Document de Prototypage

Contexte de la Démo : Le joueur doit survivre face à des vagues d’ennemis très basiques. Il peut se déplacer, sauter, et courir, ainsi qu’utiliser un panel d’armes à sa disposition.

**Mécanique choisie :**

Système d’armes à roulement. Le joueur possède un certain nombre d’armes aux propriétés variées. Via un système de roulement des armes, il peut sélectionner n’importe quelle arme parmi toutes celles qu’il possède.

**Implémentation Technique :**

L’implémentation de ce système d’armes s’est faite sur la base de cinq scripts majeurs.

GunSwitch : (MonoBehaviour)

Le premier GunSwitch, qui permet le roulement des armes. C’est un script très simple qui prend en Input la molette de la souris, et augmente ou diminue un index de sélection des armes en fonction de la direction de l’input de la molette.

ShootBehaviour : (MonoBehaviour)

L’information du GunSwitch est ensuite utilisée par le script ShootBehaviour, afin de sélectionner la bonne arme, parmi une liste d’armes disponibles. Le script ShootBehaviour est responsable de l’exécution du tir, en prenant en compte tous les paramètres des différentes armes. A ce titre, ce script permet d’effectuer des tirs simples ou en continue, ainsi que des tirs en rafales, et également des tirs en cône, avec une randomisation des rayons dans ce cône. Il s’occupe ensuite d’envoyer des raycasts et/ou capsulecasts afin de détecter les surfaces touchées ainsi que les ennemis, et y appliquer ou non des dégâts, et des feedbacks.

ShootControls : (MonoBehaviour)

Ce script permet de s’abonner à un système d’Input customisé, en définissant une touche pour le tir, puis en recevant des callbacks dès que cette touche est pressée, et relevée. Le script ShootBehaviour utilise cette information afin d’enclencher les tirs.

AimBehaviour : (MonoBehaviour)

Ce script permet de viser en définissant une position et vitesse de rotation de la caméra ainsi que de mouvement du personnage, lorsqu’en visée. Le script ShootBehaviour utilise l’information de visée afin d’enclencher ou non les tirs.

GunMode : (ScriptableObject)

Enfin, les différents types d’armes sont créés via Scriptable Object, avec le script GunMode. Le GunMode permet de choisir entre un tir de type continu ou simple. De choisir entre un tir en rafale ou solo, ainsi que de choisir de faire un tir en cône, avec plus ou moins de rayons tirés. Et également la quantité de dégâts ainsi que la distance de tir maximale.

En plus de cela, il y a un système d’IA, composé de deux scripts.

EnemyManager : (MonoBehaviour)

Le premier script gère l’apparition de vagues d’ennemis, via des spawners définis dans l’éditeur, possédant un point d’apparition, ainsi qu’une range autour de se point. L’apparition se fait aléatoirement à l’intérieur de cette range et sur ces spawners.

Ce script possède une liste statique d’ennemis, permettant aux ennemis de s’ajouter eux-mêmes dedans, lorsqu’ils apparaissent, et s’y retirer lorsqu’ils disparaissent.

Enemy : (MonoBehaviour)

Le second script, gère la totalité de l’ennemi, sa vie, son comportement, et sa mort.

Le comportement utilise un NavMeshAgent, qui récupère la position du joueur et se déplace vers celui-ci. Lorsque l’ennemi est à portée, il attaque le joueur et lui fait perdre de la vie, si le joueur est dans sa zone d’attaque.

L’ennemi possède une méthode publique afin de recevoir des damages, en outre, il update lui-même l’affichage de sa vie, et a un comportement de mort avec des feedbacks.

**3 combinaisons de réglages d’armes :**

Automatic : Tir en continu rapide, pas de tir en cône, dégâts légers, distance de tir moyenne

Salve : Tir simple en salve de 2 coups, pas de tir en cône, dégâts élevés, distance de tir longue

Spread : Tir simple sans salve, tir en cône avec 5 rayons, dégâts moyens, distance de tir courte

**Problèmes de Tech/Design :**

Difficultés à coder rapidement un système de cône moins aléatoire, utilisant l’algorithme du sunflower, permettant de placer des points à équidistance les uns des autres, sur un cercle. Finalement opté pour la solution de facilité avec une randomisation des rayons. Malheureusement, la précision est assez mauvaise lorsqu’à distance de tir moyenne.

L’ajout de systèmes de particules avec HDRP, sans utiliser ShaderGraph. Le projet ainsi que son Controller TPS provenant d’un projet HDRP sur lequel je travaille. Et HDRP ne possédant pas de shaders de base pour particules.

L’édition d’animations importées dans le moteur qui étaient readonly à la base. Contournement du problème via de nouvelles animations.

Arme à salve beaucoup plus puissante que toutes les autres. Equilibrage cassé.

Faire un jeu TPS fun, modulaire, et original en 6h. Impossible.

Bugs :

Un bug de visée survenu aléatoirement. Impossible de sortir du mode visé. (Very Uncommon)

Un bug de tir survenu aléatoirement. Rayons envoyés au mauvais endroit. (Very Uncommon)