Multiplayer Prototyping and Quest Design for an Educational Massively Multiplayer Online Role-Playing Game

By

Michaela LaVan

B.S., Massachusetts Institute of Technology (2012)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

June 2013

Signature of Author ....
Department of Electrical Engineering and Computer Science
May 24, 2013

Certified by ....
Eric Klopfer
Professor of Urban Studies and Planning
Thesis Supervisor

Accepted by ....
Dennis M. Freeman
Professor of Electrical Engineering and Computer Science
Chairman, Masters of Engineering Thesis Committee

Multiplayer Prototyping and Quest Design for an Educational Massively Multiplayer Online Role-Playing Game

By

Michaela LaVan

B.S., Massachusetts Institute of Technology (2012)

Submitted to the
Department of Electrical Engineering and Computer Science
in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering in Electrical Engineering and Computer Science

at the

Massachusetts Institute of Technology

June 2013

ABSTRACT

Digital games have proven to be powerful learning tools, and may hold the key to closing the STEM (Science, Technology, Engineering, and Mathematics) achievement gap in the United States. Multiplayer play in particular can foster peer instruction, mentorship, and other productive learning interactions. However, few models of effective multiplayer mechanics exist for educational games. This paper outlines a scheme for developing synchronous multiplayer tasks for a Flash-based massively multiplayer online game (MMO), and discusses design considerations for creating meaningful multiplayer interactions. The results were applied to the design and implementation of multiplayer quest prototypes for the Radix Endeavor, an educational MMO.

Thesis Supervisor: Eric Klopfer
Title: Professor of Urban Studies and Planning

Multiplayer Prototyping and Quest Design for an Educational Massively Multiplayer Online Role-Playing Game

Michaela LaVan
The Education Arcade
Massachusetts Institute of Technology
mlavan@mit.edu

Eric Klopfer
The Education Arcade
Massachusetts Institute of Technology
klopfer@mit.edu

ABSTRACT- Digital games have proven to be powerful learning tools, and may hold the key to closing the STEM (Science, Technology, Engineering, and Mathematics) achievement gap in the United States. Multiplayer play in particular can foster peer instruction, mentorship, and other productive learning interactions. However, few models of effective multiplayer mechanics exist for educational games. This paper outlines a scheme for developing synchronous multiplayer tasks for a Flash-based massively multiplayer online game (MMO), and discusses design considerations for creating meaningful multiplayer interactions. The results were applied to the design and implementation of multiplayer quest prototypes for the Radix Endeavor, an educational MMO.

ACKNOWLEDGEMENTS

I would like to extend my heartfelt thanks to the following people:

My primary thesis advisor, Eric Klopfer, for giving me to opportunity to pursue this project and for introducing me to the wonderful world of educational gaming;

My supervisor Louisa Rosenheck, for her invaluable guidance, feedback, and assistance, and for pushing me to ask the interesting questions;

Colin Greenhill, for being a second set of eyes when the code Just Wasn't Working, and for his patience as I untangled the more arcane features of ActionScript 3 and SmartFoxServer;

Susannah Gordon-Messer and Angie Tung, for their input and help with play testing the prototypes, especially the finicky and frustrating early iterations;

Alex Gurany for his initial work paper prototyping the Marketplace trading concept;

Kristina Buenafe and her students at Josiah Quincy Upper School for taking time out of the regular school schedule to play test the prototypes;

David Zou for the late-night moral support and comic relief;

And finally, I would like to thank my parents and grandparents for the love, support, and encouragement they have given me for the past 23 years. I would never have made it this far without you guys.

I. INTRODUCTION

A. STEM Subjects and the United States

The United States is currently facing an educational crisis. Science, technology, engineering and mathematics (STEM) are of critical importance to the United States' economic strength, and job growth in these fields has accelerated in recent years, with new STEM jobs being created at three times the rate of jobs in other sectors (Langdon, 2011). At the same time, most students in the United States are struggling with math and science. The mathematical and scientific literacy of American students, including top performers, lags well behind that of their peers in other industrialized countries (National Science Foundation, 2012). This achievement gap is coupled with a pernicious lack of enthusiasm for scientific pursuits. Even among students who enter high school expressing an interest in these fields, the majority will lose interest by graduation (Morella, 2013). If this trend is allowed to continue, the United States will soon cease to be a major scientific force, and the breakthrough technologies of the future will be developed elsewhere.

Many top policy-makers and educators have identified the need to increase the number of STEM degree-holders as a top priority for the United States moving forward. If this goal is to be realized, concrete steps must be taken at the high school level, both to improve students' STEM proficiency and knowledge, and also to promote the excitement and interest in these fields that will encourage students to pursue further study of these subjects in college and beyond.

B. Educational Gaming

In order to learn effectively, students must be encouraged to try, enticed to put in effort, and rewarded with success after working hard. In this respect, games are excellent teachers, and the effectiveness of video games as educational tools in this regard has been well documented (Hung and Chen, 2009). However, games are not merely good motivators; they are also ideal vehicles for delivering educational content. James Paul Gee, one of the leaders of the serious games movement, has identified 36 characteristics of games that make them particularly well suited to learning. Some key insights include the "regime of competence" principle (players are constantly met with challenges appropriate to their ability), the practice principle (players are given ample opportunity to exercise new skills and work towards mastery of a task), and the discovery principle (players make progress through experiments and open-ended exploration) (Gee, 2007).

C. Massively Multiplayer Online Games

Massively Multiplayer Online Role-Playing Games (MMORPG's, or MMO's) are online video games in which many players can interact simultaneously. They are characterized by rich, immersive worlds and a persistent game environment.

MMO's have several properties that make them especially promising tools for tackling the STEM achievement gap. Most MMO's are "open world" games, meaning that players have the freedom to explore the entire game space. This nonlinear format challenges players to take initiative and discover puzzles on their own. Games that encourage players to proactively investigate the way the world works and to develop solutions to problems based on this understanding are not only teaching students the answers, but also how to ask questions — in effect, teaching them to think like scientists. In addition, the multiplayer facet of MMO's is an integral part of the gameplay experience, and many quests are difficult or impossible to solve as a single player working alone. This emphasis on teamwork mirrors the cooperative nature of research and underscores the importance of collaboration in the scientific community. Despite these benefits, the relative newness of online gaming combined with a culture of skepticism regarding the educational merits of games means that there is a paucity of research focused on multiplayer gaming within the educational sphere.

D. The Radix Endeavor

The Radix Endeavor ("Radix") is an MMO aimed at increasing high school students' proficiency and engagement in STEM fields. Funded by the Gates Foundation, the game is being developed as a collaborative effort between the MIT Media Lab and Filament Games. The initial curriculum will align with the Common Core standards in mathematics and the Next Generation Science Standards, and will focus on algebra, geometry, probability, statistics, genetics, and environmental science. Through the persistent, shared world of an MMO, Radix aims to provide an immersive environment that will foster critical thinking skills and a spirit of scientific inquiry.

1. The Scheller Teacher Education Program

The Scheller Teacher Education Program (STEP) is a research group at the MIT Media Lab that conducts research and development of innovative learning technologies, with an emphasis on games and simulations (Scheller Teacher Education Program website). They are the creators of the popular StarLogo visual programming language and have developed a number of educational games in conjunction with The Education Arcade, another research group at MIT (Hass, 2008). Much of their work focuses on making STEM subjects more fun and accessible to middle- and high-school students, although the full scope of their projects covers subjects as diverse as history and language learning.

2. Narrative

Radix takes place on a thriving island located in the center of the sea. The island is in many ways a utopia- it is a self-governing democracy and an international trading hub, with a culture dedicated to learning and discovery. A rich profusion of diverse plant and animal life populate its shores. However, recently a new leader has come to power whose greedy pursuit of profit and exploitation of natural resources is leading the island into decline. At the same time, many strange and unexplained things have been occurring, including the disappearance of the island's founder. Players join a group of knowledge-seeking vigilantes known as the Curiosi working to uncover these mysteries and restore balance to the island.

3. Prototyping System

Although Filament Games will do the development work for the final production version of Radix, STEP is also developing in-house prototypes for proof-of-concept design work and for field-testing new game ideas. This prototyping framework is discussed in more detail in Section III.

4. Multiplayer Prototypes

One of the motivating goals of the Radix Endeavor is to offer a collaborative learning environment that enables students to engage with one another as well as with the game itself. To achieve this, STEP researchers must design prototypes that incorporate meaningful multiplayer interaction into the learning process. The design, implementation and evaluation of these multiplayer prototypes is the primary focus of this paper. The contribution of my thesis work is twofold: first, I extended Radix's existing prototyping framework to support simultaneous play and interaction between multiple players, and second, I created several prototypes in which multiplayer features play a central role. Section III outlines the technical implementation of the multiplayer prototyping framework, while Section IV discusses the design goals and details of the prototypes themselves.

II. RELATED WORK

Some educators have taken advantage of the versatility of MMO's to build educational curricula on top of existing MMO platforms. World of Warcraft, one of the most popular online fantasy role-playing games at over 10 million users, has served as the basis for curricula in statistics and environmental science (Bainbridge, 2007). A team at Illinois State University designed an economics curriculum based on the in-game economy of EverQuest II, and outlined potential curricula based on the same game in fields as diverse as physics, linguistics, sociology and philosophy (Riegle and Matejka, 2006). Piggybacking off of existing MMO's is an attractive option, as these games can offer rich, immersive worlds with a vast amount of technological and financial resources behind them. However, this approach is not without its downsides. Building an educational experience on top of a game designed for entertainment means that learning and gameplay are decoupled. In addition, parents may have concerns about the use of these online environments in the classroom, where teachers have little control over the kind of people or content to which students may be exposed.

There are also a number of online virtual worlds created with an educational mission in mind. The River City project aims to teach students about epidemiology and the scientific method by means of an immersive, 3-D environment where students solve scientific puzzles as they investigate the spread of disease in the fictional River City community (River City Research Team, 2007). Although this game borrows features of online role-playing games to create immersive and engaging educational experiences, interaction with other players is peripheral to the gameplay; students mainly complete single-player quests and individual investigations. Another project, called Quest Atlantis, also aims to engage students in curricular tasks by presenting them as "Quests" in an online virtual world (Atlantis Remixed Project Team). In Quest Atlantis, the game community serves as an important forum for aggregating scientific data and discussing social and environmental issues. However, the Quests themselves are completed individually. The Radix Endeavor aims to emulate the successes of these games and to push the educational potential of online virtual worlds one step further, by offering puzzles that require real-time collaboration between students.

III. MULTIPLAYER PROTOTYPING FRAMEWORK

The Radix prototyping framework enables rapid development and testing of new learning and game-play concepts. Before researchers on the Radix team can implement and test multiplayer scenarios, the existing prototyping framework must be extended to support multiplayer interactions. The first contribution of this thesis was to develop a system backend to support real-time player interaction and cooperative play. This section outlines the architecture and functionality of this multiplayer prototyping framework.

A. Radix Prototyping System

Prior to the start of this thesis project, the Radix Endeavor already had a prototyping system in place for the development of single-player quests. The system was implemented using ActionScript3 and can build playable quest prototypes from a designer-specified XML document. A "quest" is an in-game task designed to accomplish a learning objective, usually as part of a larger series of quests, or "quest line." The following paragraphs describe the basic structure of a quest in a Radix prototype.

1. Scenes

A scene in a single contiguous space within the game, containing a set of objects with which the player can interact in different ways. A scene is typically associated with a single quest, and contains all the items needed for the player to complete that quest. The contents of a scene are defined in an XML document and are rendered by the prototyping engine into a playable Flash game.

```
        </quest>
    </npc>

    <sign x="10" y="1" text="Marketplace"/>
    <shop x="10" y="10"></shop>
    <shop x="10" y="13"></shop>
    <shop x="10" y="7"></shop>
    <shop x="10" y="4"></shop>
    <shop x="10" y="16"></shop>


    <vendor x="11" y="10" name="Vendor Charles" vendortype="standard4" introText="Welcome to my shop!">
    <inventory>
        <item name="vanillaflower" quantity="2">
            <object name="vanillaflower">
                <vanillaflower></vanillaflower>
            </object>
            <trade name ="button" quantity="1">
                <object name="button">
                    <button></button>
                </object>
            </trade>
        </item>
        <item name="Cheese" quantity="2">
            <object name="cheese" quantity="2">
                <cheese></cheese>
            </object>
            <trade name ="banana">
                <object name="banana">
                    <banana></banana>
                </object>
```

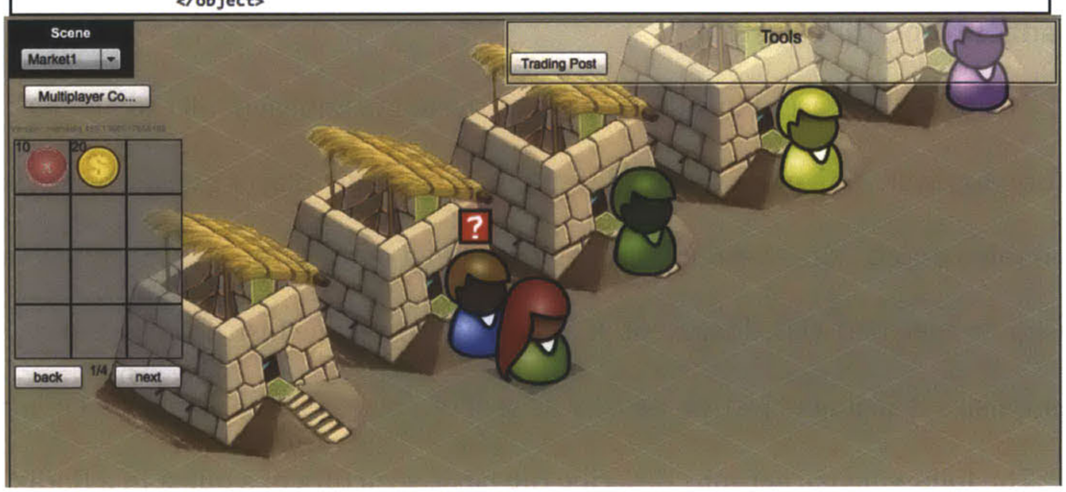


Figure 1. A snippet of XML and the corresponding rendered scene.

Some key components of a functional scene include the player avatar, scene elements, and non-player characters.

i. Player

The player character is represented by an on-screen avatar (the redheaded character in Figure 1 above), with starting location defined in the scene XML. A student can move his or her avatar to any empty location in the scene by clicking on that location.

The player also has an inventory for storing any items collected. The inventory is represented by a grid of 12 slots in the upper-left corner of the screen. Icons in these slots indicate items that are currently in the player's possession; multiple copies of the same type of item may occupy the same slot. The inventory can hold up to 50 different types of items.

ii. Non-Player Characters

Non-player characters, or NPC's, are computer-controlled characters. The player can interact with an NPC by clicking on them. NPC's may deliver important information via lines of dialogue, or they may give the player items. NPC's can also give a quest to the player. NPC's who have a quest associated with them are marked by a yellow star floating over their head.

iii. Scene Elements

All other visible items in the scene are scene elements. Each scene element has its own XML tag, such as <flower> or <shop>. The player can interact with a scene element by clicking on it. Many scene elements are collectible; clicking on these elements will add them to the player's inventory. Other scene elements may simply be decorative background items to add flavor to the scene. Many quests involve collecting a particular type of scene element, or investigating a scene element's hidden properties, such as the genotype of a flower.

iv. Tools

Tools enable the player to interact with scene elements in more complex ways. For example, the ruler tool allows the player to measure the height or length of many elements, while the crossbreeding tool lets the player perform a test cross between two flowers and view the resulting offspring. If a player currently has a tool in his possession, the interface to the tool is visible in the upper-right-hand corner of the screen.

2. Quest Dialogs

As mentioned above, quests are given to players by NPC's. Clicking on an NPC with a quest triggers the appearance of a quest dialog.

i. Initialization

Figure 2. A quest initialization dialog.

After clicking on a quest-giving NPC for the first time, a quest initialization dialog appears. This dialog contains the narrative background of the quest and challenges the player to complete a particular task. After the player accepts the quest, the star icon over the NPC's head changes to a red question mark to indicate that the quest is in-progress.

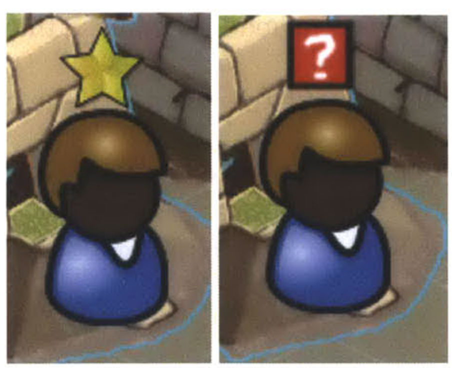


Figure 3. An NPC with an inactive quest (left) and an in-progress quest (right).

ii. Turn-in

Figure 4. Two different types of quest turn-in dialogs.

Once the player has acquired the items or information requested, the player can click on the NPC again to complete the quest. Depending on the nature of the quest, the turn-in dialog may contain different things, such as slots for a player to drag items from the inventory or a multiple-choice question for the player to answer.

iii. Feedback

Finally, after the player has turned in the quest, the dialog will display a message indicating how well the player performed. If the player did not complete the quest successfully, she will typically be given the option to attempt the quest again, sometimes with a clue to steer her in the right direction.

B. SmartFoxServer

The multiplayer extension to the prototyping system uses SmartFoxServer (SFS) for its back-end communication. SmartFoxServer is a client-server platform for online multiplayer games. It is compatible with ActionScript3 and offers a free community edition that can support up to 100 players, which is more than adequate for the Radix Endeavor's testing and prototyping needs. SmartFoxServer's lightweight "chat room"-inspired protocol consists of three major components: users, rooms, and messages.

1. Users

Players are represented on the server by User objects. Each User has a unique ID that identifies them to other players. User Variables can be used to store data associated with a particular user, such as current location or avatar type.

2. Rooms

Rooms are objects used to group Users together; Users who are in the same Room can interact with each other. Persistent data pertaining to a Room can be stored in Room Variables. In the prototypes, each Room corresponds to a single scene. Every time a new multiplayer scene is added to the prototype, a corresponding room must be created on the server. SmartFoxServer offers administration tools with a graphical interface for creating rooms and room variables.

3. Messages

SFS Messages are the medium for transmitting events between connected users. By default a message sent by a User is broadcast to all other Users in the same

Room, but messages can also be sent to all logged-in Users or to one particular User. Messages are essential for keeping games synchronized and for multiplayer interaction.

C. System Architecture

1. Server

SmartFoxServer is installed and running on one of the STEP lab machines, and is accessible at the IP address 18.7.25.15.

2. The Login Process

When users log in, their client first connects to the server, where they are represented as a User object and assigned to a particular Room. Then their client sends a series of requests to synchronize with other in-progress games. The technical details of the process are outlined below.

i. Connection

The prototypes initially load in single-player mode. When a player wishes to connect to the server, she clicks the "Multiplayer Connect" button, located on the left. After she clicks this button, the client creates a new SmartFox instance and establishes a connection to the server. Once a connection is successfully established, a dialog launches where she can choose a screen name; this is the name that will appear to other players when they interact with her avatar.

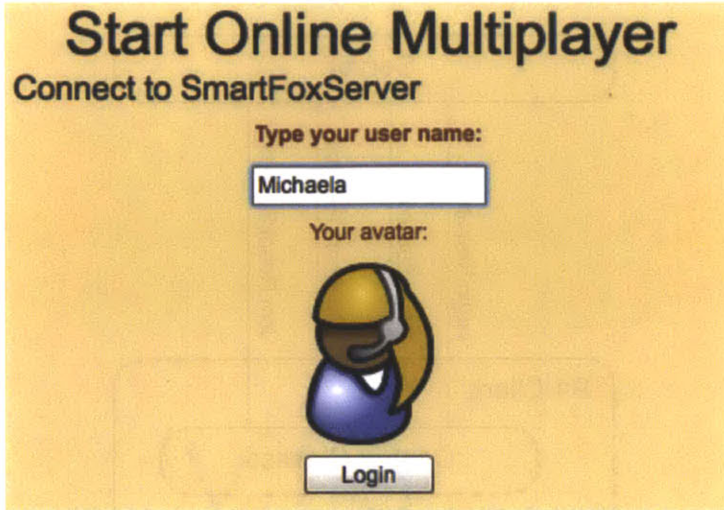


Figure 5. The login dialog.

Since these prototypes are for testing purposes only there is no need to have persistent user accounts, and a password is not necessary. Students simply log in with whatever username they choose, as long as another player with the same username is not already logged in. (Players who try to log in with a username that is already in use receive an error dialog: "Sorry, that username is already taken. Please log in with a different name.") An avatar is randomly selected for the player from a library of 16 different possibilities.

ii. Login

After the player clicks "Login", the client sends a login request to the server. This is followed by a request to join the Room corresponding to the current scene that the player is in.

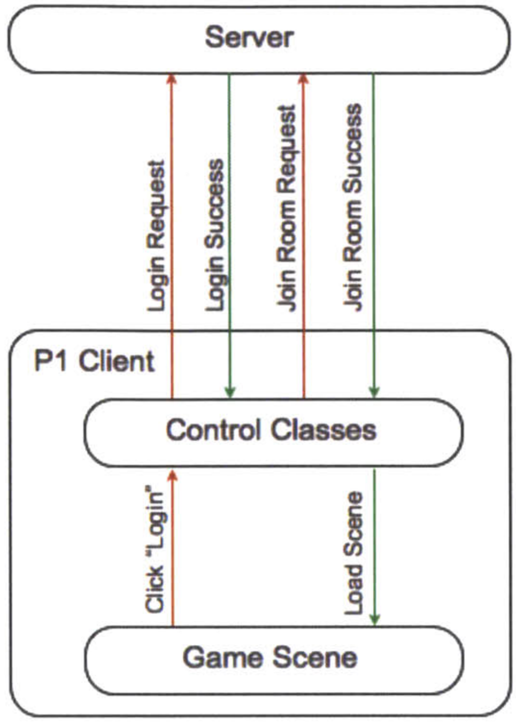


Figure 6. Logging in with SmartFoxServer.

iii. Load Users

Once the player has joined the Room, the client sends a request to the server for a list of all other Users in the room, along with their usernames, avatar types, and locations. Their avatars are added to the player's map. The client also asks the server to broadcast a message with the newly joined player's data, so that she will appear on other players' maps.

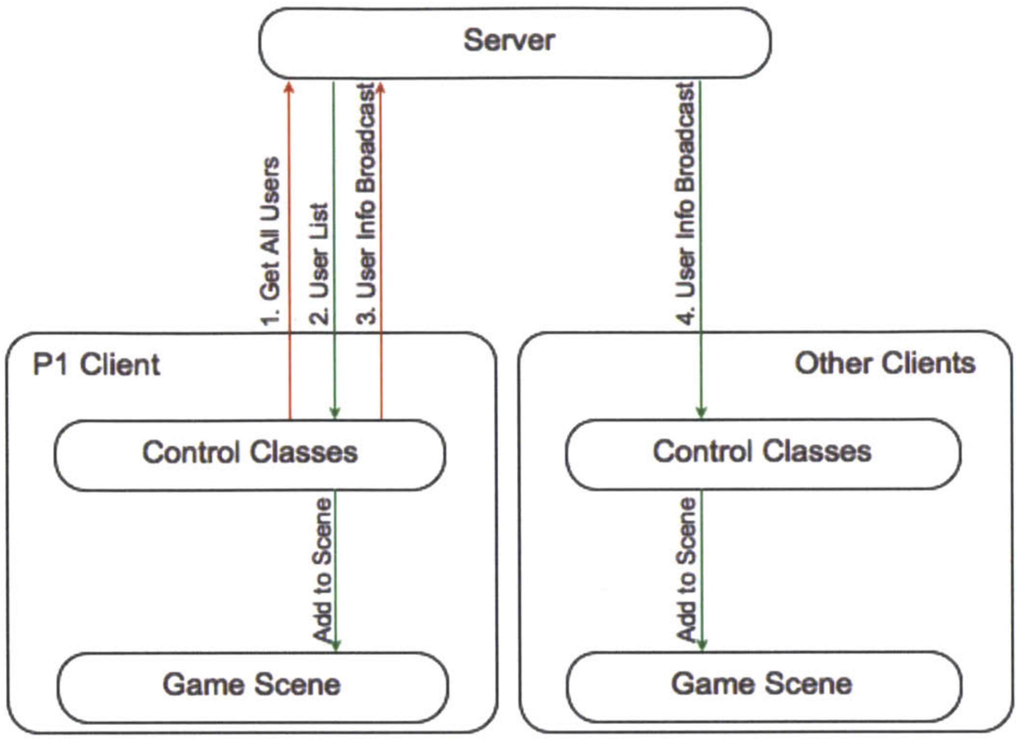


Figure 7. Populating the game scene with other users' avatars.

iv. Match State

There may be shared items in the scene whose states need to be updated. For example, if a field of flowers is a shared resource, and other players have picked some of the flowers, the current player's game scene needs to be updated to reflect this common reality. To synchronize the game state, the client selects another User in the Room at random and sends a request to that User for the current scene state, and then uses this information to update the scene accordingly.

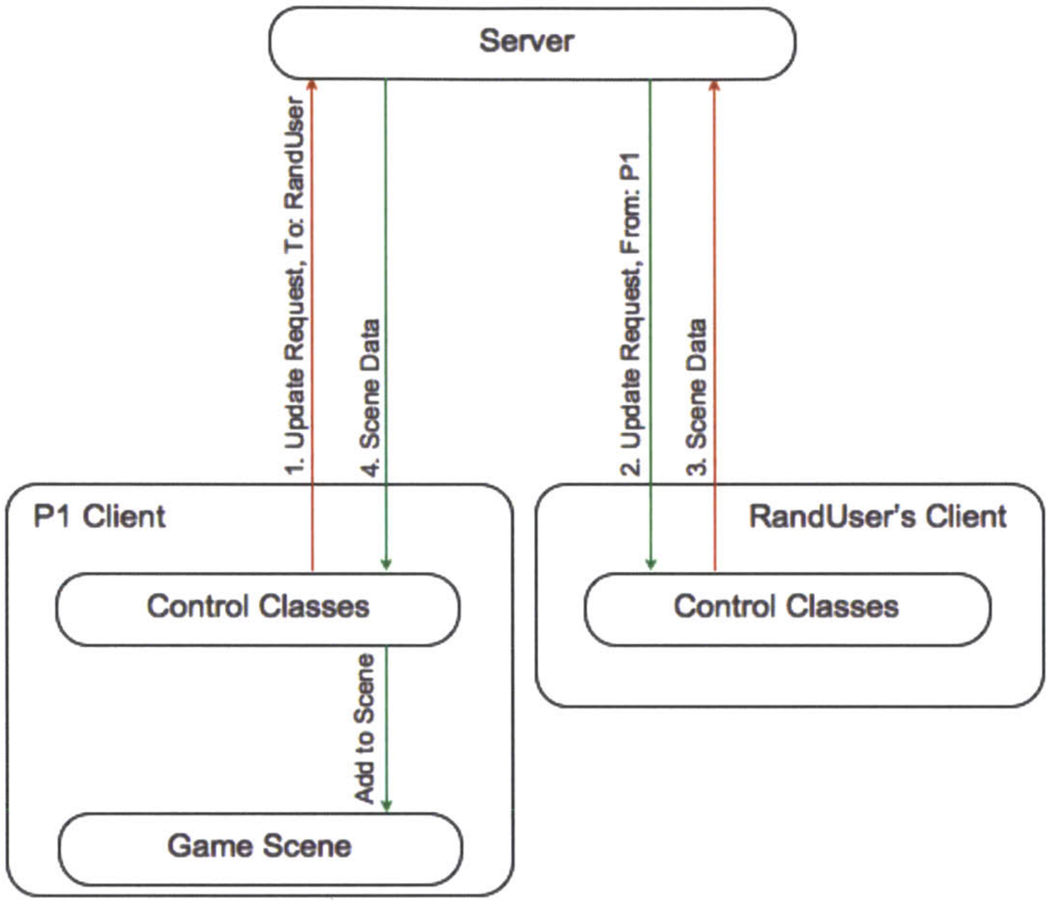


Figure 8. Matching initial game state to an in-progress multiplayer game.

v. Dynamic Loading

Finally, some scenes have dynamically loaded elements, which are dependent on the values of persistent Room Variables stored on the server. For example, an NPC may alternate the quests he assigns to players as users enter the room, or different players may find different collectable items on the ground. For these elements, the client must send a request to the server to get the values of the Room Variables indicating which variant to load. Because such variable elements may be highly idiosyncratic to a scene, the loading for dynamic elements is handled on a case-by-case basis.

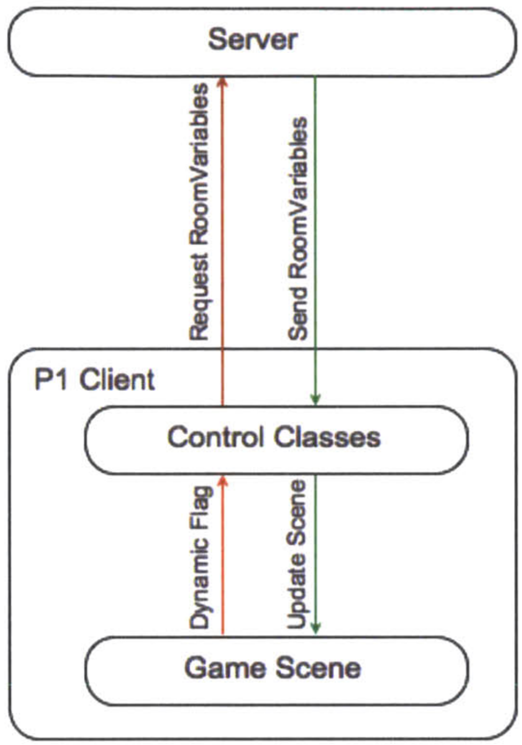


Figure 9. Dynamically loading variable scene elements.

Whenever a player changes scenes, she is logged out of the old Room and logged into a new one, and steps iii-v must be repeated.

3. Synchronization

The bulk of the prototype code is run in the browser. Once the initial login process is complete, most game events can take place client-side, with server calls only being made as needed. Whenever a player takes an action that requires other players' games to be updated, such as moving his avatar or collecting a shared item, the client sends a SmartFoxServer message containing the type of action taken along with any additional data. For example, each time a player moves his avatar, a message with type PLAYER_MOVED is sent to the server, along with the username of the player who moved, and the coordinates of his new location. SmartFoxServer then broadcasts this message to all other players who are in the Room.

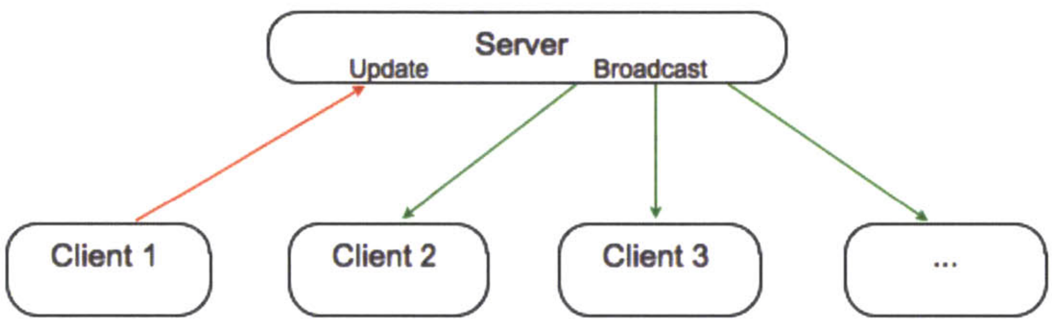


Figure 10. Keeping clients synchronized by sending updates as-needed.

4. Control Classes

On the client side, there are three classes— MultiplayerControl, CCControl and MControl— that listen for incoming messages. When a SmartFoxServer message is received, the control classes extract the message data and update the game display accordingly. In the example above, once a PLAYER_MOVED message is received, CCControl updates the location associated with that user and triggers an animation of their avatar moving from the old location to the new one.

MultiplayerControl brokers the initial login process. CCControl handles universal functions like scene loading and avatar movement, while MControl handles more specialized functionality such as updating the state of a shared tool or loading dynamic elements for specific scenes.

IV. QUEST PROTOTYPES

A prototype quest is a self-contained learning task with a goal and a set of tools associated with it. With the multiplayer prototyping framework in place, the next step is to design and implement quests that take advantage of the new functionality and explore different multiplayer concepts. In addition, these quests can be tested alongside analogous single-player quests in order to gauge whether multiplayer interaction has a positive effect on player learning and engagement. This section outlines the prototype quests that were created for this thesis project, including narrative, design, and implementation details.

A. Learning Objectives

The marketplace quest prototypes require players to apply quantitative reasoning to a complex system. They were designed to be part of the Radix Endeavor's algebra curriculum, and to focus particularly on the concepts of unit conversion, systems of equations, and variables. Players who successfully complete these quests should be able to use unit conversions to solve problems, and solve simple linear equations.

B. Marketplace

The quests all take place in the Marketplace, an open-air bazaar with a barter economy. Players can exchange goods with NPC's, trade with other characters, and redeem items for gold bars at the local Pawn Shop.

1. Narrative

Players are given a collection of coins and starter items and asked to take them to the Marketplace and make an advantageous series of trades. The player may be asked to maximize the value (in gold bars) of the items they get, or to obtain a specific goal item as cheaply as possible. The Pawn Shop, which offers gold bars in exchange for items, provides a baseline measurement of the relative values of objects. However, the trades offered at the vendors' stalls do not necessarily align with this exchange rate. Some vendors may offer bargains while others try to cheat the player with bad deals.

2. Vendors

Vendors are a special type of NPC that offer items to the player in exchange for other items. Clicking on a vendor opens a dialog with a list of that vendor's merchandise, along with the items that they are willing to accept as payment.

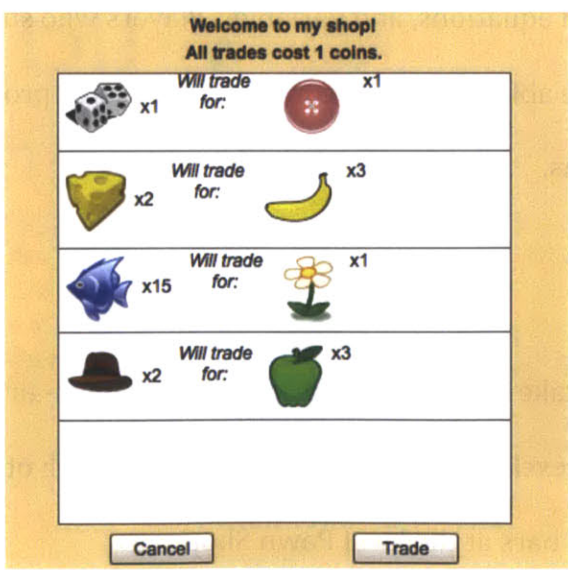
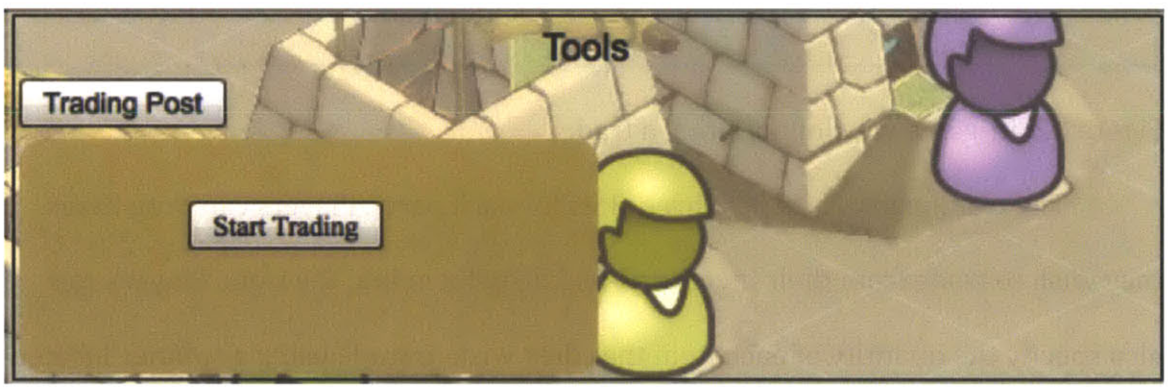


Figure 11. A vendor dialog. The left column shows items that the vendor can give the player, and the right column shows the items that the player must give in exchange.

Each trade with a vendor carries a tax of 1 coin. Note that coins are distinct from gold bars, and one cannot be exchanged for the other. This is so that, while players sell items at the Pawn Shop to increase their gold, the total number of trades that a player can do to complete a quest is still limited by the number of coins she is given to start with.

3. Trading Post

The trading post is a tool, only available in multiplayer mode, that enables players to trade items with other players. When the tool is enabled, players can click on other players to initiate a trade.



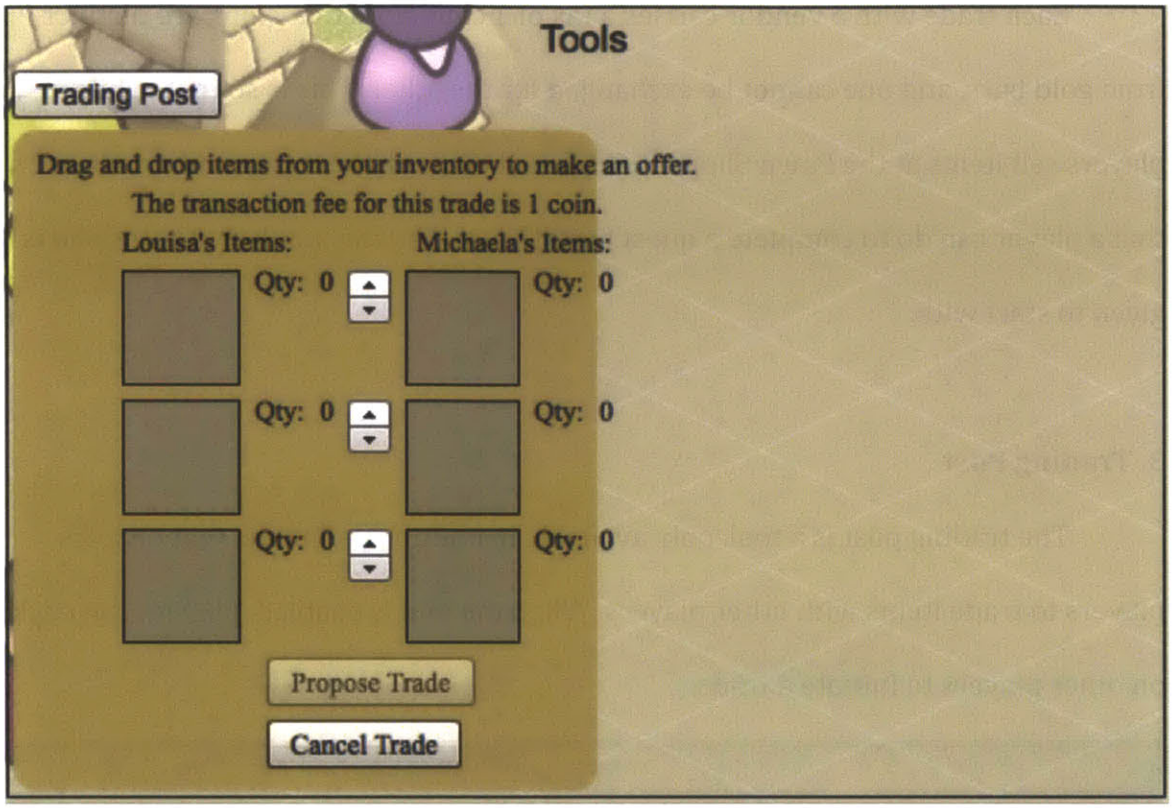


Figure 12. In-game view of initiating a trade.

The trading interface has three slots for each party. Players can drag items they wish to trade from their inventory, and drop them into the slots. Players can also specify the quantity of each item that they wish to trade using a spinner located next to the slot.

Players can use the text chat interface, located in the lower-left-hand corner, to negotiate. When a player is satisfied with the terms of the trade, he can "Propose a Trade." This locks both players from making changes to the quantity or types of items in the trading post. The second player can then "Accept" the trade, or go back and "Edit" the trade to re-open the negotiation phase.

Finally, if both players confirm the trade, then the trade is completed and the items are exchanged. Either player can also choose to "Cancel" the trade at any time, in which case the trade is terminated and both players' trading post tools are reset.

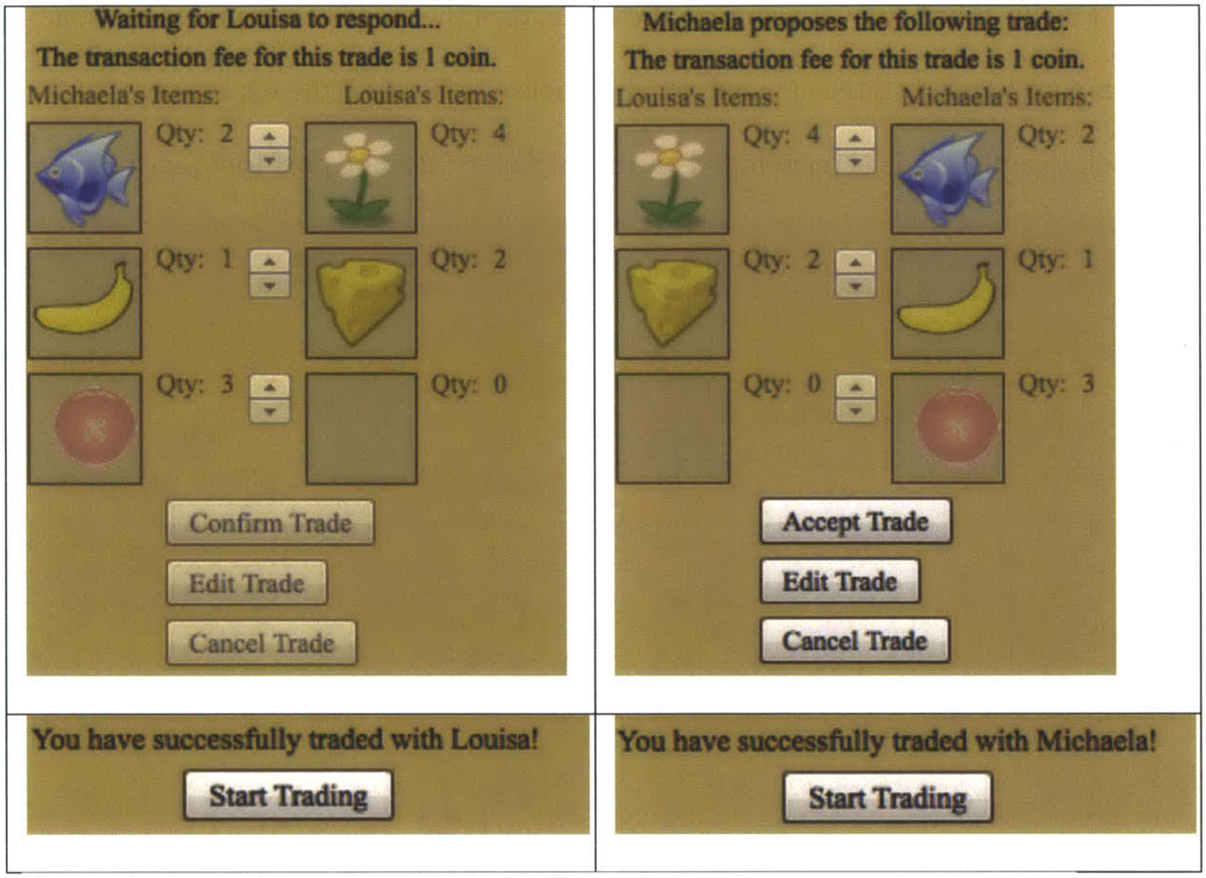


Figure 13. Two player views of the trading propose/accept/confirm process.

C. Single-Player Quests: Marketplace

In this quest line, an NPC ("Marketplace Michaela") gives the player a collection of low-value items and coins and asks the player to take these to the marketplace and return with gold bars. Players must compare the pawnshop prices with the exchange rates offered by different vendors in order to "trade up" for items of higher value and end up with as many gold bars as possible. Successful players are congratulated for their keen trading skills. If a player returns with less than the optimal number of gold bars, Marketplace Michaela thanks the player for his contribution but mentions that she has heard of other traders getting even better deals at the marketplace, and offers the player a chance to try again.

For these four quests, the player works alone and is unable to interact in-game with any other players. All four quests use the following pawnshop exchange rates:

1 button	1 gold bar
1 fish	2 gold bars
1 apple	4 gold bars
1 banana	5 gold bars
1 flower	10 gold bars
1 dice	20 gold bars
1 hat	25 gold bars

1. Quest 1

Objectives: This quest introduces players to the marketplace trading mechanic and challenges them to use quantitative reasoning skills to maximize the number of gold bars they receive.

Starting items: 10 buttons, 15 coins

Vendor Name	Player gives	Player Receives
Alice	5 fish	1 hat
	3 apples	1 dice
	2 bananas	2 flowers
Joe	2 buttons	1 apple
	3 bananas	1 dice
	1 hat	2 flowers
Bob	1 button	1 fish
	6 apples	6 bananas
	1 dice	1 hat
Sally	2 fish	1 banana
	1 hat	3 bananas
	1 dice	2 flowers

Solution: In the optimal solution, the player can get 50 gold bars. This can be achieved by trading 10 buttons for 10 fish (via Vendor Bob), and 10 fish for 2 hats (Vendor Alice). Then trade the hats in for gold bars.

2. Quest 2

Objectives: This quest closely resembles the previous quest, with the difference that none of the vendors now offer hats. This forces players to find an alternative solution within a system with many of the same parameters. The aim of this variation is to probe student understanding of the task more deeply, as players who employed more simplistic heuristics to complete the previous quest (such as "Trade for the most valuable item") will be forced to rethink their strategy. This is also the only quest where there are multiple ways to acquire the maximum number of gold bars, some costing more coins than others, so this is a good test to see whether students are considering efficiency when they trade or if they are merely looking at the final payout.

Starting items: 10 buttons, 15 coins

Vendor Name	Player gives	Player Receives
Alice	3 apples	1 dice
	2 bananas	2 flowers
Joe	2 buttons	1 apple
	3 bananas	1 dice
Bob	1 button	1 fish
	6 apples	6 bananas
Sally	2 fish	1 banana
	1 dice	2 flowers

Solution: In the optimal solution, the player can get 28 gold bars (and save 9 coins), with the following sequence of trades— 10 buttons for 5 apples (Joe), 3 apples for 1 dice (Alice). Then trade the dice and apples in for gold bars.

3. Quest 3

Objectives: This quest has the same vendor configuration as Quest 1, but players are given fewer materials to start with. The goal of this variation is to get students to think more carefully about the length of the path their trade sequences take, in addition to the items that they end up with.

Starting items: 4 buttons, 6 coins

Vendors: The vendors for this quest are identical to those in Quest 1.

Solution: In the optimal solution, the player can get 10 gold bars, with the following sequence of trades— 4 buttons for 4 fish (Bob), 4 fish for 2 bananas (Sally). Then trade the bananas in for gold bars.

4. Quest 4

Objectives: In this quest, rather than a flat tax of 1 coin/trade, each vendor charges a different amount. Thus, vendors who offered good value before may no longer be such a good deal. This quest challenges students to add yet another dimension to their problem solving: they must now consider not only the relative values of items and the number of trades it takes to obtain them, but also *who* they are trading with. In mathematical terms, this is equivalent to adding a multiplicative factor to a subset of equations in the system, or adding different weights to the edges of a graph.

Starting items: 10 buttons, 15 coins

Vendor Name	Trade Tax
Alice	1 coin
Joe	3 coins
Bob	2 coins
Sally	1 coin

Items and barter prices are identical to those in Quest 1.

Solution: In the optimal solution, the player can get 32 gold bars, with the following sequence of trades— 5 buttons for 5 fish (Bob), 5 fish for 1 hat (Alice), 2 buttons for 1 apple (Joe). Then trade the hat, apple, and remaining buttons in for gold bars.

D. Multiplayer Quests: Trading Post

The multiplayer quests are based on the same Marketplace scenario as the quests described above; however, for these quests players need to work with other players to achieve the optimal results. In many games, MMO's in particular, multiplayer contact occurs primarily via real-time combat. However, since the game play of Radix is focused more on puzzle-solving and creative analysis, it was necessary to explore alternative forms of multiplayer interaction. Each multiplayer quest is an experiment in a different type of multiplayer mechanic.

The multiplayer quest line introduces the Trading Post tool. Before starting a quest, players are prompted to log in. Once online, players can see each other moving around the marketplace and have the option of using the trading post to swap items with one another. Players can also communicate using text chat.

1. Multiplayer Design

From an educational standpoint, multiplayer games have several advantages over single player ones. The ability to play with peers can increase students' interest and engagement in the game, and shared in-game experiences can spark dialogue about the material outside of the classroom. In addition, a multiplayer environment provides a built-in mechanism for human feedback to assist students who are struggling. We have seen from commercial MMO's that multiplayer gameplay can be a source of rich and compelling player interactions, and with the advent of mobile computing and the proliferation of ubiquitous Internet access, creative applications of social gaming have been growing in popularity.

However, this innovation has been slow to make its way into the educational sector. Most learning games available today have patterned themselves after a schooling system that has traditionally focused on individual assessment. As a result, few models of educational multiplayer gameplay currently exist. Because of this, I faced significant challenges in designing multiplayer interactions that would contribute to learning in an effective and meaningful way.

i. Cooperative vs. Competitive

One of the first design choices I faced was whether to prototype cooperative quests, where players work together to achieve their goals, or competitive quests, where each player is trying to outdo the others. I chose to focus on cooperative rather than competitive play for several reasons. First, cooperative play better embodies the collaborative nature of scientific research. Cooperative play is also more copacetic with Radix's goals as a tool of instruction. Radix is intended to be used in an educational setting, and each quest functions to some degree as an evaluation of students' abilities. If quests were competitive, performance would be more a function of whom players were competing against rather than how well they grasped the material, and would therefore not be a fair assessment.

The literature confirms that cooperative play is likely to be the best choice for a game like The Radix Endeavor. Research has demonstrated a link between competitive play and increased hostility and verbal aggression between players, while cooperative play is associated with heightened feelings of camaraderie and unity (Eastin, 2007). In one study, participants who played the first-person shooter game Halo II cooperatively showed more increased cooperative and prosocial behavior after playing compared to participants who played competitively (Ewoldsen, 2012). Cooperative quests should therefore create more positive player experiences, and may alleviate some parental concerns over cyber-bullying. Cooperative learning has also been found to promote greater achievement, more successful problem solving, and increased student-student interaction compared to competitive or individualistic instruction in digital settings (Johnson, Johnson and Stanne, 1986).

ii. Communication is Key

To me, the promise of player-to-player communication is one of the most compelling reasons to apply multiplayer dynamics to learning games. Advice from a friend on how to tackle a tricky puzzle is likely to be more relevant and more individually tailored than the feedback generated by an automated hint system, ultimately leading to better comprehension on the part of the student seeking help. For stronger students, being called on to provide such feedback will force them to articulate their reasoning, enabling them to achieve a deeper understanding of the subject matter and giving them an opportunity to develop the "soft" communication skills that are so critical to STEM fields. For students of equal ability, being able to bounce ideas off of one another can lead to greater insights than either could achieve if they were working alone. This is the secret behind great universities and think tanks: the emergent power of bringing many creative minds together.

I wanted to encourage this kind of synergy by creating quests where effective communication was not only possible, but necessary for players to succeed. This is one reason why I decided to make the multiplayer interaction synchronous rather than asynchronous. Asynchronicity has convenience in its favor, allowing students to play at their own pace and on their own time— hence the popularity of turn-based games like Words with Friends. But only synchronous play has the benefit of enabling players to ask one another questions, facilitating "just-in-time" learning and peer instruction. This is also why the mechanics of the quests center on trading and negotiation. For effective negotiation, clear and constant communication between partners is crucial.

If a multiplayer collaboration focuses merely on exchanging items or special abilities, the interaction becomes almost mechanical, with other players functioning as simply one more "tool" in a gamer's toolbox. With the addition of communication, they become fellow learners to engage with in the process of planning and discovery.

iii. Game Theory

On a mechanical level, designing cooperative quests such that working together benefits both parties also raises a number of interesting challenges. In many ways, competitive quests are a much more straightforward extension of individual play. If participants are all striving for the same goal or the same pool of resources, then the aim of each player is simple: to do the best, get the most, or be the fastest. With cooperative quests, the division of responsibility among individual players is much more subtle. The quest must be structured such that working together is essential to achieve the optimal result— even in a single-discipline task like the marketplace quest line, where players are bringing a similar set of knowledge and skills to the table. At the same time, I felt it was important for players to be able to solve the quest on their own. I wanted them to really think about *why* they were working together and how that could benefit them, rather than collaborating simply because the game seemed to demand it.

I quickly realized that the format of the single-player trading quests was not conducive to interesting multiplayer dynamics. If students are simply trying to maximize the number of gold bars they can obtain, then every trade between students has a clear "winner" and "loser" based on the gold value of the items exchanged. And even if the students traded items of equal value, the fact that both are working toward the same end goal of maximizing their gold bars means makes trading a zero-sum proposition— if one student came out ahead as the result of the trade, the other would necessarily have done better by not trading at all.

To fix this issue, the structure of the multiplayer quests differs slightly from that of the single-player prototypes. The pawnshop and gold bars have been eliminated. Instead, players are given a set of items and coins to start with and are asked to trade for a specific set of goal items, while spending as *few* of their coins as possible. By placing the emphasis on *saving* coins, the quest becomes a challenge of efficiency and students can make trades that are mutually beneficial. The new format also allows for different types of cooperation, because instead of competing for the same resources (gold), students can be given different, complimentary goals.

2. Quest 1

Objectives: This quest investigates a "divide-and-conquer" multiplayer mechanic.

Players are given the same resources to begin with and are all working towards the same goal. They must figure out how best to divide the task so that each player achieves a better outcome than if he or she were working alone.

Starting Items: 10 buttons, 20 coins

Goal Items: 1 hat, 1 dice

Vendor Name	Player gives	Player Receives
Alice	1 flower	1 banana
	2 cheese	5 apples
	10 fish	2 hats
Bob	1 button	1 fish
	4 dice	5 hats
	2 apples	3 dice
	1 cheese	1 apple
Charles	1 button	2 flowers
	2 bananas	2 cheese
	1 flower	4 fish

Solution: Individually, the best players can do is trade 1 button for 2 flowers (Charles), 2 flowers for 2 bananas (Alice), 2 bananas for 2 cheese (Charles), 2 cheese for 5 apples (Alice), and 4 apples for 6 dice (Bob), and 4 dice for 5 hats (Bob), then turn in 1 hat and 1 pair of dice. This costs them a total of 8 coins, saving 12. However, they will have many extraneous items. This should help tip players off that it might be more efficient to pursue just one of the goal items while a partner goes after the other.

Working in pairs, one player can trade 2 buttons for 4 flowers (Charles), 3 flowers for 12 fish (Charles), and 10 fish for 2 hats (Alice). Another player can trade 1 button for 2 flowers (Charles), 2 flowers for 2 bananas (Alice), 2 bananas for 2 cheese (Charles), 2 cheese for 5 apples (Alice), and 2 apples for 3 dice (Bob). If Player 1 trades a hat for one of Player 2's dice, they can each obtain the goal while only spending 7 coins, and saving 13.

3. Quest 2

Objectives: This multiplayer quest explores a "resource instancing" approach. All players are working in the same marketplace setting with the same set of vendors; however, players receive a *different* set of resources to begin with and are each working toward different goals. Players can exploit this asymmetry to find items that are of little value to them but are very valuable to others, and then use these items to negotiate advantageous trades.

Starting Items: 10 buttons (Player 1) OR 10 flowers (Player 2), 25 coins

Goal Items: 10 hats (Player 1) OR 10 apples (Player 2)

Vendor Name	Player gives	Player Receives
Alice	1 flower	1 banana
	2 cheese	5 apples
	3 fish	1 hat
	9 fish	3 hats
Bob	1 button	2 fish
	4 dice	5 hats
	2 dice	2 cheese
	1 banana	1 apple
Charles	1 button	1 dice
	3 bananas	2 cheese
	1 flower	15 fish
	3 apples	2 hats

Solution: Working alone, the best Player 1 can do is to trade 8 buttons for 8 dice (Charles), and 8 dice for 10 hats (Bob), saving 15 coins. The best Player 2 can do alone is trading 6 flowers for 6 bananas (Alice), 6 bananas for 4 cheese (Charles), and 4 cheese for 10 apples (Alice), also saving 15 coins.

If two players with complimentary goals pair up, they can save more coins. There are several possible solutions to this quest, one of which is outlined here. Player 1 can trade 4 buttons for 4 dice (Charles), and 4 dice for 4 cheeses (Bob). Player 2 can trade 2 flowers for 30 fish (Charles), and 30 fish for 10 hats (Alice). Then, using the Trading Post tool, the players can swap 4 cheeses for 10 hats. This gives Player 1 everything she needs. Finally, Player 2 can trade 4 cheeses for 10 apples (Alice) to get his goal. In this solution, Player 1 saves 18 coins and Player 2 saves 16 coins.

V. FEEDBACK

This section summarizes the feedback obtained from the initial play testing sessions for the single- and multi-player prototypes. Due to the small sample size and lack of rigorous testing methodology, these results are naturally preliminary and don't represent a formal evaluation of the game. However, they proved useful as an early evaluation of the new mechanics and in indicating fruitful directions for future development work.

A. Methodology

The single-player and multiplayer prototypes were play-tested on December 13, 2012, and April 25, 2013, respectively, with the same group of students at Josiah Quincy Upper School. Students worked through the quests in a shared computer lab during two hour-long sessions. For the multiplayer quests, students were divided into 3 different instances of the game so as not to overwhelm the servers, and were intentionally seated apart from classmates who were playing the same instance, in order to better simulate a real online multiplayer gaming experience. While they were doing the quests, students answered oral questions about their strategies and thought processes. Each student also completed a written survey (Appendix A) at the end of each play test session. During play, students were permitted to ask questions and to speak to one another.

B. Single-Player Quest Feedback

Students generally took a few minutes to become familiar with the task and to figure out how to execute a trade. Many students started off by simply selling all their buttons at the pawnshop, but after getting feedback from an NPC ("I've heard that some people managed to get even more gold!") they quickly realized that they needed to trade with the vendors. Students got excited when they saw their friends trade for different items, and wanted to know where to find them. Students seemed to find the trading interface easy to use and understand, although some expressed frustration that they couldn't open several vendors' shop windows at once in order to compare exchange rates side-by-side. A common remark was that it was difficult to remember which vendors sold which items. As a result of this, the appearance of the vendors was altered in the multiplayer prototypes to make them visually distinct.

Most students were successfully able to complete the quest by framing it as a unit-conversion task. Although they did not write out any explicit equations, students understood that they needed to look at the pawnshop to determine the value of items in gold bars. Seeing that a vendor was offering 1 pair of dice (20 gold bars each) in exchange for 2 flowers (10 gold bars each), one student scoffed, "What! They're the same!" Students were also able to quickly recognize very good (low value for high value) and very bad (high value for low value) trades. However, students seemed to struggle somewhat with the dual goal of trying to maximize gold bars *and* spend as few coins as possible. Students clearly realized that, because trades carried a tax of 1 or more coins, they had to be efficient about their trades. But many students skipped Quest 2 after getting the maximum number of gold bars without trying to achieve the goal in fewer trades.

Although these were intended as single-player quests, some interesting multiplayer dynamics emerged during play testing. Many students formed mentor-mentee groups, with a student who had already completed a quest successfully walking his or her neighbors through the process. Students also seemed to be more motivated to get more gold or to try different trades when they saw their peers do these things.

C. Multiplayer Quest Feedback

As with the previous session, students took a little while to get acclimated to the new tools. Especially because they were already familiar with the Marketplace, students tended to just jump in and start trading with vendors, without realizing that they could initiate trades with other players. Many students needed the trading post tool to be pointed out to them before they attempted to trade. Once they realized trading was an option, students seemed to find the trading interface fairly intuitive. Finding prospective trading partners and negotiating with them, however, proved to be a bit of a challenge with the public chat window. Students tended to have more success discussing and completing trades with other students who were seated physically close to them, talking through trades out loud with their partner rather than using the in-game chat interface.

Even once students fully grasped the trading process, some students elected not to trade, or started trades but then changed their minds and cancelled before exchanging any items. These students continued to work alone despite NPC feedback ("Maybe you should try working together with other wandering traders") and explicit encouragement from the play test moderators. When asked why they didn't want to trade, many students expressed a desire to be self-reliant: "I don't want to be a leecher!" one student exclaimed. "I don't want to take other people's hard-earned items." Another student commented on the written survey that the student preferred the single-player quests because "[I] don't have to rely on others."

Of those who did enjoy trading, students said that they liked the multiplayer because it was "more fun", was easier, and saved time. These students said they liked working together and helping others. On the written survey, one student noted that "When I had enough for my quest, I gave my leftovers away" even though it did not impact her performance. In total, about half (7) of the 15 students surveyed said that they preferred the multiplayer quests to the single-player ones. Four students said they preferred the single-player quests, and four students liked both kinds of quests equally.

Almost all students commented that they enjoyed seeing other students' avatars and using the player chat to talk to one another, regardless of whether they used the trading tool or preferred to work alone.

VI. FUTURE WORK

Given the feedback obtained during play testing, one useful area of future work would be to improve the in-game chat interface. As discussed earlier, communication is a key part of the learning process, and is accordingly crucial for success in Radix quests. With the single-player prototypes, students had lively discussions with their neighbors about the game and swapped strategies and tips with their friends. Ironically, during multiplayer testing these kinds of conversations were inhibited because students were often playing in different instances than the classmates sitting next to them, and the in-game text chat proved to be a poor substitute for these kinds of spontaneous spoken discussions. Many students struggled to make trades in the prototypes because it was difficult to figure out who each player was, what items they wanted and what they were willing to trade. In particular, enabling private chat between a pair or group of players would make communication and negotiation during multiplayer quests much easier by making it clear who players were working with, and providing a communication channel where they could have a focused discussion without outside distraction. An automatic player-matching tool like that available for World of Warcraft dungeons might also be useful for players who are having difficulty connecting with other players. The two quests tested were designed with pairs of cooperating players in mind, but it turned out to be difficult for players to discover this on their own. A matching tool would remove this obstacle and allow players to focus more on the really important learning tasks.

In addition, developing a narrative that encourages cooperation could alleviate some of the philosophical issues that some players had with trading. A number of students seemed to view working with other players as "cheating", wanting to complete the quests on their own even when they were encouraged to trade. If the game's narrative emphasized more strongly that all players are members of the Curiosi and are part of the same "team", perhaps some of the reservations students had about teamwork would disappear.

It would also be interesting to explore several other variants of cooperative multiplayer interaction. Perhaps players who were averse to sharing physical resources might not have a problem with sharing *information*. In this vein, it might be informative to prototype a quest where players were given the same tools to complete a task, but incomplete knowledge, and players need to share the clues they've been given before they can find a solution. It would also be interesting to experiment with cooperative interdisciplinary quests, which require the students working together to have mastery of different fields. I started work on a genetics/algebra prototype where players breed flowers and trade them in the marketplace, but wasn't able to complete it in time for testing. In addition, the quests prototyped here were all synchronous multiplayer quests. However, it may not always be practical for students to do a quest simultaneously, especially if it is given as a homework assignment for students to complete on their own time, so asynchronous multiplayer is an additional area for further exploration. One idea discussed in lab for asynchronous cooperative quests was to furnish marketplace stalls with "message boards" where players could warn others about unscrupulous vendors or highlight good deals. The addition of rudimentary Artificial Intelligence that allowed players to haggle with vendors might also enrich the experience, since players could use the message boards to share the details of interactions that would not be immediately apparent from the posted price sheets.

Finally, from a technical standpoint, there are also aspects of the prototyping system that could be further refined. To be truly designer-friendly, the system should enable designers to prototype any scene simply by writing an XML representation of the scene and creating a corresponding Room in SmartFoxServer (along with any room variables). However, scenes that contain instanced items still require writing a small amount of custom code in order for the scene to load correctly. A general-purpose script that could dynamically load elements entirely from XML and room variables would streamline the design process greatly. In addition, quest turn-ins have also all been custom-built for each quest up to this point. A handful of reusable modules for popular quest turn-in formats (multiple-choice questions, drag-and-drop items, etc) that could be instantiated from XML would also ease the burden on designers.

VII. CONCLUSION

The Radix Endeavor is an innovative approach to education in STEM fields, areas in which students in the United States consistently underperform. By leveraging the popularity and uniquely interactive potential of social gaming, Radix aims to inspire students to develop a deeper interest in the material, and to carry those interests outside of the classroom. This paper described extensions to the game's software infrastructure to support multiplayer gaming, and outlined the design and play test feedback gleaned from several early multiplayer prototypes. Future work aims to extend the Radix Endeavor's meta-tools to facilitate multiplayer communication and to explore further variations of multiplayer interaction. Ultimately, it is hoped that this work will be incorporated into a later iteration of the final production version of the game.

REFERENCES

Atlantis Remixed Project Team. Atlantis Remixed. Retrieved May 21, 2013: atlantisremixed.org

Bainbridge, William Sims (2007, July 27). The Scientific Research Potential of Virtual Worlds. In Science Magazine (vol. 317, pp. 472-76). Retrieved May 8, 2012 from Science Magazine: http://www.sciencemag.org/content/317/5837/472.full.pdf

Eastin, M. S. (2007, October). The influence of competitive and cooperative group game play on state hostility. In Human Communication Research (Vol. 33, pp. 450-66). Retrieved May 21, 2013 from Wiley: http://onlinelibrary.wiley.com/doi/10.1111/j.1468-2958.2007.00307.x/full

Ewoldsen, David R., et. al. (2012). Effect of Playing Violent Video Games Cooperatively or Competitively on Subsequent Cooperative Behavior. In Cyberpsychology, Behavior and Social Networking (Vol. 15, No. 5). Retrieved May 21, 2013 from the University of Alabama: http://osil.psy.ua.edu/pubs/halo_published.pdf

Gee, J. P. (2007). What Video Games Have to Teach Us About Learning and Literacy. New York, NY: Palgrave Macmillan.

Hass, J. (2008, May 15). About the Education Arcade. Retrieved May 21, 2013, from The Education Arcade: http://educationarcade.org/about

Hung, K., C. Kinzer & C. Chen (2009). Motivational Factors In Educational MMORPGs: Some Implications For Education. In Transactions On Edutainment III (pp. 93–104). Retrieved May 8, 2013, from SpringerLink: http://www.springerlink.com/content/j52l613178953788/fulltext.pdf

Johnson, R. T., D. W. Johnson & M. B. Stanne (1986). Comparison of Computer-Assisted Cooperative, Competitive, and Individualistic Learning. In American Educational Research Journal (Vol. 23, No. 3, pp. 382-392) Retrieved May 6, 2013, from JSTOR: http://www.jstor.org.libproxy.mit.edu/stable/1163055?seq=1&

Langdon, David et. al. (2011, July). STEM: Good Jobs Now and for the Future. Retrieved May 18, 2013, from U.S. Department of Commerce - Economics and Statistics Administration: http://www.esa.doc.gov/sites/default/files/news/documents/stemfinalyjuly14.pdf

Morella, Michael (2013, January 31). Many High Schoolers Giving Up on STEM. Retrieved May 18, 2013, from US News and World Report:

http://www.usnews.com/news/blogs/stem-education/2013/01/31/report-many-high-schoolers-giving-up-on-stem

National Science Foundation (2012). Elementary and Secondary Mathematics and Science Education. Retrieved May 18, 2013, from the National Science Foundation: http://www.nsf.gov/statistics/seind12/c1/c1s1.htm#s3

Riegle, R. P. & W. A. Matejka (2006). The Learning Guild: MMORPG's as Educational Environments. Paper presented at the 22nd Annual Conference on Distance Teaching and Learning. Retrieved May 8, 2012: http://www.uwex.edu/disted/conference/Resource_library/proceedings/06_4095.pdf

River City Research Team (2007). River City Curriculum- Overview. Retrieved May 8 2012 from Harvard University: http://muve.gse.harvard.edu/rivercityproject/curriculum.htm

Scheller Teacher Education Program. STEP History and Mission. Retrieved May 21, 2013 from the Massachusetts Institute of Technology: http://education.mit.edu/classes/step-history-and-mission

APPENDIX A

How many times did you try Quest 1 (get 1 hat + 1 dice)? _______
What was the highest number of coins you were able to save? ________

How many times did you try Quest 2 (get 10 hats or apples)? _______
What was the highest number of coins you were able to save? ________

How did you decide which trades you wanted to do?

How fun did you think the game was? (Boring) 1 2 3 4 5 6 7 (Awesome)

How hard did you think the game was? (Very easy) 1 2 3 4 5 6 7 (Very hard)

Was there anything about the game that you thought was especially cool?

Compared to the last time you played, which did you like better (circle one):

Doing the quest by myself Working with someone else

Why?

How did you negotiate with the other person you were trading with?

What do you think this game was trying to teach you? Do you think it did a good job?