

Soutenance finale

*Thibault Berland,
Gabriel Lecherf,
Etienne Samatiallian,
Mathieu Wattelet*

Juin 2019



Hide Life

Table des matières

1	Introduction	4
2	Cahier des charges	6
2.1	Introduction	6
2.2	Présentation du groupe	7
2.3	Origine et nature du projet	9
2.4	Objet de l'étude	13
2.5	Etat de l'art	14
2.6	Découpage du projet	14
2.7	Conclusion	15
3	Parties Individuelles	17
3.1	Partie d'Étienne	17
3.1.1	Modélisation de la map	18
3.1.2	Modélisation 3D des objets	19
3.1.3	Mise à l'échelle	21
3.1.4	Codage et autres missions de finition	21
3.1.5	Mission de chef de groupe	22
3.2	Partie de Mathieu	23
3.2.1	Première partie du projet	23
3.2.2	Deuxième partie du projet	26
3.2.3	Troisième partie du projet	31
3.3	Partie de Gabriel	34
3.3.1	1ère partie du projet : Acquisition des bases d'Unity et développement du gameplay de base	34
3.3.2	2ème partie du projet : Mise en place des interfaces, menus, et finalisation du gameplay de Base	40
3.3.3	Troisième partie du projet : Mise en commun, finalisation	45
3.4	Partie de Thibault	47
3.4.1	Première Etape	47
3.4.2	Deuxième Etape	50
3.4.3	Troisième Etape	53
4	Bilan de la réalisation	54
4.1	Difficultés ou déceptions	54
4.1.1	du point de vue d'Etienne	54

4.1.2	du point de vue de Mathieu	55
4.1.3	du point de vue de Gabriel	56
4.1.4	du point de vue de Thibault	57
4.2	Satisfactions	58
4.2.1	du point de vue d'Etienne	58
4.2.2	du point de vue de Gabriel	59
4.2.3	du point de vue de Thibault	59

5 Bibliographie **61**



Hide Life Blason

1 Introduction

Nous sommes réunis aujourd'hui pour vous présenter l'aboutissement de notre projet de jeu vidéo en 3D avec une vue à la première personne. Nous avons choisi de baptiser notre jeu Hide Life car il consiste en un jeu de cache cache entre des chasseurs et des caméléons échappés d'une expérience scientifique et qui ont la possibilité de se fondre par mimétisme dans le décor. Nous nous sommes inspirés du mod Prop Hunt du jeu Garry's mod. Notre équipe de travail se compose de quatre membres passionnés de jeux vidéos, Thibault Berland, Gabriel Lecherf, Mathieu Wattelet et Étienne Samatiallian comme chef de projet. Notre groupe s'est baptisé Dev Star.

Ce projet présentait plusieurs niveaux d'intérêt pour chacun d'entre nous. Il s'agissait tout d'abord de nous mettre d'accord sur un objectif commun, tant dans sa présentation finale que dans le partage des tâches et dans l'organisation de tous les tenants et aboutissants du challenge. Après être rapidement tombés d'accord sur le type de jeu et le scénario dont nous souhaitions voir le jour, nous avons convenu que la modélisation globale 3D de la map et des objets serait dévolue à Étienne. Thibault lui gérerait le site internet ainsi que la gestion de compte, la création de sessions et la liaison avec la base de données. Mathieu viendrait en renfort de Thibault sur le multijoueur et aurait en charge le coding et le débogage du jeu, ainsi que l'amélioration du gameplay avec Gabriel. Ce dernier ayant également à créer le menu d'accueil et la bande son des bruitages.

Un autre intérêt majeur était de nous familiariser avec un certain nombre de logiciels comme Unity, Blender, Brackets et avec le codage. Grâce à ce projet, nous devons apprendre le travail de groupe, c'est-à-dire apprendre à échanger nos points de vue, à

faire des compromis, à collaborer pour résoudre les difficultés, à respecter des délais et un cahier des charges et à appréhender une mission sous tous ses aspects. Ce travail présentait aussi l'intérêt de nous permettre de mieux connaître les aptitudes et qualités de chacun des membres du groupe afin de les optimiser et de les compléter.

Le but final était un but ludique et de partage. Nous souhaitions concevoir un jeu fonctionnel, à partager avec ses amis tout en retrouvant le plaisir d'un mode de jeu inspiré de Prop Hunt que nous avons tous apprécié dans Garry's mod ou dans Call of Duty. Enfin, il s'agissait aussi de nous prouver à nous-mêmes et à nos enseignants que nous étions capables de mener à bien ce projet complexe à notre niveau de connaissances en partant d'une simple idée commune.

Après cette introduction, nous avons décidé d'organiser notre rapport selon l'ordre suivant. Tout d'abord, nous reprendrons le cahier des charges puis nous ferons une présentation individuelle en fonction de la répartition des tâches que nous avons opéré. Ensuite, nous ferons un bilan de la réalisation en ce qu'elle a pu nous procurer comme difficultés ou déceptions et surtout comme satisfactions. Enfin nous conclurons.

2 Cahier des charges

2.1 Introduction

Notre groupe est constitué de quatre personnes qui sont Bertrand Thibault, Lecherf Gabriel, Wattelet Mathieu et Samatiallian Etienne. Ce dernier a été choisi comme chef de projet. Notre projet consiste en la réalisation d'un jeu vidéo qui se joue à la première personne et qui est inspiré du mod PropHunt du jeu Garry's mod. Nous avons choisi de créer ce jeu car nous partageons la même envie de réaliser un jeu en 3D avec une vue à la première personne. L'enthousiasme de tous les membres du groupe de mener cette création à bien a permis à notre projet de prendre forme. Ce jeu a pour finalité le divertissement et également le partage puisqu'il va nécessiter la participation d'au moins deux joueurs. Il se différencie des habituels FPS ou même jeux de stratégie par l'originalité de son "gameplay" (=expérience de jeu, jouabilité).

Afin de réaliser ce projet, nous allons devoir nous familiariser avec Unity et le principe de la modélisation 3D. De plus, le gameplay sera codé en C sharp. En ce qui concerne la modélisation de certains objets, Etienne va utiliser le logiciel de modélisation 3D appelé Blender. Le logiciel que va utiliser Thibault pour générer le site internet de notre création est Brackets ou sublime text. Un des aspects de ce projet est que le PropHunt dont il s'inspire est un jeu sans musique, donc nous respecterons ce critère d'origine et ne chercherons volontairement pas à trouver des musiques pour notre jeu. Néanmoins, nous nous soucierons des bruitages, éléments indispensables du gameplay pour augmenter la réalité des scènes et l'immersion du joueur.

La modélisation 3D sera principalement l'objet du travail d'Etienne. Gabriel s'occupera, quant à lui, essentiellement du gameplay. Mathieu sera chargé de la mise en place du multijoueur. Il sera soutenu dans cette tâche par Thibault qui a pour autre mission, la réalisation du site internet. La progression et les étapes de la réalisation du projet seront vues plus en détails dans la partie découpage de la mission.

2.2 Présentation du groupe

Notre groupe de travail est constitué de quatre personnes. Voici leur présentation :

Etienne Samatiallian : depuis toujours attiré par la modélisation 3D, les jeux vidéos et les projets de groupe, il s'est intéressé à certains logiciels de modélisation tel que Blender et Unreal Engine.

Thibault Berland : aime se lancer des défis, participer à la création de projets et apprendre en autodidacte pour pouvoir s'améliorer. Apprécie beaucoup le style de jeu du Prop Hunt.

Gabriel Lecherf : adepte du Prop Hunt sur Garry's mod, en quête de savoir quant au développement d'un jeu.

Mathieu Wattelet : aime coder et apprendre à utiliser tout ce qui est en rapport avec la programmation. Joueur débutant au Prop Hunt.

Présentation du groupe en général :



Hide Life

Notre nom de groupe est DevStar ; Dev pour développement et Star (étoile) pour notre espoir et notre passion pour l'informatique et la réalisation de ce projet.

2.3 Origine et nature du projet

Le projet que notre groupe souhaite voir aboutir est donc un jeu vidéo à la première personne. Ce jeu est inspiré du mod PropHunt (Hide'n'Seek en anglais) du jeu Garry's mod. Il se base sur le principe du jeu de cache cache, exception faite que ceux qui cherchent possèdent des armes pour tuer ceux qui se cachent. Ces derniers ont la possibilité de se transformer en un des objets de leur environnement immédiat. Ainsi, les joueurs qui se fondent dans le décor sont appelés caméléons car ils ont la faculté de se rendre en tout point identique à n'importe quel objet choisi dans le décor. Leur objectif est de rester en vie jusqu'à la fin du temps imparti pour le round. Ce sont ces joueurs caméléons qui ont inspiré le logo que nous avons choisi de réaliser pour Hide Life.

Passons maintenant à une explication du déroulement du jeu. Tout d'abord, au moins deux joueurs vont devoir se connecter au même serveur. Ensuite, aléatoirement, un des deux joueurs sera désigné pour être le chasseur et l'autre le caméléon. Le chasseur est pourvu au démarrage de la partie, de 100 pv (points de vie) et de deux armes, une arme à feu et une arme de corps à corps. La cible aura environ 1 minute d'avance par rapport au chasseur afin

The logo consists of the words "PROP HUNT" in a bold, orange, sans-serif font. The text is contained within a dark, rectangular background that has a slight 3D effect with a shadow on the right side.

de trouver la parfaite cachette. Ce caméléon possédera la capacité de se transformer à volonté dans les objets présents sur la map. Ainsi, pendant une minute, les proies pourront se déplacer sur la map afin de trouver une cachette la plus crédible possible en fonction de l'environnement.

Passé ce délai, les chasseurs vont entrer en piste et circuler sur la carte afin d'essayer de les débusquer. Ils vont tirer sur les objets présents dans les pièces afin d'essayer de dépister et de toucher les caméléons (une marque particulière apparaissant sur ceux-ci lorsqu'ils sont touchés). Les chasseurs devront utiliser la logique et l'instinct pour repérer les objets dont la présence leur apparaîtra comme inappropriée ou incongrue. De plus, si un chasseur touche un vrai objet alors il perdra des points de vie. Les caméléons quant à eux, posséderont aussi un certain nombre de pv en fonction de la taille de l'objet en lequel ils se seront incarnés.

La partie se découpe en round/en tour et ce nombre de tours est lui-même déterminé par le nombre de joueurs présents dans la partie. Le round se termine lorsque tous les caméléons sont morts ou que les chasseurs n'ont pas réussi à trouver ceux qui se cachent dans le temps imparti. De plus, en fonction des actions des joueurs durant la partie, un certain nombre de points leur est attribué.

La partie se termine avec un affichage des scores et la mise en avant du joueur arrivé premier avec le plus grand score. Nous pensions rajouter en bonus, si jamais quelqu'un arrive à prendre de l'avance sur le planning, un système de monnaie gagnée au fur et à mesure des parties. Cette cagnotte serait alors en lien avec un système de boutiques sur le jeu dans lesquelles il serait

possible d'acheter des skins que ce soit pour les armes ou pour les chasseurs. Si nous en avons le temps, nous allons également essayer de créer la possibilité aux chasseurs de changer d'armes, en implémentant de nouvelles armes. Pour ne pas désavantager les caméléons, nous pensions aussi leur conférer de petits pouvoirs qui leur permettraient de s'enfuir plus facilement par exemple.

Pour ce qui est de la map (la carte), le groupe a dans un premier temps souhaité reprendre la forme du bâtiment de l'Epita Lyon comme map principale. Le bâtiment sera cependant représenté sans aucune image de l'Epita pour éviter les problèmes de copyright ou d'image. Cette carte sera donc à priori constituée d'un sous-sol et de trois étages. Elle comportera un assez grand nombre d'objets différents afin que les caméléons aient un large éventail de transformations possibles.

Dans le site internet que Thibault va réaliser, on pourra trouver :

- Une présentation du projet en corrélation avec l'introduction du cahier des charges.
- De courtes présentations pour les chasseurs et les caméléons.
- L'état d'avancement du projet.
- Une sorte de Wiki.
- Les outils utilisés.
- Une boutique.
- Un tableau des scores.
- Un lien de téléchargement du jeu.

Dans le cadre du projet, nous allons créer un compte sur plusieurs réseaux sociaux afin de promouvoir notre projet avec notamment son état d'avancement, recueillir les suggestions et partager des photos.

Histoire : Des caméléons ont été capturés pour faire des expériences scientifiques sur le changement d'apparence. Suite à un test défaillant, les caméléons s'échappent et développent la capacité de prendre l'apparence et la forme d'objets de l'environnement immédiat. Des chasseurs sont chargés de les éliminer.

2.4 Objet de l'étude

Le but de ce projet est de réaliser un travail de groupe sous un aspect ludique puisque le résultat sera un jeu. De ce fait, nous serons amenés à développer notre capacité à échanger nos différents points de vue, à nous accorder sur les aspects variés de notre mission, à appréhender les multiples disciplines nécessaires à l'élaboration de notre création, à envisager le projet dans sa globalité et dans chacun de ses aspects, à utiliser au mieux les aptitudes et qualités de tous les membres de DevStar et à nous répartir les différents aspects de ce travail. Le tout en respectant des dates précises de rendu du travail ainsi que le cahier des charges, ce que nous serons amenés à être capables de produire dans notre future vie professionnelle. Ainsi, coopération et efficacité de travail devront être au rendez-vous, sur fond de bonne humeur et d'enthousiasme ! Cela nous permettra aussi de mettre en pratique les connaissances acquises tout au long de l'année, mais aussi d'en acquérir de nouvelles (modélisation 3D, Unity, bande-son,réseau).



Image tirée du jeu Prop Hunt

2.5 Etat de l'art

S'il est vrai qu'aujourd'hui le Prop Hunt est fortement associé à Garry's Mod puisque c'est le jeu qui l'a le plus rendu populaire, il existe dans de nombreux jeux en tant que mode de jeu secondaire, comme par exemple Call of Duty ou encore Team Fortress 2 pour ne citer que les plus connus. C'est d'ailleurs sur Team Fortress 2 que le Prop Hunt est né. En effet, il est apparu comme modification de serveur (appelé un "mod" dans ce milieu), avec des terrains de jeu spécifiques (la map) à ce mod.

Néanmoins, il n'y a pas vraiment eu de jeux populaires où le Prop Hunt est le mode de jeu principal. Il reste possible de trouver de petits jeux basés sur le Prop Hunt comme Witch it. Ce qui fait le succès de Garry's Mod par rapport aux autres jeux cités c'est sa communauté très active couplée à la facilité pour celle-ci de rajouter du contenu dans une grande base de données commune (le Workshop de Steam par exemple), appelée Addons. Ainsi, grâce à la grande quantité de contenu créé par la communauté, il est possible de trouver beaucoup de serveurs, pouvant tous pourtant se distinguer par les Addons utilisés.

2.6 Découpage du projet

Lors du choix du projet, nous nous sommes concertés et avons procédé à la répartition des différentes tâches de celui-ci. Thibault, qui possède déjà quelques bases dans les langages de programmation de site internet tels que le Javascript, ainsi que certaines connaissances en HTML et CSS se charge de la création du site internet. Étienne quant à lui, a déjà réalisé des modélisation

3D et possède donc de petites bases en modélisation, c'est pourquoi il s'est proposé pour réaliser celle de la map et des objets. Comme nous avons choisi de modéliser les locaux de l'Epita Lyon et que cette tâche sera assez longue et complexe, Gabriel pourra être amené à lui prêter main forte. Le codage du jeu sur Unity sera fait par Mathieu et Gabriel qui devront travailler de concert afin de trouver plus facilement les erreurs qui pourraient être à l'origine de Bug. Mathieu et Gabriel seront aussi en charge de la bande sonore du jeu ainsi que des bruitages. Notre projet a pour complexité principale la mise en réseau du jeu. Mathieu et Thibault s'occuperont ensemble de cette partie du projet. Thibault se chargera du développement du site internet en utilisant du code HTML et CSS pour mettre en page la structure ainsi que la partie graphique. Il codera des programmes en Javascript pour gérer les différents modules ainsi que les éventuels problèmes de mise en page. Il créera une base de données pour gérer des comptes et faire des rapports de bugs ou pour signaler des problèmes, puis mettra en place des sessions en fonction des comptes utilisateurs.

2.7 Conclusion

Nous considérons ce projet comme une réelle opportunité à tous points de vue et sommes impatients de progresser dans sa réalisation. Nous espérons tout particulièrement rendre un projet à la hauteur de la confiance que l'équipe de nos enseignants placent en nous.

Partie du projet/Etape	Soutenance 1	Soutenance 2	Soutenance 3
Modélisation 3D de la map	50%	80%	100 %
Optimisation de la map	10%	20%	100 %
Modélisation 3D des objets	15%	75%	100 %
Zone de tutoriel	0%	20%	100 %
Menu d'accueil	15%	75%	100 %
Coding et debug du jeu	40%	75%	100 %
Amélioration du Gameplay	15%	50%	100 %
Bande son	15%	65%	100 %
Multijoueurs	0%	50%	100 %
Graphisme du site	60%	90%	100 %
Liaison avec la base de données	30%	80%	100 %
Gestion de comptes	0%	60%	100 %
Création de sessions	0%	30%	100 %

Planning de l'avancement des différentes tâches du projet

Partie du projet/Etudiant	Etienne	Thibault	Gabriel	Mathieu
Modélisation 3D de la map	X			
Optimisation de la map	X			
Modélisation 3D des objets	X			
Zone de tutoriel	X			
Menu d'accueil			X	
Coding et debug du jeu			X	X
Amélioration du Gameplay			X	X
Bande son			X	
Multijoueurs		X		X
Graphisme du site		X		
Liaison avec la base de données		X		
Gestion de comptes		X		
Création de sessions		X		

3 Parties Individuelles

Comme nous l'avons évoqué dans le cahier des charges, chacun des membres du groupe s'est vu attribuer un certain nombre de tâches en fonction de ses aptitudes, de ses affinités ou de ses envies.

3.1 Partie d'Étienne

J'ai ainsi été désigné chef de groupe et responsable de la modélisation 3D de la map et des éléments du décor ainsi que de différentes tâches ponctuelles telles que la modélisation du caméléon, la fonction d'ouverture fermeture des portes, le changement d'arme, l'optimisation et l'amélioration de la finition de la carte, le placement aléatoire des objets au fil des parties, l'amélioration de certains bruitages et avec Gabriel de la création d'une autre

scène servant de zone de tutoriel. J’aborderai successivement les divers aspects de ma participation au projet de DevStar.

3.1.1 Modélisation de la map

Tout d’abord intéressons nous à la map par elle-même. Élément indispensable au déroulement du jeu, j’avais décidé de reproduire les locaux de notre école le plus fidèlement possible sans toutefois faire apparaître le nom de l’Epita pour des raisons de copyright. Le projet initial comprenait quatre étages que nous avons par la suite décidé de scinder en deux map distinctes de deux étages pour offrir plus de variétés de décor aux futurs participants et aussi car la map apparaissait trop grande, entraînant un déséquilibre dans le jeu. En effet, les caméléons avaient trop d’espace pour se cacher et les chasseurs trop de surface à couvrir. Suite aux conseils reçus lors de la deuxième soutenance, nous avons décidé de nous limiter à une map de deux étages.

J’ai également pris des libertés quant à l’affectation des fonctions des pièces pour créer de la diversité de lieux. J’ai ainsi organisé un accueil, deux cuisines, une salle à manger, un salon, une réserve, des couloirs, une bibliothèque, un foyer pour les étudiants, une salle de repos, des toilettes, une classe et une salle de cours informatique. Je voulais sortir du schéma foyer, couloirs, salles de classe qui offrait peu de fantaisie d’ambiance. Afin de modéliser la map en 3D, j’ai suivi des cours sur Open Classroom pour être en mesure de manipuler le logiciel Unity. C’est en effet grâce à celui-ci que j’ai pu reconstituer les locaux de l’Epita Lyon en m’aidant des plans d’évacuation incendie affichés dans nos locaux. Je pensais également que cela pourrait amuser mes camarades de tester

le jeu dans un environnement que nous parcourons quotidiennement. J'ai commencé par placer une plateforme qui m'a servi de support pour les éléments de la carte. Après avoir monté les murs du foyer pour les étudiants, je me suis attaqué aux espaces dédiés du rez-de-chaussée avec une difficulté liée au grand nombre de courbes des différentes cloisons. Pour la réalisation des étages, je me suis aidé d'une préfab qui est un dossier dans lequel on crée des modèles que l'on peut ensuite importer autant de fois que nécessaires dans le projet. Ainsi chaque fois que l'on va décider de modifier un élément A dans la préfab, tous les éléments A présents sur la scène principale du projet vont être modifiés.

La deuxième étape de la réalisation de cette map a été la préparation des textures que j'ai trouvées libres de droits sur le site Asset Store Unity et que j'ai ensuite pu appliquer aux surfaces de la map. Cette opération m'a pris plusieurs heures de travail pour obtenir une bonne application des textures aux milliers de blocs présents sur la carte.

3.1.2 Modélisation 3D des objets

Le deuxième aspect de ma mission de modélisation concernait les objets 3D du décor. Leur diversité et leur nombre étant importants puisqu'ils sont le support de transformation des caméléons et qu'ils permettent de rendre les décors réalistes et attrayants. Pour mener à bien cette étape, j'ai dû consacrer beaucoup de temps sur Open Classroom pour savoir manipuler le logiciel de modélisation 3D Blender qui offre énormément de possibilités de modélisation mais du coup qui est un peu long à apprivoiser. Il était important pour moi de créer totalement les objets afin qu'ils restent des

créations originales tout en étant reconnaissables dans leur fonction. C'est ce qui m'avait fait privilégier dans un premier temps la map au détriment du développement des objets par rapport aux prévisions de notre cahier des charges. J'avais été en mesure de rattraper ce retard pour la deuxième soutenance. Pour réaliser chacun de mes objets, j'ai d'abord commencé par faire une esquisse papier de l'objet en question. J'ai ainsi pu modéliser différentes sortes de chaises, de tables, de lampes, de bureaux mais également des canapés, des pots à crayons remplis de stylos, des canettes et leur distributeur, une télévision, un ordinateur avec son écran, sa tour, son clavier et sa souris, des extincteurs, des cônes de signalisation, un réfrigérateur, un distributeur de savon, des toilettes, un lavabo, un four, un sèche-main, un fauteuil de bureau, des feux de signalisation, une poubelle, des rouleaux de papiers toilette, une bibliothèque avec ses livres...un caméléon et un skateboard. J'ai dû pour cela surmonter des difficultés liées par exemple à la création d'arrondis ou au poids des objets ou encore à la perte de fichiers d'objets que j'ai dû intégralement refaire. Pour contourner le problème des arrondis de certains éléments du décor, j'ai débuté avec une sorte de pixellisation des objets de façon à pouvoir soustraire certains cubes et ainsi lisser les surfaces jusqu'à obtenir une impression d'arrondi. La création de la bibliothèque m'a pris environ sept à huit heures. Elle se composait au départ d'environ quatre mille six cents blocs ce qui était extrêmement lourd pour les ordinateurs des futurs joueurs. J'ai donc allégé l'ensemble en optimisant le nombre de blocs. Désormais, la grande majorité des livres se compose de quatre blocs et seulement quelques uns d'entre eux en comptent encore quatre-vingt pour la finesse du rendu esthétique. La longueur du temps nécessaire à la réalisation et à la modification de la bibliothèque est

due à l'obligation de créer le meuble puis les livres et d'imaginer la disposition de chaque élément et enfin d'appliquer les textures. Alors que pour la création d'un ballon par exemple, il suffit de créer une sphère et de lui appliquer une texture.

3.1.3 Mise à l'échelle

Une fois ces deux types de modélisation 3 D achevés, j'ai alors pu intégrer ces éléments de décors à la map. Il m'a alors fallu faire une mise à l'échelle globale en fonction de la taille prévue d'Ethan, notre prototype de personnage. J'ai dû redimensionner quasiment tous les éléments de la map et un grand nombre d'objets pour la cohérence de nos scènes. J'ai ainsi modifié la hauteur des murs, la dimension des pièces et fait la mise en adéquation de la taille du personnage par rapport aux objets et aux volumes spacieux.

3.1.4 Codage et autres missions de finition

L'aspect de ma mission le plus varié (car consistant en pleins de petites mises au point) consistait en du codage pour la perte des points de vie des joueurs avec les dégâts de chute et pour créer la possibilité de zoomer avec l'arme sur un élément choisi du décor. j'ai codé les fonctions de dégat de chute. J'ai également pu faire un peu d'animation sur le personnage.

J'ai eu l'initiative de créer la base d'une zone de tutoriel pour apprendre les fonctionnalités de déplacement du joueur avant qu'il n'intègre la map. J'ai également fini la création du logo du jeu et ai pu le mettre en valeur sur différents supports tels que des

sweatshirts ou des casquettes afin de nous créer une publicité supplémentaire et de susciter la curiosité de nos camarades.

J'ai également en phase finale procéder à quelques missions de finition telle que la modélisation du caméléon symbole de notre jeu et qui présente l'originalité de se déplacer en skateboard. Ceci en fait un élément ludique et qui m'a permis de contourner le problème de l'enroulement de sa queue qui gênait dans les déplacements. De même, j'ai veillé au bon fonctionnement de l'ouverture et de la fermeture des portes durant les déplacements des chasseurs et des caméléons. J'ai pu coder pour que le chasseur ait la possibilité de changer d'arme en cours de partie pour toujours plus de fonctionnalités. J'ai optimisé la carte et en ai amélioré la finition pour rendre le décor plus attrayant. J'ai créé la fonction d'un positionnement aléatoire des objets afin de susciter la surprise des joueurs et de ne pas tomber dans la facilité d'un décor toujours identique. J'ai amélioré certains bruitages tels que les bruits de pas variables selon la nature du sol (herbe, cailloux, béton, bois), les bruits de respiration type essoufflement pour un chasseur à cours d'énergie, des bruits liés à l'ouverture et fermeture des portes, ou le souffle du vent. Enfin avec Gabriel, nous avons retravaillé la zone de tutoriel en créant une autre scène.

3.1.5 Mission de chef de groupe

En tant que chef de groupe, j'ai coordonné au mieux le travail de chacun en suivant les avancées de chacun en fonction des échéances et en essayant de maintenir à la fois le lien parfois tendu en raison de point de vue divergent, la motivation et l'enthousiasme de mes camarades. Tâche rendue ardue par mon manque

d'expérience et de maîtrise des différentes capacités et connaissances nécessaires à l'accomplissement de notre mission. J'ai encadré au mieux l'équipe pour la mise en commun des différentes parties du projet.

3.2 Partie de Mathieu

3.2.1 Première partie du projet

En début de projet, lors de la répartition des tâches nous avons dû hiérarchiser nos priorités. Le codage des personnages nous est apparu comme un élément essentiel à travailler dès le début du projet pour plusieurs raisons. Tout d'abord, le rôle des personnages est un élément clé du jeu, sans eux le jeu n'a pas de sens. D'autre part, leur codage est relativement facile par rapport à celui du mode multijoueur par exemple, ce qui nous offre la possibilité de prendre en main le logiciel Unity 3D, dont nous ignorions tout. D'un commun accord, il a été décidé que Gabriel et moi gérerions ce travail sur les personnages et leurs caractéristiques. Dans le jeu, il existe deux rôles de personnages différents :

- le chasseur : il a des armes et doit abattre les caméléons, durant un temps limité. Le chasseur peut être seul sur la carte ou accompagné par d'autres.

- les proies : elles doivent survivre en se cachant pour échapper aux tirs des chasseurs, durant un temps limité. Elles ont la particularité de pouvoir se transformer en objet.

Bien qu'opposés dans le jeu, le chasseur et les proies présentent de nombreux points communs au niveau du code, ce qui fait qu'il est plus judicieux de les coder simultanément, tout en tenant évidemment compte de leurs spécificités. Avec Gabriel, dans un souci

d'efficacité et afin d'aborder les fonctions de chaque personnage avec le plus de précision et d'efficience possible, nous nous sommes répartis le codage de ses différents personnages. Il m'a été attribué la responsabilité de gérer la capacité du personnage à se transformer en objet.

Dans la consigne, deux logiciels étaient conseillés : Unity 3D et Blender. Blender est un logiciel de modélisation 3D et Unity 3D permet de faire interagir des modèles 3D. N'ayant pas à créer des modèles, mais devant gérer leurs interactions, j'ai fait le choix de développer la fonction sous Unity 3D. Je n'ai jamais codé avec ce logiciel, mais je l'ai découvert grâce aux jeux Flash. Ces derniers sont généralement des jeux "indé", c'est à dire des jeux indépendants qui n'ont pas été produits par des studios importants et qui n'ont pas de sponsors. La plupart du temps, ils sont créés par un seul développeur. Ils sont quasiment toujours développés via Unity 3D. Il a été très facile d'obtenir le logiciel Unity 3D qui est un logiciel gratuit pour les particuliers développant des jeux. Avant de pouvoir le télécharger, il faut impérativement répondre à un questionnaire ayant pour but de vérifier que le jeu qui sera créé n'a pas pour but de devenir un jeu sponsorisé ayant des milliers d'utilisateurs. Ensuite, ne maîtrisant pas le logiciel Unity 3D, j'ai dû me familiariser avec lui. Pour cela, j'ai regardé des tutoriels présents sur Youtube. Je les ai trouvés intéressants mais parfois abstraits, j'avais besoin de manipuler pour que leurs contenus aient un sens concret. J'ai alors essayé par moi même. Je me suis rendu compte que le logiciel Unity 3D est bien conçu et qu'il est relativement intuitif. Pour les fonctions simples que je dois réaliser en ce début de projet, son utilisation s'est bien passée. Je n'ai pas rencontré de difficultés majeures .

Comme il a été décidé par le groupe, je dois donc permettre au

personnage de prendre l'apparence d'un objet afin d'échapper au chasseur. Cela consiste en un changement de l'apparence et de la hitbox du joueur en celle de l'objet ciblé. J'ai alors répertorié les différents éléments qui sont nécessaires pour établir une fonction donnant cette capacité au personnage. Cette fonction se décompose en quatre grandes parties : - récupérer l'objet ciblé pour la transformation du joueur - modifier la hitbox du joueur - modifier l'apparence du joueur - repositionner le joueur

Premièrement, pour être capable de prendre l'apparence de l'objet, le joueur doit récupérer l'objet ciblé. En fait, la transformation ne peut se réaliser que si l'objet se trouve dans le centre du champ de vision du joueur. Ils sont forcément alignés. Face à ce protocole, j'ai opté pour l'utilisation des Array. Les Array sont une classe liée à Unity3D qui permet de détecter des objets en ligne droite. Cette fonction détecte donc l'objet ciblé par le joueur et le renvoie aux autres fonctions. Ensuite, une fois que le personnage accède à l'objet, il faut modifier sa hitbox qui doit être remplacée par celle de l'objet. Pour transformer la hitbox du joueur, il faut d'abord retirer sa hitbox initiale. Cette opération est très complexe car il existe de nombreux types de hitbox. Je ne peux pas me contenter de modifier les paramètres de la hitbox mais je dois aussi modifier sa structure. Ensuite, il suffit de copier les valeurs de la hitbox de l'objet et de les ajouter au joueur. Troisièmement, une fois la hitbox copiée, il faut ensuite modifier l'apparence du joueur, son mesh. Tout comme pour la hitbox, il faut détruire l'apparence initiale et remplacer son mesh par un clone de l'apparence de l'objet cible. Cette étape s'est révélée complexe, dû à mon manque d'expérience sur ce logiciel. J'ai laborieusement dupliqué le mesh, paramètres par paramètres, jusqu'au moment où une erreur de manipulation, m'a permis de découvrir une fonctionnalité

de Unity 3D qui permet de cloner la globalité d'un mesh et de gagner du temps et de la précision. Enfin, une fois le personnage transformé, il suffit de le repositionner directement dans le jeu. Pour cela, les coordonnées de position du joueur sont nécessaires. Dans un premier essai, je n'ai pas repositionné le joueur, et j'ai constaté que la transformation avait tendance à faire passer le joueur à travers le sol. Par conséquent, dans un deuxième essai, je l'ai fait réapparaître quelques centimètres plus haut et j'ai ainsi corrigé le problème. Une fois ces quatre étapes achevées, le personnage peut enfin prendre l'apparence d'un objet. La fonction a été testée sur la map, et ainsi par exemple, le personnage peut se transformer en mur. Ce début de travail du projet, où je devais réaliser un personnage ayant acquis la capacité de se transformer en objet pour échapper au chasseur, m'a permis de me familiariser avec un logiciel inconnu Unity 3D. Cette prise en main du logiciel faite, je peux donc aborder un domaine plus complexe : le mode multijoueur.

3.2.2 Deuxième partie du projet

Suite aux modifications apportées d'un commun accord par le groupe, mon planning pour la seconde partie du projet a été complètement remodelé. Initialement, il a été convenu que je m'occupe avec Gabriel de créer et de développer l'interface graphique des armes et du codage de leur fonctionnement. En parallèle, avec Thibault nous devons élaborer le mode multijoueur. Je devais également travailler sur les points de vie du personnage et des objets. Afin que la répartition du codage soit plus équitable, il a été décidé que je ne gèrerai plus que la partie concernant le mode

multijoueur. En contrepartie, ma tâche de travail sur la partie en ligne serait beaucoup plus conséquente. Je continuerai à travailler en étroite collaboration avec Thibault qui apportera ses connaissances sur les bases de données SQL. Lors de la phase précédente du projet, j'avais essentiellement découvert les différentes classes de Unity3D. Cette nouvelle répartition a l'avantage de m'offrir l'opportunité de me familiariser avec la partie réseau du jeu.

Pour toute la gestion du mode multijoueur, j'ai continué à utiliser le logiciel sur lequel nous avons débuté le jeu : Unity3D, qui répond efficacement à nos besoins. Mais, en concertation avec mon groupe, j'ai également opté pour le choix d'un asset, nommé Unet. Il s'agit d'un asset officiel développé et pris en charge par Unity. Cet asset a été conçu pour permettre de gérer de manière simplifiée toute la partie multijoueur sur Unity. Personnellement, je n'ai aucune connaissance de cet outil et je possède des compétences très limitées dans le domaine du programme réseau. J'ai donc étudié attentivement les différentes documentations relatives et j'ai suivi plusieurs tutoriels expliquant son fonctionnement. Sa prise en main m'a semblé complexe car cet asset gère un domaine vaste et possède plusieurs fonctionnalités diverses et variées. Cependant, sa pertinence est évidente car ayant pour fonction de simplifier les différentes démarches et protocoles, toutes les parties contraignantes qui étaient liées au logiciel sont résolues par d'autres fonctionnalités prévues à cet effet.

Dans notre projet de jeu, inspiré du prop hunt, deux options s'offrent au joueur : soit il est le chasseur (hunter), soit il est une proie (prop). Quel que soit le rôle choisi, le joueur interagit constamment avec ses coéquipiers et ses adversaires. Ce mode multijoueur implique donc que les différents joueurs aient une vision

identique du jeu et des actions. Les différentes perceptions offertes aux différents joueurs doivent être synchronisées entre elles. Pour ce faire, la mise en œuvre de ce mode multijoueur doit se décliner en 3 grandes étapes :

- la connexion/déconnexion au serveur en mode « hotseat »
- le contrôle de son personnage et seulement de lui
- la synchronisation des différentes actions.

Premièrement, en ce qui concerne la connexion et déconnexion au serveur en mode « hotseat », il paraît évident que les joueurs puissent se connecter ensemble à une même partie et interagir les uns avec les autres . Il existe d'ailleurs trois types de parties dépendants du mode de connexion choisi :

- les parties à distance où tous les joueurs sont chacun sur leur ordinateur et doivent se connecter à un serveur distant . Ces parties sont dites « en ligne »

- les parties locales, où là encore chaque joueur est sur son ordinateur, mais cette fois il se connecte à un serveur proxy. Ces parties sont dites « LAN »

- les parties où tous les joueurs n'ont pas un ordinateur qui leur est propre, mais sont sur le même ordinateur qui sert également de serveur. Ces parties sont dites « hotseat ».

Afin de ne pas bloquer le travail du groupe, j'ai dû impérativement mettre en œuvre la connexion et le déconnexion au jeu. Cependant, face à la complexité générée par les parties « en ligne » et « LAN » et face à mon inexpérience dans la maîtrise de l'asset Unet, il a été décidé avec le groupe que je ne gérerai pour cette soutenance, que le cas des parties « hotseat » qui peut être réalisé plus simplement. Les connexions et déconnexions des autres cas de figure, seront travaillées ultérieurement, en vue du projet final,

puisque nous voulions avant tout mettre l'accent sur les points permettant au jeu de prendre forme afin de présenter un rendu jouable.

Deuxièmement, une fois la réalisation de cette phase de connexion et déconnexion résolue, j'ai pu me consacrer au travail sur la gestion des déplacements et des actions du personnage. Il paraît naturel dans un jeu en ligne que le joueur ne puisse contrôler et gérer que les déplacements et les mouvements de caméra de son personnage, et surtout pas ceux de ses adversaires ou même de ses coéquipiers. Pour mettre en œuvre cette tâche, j'ai très vite été confronté à un problème majeur propre au « hotseat » : le mouvement d'un personnage impacte directement le mouvement des autres personnages. En effet, quand un joueur déplace ou fait simplement bouger son personnage, qu'il appuie sur une touche ou bouge la souris, cela génère des inputs. Aussitôt, les autres personnages effectuent le même déplacement ou le même mouvement, puisque les scripts des différents personnages sont identiques. Par exemple, si un joueur décide de faire sauter son personnage alors, l'input est directement envoyé aux autres personnages comme si l'ordre leur était donné par leur propre joueur. Résultat, quand un personnage saute tous les autres personnages sautent également, indépendamment du souhait de leur joueur. Cette situation rend impossible le jeu. Pour parer à cela, j'ai dû coder un script empêchant la transmission de l'input aux autres personnages. Chaque joueur est ainsi devenu le seul maître des mouvements de son personnage.

Troisièmement, une partie en mode multijoueur nécessite la

synchronisation de la position des différents personnages. Si un joueur décide de bouger, tous les autres joueurs doivent également le voir bouger. Cette fonctionnalité a nécessité le codage de plusieurs scripts différents, définissant la fréquence d'envoi des modifications de position et permettant aux autres joueurs de visualiser les mouvements des personnages. C'est une étape absolument indispensable dont le dosage de la fréquence doit répondre à une exigence précise et adéquate. Si l'information est transmise trop fréquemment, le serveur risque de surchauffer ou de rencontrer divers problèmes. Mais, inversement, si l'information est transmise trop rarement, les joueurs ne peuvent pas interagir entre eux et le jeu perd de son intérêt. Puisque chaque joueur a sa propre vision de la map, il a fallu ensuite coder un second script permettant d'actualiser la position et le mouvement de tous les autres joueurs du point de vue de chacun. Il en est de même pour l'apparence. Un premier problème est alors survenu : tous les joueurs paraissaient se téléporter d'un point à un autre au lieu de bouger de manière fluide. De plus, leurs mouvements étaient très saccadés, alors qu'ils auraient dû être lisses. Pour résoudre cela, il a été nécessaire d'utiliser une fonctionnalité de Unet permettant de faire apparaître le chemin qu'empruntent théoriquement les joueurs pour rallier leur nouvelle position (celle où ils se sont téléportés), depuis leur ancienne, (celle avant la téléportation). Dès lors, l'affichage du déplacement des joueurs est devenu fluide et continu. Le même travail a été fait avec le mouvement des personnages qui sont devenus lisses et plus réalistes, rendant le jeu plus agréable à regarder et plus jouable.

Une fois ces trois étapes résolues, le jeu a acquis son statut multijoueur puisque les interactions entre les différents joueurs sont rendues possibles. Les joueurs peuvent enfin se retrouver à

plusieurs sur une même map et contrôler leur personnage à leur guise.

En conclusion, lors de mon travail précédent pour la première soutenance du projet, j'avais découvert un logiciel qui m'était inconnu : Unity 3D. J'avais alors eu pour objectif de permettre au personnage de se transformer en objet pour échapper au chasseur. Mon travail avait porté sur un seul personnage à la fois. Lors de cette deuxième partie du projet, j'ai pu découvrir le mode multijoueur où j'ai été confronté aux problèmes liés aux interactions entre les personnages. Cela a donc enrichi mon expérience en augmentant la complexité de la tâche. J'ai d'ailleurs pu découvrir un asset du logiciel Unity3D. J'ai eu plus de difficultés à prendre en main et à me familiariser avec cet asset puisque je suis confronté à un pan de la programmation que je connais peu et dans lequel je ne suis pas à l'aise.

3.2.3 Troisième partie du projet

La répartition de la dernière partie du projet coulait de source et a été réalisée rapidement. Chacun s'engageait à achever la partie commencée auparavant. Ainsi, j'ai continué à gérer le mode multijoueur. La fois précédente, j'avais mis en place le mode "Hotseat", il me reste à m'occuper des modes "LAN" et "en ligne". Mais auparavant, j'ai dû rajouter des fonctionnalités au mode "Hotseat" qui seront également utilisées pour les deux autres modes :

- définition du point d'apparition des joueurs
- retrait d'un personnage qui servait de démonstration.

La définition du point d'apparition des joueurs n'avait pas en-

core été réalisée car jusqu'à présent sa présence n'était pas indispensable. Il m' a suffi de définir un nouvel élément du network manager et de lui implémenter un nouveau composant lui permettant d'identifier le point d'apparition des joueurs. Quant au retrait du personnage, il était indispensable pour notre démonstration lors de la deuxième soutenance, mais il est gênant pour le mode multijoueur. En effet, la première personne qui se connecte à ce personnage, n'apparaît pas au point de spawn mais au centre de la map, ce qui isole ce personnage des autres joueurs.

Une fois ces deux modifications effectuées, j'ai pu me consacrer pleinement à la mise en place des modes "LAN" et "en ligne". Ces deux modes ayant un fonctionnement identique, je les ai réalisés suivant un même protocole. Tout d'abord, j'ai réalisé le mode en ligne. Cela s'est fait en 2 étapes :

- création du matchmaking
- connexion à la partie

J'ai donc commencé par créer une fonction jouant un rôle de matchmaking, c'est à dire permettant au joueur de trouver une partie lorsqu'il se connecte et clique sur le bouton "jouer". La création de cette fonction s'est faite assez rapidement grâce à l'asset Unet conçu pour simplifier ce type de démarche.

Ensuite, j'ai dû écrire une fonction permettant au joueur de se connecter à la partie trouvée. Cette fonction a pour rôle de faire passer le joueur du lobby à la scène de jeu. Ainsi, le joueur est pleinement intégré au jeu. Cette fonction doit également s'occuper de synchroniser les personnages. C'est à dire de faire en sorte qu'ils voient tous le même jeu en même temps. Ainsi, quand un joueur se déplace ses mouvements sont visibles de tous. Cette étape s'est avérée plus complexe que prévu car je maîtrise mal les

scènes. J'ai eu du mal à passer d'une scène à l'autre et à gérer leurs interactions. Il faut dire que le jeu contient plusieurs scènes différentes.

Une fois le mode en ligne achevé, il m'a fallu m'occuper du mode "LAN". Puisqu'il est proche, du mode en ligne, j'ai eu recours à un protocole quasi identique. J'ai donc repris les deux étapes précédentes, en mettant d'abord en place le matchmaking puis la connexion à la partie. Ce travail sur le mode "LAN" a donc été plus simple.

Au fur et à mesure des tests, lors de la mise en place de ces deux modes multifonctions, j'ai été confronté à différents bugs de natures diverses. Le premier bug rencontré a été causé par un mauvais dosage de la quantité des informations envoyées au serveur. Elles étaient trop régulières et elles provoquaient une saturation des serveurs et de la connexion. Cela se manifestait par du lag. J'ai donc dû modifier le rythme d'envoi et d'actualisation des données auprès du serveur.

Le second bug est apparu lors des changements de scènes. Le network manager se dédoublait. Le problème a été causé par une erreur dans la fonction qui faisait que le network manager était appelé deux fois, et donc il recevait à chaque fois deux ordres identiques, d'où le dédoublement. Je n'ai eu qu'à modifier son appel pour qu'il ne soit sollicité qu'une seule fois. Le dédoublement a donc logiquement disparu.

En conclusion, cette troisième partie du projet m'a permis d'approfondir mes connaissances de l'asset Unet. J'ai été content de travailler sur le mode multijoueur car ce mode offre une vision d'ensemble du jeu, mettant en relation les différents paramètres. J'ai pris conscience à travers les différents bugs de la complexité que peut représenter la connexion entre les joueurs. Malgré la simplicité de notre jeu, j'ai été étonné par les difficultés rencontrées pour permettre l'interaction des différents joueurs. Je suis vraiment admiratif des jeux célèbres en ligne qui sont capable de gérer l'interaction de milliers de joueurs. Leur fonctionnement doit être extrêmement complexe et la gestion du mode en ligne un vrai casse tête qui a dû exiger un travail colossal pour être jouable. Peut être un jour, parviendrai-je à mon tour à collaborer à la création d'un tel mode multijoueur ...

3.3 Partie de Gabriel

3.3.1 1ère partie du projet : Acquisition des bases d'Unity et développement du gameplay de base

Pour la 1ère soutenance, je me suis occupé de la partie du gameplay basique, c'est-à-dire du déplacement et de la caméra des personnages du jeu. Comme je trouvais l'utilisation d'Unity au début plutôt compliquée en raison de l'apparition de nouvelles notions pour moi telles que l'utilisation de vecteurs, Raycast, etc..., j'ai décidé de m'aider d'un tutoriel disponible sur Youtube afin de m'aider à intégrer et à utiliser ces nouvelles notions au mieux. Cela m'a pris pas mal de temps, ce qui avait rendu au début ma progression assez lente, mais m'a permis d'avoir les bases néces-

saire à la réalisation du code du gameplay du jeu, tout au long du projet.

Au début, nous voulions avoir rapidement un personnage modélisé, pour le chasseur et pour le caméléon. Cependant, nous n'avions pas encore choisi de modèle définitif. En effet, nous voulions créer notre propre modèle de personnage ce qui impliquait de savoir manipuler Blender. Or l'utilisation de Blender s'est révélée complexe et chronophage, c'est pourquoi j'ai décidé de mettre la modélisation du personnage de côté pour la 1ère soutenance et de me focaliser sur le coding du déplacement et de la caméra.

J'avais également recherché sur internet des sons potentiels pour notre jeu que nous intégrerons dans une deuxième phase de développement de notre création. Les bruitages étant une partie importante du gameplay d'un PropHunt, il nous paraissait essentiel d'avoir déjà une petite idée des sons que nous allons choisir d'implémenter. Au final, cela n'a pas été très utile, puisque parmi tout ceux que j'avais retenu, seulement 1 à été utilisé pour le bruitage des coups de feu de l'arme principale.

Le codage du déplacement du personnage a été fait assez rapidement. Cela m'avait permis de me familiariser avec les notions de Rigidbody qui permettent d'appliquer les lois de la physique sur l'objet en question, ainsi qu'avec les notions de vecteurs et d'Input qui permettent à Unity de calculer le déplacement du joueur en fonction des touches directionnelles enfoncées.

Nous avons envisagé également d'ajouter la capacité au joueur de courir, ce qui lui permettra, en appuyant sur une touche spé-

ciale de booster sa rapidité de déplacement. Grâce à l'implémentation de la fonction de déplacement, cette tâche devait être facile à réaliser puisqu'il s'agira juste d'ajouter un booléen qui vérifie si la touche spéciale en question est actionnée, et si c'est le cas, d'augmenter la variable de vitesse du joueur. Au final, j'ai fait cela pour la 2ème soutenance. Comme je l'avais prévu, ce n'était pas très compliqué, cependant je n'avais pas pris en compte que la mise en place d'une telle fonction m'aurait obligé à créer un indicateur qui montre au joueur la quantité de stamina, nécessaire au sprint du personnage, et régler quelques soucis d'équilibrage, que je développerais dans ma partie qui touche à ma participation dans le projet pour la 2ème soutenance.

De plus, il a été prévu qu'en fonction de la taille de l'objet en lequel le caméléon (c'est-à-dire le joueur dont le but est de se cacher) choisira de se transformer, nous intégrerons un changement de vitesse associé. Par exemple, il est logique que s'il se transforme en un petit objet, il ira plus vite que s'il se transforme en un grand objet. J'ai bel et bien réalisé ceci pour la 3ème soutenance.

Pour le codage de la caméra des personnages, il m'a fallu faire une arme provisoire qui permette au joueur de voir dans quelle direction son personnage regarde. Dans un deuxième temps, j'ai aussi rajouté un réticule pour être encore plus précis. De plus, j'ai dû utiliser des vecteurs afin de calculer la rotation de la caméra. J'ai cependant rencontré un problème. En effet, la caméra pouvait faire des rotations de 360 degrés, ce qui impliquait que la tête du personnage ne soit pas fixée sur le personnage lui-même ce qui était totalement irréaliste. J'ai voulu résoudre ce problème en mettant des booléens qui limiteraient la rotation de la caméra lorsqu'elle atteindrait un angle limite minimum ou maximum de rotation,

néanmoins Cela m'a pris du temps à atteindre le résultat espéré, mais j'ai depuis solutionné cette difficulté.

En effet, j'ai résolu ce bug de caméra à l'aide de recherches sur Internet et de l'aide du tutoriel que je suivais. Il m'a fallu limiter la rotation de la caméra à une valeur minimale et une valeur maximale.

La plupart des fps utilisant une vue inversée, c'est-à-dire que lorsque le joueur bouge sa souris vers le haut, alors la caméra va vers le bas, et si la souris va vers la droite, la caméra se dirige vers la gauche. Pour la 1ère soutenance, notre jeu avait un système de vue normale, mais j'avais prévu en bonus qu'il sera possible d'ajouter la possibilité de changer de système de caméra dans les options du menu du jeu, comme pour la plupart des Fps actuels, et qu'il sera tout aussi possible de choisir dans les options, la capacité à changer la sensibilité de la caméra, c'est-à-dire à changer la vitesse et la rotation de la caméra en fonction du déplacement de la souris. Plus la sensibilité sera importante plus la vitesse de rotation de la caméra sera importante, Ce que j'ai fait pour la 2ème soutenance.

J'avais ensuite ajouté la fonction de saut au déplacement du personnage. Ceci m'avait amené à faire face au problème suivant : le personnage pouvait sauter à l'infini alors même qu'il était déjà en l'air. Après avoir fait des recherches, j'ai appris qu'il était possible de retourner le "tag" de l'objet que le personnage touche, cela étant possible entre autre grâce à une fonction d'Unity, On-CollisionEnter. Je m'étais aussi familiarisé avec les tags, qui permettent de distinguer les objets entre eux. J'avais donc ajouté un tag "ground" au sol et dans la fonction de saut, un booléen qui

est égal à true si le personnage touche un objet avec le tag sol ou s'il touche le sol. La découverte des tags était un véritable plus pour notre projet puisqu'elle nous permettra de faire la distinction entre les objets, les joueurs et le reste, et ainsi de faciliter la programmation du système de tir et de transformation. L'implémentation de cette fonction a été ensuite utilisée par Étienne pour la fonction des dégâts de chute.

J'avais commencé à élaborer le système de tir et donc à assimiler la notion de Raycast. J'avais pour objectif de finir ce système de tir avec Mathieu pour la 2ème soutenance et de :

- S'occuper du système de dégâts
- S'occuper des graphismes du tir
- S'occuper de la modélisation des armes
- Prendre en compte si l'objet touché est un objet ou un joueur
- Ajouter le système de mort du joueur et le système "d'observation" pour les joueurs morts afin qu'ils puissent observer continuer à suivre leur équipe.

Étant une des parties les plus importante du gameplay du jeu, il fallait que nous étions au moins 2 pour s'en occuper, mais Mathieu a dû lui s'occuper de la mise en place du multijoueur, ce qui nous a mis en retard dans cette partie. En effet, nous n'avons toujours pas fait le système de mort du joueur et le système "d'observation" pour les joueurs morts.

L'ajout des armes allait causer un nouveau problème. En effet, le modèle de l'arme possède un RigidBody entravant le mouvement du personnage qui pourra se retrouver bloquer par des

éléments de la map. C'est pourquoi j'ai envisagé d'enlever ce RigidBody avant de me raviser puisque faire ceci impliquerait que le modèle de l'arme passe à travers les obstacles ce qui occasionnerait un nouveau problème de réalisme. De plus, en testant mon travail j'ai pu constaté qu'il était possible de voir apparaître un bug si lors d'une manipulation le joueur saute tout en provoquant une importante rotation de la caméra alors il est possible que le personnage fasse un très grand saut. Pour l'instant je ne connais pas les origines de ce bug mais il est probable que ce soit le RigidBody de l'arme. Même si la manipulation à l'origine du bug reste très dur à mettre en place, j'ai résolu ce problème en apportant quelques modifications à la fonction de saut.

Pour conclure, nous étions largement en avance par rapport à ce que nous avions annoncé pour la partie "gameplay". Néanmoins, nous étions en retard par rapport à la partie "menu d'accueil", c'est pourquoi j'avais prévu de me focaliser sur cette partie pour la 2ème soutenance en plus du système de tir. Ainsi, toutes les prévisions faites lors de la 1ère soutenance ont bel et bien été réalisées.

3.3.2 2ème partie du projet : Mise en place des interfaces, menus, et finalisation du gameplay de Base

Pour la 2ème soutenance, j'avais touché à plus de concept différents d'Unity qu'à la première : Canvas & Slider, Sons , PlayerPrefs, Scenes, Time.

J'ai commencé par la création des menus, car elle me semblait à coup d'oeil simple et aussi je pensais qu'il me fallait que Thibault et Mathieu configurent le multijoueur pour que je puisse faire le système d'arme. J'ai donc commencé à faire le menu principal, mais n'ayant pas trop d'inspiration quant au design de celui-ci, le menu actuel n'est pas très joli. Cependant j'ai préféré m'assurer de faire un menu fonctionnel pour cette soutenance plutôt qu'un menu joli mais qui ne marcherait pas, surtout que, comme dit précédemment, je n'avais pas trop d'inspiration pour son design. Finalement, le design me paraissant correct, j'ai décidé de la garder.

Après m'être familiarisé avec les boutons, images et canvas d'Unity, je suis passé à la réalisation du menu des options. Étant donné que notre jeu est un FPS (First person shooter), il fallait ajouter dans ce menu la possibilité de changer la sensibilité de la caméra. J'avais donc utilisé un slider dont la valeur est égale à cette sensibilité. Le problème était que lorsque je changeais de scène, la valeur du slider était reset à 0. Pour remédier à ce problème j'ai dû utiliser PlayerPrefs, qui stock justement les préférences du joueur, et ces dernières ne changent pas même lors d'un changement de scène. Après quelques modifications du script de la rotation de la caméra, le changement de sensibilité fut opérationnel.

Au début, lorsque le joueur utilisait sa touche "echap" il était redirigé au menu principal ; ainsi j'utilisais le menu principal comme menu in game, cependant, lorsque le joueur allait dans le menu principal et que ensuite il utilisait le bouton "play" pour revenir en jeu, le jeu était reset puisque comme il y a eu un changement de scène, la scène de jeu et donc les anciennes actions du joueur ont été supprimées.

Pour remédier à cela, j'aurais sans doute pu utiliser dans mes scripts des DontDestroyOnLoad qui empêche des objets d'être supprimés lors d'un changement de scène, mais j'ai opté pour la méthode qui me semblait la plus efficace, c'est-à-dire créer 2 nouveaux menus utilisables en jeu. Ces 2 derniers ont la particularité d'être dans la même scène que le jeu. Le problème était que lorsque je mettais le jeu en pause en accédant au menu, les scripts continuaient. J'ai donc dû faire un script qui désactivait tous les scripts du joueur lorsque le menu était actif.

Ensuite, j'ai voulu ajouter plus de fonctionnalités dans les menu d'options. J'ai ajouté la possibilité de modifier les input du joueur. Cette tâche m'a étonnamment pris pas mal de temps, je ne savais pas encore comment m'y prendre. Après plusieurs essais, j'ai finalement réussi à faire cet InputManager. Le problème était que le joueur peut s'il le souhaite attribuer une touche à plusieurs fonctionnalités : par exemple, il peut faire en sorte que la touche "A" soit non seulement la touche pour avancer mais aussi pour sauter. J'ai voulu mettre de côté la résolution de ce problème pour plus tard afin de me focaliser sur l'amélioration du gameplay et du système d'armes entre autres.

De plus, si pendant le changement des touches le joueur décidait de configurer une des touches sur "echap" qui est la touche pour fermer le menu alors sans surprise la touche est configurée mais le menu se ferme or il est souhaitable que ce ne soit pas le cas puisque le joueur ne voulait pas fermer le menu mais bien configurer une touche. Cependant est-ce que le joueur a le droit de configurer une touche sur "echap" ? C'est débattable : soit oui et donc il faudra que je modifie mes scripts ouvrant/fermant le menu, soit non et il faudra que je modifie mon Input manager en faisant que la touche "echap" soit interdite. J'ai décidé d'y réfléchir et de faire ça plus tard.

Puisque le multijoueur n'était pas encore configuré, j'ai voulu améliorer le gameplay plutôt que de finaliser le système d'arme, car je pensait que celui-ci nécessitait cette configuration pour avoir accès aux informations des joueurs, les modifier, et les ré-envoyer au serveur. Avec l'aide d'Étienne, j'ai donc ajouté une fonction de sprint, qui permet de, si le joueur a les points de "stamina" nécessaires, avancer beaucoup plus vite. L'intérêt de cette fonction dans notre PropHunt est débattable : si je l'ai fait car il est classique a chaque personnage de FPS d'avoir la capacité de sprinter, elle peut être cependant considéré comme "GameBreaker" car elle pourrait fortement avantager les chasseurs, pouvant traquer les Props plus facilement, c'est pourquoi il faudra, lors des tests quand le jeu sera fini, voir s'il faut la modifier pour la rendre plus équitable ou pas.

J'ai donc commencé la création de l'HUD du jeu par celle de la barre de cette fameuse stamina. Pour la faire, j'ai utilisé 2 bars dif-

férentes qui se superposent : la bar de stamina remplie et la bar de stamina vide. Quand le joueur sprint, la stamina diminue et donc j'utilise un type d'image "Fill" qui fait disparaître en fonction de la stamina, la bar remplie, ne cachant plus la bar non remplie. J'ai ensuite ajouté un "cooldown" à ce sprint de 5 secondes lorsque la stamina atteint la valeur 0, afin que le joueur n'ait pas la possibilité de maintenir la touche de sprint pour continuer de courir même quand la stamina est de 0, la stamina se régénérant chaque seconde.

Puisque la mise en place du multijoueur tardait trop et que je ne voulais pas être en retard sur ce que j'avais prévu de faire pour cette soutenance, j'ai décidé d'attaquer le système d'arme. Tout d'abord, j'ai créé un script Player qui m'a permis de gérer les points de vie des joueurs. Ainsi, j'ai pu implémenter les fonctions qui font subir des dégâts aux joueurs voire même le tuer s'il se fait tirer dessus. Néanmoins puisqu'il n'y a pas encore de multijoueur la fonction de mort du joueur ne fait que détruire l'objet "Player" et sera bien évidemment modifiée dans le futur. J'ai donc créé une barre de vie similaire à la barre de stamina.

Maintenant, il me fallait résoudre un nouveau problème : comment avoir accès à tous les joueurs sans serveur ? J'ai donc créé une classe "GameManager" qui me donne accès à tous les joueurs : elle cherche les GameObject possédant le tag "Player" et ajoute à un dictionnaire les ID attachés aux scripts Player de ces GameObject. Ainsi, lorsque le joueur tire et touche un personnage, son ID permet au script d'accéder au Player et de lui infliger des dégâts ou bien même le tuer.

Il fallait donc que je m'occupe des munitions de l'arme. Avec le groupe, nous avons discuté sur l'arme qu'utiliserait notre personnage et nous avons choisi l'ak47. Il fallait donc que mon script respecte les critères qu'un ak47 a généralement dans les FPS :

- Une cadence de tir assez élevée
- Un chargeur ayant une capacité moyennement élevé
- des dégâts par balle peu élevés

Même si j'ai essayé de respecter à peu près ces critères ce n'est sans doute pas encore parfait mais on adaptera facilement plus tard lors des tests. J'ai ensuite ajouté une fonction de rechargement des munitions et affiché ces dernières sur l'HUD.

Maintenant que nous avons une arme capable d'infliger des dégâts et avec des munitions, il fallait que lorsque celle-ci tire il y ait un bruit de tir. J'ai donc utilisé un des bruitages que j'avais trouvé avant la 1ère soutenance pour ce script. J'avais eu quelques problèmes avec l'audio, qui continuait à ce jouer même quand je mettait le jeu sur pause en accédant au menu. A chaque fois que le joueur tirait, le l'audio était reset, ce qui, mélangé avec la cadence de tir plutôt élevée de l'arme, provoquait un bruit affreux. J'ai donc dû ajouter un Timer de 0.5 secondes, soit le temps qu'est joué l'audio.

J'ai finalement réussi à corriger le bug de l'input Manager, qui faisait que le joueur peut configurer plusieurs fois la même touche sur plusieurs fonctionnalités. J'ai créé un dictionnaire<string,string>, qui contient les fonctionnalités comme par exemple courir et les touches qui leur sont attribuées. Si le joueur essaye de configurer une touche, le script vérifie que cette touche n'est pas déjà

attribuée quelque part et donc si elle n'est pas déjà dans le dictionnaire.

Pour conclure, mon avancée pour cette 2ème soutenance est s'est faite lentement mais sûrement, comme j'ai dû toucher à plusieurs parties d'Unity différentes j'ai eu l'impression de passer plus de temps à apprendre qu'à faire avancer notre projet, qui était pourtant déjà assez en retard, ce qui m'a donné un certain sentiment frustrant d'inefficacité même si c'est sans doute faux. Pour la 3ème soutenance j'espère pouvoir avancer plus rapidement, et donc si jamais je serais bloqué car une partie prend du retard j'essayerai de la réaliser moi même plutôt.

3.3.3 Troisième partie du projet : Mise en commun, finalisation

Pour la soutenance final, j'ai dû finaliser le gameplay pour que le jeu soit totalement jouable lors de la finalisation de la mise en place du multijoueur. Tout d'abord j'ai voulu finir les menus que j'avais fait pour la 2ème soutenance. J'ai donc rajouté l'option qui permet de changer la langue du jeu, permettant d'avoir une version anglaise ou bien française. Cette tâche n'a pas été très compliqué puisque j'avais déjà bien acquis les connaissances nécessaires à la réalisation d'options dans un menu.

J'ai ensuite ajouté la fonction des Taunts (qui se traduit littéralement par provocations), partie importante du gameplay du jeu. En effet, la plupart des Prop Hunt utilisent un système de taunt pour limiter l'efficacité du "camping", une technique populaire chez les joueurs qui consiste à se cacher à un endroit généralement improbable et y passer le reste de la partie. Les taunts sont

des bruits provenant du Prop qui se cache à un endroit trop longtemps, permettant ainsi aux Hunters d'avoir une indication sur sa position. Les taunts peuvent être aussi des images qui s'affichent en haut de la position du joueur cachés, comme par exemple des émojis .

Pour notre jeu j'ai choisi d'utiliser des bruits pour nos taunts, car je pense qu'ils sont plus adaptés à notre carte de jeu que des images et qu'ils ajoutent un côté plus amusant au jeu. Pour réaliser cette fonction il m'a fallu vérifier que le joueur ne bouge pas, c'est-à-dire que son vecteur de déplacement est nul. S'il l'est pendant 20 secondes, alors un bruit aléatoire s'active à sa position. Pour la réalisation des taunts, je m'étais demandé quel type d'audio on pouvait utiliser. Et puis, je m'étais dit, un peu en retard, que puisque notre jeu se déroulait à Epita, alors pourquoi ne pas demander à chaque élève de faire un taunt ? Néanmoins j'ai eu cette idée un peu trop tard, c'est pourquoi je n'ai malheureusement presque pas eu de taunt d'élèves. J'ai aussi rajouté des taunts qui sont des petits easter eggs pour le groupe, comme par exemple l'audio de l'essai désastreux de faire la musique du menu principal.

J'ai ensuite commencé à finaliser la mise en commun de chacune des parties touchant le gameplay. J'ai dû trier les objets présent dans le jeu en fonction de leur taille, afin que lorsque le joueur se transforme en ces objets, ses points de vie changent en fonction de la taille de l'objet. Par exemple, s'il se transforme en un petit objet, alors il aura moins de vie que s'il s'était transformé en un grand objet afin d'équilibrer le jeu, car un petit objet a plus de facilité à se cacher dans le décors par rapport à un grand ob-

jet. Puisque la fonction de transformation de Mathieu ne marchait pas parfaitement et qu'il me fallait pouvoir tester tout cela, j'ai dû créer une fonction provisoire en attendant pour pouvoir réaliser mes tests.

De plus, il fallait que lorsque le personnage se transforme en petit objet, il aille plus vite que s'il est transformé en grand objet. Avec la mise en place du multijoueur, les scripts que j'avais réalisés avant ne marchaient plus à cause de nouvelles erreurs comme par exemple à cause du fait qu'il y'a désormais plusieurs joueurs ces scripts ne faisaient pas la différence entre le joueur Local et les autres joueurs. L'utilisation du Component Unity "Network Identity" m'a été très utile pour régler ses problèmes puisqu'il permet de savoir si le joueur est bien le joueur local.

J'ai aussi fait la partie code du tutoriel, dont la réalisation a été partagée avec Ethienne.

3.4 Partie de Thibault

3.4.1 Première Etape

Aborder la création d'un site web a demandé d'abord plusieurs points techniques nouveaux et de tenir compte de beaucoup de prérequis. N'ayant aucune connaissance en ce domaine, j'ai dû commencer par suivre de nombreux cours sur Udemmy et Open-Classrooms pour pouvoir apprendre différents langages utilisés pour la création d'un site web, ainsi que pouvoir utiliser un fra-

mework. La première partie pour créer un site web est de mettre en place sa structure. Pour se faire on doit utiliser le langage HTML5 pour structurer chacune des pages du site. La mission suivante est de le mettre en forme pour changer son aspect graphique. On utilisera pour cela le langage CSS3. Il faut aussi tenir de l'importance de l'architecture du dossier et sous dossiers et fichiers du site web. crée le fichier index.html, celui qui sera lancé par défaut pour afficher le site internet. Puis la création d'un sous dossier, style, qui permettra de stocker tous les fichiers .css servant à la mise en forme des pages web. la création d'un fichier pour déposer les photos utilisées sur le site que l'on nommera images. De même pour les documents avec un dossier docs. Et pour finir un dossier js, pour mettre tous les scripts java.

Avant de me lancer directement sur la création du site web de notre projet «Hide Life», j'ai dû m'entraîner en créant plusieurs petit site web pour bien comprendre le fonctionnement et pouvoir bien utiliser mes connaissances au moment venu. J'ai créé un mini site d'agence de voyage qui a demandé des connaissances en HTML et CSS, ainsi qu'un site web où j'ai créé le jeu skane en ligne en utilisant du HTML et du JavaScript.

L'approfondissement de certains points ont été particulièrement important. Le positionnement correct des éléments sur la page. C'est à dire sur la largeur et la longueur, mais aussi si les éléments sont collés, c'est à dire que si la page défile, l'élément suivra la page, ou si l'élément est libre, c'est à dire qu'il disparaîtra si on fait défiler la page. C'est une partie importante, car certains éléments se confondaient, et d'autres ne se plaçaient pas

en endroits indiqués théoriquement. Un des problème à résoudre était notamment que les éléments perdaient chacun de leurs cotés lorsque l'on agrandissait ou rétrécissait la taille de la page sur le navigateur. La solution a été de remplacer la taille des éléments en pixels par des pourcentages en taille sur la page. J'ai ensuite appris à mieux connaître et utiliser JavaScript, il m'a servi à gérer des éléments en fonction de leur position dans le temps, pour ainsi créer des événements et même des animations. J'ai appris à utiliser JQuery qui utilise JavaScript, il permet de faire de nombreuses animations et actions lorsque par exemple un utilisateur met sa souris sur un objet particulier, ainsi que de permettre une succession d'évènements.

Mon objectif suivant était de savoir utiliser le framework Bootstrap, versions 3 et 4, qui présente le grand avantage de rendre une page internet responsive. Une page internet responsive signifie : qui peut s'adapter à la majorité des navigateurs internet et qui peut s'adapter à tous types de plateformes, mobile, tablette, ordinateur et télévision. En effet, le framework Bootstrap s'adapte aux différentes largeurs d'écran de ces appareils. Il fonctionne avec un système de grilles. Si l'écran est grand, il pourra afficher quatre éléments par ligne contre deux pour une tablette ou afficher seulement en colonnes si le support est un téléphone portable. Il possède aussi des fonctions prédéfinies qui permettent de simplifier la taille du code.

L'association de mes nouvelles connaissances en html, css, javascript, jquery avec Bootstrap m'a permis de pouvoir concevoir un site internet qui peut s'adapter à tous les navigateurs et à

toutes les plateformes. J'ai acquis diverses connaissances comme celle de créer des bordures autour d'une image, de faire apparaître des alertes ou popups sur une page, de créer des annotations, des boutons, des carousels (images ou textes défilants). Je peux désormais aussi manipuler à souhait des barres de navigation pour pouvoir avoir un point de repère sur les différents éléments de la page du site web et des barres de progressions pour montrer ou illustrer l'avancement d'une tâche. Tâche que l'on peut ensuite animer si l'on veut avec jQuery. On peut également utiliser le tag "important" pour avoir la priorité sur les propriétés de bases de bootstrap.

J'ai réalisé lors de la première soutenance le site internet sur une seule page déroulante. J'ai trouvé au début agréable de tout avoir sur la même page en le segmentant par catégorie, tout en permettant d'accéder rapidement à toutes les catégories via une barre de navigation qui nous suis en haut de la page. Mais je prévois à terme de le segmenter en différentes pages pour que celui-ci soit plus léger. J'ai encore des correctifs à opérer concernant quelques bugs mineurs que j'ai rencontré comme un élément de la barre de navigation qui passe en dessous avant de se transformer en menu déroulant et un bug de placement d'éléments qui intervient lors du changement d'appareil.

3.4.2 Deuxième Etape

Pour la 2ème partie j'avais prévu d'avancer le développement du site internet, en modifiant le graphisme et l'affichage des différentes pages ainsi que la création de la base de donnée pour conser-

ver les données des joueurs en jeux et depuis le site internet pour que les joueurs se connectent au jeux et au site avec leurs propres comptes. J'avais aussi prévu de mettre en place des sessions sur le site web et gérer les différentes connexions et inscriptions depuis le site mais aussi d'empêcher la visibilité de certaines parti du site si le joueur n'est pas connecté. J'ai acquis de nouvelle connaissance pour pouvoir mettre en place tout ce que j'avais prévu. J'ai appris le langage php et Sql pour parvenir à réaliser mes objectifs. Par la suite j'ai réalisé certains projet pour pouvoir mettre en pratique mes connaissance et pouvoir pratiquer plus aisément par la suite.

Lors de la réalisation de mes projets comme l'envoi d'un mail automatique pour contacter le support et pour inscrire, connecter et déconnecter les utilisateurs sur le site en faisant des requêtes avec la base de données j'ai été confronté à de nombreux problèmes. Une des plus longue partie a été de mettre en commun le code php, html et sql et les faire communiquer ensemble. Je n'arrivais pas à stocker les valeurs dans des variables pour les envoyer à la base de données ni pour les récupérer depuis la base de données. Les redirections vers les différentes pages pour les formulaires d'inscriptions et connexions ont été vraiment très compliqués. j'ai mis beaucoup de temps pour trouvé des solutions adapté et parfois en passant par différentes techniques. J'ai stocké les valeurs dans un tableau avec php. J'ai appris à utiliser les différentes méthodes de récupération de données avec php avec les super globales POST, GET, SESSION et SERVER.

En récoltant des données j'ai dû apprendre à les stocker correctement et en les sécurisant. Pour cela j'ai dû apprendre quelques

notions de sécurité, comme empêcher certaines failles dans le php qui auraient permis l'exécution de code involontaire par des personnes malveillantes. Le chiffage des mots de passe avec des morceaux de textes ajoutés à l'original et en utilisant plusieurs fois le chiffage SHA1 à la suite. J'ai dû vérifier que ce qui est écrit dans le formulaire correspondait bien à ce qu'il y a écrit dans les cases associées que ce soit dans le frontend (partie visible des utilisateurs) ou backend (partie serveur, invisible des utilisateurs), car cela aurait pu entraîner de nombreux problèmes notamment un email qui est en fait du texte et qui quand le serveur l'utilise pour d'autres actions peut planter, mais aussi envoyer des variables de types différentes que celles attendues. Cela peut engendrer de nombreux bugs voire des crashes.

Pour connecter un utilisateur sur le site, j'ai utilisé la super globale SESSION. Mais cela permet à l'utilisateur de rester connecté seulement pendant la période où l'utilisateur est sur la page, dès qu'il la ferme et qu'il la réouvre, il doit se reconnecter. Pour y remédier j'ai dû apprendre à créer des cookies de navigations pour garder l'utilisateur connecté s'il le souhaite au delà du temps qu'il est sur la page en conservant ses données à l'intérieur du cookie de suivis. J'ai dû apprendre à le chiffrer et à le complexifier comme pour le mot de passe afin de protéger les données des utilisateurs. Après ces différentes acquisitions j'ai créé un pop up pour identifier les utilisateurs. J'ai rencontré aussi quelques problèmes comme la redirection vers d'autres pages, ne pas pouvoir vérifier les données envoyées dans le formulaire, le pop up qui se ferme sans que ce ne soit voulu. J'ai pu les régler correctement par la suite.

J'ai appris à utiliser Ajax, il permet de charger à l'avance certaines pages et garder des pages ou seulement certaines contenues chargés. Il utilise javascript et c'est une fonctionnalité de jQuery. C'est très pratique je l'ai utilisé pour transformer le site internet et les découper en plusieurs catégories pour éviter de laisser tout sur une même page en évitant ainsi la surcharge d'information.

3.4.3 Troisième Etape

La troisième étape de mon travail se divise en 2 catégories. Premièrement la liaison entre le site web et la base de données avec le jeu. Et une seconde catégorie avec la finalisation du site internet et le rajout de fonctions intéressante pour les utilisateurs.

Pour ce qui est de la liaison entre le site et la base de données avec le jeu, il y a plusieurs parties à gérer. Dorénavant lorsque l'on ouvrira le jeu il y aura d'abord un menu pour se connecter avant de pouvoir accéder au menu principal du jeu. Pour réaliser cette partie j'ai dû me familiariser avec les scènes de Unity3D. J'ai utilisé un Nuget Csharp pour pouvoir me connecter à la base de données et pouvoir effectuer différentes requêtes. J'ai dû refaire le chiffage avec les mêmes transformations en utilisant la méthode SHA1 pour pouvoir comparer les 2 bouts de mots de passe chiffrés. J'ai décidé que pour se connecter les joueurs devront utiliser leur adresse mail et leur mots de passe car ils resteront fixes. J'ai eu quelques soucis pour changer de scène mais l'interface fonctionne parfaitement. Ensuite avec Mathieu on va s'occuper de gérer l'attribution des pseudo aux joueurs dans le jeu. On va devoir vérifier l'identité de chaque joueurs avec l'entité qui lui correspond.

Ensuite je vais finaliser les graphismes du site que j'ai du un peu baclé la dernière fois pour privilégier le fonctionnement opérationnel des différentes fonctionnalités. Je vais donc correctement mettre en forme les pages d'inscription et de connexion. Une fois le jeu terminé je pourrais mettre le site correctement à jour en mettant le contenu actuel du jeu, que ce soit pour les photos ou pour l'histoire et les objets du jeu. Je créerai ensuite une nouvelle section pour pouvoir gérer son compte depuis le site. Pouvoir changer son pseudo par exemple. Et sur la page principale un endroit où je mettrais un leaderboard avec les joueurs ayants remportés le plus de parties.

4 Bilan de la réalisation

4.1 Difficultés ou déceptions

4.1.1 du point de vue d'Etienne

Chacun d'entre nous a vécu le projet différemment en fonction des attentes qu'il en avait. Mais je crois qu'une des difficultés à laquelle nous avons tous les quatre été confronté, a été la sous-estimation du travail à fournir en apprentissage par rapport aux différents logiciels que nous avons eu à mettre en oeuvre et en quantité de travail à fournir. J'ai pour ma part mis beaucoup plus de temps que prévu à être capable de manipuler correctement Blender.

J'ai également trouvé difficile de tenir les délais tout en menant le reste de notre scolarité.

La synchronisation de nos avancées a également été problématique car nous avons peu d'occasion de passer du temps ensemble pour avancer conjointement.

Pour ma part, j'ai vécu assez durement le départ de mon copain Kylian qui devait faire partie de DevStar et qui a démissionné de l'Epita.

Le rôle de leader a été un réel enjeu pour moi qui n'ai pas réellement un tempérament "à commander les troupes" particulièrement quand je ne maîtrise pas tous les éléments.

J'ai été déçu de devoir renoncer à faire des finitions plus abouties en raison du poids final des objets. Cette carence de finition nuit un peu à la qualité des décors que j'avais l'ambition de créer.

Cela a été également une grosse déception pour moi de renoncer aux troisième et quatrième étages c'est à dire à la deuxième map en raison de la quantité de travail à fournir qui l'aurait été au détriment de la finition de la première map et à la finalisation de tous les autres points de détails.

4.1.2 du point de vue de Mathieu

Ma première approche d'un travail d'équipe sur un projet scolaire avait eu lieu l'année précédente, en terminale pour la validation de l'option sciences de l'ingénieur. Notre groupe était alors composé de quatre membres, mais très vite le travail avait été accompli seulement par deux personnes, un camarade et moi. Dès lors, j'étais sorti amer de cette expérience et je redoutais le deuxième semestre de mon année à EPITA où je savais qu'un

travail de groupe était imposé.

Durant la première partie du projet j'avais l'impression que le travail demandé était titanesque et je me demandais comment arriver à le mener à bien. Je n'avais pas les connaissances suffisantes et j'avais l'impression que peut être je ne serai pas capable de le faire.

Je me suis aussi beaucoup inquiété lorsque je commençais à travailler la deuxième partie du projet en vue de la deuxième soutenance. Je devais aborder un aspect du programme plus complexe : le mode multijoueur. D'ailleurs, j'ai rencontré des difficultés pour la réalisation de ce mode multijoueur. Souvent, j'étais découragé car je ne comprenais pas pourquoi ma fonction ne donnait pas le résultat attendu. J'abordais donc la dernière partie du travail avec inquiétude. Pour me rassurer, je commençais simplement par rajouter des fonctionnalités à mon travail précédent. Ensuite, j'ai entrepris le mode LAN et en ligne. En fait, le travail s'est avéré plus simple que je ne le pensais et cela m'a redonné confiance en moi.

4.1.3 du point de vue de Gabriel

J'ai été assez déçu de ce projet, tout d'abord car l'idée de faire un PropHunt ne m'attirait pas trop. S'il est vrai que j'ai néanmoins eu beaucoup de plaisir à réaliser ma partie, j'ai trouvé l'organisation du groupe très mauvaise. En effet, la mise en commun de chacune de nos parties a pris beaucoup trop de temps, tout comme l'utilisation d'un Git commun pour le groupe. De plus, étant chargé du codage du jeu en général, j'ai souvent eu besoin de la partie des autres membres pour pouvoir avancer de mon côté.

Le reste du groupe ayant des difficultés quant à la compréhension et utilisation de Git et le fait que souvent les autres étaient en retard sur ce qui était prévu de faire à fait que j'ai dû stopper inutilement mon avancée.

De plus, au commencement du projet, j'ai trouvé que j'ai pris beaucoup trop de temps à apprendre à utiliser Unity plutôt qu'à faire avancer le projet.

4.1.4 du point de vue de Thibault

Ce projet a été le premier vrai projet de mon côté. La création d'un site internet de A à Z entièrement en abordant une grande partie des sujets dans ce domaine, que ce soit avec les différents langages que j'ai eut à gérer ou avec le responsive et les bases de données a été une totale découverte, j'ai mis énormément de temps pour apprendre toutes ces notions et j'ai fais de nombreuses erreurs. J'aurais aimé repartir de zéro pour pouvoir utiliser tout ce que je sais maintenant exploiter, mais faute de temps ce n'est pas possible. Je n'ai pas eut énormément de retour au niveau du code de mes coéquipiers car ils ne connaissent pas tous les langages et spécificités de chacun. J'aurais aimé toucher un peu plus au code du jeu, je n'en ai pas eut beaucoup l'occasion car je voulais que le site soit bien fini et que pour ceux qui s'occupaient du code du jeu il ne me confiait pas grandes tâches à faire de ce côté. De plus j'ai été un peu brusqué par la charge de travail à faire que j'ai surrestimé après avoir rencontrés des problèmes que jamais je n'aurais soupçonné ou par le manque de connaissance qui m'a empêcher de voir au premier abord toutes les tâches à réaliser pour réaliser certaines parties.

4.2 Satisfactions

4.2.1 du point de vue d'Etienne

J'ai le sentiment du devoir accompli puisque nous rendons un projet de jeu ludique, fonctionnel et à partager entre amis. J'ai eu le plaisir de progresser dans de nombreux apprentissages de logiciels, de codage, de bruitage, d'animation et de fonctionnalités. Notre groupe est resté soudé face aux difficultés et chacun a, pour le coup, joué le jeu en donnant le meilleur de lui-même. Je suis content d'avoir réussi à apaiser certaines tensions liées à la pression et à la fatigue. Il y a eu au final un réel plaisir partagé par les membres de DevStar lors de l'accomplissement de cette mission. J'ai en partie vaincu ma timidité pour manager mon équipe et faire en sorte que le cahier des charges soit au mieux respecté. J'ai adoré modéliser, que ce soit la map ou les éléments de décor et tout particulièrement le caméléon qui était mon petit clin d'oeil personnel. J'ai réussi à être organisé et vigilant aux détails pour mener à bien ce projet qui me tenait réellement à coeur. Même si le niveau de finition n'est pas tout à fait celui que je visais, je suis extrêmement satisfait du rendu général de Hide Life et du graphisme qui passe bien pour ce type de jeu avec un aspect type Team Fortress. J'ai réussi à jouer mon rôle de chef de projet en récupérant à temps tous les éléments, en réajustant au fur et à mesure les tâches de chacun et en faisant la synthèse du projet. J'ai également pu voir de la fierté dans les yeux de mes parents quand je leur ai présenté notre travail (même si j'ai bien conscience qu'ils ne sont peut-être pas totalement objectifs!) J'ai été fier de recevoir des encouragements de notre jury pour notre deuxième soutenance et de voir qu'il nous croyait capable d'aboutir dans de

bonnes conditions.

4.2.2 du point de vue de Gabriel

J'ai eu beaucoup de plaisir quant à la réalisation de ce jeu. Tout d'abord, car j'ai toujours eu envie de faire un jeu, mais ne sachant pas par où commencer je ne l'ai jamais fait. Cela a donc été pour moi une expérience très enrichissante, à un tel point que je vais sûrement refaire mon propre jeu pendant les grandes vacances. Étant une personne assez timide qui n'ose pas trop s'imposer en général, j'ai eu la possibilité de vaincre cette timidité lors du projet en confrontant mon avis à celui des autres membres, et en essayant de manager ceux-ci quant à la mise en commun des différentes parties qui composent le jeu et le git du projet. Même si j'ai été déçu par le groupe, avoir un projet en commun avec un groupe est une expérience que j'ai trouvée unique tout d'abord sur le plan humain, mais aussi technique,

4.2.3 du point de vue de Thibault

Je suis content d'avoir soumis l'idée du prop'hunt qui a été soutenu par la majorité, j'ai trouvé ce projet très intéressant et ça m'a apporté beaucoup de nouvelles connaissances. Ça m'a donné envie de réaliser de nouveaux projets, seul ou en groupe. L'entente dans le groupe était assez bonne, chacun travaillait bien et malgré quelques retards, il n'y a pas eu d'embrouilles entre nous. Nous sommes un groupe assez soudé et qui sait gérer ses périodes

de crises correctement. Je suis très content d'avoir participé à ce projet et d'avoir collaboré avec mon groupe.

5 Bibliographie

- **Unity** : <https://unity3d.com/fr>
- **Git** : <https://github.com/HideLifeFr>
- **Bruitages** : <http://www.universal-soundbank.com/bruitages-humains.html>
- **Tuto Unity** : https://www.youtube.com/watch?v=AL1Ti31ulq4&list=PLUxWDlz8PYJ7FxyQtdiLgLhRBBI6fr_-
- **Cours unity openclassroom** : <https://openclassrooms.com/fr/courses/1946836-realisez-votre-premier-jeu-video-avec-unity>
- **cours créer un site web en html5 et css3** : <https://openclassrooms.com/fr/courses/1603881-apprenez-a-creer-un-site-web-en-html5-et-css3>
- **Apprendre à créer des sites professionnels avec HTML5 et CSS** : <https://www.udemy.com/apprendre-a-creer-des-sites-professionnels-avec-html5-learn/v4/overview>
- **Formation Complète Développeur Web** : <https://www.udemy.com/formation-developpeur-web/learn/v4/overview>
- **https ://jquery.com/ (site de jQuery)** : [https://getbootstrap.com/docs/4.0/getting-started/introduction/\(frameworkbootstrap4documentation\)](https://getbootstrap.com/docs/4.0/getting-started/introduction/(frameworkbootstrap4documentation))