

Team: The Missing Semicolons

Project: Gamification

Brendan Lauck

Nigel Haney

Stefon Martin

Jose Cuevas

Ash Mahein

Mentor Chris Cain

Class Computer Science 423

Date September 7, 2018

## Table of Contents

1. Executive Summary .....	3
2. Introduction and Background.....	3
3. Description of Final Project Design .....	4
4. Unity Game Frontend.....	5
5. Mini Game Design.....	6
6. Project Management.....	7
7. Results.....	9
8. Broader Impacts.....	11
9. Limitations and Recommendations for Future Work.....	11
10. Conclusions.....	12
11. Acknowledgements.....	12
12. References.....	13

## Executive Summary:

The Gamification project was presented to our team in a very delicate state in the Spring of 2018. Our mentor, Chris Cain, presented our team with a unique opportunity to take on a large scale game development project. Our team consisted of students with various backgrounds and a curiosity for learning game development. With the various skill sets our team possessed we needed to get up to par with the skills required to start working on the Gamification project. In order to make any progress on the project, our team spent a few weeks trying to learn the appropriate tools and skills to begin fixing the game and implementing new features.

Throughout spring semester our team was busy fixing errors and bugs that were left unchecked and unfixed by teams in previous years. This took most of the team's time as the project was in a very delicate state. Our team had to look deep into the Gamification project in order to find errors, faults, and bugs that needed fixing. As the end of the semester neared our team had become certain that many of the major bugs affecting gameplay had been fixed and that the game was in a playable state for our users.

With the start of the Fall 2018 semester, our team was given the task to improve upon the current state of the game, and to implement a few more features that could enhance the user experience while playing the game. Our mentor expressed we add features that make the game enjoyable for users without needing to add new features every year. Our team accomplished most of what our mentor asked for, and the remainder of this paper will outline the current state of the project, the work we accomplished, and an outline of work that needs to be done by future teams taking on this project.

## Introduction and Background:

Students often attend lecture, but many tend to doze off and some students begin to skip as the semester continues on. As students skip or doze off, we begin to see a decrease in student retention and an increase in student dropouts. Our solution to keep students interested all semester long is content-independent classroom gamification. Classroom gamification will be used to show students that it is possible to learn and engage in a lecture hall setting and having fun while doing so.

Gamification is a technique used to motivate and influence students to pay attention in class, engage with peers, and learn along the way. To implement the idea of Gamification, our team is creating a game that is completely classroom-independent, but the progress of the players in the game is directly reliant on the students attending lectures and completing assignments. Once our team has finished development of our game, we hope to see improved academic performance of students who use it.

The Gamification project was given to us Spring 2018 with its structure in place. The project has been based off of a preexisting game on Kongregate called Anti-Idle. The idea of an incremental game consisting of mini games originated from the concept of Anti-Idle. Teams in previous years had built the framework for the entire incremental game including user logins, coin and experience accumulation, and three minigames. Most of the structure for the minigames was in place as well, but the details were either missing or not well implemented. Although the game looked complete to a user, it was in an unplayable state. Users would find themselves in constant loops and unable to move in many of the minigames. The gameplay and game balance was frustrating for our developers to even test by playing let alone having users test. Each minigame needed to be restarted when our developers got stuck prolonging the process of getting acquainted with the game.

Our team's main focus for the first half of the Spring 2018 semester was to get the game to a playable state. Our mentor Chris Cain insisted the game be playable, so that he could test it by releasing it to a intro to Computer Science class at Washington State University. Our team of developers began finding and fixing bugs to the best of our ability. As the game reached its alpha prototype phase our team moved forward and began the phase of implementing new features in Fall 2018.

#### Description of Final Project Design:

The Gamification system involves three major components communicating with each other over the internet using the websocket protocol. In this section each of these components will be elaborated to show what they do and why they are useful in our project (even if our team was not the people who originally selected them).

First and foremost, the frontend game engine that our project uses is Unity. The Unity codebase and features are very elaborate, so there will be another section after this one describing all of the features of our Unity project. To connect to our backend service (the next two components being discussed), we use C# scripts to create a websocket to our backend from Unity, so that messages can be exchanged with player save data and requests. Unity allows for a fairly simple game development pipeline, making it easy to pump out rapid prototyping of new features to add. This is helpful because the Gamification project lives and dies off of new features, so the ability to quickly add them is important to keep the game fresh. The reliance on C# scripts is helpful because everyone on the team has had prior exposure to the language before starting the project from previous classes that we have taken.

The next part of the project to discuss is the actual server, which is being hosted by EECS. Our game server is written entirely in C# to stay consistent with the other code being used in our frontend game. The server sets up a connection with the websocket protocol for clients when they start the game. The reason the websocket protocol is used is because it allows for a reliable connection between the game and the server as long as the user is logged

in. The scripts on the server perform functions that create, read, update, and delete data from our backend service upon receiving messages from our game client.

The last major component of our project is our database. We use a MongoDB for the Gamification project because it is a NoSQL database that allows for the structure of our saves to easily be modified to how we see fit as new features are added into the game. This coincides with the idea that we presented earlier about rapid development of new features. All of the server side functions act as a messenger as player data is consistently being read and written from our database.

### Unity Game Frontend:

In this section the general design of the frontend game client is broken down, as it is the most complex component present in our project.

The Gamification project is broken down into an incremental game with five minigames within the incremental game. The five minigames are based on the BrainHex player typology (Nacke et al), which are meant to appeal to the following player types: the Seeker (RPG game), the Conqueror, the Daredevil, and the Mastermind (Sudoku and Sokoban). Our team believes that these player types can be used to describe a large portion of our intended user base. The incremental game can be considered a mother-game which hosts the five different mini games within it. Playing any of the five minigames helps the user progress in the incremental game. In order for users to access any of the Gamification project content they must first create an account. In order to create an account the user must enter in their new username and password alongside a verification code given to them in class at the start of the semester.

Once logged in, if the user is a first time user they are given a tutorial on how the incremental game works. This tutorial includes a basic walk through of the game features such as passive and active progression, coins, experience, how to earn perks in various games, stamina, and the countdown timer which ticks down as a user actively plays a game. The tutorial gives the users a general idea of how to play the game, so that users don't have trouble starting the game.

The incremental game is responsible for keeping track of user coin and experience accumulation whether it is in active or passive mode. As users navigate through the various mini games they can play each game in one of two mode, passive or active. If a user chooses to play a minigame in active mode they earn coins, experience, and progression points faster but a timer runs down while they spend time in the game. When the user runs out of active playing time they are unable to toggle active mode back on until they get more as a classroom reward. On the other side of things, when a user plays any mini game in passive mode they don't earn the items listed previously as quickly. Also, if the user logs out, they will continue to earn passive progress even when they are not playing. Passive and active coins are used to purchase perks that allow faster progression. Experience causes the player to level up, and after leveling they may select a potion as a reward which serves to temporarily boost their rate

of progression. Another feature of the incremental game is that the stamina points decrement each time a user begins playing any of the mini games. Once the user runs out of stamina they must wait to earn a bonus code in class to get more stamina to play the game. This limits user time on the game and make sure students aren't getting carried away with playing the game instead of doing school work.

Another key feature of the game is ascension. When the user reaches a certain level, they can choose to reset their progress back to level 1 and 0 active and passive coins in exchange for a permanent increase on their rate of progress as well as a perk point in the game of their choosing. The goal of the perk point is to give the player something that will help them in their favorite game, and be completely independent of the incremental game. This helps for a more customized and replayable gameplay experience because players may choose to increase the effectiveness of the time spent in their game of choice. As a competitive player ascends, they can keep track of their progress against fellow classmates on an in-game leaderboard. The higher your ascension, the higher on the leaderboard you will be. This leaderboard can be opted-out of if the player does not want to compete while playing the game.

#### Mini Game Design:

Each mini game has been made to challenge the users playing it. The various minigames in the incremental game have been made to appeal to various player types described by the BrainHex player typology above. Our project allow our users to enjoy a variety of games, so they don't get bored of having to play the same game everyday. All of the mini games include numerous levels and quests which increase in difficulty as the user progresses through each mini game.

The conqueror game was developed with the idea of appealing to players with skills in the shoot'em up genre of game. Our mentor had asked the team from last year to develop this game, but when our team received this project the conqueror game needed to be reworked. The gameplay for the conqueror game was not something a user would want to play, so our team reimplemented the game and fixed a numerous amount of bugs so that the game was playable upon full release. The game now allows users to fight enemies and earn perks such as better guns or power ups they can use toward their battles in various levels in the game. This game also allows users to gain progression, experience, and coins they can put towards permanent power ups within the incremental game. The higher the level that the player goes, the higher their reward will be both in game and in the incremental once they defeat the boss of that level. Once the user starts the game, they lose one stamina point and they can keep playing until they die. The perk point the player can choose for this game once they ascend increases the power of new weapons earned, as well as the speed and total health of the player.

The RPG (Role Playing Game) was designed to allow players with the need to adventure and craft to explore a world where the decisions they make can impact the sorts of quest they take and go on. The RPG allows users to take on quests to earn progression in the

outside incremental game. The in-game system of the RPG is the most complex one out of all of the games. There are stats, levels, and an in-game currency that is completely separate from the incremental game, all contained inside the game. Incremental stamina is used when a player chooses to travel to a dungeon. Players can use the in-game currency (known as “reputation”, which is earned from crafting, looting, and completing quests) to buy progress in the incremental game if they desire, or to earn progress in the RPG game itself. By spending a perk point in the RPG game, the player will receive more “reputation” when selling items in the in-game store.

The Sudoku game was designed to allow players that enjoy solving puzzles to think critically and challenge themselves. Players are allowed to choose the difficulty of puzzle they would like to solve. Puzzle difficulty is categorized by easy, medium, or hard. When a player chooses a puzzle they can solve it on their own time. If a player chooses to leave the game and come back the progress gets saved. The Sudoku game was one of the only games that didn't have very many bugs in it when the current team took over the Gamification project. Stamina is spent when starting a new game and incremental progress is rewarded upon accurate completion of a puzzle. Spending a perk point in the Sudoku game will result in a greater reward upon the completion of a puzzle. This perk point is something that we would like to rework in the future, as we want to perk points to not directly affect the incremental, just the game they are active in.

Sokoban was another game that was developed for players that enjoy the challenge of solving puzzles. Sokoban is a 38-year-old game originally created in Japan. The objective of the game is to maneuver a character to push boxes around a maze and onto different targets. Once all of the boxes are pushed onto targets, the player advances to the next level. As the player proceeds through each level, the level of difficulty increases slightly. Incremental stamina is spent upon starting a new game, and like the sokoban game, incremental progress is rewarded upon completing a puzzle. Like the Sudoku game, a perk point spent on the Sokoban game will give the player a greater incremental reward upon completing a level.

Daredevil was designed to appeal to players that enjoy a fast-paced game experience. There are 3 enemies that are meant to be dodged and they are each set to have a different flying speed. As the players score increases the difficulty and speed of the level will progress. The player will also need to keep an eye out for rare pickup items as well. The first rare item can be used to slow the level back down, the second item is a coin that can be picked up to reward the player with progress in the incremental game. If a player chooses to spend their perk point on the Daredevil game, the amount of powerups present in the game increases.

#### Project Management:

For this project, Nigel primarily worked on fixing bugs present in the games. In the first semester, a lot of work needed to be put in in order to make sure that the game was fully functional, as there were problems with every major system in place. By making the entire

project playable by the first semester, this set up well for more features to be integrated during the second semester. Initially, it was impossible to progress in the RPG minigame, so Nigel spent a lot of time making that game fully playable as it was originally intended. The Sudoku game, which existed at the time we inherited this project, did not need much help, but there were some game breaking bugs as well as various things that needed to be polished. Another major addition that Nigel made to the project was integrating the new conqueror mini game because the original one was in an unplayable state when we first got our hands on the project. After obtaining a framework for the new game, made by a previous team member, Nigel made sure that it was functioning properly and also properly integrated with the online saving system.

In the second semester, Nigel continued to fix bugs, but also took on additional responsibilities by adding new features to the incremental game and also was in charge of integrating the two new minigames made by Brendan and Stefon. The two major systems that Nigel introduced as new features into the game were a new leveling system and the leaderboard. Both of these features involved some overhauls on the server side code, so that the database could be more easily queried to provide accurate information to the client. This restructuring took up quite a bit of the development time this semester. In the end, both the ascension and the new leveling system were fully integrated. These systems were designed in order to keep players interested in the game for a longer period of time, by adding both replayability and a competitive factor for all players. In addition to these new features, Nigel worked closely with Brendan and Stefon to put their new mini games into the overarching “mother game” and successfully integrated them both. This required some refactoring of the code on the server in order to accomodate for two additional fields that needed to be saved in the database as well as reworking the UI on the frontend to actually allow the player to navigate to and away from these games in the main menu.

For this project Brendan worked on building a new puzzle-based minigame for the incremental called Sokoban. The development of this game started as a standalone project. The core functionality was developed in the first half of the senior design course. In the second half, the game was integrated into the incremental. From there, Brendan and two other subjects play-tested the game and uncovered many bugs. Each were documented and later fixed by Brendan. While play-testing the game Brendan improved the UI, music, and game art to fit Sokoban’s original Japanese themes in order to create a more immersive experience for the player. The addition of Sokoban was important to cater to puzzle-based gamers. It provided more variety for the user and to Gamification as a whole.

For this project Stefon worked on the new daredevil minigame. From the start daredevil had to be planned and confirmed with the group. After diagrams and illustrations were approved the first half of daredevil needed to be complete by the end of the semester. The first half of creating daredevil involved basic game playability and UI functionality. When the first half was complete the next half included adding features to make the game more involved with the incremental and overall more enjoyable to play. Stefon also created a GameController script that could be used as an API for the incremental to collect game stats such as highscores and



play counts. Daredevil is at a stage where it is playable, but new features will eventually need to be added including cleaner level progression, more in-game perks, sound effects and an improved UI.

For this project Ash worked on the game balancing of the incremental and mini games. Game balancing refers to the level of ease and difficulty of the game. When users plays the game and if the game is balanced too much in any one way it can negatively impact the users view of the game. If the game is too easy and users feel as if they are easily able to achieve goals, level up, and earn points, many users will quit playing because they wouldn't be challenged enough. On the other hand when a game is too hard and if users feel that they can't level up and earn points as easily they will quit playing the game as well. Ash was tasked with the job to ensure users that the incremental game would be balanced enough so users would earn an adequate amount of experience, progress, and coins depending on the mode the user wanted to play the game in. Ensuring that the user didn't earn too much experience, progress, and coin or too little was a difficult task to do, but Ash made sure that the incremental game was well balanced.

Along with balancing the incremental game, the mini games needed to also be well balanced. Each game needed to be played and tested to see if it was easy to reach new levels and unlock more perks. Users needed to be able to play each of the games and feel that they were easily able to defeat enemies, quests, and move through each game with just the right amount of difficulty. In order to fix any issues in the game balancing Ash had to test the game by playing it. Testing included spending hours playing each game and monitoring the coin, experience, progress accumulations, time spent playing games, and stamina decreases. All the items listed have to work cooperatively in order for the game to work properly. If any of the items listed above stopped working they could break the balancing of the entire game. Along with the other developers, Ash also helped fix bugs and add new features to the incremental and conqueror game.

For this project, Jose also worked on the game balancing and incremental game bug fixes. Jose was a late addition to our team, so he was used as a utility person that would help fix bugs or help other developers that were struggling to get their parts completed.

## Results:

The final prototype is fully functional. There are no apparent bugs within the game that hinder users from playing it. The incremental game including the mini games are working and logging all user action in the game. All user information is stored correctly and loaded correctly from the MongoDB database using the EECS servers. Our team feels that we have made our mentor's vision for this project much clearer. We have taken the idea of classroom gamification and made it a reality. With the solid base created by our team, the gamification project can be taken further by future teams who continue to develop new features.

Our Gamification project now includes five playable mini games, more replayability, and a smoother gameplay experience overall. Our incremental game now allows students with various gaming styles to have fun playing our games and have enough content to keep them occupied throughout the entire semester.

The project at the end of the last semester was functional and included two new mini games that our mentor wanted to integrate into the current project. The new mastermind and daredevil game were in an early development phase with only a skeleton structure in place that needed to be filled in with more functionality and game levels. These new games also needed to be well balanced and our team believed that this project was best left to be done in the upcoming semester. Our mentor also informed our team that he wanted to add a few new features to the project. These new features include an ascension system, leaderboard, and new perks that users can use in each minigame. Our team met all of the requirements that were stated at the beginning of this semester, however there were some differences in where the project stands today and the original product requirements that were stated two semesters ago.

The bonus code portion of our project was something that was originally supposed to be addressed by our group, but it was removed for reimplementation at a later time. Our team wanted to re-evaluate the bonus code system, so users couldn't try to gain access in a malicious way. The bonus code system works such that once the user runs out of stamina they must wait till they complete an assignment and attend class to get a bonus code to earn more stamina in the game. The bonus code was to be made up by a random generator and sent out via email to the students. Each code should be random for every student so they can't share bonus codes to get access. Once the students enters in the correct bonus code they can gain access to the game. Unfortunately, our team wasn't able to get this feature in because we could not come up with a simple system that could make sure the students were receiving this bonus code in class. We played around with ideas of geolocation on smartphones to send out the codes to only students in class, or codes that are only active for a very short period of time, but we decided these approaches would take too much time to implement and instead we focused on developing core features of the game to make it more fun and playable.

Another original requirement that was missed by our final product was creating a WebGL web-based version of the game. This would be used to allow the game to be more accessible to all users, without having to download a new version of the game every time a new update came out. We attempted exporting our game using the built in Unity functionality, but there was problems connecting to the server using this version, as the websocket framework for C# was not compatible with the JavaScript the code got transformed into. We decided to leave this task up to the next team who takes on this project, as it was an original goal of the project, but as a group we decided that pushing new features to make the games more fun and playable was a priority at that point in the project.

Testing was mainly conducted by our developers by looking into the code base and playing the game. Our developers started this project by doing various game play tests. This

included looking into the code by playing the game and recording the various gameplay issues. Some of the issues that were recorded were crashes and freezes in the mini games. Running these tests allowed our developers to get a better understanding of what features needed to be fixed, added, or removed to bring our users a game that is fun and complete.

User testing took place at various phases of the game design and implementation process. As our developers finished a feature or completed a round of bug fixing, we thought it was an important to have our users involved in our development process. At the end of every major development phase we had one of our developers conduct user tests. User tests included giving a few students the game and letting them play it. After allowing some users to play the game we asked conducted a verbal survey. The verbal survey consisted of question such as what users liked and disliked about the current state of the game, what new features they wanted to see, what gameplay issues they encountered in the different mini games they played. The responses received by the users were then taken into account when adding or removing gameplay features in the upcoming weeks.

The results from both user and developer tests were very helpful in our teams success and development process with this project. Our team's internal testing results guided our team to fixing bugs and cleaning up our code. Users tests that were conducted allowed our developers to get a better understanding of what issues our users wanted our team to get rid of in order to better their experience while playing the game.

#### Broader Impacts:

Our team believes there were no broader impacts on our project, other products, or environments. Every external resource we used in our project has worked well for our team and it never crashed on us. On occasion, our team ran into an issue of the EECS servers going down, but we quickly were able to restart our server and it never impacted our team's ability to develop the game, as a lot of features could be worked on offline.

#### Limitations and Recommendations for Future Work:

There are many features that still need to be implemented to realize the goal of the project. The major component of the project that still needs to be addressed is how to integrate it with the classroom in a meaningful way. Our team has come up with a rough outline of how this should be done, but it has not yet been implemented. Since this idea is basically the goal of the entire project, adding this component on should be done very carefully. A lot of testing and prototyping will be necessary to completely integrate the classroom in a unique, engaging manner.

Other features that could be considered are as follows: additional minigames, expansions upon existing minigames, a web-version of the game or a client that downloads new updates, an integrated user feedback system, real time online multiplayer interactions (chat,

friends, online games), and expansion upon incremental features (achievements, additional power-ups, etc.).

As for the general approach a new team should have on the project, we would recommend the following outline:

The first thing that needs to be done is to polish the version of the game that our group has made in order for it to be acceptable to release to classrooms. Specific attention needs to be put into the classroom integration features outlined above, as well as a system that allows players to update their game, whether it be a download client or a web-based version of the game. This could be the goal of the first semester with the goal being a deployment at the end.

Once the game is deployable, break into subgroups with specific assignments. Some people should be working on testing and bug-fixing, and others should work on implementing new features. When there are not many bugs to be fixed, more members should be focused on the implementation of the new features and when a new feature is rolled out, more people should be focused on testing and bug fixing. This cyclic approach is useful for the nature of the Gamification project because it will allow new high quality features to be pumped out efficiently to the players. This feature cycle will be especially important when the game is live, because new content based on player feedback will keep people coming back to the game.

Obviously it is very hard to give an outline of future work without knowing the strengths and weaknesses of the next team, but if our team were to continue with the project, the steps outlined above are how we would expand it.

#### Conclusion:

The Gamification project has been a project filled with numerous learning experiences for our entire team. Our team has learned game development, database maintenance, and testing skills. We have been able to take a partially finished project and turn it into a playable game where users can enjoy playing it and still be motivated to attend lecture and actively participate. The project is now functional and playable by our users. Our team has taken the Gamification project and achieved our goal of adding new a daredevil and mastermind game, fixing gameplay issues, and balancing problems throughout the entire game.

#### Acknowledgement:

Our entire team would like to thank our mentor Chris Cain for supporting our team throughout the entire design and development process of the the Gamification project. Without the support and guidance of our mentor many of our achievements wouldn't have been possible. We would like to also thank Dr. Shirazi for allowing us the opportunity to schedule weekly meeting to discuss and brainstorm ideas during the duration of the semester.

## References:

Nacke, Lennart E., et al. "BrainHex: A Neurobiological Gamer Typology Survey." *Entertainment Computing*, vol. 5, no. 1, 27 July 2013, pp. 55–62., doi:10.1016/j.entcom.2013.06.002.