Player Controller Plan

Aim :

* To build an all round controller for third person games which can traverse over complicated terrains and be extendable enough to be used for custom purposes such as ledge climbing.
* Clean and efficient code that can be reused when needed and variables can be edited so the motor is what will be moved while the controller and other scripts can control the movement.
* The ability to make this an all round controller for the audience who will want to learn about how this type of mechanisms work.

Expected Base Features :

* Simple Movement
* Slope Handling
* State Machine
* Collision + Multi Edge Detection
* Artificial Damping *based on surface normal*
* Climbing Stairs
* Terrain Movement and Proper Lerping
  + Sticking to ground while gliding over fractures
* Smooth Frame Independent Movement

Expected Complex Features :

* Slope Slipping
  + Realistic (Not Instant Slip)
* Jump Holding power
* Crouching and Sprinting
* Sliding and Collider Size Changing for going under logs etc…
* Strafing
* Ledge Climbing (Proper Prediction System based on Input)

There are two aspects of the controller :

* Player Controller
* Player Motor

**The controller** is what will actually receive input and determines how that input is used, the controller is what will have the actually state machine inside it always calculating the current state of the player based on the environment and input. Then the inputs and movement will be conducted differently.

**Player Motor** will be the system that takes the input and just does what the input tells it to do, it needs to be very flexible with many exposed variables and functions for any use case. So that even non-player controller scripts can change the variables if need be.

Different States:

These are the base movement states :

* Crouching
* Strafing
* Jumping
* Landing
* Sprinting
* Walking
* Still
* Falling

These are the base control states :

* Aiming
* Shooting
* Attacking
* Climbing
* Sliding

Motor:

* Needs to be frame independent so that some devices don’t run the player faster or slower than others.
* *Needs a controlling script at that point, because a ladder can have it’s own script that controls the player when it is on that ladder. So the motor will always have a controlling script that can take control from the player controller at any point – when this happens any input given from player controller script will be ignored until that controlling script is removed…*

Different Controllers –

**Base Movement Controller**

**Ladder Movement Controller**

**Ledge Controller**

How will the controller be built from scratch?

Will there be use of any assets?

Integration of animations?