

Rapport de projet

La Confrérie

Pierre Kelbert, Morgan Vaterkowski, Léo Hantsch,
Sébastien Hoehn, Anass Kasmi

25 mai 2018



Table des matières

1	Introduction	3
2	Reprise du cahier des charges	5
3	Bonus	6
4	Détail du travail réalisé	7
4.1	Pierre	7
4.1.1	Introduction	7
4.1.2	Reprise des objectifs	7
4.1.3	Travail personnel	8
4.1.4	Bénéfices du projet	17
4.1.5	Conclusion	19
4.2	Morgan	20
4.2.1	Introduction	20
4.2.2	Travail personnel	20
4.2.3	Entraide	23
4.2.4	Conclusion	24
4.3	Léo	25
4.3.1	Introduction	25
4.3.2	Menu du jeu	25
4.3.3	Commandes/Périphériques	30
4.3.4	Niveaux	33
4.3.5	Conclusion	35
4.4	Sébastien	36
4.4.1	Introduction	36
4.4.2	Reprise des objectifs	37
4.4.3	Travail personnel	38
4.4.4	Bénéfices du projet	44
4.4.5	Conclusion	45
4.5	Anass	46
4.5.1	Introduction	46
4.5.2	Travail personnel	46
4.5.3	Bénéfices du projet	48
4.5.4	Conclusion	48
5	Travail en cours	49
6	Conclusion	49

1 Introduction

Avez-vous déjà entendu parler du jeu vidéo *Through*? Si ce n'est pas le cas, en plus de vous dire à quel point c'est dommage pour vous, nous vous encourageons davantage à lire la suite de ce document...

Parmi la riche diversité qui caractérise le monde du jeu vidéo, nous avons eu comme souhait de nous démarquer en créant un jeu mêlant aventure et énigmes. La grande question était alors : « Comment y parvenir? ». La solution à cette grande question n'était pas 42 pour une fois, mais plutôt le fait de combiner labyrinthes et système de portes. Imaginez-vous un labyrinthe. En trouver la sortie reste quelque chose de classique (attention, nous n'avons pas dit simple pour autant). Maintenant, si vous rajoutez des portes qui bloquent le chemin vers la sortie alors il s'agit tout de suite d'une autre histoire... C'est ainsi que le concept de *Through* est né!

Placé dans un labyrinthe, le joueur, pour parvenir à la sortie, se doit de trouver les boutons ouvrant les portes et ceci dans un ordre bien précis. Nous insistons sur l'ordre d'ouverture car, dans certains niveaux, le joueur n'est pas seul : il est accompagné d'un monstre prêt à tout pour le retrouver. Malheureusement pour le monstre, celui-ci n'est pas en capacité d'ouvrir ces fameuses portes, seul le joueur peut le faire par inadvertance!

Avant de débiter le développement, nous avons élaboré un cahier des charges donnant les objectifs et les délais de chaque membre du groupe. Si vous ne l'avez pas lu, ne vous en faites pas, nous allons reprendre pour vous les différents points dans la suite du document¹.

Avec nos objectifs en tête, nous avons travaillé à la conception de ce jeu pendant plusieurs mois. Travaillant parfois jour et nuit pour tenir nos délais, nous avons chacun donné le meilleur de nous-mêmes dans le seul but de proposer la meilleure expérience aux joueurs. Tout au long du développement, nous avons eu l'occasion d'apprendre beaucoup de choses. Tout le monde n'ayant pas forcément programmé de jeu vidéo par le passé, il a fallu apprendre de nouvelles méthodes et découvrir de nouveaux logiciels.

Malgré un travail qui nous aura demandé une attention particulière et beaucoup de patience, nous avons finalement rempli notre objectif et nous prendrons soin de vous le démontrer lors de la dernière soutenance.

1. Vous pouvez également retrouver tous les documents sur notre site throughgame.fr.

En effet, dans quelques jours sonnera pour nous la fin du projet avec la soutenance finale. Vous assisterez à la sortie, en avant-première s'il-vous-plaît, du jeu **THROUGH**...

Ne précipitons cependant pas les choses car avant de parler de la fin, il est temps pour nous de vous livrer tour à tour notre rapport personnel. Nous terminerons cette introduction en vous souhaitant d'ores et déjà une très bonne lecture².

2. Pour plus de confort, n'hésitez pas à prendre un café, la longueur du document vous le permet...

2 Reprise du cahier des charges

TABLE 1 – Avancement des tâches à réaliser

Tâches	Planning	Réalisé
Niveaux solo	100%	100%
Site internet	100%	100%
Design 2D	100%	100%
Système de portes	100%	100%
Tutoriel	100%	100%
Commandes	100%	100%
IA des monstres	100%	100%
TEX	100%	100%
Menu	100%	100%
Map multijoueur	100%	100%
Mini-map	100%	100%
Réseau	100%	100%

D'après le tableau 1, le projet est entièrement terminé. En effet, toutes les tâches seront réalisées à 100% pour la soutenance finale. Ainsi, le menu permettra au joueur d'accéder à n'importe quel niveau mais également aux options. Le joueur pourra apprendre les contrôles du jeu en traversant le tutoriel avant de se frotter aux 16 niveaux répartis en 4 mondes. Dans ces niveaux, le joueur pourra se déplacer et ouvrir des portes afin d'atteindre la sortie. Une mini-map sera à sa disposition dans un coin de l'écran pour l'aider à se repérer.

Definition 1 (Tutoriel)

Niveaux simples de début de jeu montrant les bases des commandes au joueur. Ces niveaux lui permettent d'apprendre à jouer.

Definition 2 (Mini-map)

Carte du monde en vue du dessus qui permet au joueur de voir où son personnage est situé dans l'environnement du niveau. La mini-map est placée dans un coin de l'écran de jeu.

Attention cependant au monstre qui rôde dans les couloirs dans le but de tuer le joueur. S'il devait réussir, le joueur devrait recommencer le niveau. Enfin, si le joueur souhaite s'amuser avec un ami, il pourra rejoindre la map multijoueur avec celui-ci afin de résoudre une énigme à deux.

Enfin, les documents liés au projet seront rédigés. En effet, vous êtes en train de lire le rapport de projet et le site internet³ pourra apporter un complément d'informations en différentes langues. Vous y retrouverez également tous les documents liés au projet.

3 Bonus

Grâce à l'intégration d'un membre supplémentaire au sein du groupe, le jeu sera rendu avec des bonus, c'est-à-dire du contenu supplémentaire par rapport à ce qui était annoncé dans le cahier des charges. En effet, si le joueur souhaite une difficulté accrue, un mode « Hardcore » lui sera proposé. De plus, un menu pause sera accessible en jeu depuis chaque niveau et permettra au joueur de revenir au menu ou de régler les paramètres.

3. Toujours throughgame.fr.

4 Détail du travail réalisé

4.1 Pierre

4.1.1 Introduction

Il y a quelques mois de cela, le développement d'un jeu vidéo était quelque chose de totalement inconnu pour moi. N'étant pas un grand joueur de nature, je ne me suis jamais intéressé à cette partie du monde informatique. Cependant, mon entrée à l'EPITA avec le projet de S2 m'aura permis d'y toucher et d'en découvrir les méthodes de travail. Sur toute la durée du projet, beaucoup de choses se sont passées, aussi bien au niveau du travail que du reste... Il est donc temps pour moi de faire le point et de reprendre tout ce que j'ai réalisé afin de conclure cette grande expérience.

Pour garder un cheminement rigoureux dans la présentation de mon travail, je vais commencer mon écriture par la reprise de tous mes objectifs tels qu'ils sont définis dans le cahier des charges. Ce n'est que dans un second temps, et à partir de cette base, que je vais me charger de développer les problèmes que j'ai pu rencontrer (ainsi que leurs solutions). Pour terminer, je prendrai le temps de détailler ce que le projet m'a apporté, ce que j'ai appris de cette épopée informatique et comment s'est passé le travail en groupe. En d'autres termes, pourquoi dire aux futures premières années : « C'est super! ».

4.1.2 Reprise des objectifs

Reprenons mes objectifs si vous le voulez bien. Pour ce faire, faisons un petit retour en arrière, soit lors de notre première réunion de projet le 12 janvier 2018. A ce moment là, nous étions fixés seulement sur la composition du groupe, tout le reste était à définir pour la création du cahier des charges. Les premiers moments du projet auront été un débat sur « que faire », une question qui finalement n'est pas si bête. Après un moment de discussion, nous avons décidé de partir sur ce qui est aujourd'hui un jeu de labyrinthe mêlé à un jeu d'énigmes. Pour plus de détails, je vous invite à reprendre l'introduction ou à vous rendre sur le site web⁴. Cette notion de labyrinthe, nous la retrouvons d'ailleurs dans le nom *Through*, à savoir passer au travers. A partir de ce premier point, nous avons fait un brainstorming pour lister toutes les choses à réaliser. Pour vous donner une idée, nous avons réussi à remplir complètement un tableau blanc (certes pas autant qu'en physique). Plus sérieusement, de cette liste nous avons ensuite distribué les tâches entre les membres du groupe en fonction des préférences de chacun mais également avec un souci d'équité de quantité de

4. Encore et toujours throughgame.fr.

travail. Par exemple, me concernant j'ai déjà réalisé plusieurs sites internet, c'est donc tout naturellement que je me suis proposé pour la création de celui du projet. En plus de cela m'est également revenue la responsabilité du mode multijoueur, la création des particules et la création du son. Une tâche supplémentaire, mais commune cette fois-ci, aura été la création des niveaux. Nous étions d'avis que pour offrir une plus grande diversité dans les niveaux et pour éviter une inégalité dans la quantité de travail, il était préférable que chacun crée des labyrinthes.

Une fois la répartition des tâches terminée, nous nous sommes empressés de tout noter dans ce fameux cahier des charges. Après quelques modifications, celui-ci a été approuvé déclenchant immédiatement notre mise au travail. Comme dit précédemment, le jeu vidéo n'est pas mon domaine de prédilection aussi, pour éviter toute mauvaise surprise, commencer le plus tôt possible n'était pas forcément une mauvaise idée.

Dans ce début de travail et avec le flou total auquel je faisais face, l'une des premières choses que j'ai pu faire a été de regarder la conférence « Bien démarrer son projet 2018 » (BDSP). J'y ferai fréquemment allusion et pour cause : je dois l'aboutissement de mon travail en grande partie grâce aux intervenants et à leurs explications ! Quoi qu'il en soit, cette présentation m'aura donné un premier élan pour la création de mes niveaux.

4.1.3 Travail personnel

Contrairement à mes rapports précédents, je ne vais pas développer mon travail dans un ordre chronologique car je risque de croiser les données beaucoup trop souvent. Sur le long terme ce serait aussi bien un inconvénient pour moi que pour vous chers lecteurs.

La création de niveaux, dans tout jeu me direz-vous, est un point essentiel qui demande une attention particulière. Il faut être d'une extrême rigueur si l'on souhaite que le joueur évolue dans les meilleures conditions possibles. Ainsi j'ai, et nous avons tous au sein du projet, commencé par des dessins sur papier. Créer un labyrinthe, c'est très simple : prenez le plan du campus de Paris et vous en aurez un. Mais créer un bon labyrinthe, dans le sens où il reste ludique pour le joueur, constitue une autre paire de manches !

Nous avons donc griffonné plusieurs prototypes, essayé plusieurs combinaisons différentes et en avons débattu lors de plusieurs réunions pour finalement arriver à une version finale. Chacun devait réaliser 4 cartes.

Une fois que nous étions tous satisfaits de ces cartes, nous avons pu nous concentrer sur leur réalisation en 3D.

Pour se faire, coté logiciel mon choix s'était porté sur *Blender* car dans la conférence BDSP, une attention supplémentaire avait été portée sur ce logiciel. Ceci contrairement à *Unity* qui n'était exploité que pour la partie « programmation ».

Blender reste pour moi un logiciel qui n'est pas évident à prendre en main en ce sens où les commandes ne sont pas intuitives... J'ai par le passé déjà eu l'occasion de faire de la modélisation 3D avec le logiciel *SolidWorks*, plus orienté sur un aspect « mécanique » que « artistique ». Je m'étais habitué aux commandes de ce logiciel, or *Blender* est pensé d'une manière totalement différente, ce qui m'a valu une longue prise en main.

Créer des niveaux sur *Blender* est une opération qui m'aura demandé plusieurs jours de travail, du moins pour les deux premiers qui ont été réalisés juste avant la première soutenance en mars. Malgré plusieurs échecs, je suis tout de même parvenu à respecter mes délais pour la première revue de projet.

Après la première soutenance, j'ai mis de côté la création des cartes au profit d'autres activités intéressantes (une prépa par exemple) et ce n'est que deux semaines avant la seconde soutenance, lors d'une semaine sans cours, que je me suis remis au travail. J'avais déjà raconté la petite histoire qui va suivre mais je me permets de la décrire à nouveau vu qu'elle a son importance dans ce rapport. Au jour près où j'allais poursuivre mon travail sur *Blender*, un ami avec qui j'avais discuté du projet m'avait proposé une petite démonstration sur la manière dont on crée des niveaux sur *Unity*. Cette personne avait déjà été amenée à créer un jeu sur *Unity* mais plutôt dans un cadre personnel. Après quelques heures d'explications, je reconnais avoir totalement changé d'avis en ayant compris à quel point cette méthode était bien plus adaptée à la création de terrains. En plus, comme une bonne nouvelle n'arrive jamais seule, les commandes sur *Unity* sont beaucoup plus intuitives ! Face à cela, j'ai finalement pris la décision de laisser tomber tout le travail réalisé sur *Blender* (ce qui se résumait à deux cartes sans texture) pour tout reprendre sur *Unity*. Reprendre et terminer les niveaux m'aura certes pris plusieurs jours complets mais correspond à ce que je souhaitais obtenir. En effet, ce long délai vient du fait que je ne me suis pas limité à la création de simples murs, mais plutôt à la création de cartes représentant des situations « réalistes » (un appartement,

un château, une supérette, ...). Or, ce genre de réalisation comporte beaucoup d'objets, de lumières, et ainsi de suite, qui embellissent l'expérience du joueur. Il m'a également fallu plus de temps car les calculs graphiques réalisés par l'ordinateur étaient simplement plus poussés comme je l'avais décrit lors du rapport précédent.

La création 3D de mes niveaux terminée, la dernière étape était celle de la mise en place des portes, car oui jusque-là quand je parlais de créer mes niveaux je ne parlais pas des portes. Pour cela, j'ai repris mes cartes (de zéro) pour travailler l'emplacement de chaque porte et de chaque bouton correspondant. C'est une partie qui demande un peu plus de réflexion car une erreur dans l'ordre d'ouverture des portes peut vite rendre le niveau inutilisable. Pour éviter ce genre d'erreurs, mes cartes, avant leur version finale, ressemblaient à ceci : un fléchage permet de vérifier qu'une porte ne bloque pas complètement le joueur (voir figure 1).

FIGURE 1 – Carte du niveau du château



Ce n'est que ces « tests » accomplis que j'ai réalisé la version finale, cette fois-ci sans flèche, mais avec des identifiants uniques. Vous pourrez constater que des boutons peuvent actionner plusieurs portes à la fois (voir figure 2).

FIGURE 2 – Carte du niveau du supermarché



Pour créer mes plans, j'ai utilisé le logiciel de Google : *Drawn.io*, qui permet d'avoir un rendu très propre. Je vous laisse en juger.

Avec ces différentes cartes en main, il ne me restait plus qu'à intégrer mes portes dans mes niveaux. Bien qu'initialement il s'agissait de mon travail, Morgan a eu la gentillesse de s'en charger à ma place, ayant déjà l'habitude de gérer l'outil d'animation sur *Unity*.

Passons si vous le voulez bien à une seconde partie très intéressante : le multijoueur. Encore une fois j'étais face à un mur, et encore une fois la conférence BDSP aura changé la donne. Lors de cette présentation, les intervenants ont utilisé le système *Photon Netwok Unity* (PNU) pour faire fonctionner leur exemple. Convaincu très rapidement de la facilité de mise en place, je me suis orienté vers cette solution pour mettre notre jeu en réseau. Lors de la première soutenance, mon objectif fixé par le cahier des charges était de commencer le réseau. A ce stade de progression générale du projet, je ne pouvais malheureusement qu'avancer jusqu'à la création d'un compte sur l'interface en ligne de Photon. En effet, pour mettre en réseau un jeu il faut avant tout ... un jeu. Même si l'idée paraît frivole, elle ne l'est pas tant que ça. La première soutenance n'était donc pas sujette à beaucoup de détails intéressants sur cette partie. Soit ! Ce n'est que par la suite, lorsque les autres membres du groupe ont avancé sur le jeu, que j'ai pu me mettre sérieusement au travail. Quelques jours avant la seconde soutenance, je me suis repenché sur le fonctionnement de PNU à travers les documentations officielles (en plus de BDSP). Après quelques heures de travail sur le script des joueurs, j'ai pu passer une première étape : à savoir faire apparaître deux personnages sur une même room.

Bien que cela constituait déjà une énorme joie (pour le développeur), l'objectif tel qu'il est défini n'était pas encore atteint ! Si nous reprenons les termes du multijoueur, nous pouvons constater qu'il m'est demandé de faire apparaître au maximum deux personnages sur une carte et à des endroits différents. La seule chose qu'il me reste à faire est donc de me remettre au travail...

Je vais dès à présent m'attarder sur une partie qui n'a rien à voir avec du code : la charte graphique ! Comme nous utilisons $\text{\LaTeX}$, j'en profite pour faire une bulle de définition...

Definition 3 (Charte graphique)

Document de travail qui contient l'ensemble des règles fondamentales d'utilisation des signes graphiques qui constituent l'identité graphique d'une organisation, d'un projet, d'une entreprise...

Même s'il ne s'agit pas d'un élément inscrit dans le cahier des charges, je me suis tout de même permis de m'attarder sur l'aspect graphique du projet. J'entends par là, en plus de la charte, la création d'un logo, de présentations *PowerPoint* et ainsi de suite. Le logo a été créé avec la même

idée que le choix du nom, c'est-à-dire le fait de passer au travers d'un labyrinthe. Pour représenter cela, je me suis tourné vers des flèches à double sens qui « englobent » le nom du projet comme vous pouvez le constater au bas de la page. Les logos sont dans la majorité des cas réalisés sous format vectoriel par soucis de propreté. Ainsi, pour notre logo je me suis aidé du logiciel *Inkscape*, open source en plus, qui a fait parfaitement l'affaire ! Pour les rares personnes qui ne savent pas ce qu'est une image vectorielle, je me contenterai de dire qu'il s'agit d'une image basée sur un tracé mathématique. En soit, on peut agrandir l'image à l'infini sans jamais voir de pixel.

Definition 4 (Open-source)

Méthode d'ingénierie logicielle qui consiste à développer un logiciel, ou des composants logiciels, et de laisser en libre accès le code source produit.

Les couleurs du logo ont quant à elles été choisies bien avant grâce à l'outil en ligne *Adobe Color Cc* et de sorte à créer un dégradé pour le moins sympathique. Cependant, ne cherchez aucune symbolique derrière les couleurs, la seule contrainte était « il faut que ce soit joli... », rien de plus.

Venons-en à ma partie préférée : le site web !

N'étant pas nécessaire lors de la première soutenance, et ayant déjà assez de travail sur les épaules, je n'ai pas débuté avant celle-ci à l'exception de l'enregistrement du nom de domaine. Notre jeu s'appelle bien *Through*, mais malheureusement le nom de domaine associé étant déjà utilisé, nous nous sommes portés sur une variante à savoir *throughgame.fr*. Petite parenthèse pour dire que nous en avons profité pour créer des adresses mail associées au projet. Par exemple, en cas de réclamation, n'hésitez pas à m'écrire à l'adresse sebastien.hoehn@throughgame.fr. Je referme cette parenthèse.

Ce n'est que par la suite donc, lors de la semaine des oraux *Advance* pour vous donner un repère temporel, que je me suis occupé sérieusement du site. D'un point de vue hébergement, je ne suis pas parti sur un système en local, bien que tout à fait possible, pour des raisons de bande passante et de fiabilité. Ayant déjà un contrat depuis 2 ans chez un hébergeur dont je ne citerai pas le nom⁵, j'ai tout simplement associé notre nom de domaine

5. 1&1 Internet

à mon hébergement existant. throughgame.fr est donc hébergé sur une solution payante. Niveau code, pour créer un site internet, il faut reconnaître que beaucoup de possibilités se sont offertes à moi. Entre tout faire soi-même, ou utiliser un framework, mon choix s'est finalement orienté vers l'utilisation d'un CMS (Content Management Software) à savoir le plus que célèbre : *Wordpress*. Cela peut paraître comme étant une solution de facilité, mais si j'ai effectué ce choix c'est avant tout pour assurer une grande stabilité au site et aussi une grande sécurité. Nous ne sommes jamais à l'abri d'un petit piratage d'un groupe adverse. *Wordpress* est donc effectivement un système extrêmement fiable. A titre d'exemple, il est tout de même utilisé par 74,6 millions de sites dont celui de *The Walt Disney Compagny*, *UPS*, *Microsoft/Windows* (blog) et *Sony*.

Definition 5 (Framework)

Ensemble d'outils constituant les fondations d'un logiciel informatique ou d'applications web, et destiné autant à faciliter le travail qu'à augmenter la productivité du programmeur qui l'utilisera.

Definition 6 (CMS)

Programme web destiné à la création et la gestion de sites internet.

Pour le site de *Through*, même si *Wordpress* se charge du fond, la réalisation graphique a été composée par mes soins. Le thème actuellement en place ne vient pas d'un éditeur mais s'inspire d'un thème parent coté scripts PHP et d'un travail personnel côté mise en forme. Je ne dirais pas que créer un site est un travail compliqué mais plutôt qu'il s'agit d'un travail long. En effet, cela ne se résume pas à une seule tâche mais plutôt en une multitude qui demande beaucoup de temps (CSS, création d'images, écriture de contenu, etc.).

Le CSS par exemple, responsable de la mise en page, n'est pas de nature très coopératif. Il s'agit vraiment d'un monde de la programmation à part qui demande parfois beaucoup de temps pour arriver à faire des choses simples. Je tiens d'ailleurs à remercier mon partenaire stackoverflow.com.

Il m'a fallu beaucoup de patience pour obtenir un site en accord avec la charte graphique établie (couleurs, police, ...) mais je trouve qu'il ne s'agit

pas de temps gaspillé au vu de l'importance de la chose. Je vous invite à vous rendre sur le site ⁶ pour vous forger un avis sur le sujet.

Sortons un peu du cadre de la programmation pour parler un petit peu plus marketing, qui malgré tout représente un point très important dans la conception d'un site web. Prenons comme exemple la page d'accueil. Faisant office de vitrine pour notre jeu, sa conception a fait l'objet d'une attention toute particulière. On y retrouve un texte introduisant le jeu *Through* mais aussi un lien de téléchargement pour y jouer. Pour attirer l'œil du visiteur et pour véhiculer des phrases clés, j'ai pris la décision d'ajouter des images « full_width » (voir figure 3). Ces images ont un concept simple : véhiculer une idée avec un style épuré. En plus d'accrocher le regard, cela ajoute de la couleur et rend la lecture plus agréable.

FIGURE 3 – Image « pleine largeur » sur le site throughgame.fr



Le site ne comporte évidemment pas qu'une page d'accueil. On y retrouve aussi une page avec l'avancement des différents points du cahier des charges, une description détaillée de l'équipe et un historique. Juste après la seconde soutenance, je me suis permis de rajouter une page « Documents » qui rassemble tous les fichiers réalisés depuis le début du projet. Cela inclut le cahier des charges, les différents rapports/plans de soutenances et même nos présentations *PowerPoint*.

Par la suite ont aussi été rajoutés les différentes versions du jeu.

Un dernier point sur le site internet : the different languages, of course !

Lorsque nous avons appris que la présentation d'un site bilingue était encouragée, nous avons tout de suite débuté la mise en place d'une telle

6. ... throughgame.fr

structure. Encore une fois, le fond étant géré par *Wordpress*, la mise en place de pages dans une seconde langue a été facile sur le plan technique. La rédaction quant à elle a été assurée par Léo, bien meilleur que moi dans le domaine.

Le travail que je vais décrire dans les prochaines lignes (sons/particules) n'a été réalisé qu'à la fin du projet car il ne s'agit pas de quelque chose d'important sur le plan technique. Se retrouver avec du son dans un jeu est bien évidemment plaisant mais pas d'une nécessité première.

Cette partie, à l'instar du mode multijoueur, m'a été attribuée sans raison particulière au sens où je n'ai jamais réalisé ce genre de travail auparavant. Cependant, contrairement à la mise en réseau, les particules ne représentent pas la même importance dans la conception du jeu, n'étant que « fioritures ». Là où je veux en venir, c'est que pour ce rapport de projet je ne suis malheureusement pas en mesure de développer sur cette partie. En effet, à l'heure où j'écris ces lignes nous sommes à quelques jours des partiels, j'ai donc dû choisir entre mettre de la décoration dans notre jeu ou réussir mon année. Autant vous dire que je n'ai pas tergiversé longtemps pour finalement repousser à plus tard la mise en place des particules. Je tiens à préciser que je ne laisse pas tomber cette partie pour autant, vous aurez donc droit à de belles particules lors de notre soutenance, n'ayez crainte ! Pour vous donner une idée, je compte réaliser quelques animations au niveau des feux dans le niveau du château fort ou encore réaliser quelques animations de poussière lors du déplacement des portes. Comme vous l'aurez constaté, le nombre de possibilités est élevé ! « Mais comme dirais l'autre, ceci est une autre histoire ! »⁷

Je tiens à préciser, avant de poursuivre, que lors des choix en termes de musiques, j'ai été particulièrement vigilant quant aux droits d'auteurs. Les musiques/bruitages présents dans notre jeu sont donc soit libres de droits soit « home-made ». Poursuivons...

Bien en proie au changement au vu de la date à laquelle je rédige ces lignes (voir paragraphe précédent), je souhaitais préciser que tous les bruitages utilisés proviennent de bibliothèques en ligne, nous n'avons pas réalisé d'enregistrements nous-même. Là où pour la musique je comptais faire de même, j'ai finalement changé mes plans à la suite d'une proposition que m'a faite Léo. Celui-ci jouant de la guitare, il m'a proposé de créer de petites compositions, une très bonne idée selon moi pour apporter de l'originalité dans le jeu. Nous n'avons pas encore réalisé ces enregistrements donc je

7. © Krisboul

ne saurais malheureusement pas, encore une fois, vous développer davantage cette partie.

4.1.4 Bénéfices du projet

Maintenant que j'ai pris le temps de parler de ce que j'ai réalisé, je vais développer autour de ce que le projet m'a appris. Évidemment, je ne vais pas reprendre les points précédents, mais plutôt des points qui ne peuvent être abordés dans la partie réservée à mon travail.

Le projet ne se résume pas que à mon travail mais également à tout ce qui l'entoure, à savoir la méthode. Ayant également beaucoup appris de ce point de vue là, je profite de ce rapport final pour vous en faire part.

Première chose, les outils. Il m'est déjà arrivé bon nombre de fois de réaliser des projets informatiques mais jamais en groupe, l'occasion a donc été pour moi de découvrir de nouveaux outils collaboratifs. Un des meilleurs exemples est celui de la gestion des fichiers.

Lorsque vous êtes à plusieurs, il faut que tout le monde puisse avoir accès au projet sans pour autant risquer d'anéantir le travail d'un autre en modifiant le même code. Une solution à ce problème, vous l'aurez deviné, c'est *Git*. Autant être honnête, nous n'avons pas tergiversé de longues heures sur la nécessité ou non d'utiliser ce logiciel tellement il est recommandé par les « anciens » et par les développeurs de manière générale.

Nous avons donc dès le début mis en place ce système à travers la forge *GitLab*.

Pour ceux qui pensent que je parle de métallurgie, permettez-moi de vous donner la définition suivante :

Definition 7 (Forge)

L'objectif d'une forge est de permettre à plusieurs développeurs de participer ensemble au développement d'un ou plusieurs logiciels, le plus souvent à travers le réseau Internet.

Là où *Git* (seul) permet de suivre les versions d'un même fichier, *GitLab* (et *GitHub* aussi d'ailleurs) permet de le faire en réseau.

J'avais déjà entendu parler de ce logiciel mais que très vaguement et sans jamais m'y intéresser davantage. Ce n'est donc que dans le cadre du projet que j'ai appris à m'en servir. *Git* n'est pas un logiciel des plus attrayants. Habitué aux interfaces graphiques comme la plupart des utilisateurs lambdas, je n'étais pas très à l'aise au début. Je dois la prise en main de ce logiciel avant tout aux autres membres de La Confrérie m'ayant montré les différentes commandes. Après ce temps d'adaptation passé, j'ai vite compris à quel point il s'agit d'un système de collaboration indispensable. A peine quelques jours après l'installation de *Git* sur mon ordinateur j'ai été confronté à un bug sur *Unity*. *Git* m'a alors permis de restaurer une ancienne version ce qui aura été beaucoup plus rapide que de trouver l'origine du problème. Je ne peux que souhaiter bonne chance à ceux qui auront préféré la clé USB à *Git*...

Si je devais parler de la gestion et la répartition du travail, je dirais que je ne m'attendais pas à une si faible répartition chronologique. Lorsque nous avons écouté les anciens parler de leurs projets, ceux-ci ont toujours dit qu'ils avaient travaillé peu avant les soutenances sous forme de « rushs ». Nous pensions, à tort, que ce ne serait pas notre cas, mais finalement nous n'avons pas échappé à ces périodes de code intensives. L'explication est simple, nous avons une quantité de travail constante pendant les cours et ce n'est qu'avant les soutenances que le rythme diminue. Ainsi, même si une partie du travail est effectuée en amont, le plus gros n'est réalisé que juste avant les soutenances, quand il n'y a plus de cours... Cela ne nous a cependant pas empêché de tenir nos délais pour chaque soutenance. Même s'il m'arrivait d'avoir quelques décalages avec le planning du cahier des charges (ce qui reste normal), j'ai tout de même tenu tous mes délais ainsi que tous mes objectifs.

Une autre chose que j'ai pu apprendre durant ces longs mois n'est autre que l'écriture de documents « formels ». Écrire des cahiers des charges, des rapports de soutenance et ce genre de choses ne faisait pas partie de mes habitudes. Mes débuts ont été laborieux mais comme la plupart des choses, un peu d'entraînement permet d'y arriver haut la main ! Écrire, nous savons tous le faire, mais rédiger un document agréable à la lecture, avec un langage plus soutenu, reste d'une tout autre compétence. Je n'aurais pour exemple que le fait d'être parvenu à intégrer des mots tels que « fioritures » ou encore « frivole » dans un document tel que celui-ci. Abracadabrantique !

Cependant, cette compétence n'est rien sans la capacité de créer du contenu. Je m'explique. Dans le cadre du projet, il nous est demandé de faire des rapports, mais nous avons malheureusement un nombre de pages imposé.

Il faut donc savoir réaliser une quantité importante de contenu avec une quantité d'information moindre. Si vous en doutez, relisez ce paragraphe dédié à l'écriture de paragraphes...

4.1.5 Conclusion

Ce projet reste pour moi une belle expérience car, au-delà de l'aspect informatique, il y a également l'aspect du travail en groupe. Lorsque nous sommes à plusieurs, et ce n'est pas un scoop, la gestion du travail n'est absolument plus la même. Il faut travailler, certes, mais de manière coordonnée et, même si sur le papier cela paraît facile, appliqué à la réalité ça n'est pas la même chose.

L'importance de la communication, nous l'avons heureusement comprise très vite. Pour preuve, nos réunions étaient très régulières : tous les jeudis étaient synonymes de réunion. Sans pour autant avoir énormément de choses à raconter, le plus important était de savoir où en étaient les autres.

Pour quelqu'un qui n'avait jamais pris soin de comprendre le monde du jeu vidéo, comme je dit en introduction, j'ai énormément apprécié le fait de travailler sur ce jeu et cela constitue pour moi une première victoire. Même si nous avons rencontré des soucis et que ça n'a pas été toujours évident à gérer, je reconnais que je ne suis pas à plaindre ! Nous n'avons pas rencontré de gros problème remettant en question notre objectif, mais seulement de petites « coquilles » auxquelles nous avons su faire face. Après relecture de toutes mes notes vis-à-vis de ce que j'ai traversé avec mon groupe, je ne peux qu'être fier d'avoir travaillé à l'élaboration de ce projet et ce dans une si bonne entente !

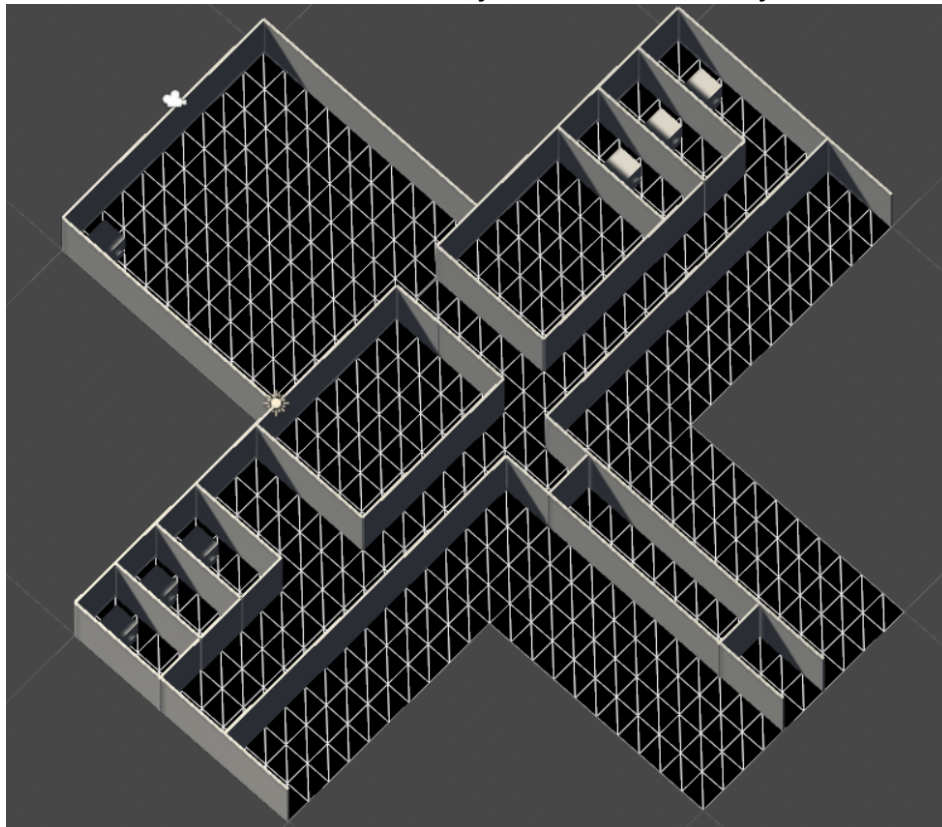
4.2 Morgan

4.2.1 Introduction

Je vais donc, à mon tour, décrire ce que j'ai fait tout au long du dernier semestre dans le cadre du projet en groupe. Depuis le début de cette aventure, j'ai appris beaucoup de choses sur les gens qui m'entouraient. Grâce à ce travail, j'ai pu comprendre comment les jeux vidéo, qui occupent une grande partie de mon loisir, fonctionnent. Cela a donc élargi ma tolérance concernant certains aspects des dysfonctionnements des jeux vidéo et, ayant un esprit très critique et très taquin, j'apprécie deviner (ou essayer au moins) comment certains éléments que je rencontre dans mes loisirs sont créés et comment ils sont animés (ragdolls, booléens, ...).

4.2.2 Travail personnel

Nous avons pour débiter décidé que tous les membres du groupe devaient dessiner des niveaux et les transposer dans *Unity* pour que tout le monde ait les bases. Cependant, la majorité des membres du groupe a utilisé *Blender* pour créer ses niveaux. Personnellement, je me suis contenté de *Unity* (évidemment j'ai dû les refaire, puisque je les ai réalisés bloc par bloc et non mur par mur) (voir figure 4). Ensuite, chaque membre du groupe s'est spécialisé dans la partie qui lui avait été attribuée.

FIGURE 4 – Niveau multijoueur réalisé sur *Unity*

Dans ce projet, je me suis donc chargé des portes et de leur fonctionnement. Beaucoup de choses les concernant ont été abandonnées pour cause de charges de travail démesurées par rapport à l'apport de l'élément en question. De plus, au début du projet, nous avons eu du mal à nous mettre d'accord sur certaines normes (qui sont parfois non respectées dans certains niveaux) concernant les dimensions des niveaux (ex : largeur des couloirs, des murs, ...). De plus, certaines animations concernant les boutons étaient totalement superflues, puisqu'au rendu de la première phase de test elles ne donnaient pas d'information concernant l'état du bouton (activé/désactivé) puisque le cylindre à l'intérieur du cube ne bougeait que de dix centimètres et que les deux états étaient trop peu distincts. Cependant, déplacer le cylindre de plus de dix centimètres rendaient la chose démesurée d'après des avis extérieurs.

Pour commencer à construire ce système, je me suis placé dans une scène vide qui serait importée dans chacune des scènes afin de placer les boutons et les portes aux endroits prévus. Mais c'était sans compter sur le

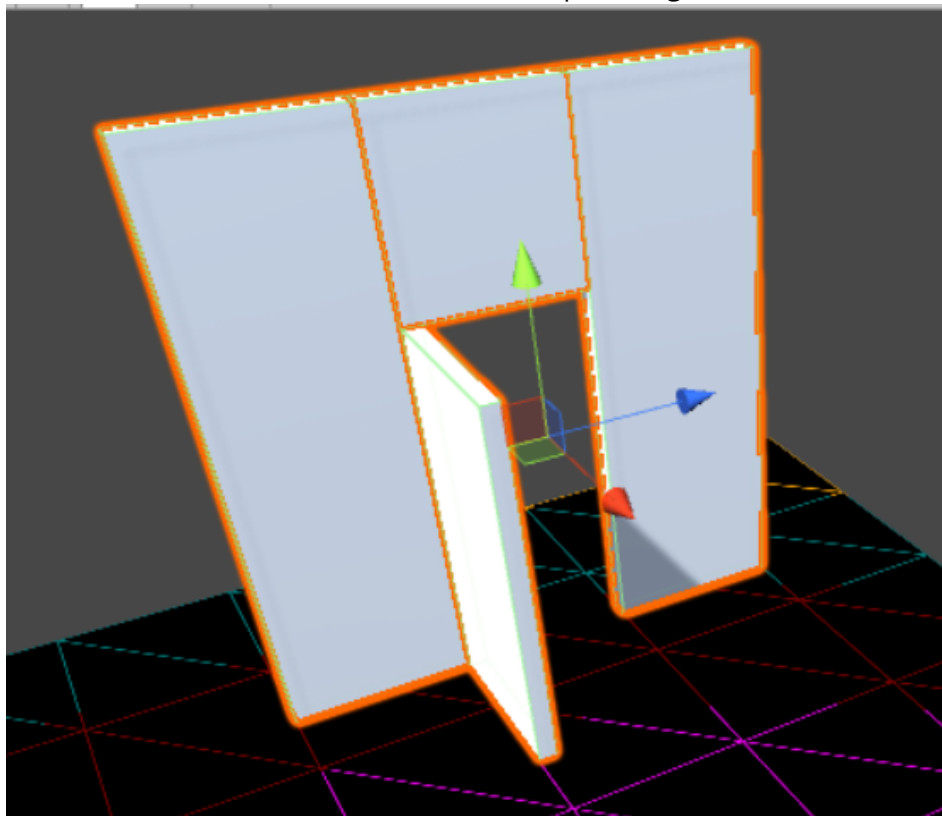
fait que mes animations reposent sur des coordonnées et des rotations et non sur des vecteurs. A la phase de test des autres niveaux, les portes se téléportaient aux coordonnées de leur scène de départ, soit pas du tout où nous les voulions. Sébastien et moi avons donc récréé chaque porte et son animation propre. C'est ainsi que voyant la fin de notre travail approcher, j'eut une révélation : il me semble avoir trouvé comment réaliser la porte universelle pour tous les niveaux. Il suffit de placer la porte dans un nouveau « game object », ce qui donne une nouvelle origine aux objets placés à l'intérieur de celui-ci. Nous n'aurions plus qu'à dupliquer autant de fois que nécessaire ce game object et de le placer aux endroits où nous en avons besoin. Mais comme sous-entendu plus haut, nous n'avons pas utilisé ce stratagème, nous ne sommes donc pas sûrs que cette technique fonctionne à 100%.

Nos commandes étant incomplètes au moment où j'ai terminé les portes, nous avons décidé de remplacer la touche « E » par un collider, et cela a été possible en grande partie grâce à Maxime. Et à partir de ce moment, mes portes étaient fonctionnelles à 100% (voir figure 5).

Definition 8 (Collider)

Le masque de collision (en français), aussi appelé hitbox, est la zone sensible d'un élément du jeu aux autres éléments de ce jeu.

FIGURE 5 – Porte réalisée par Morgan



4.2.3 Entraide

Je vais, dans ce paragraphe, souligner l'importance de l'entraide entre les groupes. Personnellement, j'ai surtout travaillé avec Maxime et Zoé du groupe Awesome Seals. Ils nous ont apporté beaucoup et nous leur avons apporté tout autant. En plus du soutien matériel, ils nous ont apporté du soutien moral. J'avais personnellement tendance à oublier que nous n'étions pas les seuls à travailler sur un projet complexe, et que nous n'étions pas les seuls à devoir gérer des délais. De plus, l'arrivée d'Anass dans notre groupe de projet a amené une bouffée d'air frais. Sébastien et moi nous sommes occupés d'intégrer Anass au mieux dans le groupe. A cause du retour que nous avons de son ancien groupe, nous avons été très surpris, et dans le bon sens du terme. En effet, au bout de trente minutes, Anass savait d'où nous étions partis et où nous allions. Et après une heure, il est revenu vers moi avec une idée et m'a demandé si c'était réalisable et si on pouvait l'intégrer dans notre jeu. Son idée n'était autre que de générer des labyrinthes procéduralement. En plus du fait que se soit une idée superbe, elle m'a été très utile pour créer les deux derniers labyrinthes que j'avais perdus (on ne

sait où).

Definition 9 (Génération procédurale)

Création de contenu numérique à une grande échelle et de manière automatisée répondant à un ensemble de règles définies par des algorithmes.

4.2.4 Conclusion

Comme beaucoup, à posteriori je changerais beaucoup de choses comme les portes et les relations que j'ai eues avec les autres personnes. Sauf pour les relations que j'ai eues avec les personnes du groupe Awesome Seals, je n'y changerai rien. Ce projet m'a appris beaucoup de choses, non pas uniquement sur *Unity* ou *C#*. Mais sur les gens, notamment qu'il ne faut pas s'arrêter à ce que les gens disent ou à ce qu'ils semblent être. Si on prend l'exemple de Maxime, pour moi à première vue il me semblait qu'il s'agissait d'une personne qui n'est jamais d'accord avec personne et du coup un personnage ennuyeux et redondant de mon point de vue. Mais en discutant sur divers sujets avec lui lors de repas le soir, j'ai découvert qu'il avait une réflexion très poussée et intelligente sur bien des sujets et qu'en plus il est une personne qui se donne du mal pour faire les choses bien. De plus, je me suis donc lié d'amitié avec lui et ceci est bénéfique (surtout) pour moi comme pour lui.

4.3 Léo

4.3.1 Introduction

Cette année, dans le cadre de notre module d'informatique pratique, notre classe a dû constituer de petits groupes de quatre (ou cinq) personnes avec le seul et même objectif : réaliser un projet en informatique. Au début, je dois admettre que j'appréhendais un peu ce projet, et cela pour les raisons suivantes : allais-je trouver un groupe avec des camarades sympathiques (et travailleurs)? Arriverais-je à me débrouiller tout seul en codage sans trop solliciter l'aide de mes camarades? Aurais-je un niveau suffisant en programmation?

Toutes ces interrogations disparurent quand j'ai rencontré ceux qui allaient devenir mes collègues de travail et amis (pour le meilleur et pour le pire) : Sébastien, Morgan, Pierre, et dernièrement Anass. En dehors du fait que les élèves cités précédemment soient tous sérieux et impliqués dans leur travail, j'ai d'autant plus été rassuré car nous avons tous immédiatement accrochés à l'idée de construire un labyrinthe rempli de portes à déverrouiller. Mais ce qui m'a surtout semblé génial au début de ce projet est le fait que, lors de la répartition des tâches, chaque membre du groupe était volontaire et motivé par les différentes missions à remplir. Moi qui avait peur au début de ne pas trouver de tâche que je ne puisse faire tout seul, je suis ressorti content de cette répartition des tâches, car j'ai moi-même pu choisir des missions qui me semblaient atteignables, à savoir : la création et la modélisation des niveaux en 3D (tâche commune à chaque membre du groupe), la réalisation du menu du jeu ainsi que l'implémentation des commandes du jeu (couplée avec la gestion de périphériques).

Chaque membre ressortit donc satisfait de cette répartition des tâches. Le projet *Through* pouvait donc enfin commencer!

4.3.2 Menu du jeu

Lors de la répartition des tâches, je me suis tout de suite proposé pour réaliser le menu de notre jeu, car ayant déjà de l'expérience dans le domaine. En effet, j'ai choisi l'année dernière l'option Informatique et Science du Numérique (ISN). Dans cette matière, j'ai codé avec mon collègue un jeu, et plus précisément une bataille navale, que nous avons dû ensuite présenter devant un jury dans le cadre de l'épreuve du baccalauréat. Et c'est moi qui me suis chargé de l'implémentation du menu de cette bataille navale.

Cependant, il y a une différence fondamentale entre la réalisation du menu de la bataille navale et celui de *Through*. En effet, pour le menu de

mon projet d'ISN, j'ai choisi de le coder en CSS3 et en HTML5 puisque celui-ci devait être accessible depuis Internet. Alors que pour le menu de *Through*, j'ai décidé d'utiliser le logiciel *Unity* avec lequel j'étais de toute manière amené à me familiariser puisque les finitions de mes niveaux de jeu devaient être faites sur *Unity*. C'est ainsi que je me lançai dans un nouveau défi : celui de réaliser un menu avec un logiciel que je ne maîtrisais pas à l'époque.

La première étape dans l'implémentation de mon menu était de réaliser au brouillon un « cahier des charges », en consultation directe avec mes camarades, afin de fixer dès le début les éléments qui devaient absolument figurer sur ce menu. En dehors des spécificités techniques, il était important pour moi de discuter avec mes camarades pour connaître l'architecture globale que devait adopter ce menu. Après quelques discussions, l'architecture choisie pour notre niveau fût la suivante. Au début, le panel de départ, contenant trois boutons : un pour choisir les niveaux « Choose levels », un autre pour accéder aux options du jeu « Options », et enfin un dernier bouton pour quitter le jeu « Quit Game ». Puis, les scènes mettant en avant les différents « mondes » de chaque joueur. Chacune de ces scènes présente quatre boutons (un pour chaque niveau), un bouton pour revenir à la scène précédente et un dernier pour passer à la scène suivante. Enfin une scène pour le tutoriel, et une dernière pour le mode Hardcore.

La seconde étape pour implémenter ce menu fût donc de choisir la manière d'apprendre à utiliser *Unity*. Plusieurs choix s'offraient à moi : d'abord, la documentation *Unity* disponible sur le site homonyme. Après quelques recherches, j'ai abandonné cette méthode de travail car trop coûteuse en temps selon moi. C'est pourquoi je me suis tourné vers des sites qui proposaient des guides vidéo complets pour implémenter son menu sur *Unity*. Et c'est finalement cette méthode que j'ai décidé d'adopter, justement parce que je trouvais que ces guides étaient complets, dans la mesure où l'auteur prenait le temps d'expliquer et de détailler les classes utilisées.

Après cela, j'étais prêt à implémenter le menu. Rapidement, je me rendis compte que l'implémentation de ce menu allait être divisée en deux grandes parties. Tout d'abord la création et le design des panels, c'est à dire des surfaces colorées sur lesquelles sont disposées les boutons cliquables par le joueur. La seconde partie du travail consistait à relier les panels, ou plutôt les scènes sur lesquelles figurent chaque panel.

La création des panels était la partie qui nécessitait le plus de temps. En effet, les étapes étaient assez nombreuses et répétitives. Pour commencer, il faut créer un panel. Ensuite, il faut créer un bouton cliquable, sans oublier

de modifier le texte de ce bouton. Suite à quoi on crée un canvas dans lequel on place les différents boutons précédemment créés. Enfin, on glisse le code C# reliant les deux scènes sur chaque bouton cliquable afin qu'ils deviennent effectifs.

Definition 10 (Canvas)

Le canevas (en français) est la zone dans laquelle tous les éléments d'interface utilisateur doivent être placés.

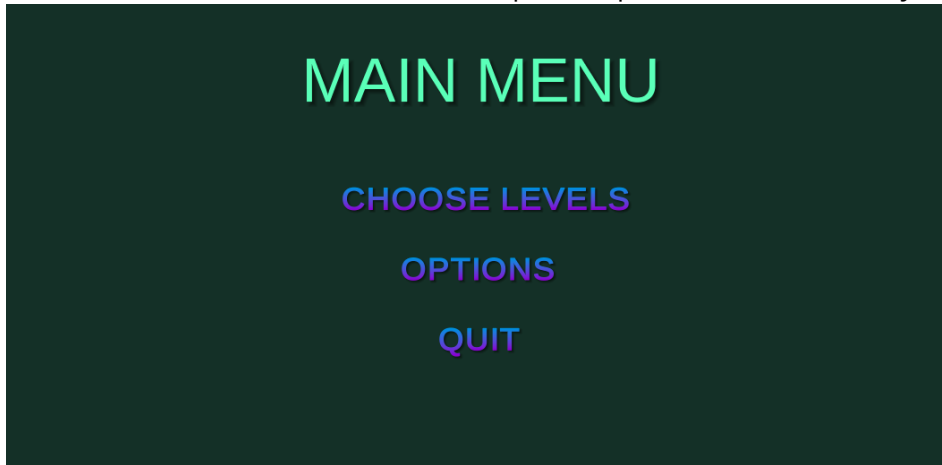
Le problème, c'est que je viens de vous citer les étapes de surface. En réalité, ce qui me prit beaucoup de temps dans la création de chaque panel était les petits détails, nécessaires au bon déroulement de la création de la scène. Par exemple, il est nécessaire de nommer chaque objet présent sur la scène pour ne pas se perdre dans tous les objets présents sur celle-ci. Autre chose, les différentes couleurs du bouton (sans que la souris ne le survole, lorsque la souris survole le bouton et lorsque le joueur clique sur le bouton) étaient systématiquement à refaire puisque la couleur principale de chaque panel varie d'une scène à une autre.

L'architecture du menu prévoyait donc une scène « d'accueil » (voir figure 6) et six autres scènes correspondant chacune à un monde (en comptant le tutoriel et le mode Hardcore). En temps normal, j'aurais eu la possibilité de réutiliser le canvas du premier monde étant donné que les ces six scènes présentent les mêmes boutons aux mêmes endroits. C'était sans compter sur *Unity* qui, apparemment, n'était pas d'accord avec cette méthode (c'est pourquoi la création des différents panels fût répétitive).

J'ai réussi à créer les premières scènes du menu sans toutefois les relier pour la première soutenance. L'objectif pour la deuxième soutenance était donc de créer les dernières scènes manquantes et d'implémenter le code reliant ces scènes entre elles.

Pour ce faire, deux méthodes s'offraient à moi : la méthode dite « nominative », qui met en avant le nom de la scène, et la méthode dite « numérique » qui prend en compte la place de la scène dans la fenêtre « Build Settings » de *Unity*. Ces deux méthodes présentent chacune des avantages et inconvénients.

FIGURE 6 – Scène d'entrée du menu après un premier essai sur Unity



Le gros avantage de la méthode dite « nominative » est que le lien entre le bouton et la scène associée sera toujours effectif, et cela même si l'on rajoute des scènes dans la fenêtre « Build Settings ». En effet, étant donné que la méthode nominative prend en compte le nom de la scène, peu importe le nombre de scènes que l'on rajoute, le lien reste valide tant que la scène ne change pas de nom.

L'inconvénient de cette méthode vient du fait que comme chaque nom de scène est obligatoirement unique (auquel cas le lien ne fonctionnerait pas), il faut nécessairement coder une fonction pour chaque lien, ce qui peut donner un code un peu redondant (voir figure 7).

FIGURE 7 – Méthode nominative, mettant en avant un code redondant

```
public void HardToMain()
{
    //SceneManager.LoadScene (SceneManager.GetActiveScene ().buildIndex - 6);
    Application.LoadLevel("Scene_MenuPrincipal");
}

public void MenuToMulti()
{
    Application.LoadLevel("multi level1");
}

public void TutoToMenu()
{
    Application.LoadLevel("Scene_MenuPrincipal");
}

public void W1LVL1()
{
    Application.LoadLevel("Niveau 1");
}

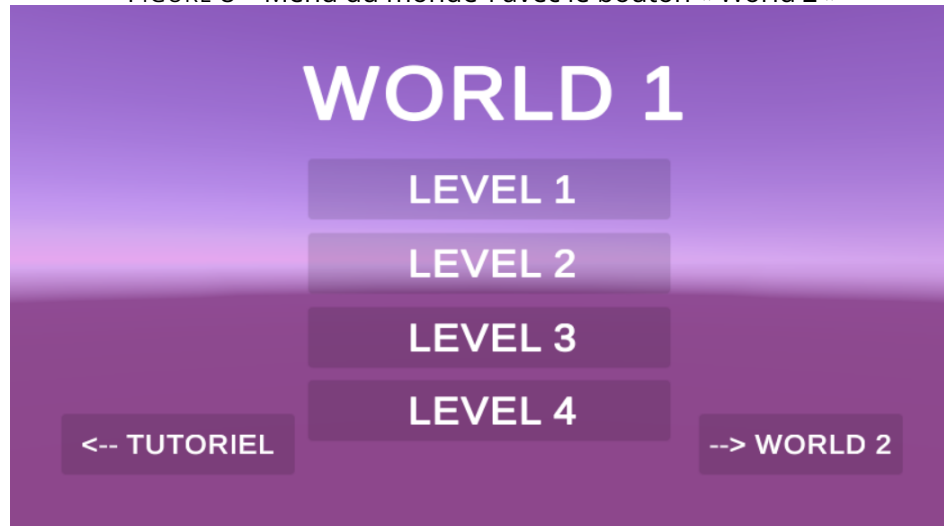
public void W1LVL2()
{
    Application.LoadLevel("Niveau 2");
}

public void W1LVL3()
{
    Application.LoadLevel("Niveau 3");
}

public void W1LVL4()
{
    Application.LoadLevel("Niveau 4");
}
```

Avec la méthode dite « numérique », une fonction codant un lien peut suffire pour plusieurs boutons. Voici le fonctionnement de cette méthode : toutes les scènes de mon menu ont été ajoutées dans la fenêtre « Build Settings ». Dans cette fenêtre, chaque scène est assignée à une position bien précise. Ainsi, lorsque l'on code cette méthode, on rentre le nombre de scènes que l'on souhaite passer (dans la fenêtre « Build Settings ») en cliquant sur le bouton. Ainsi, si je souhaite passer d'une scène à une autre, je choisis d'avancer de « +1 ». De plus, si je souhaite accéder à la scène précédente, je vais donc avancer de « -1 ». Ainsi, pour le bouton faisant passer d'un monde à un autre (qui apparaît plusieurs fois dans mon menu), il suffit de coder une fonction qui avance de « +5 » puisqu'il y a quatre niveaux dans chaque monde (voir figure 8).

FIGURE 8 – Menu du monde 1 avec le bouton « World 2 »



Vous l'aurez compris, en plus de demander quelques astuces de calcul, cette méthode est totalement dépendante du nombre de scènes dans la fenêtre « Build Settings ». Ainsi, il suffit de rajouter ou d'enlever une scène pour que l'ensemble des liens devienne invalide (puisqu'il y aurait alors un décalage dans les « Build Settings »). Et c'est par rapport à cette fragilité que j'ai décidé de me tourner vers la méthode « nominative ».

4.3.3 Commandes/Périphériques

En dehors du menu du jeu, l'implémentation des commandes du jeu ainsi que des périphériques constitue la seconde tâche que j'ai choisi de réaliser au sein du groupe. En effet, étant un joueur de jeux vidéo depuis mon enfance, j'ai toujours souhaité savoir comment fonctionnaient les commandes des périphériques de mes jeux (en particulier le déplacement de mon personnage). Cette mission constituait donc un tout nouveau défi pour moi, qui n'avais jamais codé de commandes auparavant.

Tout comme le menu, j'ai d'abord concerté mes camarades afin de connaître précisément leurs attentes. L'implémentation des commandes devait donc comprendre toutes les commandes liées au déplacement du joueur, à savoir avancer tout droit, reculer, aller à gauche, aller à droite. De plus, étant donné que nous codions un jeu à la première personne (plus couramment appelé FPS pour First Person Shooter), il était essentiel que la caméra du jeu (la « Main Camera » dans *Unity*) soit intégrée dans la tête du joueur, mais aussi que la caméra soit liée aux mouvements de la souris. Enfin, il fallait

intégrer au jeu une commande de plus pour permettre au joueur d'appuyer sur un bouton ouvrant des portes.

J'ai d'abord décidé de concentrer mes efforts sur les commandes du déplacement du personnage car elles constituaient selon moi la part la plus importante dans l'implémentation des commandes. Et tout comme le menu, j'ai choisi d'implémenter les commandes sur *Unity*.

Implémenter les commandes du jeu m'a fait découvrir un aspect du code que j'aurais préféré ne jamais découvrir : la petite part de hasard dans le fonctionnement du code.

En effet, j'ai d'abord choisi de me diriger vers la méthode proposée dans la conférence « Bien démarrer son projet », filmée et mise en ligne par GCONFS. Le gros avantage de cette vidéo était qu'elle proposait une méthode simple à comprendre (voir figure 9).

FIGURE 9 – Méthode proposée par la vidéo GCONFS

```
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4
5 public class controleur : MonoBehaviour {
6
7     // Use this for initialization
8     private Transform _t;
9     void Start () {
10         Debug.Log("Start");
11         _t = this.gameObject.GetComponent<Transform>();
12     }
13
14     // Update is called once per frame
15     void Update () {
16         Debug.Log("Update");
17         Vector3 dir = Vector3.zero;
18         if(Input.GetKey(KeyCode.LeftArrow))
19             dir = Vector3.left;
20
21         if(Input.GetKey(KeyCode.RightArrow))
22             dir = Vector3.right;
23
24         if(Input.GetKey(KeyCode.UpArrow))
25             dir = Vector3.forward;
26         if(Input.GetKey(KeyCode.DownArrow))
27             dir = Vector3.back;
28     }
29 }
```

Un coup d'œil à ce code suffit pour en comprendre la logique : la commande `Input.GetKey` permet « d'accéder » à la touche entrée en paramètre. Ainsi, le mot qui suit `KeyCode` correspond à la touche du clavier à laquelle on souhaite rattacher une commande, qui suit alors la condition.

Avant de pleinement me concentrer sur cette méthode, j'ai d'abord choisi de me tourner vers des alternatives. Et après avoir trouvé une méthode qui me paraissait intéressante, j'ai décidé de me baser sur cette dernière pour le début de l'implémentation de mes commandes (voir figure 10). Et sans aucune explication, le code ne fonctionne pas. *Unity* me signale plusieurs erreurs qui n'apparaissent pas dans la vidéo. C'est pourquoi je me suis replié sur la méthode proposée par GCONFS, qui elle fonctionnait bien.

FIGURE 10 – Méthode trouvée sur un site différent de GCONFS

```
using UnityEngine;
using System.Collections;

public class PlayerController : MonoBehaviour {

    public int speed = 10;

    // Update is called once per frame
    void FixedUpdate () {

        float mouveHorizontal = Input.GetAxis("Horizontal");
        float mouveVertical = Input.GetAxis("Vertical");

        Vector3 movment = new Vector3(mouveHorizontal, 0, mouveVertical);
        rigidbody.AddForce(movment * speed * Time.deltaTime);

    }
}
```

J'avais réussi à implémenter les commandes du déplacement du joueur (avancer, reculer, aller à gauche, aller à droite) pour la première soutenance orale. De plus, je m'étais également occupé de la vue à la première personne du joueur. Il ne me restait donc plus que la commande pour appuyer sur le bouton, après avoir liée la Main Camera aux mouvements de la souris.

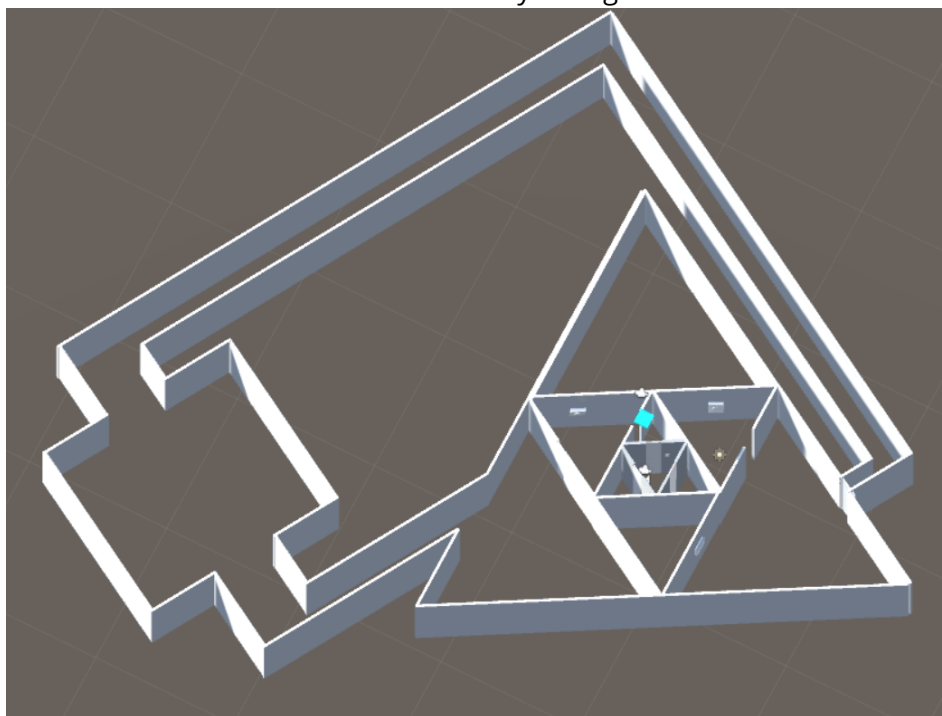
Et je dois reconnaître que ces deux dernières missions m'ont posées des difficultés, qui je pense n'auraient pas été surmontables sans l'aide précieuse de mes camarades.

Enfin, Morgan a apporté son collider pour la commande appuyant sur le bouton. C'est donc avec mes camarades que j'ai réussi l'implémentation des commandes du jeu dans les temps.

Et tout comme pour le menu, je me suis basé sur un site proposant des guides vidéo complets pour implémenter les commandes permettant au joueur de jouer avec des périphériques (à savoir des manettes).

4.3.4 Niveaux

FIGURE 11 – Niveau « Crazy Triangle » de Léo



La création des niveaux était une tâche commune à tous les membres du groupe. Chaque personne devait inventer et concevoir quatre niveaux différents. Et je dois dire que j'ai bien aimé cette mission. En effet, la création de niveaux en 3D est une mission qui sollicite des compétences artistiques, et non pas scientifiques comme mes précédentes tâches.

Avant de m'attaquer à la modélisation 3D, j'ai d'abord tenu à dessiner les modèles de mes niveaux sur papier pour que Morgan et Pierre puissent

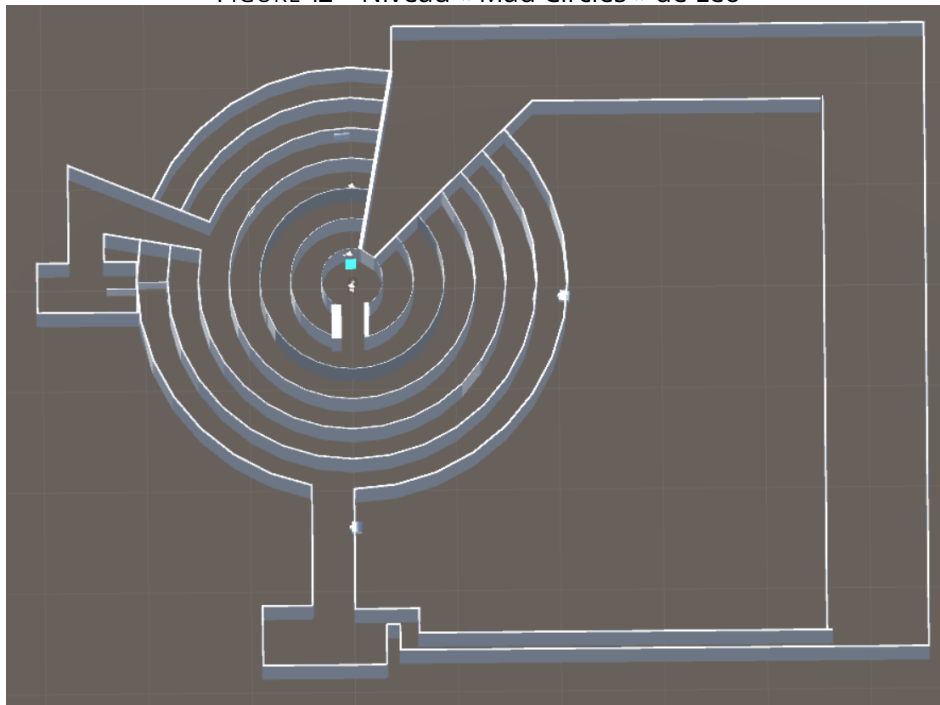
correctement effectuer la modélisation 2D sur ordinateur. C'est à ce moment-là que j'ai imaginé les concepts qui allaient revenir dans mes niveaux, et notamment les raccourcis. Ainsi, le joueur aura la possibilité de finir très rapidement les niveaux si ce dernier trouve le raccourci caché dans mon niveau.

Sur conseil de mon ami Pierre, j'ai fait le choix de d'abord passer par le logiciel de modélisation 3D *Blender* avant d'insérer mes niveaux sur *Unity*. Je ne regrette pas ce choix car personnellement, je trouve que *Blender* propose une interface beaucoup plus accessible à des novices de la modélisation 3D (comme moi à ce moment-là).

Après avoir créé deux niveaux aux formes rectangulaires, j'ai choisi de changer les formes de mes niveaux. Ainsi, mon troisième niveau se résume à un emboîtement de triangles les uns dans les autres. Et ce n'est qu'après avoir terminé ce niveau que je me rendis compte de l'immense potentiel que présentait ce concept d'emboîtement. En effet, dans mon troisième niveau, le joueur est face à trois portes lorsqu'il apparaît sur la carte. Deux choix s'offrent alors à lui : continuer tout droit si la porte choisie donnait accès au bon chemin ou bien faire marche arrière. Ainsi, pour ajouter de la difficulté au niveau, il suffit de superposer un triangle supplémentaire pour ajouter de la difficulté au niveau (voir figure 11).

J'ai donc utilisé le même concept pour créer mon dernier niveau, Mad Circles, à la différence que cette fois-ci le concept était appliqué à des formes circulaires (voir figure 12).

FIGURE 12 – Niveau « Mad Circles » de Léo



4.3.5 Conclusion

Ce n'est pas sans éprouver une certaine nostalgie que je conclus donc ma partie sur ce rapport de projet. J'ai réellement accroché au côté professionnalisant de cette expérience, et par professionnalisant je fais référence au travail en équipe (pendant plus de 18 heures d'affilées) mais aussi au fait de devoir rédiger les différents rapports de soutenances afin de rendre compte du travail réalisé.

Comme dit plus haut, le travail en équipe effectué ensemble m'aura enseigné bon nombre de valeurs primordiales dans le monde de l'entreprise, comme par exemple la tolérance, la communication ou encore la solidarité face à la charge de travail.

À présent, mes amis et moi sommes fiers de vous présenter le résultat final d'un travail acharné qui aura duré plus de six mois : j'ai nommé *Through* !

4.4 Sébastien

4.4.1 Introduction

Cela fait longtemps que je suis intéressé et que je joue aux jeux vidéo. En effet, mes premiers jeux, qui étaient alors des jeux éducatifs⁸, remontent à l'époque où j'étais en primaire. Au fur et à mesure, et notamment à partir de la classe de quatrième, j'ai développé une passion pour ce média. En même temps que j'accumulais les expériences de jeu et que je développais ma culture autour de ce domaine, je me suis intéressé à la création et au développement de ces jeux. De cette manière, mon projet professionnel est d'abord passé par programmeur de jeux vidéo, puis modelleur et animateur 3D avant de s'arrêter sur game designer. De ce fait, entre ma première et ma terminale j'ai réalisé des summer camps dans deux écoles de jeux vidéo : Rubika, située à Valenciennes dans le Nord de la France, et Isart Digital, située à Paris. Durant ces summer camps, j'ai eu l'occasion de réaliser des jeux de plateau ainsi qu'un jeu vidéo (sur le moteur de jeu éducatif Scratch) sur des périodes allant d'une journée à une semaine pour chaque projet.

Definition 11 (Game designer)

Personne qui établit les règles d'un jeu ainsi que la manière d'y jouer. C'est la personne qui imagine et crée l'expérience du joueur.

Le jeu vidéo réalisé lors du summer camp à Isart Digital constitue l'inspiration principale pour le jeu *Through* que nous développons durant ce projet à l'EPITA. En effet, ce jeu nommé *Alone in Space* reposait sur un principe simple : le joueur se retrouvait seul dans un vaisseau spatial et devait échapper à un alien. Pour cela, le joueur devait traverser les pièces du vaisseau en résolvant des énigmes basées sur l'ouverture de portes (voir figure 13). Il y a cependant quelques différences à noter : le jeu était en vue du dessus et les salles n'étaient pas encore des labyrinthes. Nous n'avons donc pas copié le concept, nous l'avons simplement amélioré et avons modifié sa portée puisque le but n'est plus seulement de faire résoudre des énigmes au joueur mais également de faire monter son stress, voire sa peur, en créant un sentiment de claustrophobie par des environnements étroits et labyrinthiques et en lui donnant une visibilité réduite.

8. Par exemple *Adibou* ou encore *Sethi*.

FIGURE 13 – Niveau de *Alone in Space*

Après ce petit résumé sur l'histoire du projet de mon point de vue, je vais désormais vous présenter mon travail sur le projet. Pour cela, je vais tout d'abord reprendre les objectifs qui m'étaient fixés, puis je vais présenter le travail réalisé pour finalement lister ce que le projet m'a apporté et en quoi cela m'a été bénéfique. Enfin, je vous laisserai sur une petite conclusion.

4.4.2 Reprise des objectifs

Pour ce projet, plusieurs tâches m'ont été assignées. Tout d'abord, j'ai été nommé chef de projet, ce qui n'est pas une mince affaire. En effet, cette tâche consiste à organiser le planning de travail ainsi que les réunions afin que le projet respecte bien les deadlines définies dans le cahier des charges. Il faut également superviser la mise en commun de toutes les parties du projet.

En plus de ce travail, j'ai partagé la réalisation du projet avec les autres membres du groupe. Comme eux, je devais réaliser quatre maps afin de constituer un monde du jeu, c'est-à-dire les dessiner puis les modéliser en 3D et ajouter les textures. Je devais également m'occuper de l'intelligence artificielle du monstre qui serait présent dans les niveaux. Pour cela, il fallait réaliser un algorithme de pathfinding ainsi qu'un algorithme permettant au monstre de savoir quand attaquer le joueur. Enfin, je devais créer la mini-

map permettant au joueur de se repérer dans les niveaux. Celle-ci devait être assez claire et compréhensible pour aider le joueur. Pour cela, elle devait notamment ne pas être surchargée d'informations.

Definition 12 (Pathfinding)

Algorithme permettant à une intelligence artificielle de trouver le chemin à parcourir jusqu'à son objectif.

En parallèle, j'étais chargé de mettre en page les documents à rendre à l'aide du langage $\text{\LaTeX}$. Cela signifie prendre les textes écrits par les différents membres du groupe et les adapter à ce langage pour que le document PDF final ait une mise en page correcte. Il fallait également ajouter les images au document.

Des tâches supplémentaires pouvaient être ajoutées ponctuellement pour aider d'autres membres du groupe en cas de problème ou de manque de temps.

4.4.3 Travail personnel

Tout d'abord commençons par les noms. En effet, les noms « La Confrérie » puis « Through » ont été proposés par moi-même. L'idée du nom « La Confrérie » vient des jeux *Assassin's creed*⁹ auxquels j'ai beaucoup joué. J'ai- mais beaucoup l'idée d'un groupe soudé dont les membres sont des frères regroupés autour d'un même objectif. Le nom « Through » quant à lui représentait bien le concept du jeu puisque le joueur doit passer « à travers (*through*) » les labyrinthes. De plus, ce nom est simple et facile à retenir. Après quelques discussions et malgré les réticences de certains membres du groupe, ces noms ont été adoptés.

Dès le début du projet, lors de la création du groupe, je me suis proposé pour devenir le chef de groupe. En effet, je souhaitais acquérir de l'expérience dans le domaine de la gestion de projet car j'avais déjà eu l'occasion d'être chef de projet¹⁰ et que cela m'avait beaucoup intéressé. De plus, des compétences dans ce domaine s'avèrent utiles dans le monde professionnel. J'ai rapidement été choisi et je me suis alors rendu compte de ce que ce poste impliquait sur un projet de cette ampleur. En effet, il n'est pas toujours évident de faire en sorte que les deadlines soient respectées. Pour y

9. Edités par *Ubisoft* à partir de 2007.

10. Lors du TPE en classe de première et du projet de Sciences de l'Ingénieur en terminale

parvenir, j'ai tout d'abord organisé des réunions de groupe chaque semaine (le jeudi soir). Des objectifs étaient définis pour chacune d'elle et chaque réunion démarrait par un compte-rendu sur ces objectifs afin de vérifier que ceux-ci étaient bien respectés. Ensuite, les membres se donnaient des conseils afin d'améliorer les parties de chacun. Enfin, les nouveaux objectifs étaient définis et chacun commençait à réfléchir à la manière d'y arriver. Les membres pouvaient alors poser des questions à leur collègues. Après chaque réunion, un compte-rendu était rédigé afin d'avoir une trace écrite de celle-ci à laquelle chaque membre du groupe avait accès, notamment pour retrouver les objectifs énoncés. Un autre moyen de respecter les deadlines a été la mise en place de séances intensives de projet avant les soutenances afin de finir le travail avec une efficacité maximale. Lors de ces séances le groupe était réuni et les membres pouvaient ainsi aisément échanger des idées et s'entraider. De plus, comme nous ne faisons que ça et que nous n'étions pas seuls chez nous, nous pouvions mieux nous concentrer sur le projet. En tant que chef de projet, je me suis également rendu compte que les heures de sommeil pouvaient facilement diminuer.

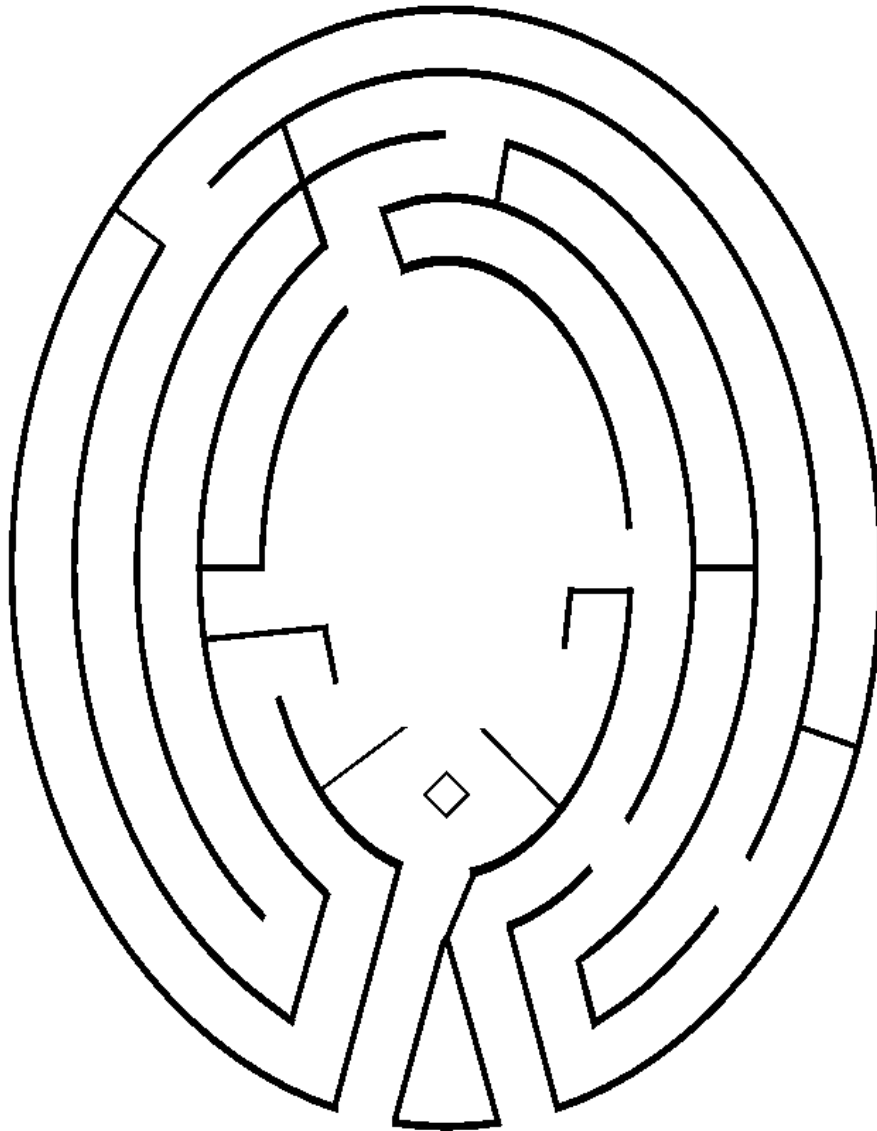
Je vais terminer avec mon rôle de chef de projet (avant d'y revenir à la fin) par l'arrivée d'un nouveau membre dans le groupe. En effet, une semaine avant la deuxième soutenance, Anass Kasmi nous a rejoint. Son précédent groupe venait de se dissoudre et on nous a demandé de l'accepter dans La Confrérie. Malgré quelques réticences, nous avons décidé de l'intégrer au groupe et cela s'est parfaitement bien passé dès le premier jour. Nous lui avons tout d'abord présenté le projet puis lui avons demandé s'il avait des idées concernant les bonus qu'il pouvait ajouter au projet. A partir de là, Anass faisait partie du groupe et travaillait comme n'importe quel membre.

Je vais maintenant vous parler de la création des niveaux. J'ai tout d'abord commencé par les dessiner sur papier afin de pouvoir aisément les corriger, les améliorer et effectuer des modifications. J'ai tout d'abord eu l'idée d'un niveau dont la fin serait visible depuis l'endroit de départ. Une sortie en hauteur s'est donc rapidement imposée à moi. Cette idée a donné mon premier niveau : le niveau carré dans lequel on accède à la sortie en montant à l'aide d'un plan incliné. Pour mon deuxième niveau, j'ai puisé mon inspiration dans le film *Le Labyrinthe* puisque j'ai créé un niveau de forme ronde. J'ai construit ce niveau comme une course puisque dès le départ le monstre du niveau nous poursuit¹¹ (quand le joueur arrive dans le niveau il s'agit de la première chose qu'il voit) et le joueur doit atteindre le centre du labyrinthe avant de se faire attraper. Le troisième niveau quant à lui vient directement de la série *WestWorld* puisque j'ai repris la forme du labyrinthe du logo de cette série. Je l'ai ensuite adaptée pour créer un labyrinthe

11. En anglais, le titre film *Le Labyrinthe* est *The Maze Runner*...

plus complexe au centre duquel est placé le monstre (voir figure 14). Enfin, pour mon dernier niveau j'ai adapté un niveau de *Pacman* en lui ajoutant de nombreuses portes ainsi que des boutons contrôlant plusieurs portes à la fois ce qui peut bloquer certains passages en même temps qu'en ouvrir de nouveaux. J'ai essayé d'apporter une certaine difficulté dans mes niveaux pour que le joueur ait besoin de réfléchir pour s'en sortir. Le niveau 2 en revanche a été construit afin de faire monter la tension pour que le joueur n'ait pas le temps de réfléchir et doive agir avec réflexe. J'avais également commencé la création d'un niveau basé sur le modèle des pyramides mais l'idée a dû être abandonnée car je n'étais pas satisfait du résultat.

Une fois les dessins terminés, j'ai modélisé mes niveaux à l'aide du logiciel de modélisation 3D *Blender 3D*. Ce logiciel demande un petit temps d'apprentissage mais une fois maîtrisé (en tout cas les commandes basiques) la modélisation de niveaux devient plutôt rapide. Les niveaux créés sur *Blender* peuvent ensuite être aisément importés sur *Unity* afin d'être utilisés dans le jeu. A partir de là, le sol ainsi que les portes et les boutons ont pu être rajoutés au niveau.

FIGURE 14 – Plan du niveau *Westworld*

Je me suis également occupé de l'intelligence artificielle du monstre. Pour cela, j'ai tout d'abord programmé le pathfinding. Cet aspect m'avait beaucoup intéressé durant le TP C# « I/O » dans lequel nous avons dû tester plusieurs algorithmes de pathfinding avant d'essayer d'en implémenter un meilleur. J'avais alors déjà étudié les meilleurs algorithmes de pathfinding, notamment ceux utilisés dans les jeux vidéo. L'algorithme A* créé par Dijks-

tra¹² est le plus couramment utilisé dans ceux-ci car il est rapide et précis. Ce dernier repose sur la création d'une grille invisible sur le sol du niveau. Cette grille permet à l'intelligence artificielle de calculer le chemin le plus court jusqu'au joueur et de suivre celui-ci. Pour ce calcul, l'intelligence artificielle additionne la longueur du chemin du départ jusqu'à une certaine case et une estimation de la longueur du chemin de cette case jusqu'à la fin. Ce calcul est fait pour chaque case du chemin estimé et permet de trouver le chemin le plus court. De plus, le chemin est recalculé à chaque frame du jeu, les déplacements du joueur sont ainsi pris en compte. Ce pathfinding est facilement implémentable sur différents niveaux puisqu'aisément adaptable. Ensuite, il faut faire en sorte que l'intelligence artificielle reconnaisse les portes fermées et ouvertes pour savoir s'il peut les traverser. Il faut également que le monstre ait la capacité de tuer le joueur s'il arrive sur sa position.

Definition 13 (Frame)

Terme désignant une image. Il est très utilisé dans le domaine des jeux vidéo. Le terme fps (frame per second) désigne le nombre d'images par seconde.

D'autre part, j'ai créé la mini-map. Cette mini-map est placée en haut à droite de l'écran de jeu. Elle a été créée à l'aide d'une deuxième caméra qui est placée au-dessus du joueur et tournée vers le sol. Pour que le joueur soit visible, un carré de couleur bleue a été placé au-dessus de lui. Ce carré est en fait un plan, cela permet qu'il soit invisible si regardé par-dessous, c'est-à-dire depuis le point de vue du joueur. La caméra de la mini-map renvoie donc une image du niveau vu du dessus qui affiche le joueur, les murs ainsi que le monstre (voir figure 15). Cette caméra suit le personnage et est aisément adaptable à tous les niveaux.

Plus tard, j'ai ajouté le personnage contrôlé par le joueur dans chaque niveau. Pour cela, j'ai créé un asset contenant la hitbox du personnage, le script de déplacement, la caméra ainsi que le système servant à la mini-map. Dans le même temps, le monstre a également été placé dans chaque niveau. Enfin, une sortie renvoyant sur le menu a été ajoutée à la fin de chaque niveau.

12. Edsger Dijkstra est un mathématicien et informaticien néerlandais ayant reçu plusieurs prix, dont le prix Turing, pour ses travaux sur les algorithmes.

FIGURE 15 – Mini-map

**Definition 14 (Asset)**

Préfabriqué correspondant à un objet à placer dans un niveau de jeu. Il peut contenir du code, un modèle 3D... ou une combinaison de plusieurs éléments.

Pour chaque soutenance, j'ai également rédigé les documents en $\text{\LaTeX}$. Cela signifie que chaque membre du groupe a rédigé sa partie puis que je me suis occupé de la mise en page sur $\text{\LaTeX}$. Ce langage est assez facile à prendre en main, même s'il ne fait pas toujours ce à quoi nous nous attendons. Il faut alors utiliser quelques petites astuces pour obtenir un résultat proche des attentes de départ. Vous pouvez d'ailleurs constater que les images ne sont pas toujours sur les pages sur lesquelles elles sont citées, mais je peux vous assurer que sans utiliser d'astuce la plupart seraient en fin de document.

Enfin, en tant que chef de projet, j'ai supervisé la mise en commun des parties de chacun pour créer le résultat final. Cela nécessite une connaissance de l'ensemble du travail des différents membres du groupe ainsi que la capacité à les aider sur les finitions pour que tout fonctionne correctement et soit lié de la bonne manière au sein du projet. Lors de la mise

en commun, les membres doivent s'aider les uns les autres en fonction de l'avancement des différentes parties pour que tout soit fini en même temps.

4.4.4 Bénéfices du projet

Ce projet, notamment grâce à la diversité des tâches à réaliser, m'a permis d'apprendre de nombreuses compétences, ou au moins d'acquérir plus d'expériences dans de nombreux domaines. En effet, la gestion de projet en équipe m'était presque inconnue. J'ai donc eu l'occasion de développer mes compétences dans ce domaine, d'autant plus que j'étais chef de projet. Je me suis rendu compte que cette tâche est loin d'être aussi aisée qu'on pourrait le penser. Il y a plusieurs raisons à cela. Tout d'abord, des imprévus apparaissent toujours¹³, il faut alors savoir réagir assez rapidement pour surmonter les problèmes rencontrés. Il est également difficile de respecter des deadlines puisque dans le même temps il est facile de prendre du retard. Une autre difficulté est celle de la motivation. Par ce mot je veux signifier qu'il faut réussir à conserver et alimenter la motivation de l'ensemble du groupe pour que le projet avance toujours avec une bonne vitesse. Il faut aussi savoir régler les petits problèmes de dissensions au sein du groupe, mais également donner de son temps (surtout avant les deadlines) pour que tout fonctionne correctement. En permanence, la communication a été un élément important qui nous a permis d'avancer. *GitLab* nous a également beaucoup aidé pour partager notre travail. Il demande un petit temps d'adaptation mais se révèle ensuite très utile et pratique. C'est pourquoi je recommande *Git* ainsi que ses dérivés *GitHub* et *GitLab* à toute personne souhaitant travailler sur un projet.

Dans le même style, j'ai pu découvrir la personnalité de mes collègues dans un environnement de travail plus contraignant. J'ai ainsi pu remarquer les différences de caractère en fonction des situations et j'ai appris à adapter le mien aux différentes situations. Parallèlement, la coopération avec certains membres du groupe Awesome Seals s'est renforcée.

Par ailleurs, j'ai également appris beaucoup de choses sur la réalisation d'un jeu vidéo. J'ai notamment pu, grâce aux tâches variées qui m'étaient assignées, me rendre compte du travail de plusieurs personnes composant une équipe de développement de jeu vidéo en général. J'ai ainsi pu comprendre leurs contraintes et je serai également plus indulgent à l'avenir lorsque je rencontrerai un bug dans un jeu. De plus, j'ai pu acquérir des bases dans les domaines du level design, de la modélisation 3D et de la programmation d'intelligences artificielles pour un jeu vidéo.

13. Beaucoup d'imprévus...

Definition 15 (Level design)

Création des niveaux d'un jeu vidéo.

Enfin, ce projet m'a confirmé dans mon choix (que j'ai fait en fin de terminale) de m'éloigner du jeu vidéo pour me tourner vers les autres domaines de l'informatique et de la programmation.

4.4.5 Conclusion

Pour conclure, ce projet, grâce à la diversité des tâches demandées et malgré les rares accrochages, m'a appris de nouvelles choses et m'a surtout montré comment se déroule un long projet de la phase de création au produit fini en passant par la réalisation. Il s'agit en quelque sorte d'une découverte du monde professionnel tout en restant dans un environnement plus encadré et moins dangereux et punitif en cas d'erreur. Ce projet était une bonne expérience, et le groupe La Confrérie un bon groupe. Reste à savoir comment s'organisera le projet de deuxième année.

4.5 Anass

4.5.1 Introduction

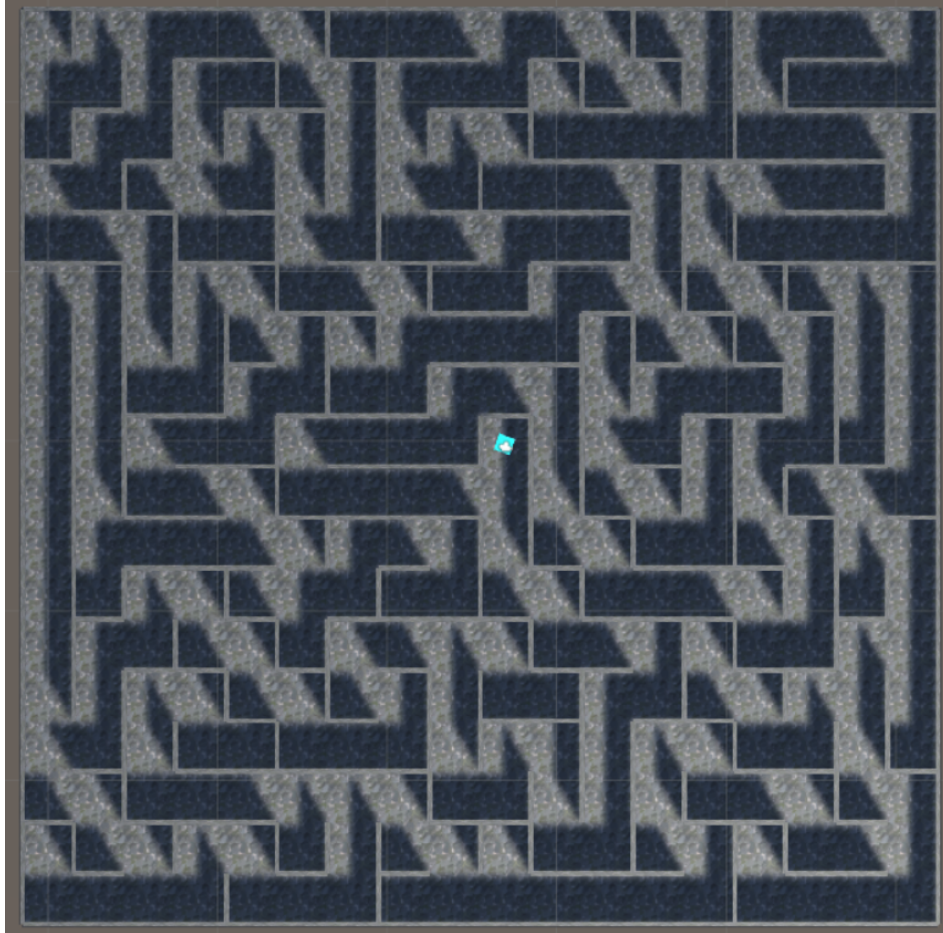
Et voilà déjà la fin de ce magnifique projet dans ce nouveau groupe génial qu'est La Confrérie, car oui il est vrai que j'ai changé de groupe de projet et que je suis arrivé juste avant la deuxième soutenance mais mon adaptation a été très bonne et j'en suis très satisfait. Très satisfait car malgré mon arrivée soudaine, le groupe a été derrière moi, m'a tout de suite mis à l'aise et m'a considéré comme un membre à part entière.

4.5.2 Travail personnel

Une fois bien installé dans le groupe, on m'a mis au courant des différents objectifs que j'avais à atteindre, en l'occurrence des bonus du jeu. Après quelques heures de réflexion nous avons décidé que mon travail serait de faire un mode bonus qui se nommera le mode « Hardcore » et un menu pause sur le logiciel *Unity 3D*.

Je vais tout d'abord commencer par vous présenter ce qu'est le mode « Hardcore ». Alors le mode « Hardcore » est un mode indépendant des autres niveaux de *Through*, c'est-à-dire qu'il ne prend pas en compte l'avancement des niveaux du mode histoire. Afin de réaliser ce fameux mode « Hardcore », j'ai eu la folle idée de programmer un générateur aléatoire de labyrinthe avec les dimensions que l'on souhaite (voir figure 16). La partie de code est très lourde et très différente de ce que l'on fait en TP à EPITA mais le concept est simple en réalité : on crée une cellule avec quatre murs et un sol qui va nous servir de cellule initiale et lorsque qu'on va lancer le mode de jeu elle va se dupliquer et retirer des murs selon son positionnement afin de créer un chemin jusqu'à donner un labyrinthe entier. On a aussi rajouté un bouton sur le mur de fin du labyrinthe pour ensuite lancer l'écran qui dira au joueur qu'il a réussi et aussi une fonction qui vérifie que le labyrinthe contient bien une issue de sortie (ce serait peu malin si le joueur se retrouvait coincé sans aucune chance de sortie).

FIGURE 16 – Exemple de labyrinthe généré pour le mode « Hardcore »



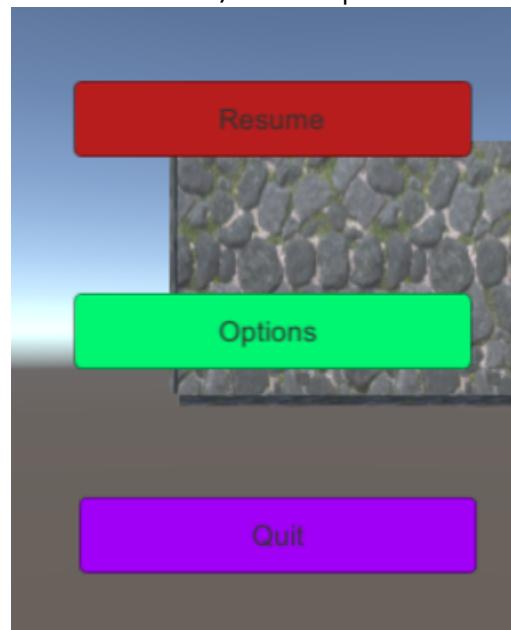
Vous me direz : « D'accord, il y a un beau labyrinthe et après? ». Et c'est maintenant que je vous explique le concept de ce mode de jeu.

Une fois le mode lancé, le joueur sera positionné au milieu du labyrinthe et devra donc en sortir avec le maximum de points en poche. On rajoute un peu de piment car le joueur aura un temps pour sortir de ce piège sinon il y restera à jamais... Les points se feront en fonction du temps d'arrivée, plus il finira le niveau rapidement, plus il aura de points (je me suis inspiré du système de points lors de petits jeux techniques dans le célèbre jeu de football FIFA).

Ensuite, ma seconde de tâche était de réaliser un menu pause qui nous permet d'arrêter le jeu, le reprendre, de modifier quelques options ou bien de quitter le niveau dans lequel on se trouve.

Pour réaliser ce menu, j'ai créé un arrière-plan sur *Unity 3D* et insérer les boutons « RESUME », « OPTIONS » et « QUIT » (voir figure 17). Le menu se lancera lorsque le joueur appuiera sur la touche « Echap » de son clavier d'ordinateur.

FIGURE 17 – Menu pause



4.5.3 Bénéfices du projet

J'ai beaucoup appris sur moi-même lors de ces quelques mois de travail au niveau relationnel car j'ai appris à travailler en groupe, suivre les consignes d'un chef de projet mais aussi j'ai appris à m'adapter en tant que nouveau membre d'un groupe et je suis très satisfait de ce que notre groupe a produit lors de ce projet.

4.5.4 Conclusion

Pour conclure, ces quelques mois de projet dans mon nouveau groupe ont été très enrichissants pour moi. Tant au niveau professionnel qu'au niveau relationnel car, oui, j'ai appris à faire des choses complètement différentes de ce que l'on fait généralement à EPITA. C'est la première fois que je participe à la conception d'un jeu vidéo et j'ai trouvé cette expérience extrêmement enrichissante et ce travail me rend en quelque sorte fier car je pourrai montrer le jeu à ma famille et mes amis.

5 Travail en cours

Au moment d'écrire ce rapport de projet, ce dernier n'est pas encore terminé. En effet, les partiels ont constitué notre priorité pour l'instant. Ainsi, le projet a été mis en pause (excepté bien sûr la rédaction du rapport de projet). Celui-ci reprendra dès la fin des partiels afin que tout soit terminé à temps.

Cependant, je peux vous annoncer ce qui devra être terminé d'ici la fin du projet. Tout d'abord, l'aspect esthétique des niveaux doit être achevé, c'est-à-dire placer les textures. Les portes ainsi que les monstres doivent également être ajoutés aux niveaux. En parlant de ce monstre, son intelligence artificielle doit être terminée. Le monstre devra ainsi poursuivre le joueur et pouvoir le tuer. Les menus devront être totalement opérationnels, à savoir permettre au joueur de modifier des paramètres. Enfin, le mode multijoueur ainsi que le mode « hardcore » doivent recevoir les finitions attendues.

De plus, le dossier d'exploitation ainsi que la procédure d'installation et de désinstallation doivent être réalisés pour la présentation du jeu.

Enfin, le jeu étant encore en développement, de légers changements peuvent encore apparaître entre ce rapport de projet et le rendu final.

6 Conclusion

Comme vous l'aurez compris, développer un jeu vidéo nous aura réservé pas mal de surprises, tantôt bonnes et tantôt mauvaises. Nous n'allons pas reprendre ces points en conclusion puisque chacun les a déjà développés dans sa partie personnelle. Cependant, nous serons tous d'accord pour vous dire que le projet *Through* aura été une belle expérience, aussi bien individuellement qu'en groupe.

Sur le plan « technique », nous avons eu l'occasion de prendre en main de nouveaux outils, découvrir de nouvelles méthodes de travail et bien plus encore.

Ce ne sont cependant pas les seules choses que nous avons dû réaliser, nous faisons allusion par là au travail de groupe et à l'entente entre les membres d'une même équipe.

Comme dans la majeure partie des projets, nous avons pu constater une certaine différence entre les membres de La Confrérie en ce sens où nous n'avons pas les mêmes centres d'intérêt, les mêmes méthodes de travail etc. Cette différence ne nous aura pas empêché de poursuivre le projet, au contraire! Nous devons beaucoup à cette polyvalence en ce qui concerne la réussite et la tenue de nos objectifs.

Finalement, Le travail que nous avons réalisé nous représente tous d'une quelconque manière, ce qui nous apportera lors de cette dernière soutenance, une grande fierté à vous présenter notre jeu ¹⁴,



Merci pour l'attention que vous avez portée à notre rapport.

14. Vraiment, visitez throughgame.fr...