Endless Runner Game

▼ Description :

Your project will be an endless runner game like subway or temple run , where the player will run in one direction switching between lanes with obstacles in the opposite way.

▼ Features should be included :

- Obstacles in the opposite way (static and dynamic ones)
- Collectable things “e.g. coins”
- Collectable PowerUps
- GUI include score and count of the collectable objects
- SoundEffects and Background Music
- Animation for your player
- Basic Main Menu
- Choose your name
- Extra Bonus : Save your name and your highest score of all time

▼ Tasks :

Task 1: Basic Game Structure and Mechanics

1. **Game Environment Setup:**
    - Create a simple 3D environment with a road and lanes.
    - Implement player character movement left and right within the lanes.
2. **Obstacle Generation:**
    - Introduce static obstacles on the road.
    - Implement the generation of dynamic obstacles (moving obstacles).
3. **Collectibles:**
    - Add collectible coins along the lanes.
    - Implement the collection mechanism.

4. **Scoring System:**
    - Set up a scoring system based on distance covered or number of collected items.
    - Display the score on the GUI.
5. **Basic GUI:**
    - Create a simple GUI displaying the player's score and collected items count.
    - Include buttons for starting, pausing, and restarting the game.

Task 2: Enhancements and Interactivity

1. **Power-Ups:**
    - Introduce power-ups (speed boost, invincibility, etc.).
    - Implement the collection and activation of power-ups.
2. **Player Animation:**
    - Add animations for the player character (running, jumping, sliding).
    - Include smooth transitions between animations.
3. **Sound Effects and Background Music:**
    - Integrate sound effects for actions like jumping, collecting items, and hitting obstacles.
    - Include a background music track for an immersive experience.
4. **Main Menu:**
    - Design a basic main menu with options for starting the game, accessing settings, and viewing high scores.
    - Implement functionality to navigate between menu screens.
5. **Player Name Input:**
    - Allow players to input their name before starting the game.
    - Save the entered name for future sessions.

Task 3: Persistence and Polish

1. **High Score System:**
    - Implement a system to track and display the player's highest score of all time.
    - Save the high score data persistently (e.g., in a file or database).
2. **Save/Load System:**
    - Create a system to save and load the game state, including the player's name and current score.
    - Implement options to continue the game from the last session.
3. **Bug Fixing and Testing:**
    - Conduct thorough testing to identify and fix bugs.
    - Optimize performance for a smooth gaming experience.
4. **Final Polish:**
    - Add polish to the game by refining graphics, adjusting difficulty levels, and improving overall user experience.
    - Fine-tune parameters such as obstacle spawn rates, power-up effects, and scoring mechanics.

By breaking down the project into these three tasks, you can focus on specific aspects of the game development process, ensuring a systematic and efficient development workflow.