

Project #2 Write Up

Hirsh Guha, Ryan Pope, Jordan Love, John Quitno, and Giang Nguyen

Meeting Log

First Meeting (3/2/19):

1. Location: LEEP2 Study Room
2. Members Attending: Giang, Jordan, Hirsh, Ryan, John
3. Outcomes: Decided on additional feature, defined the scope of the project, divied up the work.

Second Meeting (3/18/19):

1. Location: LEEP2 Room 1410
2. Members Attending: Giang, Hirsh, John
3. Outcomes: Checked in on progress, made sure Spring Break would be no obstacle.

*Most parts were completed on our own time, and communication was down primary through a group chat. This ended up being extremely effective, and wasted far less time than additional meetings would have.

Proces

Our inherited project was written in multiple language, with the backend logic and server written in Python, and the front end display written in JavaScript and HTML. Since we had a varied project to work on, our first step was more conceptual than code. We met in order to understand how we wanted to divide and conquer, and it became clear that the order things needed to happen was 1) bug fixes and restyling, 2) timer and additional features added, 3) cheat mode, and lastly 4) documentation and final testing. Since we divided up the work such that different people were working on different steps, no one was really working on the project at the same time. Instead, someone started, finished there piece, and handed it off to the next person. This ended up being a very efficient and productive strategy. Whenever anyone ran into an issue they didn't know how to solve, they simply pushed the code, and another person pushed a fix so that the original person could get back to their part.

Working in this way, we finished the whole project in about a week.

How Work Was Split Up

- **Hirsh** - Restyled the board, wrote the timer, displayed the timer, wrote an algorithm to calculate board difficulty, worked on leaderboard backend
- **Jordan** - Wrote the client side display of the cheat mode.
- **Ryan** - Wrote the server side code for a cheat mode
- **Giang** - Revamped and rewrote large amounts of backend code to improve efficiency and speed, changed styling of win and lose states.
- **John** - Bug fixes, worked on the backend of the leaderboard, finalized the

frontend of the leaderboard, styling changes.

Our Additional Feature

Our feature uses both the front end and back end of the code we inherited significantly, being a leaderboard. We have the ability for a user to input a username/ID, and then if they play with a 10x10 board with 10 mines(Though we originally allowed any size board and mines on the leaderboard, we limited it to this on purpose in order to assure every score on the leaderboard is consistent.), and they win, their best time and username is tracked to a leaderboard. Up to 10 scores are kept there, and we added a timer so they could see exactly how they were doing, comparing themselves either to other users or themselves. We also moved our game to the cloud, so the leaderboard can be accessed by anyone here: <https://minesweeper448.appspot.com/>(though the cheat mode is not on the cloud).

Challenges Faced

We were lucky in that multiple of us were already familiar with our inherited language – python. Not only that, but the front end of our inherited project was written in JavaScript, which we had just written the game of Minesweeper in. This made for a very smooth transition, and no challenges.

The area that did present a challenge was in our implementation of our additional feature, where we had trouble writing an algorithm we felt accurately represented the difficulty of the board, given that "difficulty" is a subjective variable. After multiple tries to this end, we chose instead to switch to a leaderboard design. This was still challenging, as there needed to be immense communication between our Python server and the front end JavaScript, but this was a challenge that was overcome in the end.

Features that didn't make the demo

Every feature we needed in order to meet all requirements perfectly we were able to implement. However, we were forced to change course on our additional feature. Originally, our plan was a multi-faceted scoreboard, where a real-time score display would keep track of time, the efficiency of solve, and the difficulty of board and display a final score at the end. Unfortunately, this was far from realistic within a two-week time frame. While we were able to write an algorithm that calculated board difficulty, it was never something we were fully satisfied with, and harder still was trying to quantify the efficiency of solve. Luckily, we were able to pivot from a scoreboard, to a leaderboard, where we would actually be able to store the top 10 solves in an array, and display it on a screen to create a competitive environment. We decided this version of the game is debatably more fun than if we had spent hours writing a backend just to output some number at the end.

To summarize, every feature is present in our enhanced, inherited minesweeper, however we did change plans at one point.

What we could have done different

The particular implementation of Minesweeper that we were assigned had quite a few bugs, and one since we were only locked into the language and framework, we chose to rewrite some of the messier, bug-prone parts as opposed to attempting to find and resolve the bugs in the existing code. In the end, this was probably a better decision, as it left us with a better understanding of server side python, jQuery, and of the program that we wrote. However, the approach where we could have kept as much as possible might have been faster, and would be worth attempting in some

other scenario.

Of course, another obvious place we could have done things differently is choosing the extra feature. Originally, we thought we would focus on a power-up feature, but after hearing how common this was, we decided instead on a global leaderboard. While we still believe this was the best course of action and resulted in a really cool feature, it was still a last minute decision, and there's always the lingering thought that we were wasting time on something insignificant, and taking a different course of action would have resolved these concerns.