Asus Warriors

ASUS Team :

Mouton	(Antoine Bruneel)
Cladaaliy	(Jonathan Bedo)
Xoy	(Fabien Bourgineau)
Titou	(Alexandre Gazuit)

Table des matières

Introduction

Le projet ASUS Warrios developpé tout au long de l'année par l'ASUS Team va être présenté pour la dernière fois. Cela afin de faire le point sur le travail accomplie tout au long de l'année par les membres du groupe.

Mais plus qu'un simple projet, c'est le moment pour tous les membres du groupe de faire le point sur le travail accompli tout au long de cette année. Aura-t-il était instructif et enrichissant ? Tous les membres du groupe ont-ils trouvé un intêret dans son développement ? Mais aussi, qu'est-ce qui serait à changer pour travailler encore plus efficacement les années suivantes ?

Dans ce rapport, les membres de l'ASUS Team feront donc une dernière fois le compte-rendu de leur travail. Mais cette fois, ils le feront en commençant au début du développement du projet.

Il est temps de tirer les enseignements de cette année...

1 Retour sur le cahier des charges

1.1 Avant le projet

Aucun des membres du groupe de l'Asus Team ne savait coder au début de cette année. Nous sommes donc partis tous un peu au hasard, selon nos envies et nos délires de l'instant. Heureusement, certaines exigences du jury nous ont obligées à nous organiser un minimum. Parmis elles, la plus importante était la rédaction dès le début de l'année d'un cahier des charges définissant précisément ce que serait le projet et les tâches de chacun dans son développement.

La rédaction de ce cahier des charges a donc été précédé de nombreuses séances de brainstorming afin de trouver un projet à la fois intéressant pour chaque membre du groupe et intéressant au niveau de l'application de notions qui sont vues en cours pendant l'année. Le groupe ayant bien compris que ce projet de première année serait le seul du cursus EPITA entièrement libre et laissant la possibilité à chacun de développer un jeu, c'est dans cette voie que nous nous sommes orientés. Il ne fut pas facile de se mettre d'accord sur le type de jeu que ce serait. Finalement, au bout de quelques temps, c'est un jeu de type RPG qui a été choisi. L'univers de celui-ci fut décidé sur un délire : une guerre d'ordinateur dont les héros seraient des Asus face aux redoutables Dell et Mac.

Il est vrai que le groupe n'a pas - et on s'en rend compte aujourd'hui, à tord - réfléchi à tous ce qui allait devoir être fait. Principalement au niveau graphique, où aucun membre du groupe ne savait utiliser le moindre éditeur de model 3D. Nous avons tout de même essayé de découper de manière assez précise le projet pour organiser son développement entre les membres du groupe.

Les grandes parties qui ont été retenue sont :

- conception
  - scénario
  - gameplay
- partie physique
  - déplacement
  - IA
- parti graphique
  - éditeur de map
  - moteur graphique
  - bibliothèque graphique
- partie sonore
  - bruitages
  - musiques
- promotion
  - site internet
  - goodies

1.2 Présentation du projet

Ces différents éléments du développement se sont répartis entre les membres du groupe. Encore une fois, aucun de nous ne sachant coder ou même sachant ce que représente réellement le travail dans ces parties, elles ont été choisies par les membres selon leurs préférences, un peu au hasard. Il a ensuite fallu organiser le développement en fonction des 4 soutenances qui viennent sanctionner le projet dans l'année. Chacun a donc essayé d'estimer le temps qu'il lui faudrait pour réaliser sa partie.

Même après ce découpage du projet, certaines des parties restaient floues et l'on se devait de les préciser. Chaque responsable des grandes

parties s'est donc fixé des objectifs, parfois ambitieux, afin d'affiner le planning.

1.2.1 Le gameplay

Dans la définition du projet, il est spécifié que nous développerions un RPG. Le gameplay devait donc être cohérent avec cette information. Reste à choisir quels éléments des RPG classiques nous allions retenir et essayer d'adapter dans notre jeu. Parmi les premiers dont nous avons parlé, il y avait la passibilité de lancer des quêtes en interagissant avec des PNJ (personnages non-joueurs). Mais aussi la possibilité aux personnages d'évoluer grâce à un système d'expérience similaire à celui de World of Warcraft et à un système d'équipements.

Un autre aspect très important du gameplay qui se devait d'être rapidement précisé était le mode de déplacement. Ce choix donne une direction à la partie "déplacement" du projet. Il a été très vite décidé que ceux-ci se feraient à la souris en cliquant sur la map.

Dernière spécification dans cette partie : il nous semblait plus facile pour plusieurs choses de gérer une map quadrillée. Le choix des cases a donc été validé.

1.2.2 Les déplacements

Pour permettre les déplacements à la souris, il était nécessaire d'implémenter un algorithme efficace de pathfinding. Notre choix s'est porté sur l'A*. En effet, il semblait adapté au type de jeu et à la map utilisée.

1.2.3 IA

L'intelligence artificielle a pour but de donner un peu de vie au jeu. Initialement, il était prévu de définir différent type d'ennemis, chacun ayant leur propre attitude face aux actions des joueurs.

1.2.4 Editeur de map

L'éditeur de map est un logiciel permettant de nous faciliter la tâche lors de la création des différentes map du jeu. Il était prévu de lui ajouter plusieurs options afin de le rendre utilisable par les joueurs pour créer leur propre map.

1.2.5 Moteur graphique

C'est un peu par défaut que nous avons choisi d'utiliser DirectX plutôt qu'OpenGL. Personne ne sachant réellement les utiliser et ne connaissant ni les avantages ni les inconvénients des deux.

2 Alexandre

2.1 De la première à la troisième soutenance

2.1.1 A*

Par goût personnel, je me suis très rapidement éloigné de la partie graphique du projet pour m'intéresser au physique. La première chose qui m'a attiré a été les pathfinding.

Nous avions très vite défini ce que serait, dans ses grandes lignes, le gameplay du jeu. Les déplacements à la souris à l'image d'un jeu de stratégie nécessite un algorithme qui calcul le chemin. Je me suis renseigné sur les différentes possibilités offertes par les algorithmes existants. L'A*, grâce à son implémentation relativement simple et à ses performances sur des map dynamiques de petite taille, semblait adapté à nos attentes.

Son développement a été rapide puisqu'il a pu être présenté dès la première soutenance sous la forme d'une petite application 2D. Toutefois, sont implémentation réelle dans le jeu en 3D a, quant à elle, tardé un peu plus. En effet, son utilisation posait plusieurs problèmes, principalement de complexité. Ainsi, lors des multiples utilisations, le jeu était soumis à de nombreux ralentissements. Avec l'aide d'Antoine qui s'est intéressé aux threads, nous avons pu limiter ces ralentissements et garder un jeu relativement fluide.

De plus, plus le projet avançait et plus il me paraissait évident que l'A* à lui tout seul ne suffirait pas à gérer l'intégralité des déplacements. Il fallait lui ajouter des "options" applicable en fonction du personnage

souhaitant se déplacer. En effet, il y avait des contraintes de vitesse, de zone d'action, de cible, de direction...

La dernière difficulté a été d'utiliser le picking pour déclencher l'A*. Cette tâche fut celle de Fabien.

Lors de la première soutenance c'est donc une simple application graphique sur un tableau en 2D qui permettait de voir le résultat de l'A* qui fut présentée. Son fonctionnement était simple : un tableau à deux dimensions sur lequel l'utilisateur pouvait placer à sa guise des mûrs, un départ et un but. En cliquant sur le bouton "A*", l'algorithme calcul le chemin allant du départ au but. Ce chemin, s'affiche alors sur le tableau. Sur cette application, on pouvait déjà voir le temps de calcul conséquent que prenait cette première version de l'A*. Il était donc nécessaire de l'optimiser.

Une première tentative d'implémentation en 3D avait aussi été présentée. Celle-ci était très minime puisque l'A* ne pouvait se calculer encore une fois qu'à partir de l'éditeur de map en 2D d'Antoine. Ainsi, une fois que la map se chargeait en 3D dans DirectX, le chemin préalablement calculé était appliqué sur le personnage de la map. Ce premier essai, bien que peut évolué, nous a permis de faire nos premières dents avec les déplacements 3D et nous a aidé à mieux aborder l'implémentation 3D de l'A* pour la suite.

Pour la deuxième soutenance, nous avons continué le travail d'implémentation. Le picking n'étant pas encore disponible, il a fallu utiliser un curseur qui se déplaçait à l'aide des touches du clavier sur la case où l'on souhaitait déclencher l'A*. Ce dernier a été d'ailleurs recodé pour l'optimiser par rapport à la version de la première soutenance. Ainsi, il était possible, à l'aide de ce curseur, de se déplacer où l'on voulait sur la map.

La troisième soutenance a été marquée par l'arrivée du picking. Antoine c'est chargé de son implémentation. Quant à moi, il a fallu que je m'occupe de le rendre définitivement fonctionnel dans le projet. En effet, les versions précédentes fonctionnaient mais ne permettait pas d'ajouter comme on le souhaitait divers éléments dans le projet sans avoir besoin de modifier le code de l'A*. Ainsi, je me suis mis aux "class" dans Delphi pour obtenir un objet A* utilisable très facilement avec divers éléments dont plusieurs pouvaient l'utiliser en même temps sans trop ralentir le jeu.

2.1.2 IA

L'I.A était une partie que je voulais relativement ambitieuse dans le projet. Le but était de créer une interactivité entre les ennemis et les joueurs. Les ennemis devaient réagir en fonction de ces actions. De plus, j'ai très vite pensé à utiliser différents type d'ennemis pour encore augmenter la vie dans le jeu. Parmi les types d'ennemis, voici ceux qui me semblaient intéressant :

- Cypher : des espions qui peuvent voir loin et repérer le joueur mais qui ne peuvent pas attaquer. S'ils repèrent le joueur, ils fuient et ils appellent des Kamikazes pour l' attaquer.
- Kamikaze : des guerriers qui n'ont peur de rien, ils combattent jusqu'à la mort quoiqu'il arrive. Ils peuvent repérer le joueur mais sont plus souvent utiliser en coopération avec les Cyphers.
- Warrior : des soldats indépendants qui font des rondes un peu partout dans la map. S'ils repèrent le joueur, ils l'attaquent. Ils peuvent être amenés à fuir s'ils sont en difficulté.

Ces personnages utilisent eux aussi l'A* pour se déplacer, d'où la nécessité d'optimiser ce dernier.

Mon travail pour la première soutenance étant principalement destiné aux pathfinding, l'I.A ne fit sont apparition qu'à la deuxième. Les premiers ennemis qui ont été développé étaient les Cyphers et les Kamikazes. Une application avec ces deux ennemis a été présentée lors de cette soutenance. On pouvait diriger à l'aide de l'A* et du curseur le joueur. Sur la map si trouvait un Cypher mis en évidence et un Kamikaze caché derrière un mur. Lorsque l'on s'approche trop du Cypher, celui-ci fuit et appelle le Kamikaze qui vient à notre rencontre. Leur comportement était très basique mais ça m'a permis de me donner une idée du travail à effectuer.

Initialement, il était prévu pour la troisième soutenance d'avancer l'I.A. Malheureusement, l'A* m'a demandé un trop gros travail pour l'adapter intégralement dans le projet et le manque de temps ne m'a pas permis de présenter une IA fonctionnelle. Toutefois, le code avait bien avancé et cela me mettait sur la bonne voie pour la soutenance finale.

2.2 Quatrième soutenance

Le but de cette soutenance est de présenter un produit terminé. Pour cela, il a fallu essayer de terminer l'ensemble des parties. Pour ma part, l'A*, l'I.A mais aussi les combats ainsi que participer à l'implémentation du réseau.

2.2.1 A*

L'A* était complètement intégré au projet. Mais le trop grand nombre d'utilisation, nous a obligé à implémenter des threads. Cela a été réalisé avec Antoine dès le début de la coding week. Il a aussi fallu l'implémenter dans le réseau, ce qui, encore une fois, a été fait avec Antoine.

Finalement, l'utilisation de l'A* s'est faite progressivement tout au long du projet mais n'a pas posé énormément de problème. C'est le premier aspect complètement terminé du projet.

2.2.2 IA

Après la non-présentation de cette partie lors de la troisième soutenance, il a fallu procéder à une grosse séance de débug. Ensuite, l'objectif était d'étoffer au maximum les réactions des ennemis. Par exemple, quand on en attaque un, celui-ci fuit ou riposte selon son type et sa situation. De plus, lors des attaques, les Kamikazes se différencient des Warriors en attaquant au corps à corps. Le manque de temps m'a empêché de développer autant de possibilité que je l'aurai souhaité, mais le résultat reste tout de même amusant. Les ennemis peuvent réellement mettre en difficulté le joueur, notamment avec les attaques en groupe des Kamikazes.

2.2.3 Combat

Des ennemis contrôlés par l'ordinateur impliquent l'implémentation de fonctions de combat. Celles-ci permettent aux joueurs d'infliger des dégâts aux ennemis jusqu'à leur mort, et inversement. De plus, selon les types d'ennemis, ils font plus ou moins de dégâts.

Pour faciliter les combats, le joueur a la possibilité de locker un ennemi. Grâce à une implémentation faite en commun avec Jonathan, la vie de l'ennemis locké est représentée graphiquement sur l'écran de jeu.

2.2.4 Gestion des caractéristiques

Comme tout bon RPG, il fallait que le joueur puisse évoluer et que ses caractéristiques évoluent avec lui. Ainsi, un système d'expérience a été développé en collaboration avec Jonathan. Chaque ennemi tué

donne un certain nombre de points d'expérience au joueur, qui gagne un niveau une fois une barre d'expérience remplie. Le gain d'un niveau met à jour des caractéristiques tel que la vie ou la force. Pour chaque caractéristique il existe une barre qui donne son état d'avancement dans un des menus qui ont été intégré au jeu par Jonathan.

La barre de vie est aussi visible sur la fenêtre de jeu principale. Elle diminue en fonction des dégâts mais aussi en fonction de l'armure du joueur. Bref ! J'ai essayé de lier les caractéristiques entre elles pour donner un résultat assez amusant.

2.2.5 Réseau

L'un des objectifs de cette soutenance était aussi de présenter un réseau fonctionnel. Développé par Antoine, je l'ai aidé à implémenter l'I.A ainsi que les combats. Finalement, le jeu en réseau se trouve être différent du jeu solo. En effet, il s'agit simplement d'un mini jeu à jouer entre amis. Il consiste en un concourt de kill : le gagnant est celui qui a obtenu le plus de point en tuant des ennemis. Le réseau a été testé jusqu'à 4, mais il est probablement fonctionnel avec plus de joueurs.

2.3 Conclusion partielle

Mon travail au cours de ce projet s'est dès le début centré sur le côté physique. Il m'a permis de beaucoup progresser en code, mais aussi de comprendre pas mal de choses sur le plan algorithmique.

Mon travail au cours de ce projet s'est dès le début centré sur le côté physique. Il m'a permis de beaucoup progresser en code, mais aussi de comprendre pas mal de choses sur le plan algorithmique.

3 Antoine

3.1 Première soutenance

3.1.1 Editeur de map

Pour tout jeu évolutif un éditeur de map est obligatoire. De plus notre jeu se jouant en réseau il faudra bien créer ses propres map. Le but a été ici de faire un éditeur de map simple d'utilisation tout en étant très complet. On peut bien sur charger et sauvegarder des map. Mais sa puissance ne s'arrête pas là car les joueurs peuvent y additionner leurs propres objets. Une base de données contenant tous les objets avec leurs caractéristiques a donc été crée et l'utilisateur peut en 3 clics ajouter un objet en précisant le nom, le type ainsi que les caractéristiques correspondant au type de l'objet. Pour créer une map il suffit au début de choisir la taille, ensuite avec une liste déroulante on accède à tous les types pour pouvoir enfin choisir le type d'objet à mettre sur la map. Il suffit enfin de cliquer sur la ou les cases de la map où l'on veut rajouter l'objet. Une simple clique droite suffit pour les supprimer. Pour les objets, plus complexes comme les boss, les livres ou les "gentil petit asus" une fenêtre s'ouvre quand on la ajoute pour pouvoir, pour les livres ajouter du texte, et pour les boss ou les "gentil petit asus" ajouter des scripts en asus++. De plus quand on rajoute un ennemi, un coffre ou un boss on peut préciser l'objet à ramasser à la mort ou l'ouverture de celui-ci. On peut aussi contrôler la quantité d'argent que l'on met ou la mettre en aléatoire. Enfin on peut choisir l'orientation des objets que l'on pose sur la map (utile pour les portes !). Les map peuvent aussi comprendre plusieurs étages et la navigation entre ceux-ci est facilitée par l'ajout d'onglets et de raccourcis claviers pour les créer ou les détruire. Une des parties les

plus dures de cette réalisation a été de trouver un format de fichiers facilement exploitable.

3.1.2 Affichage 3D

Notre jeu étant en 3D il a fallu s'intéresser à directx et à l'affichage des formes en 3D dans une form Windows. Après les recherches et essais de Jonathan je m'y suis intéressé pour pouvoir créer une interface 3d chargée à partir d'une map. N'ayant encore très peu d'objets à afficher j'ai du créer un éditeur de map très simple ou on pouvait juste mettre le départ, ajouter des murs, dés étages et des escaliers. J'ai donc implémenté la gestion des caméras faite par Jonathan ainsi que les premières bases construites par Jonathan. J'y ai ajouté un mesh plus beau avec des textures ainsi qu'une skybox. Celle-ci était en fait une sphère recouverte d'une texture bleutée qui faisait penser à du ciel. Pour cette première map les déplacements des joueurs étaient gérés au clavier. Le plus dur ici a été de pouvoir charger une map crée par l'éditeur de map et d'afficher les murs, les escaliers et le sol au bon endroit ainsi que de récupérer les touches pressées pour ensuite faire bouger notre joueur. Mais il fallait aussi tester les positions du joueur pour le faire monter ou descendre d'un étage s'il était sur un escalier ou pas. Ce début de 3D assez simplistes a été très formateur pour les réalisations des autres soutenances.

3.1.3 Implementation A*

Alexandre travaillant depuis le début de l'année sur un Astar, il est enfin arrivé à une version stable. Ayant déjà fait l'interface de l'éditeur de map j'ai pu implémenter l'Astar dans une form Windows. On pouvait voir une map de 100 * 100 cases ou on pouvait rajouter des murs ainsi qu'un départ et une arrivée. Mais une fois cette interface faite il a fallu appeler l'Astar et afficher les résultats dans la form. On a donc du travailler en commun pour ceci et avons commencé à voir les difficultés de l'implémentation de toutes les parties d'un groupe. Mais

avec de la motivation on arrive à tout, et on a donc pu voir le calcul d'Astar en temps réel et ceci a donc pu aussi servir de tests pour les optimisations futures de l'Astar.

3.2 Deuxième soutenance

3.2.1 Débuts du réseau

Asus Warri ors étant un jeu non seulement RPG mais aussi multi-joueurs (limité à 4 !) je me suis donc intéressé au réseau. Après avoir essayé plusieurs librairies j'ai choisi Indy car elle me semblait la plus facile à utiliser (pour ne pas perdre trop de temps) et adaptée à notre jeu qui ne nécessite pas d'énormes envois de données. J'ai donc tout d'abord créé un chat en ligne de commande pour bien en comprendre le fonctionnement. Puis notre jeu étant en 3D j'ai implémenté un chat dans une fenêtre directx, c'est à dire l'affichage du texte avec directx. Ceci a été simplifié grâce à notre approfondissement dans la programmation orientée objet. On avait donc ici le choix d'être serveur ou client et ensuite de jouer en communiquant avec le chat de fenêtre. Mais le réseau ne s'arrêtait pas là car les joueurs pouvaient se voir et évoluer sur la même carte. On pouvait donc déjà commencer à rêver de quêtes effectuées en réseau avec plusieurs joueurs comme dans les grands MMO ! Mais cette avancée était entachée d'un certain problème : les joueurs se dédoublaient souvent, ceci du à un problème de gestion du mouvement de joueur.

3.2.2 Premiers menus

Comme tout jeu existant il a fallu créer des menus simples d'utilisations et graphiquement "beaux". J'ai donc du afficher une texture sur l'écran ou étaient affichées les options, c'est à dire, grâce au réseau, client ou serveur. En fonction de ce choix il a fallu afficher une

autre texture pour pouvoir prendre les informations désirées, nom de la map et pseudo pour le serveur, pseudo et adresse IP du serveur pour le client. Ces champs remplis il a ensuite fallu traiter comme il fallait pour pouvoir ensuite lancer le jeu avec le réseau. A ce niveau là, n'ayant pas encore découvert les sprites, les textures de menus étaient affichées sur un simple mesh carré qui se positionnait juste devant la caméra.

3.2.3 Intégration de l'IA

Mais le travail avec Alexandre ne s'arrête pas là, car il s'est non seulement occupe de l'Astar mais aussi de l'intelligence artificielle qui régissait les ennemis. Il a donc crée 2 types d'ennemis, les kamikazes et les espions. Le but, ici, a été de faire un rendu 3D de leurs réactions. C'est à dire qu'ici l'espion devait détecter le joueur dans la map pour ensuite déclencher l'arrivée du kamikaze. On pouvait de plus faire sa map avec l'éditeur pour pouvoir poser les ennemis et le joueur ou on voulait. Il a donc fallu se plonger l'un et l'autre dans le code de l'autre pour pouvoir implémenter le tout sans problèmes. Le but a été de faire une map accessible pour pouvoir afficher la position du joueur et accessible aussi pour les ennemis et le déclenchement de leur actions. On a donc ici eu une implémentation de l'Astar sur une map avec le rendu 3D. L'Astar était calculé soit pour la fuite de l'espion, l'arrivée du kamikaze ou le joueur qui sélectionnait d'abord la case ou il voulait aller et ensuite appuyait sur entrée pour déclencher le calcul de l'Astar puis le joueur se mettait en mouvement. A part quelques difficultés rencontrées au début par rapport à un calcul de l'Astar qui se faisait tout le temps, il n'y a pas eu de problèmes pendant l'implémentation, ce qui nous a mis en confiance pour la suite du projet. Mais nous a aussi appris qu'il fallait énormément communiquer pour que tout ce que l'on fait soit compatible avec les autres.

3.3 Troisième soutenance

3.3.1 Deuxièmes menus

Les premiers menus étant assez statiques j'ai du en refaire de nouveaux, d'une part plus beaux et d'autres part réactifs aux actions du joueur. J'ai donc fait réagir les zones du menu en fonction de la position de la souris. Les zones survolées devenaient rouges. Le plus dur ici a été d'adapter à toutes les résolutions, j'ai donc crée les zones pour une certaine résolution et j'ai ensuite fait les rapports pour pourvoir actualiser en fonction de la résolution. La réalisation des menus est assez longue car on est obligés de tout "coder en dur" c'est à dire d'anticiper chaque réaction du joueur par rapport à la position de la souris. Pour le dynamisme j'ai ajouté une rotation pour changer de pages de menus. Dès que l'utilisateur change de page de menu, celle sur laquelle il était tourne pour donner place à la page suivante. J'ai rajouté aussi l'option solo dans les menus pour pouvoir jouer seul.

3.3.2 Amélioration du réseau

Cette soutenance, le picking ayant enfin été réalisé et ayant déjà implémenté l'Astar je me suis occupé de l'implémentation de ces deux parties. Ceci a été assez rapide. Mais l'avancée a été que l'on pouvait jouer en réseau avec le picking implémenté. De plus j'avais déjà commencé à regarder comment implémenter l'intelligence artificielle dans le réseau, donc quelques parties avaient été travaillées mais n'étais pas encore utilisées. Quelques améliorations ont été apportées au réseau, notamment plus de problèmes de dédoublement des joueurs. Il y a eu aussi l'arrivée de menus pendant le jeu. J'ai créé une "shoutbox" dans laquelle s'affichaient les messages que les joueurs s'envoyaient en réseau. Celle-ci était déplaçable grâce au clavier et on pouvait la réduire en cliquant sur le coin supérieur droit comme les fenêtres et ensuite la ra-

grandir en cliquant sur une icône. Ceci a été rendu possible grâce à ma découverte des sprites, qui servent à afficher de simples images, malgré quelques problèmes mystiques sur le tracé (les textures s'agrandissent à la puissance de 2 supérieure) après plusieurs heures de travaille rendu était plus qu'acceptable et sera même amélioré plus tard par Jonathan. De plus cette shoutbox servira plus tard à afficher les dialogues avec les "gentil petit asus" et à leur répondre.

3.4 Quatrième soutenance

3.4.1 Menus finaux

Cette soutenance, le picking ayant enfin été réalisé et ayant déjà implémenté l'Astar je me suis occupé de l'implémentation de ces deux parties. Ceci a été assez rapide. Mais l'avancée a été que l'on pouvait jouer en réseau avec le picking implémenté. De plus j'avais déjà commencé à regarder comment implémenter l'intelligence artificielle dans le réseau, donc quelques parties avaient été travaillées mais n'étais pas encore utilisées. Quelques améliorations ont été apportées au réseau, notamment plus de problèmes de dédoublement des joueurs. Il y a eu aussi l'arrivée de menus pendant le jeu. J'ai créé une "shoutbox" dans laquelle s'affichaient les messages que les joueurs s'envoyaient en réseau. Celle-ci était déplaçable grâce au clavier et on pouvait la réduire en cliquant sur le coin supérieur droit comme les fenêtres et ensuite la ragrandir en cliquant sur une icône. Ceci a été rendu possible grâce à ma découverte des sprites, qui servent à afficher de simples images, malgré quelques problèmes mystiques sur le tracé (les textures s'agrandissent à la puissance de 2 supérieure) après plusieurs heures de travaille rendu était plus qu'acceptable et sera même amélioré plus tard par Jonathan.

De plus cette shoutbox servira plus tard à afficher les dialogues avec les "gentil petit asus" et à leur répondre.

3.4.2 Les sons

Cette soutenance, le picking ayant enfin été réalisé et ayant déjà implémenté l'Astar je me suis occupé de l'implémentation de ces deux parties. Ceci a été assez rapide. Mais l'avancée a été que l'on pouvait jouer en réseau avec le picking implémenté. De plus j'avais déjà commencé à regarder comment implémenter l'intelligence artificielle dans le réseau, donc quelques parties avaient été travaillées mais n'étais pas encore utilisées. Quelques améliorations ont été apportées au réseau, notamment plus de problèmes de dédoublement des joueurs. Il y a eu aussi l'arrivée de menus pendant le jeu. J'ai créé une "shoutbox" dans laquelle s'affichaient les messages que les joueurs s'envoyaient en réseau. Celle-ci était déplaçable grâce au clavier et on pouvait la réduire en cliquant sur le coin supérieur droit comme les fenêtres et ensuite la ragrandir en cliquant sur une icône. Ceci a été rendu possible grâce à ma découverte des sprites, qui servent à afficher de simples images, malgré quelques problèmes mystiques sur le tracé (les textures s'agrandissent à la puissance de 2 supérieure) après plusieurs heures de travaille rendu était plus qu'acceptable et sera même amélioré plus tard par Jonathan. De plus cette shoutbox servira plus tard à afficher les dialogues avec les "gentil petit asus" et à leur répondre.

3.4.3 La fin (enfin) du réseau

Améliorant de plus en plus mes connaissances en Indy. On a pu Alexandre et moi mettre en réseau l'intelligence artificielle. Le plus dur, une fois que l'intelligence artificielle fonctionnait a été de la faire apparaitre en réseau. Nous avons décidé de laisser le serveur continuer à calculer l'intelligence artificielle, dont j'ai threadé les calculs. Il a donc fallu désactiver la gestion des ennemis côté client. Il faut donc que quand un ennemi bouge le serveur envoie à tous les clients la nouvelle

position de celui-ci. Mais les clients doivent aussi envoyer leur position quand ils bougent pour que le serveur renvoie la position à tous les autres clients. Une fois les ennemis bougeant sur le serveur et sur les clients il a fallu que les ennemis voient les clients sur la map. Ceci a été le travail d'Alexandre d'intégrer ceci dans son intelligence artificielle. Une fois que les ennemis purent voir les clients (et le serveur !) nous avons donc pu implémenter un mini-jeu qui consistait à tuer le plus d'ennemi possibles sur la map, les kamikazes rapportent 3 points, les warriors 2 et les espions 1 point. Le vainqueur est celui qui a le plus de points quand tous les ennemis sont morts. Cette partie a été à mon goût la plus dure du projet car très longue à déboguer en raison de pleins de problèmes mystérieux qu'il a fallu résoudre un par un, la résolution d'un problème en créant d'autres... Mais que de bonheur quand tout fonctionne !

3.4.4 Début du moteur à particules

Améliorant de plus en plus mes connaissances en Indy. On a pu Alexandre et moi mettre en réseau l'intelligence artificielle. Le plus dur, une fois que l'intelligence artificielle fonctionnait a été de la faire apparaitre en réseau. Nous avons décidé de laisser le serveur continuer à calculer l'intelligence artificielle, dont j'ai threadé les calculs. Il a donc fallu désactiver la gestion des ennemis côté client. Il faut donc que quand un ennemi bouge le serveur envoie à tous les clients la nouvelle position de celui-ci. Mais les clients doivent aussi envoyer leur position quand ils bougent pour que le serveur renvoie la position à tous les autres clients. Une fois les ennemis bougeant sur le serveur et sur les clients il a fallu que les ennemis voient les clients sur la map. Ceci a été le travail d'Alexandre d'intégrer ceci dans son intelligence artificielle. Une fois que les ennemis purent voir les clients (et le serveur !) nous avons donc pu implémenter un mini-jeu qui consistait à tuer le plus d'ennemi possibles sur la map, les kamikazes rapportent 3 points, les warriors 2 et les espions 1 point. Le vainqueur est celui qui a le plus

de points quand tous les ennemis sont morts. Cette partie a été à mon goût la plus dure du projet car très longue à déboguer en raison de pleins de problèmes mystérieux qu'il a fallu résoudre un par un, la résolution d'un problème en créant d'autres... Mais que de bonheur quand tout fonctionne !

3.5 Conclusion partielle

La réalisation de ce projet m'a apporté des connaissances assez énormes en informatique. J'ai pu aussi apprendre à bien travailler et communiquer en groupe, à faire confiance aux autres. J'ai aussi appris à déléguer les tâches à faire même si la grande majorité des décisions se faisaient en commun. Malgré quelques passages éprouvants (Les coding week !) j'en garde de bons souvenirs. Il m'a aussi apporté l'envie de finir les choses commencées, de ne pas laisser tomber quitte à tout refaire ou tout reprendre.

4 Jonathan

4.1 Première soutenance

4.1.1 ASUS ++

La réalisation de ce projet m'a apporté des connaissances assez énormes en informatique. J'ai pu aussi apprendre à bien travailler et communiquer en groupe, à faire confiance aux autres. J'ai aussi appris à déléguer les tâches à faire même si la grande majorité des décisions se faisaient en commun. Malgré quelques passages éprouvants (Les coding week !) j'en garde de bons souvenirs. Il m'a aussi apporté l'envie de finir les choses commencées, de ne pas laisser tomber quitte à tout refaire ou tout reprendre.

Pour remédier à ce problème j'ai pensé à réaliser un système de balisage rapide. Celui comporte 5 types de balises :

- <quest> : celle-ci est réservée aux paroles des gentils petits asus, il peut nous poser des questions via cette balise et nous donner des quêtes.
- <rep> : celle-ci correspond à la réponse du joueur, il peut également choisir entre plusieurs réponses.
- <obj> : cette balise permet de vérifier si le joueur donne un objet a un gentil petit asus.
- <givobj> : cette balise permet l'inverse de la dernière c'est-à-dire quelle vérifie si un gentil petit asus nous donne un objet.

Voila donc le système de balisage en place, on pouvait créer nos script dans l'éditeur de map en ajoutant des alliés, il fallait maintenant pouvoir le lire et l'analyser. Donc j'ai fait en sorte de parcourir le texte de le parser et de rentrer chaque partie dans un tableau dynamique de tel sorte que l'on puisse toujours trouver un lien rapide entre la partie de script avant et celle d'après celle qu'on doit gérer. J'ai également créé un petite interface permettant via la console de tester si le script était valide ou pas.

4.1.2 Rendu 3D

Ma principale partie fut de s'intéresser à Directx l'API graphique que nous avions choisie. Cette API nous l'avions choisie en partie à cause de moi. Car je connais depuis longtemps un élève de l'EPITA (un spé de cette année) et il m'a beaucoup parlé et conseillé dans ce sens et m'a bien motivé et donc de ce fait j'ai convaincu les autres membres du groupe d'adopter cette technologie. Du fait de ma motivation je me suis proposé pour la partie graphique du jeu. Avant la première soutenance directx était vraiment obscur pour moi je n'avais encore jamais codé avant d'entrer a l'EPITA. Une fois ce petit choc passé mon objectif était d'ouvrir une fenêtre directx et d'y charger un model 3D. J'ai rapidement trouvé un tutoriel me permettant d'ouvrir une fenêtre et d'y dessiner un triangle. Jusque la pas encore trop de difficultés si ce n'est le mal à m'adapter a utilisée des fonctions déjà codées dans les librairies et de ne pas comprendre leur fonctionnement. Ensuite afin de gérer notre éditeur de map j'ai essayé d'afficher une map pour en importé une de l'éditeur. Mais avant cette première soutenance j'affichais une map mais pas une charger depuis l'éditeur de map c'est Antoine qui s'en est chargé. Pour cette soutenance j'ai également à afficher un mesh texturé (un tigre).

4.1.3 Déplacements

Les déplacements (du tigre) pour cette soutenance s'effectuaient au clavier le mesh pouvait se déplacer dans la matrice du monde dans tous les sens mais ne pouvais pas encore pivoter. Lors de ces premiers pas dans le domaine des déplacements j'ai découvert une petite subtilité du rendu 3D en ce qui concerne les pixels qui se superpose, donc j'ai trouvé un moyen d'afficher les pixels au bon endroit.

4.1.4 Camera

En ce qui concerne les cameras, j'ai fait en sorte de pouvoir également tourner autour du mesh dans tous les sens et de zoomer sur celui-ci.

4.2 Deuxième soutenance

4.2.1 Optimisation du loader de map

Lors de la première soutenance nous pouvions charger une map en 3D mais nous chargions seulement les mesh et les textures sans optimisation particulière plutôt de manière bourrine je dirais. Dans un souci de chargement de la map j'ai du repenser le système de chargement des map qui viennent de l'éditeur. Antoine et moi avons pensé toutes les caractéristiques et tous les type d'objets qui pourraient nous servir et de ce fait on en à conclu un type d'objet map qui pourras contenir tous ce qu'il nous faudra pour chaque objet. Cette soutenance je pouvais sauvegarder toutes caractéristiques des objets par étages car je le rappelle la zone de jeu est un bâtiment à plusieurs étages. Par étage je tris tous les objets, je fais un tableau d'objet map et pour chaque objet j'ai la liste de leurs coordonnées à afficher dans la map. De ce fait on

charger tout les mesh du même type en même temps pour améliorer le chargement et l'affichage.

4.2.2 Refaire le projet en langage objet

Environ un mois avant la 2eme soutenance les spe nous on fait une conférence Delphi sur le langage objet. En voyant comment se déroulait le projet pour une facilité dans le futur je me suis di qu'il serait très très pratique de tout refaire sous forme d'objet car au début le projet n'était pas vraiment clean et partait un peu dans tous les sens. C'est pourquoi j'ai décidé de tout reprendre lors de la coding week avant cette soutenance. Dans un souci de clarté maximum j'ai créé un objet pour chaque partie importante qui s'y prêtait :

- Un objet pour la gestion de la fenêtre.
- Un objet pour l'affichage, le moteur du jeu.
- Un objet pour la gestion des map avec les mesh et tous les objets.
- Un objet pour la gestion de la camera.
- Un objet pour le chargement de la map de l'éditeur de map vers le jeu.
- Un objet pour la gestion des inputs du type clavier et souris.
- Un objet pour la gestion des sons.
- Un objet pour la gestion du texte dans le jeu.

Ces deux derniers objets ont été faits après plus tard dans la coding week, fait car il n'y avait pas de base.

La gestion de tout le projet en langage objet m'a pris une grande partie de la coding week sans compter la phase de débogage qui ne fut pas si compliqué que ça finalement. De la est reparti un projet tout propre. Comme le projet ne pouvais pas vraiment être testé avant la fin les mes 3 confrère ont continué à travailler sur l'ancienne version

qui a été remis en commun peu de temps après la deuxième soutenance par manque de temps nous n'avions pas pu le rassembler pour la soutenance.

4.2.3 Début du son

Une fois ces deux dernières tâches accomplies j'ai voulu regarder dans le temps qui me restait comment jouer des sons. J'ai trouvé principalement comment jouer des musiques d'ambiance, pour cela je me sers de Direct Sound (la librairie de son de directx), à l'époque comme il n'y avait pas vraiment de gameplay ni de scénario j'avais décidé de mettre les sons dans une playlist et permettre d'arrêter, de relancer, de passer une musique.

4.2.4 Affichage du texte

Pour des raisons pratiques, nous avions besoin d'afficher du texte, c'est-à-dire pour les menu qui venait moche qui d'apparaitre et pour le réseau pour choisir l'adresse IP ou afficher du texte dans la barre en bas de la fenêtre qu'Antoine.

4.3 Troisième soutenance

4.3.1 Gestion des objets dans la map

Comme dans les vrais RPG notre personnage a un inventaire sur lui avec un nombre limité d'emplacements. Je n'ai pas pu m'en occuper avant car il me fallait le picking de Fabien pour pouvoir sélectionner un objet posé sur le sol et le rajouter dans notre inventaire. Comme c'est pendant la coding week que le picking est arrivé j'ai pu m'en occuper, à ce moment il était question de sélectionner un objet de

se déplacer jusqu'à lui et ensuite de la ramasser. Pour cela j'ai trié les objets en trois catégories les objets statiques, dynamiques et les conteneurs. Sachant que pour les objets statiques je ne faisais aucun traitement supplémentaire dessus. Pour les conteneurs le joueur arrive devant et une petite fenêtre s'ouvre pour que celui choisisse ou non de prendre l'objet, pour les objets dynamiques sont les objets trouvés à même le sol.

4.3.2 Le brouillard

Pour améliorer l'affichage car nous avons malgré nous un fps pas super élever alors pour remédier a ce problème nous avons décidé d'afficher du brouillard comme cela nous pouvons ne pas afficher totalité de la map tout le temps sans que cela ce voit. Cette technique nous permet également de limiter la distance d'affichage car, dans un souci de gameplay, nous ne voulons pas que le joueur découvre la map trop vite.

4.4 Quatrième soutenance

4.4.1 Gestion des caractéristiques

Lors de cette soutenance avec l'intelligence artificielle et les combats, il devenait indispensable d'avoir un menu de caracteristiques pour gérer les caractéristiques des joueurs. Pour ce faire j'ai du créer un équipement pour le joueur qui améliore ses caractéristiques Pour cela je me suis intéressé au sprite sachant qu'Antoine avait déjà un peu travaillé dessus pendant la 3eme soutenance ce fut assez simple pour moi de m'y mettre. Alors maintenant sur la touche « c » comme caractéristique on peut afficher un menu qui gère en temps réel :

– La vie
– L'expérience
– La force
– La vitesse
– La discrétion
– Les dégats
– La portée de l'arme
– L'armure

Maintenant le joueur perd de la vie quand il se fait attaquer, il gagne de l'expérience en tuant des méchants ou en réalisant des quêtes pour des petits asus. Les processeurs nous permettent de gagner en force, la carte graphique en discrétion, la ram en vitesse, les disques durs en emplacement dans notre inventaire et les armes modifient les dégâts et la portée de l'arme.

4.4.2 Gestion des données

Pour que le joueur à tout moment puisse voir ça vie et l'ensemble de ses caractéristiques j'ai créé un menu que l'on peut afficher grâce à la touche « c » ce menu affiche sous forme de barre de progression, mais également numériquement avec une valeur maximum actuel et minimum, de la vie, de l'expérience, de la force, de la vitesse, de la discrétion ainsi que le niveau du joueur ses dégâts et portée de son arme. Il peut aussi voir les emplacements libres, c'est-à-dire ou il peut encore équiper un objet.

Maintenant le joueur peux voir son équipement à l'aide de la touche « p » de plus super pratique si le joueur passe la souris sur l'icône de l'objet il verra ses caractéristiques.

L'inventaire est également disponible avec les infos bulles de l'objet, on peut également voir combien il nous reste d'emplacement libre et notre porte monnaie.

4.4.3 Gestion menus in game

J'ai créé un menu « echap » avec les sprite animés d'Antoine le menu contient le menu option une possibilité de revenir au menu principal de quitter et bien sur de continuer.

J'ai amélioré le design de la shoutbox et j'ai également fait en sorte qu'elle se déplace a la souris c'est plus marrant.

Maintenant en bas de la fenêtre on a une petite barre d'interface qui permet de voir le pseudo du joueur la photo de l'ordinateur que l'on dirige ainsi que notre vie. Sinon cette barre sert également lorsque que l'on sélectionne un ennemi ou un gentil petit asus auquel cas on affiche son type ainsi qu'une image leur correspondant et leur barre de vie.

4.4.4 Gestion Surcouf

J'ai aussi fait en sorte que l'on puisse dépenser l'argent que l'on récolte a chaque étage, on peut le dépenser en prenant un cône téléporteur entre chaque étage on se retrouve chez Surcouf et on peut faire le plein d'équipement.

5 Fabien

Pour ma part dans, le projet de cette année, j'ai participé essentiellement au développement du moteur physique de notre jeu. Le début étant assez laborieux, ce fut Jonathan qui me permit de réellement commencé à travailler, en effet a partir du moment où il a réussit à afficher un mesh sur une carte, j'ai pu commencer à gérer les mouvements et la gestion de la caméra. Au départ la gestion des déplacements se faisait au clavier, mais cela impliquait que les mouvements du joueur était indépendant de l'angle de vue de la camera.

5.1 Le picking

Il nous est vite apparut que tout jeux de rôle sur PC, nécessitait un déplacement à la souris de notre joueur, je me suis donc mis à la tache d'implanter un picking fonctionnelle.

J'ai d'abord pensé à faire une simple projection du plan de la souris sur la map en effectuant des calculs de projection. Cette technique ne donnait de bons résultats que sur un périmètre de 2 cases autour de notre joueur, au delà de cette limite le résultat était pour le moins surprenant.

Après moult essais, j'ai réussi à transformer les coordonnées 2D du pointeur en coordonnées 3D, mais les résultats obtenus n'étaient pas encore ceux escomptés. J'ai donc remanié ma formule afin de pouvoir sélectionner avec précision la case ou l'on voulait se déplacer.

Pour ce faire, j'ai dû apprendre à manipuler les matrices, qui jusqu'alors m'était inconnues, avec des calculs complexes j'ai réussi à at-

teindre mon but avec une précision de deux cases autour de celle que je voulais réellement atteindre.

A ce stade je me suis aperçu que je pourrais peut être en effectuant des calculs a partir des différente matrice, que l'on utilisait pour afficher notre jeu, pouvoir trouver les bonne coordonne. J'effectuais mon calcul à partir de la matrice de projection, de vue, du monde, de perspective. La zone d'action valide de cette fonction était élargi par rapport à a première version de 5 case environ.

Mais cette formule ne correspondait pas à nos attentes, d'une part trop lourde à gérer au niveau de la mémoire et d'autre part ayant précision qui n'étant pas suffisamment fine pour pouvoir jouer.

Recommençant mes recherches, avec un peu plus de connaissances, j'ai réussi à trouver des fonctions qui me permettaient d'effectuer mes lourds calculs matricielles sans perte de mémoire, dès lors j'ai pu présenter à mon coéquipier un picking fonctionnelle.

Pour pouvoir convertir les coordonnés 2D en 3D je construis 2 point dans l'espace 3D, un proche du plan de la camera et un lointain. A partir de c'est 2 point je construis 2 vecteur qui lorsqu'il croise le plan ou évolue notre joueur définisse les nouveau coordonne de ce dernier ;

Je me suis ensuite essayé à la création de modèle pour que notre jeu ait un rendu graphique acceptable. Devant mes pitoyables résultats j'ai vite abandonné l'idée de nous créer de joli mesh à afficher. Quand a faire des vidéo d'introduction comment dire : je suis venu, j'ai vu et ça ne ma pas plus. Non en fait il y a une un essai avec un cube mais ce n'étais pas possible je n'arrivais même pas a faire bouger la vidéo comme je voulais, et étant donné que l'on n'avait pas de joli modèle

j'ai remis cela a plus tard, et comme on n'en a pas eu je ne l'ai pas fait.

5.2 Les travaux graphiques et le moteur à particules

Pour permette à mouton de travailler sur ses menus, j'ai dû lui créer des boutons pour qu'il puisse les affichés.

Les combats étant enfin implémentés, j'ai donc crée, sous la forme d'un moteur de particule, des animations lorsque l'on combattait les ennemis sur notre map.

Ce moteur de particule me permet de créer un nombre de particule donné et de les déplacées dans la map.

Le moteur de particule fut rapidement mise en place, ce qui a permit de m'attarder un peu sur les différentes utilisations que l'on pouvait en faire ainsi, on l'utilise aussi lorsque notre personnage prend un niveau ou encore lorsque les ennemis nous attaque.

Ceci étant fait, et ayant géré les fuites de mémoire occasionnées, je me suis occupé de l'interaction que l'on pouvait avoir avec nos alliés.

5.3 Les alliés

En chargeant notre map, nous définissons pour nos alliés des réponses prédéfinies qui selon les réactions du joueur pourront ou ne pourront pas être affiché. Nous avons décidés de donner au joueur le choix entre des réponses prédéfinies pour pouvoir créer le scénario du jeu. Ainsi l'utilisateur aura le choix entre plusieurs possibilités pour faire évoluer le jeu à sa guise.

Jonathan ayant déjà codé un système de script pour gérer le dialogue avec ces derniers il m'a suffit de ré implémenter le système de question/réponse de l'asus++.J'ai ensuite réutilisé ce système afin que les personnages neutres puisse nous donner des informations ou des quêtes pour avancer dans le jeu. Il a fallu pour cela que je repense la manière d'écrire nos scripts et que j'intègre, lorsque l'on effectue une quête les interactions avec notre monde.

J'ai opté pour une solution simple de gestion des quêtes par l'utilisateur, il va parle à un allié si l'allié a une quête disponible sois le joueur l'accepte soi il refuse. Si il accepte la quête un menu se lance dans lequel le joueur peut voir quel est le but de sa quête et ou il en est.

Comme le script était dans notre map à l'état brut il a d'abord fallu le traité et mettre les relations d'ordre a jour, pour permettre à ma fonction de parcourir un seul chemin du script. Or ce chemin incluait de pouvoir récupérer la réponse de l'utilisateur tous ceci sans ralentir le jeu. Cette attente de réponse incluait de pouvoir sortir du script traité sans a avoir à recommencer du début mais de la question précédant la demande de réponse.

Le traitement de ce script se fait donc logiquement par un arbre, dont chaque nœud contient l'expression à écrire et dans ses fils les choix de l'utilisateur. Et simplement lorsque l'on récupère la réponse du joueur on reprend le parcours de l'arbre à partir du père du nœud que l'on avait quitte.

Après réflexion je me suis dit qu'il serait intéressant que les alliés puissent nous donner, comme dans les jeux de rôle actuellement dans le commerce, des suites de quête. J'ai donc encore une fois modifie ma gestion du script pour pouvoir le faire.

Nous avions donc des PNJ (personnage non joueur) qui nous donnait désormais des missions, or qui dit quêtes dis récompense de quête, il a donc fallu que je gère, avec les quêtes que l'on nous donnait, les gains d'expérience ainsi que les objets que l'on nous donnait.

Nous avions donc des PNJ (personnage non joueur) qui nous donnait désormais des missions, or qui dit quêtes dis récompense de quête, il a donc fallu que je gère, avec les quêtes que l'on nous donnait, les gains d'expérience ainsi que les objets que l'on nous donnait.

J'ai ensuite renvoyé les donnés que je récupérais dans la chatbox codé par mouton, il a donc fallu que j'intègre les contrainte lié a l'espace ou je voulais afficher mon script.

Après consultation des membres de l'Asus team, il est apparu que le joueur devait avoir la possibilité d'abandonner une quête en cour. Je ne voyais tout d'abord pas comment j'aurais pu faire pour que le joueur ayant déjà accepte une quête puisse l'abandonner n'importe quelle moment, puis en réfléchissant je me suis aperçu que la structure de l'arbre permettait déjà de le faire puisque nécessitant la validation de l'objectif de mission pour continuer.

Le fait de relancer l'arbre en n'ayant pas cette validation me permettait de posez le choix et si l'utilisateur voulait continuer je repartais en attente de la validation de la quête et sinon je réinitialisais l'arbre.

5.4 Conclusion partielle

La réalisation de ce projet m'a apporté la joie des mises en commun d'éléments codé indépendamment, en fait par joie j'attends surtout les désagréments engendrés par ces mises en commun.

Le plaisir de voir se créer sous mes mains un jeu qui au fur et à mesure gagnais en intérêt. Le plaisir de créer pour la première fois un jeu vidéo qui me plaisait, et de voir les jeux vidéo de l'autre coté du miroir. Le plaisir de voir que notre travail évolue et porte ses fruits.

Il y a eu aussi les désagréments liés au débogage de mon code, les heures passés pour se rendre compte que l'on avait fait une petite faute dans un test que l'on faisait.

Ce projet m'a aussi appris à mener à bien un projet informatique sur une année complète, se contraindre a un cahier des charges. Tout cela dans la bonne humeur, malgré les nombreux bugs que l'on a eus et la mort de la première carte mère de mon fidèle ordinateur.

Conclusion

Le projet ASUS Warriors est le premier réalisé par chacun des membre du groupe. Cette expérience semble avoir été bénéfique pour chaucun. En effet, en plus de former le groupe au niveau informatique, il participe à l'apprentissage du travail en équipe et en entreprise.

Sur ce dernier point, il semble que le groupe ait encore des choses à apprendre. Toutefois, certaines erreurs qui ont été commises lors de ce projet pourront désormais être éviter. De plus, il faut reconnaître que globalement, le développement ne s'est pas si mal passé. Il s'agit donc bien d'une réussite à la fois au niveau de l'apprentissage technique que des méthodes de développement.

Ce projet pourra donc servir pour chacun des membres d'exemple pour leurs expériences futurs au sein de l'école ou professionnelles.