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

Beta Prototype Description Updates

One of the biggest successes of our team since the start of the semester is regarding completing and integrated two new minigames into our project. Brendan and Stefon, who were tasked with making these two games, have finished developing the Sokoban and the Daredevil games respectively and both games are now fully functional. We have begun integrating the two new minigames into the incremental game and now they are nearing the point of being fully functional within our entire system. Brendan and Stefon will now be working on polishing and balancing their games so that they are satisfying and interesting for the user to play in the context of our "mother-game".

After these games were integrated, some server-side functionality needed to be given to allow the user to save their progress in these games. The server for our project includes many scripts designed to manipulate the database, written by the previous team who worked on this project. In order to make sure that the new games could be functional in the database, these scripts needed to be updated in order to accommodate the new additions. At the moment, saving in the new games is functional, but we still need to work as a team to determine what save data should be stored to create an impactful experience for the user when playing these two new games.

Our team opted out of making a web version of the game for the time being, so we may focus on making the game more appealing to play and well balanced before the release. A few features that we have come up with to implement to make the game more appealing are as follows:

Our team has been instructed by our mentor to develop a leaderboard, so students can see which students in their class are ranked in the entire game. This leaderboard will be displayed in the main page of the incremental game and will include a tab to allow students to opt-in or opt-out of seeing the leaderboard. Our team wants to make viewing the leaderboard optional, so students won't be discouraged by other students that advancing further into the game than they are.

Our mentor has also requested that we begin implementing an ascension feature in our game. Ascension will be used to allow player to reset their experience in the incremental game in order to earn a permanent perk within one of the minigames. Once a user reaches ascension, it will become harder and harder to reach the next ascension as they continue playing the game. With each ascension comes a better perk, incentivizing the player to keep coming back and playing the game throughout their semester.

These two features are things that our team has decided will make the game more exciting and replayable for users, as well as reasonable to implement before the end of the semester.

Beta Prototype Test Results

After the development of the two new minigames our team began to run numerous tests to find errors and vulnerabilities within the game. For the first few weeks before integrating Sokoban and the Daredevil game we began unit testing our code for the two new games. The developers for the two new minigames ran extensive tests by isolating their individual units to see how their games were affected.

Currently daredevil is in a functional but delicate state. There have been a couple of gameplay and unit test conducted, but solely for the purpose of seeing how the game runs and for finding major bugs. At the end of the unit test our team came to the conclusion that we need to refactor Daredevil some more. First, the runtime we will need to be tested. If all game objects are being instantiated at the current game times, this test will make sure that we are not getting a mix up of the game states like pausing the game and going back into play mode. This could cause a variety of issues not only for scoring, but for level progression. Previously the game loop reset after the user unpaused the game. Second, there was a unit testing done on the player controls and animations. Since there is an animator controller that has a whole other system that overlays the game play, this system uses some deep testing. Animations and key inputs need to have the correct transitions and play times, if they are off by any amount the game animations will run slow and feel very unrealistic. Lastly, we had to come up with the idea to run a score system to contribute to the players score in the incremental game. We tested the scores that were being calculated to see if they saved correctly. These test were helped uncover multiple bugs within the game that our team will have to fix.

For the Sokoban minigame, the unit testing began while the minigame was in a standalone state. It was ensured that each individual function controlling the player and events worked in all cases. After the unit testing had been done, integration testing was done. The project was quite small, so this testing phase was quick. The most significant part of Sokoban testing was functional testing. This revealed the difficulty of levels and the overall flow of the minigame. Many bugs were uncovered in this testing phase. Such as an error when the player started on a goal square, an error when a player reached a wall, level generation errors, etc. Again, functional testing was vital to the development of this minigame. Edge and corner cases were also explored and all bugs were fixed in the minigame standalone state before integrating it into the main game.

After unit tests were completed our developers began to run integration tests to see how they games would interact in the incremental game. This allowed us to demonstrate the cohesion of the two new minigames with the incremental. The first step to make sure the games were integrated into the incremental successfully. This included adding new buttons to the main menu to access the games as well as making them fit inside the pre-existing UI that we had in place that shows the player's incremental progress. To do this some minor graphical tweaks were needed in both games. After completing the frontend work to make sure the games were actually accessible while playing, some work had to be done on the backend to make sure that we will be able access our server and be fully integrated. To do this, code needed to be altered both in our Unity project and on our backend server. During every step of this process, extensive integration testing took place to make sure that nothing was overlooked when adding these two components to our overarching system.

Throughout this entire test phase our developers focused on trying to find as many bugs as we could. We spent approximately two weeks on this process. The first week of testing included running system tests on the entire game. Developers took the time to play the individual minigames and write down any vulnerabilities or bugs they encountered along the way. Our team compiled a list of bugs we found in the minigames and also the incremental game. We decided on prioritizing each of the bugs from high to low in order to know which bugs or issues need to be fixed first in order to move onto the next process in the game's development. Our team spent the following week fixing the high priority issues and bugs within the game before moving onto low priority one. Some high priority bugs included player not being able to access minigames or exit from minigames, player mechanics in the daredevil and conqueror game, and fixing issues with graphics in the two new minigames. Since the game was still in a very early state after the integration of the two new minigames, our team had to reevaluate our timetable for player testing.

After unit, integration, and systems tests within the team were out of the way we concluded that the game was still in a very early state and the feedback we would receive from students wouldn't be very positive since the game was still lacking features to give the user a reason to continue coming back and playing our game. Our team decided to go ahead and implement some proper checks and balances within the game before we allow students to test our game. These checks and balances included overall game play balancing, leaderboard and ascension integration, and meaningful database saves for player data regarding the new features and games.

Beta Prototype Validation Results

During the entire software development life cycle of our project, every member of our team has been validating that the game continues to work as they intend it and that the new features they add work in the context of the entire system after every major change. Since we have not released the game to a test group as of the time of writing this paper the only form of validating the results of our work is from inner testing within our software development team. Validation results will be brought up at our weekly team meetings as we all continue extensively playtesting our game. As we expand upon our game and make it more feature-rich we hope to release the game to a group of testers and get validation results in the form of feedback them.

Summary of Work Remaining This Semester

For the rest of the semester the team will split into 3 sub-teams to be focus on getting the incremental game balanced, fix bugs, polish existing parts of the game, add the ascension scheme, and a leaderboard. The sub-teams will include a balancing team, bug fixing/polishing team, and a new feature team. Some members of our team will be on different sub-teams depending on what stage of development we currently are at, but by dividing up these tasks we will prevent merge conflicts as more and more code is added to each feature and the game as a whole.

The biggest task to complete still is the balancing the game so that it is fun to play and rewarding for the user. This will require the expertise of all members of our team to complete. This will require having to go through all five games and consider the different ways we can make the games harder or easier for students play. This is a crucial step in any game development process since it can make or break the entire game. If the balancing of the game leans too far to one side students might shy away from playing our game. If students begin game play and come to the realization that it is too hard to play the minigames, earn coins, experience, or power ups they might quit playing the game. If some students feel that they game is too easy they might lose interest as well. The game has to appeal to all students in the class whether their skill level with playing games is high, moderate, or low. The developers in charge of the balancing process will not only need to balance each game individually, but also the incremental game as a whole and how all the components of our system interact.

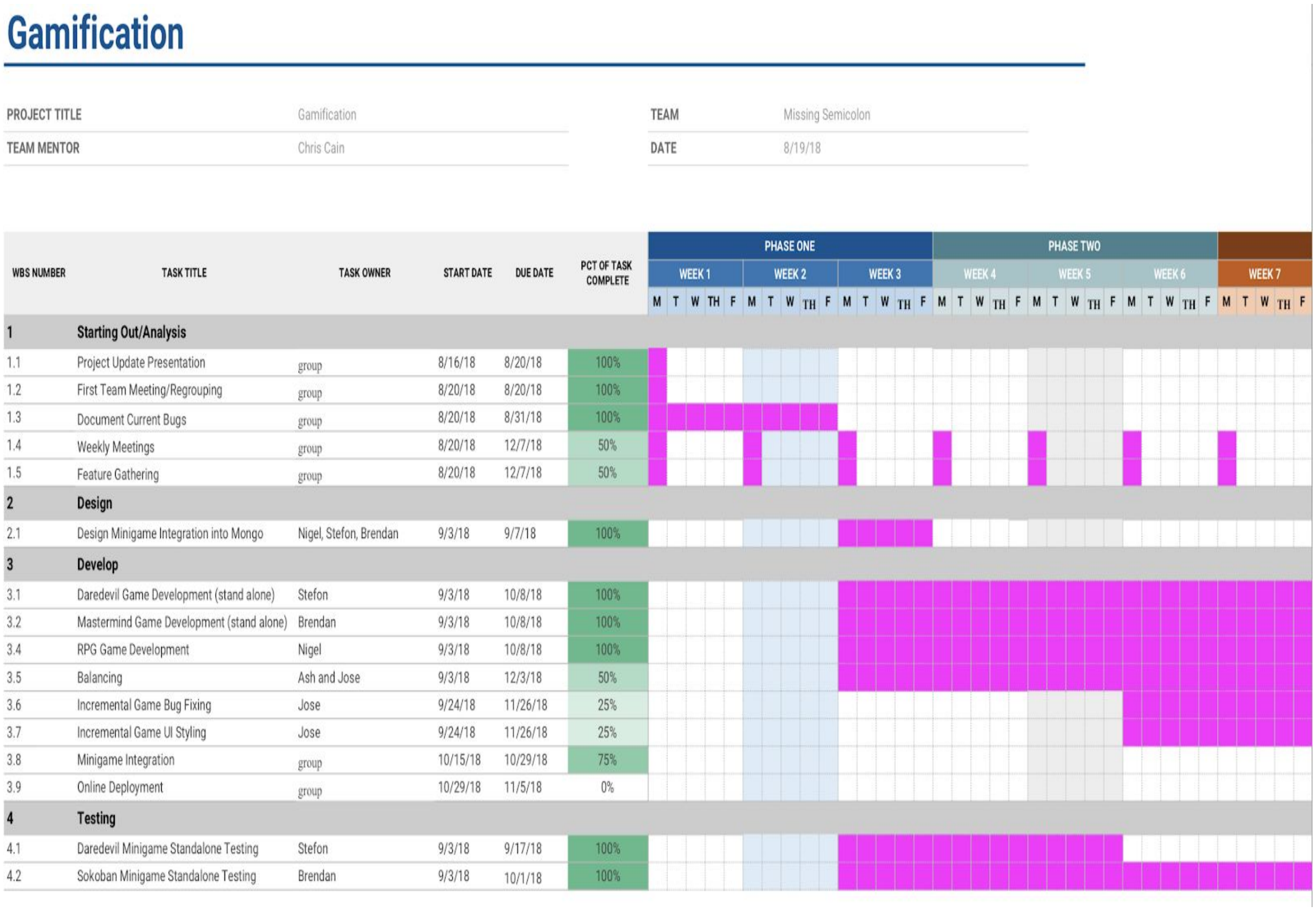
Two developers on the team will take the task of balancing upon themselves to enhance user experience when playing the game. This will require looking into various features of the game such as stamina expenditure, coin and experience accumulation, and time spent playing games. We want users to have to work hard in the classroom and not focus too much on the game because that will negatively affect their education and take away from our goal for this project. We want to make users only have a certain amount of stamina that they can spend when playing a game. After the user have expended all their stamina points they must wait to earn a bonus code from attending class to play again, which in turn will reset their stamina. The last issue that needs to be addressed by the balancing team is the coin and experience accumulation of when playing the various minigames. The balancing team will need to come up with plans regarding how to earn coins and experience in the two new minigames. This balancing portion needs to be completed before the team can carry on with the full release at the end of the semester.

The balancing team plans to add an extra five stamina points in for the time being, so users are allowed to spend more time playing the two new minigames. Depending on the minigame the user is playing we will take away a certain amount of stamina points, so not all users gravitate towards one game. The balancing team will be in charge of setting up a coin and experience accumulation scheme in the two new minigames, so users don’t find them too hard or too dull. The final thing that will need to be done by the balancing team is setting up how the two new minigames tie into the original incremental game in regard to coins and experience earned, stamina lost during playing a game, and other perks in the upgrade shop.

The bug fixing and polishing team will plan to find and fix as many bugs as possible before the full release of the game. In order to do so, the team will need to continue running unit, integration, and system tests on the entire game. This is a hard task to fully complete since the code base is very large and some of the code implemented has been developed by teams in previous years. When bugs are not apparent, members of this team will add new features that complete the user experience when playing the games such as updated graphics, sound effects, and music.

The new feature team will be in charge of adding the ascension and leaderboard features to the incremental game. Since this is a completely new feature that needs to be implemented many developers will partake in the development. One of our developers has already begun implementing a general framework for ascension and he will continue to work on that. Once the balancing team finishes the game balancing they will help finish the ascension feature and the leaderboard.

Updates to the Gantt Chart for completion of the project



Gamification

PROJECT TITLE	Gamification	TEAM	Missing Semicolon
TEAM MENTOR	Chris Cain	DATE	8/19/18

WBS NUMBER	TASK TITLE	TASK OWNER	START DATE	DUE DATE	PCT OF TASK COMPLETE	PHASE ONE															PHASE TWO																			
						WEEK 1					WEEK 2					WEEK 3					WEEK 4					WEEK 5					WEEK 6					WEEK 7				
						M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F	M	T	W	TH	F
1	Starting Out/Analysis																																							
1.1	Project Update Presentation	group	8/16/18	8/20/18	100%	🟪																																		
1.2	First Team Meeting/Regrouping	group	8/20/18	8/20/18	100%	🟪																																		
1.3	Document Current Bugs	group	8/20/18	8/31/18	100%	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪																									
1.4	Weekly Meetings	group	8/20/18	12/7/18	50%	🟪					🟪					🟪					🟪					🟪					🟪					🟪				
1.5	Feature Gathering	group	8/20/18	12/7/18	50%	🟪					🟪					🟪					🟪					🟪					🟪					🟪				
2	Design																																							
2.1	Design Minigame Integration into Mongo	Nigel, Stefon, Brendan	9/3/18	9/7/18	100%											🟪	🟪	🟪	🟪	🟪																				
3	Develop																																							
3.1	Daredevil Game Development (stand alone)	Stefon	9/3/18	10/8/18	100%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.2	Mastermind Game Development (stand alone)	Brendan	9/3/18	10/8/18	100%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.4	RPG Game Development	Nigel	9/3/18	10/8/18	100%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.5	Balancing	Ash and Jose	9/3/18	12/3/18	50%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.6	Incremental Game Bug Fixing	Jose	9/24/18	11/26/18	25%																										🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.7	Incremental Game UI Styling	Jose	9/24/18	11/26/18	25%																										🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪
3.8	Minigame Integration	group	10/15/18	10/29/18	75%																																			
3.9	Online Deployment	group	10/29/18	11/5/18	0%																																			
4	Testing																																							
4.1	Daredevil Minigame Standalone Testing	Stefon	9/3/18	9/17/18	100%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪										
4.2	Sokoban Minigame Standalone Testing	Brendan	9/3/18	10/1/18	100%											🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪	🟪