The Micro-Rhetorics of *Game-O-Matic*

Mike Treanor, Bobby Schweizer, Ian Bogost, Michael Mateas
University of California Santa Cruz, Expressive Intelligence Studio
Georgia Institute of Technology, Newsgames
{mtreanor, michaelm}@soe.ucsc.edu
{schweizer, ibogost}@gatech.edu

ABSTRACT

Micro-rhetorics are the representational units of meaning that emerge from the rhetorical affordances of videogame mechanics, abstract gameplay patterns, and thematic depiction. This paper explains the concept of micro-rhetorics, how game dynamics can be interpreted, and how designers can make use of game mechanics to express ideas through simple videogames. This theoretical framework is informed by the design of *Game-O-Matic*, a videogame authoring tool that generates games to represent ideas. It takes a network of basic relationships between actors and assembles simple arcade-style game mechanics into videogames that are able to make arguments and depict ideas.

Categories and Subject Descriptors

K.8.0 **[Personal Computing]:** General – Games. I.2.4 [Artificial Intelligence]: Knowledge Representation Formalism and Methods – Representations (procedural and rule-based).

General Terms

Design, Theory

Keywords

Game interpretation, game design, procedural rhetoric

1. INTRODUCTION

The ability of videogames to express ideas and meaning has been extensively discussed. Most closely related to our work, the proceduralist interpretation of meaning describes how the dynamics of systems are expressed through rules and procedures. In previous work, we have described meaning derivations—a method for interpreting games with graphical logics. In a proceduralist reading, the mechanic-dynamic-aesthetic (MDA) framework [5] is modified and expanded to explicitly include theme and representation [12]. By following the process of a meaning derivation, designers and players alike can make formal claims about a game's meaning. We are not arguing that a game has a single correct interpretation; rather, the validity of an interpretation hinges on the effectiveness of the argument made in the meaning derivation.

FDG '12, May 29-June 1, 2012 Raleigh, NC, USA.

Game-O-Matic, a Knight News Challenge funded collaboration between the Georgia Institute of Technology and the University of California at Santa Cruz [4], is a generator that can generate simple games based on input that lists objects, actors and their relationships. *Game-O-Matic* addresses a problem facing newsgames [3][10]: 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.

Though designed for newsgames, *Game-O-Matic* can be expanded into other domains. Though currently limited to mostly graphical logics (based on collisions, movement, position, and displays of meters and score) [6] the framework that it is built upon could conceivably handle any sort of dynamic model. It could depict the battle for consumer attention in the mobile phone market, weather patterns, 401K plans, or Heidegger's philosophy of enframing. These are made possible by understanding how games are able to represent ideas through detailed mechanics and thematic choices. *Game-O-Matic* assembles micro-rhetorics based on nouns that relate to each other with verbs that have been thematically skinned with graphical elements.

Often, 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 of imprecise targeting and the process of the mourning civilian turning into an angry terrorist.

Game-O-Matic utilizes explicit reasoning about micro-rhetorics to create games that reasonably represent specified relationships between objects. In this paper we describe the micro-rhetoric design framework, how *Game-O-Matic* interprets input concept-maps, and how these pieces are assembled into coherent games.

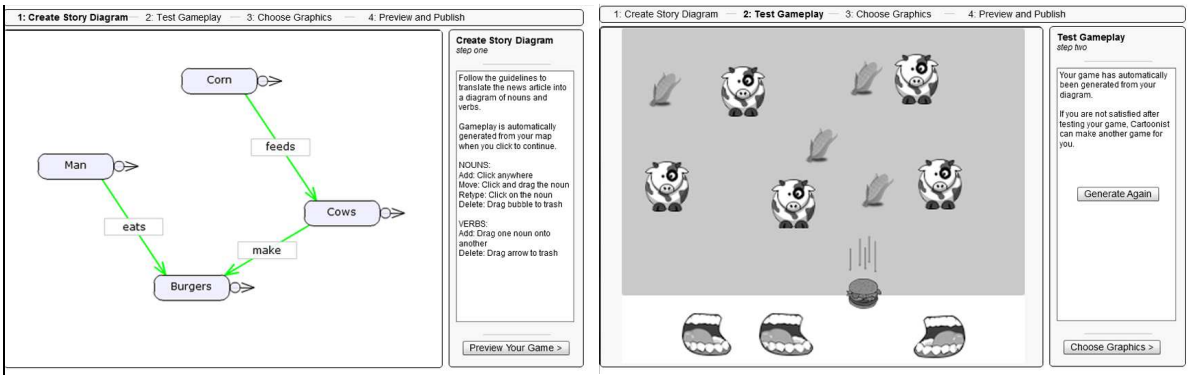

Figure 1. *Game-O-Matic*'s concept map interface (left) and a game that was generated from it (right).

2. MICRO-RHETORICS

What makes us believe that Pac-Man is eating pellets or that the player's missile in *Missile Command* is defending the planet? These easily taken for granted observations become difficult to design for when creating a videogame to represent a specific idea. Unlike static visual media, such as film and comics, the interpretable content of a videogame is generated by the interaction between its rules and players. Because of this, we believe it is crucial to carefully consider a game's rules when addressing representation.

Consider *Space Invaders*. 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.

However, this level of description still takes much for granted. Why do we believe that white lines that appear at the center of the objects and either moving up toward the aliens or down toward the player are bullets of some sort? Why do we ascribe intentionality to the alien's movement?

The representational power of videogames lies in the answers to these questions. This section describes the concept of micro-rhetorics—patterns of game mechanics and beliefs about the instantial assets that can be said to coherently represent an idea—which form the foundation of representation.

2.1 Interpreting Game Mechanics

While instantial assets and cultural beliefs may influence a player's choices, a game's mechanics are ultimately what restricts interaction and interpretation of the entities on the screen. Aarseth uses the term ergodic to describe the work required by the player to advance the system. 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 [1]. In short, interpretation necessarily involves considering a game's interactivity and thus the rules that govern it.

The first component of a micro-rhetoric are the game mechanics that represent a particular idea. Game mechanics are defined abstractly to separate the culturally assumed behavior of the represented objects from the hardcoded, material behavior written into the software. For example, the micro-rhetorical representation of harm comes from a game mechanic that can be described as, "the collision between entity A and entity B causes the removal of entity B." We have previously adopted a formal but simple logical notation: collision(A,B)→remove(B) (the event on the left of the arrow causes the event on the right). When A and B are instantiated—having been given concrete representations—this simple micro-rhetoric is complete. For example, if we were to imagine A as a shoe, and B as an ant, it is reasonable that one would believe that the shoe was harming the ant.

The game mechanics listed in a micro-rhetoric should be only those necessary to produce the events supporting the assumed interpretation. For example, if our simple "harms" micro-rhetoric were to involve the additional mechanic that the ant chases the shoe, we would no longer be able to say that the pattern only represents that *shoe harms ant*, as it would now seem as though the ant desires to be harmed given its suicidal trajectory toward the entity which will ultimately destroy it (a different, more nuanced micro-rhetoric).

2.2 Instantial Assets

Abstract mechanics do not represent concrete ideas in themselves. For example, the collision(A,B)→remove(B) mechanic, could be used to represent A harms B, A makes B invisible, A eats B, or A catches B.

The interpreter's beliefs about instantial assets determine how a set of abstract mechanics are understood. For our purposes, instantial assets refer to a game's static visuals and sounds. 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 squishing the ant. Whereas if A was a bunny and B was a carrot, the interpreter would understand collision(bunny,carrot)→remove(carrot) as the bunny eating the carrot.

The different common-sense beliefs about bunnies, shoes and ants can completely change what an abstract set of game mechanics can be said to represent. Previous work interpreting the abstract game mechanics of Activision's *Kaboom!* demonstrated that is possible to represent ideas like protection, theft, and kidnapping

in that game by keeping the mechanics fixed while only changing the game's visuals [9].

Groups of game mechanics can be said to have *rhetorical affordances*. Rhetorical affordances are the opportunities for representation made available by the rules that govern the relationship between objects and processes in a system. The meaning that is being selected from a set of possible meanings afforded by a game mechanic 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, stealing, etc.

The definition of a micro-rhetoric also contains the interpreter's assumed beliefs about the instantial assets and the relationships between them. For example, because we believe that an ant is vulnerable to shoes, we believe that collision(shoe,ant)→remove(ant) would represent killing, while because we believe carrots are edible, and have been raised to believe that bunnies desire carrots, we interpret collision(bunny,carrot)→remove(carrot) to represent eating.

2.3 Meaning Derivations

Each micro-rhetoric can be justified using a meaning derivation. A meaning derivation is a hierarchical, structured graph that explicitly states what an interpreter believes about a segment of gameplay and why. In previous work, meaning derivations were defined to be constructed out of three types of considerations: Code, Cultural and Interpretive [12]. Statements about the code can be understood as being the same as the abstract game mechanics defined in 2.1.

Cultural considerations are axiomatic assumptions that are true for a group of people. In other words, an interpreter either belongs to that group and believes them, or doesn't. 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.

The interpretive considerations are where the bulk of a meaning derivation lies. All interpretive considerations are constructed from code, cultural or other interpretive considerations. In other words, it is not possible for an interpretive consideration to be a leaf in a meaning derivation graph (see figure 2 below).

Interpretive considerations are of four types: dynamic, thematic, aesthetic and representational.

Dynamics are "the run-time behavior of the mechanics acting on player inputs and each others' outputs over time" [5]. Dynamics describe what happens during run time. For example, if the code hidden from the player defines that B is removed upon a collision with A, they will only understand that A destroys B once that event occurs. More complicated dynamics are emergent and often unpredictable.

Thematic statements—beliefs about the instantial assets—are almost entirely grounded in cultural assumptions or other interpretive statements. For 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, the appearance of pursuit could also be used as further support that red is a villain of some sort. Thematic considerations need not require that the interpreter justifies their beliefs, but relying wholly on assumptions about the visual rhetoric of instantial assets may weaken an argument.

Aesthetic considerations are used 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").

Finally, representational considerations are statements about what a segment of gameplay means in a symbolic, representational sense. For micro-rhetorics, these are the target of a meaning derivation, though meaning derivations could be created targeting conclusions about a game's aesthetics, theme or dynamics as well.

Meaning derivations 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.

2.4 Forming Micro-Rhetorics

The following section demonstrates several simple micro-rhetorics and discusses how the claimed meanings can be derived. Each micro-rhetoric consists of a collection of mechanics, as well as the assumed thematic, dynamic and aesthetic interpretations that are necessary to establish the micro-rhetoric's representation. Of course, the validity of each micro-rhetoric is informed by an interpreter's culture.

Like any complex argument, any assumptions made when creating a game will open it up to different interpretations. Implicit interpretive observations may either be ignored by the player or filled in with interpretations unintended by the author. For example, Molleindustria's *McDonald's Videogame* is intended to be about the evils of globalization and the underhanded tactics required to run a multi-national corporation. But an alternative reading of the difficulties and hardships of trying to keep a business running may emerge from players unsympathetic to the aims of the game [3].

This example shows how the more micro-rhetorics employed, and thus the more complex a meaning derivation gets, the less sure a designer can be that players will interpret the game as desired. This is not necessarily a problem, and is certainly not unique to videogames. Micro-rhetorics can be seen as analogous to the paint strokes of a painter. It is up to the designer to assemble/sequence them beautifully or coherently.

2.4.1 A destroys B

Mechanics:

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

Theme

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

This is one of the simplest micro-rhetorics, and one that is common to almost all classic arcade games. Simply, A collides

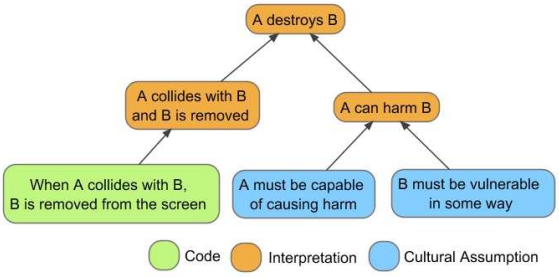

Figure 2. A graphical representation of a meaning derivation for A destroying B. In it, three interpretive assumptions are shown.

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 be able of causing 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, the instantiated micro-rhetoric represents that an axe destroys 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.

Figure 2 illustrates this micro-rhetoric's meaning derivation, and explicitly illustrates the three *interpretive leaps*, the assumed reasoning of an interpreter (arrows between the nodes), that underlie the representation that A destroys B.

2.4.2 A needs B

Mechanics:

- A's physical size decreases over time
- When A collides with B, A's size increases

Dynamics:

- If enough time passes, A will no longer be visible on the screen

This example demonstrates how even without thematic considerations, a micro-rhetoric's representation can be argued using only abstract mechanics [2]. This results from the tight coupling between the meaning of the word *need* and what the mechanics make happen on the screen. The word *need* begs the question, "what is the consequence of A not having B?" As described, the mechanics provide that answer: A needs B in order to exist in the field of view. This results from the dynamic that if enough time passes, A will cease to be on the screen anymore.

Though this micro-rhetoric doesn't require any thematic constraints, it is possible that an interpreter will employ their own thematic considerations. This micro-rhetoric will likely create a situation on the screen that represents much more than A needing B. For example, if the interpreter considered B to be harmful to A, the game would represent a sort of tragic and hopeless view of the existence of A in which the only way to exist on the screen is to collide with something harmful. While the notion that A needs B is still present in this interpretation, the additional thematic considerations have altered the player's interpretation considerably.

2.4.3 A attacks B

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:

- 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. Even if the abstract shape was not small, it is likely that the cultural convention that videogames often have entities shooting bullets leads most to understand this collection of mechanics as shooting.

An additional mechanic that A follows B would make this micro-rhetoric more convincing as without it one could argue that A does not intend to harm B. The question of perceived volition in an entity's behavior was found to be one of the main thematic considerations in the previous work with *Kaboom* [9]. Rather than require this to be a mechanic, we have made the definition more general, and moved this consideration into the interpreted consideration involving theme and dynamics.

2.4.4 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*, where a region on the screen protects the player's ships from the enemy's bullets. This micro-rhetoric contains the *destroys* micro-rhetoric from 2.4.1. 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 four examples hardly break the surface of what is possible to represent in 2D arcade-style games.

3. Building Games with Micro-Rhetorics

The previous section describes how game mechanics and dynamics are able to depict relationships between entities. Micro-rhetorics are given additional meaning when combined with each other and other dynamics.

Game-O-Matic, a videogame generator that creates games from relational input, uses micro-rhetorics to assemble its games (figures 1 and 3). Rather than give a detailed system description, an overview of *Game-O-Matic*'s function and the kind of games it generates illustrates how meaning can be authored by combining micro-rhetorics.

3.1 *Game-O-Matic*

The input interface of *Game-O-Matic* is modeled after a concept map with nodes (actors/nouns) linked by arrows (relationships/verbs). Users fill in the bubbles and draw arrows between them to represent the dynamic they wish to portray. Nodes (the circles) can be filled with any noun, be it proper like "White House" or conceptual like "lending practices." A node can have several arrows coming out of it, pointing at other nodes while others point back at it. Arrows are assigned transitive verbs such as "A neglects B."

After creating a concept map, the user clicks a button to generate a game, which is immediately displayed on screen and ready for the user to play. If they decide the game does not adequately represent their concept map, they can click another button to immediately generate another game. The generated games represent game entities using circles labeled with nouns from the concept map. When the user decides they like the game that has been generated, they can add artwork from a provided library or upload their own. Satisfied with their creation, a final button click will export the game and its assets for users to embed on their own website.

Game-O-Matic makes use of a library of micro-rhetorics that are related to the verbs input by the user. Because there is no way to enforce that a player will produce a given dynamic, carry a particular belief about the instantial assets, or have a particular aesthetic response, each of *Game-O-Matic's* micro-rhetorics only specify which mechanics should be present to represent a given relationship.

For every verb *Game-O-Matic* supports, there are many micro-rhetorics. For example, in the relationship "A attacks B," the micro-rhetoric for attack must involve some sort of harm which, using graphical logics, can be depicted as B being removed on collision, slowed down or frozen in place, or being prevented from carrying out its assigned actions. Not only does attack imply harm, it also implies the process is active. A may physically collide with B, it may shoot projectiles at it, or it may harm B indirectly through a third party.

When *Game-O-Matic* interprets a concept map, it creates entities from the nodes and assigns them game mechanics based on the micro-rhetoric of the verb. Given that concept maps are composed of multiple verbs and multiple mechanics for each of those verbs, there are many possible games that *Game-O-Matic* can create from a given input. As it generates games, it is able to try different combinations of mechanics which in turn produce different rhetorical interpretations. Because they are quickly generated, the author is able to cycle through interpretations to find one that best aligns with their intention. An important consequence of this process of generation and examination is the serendipitous nature of unexpected interpretations. *Game-O-Matic* is not merely an

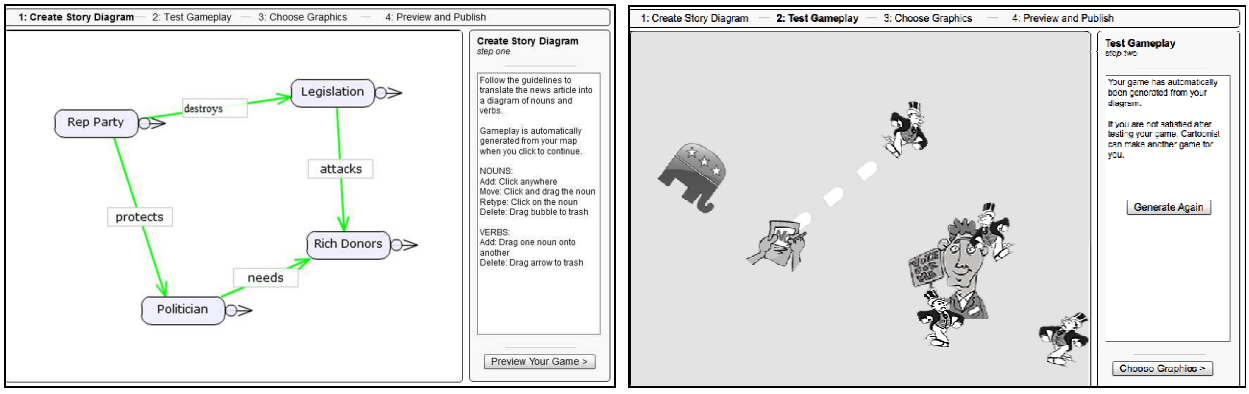

Figure 3. With the concept map (left) and representational input from the user, *Game-O-Matic* generated a game (right) where legislation shoots at rich donors, which move toward the presidential candidate to stop him from shrinking. The Republican elephant chases the legislation in an effort to stop the legislation.

authoring tool, but a way of thinking through relationships.

3.2 Testing Micro-Rhetorics

For this example, we will use a hypothetical current event about Republicans in the House of Representatives blocking legislation that would limit the size of campaign contributions. Someone who is sympathetic to the Republican Party might see this situation as the Republicans *protecting* a Republican presidential candidate and represent the situation in *Game-O-Matic*'s concept map form as shown in figure 3.

Game-O-Matic constructed the game in figure 3, using the micro-rhetorics defined in section 2.4, to represent a situation where Republicans protect a Republican presidential candidate from legislation that would limit campaign finance. First, it applied the *needs* micro-rhetoric (2.4.2) between the Republican Presidential Candidate and Rich Donors (meant to represent campaign financers) to represent that the Presidential Candidate needs the Rich Donors:

Mechanics:

- The Republican Presidential Candidate's scale decreases over time
- When The Republican Presidential Candidate collides with a Rich Donor, The Republican Presidential Candidate's scale increases

Dynamics:

- If enough time passes, The Republican Presidential Candidate will no longer be visible on the screen

Next, it applied the attacks micro-rhetoric to the legislation and the rich donors (2.4.3)

Mechanics:

- Legislation spawns a Generic Shape
- The Generic Shape moves in a straight line along the vector that the Legislation is facing when it is spawned
- When the Generic Shape collides with a Rich Donor, the Rich Donor is removed from the screen

Finally, it applied the micro-rhetoric for destroys to the Majority Party and Legislation (2.4.1):

Mechanics:

- When the Majority Party collides with Legislation, Legislation is removed from the screen

Figure 3 shows a screen shot of a game that has all of the mechanics and dynamics described above. Note that users of *Game-O-Matic* provide images to represent the nouns in their concept map input. The majority party is represented by a picture of a Republican elephant, the politician is represented by the man in a suit, the legislation is represented by a document and hand writing the word "No" and the rich donors are represented by men with monocles.

Of course, one of *Game-O-Matic's* strengths is its ability to rapidly generate many games from the same input. If the user isn't satisfied with the game in figure 4, for example the game isn't compelling or has other representational issues (see section 4), he can simply generate a new game.

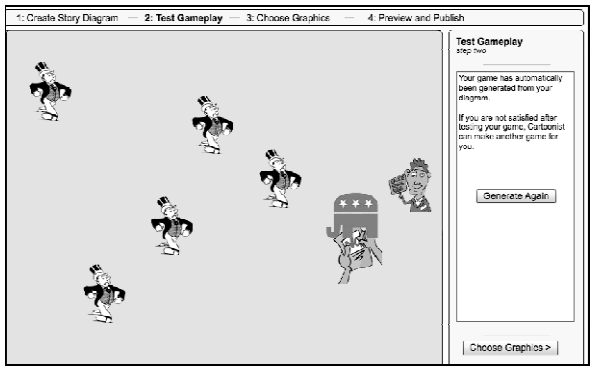

Figure 4. A second game that represents the concept map input in figure 3.

For example, figure 4 shows a game where the Majority Party prevents the legislation from being able to affect the Rich Donors. Donors move across the screen, left to right, and the Candidate positions himself to catch the Donor. But the Legislation also moves along a vertical line to try to intercept the Donors, removing them from the screen. It is up to the Majority Party, which moves along the same line as the Legislation, to slow down the interfering Legislation. The mechanics of this implementation are quite different, but are based on the same concept map input as the previous example.

3.3 Future Work

In the game described above, there are several additional mechanics that could be added to make the game both more engaging and more rhetorically successful. For example, the first game would be much better if the Legislation and Rich Donors were frequently respawning, as without this mechanic, the moment one disappeared the game would be over, communicating its intended meaning exactly once.

Already in progress are several additional layers to *Game-O-Matic* that help add structure to the games it generates both rhetorically and in terms of gameplay. Our initial implementation of *Game-O-Matic* relies on simple graphical logics constructed through collisions and movement. The interaction of micro-rhetorics play out on the screen, but the shape we take may be unfamiliar to the average user. In order for the games produced by *Game-O-Matic* to more successfully communicate their desired meaning, their interactions and goals need to be apparent. This can be accomplished by implementing design constraints that create identifiable patterns of gameplay.

Rather than emulating familiar games—a *Kaboom* type, a *Space Invaders* type, an *Asteroids* type—*Game-O-Matic* is being designed to use a library of familiar gameplay patterns that overlap with the collection of micro-rhetoric mechanic patterns, giving the generator suggestions for how to arrange objects on the screen, assign speeds of movement, game controls, and goals. With short-form rhetorical games such as newsgames, the instructions need to be immediately clear to the player, especially since the bite-sized games produced by the system are designed to last less than a minute. Nelson and Mateas's discussion of *WarioWare*-scale generated games informs these constraints [8]. The familiar gameplay patterns, or recipes, help ensure that the player can quickly orient to the goals and dynamics of the game.

The problem with generating games that explicitly follow design patterns of well-known arcade games is that the underlying arcade pattern is immediately apparent to players (e.g. "Oh, that's just a skin of *Space Invaders*"). Additionally, following such explicit patterns would dramatically limit the generative possibilities of *Game-O-Matic*. By extracting more abstract gameplay patterns from classic arcade games, the recipes can be combined in large number of combinations to constrain and provide parameters for selected micro-rhetorics, producing a large number of well-formed games with novel combinations of dynamics, controls and goals.

4. DISCUSSION

As alluded to earlier, games created by simply combining micro-rhetorics may not represent what a designer intends. Even putting aside the complications that arise from each player's highly variable personal histories and cultural contexts, what a combination of micro-rhetorics represents may lose some of its meaning or have different meaning than simply the sum of their representations.

This complication is evident in Molleindustria's *Kosmosis*, a short form game created to be a "procedural representation of collectivist/revolutionary statements..." [7]. 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 labeled the "war machine." If the player collides with the war machines, all the space prolets are dispersed and stop following 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 proletariat 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, as argued previously in [13], 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."

Previous work performing a proceduralist reading of Data East's 1982 arcade game *BurgerTime*, shed light on further complications in interpreting a videogame's dynamics [11]. Much effort was spent trying to reconcile the seeming inconsistency in the obvious interpretation of the collisions between the chef, controlled by the player, and the various types of foods. In *BurgerTime*, as the chef collides with pieces of burger, the parts stack into an assembled burger. These collisions can represent preparing the food. However, this interpretation doesn't seem to hold when considering collisions between the chef and other ingredients. Hot dogs, eggs, and pickles cause the chef to collapse and lose a life, but the enemy foods remain unaltered.

Only after investigating the strategies of expert players were dynamics discovered that contributed to a comprehensive and consistent interpretation. Expert players of *BurgerTime* can group enemy foods together by exploiting their navigational algorithms and controlling their movement with the chef's pepper attack. Once grouped, the experts drop the burger ingredients on top of them to receive a hefty boost to their score. Given this, one can interpret *BurgerTime* as representing a chef mixing seasoning (the enemy foods) and putting it into the burgers. The antagonism between enemy foods and the chef force the player's behavior, but the meaning of those collisions does not have to be part of this interpretation.

This demonstrates how the particular dynamics a player chooses to focus on—and the relative importance they place on these dynamics—will produce different interpretations. As illustrated by the expert *BurgerTime* players, the nuances of dynamics experienced by players of different skills also affects interpretation. Micro-rhetorics encourage interpretations, but they do not guarantee an absolute reading of authorial intent.

Designers who intend to create games to represent particular ideas should carefully consider these complications that arise when making use of procedural rhetoric. In *Game-O-Matic*, these complications are sidestepped by the ease of generating another game. If a user doesn't like the meaning or gameplay of the game produced, another can be quickly generated with just the click of a button.

5. CONCLUSION

Micro-rhetorics allow designers to establish logical claims about their representational intentions, while also providing players with a grounded way to validate their interpretations. A micro-rhetoric is always a specific instantiation of an idea as gameplay, but is composed of game mechanics that can be extended to many situations.

The pattern of design used in *Game-O-Matic*, based on objects in micro-rhetorical relationships, is extensible to a whole spectrum of game scales. Micro-rhetorics are bundles of representation, so while *Game-O-Matic* relies on simple graphical logics, it is conceivable that a group of complex dynamics could be considered a micro-rhetoric in a much larger system.

Game-O-Matic demonstrates the combinatorial powers of game mechanics for rhetorical use. By generating games that can be interpreted through the method of meaning derivation, it tests the plausibility of micro-rhetorics as a unit of understanding. The rhetorical affordances of game mechanics, when realized through thematic and cultural considerations, can procedurally generate varieties of interpretable games for play and examination.

6. ACKOWLEDGEMENTS

This paper was informed by discussion and research done in conjunction with Simon Ferrari, Chris DeLeon, Bryan Blackford, and Emmy Zhang.

7. REFERENCES

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

2. Begy, J. Experiential Metaphors in Abstract Games. *Conference of the Digital Games Research Association - DIGRA 2011*, (2011).

3. Bogost, I., Ferrari, S., and Schweizer, B. *Newsgames: Journalism at Play*. MIT Press, Cambridge, MA, 2010.

4. Bogost, I. The Cartoonist Aims to Bring Newsgames to the Masses. *PBS MediaShift Idea Lab*. http://www.pbs.org/idealab/2010/09/the-cartoonist-aims-to-bring-newsgames-to-the-masses243.html.

5. Hunicke, R., Leblanc, M., and Zubek, R. MDA : A Formal Approach to Game Design and Game Research. *Challenges in Game AI Workshop Nineteenth National Conference on Artificial Intelligence*, (2004).

6. Mateas, M. and Wardrip-Fruin, N. Defining Operational Logics. *Digital Games Research Association (DiGRA)*, (2009).

7. Molleindustria. Kosmosis. http://www.molleindustria.org/kosmosis/kosmosis.html.

8. Nelson, M.J. and Mateas, M. Towards Automated Game Design. *In AI*IA 2007: Artificial Intelligence and Human-Oriented Computing*, (2007), 626-637.

9. Treanor, M., Mateas, M., and Wardrip-Fruin, N. Kaboom! is a Many-Splendored Thing : An interpretation and design methodology for message-driven games using graphical logics. *Foundations of Digital Games*, (2010).

10. Treanor, M. and Mateas, M. Newsgames: Procedural Rhetoric meets Political Cartoons. *Digital Games Research Association - DIGRA. 2009*, (2009).

11. Treanor, M. and Mateas, M. BurgerTime: A Proceduralist Investigation. *Conference of the Digital Games Research Association - DIGRA 2011*, (2011).

12. Treanor, M., Schweizer, B., Bogost, I., and Mateas, M. Proceduralist Readings: How to find meaning in games with graphical logics. *Proceedings of Foundations of Digital Games (FDG 2011)*, (2011).

13. Treanor, M. Kosmosis – Procedural rhetoric gone wrong (as usual). *Expressive Intelligence Studio Blog*, 2009. http://eis-blog.ucsc.edu/2009/07/kosmosis-procedural-rhetoric-gone-wrong-as-usual/.