Principles of Emergent Design in Online Games: *Mermaids* Phase 1 Prototype

Celia Pearce*
Emergent Game Group {EGG}
Georgia Institute of Technology

Calvin Ashmore†
Emergent Game Group {EGG}
Georgia Institute of Technology

Abstract

This paper outlines the first phase prototype of *Mermaids*, a massively multiplayer online game (MMOG) being developed by Georgia Tech's Emergent Game Group {EGG}. We describe *Mermaids* in the context of the group's research mission, to develop specific games, techniques and design features that promote large-scale emergent social behavior in multiplayer games. We also discuss some of the innovative design features of the *Mermaids* game, and describe the rapid prototyping and iterative development process that enabled us to create a working prototype in a relatively short period of time on a zero budget project using a student-based development team. We also discuss the special challenges encountered when trying to develop a nontraditional game, one of whose stated research goals is to interrogate MMOG conventions, using a relatively conventional game engine.

Keywords: online games, massively multiplayer online games, MMOGs, emergence in games, networked games, real-time 3D

1 Emergent Behavior in Online Games

The Emergent Game Group {EGG} at Georgia Tech was formed in 2006 to explore questions that had arisen from co-author Pearce's Ph.D. thesis work between 2004 and 2006. [2006] The thesis project, framed as "design research," analyzed large-scale social emergence in online games to identify the role of software design in promoting or hindering emergent behavior. The thesis study focused on inter-game immigration, following two groups who had become refugees after the closure of the beta test of *Uru Live* [Cyan 2003) in 2004. (*Uru* was re-released in 2007 as *Myst Online* by Gametap.) The construct of observing inter-game immigration provided a unique opportunity to observe how the emergent behavior of a single group might take different forms in different games based on the software affordances of each. The primary group studied, an Uru "hood" (similar to a guild in other massively multiplayer games or MMOGs) that comprised 300 players, moved between five different MMOGs or online virtual worlds (including two they built themselves) during the course of the 18-month study. In each, their play style intersected with the software of the game to create new emergent play patterns. One of the key findings of the study was that the type of games that players are attracted to will create a foundation for the types of emergent behavior they are likely to exhibit. Another was that the properties of games themselves promote the honing of particular play styles and skills that will lead to emergent behavior. For instance, in *Uru*, exploration and puzzle-solving were predominate play patterns at which players of *Uru* and previous *Myst* games had developed considerable mastery. When this expertise migrated into other virtual worlds, it led to the emergence of particular types of exploration and questing patterns that were characteristic of Uru players. [Pearce 2006]

The study identified a spectrum of MMOG and virtual world types spanning from the "fixed synthetic world," in which the world is entirely designed and built by the game's developers and in which players have little influence on the world itself, to the "co-constructed world" in which players have a significant hand in actually designing and building the world. Examples of the former include *Everquest* [Verant 1999] and *World of Warcraft*, [Blizzard 2004]; fixed synthetic worlds tend to fall under the rubric of "games." At the opposite end of the spectrum one might find a world like *Second Life* [Linden Lab 2003], in which everything in the world is entirely user-created. A co-constructed world such as *There.com* [There, Inc. 2003] falls a little closer to the fixed synthetic world spectrum because the designers still exercise a certain measure of control over the world; for instance, players cannot alter the terrain, and all player-created items must be approved by management. These two examples are more open-ended play spaces than games per se, but the share in common with the previous examples that their primary purpose is networked play. A game such as *Star Wars Galaxies* [Sony 2003], while technically a fixed synthetic world, may fall a little towards the co-constructed end of the spectrum because it has affordances for players to own businesses and design their own homes.

A key finding of Pearce's study is that "emergence happens," regardless of the word type. In addition, emergence is generally the result of the intersection of play patterns with software affordances. For instance, in *Lineage* [NCsoft 2001], because items dropped on the ground remained visible, these could be used as décor for such emergent play patterns as in-game weddings. *Uru* had the unusual affordance of items not being affixed to the ground. Thus stones, wooden logs, and other objects could be rolled, usually by walk-kicking them since avatars have little use of their hands. An example of such an object was the traffic cone, which was abundant throughout the underground ruin of the city of D'ni, the home base of "Uru explorers." These cones were intended to mark reconstruction zones, but because they were not tied to the ground, players adopted and subverted this object in various ways, first kicking it around, and eventually, through experimentation, developing more structured games. One such game was "avie bowling." Players discovered a flaw in the collision in one area of the world that enabled them to submerge

* e-mail: celia.pearce@lcc.gatech.edu

† e-mail: ashmore@gmail.com

their avatars up to the neck in the floor. They could then run quickly along the floor using their heads as bowling balls and the traffic cones as pins. Later, players invented the "D'ni Olympics," a series of sporting activities that exploited various aspects of the *Uru* world. Among these sports was "cone balancing," which entailed upending a cone and standing on the bottom while balancing on its point. A related game invented by Uru players in *There.com* was "buggie polo," a soccer-like sport that was played in dune buggies. The ball was actually a vehicle, a large orb that was occupied by a player. The player inside the orb-ball remained passive through most of the game, allowing the two teams to knock the ball about where they wished. She would only actively drive the ball when it went out of bounds. Since there were no loose objects in *There.com*, this was the only way to actually play a ball game. Buggie polo is reminiscent of avie bowling in that players once again took the role of a ball. [Pearce 2006]

2 Emergence as a Design Material

The Emergent Game Group {EGG} was formed to extend this research in emergent group behaviors to the application of design practice. The premise of the group's research is that, while emergent behavior is by definition unpredictable, a game's design can consciously include mechanisms and affordances that will increase its likelihood and influence its direction. The premise of *Mermaids* was, therefore, to treat emergence as a design material, factoring it into the game's overall design and storyline, as well as crafting specific game features that would promote emergent behavior. *Mermaids* also interrogates a number of core assumptions and conventions of MMOG design in order to promote new forms of social play and allow for new play patterns to emerge.

Mermaids is set in an underwater world in which players take the roles of hatchlings coming to life in the ruins of a long-extinct mermaid culture. The over-arching goal and storyline is to rebuild their lost culture and reclaim the various skills and cultural practices (such as magic), while at the same time trying to avoiding the mistakes that caused the extinction of their ancestors. Players will quickly discover that in order to survive, they must also revive the damaged ecosystem, and ultimately learn to work with it in a harmonious fashion.

This theme is very strongly influenced by *Uru*, in which players begin as explorers in the ruins of the once-glorious but now abandoned underground city of D'ni Ae'gura. *Mermaids* draws its approach to storytelling from *Uru*, in which players construct the story through social interaction. Unlike many other MMOGs, *Mermaids*, like *Uru*, has no exposition describing what happened and telling players what they are to do about it. The premise of a ruined city provides an incomplete story, one that players must re-construct through social negotiation. Hypotheses and theories abound, and there may ultimately be no "right" answer.

Chris Crawford has said that games should be thought of in terms of verbs. [1984] *Mermaids* deliberately moves away from the predominate verb of many MMOGs which is "to kill," and instead emphasizes verbs such as "build," "grow," and "explore." These three verbs are inherently more likely to promote emergence because the first two in particular allow players to actually transform the world through their own agency.

We also wanted to move away form the "grind" framework of many games in which multiple repetitive actions produce "experience points." *Mermaids* has no point system and its experiential aim is to make gameplay activities more varied and creative and less routinized.

3 Design Overview

Mermaids has a number of features that challenge the traditional conventions of MMOGs in a variety of ways. In this section, we will describe the game's key design principles and features; later, in the section on implementation, we will describe the ways in which these features push the boundaries of conventional engines.

Mermaids has three primary design credos that the team was given as its "high-level" design constraints:

- Think socially; act procedurally
- Question conventions
- Appeal to diverse players

From these three constrains arose the following design features, many of which, though influenced by other games, are highly innovative in terms of MMOG design:

Swimming: The first and perhaps most innovative aspect of *Mermaids* is the fact that players swim rather than walking. While this does not seem particularly innovative, it introduces an order of magnitude more complexity. While their graphics are 3D, in terms of actual gameplay, most MMOGs are actually two-dimensional, or what we characterize as "2D with bumps": players usually move over terrain hugging the ground. With the exception of a handful of virtual worlds that fature flying, such as *There.com* and *Second Life*, when players do move in 3D, it is generally onto platforms or terrain contours and not in open space; in a small number of case, avatars can swim in bodies of water. Avatars, with a few exceptions, are typically fixed to the ground. Adding the vertical (z) axis to players' movement introduced a number of challenges and opportunities in terms of input and navigation that we are still continuing to address. In our research, we quickly discovered that most game engines are designed with the basic assumption that players will be walking on the ground, with only two axes of navigation, or two degrees of freedom.

Mouse-Based Gestural Interface: The second innovation in *Mermaids* is the player's input. This is a feature of *Mermaids* that was developed in direct response *against* the prevalent conventions of MMOGs. While the hands are the predominate interface that players have with most virtual worlds, via keyboard, their avatar characters generally have almost no use of their hands whatever. Inspired by single-player games *Okami* [Clover 2006] *Arx Fatalis* [Arkane 2002], and *Myst V* [Cyan 2005] we developed a mouse-driven gesture system. The gesture system was developed by Rob Fitzpatrick and Chris Langston, and prototyped and implemented by Fitzpatrick. Players enact agency in the world through a mouse-controlled gesture-based interaction that allows them to draw in three-space. Depending on what tool, weapon, or wand they are holding, these gestures will have different effects. Without holding onto anything, players can engage in free-drawing with bubbles, a mechanic that is intended to promote open-ended play and emergent behavior. If they have a wand, they can draw a number of symbols and characters that the

computer will recognize and translate into an effect on a selected object. Over time, each player can accumulate a variety of different tools, each with its own gestural commands and effects. These might include harpoons that can be thrown, special pens or paintbrushes for map-drawing, spindles for net-weaving, etc. This open-ended gesture system creates an extensible interface paradigm to which can be expanded by adding new tools.

Non-Linear Progression: Leveling is one of the most prevalent conventions in MMOGs. In most cases, it takes the form of experience points that translate into a numerical value that indicates your progress in the game, or your level. This is a linear process, and while there are a number of ways to gain points, there tends to be an optimum formula for success: harvest experience, primarily by killing things or completing quests. The leveling mechanic promotes the feeling of accomplishment in players, but usually at the cost of repetitive and tedious activity. Because game challenges are tuned to players' levels, this mechanic has the unfortunate consequence that players outside of three levels from each other cannot effectively play together, thus setting up an artificial skills stratification that inadvertently penalizes more experienced players for helping newer players. We wanted a system where players could play across diverse skill levels, and we are also developing a system to reward higher-level players for helping "newbies." With few exceptions, levels also do not allow player directed control over their avatar's abilities; instead, the construct of race and class places significant constraints on players as to which skills and abilities they can acquire. We therefore decided to do away with leveling, as well as with race and class. In order to maintain a feeling of accomplishment without levels, we developed the concept of "non-linear progression," which has the following tenets: 1) players do not have to pick a race and class at the beginning of the game that predetermines their fate for the rest of their game experience; 2) Players thus are not compelled or restricted from learning skills based on their "class," but rather, they learn skills and develop capabilities based on their own play interests; 3) players do not have to achieve goals in any particular order, but can work at their own pace, emphasizing the activities they most enjoy based on their play style; 4) Players can build their skills portfolios provisionally within a given play session, or alter their specialties over time. A player may start out wanting to be a hunter, but later, may decide she wants to move into a healing role. There should be no reason why she cannot do both of these things simultaneously, nor why she should not be able to change her mind from time to time. 5) The marker of progress in the game is determined by what you wear. One consistent feature in the mythology of mermaids that turned up in our research was that they like to decorate themselves. Thus, when mermaids accomplish a task, they get some kind of ornament or talisman, perhaps a shark's tooth to indicate a completed hunt or a tattoo to display a skill acquired, or even a scar to indicate survival of a dangerous mission. They wear whichever wands, weapons or tools they are using on a belt around their waist so that other players can see the skills they are currently sporting. This will create a basis for grouping since players can literally see each other's capabilities. It also supports a dress-up mechanic, which Pearce and her collaborators in Ludica, have found is a prevalent emergent play pattern. [Ludica 2007]

Diverse and Underserved Player Types: The common paradigm for online games tends to be at one of two extremes: one is hardcore MMOG which requires three or more hours to complete a quest or task, the other is the casual game with a twenty-minute average play time. A few games, such as *Puzzle Pirates* [Three Rings 2003] have tried to fill a niche between the two. This player, which has been referred to as "game enthusiast," might wish to spend an hour or two in a game but can't commit the 3-6 hours required of a typical *World of Warcraft* [Blizzard 2004] dungeon quest. In addition, we wanted to readdress the Bartle player types (Killer, Achiever, Socializer, Explorer) [1996], which Pearce has argued are very genre-specific. [2006] Furthermore, many MMOGs strongly favor the killer and achiever types, providing nominal and token rewards for exploration and, more often than not, penalizing the socializer. Other types of players are also ignored entirely by the Bartle taxonomy. Economic achievement, for instance, is not fully accounted for, nor is puzzle-solving; creative activities, such as crafting or fashion design are also neglected in this scheme. *Mermaids* is designed to favor explorer and socializer types, while including activities for achievers and killers, and opening up play to unaccounted for types, such as creators and economic achievers. We also wanted to integrate socializing into the actual gameplay, rather than making it predominately a downtime activity. The open-ended storyline and mystery-solving aspect of the game provides a framework for socializing *while* playing. A multiplayer net-creation mechanic, for example, has been designed to engage players in a social crafting activity. There will be no NPCs, in part because we don't like them, and in part because we do not have the resources to generate a large amount of extra assets and content. Instead, any quest-like activities in the game will be given to players by other players, another instance of "think socially, at procedurally." We plan to design a mechanism for players to define their own quests, which might entail getting other players to do specific tasks for them to assist in their own goals, e.g., gather resources for a craft, provide protection on a harvesting mission, etc.

Exploration/Scavenger Hunt: Drawing on Pearce's previous work on Iwerks and Evans & Sutherland's *Virtual Adventures: The Loch Ness Expedition* [1994], the game develops exploration and scavenger hunt mechanics. Exploration is its own inherent game mechanic, but in addition players will explore to find treasures and resources throughout the world. In the first prototype, for instance, players must collect pieces of a magic wand, which, once found, they can activate to restore dead coral using the spell-casting mechanism described above.

Dynamic Ecosystem: The game will have a dynamic ecosystem. Part of the player's goal will be to learn how the ecosystem works in order to a) be a good caretaker of the environment and b) learn how to use its resources in a harmonious and sustainable way. Selected plants and creatures can have various uses depending on player skills and knowledge. Some can be harvested, others can be grown and cultivated in controlled ways. There will be an intricate net of relationships between the different types of flora and fauna. Discovering and learning how to strategically leverage these relationships in a balanced fashion will be part of the gameplay. For instance, certain plants might attract certain creatures, which might in turn serve as prey to other creatures. Players can use this information to locate creatures or plants that they want to use, or to plan their own gardens around a particular set of desired resources.

Constructivist Gameplay: While there will be some hunting and killing in the game, the majority of *Mermaids* game activities are

constructive in nature. Players will restore dead coral and plant life, and eventually learn to plant their own gardens. Loose objects will be available, including the relics and ruins of the extinct mermaid culture. Players may wish to reconstruct structures as they originally existed, or perhaps use the materials to create new kinds of structures. One of the activities sketched out using our rapid-prototyping process is a map-drawing game. Here a player may find a map in a bottle, which, when opened, starts to fade. Working together, a group of players can trace the map, filling it in before the original disappears. Another activity that has been prototyped is a group net-building mechanic. Players encircle a spire, in a similar fashion to a maypole, and by turning in a circle, weave a spider-web like net. Depending on the number of mermaids and the speed at which they move, they can control the mesh density of the net; they can also pick materials to make the net specialized for capturing different creatures, e.g, heavy material in a loose mesh for large creatures, finer material and mesh for smaller creatures. Groups and tribes can also customize their nets to develop characteristic designs and color schemes.

By negotiating between player content creation and initial world design, *Mermaids* falls happily somewhere along the center of the spectrum between fixed synthetic and co-constructed worlds. Players can create content and modify the game world, but ultimately the world possesses a background story and dynamics that situate the players and their created content.

Emergent Economy: In our early discussions we grappled with whether or not to create an economy, but decided instead to treat the economy as another emergent feature of the game. We are therefore developing a strategy in which items can emergently take on a social value as determined by the players. Some items may be rare and powerful, others might be abundant but particularly useful, or beautiful with special properties. With crafting, players can create items of value from raw materials within the game, adding an artistic dimension to the economy. Some players may develop skills that enable them to create desirable items by transforming resources not immediately recognizable as being of value. In order to have an economy we are aware that it will be necessary to have players own individual items, but not necessarily own land. We are continuing to develop these ideas with the next iteration of the game.

Change Over Time: One of the hallmark characteristics of emergence is change over time. In the majority of MMOGs, especially those set in the context of an imaginary narrative, the world changes little from visit to visit. In *Mermaids,* players should experience a notable change in the world over time, and they will also recognize the extent to which their agency has contributed to that change.

4 Prototype Development Process

Mermaids would not have been possible without a particular institutional course format that is unique to Georgia Tech's School of Literature, Communication and Culture: the project studio. The project studio framework allows faculty to pursue highly exploratory research that may not fit into traditional funding frameworks. Students participate in the project studio for credit, or in some cases, on a volunteer basis, often taking a highly collaborative and creative role on the project. This format provides students the opportunity to make both a creative and technical contribution to a project that they can then include in their portfolios. It also gives faculty (particularly new faculty) the ability to "hit the ground running" and begin working on a practical project immediately.

The design process for *Mermaids* has been highly collaborative and highly iterative. The faculty director provided a number of design constraints. From these constraints, the overall design (including the storyline premise) was developed through an iterative, collaborative brainstorming process. Some initial research was done into mermaid mythology; the result was that, other than the classic fairly tale *The Little Mermaid*, there seems to be very little, thus leaving us open to develop our own narrative. We also made a visit to the Atlanta Aquarium, which provided inspiration for design and art direction.

The traditional game development process typically entails a long period of design in which elaborate documents are prepared prior to any actual implementation. We could not follow this process for several reasons. First, although the game is defined as a two-year project, the fact that it is being developed in the context of a class meant that we had to have milestones at the end of each semester. This creates both the discipline and constraint of a much more rapid and iterative process than is the case in traditional game development. In the first semester, we had five students, with some contribution from two others. We developed the design document collaboratively via wiki, while simultaneously researching MMOG game engines, and developing concept sketches and a visual direction for the game.

The first semester culminated with rapid prototyping using Java Processing in which two of the key game mechanics, the spell-casting/gesture mechanic and the net-creation mechanic, were prototyped in a simple, 2D, interactive, single-player sketch format. (Figures 1 and 2) This enabled us to quickly determine the viability of these mechanics prior to committing them to the full 3D prototype. We also found that prototyping elements of the game in 2D was a good precursor to implementing them in 3D. We also anticipated, correctly, that the introduction of a third axis added significant complexity, thus the 2D mock-up provided a tiered, iterative approach.

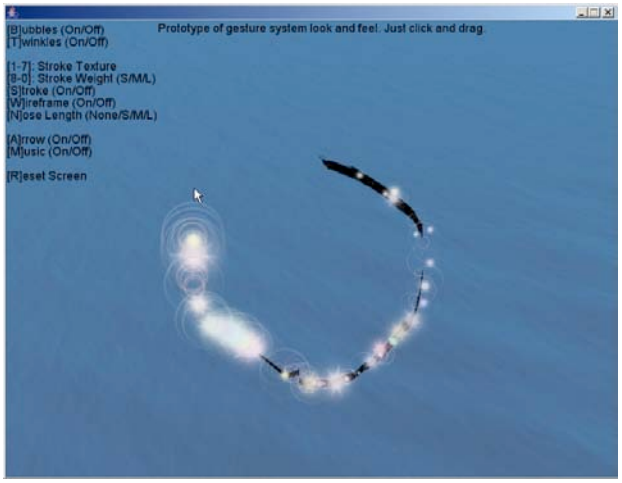

Figure 1: Gesture/spell mechanic rapid-protoype in Java Processing. Gesture system designed by Rob Fitzpatrick and Chris Langson; developed and programmed by Fitzpatrick.

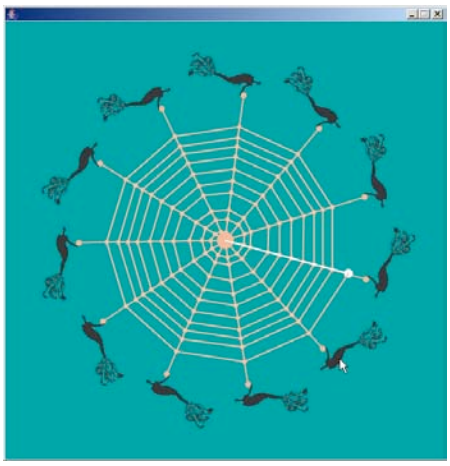

Figure 2: Net-creation mechanic, designed and prototyped in Java Processing by Will Riley with Walter Kim.

We had also set out a goal for the first semester to select the development environment/game engine we wished to work in. Most commercial MMOGs develop proprietary, custom game engines. Due to constraints of time, personnel, and budget (i.e., little of the former two and none of the latter), it was necessary to work with an existing game engine. Unfortunately, there are very few end-to-end solutions for massively multiplayer development. We had considered *Torque* (Garage Games 2001-2007), which many universities use because the site license is inexpensive relative to other engines and it has a very strong developer community. We also looked at *Unity* (OTEE 2005-2007) and *Panda 3D* originally created by Disney but currently overseen as free software by Carnegie Mellon. [2002] Neither of these engines had a server back-end that would support our goal to make a large-scale network game. Emergent Technologies and Sun Microsystems offered server-side technologies, but no client front-end. The challenge we faced was that, because our game was doing a number of novel things, we needed an end-to-end solution, but one with which we could get under the hood and make direct changes to the code. The two finalists were *Kaneva* [2006] and *Multiverse*. [2006]

The rationale for selecting *Multiverse* was as follows:

- *Multiverse* used familiar and conventional languages: Python and XML for the client, Java and Python for the server. (Several game engines use their own scripting languages that would have taken too much time to teach given our rapid turnover in personnel, which changes significantly each semester; we also wanted students to develop skills that would be useful to their later career goals.)
- Many of the file formats used in *Multiverse* are open standards, co-opted from the OGRE open source game engine. Those file formats which are not standards are XML. This combination allows us as developers substantial understanding and control over our assets.
- *Multiverse* was in beta, which, though it meant the software was less stable, allowed us to give input as various iterations came online. Although we did encounter bugs along the way, we found the software much more stable than we had anticipated.
- *Multiverse* has the feature that the client and the server can be updated separately, allowing us to modify assets and interface elements without needing to restart the server.
- *Multiverse*'s economic model requires no up-front charge to developers for creating games within its system, but takes a percentage off the back-end when developers start charging for their games. This was indispensable for an unfunded academic project; up-front payment for development tools would have made the project impossible.
- As the project progressed we found *Multiverse*'s customer support, as well as its developer community, an invaluable resource in tackling the various challenges we encountered during the implementation process.
- And most notably, while the server provides affordances for some common MMOG conventions, it defines these via an open source plug-in architecture. Because our project challenges so many tropes and conventions, we can very easily modify the server plug-ins so that we may wipe away the existing conventions and define our own strategies for the server side logic.

Just prior to the start of the second semester, we were asked by Multiverse to present the project at its booth at the 2007 Game Developers Conference. It should be noted that, short of the two processing prototypes described above, we had no working prototype in the *Multiverse* environment at this time. The Game Developer's Conference was less than two months away from the first day of our new semester. However, given that this seemed like an opportunity worth pursuing, we decided to embark on an extremely rapid phase 1 development process.

The first month of development was spent getting up to speed in *Multiverse*, a process that went much more quickly for those members of the team who had previous experience with game engines. Python and XML being fairly standard, this was a much shorter leap for students who had worked in those environments previously.

During this period, Calvin Ashmore, a Ph.D. candidate, who was eventually given the title "Chief Game Architect," and a major creative force behind the project, began building what we characterized as a "sandbox" using *Multiverse.* In its first iteration, the sandbox used the sample world provided with the *Multiverse* tool kit. During the second month, we began a process of slowly integrating our own game elements, starting with server side code, to test behavior within the sandbox. We gradually exchanged assets and client side code with the generic elements of the *Multiverse* demo until our material was the only content that remained. This process led to some interesting interstitial effects. For instance, before we integrated the mermaid model, the standing avatar from the *Multiverse* demo was floating around in the world as if walking in the air.

One advantage to this approach was that we could instantly see and test how each new feature worked, before adding the next set of components. In this way we could add features and make changes incrementally, which turned out to be extremely useful. Partly because of this method, and also because of the affordances of the *Multiverse* architecture, our team members were able to collaborate very effectively. One team member could define server side code in one module, another could do world building,

and others could write client scripts or update models and textures. Each of these aspects could be implemented and tested locally on the student's computer, and then once functioning properly, be easily and quickly integrated into the shared sandbox.

Once we were in full swing on this incremental development process, we were able to put together the GDC prototype in a matter of about a month. Working from a storyboard developed by Sara Greer and Addy Lee with Will Riley, Calvin Ashmore built the main architecture of the game, the terrain (with contribution from Chris Langston) and the majority of models, with the exception of the mermaid, which was created by Michael Arteaga. Rob Fitzpatrick continued his work on the gesture system, initially developing a 2D version in Processing, and then adding the tree-dimensional gesture and particle system to the *Multiverse* sandbox. Will Riley and Wes St. John are currently developing the ecosystem which will be integrated in the next phase of the prototype. Other contributing team members included Walter Kim and Judith Siegal.

During this development process, the *Multiverse* engine, which was in Beta 1 and Beta 2 respectively, was subject to ongoing revision. *Multiverse*'s core system provides a single engine and environment on which a variety of games can sit. Because the engine is de-coupled from content, we were able to update our game-specific asset, code and script files independently of updates to the engine. One of the more critical updates to the engine entailed the addition of the capability for flight, a feature that was vital to our success. We are continuing to work iteratively with the Multiverse staff to improve methods for programming the swim path of the mermaid, which, as anticipated, requires some alternative strategies to the traditional walking paradigm.

By March 7, 2007, we arrived at the Game Developer's Conference with a fully working prototype, which we showed at the Multiverse booth during the Expo portion of the conference. (Figures 3 and 4) For the next phase of the project, we plan to do more extensive user testing, refinement and addition of a preliminary ecosystem prototype, and will continue to add features incrementally to our *Multiverse* sandbox. Our strategy is to then create a second Beta server where we can upload final features as they are complete for beta testing. Since *Mermaids* is specifically designed to be extensible, we expect to continue to add features incrementally even after public release of the game.

Figures 3 and 4: The spell-casting mechanic as implemented in *Multiverse* by Rob Fitzpatrick. Mermaid by Michael Artega.

Figure 5: Mermaid avatar navigating through *Mermaids* world.

5 Conclusion

This paper provides an overview of the first phase design and development process of the *Mermaids* MMOG, a work in progress of the Emergent Game Group {EGG} at Georgia Tech. We have outlined our development process and described the ways in which our game challenged the traditional paradigms of both MMOG gameplay and technology. We also outlined our initial process of 2D rapid prototyping in Java Processing followed by a rapid, iterative development process using a "sandbox" created in the Multiverse MMOG development environment. These techniques provide an alterative to traditional game development methods that can be implemented within a semester-based scheduling framework in the context of a faculty-led University research project. These techniques may also be applicable to independent development or contexts in which resources and peoplepower are limited to help make small-scale non-traditional MMOG projects both feasible and successful.

Special thanks to Ron Meiners and the team at Multiverse; also to James Hursthouse and OGSI for providing *Mermaids* with hosting services.

6 References

ARKANE STUDIOS. 2002. *Arx Fatalis*. Dreamcatcher Interactive.

BARTLE, R. 1996. *Hearts, Clubs, Diamonds, Spades: Players Who Suit MUDs*. http://www.mud.co.uk/richard/hcds.htm

BLIZZARD ENTERTAINMENT. 2004. *World of Warcraft.* Blizzard.

CLOVER STUDIO. 2006. *Okami*. Capcom.

CRAWFORD, C. 1984. The Art of Computer Game Design. Berkeley: McGraw Hill/Osborne Media.

CYAN WORLDS. 2003. *Uru: Ages Beyond Myst*. Ubisoft. Re-released by Gametap in 2007.

CYAN WORLDS. 2005. *Myst V*. Ubisoft.

DISNEY. 2002. Entertainment Technology Center, Carnegie Mellon University. www.panda3d.org

IWERKS and EVANS & SUTHERLAND. 1994. *Virtual Adventures: The Loch Ness Expedition.* (Creative Directed by Pearce; Presented in the Electronic Theater, SIGGRAPH 1994)

KANEVA. 2006. *Kaneva.* www.kaneva.com

LINDEN LAB. 2003. *Second Life.* Linden Lab. www.secondlife.com

MULTIVERSE NETWORK, INC. 2006. *Multiverse*. www.multiverse.net

NCSOFT. 2001. *Lineage*. NCsoft.

OTEE. 2005-2007. *Unity 3D.*

PEARCE, C. 2006. *Playing Ethnography: A study of emergent behaviour in online games and virtual worlds.* Ph.D. Thesis, Central Saint Martins College of Art and Design, University of the Arts, London.

SONY ONLINE ENTERTAINMENT. 2003. *Star Wars Galaxies*. LucasArts.

THERE, INC. *There.com.* Makena Technologies. www.there.com

VERANT/989 STUDIOS. 1999. *Everquest.* Sony Online Entertainment.

www.vancouver.wsu.edu/fac/peabody/game-book/Coverpage.html