FPS Multiplayer Kit Documentation Example

Table Of Contents

Steam Integration

Enabling Online Services Plugins

1. Head over to your projects plugins, you can do that by going to edit (found in top left of screen) → project settings

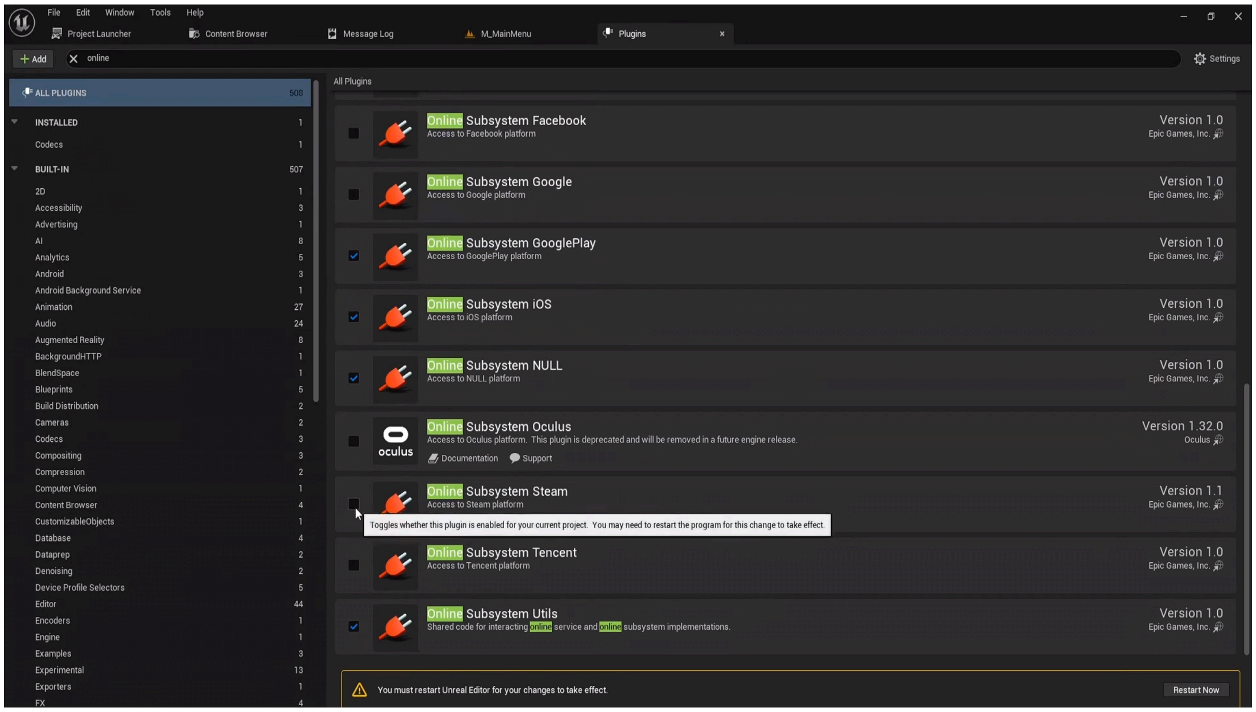
2. Select All plugins heading, and make sure the following plugins are enabled:
   → Online Services
   → Online Subsystem
   → Online Services OSS Adapter
   → Online Sub System Steam

   Note: *(Some of these plugins may or may not already be enabled)*

3. Restart your editor

Note: *Steam will have to be open when launching a packaged version of the project to use its functionalities*

[Integrating Steam Subsystem Video Tutorial Version]



ALL PLUGINS	508
INSTALLED	1
Codecs	1
BUILT-IN	507
2D	1
Accessibility	3
Advertising	1
AI	8
Analytics	5
Android	3
Android Background Service	1
Animation	27
Audio	24
Augmented Reality	8
BackgroundHTTP	1
BlendSpace	1
Blueprints	5
Build Distribution	2
Cameras	2
Codecs	3
Compositing	3
Compression	2
Computer Vision	1
Content Browser	4
CustomizableObjects	1
Database	4
Dataprep	2
Denoising	2
Device Profile Selectors	5
Editor	44
Encoders	1
Engine	1
Examples	3
Experimental	13
Exporters	1
FX	4

	Online Subsystem Facebook Access to Facebook platform	Version 1.0 Epic Games, Inc.
	Online Subsystem Google Access to Google platform	Version 1.0 Epic Games, Inc.
✓	Online Subsystem GooglePlay Access to GooglePlay platform	Version 1.0 Epic Games, Inc.
✓	Online Subsystem iOS Access to iOS platform	Version 1.0 Epic Games, Inc.
✓	Online Subsystem NULL Access to NULL platform	Version 1.0 Epic Games, Inc.
	Online Subsystem Oculus Access to Oculus platform. This plugin is deprecated and will be removed in a future engine release. Documentation Support	Version 1.32.0 Oculus
	Online Subsystem Steam Access to Steam platform	Version 1.1 Epic Games, Inc.
	Online Subsystem Tencent Access to Tencent platform	Version 1.0 Epic Games, Inc.
✓	Online Subsystem Utils Shared code for interacting online service and online subsystem implementations.	Version 1.0 Epic Games, Inc.

Toggles whether this plugin is enabled for your current project. You may need to restart the program for this change to take effect.

You must restart Unreal Editor for your changes to take effect. Restart Now

Setting Up Custom Attachments

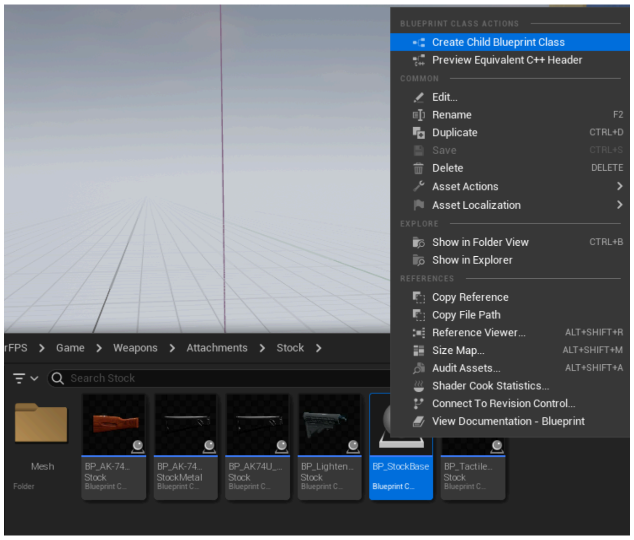
If you are looking to add sights, there is a separate more in depth section to adding sight attachments as the process is a little different

Creating A New Attachment Actor

1. Head Over to Content → Multiplayer FPS → Attachments → Select A Folder based on your new attachment, i.e. New Stock → Stock folder

2. Create a New Folder within the Stock Folder, name it according to your new attachment, i.e. tactical stock. Within that new folder, create a folder named Mesh

3. Right Click BP_Stock and click Create Child Blueprint class, this will inherit any logic and variables from the parent class BP_Stock into our new custom stock attachment, without us having to redo any code or logic.


BLUEPRINT CLASS ACTIONS
Create Child Blueprint Class
Preview Equivalent C++ Header
COMMON
Edit...
Rename F2
Duplicate CTRL+D
Save CTRL+S
Delete DELETE
Asset Actions
Asset Localization
EXPLORE
Show in Folder View CTRL+B
Show in Explorer
REFERENCES
Copy Reference
Copy File Path
Reference Viewer... ALT+SHIFT+R
Size Map... ALT+SHIFT+M
Audit Assets... ALT+SHIFT+A
Shader Cook Statistics...
Connect To Revision Control...
View Documentation - Blueprint
rFPS > Game > Weapons > Attachments > Stock >
Search Stock
Mesh Folder
BP_AK-74... Stock Blueprint C...
BP_AK-74... StockMetal Blueprint C...
BP_AK74U_... Stock Blueprint C...
BP_Lighten... Stock Blueprint C...
BP_StockBase Blueprint C...
BP_Tactile... Stock Blueprint C...

4. Open up your new actor created, and select the Static Mesh within it. Fill out its mesh and you can then close the new actor created

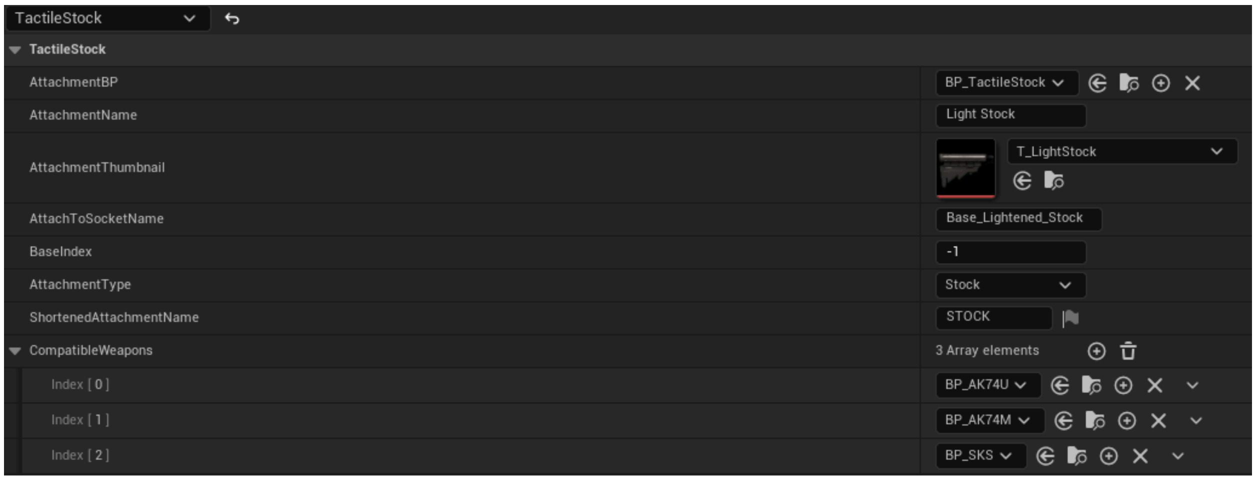
5. Now we need to update our weapon customization system to know that there is a new attachment in our game. Head on over to Content → Multiplayer FPS → Blueprints → Customization → Data, and Open up DT Attachments

6. In the data table, select Add at the top to add a new row to the table. We now have a few properties we can edit such as the actor to spawn, its name in game, sockets to attach to ect.

7. Fill out the actor to use with your new actor, give it an attachment name, apply a thumbnail texture image to represent the attachment, give it a socket name e.g. Base_TacticalSocket. This socket name will be used later to add to our gun's skeleton asset. Be sure to use a consistent naming convention. I use Base_ for weapon sockets, so its likely easiest to continue using Base_. Next set the BaseIndex to -1, select the attachment type to the appropriate ENUM, i.e. Stocks for a stock, pistol grip for a pistol grip, Muzzle for muzzle devices and so on.

8. The next step is to add a shortened name, after that add an element to CompatibleWeapons array, this allows you to select what weapons specifically you want to have access to your new attachment.

Example of a pre-existing attachment set-up:



TactileStock

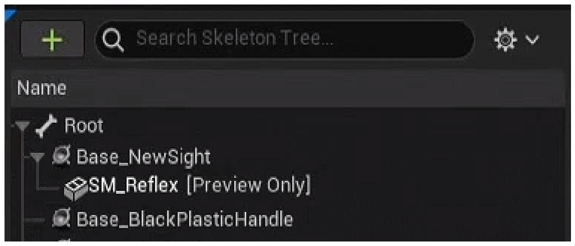
AttachmentBP	BP_TactileStock
AttachmentName	Light Stock
AttachmentThumbnail	T_LightStock
AttachToSocketName	Base_Lightened_Stock
BaseIndex	-1
AttachmentType	Stock
ShortenedAttachmentName	STOCK
CompatibleWeapons	3 Array elements
Index [0]	BP_AK74U
Index [1]	BP_AK74M
Index [2]	BP_SKS

Creating the Attachment Socket

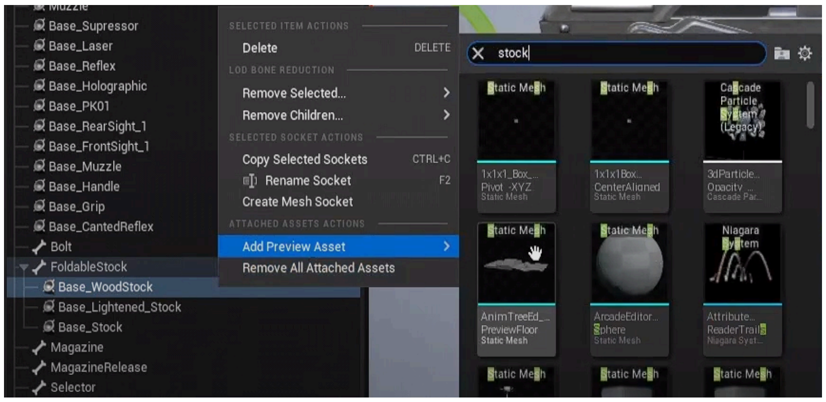
The last step is to create the socket in which to attach the new actor to.

1. First, head over to your weapon's skeleton, in my case im going to use the SKM_AK74U_Frame found over at Multiplayer FPS → Game → Weapons → AK74U → Mesh

2. Press the green plus icon within the skeleton to create a new socket, name it the exact same name you used for the AttachToSocketName in the new data table row you just made.



3. To make things easier, right click the new socket and Add Preview Asset, search for your new custom mesh attachment and it will give you a preview to move around and position to make it easier



Once you have created the socket, you can go ahead and test things out by playing, and selecting the weapon you assigned and giving it a go in the game.

[Adding Custom Attachments Video Tutorial]

Setting Up Custom Optic Attachments

Setting up Custom Optic Attachments is nearly identical to custom attachments, however custom sight attachments require a socket created within the mesh itself. This socket is responsible for Aiming Down Sights, the socket is positioned center of the optic and the location and rotation of the socket is then fed into the ADS system where the socket transform can be interpolated from the base VB Sight position (aka the position of both hands) to the new sight socket transform.

Prerequisites: It is recommended to have any Optic meshes Facing Positive X before import, it makes creating the sockets for ADS easier, its not needed but makes life easier. Otherwise you may have to offset the Aim Socket by -90 degrees.

Importing your Optic

1. Head over to Multiplayer FPS → Game → Weapons → Attachments, and Open up Optics folder.

2. Create a new folder for your Optic, i.e. Reflex Sight or Trijicon Sight. Within the folder, create a mesh folder, and within the mesh folder create a materials folder and within that create a textures folder. It should end up being Trijicon Sight → Mesh → Materials → Textures.

3. Import your mesh to the mesh root folder, and drag your textures to the textures folder

4. Head on over to MultiplayerFPS → Game → Weapons → MasterMaterials, right click and create a material instance. Rename this to whatever your new Sight Name, i.e. M_TrijiconSight. Drag this Instance into your Materials folder for your sight.

5. Open up the material and fill out your Ambient Occlusion, Roughess, Metallic, Albedo and Normal Map textures. Then head over to your new Optic Mesh and apply the texture to its respective Material Slots.

Adding the Parallax Sight Material

To add a reticle to the sight, add any of the reticle materials to your glass of the optic. These materials can be found in Content → MultiplayerFPS → Game → Weapons → Attachments → Optic → Materials. Alternatively you can create a instance via M_SightBase and customize it to your liking

Creating the Optic Actor

1. Right Click BP_Sight and create a child blueprint class. The parent class can be found at Content → MultiplayerFPS → Game → Weapons → Attachments → Optic.

2. Drag the child class into your new Sight Folder, open it up and replace its mesh with your own.

3. Now we need to update our weapon customization system to know that there is a new Optic in our game. Head on over to Content → Multiplayer FPS → Blueprints → Customization → Data, and Open up DT Attachments

4. In the data table, select Add at the top to add a new row to the table. We now have a few properties we can edit such as the actor to spawn, its name in game, sockets to attach to ect.

5. Fill out the actor to use with your new actor, give it an attachment name, apply a thumbnail texture image to represent the attachment, give it a socket name e.g. Base_Trijicon. This socket name will be used later to add to our gun's skeleton asset. Be sure to use a consistent naming convention. I use Base_ for weapon sockets, so its likely easiest to continue using Base_. Next set the BaseIndex to -1, select the attachment type to the appropriate ENUM, i.e. Stocks for a stock, pistol grip for a pistol grip, Muzzle for muzzle devices and so on.

6. The next step is to add a shortened name, after that add an element to CompatibleWeapons array, this allows you to select what weapons specifically you want to have access to your new Optic.

Creating the Aim Socket

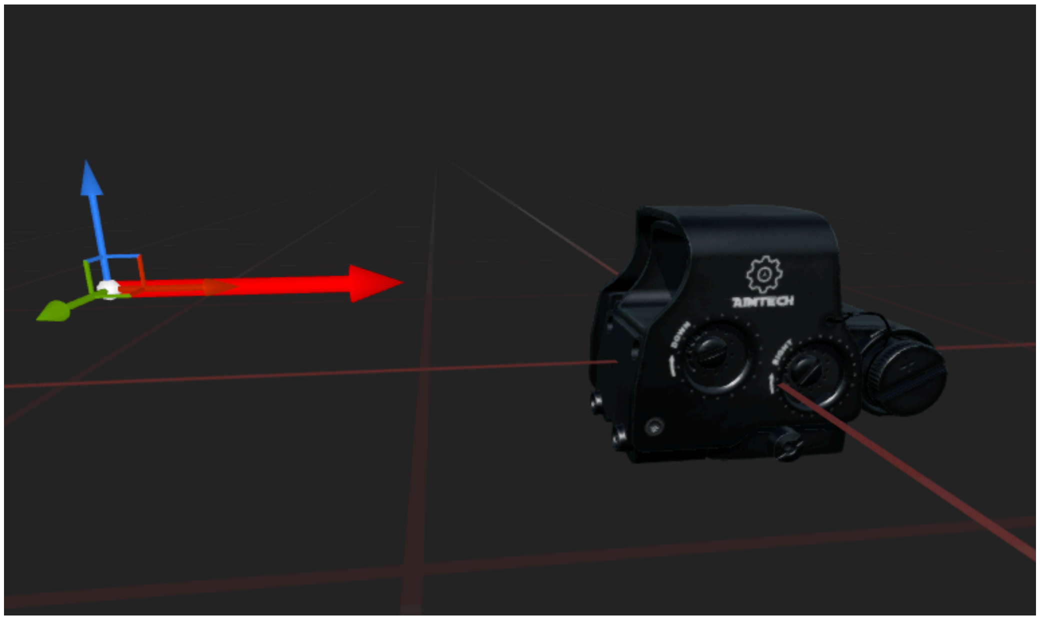
NOTE: *The process is the exact same for iron sights, except you only need to do the process for the rear ironsight. I recommend duplicating one of the existing ironsight actors, i.e. M4A1 rear ironsight, and then continuing the process. If your ironsight should flip when a optic is on the rail, make sure you plug in the Flipped Static mesh in the FlipMesh variable in default settings of the Ironsight Actor you duplicated.*

The ADS system has been completely reworked. Making it even easier to use. No more sockets, X forward, static or skeletal; it works straight out of the box with any rotation or scale.

1. Locate your newly created optic child class, head over to viewport and select Aimpoint

2. Position the aimpoint component in the middle of the optic lens, that way it is centered when aiming down sight

3. That's it! You can head in game and adjust the optic position by readjusting aim point to finalize the aiming location.



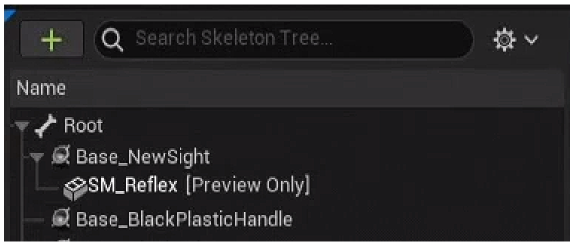
Note: *You can rotate, and move the Aimpoint backward to simulate how far the player should move the gun from the players eyes, you may find this useful for scopes like 8x that take up much more of the screen vs reflex pistol sights*

Creating Optic Attach Socket

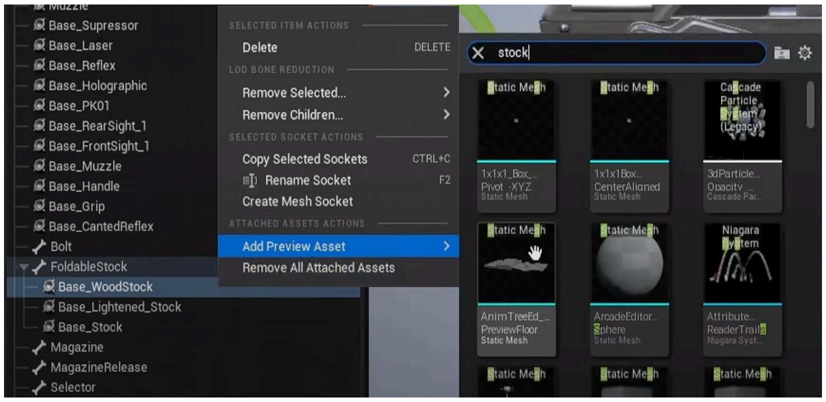
The last step is to create the socket in which to attach the new actor to.

4. First, head over to your weapon's skeleton, in my case im going to use the SKM_AK74U_Frame found over at Multiplayer FPS → Game → Weapons → AK74U → Mesh

5. Press the green plus icon within the skeleton to create a new socket, name it the exact same name you used for the AttachToSocketName in the new data table row you just made.



6. To make things easier, right click the new socket and Add Preview Asset, search for your new custom mesh attachment and it will give you a preview to move around and position to make it easier



Once you have created the socket, you can go ahead and test things out by playing, and selecting the weapon you assigned and giving it a go in the game.

Note: If the aim down sight doesn't work as expected, adjust the Aim Socket you created in the Optics Mesh, adjust its rotation and location to get it lining up.

[Adding Custom Optics Video Tutorial]

Setting Up PIP Or Thermals:

The process for a thermal PIP or a PIP Sniper scope is near identical, however we need to make sure we use the BP_BasePIPScope parent class.

This can be found in Multiplayer FPS → Game → Weapons → Attachments → Optic → 8X.

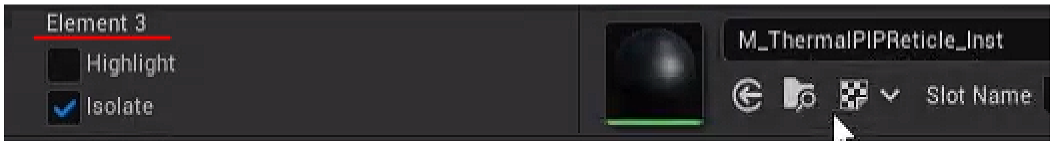
1. Right click BP_BasePIP and click create child class

2. Rename this to what your PIP sight is called and drag it into your folder created in Multiplayer FPS → Game → Weapons → Attachments → Optic

3. Open up the child class created and swap out its mesh.

4. Inside the new PIP scope mesh, create a Aim socket as usual

5. Inside the mesh, find out which material slot is responsible for your scope's glass. For example Element 3 on the SM_Thermal is responsible for the glass material. With this information, head over to the PIP actor class you created and in default variables in the PIPMaterialIndex, fill out the index to be the corresponding Element Number from your mesh material responsible for the glass material, in my case its 3 for SM_Thermal.



Element 3
Highlight
Isolate
M_ThermalPIPReticle_Inst
Slot Name

6. The rest of the steps are exactly the same as a normal sight

Setting Up PIP and Thermals Video Tutorial

Adding a Custom Character Mesh:

Firstly import or add your custom character into the project, this guide uses with the Unreal Engine Mannequin Skeleton Hierarchy

Reassigning the New Mesh Skeleton

1. Firstly, locate your new custom character mesh, right click and at the top of the drop-down menu, hit assign skeleton.

2. Search for SK_Mannequin within the Choose Skeleton browser, and make sure the bones match up.

3. Now open your mesh after the transfer and adjust any bones accordingly

Replacing BP FPS Character’s Mesh

Head over to Multiplayer FPS → Game → Blueprints → Core and Open BP_FPSCharacter.

1. In the viewport, select the Mesh (CharacterMesh0) and swap your new character mesh in. Repeat the same process for the FP Legs mesh

2. Compile and Save, test it out.

Replacing the FPS Arms

With your FPS arms, follow the same process with assigning its skeleton, after that head over to Multiplayer FPS → Game → Blueprints → Core and Open BP_FPSCharacter and select FP Arms, and replace its mesh.

You do not need to replace the Virtual Bones for the FP arms.

If you do not have the FP arms, head over to your mesh, right click → Asset Actions → Export.

1. Open the FBX in a 3D modeling program such as blender, select all faces except the arms and delete those.

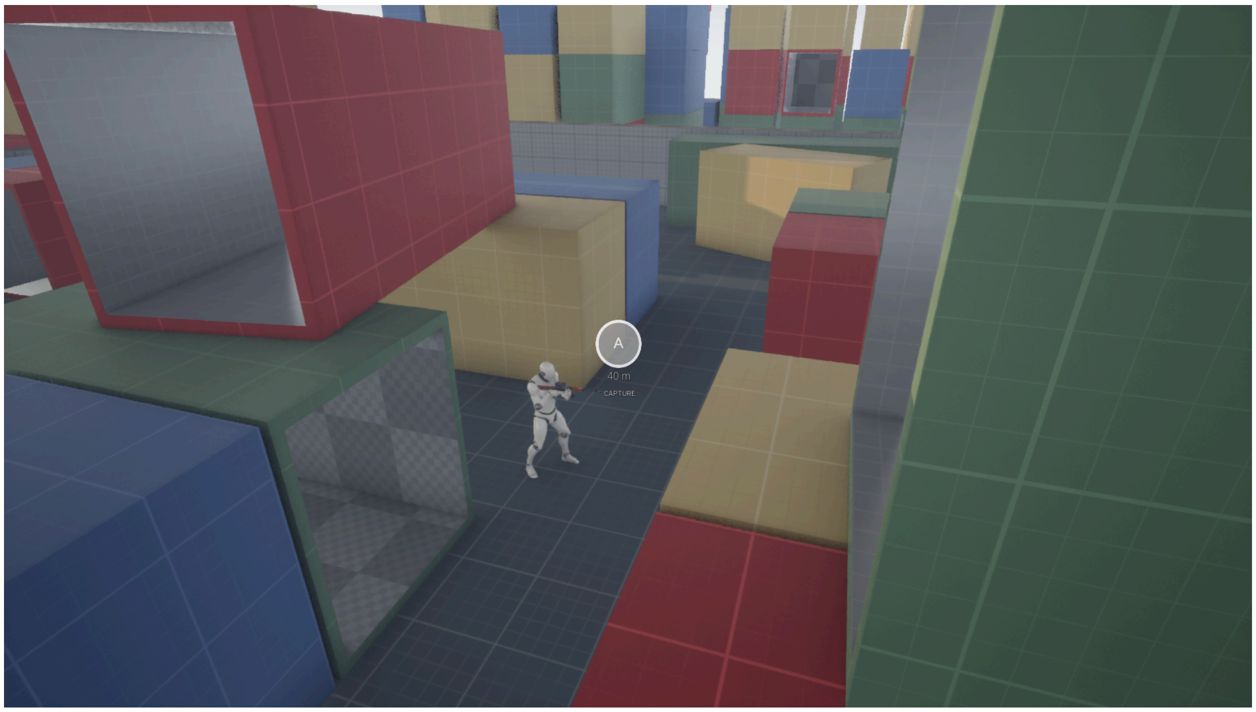
2. Export the Arms only mesh as FBX and on import, assign it the SK_Mannequin Skeleton.

Note: The video tutorial version provides greater info on how to separate the faces in blender. Timestamp is 4:00.

Setting Up A Character Video Tutorial

Setting Up New Maps

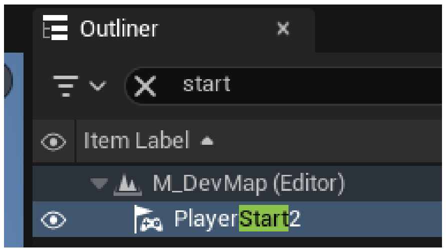
Adding new maps has become 100x easier. Read further to see how to integrate new maps



First thing you need to do is move your new map into the Maps folder, the directory for the maps folder is Multiplayer FPS → Maps.

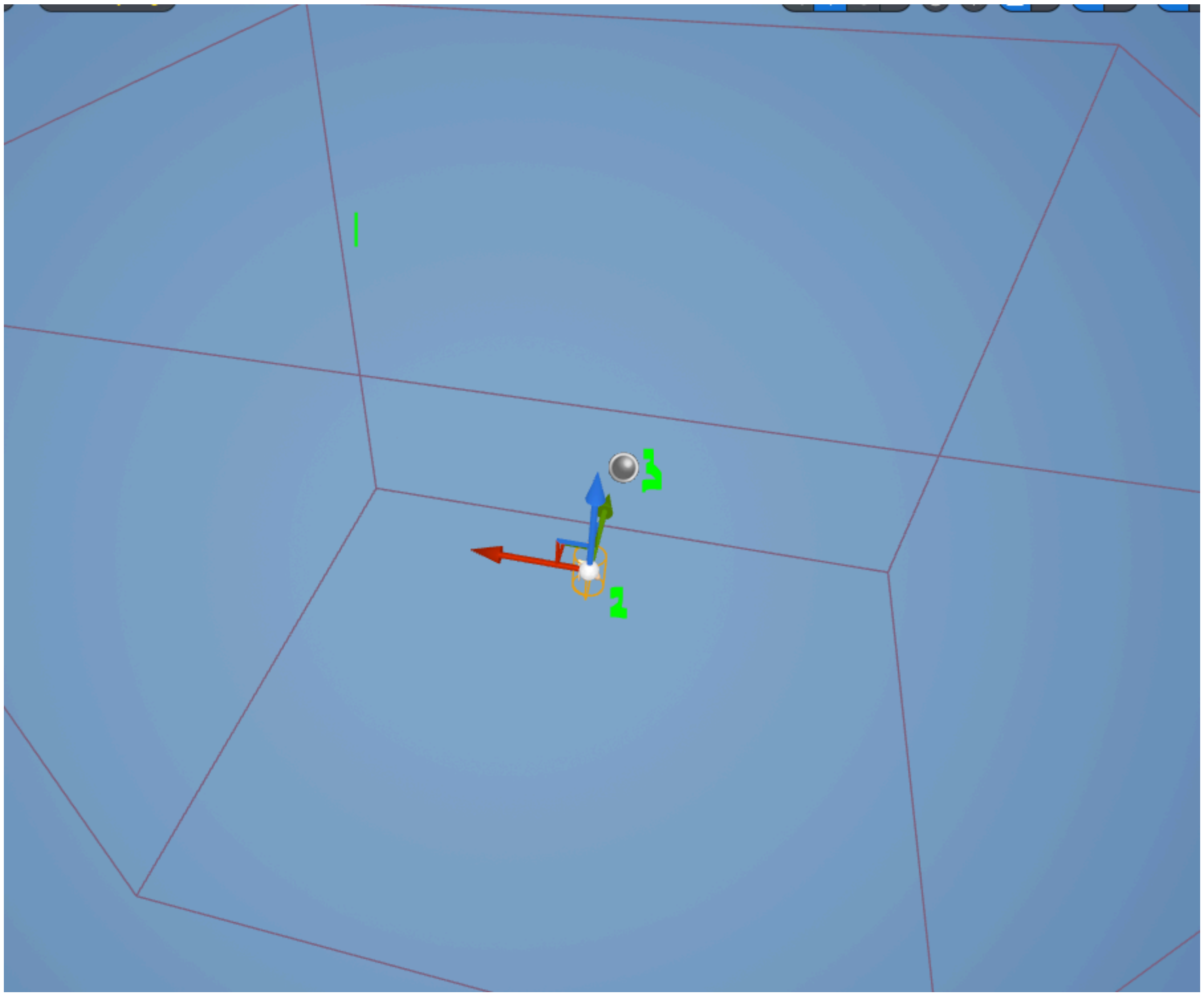
Within Maps, create a new folder, call it the name of your new map. Within that folder, create 2 sub folders, one for the day version of your map and another for a night version of your map.

1. In your new map, make sure to delete any existing PlayerStart, and only have 1 player start within your map. Ideally place this somewhere underneath your map.



NOTE: *PlayerStart is responsible for spawning in all the customization sphere, weapon preview locations etc at runtime. This means you no longer need to drag all those blueprints manually per map.*

Example:



1 = Post process, you can copy and paste this from one of the existing levels. It is not required to get customization working

2 = PlayerStart, this is responsible for handling where to spawn customization at run time. It is required to get customization working

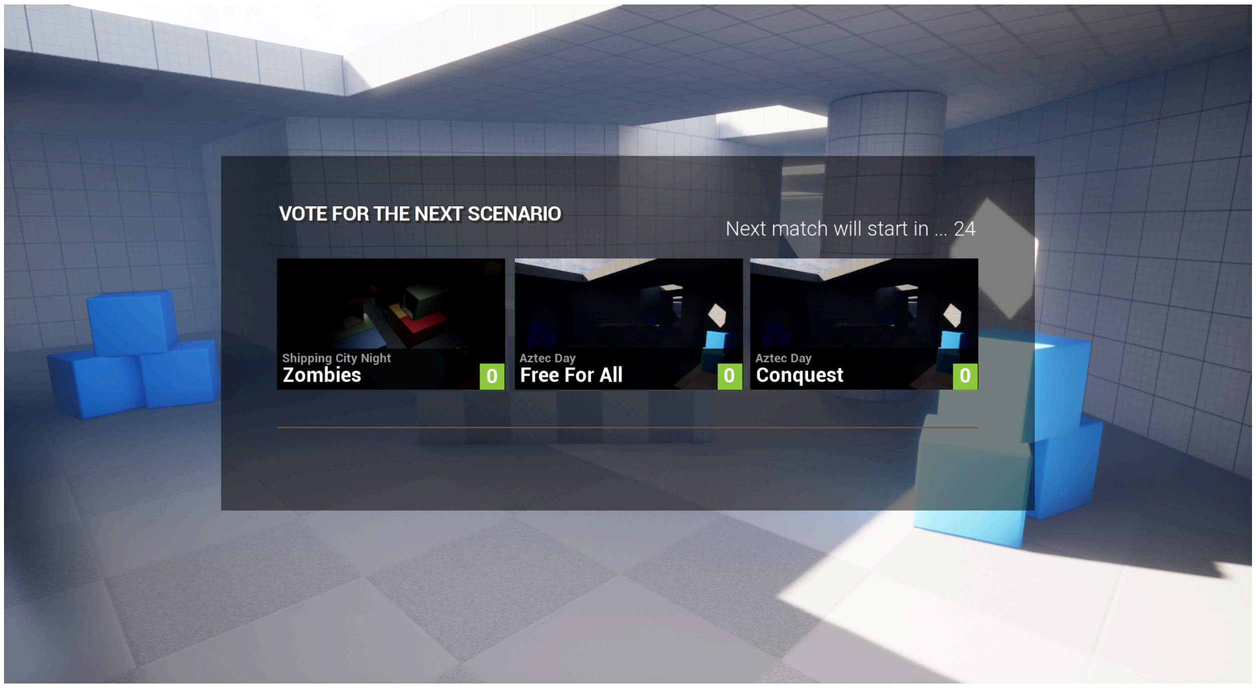
3 = SphereReflectionsCapture, this handles reflections and calculates them within the set radius, it is not required to get customization working.

2. You should have only 1 player start within your level and place that inside your BP_Customization.

3. The next step is to drag in BP_CustomizationSpawnPoints, position that somewhere inside the BP_Customization white sphere. You can adjust the individual weapon spawn points to your liking

Note: The easiest way to set this up is by copying the player start, BP_Customization and BP_Customization spawn points within another level such as the FFA Aztec and pasting it into your new level.

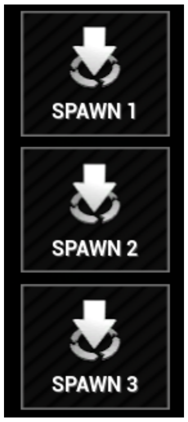
Adding the Game Over Camera



1. Head over to Multiplayer FPS → Game → Blueprints → Gamemodes and drag the BP_EndGameCamera somewhere within your level.

   This blueprint is a camera that the server switches to when the game is over, it pans over the entire map while scoreboards show and votes start.

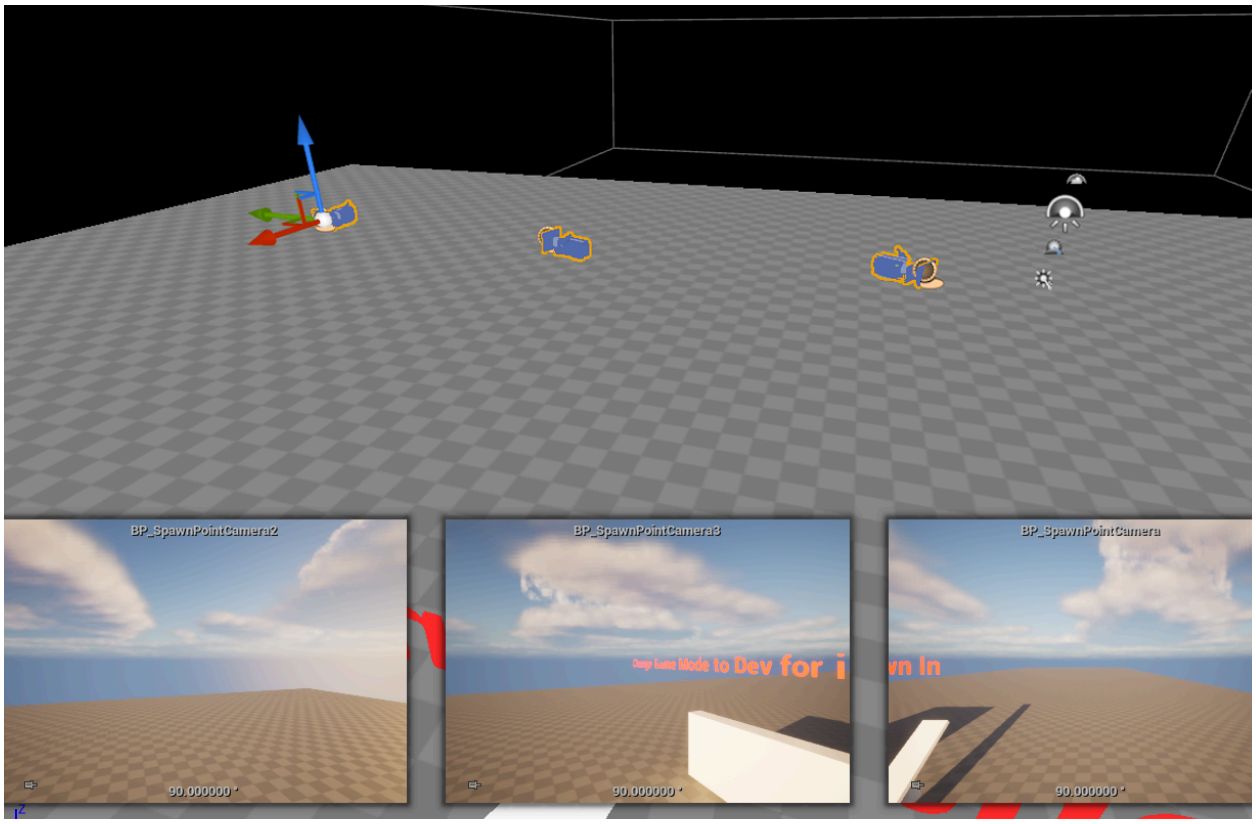
Adding Spawn Selection Cameras



Head over to MultiplayerFPS → Game → Blueprints → GameModes, drag in 6x BP_SpawnPointCamera into the level.

Position the first 3 camera's in Team A spawn point. Position them near the first 3 spawns where you have placed your team A spawn points. Make sure Team is set to Team A and the SpawnPoint int value is ascending, i.e. Camera 1 = 1, Camera 2 = 2, Camera 3 = 3.

Repeat this same process for the other 3x BP_SpawnPointCamera, only difference is make sure Team is TeamB.



Note: *Must be in a team based Game Mode, i.e TDM, Skirmish, Conquest. Will not work for Free for all based game modes.*

Adding New Map To Main Menu Map Selection

To add your map to the main menu list, you need to add it to the map's table, the main menu simply fetches every row within the table, filtered out by Gamemode and Time Of Day. What we need to do is tell the game we have a new map we want to include

1. Head over to Multiplayer FPS → Game → Maps. Open up DT_Maps

2. Within DT Maps, create a new Row by pressing the Add button at the top

3. Now, we need to fill out the new Rows information using the row editor. Set the Game Mode to your preferred map game mode - This doesn't set the actual map's game mode, rather it means what game mode will the map show up in.
4. Fill out the description, image and Day Or Night Variable

5. The LevelName field must be **EXACTLY** as your map name is in the unreal engine editor

6. The Command tab is used for server travel, i.e. for voting ect. Your command will be servertravel /(Your map's directory)

   For Example server travel for the TDM day map will be:

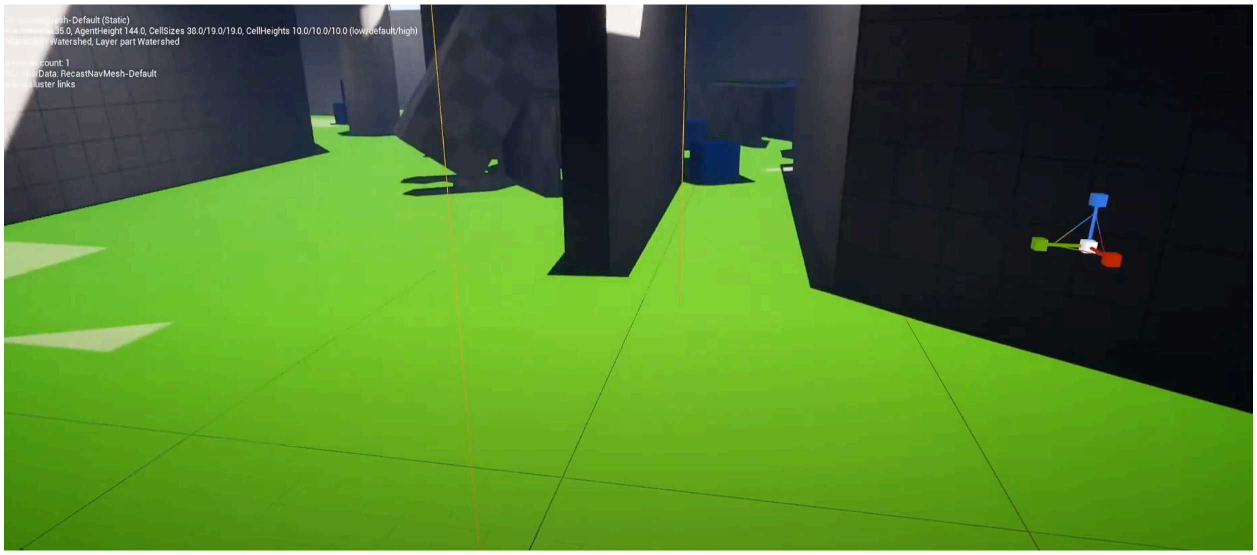
   servertravel /Game/MultiplayerFPS/Maps/Levels/TDM/M_TDM_Aztec_Day

7. The next step is to add a Map name, this can be called anything as its just a name for the Main Menu Map Selector Widget to show

Adding A Nav Mesh bounds volume

A Nav mesh bounds volume put simply is an area that AI are able to navigate. In Unreal engine, select the Get Content Menu at the Top Left, then Go to Volumes → Nav Mesh Bounds Volume

Scale the Nav Mesh Bounds Volume to the size of your map. Hit 'P' to visualize the area once it generates



```
[illegible]sh-Default (Static)
[illegible]35.0, AgentHeight 144.0, CellSizes 38.0/19.0/19.0, CellHeights 10.0/10.0/10.0 (low/default/high)
[illegible] Watershed, Layer part Watershed

[illegible]count: 1
[illegible]Data: RecastNavMesh-Default
[illegible]luster links
```

Once you have completed these steps, go to the main menu map and test out if your map is showing, along with creating a match with bots.

How to Change Maps Game Mode

Setting Up Free For All Maps

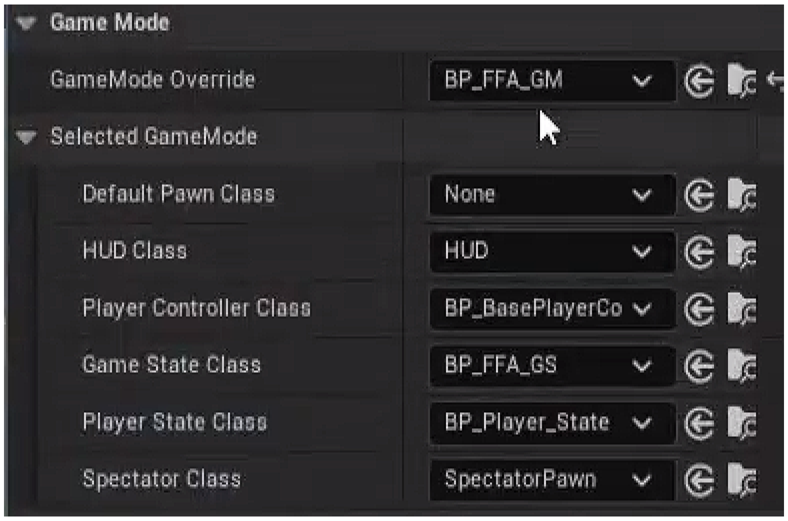
World Settings

First you need to follow the Map Setup guide, if you have already completed this step, you can follow along.

1. Head over to your maps world settings, this can be found on the right hand side of your screen

2. Change the GameMode Override to BP_FFA_GM

Example:



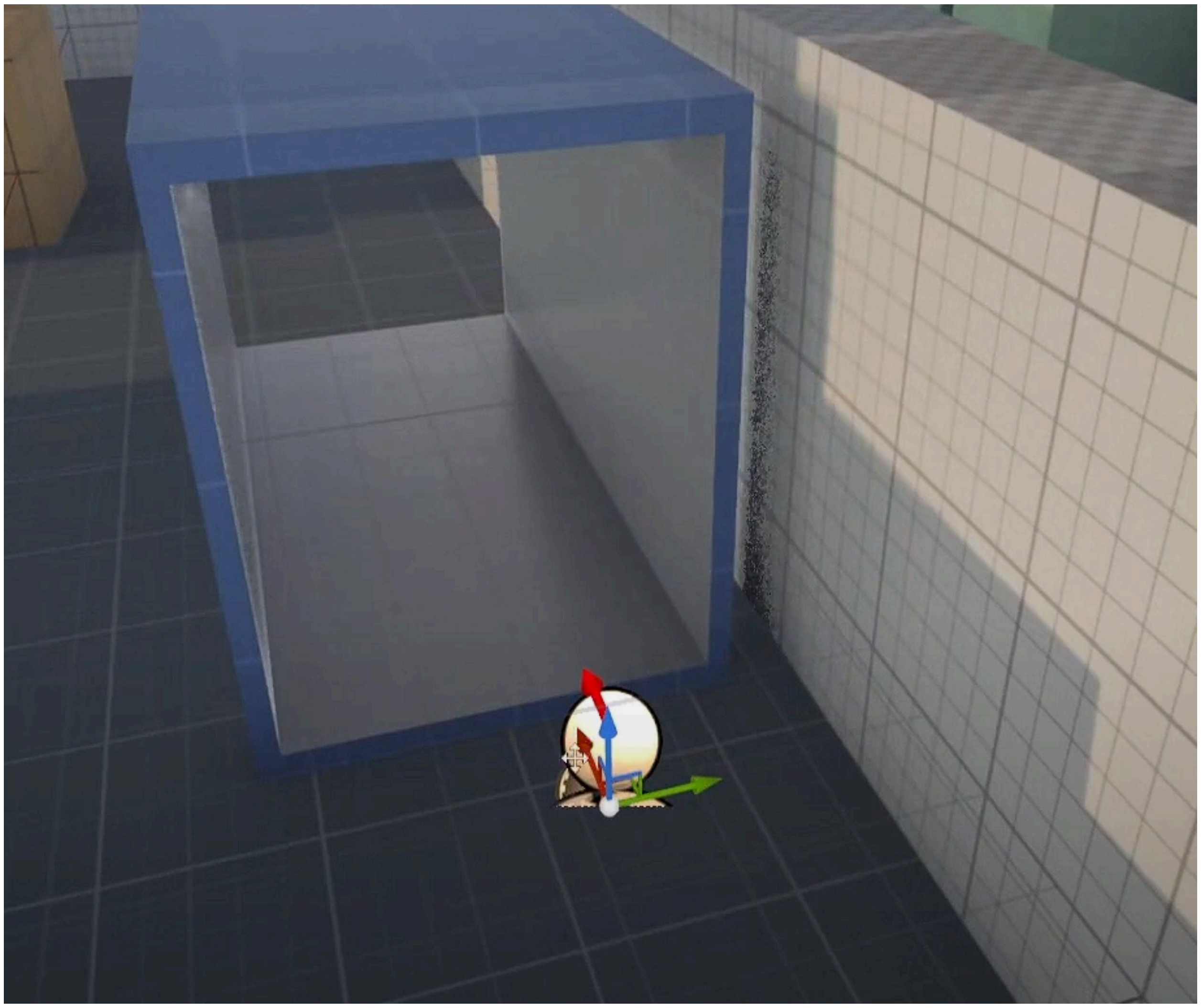
Game Mode

GameMode Override	BP_FFA_GM
Selected GameMode	
Default Pawn Class	None
HUD Class	HUD
Player Controller Class	BP_BasePlayerCo
Game State Class	BP_FFA_GS
Player State Class	BP_Player_State
Spectator Class	SpectatorPawn

Free For All Spawn Points

Next, add the Game Modes respective spawn points. Head over to Multiplayer FPS → Game → Blueprints → Core → Spawn Points → Spawns, and drag BP_SpawnPoint_FFA into your level.

You can add as many spawn points as you like but the maximum amount is 24. Its best to space out your spawn points evenly and apart to make AI's pathfinding work better


Setting up Free For all maps Video Tutorial

Setting Up Team Deathmatch Maps

World Settings

First you need to follow the Map Setup guide, if you have already completed this step, you can follow along.

1. Head over to your maps world settings, this can be found on the right hand side of your screen

2. Change the GameMode Override to BP_TDM_GM

Team Deathmatch Spawn Points

Next, add the Game Modes respective spawn points. Head over to Multiplayer FPS → Game → Blueprints → Core → Spawn Points → Spawns, and drag BP_SpawnPoint_A and BP_SpawnPoint_B into your level.

You can add as many spawn points as you like but the maximum amount is 12 per A and B. Its best to space out your spawn points evenly and apart to make AI's pathfinding work better

Drag the side A spawn points on one side of the map, and drag the Spawn Point Bs on the other side of your map.

Setting Up Team Deathmatch Maps Video Tutorial

Setting Up Skirmish Maps

World Settings

First you need to follow the Map Setup guide, if you have already completed this step, you can follow along.

1. Head over to your maps world settings, this can be found on the right hand side of your screen

2. Change the GameMode Override to BP_Skirmish_GM

Skirmish Spawn Points

Next, add the Game Modes respective spawn points. Head over to Multiplayer FPS → Game → Blueprints → Core → Spawn Points → Spawns, and drag BP_SpawnPoint_A and BP_SpawnPoint_B into your level.

You can add as many spawn points as you like but the maximum amount is 12 per A and B. Its best to space out your spawn points evenly and apart to make AI's pathfinding work better

Drag the side A spawn points on one side of the map, and drag the Spawn Point Bs on the other side of your map.

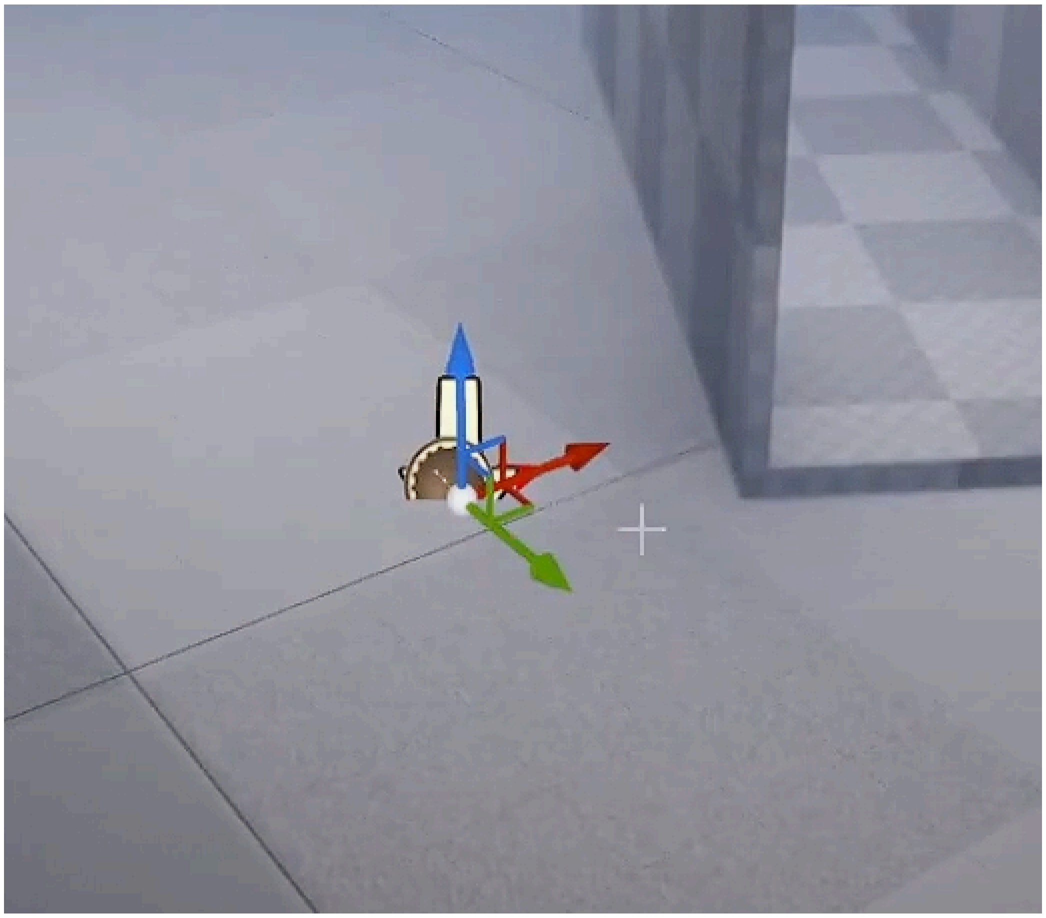
Skirmish Capture Points

Head over to Multiplayer FPS → Game → Blueprints → Game Modes→ Skirmish → Spawns

1. Drag BP_DominationPointSpawnPointAlpha into your level
2. Drag BP_DominationPointSpawnPointBeta into your level
3. Drag BP_DominationPointSpawnPointCharlie into your level
4. Drag BP_CachePoint into your level

Note: The location you drag the Point Spawners is where the Skirmish Capture Points will spawn.

Setting Up Skirmish Maps Video Tutorial



Setting Up Domination Maps

World Settings

First you need to follow the Map Setup guide, if you have already completed this step, you can follow along.

1. Head over to your maps world settings, this can be found on the right hand side of your screen

2. Change the GameMode Override to BP_Conquest_GM

Domination Spawn Points

Next, add the Game Modes respective spawn points. Head over to Multiplayer FPS → Game → Blueprints → Core → Spawn Points → Spawns, and drag BP_SpawnPoint_A and BP_SpawnPoint_B into your level.

You can add as many spawn points as you like but the maximum amount is 12 per A and B. Its best to space out your spawn points evenly and apart to make AI's pathfinding work better

Drag the side A spawn points on one side of the map, and drag the Spawn Point Bs on the other side of your map.

Domination Capture Points

Head over to Multiplayer FPS → Game → Blueprints → Game Modes→ Skirmish → Spawns

1. Drag BP_DominationPointSpawnPointAlpha into your level

2. Drag BP_DominationPointSpawnPointBeta into your level

3. Drag BP_DominationPointSpawnPointCharlie into your level

Note: The location you drag the Point Spawners is where the Skirmish Capture Points will spawn.

Setting Up Domination Map Video Tutorial

Setting Up Zombies Maps

World Settings

First you need to follow the Map Setup guide, if you have already completed this step, you can follow along.

1. Head over to your maps world settings, this can be found on the right hand side of your screen

2. Change the GameMode Override to BP_Zombies_GM

Zombie Mode Player Spawn Points

Next, add the Game Modes respective spawn points. Head over to Multiplayer FPS → Game → Blueprints → Core → Spawn Points → Spawns, and drag BP_SpawnPoint_FFA into your level.

You can add as many spawn points as you like but the maximum amount is 24. Its best to space out your spawn points evenly and apart to make AI's pathfinding work better

Zombies Wave Spawners

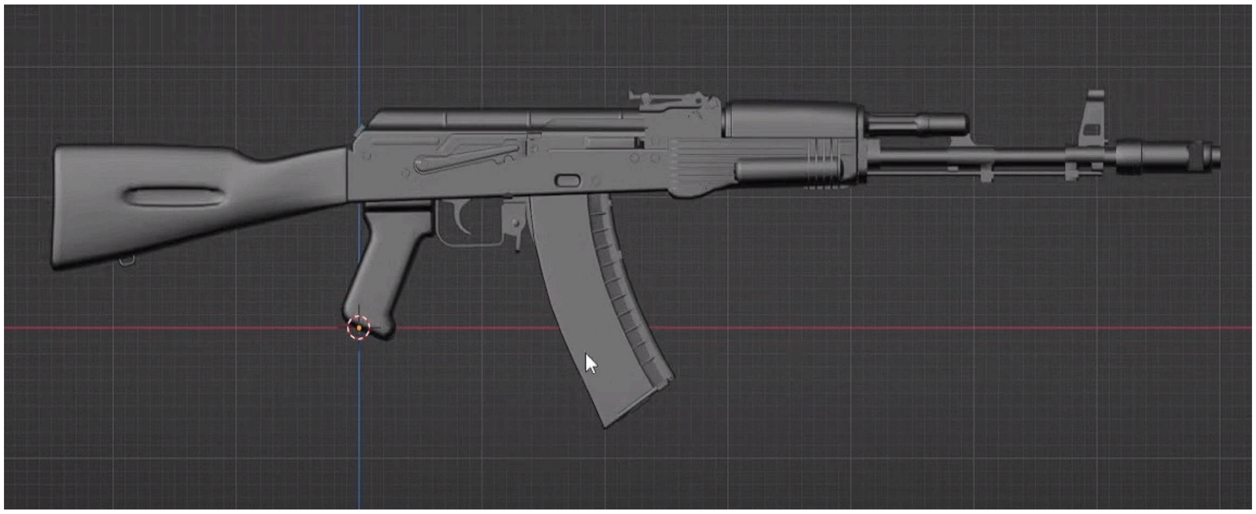
Head over to Multiplayer FPS → Game → Blueprints → Core → AI → Zombie → Wave and drag in BP_WaveBasedSpawner. This is the brain of the wave system and only 1 is required.

Next drag in BP_WaveSpawnPoints into your level, these are the spawn points for the new zombies each wave. You can drag as many as you'd like into the level.

Setting Up Zombies Map Video Tutorial

Adding Custom Weapons

Prepping Our Gun



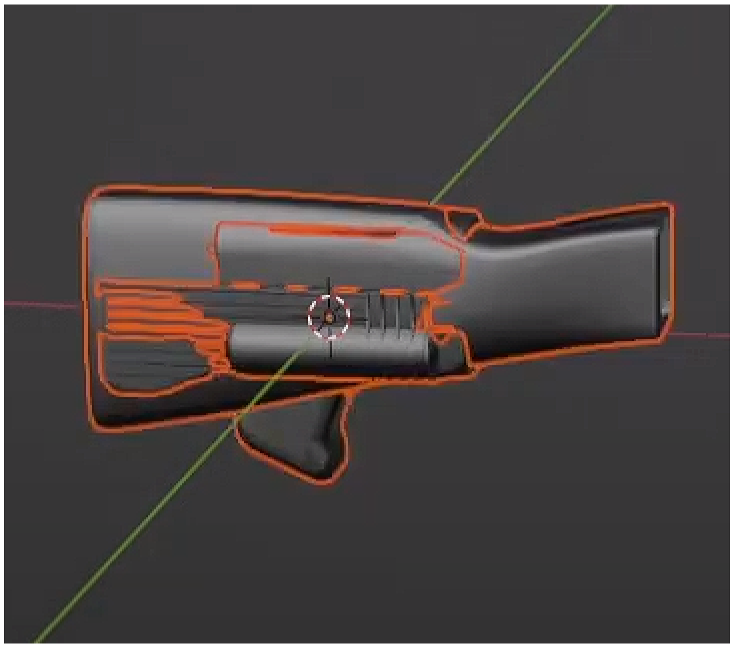
To use the Weapon with the customization system, we need to remove the non essential components of the weapon, like pistol grips, stock, handle, front and rear sights and the muzzle device.

In edit mode in blender, we remove each part and separate them as its own static mesh which we will later import into unreal engine.

In blender, make sure the weapon's origin is at its pistol grip, and that the weapon faces **POSITIVE X** axis, with **Y positive** as its upward axis.

The end result should be something similar to the example below

Example:



Our Weapon parts are split off into their own individual objects and then later imported into unreal engine as static meshes.

For the base frame of the weapon, we want to export it out of blender into unreal engine using Positive X as its forward axis, and Positive Y as its upward axis. This is so that it faces the correct way in unreal engine.

The same process can be repeated for exporting the weapon parts as static meshes.

For more information on how to do this process, follow the video tutorial from 0:40 onward.

Ultimate Multiplayer FPS Kit: Setting Up A Weapon | Unreal Engine 5

Importing your Weapon Files

Import your weapon's frame rig as a skeletal mesh, whereas import the weapon default parts as static meshes.

Setting up the Weapon Material

1. Head over to Multiplayer FPS → Game → Weapons → Primary / Secondary (depending on weapon)

2. Create a new folder for your weapon, i.e. Five-Seven or P90.

3. Within the new Folder, create new folders to match the hierarchy.

   →Animations → FP, TP, Weapon Animations
   →DefaultParts
   →Mesh → Materials → Textures

   Note: You can refer to the pre existing weapon's folders to match the hierarchy

4. Import your mesh to the mesh root folder, and drag your textures to the textures folder

5. Import your static mesh default parts into the default part folder

6. Head on over to MultiplayerFPS → Game → Weapons → MasterMaterials, right click and create a material instance. Rename this to whatever your new Weapon Name, i.e. M_P90_Body_Inst. Drag this Instance into your Materials folder for your Weapon.

6. Open up the material and fill out your Ambient Occlusion, Roughess, Metallic, Albedo and Normal Map textures. Then head over to your new Optic Mesh and apply the texture to its respective Material Slots.

7. Do the same thing for any parts or you can apply the pre-existing material onto the default parts if they share the same textures.

Setting up the First Person And Third Person Animations

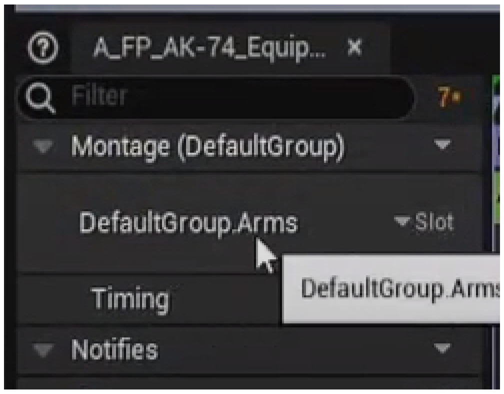
Import your Animations for First Person and third person into their respective folders. You can find these folders you created at the hierarchy MultiplayerFPS → Game → Weapons → P90 → Animations.

Alternatively, you may share the same animations between FP and TP if you wish to.

The next step you need to do is create animation montages for your main action animations.

1. For your Reload, Reload Empty, Unequip, Equip and Melee animations, right click and create montage.

2. In each montage, make sure the default slot is set to Arms. This is the slot responsible for handling procedural movements such as swaying, IK, ADS reloads ect



3. For Third person, create montages again for Reload, Reload Empty, Unequip, Equip and Melee.

4. However this time, set the montages default slot to upper body for the Third Person montages

Adding Animation Notifies

Now the next step is to add notifies into our animations. Notifies are simply a custom event for animations. This allows us to call logic at a specific time in an animation.

For example, we use notifies to play sounds for reloads, camera shakes in weapon movements and handling or hiding the weapon when swapped

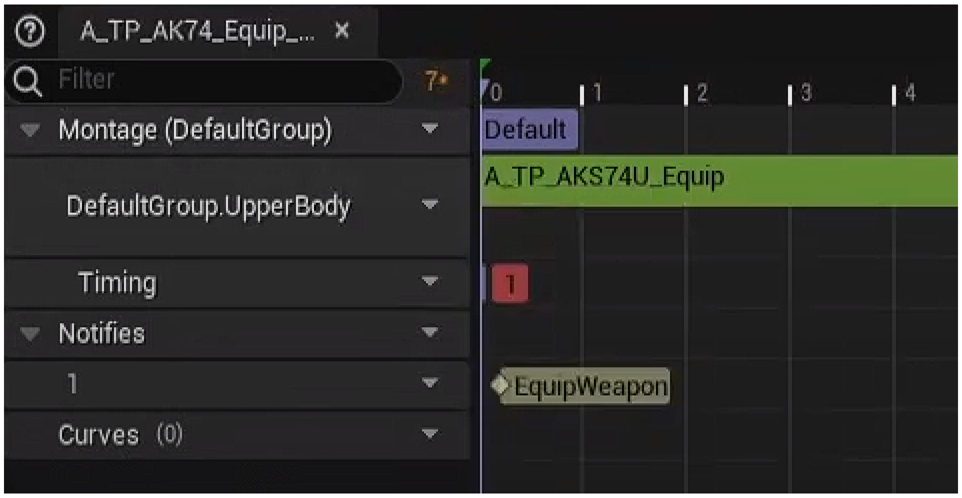
Third Person Notifies

1. In the Equip Montage for the Third Person animation, at the beginning of the track, right click on the notify track → Add Notify → Skeleton Notify → Equip Weapon

2. Drag the notify to the beginning of the Equip Montage

3. Now in the Unequip Montage for the Third Person animation, right click the notify track → Add Notify → Skeleton Notify → Unequip weapon. Drag this notify toward the end of the Unequip Montage sequence

Example:



First Person Notifies

First we will start with the essential Notifies needed to get the necessary Montages working, then we will move onto the Notifies that are only needed for effect, such as sounds and camera shake.

Here is a list of the essential Notifies we will use for FP montages
→ RemoveMag
→ InsertMag
→ ReloadComplete

Implementing these notifies

1. Open up the First Person reload montage, at the point in the animation where the magazine is removed and out of view of the player, add the RemoveMag Notify

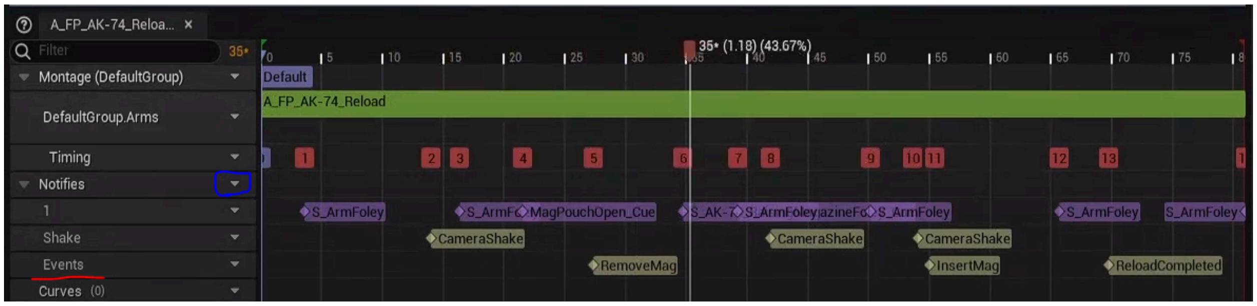
2. At the point in the animation where the Magazine is Reinserted, add the notify InsertMag

3. At the end of the animation Sequence, add the notify ReloadComplete.

To add a notify

→ Right click the notify track → Add Notify → Skeleton Notify → e.g. Reload Complete

Example of Reload Montage:



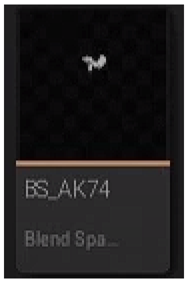
We only need to focus on the Events Track. To add a track, select the drop down arrow highlighted in blue → Add New Track. You can then name it accordingly. Tracks allow you to layer notifies on top of one another, making the Notify tracks easier to read.

You can repeat this same process for the Reload Empty.

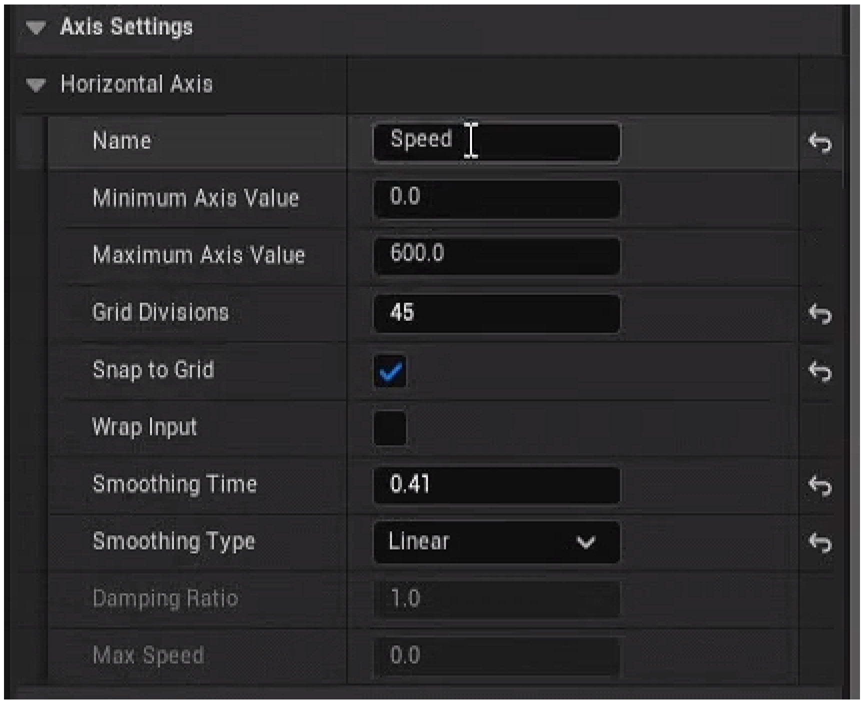
[If you are having trouble at this step, watch the video tutorial version]

Creating the FP Blendspace

Head over to MultiplayerFPS → Game → Weapons → Primary / Secondary → Your Weapon



Within the folder, create a new 1D Blendspace, you can find that by right clicking in the content browser → Animation → Legacy → BlendSpace 1D. The blendspace will be responsible for handling our transitions between Idle, Walk and Sprint.



Axis Settings

Horizontal Axis

Name	Speed
Minimum Axis Value	0.0
Maximum Axis Value	600.0
Grid Divisions	45
Snap to Grid	✓
Wrap Input	
Smoothing Time	0.41
Smoothing Type	Linear
Damping Ratio	1.0
Max Speed	0.0

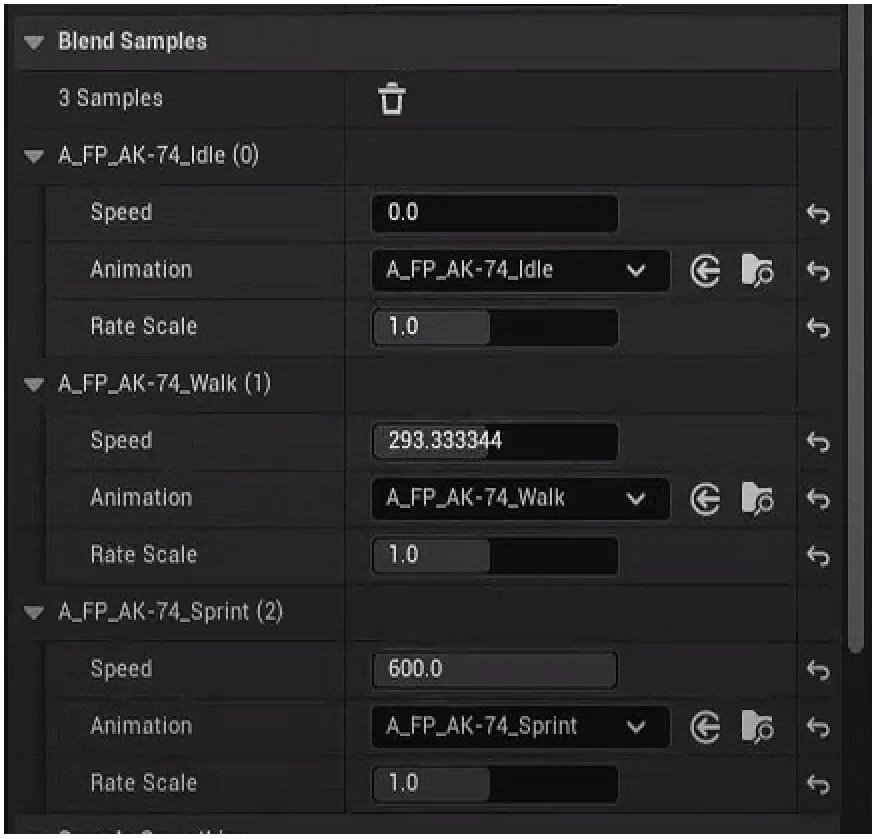
Adjusting the blendspace settings

1. Create an axis called Speed

2. Then give the Horizontal Axis settings a minimum value of 0, and a maximum value of 600.

3. Set the grid divisions to 45

4. Adjust the smoothing time to something like 0.41, you can adjust this based on how smooth you want the blending to be between animations.



Blend Samples

3 Samples	
A_FP_AK-74_Idle (0)	
Speed	0.0
Animation	A_FP_AK-74_Idle
Rate Scale	1.0
A_FP_AK-74_Walk (1)	
Speed	293.333344
Animation	A_FP_AK-74_Walk
Rate Scale	1.0
A_FP_AK-74_Sprint (2)	
Speed	600.0
Animation	A_FP_AK-74_Sprint
Rate Scale	1.0

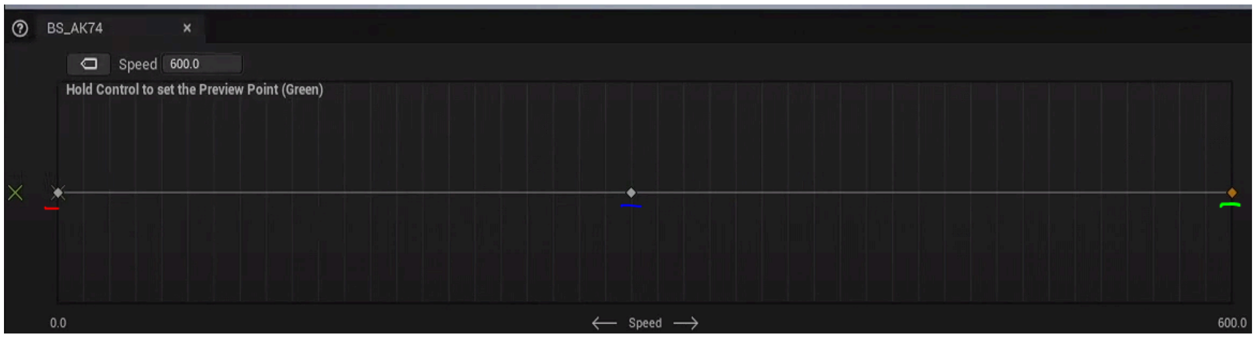
Adding animations into our blendspace:

→ First, add an idle animation at 0 speed. You can do this by searching for your idle animation in the asset browser, and drag it into the blendspace sequence.

→ Then, add the Walk animation at the midpoint of the blendspace sequence

→ Finally, add the Sprint animation at the end of the blendspace sequence

Example Blendspace Sequence:



Hold Control to set the Preview Point (Green)

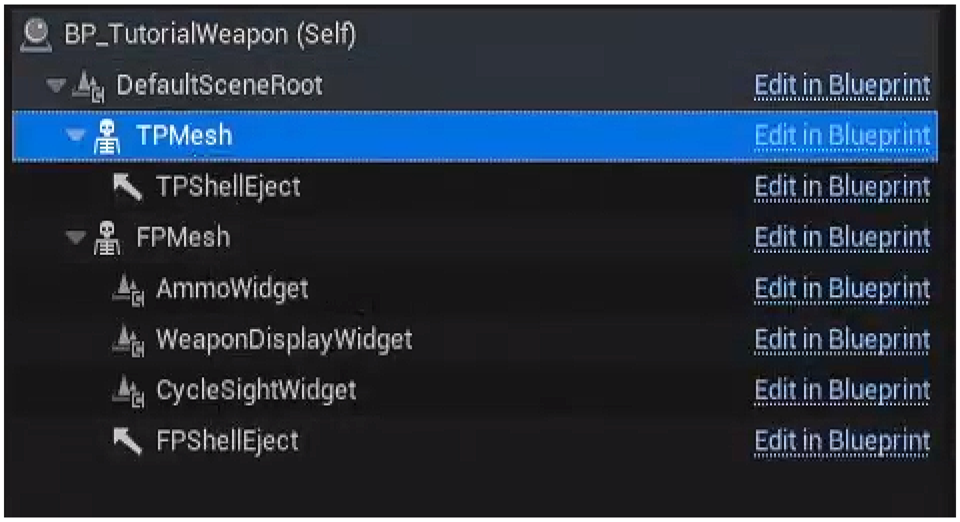
Creating the New Weapon Class

Head over to MultiplayerFPS → Game → Blueprints → Core, right click the BP_WeaponBase and make a child blueprint class.

Rename it to your new weapon class,for example BP_P90 or BP_M249. Move the child weapon class into its respective folder.

Setting up the Child Weapon Class

Inside the new weapon class you have created, we need to plug in our weapon's skeletal mesh frame for the TP and FP meshes.


BP_TutorialWeapon (Self)
- DefaultSceneRoot — Edit in Blueprint
  - TPMesh — Edit in Blueprint
    - TPShellEject — Edit in Blueprint
  - FPMesh — Edit in Blueprint
    - AmmoWidget — Edit in Blueprint
    - WeaponDisplayWidget — Edit in Blueprint
    - CycleSightWidget — Edit in Blueprint
    - FPShellEject — Edit in Blueprint

1. First, select the TP mesh and plug in your new Skeletal Mesh Weapon, and repeat the same process for the FP mesh.

2. The next step is adjusting the TPShellEject and the FP Shell Eject, in the components tab select the arrow components and adjust them to line up with your weapon's slide. The process is the exact same for the FP and TP.

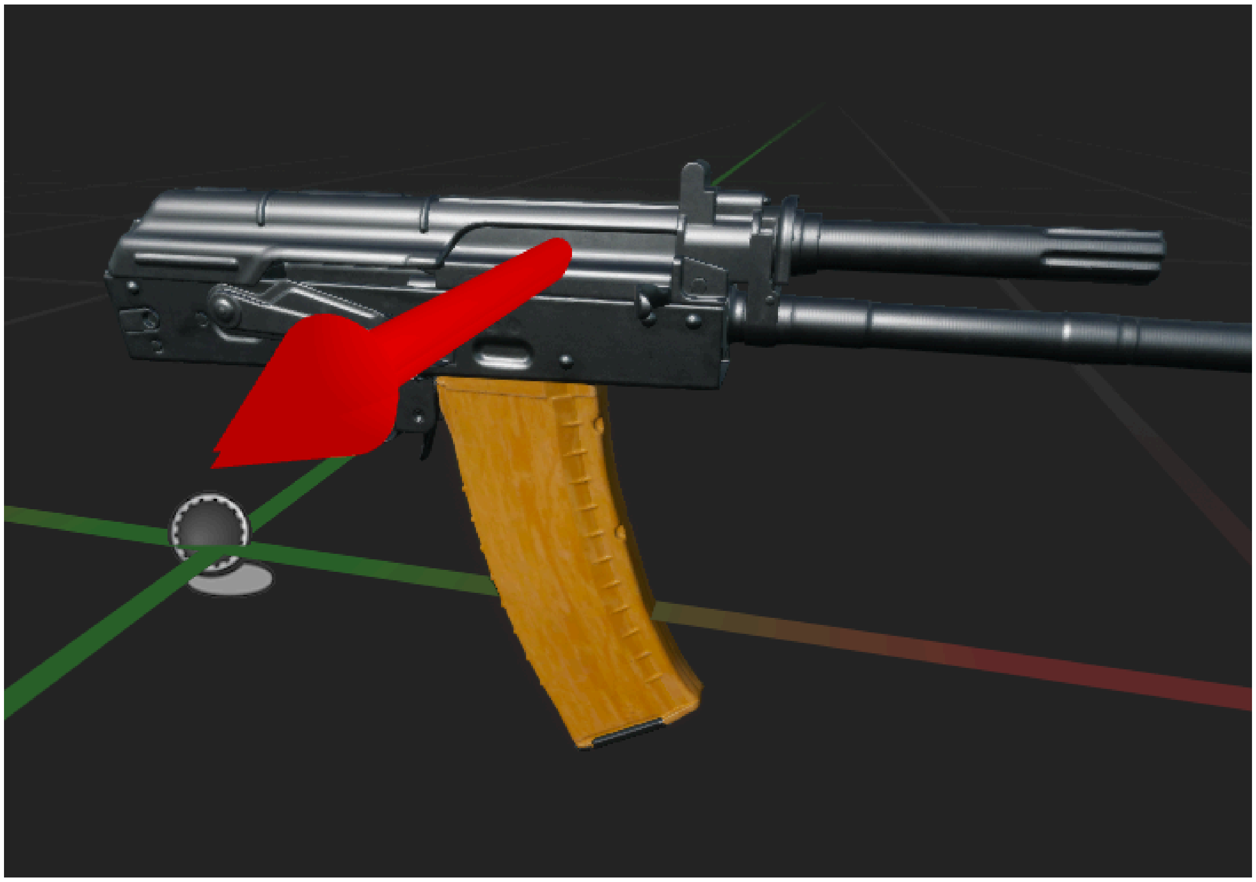
3. Once you have adjusted the Shell Ejects, the next step is to adjust the AmmoWidget and Weapon Display widget. You can adjust the ammo widget anywhere on your weapon, however I recommend it is somewhere near the magazine of your gun.

4. The same process as the previous step can be applied to the Weapon Display Widget, I recommend positioning it near the barrel of the weapon.

Note: *If you are stuck at any step of the Weapon Child Class setup, refer to either the video tutorial or reference the preset weapon classes included with the pack.*

www.youtube.com/watch?v=zpL9pkQ0fHo&t=1020s

Example:



Adjusting Weapon Characteristics

Weapon Settings

1. For Damage Type, select either one of the presets, or you can create your own. To create your own, head over to MultiplayerFPS → Game → Blueprints → Data → DamageTypes. Duplicate one of the presets and fill out an icon and a name for your weapon. Then simply use that new DamageType In the Weapon Settings.

2. The length of the weapon is simply a unit of measurement to calculate the barrel length of the gun, it's used for procedural weapon blocking. The Value 80 works best for most

AR styled barrel guns, and a value of 120 works well for long barreled weapons like snipers or DMRs.

3. Fill out your damage, noise range, loudness, weapon range and velocity values

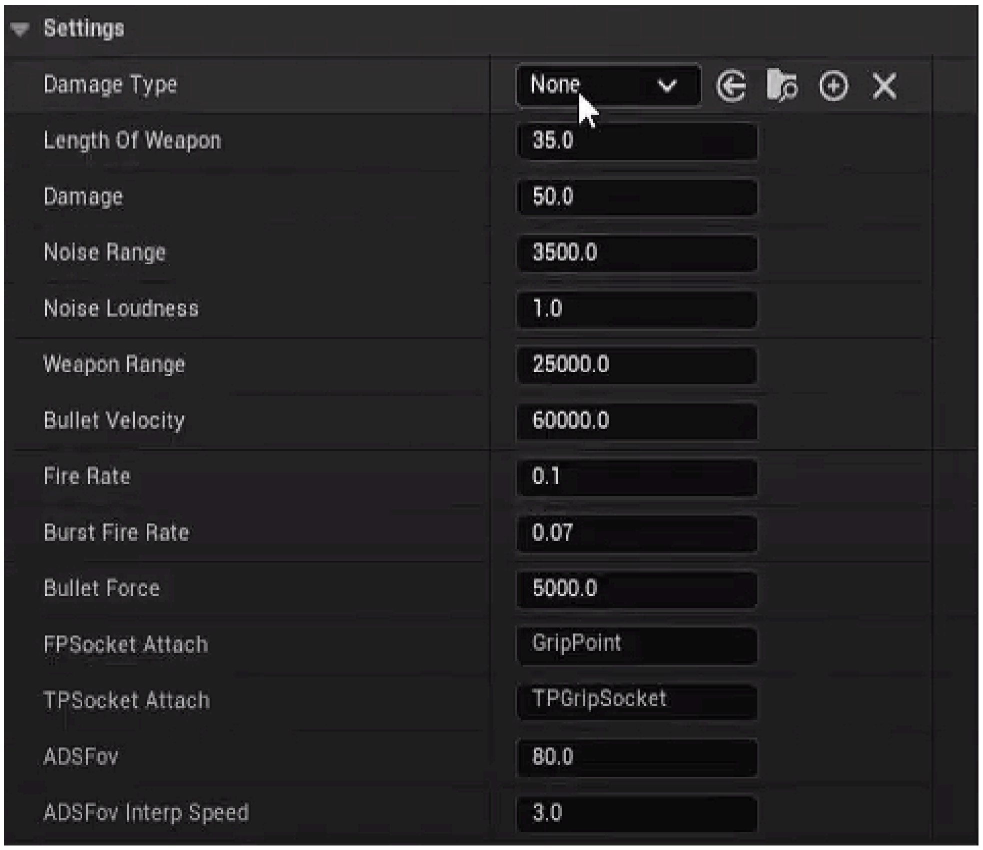
4. Adjust your weapon fire rates, burst and force.

5. The next step is to create our sockets for attaching the weapon. Head over to MultiplayerFPS → Game → Characters → Mannequin → FP → Mesh. Select the hand_r right click and create a new socket. Name this socket according to your weapon using the prefix Socket_. Use the same socket and fill in the FP, repeat the same process for the TP socket and fill out the TP socket parameter.

   Note: If you are struggling with this step, follow the video tutorial

6. You can adjust your FOV and FOV interpolation speed depending on your weapon type and weight.

Example:



Settings

Damage Type	None
Length Of Weapon	35.0
Damage	50.0
Noise Range	3500.0
Noise Loudness	1.0
Weapon Range	25000.0
Bullet Velocity	60000.0
Fire Rate	0.1
Burst Fire Rate	0.07
Bullet Force	5000.0
FPSocket Attach	GripPoint
TPSocket Attach	TPGripSocket
ADSFov	80.0
ADSFov Interp Speed	3.0

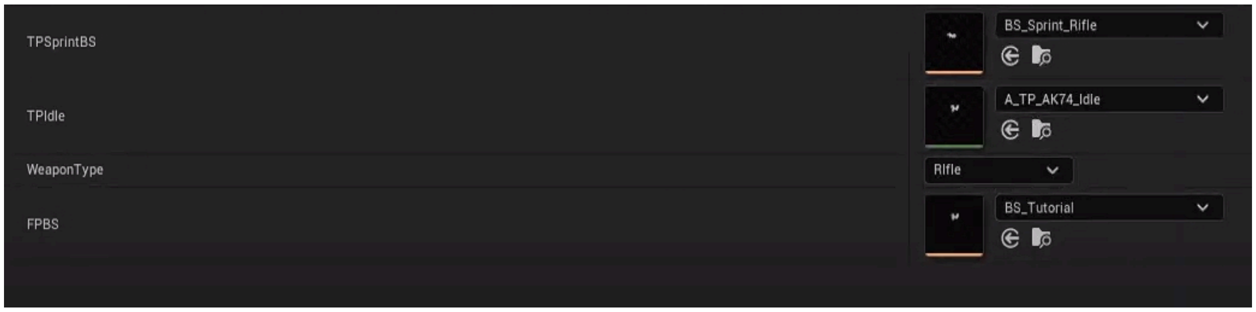
Weapon Type

1. Fill out your Weapon UI image

2. Add a GameName, this name will be used for things like the equipped Weapon Slot in the HUD hotbar

3. For the animation slot, head over to MultiplayerFPS → Game → Blueprints → Data and open up DT_AnimationData. Within the table, add a new row and give the Row Name a value thats +1 of the previous row. For example if the previous row is 4, name the new row 5. Fill out your Row's animation details. Once you completed filling out the row, plug in your Row Name's number into the Animation Slot in the Weapon blueprint.

Example:



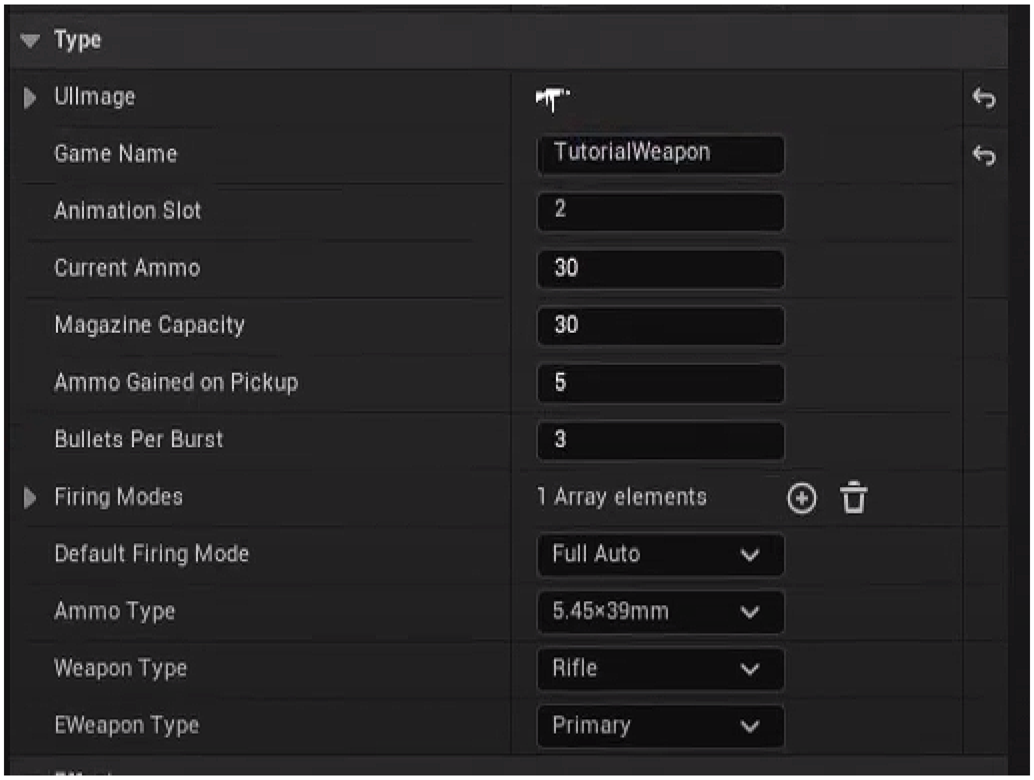
TPSprintBS: BS_Sprint_Rifle
TPIdle: A_TP_AK74_Idle
WeaponType: Rifle
FPBS: BS_Tutorial

4. Adjust your weapon's Current ammo and Max ammo. Current ammo means the amount of magazine the gun has per magazine, and max ammo is the maximum ammo the player can store in its pouch

5. Fill out your firing mode array and the ammo type, along with the weapon type and EWeapon Type slot.

   The EWeapon Type slot identifies if the weapon takes up a Primary Or Secondary slot

Example:



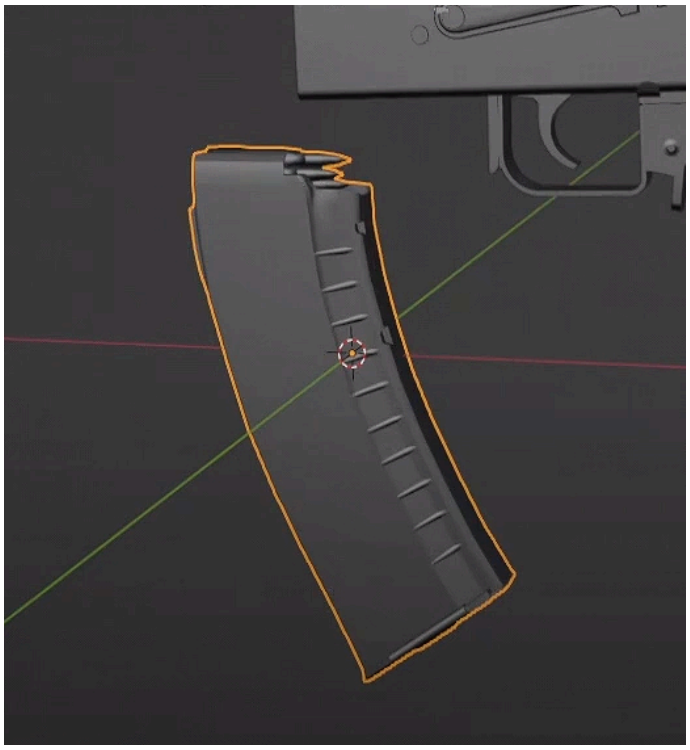
Type	
UIImage	
Game Name	TutorialWeapon
Animation Slot	2
Current Ammo	30
Magazine Capacity	30
Ammo Gained on Pickup	5
Bullets Per Burst	3
Firing Modes	1 Array elements
Default Firing Mode	Full Auto
Ammo Type	5.45×39mm
Weapon Type	Rifle
EWeapon Type	Primary

Adding Effects to the New Weapon Class

In blender, head over to your model and select the magazine in edit mode. We want the magazine to be a separate skeletal mesh to use for dropping the magazine in game

Note: *If you are struggling on this step, head on over to the video tutorial on how to remove the magazine*

Example:



Export this as a .fbx, making sure it faces the Positive X axis with Y positive as its upward axis.

Setting up the Magazine and Weapon Physics

→ Import the Magazine as a skeletal mesh. Unreal Engine will automatically generate a bone for it, converting it into a Skeletal mesh, along with a physics asset and Skeleton.

→ Head over to MultiplayerFPS → Game → Weapons → Primary → YourWeapon → Mesh, inside this directory, create a new folder called Magazine and move your Magazine files in here.

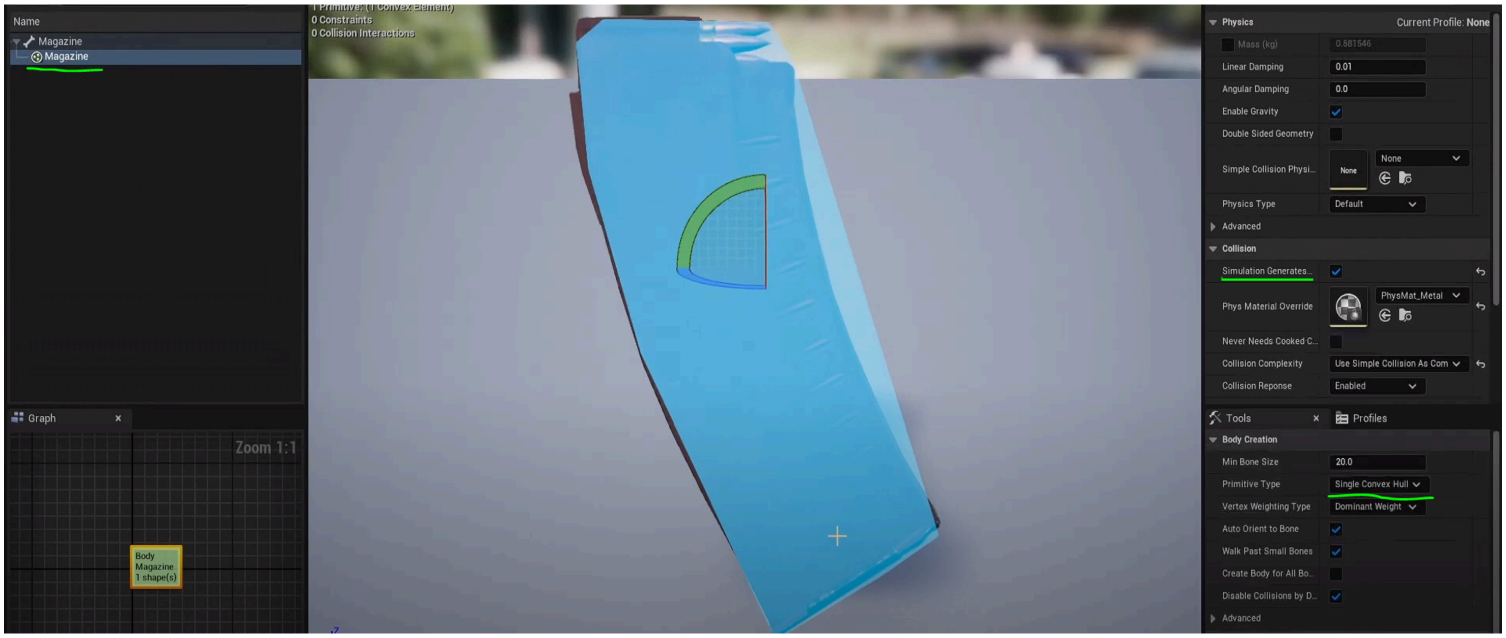
1. Open up your Physics asset for the magazine, remove any premade collisions in the left hand side skeleton tree.

2. Select the settings button → Show All bones → Select the Bone → In the right hand side, Select Single Convex Hull and generate bodies

3. Select your newly created body, and in the details panel, enable simulation generates hit overlaps. The next step is to apply a physics material, you can use PM_Metal.

4. Repeat the exact same process for the physical material of your own weapon frame

Example:



Physics — Current Profile: None

Mass (kg)	0.581546
Linear Damping	0.01
Angular Damping	0.0
Enable Gravity	✓
Double Sided Geometry	
Simple Collision Physi...	None
Physics Type	Default
Advanced	

Collision

Simulation Generates...	✓
Phys Material Override	PhysMat_Metal
Never Needs Cooked C...	
Collision Complexity	Use Simple Collision As Com
Collision Reponse	Enabled

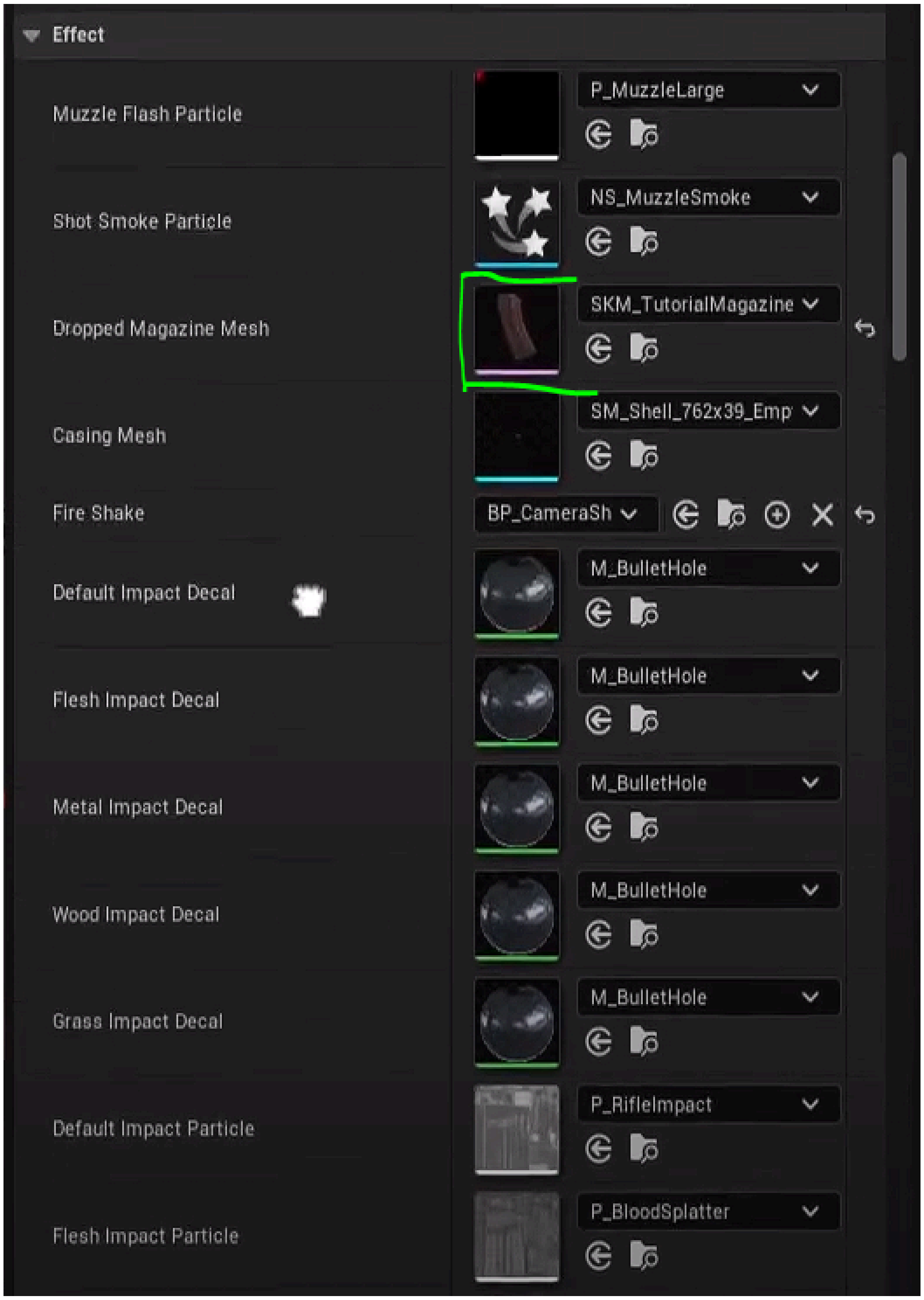
Body Creation

Min Bone Size	20.0
Primitive Type	Single Convex Hull
Vertex Weighting Type	Dominant Weight
Auto Orient to Bone	✓
Walk Past Small Bones	✓
Create Body for All Bo...	
Disable Collisions by D...	✓
Advanced	

Once you have completed the new bodies for the physics asset, you can return to the weapon and fill out the new Skeletal Mesh for the Dropped Magazine mesh.

For the rest of the parameters, you can either use the preset particles and decals, or add your own.

Example:



Effect

Muzzle Flash Particle	P_MuzzleLarge
Shot Smoke Particle	NS_MuzzleSmoke
Dropped Magazine Mesh	SKM_TutorialMagazine
Casing Mesh	SM_Shell_762x39_Emp
Fire Shake	BP_CameraSh
Default Impact Decal	M_BulletHole
Flesh Impact Decal	M_BulletHole
Metal Impact Decal	M_BulletHole
Wood Impact Decal	M_BulletHole
Grass Impact Decal	M_BulletHole
Default Impact Particle	P_RifleImpact
Flesh Impact Particle	P_BloodSplatter

Plugging in our new Animations

Plugging in Weapon Animations

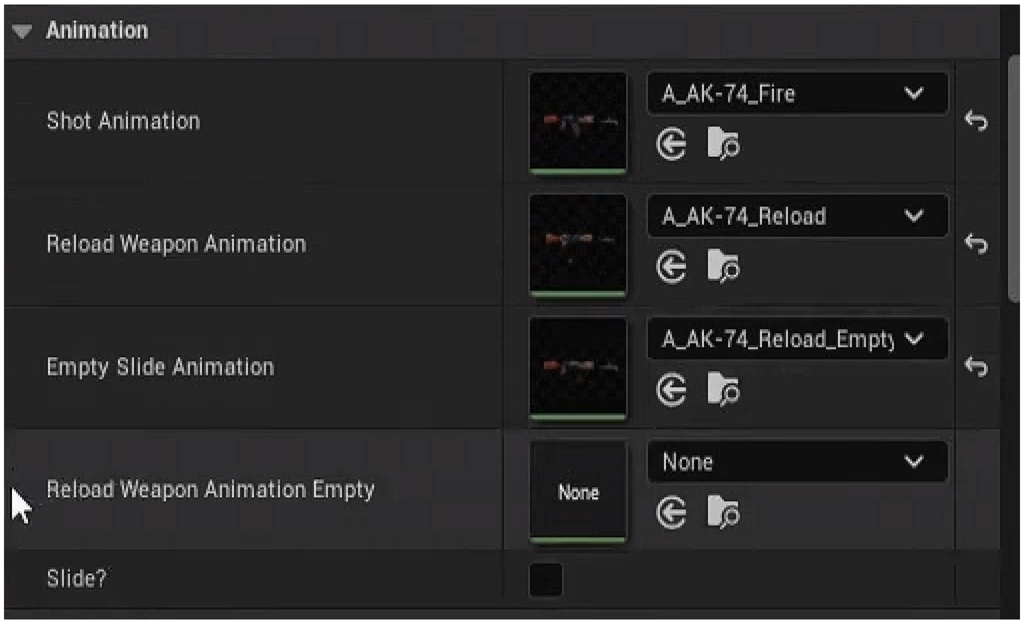
Now we can move onto adding our Weapon FP and TP animation montages that we created earlier

Head on over to the Animations tab within the Weapon Class you created. The first animations we are going to fill in is:

→Weapon Shot Animation
→Weapon Reload Animation
→Weapon Reload Empty Animation

Optional:
→ Empty Slide Animation
Note: *this optional animation only applies to weapon's that feature a bolt system, for example pistols have their slide locked back when the magazine is empty. If this is the case for your gun, enable Slide? and plug in your own animation for the weapon slide locked back*

Example:



Animation

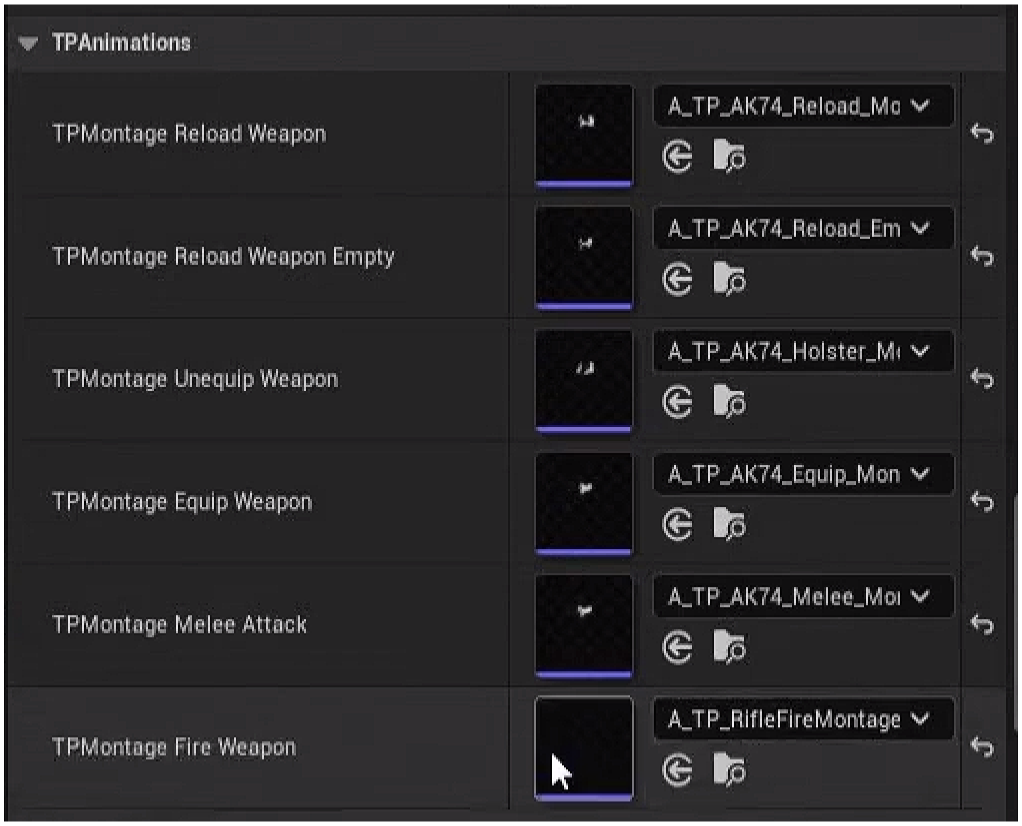
Shot Animation	A_AK-74_Fire
Reload Weapon Animation	A_AK-74_Reload
Empty Slide Animation	A_AK-74_Reload_Empty
Reload Weapon Animation Empty	None
Slide?	☐

Plugging in TP Animations

The next step is to plug in your Third Person montage animations we created earlier. Its as simple as matching up the montage to the TP animation slots in the Weapon BP, you'll need to add:

→ Third Person Body Reload
→ Third Person Body Reload Empty
→ Third Person Body Unequip
→ Third Person Body Equip
→ Third Person Melee Attack
→ Third Person Fire montage *(Optional)*

Example:



TPAnimations

TPMontage Reload Weapon	A_TP_AK74_Reload_Mo
TPMontage Reload Weapon Empty	A_TP_AK74_Reload_Em
TPMontage Unequip Weapon	A_TP_AK74_Holster_M
TPMontage Equip Weapon	A_TP_AK74_Equip_Mon
TPMontage Melee Attack	A_TP_AK74_Melee_Mo
TPMontage Fire Weapon	A_TP_RifleFireMontage

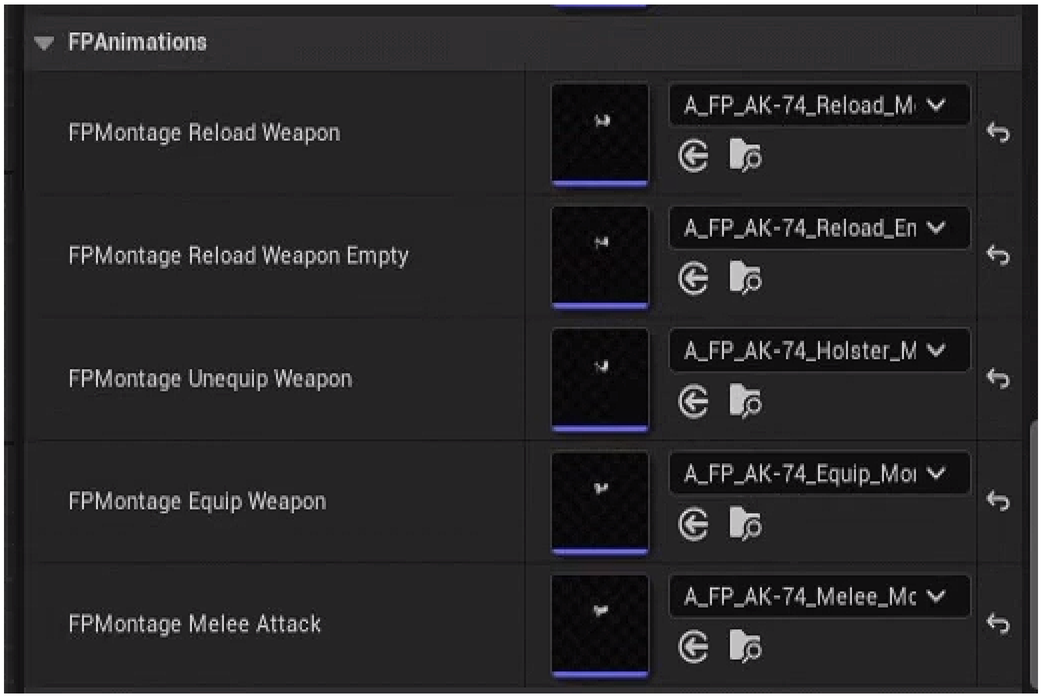
Plugging in FP Animations

The next step is to plug in your First Person montage animations we created earlier. Its the exact same process as the TP animation montages. You'll need to add:

→ First Person Arms Reload
→ First Person Arms Reload Empty
→ First Person Arms Unequip
→ First Person Arms Equip
→ First Person Arms Melee Attack

Note: *We don't need any first person fire weapon montages since firing animations are handled procedurally. You'll be able to customize the Procedural Firing System in the next few upcoming sections.*

Example:



FPAnimations

FPMontage Reload Weapon	A_FP_AK-74_Reload_M
FPMontage Reload Weapon Empty	A_FP_AK-74_Reload_En
FPMontage Unequip Weapon	A_FP_AK-74_Holster_M
FPMontage Equip Weapon	A_FP_AK-74_Equip_Mo
FPMontage Melee Attack	A_FP_AK-74_Melee_Mc

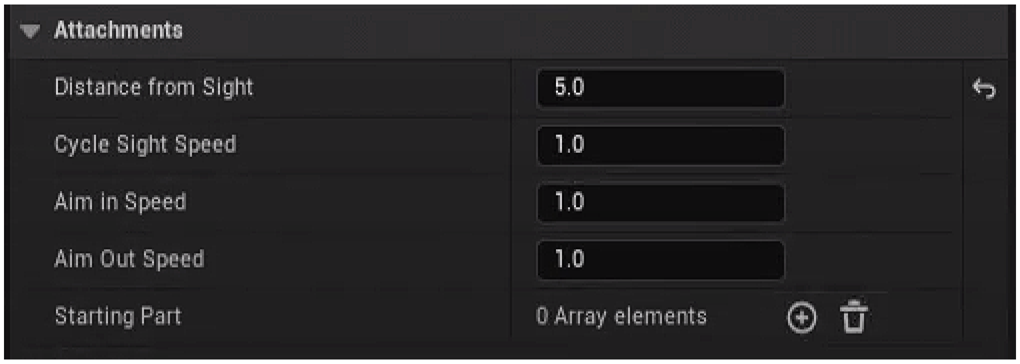
Setting Up Attachment Characteristics

We can now move onto changing our attachment and weapon movement characteristics, things like the ADS in speeds, ADS Out speeds, Cycling speeds can all be modified individually per weapon.

Head on over to the attachment section and you can modify these values. If you aren't sure what values to use, you can refer to the preset weapons values that come with the pack.

Note: *Generally 0.5s for ADS in and Out are the most common values used for ADS*

Example:



Attachments

Distance from Sight	5.0
Cycle Sight Speed	1.0
Aim in Speed	1.0
Aim Out Speed	1.0
Starting Part	0 Array elements

Note: *We don't need to add anything to the Starting Part array since our weapon handles adding parts via the customization system, however if you want an attachment to be hard coded to a weapon, you can do so by adding it in here.*

Setting Up Weapon Recoil

In this section, we will cover IK based Recoil and recoil patterns.

Head over to the recoil section within the Weapon Blueprint Class you have created, the first few things you'll notice is the ability to add custom recoil curves. You can use either the preset recoil curves or create your own.

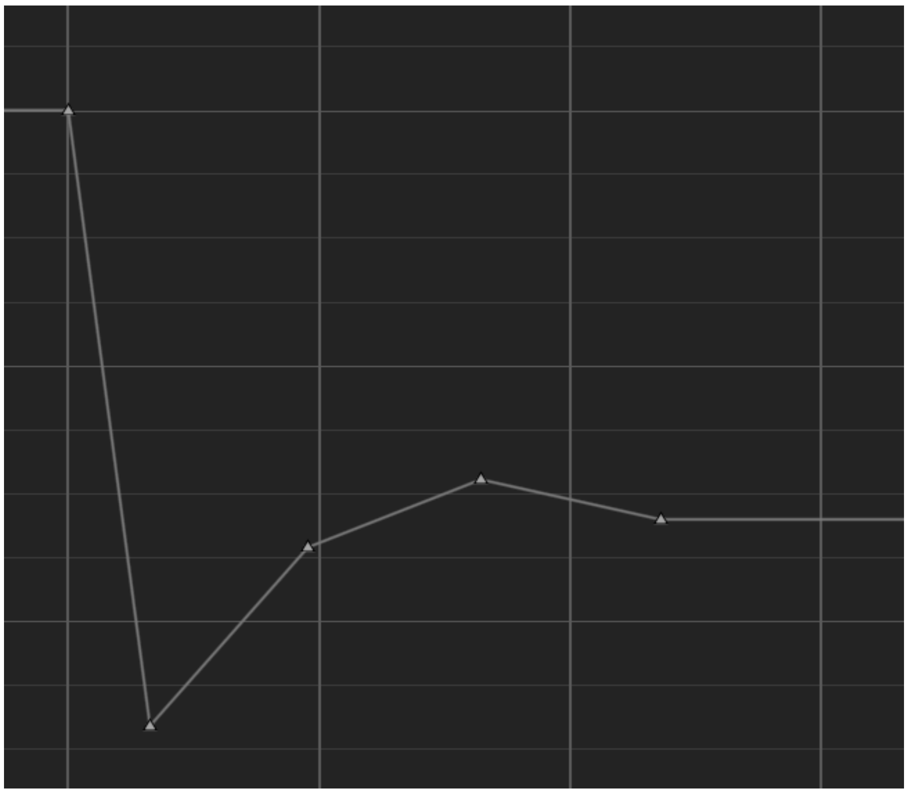
What does Pitch Recoil and Yaw Recoil mean:

→ Pitch Recoil kicks the screen upward each shot
→ Yaw Recoil kicks the screen sidewards each shot

To create your own recoil curves, head over to MultiplayerFPS → Game → Weapons → Primary → AK74U. In here you can find both the Pitch And Yaw curves. Copy these and move them to your own weapon folder.

Alternatively, if you want to create the curve from scratch, right click in the content browser → Miscellaneous → Curve → CurveFloat.

Example:



→ In this example, we can see the recoil pitch curve dramatically drop, and slowly rise back up. Negative means upward and Positive means downward. This is because we use the nodes Add Controller Pitch and Yaw.

What does Recoil Multipliers do:

In this pack, we included recoil multipliers, in some scenarios you may want to make recoil more apparent, i.e. when there is no optics or if you are hip firing.

→ Hip Fire Recoil Curve multiplier: This makes the recoil curves stronger based on if you are hip firing or not. So your screen may kick more or less based on the value. If you want to remove this feature, set it to 1.

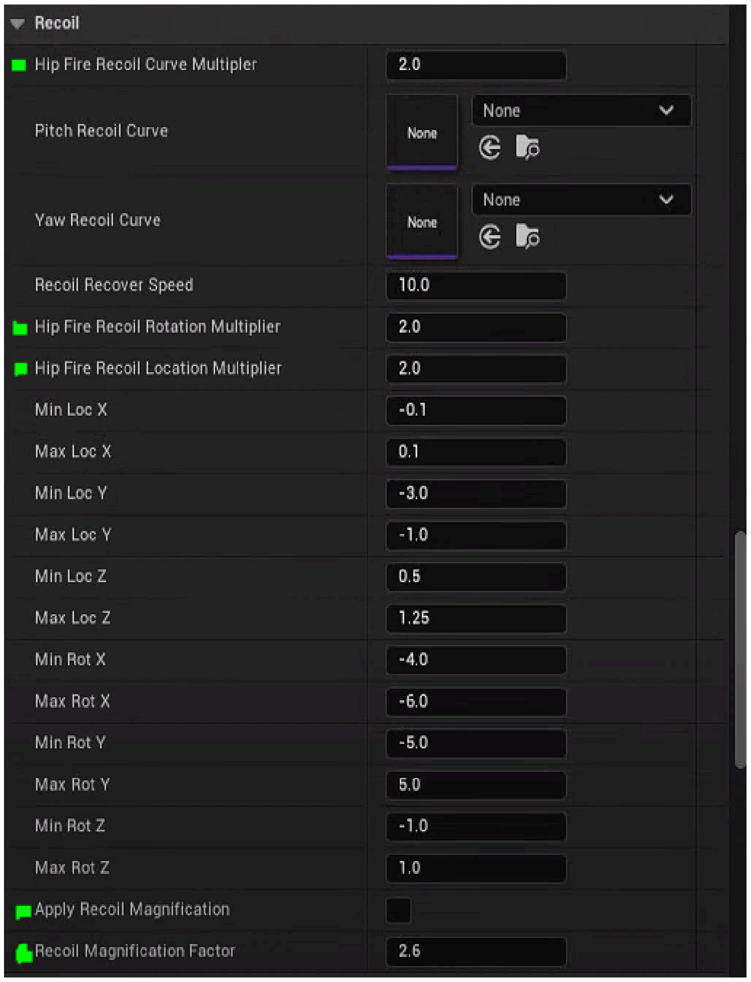
→ Hip fire Recoil Rotation Multiplier: This is purely IK based, this increases or decreases the amount of recoil applied to the hands when Hip Firing. The greater the value the more the hands rotate when firing. To eliminate this effect, set it to 1

→ Hip Fire Recoil Location Multiplier: Similarly to the Rotation Multiplier, it increases the location values of the IK based recoil, so the hand animation for recoil may appear more exaggerated. To eliminate this effect, set it to 1

→ Recoil Magnification Factor increases the amount of recoil if you have no sights equipped, to enable this feature, make sure Apply Recoil Magnification is Ticked.

These Variables are Highlighted in Green in the example below

Example:



Recoil

🟩 Hip Fire Recoil Curve Multipler	2.0
Pitch Recoil Curve	None
Yaw Recoil Curve	None
Recoil Recover Speed	10.0
🟩 Hip Fire Recoil Rotation Multiplier	2.0
🟩 Hip Fire Recoil Location Multiplier	2.0
Min Loc X	-0.1
Max Loc X	0.1
Min Loc Y	-3.0
Max Loc Y	-1.0
Min Loc Z	0.5
Max Loc Z	1.25
Min Rot X	-4.0
Max Rot X	-6.0
Min Rot Y	-5.0
Max Rot Y	5.0
Min Rot Z	-1.0
Max Rot Z	1.0
🟩 Apply Recoil Magnification	☐
🟩 Recoil Magnification Factor	2.6

Adjusting the IK based Recoil values

Here we are able to now modify how the Recoil Animations look. The parameters that start with either Min and Max are responsible for the recoil

→ Min means the minimum value the recoil can reach
→ Max means the maximum value the recoil can reach

So for example, the recoil on the X axis can only go from between -0.1 and 0.1. It will randomly select a variable within that range for the recoil on the X axis. The same is for the rest of the values such as Min Loc Z and Max Loc Z.

What do the Recoil X Y and Z values mean

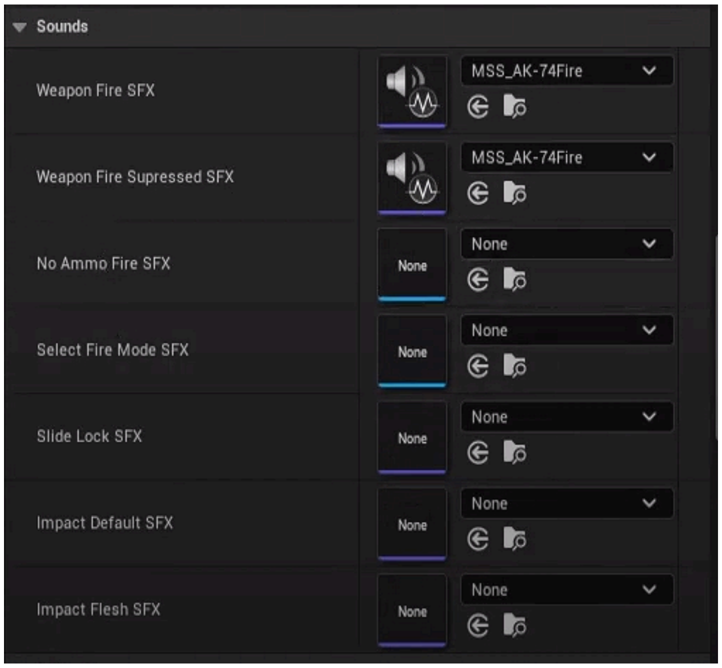
→ Location X - Sideways Location for Recoil
→ Location Y - Forwards And Backwards Location For Recoil
→ Location Z - Up and Down Location For Recoil

→ Rotation X - Upwards and Downwards Rotational Recoil
→ Rotation Y - Roll Rotational Recoil
→ Rotation Z - Sideways Yaw Rotational Recoil

Note: *The closer Min and Max values are, the tighter the weapon recoil feels, the more gap there is between the values the more jumpy the recoil becomes*

Setting Up Weapon Sounds

In this section, you can learn how to add your own weapon sounds. Head over to the Sounds section and plug in your own Meta Sound effects. You can copy the preset sound effects and swap out your own sound effects in the shot parameter array



Sounds

Weapon Fire SFX	MSS_AK-74Fire
Weapon Fire Supressed SFX	MSS_AK-74Fire
No Ammo Fire SFX	None
Select Fire Mode SFX	None
Slide Lock SFX	None
Impact Default SFX	None
Impact Flesh SFX	None

→ Swap in your own Dry Fire SFX
→ Default Impact SFX are sound impacts for surfaces with no physics materials
→ Impact Flesh SFX are sounds for hitting players or zombies
→ Slide lock SFX are for bolt fed rifles or pistols with a slide

Adding Our Weapon to Weapon List

The next step is to add our gun to the loadout weapon list. Also, we have to add the parts that are default to the weapon, for example the stock that is originally on the weapon. We can do this by heading over to MultiplayerFPS → Game → Blueprints → Customization → Data. Within this directory, open up DT_WeaponSelection.

Within the table, create a new row by pressing the add button. We need to fill out information like the weapon image to use, the name of the weapon, its mesh and the parts to use.

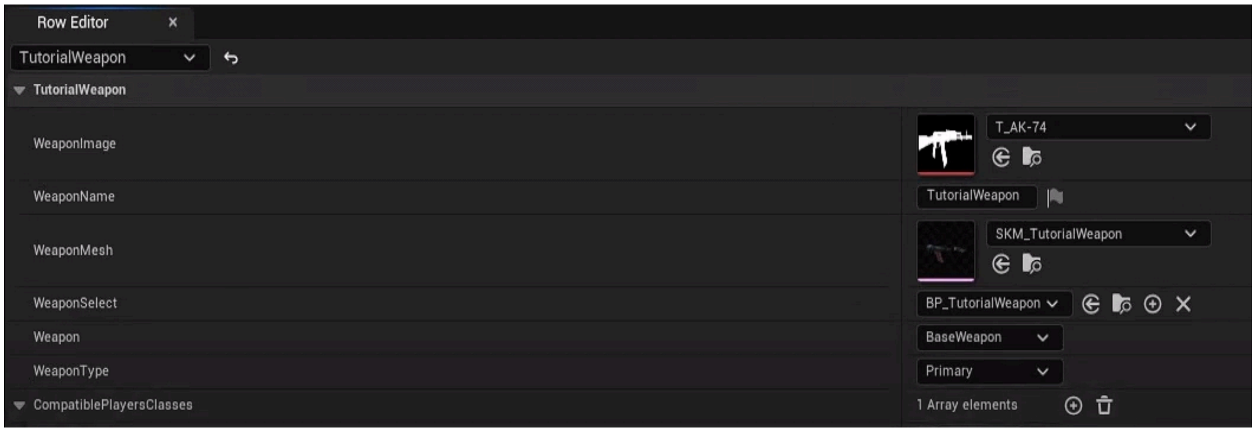
→ Weapon Select, this is where you plug in your weapon blueprint

→ Weapon, change the ENUM value to BaseWeapon

→ CompatiblePlayerClasses, add a new element(s), you can select what classes have the option to equip the weapon

→ Weapon Type, select the slot the weapon takes.

Example:



Row Editor

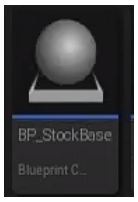
TutorialWeapon

TutorialWeapon	
WeaponImage	T_AK-74
WeaponName	TutorialWeapon
WeaponMesh	SKM_TutorialWeapon
WeaponSelect	BP_TutorialWeapon
Weapon	BaseWeapon
WeaponType	Primary
CompatiblePlayersClasses	1 Array elements

Creating our Default Part Actors

Within the table, you may have noticed parameters such as DefaultStock, DefaultPistolGrip, DefaultMuzzle ect asking for a Type Actor. These are responsible for spawning in the original parts of the weapon, they may be replaced with other actors such as a custom stock or suppressor, but if removed they revert to the original parts. We need to create actors for the original weapon parts mesh you imported earlier.

Default Stock



Head over to MultiplayerFPS → Game → Weapons → Attachments → Stock. Copy any of the premade stocks by right clicking → duplicate, or alternatively right click the BP_StockBase, right click → create child class.

Open up your new Stock actor, and fill in your weapons default stock mesh in the static mesh section in the components tab.

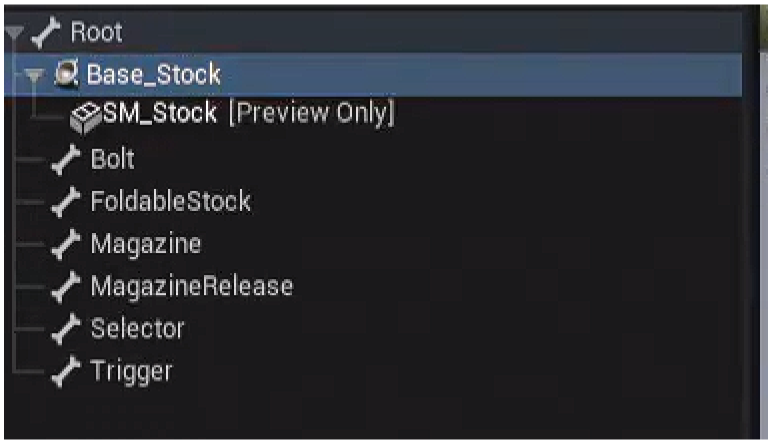
Creating Stock Socket

The next thing we need to do is create the stock's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_Stock**

Note: *It must be called Base_Stock, CaSe SenSitive*.

To help you position the stock better, right click → add preview mesh → search for your mesh.

Example:



Default Pistol Grip

Head over to MultiplayerFPS → Game → Weapons → Attachments → Grip. Copy any of the premade pistol grips by right clicking → duplicate, or alternatively right click the BP_PistolGripBase, right click → create child class.

Open up your new grip actor, and fill in your weapons default grip mesh in the static mesh section in the components tab.

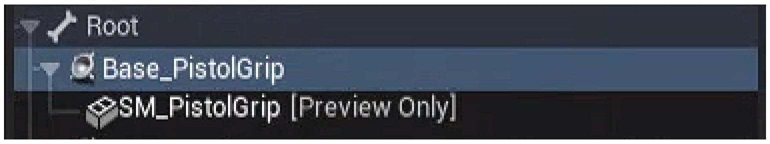
Creating Grip Socket

The next thing we need to do is create the grip's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_Grip**

Note: *It must be called Base_Grip, CaSe SenSitive*.

To help you position the grip better, right click → add preview mesh → search for your mesh.

Example:



Default Hand Guard

Head over to MultiplayerFPS → Game → Weapons → Attachments → Handguard. Copy any of the premade handguards by right clicking → duplicate.

Open up your new Handguard actor, and fill in your weapons default Handguard mesh in the static mesh section in the components tab.

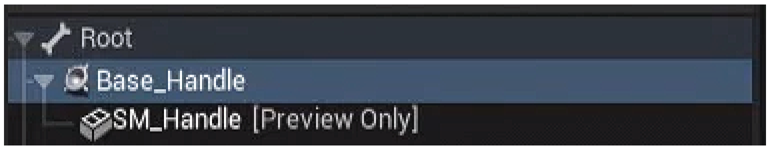
Creating Hand Guard Socket

The next thing we need to do is create the handguard's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_Handle**

Note: *It must be called Base_Handle, CaSe SenSitive*.

To help you position the handguard better, right click → add preview mesh → search for your mesh.

Example:


Root
Base_Handle
SM_Handle [Preview Only]

Default Muzzle

Head over to MultiplayerFPS → Game → Weapons → Attachments → Muzzle. Copy any of the premade muzzles by right clicking → duplicate.

Open up your new muzzle actor, and fill in your weapons default muzzle mesh in the static mesh section in the components tab.

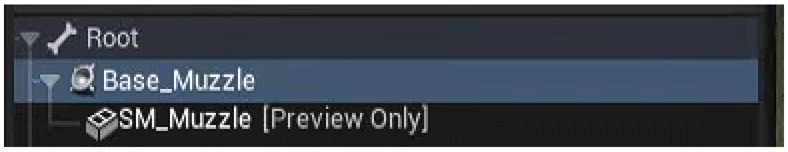
Creating Muzzle Socket

The next thing we need to do is create the muzzle's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_Muzzle**

Note: *It must be called Base_Muzzle, CaSe SenSitive*.

To help you position the muzzle better, right click → add preview mesh → search for your mesh.

Example:


Root
Base_Muzzle
SM_Muzzle [Preview Only]

Default FrontSight

Head over to MultiplayerFPS → Game → Weapons → Attachments → FrontSight. Copy any of the premade FrontSights by right clicking → duplicate.

Open up your new FrontSight actor, and fill in your weapons default FrontSight mesh in the static mesh section in the components tab.

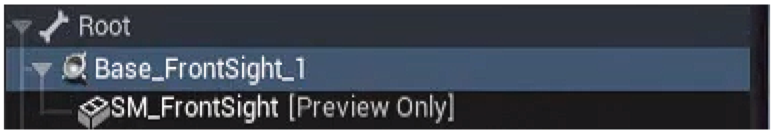
Creating FrontSight Socket

The next thing we need to do is create the FrontSight's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_FrontSight_1**

Note: *It must be called Base_FrontSight_1, CaSe SenSitive*.

To help you position the FrontSight better, right click → add preview mesh → search for your mesh.

Example:



Root
Base_FrontSight_1
SM_FrontSight [Preview Only]

Default RearSight

Head over to MultiplayerFPS → Game → Weapons → Attachments → RearSight. Copy any of the premade RearSight by right clicking → duplicate.

Open up your new RearSight actor, and fill in your weapons default RearSight mesh in the static mesh section in the components tab.

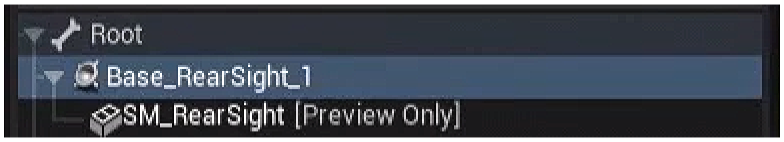
Creating RearSight Socket

The next thing we need to do is create the RearSight's socket. Head over to your weapon skeleton in MultiplayerFPS → Game → Weapons → YourGun → Mesh. Right click your Root bone and create a new socket. Name it **Base_RearSight_1**

Note: *It must be called Base_RearSight_1, CaSe SenSitive*.

To help you position the RearSight better, right click → add preview mesh → search for your mesh.

Example:



Note: *Remember to get the rear and front sight level with one another so that ironsights line up.*

Video Tutorial Index

Map tutorials:

- Ultimate Multiplayer FPS Kit: Setting Up Zombie Maps | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up Team Death Match | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up FFA Maps | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up Skirmish Maps | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up Conquest Maps | Unreal Engine 5

Attachment Tutorials:

- Ultimate Multiplayer FPS Kit: Setting Up Attachments | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up PIP Sights And Thermals | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up Optic Sights | Unreal Engine 5

Online SubSystem Tutorials:

- Ultimate Multiplayer FPS Kit: Setting Up Steam Online Sub System | Unreal Engine 5

Effects And Character Tutorials:

- Ultimate Multiplayer FPS Kit: Custom Dismemberment Mesh | Unreal Engine 5
- Ultimate Multiplayer FPS Kit: Setting Up A Character | Unreal Engine 5

Weapon Setup Tutorials:

- Ultimate Multiplayer FPS Kit: Setting Up A Weapon | Unreal Engine 5