

# SKATER TASK OVERVIEW

By Juan Fernando Ramírez

## PLANNING

Immediately after I received the task I began a short brainstorming phase, taking the task's requirements as a start and then thinking what could be added in order to enhance the base mechanics of the experience. After getting a rough idea of what I wanted to build I did some research to get concrete references on similar games, getting a better idea of what players expect. In a funny coincidence, a friend sent me an instagram reel showcasing a mod for a similar game, I added it to my reference list.

## FIRST STEPS

Brainstorming session done. Time to get my hands dirty. A couple of hours later I booted up Unreal, created the repo and began setting up the project. I added the Third Person template and took the ThirdPersonCharacter as a base for the project's player. This let me start implementing the core mechanics faster as the basic character setup was already done.

I modified the character's movement to better reflect a skateboard, mainly changing its lateral movement to feel more like steering a vehicle rather than just turning left or right. I added some friction to better control how speed decays over time.

I took some time to thoroughly test and tweak the player's movement so it felt smooth, responsive and interesting.

## OBSTACLES AND SCORING

After getting the player to a satisfying baseline, I started working on the obstacle logic and the scoring system. I really wanted to make it as scalable and modular as possible so I took some extra steps. Even if this project is small it is better to create a strong foundation so it is easier to add to it.

I created a basic obstacle with a box collision that would serve as the scoring check. Then created an Interface so it could communicate with the player without any hard references. After that I created a ScoreManager to attach to the player, in a larger project this component could be used to manage combos, bonuses, penalties, streaks and even more complex stuff.

After having the basic scoring functionality I turned my attention to the UI, creating a HUD class to serve as intermediary between the player (or any other actor for that matter) and the project's widgets, this would also be managed by interface communication.

## ANIMATIONS

I replaced the base third person character with a Mixamo model and imported its respective animations. I tweaked some of them on blender and then created their respective animation montages, using Notifies to make the pump functionality match its animation.

## **LEVEL**

After having a fully functional player I began setting up the level using the Synty Studios Sci-Fi City package. I created child variants of the base obstacle, gave them their distinctive score and base mesh. After importing the assets from Synty I began the process of polishing the level after the grayboxing phase.

## **FINAL ADJUSTMENTS**

After integrating the full experience I went through an exhaustive testing process and tweaked the player's stats (mainly the friction and pump impulse) so the whole level felt fluid.

## **POST-MORTEM**

This project really helped me polish my weak points when it comes to C++ development. Making a full C++ project was an interesting challenge (I usually work with a hybrid C++/Blueprint approach), I must improve my coding speed a little as some parts of the process took me a little more than expected (having an appointment far from my place on saturday also slowed me down a little).

Base character implementation - 5h

Animation setup/Implementation - 3h

Obstacles and Interfaces - 2h

Assets import/level construction - 3h

Testing/Adjusting - 7h

Total - 20h