MS2 Beta: Project Report (self-evaluation)

Team Members

Max Pace (netID map438) • Arjun Shah (netID ans96) • Jerry Xu (netID jjx6)

CS3110 Fall 2021

Vision:

Our core vision still has not changed. Our plans to build a version of Minesweeper in OCaml have progressed well, and we have not made any major changes to our plans. One small change was we implemented a stopwatch instead of a timer because we realized this is what is actually present in most versions of the game. We are on track with key features that we want to implement. However, for our next milestone, we are not too sure how much time will need to be invested to create a graphical user interface, so we are considering alternative "quality of life" upgrades if we lack sufficient time to create a good UI.

Summary of progress:

A game is not fun if it can't be played, so the bulk of our work this milestone focused on making our game playable and taking user input. This involved reading terminal inputs, parsing them (and handling malformed input), and then using the user's commands to manipulate the game state. Finally, the user needs to be able to see the board, so we print the board in a colorful and understandable way, cycling through the process again. Another part of this milestone involved implementing the flood-fill algorithm to clear empty squares (i.e. squares surrounded by zero mines)—a feature found in most (if not all) versions of the game. Finally, the game needs to end at some point, so we wrote code to check and handle when either the player wins (and clears all non-mine squares) or loses (clears a mine). We also implemented a stopwatch so that the player can measure how well they performed.

During our demo, we will demonstrate all of the above—playing the game in the terminal, winning, and losing. This will also demonstrate the functionality of the stopwatch.

Max Pace, Arjun Shah, Jerry Xu
CS3110 FA21

Activity breakdown:

Jerry and Arjun both focused on the terminal interface of the game, with Jerry writing initial scaffolding and Arjun putting in the bulk of the work to convert it from a prototype to a real playable game with good quality of life. Jerry also implemented and tested a floodfill expansion method that kicks off when empty squares are dug upon. Max implemented various aspects related to testing and also integrating a stopwatch with the terminal interface.

- Max
    - His delivered features include the stopwatch and pretty-printing in the terminal interface
    - Hours worked: 6
- Arjun
    - His delivered features include robust testing and completion of the terminal interface
    - Hours worked: 7
- Jerry
    - His delivered features include the floodfill expansion method and the terminal interface.
    - Hours worked: 6

Productivity analysis:

We took much more initiative than last time, and had more consistent meetings and a more stable workload than last time. We did accomplish what we had planned, and while our code organization especially in our terminal interface was a bit rough around the edges initially, we still have a very good grasp on our code organization and features and have room to scale to implement additional features as they come.

Scope grade:

We achieved Excellent scope in this assignment. To begin with, we'll consider what we laid out for Satisfactory scope. Our plans stated that we would be able to instantiate a board and

manipulate it according to user input via the terminal while also never erroring out, which we successfully have implemented.

We also achieved Good and Excellent scope since we are able to both floodfill and open up new squares recursively, in addition to offering to play again, reveal where all the mines are when the user loses, and also have a stopwatch running during the game to inform the user of their progress. Therefore, we met our plans for good and excellent scope.

Goals for next sprint (if GUI, most likely):

- ☐ Satisfactory: *Add a Graphical User Interface for the user to interact with. The GUI should allow clicking on the board directly instead of typing in coordinate values for the board, making for a more intuitive user experience.*
- ☐ Great: *Allow the user to choose different styling options or themes so they can view the game in the way which suits their needs and preferences.*
- ☐ Excellent: *Display stopwatch on the GUI, display win-loss situations*

Goals for next sprint (if not GUI):

- ☐ Satisfactory: *Be able to save game state to disk via CSV and JSON. Display flag count and mines left.*
- ☐ Good: *Enable auto-dig auto-completion feature*
- ☐ Excellent: *Prettify terminal interface by displaying stopwatch updates in real-time, overwriting previous terminal output. Allow the user to choose different styling options via the CLI so they can view the game in the way which suits their needs and preferences.*