

Layers of Fear

Cristina ZHENG

Analyse théorique :

1. De quel type de jeu s'agit-il ?

Il s'agit d'un jeu d'horreur psychologique à la première personne, singleplayer.

2. Après une heure de jeu, nommez les différentes mécaniques de gameplay rencontrées.

Les différents types de mécaniques sont :

- Le déplacement en 2D (on ne peut pas sauter)
- Le mouvement de caméra en 2D qui commence lent et s'accélère ensuite
- Le zoom de caméra
- L'ouverture d'armoires/tiroirs/portes
- L'analyse d'objet quand on interagit avec
- Le changement d'environnement selon où on se trouve dans une pièce
- Le changement d'environnement selon où on regarde
- La fermeture automatique des portes

3. Ce jeu mise son attrait sur l'immersion du joueur. Expliquez en détail comment cela est mis en place (à l'aide de quelles mécaniques, feedbacks, ambiance,...).

Layers of Fear est un jeu qui mise son attrait sur l'immersion du joueur tout d'abord par le fait que ça soit un jeu à la première personne, du point de vue du personnage directement. L'atmosphère sombre du manoir nous plonge directement dans cet esprit d'horreur. Nous en savons autant que le personnage, augmentant notre curiosité à aller chercher les éléments qui nous aideront à résoudre le mystère du jeu. A l'aide de ce point de vue, nous pouvons remarquer que les mouvements de caméra sont en accord avec le personnage que nous jouons. En effet, quand nous sommes immobile, nous pouvons sentir sa respiration et parfois une légère inclinaison de la caméra, tout comme quand nous marchons : nous pouvons deviner qu'il boite du pied droit. Nous découvrons par une ordonnance que le personnage s'est fait une prothèse à une jambe, mais qu'elle n'était pas de la bonne taille. De plus, au cours du jeu, sa canne apparaît plusieurs fois.

La mécanique des changements de salles et des éléments qui bougent dans les salles elles-mêmes est une manière facile de procurer des sursauts (jumpscares) car nous ne nous attendions pas à ce qui peut surgir derrière une porte ou quand nous tournons la caméra. De plus, cette mécanique nous rend tout aussi confus que le personnage dans ce labyrinthe qui semble infini. Nous ne savons pas ce qu'il va se passer dans chaque

mouvement que nous faisons et il n'y a aucun repère spatial. Néanmoins certaines pièces reviennent, nous indiquant que nous sommes toujours dans le même manoir.

Dans ce labyrinthe, nous sommes cependant toujours guidés de manière subtile : il n'y a pas d'éléments extradiégétique, mais des éléments qui semblent étranges/qui bougent dans le décor et des lumières pour attirer notre regard et faire continuer le jeu ou simplement des portes pour nous indiquer où aller. Les portes sont soit non verrouillées (suite du jeu) soit fermées (peuvent être décoratives, nécessité d'une clé ou la salle qui n'a pas encore été totalement explorée). Et les feedbacks du jeu, comme le déverrouillage d'une porte, nous permettent de comprendre quand nous avons débloqué quelque chose.

Au fur et à mesure du jeu, à chaque fois que nous retournons dans l'atelier du peintre avec la toile, nous comprenons que nous avons fini un "chapitre" du jeu. Les éléments dans l'armoire à sa gauche quand il peint nous permettent de voir l'avancée du jeu et combien d'éléments importants il reste à découvrir.

4. Le gamefeel du jeu passe par énormément de petits détails. Qu'est-ce que le gamefeel ? Donnez une liste des détails que vous avez perçus durant votre partie.

Le gamefeel est la sensation que nous ressentons quand nous jouons au jeu. Il s'agit d'émotions, de sensations et de ressentis que provoquent le jeu par ses différentes mécaniques, son ambiance et tous les détails qu'il peut inclure.

Un des détails que nous pouvons citer pour *Layers of Fear* est que, comme mentionné précédemment, nous voyons ce que le personnage voit, de son point de vue aux mouvements de la caméra, tout est fait pour que nous soyons mis à sa place. Les bruits de pas ressemblent à des battements de cœur, rendant chaque moment très anxieux.

De plus, les effets sur la vision du personnage nous font bien ressentir le mal-être dans lequel il était. Nous avons tout d'abord des pièces qui "bougent", puis des hallucinations, des visions de couleurs vives comme si nous étions sous drogue.

Et l'atmosphère sombre du manoir ne fait que s'alourdir : il y a de plus en plus d'éléments bizarres comme des écritures morbides, des peintures défigurées, des pièces totalement délabrées, de la musique qui se lance tout seul, des voix et même des "esprits" (nous pouvons deviner qu'il s'agit de la femme du peintre et de leur enfant).

Tout cela crée des sentiments de peur (de ce qu'il peut arriver), de stress, de mystère. Mais aussi de culpabilité (quand nous découvrons ce que le peintre a pu faire aux personnes autour de lui), de dégoût et de confusion. Le jeu joue beaucoup sur les jumpscare aussi pour nous procurer des moments de sursaut.

5. Expliquez comment est mis en place le game over dans ce jeu.

Je n'ai rencontré le game over qu'une fois durant mes sessions de jeu : en tombant de la bibliothèque dans le bureau. Le jeu n'ayant pas de logique spatiale a donc su me faire respawn de manière fluide dans une pièce menant là où je m'étais arrêtée. Comme les nombreuses autres fois où j'ai été téléporté dans d'autres pièces pour continuer le jeu, ma vision s'est assombrie et est revenue. La transition était tellement similaire que j'ai eu des doutes sur le fait que je sois vraiment mort.

6. Layers of Fear essaye de faire perdre le joueur dans son labyrinthe dynamique. En effet, les salles "bougent" toutes seules. (un exemple à partir de ce moment de la vidéo : <https://youtu.be/fMgmb-J0Guc?feature=shared&t=1705>)

Expliquez en détail et en termes de programmation comment cela est mis en place.

Nous avons donc la room 1 où nous sommes et nous devons aller dans la room 2 (qui a "bougé") par une porte.

Méthode 1 : la porte est un portail

Il faut tout d'abord créer une classe actor Portail pour stocker les informations relatives au portail. Il va donc gérer : s'il est actif ou inactif (porte ouverte ou fermée), le côté du portail que le joueur traverse, la position du portail dans la room 1, la position du portail d'arrivée dans la room 2, la position du joueur pour savoir s'il a traversé la porte, la caméra du joueur pour pouvoir renvoyer l'image de la room 2 avec le bon angle à travers la porte ouverte.

Si le joueur ouvre la porte :

Portail actif vrai

Si le joueur ferme la porte :

Portail actif faux

Si portail actif :

Si à un Tick la position du joueur > position du portail d'entrée :

On téléporte le joueur à la room 2 avec sa position/rotation

Si à un Tick la position du joueur < position du portail de sortie :

On téléporte le joueur à la room 1 avec sa position/rotation

Si le joueur est bien entré dans la zone de la room 2 :

Fermer la porte derrière lui

Changer le portail 2 par le portail de la room suivante (incrémenter)

Méthode 2 : simple téléportation

La map 2 de la room 2 contient la room 1.

On est dans la room 1 de la map 1. Quand on est proche de la porte, on se fait téléporter dans la room 1 de la map 2, avec notre position relative à la porte pour qu'on ne voit pas le changement.

Méthode 3 : spawn la room 2

Quand la porte est fermée, on delete la room 0 et fait apparaître la room 2 derrière la porte.

7. Personnel : le jeu vous a-t-il plu / déplu ? Pourquoi ?

Le jeu m'a plu, mais sans plus. J'ai trouvé l'immersion bien faite avec les nombreux détails présents. Mais beaucoup trop long pour un gameplay aussi simple. Au bout de la moitié du jeu, j'ai commencé à le trouver très répétitif et je voulais juste avancer dans l'histoire, cet ennui m'a déconnecté de l'immersion.

Pratique :

1. Définissez, selon vous, une des mécanique de gameplay principale du jeu, et reproduisez là sous le moteur de jeu Unreal Engine 5 (vous pouvez faire cette mécanique sur un projet de n'importe quel type, en 2D ou 3D, du moment que l'on "ressent" une similitude avec la mécanique originale).

Selon moi, la mécanique du mouvement de la caméra et de la marche. La caméra bouge légèrement quand il est en idle, représentant son souffle. Et d'un rythme lent et saccadé, il marche en boitant, créant un effet de caméra non symétrique que ça soit sur le pattern qu'il fait ou sur la vitesse quand elle monte et quand elle descend. En effet, le mouvement penche d'un côté et est plus lent quand il monte que quand il retombe.

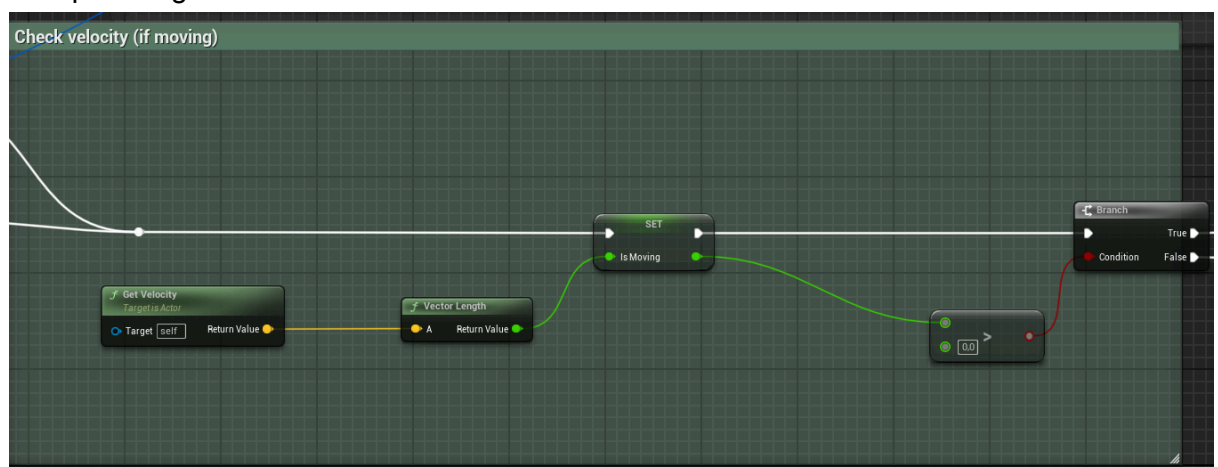
2. Décrivez, étape par étape, comment vous avez mis en place cette mécanique.

Tout d'abord j'ai créé 2 Camera Shake pour le idle et la marche, BP_CamShakeIdle et BP_CamShakeMove.

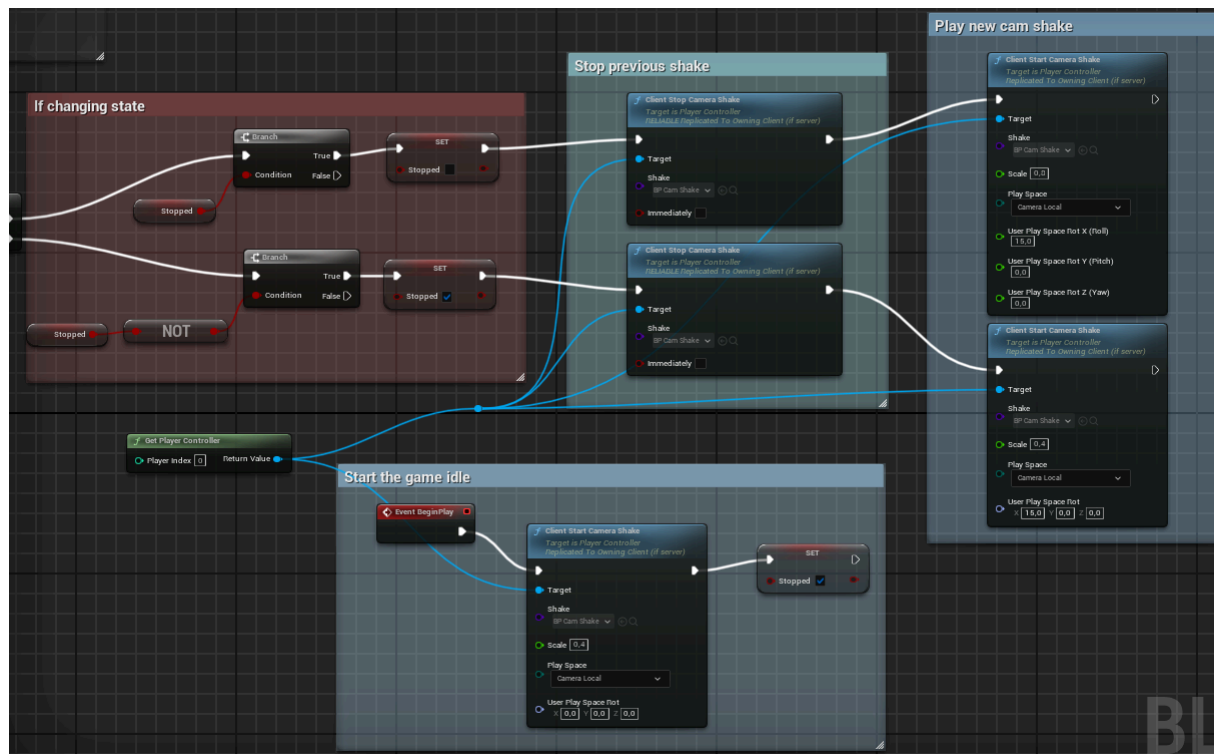
Pour qu'ils puissent se boucler, j'ai mis la durée à 0s.

Ensuite dans le *BP_FirstPersonCharacter* :

- EventBegin → je lance le *BP_CamShakeIdle*
- Il me fallait la première frame de quand il bouge/s'arrête pour lancer et arrêter mes Camera Shake. J'ai donc check la vitesse du joueur. Si la vitesse $\neq 0$, alors cela veut dire qu'il bouge.

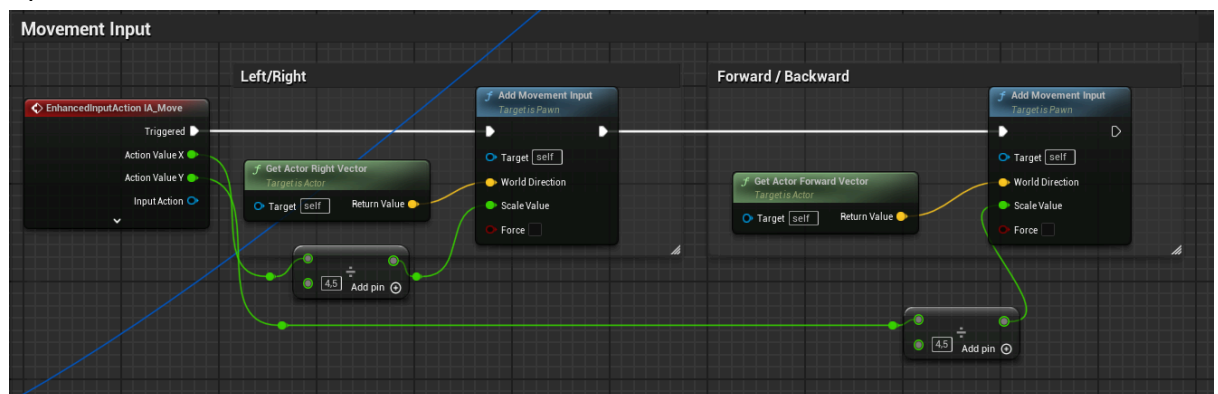


- EventTick → si isMoving alors je stop *BP_CamShakeIdle* et je lance *BP_CamShakeMove*; sinon je stop *BP_CamShakeMove* et je lance *BP_CamShakeIdle*

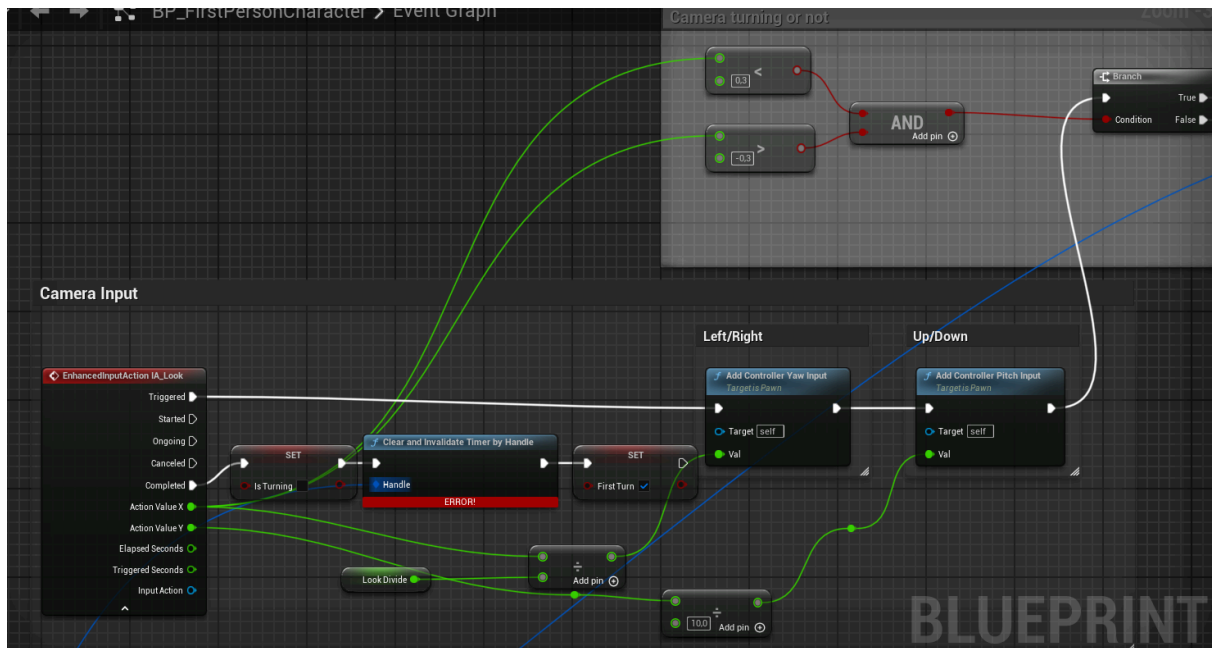


J'aurai pu faire un AND booléen pour les conditions.

- Pour la vitesse de déplacement, j'ai simplement divisé les valeurs envoyées par les inputs.

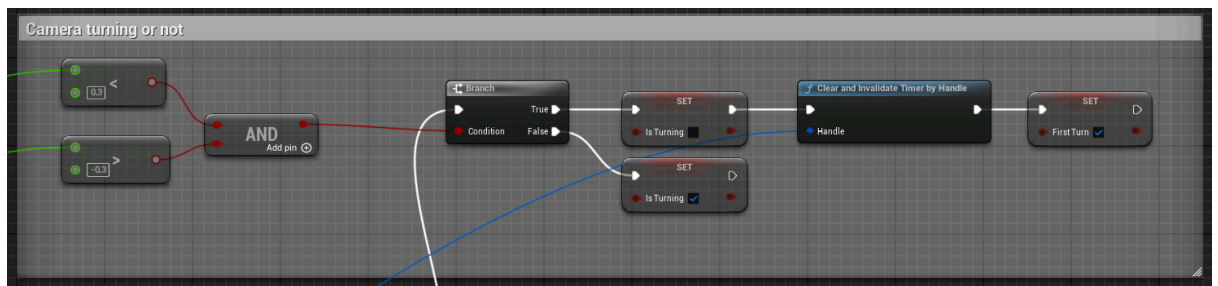


Après avoir réussi à régler le changement entre le idle et le move, je me suis dit pourquoi pas faire le changement de vitesse dans la caméra aussi. Cette partie fut plus compliquée que la première, car dans l'événement *IA_Look*, le plus petit mouvement de caméra était détecté comme un "event started" et je ne pouvais pas utiliser le "started" pour lancer mon timer pour l'accélération de caméra. J'ai donc pris les valeurs des inputs horizontaux et tant qu'ils ne dépassent pas une certaine valeur, alors je les néglige.



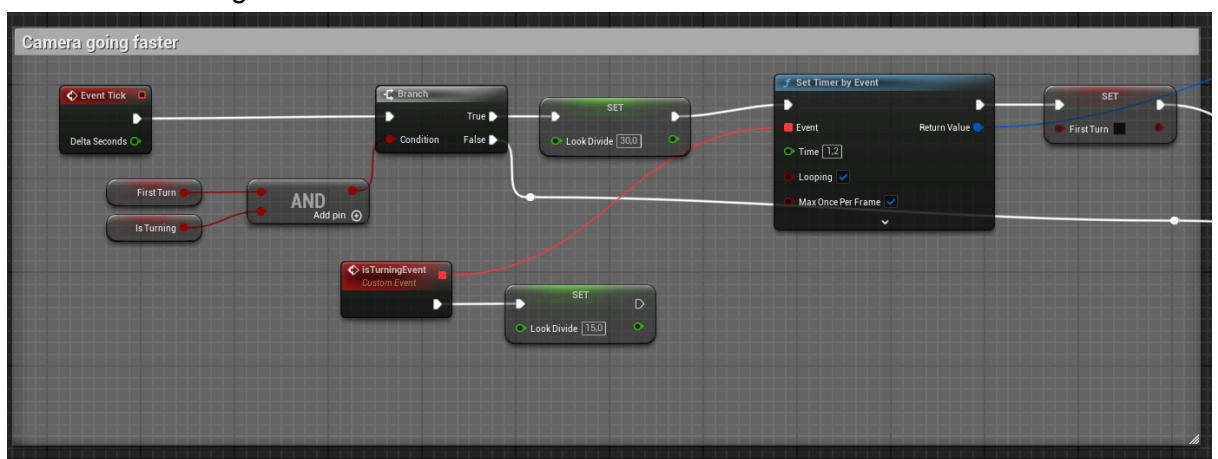
Et puisque encore une fois, je n'ai plus le moment où il commence à tourner sa caméra, il m'a fallu un nouveau booléen : FirstTurn.

- Si le joueur arrête l'action IA_Look, alors on reset le timer et le FirstTurn (car la prochaine fois qu'il va tourner, ça sera son FirstTurn).



Ainsi je peux procéder à mon algorithme :

- Si le joueur est en train de tourner et c'est son FirstTurn, alors on démarre la vitesse de la caméra à lent, on lance le timer qui va changer la vitesse et on reset FirstTurn à false.
- Sinon on ne change rien.



3. Expliquez en quoi cette mécanique est primordiale tout au long du jeu.

Cette mécanique est primordiale, car c'est celle de base : on peut regarder autour de nous et on peut bouger. Le jeu repose sur cela, étant un jeu d'horreur où il faut explorer et qui n'a pas d'autres moyens de se déplacer.

4. Pourquoi cette mécanique vous a-t-elle plu ?

Il s'agissait d'un détail auquel on ne pense pas toujours, mais qui est totalement logique. J'ai pu deviner que le personnage boitait avant même de le lire dans le jeu et cela m'a impressionné.