GAME250: Technical Game Development (Spring 2023)

Lab 07

Objectives

By the end of this lab, you'll be able to:

- Create your own functions
- Use time delays

Background

Let's create a basic combat and movement system for a third person game! We'll be giving our character dash forward and attack. The enemy will get health, armor, and a dodge ability.

In-Lab Instructions

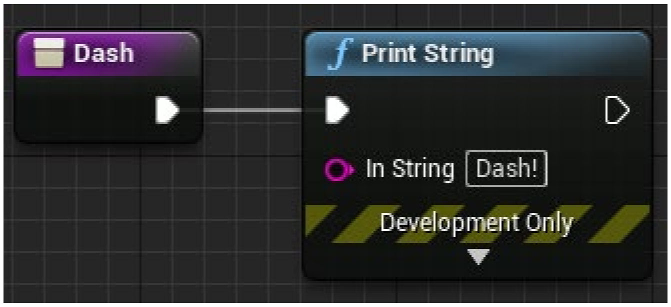
Create a new Third Person game with Starter Content.

Enemy

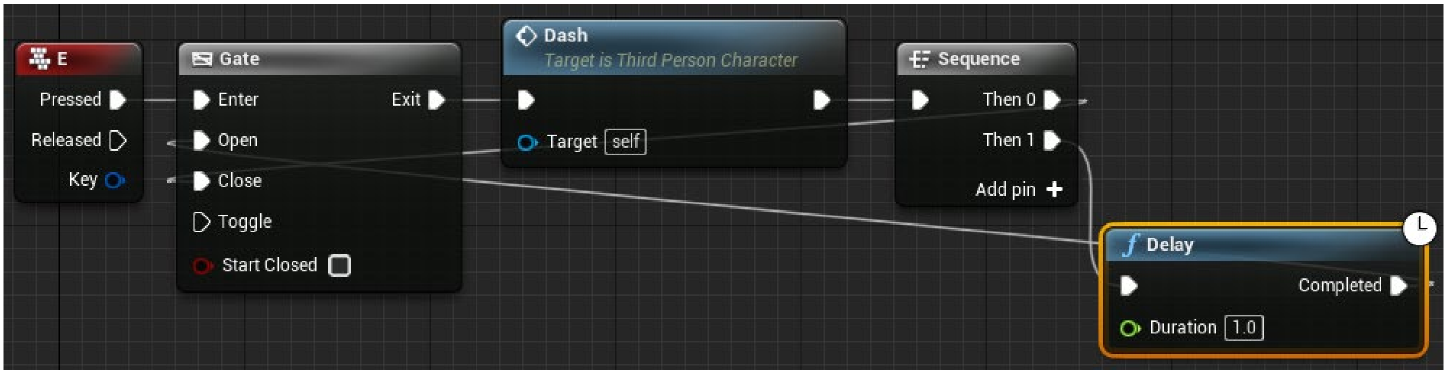
For now, we just need a copy of the ThirdPersonCharacter blueprint. Find it in the Content Browser and duplicate it and name it Enemy.

Dash Function: Cooldown

In the Third Person Character Blueprint, create a new function named Dash. For now, simply add a Print String to indicate that a dash has occurred.



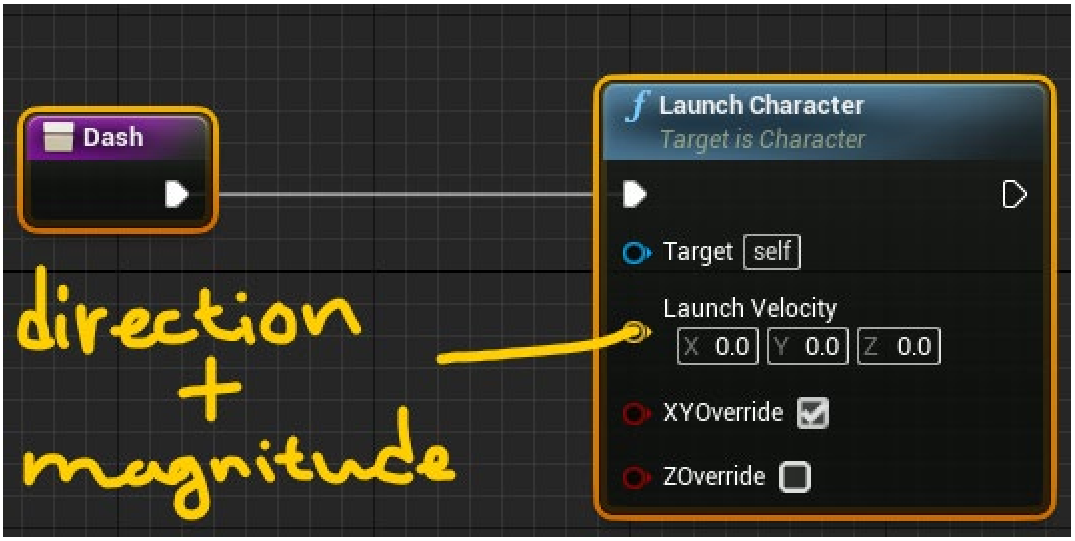
First, let's add a cooldown of 1 second between each dash. More specifically: for one second after the player has dashed, all attempts to dash again should be ignored. After the second has passed, the player can dash again.



As you may have guessed, the Gate node can be used here to decide whether to allow the dash button (E) to call the Dash function or not. Also, here's a rare instance in which Sequences come in handy: when time delays are used. Sequences execute its out Exec (white) pins in order, so the first pin (Then 0) will close the Gate and the second pin (Then 1) will open the Gate, but only after the one second delay.

Dash Function: Movement

The new node Launch Character will launch the character based on the vector input Launch Velocity. Add your own nodes to dash in the direction and magnitude of your choice. The direction should depend on where the player is facing and the magnitude should depend on the character's variable named Dash Magnitude. (We won't add powerups here, but if the player could upgrade their dash, you could simply adjust Dash Magnitude's value to let them dash further.)

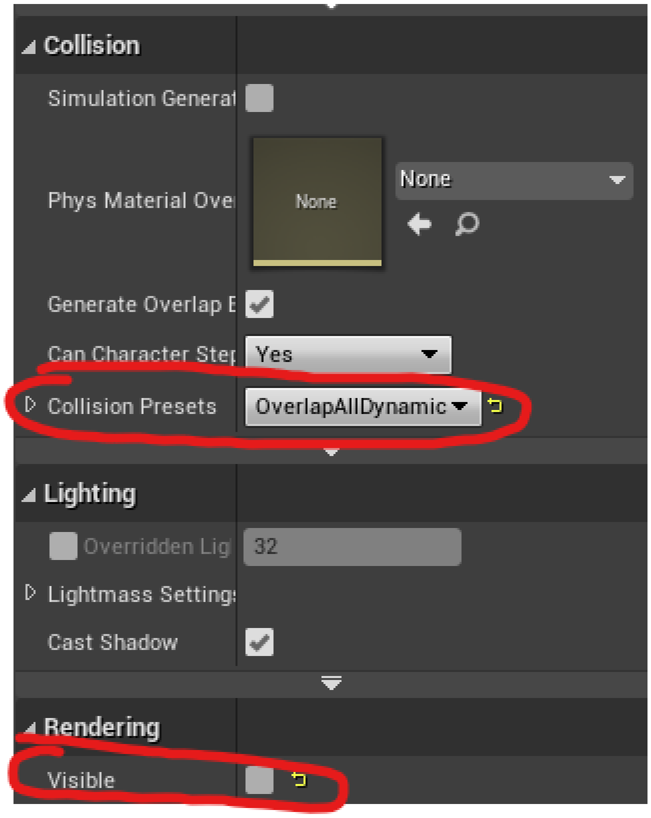


Attack: Cooldown and Trigger

When the player approaches an enemy and presses left mouse button, a light attack should occur. Since the light attack will have a 0.5 second cooldown, the cooldown part of Attack Function implementation is very similar to Dash.

Then, implement the following in the Third Person Character blueprint :

- Add a Cone Static Mesh component named Attack Hitbox to indicate the range of the attack, making it visible only when an attack is successfully made (and invisible as soon as the player releases the left mouse button)
- Attack Function should only occur when the Attack Hitbox is overlapping with an enemy (use Get Overlapping Actors). Make sure to uncheck its default visibility and set its collision presets to OverlapAllDynamic!



Collision

Simulation Generat ☐

Phys Material Ove None None

Generate Overlap E ☑

Can Character Step Yes

Collision Presets OverlapAllDynamic

Lighting

☐ Overridden Lig 32

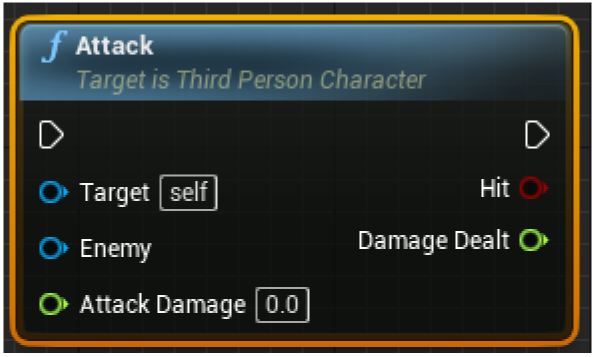
Lightmass Setting:

Cast Shadow ☑

Rendering

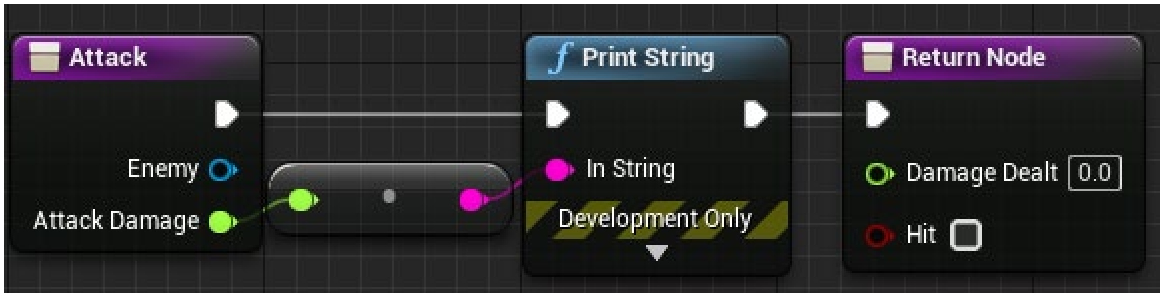
Visible ☐

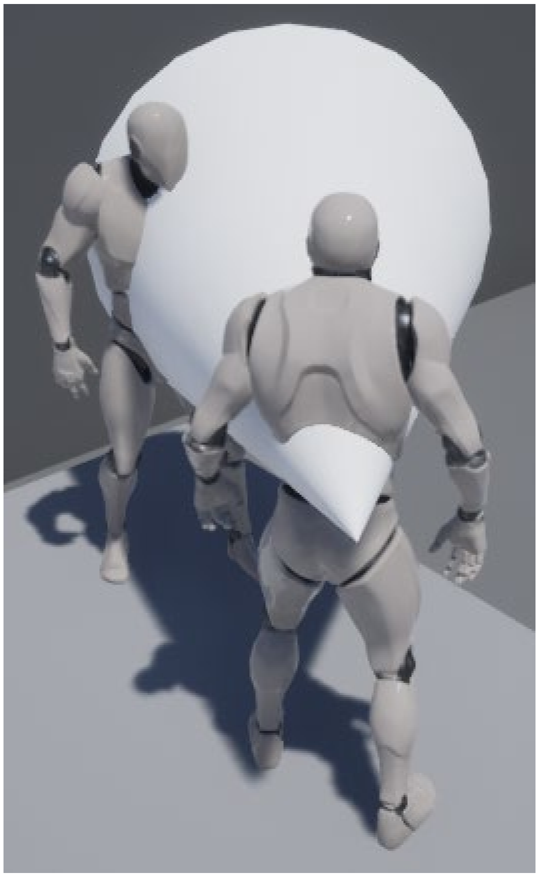
- The Attack function should have the following inputs: Attack Damage (Float) and (Enemy Object); and the following outputs Hit (Boolean; did any damage occur?) and Damage Dealt (Float; how much damage did the enemy actually take, after considering armor, dodge, etc?)



- Create a variable for the Third Person Character named Attack Damage. That will be used as an input to the node of the same name.

Use a Print String node to confirm that the Attack function is only occurring when the player is near an enemy and the cooldown is complete.





In-Lab Submission

Take a screenshot of the Dash function and the Third Person Character blueprints. Submit the screenshots to Canvas before the deadline.

Lab Instructions

Now we can add the damage dealing system, including Health, Armor, and Dodge!

First, create the Health, Armor, and DodgeProb variables (all Float) to the Enemy blueprint. Then create a Take Damage function for the Enemy:

- The Enemy has a chance to dodge the attack and receive 0 damage. If DodgeProb is .2, then there is a 20% chance.
- The Enemy has a starting Health value. Whenever it takes damage, its health goes down. If it reaches 0, it should be destroyed.
- The Enemy uses a "subtractive armor system" that doesn't run out, which means for every attack, the attack damage is subtracted by the armor. If the armor is larger than the attack, then the Enemy receives 1.0 damage.
    - If the attack damage was 15 and the armor is 10, then the Enemy receives 5 damage.
    - If the attack damage was 8 and the armor is 10, then the Enemy receives 1 damage.
- Make sure to connect the Output pins in the Return node so the Third Person Character blueprint knows if the hit was successful and how much damage was dealt.

Also, update the Third Person Character's Attack function to actually deal damage by using the Enemy's Take Damage function. The attack value should randomly range 10 below and 10 above the player's Attack Damage value.

Lab Submission

Take screenshots of the Third Person Character's Attack function and event graph blueprints, and the Enemy's Take Damage blueprint. Submit the screenshots to Canvas before the deadline.