Part 1: 3D Character Controller

Brief – 07/30/2021

# Overview

I have programmed 3D character controllers before (like in *Legacy of Embers* and some other unshipped projects) but have never created one that feels like a controller worthy of a first-person style game. Therefore, the first step in this project will be to create the movement portion of a 3D character controller with a first-person camera. I will be using a sandbox project to create new features before integrating them into the finalized GitHub repository of the project.

# Goals

* Create movement for a 3D character controller that obeys a NavMesh
* Use the Command Pattern to eliminate player controller’s dependence on player object
* Use Cinemachine to control the first-person camera
* Use the New Unity Input System to allow seamless transition between mouse & keyboard and gamepad input
* Allow for rebinding of movement controls

# Research

## 3D Character Controller Movement

I began by watching a quick video by Ciro Continisio on YouTube titled “Using NavMesh to create a character controller in Unity | Mini-tutorial.” Here, Ciro uses a NavMeshHit object every frame to determine if the player’s desired position is a valid one on a NavMesh. If it is, he “teleports” the player to that position by changing its transform’s position directly. This method seems to work nicely for quick and responsive movement, but I think I want my character controller to have some sort of acceleration component. I could amend this script to use Vector3.Lerp to smooth the movement and have some control over acceleration.

Next, I remembered in my very early days of game development that a Udemy class I took taught new programmers to make a first-person 3D game which might help with this character controller. Here the project used Unity’s Character Controller component for movement, which is another possible route to take for this controller. This technique is also used in Comp-3 Interactive’s comprehensive tutorials on a full First-Person Controller. I can also refer to these tutorials for head-bobbing, jumping, and crouching if I find myself stuck later.

Finally, it seems there are some very useful free assets on the Unity Asset Store such as “First-Person All in One” that handle this all for me, but the goal of this project is to code things myself. Therefore, I will not use these assets but perhaps reference their code later if I need some guidance.

## Command Pattern

In many unshipped projects I have used the command pattern extensively and am confident that I can implement it here without research. This could always change based on new problems that arise during development.

## Cinemachine

I am relatively new to Cinemachine, but I have used it in the 3D capacity for *Legacy of Embers* and the 2D capacity on an unshipped project. I am confident I can utilize Cinemachine in this project without much help. I think the only issue I may have is head-bobbing, and it seems like I can utilize the material covered in Code Monkey’s “How to do Camera Shake with Cinemachine!” to achieve this effect should I choose to add it.

## New Unity Input System

I am very comfortable with the New Unity Input System. In some of my unshipped projects I have used it in conjunction with the Command Pattern and they fit together seamlessly. Though if I find myself stuck, there are some great YouTube channels like samyam that cover this system in depth.

## Rebinding of Controls

samyam has a comprehensive tutorial on how to rebind controls using the New Unity Input System, however it is done through some additional, experimental scripts that come along with the system. Sine the system is relatively new and likely to be changed, I do not want to rely on it for rebinding. I know the command pattern is useful for rebinding, so I would like to instead focus on that instead of using pre-written code.

There is a wonderful online literature resource called “Game Programming Patterns” that covers the Command Pattern and specifically outlines how to implement it to rebind controls. I think I will be able to use this as a reference when creating the rebinding functionality.