**Ping-Pong Game Documentation**

**Introduction**

This Python script implements a simple Ping-Pong game using the Turtle module in Python. It allows a player to play against the computer with varying difficulty levels.

**Features**

1. **Player Names and Difficulty Level**: Asks for player names and difficulty level at the start of the game.
2. **Power-up Mechanism**: Occasionally introduces a power-up ball that increases the speed of the game.
3. **Scorekeeping**: Keeps track of player scores and displays them on the screen.
4. **Sound Effects**: Plays sound effects for paddle-ball collisions, scoring, and game over events.
5. **AI Opponent**: Implements an AI-controlled opponent with adjustable difficulty levels.
6. **Keyboard Controls**: Allows players to control their paddles using arrow keys.
7. **Game Over Handling**: Handles game over scenarios and prompts for starting a new game.
8. **Player Data Storage**: Stores player data such as name and difficulty level in a CSV file.

**Code Structure**

* **Initialization**: Imports necessary libraries and initializes the game window.
* **Player Setup**: Prompts for player names and difficulty level.
* **Game Objects Initialization**: Creates paddles, ball, score display, and winner display objects.

**Game Functions**:

* + start\_game(): Manages game logic including ball movement, collision detection, power-up appearances, and game over conditions.
  + reset\_game(): Resets the game when prompted by the user.
  + powerup\_timer(): Manages the appearance and disappearance of power-up balls.
  + game\_over(): Handles game over scenarios and prompts for starting a new game.
  + play\_sound\_async(): Asynchronously plays sound effects using the Pygame library.
  + difficulty\_level(): Prompts for and sets the game difficulty level.
  + new\_player(): Prompts for creating a new player profile.
  + write\_player\_data(): Writes player data to a CSV file.
  + read\_player\_data(): Reads player data from a CSV file.
  + new\_game(): Starts a new game based on user input.
* **Keyboard Bindings**: Binds keyboard keys to game functions.
* **Main Loop**: Listens for events and updates the game window accordingly.

**Usage**

1. Run the script.
2. Enter player names and select a difficulty level.
3. Use arrow keys to control the paddle.
4. Press 's' to start the game.
5. Avoid missing the ball and try to score against the opponent.
6. When the game is over, choose to start a new game or exit.

**Additional Notes**

* Sound files should be placed in a directory named "sounds" relative to the script's location.
* Player data is stored in a CSV file named "player\_data.csv" located in the user's "Downloads" folder.

**Roadblocks**