

**CRA - Women  
Distributed Mentor Project**

**Summer 2006**

**Eve Powell  
Dr. Tiffany Barnes**

## Table of Contents

<b>Introduction .....</b>	<b>3</b>
<b>Mission Statement .....</b>	<b>3</b>
<b>Background .....</b>	<b>4</b>
Old Story.....	5
Old Story 2.....	6
<b>Summer Game Planning.....</b>	<b>7</b>
<b>Saving Sera .....</b>	<b>9</b>
Making the Game.....	10
Storyline.....	11
The Saving Sera Team.....	11
Game Design.....	12
<b>User Reactions .....</b>	<b>14</b>
Visuals .....	14
Sounds .....	14
Quests.....	15
<b>Future Work.....</b>	<b>15</b>
<b>Conclusion .....</b>	<b>16</b>

## Introduction

My research interests in computer science involve using the current game developing technologies to promote interest in minorities (especially women) in the computer science field. Currently I am participating in undergraduate research on a project called Game2Learn at UNC Charlotte. The current goal of this project is to create a massively multiplayer role-playing game that supplements an introductory computer science class. My current role within this project is lead programmer and lead level designer. Past contributions to this project include some sound engineering, storyboarding, and some work with modeling, animation, and rapid prototyping. In a nutshell, I do just about anything and everything required in creating appealing levels for our game that encourage equal participation among men and women of all ethnicities.

Recent findings in the gaming world suggest that the extravagant amounts of time spent by high school and college age students in virtual environments or game worlds will only increase as technology becomes more sophisticated. Also, statistics show that there's almost an even amount of female and male players on some of the more popular massively interactive games. With these encouraging findings, we hope to reach out to this new group of game enthusiast and ultimately increase the amount of female's in the computer science field. From my own experience, I feel strongly about the success of this project and the possible uses for the project once we succeed in building a game that people will learn as well as play.

This project is my first experience with a big research project. Games2Learn has proved to be a very exciting project that has drawn me in deeper than any other video game. As an avid gamer, as well as a computer science student, I find the idea of a game supplemented computer class extremely appealing. As a prospective game developer, I can't imagine a better project for myself. I've learned many fundamentals of game design and development that will take me far when I pursue my career in games. Since I joined the project, it has been officially given a spot in the Future Computing Lab of UNC Charlotte. This new lab has offered many other opportunities to the Games 2 Learn Project. The Future Computing Lab (FCL) is a wonderful ambitious setting that host many similar projects dealing with using virtual realities and environment in helpful ways such as treating Anxiety Disorders and providing supplemental training to medical nurses. We have demonstrated our work to many visitors of our school and have had a good amount of positive feedback from the work that we have done.

I've had an amazing experience working with a dedicated group of motivated people specific to this project as well as REU(Research Experience for Undergraduates) and graduate students working on other various projects in the Future Computing Lab. Despite long hours (40 to 50 hours a week) and countless frustrations, this project has managed to hold my interests as a research project I would like to continue in a graduate study setting. This experience has also inspired me to ask my own questions and seek my own solutions concerning modern gaming and virtual environments.

## Mission Statement

There is a large discrepancy between male and female interest in computer science and other information technologies. Recently, incoming females majoring in computer science has dropped to proportions unseen since the 1970's (Vegso, 2005). Even more discouraging, computer science is one of the only fields of study that has lost female representation over time (Vegso, 2005). The Game2Learn research project is an attempt to alleviate this imbalance by developing software that appeals to both women and other minorities. The games that we create are also intended to supplement text based programming assignments and lectures in introductory computer science classes.

Our long-term goal is to create a massively multiplayer online role playing game that students can play for fun as well as to get homework grades in an introductory computer science course. Currently, massively multiplayer games have the greatest audience of diverse individuals. There isn't a significant difference between female and male players of massively multiplayer games. Massively multiplayer online games also attract a wide age range of game players (Yee, 2004). With the impressively diverse player pool of massively multiplayer games, it's no surprise that there are many projects dedicated to using the MMO experience to create an educational environment. Multiplayer games like *Second Life* have already toyed with the idea of programming in game.

What we hope to accomplish is to create a game in which students will be able to program in game using many different interfaces, each interface challenging the player in different ways. For example, the player may snap together a "program" by putting together the pieces of a diagram. Alternatively, the user may type in their scripts directly into a console that can then be compiled by a script interpreter.

The players would have to work together with their peers to succeed in the game, each player having a very specific in-game job. Ideally, the students would try to form well-balanced parties to successfully complete quest in the world. Similar to a dungeons and dragons type of game, where a warrior may want to join a party with a bard, mage, and a healer, the players in our game may want a debugger, a hacker, an engineer, etc. These real world types of jobs would be mixed in somehow with the more appealing mythical jobs I mentioned earlier. We would like for the in-game play to closely resemble the types of teams that the players/students would have to interact with after graduation.

## Background

In the world of computer science, the use of visualization and interactive games as a means of education has had some mixed results. Alice, an 3d introductory to object oriented programming software, has been proven effective in retaining interest in computer science and improving classes in later computer science. Even difficult topics such as recursion were mastered by students through use of Alice in the classroom (Dan, 2004).



On the other hand, many games and computer visualizations have been found to have a negative impact when used to assist in learning computer science and engineering, despite their high acceptance rate of games in the classroom (Allert, 2004).

Despite the conflicting data of past experiments, the fact remains that some kind of learning does occur in all video games. By default, games challenge people to think and seek solutions. In setting such as this, it's not hard to see why this would be an ideal learning environment, even for those that it didn't help comprehension of the course material (Allert, 2004). For Game2Learn, the initiative became not just to create a game that teaches, but also encourages students to consider and seek solutions the *correct* questions. In the past, video games to learn computer science have served to only present the material in ways that convolute and confuse the subject matter. The games that we create need to inspire a method of thinking similar to how we are engaged in a classroom lecture or when reading a textbook (i.e. encourage reflective learning in an active environment).

## Game2Learn History

When I first joined the Game2Learn team one year ago, our goals and ideas were scattered. The team knew that we wanted to create a massively multiplayer online role playing game, there would be a steganography quest, and that we wanted to create such a game in 3d Game Studio. Overall, we had very little experience with game programming and none of the team had any extensive knowledge of the game engine. Most of the ideas in my first semester on this project were abandoned due to how unfocused and unprocessed those ideas were. The following is the story idea from semester one:

## Old Story

Logan Pryce has just graduated from her average small town highschool in Kansas, barely gaining acceptance into a new yet highly competitive institute of technology in Michigan. Her first week at SPAM proves to be very strenuous. The classes are fast paced and there are tons of homework assignments to do every night. Logan seems to be doing well, despite the work load until she get a letter in the mail from the financial aid department. In a nutshell, Logan is poor and she will need another \$15,000 to stay in school for the semester. The school decides to give her 4 weeks to remedy this matter.

Logan has 4 weeks to collect \$15,000 for her tuition. Desperate for a solution, Logan opens her Spam folder in her HooRay email account and notices a cash reward for the safe return of Xernthander McMillan, world renowned digital artist that lived in a mansion just north of SPAM institute. Logan decides to take the case and sneaks into the mansion late at night to gather clues on his whereabouts...

For our final project that semester, the



unnamed main character (as it had not reached multiplayer status at the time), began the level on a search for the missing artist, Xernthander McMillan (this name was never agreed on, I'm not sure what the artists real name was). The storyline turned out to be incompatible with the game type because massively multiplayer games have multiple characters, and very few of them are Blonde haired/ blue eyes females. But the overall story was kept for a semester.

The following semester we began development on a slightly more focused project named "StormHaven". They decided to completely scrap the idea of a modern day game and take us back to a different time, a different place, essentially a different world. By opening up the story to something unreal, suddenly the idea of computer science and magic could coalesce into one game. In StormHaven, I was able to contribute much more to the game than I had in the first semester.

## Old Story 2

*A mystical plague, a diseased Prince, a heinous conspiracy, and a kingdom in chaos. Welcome to Geronia. A land, once beautiful and at peace, now ravaged by politics, war, and disease. The people are either enslaved or terrified, the Royal family destroyed, and the plague threatens the life of every man, woman and child in the realm. The plague that brings fever, madness, and inevitable death.*

*Succumbing to the plague's madness, Prince Alexander destroyed his entire family, including his betrothed. Ranger Darathorn saved the Prince from the flaming pyre and, through a process that would have killed a lesser man, cleansed the Prince of any taint of the plague. Wracked with remorse and guilt, the Prince vowed to cleanse the land of the insidious plague and destroy the people who created it. Then, and only then, can he atone for his deeds, for becoming a KinSlayer.*

*The adventure of a lifetime awaits those who answer the Prince's call for aid. He asks only for those brave of heart, stout of spirit, and loyal to their realm to join in the quest to rid Geronia of the plague and to bring justice to those that created it. Glory seekers, justice makers, or even those people that want a lot of treasure, all will be welcome under the Prince's banner. Disease, battle, glory, treasure, and an almost certain death awaits those that take up the challenge. Will you be the one to put Justice into the Prince's hands?*

*Or will you turn aside and watch your Kingdom fall? Or will you turn the situation to your advantage and try to take the Throne?*

*-Amanda Chaffin*



This time, the Game2Learn group attempted to create this game as an Unreal Tournament modification. The game looked better and was appealing to a larger audience but by the end of the second semester it was obvious that development just wasn't

going fast enough. Our sights were set too high and we were far too inexperienced to create the game we wanted using the tools that we had chosen.

By this time I had started rapid prototyping of our game in RPG Maker, a much simpler role-playing game toolkit that allowed me to quickly create the world that our game took place in so that I could start developing game-play ideas and various missions for computer science students to test out. Other students had begun looking at other games such as Neverwinter Nights to use to develop a Game2Learn game. This is where we picked up during the summer.

## Summer Game Planning

Our first two weeks working on this project was spent planning out games that we thought would teach concepts based off of earlier projects assigned by introductory programming professors. Computer science professors at this university were gracious enough to send us copies of their syllabi and project lists. From this data we were able to come up with in-game assignments very similar to the assignments given in a normal computer science class. Within these assignments we attempted to break down the assignments to the foundation concepts that the student would have to master in order to successfully complete the assignment.

**Assignment:** Write a complete C++ program to do the following:

Display (print on the monitor) your personal programmer information. This information is to be coded in comment form at the beginning of your program file AND printed on the screen when the program runs. Save your program as **programmer.cpp**

**Purpose:** This assignment is designed to demonstrate your understanding of the creation, compilation, and program submission process along with the inclusion of the required programmer information in both commented and printed form. Note: we will be using this process and including this information for the entire semester/.

**Breakdown:**

- Create a quest in which the player has to overcome syntax errors
- Create a quest in which the player has to compile the script
- After the user completes this quest they should be able to use this method to do similar projects in game.

Using this method we came up with several game ideas that we would then narrow down into the most interesting and most doable within ten weeks. Here is a list of the most popular game ideas of the summer.

- **Nedrick and the bulletin boards-** Nedrick is having trouble posting his flyers on the town bulletin boards. As a result, less people have been asking for his services as the town shrubber. We also refer to this quest as

the print statement quest. The player types in the appropriate print statements in the correct order and then has to post (compile) the scripts onto the bulletin board. This quest turned out to be one of the most confusing because we also threw in ideas about declaring variables in this quest (the strings are pre-written and given variable names of type String). We ended up using this quest, but recognize that there are some parts to this quest that should be rethought.

- **Musician's Quest-** Person will approach the king's court only to find that the king is miserable. His royal conductor has quit and left behind a stack of sheet music that is disorganized but written very similarly. The player must now put the correct sheets of music in the correct case-statements.
- **Dragon Racing-** Students will go to an arena within StormHaven especially built for this mini-game/quest. They will start out with a very basic car (dragon) and track. The car will be controlled via scripts and methods that must be preprogrammed before races. Students have no control what-so-ever during game play, so they must write scripts that can react to all possible obstacles within a track. The first track will be very basic, just a simple straight road where students must declare and change a variable that controls the speed so that they beat a computer player. Once they have finished that race, they are allowed to purchase a turning script, where they must write the methods that the car would use to turn in a certain direction, and then have the car call the method at the right time.

New tracks and scripts will be unlocked as the students continue to play, until in the end they have a car that can turn, avoid collisions, slow for sharp turns, pass other players, and keep other players from passing them. Tournaments and various other racing events can be held for special prizes, organized by teachers or game admin.

- **Key sort-** A final quest idea. This quest idea was an attempt to tackle the difficult concept of recursion. The player has overcome all of the other quest ideas and has found herself at a final locked door. The door must be unlocked with a unique key but unlike most quests, where the player must search everywhere for one lost key, the player is immediately given the correct key... as well as 4999 incorrect keys. The player is given a limited amount of time to find the correct key. To complete this quest the user must successfully piece together a quicksort algorithm.
- **Egg quest** – This game idea was designed to teach the user how to write nested for loops. Initially the user would have had to write the for loops from scratch but it was difficult to envision how the user would do this in-game without us having to let them do it free style and then us creating a script interpreter. As a result, the quest became a debug exercise. The user was given a faulty nested for loop in which it was supposed to drop



eggs into an egg carton, but instead it filled the carton as well as dropped egg outside of the carton, causing the eggs to break. The user had to figure out that the loops were counting too high and what to do to fix this problem.

- **Fishing Quest-** The player finds an old man on a dock, fishing. The man seems to be having problems remembering how many fish he's caught because he's so old and his mind is all jumbled up. When the player agrees to help him, a screen pops up that shows the player the man's mind. The player then has to reorder his mind and then compile it. If it works, the old man will give the player some money and potions. This quest is an attempt to use a script jumble to teach a programming concept. We wanted to expose the player with an opportunity to guess and check to see if the player would spend less time on the quest or take the quest less seriously.
- **In game tutor-** For quest that student's don't understand, the player has the option of going to see the town tutor. The tutor asks you which quest you're having problems with and then walks through you through the concepts that you are supposed to be learning. The tutor proved to be the most useful for the print statement quest.
- **Catacombs-** Two young children went playing down in the catacombs and haven't returned for dinner. The player runs into a neurotic mother that's worried that her children are trapped. She explains that she sent the father after them but now he is missing as well. The catacombs idea is a series of missions that the player must overcome to get through the catacombs to save the missing children.
- **Saving Sera-** The objective is to save Princess Sera from the evil sorcerer and ultimately save the world from his evil clutches. You are Arshes, a lowly peasant that has admired Sera since he was a small child. He also would like to be a hero. Perhaps with your help he can become the hero he was destined to become. Saving Sera was another attempt to compile the game ideas into a cohesive Game2Learn game.



## Saving Sera

After the Game2Learn team agreed on the ideas to work on over the summer, the team split up between two projects, Saving Sera and The Catacombs. I decided to work on the Saving Sera quest since it was decided that it would be done in RPG Maker and I had the most background with that particular

toolset. We thought that Saving Sera would be fairly simple to create with the given toolset but it was a little difficult to add the extra functionality required for the tasks that we wanted our players to complete. In the end, it took both teams all summer to develop Saving Sera and begin testing subjects. The games were each created differently and we attempted to test many different aspects of game play with each game. Saving Sera was a two dimensional game whereas the Catacombs was a three dimensional game. The missions within the Catacombs quests were very syntax oriented where Saving Sera concentrated mostly on theory. The Catacombs was a linear game where Saving Sera was exploratory. With each of these aspects we hoped to gain qualitative data on what people like and dislike in a learning game, and which teaching methods in game were most effective.

## Making the Game

Saving Sera was designed in RPG Maker, a simple toolset designed for making snap together role playing games. RPG Maker is created by Enterbrain. They've developed this toolset as a game for many different gaming consoles such as PC, Playstation, Playstation 2, GameBoy Advance, Famicom, Sega Saturn, etc. RPG Maker was originally actually created for game enthusiast with little or no programming experience. We decided to use RPG Maker for these



reasons. We wanted to develop a game to test in the shortest amount of time so that we could begin the study as soon as possible. The RPG Maker toolset also allowed us to quickly create a game that was very different from today's norm. Do realistic three dimensional graphics really matter in a game that teaches programming? Should the user interface seem childish and simple or will a beginner programmer feel more at home with a complicated but flexible interface to play with. These kinds of questions we thought could be addressed if we created the game in this environment, especially when the second team was creating a game as a modification in Neverwinter Nights, a popular three dimensional role playing game.

The games designed in RPG Maker are put together in an environment very similar to a programming environment but simple enough to piece together without having to write a single line of code. Unfortunately, for our purposes, this was not the case. Almost everything we had was scripted in the scripting editor, using the increasingly popular RubyScript.

RubyScript is the active scripting version of Ruby. While very popular in Japan, Ruby and RubyScript aren't very well represented in America. Documentation is sparse and comments in the original script are all in Japanese. It proved very difficult for my team members to learn, even to the point that it became a set back in development. For

me, I had prior experience with the language but I had an equally hard time teaching the language to my peers. Developing in this environment, though initially thought to be very simple, turned out to be surprisingly challenging, and a learning experience. In fact, a few of the teaching methods that we implemented this semester are ideas that came from attempting to teach a foreign programming language to my teammates.

## **Storyline**

The story of Saving Sera started out very in depth, as is everything I decide to do for a project. For our purposes, the story had to be severely cut down to something that could be used in a fifteen minute study, which was relatively hard to do. Actually, in some ways, that goal was never realized. While the game can be completed in under nine minutes, the average game play is somewhere between eighteen and twenty-two minutes.

The story is revealed through cut-scenes and the eyes of a poor young man. The beginning movie sequence reveals that the lovely Princess Sera was celebrating her birthday with her father (King Logan) and the royal court. All was well and good until the infamous sorcerer (Gargamel) forces his way into the castle and cast a spell, stopping time long enough kidnap the princess and flee. The King, desperate and panicked, calls upon any adventurer brave enough to go after Gargamel and retrieve his precious daughter. King Logan promises a handsome reward to anyone that can successfully bring his daughter back and destroy Gargamel before the deadline (three days).

There was a lot of controversy in using this story for a Game2Learn game. As I stated earlier, the game is intended to attract a diverse audience, but the main character is clearly a white male, and the goal is to save a princess, a white female. It stands to reason that a story of this nature would attract a white male fan base. However, it was in developing the game characters that we attempted to add diversity.

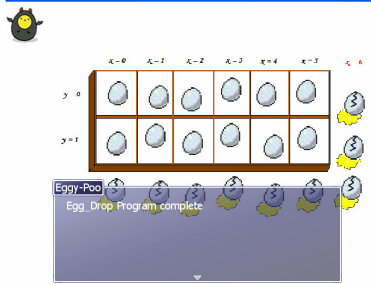
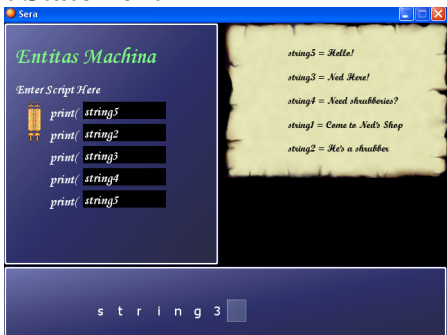
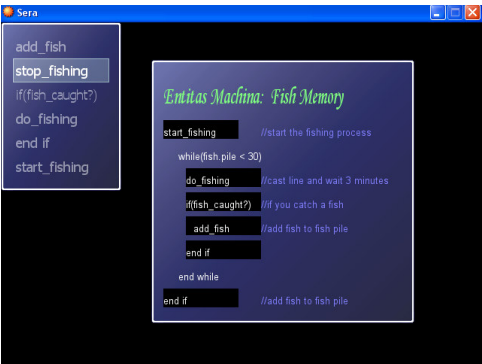
Princess Sera can also be thought of as your main tutor. Between her, the Dark Mage, and your brutish and hot-headed friend that travels with you, the main character learns to depend on these characters input to defeat Gargamel. Had the game been allowed to be longer, more time would have been dedicated to breaking this game and its characters from the mold. Like real life suggests, the player must collaborate with many people to learn and grow. The people he must collaborate with are of different backgrounds, different ethnicities, and different genders.

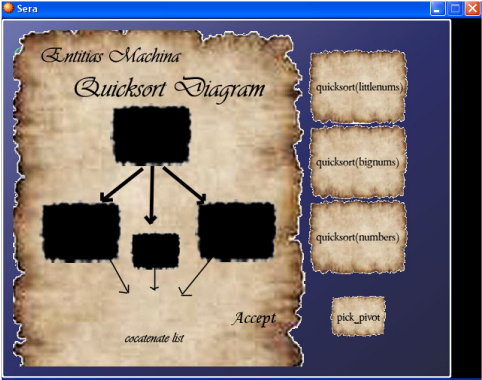
## **The Saving Sera Team**

Saving Sera was developed by three members of the Game2Learn team: myself, Hyun Jordan, and Paige Matthews. This was my first time working with some many people that could be as sympathetic the Game2Learn cause as I am. The team was composed of three women, one white, one black, and one Asian. The team was also well rounded as we had one person in charge of programming, one taking care of documentation and testing, and one responsible for data analysis. Everyone worked well together, and we also had many opportunities to work with the other team as well, which included Tiffany Ralph, Amanda Chaffin, and Alex Godwin.

## Game Design

The Saving Sera game was designed in a way to use different game scenes for the player to complete programming assignments within. Each scene put into practice a different programming technique. Afterwards, we asked the user to identify their favorite technique for completing tasks in game. We provided ways for the player to complete task using diagrams, drag and drop, freestyle (typing/ fill in the blank), code jumbles, and dialogue trees. Overall, the game seemed fairly disconnected to the user who was never able to use the same technique twice but this design was successful in testing out several different ideas at once and seeing which ones students were most comfortable with.

<p><b>For Loops</b></p> 	<ul style="list-style-type: none"> <li>• Dialogue Tree</li> <li>• Shows implementation in action</li> <li>• No penalty for incorrect answers</li> </ul>
<p><b>Print Statement</b></p> 	<ul style="list-style-type: none"> <li>• Free style</li> <li>• Most confusing quest (biggest payoff)</li> <li>• In-game tutorial available</li> <li>• in-game typing functionality</li> <li>• battle “script bugs” for incorrect answers</li> </ul>
<p><b>While Loops</b></p> 	<ul style="list-style-type: none"> <li>• custom code jumble</li> <li>• battle “script bugs” for incorrect answers</li> </ul>
<p><b>Recursion/ Quick sort</b></p>	<ul style="list-style-type: none"> <li>• drag-and-drop</li> </ul>

	<ul style="list-style-type: none"> <li>• diagram</li> <li>• dialogue tree</li> <li>• final battle, fast paced and exciting</li> <li>• health depletion/ battle “script bugs” for incorrect answers</li> </ul>
---	---

## Modifications

RPG Maker was originally very limited in the ways the player could interact with the game. Originally, the player could not type directly into the game and the game did not have mouse functionality. Those types of things as well as the custom programming scenes had to be created by me. Other modifications would have been made if time had permitted. We would have liked to create a better, more interactive battle system that would have offered another way of review introductory course material.

RPG Maker XP is different from its predecessors in that it allows the developers to customize their games using the Script Editor. While the editor was primitive and there was very limited documentation, we were able to find a few RPG Maker help sites in English, and places where people would share RPG Maker scripts and ideas. Sites like RMXF.net, The RPG Maker Resource Kit, I found help from developers that had already figured out how to use dynamic link libraries with Rubyscript and RPG Maker.

```
GetKeyState =
Win32API.new("user32", "GetAsyncKeyState", ['i'], 'i')

module_function

def keyboard(rkey)
  GetKeyState.call(rkey) & 0x01 == 1 # key 0
end
```

*Code for windows key calls- created by cybersam (RMXP)*

From here we were able to quickly add functionality to our game and custom scripts that otherwise wouldn't have been available to us.

```
def update
  #@evie_debug = Window_Help.new
  #@evie_debug.set_text($statement_counter.to_s)
```

```
@edit_window.update
#@input_window.update
if keyboard(R_Key_A)
  if (@shift_key == true)
    @character = "A"
    @shift_key = false
  else
    @character = "a"
  end
  $game_system.se_play($data_system.decision_se)
  @edit_window.add(@character)
```

*Code snippet from Saving Sera. A quick implementation of a small in game text editor, Scene\_PrintStatements.*

## User Reactions

Game2Learn is still in the middle of testing Saving Sera so the data that we have acquired hasn't been fully analyzed. Reactions to Saving Sera have been wide and varied. There is about an equal amount of people that like Saving Sera or found it helpful than there are people that disliked the game and found it unhelpful.



## Visuals

Something I noticed that Jordan pointed out among participants is that most females preferred the unrealistic nature of the Saving Sera over the realism provided in the Catacombs modification. Conversely, males preferred the Catacombs quest visually. Of course there were instances where this generalization did not hold true, but it is something that I'd like to study more in depth. Like the majority, I found the Saving Sera quest more appealing than that catacombs quest because the latter was darker and more serious. If there is any merit to this observation, how is Game2Learn to overcome this bias? Is there a healthy medium that will appeal to both men and women?

## Sounds

Few people responded to the sound in either game, however, those that did notice the sound had a lot to say about how it affected their mood and the ambiance of the game. Some of the previous readings I've done with sound in video games suggest that sound and music plays a very critical role in how immersive the game is, but is usually the least

noticed. I doubt that any useful data will be collected in that area for this study, but I'd like to investigate the role of sound/ music with in-game learning in a future study.

## **Quests**

Overall, some quests turned out to be more confusing than others. The least liked quest turned out to be the print statement quest, the quest that left the most room for error. Generally, people enjoyed the quest that they could be reasoned out in two or three tries. Because the print statement quest was freestyle, it was much more frustrating to a lot of the participants. It seemed that people had better luck with the quest after a brief in-game tutorial. I think that this quest will get a better response if the tutorial was offered immediately after the first try.

I had a mostly positive response to the for-loop quest. Participants enjoyed that they actually got to see how their response affected the programs instead of just telling them that their answer was right or wrong. In the future, I'd like to add this kind of game response to the rest of the quests and see if that has any impact in user reactions and performance.

## **Future Work**

Overall, our game was well received but there were still parts of it that the majority preferred over others. The print statement quest proved to be the most difficult and confusing quest for our participants. I'd like to look at why that is the case. Originally, this quest was created to be the easiest quest, but fewer people finished this quest than any other quest. I would like to spend a little more time on the first quest and possibly revise it. This quest also had the most challenging user interface (challenging to use and challenging to create). Perhaps too much time was spent on the interface and not enough on the quest design. As a result, the request seemed only to confuse participants.

Also, reactions to the games suggest that these learning environments affect player participation and involvement. People that didn't find a particular game appealing began not to pay as much attention to the missions they had to complete in the game. When a game failed to captivate the player, the user would miss hints and other information critical to learning and/or plot advancement. Because of these findings, I'd like to look into ways of molding game play into what the user requires most at the time.

Collaborating with my advisor and other peers/ professors on campus gave me really good ideas on how to approach this problem. Dr. Michael Youngblood, another game programming teacher at UNC- Charlotte, mentioned having game logs keep track of important interactions between the user and the game that will ultimately "teach" the game what teaching tactics work best for that particular student. For the student that skips over important readings because he or she hates reading through dialogue, perhaps the game could offer more visual learning alternatives and vice-versa for the student that hates waiting through cut-scenes. I found this idea most useful for the games that we are working on now, but I'm curious to see if the same idea could work in a multiplayer setting, which we are heading towards really soon.

We also need to look at other ways of making a game of this nature appealing to a more diverse audience. So far, each game appeals to one type of gamer or another. What can we do to make everyone happy? Can anything be done?

## Conclusion

I'd like to thank everyone that has contributed in giving me this incredible research opportunity over the summer. I will be continuing this project for the next school year and I look forward to seeing what direction this project takes. The Game2Learn project is the perfect project for any gamer enthusiast or developer. It pushes everyone involved in thinking of fun, new ways of learning. It pushes programmers to use develop new and powerful tools as well as use existing tools in new ways. I'm ecstatic to have been a part of such a fantastic team of developers. Thanks again for allowing me with wonderful opportunity.

-Eve Powell





## References

Allert, James. Learning Style and Factors Contributing to Success in an Introductory Computer Science Course. University of Minnesota Duluth. 2004.

Dann, Wanda and R. Pausch. "Using Visualization to Teach Novices Recursion." Computer Science Dept. Ithaca College Ithaca. 2001. pp 111-112  
<<http://www.sju.edu/~scooper/alice/p109-dann.pdf#search=%22SIGCE%20Alice%22>>

Hill, John M.D. Puzzles and games: addressing different learning styles in teaching operating systems concepts. ACM SIGCSE Bulletin, Proceedings of the 34<sup>th</sup> SIGCSE technical symposium on Computer science education SIGCSE '03. January 2003. pp 182- 186

RMXP.org RMXP tutorials. 2006  
<<http://www.rmxp.org/forums>>

Steinkeuhler, Constance A. Learning in Massively Multiplayer Online Games. Proceedings of the 6<sup>th</sup> international conference on Learning sciences. 2004. pp. 512-568.

Vesgo, Jay. "Interest in CS as a Major Drops Among Incoming Freshmen." Computing Research Association. May 2005. Computing Research News.  
<<http://www.cra.org/CRN/articles/may05/vesgo>>

Voth, Danna. Gaming Technology Helps Troops Learn Language. Intelligent Systems, IEEE. Volume 19, Issue 5, Sept. 2004. pp. 4-6

Yee, Nick. "The Daedalus Project: The Psychology of MMORPGs." Nick Yee. Sept. 2006. Volume 4-4.  
< <http://www.nickyee.com/daedalus/>>