

Physics-Based Skateboard System – Development Report

System Overview

The project involved developing a physics-based skateboard system from scratch using Unreal Engine. The core mechanics included movement, acceleration, braking, and a suspension system simulated with line traces and applied forces. The game also featured a point system based on obstacle traversal, a UI for score tracking, and a simple level setup.

Thought Process & Challenges Faced

Initially, I considered using Unreal Engine's **Gameplay Ability System (GAS)** to handle input and future-proof the system for multiplayer expansion. However, I encountered issues with input handling, which consumed my first day entirely. The time spent resolving GAS-related problems significantly impacted my ability to develop core mechanics early on.

On the second day, I pivoted to a more traditional character-based approach, but the movement felt unresponsive. I briefly considered using the **Chaos Vehicle Movement Component**, but due to time constraints, I opted to build a **custom physics-based movement system** from scratch using a Pawn class. This system relied on **line traces and force-based suspension**, which took up several hours as I debugged multiple physics-related issues.

Some of the most difficult bugs included:

- **Initial bouncing on game start**, which persisted despite attempts to modify damping and reset physics simulations.
- **Car flipping unexpectedly**, an issue I lacked time to fully debug.
- **Inconsistent point rewards**, where the dot product comparison sometimes failed to register a valid traversal.

By the final hours, I managed to implement **basic UI, braking, and a scoring system**, though many features remained unpolished.

Personal Assessment

This is not my best work, and I know I can deliver much stronger results when working with systems I'm more familiar with. I can definitely prove my coding skills, but the physics-based movement introduced unexpected challenges that affected the final result. The biggest setback was the time spent integrating GAS, which will be useful for scalability, but took up crucial development hours that could have been used to refine core mechanics earlier.

Physics-based movement was the most challenging aspect of this task, requiring extensive debugging and research. While I was able to build a functional system and implement key mechanics, the physics interactions didn't always behave as expected, leading to additional time spent troubleshooting. Despite these obstacles, I pushed through and delivered a semi-working prototype.

I know I can do much better, and given more time or a chance to iterate, I would refine the physics further to match my usual coding standards.

Total Hours Spent: ~20-25 hours