MicroBuild 3 - Arkham Knight Batmobile Clone

Hypothesis

Can I reproduce similar behavior to the real Batmobile made by Rocksteady using similar code to my previous projects?
This is a call back to my affection for the Arkham games, especially Arkham Knight. The batmobile is the Dark Knight's vehicle in which he goes to war with the Militia. My goal is to attempt to replicate the core behaviors of the car, being changing modes, moving in those modes, and the cannon.

Proposed Process

Use my previous controller script in tandem with a car controller.
The previous script will be altered so the vehicle rotates more smoothly using lerp functions, and to include the directional dodges.
The new script will explore the Rigidbody class more by utilizing AddForceAtPosition for more accurate physics - the batmobile feels very heavy because of its primarily rear wheel power output and being highly prone to slide around corners, thus power must focused more to the rear of the vehicle and braking more in the front.
For now this is going to be Keyboard and mouse but will hopefully be able to be used with a controller. I will be testing with a PS4 controller.

Actual Process

With my choice to use the new input system package, the plan to use old code fell apart quickly since the inputs differ completely in how they are to be implemented. This meant rebuilding the main movement script.
From there I tried to implement both movement modes in a single script but that caused many issues. Thus I had to make multiple scripts so that the components didn't interfere with one another. Thus a script for switching between the modes and scripts for both modes was required.
I also chose to implement wheel colliders for the Pursuit mode movement which made even more issues due to the less than forthcoming information about how the collider components affect the colliders themselves and thus the force output.

Once the movement system was working, the focus shifted to cameras and how they should be behaving. For this I thought Cinemachine would be the obvious choice and then have it follow a separate anchor that is a child of the main character transform. This was good until I needed to alter the rotation of the mesh manually since changing rotation of the mesh would sometimes leave the wheels behind due to the wheel colliders and some other factors. However the cameras and the mesh combined proved to be much more hassle than I believe they could have been worth.

However using code from a previous group assignment I was able to get the cannon to work, however, the particle effects are a bit too much simply because the spheres used as the particle spikes the number of polygons into the range of 450k which didn't make noticeable stutter but the scene is still extremely minimal. I will replace these spheres with lower poly meshes in the future.

Future Improvements

The controller needs multiple input maps to improve how the controls can be enabled, disabled and generally kept separate.
There need to be some improvements to the UI and the Camera design.
Pursuit Mode needs to bring the car's velocity to an exact zero, this may be a side effect of the wheel collider suspension values, or even the damping values not being high enough.
Improved mesh rotation during battle mode so that the mesh is not directly locked to camera rotation.

Conclusion

This project has all the kinks and quirks you'd expect from a first level prototype, but that being said this is the farthest I have pushed myself to building something technically complex and building a large scale system.
For the exam I want to expand this project to place it in some kind of context since it currently exists in a vacuum without any real direction. I am happy with the project where it stands and how it has pushed me to expand my knowledge and skill set to this point.