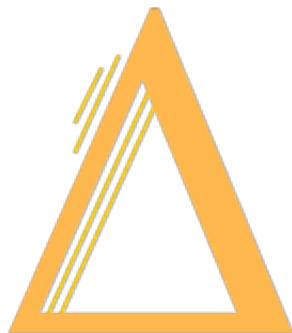


E.S.U.
Emergency Service Unit

Rapport de soutenance 2



Deltaplane

Louis Place
Maxence Oden
Nathan Rabet
Vincent Libeskind

Contents

1	Introduction	3
2	Avancements du projet	3
2.1	Mécanisme de combat	3
2.1.1	Mise en place du système de combat	3
2.1.2	Mitraillette	4
2.1.3	Hache	5
2.1.4	Pack de soin	6
2.2	Le système d'animation et de layers	7
2.2.1	Introduction	7
2.2.2	Le système de déplacement final	8
2.2.3	Les animations spécifiques	8
2.2.4	Le système de visée	10
2.2.5	Le système de Layers	12
2.3	Les Cartes	13
2.3.1	Carte n°1	13
2.3.2	Carte n°2	18
2.4	Pouvoirs Spéciaux:	20
2.5	Paint Tree et nouveaux éléments graphiques:	21
2.6	Modification du site Web:	21
2.7	Les musiques et les sons:	21
2.7.1	Choix artistiques	21
2.7.2	Développement technique des musiques	22
2.7.3	Les bruitages	23
2.8	Les menus et HUD:	24
2.8.1	Introduction	24
2.8.2	Le menu	24
2.8.3	Tableau des scores	25
2.8.4	La barre de vie	26
2.8.5	La barre de munitions	27
2.8.6	Les informations sur les armes	27
3	Conclusion	29

1 Introduction

Pour cette seconde soutenance nous avons suivi nos prévisions afin de réaliser de nombreux nouveaux ajouts qui permettent maintenant d'affirmer l'identité du jeu ainsi que son gameplay. Grâce à ses nombreux ajouts, notamment en matière de mécanique de gameplay, le jeu est maintenant beaucoup plus riche. Il offre au joueur un bon environnement de jeu que ce soit grâce aux nouveaux HUD, à l'ajout de la partie sonore, de nouvelles animations ou bien par les deux cartes qui se sont largement étoffées. Passé les heures compliquées comme les problèmes liés au Git, l'ensemble du groupe se montre maintenant encore plus efficace quant à la réalisation générale du jeu et à l'ajout de nouvelles fonctionnalités. Pour finir nous sommes tous confiant quant à l'avancement du projet.

2 Avancements du projet

2.1 Mécanisme de combat

2.1.1 Mise en place du système de combat

Tout d'abord, nous avons besoin d'un premier script qui doit gérer le statut et l'inventaire des personnages. Pour cela, nous avons créé un script nommé: "Player_Manager". Ce script contient la Classe, les Armes et la vie du personnage. Il a été conçu pour être modulable afin de n'avoir qu'un seul script pour toutes les classes.

```
62     //Donne les caractéristiques de la classe
63     switch(myClass)
64     {
65         case Classe.Policier:
66             maxhealth = 100;
67             weaponsInventory.Add(Armes.Hand);
68             weaponsInventory.Add(Armes.Pistolet);
69             break;
70         case Classe.Pompier:
71             maxhealth = 100;
72             weaponsInventory.Add(Armes.Hand);
73             weaponsInventory.Add(Armes.Hache);
74             weaponsInventory.Add(Armes.Extincteur);
75             break;
76         case Classe.Medecin:
77             maxhealth = 100;
78             weaponsInventory.Add(Armes.Hand);
79             weaponsInventory.Add(Armes.Medpack);
80             break;
81         case Classe.Mercenaire:
82             maxhealth = 100;
83             weaponsInventory.Add(Armes.Hand);
84             weaponsInventory.Add(Armes.Pistolet);
85             break;
86         case Classe.Pyromane:
87             maxhealth = 100;
88             weaponsInventory.Add(Armes.Hand);
89             weaponsInventory.Add(Armes.LanceFlamme);
90             break;
91         case Classe.Drogueur:
92             maxhealth = 100;
93             weaponsInventory.Add(Armes.Hand);
94             weaponsInventory.Add(Armes.Medpack);
95             break;
96     }
97     health = maxhealth;
98     weaponsInventoryLength = weaponsInventory.Count;
```

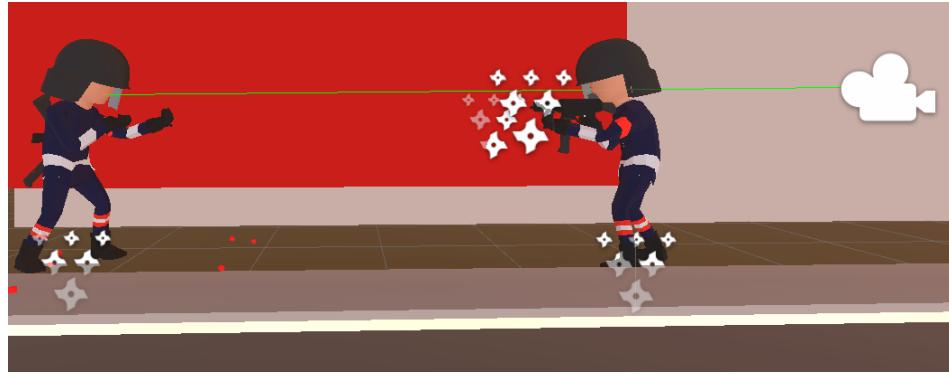
Code lors de la création du personnage

Ce script contient donc les fonctions fondamentales des personnages tel que: la perte de vie, le gain de vie, la mort du personnage et le roulement d'armes. Il va donc activer ou désactiver les scripts d'armes du personnage en fonction de l'arme en main.

2.1.2 Mitraillette

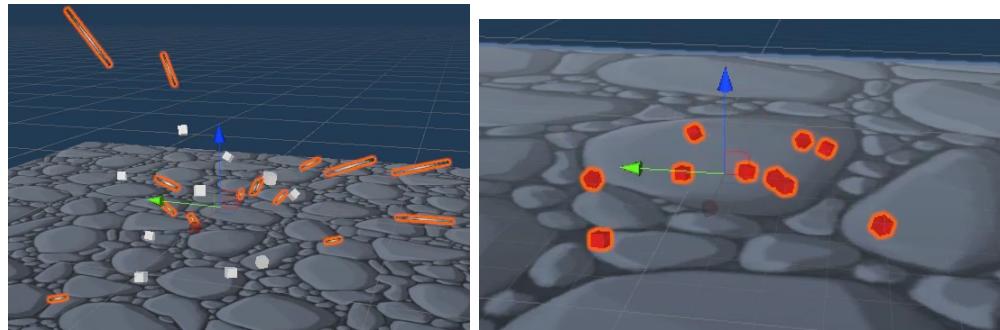
Les classes: “Policier” et “Mercenaire” ont une mitraillette. Nous avons donc un script qui s’occupe de celle-ci. Ce script contient les fonctions de tir et de rechargeement. Pour faire tirer l’arme, on utilise une fonctionnalité d’Unity

qui est les *Raycast*. Les Raycast permettent de détecter le premier obstacle en fonction d'un point de départ et une direction. De ce fait si l'obstacle est un joueur on lui enlève de la vie. Ci-dessous la représentation d'un *Raycast* symbolisé par un trait vert.



Exemple du Raycast d'un tir

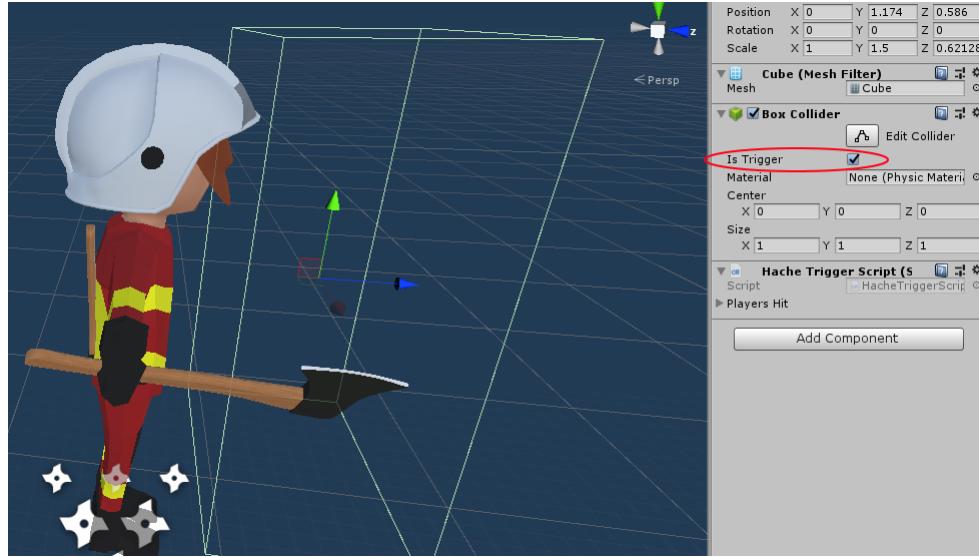
Afin de permettre au joueur de savoir où la balle est arrivé, nous avons ajouté d'effet de particule. Un quand un joueur est touché et un autre quand un objet quelconque est touché.



Particule d'impact sur une surface Particule d'impact sur un joueur

2.1.3 Hache

Pour la hache, le fonctionnement est différent car c'est une arme de corps à corps. De ce fait pour chaque coup de hache, nous avons besoin de récupérer les joueurs dans une certaine zone. Pour cela le pompier possède une boîte avec la case “Is Trigger” activé.

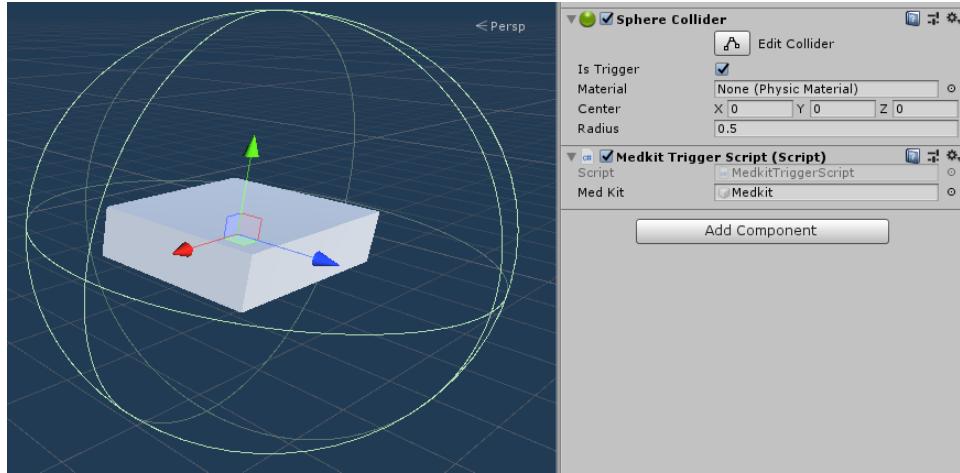


Zone de détection du Pompier

Cette case nous permet d'utiliser les fonctions “OnTriggerEnter” et “OnTriggerExit”. Ces fonctions nous permettent d'avoir les personnages qui sont actuellement dans cette zone. Il suffit donc de donner des dégâts à toutes les personnes dans cette zone à chaque coup de hache.

2.1.4 Pack de soin

Le pack de soin contrairement aux autres armes, est un objet à part entière. Cet objet est composé du modèle du pack de soin et d'une sphère de collision. Cette sphère nous permet, comme la hache, de récupérer les joueurs dans la zone et leur donner de la vie.

*Zone de détection du pack de soin*

Le script qui s'occupe d'attribuer le gain de vie est très simple:

```

12     void OnTriggerEnter(Collider other) {
13         if (canHeal && other.tag == "Player")
14         {
15             PhotonView view = other.GetComponent<PhotonView> ();
16             view.RPC("healing", RpcTarget.All, view.ViewID, 20);
17             Destroy(medKit);
18         }
19     }

```

Function de détection du pack de soin

Quand un objet rentre dans la sphère, on vérifie qu'il porte le “tag” Player. Si oui, il appelle la fonction de gain de vie “healing” à tous les clients. Puis il s'autodétruit.

2.2 Le système d'animation et de layers

2.2.1 Introduction

Le système d'animation a subi de nombreuses modifications entre la première et la seconde soutenance. Tout d'abord, l'implémentation des personnages a entraîné une réadaptation des animations. En effet, notre jeu possède différents types de personnages. Ces personnages sont basés, en partie, sur les mêmes animations. Il a donc fallu modifier le type des animations en

“humanoïde” afin qu’elles puissent être appliquées à tous les personnages sans problème. Ce type d’animation est également utile afin d’utiliser la superposition d’animations avec les “*layers*”.

2.2.2 Le système de déplacement final

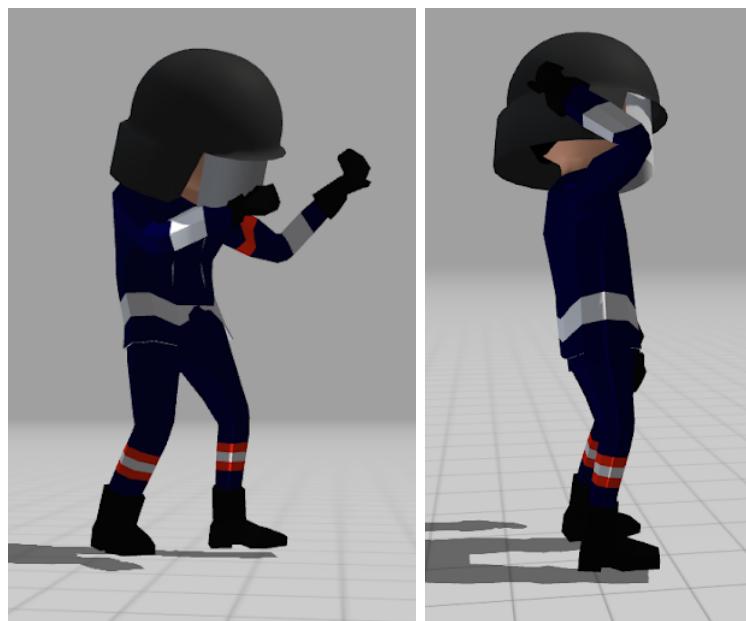
De nouvelles animations ont été ajoutés au système de déplacement final, qui concerne tous les personnages. On retrouve l’animation de marche avant, arrière, droite, gauche, course avant, course arrière, diagonale avant droite, diagonale avant gauche, idem pour l’arrière ainsi que l’animation de saut et de réception. Ce système de déplacement est le cœur du système d’animation des personnages: il permet aux joueurs de se déplacer sur les cartes.

2.2.3 Les animations spécifiques

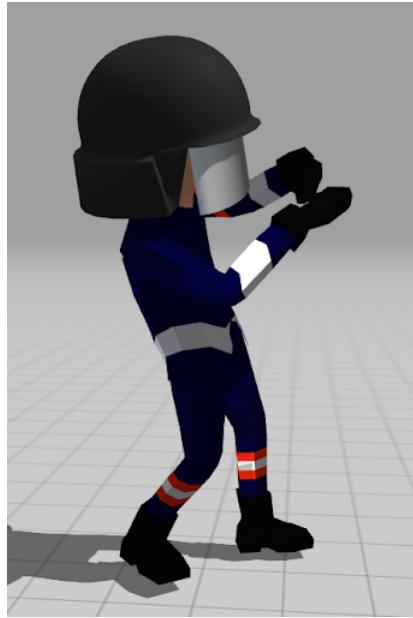
Afin de répondre aux caractéristiques des différents personnages, des animations spécifiques doivent être implémentées en plus des animations de base.

Mercenaire et Policier:

Ces deux personnages doivent être capable de porter une arme et de tirer. De nouvelles animations doivent donc être implémentées.



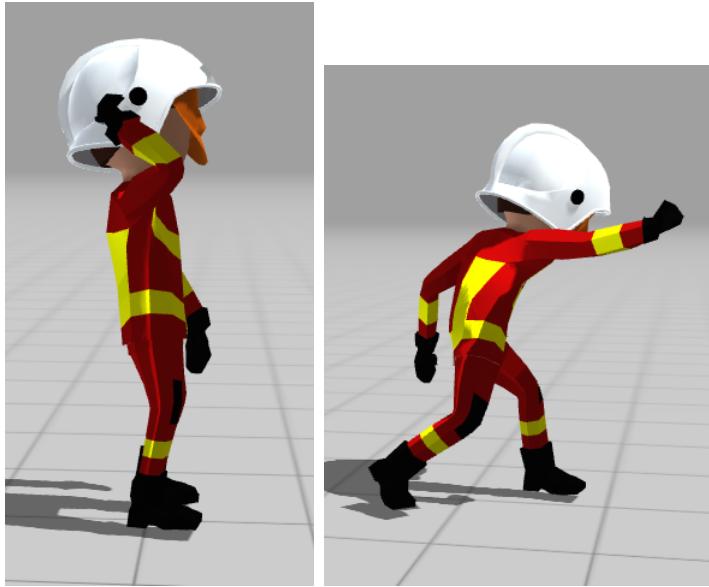
Animation de visée Animation de prise et de rangement d’arme



Animation de rechargement

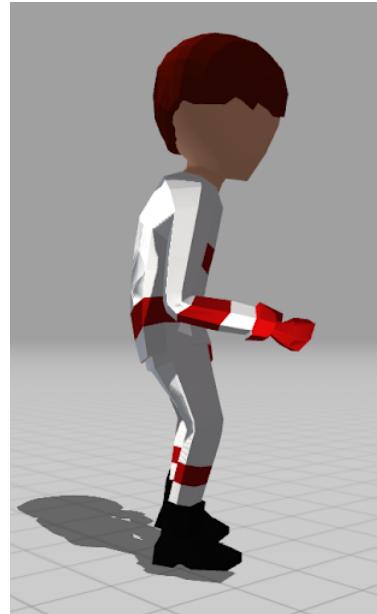
Pompier:

Ce personnage doit être capable de combattre avec une hache



Animation de prise et de rangement d'arme / Animation de combat

Médecin et Drogueur:



Animation de lancer de soin

2.2.4 Le système de visée

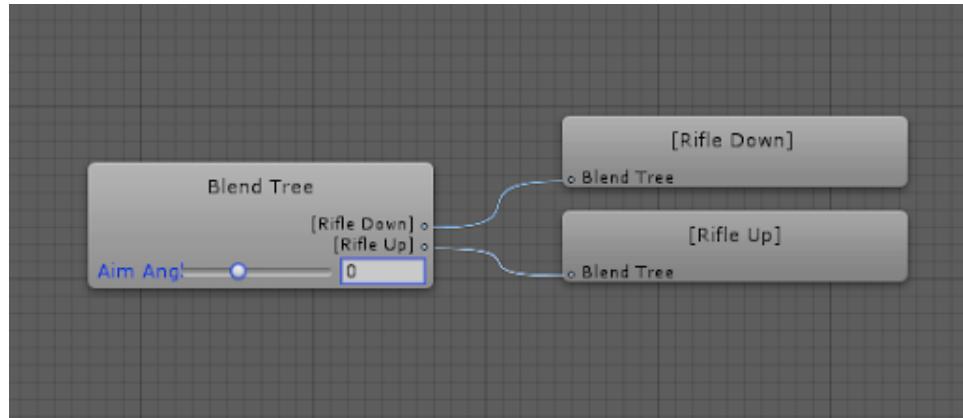
Le système de visée s'est révélé rapidement complexe : comment faire pour que l'animation de visée puisse s'orienter en haut ou en bas ?.Le problème de la droite et la gauche ne se posent pas puisque c'est le personnage entier qui subit une rotation. Unity propose au sein du composant animator (le composant gérant les animations), un système de “Blend Tree”. Le blend tree permet de mélanger plusieurs animations.

Par exemple dans notre cas, nous lui avons assigné deux animations : une animation de visée vers le haut et une animation de visée vers le bas.



Animation de visée basse et haute

Une valeur, nommé “**Aim Angle**” permet de gérer le taux d'utilisation d'une animation ou d'une autre.



Blend Tree du système de visée

Si cette valeur est au maximum, le blend tree jouera l'animation de visée en haut, et si cette valeur est au minimum, il jouera celle de visée vers le bas. Si c'est une valeur intermédiaire, il jouera un mélange des deux animations et le personnage pourra viser au milieu.

Cette valeur doit donc être modifiée en temps réel en fonction de l'endroit où vise le joueur avec la souris.

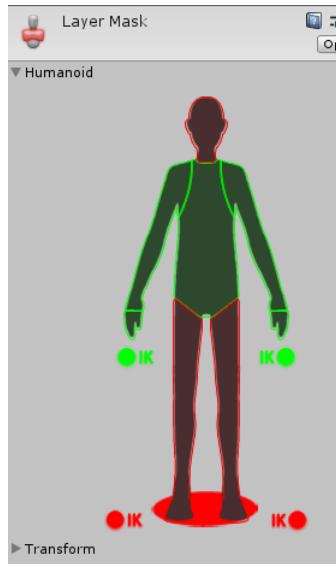
2.2.5 Le système de Layers

Plusieurs problèmes sont apparus lors de l'ajout des animations: comment faire courir et combattre le personnage en même temps.

Les “*Layers Mask*”

Pour résoudre cela, unity possède un système de superposition d'animation, appelé “*layers*”. Avec cela, on peut définir des masques et indiquer au composant animator (le composant gérant les animations) sur quelles parties du corps les animations doivent être jouées.

Pour les animations spécifiques, on les applique uniquement sur le haut du personnage.

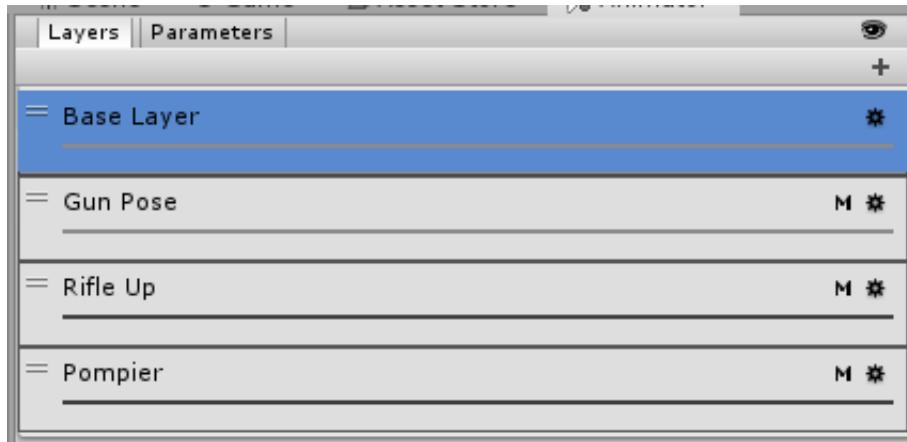


Layers Mask des animations de combat

Par exemple, ce masque est utilisé pour les animations de combat. Les zones vertes correspondent aux zones du personnage auquel on souhaite appliquer les animations. Les rouges sont donc les zones exclues.

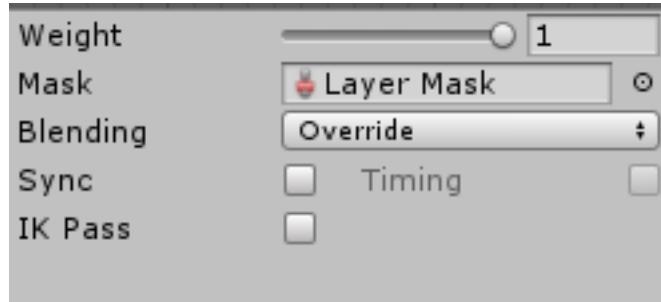
Les “*Layers*”

Différents “*layers*” ont été créés afin de superposer des animations. Ainsi, les animations du système de déplacement final sont jouées uniquement sur le bas du personnage et les animations spécifiques sur la haut. Cela permet de limiter le nombre d'animation et de ne pas avoir recours à des animations faisant courue et visée le personnage en même temps par exemple.

*Liste des “Layers”*

Base Layer est le layer du système de déplacement final.

En dessous, ce sont les “*Layers*” appliqués uniquement sur le haut du personnage. “*Gun Pose*” et “*Rifle Up*” permettent de jouer les animations spécifiques au policier et mercenaire. “*Pompier*” concerne les animations spécifiques du pompier.

*LayersMask*

Comme expliqué précédemment, un masque est appliqué à chacun “*layer*” (il permet de définir, pour rappel, les zones d’application du layer). Une valeur, nommé “*Weight*”, permet de gérer l’activation du “*layer*”. Le poids peut donc être géré dans les scripts afin d’activer ou de désactiver les “*layers*”.

2.3 Les Cartes

2.3.1 Carte n°1

La carte numéro 1 continue son développement.



Évolution de la carte entre la première et seconde soutenance

La partie urbaine

La partie urbaine de la carte à évolué: de nouveaux bâtiments ont été implémentés, la route a été agrandie et complétée et des voitures ont été ajoutées. Il y a maintenant un véritable centre-ville avec des commerces.



Centre ville



Carrefour

De nouveaux détails ont été ajoutés comme des bouches d'égout avec des particules de fumée, des feux de signalisation, des plots indiquant des travaux, des bornes d'incendie (qui pourront être utilisées par les pompiers), des bennes à ordures, des sacs à ordures etc.

Le but est de réaliser une zone urbaine simpliste qui est composée de trois éléments principaux: des routes, des bâtiments, des véhicules. La carte ne doit pas être surchargée de décors et détails afin de respecter le style graphique choisi: le Low poly. Les bâtiments et voitures joueront un rôle important dans le déroulement d'une partie. Ils pourront être détruits et donc faire évoluer la partie.

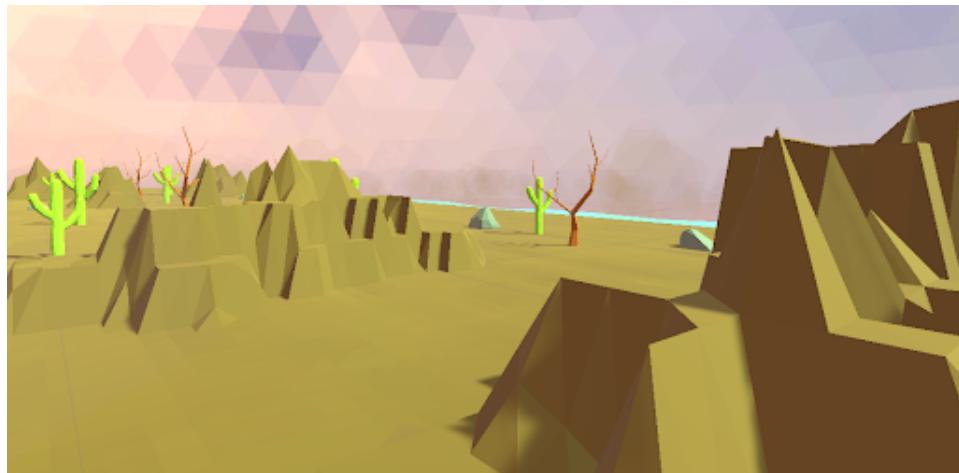
La partie environnement

Cette partie a subi de nombreux changements.



Zone montagneuse

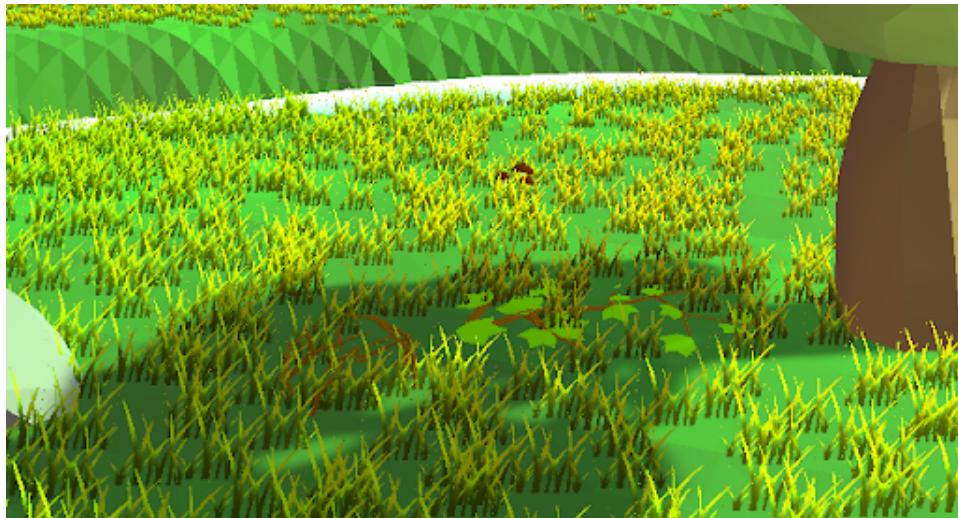
La zone montagneuse en bas de la carte est entièrement finie. Cette montagne est visible de pratiquement toute la carte et permet de créer à la fois un élément d'ambiance important ainsi qu'un repère visuel pour les joueurs. Au sommet de celle-ci, on retrouve de la neige. Une nouvelle zone a été créée: celle du désert.



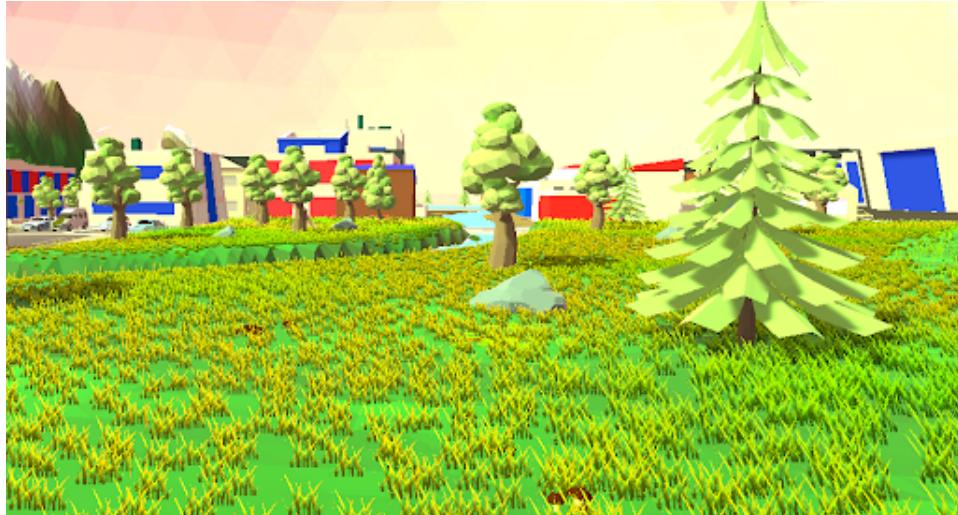
Zone désertique

On y retrouve des cactus, des arbres morts, des roches ainsi du relief. Cela permet aux joueurs de se cacher derrière et de se protéger des potentiels ennemis. Cette zone est donc importante dans le déroulement d'une partie.

Enfin, la zone du parc et forêt a été finalisée. On retrouve maintenant des rochers, un nouveau type d'herbes, des champignons, des branches, des particules de feuilles tombantes des arbres.



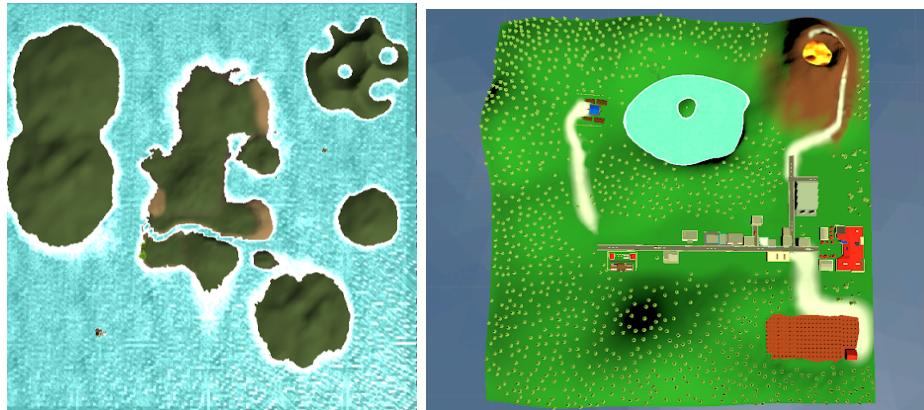
Zone parc/forêt



Zone parc/forêt

On peut remarquer qu'il y a un contraste important entre les différentes zones de l'environnement. Le but est que les joueurs puissent changer rapidement d'ambiance et qu'ils puissent découvrir sur une même carte une zone montagneuse, une zone désertique et une zone de parc et forêts.

2.3.2 Carte n°2



Évolution de la carte entre la première et seconde soutenance

La carte numéro 2, a subi de nombreux changements et adaptations au gameplay.

Ainsi la carte initiale comportant différentes îles a été abandonné au profit d'une nouvelle carte plus propre et qui correspond maintenant mieux au gameplay que nous souhaitons réaliser. En effet dans cette nouvelle carte on retrouve:

1. Une forêt en zone montagneuse qui recouvre la plupart de la surface de la carte.
2. Une ville qui comporte plusieurs nouveaux bâtiments.
3. Un volcan situé à proximité de la ville.
4. Un lac situé en zone montagneuse où on retrouve quelques bâtiments de bûcheron aux alentours.

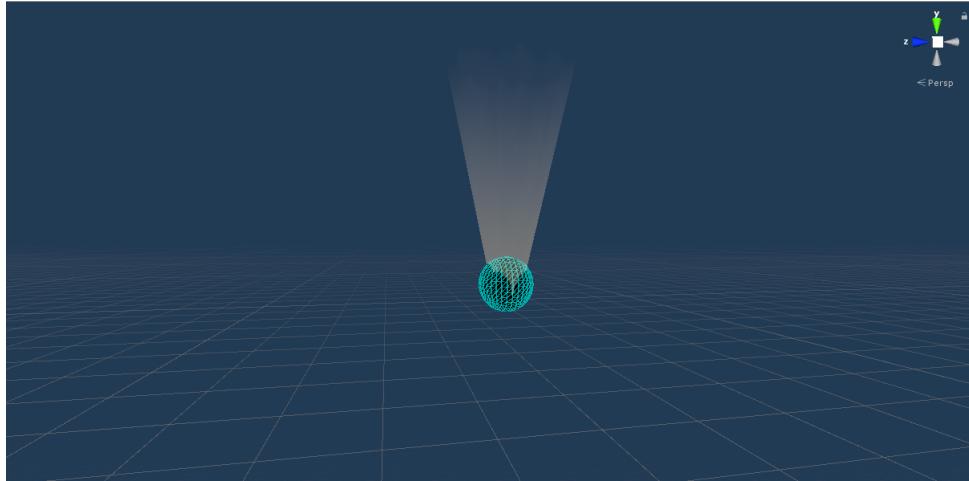


Zone urbaine



Zone montagneuse

2.4 Pouvoirs Spéciaux:



Pouvoirs Spéciaux

Les premiers consommables ont fait leur apparition dans le jeu. Ces derniers sont les pouvoirs spéciaux évoqués à la dernière soutenance. Ils permettent de modifier les caractéristiques du joueur pour les améliorer ou bien les dégrader. Les caractéristiques modifiables sont donc:

1. La taille du joueur — Le joueur peut se transformer en géant ou bien en homme minuscule, il s'agit ici d'une simple modification graphique qui aurait pour conséquence de rendre le joueur plus vulnérable (en tant que géant) ou moins vulnérable (en tant que personnage minuscule) aux balles.
2. La vitesse de déplacement — Le joueur peut ainsi devenir plus lent ou bien plus rapide.
3. La vie — Le joueur obtient plus de vie.
4. La hauteur des sauts — Le joueur peut sauter plus ou moins haut.

Toutes ses possibilités nous permettent d'ajouter une nouvelle mécanique de gameplay basée sur la recherche de pouvoirs spéciaux, permettant de faire chavirer le court de la partie.

Les pouvoirs spéciaux ont par ailleurs un temps d'effets et un temps de réapparition après qu'un joueur en ait utilisé un.

2.5 Paint Tree et nouveaux éléments graphiques:

La conception de la carte 2 a notamment permis d'apporter quelques éléments de conceptions de cartes qui nous faciliteront grandement la modification ou l'ajout de nouvelles cartes à l'avenir.

Le "paint tree", outil d'ajout d'arbre sur Unity était jusqu'alors non utilisé car il ne permettait pas de garder les boîtes de collisions des arbres ajoutés. Mais durant la conception de la carte 2, le problème a été résolu et maintenant il est bien plus rapide de peindre la carte avec un grand nombre d'arbres en très peu de temps.

Les nuages ont aussi été ajouté sur la carte 2 et sont une trouvaille tout à fait intéressante, qui renforce le réalisme des cartes et complète les skyboxes. Un script a par ailleurs été confectionné pour permettre aux nuages de bouger de façon réaliste.

2.6 Modification du site Web:

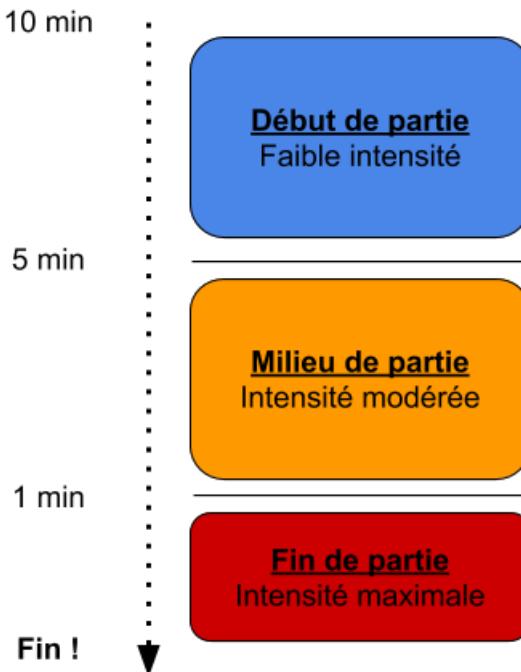
Ajout d'une option de téléchargement du jeu à sa version actuelle.

2.7 Les musiques et les sons:

Afin de rendre le jeu un peu plus stimulant et de lui donner une ambiance en adéquation avec l'avancement de la partie, nous avons décidé de créer différents sons et musiques.

2.7.1 Choix artistiques

Une partie étant d'une durée de 10 min, nous avons décidé de séparer cette durée en 3 phases:



Afin d'illustrer le niveau d'intensité de chacune de ces phases, nous avons créé 3 musiques possédant la même base rythmique... Mais ayant un niveau de détail différent.

Ainsi, lors de la première phase, la musique n'est presque **constituée que d'une nappe et d'un petit accompagnement de guitare**, alors que lors de la dernière phase, la musique devient **aussi détaillée qu'une musique de trap**.

2.7.2 Développement technique des musiques

Afin de composer ces sons, j'ai utilisé le logiciel **Ableton Live** ainsi que la suite de VST de **Native Instrument : Komplete**



L'utilisation d'Ableton en tant que séquenceur m'a permis de composer mes sons et de les arranger dans une timeline, tandis que les VST de Native Instrument m'ont permis de produire des sons qualitatifs en un minimum de temps. Les VST principalement utilisés dans la suite Native sont **Kontakt** et **Massive**.



2.7.3 Les bruitages

En complément des musiques, j'ai aussi créé des bruits pour le jeu. Ces bruits comportent des pas, des tirs, des coups de corps-à-corps. Quelques bruits (comme les tirs) sont produits en plusieurs versions (pour le cas des tirs, on a: **Tir rafale**, **tir unitaire**, **tir à 5 m**, **tir à 10 m**)



2.8 Les menus et HUD:

2.8.1 Introduction

Afin d'harmoniser le style graphique des menus et HUD (Affichage tête haute), nous avons décidé d'utiliser une palette de couleurs provenant du site "flatuiccolors.com". Cette palette a été choisie car ses couleurs possèdent peu d'éclat, elles ressortent plus agréable visuellement, ce qui est primordiale pour les menus et les HUD.



Russian Palette

RU

2.8.2 Le menu

Nous avons ajouté un menu pause, pour que lors d'une partie les joueurs puissent:

1. Changer de classe

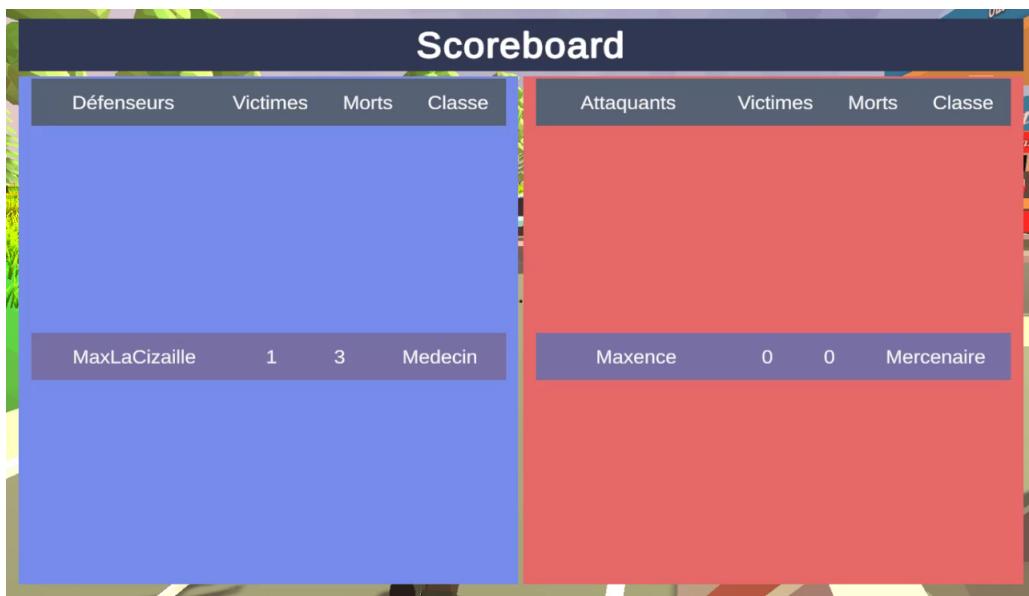
2. Changer les options du jeu
3. Quitter la partie
4. Quitter le jeu



Menu pause

2.8.3 Tableau des scores

Nous avons aussi ajouté un tableau des scores afin d'améliorer l'expérience des joueurs. De ce fait ils peuvent voir la composition de leur équipe et de l'équipe adverse.

*Tableau des scores*

2.8.4 La barre de vie

La barre de vie et son script avaient déjà été créé lors de la dernière soutenance mais elle n'était pas implémentée et non fonctionnel. Nous l'avons donc implémentée et modifier l'aspect graphique.

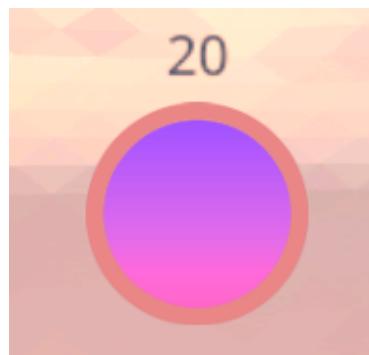
*Barre de vie*

Elle est donc représentée par une barre bleue complétée par un cadre. Un script, attaché à la barre de vie, s'occupe de récupérer la valeur théorique de la vie du joueur. Il va la comparer avec sa valeur actuelle et dans le cas où elles sont différentes, il modifie le taux de remplissage de l'image. Si la vie est à 50, son taux de remplissage est de 50, et donc seule la moitié de la barre bleue s'affiche. Afin que le changement de remplissage de l'image se fasse naturellement graphiquement, l'interpolation linéaire est utilisée avec la méthode **Mathf.Lerp** prend en argument la valeur actuelle, la valeur souhaitée et enfin la valeur d'interpolation (comprise entre 0 et 1). Pour

faire simple, la valeur égale à **Mathf.Lerp** va progressivement prendre la valeur souhaitée, proportionnellement à la valeur d’interpolation. Cela permet d’avoir un changement graphique de la barre de vie de manière progressive.

2.8.5 La barre de munitions

Exactement le même procédé est utilisé pour la barre de munitions, utilisée par exemple pour le policier et le mercenaire.



Barre de munitions

Elle est sous forme circulaire et diminue en fonction du nombre de munitions restantes. Le cercle au milieu subit une animation lorsque le joueur recharge. Un chiffre indique le nombre de munitions restantes. Un script modifie la valeur de son texte en fonction du nombre de munitions.

2.8.6 Les informations sur les armes

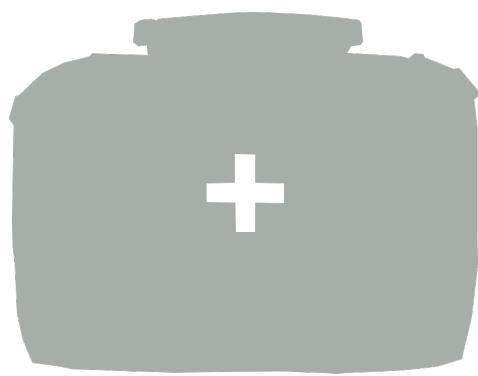
Nous avons créé, à l'aide du logiciel “**Photoshop**”, les HUD des armes.



L'arme du policier et du mercenaire



La hache du pompier



Le kit de soin du drogueur et du médecin

3 Conclusion

Pour cette seconde soutenance, nous avons donc mis en place le gameplay principal du jeu qui permet déjà d'avoir une expérience agréable pour les joueurs.