Homework Week 2
Group Project

Team Members:

- Alisha Musamabik
- Siân Tipping
- Timea Kiraly
- Jayashree Karthikeyan
- Lima Jamal

PROJECT TASK

For our project we are building an educational gaming application. This application is going to be built on the concept of the traditional Snake game.

As children love to spend most of their time playing games, we want to provide an environment for them to also learn in a fun way. In order to achieve this goal we are planning to implement the following features:

- Finding the word from the given letters
- Solving an arithmetic problem

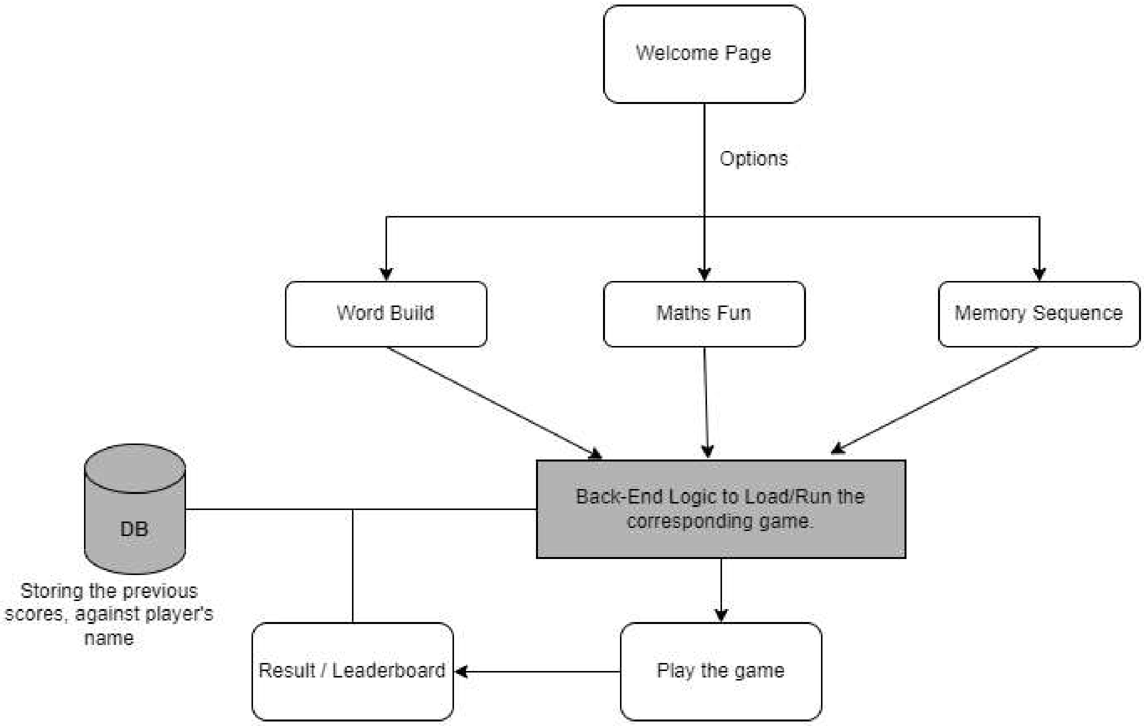
In addition to the above, we want to have:

- A leaderboard containing the individual scores for each category.
- Multiple levels of complexity in the game.

A brief summary of the game:

- The ‘snake’ will collect letters or numbers instead of the standard ‘apple’ from the traditional Snake game.
- Once collected, it will be stored in a list and compared to the correct answer from the dictionary.
- We are using Python’s time built in library to create a countdown for the level. The aim is to collect as many items in the correct order as possible within the time limit.
- The score will then be stored in a database, and the scoreboard will be based on this.

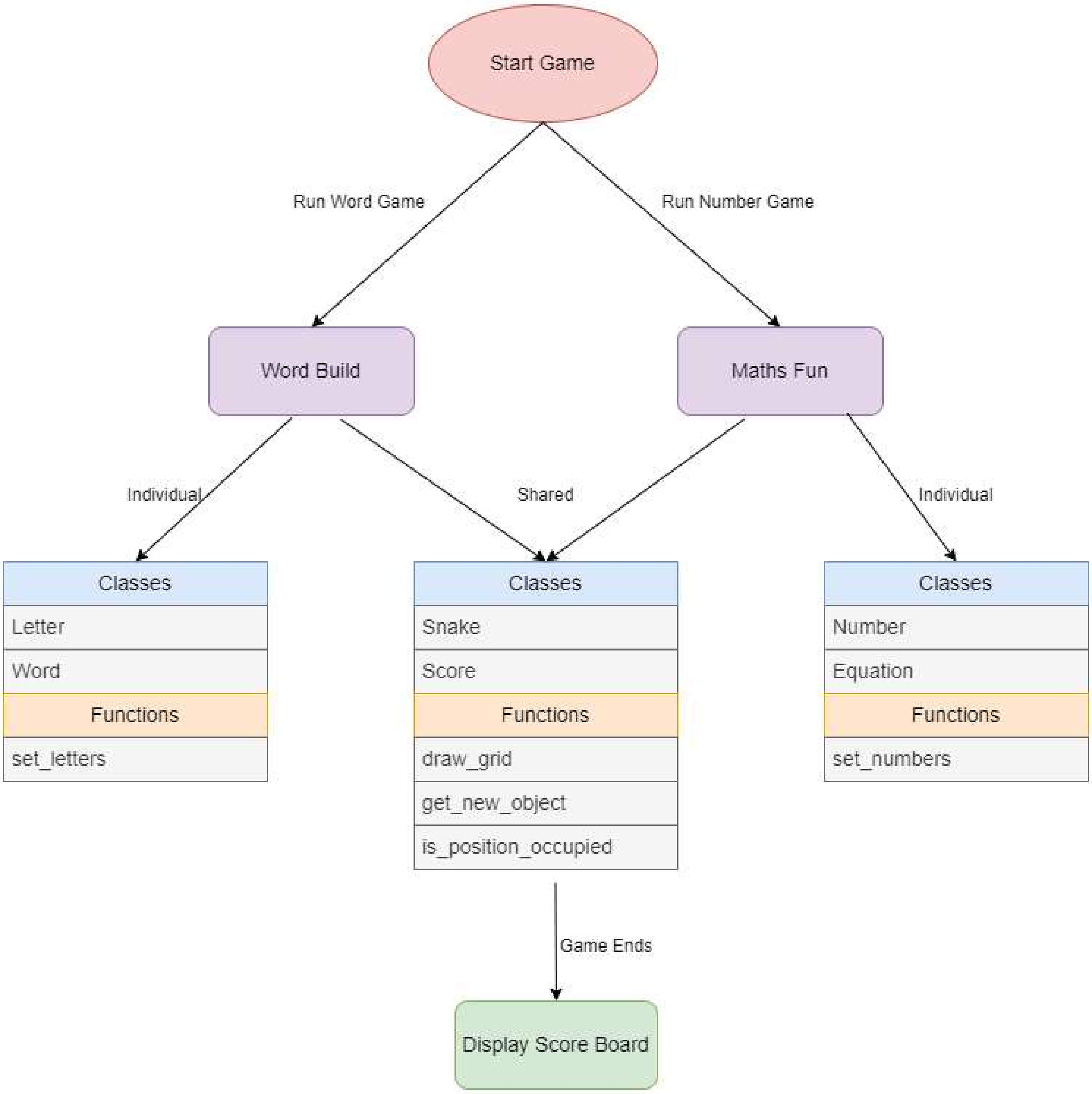
Architecture Diagram of System:


Welcome Page
Options
Word Build
Maths Fun
Memory Sequence
Back-End Logic to Load/Run the corresponding game.
DB
Storing the previous scores, against player's name
Result / Leaderboard
Play the game

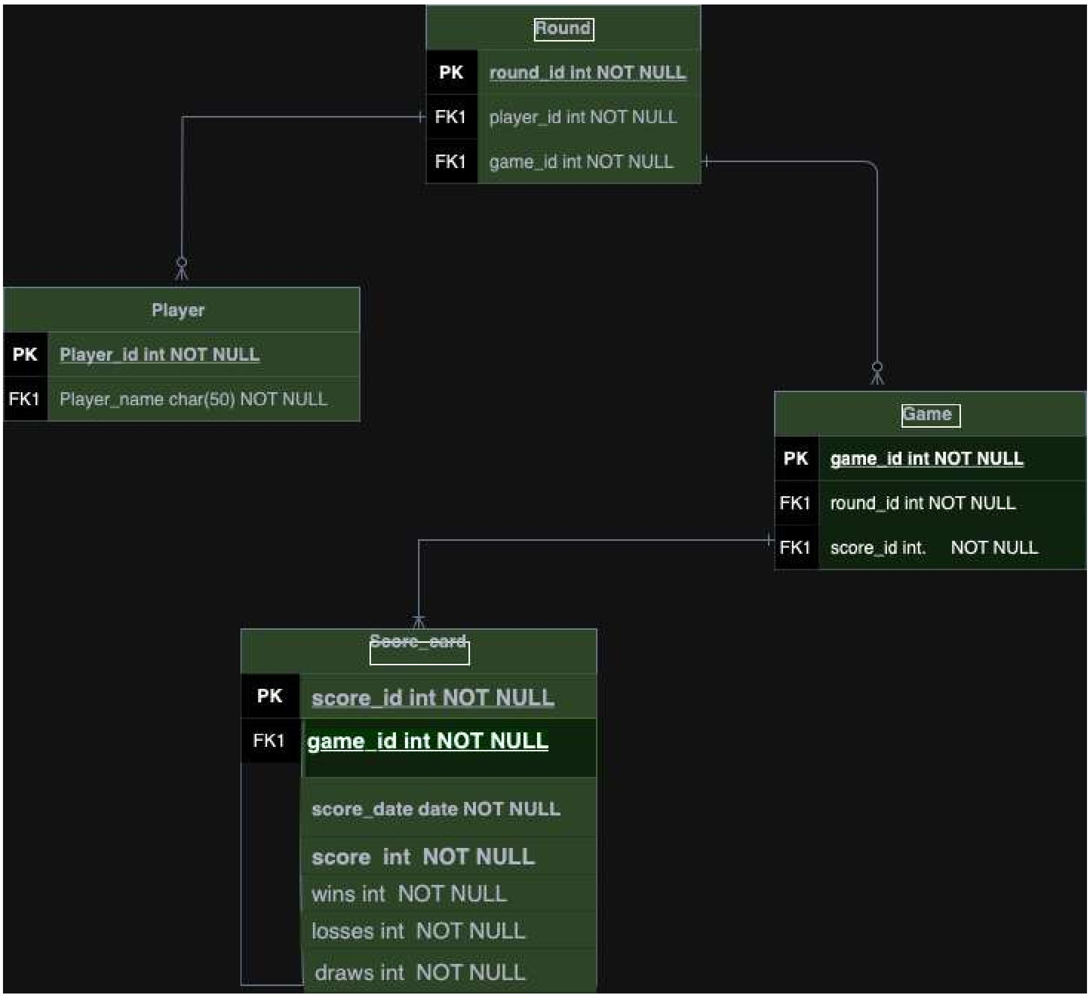
Our base structure. Note: this may be built on during development, if required.

Detailed Backend Logic:

The following diagram shows the proposed backend logic in more detail. When the game application is started, multiple buttons are displayed, enabling the user to choose the type of game they would like to play. After a button is clicked, the corresponding function ('run_word_game' or 'run_number_game') is called to start and run the game. Once the game is over, the scoreboard is displayed containing the result of the game, along with previous information from the database.



Database Structure:



Round	
PK	round_id int NOT NULL
FK1	player_id int NOT NULL
FK1	game_id int NOT NULL

Player	
PK	Player_id int NOT NULL
FK1	Player_name char(50) NOT NULL

Game	
PK	game_id int NOT NULL
FK1	round_id int NOT NULL
FK1	score_id int. NOT NULL

~~Score_card~~	
PK	score_id int NOT NULL
FK1	game_id int NOT NULL
	score_date date NOT NULL
	score int NOT NULL
	wins int NOT NULL
	losses int NOT NULL
	draws int NOT NULL

Team Approach

We considered different project management options. We have decided on GitHub so we can keep the code and task organisation in one place. We have opted for the agile project management method to allow for greater flexibility and control over each stage of the project. The GitHub repository is already set up within the CFGSoftware2-23 space.

Each section of code will be in a separate file and then any changes will be raised as an issue within the project in GitHub, and each issue will therefore have its own branch. Each member of the group will have at least one file to work on, and also designate themselves to a reasonable amount of issues. These will be peer reviewed before being pushed to main. We are aiming to keep branching on two levels so the project is organised in as simple a way as possible.

All code will be in snake case, and we will use underscores to separate words within a file name or name in the code. Naming of files will make it extremely clear what the piece of code is for, and comments are being added in a separate line above each bit of code it refers to, in some cases where comments are very short they will be on the same line as the code. Regarding testing, we will use unittest to create all relevant tests on our code, and keep these in a separate file.

Task designation is taking place during our regular calls, every other day after the live session. Each member of the team is voicing what area they feel confident on and what tasks they are happy to take. They will then self-assign to an issue on GitHub. We are also using Slack as our main communication channel and are consistently communicating there outside of calls.