Systems and Thought Process

The development was carried out iteratively, prioritizing core functionality and responsive player control before refining visual or structural aspects. Starting with the movement system in C++ was a deliberate choice to ensure full control over physics, collision, and the dynamic behavior of the skateboard, establishing a solid foundation for future features.

For animations, I chose to integrate them in a way that respects their actual execution timing—such as delaying the push impulse until after the pushing animation starts—using timers and montages to sync logic and visuals. For jumping, I introduced vector curves to control the skateboard's rotation while in the air, adding dynamism without compromising structural clarity.

1. Movement Functionality (**A - D**)
2. Small Designed Level with Jumping obstacles
3. Jumping- Utilize any Jump animation from Mixamo. (**Space**)
4. Pushing Functionality - Speed up (**W**)
5. Slow Down Functionality (**S**)
6. Points system - for every obstacle jumped over

In total, only 8.5 hours of work were invested, divided into two overnight sessions: from 11:30 PM to 5:00 AM on the first night, and from 10:00 PM to 5:00 AM on the second.

My Performance

Finally, I'd like to apologize for not being able to complete the test. Despite the focused effort during the nights, I am currently working full-time, and unfortunately, receiving the test on Monday significantly affected my ability to deliver everything within the expected timeframe—more than I initially anticipated. I'm confident that in a different context, I could perform at a much higher level. Thank you very much for the opportunity, and I hope to return in the future and show the full extent of what I'm capable of.