**Crouch**

1/22/2024

Implementing crouch is not hard, as it is inherently a part of the character movement component. I’m attempting to adjust the “height” of the crouch, allowing the player to hide underneath surfaces that they could reasonably fit under using a trigger volume around the surface and an overlap check to see if the player is able to crouch lower in the area. If both the player is crouch and able to crouch lower, then the player successfully crouches lower. If either of those Booleans are false, nothing happens.

The latter part of that interaction is a success, however if the intended crouch functionality is successfully fully executed, the character movement suddenly seems to lose all friction in the environment while crouched. Un-crouching fixes this effect until the implementation is activated again. In the meantime, whenever the player attempts to go underneath an object, they continue sliding in the direction of their movement, and lose full mobility control. I don’t even know where to begin debugging this, but I did record a [short video](https://youtu.be/liGoGdVIVcc) of the issue and posted it into my programming text channel in my server.

1/23/2024

Brian recommended adjusting the radius of the capsule, since a capsule can only shrink to be as small as a sphere. Outside of that, he also thinks that the collision is just pushing the player out because the player shouldn’t fit. Maybe take away the geometry in the scene just so I can test the trigger volume?

1/24/2024

Brian’s suggestions worked. The crouch height now automatically adjusts depending on whether the player is trying to hide underneath something or not. While hiding underneath an object, the player retains full control over their movement, and the collision does not attempt to force the player out from hiding. This was the cause of the “ice mechanics.”

Now, there are still a couple of issues left. When the player un-crouches while they are hiding underneath something, upon leaving the hiding area, their height is set to the regular crouch height, even though they are no longer crouching. This issue resolves once the player crouches + un-crouches again (no need to activate the lowered crouch). The other issue is that when the player is approaching the trigger to lower the crouch (while crouched) and stops at the edge of the trigger volume, they’re camera will jump between the lower crouch height and regular crouch height.

I have a feeling that the latter issue will be solveable by adjusting the timing on the retriggerable delay nodes on the On Component Begin and On Component End Overlap events. This adjustment could either be a longer time, or replacing the retriggerable delay node with a standard delay node. *The delay needs to be there regardless or the program will crash due to a call stack infinite loop.* I would also like to add interpolation between the values to make the transition smoother, which could also fix the issue.

Interestingly enough, this “glitch” is present in Dishonored 1, though not in Dishonored 2.

The former issue could be resolved through adjusting my Boolean branch statements, possibly. To ensure that the functionality is called at the appropriate time, I have to bools, one bIsCrouched? Bool casted from the FP\_Character blueprint and another bIsOverlapped? Bool inherent in the BP\_LowerCrouch blueprint. If both are true, the player crouches lower. I need to add an exception to allow for the bIsCrouched? Bool to be false while still allowing the player to crouch lower while they are already hiding underneath an object without the capsule collision causing the environment to turn into an ice rink again.