Drake The Snake (menu)

This project is a graphical user interface (GUI) for the Snake Game Menu using Tkinter in Python. The menu allows users to select different game modes, view highscores, show game rules, and exit the game. The game modes include Classic, Trivia, and Multiplayer. library for the graphical interface and Pygame for sound effects.

Table of Contents

Installation

Steps

1. Clone the repository:

```
git clone https://github.com/yourusername/snake-game-menu.git
 cd snake-game-menu
```

2. Ensure the following files and directories are in place:

- Interfaces/ClassicMenu.py
- Interfaces/TriviaMenu.py
- Interfaces/MultiplayerMenu.py
- Interfaces/HighscoresWindow.py
- database/rules.json

Usage

1. Run the main menu script:

```
python main_menu.py
```

2. Interacting with the menu:

- Classic Mode: Start the classic Snake game.
- Trivia Mode: Start the trivia version of the Snake game.
- Multiplayer Mode: Start the multiplayer version of the Snake game.
- Highscores: View the highscores.
- Show Rules: Display the main rules of the game.
- Leave Game: Exit the application.

Project Structure

- `main_menu.py`: The main script to launch the game menu.
- `Interfaces/`: Directory containing the modules for different game modes and highscores.
- `ClassicMenu.py`
- `TriviaMenu.py`
- `MultiplayerMenu.py`
- `HighscoresWindow.py`
- `database/rules.json` : JSON file containing the game rules.

Code Explanation

1. `create_main_menu()` : This function creates the main window for the game menu and adds buttons for each game mode, highscores, game rules, and exit.
2. `start_game(mode)`: A placeholder function to start a game mode.
3. `show_highscores()`: A placeholder function to display the highscores.
4. `show_rules(root)`: Reads and displays the game rules from `rules.json` in a message box.
5. `leave_game(root)`: Closes the application.

Drake The Snake (classic mode)

This is a classic Snake game implemented using Python's Tkinter library for the graphical interface and Pygame for sound effects.

Table of Contents

Installation

Instructions on how to install the project.### Prerequisites

- Python 3.x
- Tkinter (usually comes pre-installed with Python)
- Pygame

Steps

1. Clone the repository:

   ```
   git clone https://github.com/yourusername/drake-the-snake.git
   cd drake-the-snake
   ```

2. Install Pygame:

   ```
   pip install pygame
   ```

3. Ensure you have the `correct.mp3` sound file in the same directory as your script.

How to Play

Use the arrow keys to control the direction of the snake. The goal is to eat the red fruits that appear on the screen. Each fruit increases your score by 1 and makes the snake longer. The game ends if the snake collides with the screen edges or itself.

Code Overview

The main components of the code are:

1. **Initialization**: The game initializes the Tkinter window, Pygame sound, and sets up the game dimensions

based on the screen size.

2. **Drawing Functions**:

   - `remplir_case(x, y)`: Draws a filled rectangle representing a part of the snake.
   - `dessine_serpent(snake)`: Draws the entire snake.
   - `dessine_fruit()`: Draws the fruit as a red oval.

3. **Game Mechanics**:

   - `case_aleatoire()`: Generates random positions for the snake and fruit.
   - `etre_dans_snake(case)`: Checks if a position is part of the snake.
   - `fruit_aleatoire()`: Generates a random fruit position that is not occupied by the snake.
   - `mise_a_jour_snake()`: Updates the snake's position and handles collisions with the fruit and edges.

4. **Controls**:

   - Functions like `left_key(event)`, `right_key(event)`, etc., handle the arrow key inputs to change the snake's direction.

5. **Game Loop**:

   - `tache()`: The main game loop that updates the game state and redraws the graphics every 70 milliseconds.

6. **Game Over**:

   - `serpent_mort(NouvelleTete)`: Checks if the snake has collided with the edges or itself.
   - `afficher_fenetre_perdu()`: Displays a game-over window with the score and options to replay.

Drake The Snake (multiplayer mode)

This is a Python implementation of a multiplayer snake game using the Tkinter library for the graphical user interface.

Table of Contents

Features

- Fullscreen gameplay
- Two-player support :
  - Player 1: Arrow keys
  - Player 2: Configurable control scheme (WASD or ZQSD)
- Three speed settings: Slow, Medium, Fast
- Pause and resume functionality
- Game over screen with play again and main menu options

Installation

Steps

1.Make sure you have Python installed on your machine. You can download it from python.org.

2.Clone this repository or download the zip file and extract it.

3.Navigate to the project directory.

Usage

1.controls

-Player 1 (Arrow keys)

- Up: Up Arrow
- Down: Down Arrow
- Left: Left Arrow
- Right: Right Arrow

-Player 2 (WASD or ZQSD) Default control scheme is WASD.

- Up: W
-

Down: S
- Left: A
- Right: D

2.Playing the Game:

- Use the arrow keys to move the snake..
- Answer trivia questions by moving the snake to the correct answer position.
- The game updates the score for correct answers and ends if the snake collides with itself or the walls.

Configuration

You can customize the game speed and control scheme when starting the game by calling the start_multiplayer_snake_game function with the desired parameters.

- example : start_multiplayer_snake_game('Fast', 'WASD')

Code Explanation

- **init**: Initializes the game window, canvas, controls, and other settings.
- setup_controls: Sets up the control bindings for both players.
- change_pause: Toggles the game's pause state.
- remplir_case: Fills a grid cell with a specified color.
- case_aleatoire: Generates a random grid cell.
- dessine_serpent: Draws a snake on the canvas.
- etre_dans_snake: Checks if a cell is part of a snake.
- fruit_aleatoire: Generates a random position for the fruit.
- dessine_fruit: Draws the fruit on the canvas.
- serpent_mort: Checks if a snake has collided with itself or the boundaries.
- mise_a_jour_score: Updates the score display.
- mise_a_jour_snake: Updates the position of the snakes.
- reinitialiser_jeu: Resets the game state.
- tache: Main game loop, handling the game state updates and rendering.
- show_result: Displays the game over screen with options to play again or return to the main menu.

Drake The Snake (trivia mode)

This project is a trivia mode for the "Drake The Snake" game, combining the classic Snake gameplay with trivia questions. The game is developed using Tkinter for the GUI and Pygame for sound effects.

Table of Contents

Installation

Steps

1. Clone the repository:

```
git clone https://github.com/yourusername/drake-the-snake-trivia-mode.git
 cd drake-the-snake-trivia-mode
```

2. Install required modules:

```
pip install pygame
```

3. Ensure the following files are in place:

- questions.json : A JSON file containing the trivia questions and answers.
- correct.mp3 : A sound file played when the correct answer is chosen
- wrong.mp3 : A sound file played when an incorrect answer is chosen.

Usage

1.Run the main menu script:

```
`python trivia_mode.py`
```

2.Playing the Game:

- Use the arrow keys to move the snake..
- Answer trivia questions by moving the snake to the correct answer position.
- The game updates the score for correct answers and ends if the snake collides with itself or the walls.

Project Structure

- trivia_mode.py: The main script to launch the trivia mode of the game.
- questions.json: JSON file containing the trivia questions and answers.

- correct.mp3: Sound file played when the correct answer is chosen.
- wrong.mp3: Sound file played when an incorrect answer is chosen.

Code Explanation

-charger_questions(): Loads trivia questions from the questions.json file.

-remplir_case(x, y): Fills a grid cell at the given coordinates.

-case_aleatoire(): Returns random coordinates for a grid cell.

-dessine_serpent(snake): Draws the snake on the canvas.

-etre_dans_snake(case): Checks if a given cell is part of the snake.

-Arrow Key Handlers Functions to update the direction of the snake based on arrow key presses:

1.left_key(event)

2.right_key(event)

3.up_key(event)

4.down_key(event)

-serpent_mort(NouvelleTete): Checks if the snake has collided with itself or the boundaries.

-mise_a_jour_score(): Updates the score displayed on the screen.

-mise_a_jour_snake(): Updates the snake's position and checks for collisions.

-reinitialiser_jeu(): Resets the game state for a new game.

-afficher_question(): Displays the current trivia question and options on the canvas.

-verifier_reponse(): Checks if the snake's position matches the correct answer and updates the game state accordingly.

-afficher_felicitations(): Displays a congratulatory message when all questions are answered correctly.

-tache(): The main game loop that updates the game state and GUI.