**STEALTH GAME DOCUMENTATION**

**Ahmet Emre Sağcan**

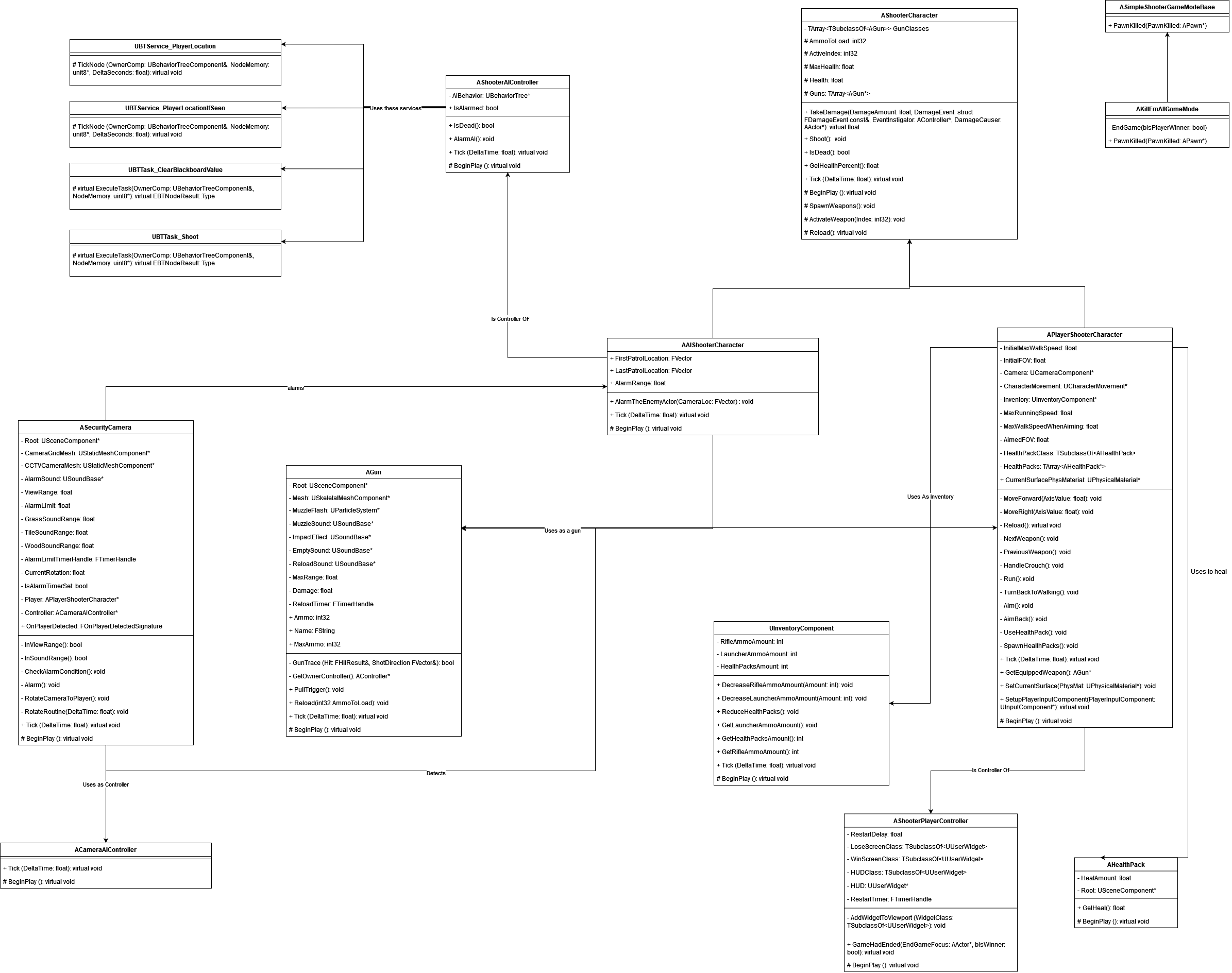
The patterns I choose:

* Abstract Factory Pattern: For creating guns and health packs for the player and enemy.
* Observer Pattern: For alarming enemies through cameras. Broadcasting the alarm is a really simple and event-driven technique to reach at other classes.

The challenges I encountered during this case:

* First challenge I encountered was that I didn’t have any crouch animation so I had to go with the most similar animation in my skeleton asset.
* For some reason I can’t figure out, in the build version of the game enemies don’t patrol probably because AI Behavior Tree is not working correctly in build version. I looked it up on internet and couldn’t find any solution. It works perfectly in editor mode. I will upload also upload a video in editor mode.
* At first AI and Player were using the same animation blueprint. In that animation blueprint footstep notify events happen. It should only happen for the player. Although the blueprint scripts still works for player, player was not making any movement. Enemies are making movement and it was enough to create a footstep event. So this created a problem where all footstep notify events make the character to make walking noise. Ending up with a really noisy sound. When I discovered the problem I made copy of the animation blueprint and set it to enemy blueprint. Now they are separate and enemy blueprint doesnt use any notify events.
* At first the AI was not finishing its patrol before moving to the player when detected. I fixed it with a blackboard check.
* At first I though I can use EphysicalSurfaceType as a global variable in my player shooter character header. It didn’t go as expected so I switched to getting physicsmaterial.

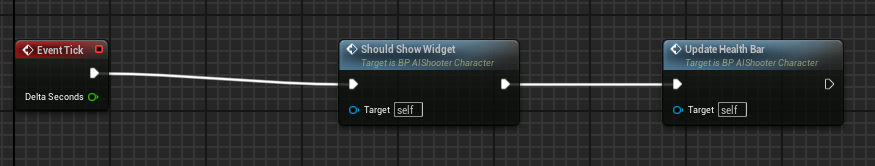
UML CLASS DIAGRAM

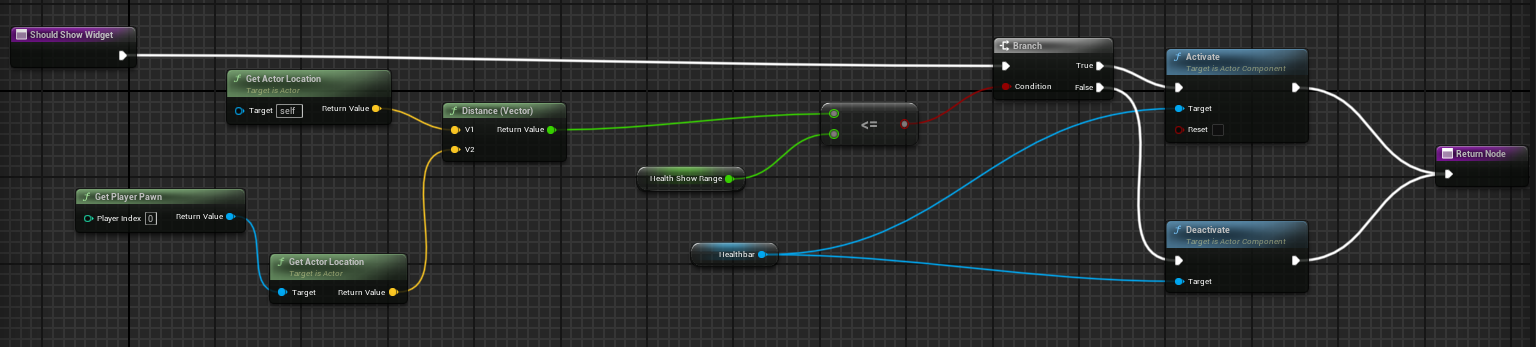


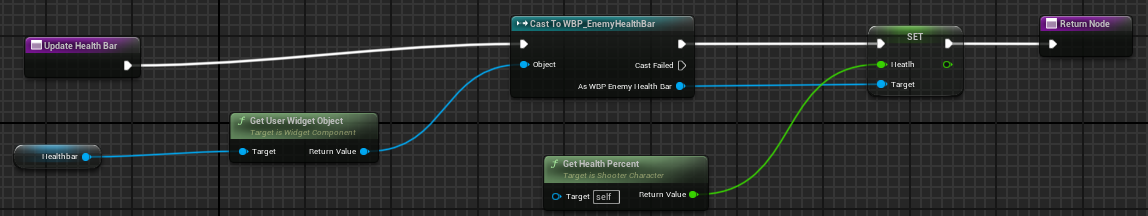
**AIShooterCharacter**  
  
BeginPlay: AI gets the references of all the camera actors that are currently in level. Then subscribes to their OnPlayerDetected events one-by-one.

AlarmTheEnemyActor: Alarms the enemy actor when the camera alarms this object. If it is close enough to that camera. Alarming happens.

Inside its blueprint.







ShouldShowWidget: If player is close enough to the enemy show the enemy healthbar.

UpdateHealthBar: Updates the healthbar slider in every tick.

**BTService\_PlayerLocation**

Updates the last known player location (as blackboard value) in the enemy AI behavior tree.

**BTService\_PlayerLocationIfSeen**

Updates the Player (as blackboard object) in the enemy AI behavior tree. If player is in the lineofsight of enemy, this value is set.

**BTTask\_ClearBlackboardValue**

Clears the selected blackboard key of that node in the AI behavior tree.

**BTTask\_Shoot**

Orders enemy to shoot at player.

**Gun**

PullTrigger(): All of the action after pulling the trigger happens here. Muzzle flashes and muzzle sound happen here. Gets the hit from GunTrace. Creates another impact effect and sound using that hit location. Then calls the take damage of that actor if all things go well.  
  
Reload(): First sets some timer for reloading sound. Then adds ammo to the current ammo.

GunTrace(): Gets the trace of the gun using the player viewpoint. Ignores the gun and the character itself.

GetOwnerController(): Gets the player controller of the owner.

**HealthPack**

GetHeal(): Basically gets healed from heal amount.

**InventoryComponent**

DecreaseRifleAmmoAmount(): Decreases the total rifle ammo in the inventory.

DecreaseLauncherAmmoAmount(): Decreases the total launcher ammo in the inventory.

ReduceHealthPacks(): Decreases the total health packs in the inventory.

**KillEmAllGameMode**

PawnKilled(): If a pawn is killed, check whether it is player. If it is player, player loses the game. If it is an AI, check whether all other AIs are dead. If they are, player wins. If they are not don’t touch anything.

EndGame(): Ends the game state according to the bIsPlayerWinner.

**PlayerShooterCharacter**

BeginPlay(): Spawn health packs at first and set initial values.

Reload(): Reload happens and updates the inventory values.

Aim(): Walk slow when aimed.

UseHealthPack(): Get healed with health pack. If health will be full when healed, set it to max health. Then destroy it.

SetCurrentSurface(): Sets the current surface that the player walks on.

**Security Camera**

Tick(): If player is in view or sound range rotate the camera to player.

InViewRange(): If distance to player is enough and if the player is in line of sight then camera can see player.

InSoundRange(): If distance to player according to its walking surface is enough for that surface’s distance range and player is in line of sight then camera can hear player.

CheckAlarmCondition(): If player is not dead and still in the view or sound range, alarm the subscribers to this camera. Alarm timer can be set again.

Alarm(): Alarm the subscribers to this camera. Make an alarm sound.

RotateCameraToPlayer(): Finds the rotation vector to the target(player). Then if alarm timer is not set, set the alarm timer.

**ShooterAIController**

BeginPlay(): Run the behavior tree and then set the patrol locations, get them from the character object.

AlarmAI(): Called when character is alarmed. Set its last known player location to the player location.

**ShooterCharacter**

TakeDamage(): Character takes damage then checks if it is dead. If dead calls pawnkilled function of gamemode.

Shoot(): Call the current gun’s pull trigger method.

Reload(): Add remaining ammo to the current gun.

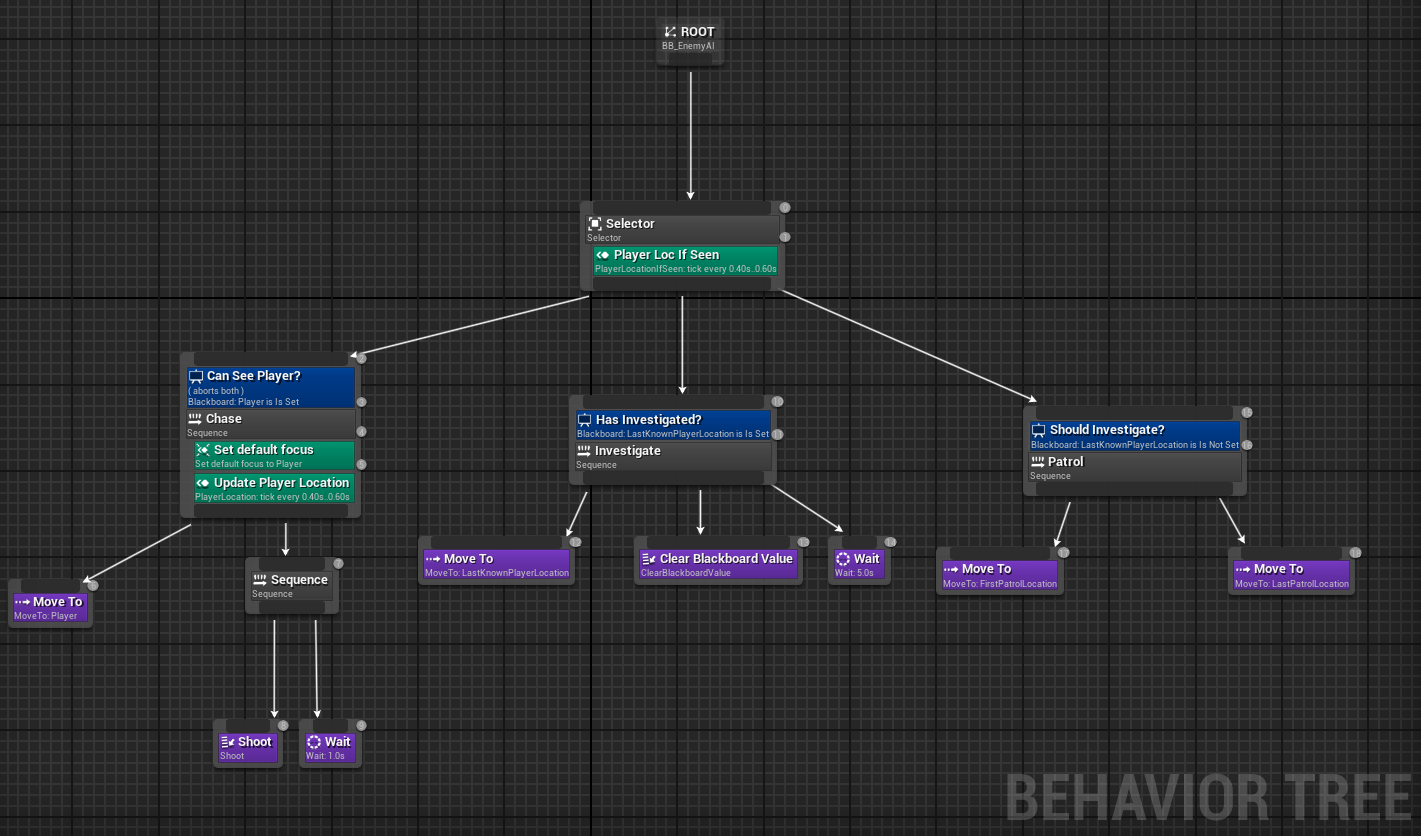
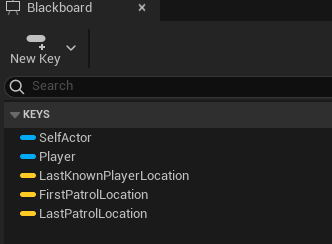
ActivateWeapon(): Sets the activate index and hides all the other guns.

GetHealthPercent(): Get the health percent for health bar widgets.

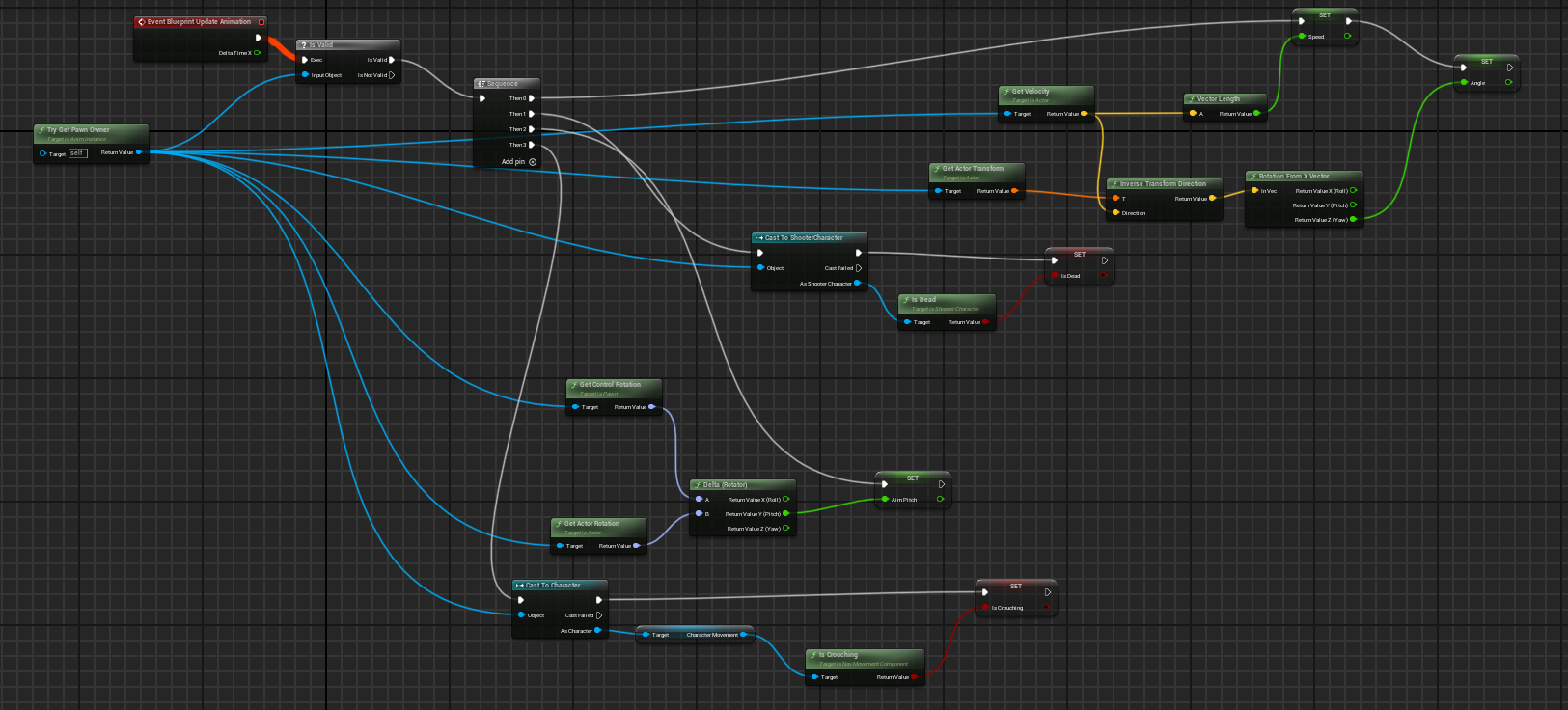
**ShooterPlayerController**

GameHasEnded(): If player won render that screen, if not render lose screen. Then restart in a delay amount.

**Enemy Behavior Tree**

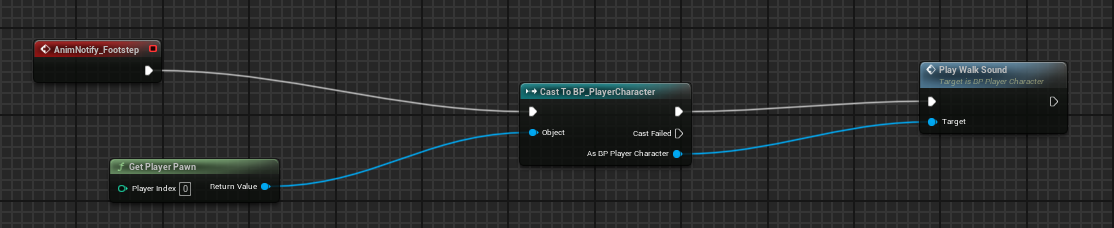
****

If enemy can see player move to player then shoot, then wait for 1 s. Then keep doing this until it can’t see player. If it can’t see player. Move to the last seen location. Wait for some time. Then go back to the regular patrolling.

**AIShooter Animation Blueprint**

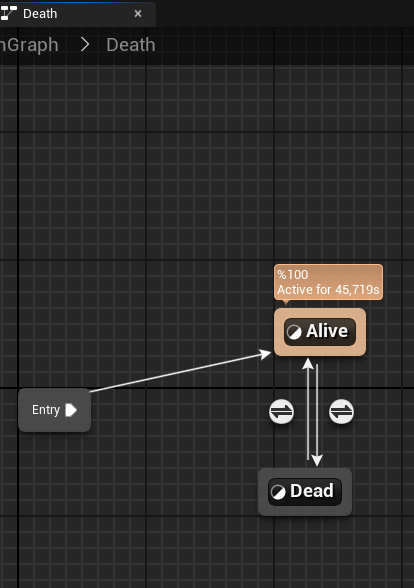
Sets the values inside the blueprint through here. Which is later on used for animation states.

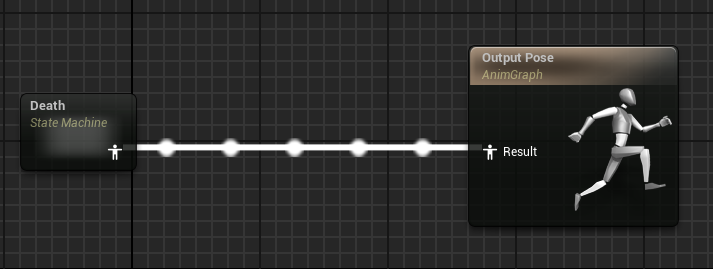
Looks for whether player does certain actions.

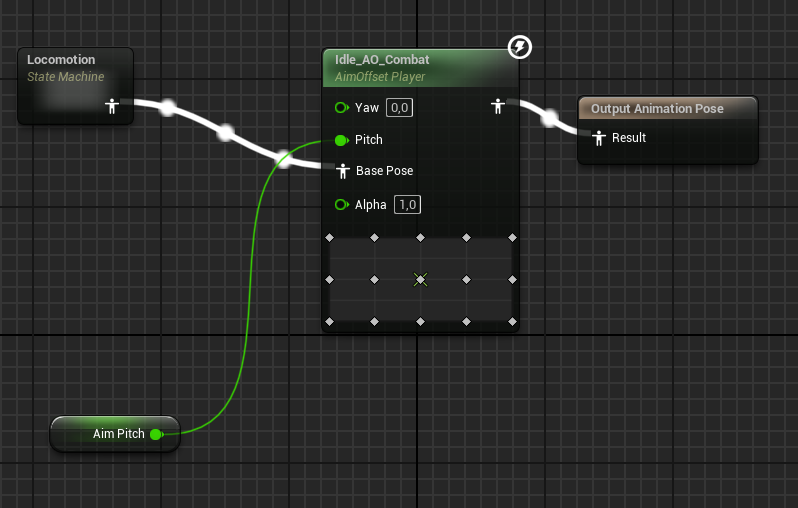
**PlayerShooter Animation Blueprint**

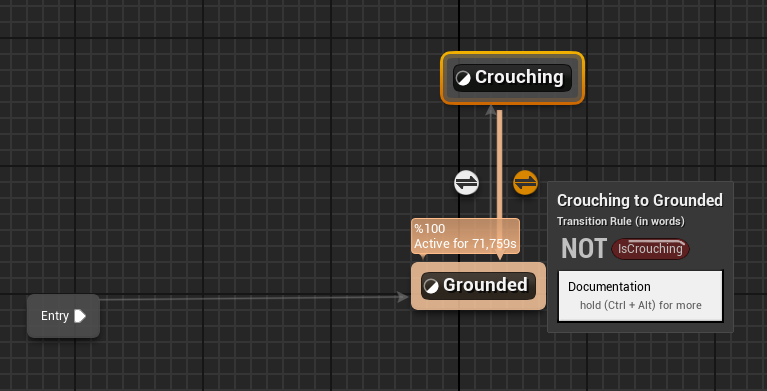
In addition to the previous blueprint scripts, this blueprint also plays the walk sound.

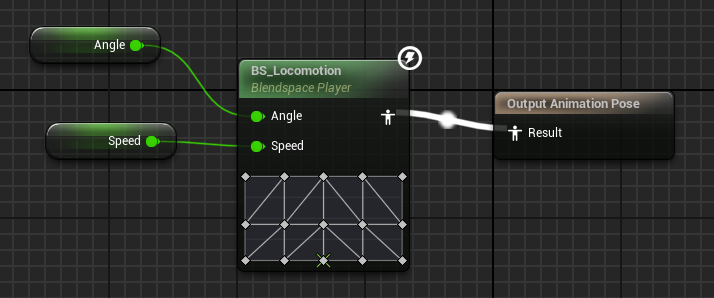
They share the same anim graph. Inside Death State Machine

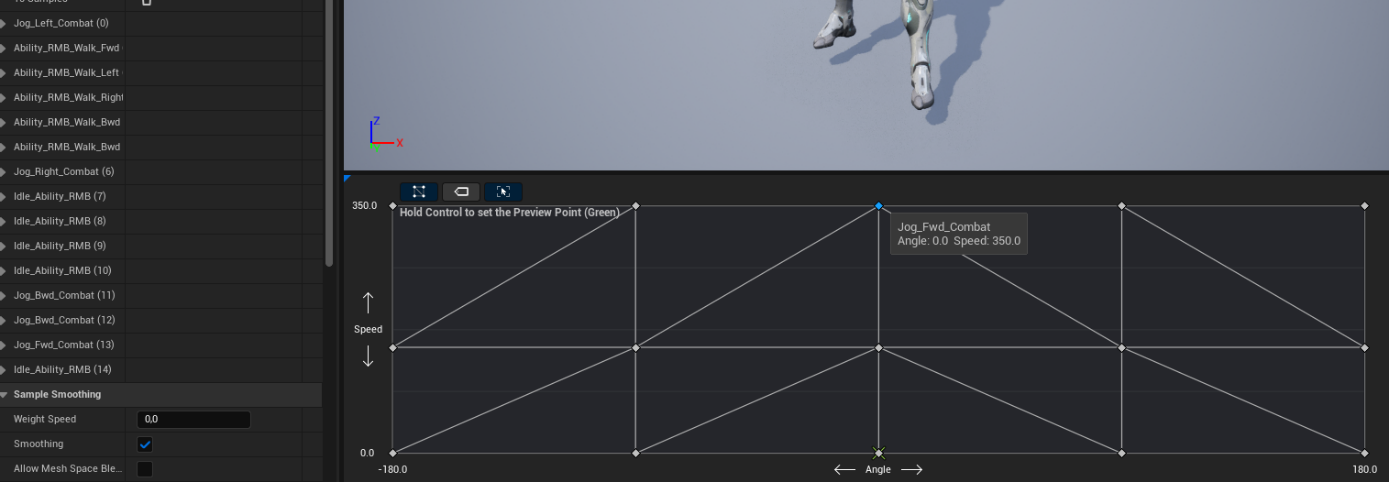




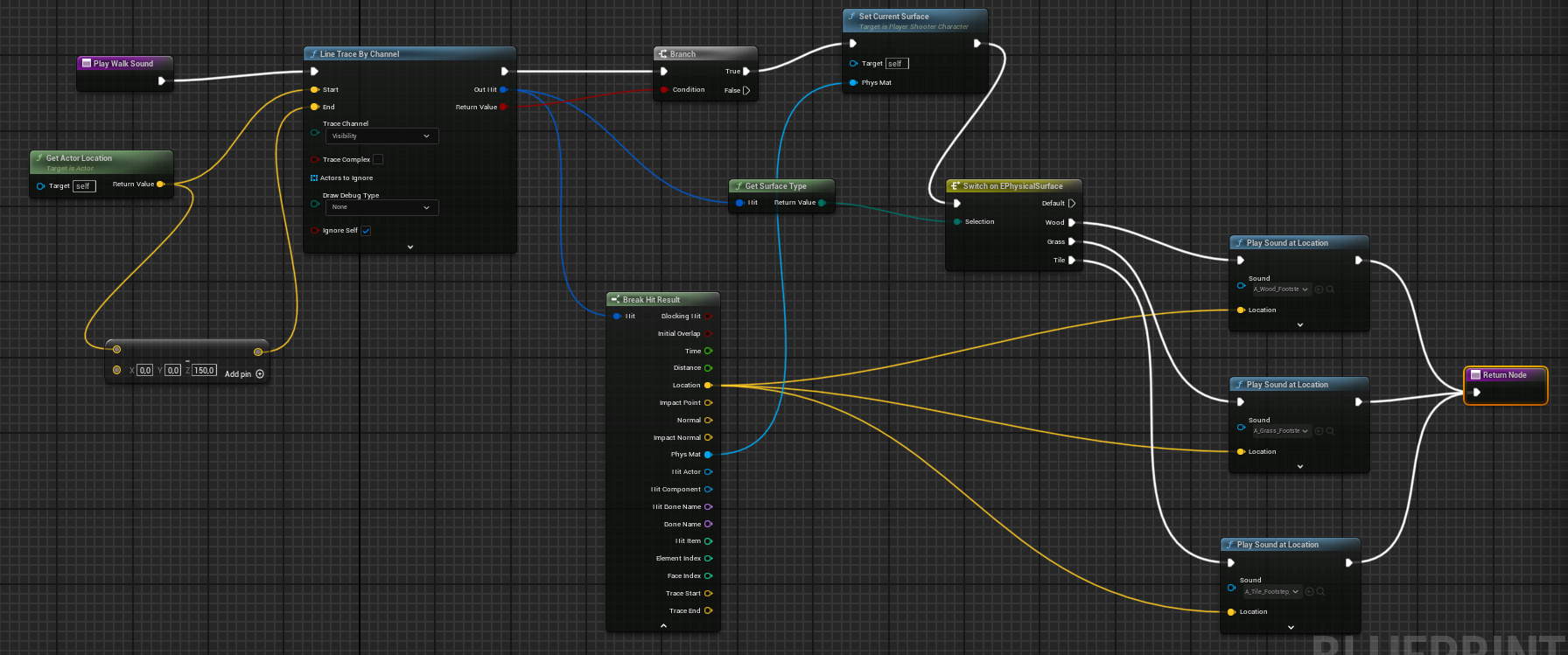
Inside Alive State

 Inside the locomotion state machine

Inside the grounded state

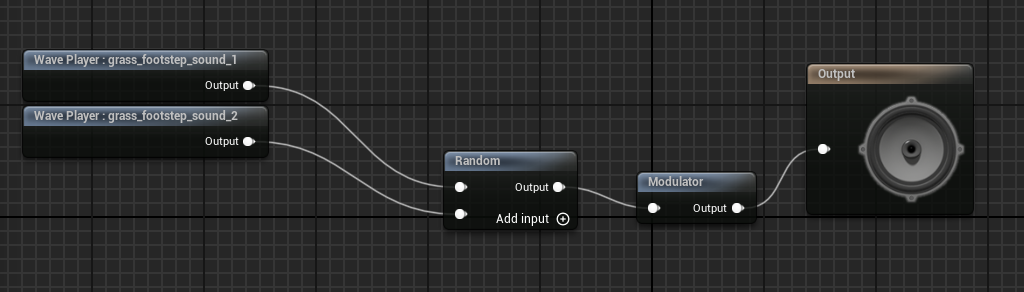
BS\_Locomotion

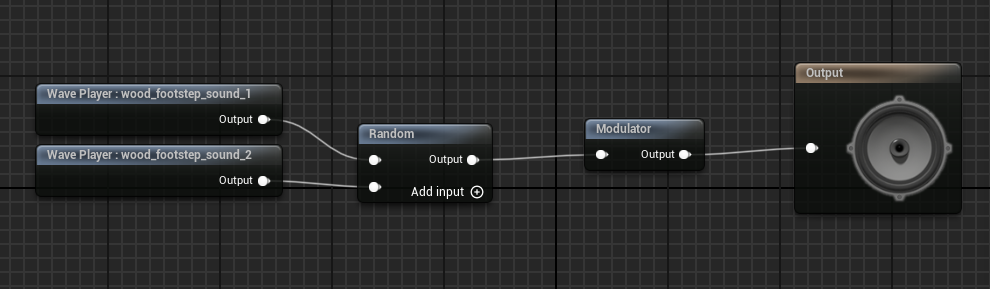
**PlaySurfaceSound Blueprint Function**

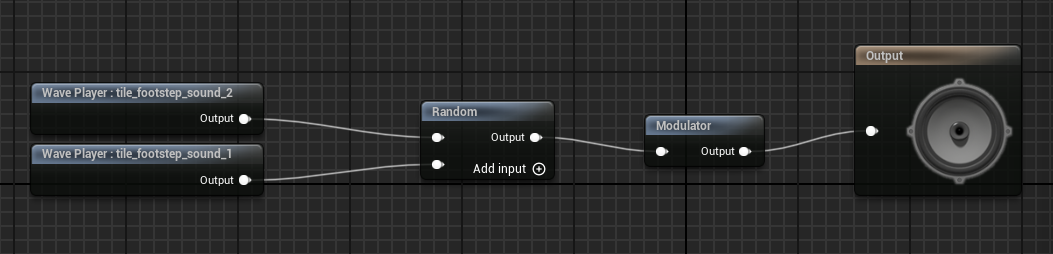
****

Line trace from the player detects the underlying surface and I get its physics material to store using setcurrentsurface from c++ class. Security Cameras look at this value. Then surface type switch decides which sound cue to play.

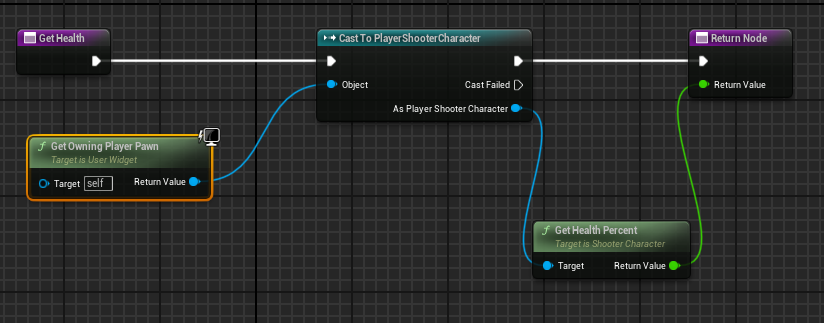
**Sound Cues**



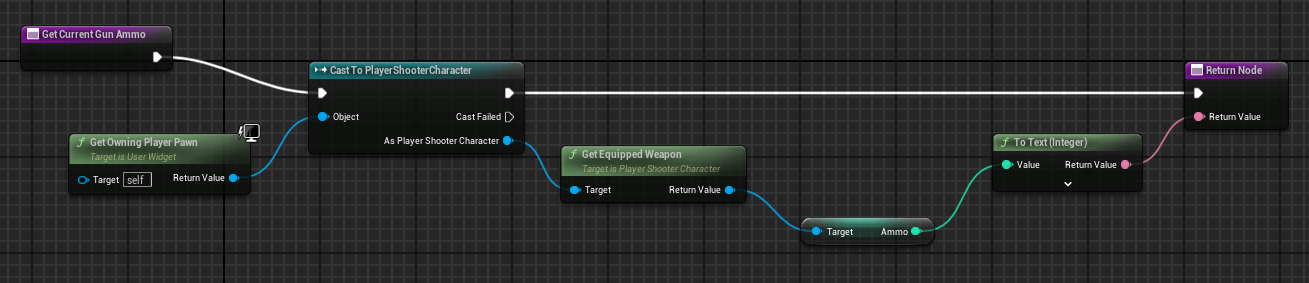




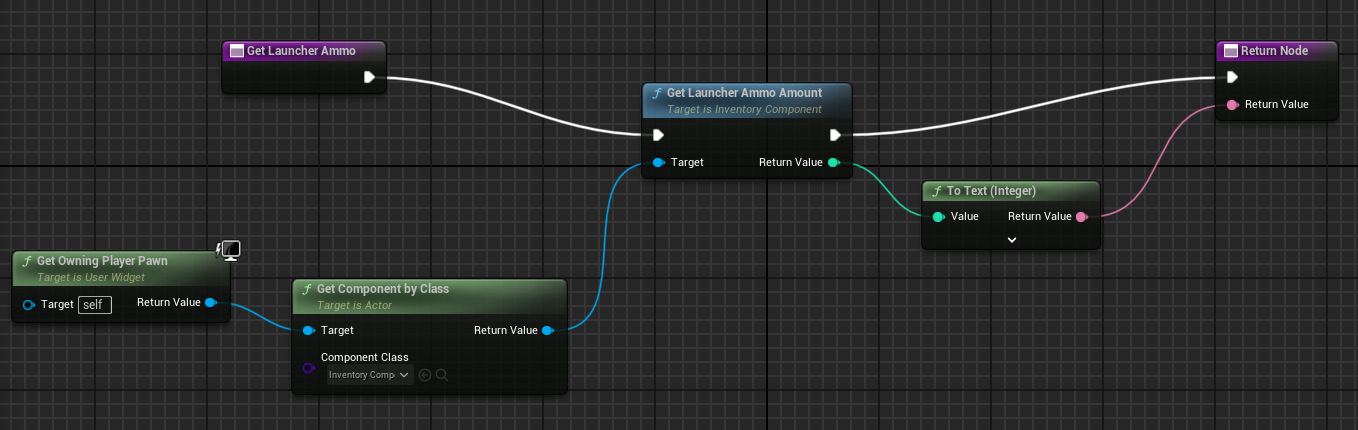
**Widget Blueprint Event Graphs**

Getting the current health percent to show in UI.

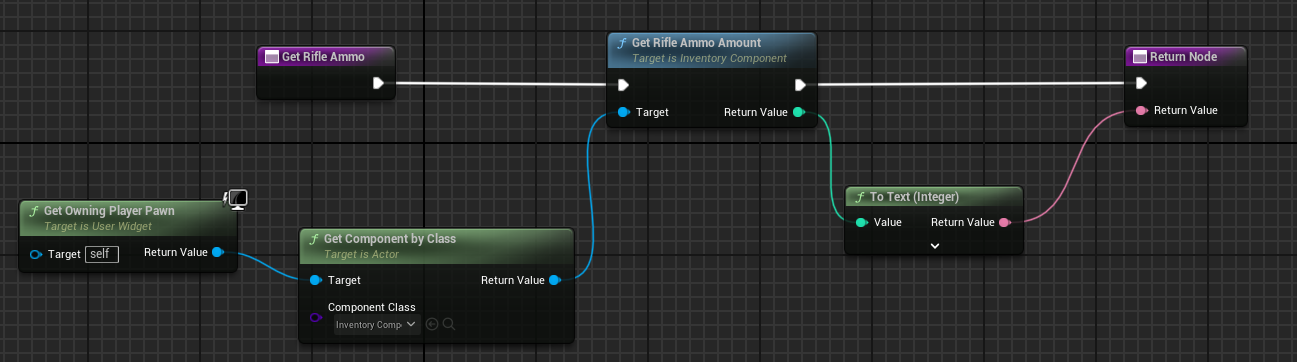
Getting the current gun ammo.



Getting the current remaining launcher ammo to display in the UI.



Getting the current remaining rifle ammo to display in the UI.



Getting the remaining health packs amount to display in the UI.

