Mineswepper Transition Documentation

How the Game Meets Requirements

**Single Player**

* The game is strictly for one player. No multiplayer features are included.

**Quick Play**

* The 10x10 grid and simple mechanics ensure the game can be played quickly, making it perfect for short breaks or walks between classes.

**Scoring**

* A timer is already in place to track gameplay duration. This will need to be expanded to create a scoring system based on time and tiles revealed.

**Mobile Friendly**

* The game works “well” on smartphones:
  + It’s responsive.
  + Touch controls like tapping and long-press for flagging are already implemented.

**Small File Size**

* The entire game, including assets, is under the 10MB limit.

**Compact Layout**

* The game fits within the required 400x800px grid.

**Minimal Data Collection**

* The only data we’ll collect is the player’s initials for the leaderboard.

What Still Needs Work & Next Steps

* End screen
  + Shows statistics
  + A way to calculate score
  + A way to ask the user the three initials for the leaderboard
* General improvement of the UI aesthetic
* Better way of swapping touch functionality
* Fix cell flagging on mobile
  + When a cell is flagged and then unflagged, the cell is revealed.
* Additional testing coverage on Mineswepper.js

File Breakdown

* Index.html
  + Main entry point for the game.
  + Needs hooks for the leaderboard and game over UI.
  + Contains elements for the game (e.g., <div id="game-board">) and controls (restart button, timer).
  + Links to style.css for layout and visual design, and Mineswepper.js for game logic.
  + Includes a temporary button linking to the rules page (rules.html).
* style.css
  + Provides styling for the game (subject to change)
* Mineswepper.js
  + Core game logic
  + Interacts with Tile.js for *individual* tile behavior.
  + Key methods: initializeGame(), restartGame(), generateMinesAfterFirstClick().
* Tile.js
  + Defines the Tile class:
    - Handles individual tile states: mine, revealed, flagged, adjacentMines.
    - Methods for tile actions: setMine(), reveal(), toggleFlag().
  + Supports mobile touch gestures (long-press to flag)
* Tile.test.js
  + Performs insane DOM mocking to test:
    - Tile initialization
    - Flagging
    - Revealing
    - Mechanics
    - gameOver state
* rules.html
  + Provides objectives, rules mechanics for revealing and flagging tiles.
  + Contains a navigation button to return to the game.
    - Navigation to rules will need to be updated when integrated with final infrastructure.
* color\_scheme.md
  + Defines granular classic Minesweeper color palette.
* main.py
  + Selenium UI testing file, performs key functions of the game automatically such as flagging and unflagging cells and revealing cells.
  + Flags and Unflags cells first
  + Simulates random clicking to reveal cells until game ends

Summary

*The Minesweeper project meets key requirements such as single-player functionality, quick gameplay, mobile responsiveness, and small file size. While the core mechanics are implemented, further enhancements are needed for scoring, UI aesthetics, touch functionality, and bug fixes. Final testing and integration steps remain to ensure a polished and enjoyable user experience*.

Credits

**Team Mineswepper**

**Product Owner:**

Curtis Reece

**Scrum Master:**

Christopher Oaks

**Development Team:**

Chris Powers

Christian Rock

Gavin Frisby

Nathan Mink

Scotty Snyder

Tyler Campbell

Zach Farmer