

Skateboard Unreal Project

For this test, the first thing I did was get the list of required features, and order it by difficulty and importance, so the most important thing was definitely the skateboard physics, followed by the movement on top of it, jumping, and lastly the points system, UI and animations.

Skateboarding Feature

My first approach to this was some testing, first simply applying forces and impulses to the Unreal default third person character controller, I quickly realized that would not produce satisfactory results, mainly because of how a skateboard must react to ramps and slopes, I also did a quick test with the vehicle template Unreal provides, and although a car is more similar to a skateboard than a default capsule pawn, that seemed overkill for the assignment, since it has a bunch of motor, gear and torque functionality that I would not require, so it seemed a better idea to make a custom character movement component.

Custom Skateboard Movement Component

Although, overall, I know how Unreal handles movement, this was my first time overriding the movement component, so it required a more in depth knowledge of the component, so my first job was to learn it, mostly from this [resource](#).

The approach here is to add a new movement mode to the character, in our case, the Skate, and provide a custom physics function that Unreal will use when the character is in this custom mode.

By overriding `PhysCustom()` and calling our internal `PhysSkate()` function when we are in the skate custom movement mode, we take full control of the characters reaction to physics, to solve our main problem, that is sliding along ramps, we cast a line trace from the character down to the floor, and apply a velocity change in the floor normal vector, along with regular gravity, effectively making the character slide on ramps, here we also use the floor normal vector to adjust the character rotation.

Inputs

With the most challenging task finished, I was ready to add movement functionality to the character.

Using Unreal's enhanced input system, I added an accelerate action, that adds an impulse in the forward direction of the character, also made a delay so the player can't spam this. Added a break input, that simply increases friction by a scalar while it's held down, simulating a skater pressing his feet on the ground to decelerate the skateboard. Added a simple jump for the skateboard, that added and impulse on the upwards vector of the character, as the character is being rotate along with the floor normal vector, this ensures that when jumping on a ramp, the impulse is added in the opposite direction of the floor, like a regular skateboard would. Added left and right lean input, that adds a continuous force in the right direction vector of the character. Also added a toggle between regular walking and skateboarding, as my custom skateboard movement component is not perfect, it can get stuck on ledges and steep angles, so changing the movement mode to walking allows the player to get out of this situations. I was also missing a free look camera while skateboarding, so I added a toggle between free look and fixed camera modes.

Animations

With all the inputs ready, I imported a character and a couple animations from [Mixamo](#), I could not find properly made animations for this project, mainly due to root motion and skeleton position on key frames, so the character is not properly aligned with the skate and floor, I am not an animator or rigger, and time was not on my side, so I left this as is.

With my character and animations properly imported and setup, I created my custom animation blueprint, copied the states for regular walking movement mode from Unreal's default animation blueprint, and added switches to handle the skateboarding movement mode, here I also liked the idea of a visual feedback for leaning, so I added a spine bone rotation to the pose.

Score

Added a game instance score subsystem, with basic functionality to store the player's score.

Created a Score Obstacle actor class, that grants player points when overlapped, added a simple cube mesh, with a fresnel material, and made it disable itself for 10s after the player hits it. The trigger only gets activated if the player is on the skateboard movement mode and if it's jumping, so smart positioning of this actor is essential to achieve the desired effect of granting points when jumping over obstacles.

Final Notes

This was definitely a challenge for the time frame, trying to implement a custom skateboard movement mode properly is a task that would require a lot more time for research and testing, so I had to settle for a very simple version, that does not handle edge cases well, otherwise I would be out of time to implement the remaining features, those were very straightforward to add.

I did not use a timer while working on this, but from my work history, I would say it took around 12 hours on the custom skateboard movement mode, and another 8 hours for all other features, input, animations, score and UI.