

DOCUMENTATION – PROTO ADVENTURE

Description :

Ce projet nommé « **PROTO Adventure** », et un prototype de projet Unreal Engine 5.4.3, permettant la facilitation de la création d'un jeu d'aventure en TPS.

Mécanique Implémentées :

- Système de jump configurable
- Système de gestion de la vie
- Système d'attaque
- Système de dialogue
- Système d'inventaire
- Système de quête
- Switch camera
- Interactable Object (ex : chest)
- Ai basique se déplacent uniquement
- Ai Random position
- Pause Manager
- Audio Manager
- Save Manager
- End Game

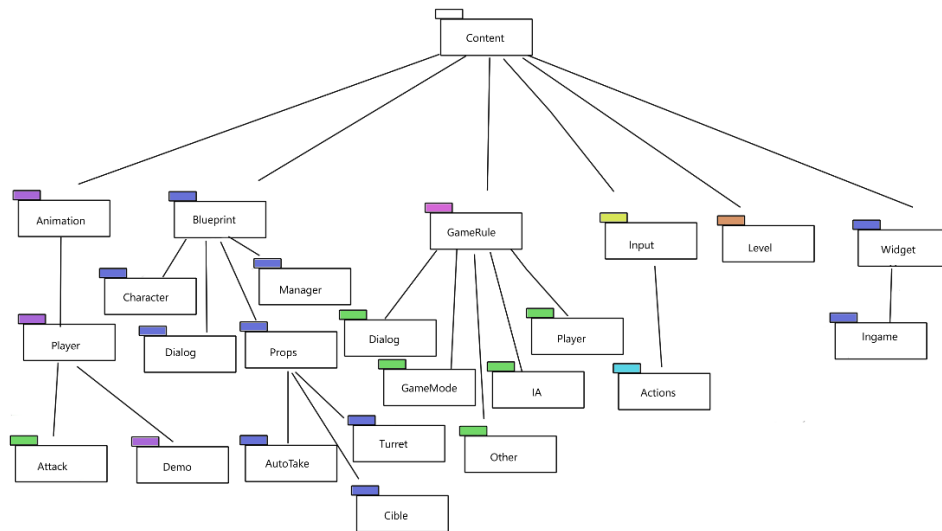
Sommaire :

- Architecture -----	p2
o Projet -----	p2
o Plugins -----	p2
- Fonctionnement -----	p5
o Projet -----	p5
o Joueur -----	p7
o Game Mode -----	p10
o Objets -----	p12
o Quête Système -----	p17
o Dialogue Système -----	p19
o Minimap Système -----	p20
o IA -----	p21
- Conclusion -----	p23

Architecture :

- Projet :

Diagramme architecture projet.



- Plugins :

Diagramme architecture plugins - IASystem

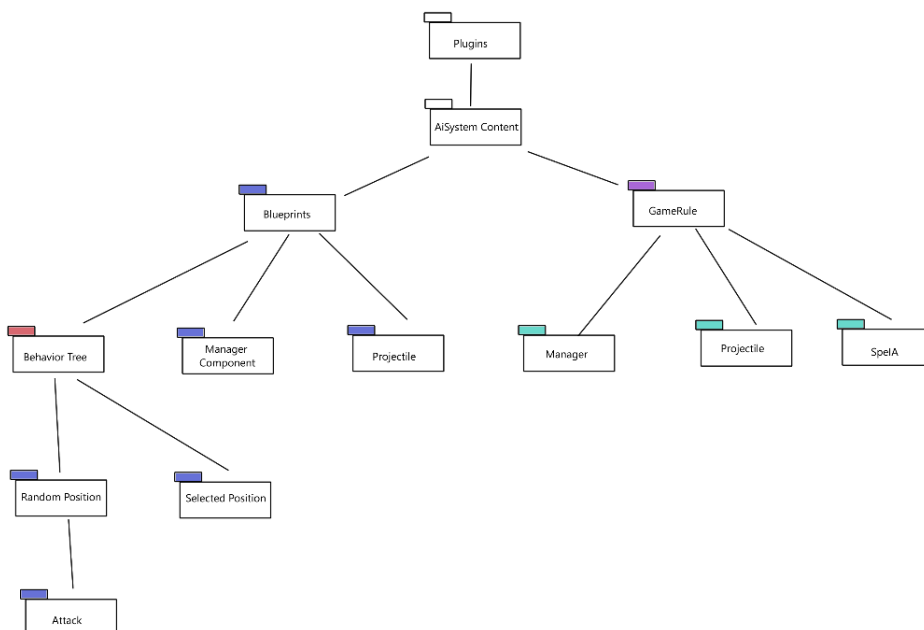


Diagramme architecture plugins – DashSysytem

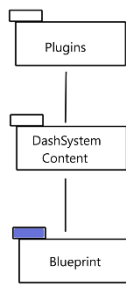


Diagramme architecture plugins – HealthSystem

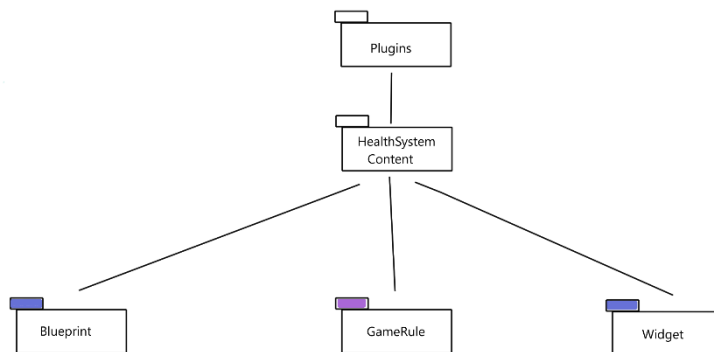


Diagramme architecture plugins – MinimapSystem

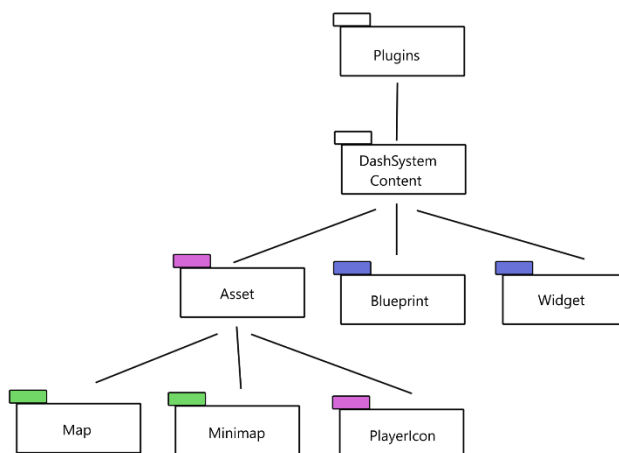


Diagramme architecture plugins –PoolSystem

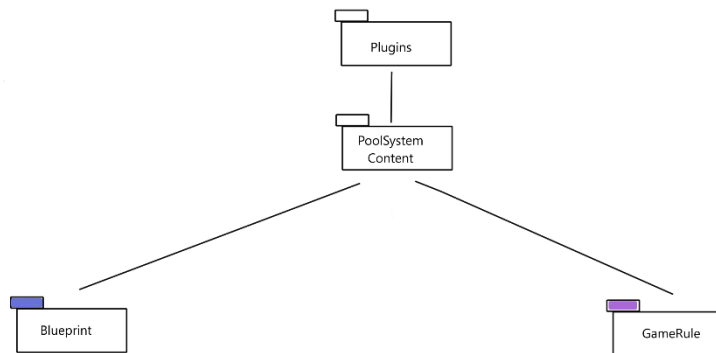


Diagramme architecture plugins – QuestSystem

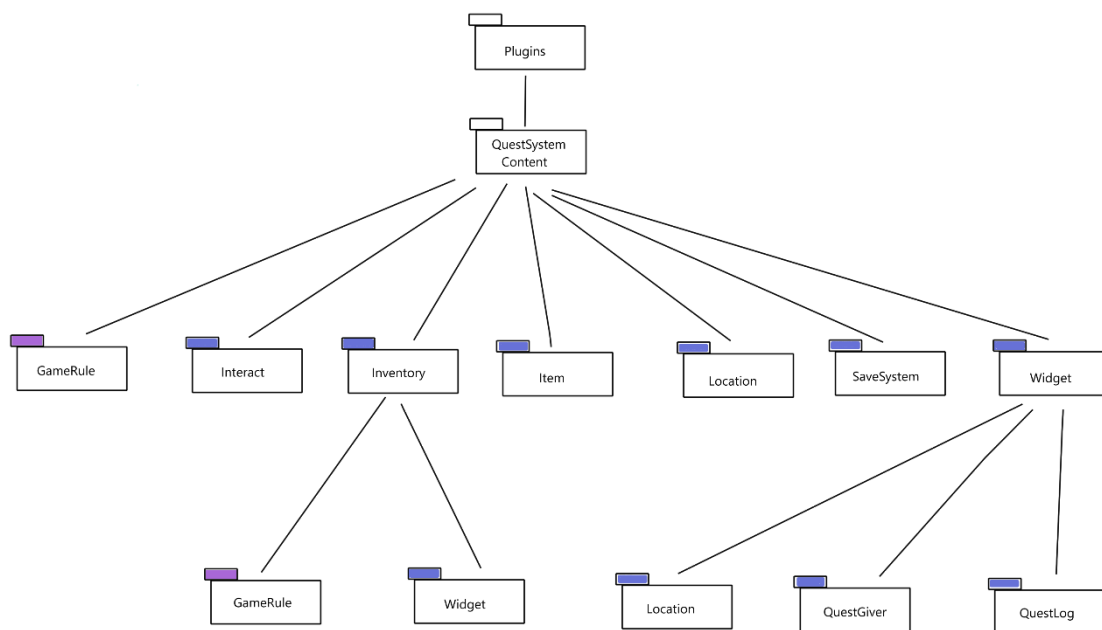
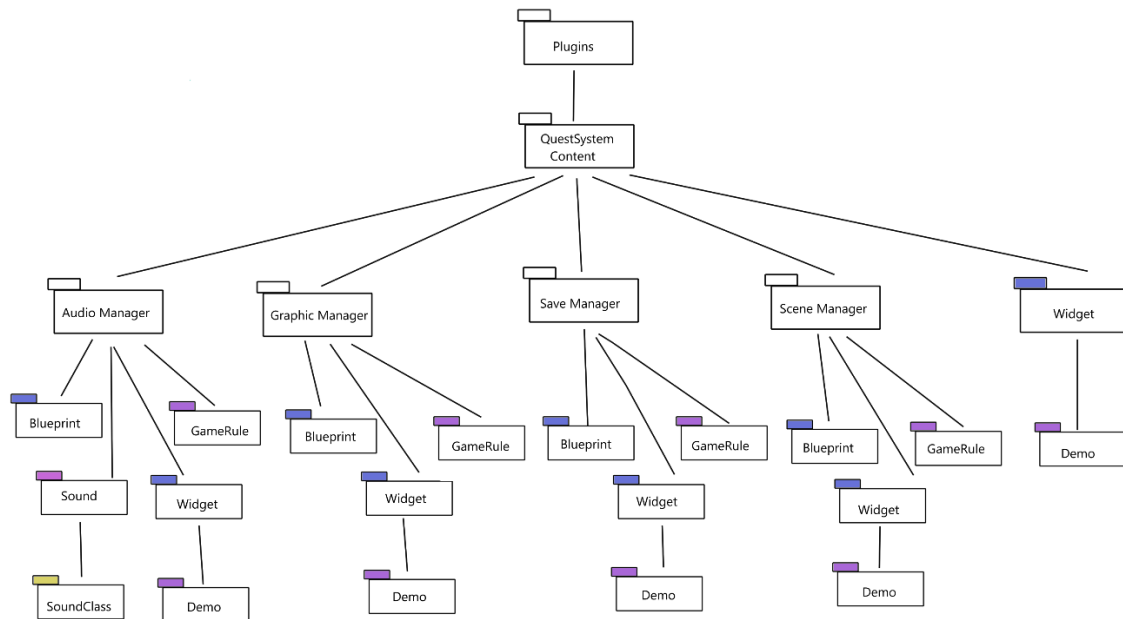


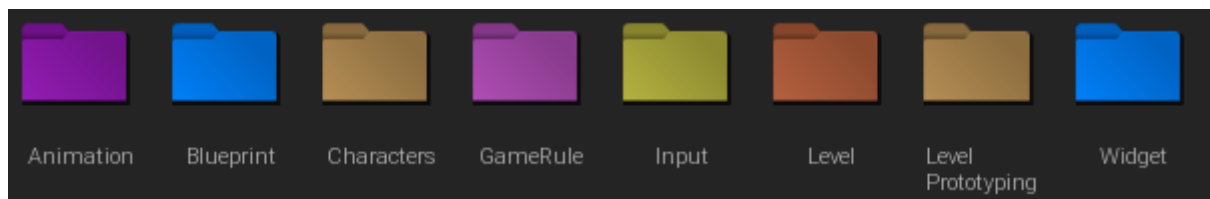
Diagramme architecture plugins – SettingManager



Fonctionnement :

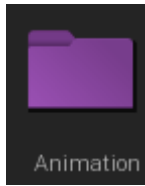
- Projet :

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



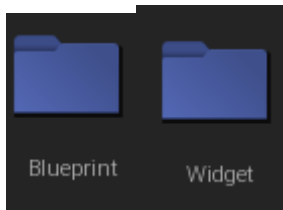
Pour commencer tous les dossiers colorés sont des dossiers non-générer de base et donc le code couleur à une signification :

- **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.

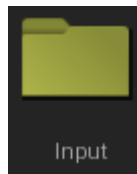
Le dossier « **Widget** » quant à lui contient tous les éléments UI/UX à mettre en place pour le jeu.

- **Rose :**

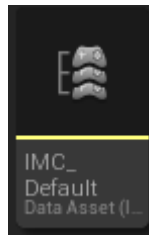


Ce dossier sera important pour vous, car il contient les tableaux de données modifiables pour rendre votre jeu unique (ex : hp du joueur).

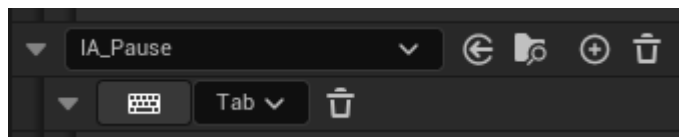
- **Jaune :**



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 :**

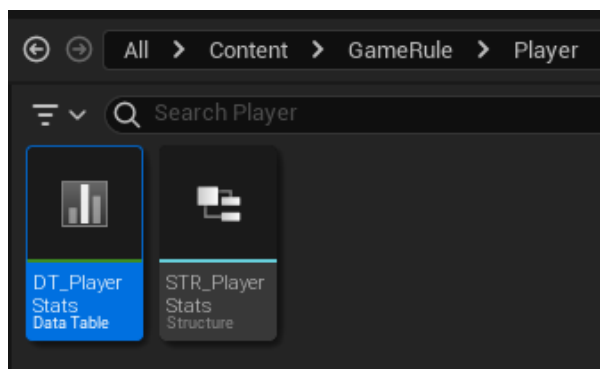


Enfin le dossier « **Level** » contiendra toute vos scènes.

- **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** »

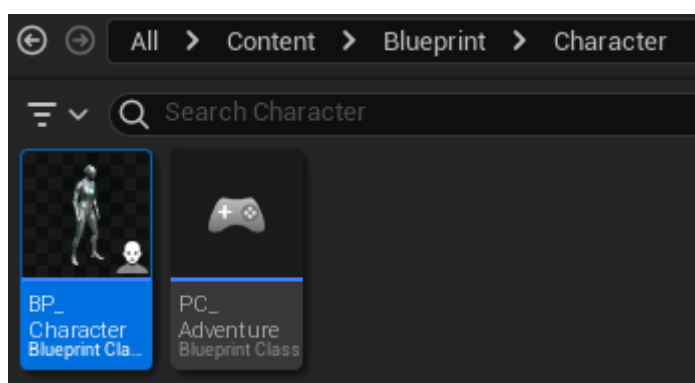


Vous y trouverez dedans les différentes valeurs modifiables du joueur

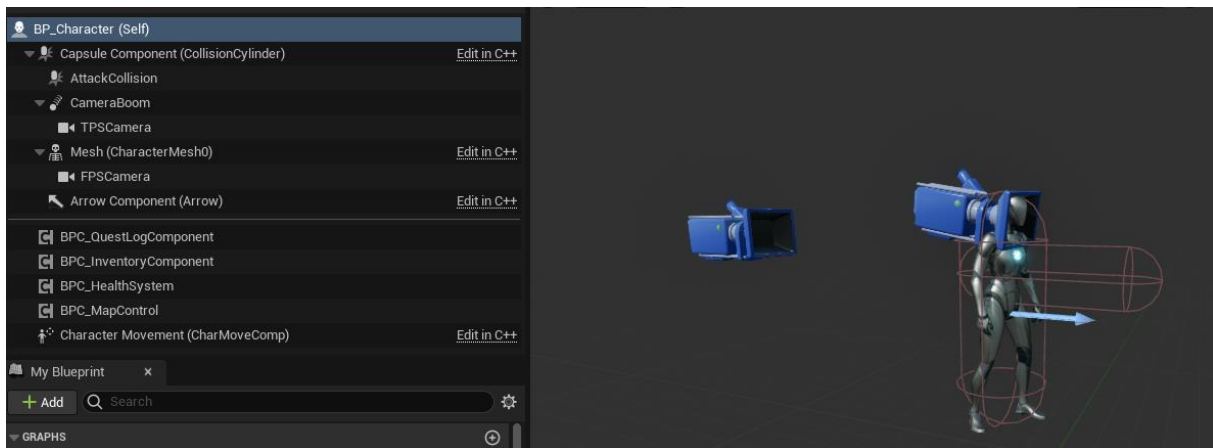
Player	▼	↩
▼ Player		
IsFPS?		<input type="checkbox"/>
Speed		500.0
SprintSpeed		750.0
AttackSpeed		1.0
AttackDamage		50
JumpVelocity		700.0
▼ HealthSystem		
MaxHealth		100
IsHealthTextActive?		<input checked="" type="checkbox"/>
IsHealthBarActive?		<input checked="" type="checkbox"/>
InvicibilityDelay		1.0
▼ RegenSystem		
IsRegenActive		<input checked="" type="checkbox"/>
RegenValue		1
RegenDelay		0.1
RegenCooldown		1.0
RegenLimit		75
RegenValueTrigger		50

Vous pourrez à l'aide de ce tableau modifier de nombreuses mécaniques du Joueur, comme activé ou non le système de régénération automatique (ex : « **IsRegnActive** »). Vous pourriez également changer des valeurs propres au joueur comme la vie ou encore le mode de vue du jeu « **IsFPS ?** » (Permet de jouer en mode FPS).

Vous pourrez également modifier le visuel du joueur en ouvrant le Blueprint
« **BP_Character** ».

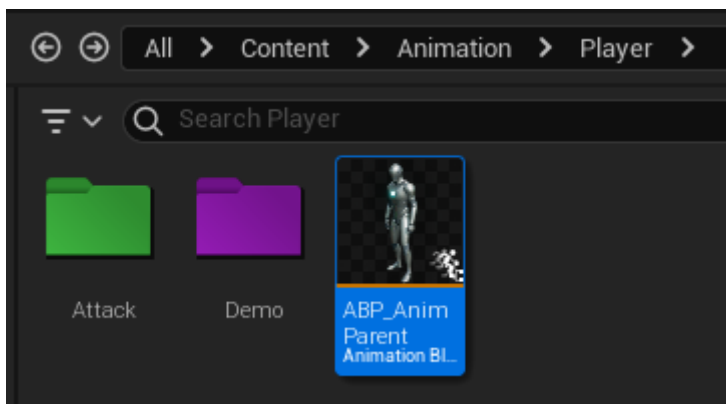


Une fois ouvert, vous aurez accès au différent composant modifiable du joueur.



Si jamais vous changez le modèle, il est très important de penser à modifier le « **ParentSocket** » de ma caméra « **FPSCamera** » pour qu'elle soit attachée à un socket de la tête de votre nouveau modèle.

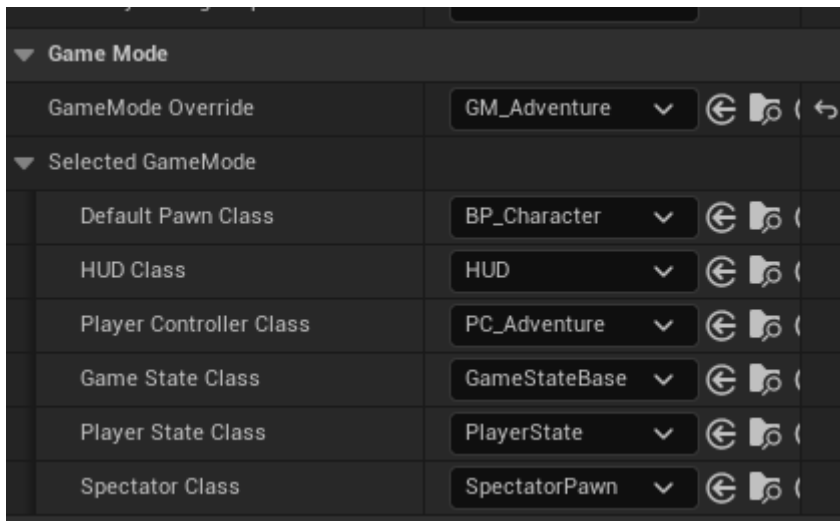
Enfin il ne faut pas oublier de faire un **AnimBP** et de lui assigner pour pouvoir avoir vos propres animations sur le personnage, il faudra que ce soient des enfants de « **ABP_AnimParent** », vous avez la partie démo qui vous montre un exemple.



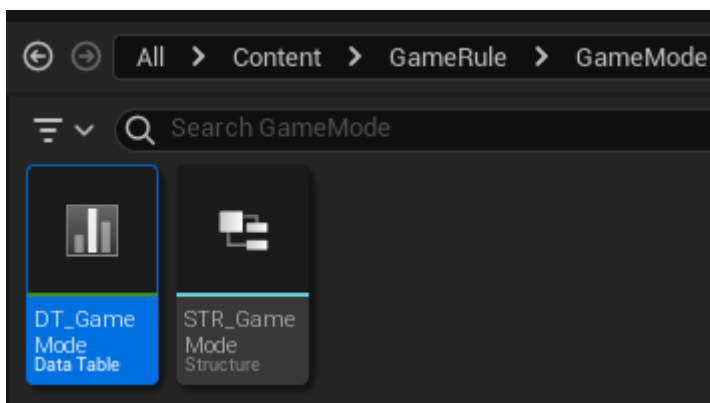
Et imaginons que dans cette enfant vous vouliez changer l'animation d'attaque du joueur, il vous suffira de changer la valeur de « **MM_Attack** ».

- **GameMode :**

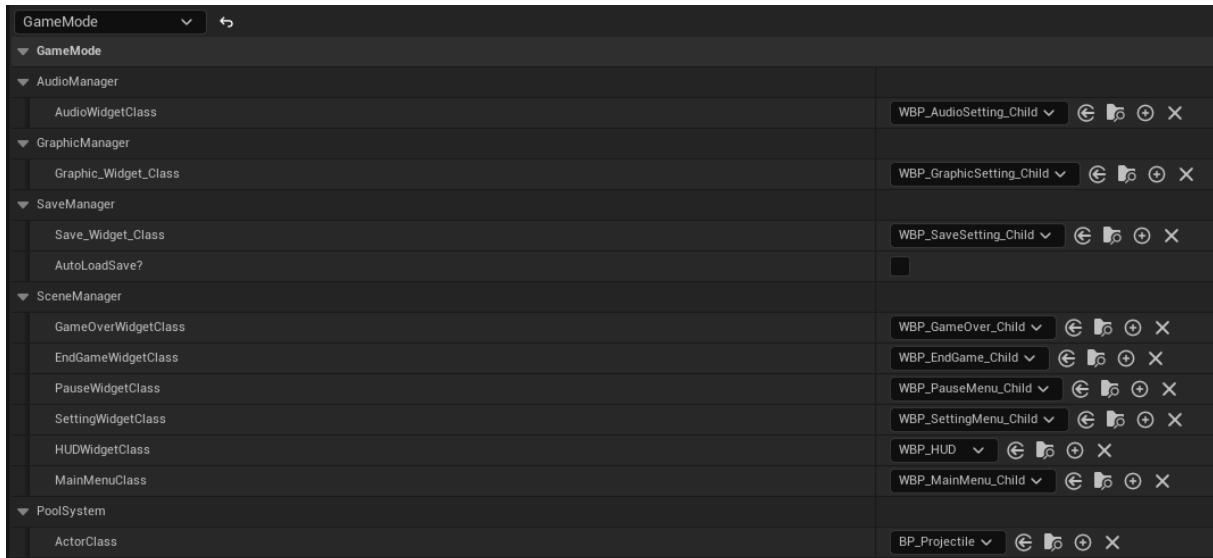
Pour la partie GameMode, il vous suffira de sélectionner le « **GM_Adventure** » dans les « **WorldSetting** » de l'éditeur Unreal Engine.



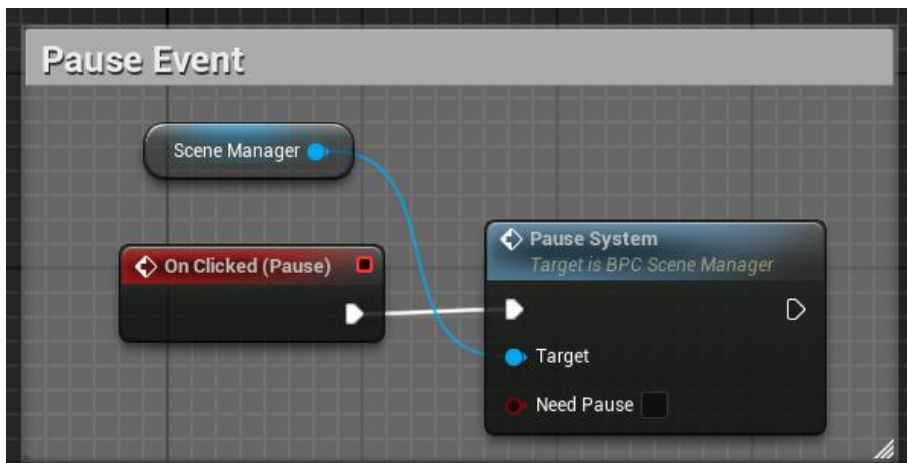
Puis, pour toutes la partie customisation, vous pourrez modifier les valeurs du tableaux « **DT_GameMode** ».



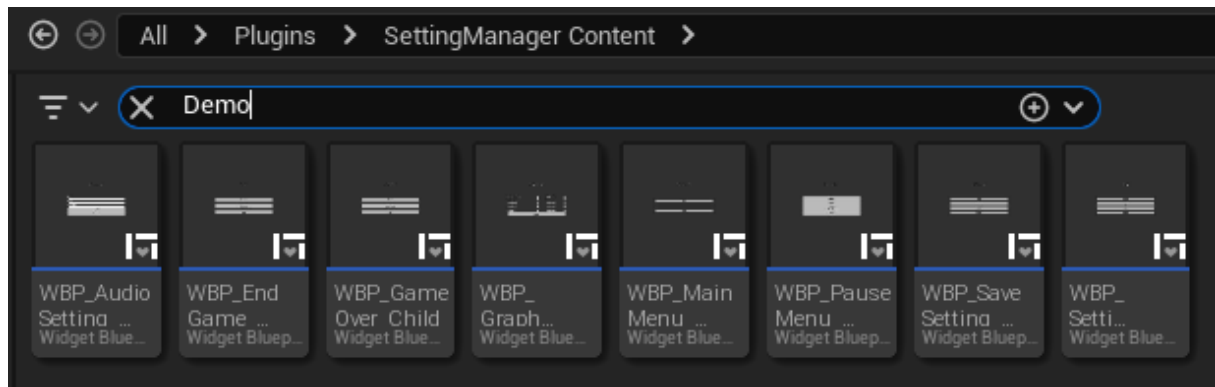
Une fois dedans, vous pourrez changer le type de projectile du système de pool. Il vous faudra également référencer les différentes UI du jeu (**AudioSetting, GameOver, EndGame, Pause, HUD, Setting**) qui seront des enfants de widget créé dans le plugin **Setting Menu**, il vous faudra les sélectionner au moment de la création de votre widget.



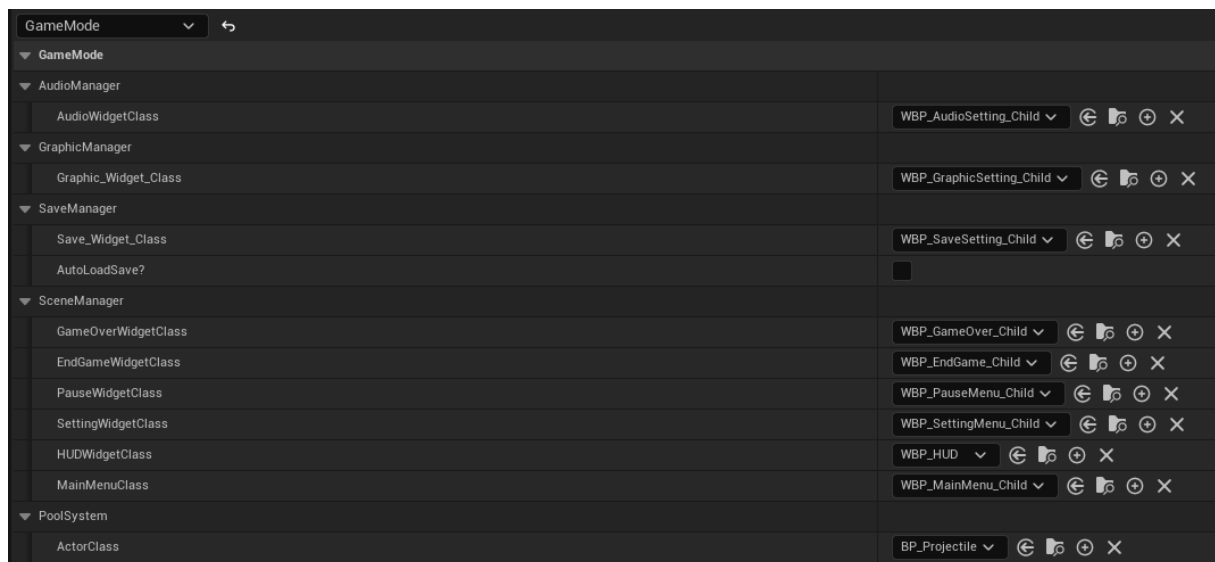
Ici c'est un exemple de quoi sélectionner comme parent si vous voulez créer un nouveau **Menu** Pause. Une fois créer il vous faudra faire vos UI et les connecté vous-même aux fonctions. Exemple du système de pause



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.

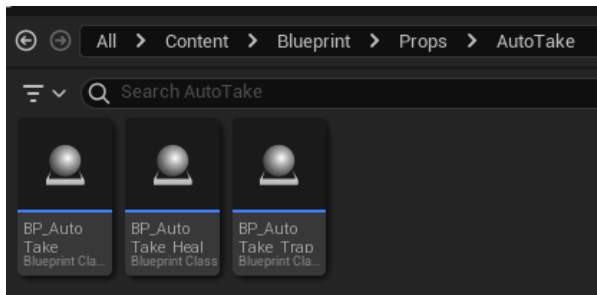


- Objet :

Du coté des objets, il existe de nombreux type d'objets différent. Ils sont découpés en 3 type distinct : **AutoTake**, **Turret** et **Props**.

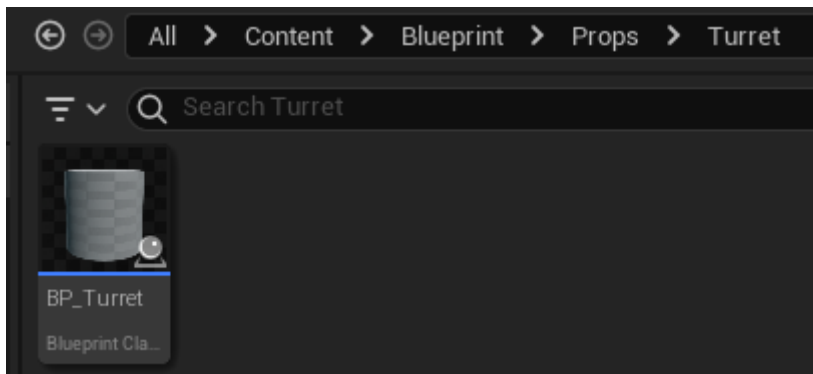
AutoTake :

correspond aux objets étant automatiquement déclenché lorsque le joueur rentre en collision avec comme le heal ou les piège au sol.



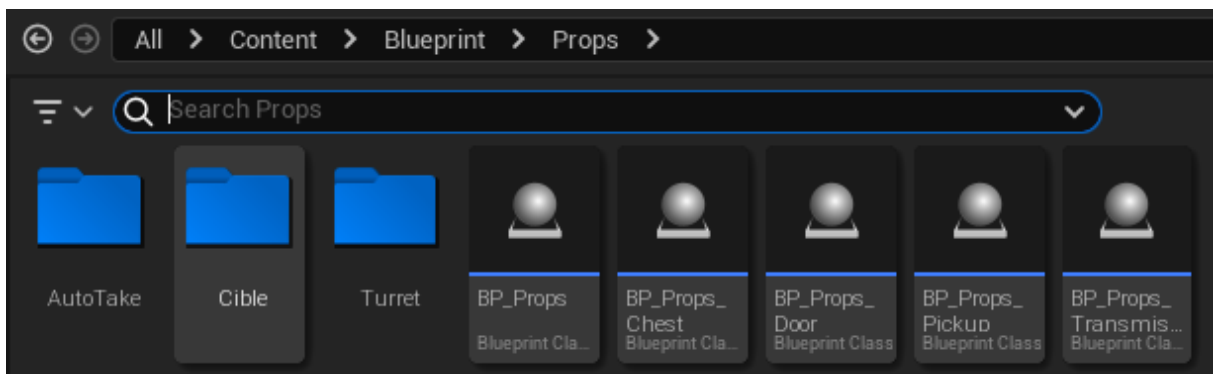
Turret :

Correspond, comme son nom l'indique, à une tourelle tirant sur le joueur.

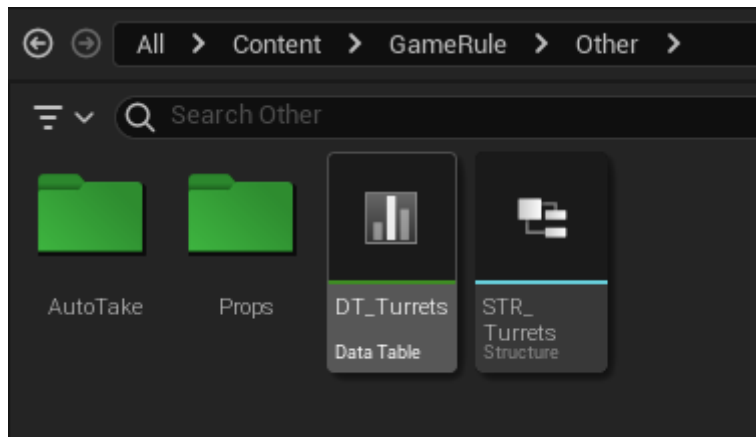


Props :

Enfin, il correspond à tous les objets ayant besoin d'une interaction pour fonctionner. Il correspond au : coffre, porte, objet au sol et les objets de transmission (ex : bouton).



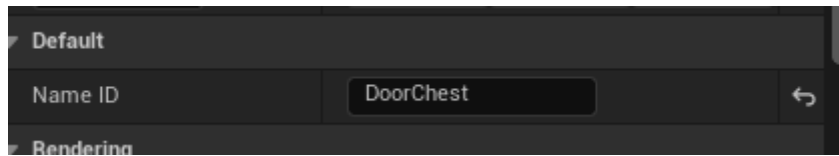
Tous ces objets sont customisable grâce au différents tableaux respectifs (« **DT_Turrets** », « **DT_AutoTakes** », « **DT_Props** »).



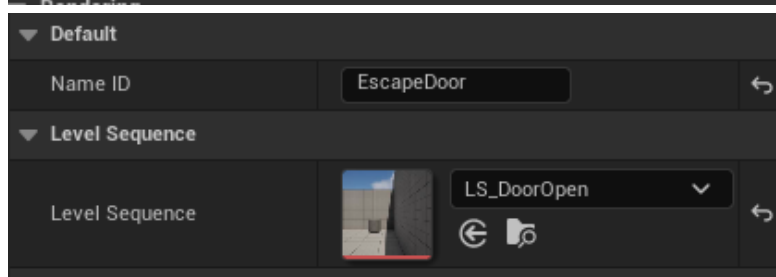
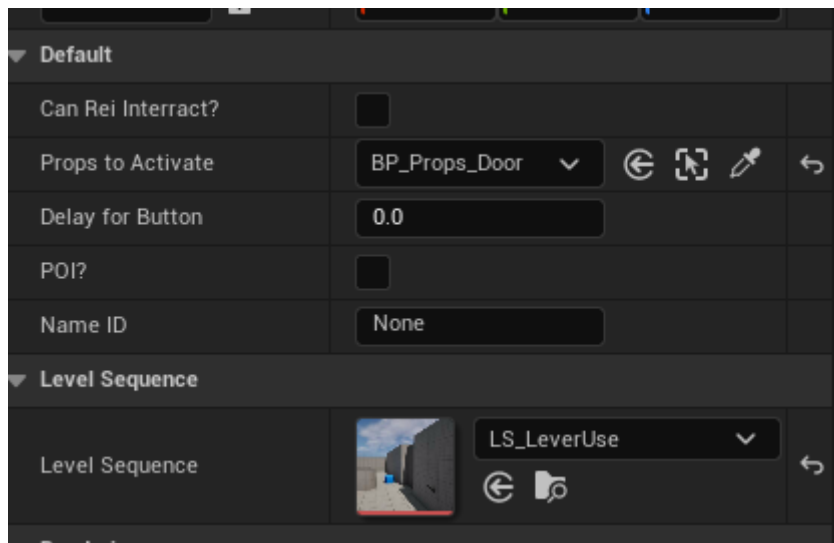
Voici un exemple du « **DT_Props** », vous pourrez modifier par exemple ID de la clé nécessaire pour ouvrir une porte, ou encore changer les objets donnés par un coffre.

Props		
▼ Props		
▼ Props_Chest		1 Map elements + -
▼ DoorChest		6 members ▼
KeyID		keyChest
IsKeyConsume		<input checked="" type="checkbox"/>
Quantity		1
▼ Item_Reward		1 Map elements + -
key		1 ▼
ObjectifID		None
POI?		<input checked="" type="checkbox"/>
▼ Props_Pickup		1 Map elements + -
▼ DragonBall		3 members ▼
ItemID		DragonBall
Quantity		1
POI?		<input type="checkbox"/>
▼ Props_Door		1 Map elements + -
▼ EscapeDoor		5 members ▼
KeyID		key
IsKeyCOnsume		<input type="checkbox"/>
Quantity		1
ObjectifID		door
POI?		<input type="checkbox"/>

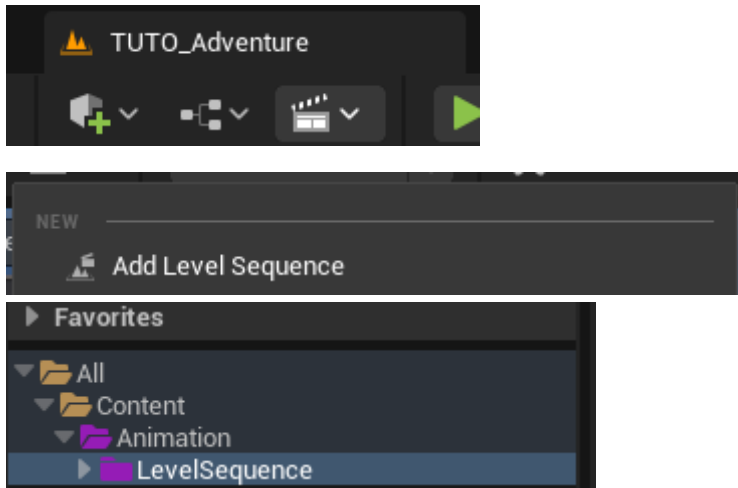
Une fois tous cela renseigné, il vous suffira juste pour n'importe quel objet, de faire correspondre son nom avec un profil créer.



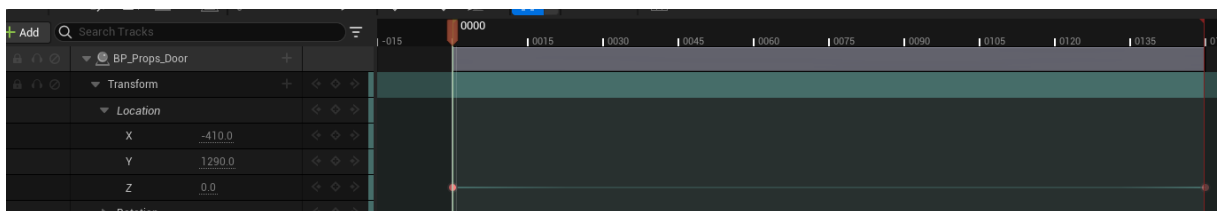
Il est important de prendre en compte que pour les Props de transmission ainsi que la partie animation de la porte, tous doit etre modifier sur l'objet.



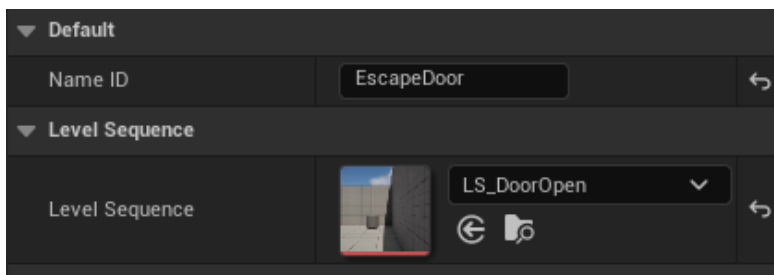
Les animations d'objet peuvent être réalisées facilement avec l'utilisation des **Level Sequence**. Pour en créer 1 il vous suffit d'appuyer sur le « **clap cinéma** » en haut de l'éditeur puis faire « **Add Level Sequence** » et choisir le dossier des level sequence pour s'y retrouver.



Une fois tous cela fait, il vous suffira simplement de réaliser la séquence que vous voulez en choisissant l'objet de la scène puis ce que vous voulez modifier. Par exemple, ma porte va bouger sur l'axe Y en 150 sec.

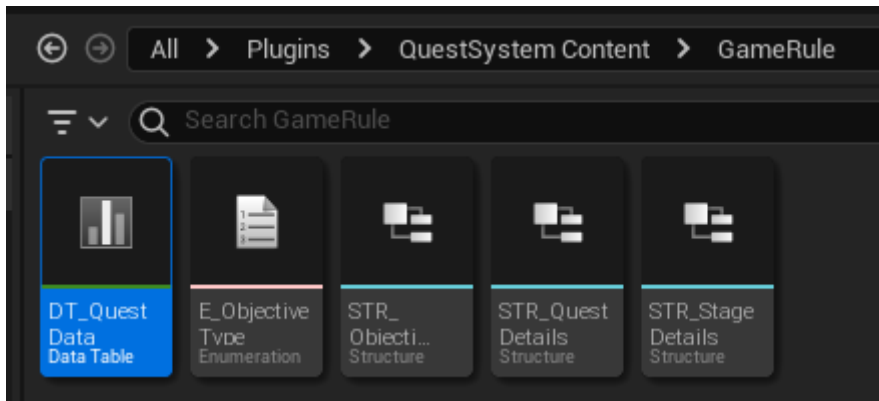


Une fois tous cela fait, il vous suffira de le renseigner dans la partie « **LevelSequence** » de l'objet pour qu'il soit utilisé lors de son activation.



- Quête Système :

Pour que le système de quête fonctionne, il vous suffit seulement d'éditer le tableau de quête « **DT_QuestData** » qui se trouve dans les plugins.



Il y a de nombreuses choses qui peuvent être modifiées comme par exemple, si l'on veut que se soit une quête principale « **isMainQuest** » ou encore si la quête doit être acceptée de base « **AutoAccept** ».

10001	
▼ 10001	
QuestName	Main Quest
LogDescription	main quest
TrackingDescription	main quest
IsMainQuest	<input checked="" type="checkbox"/>
▼ Stages	1 Array element
▼ Index [0]	6 members
StageName	get key
Description	get the key for the door
Objectives	5 Array elements
XPReward	0
ItemRewards	0 Map elements
CurrencyReward	0
AutoAccept	<input checked="" type="checkbox"/>
AutoComplete	<input checked="" type="checkbox"/>
AutoTrack	<input checked="" type="checkbox"/>

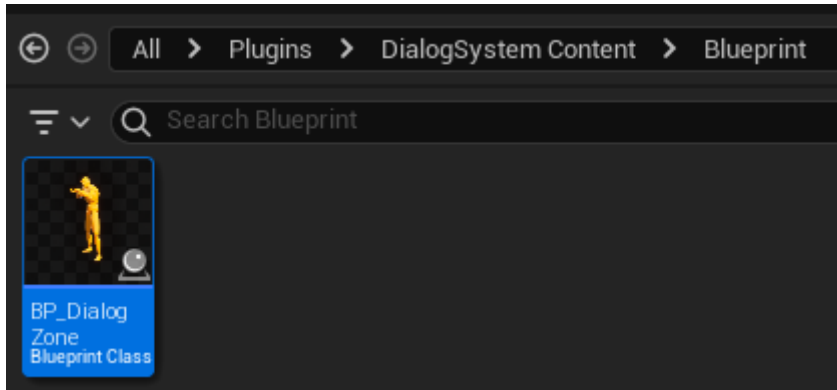
Mais la chose la plus importante à comprendre quand vous voulez créer votre propre quête est le tableau de « **Stages** » qui est un regroupement de grande étape dans votre quête qui regroupera surtout le tableau « **Objectives** » qui lui, correspond au différent objectif de l'étape. Ces objectifs peuvent être de 4 type différent : **Location, Interact, Collect, Kill**. Ils ont tous un « **ObjectiveID** » qui doit correspondre au « **NameID** » de votre objets ou ennemi et la variable « **Quantity** » correspondant donc à la quantité nécessaire pour valider l'objectif. Pour vraiment bien comprendre je vous invite à regarder comment est faite la quête du prototype.

▼ Stages	1 Array element
▼ Index [0]	6 members
StageName	escape
Description	escape from the dungeon
▼ Objectives	5 Array elements
▼ Index [0]	7 members
ObjectiveName	kill robot
Description	kill all robot
Type	Kill
ObjectiveID	robot
Quantity	1
IsOptional	<input type="checkbox"/>
BonusXP	0
▼ Index [1]	7 members
ObjectiveName	get key
Description	get the key for the door
Type	Collect
ObjectiveID	key
Quantity	1
IsOptional	<input type="checkbox"/>
BonusXP	0
▶ Index [2]	7 members
▶ Index [3]	7 members
▶ Index [4]	7 members
XPReward	0
ItemRewards	0 Map elements
CurrencyReward	0

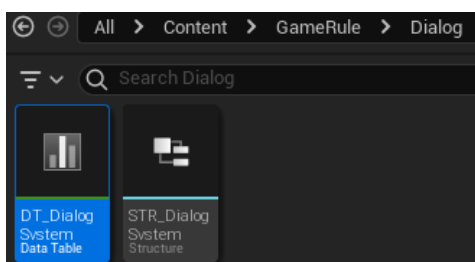
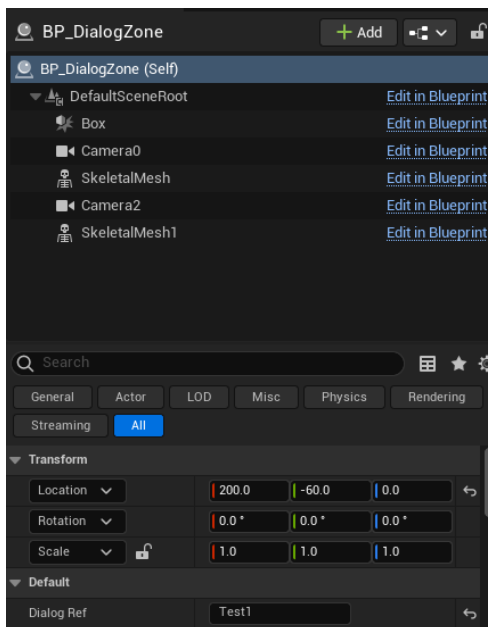
- Dialogue Système :

Pour la partie du système de dialogue, vous aurez besoin de 2 choses. Dans un premier temps vous devrez utiliser « **BP_DialogZone** » se trouvant dans le plugin

« **DialogSystem** ».



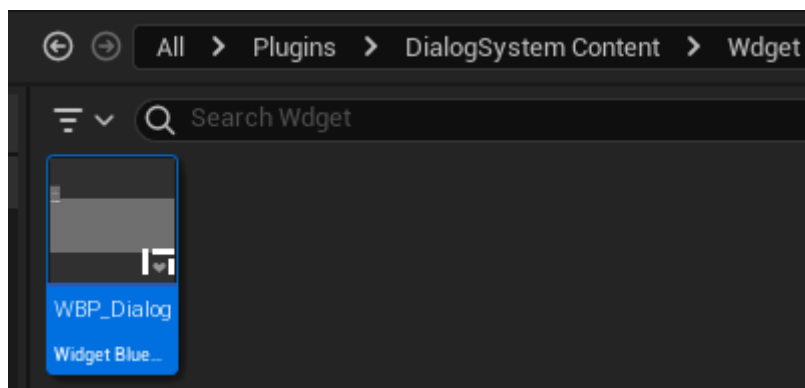
Puis il vous suffira d'ajouter autant de personnages ou de caméra que vous voudrez pour votre dialogue. Il vous faudra juste penser à vous rappeler l'ordre de vos caméras, car il faudra qu'elle corresponde lors de la création de votre dialogue. Pour pouvoir charger votre dialogue il suffira de faire correspondre « **DialogRef** » à un dialogue du tableau « **DT_DialogSystem** ».



Puis il vous suffira de créer votre propre profile de dialogue qui devra correspondre à votre valeur de « **DialogRef** ». Une fois cela fait pour chaque dialogue vous pourrez configurer, qui parle, de quoi et l'index de la caméra à utiliser.

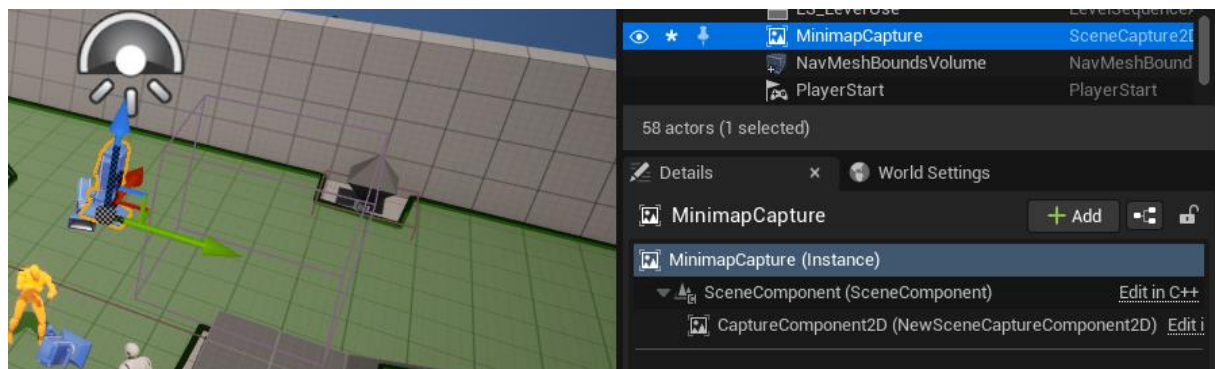


Si vous voulez changer l'aspect visuel du widget de dialogue, vous le retrouverez au nom de « **WBP_Dialog** »

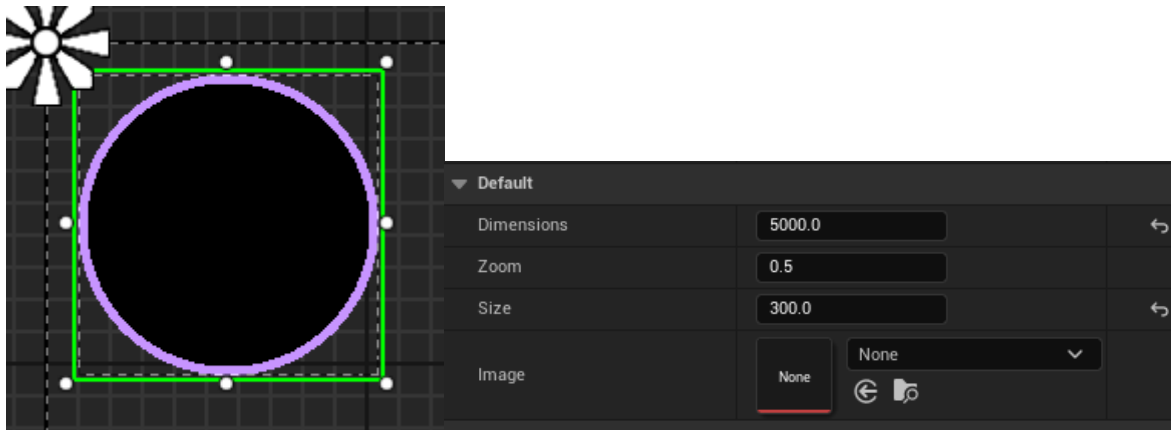


- Minimap Système :

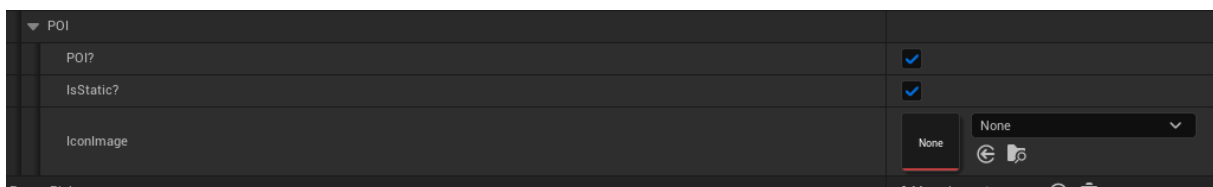
Le système de minimap est attaché à l'HUD, si jamais vous ne voulez pas l'utiliser, il vous suffit de l'enlever. Pour modifier l'aspect visuel de la minimap, il vous suffira de bouger l'acteur de la scène « **MinimapCapture** »



Vous pourrez configurer votre minimap comme vous voulez. Il faudra seulement que les valeurs dans « **WBP_HUD** » soit bonne. La « **Dimensions** » correspond à « **Ortho Width** » de « **MinimapCapture** ». « **Zoom** » correspond donc au zoom appliqué sur la caméra. « **Size** » doit correspondre à la size que vous lui donnez dans l'HUD. Et enfin la variable « **Image** » vous permet de remplacer le visuel par l'image que vous aurez fait au préalable par exemple.

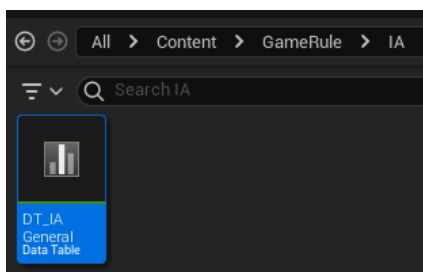


Vous aurez également la possibilité de rajouter des « **POI** » (Point Of Interest) au Props. Il vous suffit d'activer « **POI ?** » dans « **DT_Props** ». Vous pourrez également changer l'image si vous ne voulez pas utiliser le point par défaut ou encore si vous voulez que le point soit static, c'est-à-dire qu'il soit en permanence afficher sur la minimap.



- **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.

Category	Sub-category	Members
IAGeneral	AI_SelectPosition	0 Map elements
IAGeneral	AI_RandomPosition	1 Map elements
IAGeneral	Default (under AI_RandomPosition)	2 members
IAGeneral	RadiusRandom (under Default)	1000.0
IAGeneral	IA_RandomPositionAttack	2 Map elements
IAGeneral	Default (under IA_RandomPositionAttack)	10 members
IAGeneral	Cac (under IA_RandomPositionAttack)	10 members

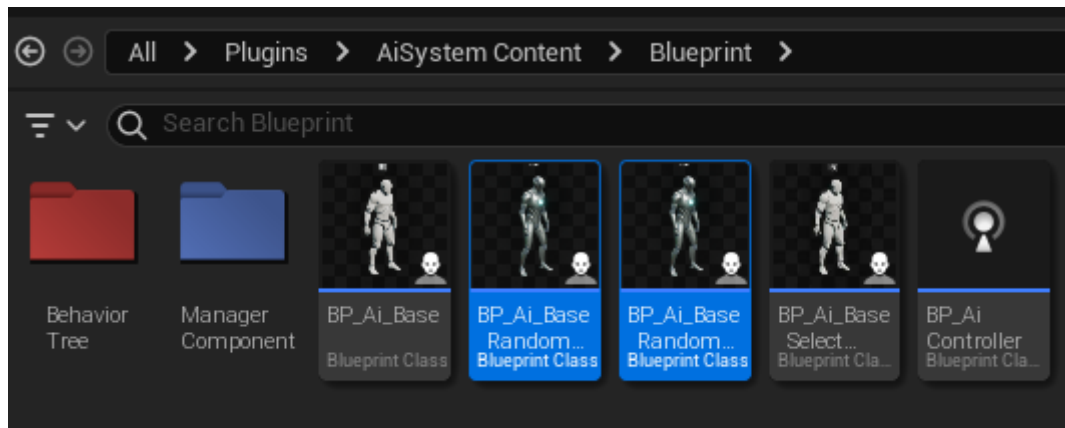
Pour les IA bougeant seulement de façon aléatoire, leur distance maximale de déplacement, vous pourrez changer leur nombre de point de vie, leur dégât et leur vitesse.

Category	Value
AI_RandomPosition	1 Map elements
Default	2 members
RadiusRandom	1000.0
BaseValue	
Health	20
Damage	10
Speed	300.0

Pour ce qui est des IA pouvant attaquer, il y a de nombreuses autres valeurs à gérer. Comme savoir si ce sont des rangs ou des mêlées, leur vitesse d'attaque ou encore leur vision de détection. Mais également les stats des projectiles.

Category	Value
Default	10 members
RadiusRandomPoint	1000.0
IsCac?	<input type="checkbox"/>
IsRange?	<input checked="" type="checkbox"/>
DelayAttack	1.0
VisionAngle	50.0
VisionDistance	4000.0
WeaponDrop	BP_Pickup_Weapon
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 jeux d'Aventure. N'hésitez pas à ouvrir les différents fichiers pour mieux comprendre le fonctionnement mais faite attention à rien casser.