**SUJET :**

Il faut : partir d'un nouveau projet template FPS ou TPS sans starter content et que les notions suivantes vues en cours soient implémentées :

- Utilisation des touches ZQSD pour le déplacement

- Menu avec 2 onglets (Main et Settings - widget switcher)

- UI avec : barre de vie, compteur d'items récupérés et timer

- Au moins 2 items utilisant une même interface

- Un line trace (qui peut servir pour appeler une fonction dans l'interface mais pas nécessairement)

- Un trigger

- Un timer by event (ou by function)

Attention, il se peut que je vous pose des questions sur l'architecture de votre code, sur la désactivation du tick ainsi que sur les paramètres des variables (catégorie et si elle est privé par exemple)

**Ne pas faire de bonus avant d'avoir fait les éléments demandés au risque d'être pénalisés !**

**Exemple de Projet :**

- Faire un labyrinthe simple avec les cubes d'unreal

- Le joueur a un temps limité pour trouver la sortie (timer by event)

- Ce temps est affiché dans l'UI

- Mettre sur les murs des interrupteurs qui allument des lumières qui indiquent le chemin : lorsque c'est le bon chemin, la lumière est d'une couleur, mauvais chemin d'une autre couleur

- Certains passages sont bloqués par une porte

- Il faut trouver les items "clé" qui ouvrent les portes

- En un premier temps toutes les clés ouvrent toutes les portes, bonus : chaque clé est liée à une porte spécifique, autre bonus : les portes ont plusieurs verrous (il faut 3 clés pour ouvrir une porte), possibilité de faire les 2 bonus : chaque verrou a un clé spécifique

- Au sol il y a des clous qui font diminuer la vie du joueur si on marche dessus

- Ajouter une barre de vie dans l'UI

- Bonus : si on reste sur les clous la vie diminue toutes les 0.5 secondes (timer by event)

- Sur le chemin il y a des trousses de soin qui font récupérer de la vie lorsqu'on les récupère

- Si la vie est à 100% on ne récupère pas la trousse de soin avec lequel on interagit

Nous avons donc :

3 items qui utilisent l'interface : les portes, les clés et les trousses de soin

1 blueprint en trigger : les clous

2 utilisations du timer by event : le temps qui s'écoule et les dégâts

1 compteur qui affiche : le nombre de clés qu'on a

UI avec : temps, compteur de clés et vie

Une utilisation du line trace : sur les items (porte, clé et trousse de soins)

**PRÉSENTATIONS**

*Notre dernier créneau est de 4h par groupe, les 6 et 7 juin. 2 heures seront réservées pour les présentations, je vous laisserai donc les 2 premières heures pour finaliser vos supports et éventuellement voir les bugs de dernière minute (mais vous devrez arriver avec une build prête). D'ici là vous pouvez me contacter par mail :* [*noellie.velez@esiee.fr*](mailto:noellie.velez@esiee.fr) *si vous êtes bloqués. Essayez tout de même de trouver les solutions par vous-même ou parmi vos camarades avant de me contacter, je resterai toutefois disponible dans la mesure du possible.*

Vous aurez 5 à 15 minutes de présentation par groupe où chaque personne doit présenter la partie qu'elle à développé en 5 minutes. Un support de présentation est demandé.

Démarrez les présentations avec une démo du projet réalisé (ne pas la faire sur le moteur mais à partir d'un exécutable - package en mode shipping).

Ensuite, vous devez présenter les notions demandées et avoir le moteur ouvert en background pour pouvoir répondre aux éventuelles questions.

Seront notés :

- La qualité de la présentation (préparation, locution, support et temps de parole)

- La réalisation des notions demandées

- La compréhension des notions utilisées (explications)

- Bonus ou malus selon ce qui à été réalisé

**Attention**, les 15 minutes comprennent le temps de questions/réponses, veuillez ne pas dépasser le timing.

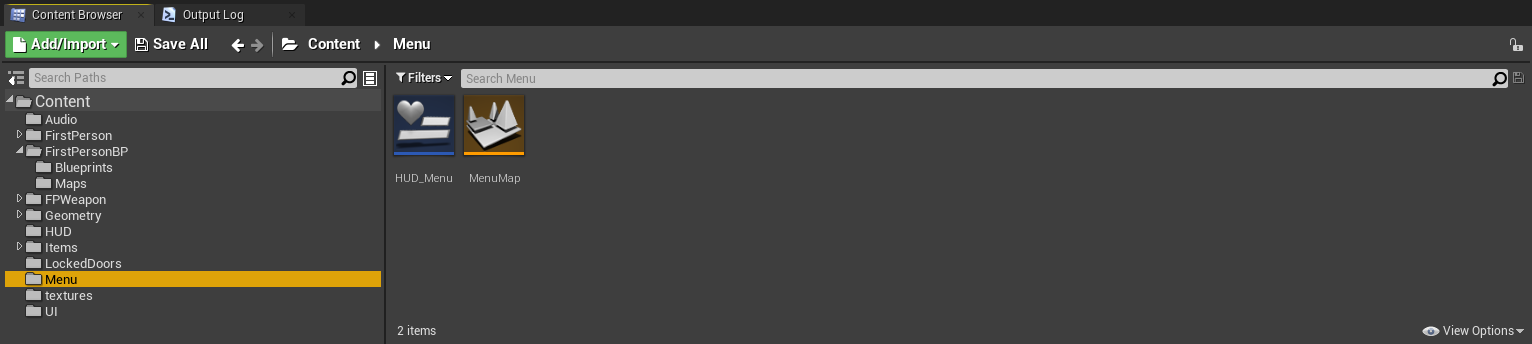
Présentation :

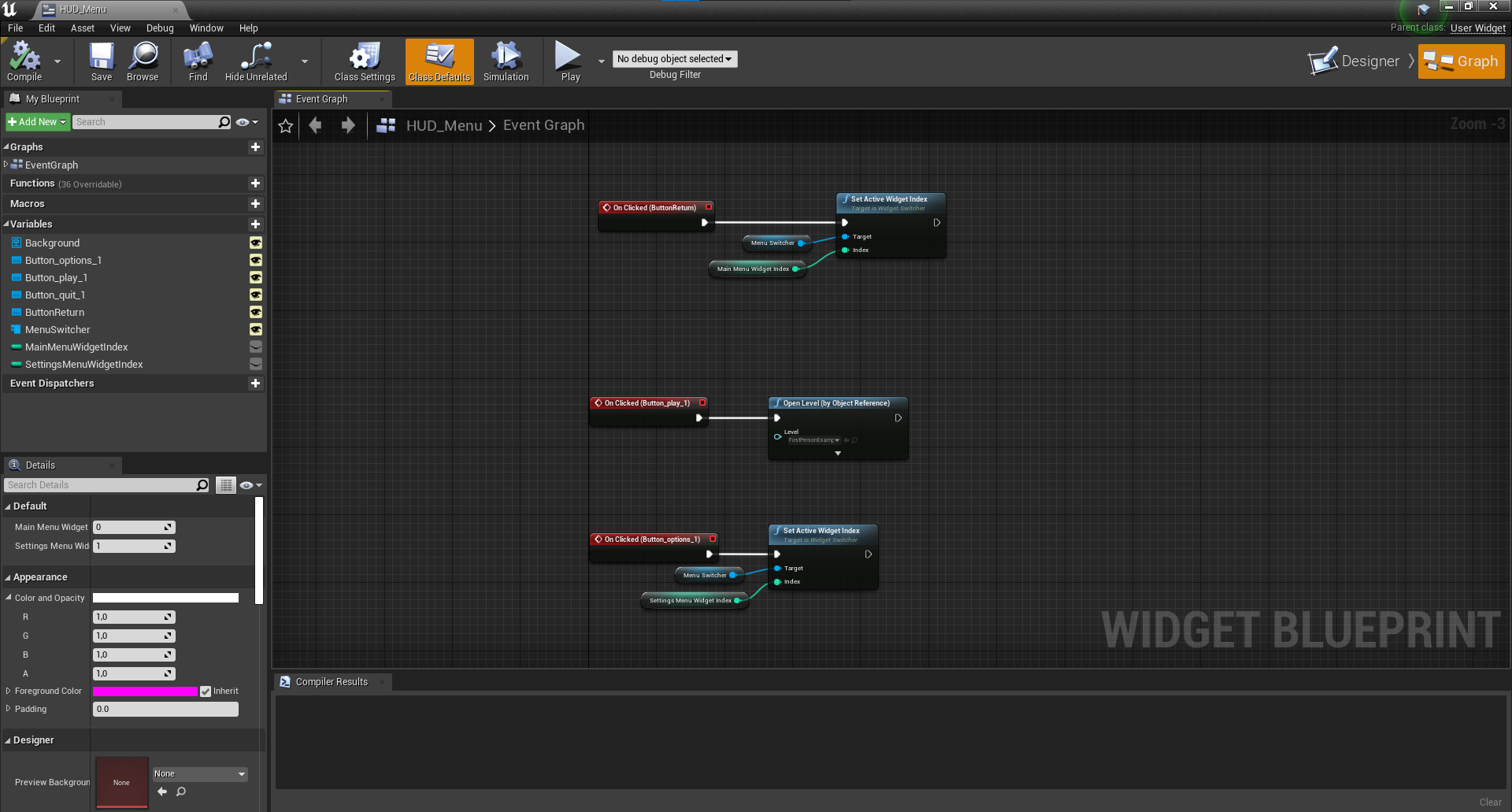
**SUJET :**

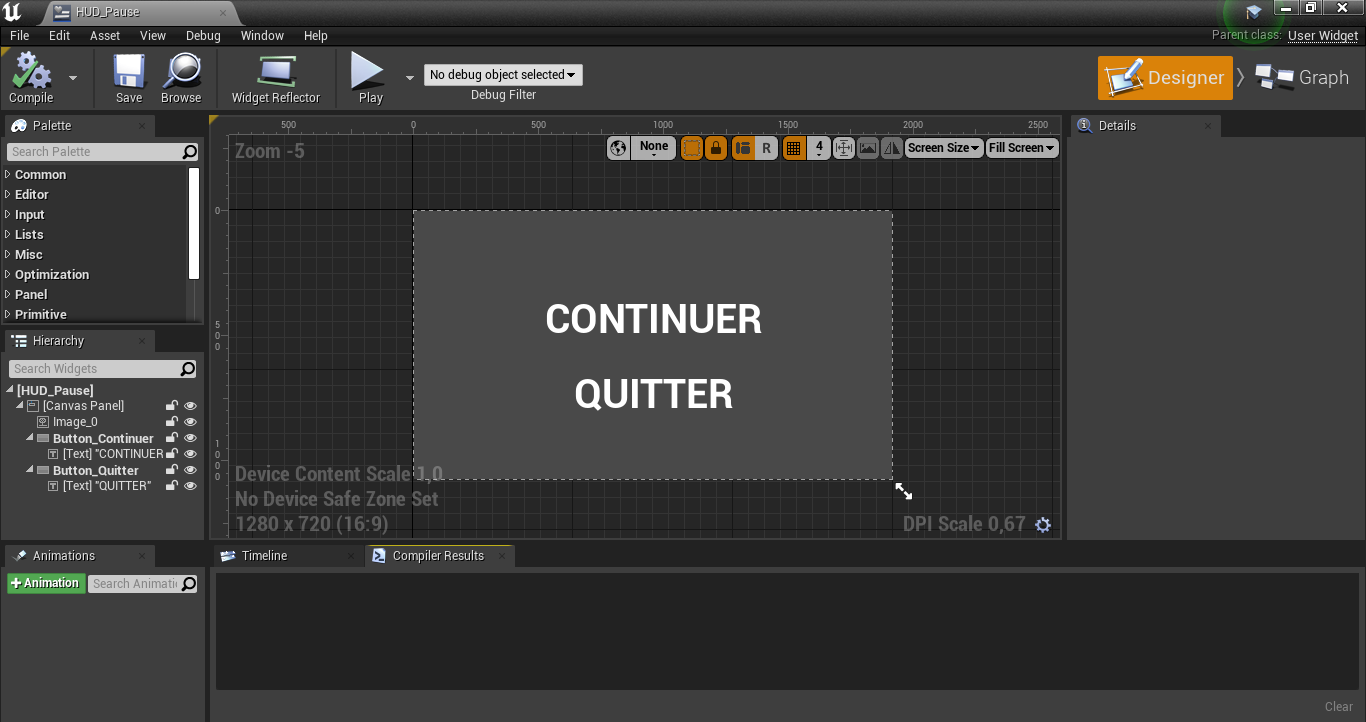
Il faut : partir d'un nouveau projet template FPS ou TPS sans starter content et que les notions suivantes vues en cours soient implémentées :

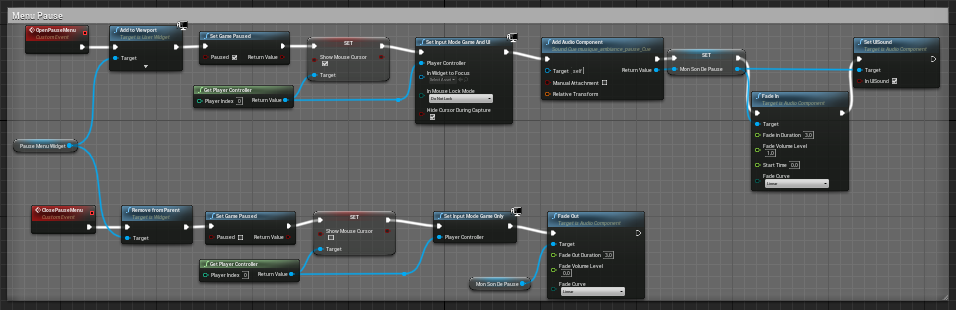
- Utilisation des touches ZQSD pour le déplacement —----------------> Fais vérifier que ça fonctionne dans le projet de val.

- Menu avec 2 onglets (Main et Settings - widget switcher) —--------> Menu de départ ! vérifier que ça fonctionne et que l’on arrive bien sur le menu de départ, éventuellement rajouter quelque chose dans les settings !



On a un level précis pour le menu du jeu avec le HUD associé.  
  
Et on utilise le widget switcher comme demandé :   


En plus, nous avons un menu pause, avec de la musique qui disparaît en fondant (en appuyant sur p).  


Dans le first player HUD :   


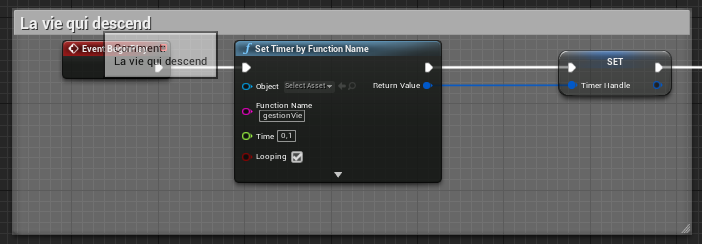
- UI avec : barre de vie, compteur d'items récupérés et timer

- Un timer by event (ou by function)

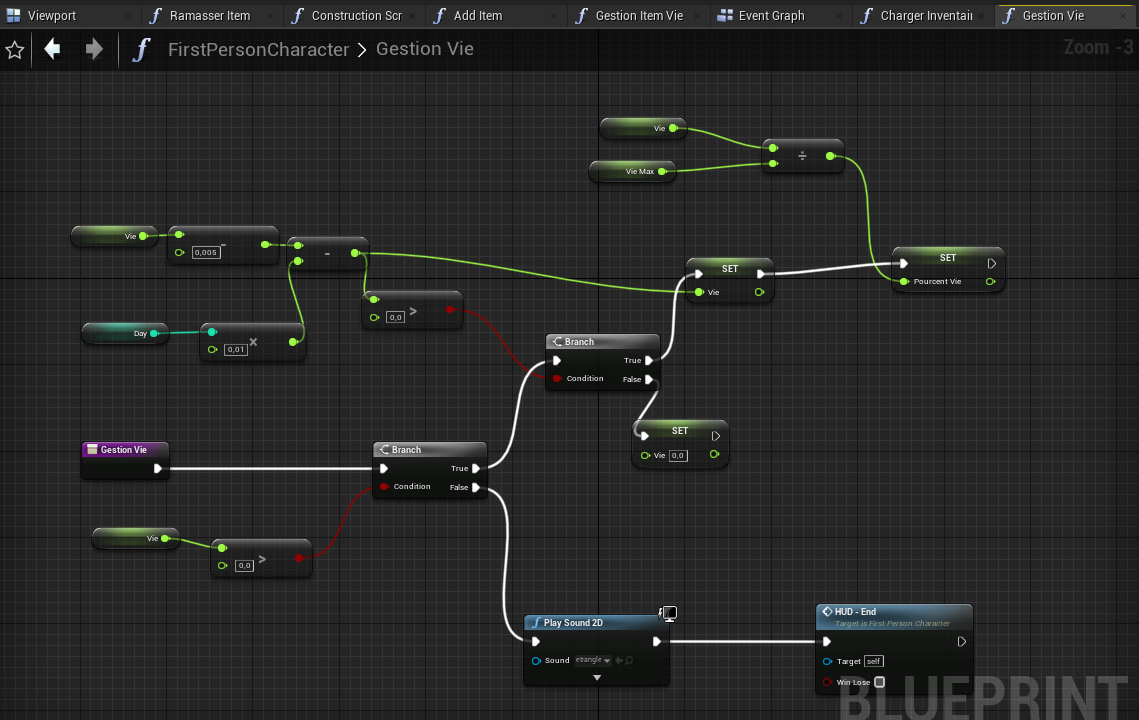
- Au moins 2 items utilisant une même interface

- Un line trace (qui peut servir pour appeler une fonction dans l'interface mais pas nécessairement)

Barre de vie, rapport à la santé mentale, qui diminue en fonction des jours (timer) qui s'écoulent. Plus le scientifique passe de temps seul, plus il perd la tête.

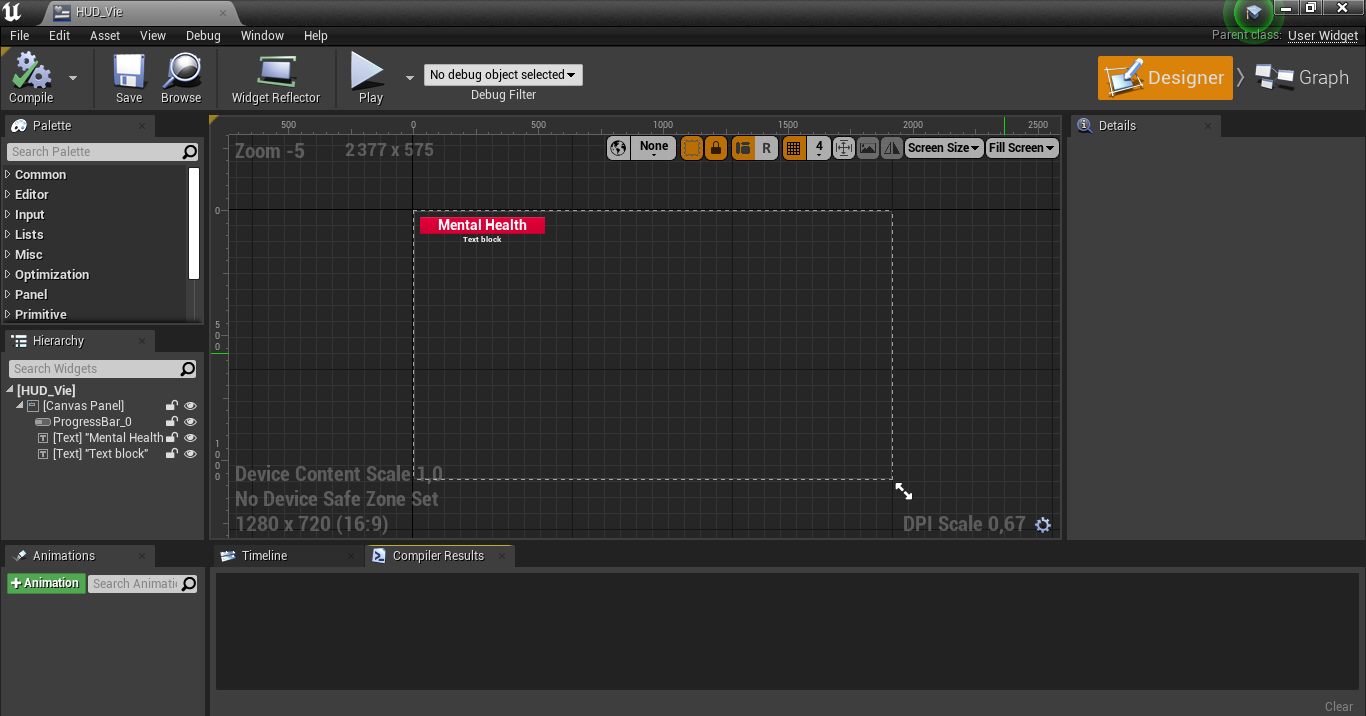
Code : 

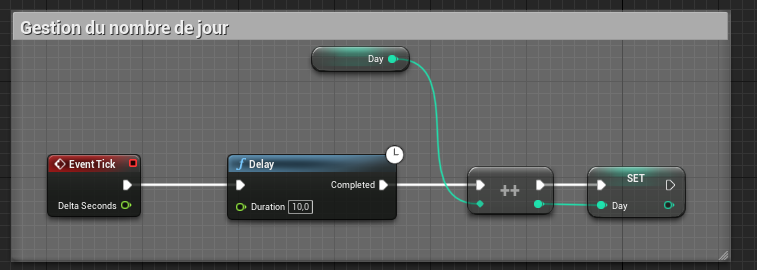
Appel de la fonction gestionVie :



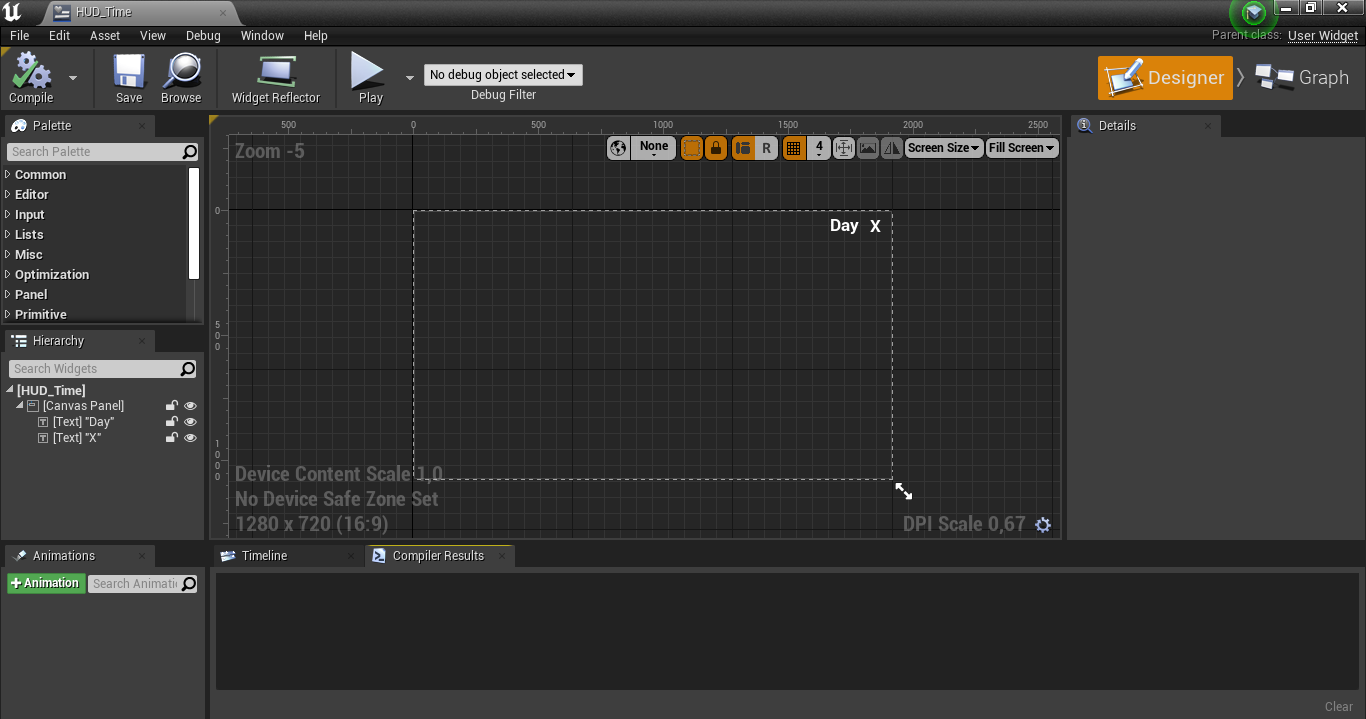
Si tout le temps s’écoule, le scientifique devient fou et un son de mort étranglé est alors joué et le Panel de fin est alors activé. Sinon la vie décrémente petit à petit en adéquation avec le nombre de jours passés.

Affichage :

  
Cette barre de vie est donc reliée au nombre de jours qui s’écoulent, autre UI en haut à droite.



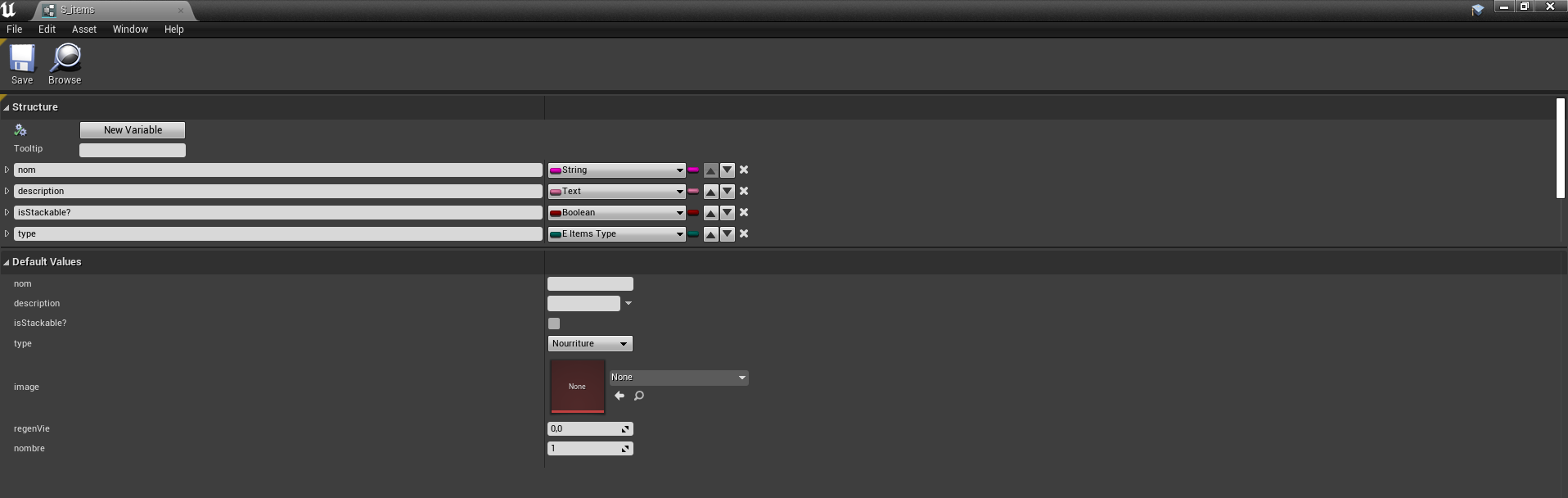
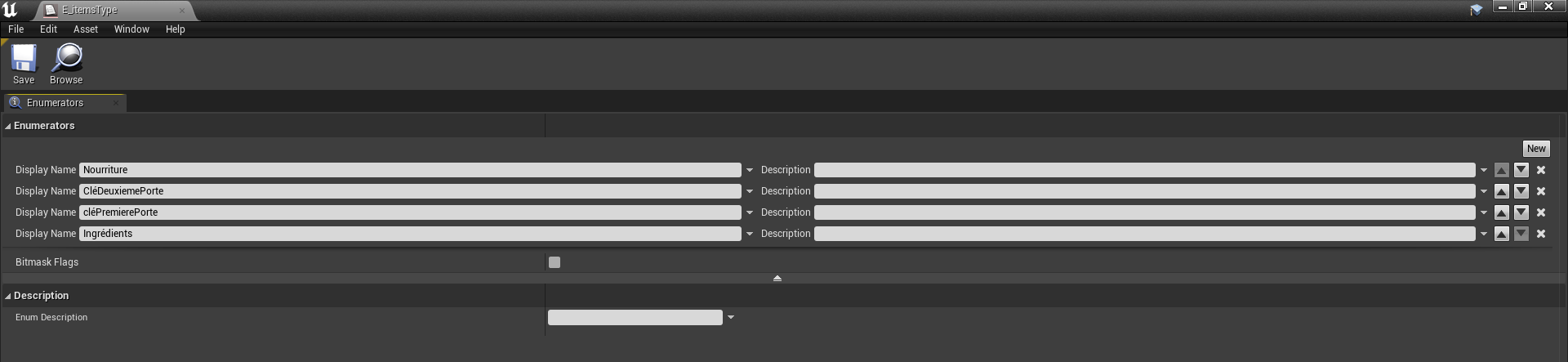
Variable dans le BP du joueur. Qui s'incrémente toutes les 10 secondes.

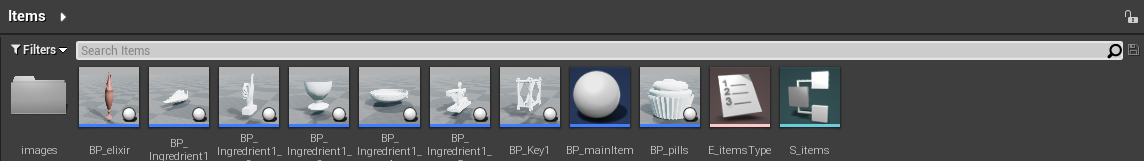
Affichage : 

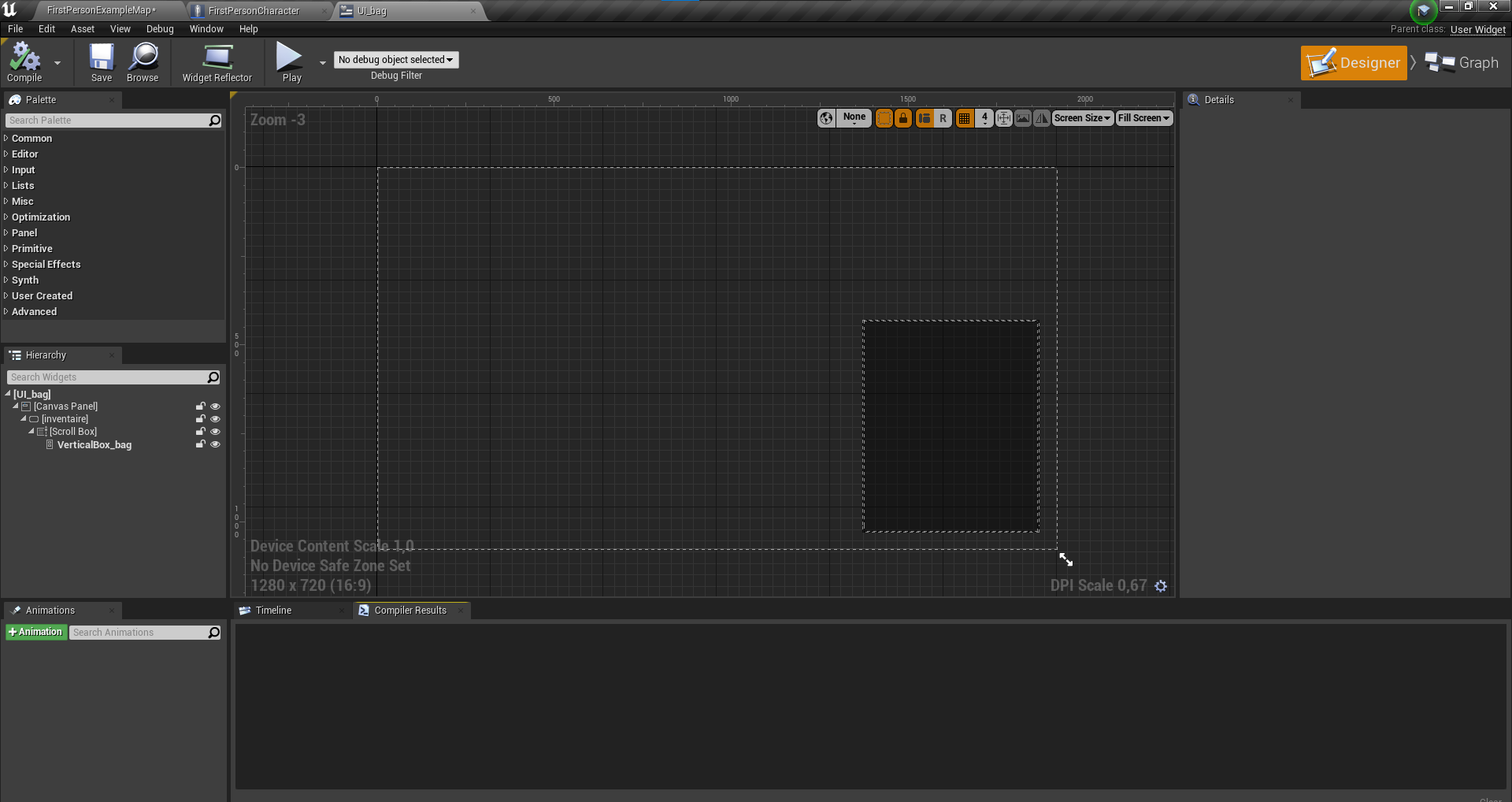
Pour le compteur d’item nous avons opté pour un inventaire qui s’active en appuyant sur TAB. Bien évidemment, avec ce fonctionnement, au moins 2 items utilisent la même interface. Pour ramasser les items on utilise le line trace. Le timer by event est utilisé pour la vie qui descend.

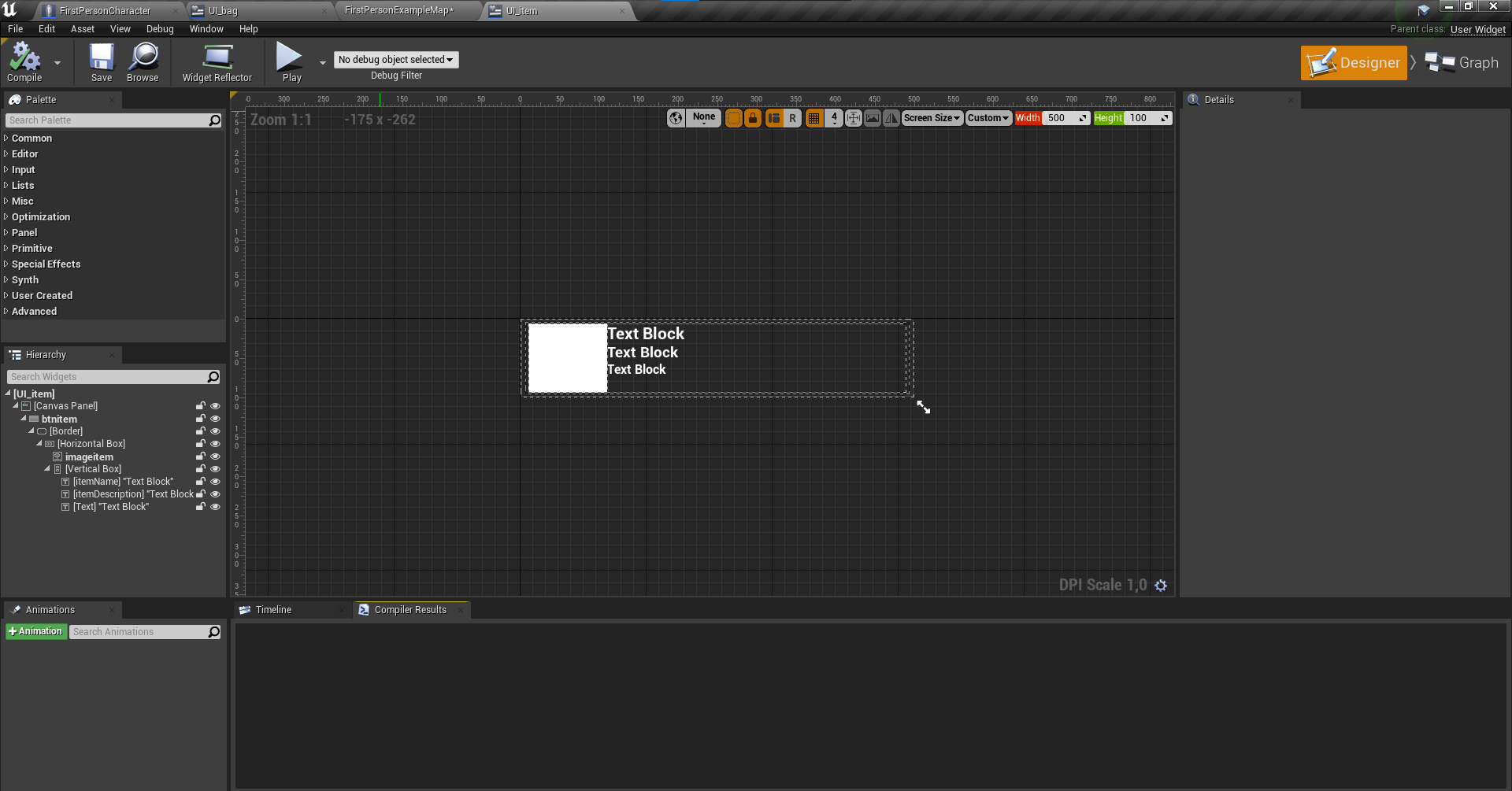
Affichage :   
  
 Mettre une capture d’écran du jeu de l’inventaire !

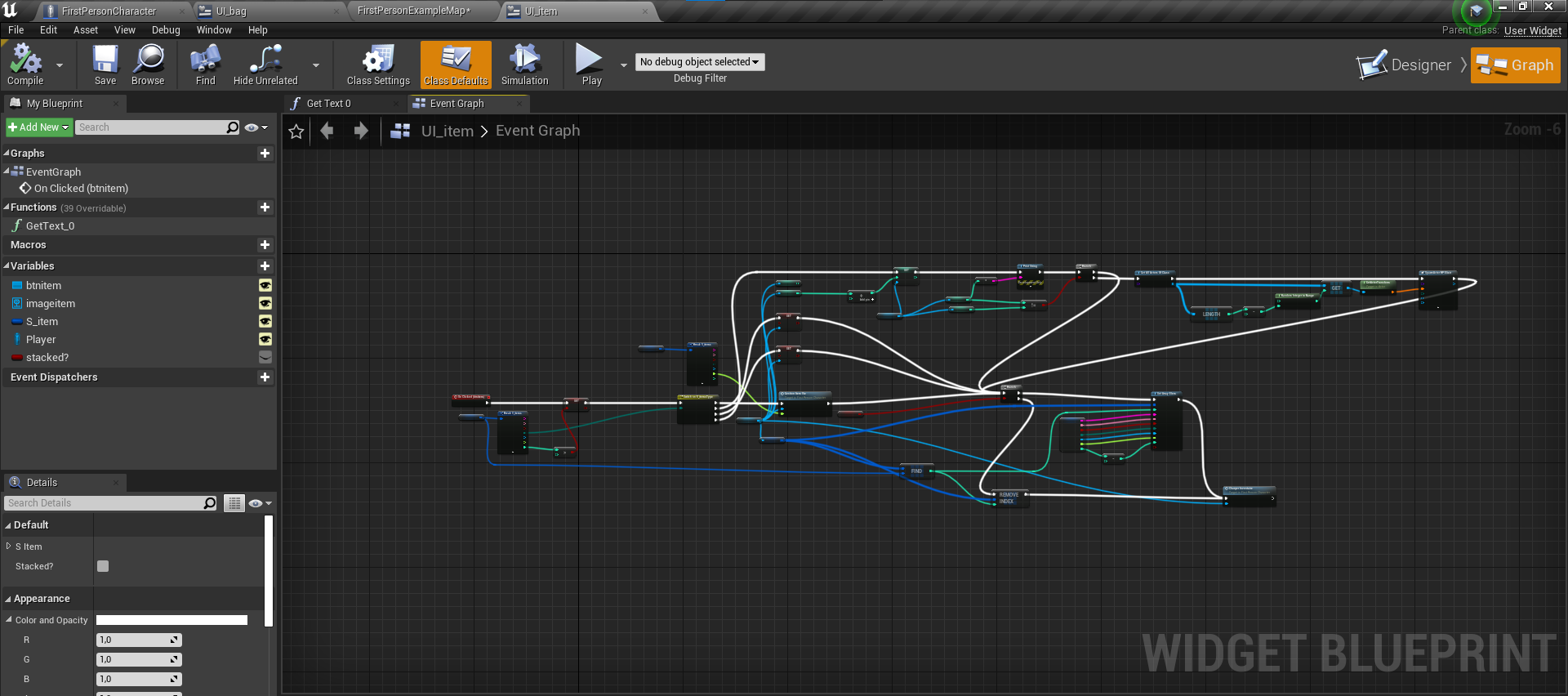
Code :

En premier lieu, nous avons une structure item qui définit les différents champs de chaque item :   
  
En deuxième temps nous avons une énumération des différents types d’item (sorte de classe) :   


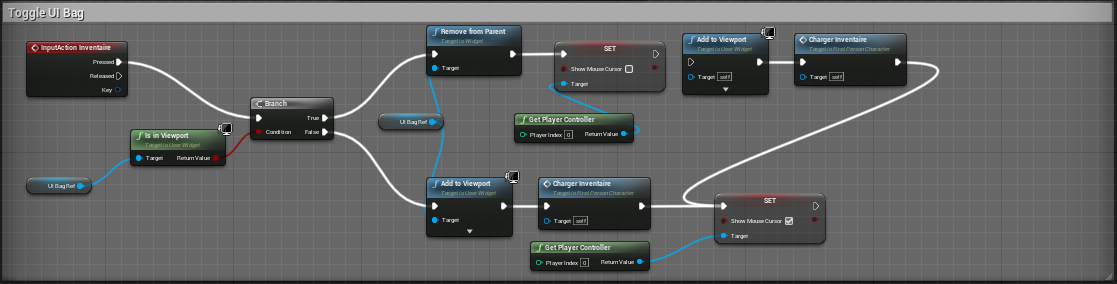
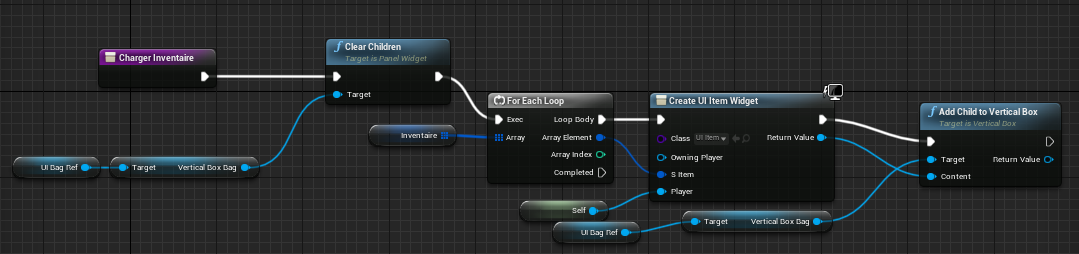
Ensuite on a les blueprints des différents items qui hérite tous du BP\_mainitem (si on veut mettre des choses en communs à tous les items) :   


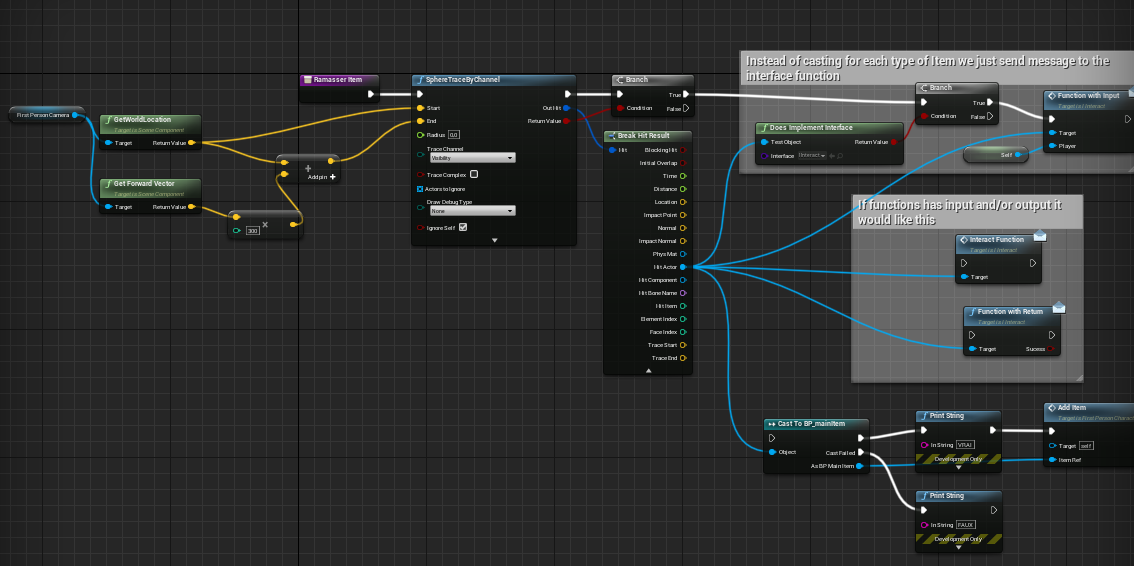
On a deux UI différentes pour les items. La globalité de l’inventaire avec tous les items listés et UI\_item qui compose juste la ligne avec comment sont disposés les informations des items dans l’inventaire :   




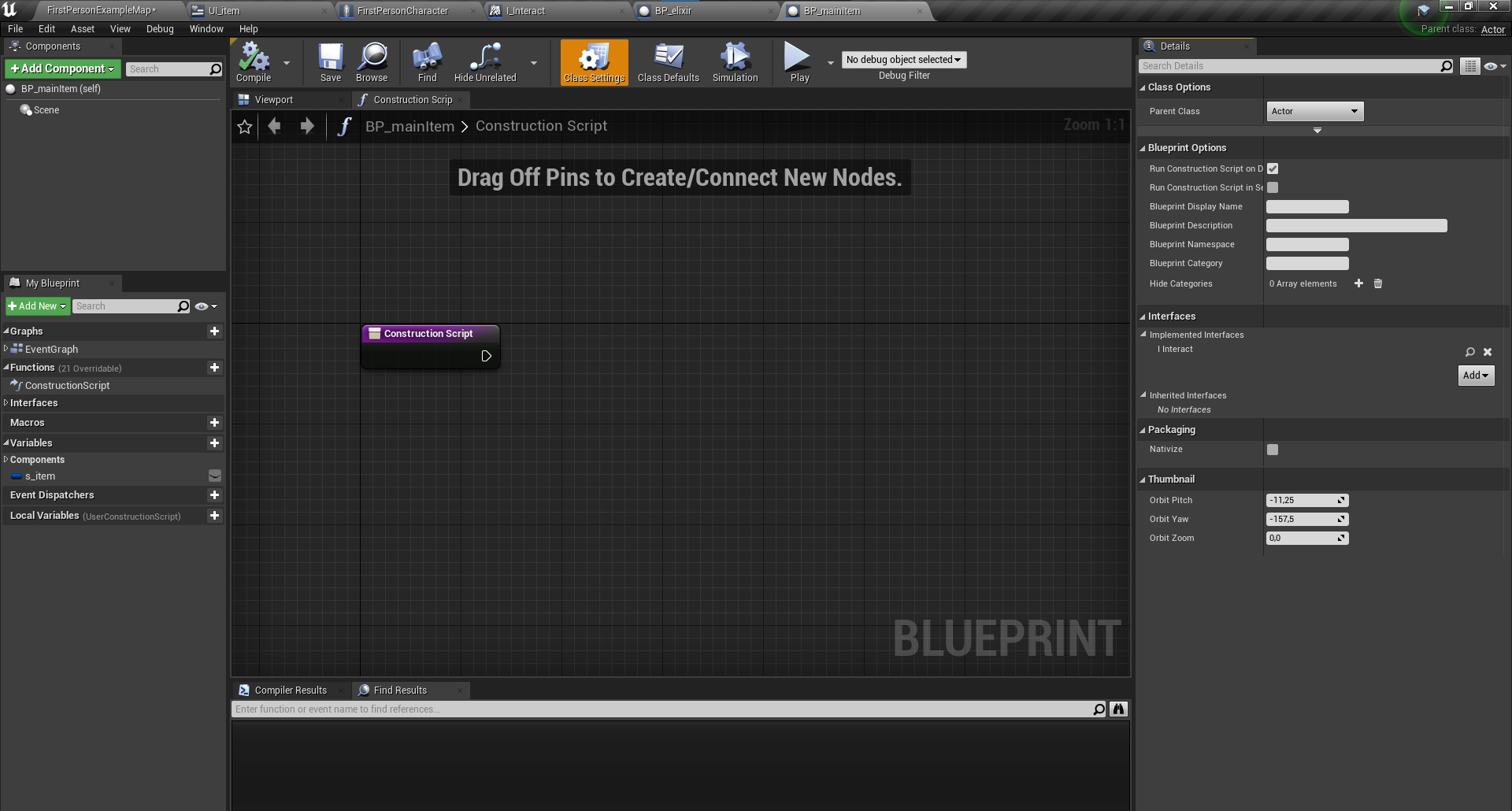


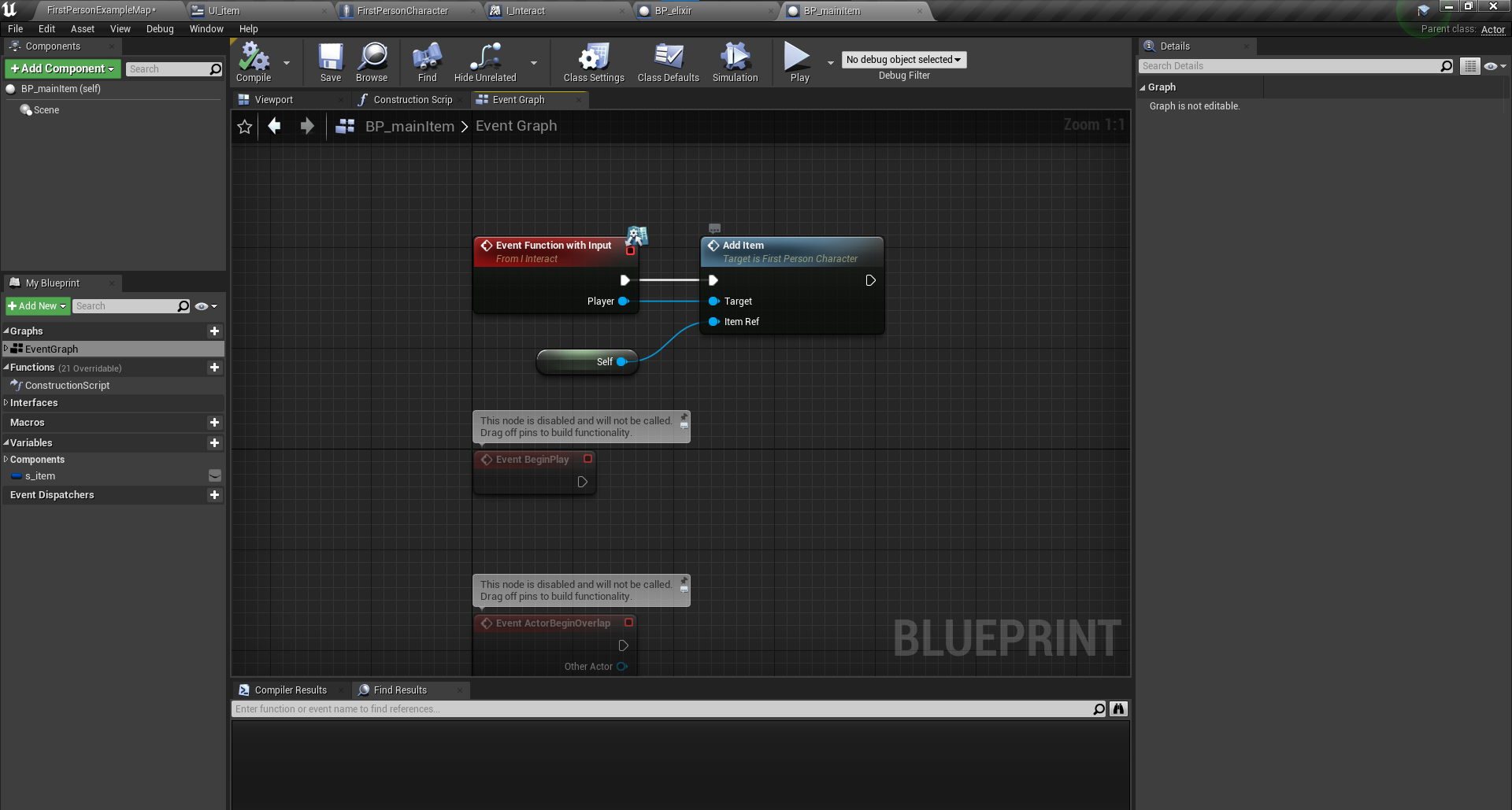
Le code de cet UI est assez complexe, car en fonction du type d’item qui est utilisé, une action différente doit avoir lieu.

Voici le code pour ouvrir et fermer l’inventaire qui fait appel à la fonction charger inventaire :   
  


Enfin, nous avons la fonction ramasser items qui utilise le line tracer pour compléter l’inventaire au fur et à mesure avec l’utilisation d’une unique interface pour tous les objets :   


Cette fonction fait appel à l’interface qui fait appel à la fonction add item (on ne la mets pas dans les screenshots car trop grande et on ne voit rien sur les screens) permettant d’ajouter l’item dans l’inventaire :

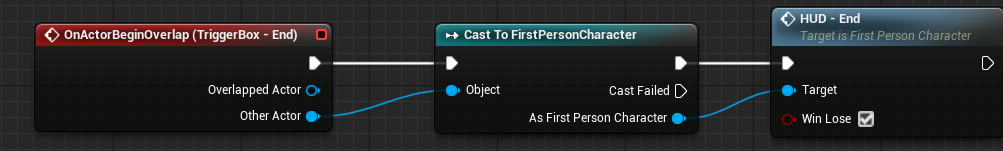




- Un trigger

Le trigger quant à lui est placé à la fin. Si le joueur entre dans la zone du trigger alors il remporte la partie. Il faut ouvrir le level blueprint pour voir la situation de victoire.

Code :



Enfin il y a les locked door (cf. video tuto).