05/09
Building basic framework

As said, I started the process of building the project. I want to design the system in a way that allows the integration of controller input - the ability to use a controller over a keyboard and mouse.

After some research I learnt of the Unity Input System package and how it could streamline the process of keybinds between MK and controllers. However there are always drawbacks, in this case I had to basically abandon my old character controller since the code uses the Unity axis system and not the Input System.

The input system appears to be a more technical way of handling inputs, very little tutorials cover the system in entirety since it's easier to teach people the simple way of using axis that have some prowess and learning but it is still just Input.GetKey(x) where x is the random key on the keyboard that you choose, this makes it difficult to map controller layouts since each layout needs its own statement in code. With the input system, keybinds are added first, then callbacks are made in code to the binding - the binding being the action, the keybind being the button pressed to activate the action.

The skill with the new input system is knowing the type of information you receive from the keybind and then how to use that information to manipulate the character to make them move or perform as expected from the action bound. This makes the task of controls a simple thing of remembering to bind all tasks for all supported inputs, and only have to reference the action once from code.

With some tweaking and testing, the basic prototype to this point works with a PS4 controller still with the need of a binding to change the player from Pursuit to Battle Mode as in the Arkham Game.

To this point it has simply been understanding the input system, but from here it is to be trying to implement the Batmobile's functionality - to start I will make Pursuit Mode. In pursuit the Batmobile is a slightly overweight car with very impressive speeds.

With a little math I have deduced it takes about 120 thousand Newtons of force to Accelerate the Bat-tank to 100km/h in the time it reportedly can - being 2.7s, an impressive number for any car, hyper or not and certainly impressive for a 3 ton armored car.

Basic principles from the previous controller remain, and thus the structure can be brought across to manipulate the input in a similar way.

15/05
Switching to WheelColliders And Building Pursuit
Pursuit mode - the standard driving mode, was not panning out as I intended, after some searching I found that wheel colliders are the way to go when making vehicles, a little obvious. I went into this process thinking it would be easy, since this is the preferred method of propulsion.

Just like the Input System, wheel colliders are not widely used by most developers due to their various difficulties. And so the documentation is quite sparse in comparison to something like a rigidbody.

In making this switch I took the opportunity to update the batmobiles model to show more of the inspiration from the 2008 film *Batman Begins* affectionately known as the Tumbler.

This new model includes wheel arms so the model does not appear as ‘Phallic’ to quote Tim, individual wheels, and now armature since animations will not be baked into the model.

Implementing Wheel Colliders is quite a technical process, the hierarchy of the object on which the colliders are being added must not include the colliders on the visual wheels but on their own empty transforms. This is so working with the colliders is as clean as possible for the developer due to the Gizmo potentially being inside the object model. Next is making sure the force being applied by the wheels is not too far beneath the vehicle's center of mass, else the car props itself on its back wheels.

The problem of wheelies persisted for a few hours until I found it had something to do with the center of mass - I did not know the center of mass was editable. Then it was a matter of meeting the speed of 200km/h according to a wiki article on Ben Affleck’s Batmobile from Batman V Superman.
With Center of mass under control I tinkered with the friction values, until the car moved to the desired speed of more or less 60m/s.

With the addition of wheel colliders I now need to rework Battle Mode to use this system. A fun and potentially very frustrating process.