Skateboarding Simulator

Code Organization

To facilitate collaboration with other teams, the project was divided mainly into art and gameplay folders. For customization features, all classes were implemented in code and templated with blueprints for enhancement.

Player Controller

The player controller was implemented using Unreal Engine Enhanced Input System. Two actions were implemented: one for general movement and another for jumping.

Character

This prototype uses the Unreal Engine character to simulate skateboarding behavior, with modifications to the following parameters:

- Brake Parameter
- Max Velocity

A rotation factor was implemented for maximum and minimum velocity, meaning the character rotates faster when stopped compared to when at full speed. Reattachment delays were also added to avoid the character continuously attaching to obstacles, which would look unnatural.

Gameplay

Obstacles were implemented using splines, allowing customization of their trajectories. This method attaches the character to the spline and continuously moves them from the start to the end of the spline, simulating a skateboard trick.

UI

Implemented the MVC design pattern for the player UI, now displaying the character's current score and velocity, along with the total time at the top.

Animations

A State Machine was implemented in the animation blueprint to handle movements, initial jump animations, and falling animations. Callback events were also added from the code to manage the jump state.

Point System

Points system was developed thinking on reusability and scalability, its ready to add more events and handle then by multipliers if needed in case to flexibility

GitHub Organization

All main features have been implemented on individual branches, simulating a collaborative workflow. However, it will be better to organize them by tasks into separate issues to improve workflow.

Pending Improvements

- Implementing FX, such as sounds for the skateboard hitting obstacles and touching the ground.
- Improving the UI, which is currently very simple and needs animated controls and colors.
- Splines need improving. While they work effectively for predefined routes, there's a need to ensure smooth character movement, especially in cases where freedom of movement is allowed.
- Adding more animations for special tricks.

Wishlist

- Converting the player controller input system to the command pattern.
- Implementing AI behaviors.

General Track	
Task	Time To Complete
Initialize project	2:45
Implemented initial player controller features	3:00
Organized project and improved player controller features	4:30
Minor fixes	0:15

Asset search and test for levels Added level assets and implemented on BPs Implemented initial UI features	2:00 2:30 3:30
Added character animations and implemented state machines Gameplay improvements Added remaining splines to level	3:00 4:30 1:00
Total Hours	27:00:00