Quiet Shower Thoughts - Jonathan Metzler, Aditya Anand, Caden Khuu, Stella Yampolsky

Software Development

P01 - Geography Game

Target Ship Date: 2024-1-17

Project Summary:

We are working on recreating a simplified version of geography games like Globle. Our project involves selecting a random country using an API. The user enters their guess, and we use a distance API to calculate how far their guess is from the target country, providing this information as feedback. The user can then make additional guesses, repeating the process until they identify the correct country. To make the game more engaging, we plan to include optional hints, such as the country's flag, a geographical map, a photo of the country retrieved via an API, or the first letter of the country's name.

Components Explanation

- **Flask**: Acts as the central hub connecting the frontend, backend, and database. It handles user requests and acts as a connector.
- **SQLite3 Database**: Stores user accounts, game statistics, and balances. Flask interacts with the database to retrieve and update data based on user actions.
- **Game Modules**: Each game module (Map feature(?), Distance, Flag) interacts with its respective API to process game logic. The results are sent to Flask for storage in the database and displayed on the frontend.
- **Front-End Templates**: Provides the interface for user interaction. Pages like home.html display personalized stats and links to games based on database queries handled by Flask.

APIs: Serves as the backbone for individual features of the game, providing hints(e.g., distance, flag, and photo) to the game modules. These outcomes are used to update the database and user score.

App Folder

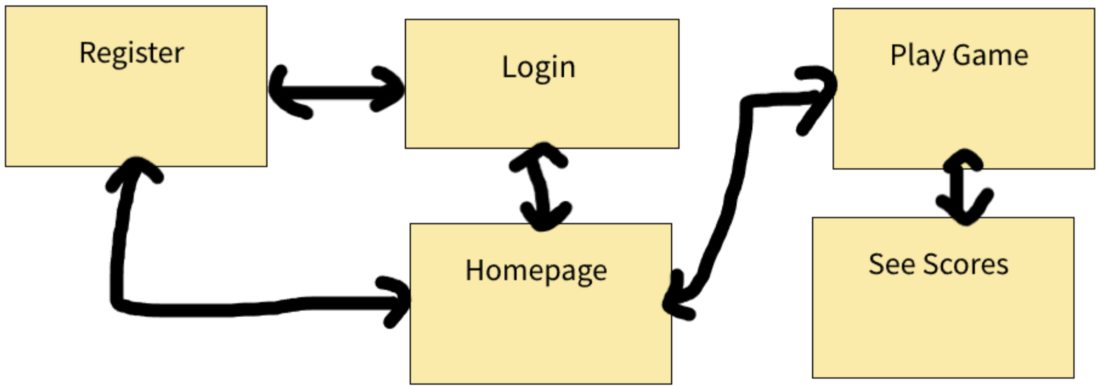
- build_db.py
  - Use sqlite3 to create a database of user information (username and password).
- __init__.py
  - Initializes the app.
- config.py
  - Checks that API keys exist and are stored in the right place.
- users.csv
  - Stores Users' username and password
- user.db
  - Stores the users and memes tables with respective information.
- keys folder
  - key_GoogleFonts.txt
  - readme
    - Explains the state of API keys
- Templates Folder
  - login.html
  - Homepage.html
    -
  - Register.html
    - Register user account
  - Game.html
    - System of 5 rounds where the player has to guess a country to get the lowest distances between their guess and the actual location
  - Scores.html
    - Shows average score out of 6
    - Shows the lowest distances for each game in the total of the 5 rounds

logout.html (just returns to login)

Application Logic

- __init__.py
  - Flask Server, designating routes and connections between pages + their functionalities
- config.py

Securely loads API keys from their respective files. It makes sure the keys aren't hard coded into the app.



- **FEF**
  - Tailwind
    - Very easy to add lots of customization to any element used, such as tweaking colors, position, padding,etc.
    - We want to add a theme to our website,related to 'games and gambling', such as using bright colors so such customization will help
    - Has built-in support for state variants (eg: hover, focus, active, etc) allowing for easy and clear game flow
    - Flexible for third-party integration since no specific structure or dependency is imposed. Easy to pair with backend system that consumes APIs

Game logic:

- **Random country is selected, guesses, distance(?) and hints are set to 0**
- **When a user guesses a country, guesses ticks up, distance from target country is displayed**
- **If the user chooses to use a hint one is displayed**
  - **First hint weather in the target country**
  - **Second hint country's flag**
  - **Third hint first letter**

    - **Letters are filled for each hint until the string is completed and the game is forfeit (Or not, depending on how stupid playtesting makes me look).**

<u>Database Organization</u>

- user.db
- Tables:
    - users
        - Formatted data of usernames and passwords, each corresponding to an index by line
        - New items appended to the end
        - Cannot be deleted
        - Saves the users average score and a list of all games that the user has played
    - scores
        - A certain game can be shown with all guesses
        - Saved to the database per game

<u>Database Table</u>

Table scores:

Play #	# of guesses	Date	Guessed locations
int	int	int,int,int	array[string]

Table users:

Username	Password	Average Score	All games
String	String	double	array[scores]

<u>Assignments</u>

<u>Task</u>	<u>Assigned To</u>
Database Engineering	Jonathan (PM)
HTML and Front End Framework	Caden
Flask	Aditya
APIs, basic game functionalities	Stella