

# Zéphyr

## Rapport de projet

Juin 2019

### Equipe Zelstrom :

Moustapha DIOP

Philippe HERNANDEZ - ALARCIA

Gabriel LION

Guillaume THOMAS



# Table des matières

<b>Introduction</b>	<b>4</b>
<b>1 Retour sur le cahier des charges</b>	<b>5</b>
1.1 Les prémices du jeu et son genre . . . . .	5
1.2 Les spécifications issues du cahier des charges et la distribution des tâches . . . . .	7
1.3 Les points revus du cahier des charges . . . . .	13
1.3.1 La répartition et le planning . . . . .	13
1.3.2 Les changements dans le projet . . . . .	15
<b>2 Le projet</b>	<b>16</b>
2.1 Le scénario . . . . .	16
2.2 Mécaniques de jeu . . . . .	18
2.2.1 Les mécaniques du joueur . . . . .	18
2.2.2 Les mécaniques de l'ennemi . . . . .	20
2.2.3 La mécanique des armes . . . . .	22
2.2.4 Mécanique générale . . . . .	22
2.3 Interface . . . . .	24
2.3.1 Menu principal . . . . .	24
2.3.2 Menu de pause . . . . .	25
2.3.3 En jeu . . . . .	25
2.4 Multijoueur . . . . .	28
2.5 Implémentation des ennemies et des alliés . . . . .	32
2.5.1 Les ennemis . . . . .	32
2.5.2 Les personnages alliés . . . . .	35
2.5.3 Les armes des alliés . . . . .	36
2.6 Les cartes . . . . .	38
2.7 Effets sonores . . . . .	40
2.8 Site internet . . . . .	41

<b>3</b>	<b>Ressenti de l'équipe sur la réalisation du projet</b>	<b>44</b>
3.1	Moustapha . . . . .	44
3.2	Philippe . . . . .	47
3.3	Gabriel . . . . .	49
3.4	Guillaume . . . . .	51
	<b>Conclusion</b>	<b>53</b>
	<b>Annexes</b>	<b>55</b>

# Introduction

Zéphyr est un action-RPG misant sur le dynamisme afin d'apporter au joueur une nouvelle expérience de jeu. Ce dernier se rapproche des roguelites dans la manière de rendre la mort nécessaire afin de pouvoir être préparé au mieux contre les ennemis. L'accumulation d'expériences du joueur est indispensable, car il doit apprendre de ses erreurs pour ne plus les refaire et donc progresser. Les roguelites s'articulent principalement autour d'un jeu : Rogue. Ce dernier est sorti en 1980, développé par Michael Toy, Glenn Wichman et Ken Arnold.

Si le jeu a vu le jour, c'est avant tout parce que Epita attendait de nous la réalisation d'un projet afin de valider notre année. Et même si Zéphyr est à la base un projet scolaire, il a une importance bien plus significative pour nous qu'une simple note. Il s'agit du condensé de six mois de travail (ou presque ...) pendant lesquels nous avons vu le projet grandir et se métamorphoser au gré de nos envies, c'est même devenu une source de fierté. C'est pourquoi nous nous sommes beaucoup donnés dans le cadre du projet et de sa réalisation. Ainsi, ce document présente les divers aspects d'un jeu et de son histoire.

# Chapitre 1

## Retour sur le cahier des charges

### 1.1 Les prémices du jeu et son genre

Le cahier des charges remis en janvier a été la première étape vers l'élaboration du projet. En effet, même si le groupe était déjà constitué depuis décembre, il n'en restait pas moins que l'idée autour du projet ne florissait pas. Nous n'avions pas d'idée réelle mise à part le fait que nous voulions tous procéder au développement d'un jeu. C'est vraiment lors de la rédaction de ce cahier des charges que nous avons commencé à nous poser des questions sur ce qu'il fallait faire. Ainsi, l'idée de s'orienter vers un jeu à inspiration « Rogue-Lite » a commencé à émerger. Ce type de jeu nous semblait adéquat comme base de projet. En effet, la mort permanente qui justifie le non besoin de mettre en place un système de sauvegarde de progression, et les graphismes minimalistes que l'on retrouve beaucoup dans le genre était un avantage. D'autre part, certains challenges proposés nous semblaient bien plus intéressants. En particulier le fait d'avoir une carte qui se génère aléatoirement à chaque fois que l'on recommence une partie nous semblait complexe mais loin d'être infaisable. Les autres points nous semblant intéressants relevaient principalement du « game design », c'est-à-dire comment parvenir à rendre notre jeu intéressant. Et en ce qui concerne notre équipe, ce genre était très apprécié.

Si cette idée de créer un tel jeu a émergé, c'est avant tout grâce à l'intérêt commun de toute l'équipe pour Enter the Gungeon. Il s'agit d'un Rogue-lite, développé par Dodge Roll dans lequel quatre aventuriers descendent dans le Gungeon afin de trouver le trésor ultime du légendaire Gungeon : l'arme qui peut tuer le passé. Son dynamisme, son design et sa difficulté en font un jeu très agréable à jouer. Ce jeu est donc la principale source d'inspiration de

Zelstrom.

Pour revenir sur le terme de Rogue-Lite, il faut savoir qu'en général il est employé pour parler d'un jeu dans lequel les niveaux sont générés aléatoirement, et où la mort est permanente, c'est-à-dire qu'une fois mort, il faut recommencer, et la progression dans le niveau est perdue.

Les Rogue-lite se différencient par le rapport à la mort que va entretenir le joueur. Ces jeux sont connus pour la plupart comme étant difficiles pour tout joueur débutant, c'est en mourant que ce dernier apprendra de ses erreurs et surmontera par la suite les difficultés qu'il a rencontrées, d'autant plus que dans le but de progresser, la mort sera souvent nécessaire. En effet, pour débloquent de nouvelles composantes de gameplay dans certains jeux, il est nécessaire d'arriver sur l'écran de game over. Par exemple, dans The Binding of Isaac : Rebirth, jeu développé par Nicalis, une fois un objectif particulier réalisé, c'est à la mort que l'on débloquent un nouvel équipement. Ainsi, même si après avoir perdu, il faut recommencer, la mort n'est qu'une étape dans la progression globale du jeu (que ce soit dans la progression personnelle, compréhension du jeu, ou dans l'obtention de nouveaux éléments de jeu).

Zéphyr est un jeu mêlant action et exploration, dans lequel les niveaux se succèdent avec une difficulté croissante, et viennent corser la progression du joueur qui n'a d'autres choix que d'apprendre de ses erreurs. L'esprit du Rogue-lite détaillé ci-dessus est donc au cœur du jeu.

## 1.2 Les spécifications issues du cahier des charges et la distribution des tâches

Lors de la création de Zéphyr, afin de pouvoir au mieux s'organiser, un premier tableau représentant la répartition des tâches au sein de l'équipe avait été réalisé. Sur ce dernier, la lettre « R » correspondait au responsable de la tâche, et la lettre « S » au suppléant de cette dernière.

Tâches	Personnes			
	D. Moustapha	H. Philippe	L. Gabriel	T. Guillaume
Mécanique de Gameplay	S	-	R	-
Interface	-	R	-	S
Multijoueur	-	-	S	R
Ennemis	R	S	-	-
Graphisme des personnages	R	S	-	-
Graphisme des cartes	-	-	R	S
Effets sonores	-	R	S	-
Site internet	S	-	-	R

Ce dernier était également accompagné d'un tableau représentant l'avancement des tâches souhaité à différent stade du développement. Cela encore dans le but de gagner en ponctualité et de se fixer des objectifs.

Tâches	Soutenance 1	Soutenance 2	Soutenance 3
Mécanique de Gameplay	70%	95%	100%
Interface	25%	50%	100%
Multijoueur	70%	90%	100%
Intelligence Artificielle	60%	80%	100%
Graphisme des personnages	30%	60%	100%
Graphisme des cartes	40%	75%	100%
Effets sonores	20%	50%	100%
Site internet	50%	85%	100%

Le contenu de chacune de ces parties avait par la suite était explicité. Voici un extrait issu du cahier des charges montrant les spécificités attendues du projet.

## Mécaniques de gameplay

Le gameplay est l'ensemble des possibilités qu'aura le joueur pour interagir avec le personnage et son environnement. Cette partie sera donc logiquement décomposée en 2 sous-parties : les interactions avec le personnage et celles avec l'environnement.

On aura la possibilité de déplacer le personnage, d'esquiver, d'utiliser sa compétence spéciale et de tirer avec son arme. Le joueur aura une arme et une compétence spéciale par défaut en fonction du personnage choisi au début de la partie. Effectivement, le joueur aura le choix entre plusieurs personnages qui auront tous un gameplay différent. La création de différents personnages et de leur action ne devrait pas être très compliquée et passera par des scripts implémentés directement dans Unity. La compétence spéciale d'un personnage est une compétence assez forte, on pourrait imaginer que celle-ci permette de ralentir le temps, ou de faire des dégâts à tous les ennemis simultanément. Celle-ci peut être utilisée de manière ponctuelle et se charge avec le temps. Le joueur aura la possibilité de trouver de nouvelles armes au cours des différents niveaux, notamment dans les coffres ou en tuant certains ennemis. Sa compétence spéciale de son côté augmentera en efficacité par palier au fur et à mesure que le joueur tuera des ennemis.

Concernant les interactions avec l'environnement, il faudra gérer les collisions entre les balles, les ennemis et le personnage. L'ouverture des coffres avec la récupération d'un butin, le passage d'une salle à une autre et la gestion des mouvements de la caméra entrent également dans cette catégorie. Toutes ces interactions ne devraient pas être très difficiles à implémenter non plus. On peut donc directement les implémenter dans les scripts de Unity. Le joueur devra choisir entre toutes ces possibilités afin d'avancer dans le donjon. Le donjon sera divisé en niveaux, eux-même divisés en salles. Chaque niveau a une ambiance (son et graphisme) qui lui est propre. A travers les salles, le joueur peut rencontrer des pièges, des ennemis ou des coffres. Le joueur devra avancer coûte que coûte, jusqu'à atteindre la dernière salle de chaque niveau dans laquelle se trouve un boss. Chaque boss aura des mécaniques différentes. L'aspect un peu plus technique des ennemis et des boss sera développé plus bas, de même pour la structure des salles et les différents objets à créer.



## Interface

L'interface est l'ensemble des informations qui seront présentées au joueur superposées au jeu. Dans cette catégorie entrent notamment :

- La page d'accueil
- L'écran de pause
- Les pages de chargement
- Les réglages
- HUD <sup>1</sup>

La page d'accueil est la première image que le joueur aura de notre jeu. Celle-ci permettra de lancer la partie, de suivre le tutoriel et d'accéder aux paramètres.

Lorsque le joueur appuie sur "échap", le jeu se met en pause et un écran de pause apparaît. Celui-ci permettra d'accéder à la page des options, de quitter la partie ou de revenir à la partie.

Les pages de chargement afficheront une petite animation lors du changement de niveau, afin de faire patienter le joueur. Celle-ci ne demandera pas beaucoup de temps à créer.

Les paramètres permettront la modification de divers aspects du jeu. Ainsi plusieurs langues (anglais, français) seront disponibles, on pourra mettre le mode plein écran, ou encore régler le son.

Le HUD rassemble l'ensemble des informations affichées en périphérie du centre de l'écran et renseignent le joueur sur son personnage ou son environnement. On peut notamment penser à la minimap, à la santé du personnage ou à l'arme utilisée.

## Multijoueur

Le joueur aura au lancement du jeu, le choix du mode auquel il veut jouer. Zéphyr comportera deux modes de jeu : le mode solo et le mode multijoueur. Le gameplay du mode solo a été développé dans la partie précédente. Celui du mode multijoueur sera assez similaire, on aura le choix entre le même ensemble de personnages, les ennemis et les armes seront également globalement les mêmes. Les deux modes se joueront sur la même carte, mais avec une difficulté adaptée au nombre de joueurs évidemment. Plus il y aura de joueurs, plus il y aura d'ennemis et plus ceux-ci seront durs à tuer, mais il y aura également plus d'armes.

---

1. Head-up Display : affichage tête haute

D'un point de vue technique, le multijoueur sera en local (et non en ligne). Le serveur hôte sera l'ordinateur ayant créé la partie. Les autres joueurs pourront alors s'y connecter. Le framework Photon Unity Network sera probablement utilisé dans ce but. Le serveur hôte s'occupera de synchroniser les différents clients. Pour cela, il récupère dans un premier temps les changements relatifs à chaque joueur. Dans un second temps, le serveur calcule les actions des ennemis. Enfin, il vérifie si toutes les interactions sont possibles et renvoie toutes les informations à chaque client.

Les changements relatifs à chaque joueur regroupent différents attributs :

- La position du personnage
- L'orientation du personnage
- Le nombre de points de vie restants
- La dernière action exécutée (roulade, ou action spéciale)

## Ennemis

Les ennemis de la même manière que le joueur pourront se déplacer et tirer. Chacun aura toutefois un comportement propre, c'est-à-dire une manière différente de suivre le joueur, de viser ainsi que des armes différentes, un nombre de points de vie différent et même une vitesse de déplacement différente.

Les ennemis peuvent se décomposer en deux catégories : les ennemis uniques et les ennemis multiples. La première catégorie regroupe l'ensemble des ennemis qui n'apparaîtront qu'une fois, notamment les boss. Tandis que la deuxième catégorie regroupe l'ensemble des ennemis qui se retrouveront au fil des salles précédentes la salle finale. Les ennemis uniques demanderont évidemment plus de travail au niveau design ou mécanique. On peut notamment imaginer que ceux-ci aient de nombreuses attaques différentes, comme par exemple tirer dans toutes les directions, avoir des balles à tête chercheuse ou encore pouvoir sauter et aggriper le joueur. En effet, ceux-ci ont pour but d'être bien plus marquants, comme ils représentent la fin du niveau et sont censés être d'une difficulté supérieure.

D'un point de vue technique, tandis que la vitesse, les points de vie et les armes des ennemis sont plutôt simples à implémenter dans les scripts, le déplacement du joueur devra passer par une intelligence artificielle un peu plus poussée. Navmesh est actuellement la solution qui paraît la plus adaptée pour le déplacement.

## Design

La partie esthétique du jeu n'est pas une des parties les plus intéressantes à faire dans le cadre d'une formation d'ingénieur, de plus celle-ci peut devenir assez chronophage si on n'y prend pas garde. Ainsi, elle ne sera pas traitée en priorité. Mais elle ne sera pas non plus négligée car, celle-ci est importante au bon rendu final du jeu. Cette partie regroupe trois sous-parties :

- Graphisme des personnages

Le graphisme des personnages décrit la tâche de création de l'apparence des différents personnages et des ennemis, mais également de leur animation. L'apparence des personnages sera créée à l'aide de Blender, alors que leur animation se fera grâce au logiciel Mixamo.

- Graphisme des cartes

Le graphisme des cartes consiste en la création des structures des salles, des différents objets qui vont remplir celles-ci et de leur design. Les structures des salles seront faites directement sur Unity. De leur côté, les différents objets (Coffres, portes, piliers ou éléments décoratifs quelconques) seront créés avec des scripts également sur Unity, tandis que leur design sera fait sur Blender.

- Design sonore

Le design sonore décrit tout ce qui se rapporte au son dans le jeu, notamment les effets sonores et la musique. Les effets sonores regroupent les bruits émis par les coffres lors de leur ouverture, lors du passage à une autre salle, lorsqu'un ennemi est tué, ou lorsque l'on tire. La musique de son côté permettra de mettre un peu d'ambiance de fond. Et comme précisé plus haut, celle-ci devra être changée entre chaque niveau. Le logiciel utilisé sera probablement Audacity.

## Site internet

Le site internet permettra d'avoir un point de vue global du jeu et de son contenu. On pourra notamment y trouver :

- Un trailer vidéo
- Des résumés par semaine de l'avancement du jeu
- Une présentation de l'équipe et de la répartition des tâches
- Tous les documents relatifs au jeu
- Le jeu disponible en téléchargement

Le trailer vidéo présentera le scénario de Zéphyr ainsi que plusieurs scènes de gameplay afin de mettre en avant le jeu. Les scènes de gameplay seront enregistrées grâce à OBS studio. Mais le logiciel de montage n'a pas encore été choisi.

Le résumé par semaine décrira les tâches faites chaque semaine, les difficultés rencontrées et des solutions trouvées. Il permettra à ceux souhaitant télécharger le jeu de voir l'ensemble du travail accompli et peut-être même aider les sups des années suivantes. De plus, cela nous obligera à rester réguliers et respecter les dates limites.

La rédaction d'une courte page qui présente les membres de l'équipe est également prévue. Cette page contiendra les compétences et les rôles de chacun dans le projet, ainsi qu'une courte biographie. Celle-ci a permis une juste répartition des tâches en fonction des compétences de chacun. De plus cela pourra combler l'envie des curieux souhaitant connaître les développeurs de ZELSTROM.

Une page sera réservée au téléchargement de tout ce qui est relatif au jeu. Ce document ainsi que les rapports de soutenance y figureront, de même que le jeu.

Concernant l'aspect un peu plus technique du site web, le nom de domaine ZELSTROM a déjà été acheté sur namecheap.me. En ce qui concerne le serveur pour l'héberger, GitHub Pages sera utilisé.

## 1.3 Les points revus du cahier des charges

### 1.3.1 La répartition et le planning

C'est lors de la première soutenance que nous avons eu à faire au premier changement par rapport au cahier des charges, en effet, la répartition des tâches donnée dans le premier cahier des charges ne correspondait pas à la réalité. En effet, Moustapha s'est beaucoup concentré sur la partie des mécaniques de jeu, au point d'en devenir responsable. Gabriel lui s'est intéressé à différentes parties au point d'en devenir suppléant. Finalement cela a mené à une répartition des tâches revisitée.

Tableau représentant la distribution des tâches finale

Tâches	Personnes			
	D. Moustapha	H. Philippe	L. Gabriel	T. Guillaume
Mécanique de Gameplay	R	-	S	-
Interface	-	R	S	S
Multijoueur	-	-	S	R
Ennemis	S	S	R	-
Graphisme des personnages	R	S	-	-
Graphisme des cartes	-	-	R	S
Effets sonores	-	R	S	-
Site internet	S	-	-	R

Au cours des différentes soutenances nous avons dû faire face à quelques changements dans le planning. Nous avons porté notre intérêt sur de nombreux domaines, et certains étaient plus en avance que prévu, ou inversement. C'est lors de la deuxième soutenance que nous sommes arrivés à un tableau montrant l'avancement réel de chaque partie.

Tableau représentant l'avancement réel du projet

Tâches	Soutenance 1	Soutenance 2	Soutenance 3
Mécanique de Gameplay	70%	95%	100%
Interface	40%	70%	100%
Multijoueur	70%	80%	100%
Ennemis	40%	80%	100%
Graphisme des personnages	30%	80%	100%
Graphisme des cartes	30%	75%	100%
Effets sonores	25%	50%	100%
Site internet	30%	90%	100%

### 1.3.2 Les changements dans le projet

Suite à quelques problèmes d'animations nous n'avons pas réussi à mettre en place une roulade permettant aux personnages alliés d'esquiver les tirs ennemis. Cependant nous ne voulions pas en rester là et c'est que nous avons décidé de mettre en place un « Dash » qui tient le même rôle que la roulade.

# Chapitre 2

## Le projet

### 2.1 Le scénario

Avant d'entrer dans les détails techniques du projet, abordons le scénario de Zéphyr. Ce dernier a été inventé grâce aux idées de l'ensemble du groupe. Et c'est peu à peu qu'est née l'histoire racontée ci-dessous.

John, le héros principal, a été bercé par les mythes durant toute son enfance ayant eu pour conséquence de l'attirer vers les jeux vidéos. En effet, il est devenu critique dans ce domaine. Mais aujourd'hui n'est pas un jour comme les autres, il est convié par la puissante compagnie ZELSTROM à tester un tout nouveau jeu révolutionnaire.

John se rend alors dans un incroyable bâtiment qui sert de quartier général à la société ZELSTROM. Il rencontre ainsi les patrons qui lui parlent du projet. Celui-ci se base sur les souvenirs du joueur pour constituer une expérience unique. Il s'équipe alors d'un casque massif branché à une multitude de fils. Mais il découvre un jeu banal contenant de vagues souvenirs de son enfance. John est plutôt déçu, il s'attendait à clairement mieux. À la fin des tests, il quitte les lieux et rentre chez lui. Tout en écoutant la télévision, il commence à écrire son article. Peu après avoir commencé, un surprenant flash spécial apparaît. Un mystérieux astéroïde vient de s'écraser à quelques kilomètres de chez lui. Rêveur, il se rappelle des légendes que lui racontait son père sur les astéroïdes, un de ses sujets préférés. Un mythe lui revient, celui d'un astéroïde mythique renfermant un pouvoir incommensurable en son coeur et qui, un jour, devait s'écraser sur Terre : Zéphyr. Peut-être s'agit-il de celui-là. Curieux, il décide donc d'aller sur place. Les lieux sont déserts, seul l'énorme astéroïde lui fait face. Il décide de se rapprocher. Mais en descendant dans le



cratère, il glisse et rentre à l'intérieur de cette énorme pierre. Une puissante voix prononce alors ces mots :

– Je suis Zéphyr l'astéroïde légendaire, pour obtenir le pouvoir divin que je détiens, tu devras traverser une série d'épreuves et te montrer vaillant. Personne n'y est jamais arrivé, mais je ressens en toi une détermination particulière.

John se lance ainsi dans cette quête. Il traverse un gigantesque donjon, affrontant tous les monstres sur son passage jusqu'à arriver au boss final. Une fois celui-ci battu, il récupère le trésor ultime. Au moment de s'en emparer, une lumière de plus en plus forte l'éblouit. C'est la lumière de la victoire ! Il voit alors le visage des patrons de ZELSTROM lui disant :

– Avez-vous aimé ?

## 2.2 Mécaniques de jeu

Pour rendre actif les personnages nous avons, dans le cahier des charges, décider de travailler sur quatre parties : les mécaniques du joueur, de l'ennemi, des armes et les mécaniques générales.

Toutes ces mécaniques de jeu ont été respectées et implémentées, hormis celle de la roulade des alliés qui a été remplacée par un « Dash ».

Des nombreux tutoriels YouTube faits par Unity ou Brackeys ainsi que des vidéos et des documentations provenant du site de Unity ont été utiles pour réaliser certains scripts et comprendre le fonctionnement même du logiciel Unity. Cependant au fur et à mesure de l'avancée du jeu, ces sources documentaires ont été de moins en moins sollicitées.

### 2.2.1 Les mécaniques du joueur

#### Ce qui a été fait par Moustapha

##### Concernant les déplacements :

Le personnage allié se déplace dans l'espace : il peut d'une part avancer, reculer, aller à droite et à gauche à l'aide des flèches ou des touches « W, S, A, D » et d'autre part, il peut pivoter sur lui même grâce au curseur de la souris qu'il suit du regard. Le script qui permet au personnage de pivoter est assez complexe. Un rayon invisible part de la souris et lorsque ce rayon entre en contact avec le sol il fait pivoter le personnage afin que celui ci regarde dans la direction de la souris. C'est un des premier script que nous avons créé, il nous a demandé beaucoup de réflexion, car il fallait une très bonne connaissance de Unity pour pouvoir connaître les différentes fonctions déjà implémentées et les utiliser correctement afin de rendre le programme performant.

Pour complexifier ces déplacement nous avons implémenté un « Dash » qui est l'équivalent d'une roulade, et qui peut être activée en appuyant sur la barre « d'espace ». Cette nouvelle caractéristique permet de se déplacer plus rapidement en un laps de temps très court et de parcourir de plus grandes distances. Durant ce temps imparti le personnage ne prend pas de dégâts. Il peut donc se sortir de situations complexes facilement. Cependant il devra utiliser le « Dash » avec parcimonie, car il met du temps à se recharger. L'ajout de cette compétence a posé de nombreux problèmes.

Tout d'abord il a fallu régler correctement la vitesse de ce déplacement afin que cela ne donne pas l'effet d'une téléportation.

Deuxièmement nous avons du trouver le moyen d'empêcher que le personnage ne perde pas de points de vie pendant le « Dash ». Pour cela dans le script, qui permet de gérer sa vie, nous avons créé une fonction qui a l'effet d'un « interrupteur » c'est à dire qu'elle rend possible ou non la prise de dégâts. Mais le problème n'était pas entièrement réglé. Le « Dash » faisait que le personnage passait à travers les murs. Il a fallu déboguer. Pour cela nous avons du revoir entièrement le déplacement sans interférer avec la façon de jouer. Ainsi dans le component « Rigidbody » nous avons mis le « Collision Detection » en mode « Continuous Dynamique » et dans le script de déplacement nous avons utilisé la méthode `rigidbody.AddForce(Vector3(x, 0, z))` pour le mettre en mouvement. Cela nous a permis de corriger de nombreux bogues et donc de rendre le jeu beaucoup plus sûr.

#### Concernant les compétences spéciales :

Chaque personnage en a une différente qui pourra être activée en appuyant sur la touche « A » ou en faisant un clic droit sur la souris. Les quatre compétences différentes se rechargent avec le temps. Le personnage devient plus fort et peut se sauver de situations assez ardues ou peut régler un différent avec un ennemi.

- La première compétence s'appelle « Plein Soin ». Elle restaure la vie du personnage. Le script n'a pas été très dur à implémenter puisqu'il interagit simplement avec le script de vie du personnage. Il fait appel à la même fonction qui permet à un ennemi d'infliger des dégâts au personnage mais au lieu que ces dégâts soient positifs et donc qu'ils réduisent sa vie, ces dégâts sont négatifs et ils ont conséquence une augmentation de sa vie.
- La deuxième compétence se prénomme « Invincible ». Elle rend le personnage invincible pendant deux secondes. Il a fallu aussi interagir avec le script de vie et rendre la prise de dégât impossible pendant ces deux secondes grâce à la fonction « interrupteur », expliquée plus haut.
- La troisième compétences s'intitule « Le Mur ». Le personnage crée un mur devant lui qui lui permet de se mettre à l'abri. Un seul mur peut être placé sur la carte, donc quand le joueur veut en placer un deuxième le premier est automatiquement détruit. Pour réaliser cela nous avons élaboré un programme qui crée un cube et qui modifie sa taille ainsi que sa forme pour qu'il puisse toujours avoir les dimensions

du personnage.

- La quatrième compétence se nomme « Onde de chocs ». Le personnage crée autour de lui une petite onde circulaire et inflige des dégâts aux ennemis qui sont dans cette zone. Pour mettre en place cette compétence un système de particules a été ajouté au joueur. Quand il active cette compétence des particules s'échappent du personnage et si leur « collider » entre en contact avec un ennemi la particule lui inflige des dégâts.

#### Concernant les armes et la vie des alliés :

Le personnage peut avoir trois armes différentes avec lui. Il peut en changer en utilisant la molette de la souris. L'arme choisie devient active et les autres sont désactivées. La réalisation de ce script s'est bien déroulée et la réalisation a été conforme à nos attentes. Le joueur peut apprécier grandement cette fonction qui rend le jeu plus interactif et plus dynamique.

On a attaché au personnage un script lui permettant de gérer sa vie. En effet quand un personnage subit des dégâts c'est ce script qui est appelé et qui lui retire de la vie jusqu'à ce que ce dernier n'en ait plus et meurt.

## **2.2.2 Les mécaniques de l'ennemi**

### **Ce qui a été fait par Moustapha**

Les ennemis ont des mécaniques assez simples.

#### Concernant les déplacements :

Les déplacements des ennemis ont été un peu complexes à réaliser et ils ont évolué tout au long de l'avancement du projet. Tous les déplacements sont régis par une sorte d'intelligence artificielle qui délimite leur territoire d'action. Cela a été possible grâce à l'utilisation du composant « Nav Mesh Agent » qui permet à l'ennemi de suivre un allié. De plus le script donne la possibilité à l'ennemi de trouver la cible la plus proche. La réalisation de ce script est passée par plusieurs étapes.

Au début du projet, quand un allié s'approchait trop près d'un l'ennemi, il le suivait automatiquement jusqu'à ce que l'un des deux protagonistes meurt. Cependant s'il y avait plusieurs personnages alliés ce script n'en reconnaissait

qu'un seul. Il ne nous satisfaisait donc pas et par ailleurs à certain moment le jeu bogait.

Par la suite nous avons rendu l'ennemi beaucoup plus intelligent. En effet nous avons entièrement refait son programme de déplacement. Dans un rayon d'action que nous définissons il cherche les cibles potentielles et se dirige vers celle qui est la plus proche de lui pour l'attaquer. Quand sa cible est détruite il se dirige automatiquement vers une autre et s'il n'a pas de cible il reste sur place.

Cependant, même avec ces améliorations, un dernier point ne nous convenait pas. En effet les ennemis avaient du mal à pivoter sur eux même et cela impactait négativement le jeu et le rendait moins immersif. Pour corriger cela nous avons ajouté une nouvelle fonction au script de déplacement afin de faire pivoter l'ennemi en direction de sa cible en temps réel. Cela lui permet donc de changer de direction plus rapidement et d'éviter d'être dos à sa cible. Nous avons également corrigé un bogue : il n'est dorénavant plus possible de se trouver dos à un ennemi sans qu'il se retourne et ne nous attaque.

#### Concernant les tirs et la vie des ennemis :

Nous avons créé des scripts permettant aux ennemis de tirer. Leur façon de tirer est assez commune à tous. Nous leur ajoutons des systèmes de particules. Quand ils veulent tirer, des particules s'échappent et lorsqu'elles rencontrent un joueur elles lui infligent des dégâts. Chaque ennemi a un script général qui lui permet de gérer son ou ses tirs et un autre qui détecte si la particule tirée a rencontré un joueur pour lui infliger des dommages. Toutes les particules des ennemis sont rouges, elles se distinguent de celles des alliés qui sont en bleu.

Par ailleurs nous avons travaillé sur la puissance des ennemis en ajustant leur vitesse, leur puissance de tir et la cadence de tir pour les adapter aux caractéristiques des alliés afin que le jeu ne soit ni trop dur ni trop simple.

Comme pour les personnages alliés, les ennemis ont un script qui gère leur vie qui et leur est associé. Il fonctionne de la même façon que celui des alliés. Cependant lorsqu'il meurt, l'ennemi s'arrête de se déplacer et est détruit au bout de deux secondes.

### 2.2.3 La mécanique des armes

#### Ce qui a été fait par Moustapha

Elles qui permettent aux personnages de tirer, elles ont un rôle principal dans le jeu, et ont nécessité des scripts adaptés.

Chaque armes détient un premier programme qui donne deux possibilités. Soit le joueur la ramasse en appuyant sur la touche « E » : en effet quand un joueur est proche de l'arme son le collider détecte le joueur et un message apparaît « Press e to take ». Soit le joueur l'abandonner en appuyant sur la touche « F ». Cependant si le joueur possède déjà trois armes celle qu'il ramasse remplace celle qu'il a en main. Ce script a posé quelques problèmes lors de sa création. Au début nous pensions mettre les armes dans une liste. Le script devait chercher dans la liste la bonne arme pour la rendre opérationnelle. De plus, quand on voulait jeter l'arme, le script devait la récupérer et la supprimer de la liste. Nous avons expérimenté plusieurs fois ce programme, mais nous nous sommes rendu compte qu'il ne fonctionnait pas à 100%, à cause d'un problème d'index. Nous avons du alors repenser entièrement le programme en accrochant les armes à un « game object » fils du personnage allié.

Au niveau du canon de chaque arme un script a été implémenté permettant le tir d'une particule bleue faisant office de projectile. Ce script gère aussi la cadence de tir de l'arme. Si la particule tirée ne rencontre pas d'obstacle elle est automatiquement détruite au bout d'un certain temps qui varie selon les armes.

Enfin au cours de notre travail nous avons modifié les projectiles ronds qui étaient tirés. Ils ne nous convenaient pas puisque nous ne les voyions que très peu. Nous avons décidé d'utiliser des particules pour les tirs des personnages qui sont bleues, rondes et ainsi plus visibles.

### 2.2.4 Mécanique générale

#### Ce qui a été fait par Moustapha

Ce sont des scripts qui ont un impact général sur le jeu.

On a ainsi réalisé un script permettant à la camera de suivre le personnage contrôlé par le joueur. Ce fut un des premier script que nous avons réalisé, sa création nous à demandé de nombreuses recherches sur internet. Nous avons aussi créé des portes qui s'ouvrent à l'approche des alliés et un coffre qui,

lorsqu'on appuie sur la touche « C » s'ouvre. Il peut, soit donner une arme soit, faire apparaitre un zombi soit, ne rien donner, ou soit exploser et faire des dégâts au personnage qui l'ouvre. Pour faire cela nous avons mis en place un script qui fait aléatoirement une de ces actions.

## 2.3 Interface

Depuis la première soutenance à l'état final du projet, l'interface a subi de nombreux changements, il s'agit d'une partie n'ayant jamais été laissée de côté, en effet elle n'a jamais connu de retard, au contraire, elle était même en avance de manière générale.

### 2.3.1 Menu principal

Guillaume a créé le menu multijoueur permettant ainsi aux joueurs de créer un lobby du nom de leur choix ainsi que de se connecter à ceux de leurs amis. Cela a nécessité d'implémenter les boutons pour se connecter et créer un lobby ainsi que les scripts correspondants. Un menu de déconnexion a également été implémenté dans le cas où le joueur perd sa connection ou s'il y a un problème sur les serveurs. Une fois le problème résolu, il pourra alors facilement se reconnecter à son lobby.

Le reste du menu principal a été réalisé par Philippe. Il est aussi à l'origine de l'esthétique et de la disposition du menu principal. Ainsi, en plus de la partie traitée par Guillaume, il y a, un bouton permettant de quitter le logiciel "Quit", un autre, d'accéder à un écran d'options avec de nouvelles interactions possibles "Options", et enfin un dernier permettant d'accéder à la sélection du personnage avant de jouer au travers du bouton "PLAY". Le bouton "Options" va changer l'interface et le joueur ne verra plus les boutons mentionnés précédemment. Il verra à la place de nouveaux éléments d'interface permettant les actions suivantes : la configuration de la fenêtre de l'application en plein écran, la gestion du volume, le choix de la qualité graphique souhaitée et enfin un bouton permettant de revenir à l'écran précédent et de sortir du menu des options "Back". La gestion du volume se fait à l'aide d'un curseur permettant de changer le volume sonore en lui affectant la valeur du curseur. L'affichage en plein écran se fait grâce à une forme d'interrupteur, il suffit de cliquer pour mettre le jeu en plein écran, et de cliquer de nouveau pour le mettre en mode fenêtré. Enfin, un menu déroulant a été mis en place pour permettre le choix de la qualité graphique souhaitée.

Zéphyr est un action-RPG, la personnalisation a donc un intérêt réel, et dans le but de permettre aux joueurs d'avoir davantage de possibilités, différents personnages jouables ont été implémentés avec chacun des caractéristiques différentes mais surtout une compétence unique pour chacun d'entre eux ! Ainsi, lorsqu'on accède au menu de sélection de personnage en cliquant sur le bouton "PLAY", plusieurs boutons ont été mis en place permettant



le choix des personnages et de lancer le jeu avec ce dernier. Quand on arrive sur ce menu un personnage par défaut est enregistré dans les préférences de l'utilisateur, cela permet entre autres que le personnage sélectionné par défaut soit le dernier avec lequel on ait lancé le jeu. Sur cet écran de sélection des personnages, quand on clique sur une des flèches (« => » ou « <= »), le personnage affiché à l'écran, et c'est un autre modèle 3d qui vient le remplacer. Chaque personnage apparait accompagné de son nom ainsi que de ses statistiques (dégâts, points de vie, etc.). Pour des raisons esthétiques nous avons décidé de représenter ces dernières par le biais d'une image superposée par-dessus une autre donnant ainsi l'impression d'avoir une jauge. Chacune de ces jauges est de couleur différente, et un mot à côté désigne ce que représente cette dernière. Une fois le personnage choisi, il devient alors possible de choisir de jouer en solo, ou en multijoueur. Le premier cas redirige vers la scène du jeu avec le bon personnage de contrôlé, le second au menu multijoueur. Pour le moment 4 personnages sont disponibles, et d'autres arriveront sans doute afin de donner encore plus de flexibilité dans la manière de jouer du consommateur.

### 2.3.2 Menu de pause

Philippe s'est chargé de l'intégralité de ce menu. Lorsque le joueur appuie sur la touche "échap" de son clavier, le jeu se met en pause, les couleurs deviennent plus sombres, et des boutons apparaissent. Ces derniers permettent de revenir au menu principal, de quitter le jeu, ou de reprendre le jeu, action qui est également possible en réappuyant sur "échap" une fois le jeu en pause. Il est également à noter que des options sont également présentes sur le menu de pause et suivent la même structure que pour le menu principal, c'est à dire qu'une fois le bouton "options" appuyé, de nouveaux éléments d'interface apparaissent changeant les possibilités de l'utilisateur. On peut notamment y changer le volume.

### 2.3.3 En jeu

Gabriel est à l'origine de la barre de vie du joueur et de celles des ennemis. Deux styles différents ont été adoptés. Pour le joueur un cercle vide disposé aux pieds et entourant le personnage a été choisi. La création de cette interface a été permis grace a un tutoriel de Unity sur la gestion des sliders. Lorsque le personnage du joueur a toute sa vie, le cercle est complet et vert. Lorsque le personnage perd de la vie le cercle se vide dans le sens horaire, et plus il perd de la vie, plus la couleur du cercle tend vers le rouge. Cela a été réalisé a l'aide d'un slider et du script de la vie du personnage. Celui-ci

a été modifié afin d'influer sur l'interface graphique de la vie du personnage. L'interface de la vie des ennemis est, quant à lui, représenté par une barre en 2 dimensions se situant au-dessus de sa tête. Cette implémentation a été effectuée grâce à un tutoriel sur youtube, ainsi que les connaissances précédentes. Lorsqu'il a toute sa vie, la barre est verte et lorsqu'il perd de la vie, la barre verte diminue à droite et à gauche et laisse apparaître une barre rouge sous celle-ci. Ainsi lorsque l'ennemi n'a plus de vie sa barre de vie est totalement rouge. La modification du script de la vie l'ennemi permet de remplir ou diminuer la vie de l'ennemi. Un autre script a été implémenté permettant de garder la barre face à la caméra. Il faut tout de même savoir qu'au moment où l'on rédige ce cahier des charges, il y a encore des doutes sur le fait de garder la barre de vie du joueur tel qu'elle est actuellement, ou de la changer pour quelque chose de plus lisible.

Le reste de cette partie de l'interface été réalisée par Philippe, en commençant par une petite carte (minimap). Pour cela, une caméra a été placée en haut du personnage, et montre en vue de dessus l'environnement du personnage. Enfin pour réaliser la carte, un carré a été ajouté à l'écran de jeu, et sur ce dernier a été superposé le rendu de ce que capturerait la caméra. Pour une meilleure lisibilité, il a également été mis en place une forme d'icone sur cette carte à taille réduite pour représenter le joueur sur cette dernière. Cela a été rendu faisable en superposant un objet par-dessus le personnage en continu. Cette carte se basant principalement sur une caméra, l'image retranscrite dessus et celle de la sphère positionnée au-dessus du personnage, d'où l'impression d'un icône sur la carte.

Un écran de chargement a également été mis en place entre le moment où l'on clique sur le bouton permettant de jouer, et le moment où le monde a fini de se générer. En effet, comme la carte est créée aléatoirement à chaque nouvelle partie, il y a toujours un petit moment durant lequel la carte doit se générer. Sans cet écran de chargement, le joueur verrait la construction de la carte ce qui nuirait à l'expérience de jeu.

Notre jeu se développe autour de l'idée d'apprendre de ses erreurs, et en particulier de mourir pour mieux revenir. Nous avons réfléchi à faire en sorte que tout nouveau joueur ait nécessairement du mal au point de subir la défaite. D'où l'importance de l'écran de fin de partie ou « Game Over » (même si lors de la première et deuxième soutenance il n'était toujours pas présent...). Ainsi, lorsque les points de vie du personnage tombent à 0, un nouvel écran apparaît présentant le texte « Game Over » et la possibilité de

quitter le jeu ou de revenir au menu principal.

En plus de tout cela, l'équipe travaille encore sur l'implémentation de quelques derniers éléments d'interface au moment où ce rapport est rédigé. On peut notamment mettre en avant la présence d'objets collectables qui permettront d'interagir avec d'autres éléments du jeu.

## 2.4 Multijoueur

Toute la partie multijoueur a été implémenté grâce au framework Photon Unity Network. Il s'agit d'un module de Unity qui facilite beaucoup la création de jeu multijoueur car il s'occupe de la gestion des serveurs et de la création de lobby. Il aide aussi à la synchronisation des différents éléments tel que la position et la rotation des personnages ainsi que des ennemis. En plus de cela il est gratuit. Ce qui a permis au jeu d'être développé avec un budget très faible. En effet, nous avons dépensé moins de 10 pour développer Zéphyr.

Aux vues de la facilité d'utilisation de Photon Unity Network, le groupe a finalement revu à la hausse les ambitions pour le multijoueur. A l'origine, nous ne comptions permettre au joueur que de se connecter en multijoueur lorsqu'ils sont connectés au même réseau ; mais vis-à-vis de la faciliter d'implémentation d'un vrai mode en ligne grâce à Photon Unity Network, nous avons décider de ne pas se limiter à un seul réseau pour jouer en ligne.

Pour revenir au jeu en lui-même, après que le joueur ait choisi son personnage, deux choix s'offrent à lui : jouer en multijoueur ou en solo. S'il choisit de jouer en multijoueur, il arrive alors sur un menu où il peut créer un nouveau lobby ou se connecter à un lobby déjà existant. Si le joueur veut créer un nouveau lobby, il peut le nommer de la manière qu'il désire puis les autres joueurs pourront alors s'y connecter. Si le joueur veut rejoindre un lobby déjà existant il entre dans la case rejoindre un lobby le nom du lobby concerné et sera directement connecter avec les autres joueurs. Si le joueur entre un nom de lobby qui n'existe pas encore, alors ne pouvant le rejoindre, il va en créer un nouveau avec ce nom ci.

L'implémentation du multijoueur a été assez compliqué, car elle était parsemée de nombreux bugs. Après la première soutenance, j'étais convaincu que l'implémentation du mode multijoueur serait bientôt terminée. Mais j'ai bien sous-estimé la complexité des bugs qui pouvait en découler. . . Ainsi pour la première soutenance, j'avais déjà créé le menu dont je viens de parler un peu plus haut et la synchronisation des déplacements de personnages. C'est ici que j'ai déjà eu mon premier problème. . .

Après que les lobbies soient créés, le joueur a besoin d'avoir son personnage qui apparaît sur sa carte. Même si cela parait simple, cela est loin d'être le cas. En effet, les premiers scripts que nous avons réalisés considéraient que le personnage était déjà sur la carte au début de la partie. Or dans le mode multijoueur, il ne peut en être ainsi. Le personnage a besoin d'être ajouté

après que la carte soit chargée. Cela permet d'une part, d'éviter tout problème de synchronisation avec les autres joueurs et d'autre part d'avoir plusieurs personnages, ce qui est pratique pour un mode multijoueur...

Mais avant que le joueur soit ajouté, des scripts appelaient ce dernier. Ce qui provoquait logiquement des erreurs... Ainsi il a fallu trouver une solution face à ce problème. Celle retenue fut de changer la conception de la plupart des scripts. Ainsi il fallait à présent que ces scripts commencent par une fonction `Start()` au lieu d'une fonction `Awake()`. Ce qui laisse donc le temps au personnage d'apparaître.

Après cela, un problème subsistait. Les scripts ne trouvaient plus le personnage. En effet, ce dernier n'est pas mis lui-même sur la carte. C'est une copie du personnage qui est ajouté. Ainsi on ne peut donc faire référence au personnage directement. Il faut alors le chercher dans la scène pour après pouvoir le manipuler. Après cela, certes le mode multijoueur a son personnage. Mais le mode solo n'en a plus. En effet, le personnage n'est plus sur la scène d'origine car il était nécessaire de l'ajouter au début de la partie. Il fallait donc également le faire apparaître dans ce mode au début de la partie.

Simple me direz-vous ?

Toujours pas... Pour ajouter notre personnage dans le mode solo, un script sur le menu principal restait de scène en scène et ajoutait le personnage lorsque la scène de jeu était chargée. Ce qui avait pour effet d'également ajouté un personnage dans le mode multijoueur. Donc 2 personnages contrôlés par un seul joueur. Pour y remédier, le script pour le personnage dans le mode multijoueur et dans le mode solo ont été unifiés. Et une séparation plus claire entre le multi et solo a été faite. Ainsi de nombreux scripts ont dû être modifiés et adaptés pour être compatibles avec le mode multijoueur.

A ce moment là un bug avec la synchronisation de l'orientation des personnages subsistait. Mais celui fut simple à régler. Il résultait d'un simple oubli d'une ligne dans le script de gestion des mouvements des joueurs. Mais le plus gros du travail en multijoueur entre la première et la deuxième soutenance fut au niveau de la synchronisation des ennemis.

L'un d'entre eux concernait la position et les attaques des ennemis. Ainsi lorsqu'un ennemi attaquait un joueur, ce dernier ne prenait pas des dégâts du point de vue des autres joueurs. J'ai alors partagé ce problème au groupe. Moustapha a modifié le script de choix des cibles et d'attaque des ennemis. Ainsi maintenant si un joueur est attaqué et meurt, il subit bien les dégâts et meurt du point de vue de tous les autres joueurs.

Lors de la première soutenance, Zéphyr souffrait également d'un problème de synchronisation des mouvements des ennemis. Après la modification des scripts de mouvements des ennemis pour synchroniser leur position, il fallait encore les faire réapparaître (comme il est nécessaire de le faire pour les personnages des joueurs). Etant donné que cette partie est gérée dans la génération procédurale de Gabriel et que je ne voulais pas trop toucher à celle-ci car Gabriel avait eu beaucoup de mal à la développer et que je ne voulais pas tout casser en plus du fait, qu'il aurait fallu que je comprenne son fonctionnement pour faire cela. Ainsi j'ai décidé d'appliquer une solution un peu moins esthétique mais beaucoup plus simple à implémenter. Celle-ci consiste à la récupération de tous les GameObject avec un tag « Ennemi » puis à l'aide d'une boucle l'instanciation de leur équivalent du mode multijoueur. Ce qui réglait le problème.

Mais le plus gros restait à venir... Peu après que Gabriel ait terminé la génération procédurale de carte, nous nous sommes rendu compte que nous avions un autre problème plutôt drôle, cette fois. Chaque joueur générait sa propre carte en se connectant au lobby. Ainsi chaque joueur jouait sa propre partie de son côté... Sans avoir les mêmes ennemis ni les mêmes salles... Ce qui était plutôt comique avec cela était qu'en jouant on voyait les autres joueurs avancés dans les murs sans aucune raison alors que de leur point de vue il était à un autre endroit et bougeait normalement.

Bref, il a fallu implémenter la génération procédurale de carte en mode multijoueur. Pour commencer, j'ai appliqué la même méthode que pour la synchronisation des ennemis et des joueurs. Mais cela ne faisait absolument rien. J'ai compris un peu plus tard que cette méthode de synchronisation ne pouvait pas marcher. Dans sa génération procédurale, Gabriel commence avec un objet carte qui est vide et puis ajoute les différentes salles sur cet objet. Ainsi lors de l'exécution du jeu, la carte était modifiée puis supposément partagée aux autres joueurs. Mais en réalité, les autres joueurs ne plaçaient sur leur version du jeu qu'une carte vide car il instanciat la carte qu'ils avaient locales (donc non modifiée, soit vide). A ce moment-là, je commençais à paniquer en me disant que j'allais devoir modifier tout le script de Gabriel pour partager toutes les modifications avec les autres joueurs. Ce qui aurait été plutôt long et rébarbatif à faire vu la taille de ce script...

Heureusement, nous avons trouvé une solution bien plus esthétique. En partageant ce problème avec Gabriel, celui-ci m'expliqua comment marchait sa génération procédurale dans l'espoir qu'une meilleure compréhension de celle-ci me permettrait de plus facilement l'implémenter en multijoueur, et ce fut le cas. Ainsi au lieu de récupérer la carte que sa génération procédurale

créait pour après la partager. La solution choisie fut de récupérer une seed de la valeur pseudo-aléatoire utilisé dans sa génération et de partager celle-ci avec les autres joueurs pour que ceux-ci de leur côté puisse générer la carte à partir de la même seed. C'est-à-dire la même carte. Le problème était réglé.

## 2.5 Implémentation des ennemies et des alliés

### 2.5.1 Les ennemis

#### Ce qui a été fait par Moustapha

La création du premier ennemi a été longue à réaliser.

Les modèles ennemis ont été récupérés sur l'Asset Store de Unity. Nous avons onze ennemis qui sont repartis dans cinq groupes différents. Tous nos ennemis sont animés. Ils ont une animation de déplacement, une animation de repos et depuis peu une animation de mort. La diversité des ennemis et leurs caractéristiques différentes ont été voulu pour rendre le jeu plus attrayant. Ils forment une « team » d'ennemis impressionnantes.

#### Le premier groupe : Zombi

C'est le groupe de base du jeu et c'est le groupe le plus facile à combattre. Il est composé de deux personnages, le Zombi et son Boss. Ils ne peuvent qu'attaquer au corps à corps leurs cibles. Ils ont un rayon d'action assez faible, ils font peu de dégâts et se déplacent rapidement. Pour les neutraliser efficacement il faut tourner autour d'eux en leur tirant dessus.

#### Le deuxième groupe : Ice

C'est le groupe de base de l'attaque à distance. Il est aussi composé de deux personnages, l'Ice et son Boss. Leur attaque est le tir d'une particule qui, lorsqu'elle entre en contact avec le collider d'un joueur, lui inflige des dégâts. Leur rayon d'action est assez faible et la particule tirée fait peu de dégâts. Cependant ces ennemis ont beaucoup de points de vie. Pour les rendre inopérants il faut les bombarder de particules.

#### Le troisième groupe : Alien

C'est un des groupes ennemis les plus redoutables. Il est aussi composé de deux personnages, l'Alien et son Boss qui ont plusieurs types d'attaque. La première est assez simple et ressemble à celle de l'Ice. Ils tirent simplement des particules en direction de leur cible. Leur deuxième attaque intervient plus rarement mais est beaucoup plus puissante. Tout autour d'eux une salve de plusieurs centaines de particules est lâchée et elle est très difficile à éviter. Leur déplacement est assez lent mais leur rayon d'action est important. Pour les mettre hors d'état de nuire, l'utilisation du « Dash »



est nécessaire.

#### Le quatrième groupe : Mage

C'est un groupe assez particulier.

Il est composé de trois personnages le Mage Baby, le Mage Parent et le Mage Boss. Ils ont une attaque simple, ils tirent une particule mais celle ci est beaucoup plus grosse que celle des autres ennemis. Elle a donc plus de chances de toucher un personnage « allié ». Mais la vraie différence de ce groupe réside dans le fait que le Mage Parent et le Mage Boss peuvent se diviser. En effet quand le Mage Boss meurt il se transforme en deux « Mage Parent » et quand le Mage Parent meurt il se transforme en deux « Mage baby ». Ce groupe se déplace très rapidement, a un rayon d'action conséquent mais cependant ne fait que très peu de dégâts. Pour les neutraliser il faut éviter qu'ils se divisent tous en même temps et donc les tuer un par un ce qui demande de la dextérité.

#### Le cinquième et dernier groupe : Gnome

C'est le groupe « sniper » des ennemis.

Il est composé de deux personnages, le Gnome et son Boss. Ils ont la particularité de tirer rarement une seule particule mais qui fait beaucoup de dégâts. Seulement deux particules sont nécessaires pour anéantir un allié. Il se déplacent très lentement mais leur rayon d'action est le plus important des ennemis. Pour pouvoir les vaincre sans être vaincus il est nécessaire de ne pas être touché.

Tableau récapitulatif des ennemis

Nom de l'ennemi	Points de vie	Vitesse de déplacement	Rayon d'action	Points de dégât	Vitesse du tir
Zombi	100	5	20	5	—
Zombi Boss	1500	5	40	10	—
Ice	110	5	30	10	10
Ice Boss	1600	6	60	15	10
Alien	80	3.5	40	Tir simple : 5 Tir spécial : 5	Tir simple : 10 Tir spécial : 10
Alien Boss	1700	4	80	Tir simple : 10 Tir spécial : 10	Tir simple : 10 Tir spécial : 10
Mage Baby	50	8	50	5	50
Mage Parent	80	10	70	7	50
Mage Boss	1700	12	100	10	50
Gnome	70	2	100	50	100
Gnome Boss	1800	3	200	70	100

## 2.5.2 Les personnages alliés

### Ce qui a été fait par Moustapha

Comme pour les ennemis le modèle de nos personnages a été récupéré sur l'Asset Store de Unity. Les quatre alliés sont tous identiques par leur forme. Leur différence est plutôt visuelle, car ils ont des couleurs qui leur sont propres. Ils ont aussi des caractéristiques et des compétences particulières.

Tableau récapitulatif des alliés

Prénom Nom	Points de vie Maximum	Dégât de l'arme de base	Compétence	Vitesse	Vitesse du dash
Thea Louest (Beige)	110	3	Plein Soin	10	12
Annie Mall (Vert)	80	7	Onde de Choc	10	12
Harry Cover (Mauve)	100	5	Le Mur	8	14
John Athan (Rouge-Rose)	100	5	Invincible	12	10

### 2.5.3 Les armes des alliés

#### Ce qui a été fait par Moustapha

Nous avons six armes opérationnelles qui ont toutes des caractéristiques différentes. Nous avons récupéré les modèles de ces armes sur l'Asset store de Unity. La création de la première arme a été la plus dur. Les autres ont été réalisées très rapidement.

Baby : C'est l'arme de base, elle sera commune à tous les joueurs au début de la partie et permettra de débiter efficacement les niveaux.

AK : C'est une arme assez basique, elle ne sera pas utile pour se sortir de situations complexes. Le joueur devra rapidement s'apercevoir que cette arme n'est pas efficace pour progresser dans le jeu.

Fusil à pompe : Cette arme tire trois balles en même temps elle sera d'une grande efficacité contre des hordes d'ennemis.

Machine Gun : Cette arme à une cadence de tir élevée mais elle fait peu de dégâts, elle est donc très efficace contre les ennemis qui ont peu de vie.

Sniper : Le sniper a une cadence de tir très faible mais fait énormément de dégâts. Il est très performant contre les ennemis qui font beaucoup de dégâts ou qui se déplacent très lentement.

Rocket : Ce lance rocket propulse une rocket sur une longue distance. Le projectile envoyé se déplace lentement mais lorsqu'il touche un ennemi il lui inflige des dégâts et explose créant aussi des dégâts de zone. Son utilité réside surtout lorsque de nombreux ennemis font face.

Concernant l'ensemble de ces armes il y a eu une amélioration au niveau de la forme des particules émises lors du tir. En ce qui concerne plus particulièrement Rocket, un nuage de fumée est créé au moment du tir et lorsque le projectile tiré explose de nombreuses particules sont projetées sous la forme d'une sphère affectant les ennemis dans la zone. L'ensemble de ces détails de tir améliore l'esthétique de notre jeu, le rendant plus attrayant.

Tableau récapitulatif des armes

Nom de l'arme	Cadence de tir	Vitesse	Dégâts (par défaut)
Baby (Orange Métallique)	0.2	50	5
AK (Or)	0.2	50	10
Fusil à Pompe (Vert métallique)	0.5	50	3 * 10
Machine Gun (Violet métallique)	0.1	40	7
Sniper (Bleu métallique)	3	130	80
Rocket (Argent)	5	20	90 + 5 * 30

## 2.6 Les cartes

Tout d’abord, nous voulions générer nous mêmes nos cartes de façon procédural à partir de salles que nous faisons nous mêmes. Nous avons commencé la création de ce dernier. Or la complexité du script est élevé. Le sujet étant peu renseigné il était difficile de s’informer afin de procéder nous même a cette génération. En effet il faut vérifier que les portes de chaque bâtiment soient bien l’une en face de l’autre, sinon il faut faire tourner la salle dans le bon sens pour qu’elles soient en face, ainsi que les salles et les couloirs ne se superposent pas.

Nous n’avons pas réussi a créer ce système permettant de générer lui même ses propres cartes de façon aléatoire a partir d’un jeu de salles déjà créé. Mais nous avons trouvé un Asset de Unity permettant de générer des cartes. Cela nous a paru être une alternative intéressante. En effet cet asset peut générer des cartes avec des salles de tailles aléatoires sur une surface et un nombre de salle définie. La modification des scripts de l’asset a permis de l’utiliser pour notre projet. Sa modification nous a permis de créer des cartes de tailles proportionnées par rapport à nos personnages et ennemis, avoir des colliders sur les murs ainsi que nos balles ne passent pas a travers ces derniers.

Il fallut plusieurs heures afin de comprendre les différentes class et fonctions implémentées. Le premier objectif fut celui d’élargir les couloirs, qui produit de nombreuses erreurs comme des couloirs avec des irrégularités, ce que nous ne recherchions pas.

Une fois ce problème résolu, le prochain objectif était de créer trois différents types de salle. Les salles de coffres, qui sont de petites salles contenant un coffre, les salles de monstres, qui contiennent un nombre aléatoire de monstres différents et enfin les salles de bosses qui contiennent un unique boss. Cette implementation fut la plus longue. En effet il fallut créer trois fonctions différentes pour les salles. Elles produirent de nombreux bogues comme l ‘apparition des couloirs uniquement, l’apparition aléatoire de salle de boss (il arrivait de ne pas avoir de salle de boss sur la carte). Une fois que la génération globale des cartes fut accomplie, il restait a faire apparaître les monstres et les coffres dans leur salles respectives. Il fallu créer une fonction permettant de faire apparaître les monstres au milieu de la salle une autre faisant apparaître un coffre il manque encore celle qui faisait apparaître le boss. Ainsi notre carte se génère aléatoirement et place les monstres aux bons endroits.

A ce moment la nos cartes n’avait pas du tout un aspect de donjon, en effet les salles étaient multicolores, les couloirs blancs et parfois beaucoup trop long entre deux salles. C’est pourquoi nous avons procédé a quelques

améliorations, nous avons implémenté les textures, elles s'appliquent à toute la carte et changeront à chaque niveaux grâce a un script. Ensuite nous avons réduit l'espace entre les salles ce qui nous a permis d'avoir des couloirs plus court afin d'avoir plus d'actions. Nos salles semblaient encore vide, nous y avons donc rajouté un peu de vie en faisant apparaître des roches aléatoirement de tailles différentes, sans quelles sortent des salles, ni quelles bloquent le passage.

Il restait pourtant des bogues, en effet les monstres pouvait passer à travers certains murs, pour cela nous avons augmenté la taille du collider derrière les murs, en quintuplant sa taille. Le sol aussi empêchait le personnage de se déplacer de façon fluide, pour pallier à ce problème nous avons réduit la taille des colliders au sol en les aplatissant totalement. La première salle n'était pas centré, le personnage apparaissait dans le vide... ce qui est quelque peu problématique, il fallut réussir a centrer la première salle à l'aide de calculs mathématiques. Les pierres n'avaient pas les bonnes dimension, ne se plaçaient pas aux bons endroits et n'avait pas de colliders, on leur mis facilement des colliders puis il fallut faire en sorte que les pierres ne sortent pas de la salle et qu'il y en ai assez pour le coté esthétique. Il fallut faire attention que les roches n'apparaissent pas sur les monstres. Car comme il faut tuer tous les monstres pour finir le niveau ce serait embêtant que certains restes coincés dans les roches...

Les cartes sont quasiment finis. Il manque les portes, nous allons changer le système d'apparition des monstres, en effet ils se généraient automatiquement au début du jeu, or cela demande beaucoup d'énergie, ainsi grâce au script des portes nous pourront faire apparaître les monstres seulement lorsque le joueur sera rentré dans la salle, les autres portes de la salle se fermeront et ne se réouvriront seulement une fois que le(s) joueur(S) auront tué tous les monstres de la salle.

Le script des portes n'a pas encore été implémenté mais il est prêt, il ne reste plus qu'à placer les portes ainsi que de faire des légères modifications dans le script de génération des cartes et les cartes seront opérationnels.

## 2.7 Effets sonores

L'immersion est un aspect important dans un jeu vidéo. Et pour favoriser ce dernier point, il ne faut pas négliger la musique ainsi que tous les bruitages provenant de ce monde virtuel. Ainsi la partie audio a toujours été un élément non négligeable dans la mise en place du projet, et cela c'est vu par le fait que de manière générale, l'audio était légèrement plus avancé que ce qui était attendu tout au long de la réalisation de Zéphyr.

La partie effets sonores a été traité exclusivement par Philippe. La musique implémentée au jeu est présente à la fois dans le menu principal ainsi que dans le jeu. Cette dernière est gérée de manière assez simple. Une classe "Sound" a été créée répertoriant les différentes musiques et effets sonores incorporés dans le jeu. Chacun des éléments de cette classe vient avec différents attributs, tel qu'un nom ou une hauteur (la hauteur de la musique) notamment. Un script a ensuite été créé, ce dernier fait appel à une liste d'éléments issus de la classe "Sound" réunis dans une liste dont on se sert à de nombreuses reprises. Ainsi, dès que le script est appelé, dépendant de la scène dans laquelle se trouve actuellement le joueur, c'est à dire dans l'état actuel du jeu, s'il se trouve dans le menu principal, ou en jeu, il appellera la musique du menu principal ou bien celle du jeu, et ce grâce à une fonction permettant de lancer une piste audio. Il est à savoir qu'actuellement l'équipe envisage un changement dans la manière de fonctionner de ce script audio, il a certaines limitations, et après des recherches plus approfondies, il semblerait qu'un moyen plus adéquat puisse être mis en place.

En plus de la musique, des effets sonores ont été introduits dans l'application. De nombreux effets sonores ont été ajouté dans le jeu grâce à divers moyen. Dont le logiciel BOSCA CEOIL permettant de créer des effets sonores assez facilement, Audacity pour enregistrer le son, ou même grâce à des bibliothèques de sons en ligne libres de droits. Différents effets sonores sont donc présents dans le jeu. Il y a par exemple le bruit que fait l'arme lorsque l'on tire avec, le son de l'esquive aussi, et bien d'autres.



## 2.8 Site internet

### Ce qui a été fait par Guillaume

Le site web était quasiment terminé à la deuxième soutenance. En effet, entre la première et la deuxième soutenance, nous avons déjà réalisé toutes les pages (exceptées la page de téléchargement) nécessaires au site web. Cela a pris beaucoup de temps car nous ne nous étions pas concentrés sur le site pour la première soutenance. A cette soutenance, nous n'avions qu'une simple page d'accueil avec deux boutons (plutôt moches) redirigeant vers la version française et anglaise du site qui était alors totalement vide.

Depuis la deuxième soutenance, la page de téléchargement ainsi que divers éléments graphiques ont été ajoutés. Sur la page de téléchargement, vous pouvez ainsi retrouver une version téléchargeable du jeu ainsi que tous les documents produits sur la période dans le cadre du Zéphyr (cahier des charges, rapport de première et deuxième soutenance ainsi que ce rapport final du projet).

Les éléments graphiques comprennent une amélioration de l'entête qui reste maintenant tout le temps accessible même lorsque l'on descend pour lire le reste de la page ; et aussi un footer en bas de toutes les pages qui permet également d'accéder à divers pages (support, page de contact, mention légales...). Un fond d'écran différent a été ajouté à chaque page, ce qui rend bien mieux que le précédent fond blanc.

Nous avons également ajouté un trailer vidéo depuis la dernière soutenance sur la page principale du site et une version gif sur la page d'accueil. Ce dernier présente le scénario de Zéphyr ainsi que plusieurs scènes de gameplay et permet de mettre en avant le jeu. Les scènes de gameplay ont été enregistrées grâce à OBS studio.

Ainsi pour résumer le site web comprend :

- Une page d'accueil qui redirige vers la version française ou la version anglaise du site
- Une page de téléchargement
- Une page expliquant les mécaniques du jeu
- Une page listant toutes les ressources utilisées
- Une page présentant le scénario de Zéphyr
- Une page détaillant l'histoire de la création du jeu
- Une page présentant l'équipe
- Une page de support
- Une page pour se connecter

La page de téléchargement contient une version téléchargeable du jeu ainsi que tous les documents liés au projet (cahier des charges, rapport de première et deuxième soutenance ainsi que ce rapport final du projet). La page avec les mécaniques du jeu détaille tout ce qui pourrait être utile au joueur. Ainsi on compte 4 parties. La première développe toutes les actions que peut faire le personnage (bouger, tirer, capacité spéciale, roulade...) ainsi que leurs touches respectives. La deuxième présente les différents personnages que le joueur peut choisir au début de son aventure avec leur différentes capacités spéciales, vitesses, points de vie... La troisième partie présente les armes que le joueur pourra utiliser au cours de son aventure avec leurs caractéristiques. Enfin, la dernière partie présente les ennemis et les boss que le joueur affrontera à travers les différents niveaux qu'il traversera.

La page listant les ressources utilisées regroupent tout ce qui a été utilisé d'extérieur dans ce projet, cela comprend des assets comme Photon Unity Network ou NavMesh ou encore des logiciels comme Rider ou Unity. La page présentant le scénario est plutôt simple, elle reprend juste le scénario de Zéphyr.

La page détaillant l'histoire de la création de Zéphyr a été un peu plus longue à créer. Déjà au niveau de l'écriture, où cela a demandé d'être régulier et rigoureux, car elle-ci est détaillée semaine par semaine. Mais aussi niveau design car pour éviter qu'elle soit trop monotone, j'ai décidé de placer l'histoire de la semaine à droite puis à gauche de la page, avec de l'autre côté une image.

La page présentant l'équipe de développement Zelstrom comprend une courte description de l'équipe et de nos ambitions. Ainsi qu'une liste de ses membres avec un mail de contact si nécessaire. Nous tenons à préciser que nous avons voulu que le site soit réactif, pour être accessible autant depuis un ordinateur, une tablette ou un smartphone. Ainsi par exemple sur cette page, alors que sur ordinateur, les développeurs de Zelstrom sont présentés horizontalement, sur mobile, ils sont présentés verticalement.

## **Ce qui a été fait par Philippe**

Sur la fin du projet, alors que Guillaume s'était concentré principalement sur le développement du site web depuis quelques semaines d'affilée, négligeant un peu le multijoueur par ailleurs. Il a dû se reconcentrer sur ce dernier pour être au point pour la deuxième soutenance.

Il a ainsi demandé un peu d'aide sur le groupe afin de l'aider à terminer le site pour la deuxième soutenance. Je me suis donc proposé pour l'aider étant donné que j'étais bien avancé sur ma partie concernant les interfaces. Je me suis donc occupé de rédiger l'avancement de l'histoire du développement de

Zéphyr et de traduire certaines pages en anglais pour la version anglophone du site. J'ai notamment traduit l'histoire de Zéphyr et la page de gameplay.

## Chapitre 3

# Ressenti de l'équipe sur la réalisation du projet

### 3.1 Moustapha

Les joies :

Mai 2019 Zéphyr fonctionne.

La joie la plus importante est que notre projet ait abouti. En décembre lorsque le projet nous a été présenté j'ai été confronté à un grand scepticisme, car je ne connaissais absolument pas ce domaine et n'ai jamais été fan de construction de jeux vidéo.

Je me suis beaucoup investi pour connaître les différentes structures à mettre en place pour concevoir le jeu. En cela le cahier des charges m'a permis, avec mes camarades de déterminer les axes que nous avions à travailler.

Le fait d'élaborer ce projet en groupe a été très motivant, car chacun apporte ses idées qui sont confrontées avec celles des autres et qui petit à petit permettent de déterminer les bonnes solutions.

La joie est aussi présente lorsqu'on arrive à faire fonctionner un élément tel qu'on l'avait prévu.

L'intérêt pour ce travail vient d'abord du fait que, lorsque je mets en place les différents scripts, je suis complètement immergé dans cette activité et je prends beaucoup de plaisir à réfléchir comment trouver une méthode pour arriver à ce que je veux produire. L'intérêt vient aussi du fait que j'ai beaucoup apprécié de faire des recherches sur Internet pour trouver des solutions aux problèmes rencontrés. Ainsi, j'ai pu connaître et comprendre comment fonctionnent certains composants comme le collider, certaines fonctions comme

awake, start, update, OnTriggerEnter, ou certains scripts comme ceux concernant le déplacement des personnages.

Quand on réussit à faire ce que l'on veut, on a un véritable sentiment de fierté d'autant plus que l'on a trouvé par soi même la solution sans aucune aide extérieure si ce n'est les cours d'algorithmique et les différents TP de programmation réalisés depuis le début de cette année à EPITA.

La recherche sur internet, la réflexion, et finalement la réussite sont des moments particulièrement enthousiasmants, du reste j'ai souvent été en avance sur les dates programmées dans le cahier des charges, car mon implication dans ce travail a été régulière. J'ai aimé l'autonomie qui nous a été laissée et qui m'a permis de planifier mon travail comme je le souhaitais. J'ai aimé aussi les échanges avec mes camarades : Ces échanges ont permis de travailler plus efficacement, de relever les challenges, de mettre en place la mécanique de jeu, le design des personnages, des ennemis et des armes. Mon travail a été accepté par les membres de l'équipe Zelstrom.

### **Les peines :**

La première difficulté a été de ne pas avoir réussi à créer des personnages spécifiques.

En effet la réalisation des graphismes me pose quelques soucis. Réaliser un personnage en 3D est pour moi difficile, car je ne suis pas un expert en dessin. J'aurais aimé donner à chacun d'entre eux un visage et une corpulence différente. J'aurais aimé dessiner de très nombreux détails, j'aurais aimé utiliser de très nombreuses nuances de couleurs et de formes. Pour remédier à ces lacunes , puisque je ne voulais pas que nos personnages prennent des formes géométriques, comme des cubes ou parallélépipèdes, je les ai récupéré sur l'Asset Store de Unity.

La deuxième difficulté est apparue souvent lors de l'ajout d'une nouvelle caractéristique au jeu. Lorsqu'un élément est ajouté ou modifié il peut créer des bogues qui n'étaient pas visibles avant. C'est par un travail minutieux et long qui demande de l'imagination que ces problèmes ont été résolus. Par exemple quand nous avons testé les personnages sur une carte représentant un espace de jeu, nous nous sommes aperçu qu'ils passaient à travers les murs lors de l'utilisation du « Dash ». Il a fallu donc revoir le déplacement des personnages sans affecter la façon de jouer.

La troisième difficulté rencontrée met en avant le fait certaines tâches peuvent être chronophages. Ainsi pour résoudre le problème du pivotement du per-

sonnage en fonction de la souris, de longues recherches ont été nécessaire sur internet car peu de source répondait efficacement à ce problème.

Enfin la dernière difficulté a été d'organiser les scripts. Au début de notre travail nous ne savions pas s'il fallait les subdiviser ou les rassembler. On a finalement choisi de faire des scripts communs à plusieurs alliés, ennemis et armes.

### **Les regrets :**

Si globalement le travail a été très positif pour moi, j'ai eu quelques regrets.

Il y a eu quelques tâches répétitives notamment pour la création des ennemis et des armes.

Je pensais par ailleurs que nous aurions du beaucoup plus coder pour faire fonctionner le jeu. Je viens de découvrir cette année la programmation et je suis très passionné : j'imaginai que ce projet m'aurait permis d'acquérir de nouvelles compétences en programmation. Cependant je me rend compte que le temps restreint ne nous permettait pas d'élaborer un projet dans lequel il y aurait eu des programmations complexes.

## 3.2 Philippe

En ce qui me concerne Zéphyr a été un projet sur lequel il a été particulièrement plaisant de travailler. Différentes raisons me font dire cela dont la dynamique de groupe dans un premier temps. Avant de former Zéphyr nous nous connaissions déjà et cela à faciliter les échanges au sein du groupe, on pouvait facilement dire ce que l'on pensait sur un sujet sans craindre que les informations soient mal comprises. Le groupe fonctionnait très bien là-aussi, nous avions des profils assez semblables en ce qui concerne nos goûts. Il était alors assez facile de se mettre d'accord sur plusieurs points esthétiques. De plus le projet a permis de renforcer la cohésion entre chacune d'entre nous, c'était souvent des occasions de se retrouver ou tout simplement parler.

Il n'en reste pas moins que Zéphir a surtout été intéressant par rapport à un point, il s'agit de l'expérience accumulée. Pour ma part je m'occupais principalement de la partie interface et son du projet. Et j'ai pu expérimenter beaucoup tout au long du projet. J'ai pu essayer de mettre en avant notamment divers choix esthétique dans les menus par exemple. Mais j'ai surtout pu voir de nouvelles manières d'appréhender le code. Jusqu'à maintenant la programmation que l'on faisait à Epita ne nous montrait que les bases dans une discipline, mais grâce au projet j'ai pu prendre en main divers logiciels permettant de compléter la formation reçue à Epita. Cela peut aller de Audacity ou Bosca Ceoil que j'ai utilisé pour la partie audio du projet à l'utilisation de Unity, framework que certaines entreprises utilisent aujourd'hui et dont l'expérience va pouvoir être un plus pour se démarquer d'autrui. Autre chose très appréciable relevé au cours du projet, il s'agit de la liberté donnée. Pour ma part c'est la première fois que j'ai autant de liberté pour réaliser un projet, et le plus dur finalement c'est de réussir à ne pas se perdre dans ce manque de directives. Il faut alors apprendre à bien se concentrer sur un point en particulier avant d'aborder une prochaine étape. Cela est particulièrement vrai avec les différents bogues qu'il a fallu corriger. En effet c'est toujours plus amusant et enrichissant d'implémenter de nouveaux éléments dans le jeu plutôt que de chercher à corriger un élément déjà présent, et ce encore plus quand on ne sait pas par où commencer à chercher. Je dirai que c'est même là que réside mon principal défaut. Lors de la deuxième soutenance, un bogue m'empêchait d'interagir convenablement avec les boutons du menu de pause, ce bogue était handicapant et empêchait de revenir au menu principal, et le vrai souci est que ce dernier subsistait depuis la première soutenance déjà.

Cette expérience a aussi pu être très frustrante en certaines occasions, mais ça permet d'avoir une nouvelle façon de réfléchir sur un problème. C'est particulièrement vrai avec les bogues. Quand j'en avais un au début du projet, mon premier réflexe était de lancer plusieurs fois le jeu pour voir si le problème disparaissait (ce qui avait peu de chances d'arriver...). Maintenant le premier réflexe est de chercher de quelle partie du programme le problème est originaire, et ensuite de chercher des solutions dans la manière de le corriger.

En résumé, Zéphir a été une expérience formidable, grâce à l'expérience acquise dans le travail en groupe et la manière de réfléchir sur des problèmes. Mais surtout un projet divertissant, source d'une fierté que je n'avais que très peu ressenti, qui est d'avoir fait quelque chose avec un groupe d'amis à partir de rien.



### 3.3 Gabriel

Tout d’abord, je pense que la réalisation de ce projet en première année est une très bonne chose car cela nous permet d’avoir un premier contact entre les études et le monde réel du travail. C’est pourquoi je trouve ce projet très enrichissant.

#### Les joies :

Une variable souvent oubliée, est le temps. En effet nous savions tous qu’il fallait s’y mettre au plus vite pour ne pas se laisser déborder comme l’ont dit d’autres groupes l’année passée. C’est pourquoi je me suis intéressé à beaucoup d’éléments de Unity qui est un monde très vaste. J’ai suivi quelques gros tutoriels en entier traitant de parties importantes du développement d’un jeu vidéo. Cela m’a permis d’acquérir une maîtrise dans le logiciel et sur des éléments très variés, allant des animations des particules, à l’interface de vie des personnages et des ennemis ainsi qu’à leur contrôle en passant par la génération aléatoire de cartes. Je suis très content de m’y être intéressé car ça m’a permis de voir tous les domaines que nous pourrions développer dans notre jeu.

Je suis aussi très satisfait de la cohésion du groupe, nous avons pu nous rapprocher d’avantage tout en faisant l’acquisition de nouvelles connaissances. En premier, je me suis penché sur la partie de la Mécanique De Gameplay à l’aide de tutoriels Unity, or voyant que Moustapha avançait à bonne allure et plus vite que moi, je décidais alors de lui donner mon rôle de Responsable. C’est alors que je me suis orienté vers Les Cartes et L’interface. J’ai trouvé très enrichissant le fait de chercher des informations ainsi que d’essayer de trouver par soi-même des solutions, c’est pourquoi j’ai beaucoup aimé faire l’interface des vies du joueur et des ennemis, que j’ai essayé de rendre assez original. J’ai entièrement développé les cartes à partir d’un générateur préexistant, je fus réellement satisfait lorsque celles-ci fonctionnèrent enfin car en effet elles furent ma plus grande peine !

#### Les peines :

Les cartes. La complexité de l’algorithme afin de générer nos propres cartes de façon procédural à partir de salles préexistantes fut très compliqué, mais à force de chercher j’ai trouvé des solutions que j’ai pu adapter à notre jeu. J’ai repris un générateur de carte que j’ai modifié par la suite. Tout n’était pas au point mais c’était très encourageant ! En effet maintenant je comprends mieux pourquoi il faut toujours commenter son code, j’ai passer

de nombreuses heures a comprendre son fonctionnement (car il n'était pas du tout commenté), à le modifier afin de l'adapter totalement à notre jeu, j'ai du faire face à de nombreux bogues qui m'ont permis de mieux comprendre le code et de toujours l'améliorer.

### **Les regrets :**

Comme j'étais seul sur les cartes cela ne m'a permis de me concentrer sur d'autres parti du jeu, même si j'ai un peu touché à tout. J'étais suppléant dans la gestion du son or je n'y ai jamais touché. De même pour le multi-joueur. Ainsi chaque personne du groupe s'est principalement concentré sur une partie du jeu. j'aurais aimé être plus présent dans d'autres partis mais cela ne nous a pas empêché de travailler en groupe !

Je me suis beaucoup impliqué dans le projet et en plus des connaissances que j'ai acquises, j'ai développé une méthode de travail et d'apprentissage qui est indispensable à tout bon ingénieur. J'ai trouvé cet exercice très enrichissant et je suis motivé à continuer sur ma lancée dans le développement de Zéphyr, afin d'arriver à un résultat proche de nos attentes et qui satisfait toutes l'équipe.

## 3.4 Guillaume

### Les joies :

Ce que je ressens le plus en pensant à ce projet est de la fierté. Alors qu’au tout début notre personnage était juste un cube moche qui affrontait d’autres cubes moches immobiles sur un terrain plat et blanc. Maintenant, le joueur incarne un vrai personnage qui peut bouger, tirer, utiliser une compétence spéciale ou faire des roulades. Il se balade dans un donjon généré aléatoirement et affronte durement ses ennemis. Le chemin parcouru a été long mais le résultat est incroyable à nos yeux. De même, le site web dont j’étais en charge, que j’ai entièrement créé, que j’ai vu évoluer, est maintenant également très beau et me rend tout aussi fier.

C’est également sur ce point que ce projet se distingue. Contrairement à ce que nous avons l’habitude de faire qui peut être assez superficiel ou théorique comme l’impose le format respectif des cours hebdomadaires de programmation encadré par les ACDC et les cours d’algorithmique. Ce projet est justement concret et approfondi grâce au temps qui nous est laissé pour le réaliser.

De plus, ce projet est notre premier gros projet. En effet, dans notre scolarité nous n’avons jamais été confronté sur une si longue durée au travail en équipe. Cela fait donc près de 5 mois que nous développons ce jeu vidéo et encore une fois, pour arriver à un résultat littéralement inimaginable à notre commencement.

Pendant cette longue période, nous avons inévitablement rencontré beaucoup de problèmes ou de bugs. Et la communication sur ce point était remarquable. Par exemple, en multijoueur, lorsqu’un ennemi attaquait un joueur, il ne prenait pas des dégâts du point de vue des autres joueurs. J’ai partagé ce problème au groupe. Et Moustapha a modifié le script de choix des cibles et d’attaque des ennemis. Ainsi maintenant si un joueur est attaqué et meurt, il subit bien les dégâts et meurt du point de vue de tous les autres joueurs. Cette entraide et cette collaboration a donc inévitablement beaucoup rapprocher les membres du groupe.

### Les peines :

Dans ce projet, j’étais en charge du multijoueur qui s’est avéré plus complexe que je ne l’avais prévu. Ainsi par exemple, j’ai pendant un long moment confronté à un problème de synchronisation de la position des ennemis. J’avais beau relire le script concerné, regarder en boucle le tutoriel concerné, vérifier s’il n’y avait pas une erreur qui entraînait en conflit avec ce script...

Impossible d'en trouver la cause... Le bug est resté sur le jeu pendant plusieurs semaines, j'étais incapable de trouver son origine. J'ai décidé de suivre à nouveau le tutoriel en entier en refaisant étape par étape tout ce qui était décrit. Le bug est donc maintenant fixé mais son origine reste inconnu...

Au début de Zéphyr, j'avais également une autre peur. Le fait est que pour commencer à développer le jeu nous avons suivi des tutoriels (beaucoup de tutoriels). C'était la seule possibilité car nous ne maîtrisions pas Unity... Ma crainte à ce moment là était que tout le projet se résume à suivre tel ou tel tutoriel pour avancer sans qu'on est la moindre ligne de code à écrire par nous-même. Heureusement ça n'a pas été le cas. Après un certain temps, on s'est habitué à l'environnement Unity et nous avons pu écrire des scripts par nous même. Je me rend compte après coup que c'est en fait le seul moyen d'avoir un jeu qui nous corresponde, car il est impossible de trouver tout ce que nous souhaitons faire en ligne. Ma peur était en quelque sorte infondé.

### **Les regrets :**

J'ai très peu de regret vis-à-vis de ce projet qui fut très plaisant à réaliser. Je regrette tout de même de ne pas avoir su être aussi régulier que j'aurais pu. Ainsi certaine semaine passait sans que j'ai avancé sur le jeu. Même si tous les éléments qui m'étaient attribués ont été fini dans les temps (en se dépêchant un peu avant les dates limites). Je pense que nous aurions pu implémenter quelques fonctionnalités supplémentaires en étant plus régulier.

# Conclusion

Finalement, Zéphyr s'est révélé être plus qu'un simple projet. C'est une source de fierté pour nous, et également de plaisir. En effet, malgré les nombreux problèmes rencontrés lors de son élaboration, ce projet a été très divertissant à réaliser et le résultat final est incroyable à nos yeux.

La majorité du groupe ne connaissait que très peu le monde du jeu vidéo. Ce projet a été un moyen de le découvrir plus en détail, de voir comment est construit un jeu vidéo et qu'elles sont les diverses étapes de sa confection. Nous avons pu vivre à une échelle beaucoup plus réduite ce type d'expérience, et avoir une meilleure idée de ce qui se cache derrière une équipe de développement.

Mais au delà, de juste savoir ce qu'est une équipe de développement, nous en sommes devenus une pendant la réalisation de ce projet. Nous avons donc développé des compétences en adéquation avec les besoins du travail en équipe. Nous sommes convaincus que ce projet nous a rendu plus mature dans le sens où nous sommes maintenant bien plus attentifs aux remarques des autres afin de rendre un produit final cohérent et qui sait plaire à tous. De plus, il est parfois nécessaire de laisser de côté sa vision ou ses envies personnels vis-à-vis du projet. Même si une idée nous paraît personnellement géniale mais qu'elle ne plaît pas au reste du groupe, il ne faut alors pas insister et la laisser tomber. Cela est absolument nécessaire pour que le groupe reste dans une ambiance saine et que le projet puisse ainsi avancer rapidement.

Sur un plan individuel l'équipe a également beaucoup évolué. En effet, le cadre scolaire auquel nous sommes confrontés depuis toujours nous apporte directement les informations qui nous sont nécessaires dans nos projets, contrôles ou plus généralement dans notre vie. Ici, de part le format du projet, nous avons été obligés d'aller chercher proactivement l'information. Cette habitude d'entreprendre est absolument indispensable pour devenir un bon ingénieur capable d'être autonome ou pour devenir entrepreneur. Au delà de la recherche d'informations, nous avons également développé nos compétences d'apprentissage. En effet, il a fallu maîtriser le logiciel Unity de fond en comble afin de produire un jeu qui nous correspond et cela avec une

contrainte temporelle assez importante, car nous n'avions que six mois pour terminer le projet.

Et le dernier point loin d'être négligeable issu de cette aventure réside dans la cohésion du groupe. Ce projet a été une occasion d'en apprendre plus d'un point de vue humain et social. Le travail en groupe a des avantages sur la répartition des tâches notamment, mais le résultat obtenu est souvent quelque peu différent de ce qui était attendu. Contrairement à ce que l'on aurait pensé, il n'y a pas eu de problème de communication. Peut-être est-ce avant tout parce que nous nous connaissions déjà avant de commencer ce projet. Toujours est-il que le travail en équipe a amené bien plus de points positifs que négatifs.

L'équipe Zelstrom

# Annexes

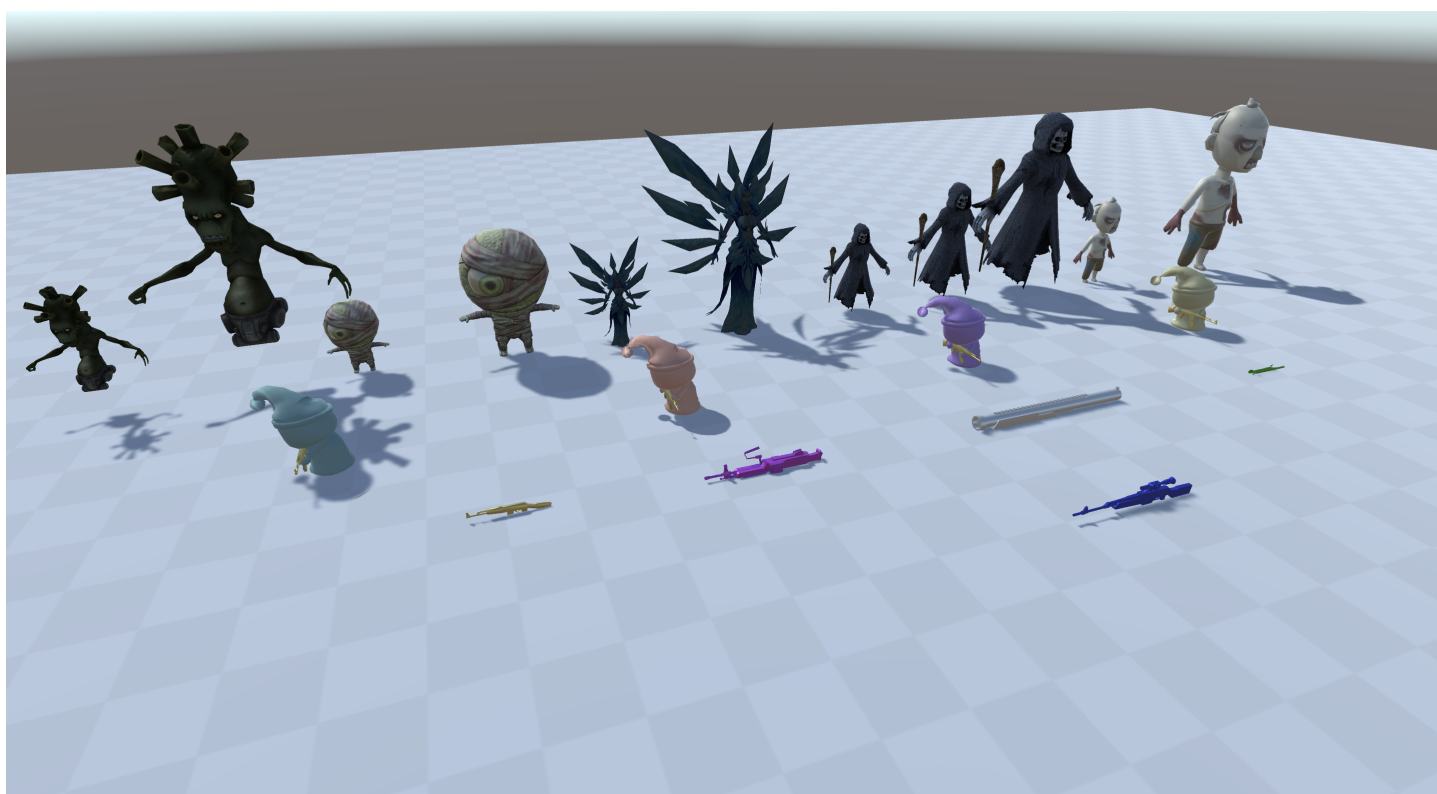
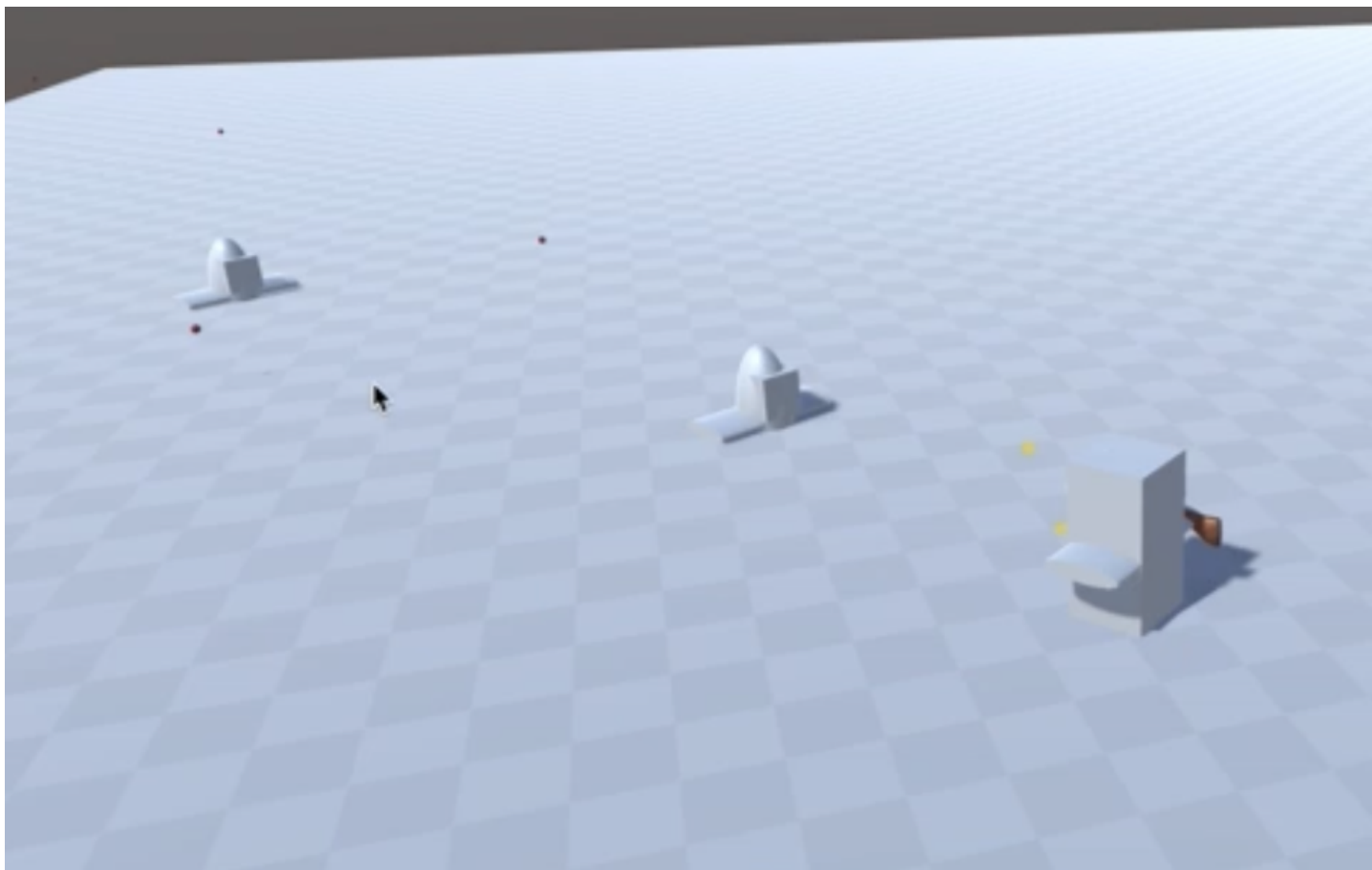


Photo de l'ensemble des belligérants de Zéphyr

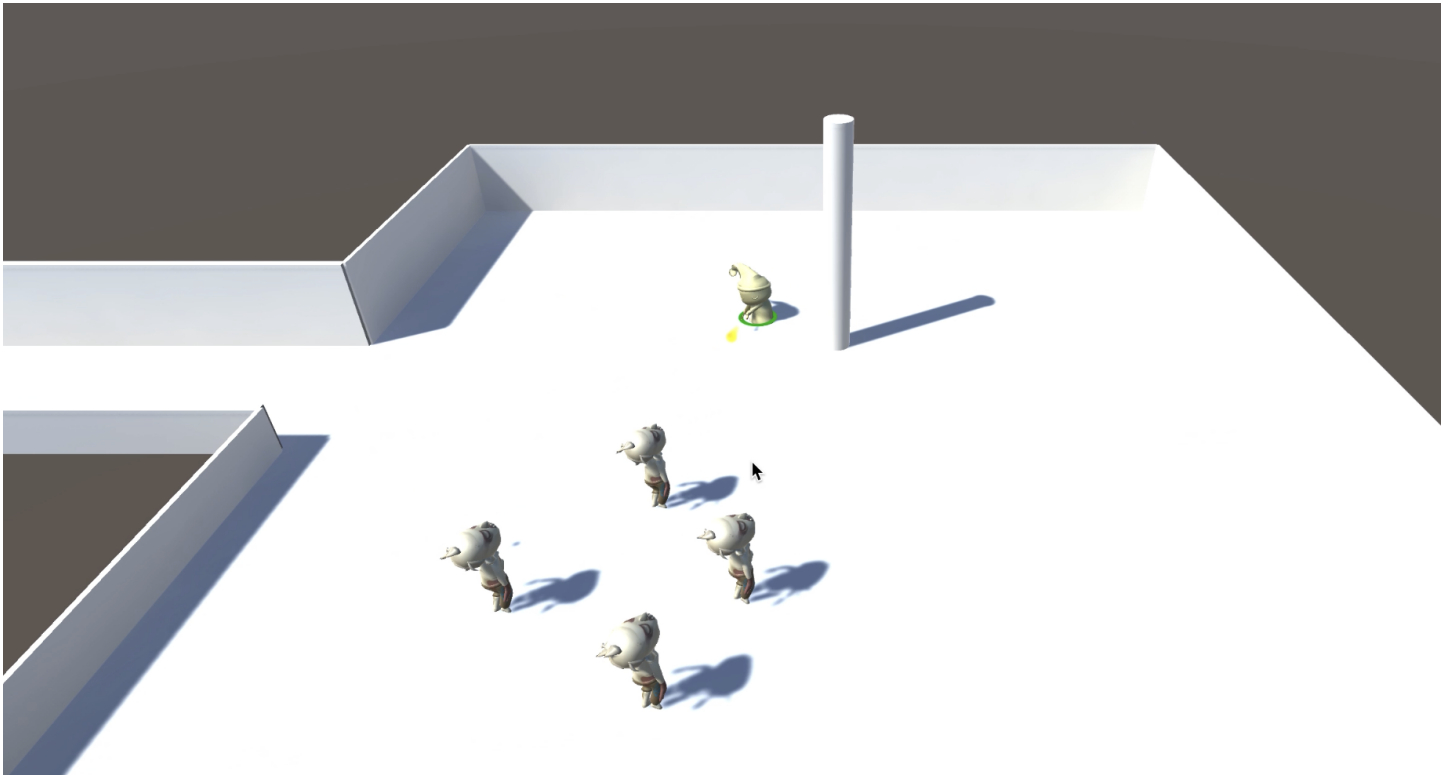
Au premier plan les armes (de gauche à droite AK, Machine Gun, Rocket, Fusil à pompe et Sniper), au deuxième plan les alliés (de gauche à droite Annie Mall, John Athan, Harry Cover, Thea Louest) et au troisième plan les ennemis (de gauche à droite le groupe Alien, le groupe Gnome, le groupe Ice, le groupe Mage et le groupe Zombi)



### Première version d'un jeu d'essai

Les ennemis à gauche poursuivent le joueur à droite qui leur tire dessus. Ils ont tous une forme géométrique.





### Deuxième version d'un jeu d'essai

Ici quatre ennemis zombis se dirigent vers le joueur qui se défend. Le décors est une maquette des cartes.