2/12/2024

Day two of working on possession resulted in my being able to possess a pawn by pressing an input key, activating a line trace, storing the overlaps in an array, checking to see if the overlapped array items contained a pawn with the tag “possessable,” and finally possessing the pawn.

I can’t seem to figure out how to make the possessed pawn receive input, unfortunately. I think it might be because my “pawn” is a default cube with the proper tag. Next time, I want to try to create a new (human) AI blueprint from scratch, give it the proper tag, and allow the player to possess it. Maybe in that instance it’ll fix the movement thing? From there, it’ll be pretty simple to add in possessable rats, bugs, fish, dogs, and all other creatures Corvo can possess in-game.

Additionally, once the actual possession works, I’m going to add in a delay timer to automatically deactivate the ability, or an option to end the ability early. I also need to make sure that the player is undetectable by other pawns while possession is active. Anything else that I should be adding in this section will either come to me later or next time.

2/14/2024

Possession now allows for movement if the possessed pawn has a movement component. I’m having issues with the player un possessing the target, but not in the way you might think.

First off, I need the possessed actor to spawn slightly in front of the player when the player deactivates possession. Possession currently just replaces the possessed target, destroying it in the process. Getting a copy of the hit actor in the line trace doesn’t seem to be working very well, either.

Then, I need to make sure the player “repossesses” their standard body when they deactivate the ability. I’m worried that if the first part gets fixed, the player will just get softlocked.

This doesn’t actually apply to the Possessable tag, either. I want it to do that.

2/21/2024

This is frustrating. I have no idea what I'm doing. I want to figure out a way to store the possession target's skeletal mesh asset and apply it to the player while possession is active (so the player is "disguised"). If I can't figure that out, I would say it didn't matter except that I need a way to store the possessed target anyways so I can respawn it once possession concludes. The disguise part is optional, but spawning back in the possessed target is not.

So in short, today was just messing with collision, line trace hit results, and skeletal mesh assets and getting nowhere important in particular. Going to try and talk with Brian or Gabe about this.

On the bright side, I have ideas for clamber.

2/22/2024

While I was at work, I decided to nix the ideas I had related to skeletal meshes entirely. Instead, I’m using the copy reference to store the hit actor, passing that through to the end of the function, and calling the Spawn actor of class function when Un-possess() is called. And. It. Worked. Kind of. Similarly to darkvision, the initial activation/deactivation seems to be bugged where the ability doesn’t behave as normal. Subsequent activations behave as expected.

To prevent the stored hit actor from spawning inside of the player, while also making the possession mechanism make some amount of visual sense, the rotation of the hit actor matches the player’s, but the location of the hit actor is the player’s current location plus the player’s forward vector. This allows the hit actor to spawn in front of the player as if nothing had happened.

However the issues are now coming from the initial deactivation, as discussed previously, as well as with the timer mechanic I currently have in place. The hit actor will spawn if possession is true after 10 seconds, but possession does not have to be true for all of those seconds for the actor to spawn. This results in the actor spawning in front of the player at unexpected times, and this is something I’m currently working to address. (There may be more to update since I plan to continue working today, but I’m getting these thoughts down now since UE just crashed.)

Managed to figure out how to remove the actor data, not that I think it really did much. Now, the issue I’m running into hinges around my ability to figure out why my bIsPossessing? Boolean is never set to true. From there, it should be easier to find out where the sources of the other issues are.

The timer works from what I can tell (but it’s difficult to test without that Boolean). I’ve condensed it into a macro for the time being.