CS 415, Fall 2022
How to Implement the "Pursuer" Enemy Type

NOTE: This tutorial covers the basics for implementing the Pursuer. It likely does not cover all the requirements for the chaser. Use intuition, creativity, and other online tutorials to aid in completing all the requirements!

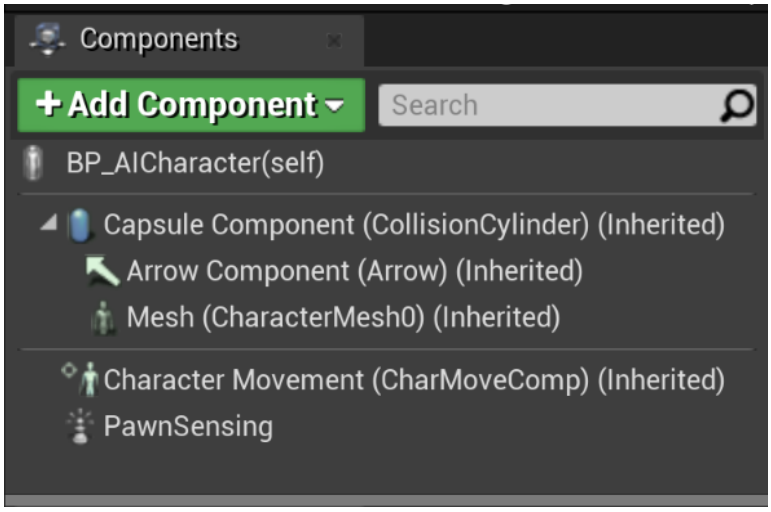
Step 1: Create the AI Controller.
Go to Add New > Blueprint Class > (type into "All Classes") > AIController and save the controller.

Step 2: Create the AI Character.
Go to Add New > Blueprint Class > Character and save the AI character. Open its blueprint editor, click on "Mesh" in the top right corner, and set the following properties on the right:

- Skeletal Mesh: SK_Mannequin
- Anim Class: Third_Person_Anim_BP

Select the "CapsuleComponent" on the left and adjust the character mesh so that the two are positioned correctly relative to each other. Finally, in "AICharacter (self)" on the left hand side menu, set its AI Controller Class to the name of the controller you created earlier.

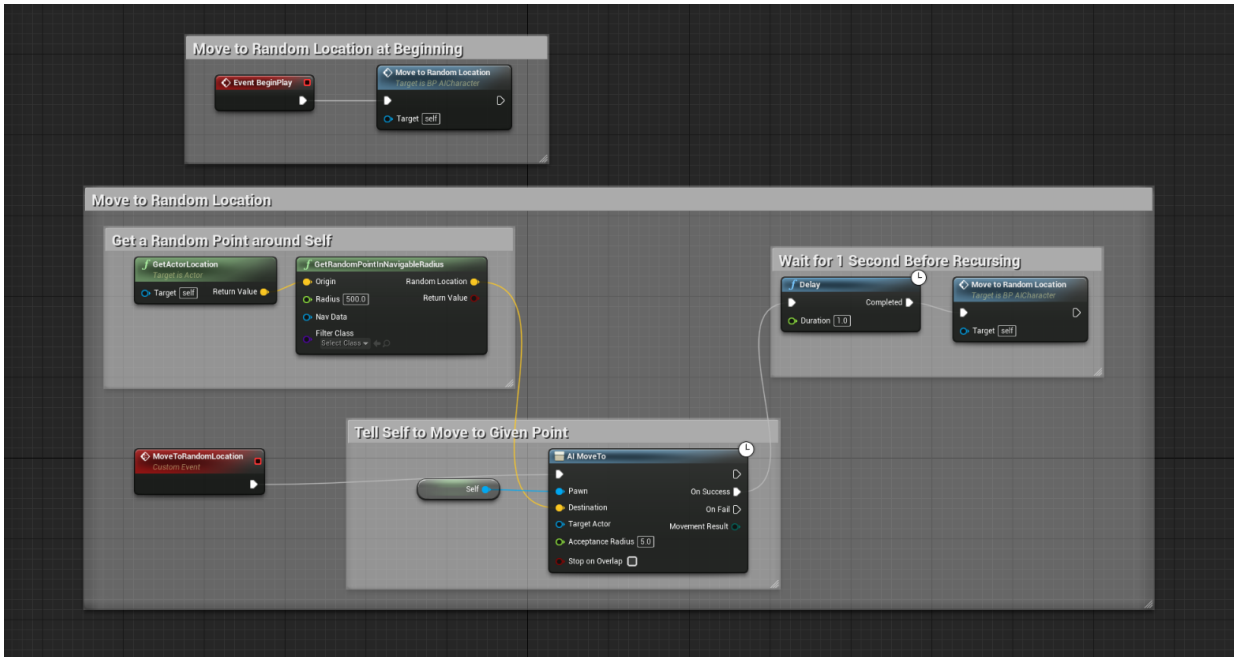
Step 3: Add Pawn Sensing.
In the AI Character, select "Add Component" and add Pawn Sensing. See below for how the component hierarchy should look in the AI Character.



Step 4: Add random movement to the AI Character.
In the Event Graph of your AI Character, create a custom event using the AI MoveTo method that can move the character to a random location within a given radius of its current location. Call this method from Event BeginPlay.

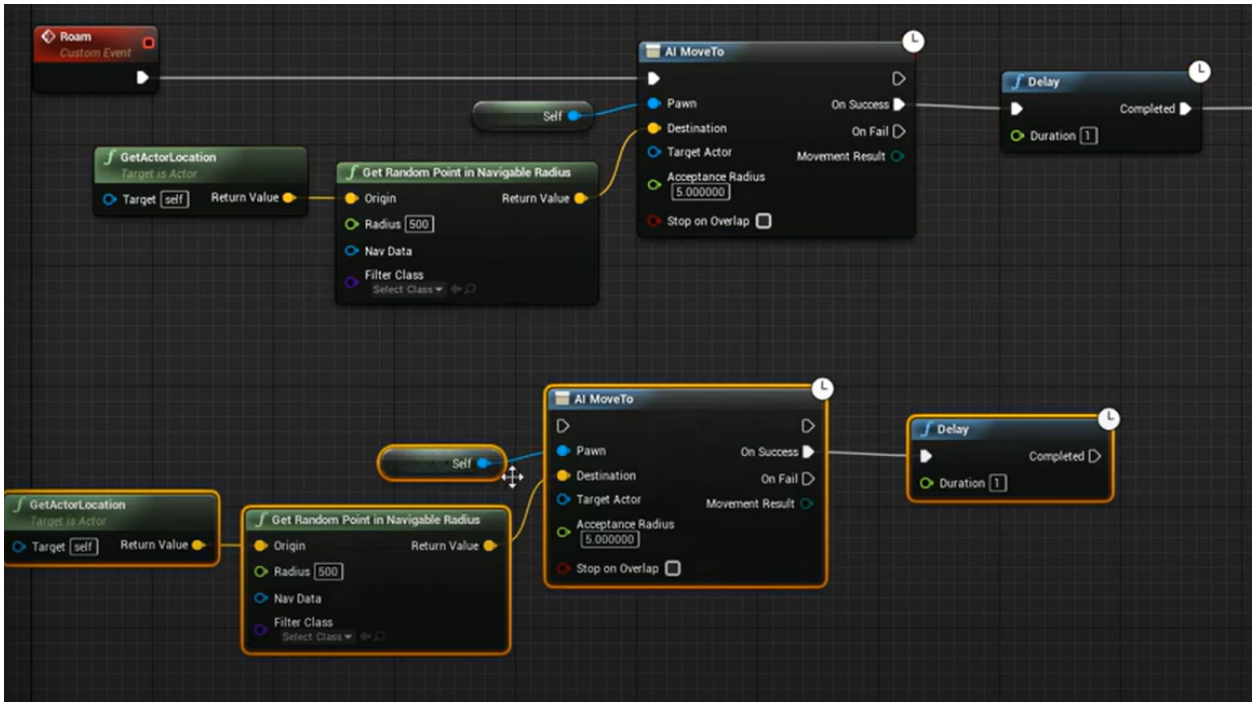
Hint: You can use the method “Get Point in Navigable Radius”.

Since you want the AI to continue moving this way, add a slight delay after the custom method and call itself. Here is how the event graph should look:



Step 5: Add Nav Mesh.
Drag and drop the AI Character into the frame. Then, on the left hand side, select Volumes > Nav Mesh Bounds Volume. Once it is added, adjust the dimensions to indicate where the AI can move.

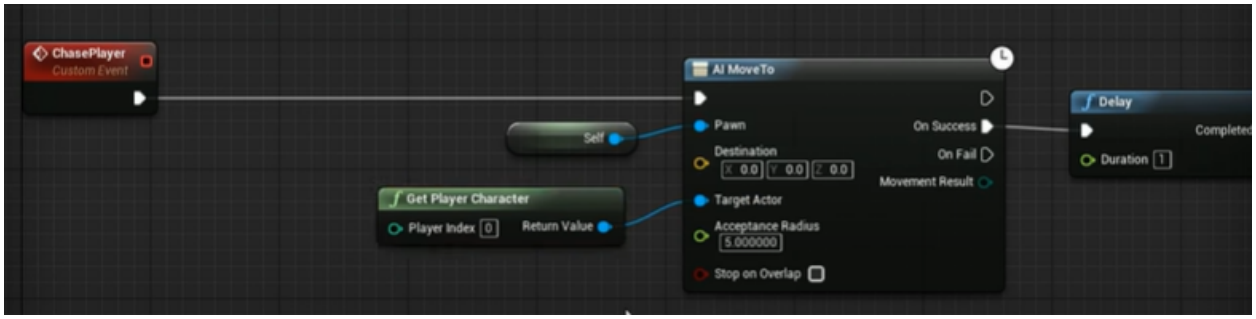
Step 6: Create ChasePlayer Event
Now that you have functionality for the “Roam” event, we will use the same logic to initiate the “ChasePlayer” event.



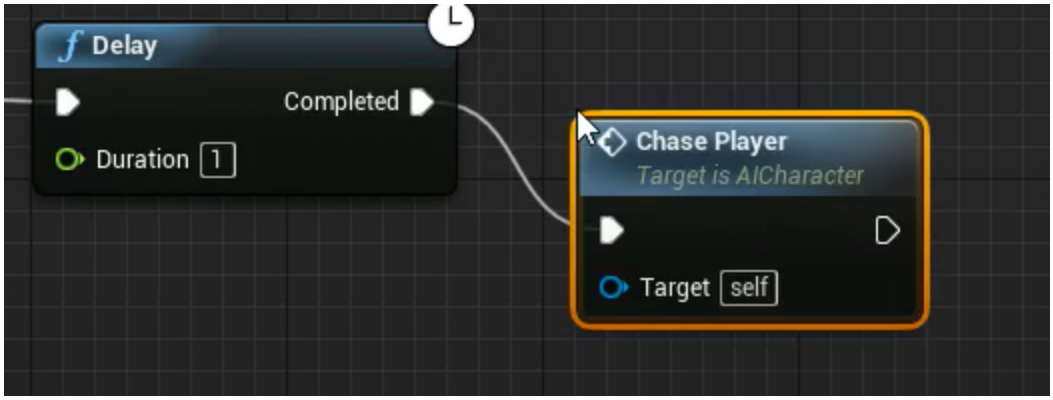
Create Custom event titled “ChasePlayer” (Right click on environment > Type “Custom Event” > Name event “ChasePlayer”

“ChasePlayer” will connect to “AI MoveTo”.

Now, replace the stuff in the “Destination” node on “AI MoveTo” with “PlayerCharacter” as the Target Actor (Delete “Get Random Point in Navigable Radius” and “Get Actor Location” > Right click on environment > Type “Get Player Character” > Connect to “Target Actor” node on “AI MoveTo”).



After Delay, call ChasePlayer.



To test, replace “Roam” in “Event Begin Play” with “ChasePlayer”. Run the code!

The next section is optional.
(OPTIONAL) Step 7: Add Enemy Functionality Upon Chase Event
Break Link between “AI MoveTo” and “Delay”. Drag from “On Success” node on “AI MoveTo” and add “Spawn Emitter at Location”.

From there, drag from “Spawn Emitter at Location” and add “Play Sound at Location” and from there, “Destroy Actor”.

Add “Get Actor Location” as the “Location” node for “Spawn Emitter at Location” and “Play Sound at Location”

You can add particle effects and sounds to your UE4 project. Add a particle effect as an “Emitter Template” on “Spawn Emitter at Location”. Add a sound to “Play Sound at Location”.

This code will cause the enemy to blow up with a sound and particle effect emitted upon collision with the player character. Change the code to fit the requirements of the MP.

(OPTIONAL) Step 8: Apply Damage
Drag from “On Success” node from “AI MoveTo” and type “Apply Damage”. “Apply Damage” will now be between “AI MoveTo” and “Spawn Emitter at Location”.

Set “Damaged Actor” node to “Get Player Character”. Play around with effects

Step 9: Enumerator Class

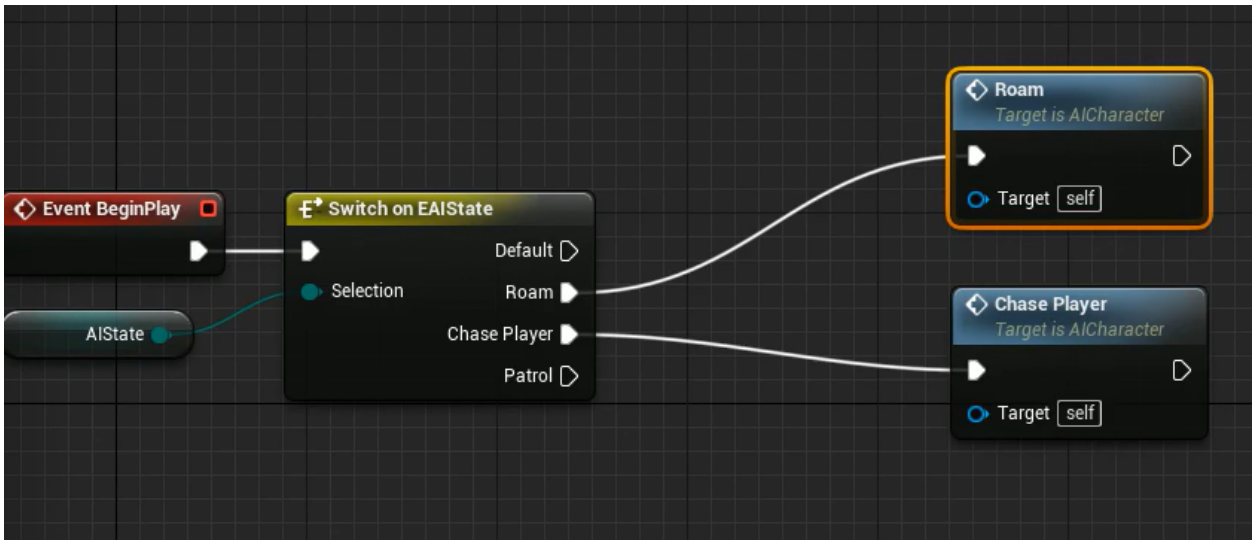
In the Content Editor, within the AI folder, create a Blueprint Enumeration class. This class will define what state our AI is in (ChasePlayer or Roam). Name this class “EAIState”.

Add three enumerators: Default, Roam, and ChasePlayer.

In the AICharacter class, create a variable called “AIState” of type “EAIState”. Make the variable public by clicking on the eye icon.

Remove connection between “Event BeginPlay” and “ChasePlayer”. Drag from “Event BeginPlay” and type “Switch on EAIState”. The AIState variable will be connected to the “Selection” node on “Switch on EAIState”.

Attach “Roam” node to “Roam” and “ChasePlayer” node to “ChasePlayer”



In viewport, click on “AICharacter” and specify default “AIState” as “Roam”. You can create multiple different AI characters with different default states.

Key Takeaways

- NavMesh Volumes are used to define bounds for AI elements in the game
    - Deals with obstacles automatically
    - Note: this handles surfaces only, doesn't work nicely for elements moving in 3D
- There are 4 main components needed to setup AI in Unreal:
    - AI Controller: Analogous to player controller, now handled by AI
    - Blackboard: A set of variables used by the AI (i.e. state variables)
    - Behavior Tree: A tree which defines how to act based on AI state
    - Behavior Tree Tasks: Subtasks to set the state of the AI

Additional Resources

In addition to the instructions above, the following set of tutorials may be helpful to you:

- https://www.youtube.com/watch?v=eBjtKsgurLU
- https://www.youtube.com/watch?v=HK3FAbIkJ-g
- https://www.youtube.com/watch?v=4UsaiOEr6Bw