Projet d’ISN 2017 :

Abricot Game



# Table des matières :

## 1 - Le cahier des charges

## 2 - Répartition du travail

## 3 - Recherche documentaire et logiciels utilisés

## 4 - Fonctionnement global

## 5 – Ma partie

## 6 - Extensions

## 7 - Conclusion

# 1 - Le cahier des charges

L’objectif de ce projet était de créer un jeu de type MMORPG (jeu de rôle multijoueur en ligne) écrit sous python 3.6, avec comme dépendance pygame.

Pour cela nous avons dû créer deux programmes : un programme client qui se charge de l’affichage des informations et des contrôles (Clics, …) et un programme serveur qui centralise toutes les informations des clients connectés, et qui se charge des calculs (IA des mobs, mouvements sur la carte, combats, …).

De plus, nous avons également mis en place un système de mise à jour automatique du programme client pour que les joueurs n’aient pas besoin de s’en occuper.

Le but final est d’aboutir à un jeu où le joueur peut se déplacer sur différentes cartes, participer à des combats sur ces cartes, interagir avec différents éléments de ces cartes (quêtes, commerce, donjons), après les combats le joueur peut récupérer de l’équipement pour améliorer sa puissance. On envisage également de mettre en place plusieurs classes de joueurs possédant chacune leurs propres caractéristiques, … .

Au lancement du projet, nous savions qu’il ne serait pas possible de réaliser toutes les fonctionnalités avant la présentation, mais nous avions l’intention de continuer à développer le jeu même après la fin du projet d’ISN.

# 2 – Répartition du travail

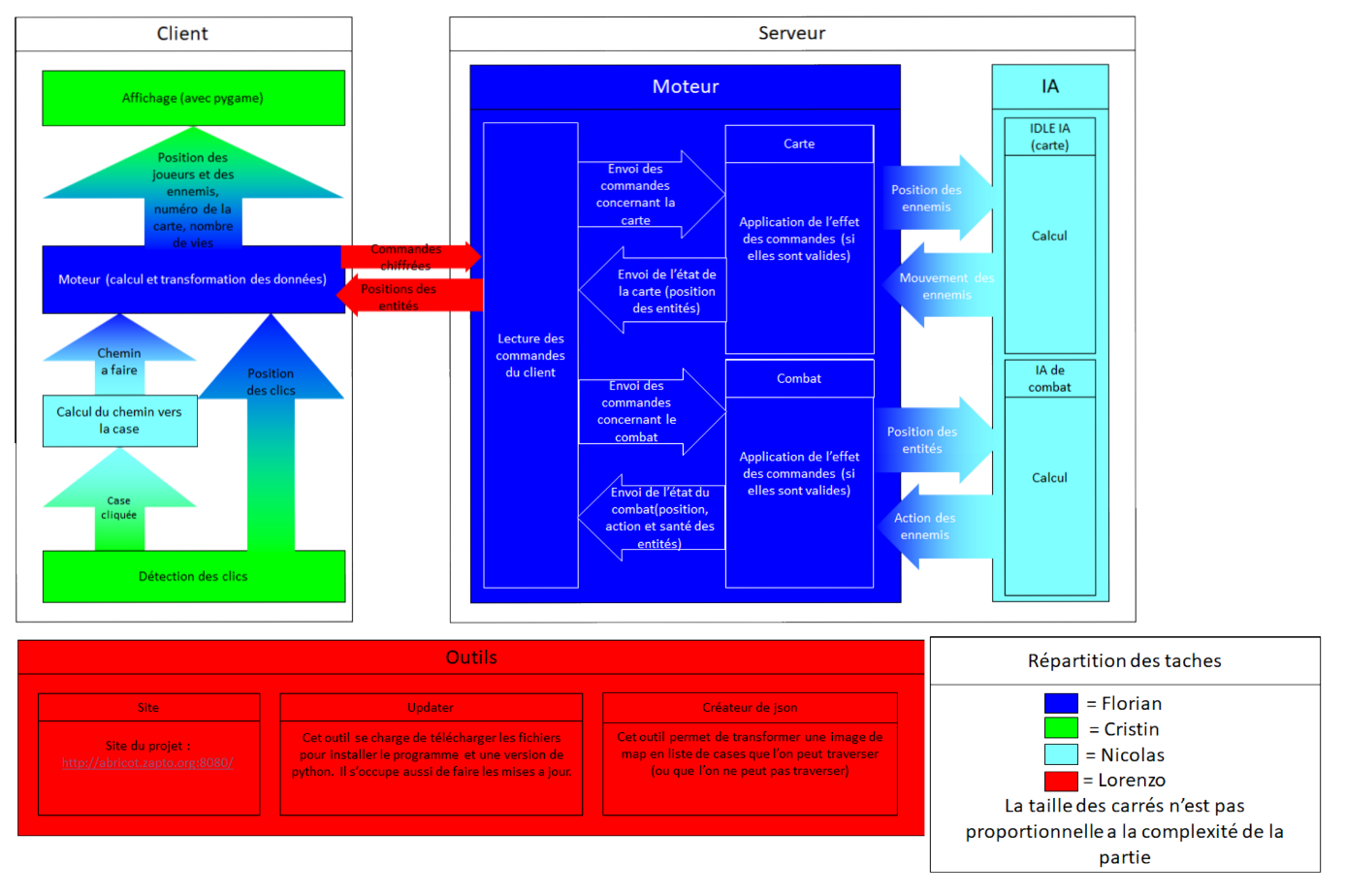
Dès que nous avons commencé notre projet, nous avons décidé de créer un groupe de 4 car nous voulions nous lancer dans un projet complexe.  
Nous avons tout de suite commencé par nous répartir les tâches.

Lorenzo a été responsable de la partie communication : il s’est occupé de transmettre les informations du client au serveur, de la connexion d’un joueur à son compte, de la sauvegarde des données lors de la déconnexion et de la création du site du projet.  
Nous lui avons donné cette responsabilité car il avait déjà travaillé sur les bases de données, les algorithmes de cryptage et l’utilisation du langage PHP. Il a aussi travaillé sur des outils externes comme un générateur de fichiers JSON, qui servent à stocker les informations des différentes cartes.

Cristin s’est chargé de la partie graphique : il s’est occupé de l’affichage des éléments sur l’écran, de la détection des clics et de la création des images.

Nicolas a développé des Intelligences Artificielles (IA) : c’est grâce à lui que les ennemis se déplacent sur la carte et agissent durant les combats. Il s’est aussi occupé des algorithmes de recherche de chemins utilisés par les joueurs et les ennemis.

Enfin, je me suis occupé du moteur : celui-ci se charge de tous les calculs qui ne sont pas faits par l’IA (mouvement des joueurs, dégâts, …) et qui s’occupe de faire le lien entre les différentes parties des autres.



Le fait de travailler à 4 personnes est nécessaire pour faire avancer le projet à un rythme convenable mais entraine des complications : il faut réussir à réunir les 4 membres du groupe au même moment, faire circuler le code et chaque partie étant développée séparément, cela peut entrainer des problèmes lors de la fusion des codes.

Pour résoudre ces problèmes, nous avons dû mettre en place plusieurs éléments :

* Pour le problème de la communication en groupe, nous avons utilisé Facebook et nous avons pu rester le mercredi après-midi dans une salle prêtée par l’école, pour pouvoir avancer de manière rapide.
* Pour le problème du partage du code, nous avons utilisé [GitHub](https://github.com/NB0174/AbricotGame), qui nous permet de travailler à plusieurs sur le même code et de le fusionner ensuite.

Le principal intérêt à travailler séparément sur les 4 parties est de pouvoir avancer sans être dépendant des autres et de se contenter de fusionner son code ensuite.  
Ce qui explique pourquoi chacun d’entre nous a peu de connaissance sur le fonctionnement précis du code des autres.

# 3 - Recherche documentaire et logiciels utilisés

Personnellement je n’ai eu que peu de recherches à faire pour ma partie du projet : en effet je devais surtout faire appel à mes connaissances en python, et je n’avais donc pas besoin de travailler sur un domaine que je ne connaissais pas.  
Les seuls sites que j’ai consultés ont servi à approfondir ma maitrise du langage python et à comprendre la partie de Cristin sur Pygame.

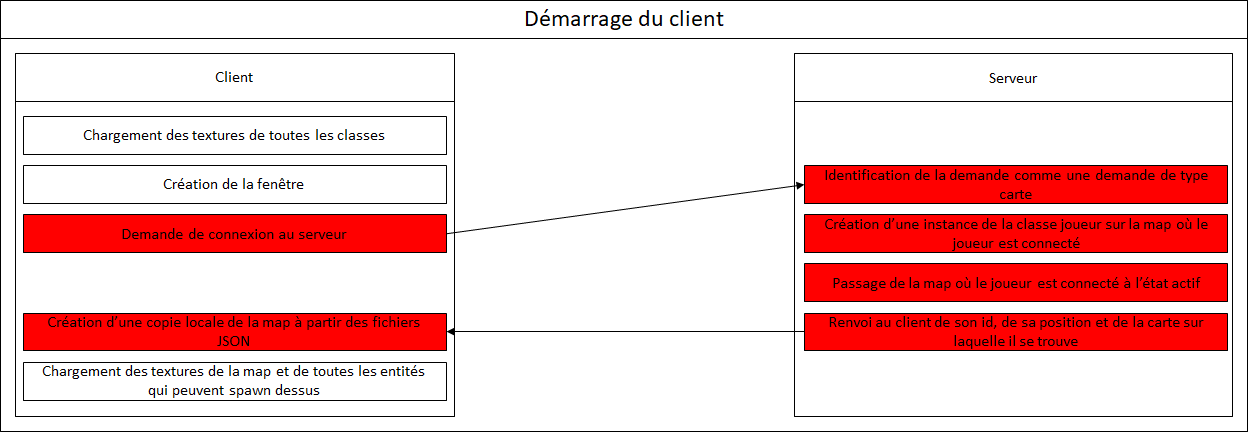
|  |  |  |
| --- | --- | --- |
| Site | Adresse | Pourquoi |
| Documentation de python | <https://www.python.org/doc/> | Pour utiliser les modules typing, copy et enum |
| Documentation de pygame | <https://www.pygame.org/docs/> | Pour vérifier certains éléments |
| Tutoriel python sur OpenClassrooms | <https://openclassrooms.com/courses/apprenez-a-programmer-en-python> | Pour apprendre à utiliser python |
| Tutoriel pygame sur OpenClassrooms | <https://openclassrooms.com/courses/interface-graphique-pygame-pour-python> | Pour des détails sur le fonctionnement des classes et de l’héritage |
| GitHub | <https://github.com/NB0174/AbricotGame> | Pour permettre de travailler à plusieurs sur le même code |

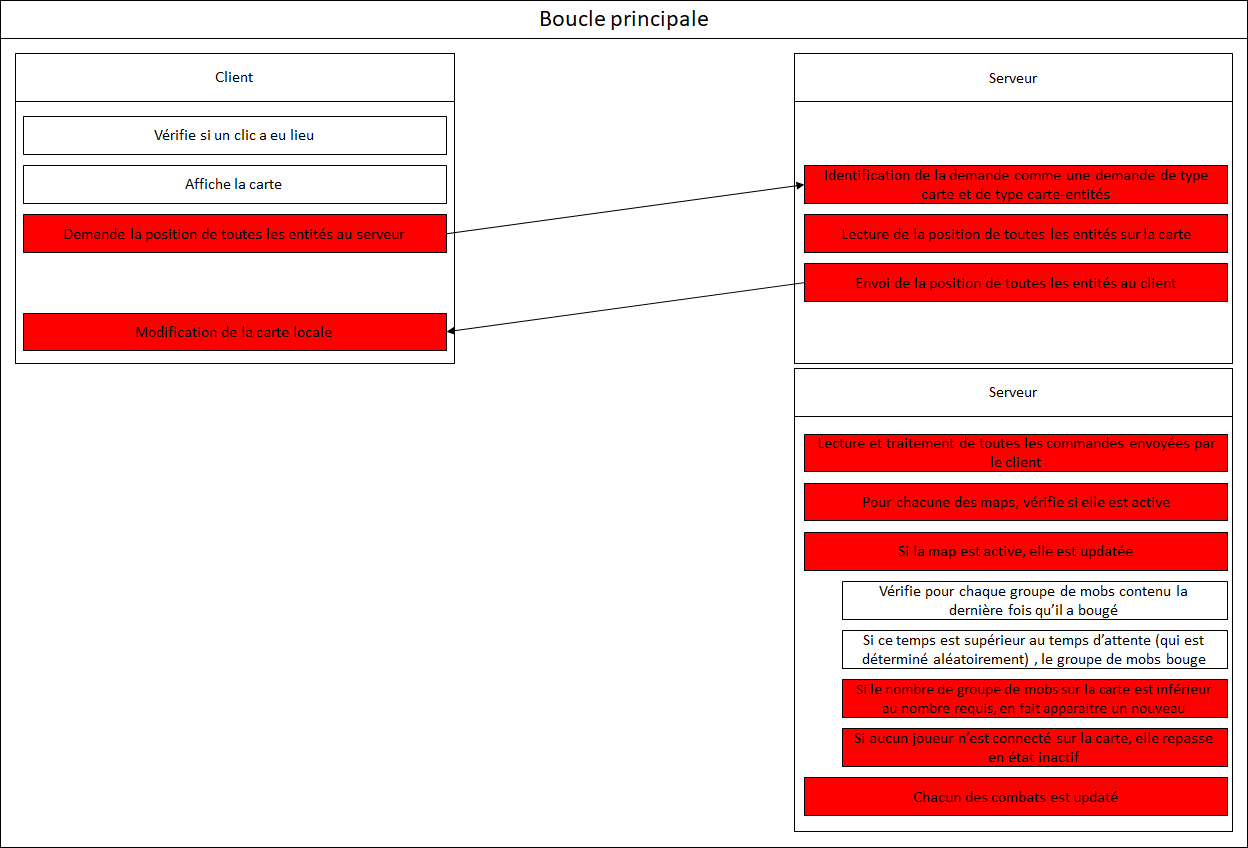
Les logiciels que j’ai eus à utiliser pour ce projet sont

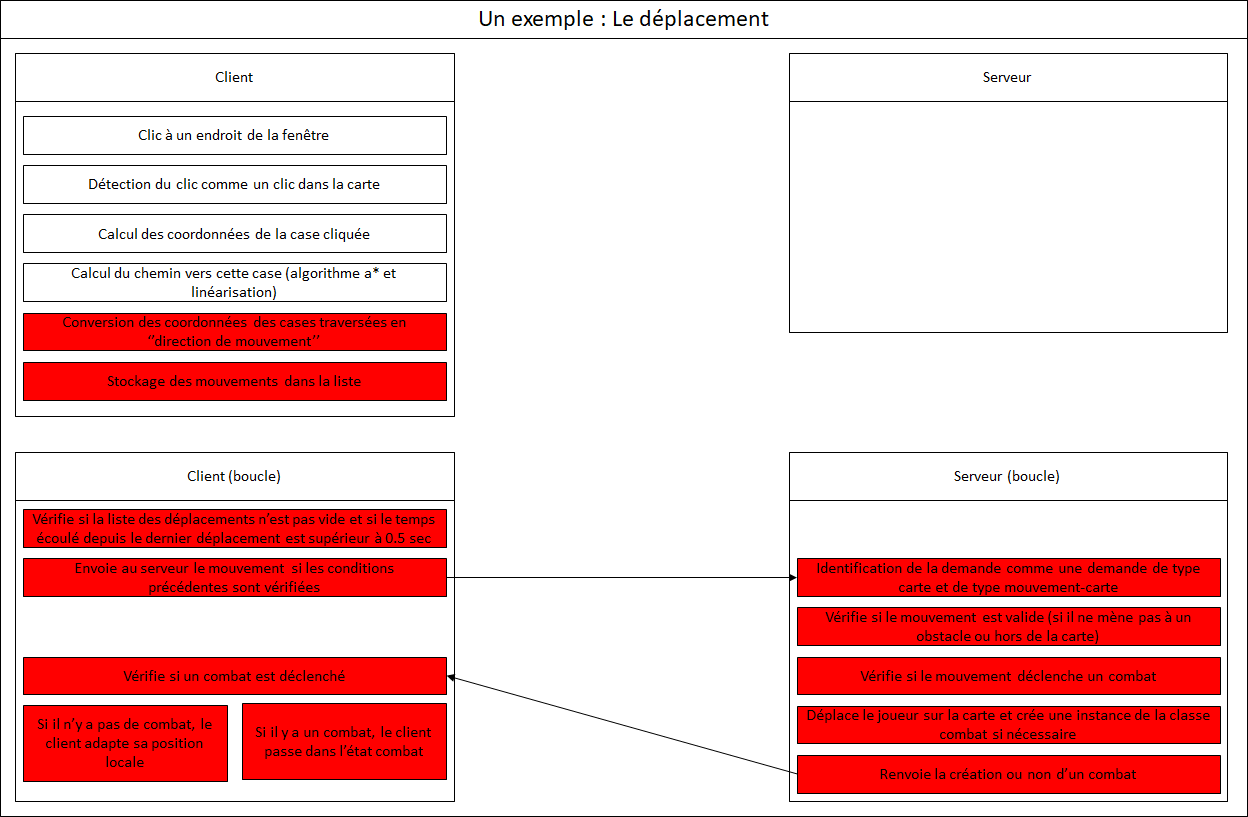
* Pycharm, un éditeur de code python offrant bien plus de fonctionnalités que IDLE, l’éditeur python de base
* GIT : Un système permettant à plusieurs personnes de travailler à distance sur le même code, ce qui nous a permis de nous synchroniser lorsque l’on travaillait
* Un programme écrit par Lorenzo en python servant à générer des fichiers JSON pour conserver les données concernant les cartes.

# 4 - Fonctionnement global

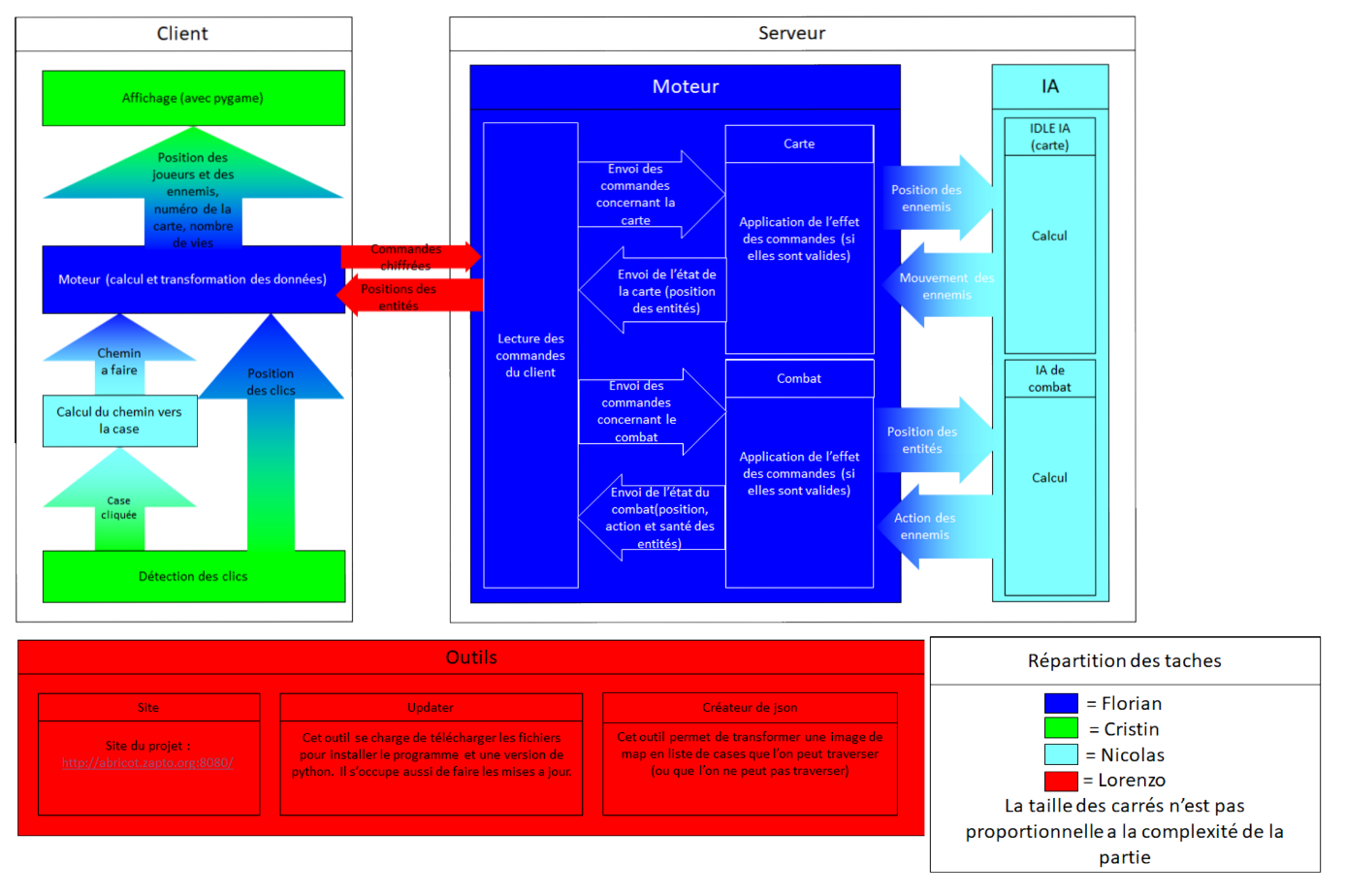
Le fonctionnement est résumé par le schéma suivant :







# 5 – Ma partie

Mon travail a été de créer le moteur du client et du serveur et de m’occuper d’intégrer les parties des autres membres du projet dans la mienne, pour avoir un système fonctionnel à la fin.  
 Je suis donc celui qui connait le mieux les parties des autres membres du projet.

Les moteurs du client et du serveur reposent sur le même principe : ils reçoivent des données, les traitent et les envoient aux parties de Nicolas et de Cristin ou se servent de la partie de Lorenzo pour communiquer entre eux.

Le moteur client fonctionne grâce à une boucle exécutée 21 fois par seconde (1 tick = 0.04761904761 secondes).  
Cette boucle se charge d’appeler la fonction de Cristin chargée de l’affichage, de réagir en cas d’action de l’utilisateur et de demander au serveur l’actualisation des données concernant la position des entités.

Le moteur du serveur a aussi un rôle lors de l’initialisation : il lit à partir des fichiers JSON les informations concernant les cartes, les ennemis et les sorts et les stocke en mémoire.

Le moteur serveur fonctionne avec 42 actualisations par seconde.  
Il reçoit les commandes envoyées par les joueurs et les classe en deux catégories :

* Celles qui concernent la carte et celles qui concernent le combat. Puis il effectue les actions liées à ces commandes : déplacer un joueur sur une carte, lancer un sort, faire changer un joueur de carte, ….
* Ensuite, il fait appel aux fonctions de Nicolas pour déplacer les ennemis et les faire agir en combat.

Par exemple, quand un joueur clique sur une case, le moteur client fait appel à l’algorithme de Nicolas pour calculer le chemin, puis se charge ensuite de décomposer ce chemin en différents mouvements et envoie régulièrement les déplacements au serveur. Ensuite, quand le serveur reçoit une commande de déplacement, il vérifie si la case dans la direction dans laquelle le joueur veut se déplacer est libre et si c’est le cas, il déplace le joueur.

Le serveur sert aussi à faire apparaitre les groupes d’ennemis et à gérer les combats.

# 6 – Extensions

Dès le début du projet, nous y étions très impliqués et nous savions qu’il ne serait pas possible de terminer un jeu de cette envergure dans un délai aussi court que celui imposé par le cours d’ISN et le Bac. Nous avions donc décidé de continuer à développer le projet même après l’avoir rendu.

Un des changements majeurs qui aura lieu après l’oral d’ISN sera de traduire notre programme dans le langage C++ au lieu d’utiliser le python.  
En effet, le C++ possède l’avantage d’être un langage compilé au lieu d’être interprété comme le python, ce qui le rend beaucoup plus rapide.

Comme on souhaite poursuivre le développement du projet, on aura donc besoin de plus de fonctionnalités incluses dans le jeu donc d’un langage plus rapide.  
Il faut aussi compter sur le fait que le nombre de joueurs risque d’augmenter, ce qui nécessitera également plus de rapidité.

Voici une petite liste des fonctionnalités que l’on ajoutera et qui ne sont pas encore prêtes au moment où j’écris ce rapport :

* Un chat entre les joueurs
* La possibilité pour les joueurs possédant un compte administrateur d’entrer des commandes pour pouvoir débugger le jeu en temps réel
* De nouvelles classes (catégories) de personnages
* Un système de montée de niveau en combattant les ennemis pour obtenir de nouvelles capacités
* De nouvelles cartes habitées par de nouveaux types d’ennemis
* Un système d’inventaire et d’équipement pour pouvoir renforcer ses statistiques de combat
* Un système qui nous apportera des récompenses à la fin de chaque combat (équipement, or, ressources, …)
* Un système de fabrication d’équipement à partir des ressources gagnées à la fin de chaque combat
* Un système de commerce utilisant l’argent gagné dans les combats pour faire des échanges avec les autres joueurs ou avec des PNJ (personnages non joueurs)
* Un scénario raconté dans le jeu à travers les quêtes
* Une augmentation du nombre de mécaniques de combats
* Une amélioration de l’IA en combat pour correspondre avec les nouvelles mécaniques
* Un système de donjons
* …

# 7 – Conclusion

En conclusion, je peux dire que ce projet m’a déjà beaucoup apporté car il m’a permis de découvrir les problèmes qui peuvent apparaitre lors d’un travail de programmation en groupe et m’a également entrainé à résoudre ces problèmes.  
Il m’a aussi permis de me perfectionner dans l’utilisation du langage python.

Je me suis aussi intéressé à la partie de Nicolas ce qui m’a permis de découvrir les algorithmes de Bresenham.

Pour conclure, je peux dire que ce projet m’a déjà beaucoup apporté et que j’attends avec impatience les futurs éléments qui vont être développés après notre passage sur le langage c++.

Je remercie aussi l’école Jeanne d’Arc de nous avoir laissé accéder aux salles le mercredi après-midi.