

Rapport de soutenance 2 : Projet SMAR Game 1

Sacha HIBON, Matthieu CAMART, Arthur BARBIER
et Ronan PEDRON

Promo 2025 : Projet S2



Table des matières

1	Introduction	3
2	Evolution de notre projet	4
2.1	Menu (Environnement du joueur)	5
2.2	Design 3D et Conception de l'environnement	7
2.3	Gameplay	10
2.3.1	Coeur	11
2.3.2	Spectateur	11
2.4	Animation	12
2.4.1	Les lignes temporelles (Timeline)	12
2.4.2	L'armature et les caractères	13
2.4.3	Énumération	17
2.4.4	Les transitions	18
2.5	Site Web	19
2.6	Réseau	20
2.6.1	Hall/Salle	21
2.6.2	Photon Animator View	21
2.6.3	Apparition (Spawner)	21
2.6.4	Type des joueurs	21
2.6.5	Manager les bots	21
2.6.6	Vie	22
2.7	Intelligence artificielle	22
2.7.1	Maintenance	22
2.7.2	Mémoire/Sauvegarde	24
2.7.3	Bot Rectiligne	24
2.7.4	Fuyard	25
2.7.5	Path Finding	25
2.7.6	Suiveur	26
2.7.7	Ajustement	27
2.7.8	Synthèse	27
2.8	Base de donnée	27
2.8.1	Implémentation dans le jeu	28
2.8.2	Apprentissage de notre IA	28

3	Musique et Bruitage	28
3.1	Musique	28
3.2	Bruitages	29
3.3	Réglage du volume	29
4	Avancement du projet	29
4.1	Réussites	30
4.2	Retards	30
5	Prévision	30
5.1	Environnement du joueur	31
5.2	Réseau	31
5.3	Animation	32
5.4	Edition 3D	32
5.5	Son	32
5.6	Gameplay	32
5.7	IA	33
6	Conclusion	33

1 Introduction

Ce compte-rendu a pour but de présenter l'avancement de notre groupe, SMAR, sur notre projet SMAR Game 1. Il détaillera notamment l'avancement de notre groupe depuis la première soutenance en mars, et nos objectifs pour la prochaine et dernière soutenance.

SMAR Game 1 est un Guess Who. Ce type de jeu comporte un joueur représentant le "chasseur" et dont le but est de traquer tous les autres participants, les "chassés". Ceux-ci se déplaceront librement dans un environnement quelconque et se cacheront derrière des PNJs (personnage non-joueur ou bot) qui auront eux aussi l'aspect humain. Lorsque le chasseur soupçonnera un humanoïde d'être un joueur, il pourra décider de l'attaquer à l'aide d'une arme à feu ou d'une arme blanche. Pour gagner, le chasseur devra attraper (tuer) tous les autres joueurs. S'il s'avère être un mauvais détective, les chassés gagneront après un certain temps. D'autre part, lorsque le chasseur tire sur une mauvaise cible, c'est à dire un PNJ, il perdra des points de vie. S'il finit par en mourir, tous les chassés gagneront.

Nous avons décidé que tout ce scénario se déroulera dans le campus de Villejuif, nous aurons donc l'enceinte d'EPITA ainsi que celle de Sup Biotech.

En ouvrant ce projet, même si ce n'est pas la première fois que nous le disons, tout ce que vous y verrez a été modélisé, animé et codé par notre équipe.

2 Evolution de notre projet

Après cette deuxième partie de projet, nous n'avons pas apporté de modification, mais il nous semblait important de redonner la répartition des tâches.

TABLE 1 – Répartition des tâches

<div>Membre</div> <div>Tâche</div>	Sacha	Matthieu	Ronan	Arthur
Graphisme		+	+	
Réseau	+			+
Site web			+	
Intelligence Artificielle	+	+		
Bruitage & Musique	+	+	+	
Gameplay	+			+
Animation		+	+	
Environnement du joueur		+		+
Manuel d'utilisation			+	+
Mémoire/Stockage	+		+	

2.1 Menu (Environnement du joueur)

Depuis la première soutenance, notre priorité pour le menu était de changer son design. Pour cela nous avons directement changé l'image en fond, puis nous avons modifié les boutons ainsi que leur police. De même, plusieurs fonctionnalités ont été ajoutées, notamment le fait que l'utilisateur doit absolument rentrer un nom pour pouvoir jouer sinon les boutons "Find Room" et "Create Room" sont désactivés. Le bouton "save" ayant été supprimé, dès qu'il entre une lettre, son nom est automatiquement sauvegardé. La possibilité de changer ses touches est désormais disponible dans le menu "Options" (cf. Figure 1). Ces changements seront sauvegardés grâce à "PlayerPrefs", un outil que nous utilisons déjà pour enregistrer le changement du nom du joueur par exemple. Enfin un menu nommé "Credits" a été ajouté dans le menu principal. Dans ce dernier, vous y retrouverez le travail des membres de notre groupe sur le jeu.

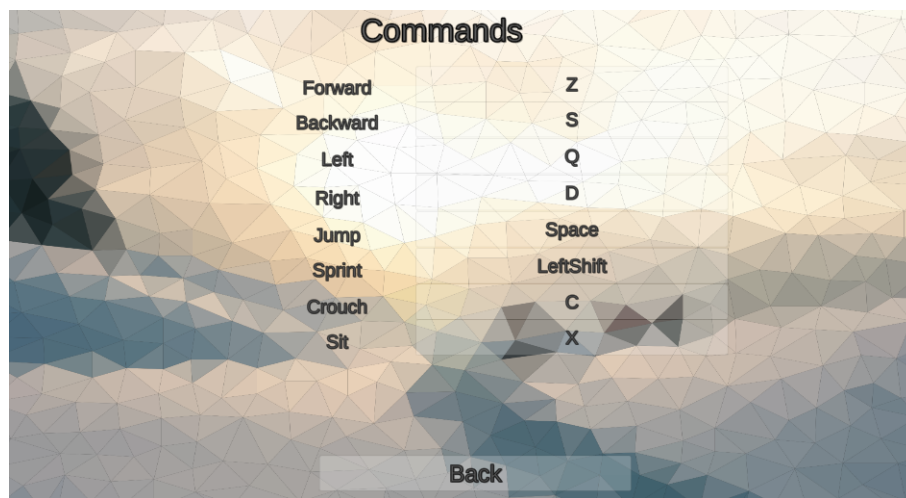


FIGURE 1 – Changement des touches

Le joueur a une "InterfaceInGame" (Interface en jeu) qui affiche toutes ses informations, notamment sa vie et le viseur. Pour la vie, nous avons implémenté un système de coeur qui se fissure au fur et à mesure que le joueur perd ses points de vie. (cf. Figure 2)



FIGURE 2 – Interface en jeu

Par ailleurs, le menu "échap" a été entièrement implémenté avec la possibilité de changer directement ses options de jeu tel que le volume, les graphismes et les changements de touches. Enfin, le joueur peut désormais voir les informations de tous les joueurs présents dans la partie, en fonction de leur rôle, en appuyant simplement sur la touche tabulation (TAB). (cd. Figure 3)



FIGURE 3 – Menu Tabulation

2.2 Design 3D et Conception de l'environnement

Lors de notre dernière soutenance nous avons dit que la prise en main de Blender n'était pas mince affaire, ceci n'a pas changé. Nous découvrons de nouvelles fonctionnalités tous les jours ce qui montre la complexité de ce logiciel. Il nous est déjà arrivé de faire quelque chose d'assez pénible pendant assez longtemps et de se rendre compte le lendemain qu'il existe une fonctionnalité déjà prédéfinie par Blender.

La création de tous les modèles qui constituent la carte s'agit d'un travail extrêmement long et minutieux. Nous avons fait d'énormes progrès sur Blender depuis nos débuts, non seulement dans la maîtrise de cet outil mais également dans la création de modèles. Vous pouvez voir une photo de la carte à la première soutenance et à l'heure actuelle.

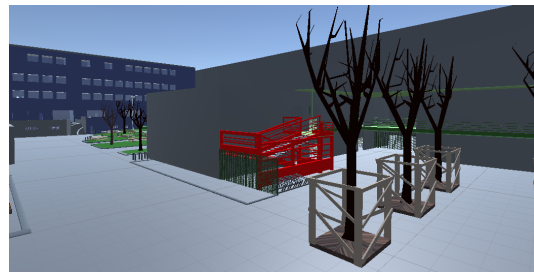
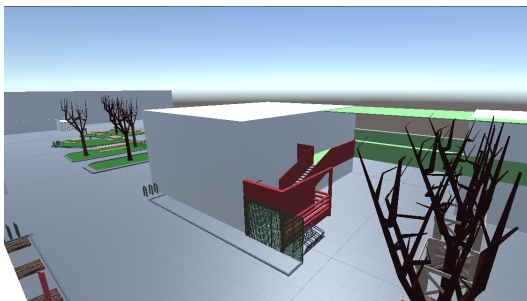




FIGURE 4 – Ancienne carte

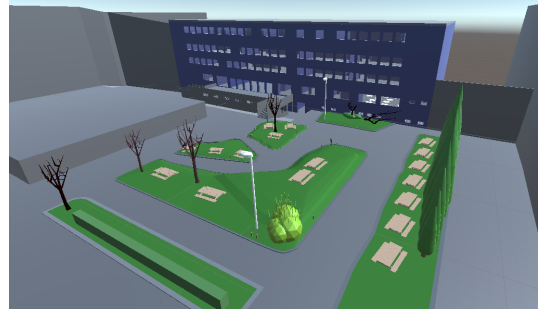
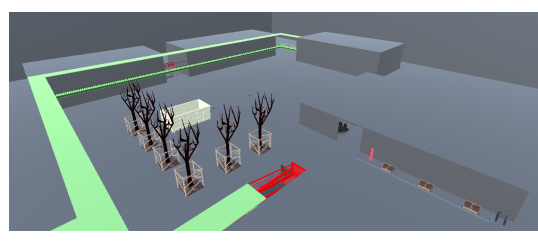
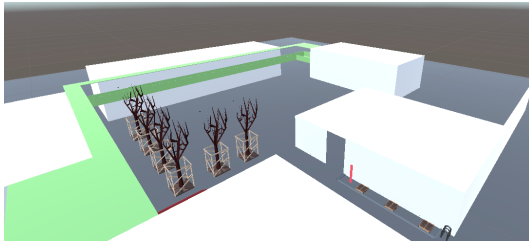


FIGURE 5 – Carte actuel



Nous sommes arrivés sur un problème auquel nous ne pensions pas avoir à faire face, celle de l'optimisation. En effet, créer ses propres modèles n'est pas forcément la chose la plus optimisée que l'on puisse avoir en terme de point sur la carte. Plus il y a de point, plus l'objet est détaillé mais plus l'objet est lourd. Nous devons donc faire attention au nombre de points que nous avons sur la carte. Nous apprenons également que faire un arrondi avec 10 rotations de face ou 42 ne change pas beaucoup quant à l'apparence de notre modèle. Il faut donc privilégier les modèles qui sont esthétiques mais qui ont également un nombre de points raisonnable. Par exemple, ces deux modèles sont quasiment identiques mais une des chaises possède moins de la moitié des points de l'autre.



FIGURE 6 – Chaise non optimisé

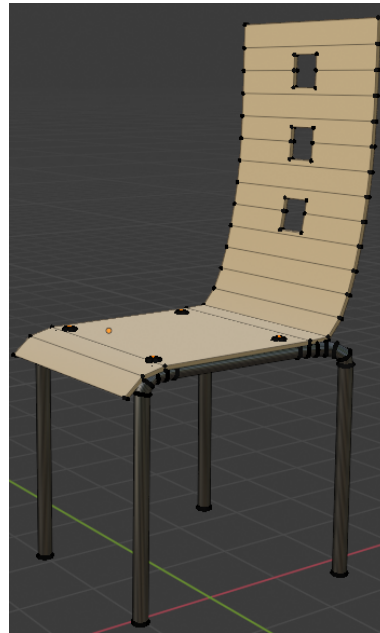


FIGURE 7 – Chaise optimisé

Depuis la dernière soutenance, nous avons fait un travail titanesque sur la carte. Nous avons tout agrandi pour que les dimensions soient les plus correctes même s'il reste encore quelques problèmes vu qu'il est très difficile de faire une carte à vu. L'utilisation du mètre sur Blender à été utilisé principalement pour la hauteur de nos bâtiments. Son utilisation régulière sur tous les objets de notre carte est une perte de temps car elle reste trop approximative. Avec ces agrandissements, nous avons ajouté la cafétéria ainsi que 6 classes dans la cour de SupBiotech. Etant donnée que nous ne connaissons pas la disposition de leurs salles, nous espérons revenir à l'école afin de faire les modifications des autres bâtiments.

La plus grosse partie est celle de l'implémentation du bâtiment d'EPITA qui est sur 5 niveaux, nous avons récupéré au préalable des plans de l'école ce qui nous a énormément aidé. Nous n'avons pas encore complété toutes les salles de classe mais cela n'est qu'une question de temps. Cependant nous ne connaissons pas l'organisation du "sous-sol" (-1). Nous attendons donc avant de faire quelconques arrangement. Nous espérons avoir accès à ce lieu, nous formulerons cette requête dès notre retour à l'école.

Au niveau des modèles 3D nous sommes particulièrement fier des baby-

foot, ou encore des chaises et tables que nous avons implémenté dans l'espace de la cafétéria de la cours de SupBiotech.



FIGURE 8 – Baby foot



FIGURE 9 – Vu d'ensemble de la cafétéria

Malheureusement cette complexité dans nos modèles nous oblige à générer énormément de points. Cependant, ce procédé ne nous permet pas de placer ce type de chaise au quatre coin de notre carte. Mais nous trouvons important de faire une cafétéria réaliste en vue de son emplacement et par conséquent du nombre de passage dans celle-ci.

2.3 Gameplay

La phase de jeu a été, dès la première soutenance, presque terminée. En effet, les personnages pouvaient se déplacer librement et les différents rôles avec leurs actions respectifs avaient été implémentés. De plus, une grande

partie du gameplay réside dans la programmation des bots. Ainsi, rien de très conséquent n'a été développé, seul du peaufinage vous sera présenté dans cette partie.

2.3.1 Coeur

Premièrement, notre carte comporte des failles comme la plupart des importations Blender. C'est à dire que certains murs et certaines grilles peuvent laisser passer des personnages. Même en rajoutant des obstacles invisibles ou non, les collisions préprogrammées ne permettaient pas un parcours complètement réaliste du campus. Ainsi, nous avons transformé ce bug en fonctionnalité. Pour cela, nous avons créé un cœur artificiel se situant au centre de l'abdomen de nos personnages. Lorsque celui-ci détecte un objet dans son périmètre, il fera perdre des points de vie à son possesseur. Ceci permet deux choses, les joueurs continueront de bénéficier de raccourcis tout en ne pouvant pas en abuser.

2.3.2 Spectateur

Lorsqu'un joueur meurt, celui-ci ne souhaite pas avoir un écran vide lui rappelant son décès tragique. Ainsi, nous avons décidé qu'il pourra suivre la vision des joueurs vivants en choisissant le porteur de sa caméra. Le « Spectateur » aura, tout comme le « Chassé » une vue à la troisième personne et pourra effectuer une rotation à 360°.

2.4 Animation

2.4.1 Les lignes temporelles (Timeline)

Après la première soutenance, nous nous sommes rendu compte que nos animations ne nous plaisaient pas. En effet, nous les avions conçues assez rapidement et nous n'en n'étions pas satisfait. Cela était dû au fait que nous les modélisions avec un squelette peu complexe et peu pratique le tout sur une seule lignée de temps, une seule timeline (comme la photo ci-dessous).

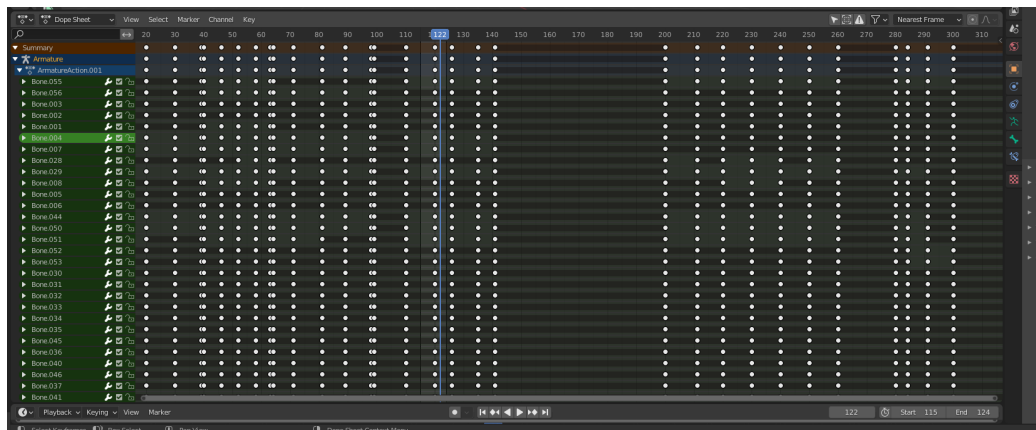


FIGURE 10 – Vieille timeline

Le problème de cette méthode étant qu'à chaque importation du personnage sur Unity, il fallait séparer à la main chaque animation en indiquant l'indice de temps du début et de fin de cette dernière. Nous avons fouillé Blender et nous avons trouvé notre sauveur. La partie spécialisée de Blender nous a permis de créer des traces d'animations, aussi appelées Track (comme ci-dessous). Vous y verrez différentes lignes dans lesquels sont ordonnées ergonomiquement les animations.

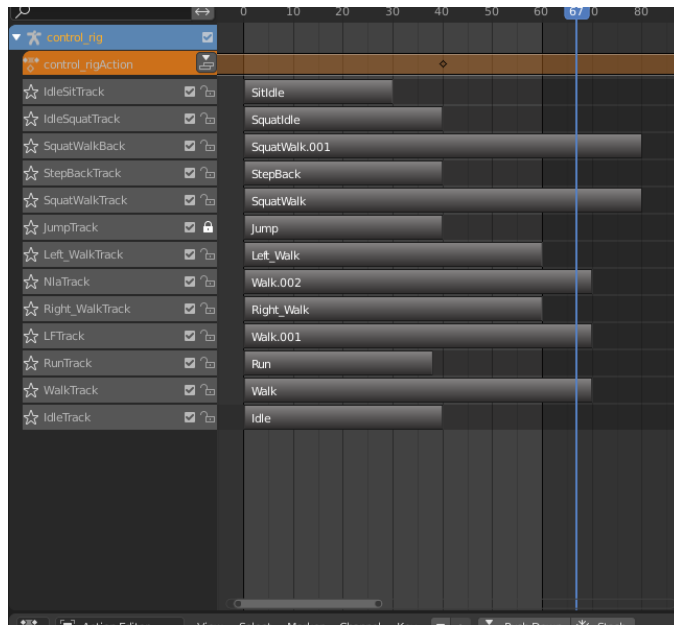


FIGURE 11 – Nouvel arrangement des timelines

2.4.2 L'armature et les caractères

Nous avons pu répartir nos animations sur plusieurs timeline qui, dès leur arrivée sur Unity, se sépare automatiquement avec le même nom Blender. De plus, comme dit précédemment, notre ancienne armature diminuait notre productivité. En effet, nous devions à chaque fois bouger les os individuellement afin de faire une animation réaliste. Nous avons une armature qui fut plus complexe à créer, certes, mais qui simplifie chaque animation. Nous avons donc utilisé des contrôleurs d'os qui agissent structurellement. Par exemples, un os permet de gérer la montée du genou en fonction de la position du pied. Nous sommes donc passé de l'armature gauche à celle de droite (voir ci-dessous).

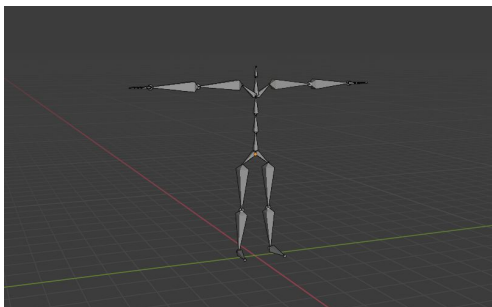


FIGURE 12 – Ancienne armature



FIGURE 13 – Nouvelle armature

Le rendu sur nos personnages est devenu différent et beaucoup plus agréable à voir. Dans le même principe, nous sommes passé de l'image de gauche à l'image de droite.



FIGURE 14 – Ancienne carrure



FIGURE 15 – Nouvelle carrure

Enfin, nous avons effectué les animations avec les armes qui sont nettement plus complexes. En effet, il a fallu récupérer l'armature générale ainsi que les animations élaborées pour les chassés (Hunted) pour éviter de trop grosses différences dans les mouvements entre les humanoïdes. Il était également nécessaire de modifier les mains pour permettre une tenue des armes. Pour chaque type d'arme nous avons un type d'animation qui leur est propre.

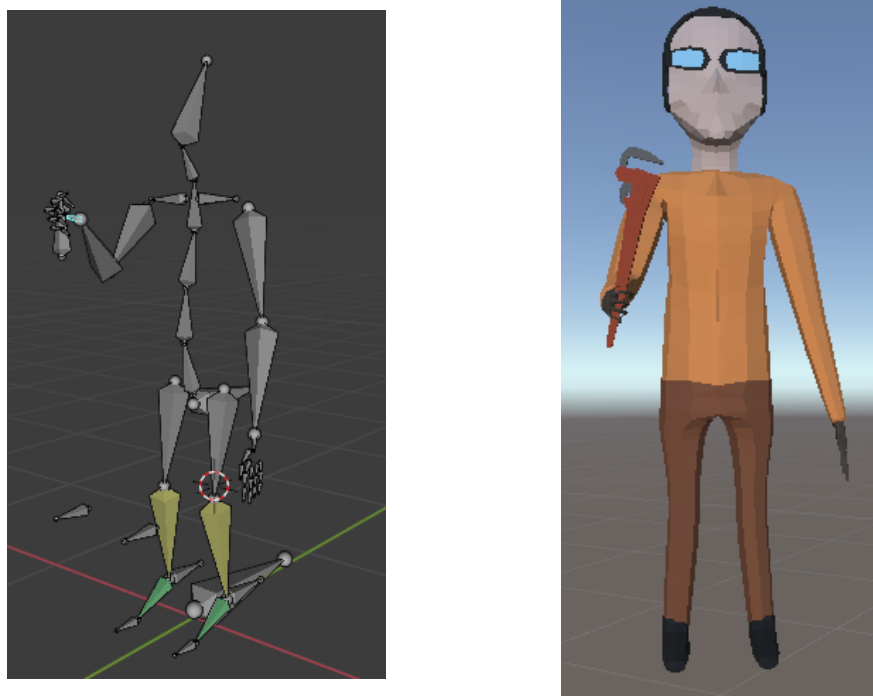


FIGURE 16 – Amarture du chassé et chassé avec la clef à molette

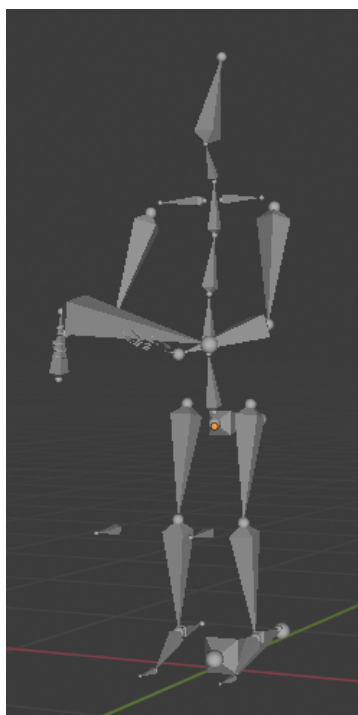


FIGURE 17 – Amature du chassé et chassé avec la AK-47

Nous avons aussi implémenté une fonctionnalité qui permet de combiner différentes animations tel que la visée et la course.

2.4.3 Énumération

Par ailleurs, les chassés et les chasseurs ont des animations communes tels que marcher, courir, se déplacer dans toutes les directions (avec des animations uniques). Ils peuvent aussi sauter et, se mouvoir dans toutes les directions lorsqu'ils s'accroupissent. Additionnées à celles-ci, les chasseurs ont la possibilité de viser, tirer ou bien de frapper. Ces dernières étant disponibles qu'en possession de l'arme adéquate.

Pour les Chassés (Hunted), une animation unique a été créée, celle de s'asseoir. Dans cette position, le joueur ne peut ni sauter ni se mouvoir.

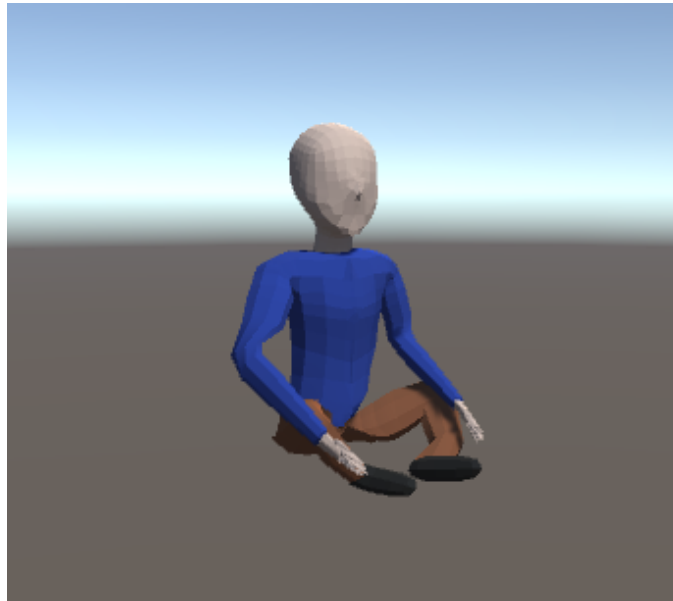


FIGURE 18 – Chassé assis

2.4.4 Les transitions

Pour que toutes ces animations ne paraissent pas saccadées, nous avons dû utiliser l'Animator d'Unity ainsi qu'une cascade de booléen. Le schéma parle de lui-même (cf. ci-dessous), il est complexe mais reste simple. Chaque flèche est une transition possible d'une animation vers une autre.

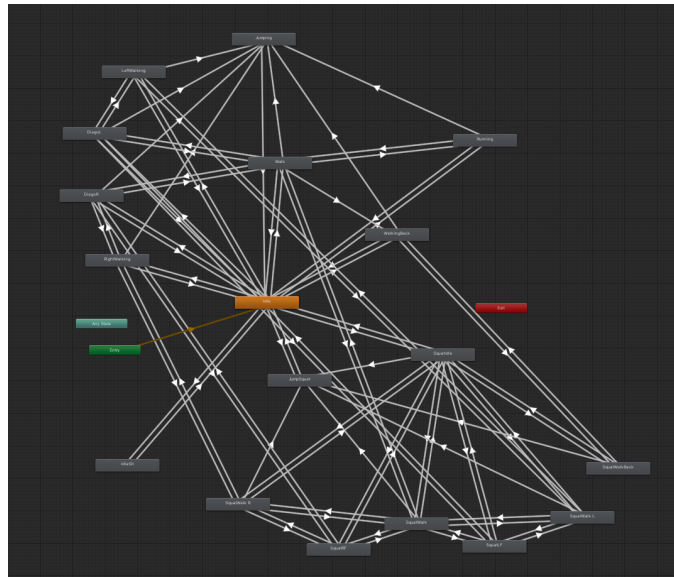


FIGURE 19 – Transition par l'Animator

2.5 Site Web

Pour la première soutenance nous avons conçu un site sur la plateforme Wix. Malgré la simplicité d'utilisation et le gain de temps qu'il nous apporterait au cours du projet, nous avons pour objectif de faire notre propre site. Mais nous ne voulions pas passer par des logiciels de "NoCode" qui nous auraient évités l'apprentissage des langages de confection de site tel que HTML, CSS ou encore Java Script. A l'heure actuel nous avons un site parfaitement opérationnel qui a été entièrement créé par nos soins. De plus, cela respecte l'objectif que nous nous étions fixé : tout créer de nos mains. De plus, cela tenait à coeur à l'un des membres de notre groupe de le développer car il était curieux d'apprendre ces langages. Nous nous sommes donc lancés dans l'apprentissage de l'HTML ainsi que le CSS. Nous ne sommes pas arrêtés ici. Afin d'implémenter un carrousel sur notre site nous avons du légèrement apprendre une petite dose de Java Script. Le carrousel est composé d'une photo d'un membre accompagnée d'une petite citation.

Chaque membre du groupe a le droit à sa page sur laquelle les visiteurs peuvent trouver d'autres informations sur les membres. Les pages sont accessibles grâce à un menu déroulant que vous pouvez voir ci-dessous.

Accueil	Nos membres	Notre Jeu
	Ronan Pédrón	
	Arthur Barbier	
	Matthieu Camart	
	Sacha Hibon	

FIGURE 20 – Menu déroulant pour nos membres

Nous avons également une page de téléchargement, où nous mettons un exécutable de notre projet afin que toute personne voulant y jouer en ait la capacité. Ces téléchargements sont accompagnés de patch note (note de mise à jour) qui indique les modifications apportées à notre jeu.

Accueil	Nos membres	Notre Jeu
		Download
		Notre IA
		Nos modèles

FIGURE 21 – Menu déroulant concernant notre jeu

La structure générale de notre site est finie, cependant nous nous trouvons limité dans l’expression de toutes les possibilités à parcourir avec ce projet, c’est pour cela que nous avons décidé de créer deux nouvelles pages.

Une première page qui s’intitule « Notre IA », où nous comptons y mettre des articles sur nos différents PNJ (Personnage Non Joueur) ainsi que leur fonctionnalité. Nous avons également rajouté une page nommée « Nos modèles » où nous allons mettre des images de tous nos modèles fait sur Blender ainsi qu’un petit texte les accompagnant décrivant toutes nos péripéties.

Notre site internet était initialement hébergé sur GitHub mais, afin de cultiver notre curiosité, Ronan a loué un serveur pour que nous soyons autonomes et pour que nous puissions apprendre davantage de choses grâce à ce projet. Vous pouvez donc retrouver notre site sur « smar.cf ».

2.6 Réseau

Après que les joueurs se soient connectés au sein d’une même partie, il reste, en définitive, peu d’ajustements synchronisés nécessaires. En effet, nous nous sommes acharnés à minimiser la planification centrale au profit de l’initiative locale.

Cependant, certaines actions doivent être obligatoirement coordonnées entre les ordinateurs des différents participants, tel que l’apparition des personnages, leurs noms et d’autres attributs relatifs au jeu en lui-même. Pour ce genre de partage d’informations, nous avons souvent utilisé d’un système de Hash. Ce procédé consiste à envoyer un message possédant une étiquette (une liste de caractères) pour le distinguer des autres messages, d’une ‘target’ pour savoir quel joueur est ciblé par ce communiqué et enfin, d’une valeur

d'un type quelconque (int, string...).

2.6.1 Hall/Salle

Le système de création de salles (room) et la possibilité d'en rejoindre une n'ont pas changé. Cependant lorsque deux joueurs ayant le même nom rejoignent une même salle, le dernier à la rejoindre aura un chiffre ajouté à son nom afin de les différencier.

2.6.2 Photon Animator View

Pour pouvoir synchroniser les animations, nous avons décidé d'utiliser une fonction déjà prédéfinie dans PUN2 (Photon Unity Network 2), le Photon Animator View. Il nous suffit de l'implémenter sur les animations en question afin que tous les joueurs puissent les voir en temps réelle.

2.6.3 Apparition (Spawner)

Ensuite, il fut impératif de dissocier l'apparition de chaque humanoïde pour éviter toute collision qui aurait pour conséquence de propulser un joueur à 150 mètres de la carte. Ainsi, chaque ordinateur ne pouvait choisir l'emplacement de son avatar sans le communiquer aux autres. L'alternative la plus simple était que seul le créateur de la partie (le MasterClient) décide de la position de tous les joueurs.

2.6.4 Type des joueurs

Il en va de même pour le type des participants. Comme nous souhaitons établir le nombre exact de « Chasseur » et de « Chassé », le MasterClient devait organiser les attributions au lieu qu'elles soient faites aléatoirement au risque de bouleverser l'équilibre du jeu.

2.6.5 Manager les bots

Lors de la première soutenance, nous avons décidé que seul le MasterClient contrôlait les PNJs. Seulement, leur grand nombre provoquait un ralentissement de l'ordinateur en question. La meilleure solution était que tous les participants aient un nombre équivalent de bots à maîtriser. Le résultat

était très satisfaisant, désormais, à 4 personnes, 56 bots est envisageable sans même baisser la qualité de jeu des joueurs.

2.6.6 Vie

Comme vu précédemment, chaque joueur peut prendre connaissance, via le menu tabulation, du nombre de points de vie des autres joueurs. Ce qui implique que chaque ordinateur doit être capable d’actualiser la santé de tous les participants mais aussi des bots. En effet, il peut s’avérer, par exemple, qu’un chasseur fasse perdre des points de vie à un PNJ, celui qui contrôle ce bot devra en être informé pour pouvoir le faire disparaître lors de sa mort.

2.7 Intelligence artificielle

Durant la première soutenance, les PNJs n’avaient que des déplacements extrêmement basiques et limités. Comme évoqué plus tôt, c’est l’une des parties les plus importantes de ce projet. Celle qui permet aux « chassés » de rester discret, autrement dit de subsister, mais aussi celle qui propose une infinité d’implémentation pour les programmeurs. Ainsi, durant ce rapport, beaucoup d’améliorations seront précisées.

2.7.1 Maintenance

Tout d’abord, une multitude de points ont été placés sur le campus à altitudes différentes et avec potentiellement des obstacles entre eux. Le travail a été de détecter si chaque point était voisin d’un autre, c’est à dire si un PNJ pouvait passer d’une coordonnée à un autre sans rencontrer d’entrave. L’objectif principale de cette manœuvre est de déterminer que les escaliers comme les dunes sont des chemins praticables en les distinguant des murs, des barrières et toute sorte d’obstacles censées être infranchissable. Comme cet algorithme devait simuler le déplacement des vrais joueurs donc user de la même vitesse que ceux-ci, ce programme devait être lancé hors d’une partie.

Ainsi, nous est venu l’idée d’une « maintenance », le programme qui déterminerait les déplacements possibles des bots avant même que l’on diffuse notre jeu. Cela implique que les résultats soient stockés dans des fichiers et qu’un code puisse les lire pour en extraire les informations pertinentes, c’est le début de la mémoire de notre intelligence artificiel.

La maintenance, en pratique, est constitué de trois étapes. On choisit les zones de notre carte que l'on veut sonder. Chaque point présent dans la zone va envoyer une capsule qui aura, la masse, les dimensions et la vitesse d'un bot. Une seule capsule à la fois sera expédiée pour que de potentielles collisions ne faussent les recherches. Enfin, lorsque chaque point a envoyé des capsules vers tous ces voisins, les résultats sont compactés et rangés dans un dossier regroupant la sauvegarde de toutes les zones du terrain. Le plan de cette maintenance peut paraître long mais sonder tout le campus (près de 300 points) dure seulement 6 minutes et atteint 97% de réussite dès le premier coup.

Vous pouvez observer une vidéo affichant son utilisation sur notre site.

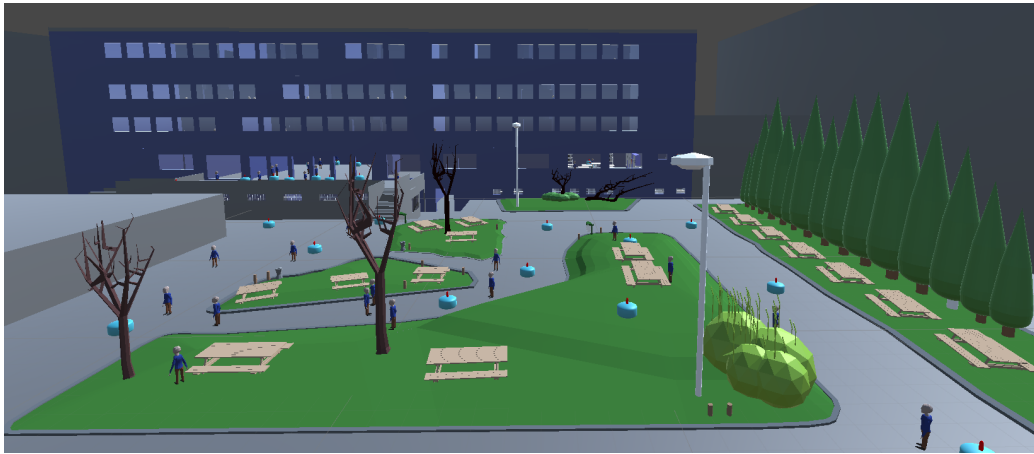


FIGURE 22 – Maintenance

2.7.2 Mémoire/Sauvegarde

```
CrossPoint (0) : ,1,3,7,10,113,114
CrossPoint (1) : ,0,2,4,5,7,8,42,110
CrossPoint (2) : ,1,3,4,5,8,40,42,114
CrossPoint (3) : ,0,2,4,7,42,113,114,115
CrossPoint (4) : ,1,2,3,5,42
CrossPoint (5) : ,1,2,4,40,42
CrossPoint (6) : ,9,38,40,41,110
CrossPoint (7) : ,0,3,10,107,110
CrossPoint (8) : ,0,1,2,4,40,41,110
CrossPoint (9) : ,6,38,40,41,109
CrossPoint (10) : ,0,1,7,11,13,101,107,109
CrossPoint (12) : ,14,17,104
CrossPoint (13) : ,10,104
CrossPoint (14) : ,12,15
CrossPoint (15) : ,14,16,111
CrossPoint (38) : ,6,9,37,40,41,109
```

FIGURE 23 – Fichier de sauvegarde

La figure 23 représente la sauvegarde précédemment évoqué, elle se décompose en deux parties. Sur chaque ligne est inscrit le nom du point. Le nom n'est pas complètement libre, il doit contenir à son bout, un nombre entouré de parenthèses. Deux points ne pouvant avoir le même numéro, ils sont tous directement référencés par leur nombre. Il est donc très facile de les ranger dans une liste lors du parsing (lecture) de tous les fichiers. Ensuite, est enregistré tous les voisins du point en question, ceux-ci sont désignés par leur nombre et séparés par une virgule.

2.7.3 Bot Rectiligne

Maintenant que nous possédons toutes les informations relatives aux déplacements des joueurs, autrement dit des PNJs, seul l'attitude des bots restait à définir.

Premièrement, le « bot rectiligne » se déplaçant d'un point à un autre. Son implémentation est ternaire, il translate vers un point, lorsqu'il est assez proche de celui-ci, il choisit aléatoirement une nouvelle destination et pivote en marchant, ainsi de suite. Ce type de mouvement permet deux choses très importantes, toutes les zones seront visitées par les bots rectilignes et, les

lieux de passages, autrement dit les espaces où les chassés auront le plus de chance de croiser les chasseurs, seront davantage parcourus par ces mêmes bots.

2.7.4 Fuyard

Le « Fuyard » a directement été inspiré des joueurs pouvant perdre leur sang-froid à l'approche d'un chasseur. Ainsi, ce bot simule la vue d'un chassé afin de détecter un chasseur rentrant dans son « champs de vision ». Dans ces circonstances, le fuyard se déplacera vers un autre PNJs le plus loin possible de son assaillant en prenant le chemin le plus rapide. Il utilisera donc l'ensemble des "points" comme un graphique relié entre eux en fonction des obstacles qui les séparent. Ce choix d'implémentation est très peu coûteux puisqu'il réutilise la maintenance post-game mais, présente un inconvénient. Nous ne plaçons pas des points à chaque coins, à chaque intersections, à chaque coordonnées de la carte. Ainsi, le chemin obtenu par ce graphique ne sera pas le plus rapide. Ce qui nous amène au « path finding ».

2.7.5 Path Finding

Le path finding est un algorithme censé trouver le chemin le plus court entre deux coordonnées quelconque d'une zone. Premièrement, nous avons fait le choix de le coder sur deux dimensions étant donné la nature des déplacements des personnages. Ensuite, il devait être efficace, c'est à dire très peu coûteux en temps puisqu'un fuyard doit presque instantanément récupérer le résultat de la recherche. De plus, il est impossible de faire une sorte de « maintenance » pour sonder la zone avant la partie puisque cela reviendrait à trouver les meilleurs chemins entre ABSOLUMENT toutes les coordonnées de la carte.

Au final, l'algorithme prend une configuration similaire au parcours largeur d'un arbre général ou binaire. Tout d'abord, on part du point d'origine, le programme vérifie si sa position est valide, c'est à dire qu'elle n'est pas situé sur un obstacle. Si ce n'est pas le cas, il s'arrête là, dans le cas contraire, il enfile quatre coordonnées (node) à une distance d de lui et recommence ce travail avec le premier élément de la file. Deux résultats sont possibles, si la file se retrouve vide sans que la destination n'ait été atteinte, il existe manifestement aucun chemin entre les deux points rentrés en argument.

Dans le cas d'une recherche positive, le chemin sera construit en reliant

les coordonnées qui se sont chaînées au fur et à mesure. En effet, chaque node retient celui qui l'a créé (enfilé). Cette implémentation fonctionne seulement parce que chaque node se trouve à équidistance de ces voisins. Ensuite, le chemin est raffiné en prenant un minimum de point à suivre. Cet algorithme est particulièrement intéressant dans un dédale désorientant. Vous pouvez d'ailleurs observer son utilisation sur notre site web grâce à un labyrinthe (de 1.2 KILOMETRES) que le responsable d'édition 3D a pris soin de modéliser.

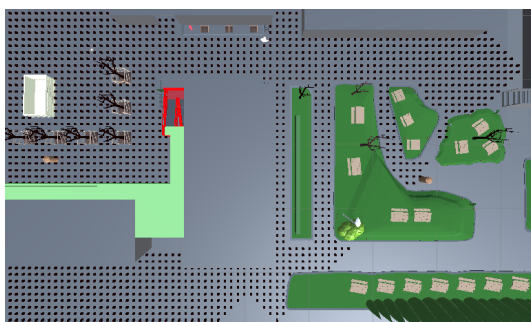


FIGURE 24 – Dispersion des nodes



FIGURE 25 – Création du chemin

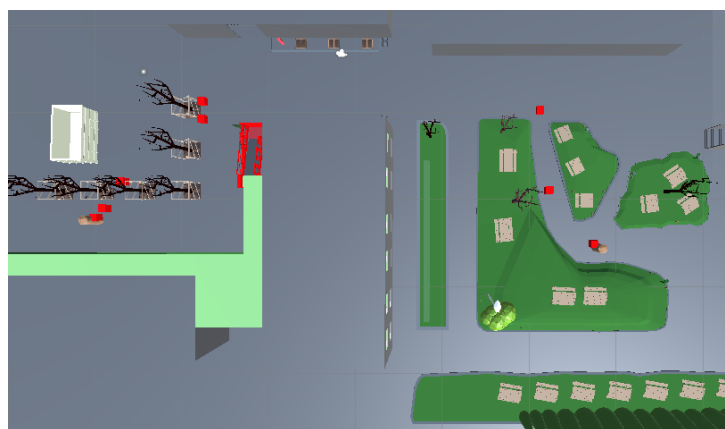


FIGURE 26 – Raffinement du parcours

2.7.6 Suiveur

Le « suiveur » est le dernier bot développé lors de cette soutenance. Ce bot servira surtout à déboussole le chasseur. Son objectif principal est de

maintenir en permanence une certaine distance avec le chasseur. Ainsi, il s'en rapprochera lorsqu'il sera trop loin, s'en éloignera lorsqu'il sera trop proche et cessera de se déplacer dans le dernier cas. Grâce à cette attitude très particulière, les chassés pourront tenter de suivre le chasseur aux côtés des suiveurs sans être soupçonnés.

2.7.7 Ajustement

Chaque PNJ à un script très précis à suivre mais, des imprévus sont toujours possibles. Ainsi, tous les bots vont réajuster leur direction lorsqu'ils reçoivent un choc ou toutes les certaines périodes. D'autre part, lorsqu'un bot reste bloqué un certain temps pour une raison inconnue, il se stoppera, se redirigera ou pivotera. Ces petits ajustements permettent de garder des PNJs opérationnels tout au long de la partie même si l'envie leur prenne de dévier de leurs instructions.

2.7.8 Synthèse

Trois bots aux attitudes différentes ont été implémentés. Cela empêche les joueurs de s'habituer aux déplacements caractéristiques qu'un PNJ pourrait posséder. Ils se feront, nous l'espérons, en permanence surprendre par leurs différences.

2.8 Base de donnée

Pour le projet nous avons décidé de rajouter un système de compte grâce à une base de données, Firebase.



Firebase

Pour jouer chaque joueur devra avoir un compte, notre système de connexion est simple il suffit d'une adresse mail et d'un mot de passe de plus de 6 caractères. Par exemple, nous pouvons voir sur l'image ci-dessous qu'il y a actuellement 4 comptes dans notre jeu.

<div> <input type="text" value="Recherchez par adresse e-mail, numéro de téléphone ou ID utilisateur"/> Ajouter un utilisateur </div>				
Identifiant	Fournisseurs	Date de création	Dernière connexion	UID utilisateur ↑
arthur.barbier@epita.fr	✉	28 avr. 2021		2O0Mdx7dfnRgo7HNPkObQMSyZ...
sacha.hibon@epita.fr	✉	28 avr. 2021		3gQ50RjD3SQSqJzVbgGQp24U4k...
ronan.pedron@epita.fr	✉	28 avr. 2021		6gBr1q0lcCSbH1p6C25y6Nrv1B13
matthieu.camart@epita.fr	✉	28 avr. 2021		1lz7yTTcGWUyNcfkYJEPYKNbYc92
<div> Lignes par page : 50 1 - 4 of 4 </div>				

FIGURE 27 – Comptes associés à notre jeu

2.8.1 Implémentation dans le jeu

Grâce à un menu login (pour se connecter) et register (pour se créer un compte) directement implémenté sur notre jeu, les utilisateurs peuvent se connecter et seront directement ajoutés à notre base de donnée.

2.8.2 Apprentissage de notre IA

L’implémentation des comptes dans notre jeu nous permettra de récolter et stocker des informations précieuses quant à l’apprentissage de notre IA. Par exemple, recueillir le temps d’utilisation des options tel que s’asseoir ou encore s’accroupir. Ainsi, nous n’avons pas encore exploité toutes les possibilités que nous offre cette fonctionnalité mais représente un pilier d’un dessein bien plus grand.

3 Musique et Bruitage

3.1 Musique

Notre premier son est une musique jazz que Ronan a joué avec des confrères avant la crise sanitaire. Cependant, nous espérons que la situation sanitaire s’améliorerait pour en enregistrer de nouvelles. Ceci n’étant pas possible, nous avons décider d’enregistrer des musiques que Ronan a joué avec un saxophone alto tout seul.

3.2 Bruitages

Pour cette partie, nous avons préparé les bruitages lorsqu'un coup de feu retentit et lorsqu'un humanoïde se fait molester.

3.3 Réglage du volume

Le joueur peut modifier le volume de la musique dans ses options du menu pour un confort optimal.

4 Avancement du projet

TABLE 2 – Avancement

	Actuel	Attendu
Édition 3D	85%	75%
Animation	80%	60%
Conception de l'environnement	75%	75%
GamePlay	95%	75%
I.A.	70%	50%
Musiques et bruitages	50%	50%
Création du site web	90%	100%
Bande annonce	0%	0%
Conception du logo	100%	100%
Mise en place multijoueur	90%	85%
Mémoire IA	50%	30%

4.1 Réussites

Nous sommes très fiers de notre avancement jusqu'à maintenant, nous avons tout fait nous même, nous avons vraiment l'impression que ce jeu nous appartient, et que chacune des pierres, chaque branche, chaque chaise ont été faites par nos mains.

4.2 Retards

Comme vous pouvez le remarqué nous avons mit un petit retard sur le site, celui-ci est du fait que nous avons voulu rajouté de nouvelle fonctionnalité à notre site qui n'étaient pas prévu à la base, cependant c'est un retard qui n'est pas grave pour la soutenance et ne nous pénalise point. Nous essayons de faire avancer le site en même temps que le projet il sera donc finalisé en même temps que notre jeu c'est à faire à la soutenance finale.

5 Prévision

Dans cette section, nous vous indiquons nos prévisions pour la prochaine et dernière soutenance.

TABLE 3 – Prévisions pour la prochaine soutenance

Tâche \ Soutenance	Actuel	Prochaine soutenance
Édition 3D	85%	100%
Animation	80%	100%
Conception de l'environnement	75%	100%
GamePlay	95%	100%
I.A.	70%	100%
Musiques et bruitages	50%	100%
Création du site web	90%	100%
Bande annonce	0%	100%
Conception du logo	100%	100%
Mise en place multijoueur	90%	100%
Mémoire IA	50%	100%

5.1 Environnement du joueur

Pour finaliser le menu, nous aimerions remodifier son design ainsi qu'ajouter des options dans celui-ci. Notamment enregistrer les préférences du joueur lorsqu'il change ses graphismes pour ne pas avoir à les fixer lors d'une réutilisation.

5.2 Réseau

Cette partie étant pratiquement finalisée, il ne reste seulement qu'à gérer les cas où l'utilisateur finirait par employer notre jeu d'une manière inappropriée. Par exemple, s'il décide de rejoindre la partie après le lancement de celle-ci, la solution serait qu'il apparaisse en temps que spectateur. Enfin, nous pourrions implémenter de nouvelles fonctionnalités tel qu'un tchat en jeu où les joueurs pourront parler entre eux.

5.3 Animation

Cette partie est quasiment terminée, concrètement, il ne reste qu'à implémenter le suivie de l'arme du chasseur en fonction de là où il regarde (que la vue "First Person Shooter" soit complète). Mais nous devons aussi ajouter de nouveau modèle de chassés avec les mêmes armatures animées ce qui est davantage un travail de modélisation 3D que d'animation.

5.4 Edition 3D

C'est la partie qui nous demande le plus de temps. Pour la prochaine soutenance nous aurons fini la carte (la map), ainsi que la confection d'autres modèles de chassés. La plus grande modification que nous voulons apporter est la création d'une salle d'attente. De notre expérience de joueur, nous préférons les jeux où nous pouvons nous mouvoir librement et ne pas être bloquer sur un menu sans pouvoir rien faire en attendant que la partie se lance. Nous allons donc créer un bar, un endroit où tout le monde se sent bien dans lequel une ambiance sera instauré par le décors mais également par le son.

Nous avons également un peu laissé de côté l'implémentation des couleurs sur nos modèles 3D alors que cela joue énormément sur l'apparence de notre jeu. Cela sera fait pour la 3ème soutenance.

5.5 Son

Le bar aura un ambiance musicale ainsi que des bruits de fonds tout cela fait spécialement pour cette salle. Nous implémenterons aussi tous les bruitages (coup de feu, coup...) ainsi que la musique du jeu.

5.6 Gameplay

Comme évoqué plus tôt cette partie est terminée, en théorie. Seulement, ce n'est que récemment, que notre jeu est opérationnel pour une partie complète. Ainsi, de nombreux ajustements pourront être proposé par les joueurs qui téléchargeront notre jeu en ligne.

5.7 IA

Il est prévu que deux nouveaux bots fassent leur apparition. Dans un premier temps, des PNJs imiteront les foules de synthèses que l'on peut observer dans des films. Plus particulièrement le principe des hirondelles développé dans la vidéo de ce lien <https://youtu.be/w-Oy4TYDnoQ>. Ensuite, un bot usant du machine learning devra être implémenté, il choisira ses déplacements en fonction des statistiques qu'il aura déterminé via sa base de donnée.

6 Conclusion

En définitive, nous sommes satisfaits de l'avancement du projet. Nous avons bien avancé dans le développement de notre jeu et avons atteint nos objectifs pour cette seconde soutenance.

La quasi-totalité du jeu a été implémentée : les principaux menus, le multijoueur, les modèles 3D, le personnage et ses actions (animations incluses), la carte, le site internet, ainsi que trois des bots aux comportements sensiblement différents.

Nous sommes très confiants à l'approche de la dernière soutenance et sommes pressés de développer les sections longuement détaillées lors de ce rapport.