EECS 581 Project 2 Minesweeper Game: System Documentation

Person-Hours Estimate

Week 1:

1. As a developer, I want to ensure that the behavior of the program is as expected.

    (6-8 hours, 2 people)

2. As a user, I want a separate mode to play against Easy AI.

    (2 hours)

3. As a user, I want a separate mode to play against a Medium AI.

    (6 hours)

4. As a user, I want a separate mode to play against a Hard AI.

    (3 hours)

5. As a user, I want to have sound effects and music played in the background while playing.

    (~9 hours, 3 people)

6. As developers, come up with app ideas for project 3.

    (N/A)

7. As a user, I want to select a mode between Easy, Medium, and Hard in AI mode.

    (1 hour)

8. As a user, I want to be able to select between AI mode and singleplayer mode.

    (1 hour)

Week 2:

1. As a user, I want to choose between an interactive and an automatic AI.

   (1 hour)

2. As a developer, I want to be able to tell which comments and documentation are new.

   (1 hour)

3. As a user, I want a separate mode to play against an Easy AI.

   (2 hours)

4. As a user, I want the first click to reveal the first cell without having to click twice.

   (1 hour)

5. As a user, I wish to be able to have the AI interact with the DOM based upon the algorithms determining what move it plays.

   (5 hours)

6. As a user, I want the first click to always reveal a safe cell.

   (1 hour)

7. As a user, I want the AI to place a flag on the grid when it determines a tile to be a mine.

   (4 hours)

8. As a developer, I want to separate the JavaScript into a separate file once all the scripting is complete.

   (1 hour)

9. As a user, I want to have project documentation with a high-level description of the system architecture.

   (1-2 hours)

10. As a user, I want to have a software architecture diagram that describes the flow and functionality of the new features.

(2 hours)

11. As a developer, I want the documentation to be well-formatted and edited.

(2 *hours*)

12. As a user, I want the program to be well tested with consistent usage.

(3 hours)

13. As a user, I want to have each user story reformatted into the clients desired template.

(1-2 hours)

14. As a user, I want to create a take turns system for whenever I play against the AI.

(2-3 hours)

15. As a user, I want a safe zone to be generated around my first click.

(1 hour)

Actual Person-Hours

Elizabeth's Person-Hours:

- As a user, I want to have sound effects and music played in the background while playing: 1 hour
- As a user, I want a safe zone to be generated around my first click: 1 hour
- As a user, I want the first click to reveal a safe cell: 1 hour
- As a developer, I want to ensure that the behavior of the program is as expected: 4 hours

Marco's Person Hours:

- As a developer, I want to ensure that the behavior of the program is as expected: 2 hours
- As a user, I want to have sound effects and music played in the background while playing: 3 hours
- As a user, I want to have project diagram with a high-level description of the system architecture: 2
- As a developer, I want to separate JavaScript into a separate file once all the scripting is complete: 30 minutes
- As a user, I want to have each user story reformatted into the clients desired template: 2 hours
- As a user, I want to create a take turns system for whenever I play against the AI: 5 hours

Addie's Person-Hours

- As a user, I want a separate mode to play against a Hard AI: 3 hours
- As a user, I wish to be able to have the AI interact with the DOM based upon the algorithms determining what move it plays: 5 hours

Janna's Person-Hours:

- As a user, I want to have sound effects and music played in the background while playing: 2 hours
- As a user, I want to select a mode between Easy, Medium, and Hard in AI mode: 1 hour
- As a user, I want to choose between an interactive and an automatic AI: 2 hours
- As a user, I want to have project documentation with a high-level description of the system architecture: 1.5 hours
- As a user, I want the program to be well tested with consistent usage: 1 hour

Anya's Person-Hours:

- As a developer, I want to ensure that the behavior of the program is as expected: 1.5 hours
- As a user, I want a separate mode to play against Easy AI: 2 hours
- As a developer, I want to be able to tell which comments and documentation are new: 30 minutes
- As a developer, I want the documentation to be well-formatted and edited: 2 hours

Hunter's Person-Hours:

- As a user, I want a separate mode to play against a Medium AI: 5 hours
- As a user, I want the AI to place a flag on the grid when it determines a tile to be a mine: 1 hour

System Architecture Overview

I. High Level Description

a. System Components

The program is built in an HTML file and contains JavaScript as well. The frontend UI is implemented with HTML and user input is then handled in JavaScript. For the project 2 requirements, the AI solver also utilizes JavaScript. The custom addition to the game is sound effects for uncovering mines, adding or removing flags, and winning or losing the game. There is an animation that plays when a mine is clicked to reveal all the mines.

b. Data Flow

The program defaults to 15 mines and the classic single player mode. The user can use this or change the game mode to interactive or automatic AI. If an AI mode is chosen, the user must specify the difficulty of the AI model (easy, medium, hard). If the user chooses an AI mode but does not specify the difficulty, an alert pops up if the start game button is clicked to tell the user that the difficulty must be chosen before starting the game.

Once the inputs are set, the game mode is handled accordingly. If the classic single player mode is chosen, functionality from project 1 remains. If the automatic AI is chosen, it will autocomplete depending on the AI difficulty. If the interactive AI is chosen, game play moves forward as a two-player game with the user and AI alternating turns. In the interactive AI mode, the user makes the first move/click and passes its turn to the AI. When the AI makes its move, the program checks if it chose a mine, and if it did, it enters the game over state. If it chooses a safe cell, the turn is passed to the player, and the process repeats until a mine is hit or until the game is won.

The sound effects are handled in the corresponding JavaScript functions for each use case. The mine uncovering and flag sound effects are in use throughout gameplay, but the win/lose sound effects are played one time once the user wins or loses.

c. Key Data Structures

The user input for the game mode and difficulty, if necessary, is handled in the HTML for the user interface, and in apply_settings() for the JavaScript. The additions in apply_settings() consist of getting and saving the user input for the game mode and difficulty and then calling the appropriate function AI move functions if the user chooses an AI game mode.

If the classic single player is chosen, the data structures remain as in project 1. The automatic AI mode is handled in aiPlayAlone function where the user does not play with the AI, and instead the AI will perform its moves/clicks one at a time until it wins or game over. The move is chosen by calling the appropriate difficulty function until the game is no longer in progress. In the interactive AI mode, an event listener waits for the player's first click and it as usually in left_click, and then the turn is passed to the AI. Once the AI makes its move, it is also checked in left_click. If a mine is hit, gameOver is called. If a safe cell is revealed then the program checks for victory, and if there no victory, it loops back to the user's turn but if there is a victory, the game ends. This process continues to loop until a mine is hit and gameOver is called or until the game is won and activateVictoryMode is called. In both cases, the game has ended.

The easy AI is handled in bestEasyMove where the AI chooses a random cell to click and returns it to aiMakesMove (if interactive) or aiPlayAlone (if automatic). The hard AI is handled in bestHardMove where first it checks if the board is initialized and clicks a random cell if not (in the automatic AI case). Then, it chooses a random cell from an array that contains the cells that are safe (no mine) and are hidden (not yet clicked). It then flags the randomly selected safe cell. If it cannot find a best safe cell, bestMediumMove is called.

The medium AI is handled in bestMediumMove. This function checks if the cell is on an edge (so that nonexistent cells are not flagged) and if the neighboring cells are hidden. Then, the function checks if it the number of hidden neighboring cells equals the number in the cell and flags the neighboring cells if so. If the number of neighboring flags equals the number in the cell, then the best medium move is found and returned to bestMoves

The sound effect files are in the /soundFiles directory. The sound effects for mine uncovering and flag placement/removal are handled in the left_click function and the f-key event listener respectively. The game over sound is also handled in the left_click function. The victory sound effect is handled in the activateVictoryMode() function and is called in the left_click function after checking for victory if there is a victory.

The mine animation is handled in the gameOver function and plays whenever gameOver is called to reveal the mines.

II. System Components Diagram

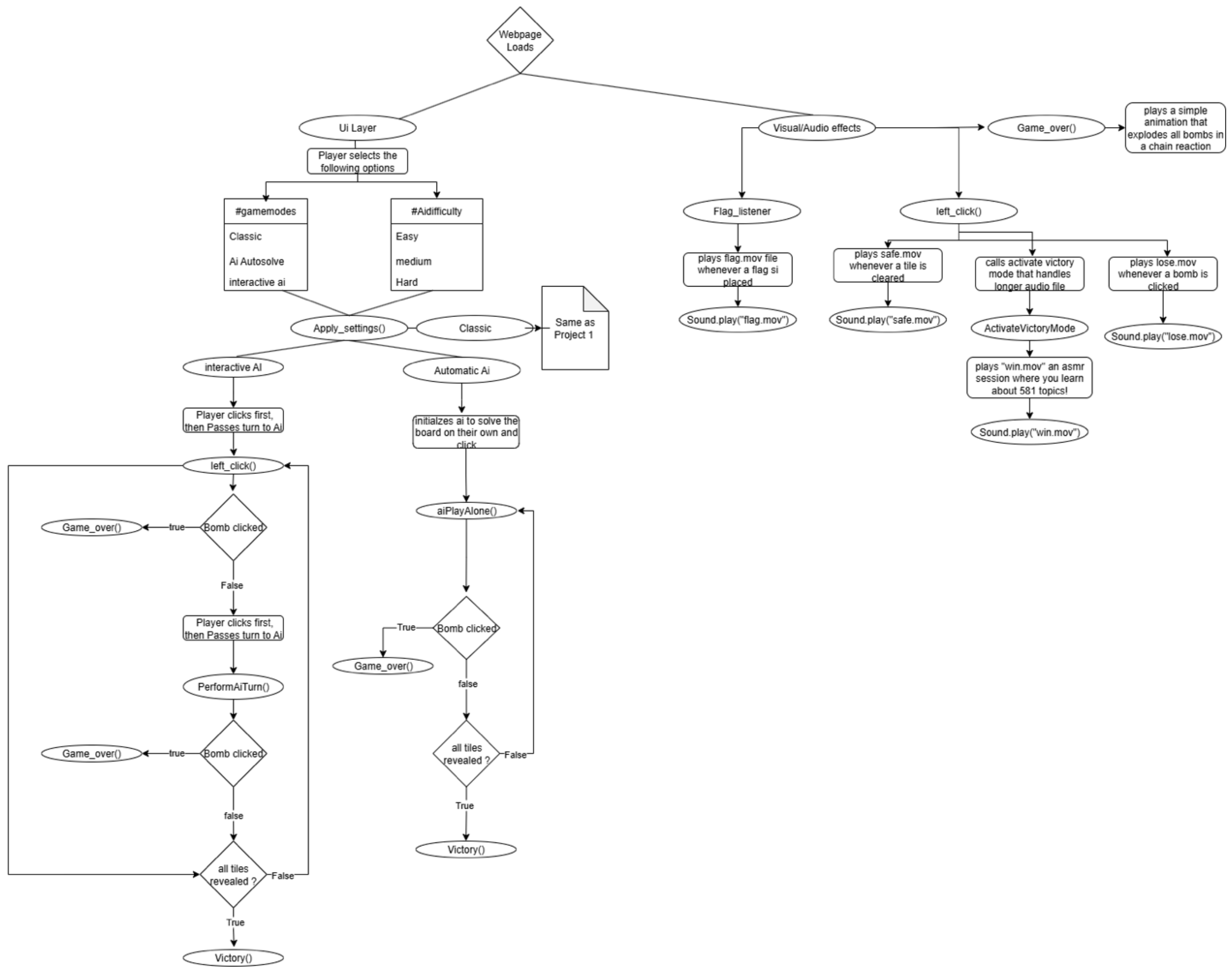

Figure 1. System Architecture Diagram. A larger image is in the GitHub repository. This figure displays how the JavaScript functions in the *src* folder are connected.