EECS 448

Project 1 – Minesweeper

* Log of all meetings
  + Date, Location (no specific addresses needed. Something like "coffee shop", "Frank's house", "Google Hangout" would suffice), and list of attending members
  + Brief description of meeting outcomes
* Description on how work was split between teammates
* Challenges and how they were overcome or dealt with
* Any features that did not make the demo version
* Retrospective on what the team would have done different

**Meeting 1:**

Friday Aug. 31st during 448 Lecture for about 10 minutes

In Spahr classroom

All 5 team members were there

We picked JavaScript as our language for the project and planned the next meeting for Wednesday morning

**Meeting 2:**

Wednesday Sept. 5th during our 448 Lab for about 2 hours

In a cubicle outside Spahr Library

All 5 team members were there

We played minesweeper to understand the game

outlined the project

decided on a grid rather than honeycomb design

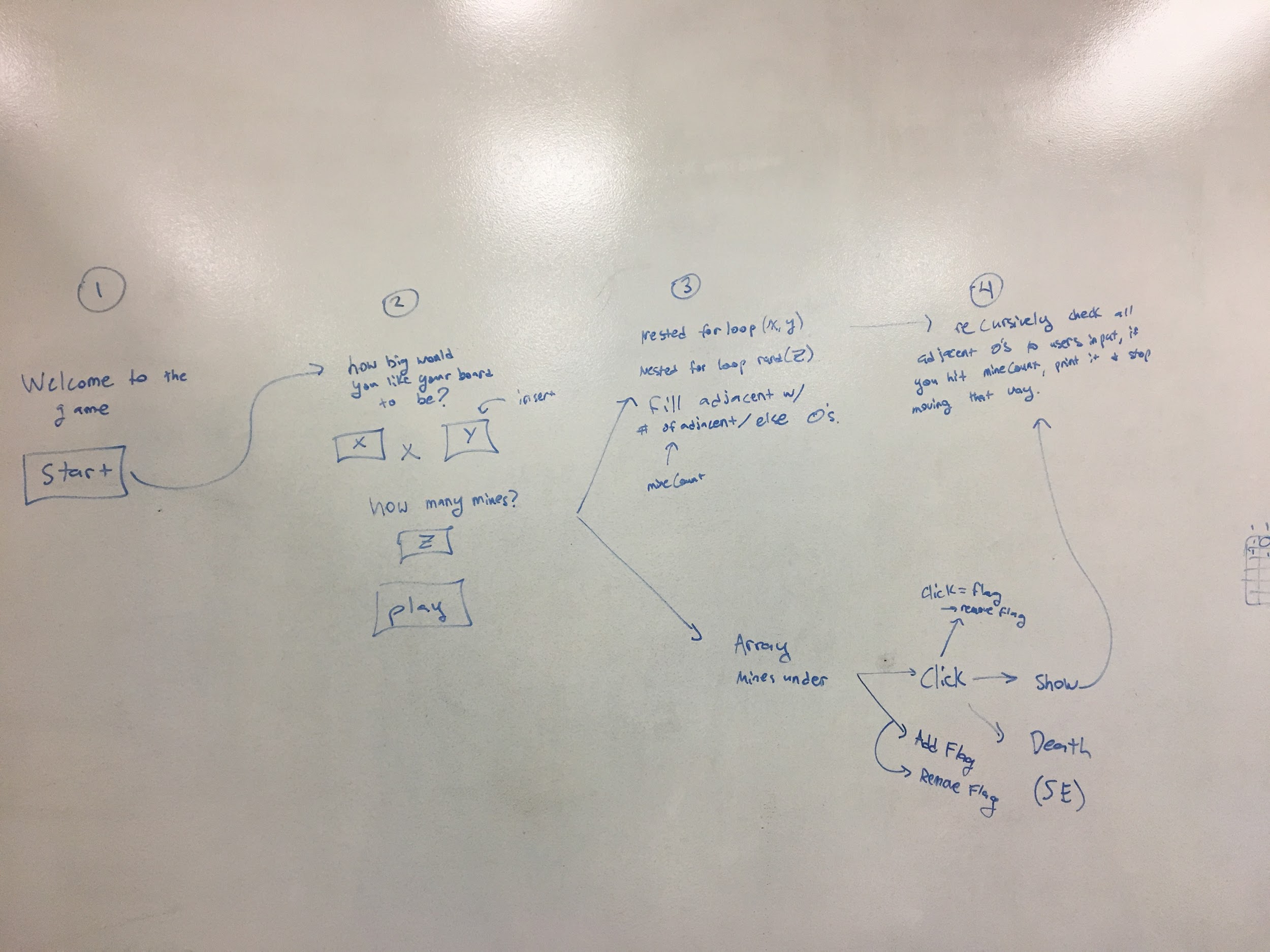
planned out the code for the project (see images following this section)

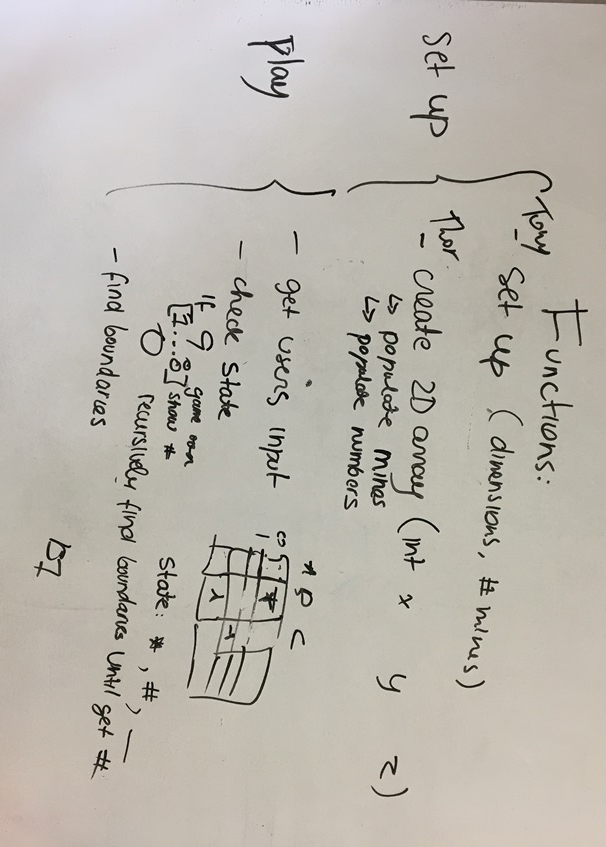
decided on a space theme for the minesweeper game

Tony made a GitHub Repo and shared it

divided the responsibilities among the 5 members (in other image following this section)

Planned next meeting for Friday after Lecture and meetings on Saturday





**Meeting 3:**

Friday Sep. 7th during 448 Lecture for about 10 minutes

In Spahr classroom

Thor, Tony, Emilia, and Rob were there. Ian was not.

We came together to discuss our progress in anticipation for our longer meeting following this.

**Meeting 4:**

Friday Sep. 7th following 448 Lecture for 2 hours

In cubicle outside Spahr Library

All 5 team members were there

We summarized our current project

Thor described his mine-laying method and we bug fixed

worked to merge our current progress

Emilia got the basic user interface up and going

**Meeting 5:**

Saturday Sep. 8th, 4:00 - 8:00

in cubicle outside Spahr Library

All 5 team members were there

We started on JSDoc to keep documentation on our code

integrated all of the game logic to work together

got the numbers surrounding the mines operational

**Meeting 6:**

Monday Sep. 10, in class for ~15 minutes

In lecture

All 5 team members were there

We got up to speed with Tony’s recursive revealing method and Rob’s flags

met with the product owner and got feedback on our prototype

planned to meet again as a group wednesday

**Meeting 7:**

Monday Sep. 10, 4:00-5:00

in Leep2

Thor and Ian were there

We cooperated on the win and lose end cases

**Meeting 8:**

Wednesday Sep 12, 9:00-11:00

in Lab

All 5 of us were there

We thoroughly played the game to come up with bugs to fix

split the known bugs amongst the group

thought up questions for the product owner

**Meeting 9:**

Saturday Sep 15, 3-6

in Leep2

All 5 of us were there

We fixed a few persistent bugs

commented the program to work with JSDoc

collaborated on the retrospective write up

**How work was split up**

Thor-

* Mine creation and population in the user specified array
* End game function to refresh the game if a player loses
* Fix bugs

Tony-

* Recursive function to reveal spaces
* Inputs for grid in HTML
* Fix bugs

Ian-

* Created the playing field including number of mines and locations
* Wrote winning game function to see if a player won the game
* Fix bugs

Rob-

* Logged all meetings
* Set up flag/ flag capabilities
* Designed numbers to fill array to display adjacent mines
* Fix bugs

Emilia-

* Created the interface in JS working the the P5 library
* Set up the grid
* Created mouseclick function based on user input to reveal spaces
* Fix bugs

**Challenges**

1. Getting the user interface to work, how we are going to make it
   1. Made a simple interface using the P5 library, once we understood the library, the interface was easy to create.
2. Game functionality, generating the number of adjacent mines to a space
   1. After accounting for the edges of the board, we created a function that looped through all the spaces adjacent to a specified box, counting the number of mines around it. Figuring out the edges and how to work with them was the bulk of this issue.
3. Validating user input
   1. We just tested the game over and over and brainstormed possible bad input, fixed each one as we went until we could not find/ think of any more bad input scenarios.
4. JSDoc does not accept the format of comments specified in the wiki, ruins the connections between files

**Features that did not make demo**

1. Sound effects
   1. While we tried to add an explosion MP3 file, we could not get it to run upon loss of the game without ruining the loss function, or requiring the user to press play on the sound.
2. Preset sizes (beginner, medium, hard)
   1. We ran out of time, but it would have been cool to create 3 pre-made sizes of board, easy, medium, and hard which would have each increased in size (2x2, 5x5, 30x30)
3. Timer
   1. We wanted to add a timer so players could see how much time they had spent playing the game but it was low on our priority list and we ended up running out of time
4. Leaderboard
   1. We wanted to build a leaderboard to keep track of the high scores. Since we did not implement a timer there would be no way of keeping track of how well someone was doing, and if we did we would have needed to implement a database to store this info, something which none of us have experience doing.

**Retrospective**

Overall we were very pleased with how this project turned out. Our entire team was enthusiastic and hit the ground running on this project. Despite a lack of experience with JavaScript, we decided to try out a new language and love the way it turned out. Things we would have done differently would be working in raw JavaScript, instead of the P5 library to give us more control over our project. We also would have divided our work up in a more practical way to get a basic working version up sooner. The way we divided up work caused the people who were building features to wait until the people who built the basic game were done to start working.