Skateboarder Movement & Scoring System

System Overview

This system simulates skateboarding physics and rewards players for passing obstacles with style in a timed session. The movement system handles acceleration, braking, leaning, pushing, and jumping while ensuring smooth, physics-based motion. The scoring system dynamically calculates points based on airtime, style (spins), and obstacle clearance to create a skill-based gameplay experience.

Thought Process

The development process followed a structured approach:

- Set up the character skeleton and animations to provide a foundation.
- Developed custom skateboarding mechanics instead of relying on built-in engine physics.
- Simulated friction using Tick events for realistic speed adjustments.
- Encapsulated all movement logic into a modular component for better reusability.
- Implemented the scoring system to reward airtime-based tricks.
- Added a level prototype to test mechanics in a gameplay environment.
- Enhanced the scoring system to factor in obstacles and style for deeper gameplay.
- Developed the UI system with gamepad compatibility (though final setup is pending).
- Added a pause menu to improve usability.
- Introduced a timed session through the GameMode to create a clear challenge and prevent the game from feeling aimless.

Personal Assessment

I am satisfied with the system design, modularity, and performance optimizations. The physics and controls could be further refined, but improving them would require extensive playtesting and feedback. Given more time, I would convert movement settings into data assets to allow for character customization, where each character has unique movement properties. This would make the game more dynamic and engaging.

Time Spent Per Task

- **Character Setup and Animations** – 1 hour
- **Movement System Implementation** – 6 hours
- **Scoring System (Airtime & Spins)** – 8 hours
- **Refactoring & Optimization** – 1 hour
- **UI and level prototype** – 1 hour
- **Testing & Debugging** – 2 hours
- **Post Testing Refactoring & Optimization** – 1 hour
- **Documentation, Packaging and Testing the Packed exe** – 1 hour

Total Time Invested

21 hours