

# **DOCUMENTATION – PROTO SHOOTER**

## **Description :**

Ce projet nommé « **PROTO Shooter** », et un prototype de projet Unreal Engine 5.4.3, permettant la facilitation de la création d'un shooter en FPS.

## **Mécanique Implémentées :**

- Système de jump configurable
- Système de dash
- Système de crouch/slide
- Système de gestion de la vie
- Système de tir
- Ai basique se déplacent uniquement
- Ai Random position
- Ai random position attack (melee + distance)
- Wave Systeme
- Pause Manager
- Audio Manager
- Save Manager
- End Game

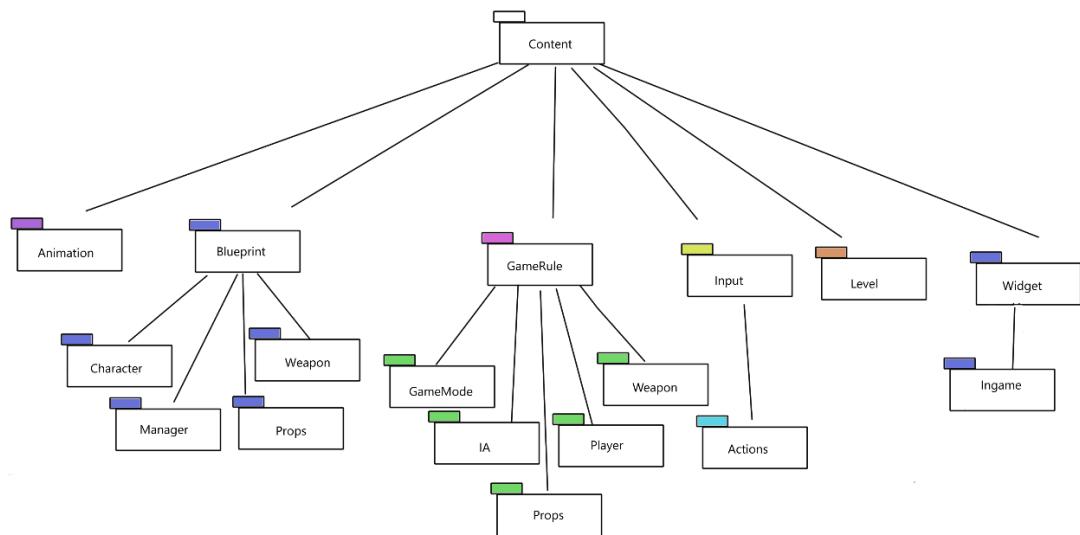
## **Sommaire :**

- Architecture -----	p2
o Projet -----	p2
o Plugins -----	p2
- Fonctionnement -----	p5
o Projet -----	p5
o Joueur -----	p7
o Game Mode -----	p9
o Weapons -----	p12
o Objets Collectables -----	p13
o Wave Système -----	p14
o IA -----	p15
- Conclusion -----	p17

## **Architecture :**

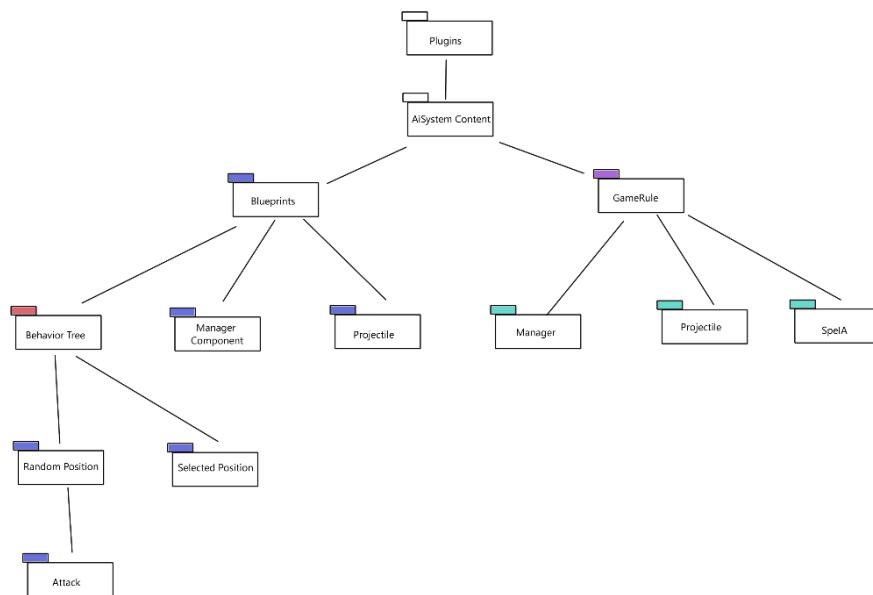
### **- Projet :**

Diagramme architecture projet.

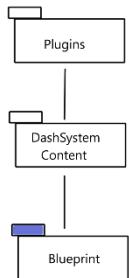


### **- Plugins :**

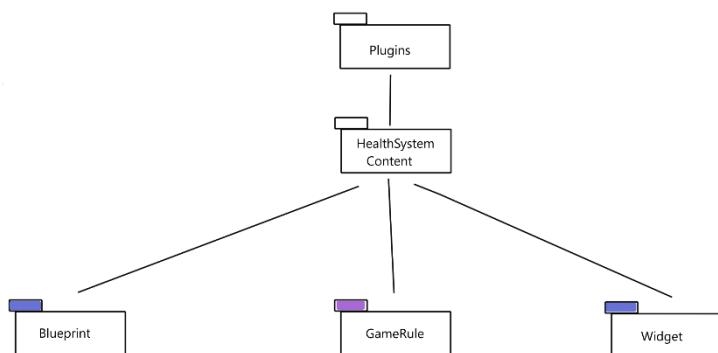
Diagramme architecture plugins - **IASystem**



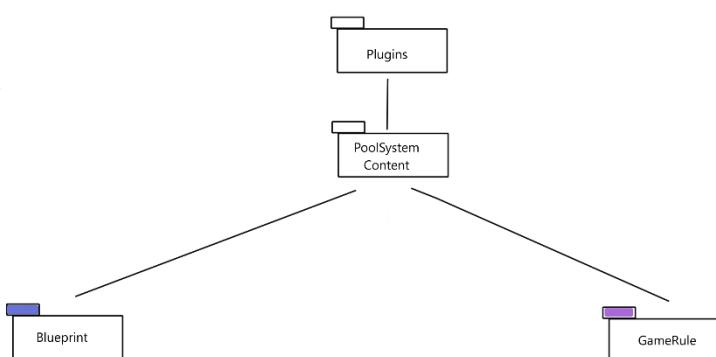
### Diagramme architecture plugins – **DashSystem**



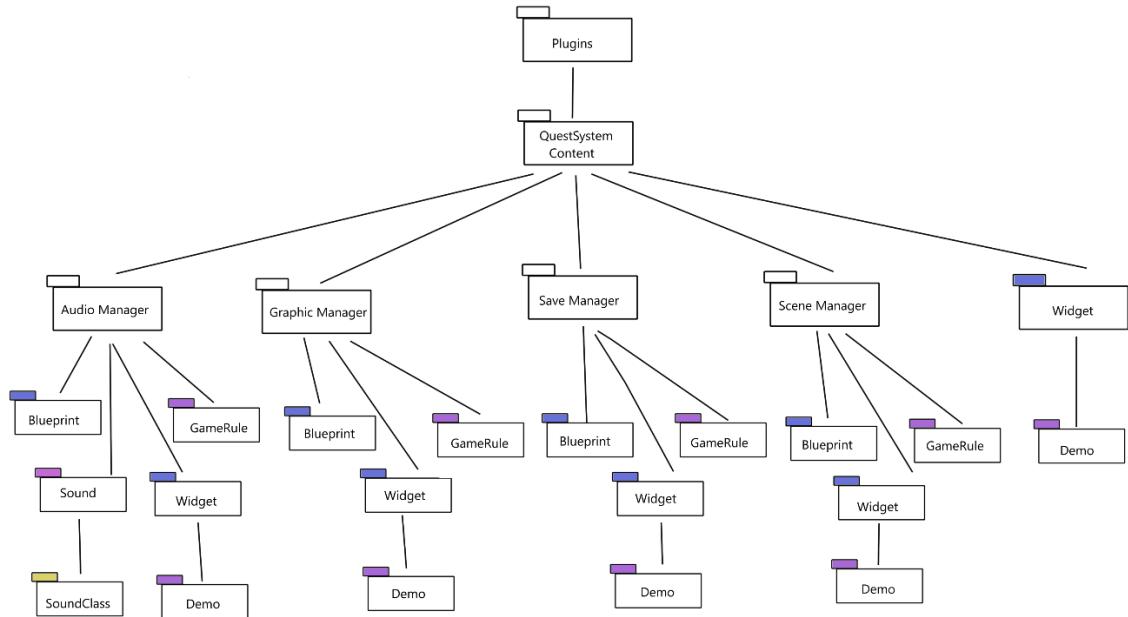
### Diagramme architecture plugins – **HealthSystem**



### Diagramme architecture plugins – **PoolSystem**



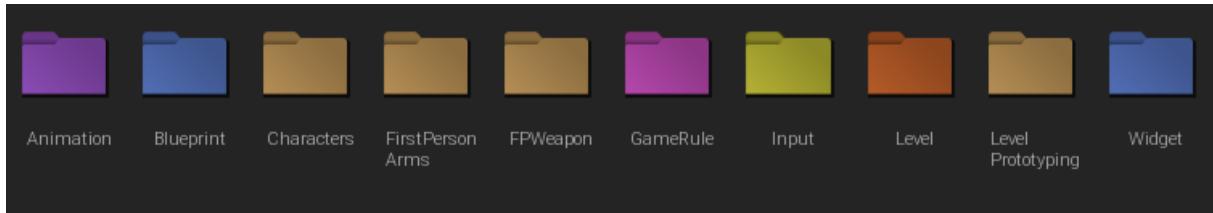
## Diagramme architecture plugins – **SettingManager**



## Fonctionnement :

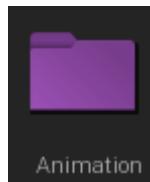
### - **Projet :**

Nous allons tous d'abord regarder l'architecture du projet pour être sûr de ne pas se perdre.



Pour commencer tous les dossiers colorés sont des dossiers non-générer de base et donc le code couleur à une signification :

### - **Violet :**



Comme son nom l'indique, vous trouverez les animations ainsi que l'Animation Blueprint (**AnimBP**) qui sera très important pour relier les actions à une animation.

### - **Bleu :**



Le dossier « **Blueprint** » contient tous les éléments de code réalisé pour le prototype. Ce sera dans ce dossier que vous pourrez retrouver par exemple les point de score, que vous pourrez personnaliser (graphiquement) et poser dans la scène.

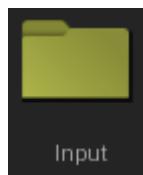
Le dossier « **Widget** » quant à lui contient tous les éléments UI/UX à mettre en place pour le jeu.

- **Rose :**

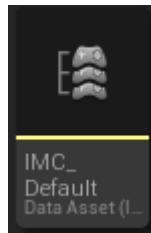


Ce dossier sera important pour vous, car il contient les tableaux de données modifiables pour rendre votre jeu unique (ex : hp du joueur).

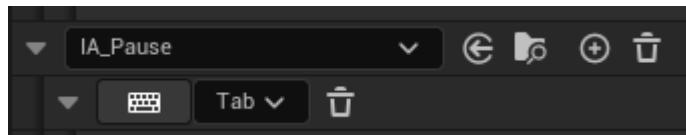
- **Jaune :**



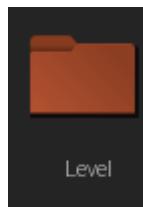
Comme son nom l'indique, il contient toutes les informations relatives au touches possible. Vous pourrez modifier les touches configurer dans le fichier :



Il vous suffira de modifier la touche rentrée ici :



- **Orange :**

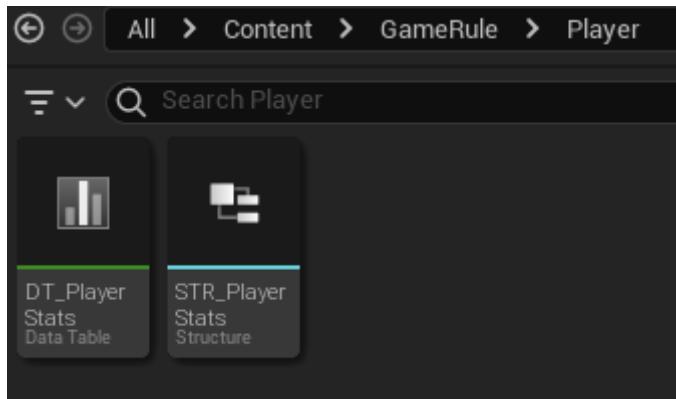


Enfin le dossier « **Level** » contiendra toute vos scènes.

- **Joueur :**

Dans le cadre du joueur vous aurais 3 fichiers à modifier pour pouvoir le personnaliser comme bon vous semble.

Pour la partie changement de valeur, vous devrais ouvrir le tableau « **DT\_PlayerStats** »

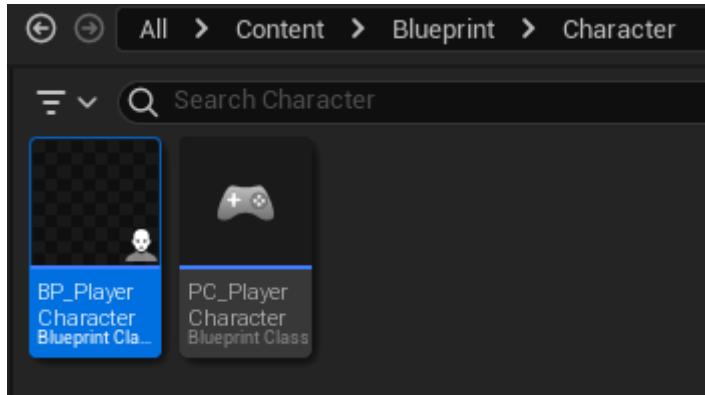


Vous y trouverez dedans les différentes valeurs modifiables du joueur

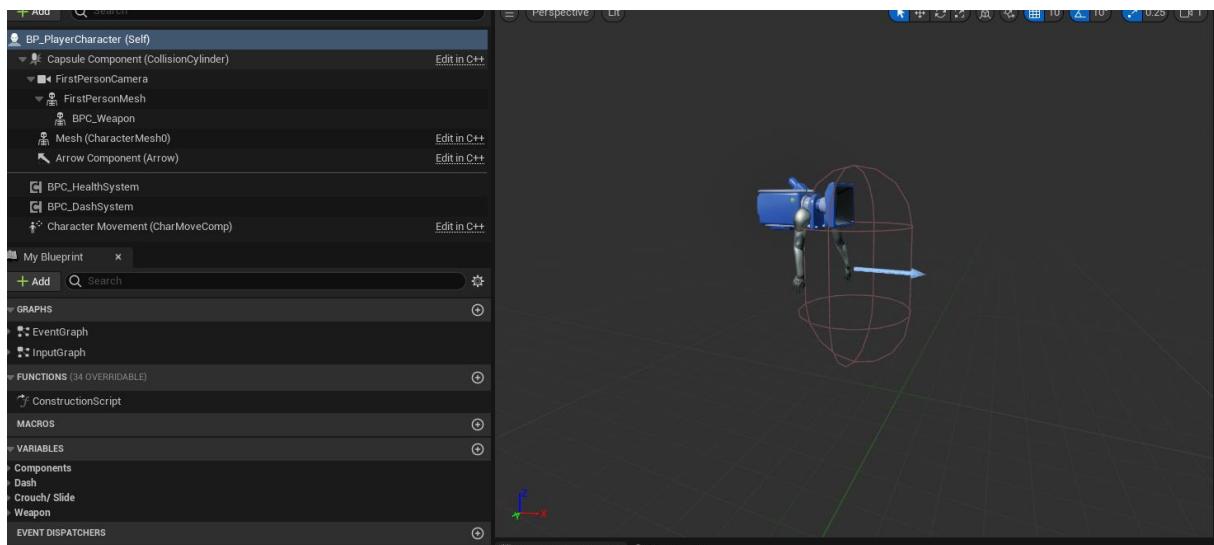
PlayerStats	
<b>PlayerStats</b>	
MaxHealth	125
IsHealthTextActive?	<input checked="" type="checkbox"/>
IsHealthBarActive?	<input checked="" type="checkbox"/>
InvincibilityDelay	1.0
IsDashActive?	<input checked="" type="checkbox"/>
<b>DashSystem</b>	
DashPower	10.0
DashDelay	1.0
IsCrouchActive?	<input checked="" type="checkbox"/>
IsSlideActive?	<input checked="" type="checkbox"/>
DefaultWeaponName	Pistol

Vous pourrez à l'aide de ce tableau modifier de nombreuses mécaniques du Joueur, comme activé ou non le Crouch (ex : « **IsCrouchActive ?** »). Vous pourriez également changer la valeur de vie du joueur ou encore l'arme de départ (ici on donne par défaut « **Pistol** »).

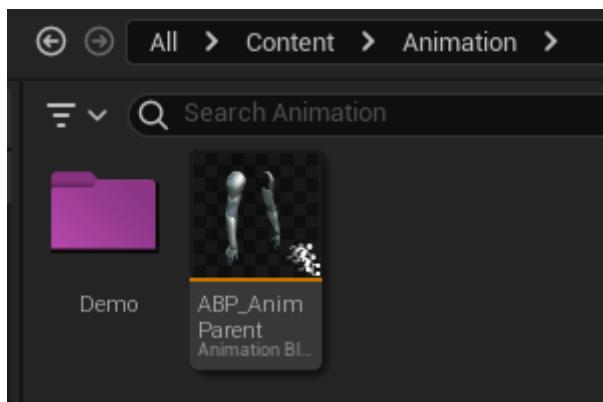
Vous pourrez également modifier le visuel du joueur en ouvrant le Blueprint « **BP\_PlayerCharacter** ».



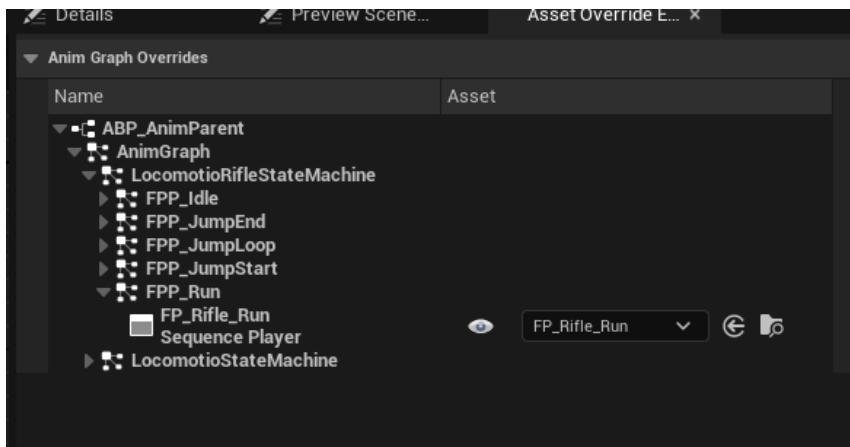
Une fois ouvert, vous aurez accès au différent composant modifiable du joueur.



Enfin il ne faut pas oublier de faire un **AnimBP** et de lui assigner pour pouvoir avoir vos propres animations sur le personnage, il faudra que ce soient des enfants de « **ABP\_ParentAnimation** », vous avez la partie démo qui vous montre un exemple.

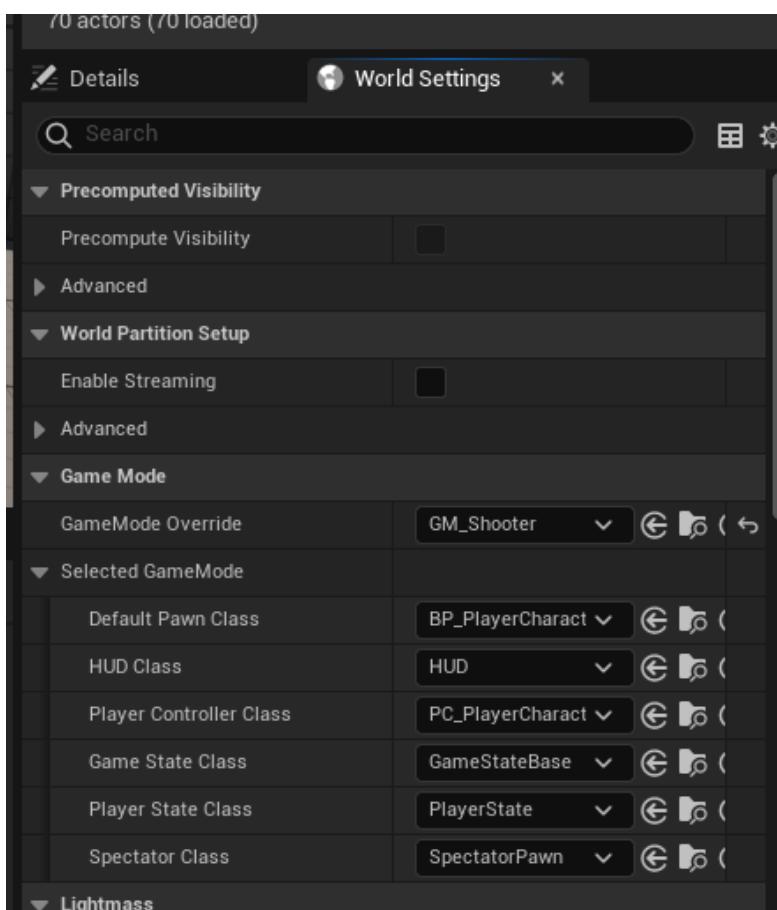


Et imaginons que dans cette enfant vous voulez changer l'animation de course, il vous suffira de changer la valeur de « **FP\_Riffle\_Run** ».

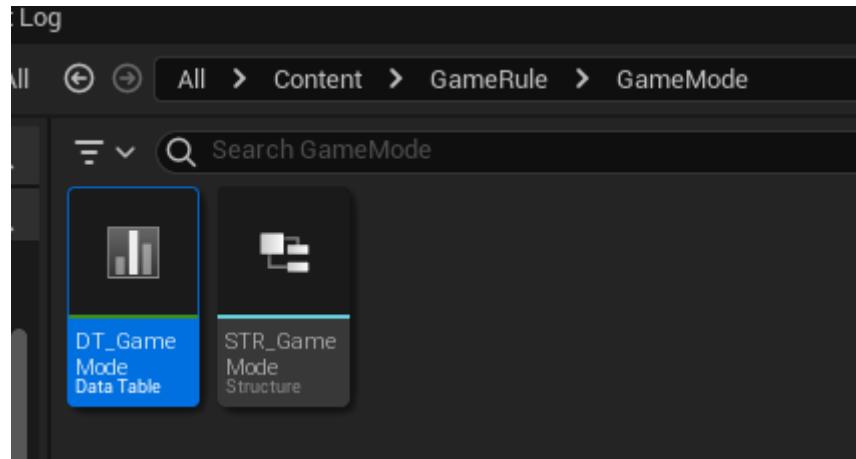


- **Game Mode :**

Pour la partie GameMode, il vous suffira de sélectionner le « **GM\_Shooter** » dans les « **WorldSetting** » de l'éditeur Unreal Engine.



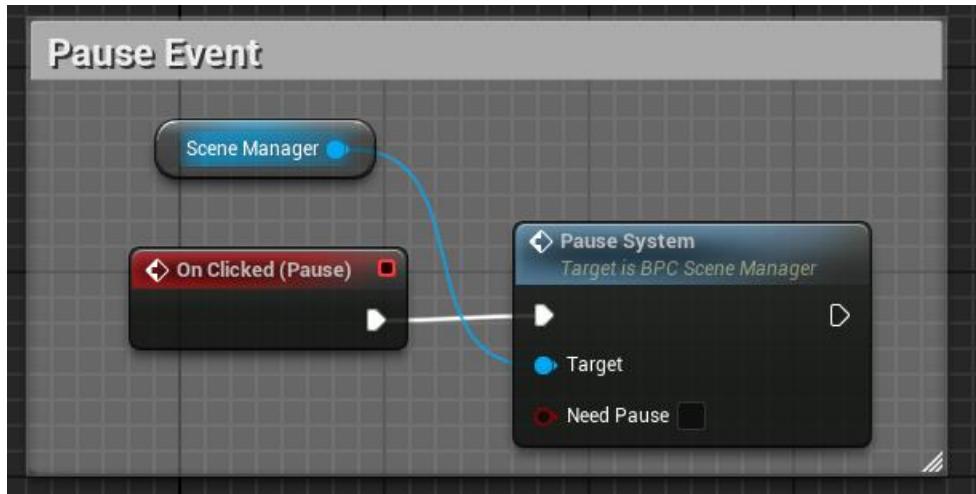
Puis, pour toutes la partie customisation, vous pourrez modifier les valeurs du tableaux « **DT\_GameMode** ».



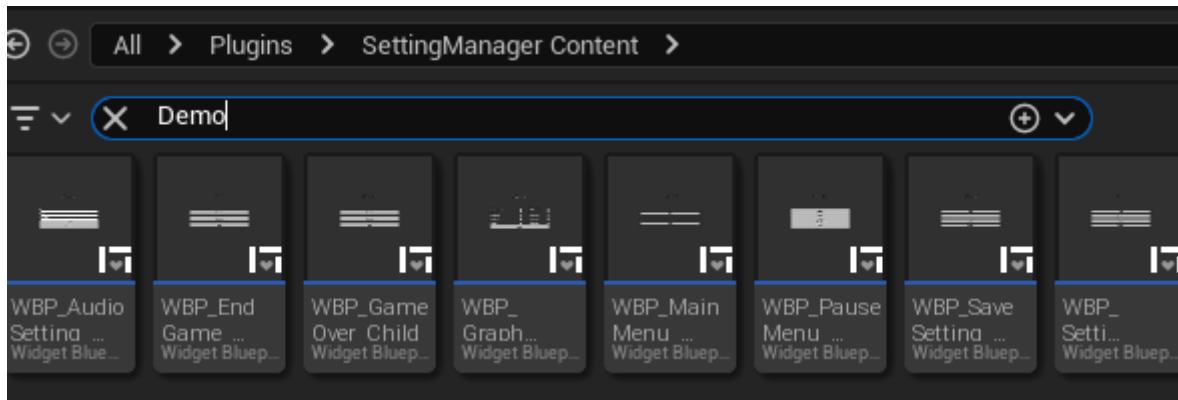
Une fois dedans, vous pourrez décide de si vous voulez changer la valeur max de score à avoir. Il vous faudra également référencer les différente UI du jeu (**AudioSetting**, **GameOver**, **EndGame**, **Pause**, **HUD**, **Setting**) qui seront des enfants de widget créé dans le plugin **Setting Menu**, il vous faudra les sélections au moment de la création de votre widget.

Flow Editor	
GameMode	▼
GameMode	
MaxScore	6
» AudioManager	
» GraphicManager	
» SaveManager	
» SceneManager	
GameOverWidgetClass	WBP_GameOver_Child ▾
EndGameWidgetClass	WBP_EndGame_Child ▾
PauseWidgetClass	WBP_PauseMenu_Child ▾
SettingWidgetClass	WBP_SettingMenu_Child ▾
HUDWidgetClass	WBP_HUD ▾
MainMenuWidgetClass	WBP_MainMenu_Child ▾
» PoolSystem	
ActorClass	BP_Projectile ▾
» WaveSystem	
» WavelInfo	3 Array elements
MinRangePlayer	500.0

Ici c'est un exemple de quoi sélectionner comme parent si vous voulez créer un nouveau **Menu Pause**. Une fois créé il vous faudra faire vos UI et les connecté vous-même aux fonctions. Exemple du système de pause



Pour mieux comprendre je vous invite à parcourir les dossiers des « **Demo** » dans le plugin « **SettingManager** » qui vous permettra de mieux comprendre.

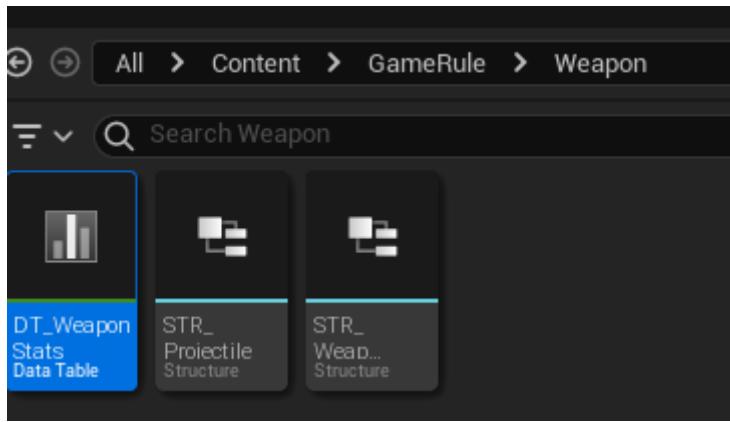


Une fois tous cela fait, il vous suffira de renseigner vos widgets dans le tableau.

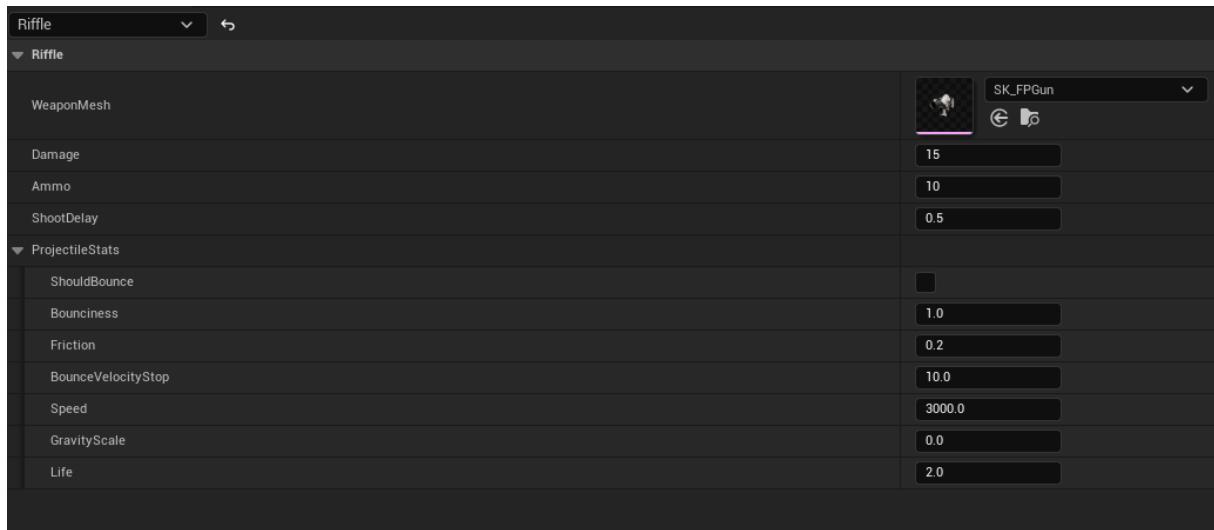
Row Editor	
GameMode	x
GameMode	
MaxScore	6
AudioManager	WBP_AudioSetting_Child ▾
GraphicManager	WBP_GraphicSetting_Child ▾
SaveManager	WBP_SaveSetting_Child ▾
SceneManager	WBP_GameOver_Child ▾
	WBP_EndGame_Child ▾
	WBP_PauseMenu_Child ▾
	WBP_SettingMenu_Child ▾
	WBP_HUD ▾
	WBP_MainMenu_Child ▾

### - **Weapons :**

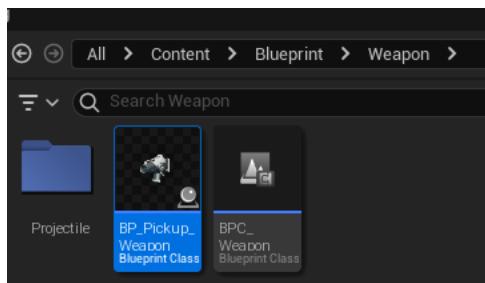
Pour ce qui est de l'ajout et des modification visuel de l'arme, tous se fait à partir du « **DT\_WeaponStats** ».



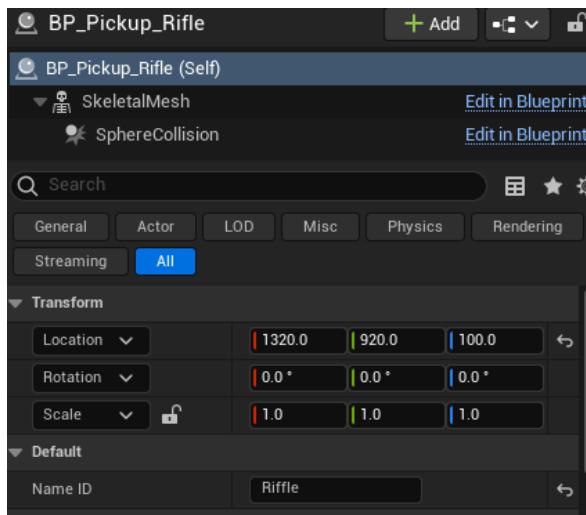
A l'aide de ce tableau, vous pourriez modifier le **mesh** de l'armes, ainsi que les stats de l'arme comme les dégâts, la vitesse d'attaque et également les valeurs des projectiles tiré par l'arme comme sa vitesse ou s'il peut rebondir sur le mur.



Puis pour le récupérer, vous pourrez soit utiliser le nom de l'arme dans la valeur « **DefaultWeapon** » du joueur, ou bien le mettre dans les « **Pickup\_Weapon** ».



Il vous faudra juste renseigner le nom de l'arme dans la variable « **NameID** ».



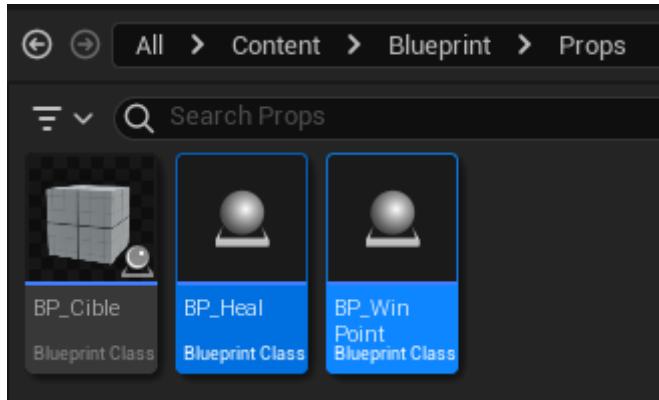
- **Objets collectables :**

Vous aurez accès à 2 types de collectable différent. Les heal configurable grâce au « **DT\_PropsInfo** ».

The screenshot shows the Unreal Engine Content Browser. At the top, there's a navigation bar with 'All', 'Content', 'GameRule', and 'Props'. Below it is a search bar and a filter icon. The main area lists three items: 'DT\_PropsInfo Data Table', 'STR\_Heal Structure', and 'STR\_PropsInfo Structure'. The 'DT\_PropsInfo Data Table' item is highlighted with a blue border. Below the browser, a detailed view of the 'PropsInfo' component is shown. It has a 'Heal' section with a 'Default' entry containing a single 'Heal' element with a value of 25.

Map elements	Members	Value
1	1	25

Vous aurez également accès à un point de fin de niveau qui vous permet d'avoir une fin dans une zone précise.



- **Wave System :**

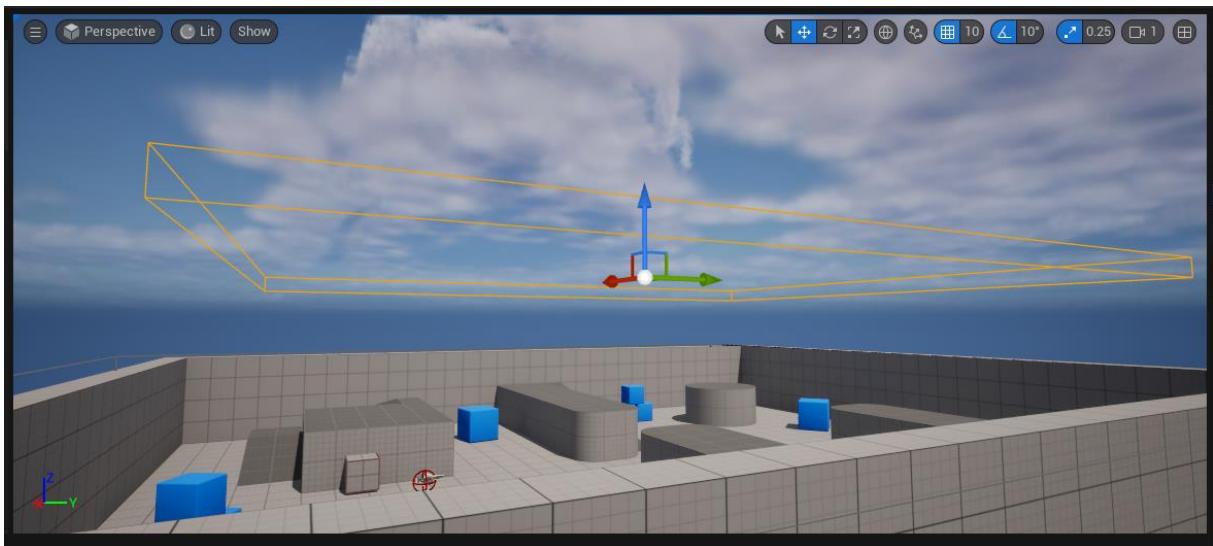
Vous aurez à disposition un système de génération de vague d'ennemi qui sera configurable dans le « **DT\_GameMode** »

WaveSystem		3 Array elements	<span style="color: #00AEEF;">+</span>	<span style="color: #D9534F;">-</span>
▼	WavelInfo			
	▼ Index [ 0 ]	5 members		
	Number	2		
	SpawnDelay	1.0		
	AiName	Default		
	AiType	RandomPosition		
	WaveSpawnerIndex	0		
	▼ Index [ 1 ]	5 members		
	Number	2		
	SpawnDelay	1.5		
	AiName	Default		
	AiType	RandomPositionAttack		
	WaveSpawnerIndex	0		
	▼ Index [ 2 ]	5 members		
	Number	2		
	SpawnDelay	2.5		
	AiName	Cac		
	AiType	RandomPositionAttack		
	WaveSpawnerIndex	0		
	MinRangePlayer	500.0		

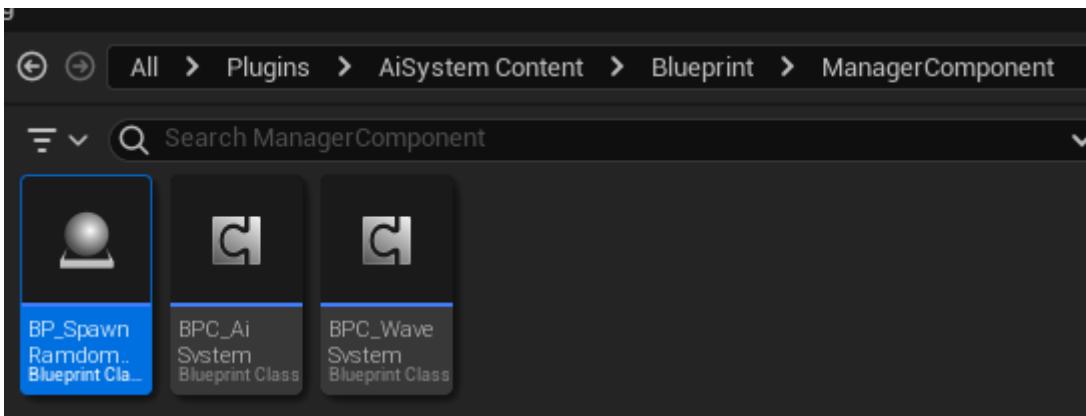
Vous pourriez rajouter autant de vague que vous voudrez, elle s'enchaîneront quand l'une d'entre elles sera terminé. Vous pourrez configurer le nombre d'ennemi, le délai d'apparition entre chaque ennemi, le type, ainsi que l'index de spawner à utiliser.

Enfin la valeur « **MinRangePlayer** » permet de changer qu'elle distance minimal d'apparition par rapport au joueur.

Donc comme indiquer il vous faudra minimum une surface d'apparition probable.

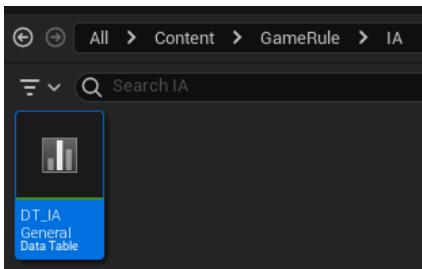


Vous le trouverez dans le plugin IA, il se nomme « **BP\_SpawnRandomObserver** »



#### - IA :

Pour ce qui est des IA, vous pourrez créer des profils d'IA grâce au « **DT\_IAGeneral** »



Vous pourrez créer différent profil selon le type d'IA voulue.

IAGeneral			
▼	IAGeneral		
	Ai_SelectPosition	0 Map elements	<input type="button" value="+"/> <input type="button" value="-"/>
▼	Ai_RandomPosition	1 Map elements	<input type="button" value="+"/> <input type="button" value="-"/>
▼	Default	2 members	<input type="button" value="▼"/>
RadiusRandom	1000.0		
► BBaseValue			
▼	IA_RandomPositionAttack	2 Map elements	<input type="button" value="+"/> <input type="button" value="-"/>
► Default	10 members	<input type="button" value="▼"/>	
► Cac	10 members	<input type="button" value="▼"/>	

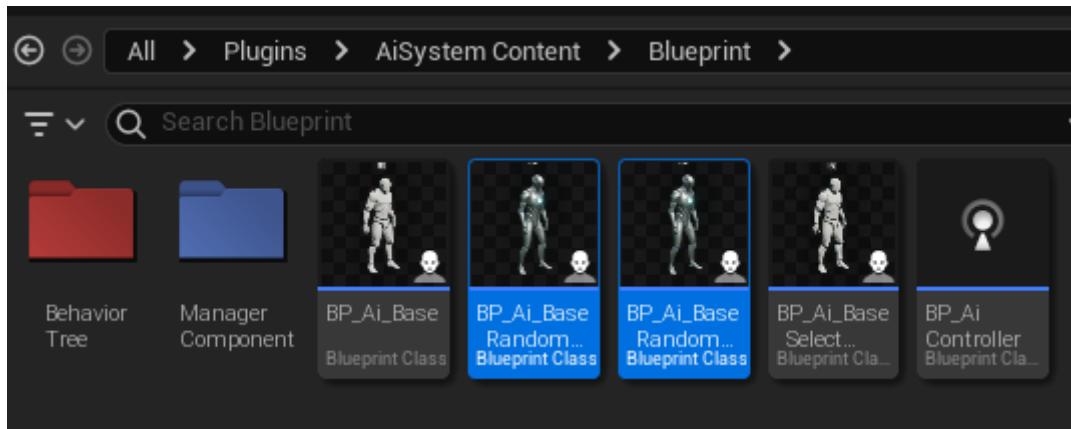
Pour les IA bougeant seulement de façon aléatoire, leur distance maximale de déplacement, vous pourrez changer leur nombre de point de vie, leur dégât et leur vitesse.

Ai_RandomPosition			
▼	Default	1 Map elements	<input type="button" value="G"/>
	RadiusRandom	2 members	<input type="button" value="▼"/>
▼	BBaseValue	1000.0	
	Health	20	
	Damage	10	
	Speed	300.0	

Pour ce qui est des IA pouvant attaquer, il y a de nombreuse autre valeur à gérer. Comme savoir si ce sont des ranges ou des mélées, leur vitesse d'attaque ou encore leur vision de détection. Mais également les stats des projectiles.

▼	Default	10 members	<input type="button" value="▼"/>
	RadiusRandomPoint	1000.0	
	IsCac?	<input checked="" type="checkbox"/>	
	IsRange?	<input checked="" type="checkbox"/>	
	DelayAttack	1.0	
	VisionAngle	50.0	
	VisionDistance	4000.0	
	WeaponDrop	BP_Pickup_Weapon	<input type="button" value="▼"/>
	DropRate	0.1	
▼	BaseValue		
	Health	20	
	Damage	15	
	Speed	300.0	
▼	ProjectileValue		
	ShouldBounce	<input checked="" type="checkbox"/>	
	Bounciness	1.0	
	Friction	0.2	
	BounceVelocityStop	10.0	
	Speed	1500.0	
	GravityScale	0.0	
	Life	2.0	

Pour les placer dans notre monde, vous pourrez soit utiliser le système de vague, soit les placer manuellement en les récupérant dans le plugin AISystem



## **Conclusion :**

Voici toutes les informations importantes à prendre en compte pour l'utilisation du Prototype de Shooter. N'hésitez pas à ouvrir les différents fichiers pour mieux comprendre le fonctionnement mais faites attention à rien casser 😊 .