We are going to make projectile launching behavior. To do this we make the projectile launching component in an actor BP and in that way it can be added to other character’s BP. Right-click and add a blueprint Scene Component as it has a transform and we are going to use it as the projectile spawn point. Name it as Projectile Launching Component. To integrate this with the AI character open the Controlling AI BP which is the AI character BP and, in the viewport, add the Projectile Launching Component. Then adjust its transform they way you want the projectiles to fire.

Create a new blueprint actor called projectile. In the viewport add a static mesh and set it to sphere. Adjust the size and set simulate physics. Make the static mesh the default scene root if you want. Add a Particle System and if the static mesh isn’t the default scene root i.e. if there’s another scene root then you need to attach the particle system to the mesh because if only the mesh moves and the root doesn’t the resulting effect will be far from the mesh. Set the particle effect to say the fire explosion effect. Adjust its transform. Add a projectile component and set the parameters such as speed, velocity, max. speed, bounciness etc.

In the Projectile Launching Component add a function/event called Launch Projectile At Location. Create a variable vector called target location. Extend the Launch projectile node and add a set Target Location node and hook its value pin to the Projectile At Location node so that an input parameter gets added to it. Extend the output exec pin of the set Target Location node and add a spawn actor from class and set the class to the projectile but here let’s use variables to make it more reusable. Extend the class pin and promote it to a variable called Projectile Class and set the default to the Projectile actor that was created. Extend the spawn transform pin and add a get world transform node as we want the projectile to be spawned at the location of the Projectile Launching Component. If you want you can increase the functionality by splitting the struct pins and then modifying the scale for variable projectile sizes.

This part is optional. Extend the return value and add a get component by class and set the component to the projectile component. Extend its return value and add an is valid node and hook its input exec pin to the output exec pin of the spawn actor at location node. Extend the is not valid pin and add a print string node and set the string to “No projectile movement”. Extend the return value of the get component by class node and add a set velocity node and hook its input exec pin to the is valid pin. Extend the velocity pin and add a vector \* float node. Extend the float pin and promote it to a variable called speed. Extend the vector pin and add a get unit direction vector. Extend the from pin and add a get world location i.e. the spawn pawn. Extend the to pin and add a target location variable. So the direction towards the target is obtained as it’s the unit vector from the current location to the target location and by multiplying it with the speed you get the velocity vector for the projectile. You can change the Projectile Class variable and speed variable’s category. Just type in anything there say “Togglable” and then make both variables public by clicking the eye button. This is just to make it easier to find in the viewport of the BP class to which we are adding the component. To check it just go to the Controlling AI BP and click on the projectile launching component. On the right-hand window under the category togglable you’ll find the variables and you can set them to the value you want.

Extend the set velocity node’s output exec pin and add a spawn sound at location node and extend the location pin and add a get world location node. Go to the content browser and right-click>sounds>sound attenuation and name it MySoundAttenuation and set the inner-radius to say 400, anything inside this distance will be in full vol. and fallout distance to say 3000, up to 3000 the sound attenuates and after that nothing will be heard. Go back to the Projectile Launching Component BP and set the value of the attenuation setting of spawn sound at location node to MySoundAttenuation. Extend the output exec pin and add a spawn emitter at location and extend the location pin and hook it to the get world location pin and set the emitter to say an explosion.

So we want the AI to be moving toward the target and at the same time fire a projectile at it. In the Controlling AI BT go to sequence which deals with the actions that are performed when you can see the player. Right now there might be the move to node. Instead of that extend the sequence node and add a simple parallel node and extend the task part (left) and hook it tot the move to node. Extend the other part and add a sequence node and this will be dealing with the firing.

Create a new task called launch projectile at target BTT. Open it and in the event graph add an event receive execute AI node. Extend the controlled pawn and add a get component by class and set the component to projectile launch component. If you want you can check the validity by extending the return value and add a check valid node and hook its input exec pin to the output exec pin of the event receive execute AI node. Extend the is valid pin and do the rest. Extend the return value of the get component by class node and add a cast to projectile launching component node. Extend the as projectile launching component pin and add a launch projectile at location. To set the location we can find the player location via blueprint but in this case we have target value stored so we extract the target location from it although it would be better to rewrite the whole thing and instead of target actor use target location instead. Anyways create a new variable called TargetActorKey add a get TargetActorKey node, extend it and add a get blackboard value as actor node extend it and add a get actor location node and hook the return value in the target location pin of the launch projectile at location node. Extend the output exec pin of the launch projectile at location node and add a finish execute node and check the success pin.

Go to the Controlling AI BT and extend the sequence node that’s attached to the simple parallel node and extend it and on the left add a launch projectile at target BTT node. Set the TargetActorKey value to Target on the right-hand window. Extend the sequence node and to the right add a wait node.

Instead of adding a wait node, to make it more reusable let’s add a wait blackboard time node. Different characters have a different cooldown time – time taken before you can use an ability or weapon again. If you add a wait node the cooldown time is a fixed value but if you add a wait blackboard time node then the wait time can be set via blackboard making it more reusable. So replace the wait node with a wait blackboard time node. Add a new BB key called CooldownTime of float type. Go to Setup\_AIC and add a get blackboard node and extend it and add a set blackboard value as float node and hook it to the rest. Set the key name etc. In this way we can set variable cooldown times for the characters.

On simulation the character chases us and fires and if it loses sight of us it stops shooting and chasing. The projectiles might fall to the ground but that might be due to gravity. If you want uncheck the enable gravity for the projectile movement and set the gravity scale to 0 or tweak the parameters.

We used a scene component for the projectile in this tutorial similarity we can use it for health, etc. and it makes it very reusable (same components for player and characters also possible) and easy to tweak and debug and is less error-prone.