Week 11 Journal of frustration

My code was not moving correctly, and I still have no idea why, but out of frustration I made a new unity 2D project and reimported the package from slate and re-did all of week 10’s tasks to get the code to respond properly. Here is the new repository link: <https://github.com/Xiomaraze/Assignment2_Platformer>

Task 1: maximum jump height and speed

This started off pretty easy, taking the given height and applying it to get the proper position for the maximum jump, and dividing by time to get the speed, but the hard part was getting the program to understand when a jump actually happened, to record the maximum jump position for that instance, vs when it was already in the air and just needed to wait for it to reach the already given maximum, or land. I ended up exploring into the raycast stuff more thoroughly, and I must admit I am a little disappointed in the unity api manual for not showing examples of how the raycasting works. I never realized that when I used the rigidbody2D.cast it used it directly from the attached collider’s EDGE instead of center. And when I was trying to understand how the method used the distance variable to denote how far the ray could actually go, I never actually understood how to measure that. In the end I had to use the raycasthit2d.distance to check if the distance was under 0.01 to verify that yes, it was because the object was touching on the side I was looking at (in this case the bottom). Also unity has issues if you forget to give it a jump height and jump time. It does not enjoy the concept of infinity.

A pixelated video game

Description automatically generated

The image left shows the edge of the boxcollider on the player in green. The red arrow shows where the raycast in the code below moves between. From the green box the raycast is sent downwards.

A screen shot of a computer code

Description automatically generated

Task 2: Terminal Speed

Alright in theory this one is pretty simple, just make sure your acceleration or addforce is a 0 in y if it’s moving faster down than the terminal speed, and make sure the velocity on the rigidbody is limited by vector2.clampmagnitude. However, I have no idea how to test it. Figured it out, just let him fall from above the scene view and monitored the velocity in the inspector.