Web-Sweeper

Table of Contents:

Group 11

Andrew Smith
Stamatios Morellas
Shivam Vashi
Minji Park

Project Summary

Our project is Web-Sweeper. It will be a web-app minesweeper game. A player will go to our web page and will choose a difficulty they wish to play. From there they will be shown the minefield with a timer on the top and a leaderboard on the side. When a player clicks on a spot in the minefield if there are no mines adjacent to the spot it will reveal all spots around until there is a border of spots that have mines adjacent. If the player clicks a spot that has mines adjacent it will reveal that spot showing the number of adjacent mines. If the player clicks on a mine then the game will end and all mines will be revealed. The player will have the option to flag a possible mine. If only the mines are flagged or only the mine spots are left then the player wins. After the player wins they will, they will be given an option to save their scores in the leaderboard if their score beats the top 100 players for that difficulty. Then they will be shown a screen with the leaderboard and an option to play again. If the player fails the game they will be given the option to play again. In either case, if the player wishes to not play again the game will end and the player taken back to the original screen.

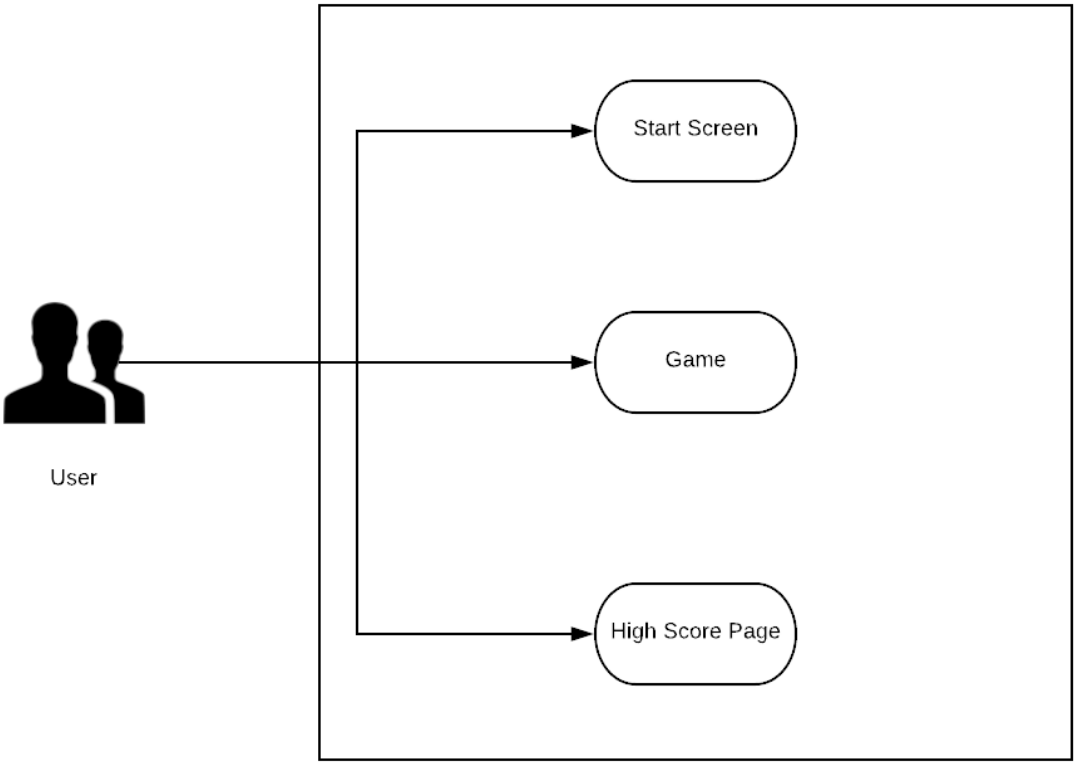
Objectives are to create a web application for players to play minesweeper. We also want to store their score in a leaderboard. Have multiple difficulties for the player to choose from.

Final outcomes are to make an application that works as described above and make the application for people who love minesweeper or people who want to play a thinking game.

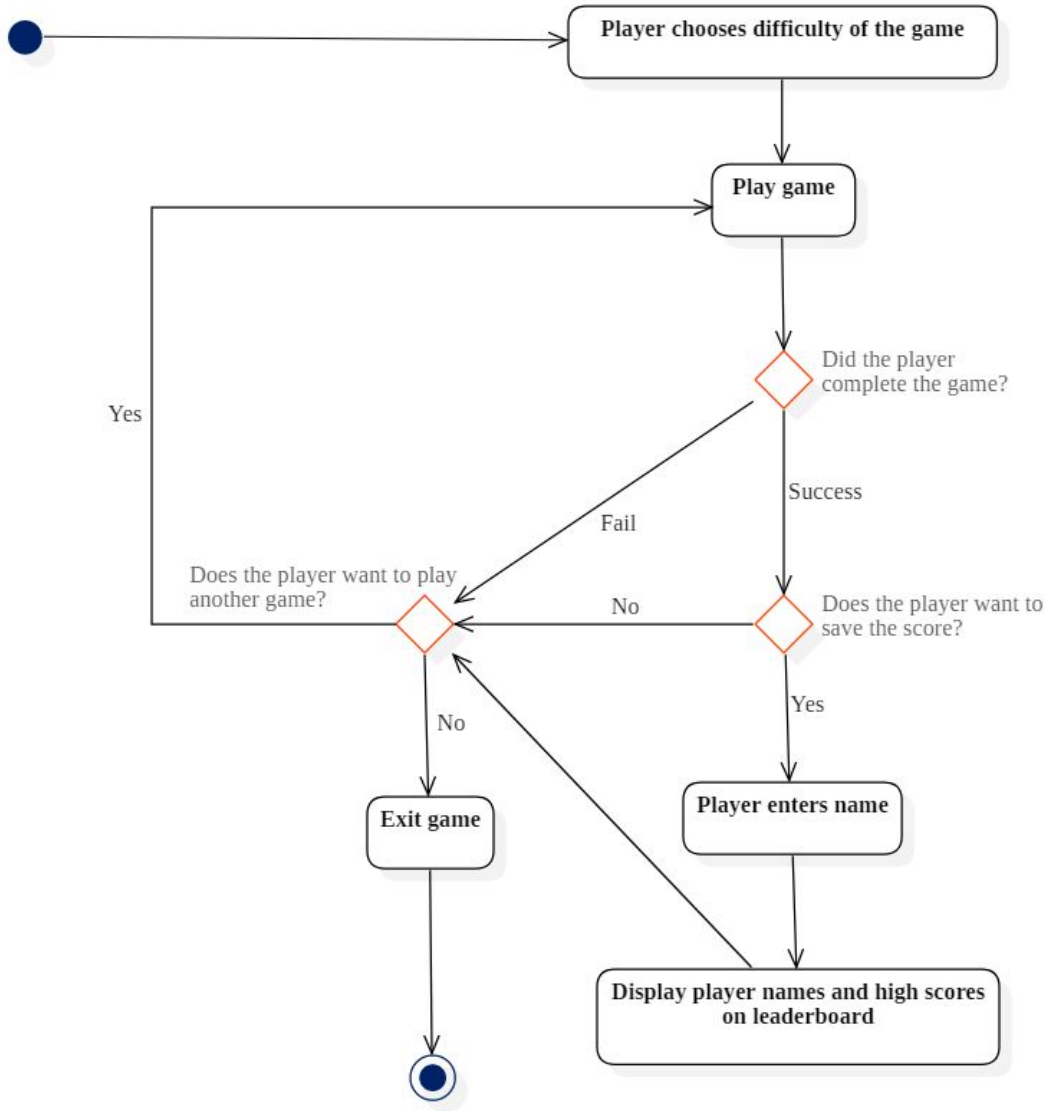
Benefits will be knowledge of NodeJS and Javascript along with internet protocols. Learning React Framework.

Limitations are going to be scheduling a time to meet as well as our commitment to other classes workload. Difficulty working with new languages and frameworks.

Use Case Diagram



Activity Diagram



Use Case Description

1. **Play Game**
    a. Participant: *Player/User*
    b. Input Condition: *Player is at the main menu*
    c. Output Condition: *The game starts*
    d. Flow of Events: *User opens the application and is greeted with the startup screen*
    e. Special Requirements: *None*
2. **Choose Difficulty**
    a. Participant: *Player/User*
    b. Input Condition: *Player chooses to play on easy, medium, or hard*
    c. Output Condition: *Makes the game tougher to play based on what is selected*
    d. Flow of Events: *Player chooses difficulty first before selecting to play the game*
    e. Special Requirements: *None*
3. **Leaderboards**
    a. Participant: *Player/User*
    b. Input Condition: *User is at the main menu*
    c. Output Condition: *Displays a table of all logged player names and high scores*
    d. Flow of Events: *User can view this automatically after a game is finished or from main menu*
    e. Special Requirements: *None*
4. **About**
    a. Participant: *Player/User*
    b. Input Condition: *User is at the main menu*
    c. Output Condition: *Displays a screen talking about how the developers made the app*
    d. Flow of Events: *Player is able to navigate here only from the main menu*
    e. Special Requirements: *None*
5. **Exit**
    a. Participant: *Player/User*
    b. Input Condition: *User completes a game or is at the menu screen*
    c. Output Condition: *Closes the app*
    d. Flow of Events: *Done from main menu*
    e. Special Requirements: *None*

Implementation Outline

Overall: We will be using an agile approach with slack to communicate with each other. We will also set up a github for version control. Our main UML diagrams will be the activity diagram and the use case diagram.

Frontend: We will be using Javascript as our language, specifically the React.js framework, to develop the front-end functionality as well as UX/UI. We will use the node package manager, as well as HTML and CSS when we need it. We will be testing it natively on our devices on localhost.

Backend: In the backend we are going to be using NodeJS as our language with MySQL as our database language. The server will be run on a personal Ubuntu server. Testing we will be using Jtest; framework we will be using Express; and our package manager npm. We will using using VSCode IDE and MySQL workbench IDE.