The game runs with default character movement and input.
The player moves forward automatically, and pressing the w or up keys makes them accelerate, while the s or down keys makes them decelerate.
Space bar makes them jump, and a/d or the mouse make the character turn.
When the character jumps, it starts tracing objects below. Any actor with the “Obstacles” custom object type that the player jumps over will be stored. If the player lands on one of the jumped over objects, it is removed from the array. After the player lands, points are awarded for each of the jumped over objects, plus the time the character spent on the air.
I based my locomotion design based on skate games I’ve played before, like the classic Tony Hawk 1 & 2 for the PS one.
I started with the character model and animations, but after encountering some issues with the animations, I decided to move to the character movement and level design.
After that it was just a matter of continuous development, testing and tuning. I tried some of the features on blueprints before implementing them fully on C++, like the line tracing for objects.
I spent the first couple of hours implementing the character model and animations, including searching for suitable free assets, creating or adjusting the blueprints and animations, trying to adjust the animations and positioning of the character, etc.
After that I spend the rest of the first day designing and iterating character movement and controls.
The second day I spent most of it creating the level with assets and props, a little more than 1 hour implementing the scoring system, and adding a simple UI.
Character animation was not completed; character movement could be improved; and the UI is very basic; all of these things are the biggest flaws of my test, but it is a simple game that could be expanded and refined easily.