presented with text that reads "Zack has his work cut out for him if he wants to be Prom King…" This line confirms the suspicion that Zack is a typical nerd character based on his messy hair, Linux t-shirt and glasses. Each story in *Prom Week* involves attempting to solve a series of goals. Zack's goals involve befriending the "prom royalty" selection committee, confronting the bully Buzz, or dating the similarly messy-haired character Lil. Achieving any one of these goals could easily be the plot of a high school television drama like *Saved by the Bell*. *How might Zack get the* *prom royalty selection committee to give him a chance to be prom king? What's going to happen when Zack confronts the school bully? Will Zack and Lil fall in love and live happily ever after?*

Gameplay and Simulation

While the above can be argued to be represented through *Prom Week* visuals and text alone, the following describes the simulation principles that can be argued to be represented through *Prom Week* gameplay and underlying simulation. To better understand how one might go about understanding *Prom Week's* simulation representation, consider the following instance where a player investigates the social actions that Zack is considering taking with Lil.

Figure 20 Players choose among the top social actions that each character wants to take with each other.

A common interaction pattern in the game involves clicking on one character, and then another. Several buttons, phrases and icons appear on the screen encapsulated in thought bubbles. Along with the first person language within each thought bubble, it is easy to conclude that these are representations of what is going on inside the character's head. The simulation carries this metaphor by uniquely populating each character's thought bubbles with social actions that might appear

Figure 21 Several of Zack's reasons for wanting to Ask Out Lil.

appropriate for the character to take. Figure 20 shows that Zack is considering sharing an interest, confiding in, asking out and bragging to Lil.

Now assume that a player is trying to complete the goal of getting Zack a "Smoking Date" which involves making it so Zack and Lil are dating (Figure 19). If the player were to click on the "Ask Out" button, Zack would ask Lil on a date.

Before this action is performed, the player could click a button next to the "Ask Out" button that shows more details about that interaction. Here, the player is presented with considerations about what Zack thinks about that interaction. Zack thinks "Ask Out Lil? Well, I do want to date Lil… a lot." When the "More…" button is clicked, more reasons why Zack is interested in asking out Lil are presented. The simulation has populated this interface element with language that is consistent with the concept of a consideration, allowing the player to interpret the phrases as Zack's thoughts about asking Lil on a date. For any interaction there are many considerations that factor into why a character may want to take a social action. Some reasons can be directly related to what a character thinks about the other. For example, Zack is attracted to Lil. Other considerations have to do with how Zack is feeling, if Zack is desperate for example.

Figure 22 A few of the reasons why Lil doesn't want to date Zack.

If the player clicks on the button labeled "Response" in a thought bubble over Lil's head, the player can see what she thinks about this potential social interaction. Figure 22 exposes Lil's mental dialogue: "Asked out by Zack? Not in million years."

Figure 23 After each social interaction a summary interface appears to give details about what just happened.

Like the interface that allows the player to explore Zack's thoughts, there is a "More…" button that the player can click through to see detailed considerations about why Lil feels the way she does. Figure 22 shows that Lil is influenced by the fact that something bad has happened to her recently. Exploring the inner thoughts of characters often reveals unexpected yet believable considerations. The third consideration in Figure 22 shows an indirect consequence of previous social interactions. Because Zack had previously become enemies with Naomi by bullying her (a social action taken several turns earlier), Lil is less likely to want to date him if he asks her out.

Once the player chooses to have Zack ask out Lil, we see a scene play out where Zack awkwardly asks Lil if she wants to go out and Lil outright rejects him. After the dialogue plays out, a screen appears that reads "Lil totally shot Zack down" and there is a row of icons that indicate upon rolling the mouse over them that "Lil thinks Zack is less cool," "Zack failed to be romantic with Lil" and "Lil was rude to Zack" (Figure 23). Given a list of why Lil reacted in the way she did, we see that she did not directly speak about any of the particular considerations and instead was just generally mean to him. Two of the reasons why Lil rejected Zack didn't even have anything to do with him (her "taking things slowly" and that something negative had happened to her recently). The ways that characters act in *Prom Week* are influenced by many factors.

After seeing several more instances like those above, several principles about how *Prom Week* represents a theory of social interaction can be generalized. Among the first generalizations are that people perform social actions in order to change the social world in some way. This principle is encoded in the game both through the interface and through the system design. Next to each social action a player can choose for one character to initiate with another is an icon which corresponds to how the initiating character is going to approach the interaction. For example, all "Share Interest" buttons contain an upward facing green arrow. In this case, green corresponds to the general feelings of friendliness between characters. Another example is the red frowning face, representing being enemies, next to the "Bully" social action buttons. Across any gameplay instances, it can be observed that these icons are always the same for any given social action. This implies that a person only performs actions such as sharing an interest to raise feelings of friendliness or only bullies to become enemies with someone.

Once a social action is performed, we find that initiating characters do not always get their desired outcome. In fact, across many instances, we find that characters will respond to social actions in surprising and often frustrating ways (assuming the player is trying to achieve some particular outcome). Sometimes, characters will have high opinions of a character (represented by a thought bubble with color coded words that correspond to different types of feelings), and they will respond negatively to a social action that has to do with that opinion. These situations can be explained as characters being multidimensional in their decision making. Two people who are complete enemies, and have many reasons to dislike one another, might bond over a shared interest, or two people who have very high romance toward one another may never choose to date because of a complicated history. As a particular example, even though Lil was rude to Zack when he asked her out, and had a set of negative considerations about him, in the following instance the player could select for Lil to "Confide In" Zack where her reasoning for wanting to perform that action would be based on a set of positive considerations (such as him being honest).

Characters in *Prom Week* are complicated and it is often difficult to predict how they will react. However, when the player looks into the considerations that influence a particular behavior, there will be concrete and reasonable considerations that motivated the action. In the example above, Lil may have considered dating Zack if she had not had something negative happen to her. At the very least, she may have not been rude in how she reject him. This example demonstrates that the social world represented in *Prom Week* is not centered on any particular character. This is very different than the world represented in most games where the NPC's social behavior is primarily directed at the player. For example, villagers in role playing games often exist solely to issue quests to the player, and aren't driven systems that model them as individuals.

Relating the above interpretation to the earlier discussion of the theoretical foundations of simulation representation, gameplay in *Prom Week* affords the above narrations. In playing through many instances, a player can conclude that social reality in *Prom Week* is composed of a set of social actions that they are influenced to take for many obvious, complicated and sometimes contradictory reasons. A responding character is also influenced by a set of complicated considerations.

At a lower level, the simulation in *Prom Week* is composed of characters that at any given point have a list of social actions that they are willing to take. A social action appears as a possibility based on how many social considerations the character has about the other character. When an action is performed, the responding character will either respond in a way that goes along with the way that the social action is labeled or not. This decision is also based on a set of considerations. Exactly how a positive or negative response is selected is not apparent but it is easy to assume that it is based on either how many or how strong the considerations are toward either response. Given a social action and a response, a particular scene of dialogue between characters is selected. This scene of dialogue does not always have to do with the social considerations. The concepts of social actions and considerations, as well as the data structures and processes that govern them are the foundation from which we are able to describe *Prom Week* as representing a social world that functions in this manner.

Given the overall game structure, we arrive at several conclusions about the game. First of all, playing the game is very difficult. Often characters will not do what is expected and reaching goals is hard. This leads the player to conclusions that *Prom Week* represents a social reality that is complicated, hard to understand yet still causal. Furthermore, an individual's social behavior is determined from the accumulation of many minor considerations and social actions have a far reaching unpredictable effect on the feelings and thoughts of others.

Conclusion

The above discussion of *Prom Week* demonstrates how a complex simulation can be understood as representational. Through the interpretive framework that understands the operation of the simulation as vehicles for metaphors about a particular domain, we were able to analyze some aspects of *Prom Week* and make claims about how it represented many themes about social reality and conclusions about how people act. A future empirical study could be imagined that would seek to confirm that actual players can arrive at the principles described. This framework of the simulative register gives researchers a language to discuss and pose hypotheses about what simulation games represent.

The experience of creating *Prom Week* brought to light the true difficulty in trying to be representational through simulation. Getting the game design and interface to allow the simulation to be accessible to players such that it could be narrated and understood as metaphorical was a very difficult process that involved many design iterations. The primary lesson here is that the operation of the simulation, or a model of a domain, is not enough to be able to argue that it is representational. A player must be able to understand how the processes relate to the thematic domain through gameplay.

Different than the micro-rhetoric theory and its implementation in *Game-O-Matic*, this approach does not immediately lend itself to implementation. This approach is more of an interpretive lens that can be used to better support the foundations of how one might understand a game. The following discussion will take a step toward formalizing the interpretive approach described in this section. In addition to summarizing the theory, it brings to the surface several unanswered questions that warrant further theorizing, and could contribute to some of the design difficulties encountered when creating *Prom Week*. Figure 24 shows a diagram of what will be described.

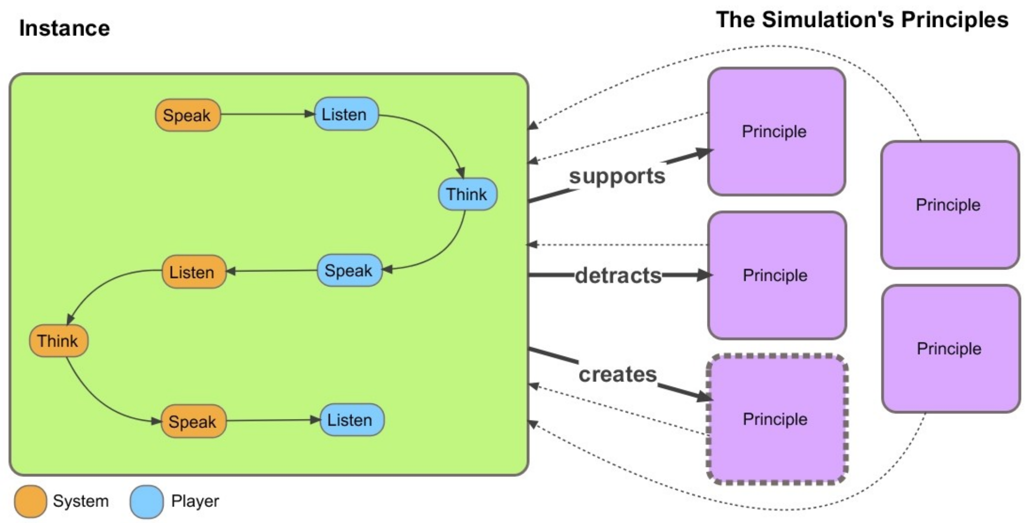

Figure 24 A diagram that summarizes the theory of simulation representation.

An instance is composed of one cycle through the following sequence. The system creates output which the player sees and considers, then selecting an action to perform. The system receives this input, performs computation and creates more output which the player notices. Not every instance will be found to be significant to an interpreter and each step of the sequence may be abstracted to include many actions over a long period of time. For example, in *SimCity*, the player may never create trash collection services, and eventually the city's ground water becomes completely polluted. At this point, the player may notice how the system has reacted to his input of not creating trash collection services. As described, there are innumerable instances at many levels of abstraction that a player may find significant. The theory does not provide any guidance for choosing which instances to consider. Such a task would necessarily involve subjective opinion. Future work could involve a theoretical model of a subject and how it recognizes instances and generalizes them into principles.

Instances are evidence that influence principles in the following three ways: they can create, support or detract. When an instance creates a principle, the principle may be very weakly supported. For example, when ground water pollution lowers a town's population in *SimCity*, a player may take the instance where he did not create trash services as grounds to create a principle that the game is representing the idea that trash services are necessary in a city. If the player were to go on to play the game, this time providing trash services, and the city maintains a healthy population, he may take this as supporting evidence for that principle. However, if the population still declined, this instance would detract from that principle. How much an instance will influence the strength of a principle is not specified by the approach. Furthermore, the mechanism from which instances generate principles is not described.

Once a principle is held, it will influence what observations a player will have, change what instances are observed, what principles will be created and how instances influence them. Exactly how the principles interact with instances is not described by this theory. To address these unresolved issues would also necessarily involve a subject, or a model of one.

This section described a language and approach for understanding how the operation of a game's complex simulation might be found representational. The description and interpretation of *Prom Week* demonstrates how, while not directly visible, the operation of a simulation is an important representational element of a game. The theory does not address areas where a subject is needed and ultimately interpretations done in this fashion will vary greatly between people. Future work could be imagined that would implement this approach in a system that could generate interpretations of simulation games, however many questions are left unaddressed and would require further research.

Like the previous chapter, this chapter is an instance of critical technical practice. The persistent design problems encountered when trying to make *Prom Week* about a particular set of ideas about social reality can be understood as the result of the unstable theory that drove the design. This chapter began by developing the theory of simulation representation where a game's processes and regarded as iconic signs and vehicles for metaphors. It was proposed that players generalize gameplay sessions, or instances, into representational principles and conclusions. Next, the social simulation game *Prom Week* was presented along with an account of the struggle to get players to understand what was happening in the system. These difficulties were the result of the unstable theory that drove the game's creation – particularly that why players might arrive at instances and conclusions is not addressed. This practice suggests future work that would attempt to incorporate formal models of players into the simulative theory.

The following chapter will build from the previous chapter's discussion on the role of instantial assets, as well as this chapter's focus on understanding complex rules, to describe an approach toward understanding games that addresses how and why different players understand games in different ways.

Chapter 4. Players and Procedurality

Meaning in games doesn't come from a single place. Game mechanics do not guarantee that players will enact them, the presence of a goal does not guarantee that a player will pursue it, and an image may not necessarily represent what it is intended to represent. The ways that a player may choose to engage a videogame are limitless. Is the meaning of *The Marriage* its reinforcement of gender stereotypes, its representation of the give and take of marriage, or its historical importance in the game design community? The answer is all of the above and more. Bogost characterized the multi-faceted, nearly boundless, ways to say what a game is *about* by the phrase "videogames are a mess" (Bogost 2009b).

Up until this point, the theories of how videogames are representational and the resulting artifacts have treated some aspects of a gameplay experience as more essential than others. Micro-rhetorics and *Game-O-Matic* are predicated on the idea that a player will pay close interpretive consideration to how the simple mechanics might be interpreted as metaphors for how the static visual elements relate to one another. The simulation theory and the presented interpretation of *Prom Week* assume that the interpreter will take as primary the operation of the complex system and metaphorical vehicles. While these perspectives led to new and more complete understandings of how games are representational, they are not sufficient to describe the many and varied ways that players can understand games.

Particularly, the role of the player was not clearly understood in the two previous theories. With micro-rhetorics it is assumed that the player will perform the actions that will trigger the rules that interact with the thematic considerations to be representational. The simulation theory doesn't address why a player might decide to interpret a principle by choosing to focus on a subset of the countless possible gameplay instances. In the first case, player action is taken for granted and in the second it is hardly accounted for at all.

The goal of this section is to create a comprehensive "anything goes" *proceduralist* theory of videogame meaning that still focuses on the role of processes while also accounting for why individual players might understand a game in different ways. After the presentation of the theory, two in depth examples will show how this approach can be applied. Following this, critiques of the proceduralist perspective in the design process are addressed.

Proceduralist Readings

Essentialist positions that place a game's rules, story, player, etc. as its defining characteristic limit our understanding of how games can be meaningful. The following conception of a videogame's meaning, referred to as a *proceduralist reading*, acknowledges that meaning can be found in any aspect of a game while maintaining the claim that all interpretations must ultimately be grounded in the operation of the game's processes. The imperatives of a proceduralist reading are to be *comprehensive* and *coherent*. A proceduralist reading can be thought of as similar to a close reading in literary analysis where an interpreter is required to commit himself to why he believes what he believes about a game.

The following sections define the components of a proceduralist reading and then show how interpretations can be constructed out of those components in a *meaning derivation*: a hierarchical, proof-like structure for an interpretation of a game and the method of a proceduralist reading. The point is not to say that meaning can be objectively proved, but instead to compensate for the lack of attention to detail in the current state of videogame interpretation. In a meaning derivation, all assumptions of the interpreter are broken into very small units and then logically constructed into rigorous cases for a claimed meaning. A meaning derivation strives to be the best argument it can be for a claimed interpretation.

Mechanism and Culture

A goal of this interpretation framework is to emphasize a game's processes while also accounting for how players necessarily have different understandings of the processes according to their individual circumstances and choices. In his writings challenging commonly held notions of the metaphysics of computation, Brian Cantwell Smith writes that "computing is best understood as a dialectical interplay of meaning and mechanism" (Cantwell Smith 2010). For Cantwell Smith, when people casually conceive of computers as processing information or manipulating symbols, they do not do "justice to [the] concrete empirical practice" in which computation exists "in the wild." In other words, while these notions may be expedient or seem like good characterizations to programmers, they do not accurately describe how computation actually functions when situated in the world. His solution is to frame computation as being characterized by a dialectical relationship between a mechanism and how an interpreter ascribes meaning to it.

Borrowing from Cantwell Smith's characterization of computation, a proceduralist should assume that a game's *meaning* arises from the dialectical interplay between the game's mechanism and the meaning ascribed to it by the player. The mechanism of a game can be best understood as the enframing aspects of the game that a player does not change during play. For videogames, this will include the game's code and physical interface elements, where for analog games this might include the rule set, tokens and physical conditions that are prerequisite to the game's operation (e.g. a table to rest the game board upon). Modding practice and house rules can change the enframing aspect of a game, but this occurs outside of play. And if it does occur during play, then there is a broader enframing aspect (mechanism) which doesn't change during play. This notion of a game's mechanism is different than what is commonly referred to as a game's mechanics. When someone says they like the jump mechanic in a game, they are already interpreting a part of the game's mechanism as representing a jump. The most important thing to recognize about a mechanism is that it is meaningless until it is encountered by players. For instance, the code inside of a machine may have been created by a programmer with a certain output in mind, and he may be able to tell stories about how it operates, but this concept of a game as mechanism strives to ensure a strict separation between the game as machine and the meaning that players ascribe to it. In other words, until the audience encounters and interprets a game, the code can be treated as nothing more than abstract causal flows that each interpreter ascribes meaning to.

It is worth noting that understanding a game's mechanism is different than looking at the source code. Parts of a game's code may be necessary for it to function but don't contribute directly to interpretations. For example, knowing the precise way collision detection is computed may not directly impact the meaning players ascribe to a game, while the fact that a game employs collision detection at all is highly relevant to meaning ascription (see the discussion of graphical logics in (Mateas and Wardrip-Fruin 2009)). By constructing an analysis of how a game is operational based on player experience, rather than studying the source code and algorithms that comprise it, the player and critic are forced to focus on the relationship between mechanism and meaning.

Cultural considerations are axiomatic assumptions that are true for a group of people. In other words, an interpreter either belongs to a group that believes them, or at least is willing to grant their validity, or does not. The culture of the interpreter necessarily influences all interpretations. For instance, entity A, which is colored blue, only might be argued to represent a male to someone who carries the cultural association that the color blue is gendered in that way.

While where cultural beliefs come from is important, the proceduralist perspective does not attempt to address these concerns as they stray from the materiality of the game itself. A proceduralist reading attempts to better understand how a videogame is able to create meaning rather than where a videogame *relates* cultural concerns. As an example of the sort of analysis that proceduralist readings are not suited to, Helen Kennedy's essay about whether *Tomb Raider's* protagonist Lara Croft is a feminist icon or not can be said to say more about theories of feminism than *Tomb Raider* as a game (Kennedy 2002). These perspectives are certainly valuable and help us understand how *Tomb Raider* functions in culture, but they fail to meet proceduralist reading's imperative of reasonable comprehensiveness as they treat *Tomb Raider* as a static artifact that produces phenomena in culture, while ignoring the specific operation of the mechanisms that comprise it.

Maintaining a strict focus on accounting for the mechanism of a game is the definitive feature of a proceduralist reading. Of course, as shown in the previous interpretation methods, cultural beliefs are of the utmost importance when interpreting a game. Players are humans after all, and humans have different cultural backgrounds and personal histories that influence their beliefs. To accommodate for cultural considerations, as many foundational cultural beliefs are to be clearly stated as possible. The blunt conception of culture being a matter of accepting a set of axiomatic claims is meant to prevent interpretations from straying from the game's mechanism. The influence of culture will be further explored below.

Interpretations

When players engage a game, they manipulate its mechanism as influenced by their cultural context. What actions are taken, how visuals are perceived, the affective state of the player and generally everything about the experience arises from the interplay between mechanism of the game and its player's cultural context.

Interpretations are the objective of a proceduralist reading. All interpretations are constructed from beliefs about the game's mechanism, the player's cultural context or other established interpretations. In other words, it is not possible for an interpretive consideration to be a leaf in a meaning derivation graph (see Figure 26). Interpretations amount to beliefs about a game and they come in many types.

The most important and troublesome claims that can be made about a game are about what *dynamics* it can be claimed to produce. Dynamics are "…the run-time behavior of the mechanics acting on player inputs and each others' outputs over time" (Hunicke, Leblanc, and Zubek 2004). Dynamics describe what happens during run time and given the interactive nature of games, they will necessarily differ between players. For example, if the code defines that B is removed upon a collision with A, a player will only understand that A destroys B once that event occurs – which is not guaranteed. More complicated dynamics are emergent and often unpredictable.

Representational interpretations can vary from simple assertions about what an instantial asset represents to to overarching claims about the *moral of the game* (story). In this case, the word representation is used to refer to the game's intentionality or aboutness. As a simple example, a particular culture may believe green to be a more *friendly* color than red, and take that into consideration when deciding the hero or villain of a game story. If the red circle was chasing the green circle, a dynamic, the appearance of pursuit could also be used as further support that red is a villain of some sort. These simple representational claims do not require that the interpreter justify their beliefs, however relying wholly on assumptions about the visual rhetoric of instantial assets may weaken an argument. More contentious claims about what a segment of gameplay means in a symbolic, representational sense can require more grounds. This will be explained in discussion of meaning derivations below.

Aesthetic claims can be made to describe an interpreter's sense of taste or feelings about a segment of gameplay. For example, one may find the repetitive and stochastic gameplay of a slot machine to be immoral, or offensive. An interpreter could use this aesthetic judgment, the governing code, and other interpretive or cultural considerations when forming their interpretation. Aesthetic considerations also encompass emotional responses of the interpreter (e.g. "seeing the ant disappear after colliding with the shoe made me sad").

It is important to note that interpretations influence each other. For example, that a player regards a green circle as friendly could influence the player's aesthetic considerations which would change their dynamic behavior and ultimately what the game can be said to represent. To complicate matters further, the idea itself that a game represents something may influence a player's aesthetic judgments which in turn could change their behavior (dynamics). Other types of interpretations include political, moral, or pretty much any other type of claim an interpreter might want to make about a game.

Meaning Derivations

A meaning derivation is where the possibly limitless observable aspects of a game with meaning potential are threaded together into comprehensive and rigorous arguments for an interpretation. A meaning derivation can be understood as a hierarchical, structured graph that explicitly states what an interpreter believes about a game and why. Meaning derivations are defined to be constructed out of the three types of considerations described above: Mechanical, Cultural and Interpretive.

Meaning derivations strive to make all relevant considerations in an interpretation explicit. The benefit of this formal structure is that it allows different interpreters to identify points of disagreement while explicating claims as to how various aspects of a videogame operate to produce a conclusion. The primary evaluative criteria of a meaning derivation are comprehensiveness, how much it accounts for obvious observations, and coherence, and how much an interpreter is comfortable with the supporting arguments that interpretations are grounded in.

Comprehensiveness

Just like any argument, a meaning derivation will not be considered strong if it ignores evidence that goes against its claim. This is particularly difficult for games, as there are so many types of evidence. Despite this, a proceduralist reading strives to account for as many of the observations about code, dynamics, aesthetics, representation, etc. and how they relate to the culture of the interpreter as possible. Of course, it will never be the case that the limitless observations and interpretations about a game are possible. The imperative of *comprehensiveness* is a qualitative measurement of the breadth and quantity of types of considerations accounted for.

This requirement can be demanding as it turns out to be rare that people pay attention to details of a game's mechanism when talking about them. One such example is Poole's claim that *Pac-Man* is a game about rampant consumerism in America based on the player's goal of relentlessly collecting dots in a seemingly endless pursuit of points (Poole 2000). A meaning derivation to support this claim would not carry much weight as there are obvious facts about the game that do not support the claimed meaning. In this case, among other things, the interpretation does not adequately explain the role of the ghosts that chase the player. Furthermore, one of the game's key mechanics, that upon colliding with the larger dots (the *power pellets*), the power shifts and ghosts are sent running from the player who can now collide with them for points, is ignored.

Even if these mechanics were addressed in Poole's argument, a proceduralist reading is not considered strong if it does not address obvious interpretations about the visuals, dynamics, aesthetics, etc. of a game. What does it mean that most players of *Pac Man* enact a dynamic where they let the ghosts get very near them before they eat the big dots to ensure that they can consume as many ghosts as possible while the power pellet power up is in effect? Also, that the enemies are themed as ghosts, along with the cultural associations that ghosts carry, should not be ignored. Poole's interpretation does not address the thematic mappings of *Pac Man*. Similarly, the abstract *feelings* and subjective judgments that result from being chased through a maze, the aesthetics, are also facts that are interpretable.

Coherence

Meaning derivations are a hierarchical structure of observations and interpretations that are held together by *interpretive leaps*. These interpretive leaps can be graphically represented as arrows between nodes, or as the right hand side of interpretive rules where the left hand side is composed of the conjunction of other beliefs about the game. Because of the subjective nature of interpretation, the only thing that validates the generation of new beliefs is that a subject is willing to grant them. For example, an interpreter may or may not believe that because a game entity is colored blue, it is supposed to thematically represent a male. All interpretive leaps are culturally grounded. Despite the appearance of formalism, an interpretive leap can be predicated on a belief about the code, or other interpretations that *aren't true*. All that matters is that the interpreter is willing to grant it as true. With the complex nature of game dynamics, this is a more common occurrence than one might expect. Finally, the statement that a meaning derivation is validated according to its coherence begs the question, *to whom must a meaning derivation be coherent?* The answer is that a meaning derivation must be coherent to the person who is evaluating it.

The following two examples will illustrate how an interpreter's beliefs about the game as a machine and their cultural beliefs can generate interpretive considerations that can be structured into meaning derivations that are comprehensive and coherent.

Examples

Below are two detailed examples of interpretations of two different games. The first is an imagined simple game and the second a game with complex dynamics that was created to represent a theory. While a proceduralist reading does not

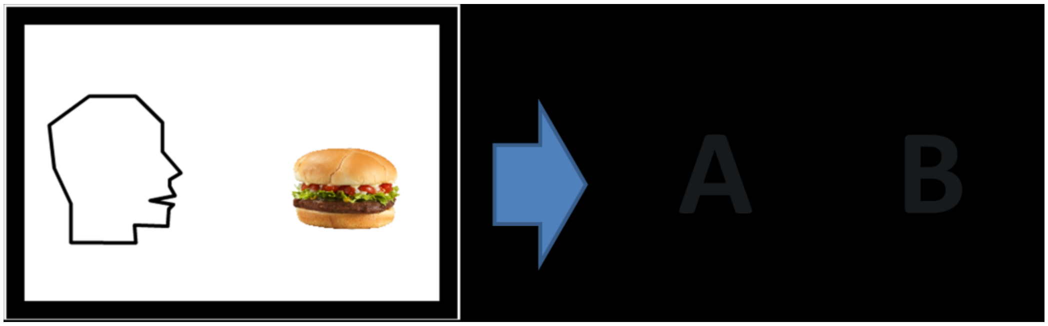

Figure 25 - To perform a proceduralist reading on the example game about a man and a burger, we first describe the game as abstractly as possible.

necessarily require a formal meaning derivation in order to be coherent, the following will show all the steps of reasoning that lead to interpretations.

A Man Eats a Burger

Imagine a very simple game where the player controls a human head with the arrow keys. There is also a hamburger on the game screen (Figure 25). When the human head collides with the hamburger, the hamburger disappears and text appears on the screen indicating that the player has won. It is evident that this game would be understood by most as a representation of a man eating a hamburger. However, what must be believed by the interpreter to believe this about the game is exposed by a proceduralist reading.

To begin a proceduralist reading on this game, the player must first identify and abstractly define the entities in this game. For example, the human head will be referred to as entity A and the burger will be entity B. Other aspects of the definition are that the player controls entity A with the keyboard. The mechanics of this game can be minimally described as “if A collides with B, B disappears” and “when B disappears, the player wins.”

The above descriptions of the code is the foundation, or mechanical truth, that will be drawn upon and synthesized through a meaning derivation to arrive at the representational interpretation that this game represents a man eating a burger.

First, we can take the definitions above, that the goal is to remove B, and that the player controls A, to infer the, almost too obvious to state, dynamic that A will move toward and collide with B. Note how the code itself does not guarantee that this will happen. We make the *interpretive leap* that this dynamic will occur.

Now we can specify our beliefs about the visuals of the game. One such belief is that A looks like a human, and I understand that humans eat food. Another is that B looks like food. One might at this point not agree with the interpretive leap that B is food if they were a vegetarian. However, we are going to build our meaning derivation with the understanding that a vegetarian, who projected themselves onto entity A, would understand the game very differently than we do in our simple interpretation.

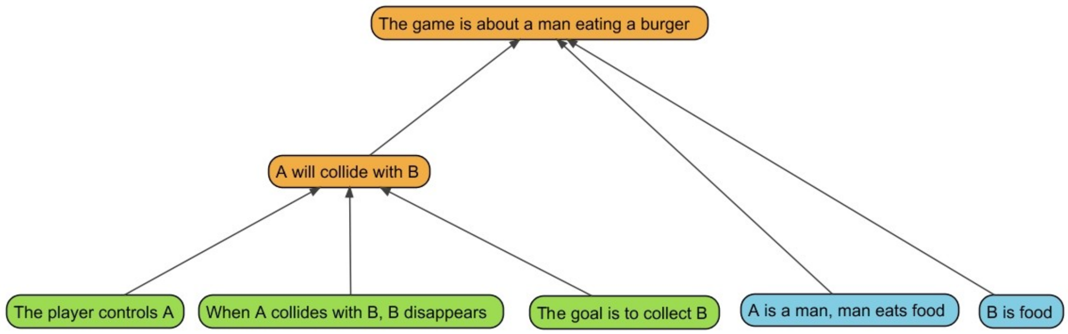

Figure 26 - A complete meaning derivation. Starting at the bottom, after a series of interpretive leaps we arrive at the result of our simple proceduralist reading that the game with the man and the hamburger represents a man eating a hamburger.

While our interpretation does not involve any aesthetic considerations, one could imagine our imagined vegetarian interpreter as finding this game *gross* or immoral. This consideration would no doubt be important to their understanding of what the game represents. Figure 26 shows how all of our considerations can be structured into an explicit, hierarchical argument for why we believe that the game represents a man eating a burger.

The Free Culture Game

In 2009, Molleindustria produced *The Free Culture Game* as a "playable theory." According to the text accompanying the game: "*The Free Culture Game* is a game about the struggle between free culture and copyright. Create and defend the common knowledge from the vectorial class. Liberate the passive consumers from the domain of the market." The theory, from McKenzie Wark's *A Hacker Manifesto* (Wark 2004), uses the phrase vectorialist to refer to the owners of data in contrast to the producers, the hackers.

The definitions and goal are explained using text before the game begins, providing us with a point of comparison between the authorial intention and our experience of the game.

To illustrate how the depth of reading can influence interpretation, we will take two passes at *The Free Culture Game*. The first represents the experience of playing the game once for a short amount of time, as one might do when curiously clicking a webpage link referred by a friend. The second is a deeper reading that adds additional definitions and dynamics that could be identified with longer engagement with the game.

We begin by defining the entities, meters, goal, and mechanics. The entities are rule-governed items that participate in mechanics. Though we cannot see any meters on screen, playing the game makes it apparent that numbers are being counted behind the scenes and that the green people fade into a duller color over time and eventually convert into the grey people in the grey outer ring. The opening text of the game explicitly states the goal of converting everybody into Commons participants and that the player is the distributor of knowledge.

Because the mechanics and considerations about *The Free Culture Game* will be more complicated than the previous example, the meaning derivations will be presented in a symbolic notation with the logical *and* and *implication* symbols, rather than in a graphical form. While the description of the mechanism below makes use of identifying labels, it is only for means of communication. All identifiers are meant to be regarded as abstract symbols. Each interpretation will be presented by an identifier and number that indicate what type of consideration they are (e.g. dynamic, aesthetic, etc.). The *conclusions* are the roots of the meaning derivation and are built out of lower level interpretations and aspects of the game as machine.

Mechanism:

Entities:

- Cursor (Blue Circle)
- Producers (Green People)
- Consumers (Grey People)
- New Ideas (Yellow Lightbulbs)
- Vectorialist (Vacuum)

Meter:

- Ideas Absorbed

Goal:

- Turn everybody green

Control:

- Player is blue circle controlled by mouse

Rules:

- Producers spawn new ideas
- New ideas are moved by an indirect force from the blue circle
- Vectorialist moves near groups of new ideas

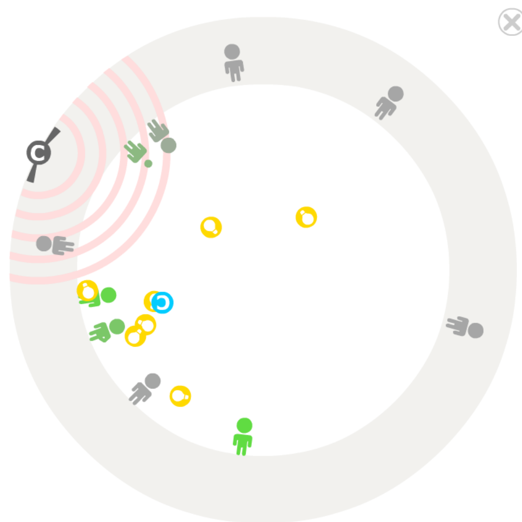

Figure 27 In The Free Culture Game the player controls the blue entity (the force of the commons) and pushes the yellow light bulbs (ideas) toward the inward facing people.

- Vectorialist pulls in new ideas
- Collision between new ideas and the Vectorialist causes new ideas to disappear
- Collision between new ideas and green person increases Ideas Absorbed
- Ideas Absorbed goes down slowly over time
- Producer with empty Ideas Absorbed meter changes to Consumer

Dynamics:

- **dynamic(1)**: Because producers create new ideas, that the player's goal is to turn everybody green and that the player exerts force on new ideas, *the player will push ideas toward green people to keep them from turning grey*

- **dynamic(2)**: Because the vectorialist pulls in new ideas and a collision between new ideas and the vectorialist causes new ideas to disappear, *the player must get between the vectorialist and new ideas to prevent them from being sucked up*

Aesthetics:

- **aesthetic(1)** → the lack of control over the indirect force that determines how the player's cursors acts on the ideas is frustrating

Representations:

- **representation(1)**: Green people are made to look happier than grey through color and animation
- **representation (2)**: New ideas are desirable objects
- **representation (3)**: Vectorialist visual design is a cold grey and its behavior is automatic, both unfavorable connotations

Conclusions:

- dynamic(1) ^ aesthetic(1) ^ representation(1) → **representation(4)**: New ideas are hard to control but, with careful attention, they will benefit everybody
- dynamic(2) ^ representation (1) ^ representation (3) ^ representation(1) → **representation (5)**: the vectorialist is out to steal ideas and does not care about the happiness of people

If a player only spends a couple minutes with *The Free Culture Game*, it is likely that their interpretation is that free ideas need to be protected from ravenous privatization of a force that turns active producers into passive consumers. This process, as it turns out, is quite difficult and needs to be tended to with care or else all producers will be converted to consumers and there will be no more new ideas.

But the system represented by Molleindustria's game does not actually spiral into a single inevitable conclusion. Instead, careful observers will note there are additional dynamics at play which demonstrate the vectorialist's need for new ideas to keep consumers happy. We will briefly define those new components.

Entities: Old Ideas

Meters: Ideas Fed

Rules:

- Vectorialist feeds the new ideas it collects as old ideas to Consumers.
- When meter Ideas Fed reaches zero, Consumer changes into Producer
- As Ideas Absorbed meter fills, Producer creates New Ideas more frequently

In the first play through it appeared that the vectorialist was taking ideas out of the commons, but the introduction of the Old Ideas element creates a direct relationship between the ideas taken and the ability to keep the Consumers happy. Additionally, when producers are happier they can generate more new ideas.

Dynamics:

- **dynamic(3)**: if the vectorialist does not have enough ideas Consumers will move back to the Commons

Conclusions:

- dynamic(3) ^ representation(2) ^ dynamic(1) → **representation (3)**: If more ideas are believed to be better than fewer ideas, then maximizing idea production takes active intervention in the commons. Otherwise, the process is cyclical but stagnant.

Because there is no end state, *The Free Culture Game* will continue indefinitely without player interaction. Unhappy consumers will return to the commons to become producers again and the vectorialist will endlessly pursue those new ideas. It is not a game to be won or lost, but rather a "playable theory" that illustrates how variables in the system are handled.

While both of these examples demonstrate how meaning derivations can result in arguments for representational claims about a game, the same process could be used to justify any interpretations about a game (dynamic, aesthetic, moral, etc.).

Critiques of Proceduralist Design

Some argue that this approach toward understanding games does not account for the rich and varied ways in which people actually play games. Wilson writes "framing game design as the art of 'system design' makes the critical mistake of focusing too intently on the media object itself" (Wilson 2012). Stenros and Waern lament that "games are most often seen as systems. This has made the play *activity* an under-explored area of game studies" (Stenros and Waern 2011). Taylor writes about how rules in games are created, negotiated and changed by players to create their own meanings: "it's not that play is either rule or nonrule based but a question of whose rules in which contexts" (Taylor 2006). These authors among others believe that a system centric view of games treats games as static artifacts rather than social or personal activities and it is these phenomena that are essential to understand if one is to understand a game.

Framed in this way, two points of view can be identified: one perspective privileging the notion of games as systems of rules, and the other emphasizing how individuals and communities create meaning through play. As with most dichotomies, it is fairly easy to present the extreme of each perspective and make them seem ridiculous. Sicart takes this approach in his essay that condemns a process-centric (or proceduralist) approach for being *totalitarian* because such a designer prescripts a player's choices before he or she ever makes them (Sicart 2011). Rather than simply continuing a debate on this subject that presents simplistic and uncharitable views of the opposing camp, this section is meant to provide a practical theoretical foundation for the proceduralist position that will help clarify the position. Also, with a stronger theoretical foundation, it is hoped that both players and designers can better understand the process centric meaning of games.

The proceduralist position strives to understand a game's meaning in the context of the processes that its system affords. This perspective can be contrasted with sociological perspectives that strive to understand a game in terms of player communities, or other accounts of games that describe how its meaning is situated in culture and history. While these other perspectives are valuable, the proceduralist strives to understand the inner workings of the game as a machine to which meaning is ascribed by players. The proceduralist project might be seen as sharing similarities with the New Criticism movement which strove to understand how language can be *charged with meaning*, without relying on authorial intention, individual experiences, or historical context. Another connection can be found in the movement's founding goal of making criticism become "more scientific, or precise and systematic" (Ransom 1938). To the proceduralist, creating and understanding games requires understanding system dynamics, which necessarily involves precise and systematic investigation into the precise operations that drive a game's system.

The claim that a game is meaningful through its processes is far from clear. The field of artificial intelligence has struggled with related issues when trying to determine what it means for a system to be *intelligent* (Agre 1997; Mateas 2001). One perspective sees computational intelligence as problem solving through the manipulation of internal symbols (mentalist AI) and the other sees it as embodied activity in an situated environment (interactionist AI). The mentalist perspective lends itself toward creating systems that focus on solving problems using approaches associated with activities that happen inside the mind (e.g. forming goals, planning, etc.). In contrast, the interactionist perspective focuses on agents that acknowledge and react to context (e.g. exhibit reactivity and improvisation). For a mentalist, the interactionist will have a hard time building systems that engage in complex symbolic behavior (e.g. language use) without recourse to concepts of symbolic representation. For an interactionist, the mentalist will have a hard time building systems that take physical action in the world because of an overemphasis on the manipulation of internal symbols.

This debate from the field of artificial intelligence helps identify what is at stake in this debate in games research. Parallels can be drawn between the mentalist and the proceduralist positions and the play-centric and interactionist positions. Where interactionists accused the mentalist approaches of attempting to create a notion of intelligence that existed without context, those of the play-centric perspective are concerned that proceduralists are striving to create games that are meaningful without players (Pratt 2012). The reaction has been to deemphasize the importance and role of procedural rhetoric in games. The proceduralist is concerned that the play-centric perspective overstates the freedom of players at the expense of the still-to-be explored field of procedural rhetoric.

Before we start choosing sides, we should remember that these characterizations do not necessarily describe the practice of any particular game designer or researcher. Those of the play-centric and interactionist-leaning position certainly would not deny that the space of possible actions is constrained by the game system. Likewise, the proceduralist does not believe that player activity is irrelevant, having no significance to the meaning of a game. However, these exaggerated positions do serve as warnings as to where errors might be made without theoretical diligence.

In this spirit, the following thought experiment will present the design process of an imagined *naive* proceduralist that falls prey to the dangers that those dissatisfied with the proceduralist approach warn of. By identifying possible faults of an approach that privileges a game's processes, a more subtle proceduralist approach that acknowledges the importance of players will be presented.

The Naive Proceduralist

The following is a caricature of the design process of an imagined process-centric, or proceduralist, designer. Through this exaggerated position, several possible problematic conceptions about how processes convey meaning can be extracted.

We begin by imagining a game developer that wants to make a game about some domain like global warming. First, the designer decides what message he wants players to walk away with. Let's say he wants to advocate that government regulation of carbon emissions is the best way to prevent global catastrophe. Because this designer is a "proceduralist," he now goes about trying to harness the "unique" potential of videogames to express messages through gameplay, rather than through just telling players a message as would be done in traditional media like literature and film. It is important to the proceduralist that players participate in creating messages through actions rather than simply be presented with information. Because he wants to advocate for the regulation of carbon emissions, he puts the player in the role of someone who can, at least symbolically, exercise this sort of power: the leader of a country.

Now the designer imagines that he just needs to get the player to choose to create the regulations in the game. This implies that the player must have the gameplay option of creating this regulation and once the regulation is applied, the problems of global warming will begin to subside. To give an incentive to choose this action, the designer creates some rules that cause water levels to rise at the start of the game. If the player doesn't manage to curb this trend, by putting into place regulations on carbon emissions, the game will end in failure. With this imagined dynamic, the designer feels comfortable that a player *should* decide to enact the regulation or lose the game. And with this, the designer feels content that the game is representing his desired message through processes, and he can start ornamenting the game with additional gameplay as well as instantial assets that represent the game state. Perhaps the designer repeats this process of imagining a message, constructing a scenario where the player is expected to enact some choice that embodies some message, creating a game that presents several messages through its processes.

Several problems arise from this naive approach. First of all, the complexity and quality of the supposed procedural messages can hardly be said to stand up to the expectations of the proceduralist evangelists, as even if this game succeeded in representing the designer's intent to a player, it is not clear that the resulting game is any more impactful or relevant than a short paragraph of text describing that carbon emissions are related to global warming. Also, this gameplay experience is lacking one of the most essential rhetorical strategies a game can employ: providing a high agency experience for the player (Treanor and Mateas 2009). The player's limited choices are not likely to leave the player feeling like he can take the actions that the domain suggests. Especially considering a hotly contested subject like global warming, one would expect that a game would allow players to explore the details of this issue, rather than be told a didactic message. While it may be the case that, as described, the imagined player wouldn't have any reason *not* to take the intended action of enacting the regulation, it seems more likely that a player would feel compelled to outright stop playing the game than be forced to choose among limited actions when other possibilities for actions obvious to the player have been so overtly excluded from the game. And surely, if a game is never played, there's no sense in which the game can be said to convey a message.

This straw man design process literally prescribes what a player will do and why. Implicit is tacit agreement with Sicart's claim that "Proceduralists believe that… behaviors can be predicted, even contained, by the rules, and therefore the meaning of the game, and of play, evolves from the way the game has been created and not how it is played" (Sicart 2011). But as Nelson points out, this condemnation of a proceduralist approach can be seen instead as "opposition, aesthetically and/or politically, to certain kinds of unsubtle, didactic rhetoric in general—of which unsubtle, didactic procedural rhetoric is one variety among many" (Nelson 2012). But what does a less didactic design process look like? Surely, a design and interpretive method that can take account of how meaning arises through interaction with a game system must have a more thorough account of both the role of the player and of computational processes than this naive proceduralist.

The Proper Proceduralist

We can now describe a more nuanced conception of a proceduralist game that accounts for the subjectivity of its players. For a proceduralist to succeed in creating a game that is meaningful through its processes, players must ascribe meaning to the game as machine. Without interpreters, a process inside a digital computer can amount to no more than abstract causal flows of electrons. Likewise, the mechanisms of physical games, like football or board games, are not meaningful until a player puts them into operation by ascribing them meaning. Game rules must be first interpreted by players and then understood as the vehicles of metaphors about some domain. The ways in which players will narrate the operation of the machine arise from an interplay between the preexisting beliefs about the represented entities (visuals, sound, story) and the ways that these entities are manipulated by the game's processes.

Viewed in this way, a proceduralist cannot be accused of treating players as mere "activators of the process that sets the meanings contained in the game in motion" (Sicart 2011). On the contrary, it is impossible for game designers to *embed* any meaning at all inside of a game, as they have no direct power over how players choose to narrate the operation of the game as machine. Furthermore, if the game does not afford interaction that the player finds meaningful in the greater scope of their life, the player will most likely seek something else to engage with, and cease to play. If the processes never even occur, it is incoherent to argue that the game as mechanism without narration *contains* meaning. If a proceduralist wants to create a game with some specific meaning, it is important that the game actually have players that want to engage with it such that they naturally create the dynamics that align with the authorial intent.

Using the language of semiotics, Mateas characterizes how systems signify by stating "Every system is doubled, consisting of both a computational and rhetorical machine" (Mateas 2003). Each machine is productive of its own signs that audiences synthesize into what they consider the system to mean. The rhetorical machine refers to the many sign systems in the world. These are the systems of signification that the author has no control over such as cultural considerations and personal history. The computational machine is then described as being made up of two semiotic systems which are productive of different syntagms. The first system (system^1) is the system architecture. This is comprised of the data structures and algorithms that manipulate them. In a game like *The Sims,* part of system^1 would be the artificial intelligence system that manages the agent's needs (such as hunger, hygiene etc.). Notice how this characterization of the AI system already involves interpretation as the concepts of *managing*, *agents* and *needs* are not inherent in the system itself, but instead narrations of system^1 that the author uses create the game. Thus, the author's understanding of this system will constrain and afford what *content* (syntagm^1) he is able to create for the system. For example, because *The Sims'* AI system is able to be conceived of as modeling physiological needs, the game's authors/programmers were able to create and tune particular representations of needs.

The second system, system^2, refers to the system in operation in front of a player. System^2 is the instantiation of the syntagm^{1}s as they execute. System^2 is productive of its own syntagm^{2}s (e.g. specific traces of behavior in a play through the game). These syntagms make heavy use of *handled* signs such as the customary meaning of animations, language spoken by the characters, etc., which are not strictly represented in the system. Syntagms^2 are not amenable to perfect prediction as the sign systems of the rhetorical machine are always out of the creator's grasp.

A proceduralist, then, is someone who can both understand how to create systems that they can reliably author for, and anticipate the ways in which cultural context is going to influence the output of the system once it is running *in the wild*. Predicting how a player might encounter system^2 is the most difficult problem of procedural rhetoric. Producing a game that expresses an intended meaning will always involve iterating upon the design of system^1, as informed by investigations into the systems of the rhetorical machine that will ultimately interact with system^2 to produce the artifact's signification (syntagm^2). This process will always be imperfect as the systems of the rhetorical machine are irreducible and impossible to formalize.

One strategy for attempting to understand or account for player subjectivity can be seen in Gingold's concept of the "human play machine" (Gingold 2009). Designer's can imagine players as a complicated system that can afford many types of interaction, or *play capacities*, with system^2. For example, a player might consider their senses, culture, language, emotions, imagination, etc. when choosing how to act. While Gingold's concept may be seen as systematizing players, the number of play capacities far exceeds the number of considerations that a designer can practically have. To maximize the chance that a desired interpretation will be arrived at, the designer should account for as many play capacities as possible. Without carefully considering the many ways that a player might engage a game, a designer makes the mistake of reducing players to pieces of the game as machine, rather than individuals. It is imprudent for a designer to make this mistake, as it is unlikely that players will choose to engage a game that does not respect their autonomy.

Above was an argument that claimed that consistent and comprehensive accountability of a game's processes is a primary value of proceduralist design. If there are aspects of a game that do not contribute, or even worst detract, from a designer's intended representation, it is less likely that players will regard the game in a desired way. While for visual rhetoric this is commonly accepted (e.g. offensive imagery for no purpose will distract from the desired representation), games will often have processes that are not accounted for or even prevent the desired interpretation from being possible. One example of this can be seen in *Bioshock* where the game's ambitious narrative critique of a philosophy is undermined by violent and conventional gameplay.

Jason Rohrer's *Passage* is an example of a proceduralist game that many people have found meaningful. *Passage* "presents an entire life, from young adulthood through old age and death, in the span of five minutes" (Rohrer 2007).

According to Rohrer, each game mechanic had specific representational authorial intent and the game is considered to be one of the founding games of what has been called the proceduralist style (Bogost 2009a). The game's success can be attributed to Rohrer's ability to design the game such that players, as culturally situated individuals, naturally interpret the rules as meaningful. It isn't the case that authorial intent is actually *embedded* in the game and players merely activate the flow of signification; players actively create and negotiate meaning as independent subjects. What makes this a proceduralist game is that the ways that most players find the game meaningful involves the procedural aspects and these interpretations *happen* to align with Rohrer's stated intentions about how the rules of the game were meant to be metaphorical (Fagone 2008; Rohrer 2007). In the semiotic language, Rohrer was able to author for a system^1 (graphical logics) that when put into operation, creating system^2, and put into contact with the sign systems of the world, were able to produce syntagm^{2}s that aligned with the intentions of the authored syntagm^{1}s.

Contributing factors for *Passage's* success are the high level of agency achieved by limiting the fidelity of the interaction and visuals (Mateas 2006) and music that sets an introspective tone. However, other factors have less to do with the artifact itself, but more to do with the cultural milieu of the time. When Passage was released, film critic Roger Ebert had recently declared that games could never be art (Ebert 2005). While these historical and cultural considerations are not the focus of a proceduralist perspective, they can still have a strong effect on how games are meaningful to players. In this case, Rohrer became the foil to Ebert in an ongoing debate about the deserved cultural status of games and this debate likely brought attention and authenticity to Rohrer's work and authorial intention.

In summary, a proceduralist is someone who treats a game's processes as primary when considering a game's meaning. A proceduralist must accept that the only aspect of the game that they have direct control over is the game as mechanism and that the meaning of the artifact is ultimately produced through the dialectical interplay between the mechanism and ways that players ascribe meaning to it.

Difficulties with Procedural Rhetoric

Creating a game with procedural rhetoric is very difficult as players create interpretations about the game from their individual contexts regardless of authorial intent. However, this is less of a problem for games designed with very specific interpretations in mind. With *The Free Culture Game*, meaning derivations were easy to construct and believe as the creator structured the game such that particular sets of observations and interpretive leaps might be developed. The *Free Culture Game* even goes so far as to outright tell the player what each game entity represents and how the player is to interpret the dynamics.

However, even for games designed to carry specific messages, it is hard to come up with proceduralist readings that stand up to scrutiny. As an example, *Bacteria Salad* is a game created in response to and released during the spinach related E. Coli outbreaks in 2006 (Figure 28). The game challenges the player with a task of running a large agribusiness. The goal is to sell as many vegetables as possible without getting the consumers sick in the process. Through building and destroying

Figure 28 Bacteria Salad makes a sophisticated procedural argument against large scale agribusiness.

farms, harvesting crops, and issuing bans on contaminated vegetables, the player has control over the distribution of the industry's vegetables.

Bacteria Salad is a form of editorial much like written editorials in newspapers. Without giving a complete meaning derivation, it can be argued that *Bacteria Salad* argues that as agricultural production becomes more industrial, the relationship between where food is consumed and where it is produced becomes very complex. This makes contamination hard to contain because distributors lose knowledge of where their product is shipped to; the only way to deal with contamination is through massive bans. The game communicates this by making the successful strategy be to actively manage small farms, where more targeted bans are possible.

However, the above interpretation can be objected to on grounds that it omits a dynamic that implies a representation that was clearly not intended. Because there is no cost or penalty for destroying industrialized farms, it is possible to always produce food using the heavy industrial farms to fill the shelves but clear the fields once the shelves are full (to avoid contamination). If the player detects that a contamination may have happened, he should issue bans and consider it a loss, but many times contamination does not occur and the player gets the benefit of rapid development and big pay outs. When taken literally, this strategy would lead one to believe that the game is advocating the rapid creation and destruction of industrialized farms. All simulations involve making decisions about what aspects of the real world to incorporate into the simulation (and in what simplified form) and which to leave out. In this case, leaving out of the simulation the costs of building and destroying industrial farms leads to an unexpected, and presumably undesired, emergent gameplay opportunity, and thus emergent representation. Appropriately simplifying and constraining the underlying procedural model such that it avoids actualizing unintended rhetorics is a very hard design problem with procedural rhetoric. In this case, while the game affords a reasonable interpretation in some ways, it fails to meet the criteria of comprehensiveness.

Figure 29 Madrid demonstrates how procedural rhetoric can send conflicting messages.

As another example, *Madrid* is a game created after the 2004 Madrid train bombings (Figure 28). Rather than simulating the event, *Madrid* tasks the player to click on candles in order to light them at a memorial gathering to honor victims of violence around the world. The candles begin lit and start to fade with time. The goal is to have the brightness meter at the bottom left to fill to a point at which the game is won. The game is clearly trying to simulate the act of memorial with the candles representing the memory of the victims. The game reminds us that in order to honor the victims it takes effort to stop their memory from fading. When read at this level of abstraction, the game uses a rhetoric of memorial to honor the victims of terrorism.

The message of *Madrid* becomes confused when a more literal reading is performed. Because the candles must be lit (clicked on) very quickly in order to raise the brightness meter, *Madrid* induces a frantic and busy state in the player. It takes much effort and grueling commitment to achieve the win state of the game. Because of this, *Madrid* has been described as using a rhetoric of “precision and diligence”

(Bogost 2007). Because grueling diligence and precision have little to nothing to do with memorial, the game's message ends up being obscured and inconsistent. The player's emotional or aesthetic response to the game should have some connection to the game's message, or at least be accounted for in some way. *Madrid* fails to do this because the primary emotional response from the player is frustration as the game primarily tests hand/eye coordination.

Once we accept code, culture and interpretations about dynamics, representation, aesthetics and more as meaningful, while also striving for comprehensiveness and coherence, we arrive at many difficulties. Because the interpretative considerations of a proceduralist reading are so interconnected and generative of other considerations, it is easy to arguably fail to be rhetorically stable. In the case of *Bacteria Salad*, an interpretation is sound as long as the interpreter chooses to not consider certain observations. With *Madrid*, a reasonable understanding of the game requires an interpreter to give rhetorical primacy to the associations with the visual aspects of the game. And with *Passage*, much of the game's expressive success can be argued to have had to do with historical factors that had nothing to do with the procedural aspects of the game's design.

Conclusion

The proceduralist approach understands games as meaningful through their processes. A player's beliefs about how the game operates as a machine will constrain the space of actions he considers taking and ultimately determine what dynamics occur that require interpretation. It seems then to imply that creating a game with a particular procedural rhetoric involves crafting its procedural aspects such that the game is able to produce the dynamics that a player can interpret in the desired way.

Of course, a player's beliefs about code and culture, as well as the rhetorical significance he might place on the possible observations about a game, cannot be known by the designer or researcher without empirical study and play testing. Even then, all it takes for a player to not adopt a belief about a game is for some aspect of the designer's imagined meaning derivation to not be convincing, or possibly even observable. This could simply be a matter of not sharing the assumed cultural context (not associating blue with male), not synthesizing an interpretation in the expected way (not interpreting that a picture of food being removed by colliding with a picture of a human head is sufficient to conclude that the gameplay represents the human eating the food) or having a different understanding of the operation of the game's mechanism (not believing that entity A is shrinking because of time passing, but rather because A has not collided with B recently).

A naive proceduralist believes that because rules are present, a player will find them meaningful. A proper proceduralist recognizes that determining meaning is an irreducible task that will necessarily involve the individual players who synthesize the meaning themselves as individual subjects. Play centric perspectives are helpful in that they remind us just how varied the perspectives can be with which someone might approach a game, but they do not help us better understand and innovate on the mechanisms that underlie the play activity.

The main purpose of indentifying and advocating for a proceduralist perspective is to enable creators to make new kinds of games, and players to understand new aspects of the world. Particularly when creating a proceduralist game, the designer must consider the rhetorical significance of the process oriented aspects of the artifact that they are considering that are difficult to grasp. It is arguably much easier to capture visual and aural renderings of perspectives than it is to render the principles that help shape experience. With games, the proceduralist sees an opportunity to attempt to represent the procedural aspects of reality that are difficult to express with other media.

Chapter 5. Alien Readings

The previous chapter explored just how unstable and unpredictable the meaning of a game can be as a result of the irreducible sociocultural context of individual players. The proceduralist framework requires that interpreters consider not just how players understand games, but also that the game as mechanism is accounted for with a maximal degree of comprehensiveness. As discussed in the previous chapter, the proceduralist approach has been accused of overemphasizing the role of the system, and underemphasizing the role of the player. This can be taken further and argued that it even marginalizes players in insisting that interpretations stand up to the imperative of comprehensiveness. Recall Agre's discussion of how the metaphors that underlie technical practice necessarily treat some aspect in the *center*, where the metaphors function well, and others in the *margins*, where the they start to break down (Agre 1997). While the previous account of proceduralist meaning strove to include a complex conceptions of players, by definition, the focus remains on the game's processes. This means that proceduralist readings may not be the ultimate analysis for understanding how players relate to games, but as illustrated, they are still useful for understanding how games can be meaningful.

One approach to better account for players would be to loosen the imperative of comprehensiveness, reducing the emphasis on the game as mechanism, and focusing more on the cultural issues that surround games. Arguably, this is what most people in game studies are already rightfully doing by researching player communities, the philosophy of play, the psychological effects of games, etc. This is important work but rather than abandon the proceduralist focus on processes, this chapter presents an interpretive approach that does the opposite. In abandoning the study of *real* players, and embracing the sort of imagined players that *proper* proceduralist designers use to attempt to design for specific interpretations, we find a fruitful interpretive exercise that shows us how games *can* be meaningful rather than how they *are*.

The following is a discussion of what can be called *alien readings*. An alien reading is a well grounded interpretation of a game where the conclusions are not generated by an individual, but rather an imagined subject with alien considerations. Because proceduralist readings can be used to justify any number of interpretations, this exercise is able to generate many understandings of a game. This sort of interpretive exercise is not as outlandish as it might sound. With other forms of media, theorists have come up with ways to argue that media artifacts represent concepts that seem both very unlikely to be the creator's intent and for which it is unlikely most audiences would achieve the same interpretation (e.g. Marxist readings, queer readings, counter readings, etc.). Where a Marxist reading privileges observations and interpretations about class struggle, an alien reading privileges the proceduralist imperatives of comprehensiveness and coherence. These readings are called alien because they do not reflect the way most people actually engage games. The alien reader strives for *any* coherent understanding of a game so long as it accounts for the apparent observations about a game as mechanism. Alternative interpretive perspectives, such as the alien reader, allow creators and audiences to understand how media artifacts often unintentionally engage issues that are not immediately apparent. These types of readings are valuable to media studies as they demonstrate that media is able to be meaningful beyond the confines of our current conceptions.

Alien readings reveal to the theorist that there are new ways that games *can* be meaningful. The first step of this exercise is to assume that the game under consideration is full of procedural metaphors. One could think of an alien reading not just as a crazy or weird interpretation of a game that no one would ever have, but instead as a perspective that could at any point become a *normal* way that games are regarded as expressive. For instance, game designer Jason Rohrer created his game *Passage* to represent the life of a man using simple visuals and simple game mechanic metaphors (similarly to the games of *Game-O-Matic*). This game became very influential in the game industry, and inspired a movement of independent game designers to create similar mechanics-as-metaphor style games about personal issues. Before *Passage*, it would have seemed very strange to regard simple collision detection in a low fidelity game as representational of personal issues, but the success of *Passage* shows that that conventional wisdom was not correct.

Towards developing this idea, the following is an alien reading of a game that was very hard to interpret as it was not likely created with any particular representation in mind: *BurgerTime* (Data East 1983). While most people who play

BurgerTime would see it as a clone of *Donkey Kong* that was simply skinned with fast food imagery, the alien reader attempts to arrive at a comprehensive account of the game's processes and visuals. Interpreting *BurgerTime* provided a challenge to the proceduralist perspective and is a detailed example of how alien readings can reveal to the interpreter new ways that games can be meaningful. In the end, a comprehensive reading is only achieved by considering the gameplay of expert players: those who understand the rules of a game the most.

Reading ***BurgerTime***

BurgerTime (Figure 30) is a 1982 arcade game created by *Data East* that is still found in many arcades. Like other arcade games from that era, it is a 2D platformer structured as a series of levels with the overall goals being to advance through levels and achieve the highest score possible. Each level contains different arrangements of staggered platforms with ladders connecting platforms on different vertical layers. The player controls the movement of a chef being chased by pickles, eggs and hot dogs. Upon colliding with any of these three foods, the chef falls over and the player loses a life. To help prevent this from happening, the player has a limited ability to direct clouds of pepper which momentarily stun these enemies. Occasionally, stationary icons of French fries, ice cream or coffee appear on the game board which replenish the chef's supply of peppers.

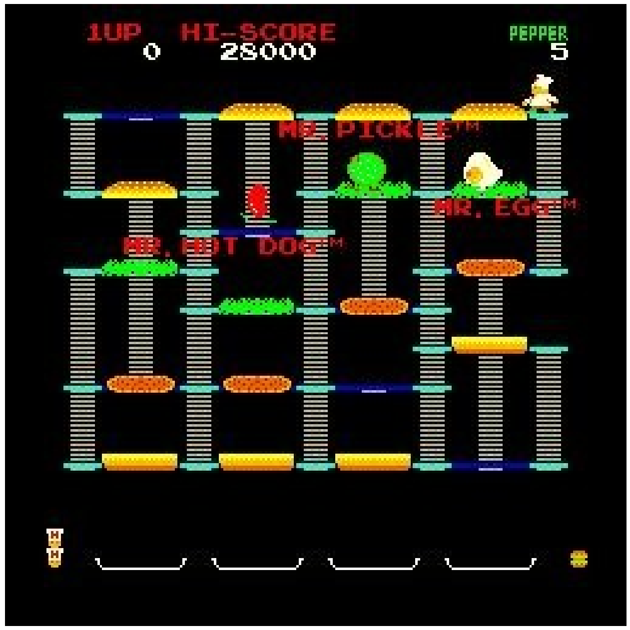

Figure 30 A screenshot from *BurgerTime* that shows the chef (top right), the plates (across the bottom), the burger parts (stacked vertically on platforms above the plates), and the enemy foods (labeled).

What makes *BurgerTime* stand apart from other classic arcade platformers can be found in each level's goal. Placed throughout the lower parts of the screen in each level are plates. On the platforms directly above each plate are various layers of burgers: buns, patties, tomatoes and lettuce. When the chef runs across the entirety of one of these *burger parts*, it falls to the platform below the one it was resting on. If there is a burger part on a platform directly below a falling burger part, both fall to the next platform, creating a cascading effect. When all of the burger parts fall on to all the plates below them, the player moves on to the next level, which contains different arrangement of platforms and burger parts.

Below are a few attempts at making sense of *BurgerTime* and its rules, dynamics, and aesthetics. The meaning derivations will not be as formal as in the examples in the previous chapter. This is partly for the sake of brevity and partly because the meanings proposed will not be strong enough to warrant the rigor.

First Attempts

Given that the goals are to complete levels and get a high score, and that forming all burger parts into complete burgers progresses the player to the next level, it is reasonable to assume that the player will attempt to run across the tops of all of the burger parts on the game board. Given that the player's avatar is themed as a chef and that the parts that are formed into burgers are themed as buns, tomatoes, lettuce and beef patties, it might be said that *BurgerTime* is a game about a chef preparing burgers.

To believe this meaning, the interpreter must accept the metaphor that running across the tops of burger parts is analogous to cooking or preparing the food. This metaphor is supported by the thematic mappings of the game as chefs are known to cook, and the various burger parts are the sort of thing that a chef would work with to create an entree: a burger. Furthermore, the relatively bland, themeless, platforms and ladders of the game board can be treated as merely supporting the ludic metaphor, and thus could arguably be rendered as invisible to the interpreter. Additionally, there is a thematic consideration in that the only means the chef has to collide with burger parts is to run over the top of them. Because of the unsanitary cultural associations with feet, it could be claimed that the game is about a chef preparing burgers in an unsanitary way.

However, this interpretation can be accused of employing the same sort of *selective* interpretation as Poole's *Pac Man* interpretation (see Chapter 4). The reading does not hold once more of the game is taken into consideration. Particularly, all considerations pertaining to the game's enemies are omitted and unaccounted for. If the chef's contact with burger parts is supposed to be understood as a metaphor for cooking, why does contact with the hot dogs, pickles and eggs cause the chef to look unhappy and the player to lose a life?

One might argue that the chef is only *supposed* to be preparing burgers, as if some undepicted customer ordered a burger and not a hot dog. To touch (cook) any other food would be a waste of time as the chef would not be preparing what was ordered. This interpretation relies on many assumptions about the game's diegesis that are hardly even hinted at and is thus an unconvincing meaning argument. Furthermore, it still doesn't address the antagonistic *behavior* of the enemy foods. Why do they try so hard to make the chef collide with them?

The following interpretation attempts to avoid such large leaps in reasoning while still trying to take a more comprehensive consideration of the game. Given that collisions between the chef and hot dogs, pickles and eggs causes the player to lose a life, it can be expected that players will avoid collision with these enemy foods. This and the enemy foods' movement behaviors that follow the chef's movements bring an anthropomorphic sense to the enemy foods. It *feels* like the enemies do not like the chef and thus do not want him to be successful. Because the chef is trying to prepare burgers, and the enemy foods do not want him to be successful, perhaps they do not *want* the burgers to be made.

Given that hamburgers, hot dogs, pickles and eggs are all popular American foods, one might argue that the enemy foods are competing with the hamburger for the chef's attention and thus their ability to fulfill a prepared item of food's purpose: to be eaten. Thus *BurgerTime* could be a game about the ubiquitous nature of the burger, and how other *neglected* foods must compete for their *plate share*. This (quite large) interpretive leap introduces questions such as why the enemy foods are volitional where the burgers lay still waiting to be prepared. While we could potentially conjure up answers to this question, the assumptions required to get to this point were perhaps already too unreasonable. Furthermore, much of the game is still left unacknowledged. Given this context, how does it make sense that pepper slows the competing foods? Why does colliding with the French fries give the chef more opportunities to bail himself out with these peppers? Why is it so hard for the chef to escape?

The above two attempts seem to confirm the intuition that *BurgerTime* is a nonsensical game. Of course this isn't surprising. The game was even advertised by emphasizing its absurdities:

> *Your job is to climb up the ladders and assemble an order of giant hamburgers. But you've got to do it fast because you're being chased by killer hot dogs, sour pickles and a very nasty fried egg. Good thing you've got your pepper shaker. One shake and they're stunned. But just make sure*

you don't run out of pepper. Because you know what happens then. You stop making lunch. And you start becoming it!

This description of the game makes no attempt to rationalize the peppers and even highlights the game's *ludonarrative dissonance* (Hocking 2007). In fact, the game itself almost appears to exist as a celebration of nonsense given its obscure subject matter and seemingly unrelated gameplay.

However, as shown above, aspects of this game do lend themselves to its theme. A chef is someone who manipulates ingredients to prepare meals. Likewise, *BurgerTime* also has the player take action to combine ingredients that are not typically eaten individually, into a whole, or complete, item of food. Regardless of the other disharmonious mechanics and themes, it seems reasonable to say that at least this relationship will *shine though* to players.

At this point, it could be that the only reasonable interpretation (i.e. one that many interpreters would agree on) is that the game is simply a poorly executed and incoherent representation of a chef cooking burgers. The following sections will demonstrate what was learned by forging forward in spite of this suspicion. By investigating *why* a more comprehensive interpretation does not seem possible, the limits of the proceduralist perspective are expanded.

Coherence, Roles and Meaning Derivations

The following section investigates the differences in function of interpretable observations, as well as the philosophical roots of meaning derivations. Through this deeper understanding of the proceduralist perspective, a way towards a comprehensive interpretation is revealed.

Dealing with Incoherence

As with all media artifacts, interpretations of *BurgerTime* will vary between interpreters. Even when socio-cultural considerations are treated as constant (i.e. a specific context is assumed), interpretations can vary wildly depending on where one focuses one's attention. If one solely considers the movement of *BurgerTime's* chef through the level, an easy argument could be made that the game is a representation of a chef running around and climbing ladders. Obvious observations such as these are taken for granted by most, but we should not forget that moving pixels on a screen are merely representations. In this case, the movement of an animated image of a chef running, moving along the top of a narrow stationary box (the platform) is understood as a representation of a chef running. Even in accepting this reasonable interpretation, one must omit violations of expectations ranging from why the platforms are seemingly suspended in midair to why the chef cannot move in three dimensions (as one would reasonably expect a *real* chef to be able to move).

For most, these minor departures from reality are not a problem. This is in part because the history of visual culture has prepared us to accept abstract images of humans moving as a metaphorical representation of their real world movement and also because the operational logics that underlie most classic arcade games are well suited for representing real world movement and collision by the nature of their spatial simulation (Mateas and Wardrip-Fruin 2009). The aspects of *BurgerTime* that do not support the representation of a chef running, such as the floating platforms, are not likely considered to be significant enough to factor into interpretations and most don't even notice them. These incoherent aspects are what *support* the higher level metaphors that the overall meanings are derived from. When it isn't clear where incoherent aspects of a game contribute to a coherent aspect, the interpretation loses its strength. For example, the attempt at interpreting *BurgerTime* as being about a competition between burgers and other foods became unbelievable because too many observations, such as why pepper *protected* the chef, did not contribute to the claimed interpretation.

As evidenced by the first, more modest, interpretation that *BurgerTime* is a game about preparing burgers, it is also possible that supporting metaphors are not always consistent. Where collisions between the chef and burger parts are in some cases understood as cooking, in other cases collision with the enemy foods causes the player to lose a life. Likewise, when the chef is standing on a platform (colliding with it), it doesn't make sense to think that he is preparing or *cooking* the platform and most wouldn't interpret it in that way. Thus, it cannot be the case that collision with the chef can always be a metaphor for cooking. Interpreters are willing to accept a certain degree of metaphorical inconsistency and detracting considerations that support metaphors that do contribute to the overall meaning. Just how much an interpreter is willing to accept depends on the individual, the game and the particular interpretation.

That procedural metaphors may be constructed out of aspects that do not necessarily support the overall meaning and need not *always* be applied consistently allows us to revisit the apparent roadblocks in our previous attempts at interpreting *BurgerTime*. However, before doing this, an exploration of the philosophical roots of meaning derivations will be presented to give direction toward a new interpretation.

Interpreting the Language of Videogames

A videogame can be understood as having a language. This perspective is alluded to in Crawford's notion that interactive systems are comprised of a *listen/think/speak loop* (Crawford 2003). A system listens and thinks when its state is modified by player interaction. How the system presents this change to the player is how a game speaks to its player. Making sure that games listen and speak clearly is a top priority of most game designers. Typically, this sentiment only applies to making the ludic aspects of the loop unambiguous. For example, a designer would certainly want to make clear to a player of *BurgerTime* how to control the chef's movement (with the left and right directions of the joystick), and that upon colliding with a hot dog, the player loses a life.

Different than considering the ways that a game speaks about its internal state, it is useful for this discussion to consider how a game communicates what it is trying to represent. Our conception of a player is that he is not solely trying to understand how to play the game, but he is also an interpreter trying to answer the question *what does the game mean?* And the system is *trying* to tell him. This player/interpreter collects observations and attempts to make sense of them. The game's utterances take the form of the components of a proceduralist reading: mechanics, dynamics, thematic mappings and aesthetics.

Given this, Wittgenstein's concept of the language-game can shed light on how an interpreter makes sense of his observations. In a language game, meaning is dependent on context and ultimately determined by how the utterances are *used* (Wittgenstein 1953). Much like how a move in Chess does not hold any sense outside of the context of a game of Chess, communicative acts only make sense in the context of a rule governed activity – a language game. Wittgenstein describes a particular language game, in which when person A says the name of some object, person B will find the object referenced by that word and hand it to person A. The meaning of person A's words are not found in the words themselves, but in the context of the rules that govern the exchange. For example, the command language enacts a rule in which when person A says a single word, person B is to find the object the word references and hand it to person A.

As an outside observer, interpreting the actions of person A and B is only possible once one has discovered the rules that they are enacting. In this particular example, this requires identifying the two roles, commander and assistant, as well as the way that the commander's words are to be interpreted - as short hand for "find the object referenced by my word, and bring it to me." Only at this point, can the actions of persons A and B make sense. In other words, to interpret person A's words as commands for person B to hand person A objects presupposes our ability to understand his words as being used in that way.

Does this mean that an interpretation of *BurgerTime* that claims that it is about cooking burgers presuppose the ability to understand it as being about cooking burgers? In some ways, this is precisely what is stated in the attempted meaning derivation above. Assuming that a chef's primary purpose is to cook, that burger parts are made complete when formed into a complete burger and that the chef's actions (collision) cause this to be the case, is what allows us to understand his actions as cooking the burgers. Just as one could not understand person A's and B's actions without understanding the language game they were engaged in, without an understanding of the *roles* and rules outlined above, *BurgerTime* cannot be interpreted.

Many videogames signal the roles and rules for interpretation before the game even begins. For example, understanding the abstract artgame *The Marriage* (Humble 2007) would be more difficult if the game were not titled *The Marriage*. For most, the title alone immediately sets expectations about two humans involved in a romantic relationship. Once the game begins, the player sees a blue and pink square, which clearly establishes that the two squares are meant to represent the two humans, as well as their gender. From this point of departure, the player can investigate the mechanics of the game, utilizing cultural assumptions about scale and transparency, to interpret the game's meaning.

The problem with *BurgerTime* is that the roles are not so well established. The title tells us little beyond that the game will involve burgers *now*. This insight is confirmed the moment the game begins, by the prevalence of the burger parts. Like

The Marriage, the game also establishes roles by its visuals. The presence of a chef sets up the expectation that he might cook food. The presence of food all around the game board appears to confirm this expectation but, as discussed above, the game complicates this by having mechanics that cause some foods to be the player's enemy.

Like Wittgenstein's command language, understanding *BurgerTime* requires us to have a preexisting idea of *how* the game's utterances are to be interpreted. Thus, our inability to comprehensively interpret *BurgerTime* is not necessarily indicative of the game having no comprehensive meaning, but it could be that we simply do not understand the rules of the game *enough*.

Achieving a Reading

That some aspects of a game can support the metaphors that build meaning, despite seemingly working against it, allows us to second guess our assumption of the role that the enemy foods are to play in the game's meaning. Rather than interpreting the chef's collisions with the various enemy foods as incoherent, given the assumption that collision with burger parts represents cooking, it is possible that the behavior and rules that send the player running from the enemy foods can *support* metaphors rather than be particular representations pertaining to cooking themselves. And as discussed above, it may just be the case that we have yet to consider, or presuppose, the correct framing from which *BurgerTime* can make sense.

A possible solution to the problem of how to interpret the hot dogs, pickles and eggs can be found in the same place the problem came from: the game's rules.

The following section describes how expert *BurgerTime* players understand the game in order to provide perspectives from which new framings can be considered. Finally, a comprehensive proceduralist reading of *BurgerTime* is proposed.

Learning from the Pros

On September 19th 2008, Bryan Wagner earned the world's highest score for *BurgerTime* on an arcade machine, with a score of 11,512,500 points**.** In an interview alongside *Mappy* (Namco 1983) world record holder Greg Bond, Wagner describes that the most important part of getting a high score on *BurgerTime* is to create *tight groupings* of the enemy foods (Tuttle and BearmanJosh 2008). By this, he means to manipulate the enemy foods into moving as a single entity by exploiting their chasing behavior. To group the enemies, the player can use peppers to stun an enemy, wait for another enemy to overlap with the stunned enemy and then use another pepper. Once the enemies start moving again, they will be moving as one entity (Figure 31).

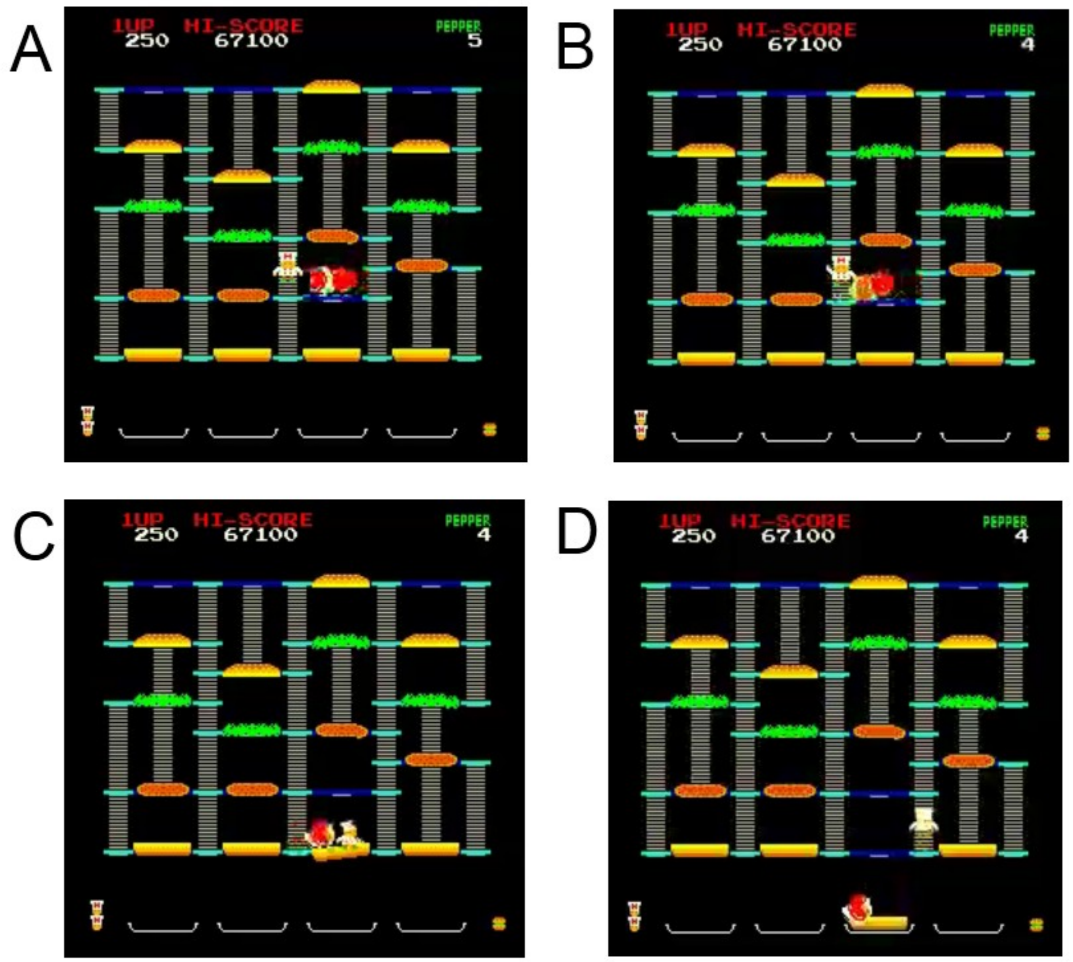

Figure 31 A and B demonstrate how a loosely packed group is made into a tight pack with peppers. C and D show how a tightly packed group can be dropped to a plate.

The interview also states that once the enemies are grouped, a strategy referred to as *dropping* becomes the best way to get a high score. If an enemy is standing on a burger part as the chef finishes running across the entirety of the same part, the enemy will fall along with the burger part to the platforms below until the burger part and enemy land on a plate and a large number of points are awarded.

Doing this when the enemies are grouped awards points for each enemy in the group (Figure 31).

The tightness of groups is important for both leading the group toward the burger part that the player is going to attempt to drop, as well as making it possible for the group and the chef to be standing on the burger part simultaneously, even if momentarily, without the group colliding with the chef. In the interview with Wagner, Bond describes how unstable the groupings are "When you make a drop on *BurgerTime,* the last thing you want to do is celebrate… because they're watching you…" (Tuttle and BearmanJosh 2008). By this, Bond is expressing the precarious nature of a group's cohesion. The only thing holding a group of enemies together, from not taking different paths from one another, is the player's precise understanding of the way they chase the chef. Because of this, even when the group *looks* tightly packed, and the player is about to drop the group, the player must still diligently monitor the way that they move the chef. One wrong move and the group will split, and with that the player throws away "a whole lot of points, and a whole lot of peppers" (Tuttle and BearmanJosh 2008).

Figure 32 *BurgerTime*'s world high score holder, Bryan Wagner (left), Mappy high score holder, Greg Bond (right bottom), and J.D Lowe, online *BurgerTime* evangelist (right top).

J.D. Lowe, another *BurgerTime* expert, posts his strategies for creating groupings for particular levels on his website for people to learn from (Lowe). He demonstrates that creating a grouping with all of the enemies on the game board requires diligent practice and a careful plan. His advice is to learn his strategies to begin with, then to eventually discover your own patterns for grouping. "Improvising while playing the game is what will make you a better player."

A (somewhat) Reasonable Comprehensive Reading

These expert perspectives enable new ways of understanding *BurgerTime*. Particularly, the enemy foods can be seen as assets, rather than enemies, as their antagonistic behavior enables opportunities to achieve the highest scores. Expert players of *BurgerTime* are not running from the enemy foods, instead they are leading them together into groups and onto burger parts. Also, the fact that the hot dogs, pickles and eggs are dropped onto the burger parts resting on the plate, where they disappear and award the player points, becomes of relevance for interpretation, as creating this situation becomes the focus of gameplay.

When interpreted in light of the game's theme of a chef preparing food, these new considerations finally provide a lens from which we can make *some* sense of the game. Creating groups, using *just the right amount* (not too much) pepper, can be interpreted as the *mixing* of ingredients. The dropping of these concoctions onto the plate can be seen as *seasoning* the burger with the mixture. The necessary patterns to group the enemy foods might be interpreted as *recipes*.

The two types of cooking, the assembling of the burger and the seasoning, are presented in ways that contribute to the interpretation. The assembling of the burger requires that the player methodically collide with each part, and can thus be argued as analogous to the craft of a *cook*. The seasoning, with its free form, artful and improvisational nature is more analogous to the art of being a *chef*.

To believe this interpretation requires that one understands the enemy status of the foods as supporting the metaphors that enable it. Their chasing behavior, and the fact that collisions cause the player to lose a life, must be seen working in the service of giving the player the ability to mix the foods into groups. The fact that collisions with enemy foods cause the player to lose a life has no place in this interpretation. However, if the collisions did not cause the player to lose a life, the player would only need to *not* move the chef and the foods would group on top of him. If this was the case, the game could not be said to be about the relationship between the artful seasoning of burgers and the pedestrian craft of assembling of buns, beef patties, tomatoes and lettuce.

Lessons from Reading *BurgerTime*

This study attempted many approaches at performing a proceduralist interpretation on a game that didn't make it easy. Performing this alien reading on a game that was not created with a coherent representation in mind faced the proceduralist perspective with many challenges. Grappling with these challenges furthered our understanding and ability to perform proceduralist readings. Particularly, the insights that many, possibly contradictory, levels of metaphor are at work simultaneously when a game produces representational meaning, along with the concept that to understand a game requires the presupposition of the *rules* of interpretation, inspired us to find the proceduralist reading for *BurgerTime* described above. As evidenced by the above, performing alien readings is beneficial to designers and theorists as they lead to the discovery of new expressive and interpretive affordances.

The observations and experiences of Wagner, Bond and Lowe can be compared to works from film studies which describe the *experience* of watching film. These foundational texts in film studies, such as Balasz' detailed discussion of the close-up shot (Balazs 1970), have greatly influenced scholarship on film. Experiential descriptions should be foundational to game studies as well. However, beyond

Sudnow's detailed account of becoming an expert at the classic game *Breakout* (Sudnow 1983), little work has been done in exploring the implications of rules as the player learns to understand them *deeply*. While games like *Passage* demonstrated that game mechanics can be expressive metaphors, this alien reading of *BurgerTime* demonstrated that rhetorically utilizing high level dynamics might be an expressive possibility for designers.

Striving to find a comprehensive and coherent interpretation of a game that seemed to resist all attempts was a very strange undertaking. Furthermore, while comprehensive and coherent, the conclusions likely do not reflect anyone's actual understanding of the game. However, this investigation into *BurgerTime* brought to light many insights. The point of the exercise was not to argue that *BurgerTime* should be regarded as a landmark expressive work, but instead to reveal how interactive artifacts contain so much potential for meaning that even an absurd game like *BurgerTime* can be argued to carry subtle and expressive ideas.

Future Work: Implementing Alien Readings

The open ended yet rigorous definition of proceduralist and alien readings lends itself to implementation in code. The following is a system and high level game description where a player is able to explore and take actions in a game world, and the system is able to interpret the gameplay and present the player with an alien reading. In this game, the player chooses simple daily actions, such as changing location, looking at objects, and various other mundane tasks, in an adventure style game. At the end of each day the system will produce a story about what happened during that day and why.

For this proposed implementation of alien readings, the architecture and game design are tightly coupled. Players of the game will take *actions*, which may or may not satisfy *desires* over the course of a day. At the end of a day, the player will be presented with a *story* that takes into account as much of the events of the day as possible, based on a library of *interpretation rules*. Actions are defined by weighted precondition rules for when it should be accessible to the player, various textual descriptions of it being performed, an amount of time it takes to perform and a set of desires it satisfies. Desires are similarly defined by a set of weighted precondition rules to determine whether it is appropriate, various textual descriptions of it being satisfied or not, and an amount of time that it will be present for. The game world is simply comprised of a set of locations and a time of day.

As the player is presented with desires, and either satisfies them or not by taking actions, a database of events is populated. Entries in this database include actions taken, actions not taken, desires present, and desires satisfied or not. All of these entries are associated with a time and place. When the end of the day occurs, the system then applies a set of interpretation rules. Interpretation rules query the database and assert new generated entries. These interpretations rules are authored to be very general and implicitly try to capture common sense notions about the nature of choice. For example, if action A is taken at the expense of desire B not being satisfied (it timing out), the system interprets these two actions as implying that the player would rather take action A than satisfy desire B. These interpretation rules encode cultural assumptions that are grounded in concrete observations about what the game's mechanism is able to produce. Below is short list of some other interpretation rules:

- Because the player chose to take action A, and another action that satisfied a desire B was present, the system could decide that the player chose action A *instead of* satisfy desire B.
- If the player never took an action that was available for a long time, the system could assume that he did not want to take that action.
- If the player did something multiple times at the expense of fulfilling desires that eventually time out, the system could infer that the player compulsively takes that action.
- If the player took an action, A, which enabled an action that he used to satisfy a desire, the system could infer that the player took action A as a means to satisfy the desire.
- The system could assume that the player does something reluctantly if he only takes the action right before a desire is about to time out.
- And so on…

Once the interpretation rules are applied, stories are scored for appropriateness based on a set of weighted precondition rules that query the database of events. The story with the highest score is selected to be instantiated. Stories carry with them templates that are filled in with the details of the events of the day. Each story can also be seen as an interpretation, as the contents of the templates often include additional statements about the feeling and intentions of the player that are not explicitly present in the database of events, but are *reasonable* to infer by the contents of the precondition rules. For example, if the player did not satisfy many desires during the day, the story may be structured around the assumption that the player had a bad day.

Because the interpretation rules are so generative, the database will become very large, and a large number of stories will be possible to tell about the player's gameplay. In the context of a meaning derivation, each summary of each story can be understood as a representational claim about the gameplay. The meat of the story comprises the supporting interpretations. The interpretation rules are the cultural influence. And the database of events before the interpretations is the results of the game as mechanism.

Being made in the tradition of critical technical practice, the process of creating this alien reading prototype and then playing it will inspire reflection about the proceduralist readings framework as well as suggest future research suggestions.

Chapter 6. Conclusion

This dissertation has presented several theories and artworks meant to inspire reflection about how videogames are procedurally about subjects. However, no single *knock down* theory was presented. As with every other form of media, interpretation is tangled up in issues of culture and subjectivity. These are not topics that can be resolved with definitive theories.

However, we can still learn qualitatively *more* about how games are about subjects through an investigative practice that seeks to find answers to these issues. Each chapter presented a lens that can help designers and players better understand how games are meaningful and why. Furthermore, the theories developed in each chapter were able to drive the creation of artifacts that advanced the state of the art in videogame technology while also furthering the theoretical discussions of how games are meaningful. A *critical technical practice* that combines humanistic investigation, artificial intelligence research and art making is productive of insights that can help further our understanding of how to create and interpret games with more intention.

Chapter 2 began with the arguably naive assumption that games can be representational through their rules alone, and from there a precise theory of how instantial assets relate to game rules was developed. This theory made explicit the implicit theories that arguably drove the creation of a subset of games that employ procedural rhetoric: newsgames and artgames. The micro-rhetoric theory argues that patterns of abstract gameplay mechanics have a space of rhetorical affordances and the player's pre-existing beliefs about what is instantially represented ultimately determine what the game is about. From there, the theory was formal enough to be implemented by running code on a machine which enabled a system that could generate games that represented ideas. This resulting artifact, *Game-O-Matic*, is a contribution in many ways. Firstly, creating it drove the theoretical investigation of how simple game mechanics can be about subjects. Secondly, the artificial intelligence research that went into implementing the micro-rhetoric theory is beneficial to the research community that is attempting to formalize and automate the game design process (see appendix). Additionally, as a tool *Game-O-Matic* enables people who have literally no knowledge of programming to create games about any subject they choose. Finally, creating and using *Game-O-Matic* revealed the margins of the micro-rhetoric theory and thus the implicit theories that drove the creation of many newsgames and artgames.

Chapter 3 developed a language to speak about how the complex processes of games are about subjects. First, a distinction was made between games that primarily represent because of beliefs about the instantial assets and games that represent because players narrate the abstract processes as icon signifiers of processes in the world (the instantial and simulative registers). Next, a theory of simulation representation was presented that presented games as a set of representational principles and conclusions that players learn about through experimentation. Given that, a discussion of the social simulation game *Prom Week* served as an example of how a complex theory of social interaction could be represented as a set of principles in a game that was enjoyable to play while also furthering the state of the art in interactive storytelling. Finally, an account was given of the persistent design problems encountered in trying to create *Prom Week* such that it represented a set of principles. This revealed that the simulation theory as presented did not adequately account for why players would recognize the procedural aspects of the game as intended and thus required further development.

In chapters Chapter 4 and 5, a theory of *proceduralist readings* was developed that described the context in which individual players interpret a game. It was argued that players create meaning independently through play separated from the creator's intent. This theory, when applied to the design process, highlights the importance of matching the design of the mechanism of a game with a player's cultural context such that the player will arrive at a desired interpretation. After exploring the many and varied ways players might understand the same game, the generative exercise of *alien readings* was presented. This showed how approaching games with uncommon considerations from the proceduralist perspective can lead to discoveries of new expressive and interpretive affordances for games. Future work was proposed that would systematize an imagined alien interpreter that would be able to interpret a player's gameplay as it was enacted. This system would be generative of the discoveries alien readings are intended to produce.

In conclusion, if the reader has gained both an appreciation for the potential of games to be meaningful because of their procedural nature as well as an appreciation for just how complex and difficult that can be, then this dissertation will have achieved its goal. With this understanding, it is my hope that we can better understand how to make and find meaningful interactive works that help us express and consider the process-oriented aspects of our lives.

Appendix

Game-O-Matic and the Videogame Generation Systems

Generating videogames is a relatively new field with few examples of combining basic rules into playable games. All of these works (including *Game-O-Matic*) create single-screen arcade-style games. The first two related works are focused on a formal specification that reliably produces playable abstract games, while the third provides a method for creating games with a sensible representation, the final paper is an earlier work on automatic game design.

Variations Forever is a game generator developed by Adam Smith and Michael Mateas (A. Smith and Mateas 2010) which generates games using a combination of answer set rules that define the game's rule set, space, controls, etc. Answer set programming offers random selection over a range of values, such as position or who is the player, yielding generative space. The generated games can be constrained to fall within certain mini-genres by adding to the rules. Variations Forever's games are selected from the solutions to the set of rules. *Game-O-Matic* also uses a rule-based approach, but the variations are selected based on a score against the narrative mapping. *Game-O-Matic* preconditions are not strict in order to provide greater variability between games or flexibility in accommodating unusual narrative maps. This may occasionally result in unwinnable games, but the genre of newsgames contains several examples of unwinnable games such as Gonzalo Frasca's *Kabul Kaboom*. These games may be interpreted as expressing a rhetoric of failure (Lee 2003). This type of game can be well suited for expressing certain ideas, so we allow for a small percentage (less than 5%) of unwinnable games, which will typically only be produced after generating several winnable games.

The ANGELINA system, developed by Michael Cook and Simon Colton (Cook and Colton 2011), like Variations Forever, generates abstract arcade games. The games are generated in 3 parts: the map, the layout (of entities), and the rule set (collisions, movement types, time limit, and score limit). Each part is evolved through many generations with separate fitness functions, and occasionally testing the fitness of the parts together. The map part is a maze-like arrangement of bricks on the grid that can impede or encase entities. This is a component which *Game-O-Matic* currently lacks. As of now, we rely on constraints to the movement of entities, and the only environmental factor is the border of the screen. Although sometimes entities will have components that cause them to behave like walls. Still, the genetic algorithms used in ANGELINA set it well apart from *Game-O-Matic*; the independent fitness function produce variations in the games, and the virtual play throughs ensure playable games.

Game-O-Matic is most similar to Nelson and Mateas' work on generating skins for games with very simple mechanics (Nelson and Mateas 2007). Given a verb and a noun, like "shoot a duck," Nelson and Mateas used a common sense knowledge base to find an appropriate skin to apply to A and an appropriate game mechanic from their library of game mechanics. For example, the system would select a game where the player controls a set of cross hairs and tries and click on a frantically moving around duck. The system chose this mechanic and skin as it was more appropriate to select than something like shooting a piano would be, as a bird is something that can be shot. As will be described below, *Game-O-Matic* relies on the user to supply sensible relationships and does not prevent strange pairings, but, it is able to combine multiple game patterns. However, using a similar approach to and putting restrains on what valid verbs are could be an interesting future direction.

Julian Togelius and Jürgen Schmidhuber's foundational work on generating videogame rules evolves games using a fitness function built on theories of fun and learning (Togelius and Schmidhuber 2008). The generator needs to play the games to evaluate the fitness function, so controllers are evolved as well. By changing a few parameters regarding the consequences of collision, setup and behavior of entities, and win/lose conditions, their system can generate *Pac-Man*-like games. *Game-O-Matic* avoids the need for evolving games by starting from a user supplied concept map and mapping those concepts to representative game mechanics, the permutations of which are scored to fit various videogame tropes, and the result is typically playable.

Social Simulation Related Work

There are many systems in the domain of modeling interactions between characters or virtual humans based on cognitive or psychological models that reason over competing capacities of a prescribed set of desires (Aylett, Louchart, Dias, Paiva, & Vala, 2005; Marsella & Gratch, 2009; Si, Marsella, & Pynadath, 2009). *CiF* is an implementation of an alternate, norms-based vision of modeling what characters should be doing. This approach gives characters the affordance to reason over what desires are appropriate for the situation and then to negotiate between those relevant desires (Evans, 2009). Through modeling normal patterns of social behavior with a context of general social norms, the amount of story space covered by each authoring effort is increased over that of authoring for a single social state.

Narrative generation systems (Lebowitz, 1984; Meehan, 1976; Turner, 1994) model enough of a story world to create stories. In comparison, *CiF* does not attempt to model an entire story world. Instead it deeply models the myriad of considerations necessary for a character to follow norms during social interactions. As such, *CiF* is meant to be the social reasoning component encompassed by a narrative generation system.

CiF represents a different take on computational social behavior by focusing on deeper models of interpersonal behavior. This stands in contrast to the more common agent-based approaches that rely on computational economics or artificial life (Langton, 1995). Instead of starting with a foundation similar to agent-based approaches, *CiF* is based on dramaturgical analysis (Goffman, 1959) and models character behavior primarily through socially normative pressures found in everyday life.

In comparison to hierarchical task networks (Erol, Hendler, & Nau, 1995) and behavior trees (Isla, 2005), the operators, or patterns of social behavior, in *CiF* make use of larger sets of domain knowledge to judge their appropriateness for the current context. Instead of encapsulating domain knowledge implicitly in hierarchically layered operators or behaviors using a small number of (possibly procedural) pre or post conditions, *CiF* chooses characters' behaviors based on all applicable rules in a large rule-base that encodes normal social behavior authored for a particular story world.

The Sims 3 is an example of a culturally influential and commercially successful video game that has a highly dynamic social space (Electronic Arts, 2009). Its characters, known as Sims, have traits and desires that inform the social practices (social norms and cluster of expectations) they perform (Evans, 2008). Two major differences between the systems are in the complexity of the statements of social norms and the use of history in those statements. *CiF* provides a level of complexity similar to first order logic in that parties outside of the social exchange can be referenced (*x* is cheating on *y* if *x* and *y* are dating and there is a character *z* also dating *x*) where *The Sims 3* can only reference the two characters in an interaction. *CiF* also allows for both back story (history of the story world before the player is involved) and play history to be used in reasoning and social exchange performance, a feature completely missing from *The Sims 3*. These richer rules found in *CiF* allow for each individual authoring effort to be more potent while enabling an entire new set of social reasoning to the characters.

Bibliography

Aarseth, Espen. 1997. *Cybertext: Perspectives on Ergodic Literature*. Baltimore, Maryland: The Johns Hopkins University Press.

Agre, Phillip E. 1997. *Computation and Human Experience*. Cambridge University Press.

Balazs, Bela. 1970. *Theory of the Film*. Dover Publications.

Berne, Eric. 1964. *Games People Play: The Basic Handbook of Transactional Analysis*. Seventh Im. New York: Ballantine Books.

Bogost, Ian. 2006. "Playing Politics: Videogames for Politics, Activism, and Advocacy." *First Monday 11* (9 Special Issue no. 9).

———. 2007. *Persuasive Games*. Cambridge, MA: MIT Press.

———. 2008. "The Rhetoric of Videogames." In *The Ecology of Games: Connecting Youth, Games, and Learning*, ed. Katie Salen, 117–140. MIT Press.

———. 2009a. "Persuasive Game: The Proceduralist Style." *Gamasutra*. http://www.gamasutra.com/view/feature/3909/persuasive_games_the_.php.

———. 2009b. "Videogames Are a Mess." http://www.bogost.com/writing/videogames_are_a_mess.shtml.

Cantwell Smith, Brian. 2010. *Age of Significance*. Self Published.

Chandler, D. 2007. *Semiotics: The Basics*. Routledge.

Cook, M., and S. Colton. 2011. "Multi-faceted Evolution of Simple Arcade Games." In *IEEE Conference on Computational Intelligence and Games*.

Crawford, Chris. 1982. "The Art of Computer Game Design." In .

———. 1986. *BALANCE OF POWER International Politics as the Ultimate Global Game*. Microsoft Press.

———. 2003. *On Game Design*. New Riders Publishing.

Ebert, Roger. 2005. “Why Did the Chicken Cross the Genders?” http://www.rogerebert.com/answer-man/why-did-the-chicken-cross-the-genders.

Eladhari, Mirjam Palosaari, and Michael Mateas. 2008. “Semi-autonomous Avatars in World of Minds.” In *Proceedings of the 2008 International Conference in Advances on Computer Entertainment Technology - ACE ’08*, 201. New York, New York, USA: ACM Press. doi:10.1145/1501750.1501798. http://portal.acm.org/citation.cfm?doid=1501750.1501798.

Fagone, Jason. 2008. “The Video-Game Programmer Saving Our 21st-Century Souls.” *Esquire*. http://www.esquire.com/features/best-and-brightest-2008/future-of-video-game-design-1208.

Frasca, Gonzalo. 2001. “Simulation 101: Simulation Versus Representation.” http://www.ludology.org/articles/sim1/simulation101.html.

Friedman, Fred. 1996. “The Semiotics of SimCity.” *First Monday* 4.

Gingold, Chaim. 2009. “Human Play Machine.” In *Game Developer’s Conference*. http://levitylab.com/cog/.

Goldberg, Marty. 2006. “Pac-Man: The Phenomenon: Part 1.” *Arcadegaming.us*.

Hocking, Clint. 2007. “Ludonarrative Dissonance in Bioshock.” http://clicknothing.typepad.com/click_nothing/2007/10/ludonarrative-d.html.

Huber, William. 2012. “The Semiotic Conditions of Videogame Authorship”. Forthcoming PhD Disseration. University of California at San Diego.

Humble, Rod. 2006. “Game Rules as Art.” *The Escapist*. http://www.escapistmagazine.com/articles/view/issues/issue_41/247-Game-Rules-as-Art.

———. 2007. “The Marriage.” http://www.rodvik.com/rodgames/marriage.html.

Hunicke, Robin, Marc Leblanc, and Robert Zubek. 2004. “MDA : A Formal Approach to Game Design and Game Research.” In *Challenges in Game AI Workshop Nineteenth National Conference on Artificial Intelligence*.

Johnson-Laird, Phillip. 1981. “Mental Models in Cognitive Science.” In *Perspectives on Congnitive Science*, 147–191. Norwood, NJ: Ablex.

Kennedy, Helen. 2002. “Lara Croft: Feminist Icon or Cyberbimbo? On the Limits of Textual Analysis.” *Game Studies* Volume 2 (issue 2).

Lakoff, G, and M Johnson. 2003. *Metaphors We Live By*. University of Chicago Press. http://books.google.com/books?id=BHPiugAACAAJ.

Lee, S. 2003. "I Lose, Therefore I Think." *Game Studies* (3).

Lowe, J.D. "J.D. Lowe's Burgertime Arcade Strategies Levels 1,2, and 3." http://www.burgertime.info/html/strategies.html.

Lowensohn, J. 2010. "GDC: What's Next for Video Game AI?" *Web Crawler - CNET News*. http://news.cnet.com/8301-27076_3-20000266-248.html.

Lycerius. 2012. "I've Been Playing the Same Game of Civilization II for Almost 10 Years. This Is the Result." *Reddit*. http://www.reddit.com/r/gaming/comments/uxpil/ive_been_playing_the_same_game_of_civilization_ii/.

Mateas, Michael. 2001. "Expressive AI: A Hybrid Art and Science Practice." *Leonardo: Journal of the International Society for Arts, Sciences, and Technology* 34 (2) (April): 147–153. doi:10.1162/002409401750184717. http://www.mitpressjournals.org/doi/abs/10.1162/002409401750184717.

———. 2003. "Semiotic Considerations in an Artificial Intelligence-Based Art Practice." *Dichtung-digital* (3).

———. 2006. "A Preliminary Poetics for Interactive Drama and Games." In *First Person: New Media as Story, Performance, and Game*. The MIT Press.

Mateas, Michael, and Noah Wardrip-Fruin. 2009. "Defining Operational Logics." In *Digital Games Research Association (DiGRA)*.

Mccoy, Josh, Michael Mateas, and Noah Wardrip-fruin. 2009. "Comme Il Faut : A System for Simulating Social Games Between Autonomous Characters." In *Digital Arts and Culture Conference*.

McCoy, Josh, Mike Treanor, Ben Samuel, Noah Wardrip-Fruin, and Michael Mateas. 2011. "Comme Il Faut: A System for Authoring Playable Social Models." In *Artificial Intelligence and Interactive Digital Entertainment*.

Molleindustria. 2009. "Kosmosis." http://www.molleindustria.org/kosmosis/kosmosis.html.

Namco. 1983. "Mappy."

Nelson, Mark J. 2012. "Sicart's 'Against Procedurality'." http://www.kmjn.org/notes/sicart_against_proceduralism.html.

Nelson, Mark J, and Michael Mateas. 2007. "Towards Automated Game Design." In *In AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, 626–637.

Poole, Steven. 2000. *Trigger Happy*. New York, NA: Arcade Publishing Inc.

Pratt, Charles. 2012. "Players Not Included." http://gamedesignadvance.com/?p=2409.

PushButton. 2011. "PushButton Engine." http://pushbuttonengine.com/.

Ransom, John Crowe Ransom. 1938. "Criticism, Inc." In *The World's Body*. Charles Scribner's Sons.

Reiss, S. *The Normal Personality*. Cambridge, MA: Cambridge University Press.

Richards, I A, and J Constable. 1936. *The Philosophy of Rhetoric*. Routledge. http://books.google.com/books?id=bRj-QgAACAAJ.

Rohrer, Jason. 2007. "What I Was Trying to Do with Passage." http://hcsoftware.sourceforge.net/passage/statement.html.

Sicart, Miquel. 2011. "Against Procedurality." *Game Studies* 11 (3).

Simon, Herbert, and Allen Newell. 1962. "Simulation of Human Thinking." In *Computers and the World of the Future*, 94–114. Cambridge, MA: MIT Press.

Smith, Adam, and Michael Mateas. 2010. "Variations Forever: Flexibly Generating Rulesets from a Sculptable Design Space of Mini-Games." In *IEEE Conference on Computational Intelligence and Games (CIG)*.

Smith, Gillian, Alexei Othenin-Girard, Jim Whitehead, and Noah Wardrip-Fruin. 2012. "PCG-Based Game Design: Creating Endless Web." In *Foundations of Digital Games Conference (FDG 2012)*.

Squire, Kurt. 2004. "Replaying History: Learning World History Through Playing Civilization III". Indiana University.

Starr, Paul. 2001. "Seductions of a Sim: Policy as a Simulation Game." *The American Prospect*, December.

Stenros, Jaakko, and Annika Waern. 2011. "Games as Activity: Correcting the Digital Fallacy." *Videogame Studies: Concepts, Cultures and Communication*.

Sudnow, David. 1983. *Pilgrim in the Microworld*. Warner Books.

Taylor, T.L. 2006. *Play Between Worlds*. MIT Press.

Togelius, Julian, and Juergen Schmidhuber. 2008. “An Experiment in Automatic Game Design.” In *IEEE Symposium on Computational Intelligence and Games (CIG)*.

Treanor, Mike, and Michael Mateas. 2009. “Newsgames: Procedural Rhetoric Meets Political Cartoons.” In *Digital Games Research Association - DIGRA. 2009*.

Tuttle, Ross, and BearmanJosh. 2008. “Burgertime-ology (with Bryan Wagner).” http://youtu.be/A-FcPb2kclU.

Wardrip-Fruin, N. 2009. *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Mit Press.

Wark, McKenzie. 2004. *A Hacker Manifesto*. Harvard University Press.

Wilson, Douglas. 2012. “Designing for the Pleasures of Disputation -or- How to Make Friends by Trying to Kick Them!” IT University of Copenhagen.

Wittgenstein, Ludwig. 1953. *Philosophical Investigations*.