This is my first foray in over a year into Unreal Engine, as such a lot of concepts I had were getting a bit rusty.
Fortunately most of the physics involved in the skateboarding system were able to be produced by using Unreal Engine's native Walking Movement system, with some severe tweaking of the drag and friction to allow for the "slippery" feeling of wheels, and using an impulse towards the Actor's forward vector it was very achievable.
Skateboards don't accelerate constantly, they have a periodic push, as such I implemented a small cooldown between each impulse to replicate this, later on I synchronized it with the Actor's animation, to make it feel better.
For turning the character using this approach though it was a different beast that i hadn't considered at first, being used to Unity's rigid bodies i hoped to use a torque approach to apply rotational force to the movement component, however this was not an option. As such i had to make a small hack that ended up working out, i forced the movement component's velocity to change towards the Actor's forward vector by using simple vector math, getting its magnitude, substituting the velocity vector by the forward vector of the Actor and then multiplying it by the saved magnitude, of course this means that the gravity constant was lost in the process, however this ended up working well because there is no air strafing, so you cannot turn while being on the air and as such no issues with the gravity presented.
Jumping was implemented using the movement component's basic behaviour, as this works really well with its internal force management.
I decided to implement animation management on blueprints and expose a few variables from C++ as this seemed like the cleanest approach. In the same way, obstacles were implemented using very simple blueprint routines.
I am not used to set dressing nor 3D level designing, so my efforts in this consumed a lot of my time trying several approaches. In the end I went for a small patch of painted landscape with some placed foliage, a circuit that encompassed the patch and obstacles placed around it. I designed 3 types of obstacles, a small bench, a wood block and a bigger two stacked wood block. Each has different point values, i tried placing them around the circuit so they all can be jumped without stopping.
Due to time constraints i was unable to add sound effects, a start or loading screen or a more elaborate HUD, this was mostly due to my derusting with the engine, as well as several technical issues i had while getting it to run (I had to switch from VS2022 to Rider due to compilation issues, and several minor package issues that consumed a lot of my time).
In the end my time was spent as such:

- Technical issues 5 hrs.
- Acceleration/Deceleration movement implementation 3 hrs.
- Turning system 4 hrs.
- Animation cutting and implementation 2hrs.
- HUD implementation 1hr.
- Obstacles creation 1hr.
- Level creation 6hrs.
- Game Input Setup 1hr.
- Value fine tuning 1hr.
- Project cleanup and compiling 3hrs.

- Testing 3hrs