**PROJECT II**

**Minesweeper**

**EECS 448**

Adam Khan

Matthew Eagle

Mitch Simmons

Natalia Leyba

Zach Harris

**List of Meetings**

Date: September 17th, 2018

Location: Eaton classroom

Topics:

* Receiving new project
* Read and Understand Code
* Discuss new features
* Discuss cheat mode
* Bug Fixes

Notes:

* 2 instances of boards, one for cheat mode, one for regular.
* Custom addition: Filling in board when only mines are remaining
* Next Meeting: September 23rd

Date: September 19th, 2018

Location: Eaton classroom

Topics:

* Code access
* New features to add
* Work distribution
* Make sure everyone can open and run code

Notes:

* 2 instances of boards, one for cheat mode, one for regular.
* Custom addition: Filling in board when only mines are remaining
* Next Meeting: In lab on wednesday September 26th

Date: September 23rd, 2018

Location: Linux Lab

Topics:

* Read Code (A LOT)
* Understanding functionality and getting the code to run
* Making connections between code files

Notes:

* Index to board.js to gameManager.js to percentBar.js
* Made new board for cheat mode, added cheat mode button and function
* Next Meeting: In class on Monday September 24th

Date: September 24th, 2018

Location: Eaton classroom

Topics:

* Code access
* New features to add
* Work distribution

Notes:

* Distribute work, centered around cheat mode functionality
* Custom addition: Filling in board when only mines are remaining
* Next Meeting: In lab on wednesday September 26th

Date: September 26th, 2018

Location: Linux Lab and Eaton Classroom

Topics:

* Scrum Meeting
* Code marathon (For 2 hours)
* Finishing up adding in cheat mode feature
* Fix Bugs associated with cheat mode

Notes:

* Bugs: Percent bar, flagging in cheat mode, removing flags to get to cheat mode
* Next Meeting: In lab on Wednesday October 3rd

Date: Wednesday October 3rd

Location: Linux Lab, Eaton classroom

Topics:

* Finish up bug fixes
* Make sure comments are up to par
* Make sure Documentation is up to par
* Make sure everything works well

**Division of the Project**

* **Adam:** 
  + Added extra feature: Clicking a number square with bombs marked to clear other squares
* **Zach:** 
  + Preserving the flagged spaces and displaying them after Un-cheat: Added the flag preservation functionality to allow user to continue game exactly as it was left before Cheat was clicked
* **Matthew:** 
  + Back-end CheatMode and GoBack functions: Function to reveal all spaces and function to go back to current game.
* **Mitch:** 
  + Preserving the percentage bar of completed game and displaying it after Un-cheat: Added the percentage bar preservation functionality to return to the percentage of completed game as before Cheat was clicked
* **Natalia:** 
  + Front-end modification: Dealt with CSS and HTML files as well as connection between user interaction and Back-end functions.

**Challenges**

* Learning a new Language: This challenge was overcomed by reading on the selected language and starting to code in it.
* Running code: When first test of code without modifications failed we had to figure out how code was developed in order to find the issue. After thoroughly reading files of program we realized the previous team used Node js on development. We had to install Node.js, and interact with it through terminal to be able to run program through localhost:3000 on browser.
* Preserving board after cheat is pressed: Revealing spaces was fairly straightforward but preserving current game proved to be a challenge as we had to store board in a separate board and then re-display it. This worked for revealed spaces but not to preserve flags and percentage of game completed. We had to these distribute tasks between team members and issues were resolved by having helper functions that were in charge of preserving and restoring these other features.

**Possible Features to be Added**

* Displaying cheat board on pop-up window: This would allow the user to still see cheat board while playing.

**Retrospective Analysis of the Project Development**

After being able to run and understand the program developed by previous team we distributed and starting developing additions to the program. After finishing development and testing we analysed our project development and noted several improvements that could have make the development process smoother and make the program better. One of the things we could have done differently is starting the analysis and testing of previous code earlier as we realized later on that this was the stage that took the most time of all development due to our lack of knowledge of previous development. Another thing we could have done differently to improve the user experience and make easier the board preservation process was to copy the current board pass to a function that would display it in a pop-up window with the spaces revealed. This would make the preservation process easier as we wouldn’t have to preserve current game because the game would still be going in main page. Overall our program and our project development was successful as we were able to implement the required functionalities using object-oriented programming by dividing fairly equally the different tasks and by communicating any issues or barriers that appeared throughout the development.