Team Kale 🥗 -- Andrew Juang, Eliza Knapp, Ella Krechmer, Lucas Lee
SoftDevP02:
2022-03-02

Description

Competitive minesweeper
We will be creating a minesweeper game with a community leaderboard for each difficulty of minesweeper board. Users will be able to play a minesweeper game on easy/medium/hard level without logging in, but are able to see the leaderboard for that board (level), based on time (less time = higher on the leaderboard). The user can log their score on the leaderboard by logging in (or creating an account). The leaderboard will be updating while other people are playing. Users who are logged in can create their own boards and anyone (logged in or not) can search for those boards by the user who created them (more detailed info on the leaderboard below).
Further Exploration: If we have time, we will also implement a play-with-friends functionality, where a player shares a code so many people can join. They can start the game whenever they want, which causes a timer to start. The person who has unearthed the most ground at the end of the timer wins.

Program Components

Backend
__init__.py
- Connections to the html forms

User.py
- Add account (username and password) to database
- Check if username is already in the database, if so, prompt user to log in instead
- Create the user database if it doesn't exist

Board.py
- Generate board- uses math and recursion to generate boards at the 3 different difficulty levels, given as a parameter
- Make board- takes in the board size and an array of the locations of bombs and saves it to the database
- Search board- given a username, you can search for that person's boards
    - This shows the person's highscores and the boards the person has made
- Update leaderboard
- Pull leaderboard data
- A board object with
    - an array where the bombs are
    - An array of booleans to see if they are visible to the player

Frontend

Game.js

- The actual js behind the minesweeper game
- Create a canvas element and edit it

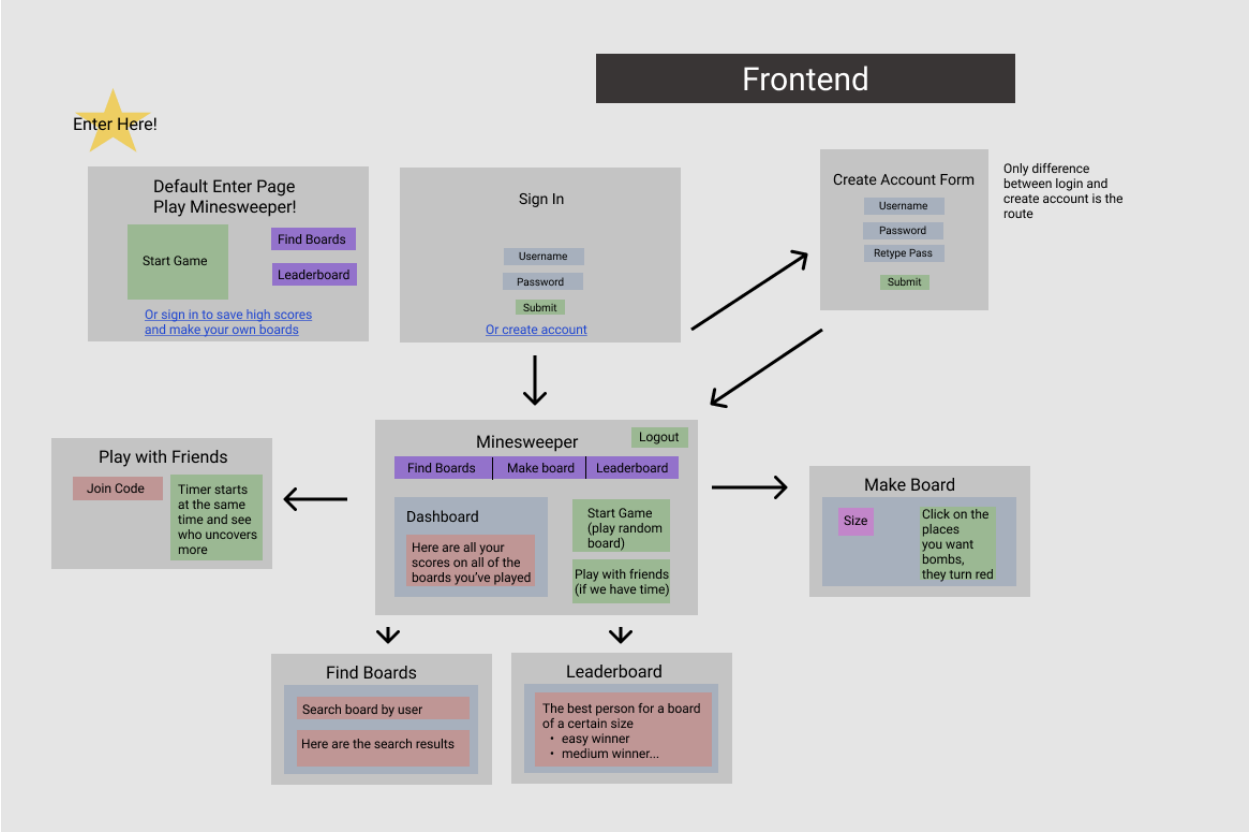
Buttons.js

- The clicking functionality of buttons and live updating
- More here with bootstrap for the css

Styling

- We will be using bootstrap because we like the component flexibility and all have used it in the last assignments
- For the stuff that bootstrap does not come with, we will be using our own css

Site Map



The leaderboard has 3 sections:

1. **Top Scores:** The top scores overall for the easy, medium and hard auto-generated boards
2. **My Scores:** YOUR top scores for the easy, medium and hard auto-generated boards

3. **High Score per Board:** High scores for each of the boards that were created by users

Database

Users

Username (text)	Password (text)	UserID (int) (primary key, auto increment)
Tooth myklick	iloveunicorns	0

User Boards [that people have created]

BoardID (int) (primary key, auto increment)	Size (int)	Array of bombs (SizexSize Array) *Stored as a string of 1s and 0s*	User with highest score	High Score	Author (int) (UserID)
0	7	0101001…	bob	40	potato
1	9	0101001…	joe	23	sloth123

Auto Boards [boards that were randomly generated by our python functions]

- For each user, a their high score on the easy, medium, and hard boards are saved

Level	User who scored	High Score
Easy	Bob	45
Hard	Joe	20
Medium	Bob	32
Hard	Bob	10

Tasks

Andrew

- Login functionality
- Basic minesweeper functionality and linking js and html

Ella

- Css at the end
- Leaderboard and general html

Eliza

- User.py database functions
- Board.py database functions
- Search boards

Lucas

- Javascript functions
- Make boards

Everyone

- Play with friends functionality