

## The Missing Semicolons

Nigel Haney  
Stefon Martin  
Kevin Tran  
Brendan Lauck  
Ash Mahein

Chris Cain

Behrooz Shirazi

Computer Science 421

4/27/18

## **Alpha Prototype Description:**

Classroom gamification is an effective method of motivating some students to engage with their coursework. Our team's project aims to show it is possible to increase the number of motivated students by removing the game from the classroom itself, and decreasing the academic content in the game, while at the same time rewarding students who show effort in the classroom with progress in the game, which is played outside of class. The Gamification project was presented to our group to develop and deploy so our mentor Chris Cain could conduct a study on whether classroom gamification can improve student attendance and participation. This would be done through the students attending class and completing classwork or homework in order to earn rewards in the game that could be in the form of coins, stamina, or other perks. Before the study could take place, our mentor wanted us to get the game into playable shape for students.

In the first few weeks after being assigned this project our team of developers was given the task of playing the game for several hours throughout the entire week to get a better understanding of where the game could be improved. After conducting our walk through of the different mini-games that were included in the overall mother-game, we concluded that there needed to be some serious reworking of the RPG, conqueror, and incremental (mother-game) game. The mastermind game seemed to be the only mini-game that was functional for students to play.

The RPG had numerous bugs, but one that stood out to the group was after 10 minutes of playing the game, the game would prevent players from progressing beyond a certain room once they entered during the tutorial stage of gameplay. Some bugs that our team dealt with were game freezes, lack of a proper tutorial, and the inability to complete some of the levels of the game. The RPG game was one of the two mini-games that need a lot of patching done before deployment.

The conqueror game was also in need of thorough revision to be playable, enjoyable, and in a state to collect good data. The main bugs that our team found in the conqueror game was that the levels were very difficult for the user to play. As soon as the game would start the player would immediately be hit by enemy bullets. There was no countdown when the player hit the play button. The game would immediately begin without giving the player any warning. This was seen as something that would potentially frustrate the user. It was necessary to fix all the bugs in this game in order to collect unbiased data.

In addition to gameplay issues, the code base had readability and efficiency issues. Fixes needed to be made in the incremental game, mini-games, and the overall balancing of the games as a whole. The code for the RPG and conqueror game had to be restructured and refactored into the incremental in order for them to work together in tracking user activity. In order to tackle all the problems the game was facing, our team decided to come together and start fixing all the bugs we had found. Our team began by fixing the RPG to allow users to have a better experience while exploring the game and being able to navigate it without becoming

stuck in a room. On the other hand, our team decided it was best to remove the conqueror game and completely redo the code.

Some minor bugs we found were with the mini-game tutorials and the grammatical errors that were made. The tutorial in the main page of the game didn't do a thorough job of giving the user details on how to play the game and what each of the items on the screen meant. Our team cleaned up the grammatical errors and fixed the tutorial so it could help the user get a better understanding of the game.

After collecting information on the state of the game our team was given individual roles to handle throughout the semester in order to make the game playable. These roles were a bug fixer, backend developer, daredevil game developer, mastermind game developer, and balancer. Nigel Haney was given the job of managing all bug fixes, Brendan Lauck was given the job of developing a new mastermind game, Stefon Martin was given the job of building the new daredevil game, Kevin Tran was given the job of picking up the slack if one of the developers would fall short, and Ash Mahein was given the job of balancing the games so users wouldn't spend too much time only on one game instead of all of them. When a developer didn't have much to do they would also become a bug fixer since there was a lot of patch work that needed to be done.

### **Design Modifications Resulting from Alpha Prototype Testing:**

After developing our alpha prototype, our teams began to conduct tests that would determine if there were flaws in our design. From our tests we concluded that our game was in working shape but there were always improvements to be made all around in order to make it playable. We came across a quite a few bugs in our games that we thought could be fixed before deployment. Following is a list describing some of these bugs, and the steps we took to resolve them.

One of our developers found a bug in the balancing of the game that we thought needed to be reworked. The bug ended up being a simple fix that involved setting the balance of the mastermind game to 40% of the overall game balance. We ended up making the RPG game balance be worth 35%. Lastly, since most user tend to play shoot'em up style games we made the conqueror game be balanced at 25% of the overall game. The original bug made each the overall game balance come out to 90% instead of 100%. The reason the most played game is ranked the least is so user gravitate towards playing a game they wouldn't usually play in order to earn more passive and active coins. We thought it would be a good idea to speed up the amount of passive and active coins earned by users throughout each game as well. There needed to be some code rework from the last teams version of the game balancing. The method used to address this issue was simply reevaluating and restructuring the code to set the total balance of the game to be 100%.

Another bug our team encountered during testing was the issue of the conqueror game being too fast-paced for the user. Initially the game didn't have a timer. When the user pressed play they were immediately attacked by enemies. We decided to slow down the firing rate of the enemies and we added a timer to the beginning of the first stage. We stayed away from adding timers in each new level of the game in order to make the user feel the need to survive the game in one run and see how far they can make it. Another design change we made was some music to the conqueror game. The game had very minimal sound in it, so our team added a soundtrack throughout the entire game to add to the immersion. This bug was fixed by reimplementing the game and running a series of tests on it to see if the game experience suited our team's expectations.

Designs for the RPG have been changed constantly to accommodate new features or fixes. One feature that our team fixed in the RPG game was allowing the player to venture into various caves and rooms without getting stuck. One of our developers put in a lot of work getting the RPG game to work and being able to meet user standards of playability. One bug that was found was when the user would get stuck in the bottom corner of the screen. No matter what the user did to get out of the location they would be hitting an invisible wall that kept them from leaving that area of the screen. After cross checking the scene and RPG script it was determined that the scene wasn't being loaded in properly. For some odd reason the scene believed that there was an object there from the previous scene that was being set to invisible so the user couldn't see it, but it still existed. Once the object was removed the user was able to move around without getting stuck in the same location. Our team's main method in fixing this bug was to reevaluate the code to find the issue and patch it.

### **Summary of the State of the Project:**

Our team has been working on the Gamification alpha prototype for 3 months now and we have made many improvements to the overall game from when it was in its initial state. During the initial state of the Gamification project a significant proportion of the game was unplayable by modern programming and gaming standards. Our team of developers ran multiple tests on the game when we were first presented with this project. The tests our team ran on the initial game were in the form of game playthroughs and parsing the code to get a better understanding of how the game was implemented.

At the moment our alpha prototype has come a long way. The game is currently in a playable state, ready to be deployed to a CptS 111 class here at WSU. Unfortunately, this deployment was delayed due to the amount of bugs we had to go through and fix, but eventually we completed a working, deployable version of the game that was ready to release on April 3rd. The player data and network communication functions of the game are currently being stored on an EECS server. The core part of the game, the incremental game, is fully functional. It keeps track of user progress and stores it in a MongoDB database on the EECS servers. The incremental game is currently keeping track of all user passive and active coins, boosters, and store upgrades.

The conqueror game has been redone to support multiple levels, new gun acquisitions, and power-ups through the various levels. Our team of developers thought that it was best to leave the game of mastermind (sudoku) relatively untouched because it was in an almost playable state when we inherited the project. After fixing only a few things with the mastermind game, it was ready to be released. The RPG game is now playable past the first five minutes of gameplay (this was the state of the game when we inherited the project) and is fully tied into the other software components in our project. A few new features were added to the RPG game as well to make it more playable and accessible to the user as well.

Currently, there are two new mini-games in development. One mini-game is Kakuro, this is the mastermind game that one developer on our team has chosen to develop. This game includes the user moving objects across a game board onto marked spots on the screen. Once the player maneuvers all the obstacles and places the objects on the marked spots they move onto the next level. The game becomes more difficult as the user moves up in levels.

The second new mini-game that our team has been working on is a daredevil game. One of our developers has taken this on as their main job throughout this semester. The daredevil game makes the user feel the speed and pressure of surviving as they are falling out of the sky. The player starts by jumping off a ledge where they need to dodge enemies and other obstacles. The user can deploy a parachute to slow down for few seconds before the parachute breaks off and they begin to fall down.

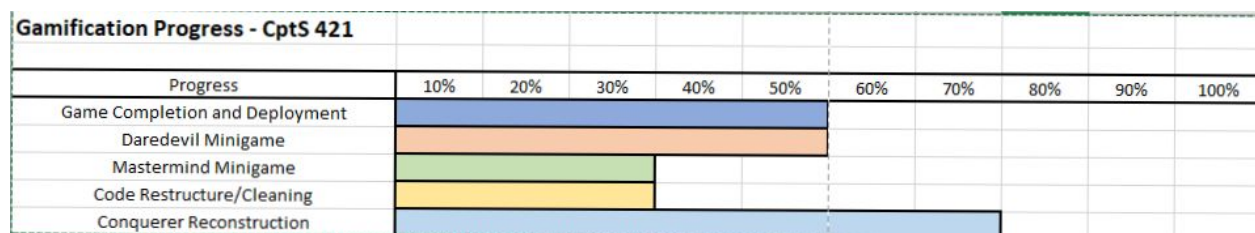
Next semester, our team has decided to continue our development of the new mastermind and daredevil game. We would like to get these new mini-games incorporated into the Incremental game before the end of the next senior design class. We would also like to keep up with all bug fixes as we improve the old or new features of the game and possibly add a user feedback system so that user's can report bugs they encounter. This will be beneficial to providing a better user experience overall because right now our testing is just limited to our team members playing through the game on our time. We will continue to maintain the database and the servers so all users are able to access the game when they choose to play. Our team of developers would also like to release a web version of the game in order to move away from having to send out new releases of the game via email each time a patch is made in the game or a new feature has been implemented. We believe that we can easily get this task done because Unity includes a feature which allows us to adapt our game to a web or mobile framework as we choose. The last task our mentor has given us to do is to implement ascension. Ascension consists of the user being able to reset their progress back to level one after reaching the max level. After a player chooses to ascend they will get access to permanent perks and new gameplay content. We think this will help retain players and keep them interested in playing throughout the entire semester.

Individual tasks for next semester will include the entire team being in charge of fixing bugs as they find them. Nigel Haney will still be charge of handling major bug fixes when the rest of the team is busy working on their pieces of the game. Brendan Lauck will be in charge of getting the

new mastermind game implemented and integrated into the incremental game, Stefon Martin will be in charge of getting the daredevil implemented and integrated into the incremental game as well, Kevin Tran will be in charge of being a utility person that picks up the slack where some members of team may fall short, and Ash Mahein will be in charge of balancing the new mini-games that will be added by next semester. Next semester we plan on getting the full game playable so we can expand our game to reach bigger audience with more varied gaming preferences. Our mentor plans to be able to take our Gamification project further and hopefully be able to present and use in his classes.

The project now has been restructured to allow students to play the game and enjoy it without running into too many issues. From our first look at the project back in January, our team believes it has improved a substantial amount. Our project aims to help our mentor study the effects of gamification in the classroom and to see if there is an increase in the number of motivated students by removing the game from the classroom. As a team we plan to aid in this research by providing our users with a game that is fun and interesting to play throughout the entire semester and analyzing their results both in-game, and in the classroom.

Progress:



Gantt Chart:

