8/31/18

All team members were present (Devin, Connor, Kyle, Ian and Daniel) the meeting took place in class.

Plan

- In our first meeting we discuss when our next team meeting will take place.
- The next meeting will take place later on today, so we can discuss our plan for the design, expectations for our work ethic, how we will go about dividing up the work , and coming up with a team name.

8/31/18

All members present at the second meeting (Kyle, Devin, Connor, Ian, & Daniel) at the fishbowl.

Plan

- We discuss how to break up the work and the agile scrum method.
- Possibly will use the 200 ok chrome app for launching the website.
- Our Team name will be RepoMen.
- We will meet Tuesday, and our goal for next meeting is to have a basic user interface.

SCRUM

Detail requirements:

- Make minesweeper
- Must be a recursive search algorithm

Design & Analysis:

- Bombs should be in a two D array too, have a two D array for the open space and number
- Have a second array that keeps track of where you clicked
- Front end we need to draw all the numbers, bombs, and have the button cover the array
- We need a panel for board size needs to be able to input size, set a limit.

9/7/18

All members present (Kyle, Devin, Connor, Ian, & Daniel) at the fishbowl.

Plan

- Meeting for Sunday.
- Devin and Danny plan on implementing the logic and flags.
- Everyone else will be finishing there branches and merging.
- Possibly changing the design to be more luxurious.

Who did what?

Kyle did formal classes in javaScript, expanding the grid.

Ian doing the input forms and setting.

Connor implemented the design of the UI.

9/9/18

All members present at the second meeting (Kyle, Devin, Connor, Ian, & Daniel) at the fishbowl.

Plan

- Merge branch setting into master.
- Get done with Logic branch possibly make the logic flag be a control click.
- Java documentation and implementing a more fleshed out design.
- We plan to meet on Tuesday

9/11/18

All members present at the second meeting (Kyle, Devin, Connor, Ian, & Daniel) at the library.

Plan

- All we have left is to work on the logic for the flags.
- We need to write the paper on Thursday get a prototype working.
- We plan to have our next meeting on Thursday 5pm.

9/15/18

All members present at the last meeting (Kyle, Devin, Connor, Ian, & Daniel) at LEEP 2.

Plan

- Finish the design for MineSweeper will also run test on the website to make sure it will not break.

Description on how work was split between teammates

When first starting the project, it was decided the group would all start working on a version of the front end html and see what was the best route. Once everyone had a very simple prototype, we took the best pieces from each version and made a strong base for each group to build off of. Connor worked on creating the visuals of the game through developing a function to create dynamically sized tables. Ian worked on the menu interface and css, installing bootstrap to assist in the process. Kyle outlined all of the classes throughout the project and defined some of their methods. Devin and Daniel worked on the majority of the logic inside the mineField class.

Once the core of the project was complete, there was less structure and the remaining work was done whenever a member of the group was free. Overall, the project was successfully divvied up and completed in a timely manner.

Challenges and how they were overcome or dealt with

While doing this project there were only a couple of problems that we faced. The first one was deciding how we wanted to approach the project and laying out the goals that we wanted to reach by the time we were done. We ended up doing a web based version of minesweeper with javascript handling the actual logic of the project. The first coding challenge was getting a clickable board with editable size to appear. This was done by having a bunch of buttons that were appended the number of buttons you want for the width and then you append a new row that can add new buttons to it.

The next problem was creating a seperate grid to house all of the logic and methods that we would need to interact with the board. This was broken down into sizable chunks, the first was creating the grid and randomly placing mines onto the board, the second was getting the table to expand when a tile with no bombs is adjacent, and third was placing flags and checking if the mines are covered by them. The grid was created in the constructor by having an 2D array of a tile class that keeps track of if it is a mine, how many mines are adjacent, and if there is a flag. The mines were then generated randomly using the math random function to choose the coordinates for a mine. All of the surrounding tiles to the mine had 1 added to their adjacent mines number as well. The expanding was built off of how the clicking worked, which was by seeing the number of adjacent mines and displaying the correct image on the tile for the number of mines adjacent or if it was a mine. The expanding built off of this by calling a recursive method if there were no adjacent mines. This method first checks if it had not already changed the tile that was passed in so it doesn't repeat, then if there are no adjacent mines to the current tile, it calls the method again with all of the adjacent tiles. If there are adjacent mines though it just calls the click function at that location to reveal the number. The final challenge was getting the flags working. This was done by adding an on right click event to the button that calls the minefield variable to add a flag to the clicked tile. This method also checks to see if the number of flags is equal to zero then it loops through the entire grid checking to see if each mine has a flag, if it does then the user wins.

Any features that did not make the demo version

There were many features we almost implemented but didn't, including

- Mobile compatibility,
- Fitting the grid size better to the window size,
- Explosions after a mine is clicked,

- Deploying the game online, hosting either on the EECS servers, Heroku, or Github,
- And guaranteeing that the first square you click is not a mine.

Features that we came up with but weren't close to implementing were

- A timer for score,
- Score based on number of clicks,
- Coloring the numbers on the board,
- Gamifying the input forms for the board size and mine number,
- Putting the win or lose message under or to the side of the board instead of as a pop-up to click out of,
- Forming mines on the board in such a way that it never forces a click to be a guess and all clicks to win can be deduced,
- Creating a leaderboard for highscores,
- A variation on Minesweeper where the grid is composed of triangles or hexagons instead of squares,
- And sound effects for certain actions performed by the user.

The bulk of these features aren't in the demo version of our Minesweeper project because we decided that they were more work than necessary, so our developmental process isn't an issue.

Retrospective on what the team would have done different

Our project strategy was effective, but it can be improved to make the next project better. Some implementations at the beginning had some small consequences down the line and, in the future, these could become bigger problems. To prevent this, the entire team should be involved in the first commit after we decide what we will work on in the later stages of the project. There should have been better planning from the start even if we have to delay writing code so everyone is on the same page and we have a nearly complete idea of what we want to do.

Another thing that we probably would have done differently is we should have worked together as a group more instead of just meeting and dividing up the work. When we worked together we were all able to communicate what we knew and were able to decide how things should be handled together. Along with that when we weren't working together it wasn't always clear who was doing what so better communication and keeping track would also be helpful.

Finally, we probably should have completed all of the necessary requirements a bit earlier so there is less of a rush at the end and we have more time to make existing code work better. This would be helpful because we didn’t meet all of the requirements until the day before and that didn’t really give us enough time to go back through the code to tinker and make things better.