

COPPO



Manuel

URBAIN-BLAUDEZ



TRISTAN



LAP
HEROES



➤ **Langage informatique :**
Python 3.5.2

➤ **Sources d'informations :**
OpenClassroom/
StackOverflow/
« Apprendre à programmer avec
Python 3 »- GERARD SWIMMEN

➤ **Temps de réalisation :**

6 mois

Terrain de jeu :

Le lycée Lapérouse

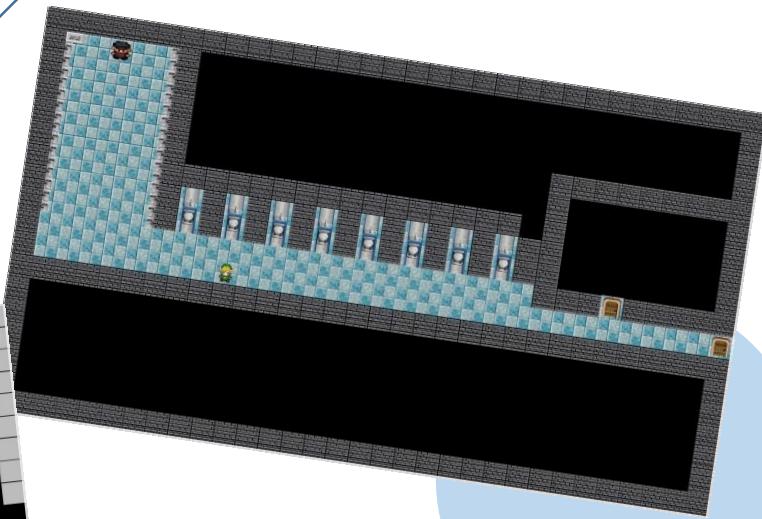
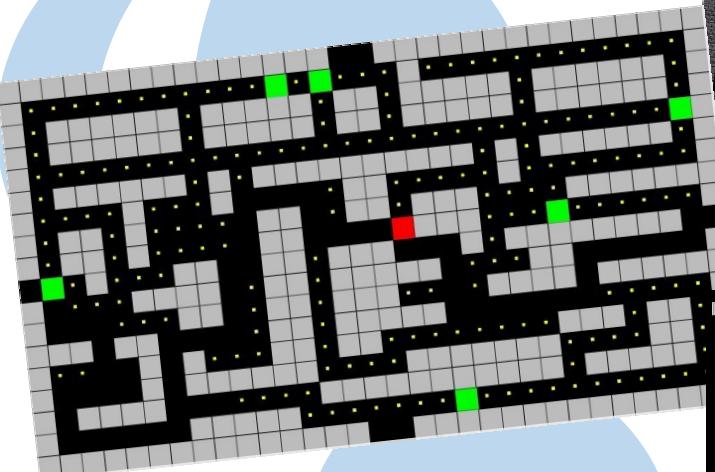
Votre But :

Trouver les profs et réussir leurs épreuves

En ayant accès à la plupart du lycée, vous essayerez de trouver les profs tout en interactuant avec des pnj (player non jouable) et en évitant ou en éliminant des surveillants qui cherchent à vous mettre en colle ! Votre vie est symbolisée par une note/20 qui descend lorsque vous perdez une épreuve ou que vous vous faites toucher par un surveillant.

Prêt à relever

Le défi ?



Notre but ? Nous voulions faire tout simplement un « zelda like » qui se passerait dans le lycée Lapérouse.

Qu'est-ce que ça veut dire ? Une vue du dessus avec des combats à l'épée des interactions avec de la populace, un système de vie, de boss. Des déplacements par bloc, des niveaux complexes...



Avoir des idées ce n'est pas ce qui manque, on a donc beaucoup de choses qui était prévue :

Une histoire, des armes et ustensiles multiples, l'accès à TOUT le lycée Lapérouse, images personnalisées et bien plus... (Indication des choses qui n'ont pas pu aboutir manque de temps)



Une image tirée du jeu vidéo

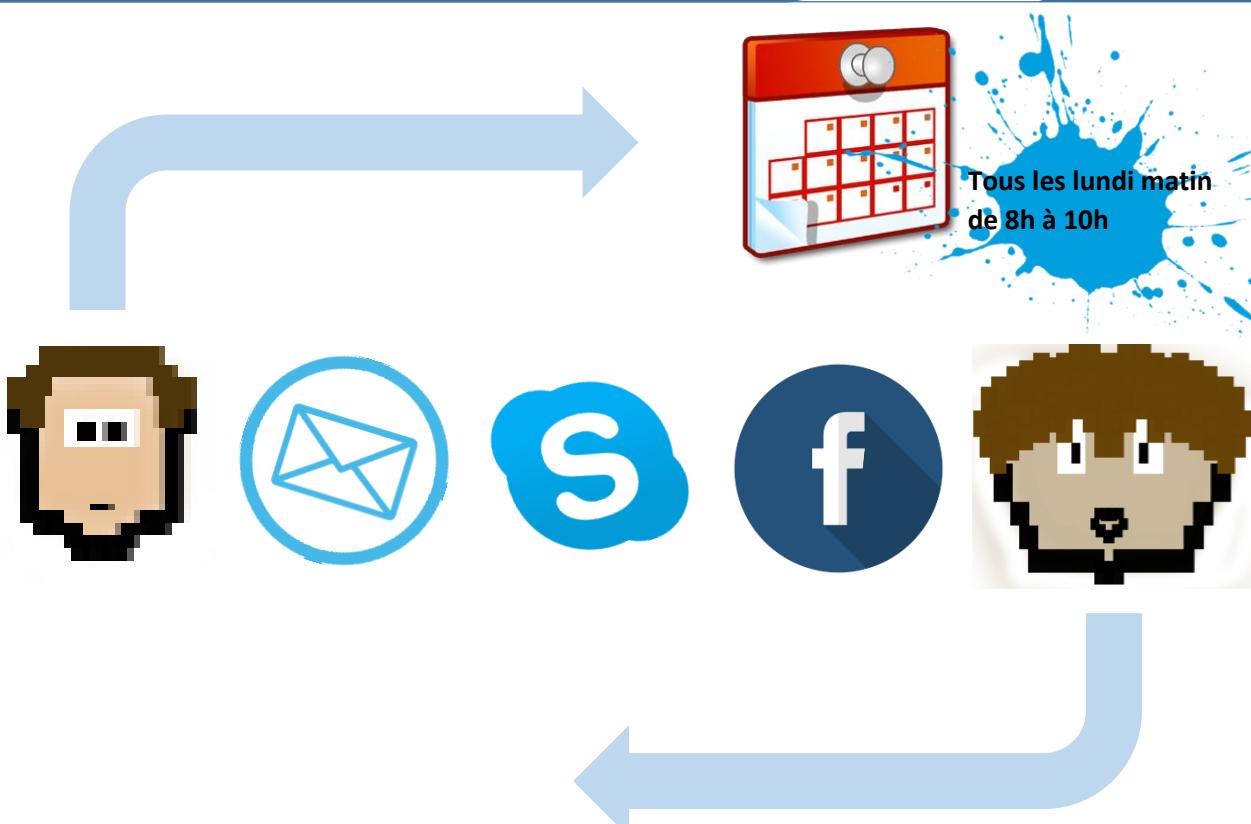
« Zelda III »



Une image de notre jeu affiché grâce au module graphique Tkinter

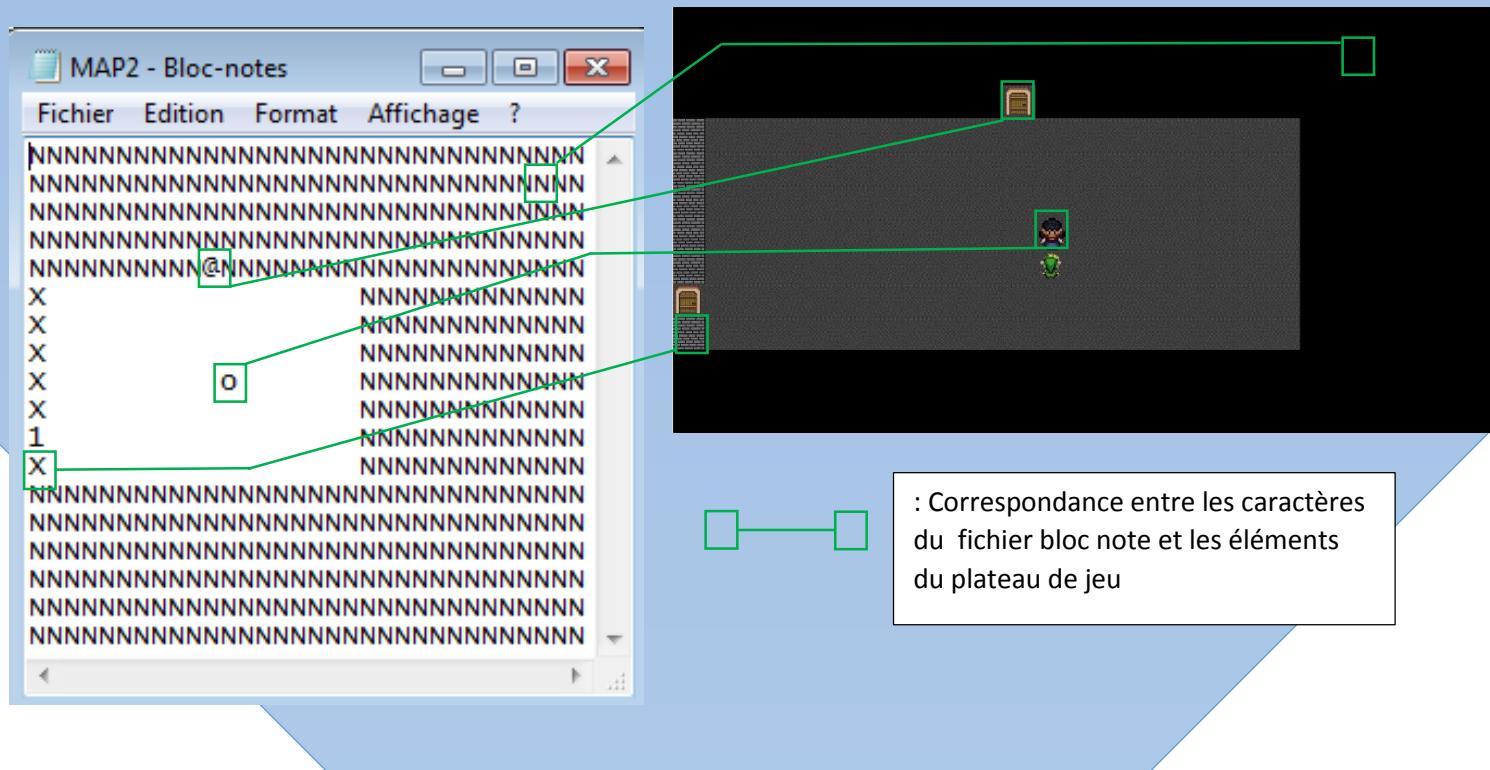
Tâches	Planning	Noms
Scan de la carte par Bloc note	Vacances juin	COPPO Manuel
Boites de dialogues scriptée ou aléatoire	14-23 juillet	UB Tristan
Intelligence Artificiel avec système de combat contre des ennemis	23 Juillet -20 août	COPPO Manuel
Modélisation du Lycée et importation de toutes les images (décor, personnages)	2 aout – 23 aout	UB Tristan
Changement de carte par accès aux portes	20 juin	COPPO Manuel
Gérer les collisions du jeu (monstre, décors.....)	Vacances juin	UB Tristan
Faire les épreuves des bosses en « mini jeux »	23 aout – 21 septembre	COPPO Manuel
Mettre le jeu en plein écran	8 juin	UB Tristan

Une communication entre nous était très facile. En effet, nous nous concertions à chaque problème rencontré pour faciliter l'avancement de notre projet



Au fur et à mesure que l'année avançait, nous nous « spécialisions » dans nos tâches et nous avancions donc de mieux en mieux. Nous apprenions à travailler en groupe et une certaine mécanique se mettait en place. Mon travail consistait à trouver l'idée, à la coder et la faire fonctionner ; ensuite mon camarade Tristan s'occupait d'optimiser et d'améliorer le programme. En effet, j'ai commencé par faire une lecture de carte par bloc note et ensuite Tristan à coder la fonction qui permet de déplacer le personnage dans le plateau créer.

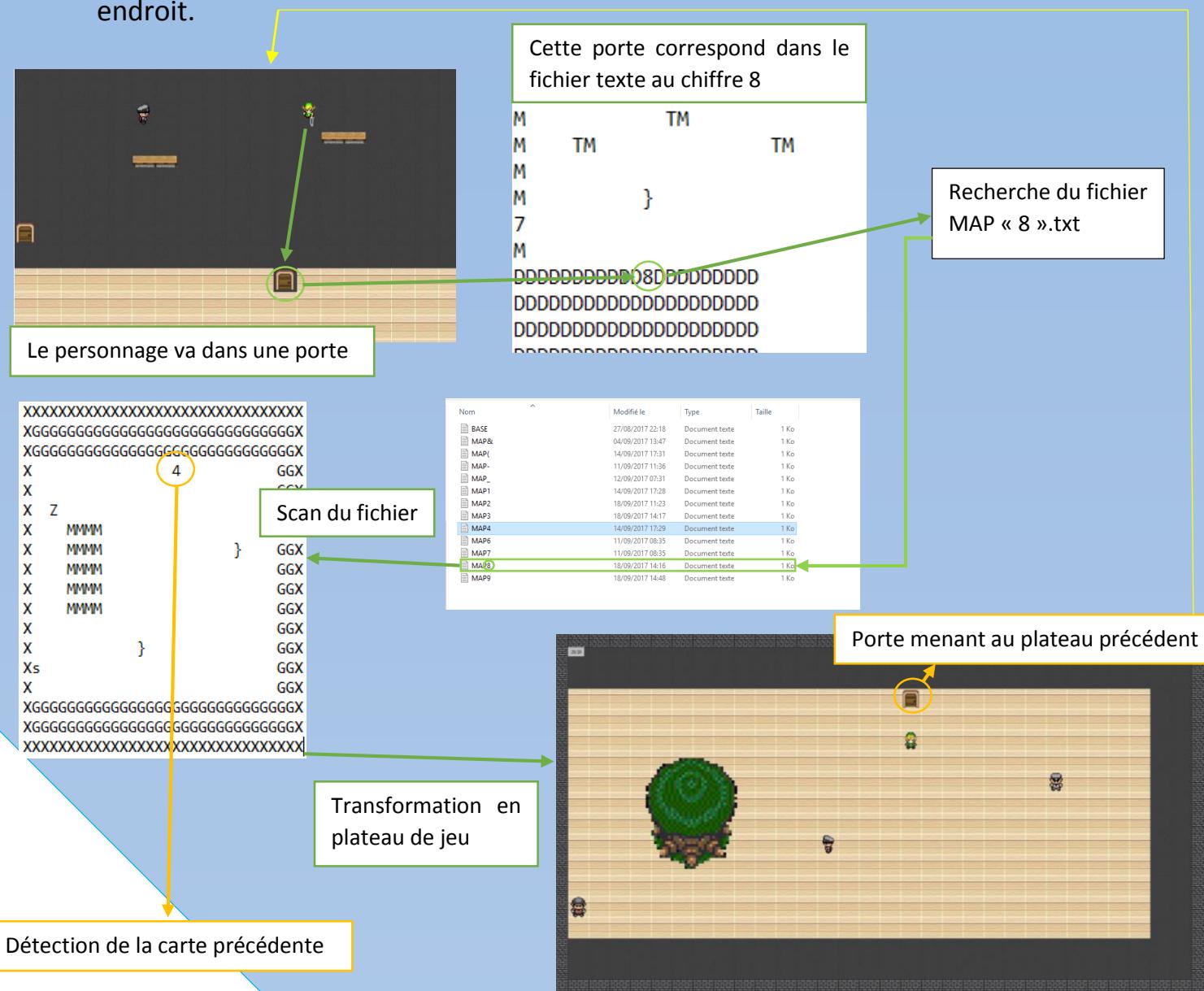
- Premièrement, notre fonction creamap() a pour but de créer le plateau de jeu, elle est « invoquer » au tout début du programme. C'est un scan d'un fichier bloc note préalablement créer avec des lettres particulières ou symboles correspondant à une image dans le plateau. Plus précisément, on lit le fichier bloc note de droite à gauche en mettant chaque caractère lus dans une liste, cette action s'effectue dans la fonction start (). On lit ensuite cette liste en supprimant les retours à la ligne et en « traduisant » ces caractères en images (on crée ainsi un plateau avec une multitude de coordonnées, en soit une grille).



Problèmes rencontrés : Nous nous sommes vite rendu compte qu'il y aurait beaucoup d'image à insérer et donc beaucoup de caractères à mettre. Nous avons donc dû faire une liste pour ne pas se tromper et ne pas s'emmêler les pinceaux...

Réalisation

- Ensuite, nous pouvons sortir d'un plateau de jeux et aller vers un autre en passant par une porte. Cette action s'effectue dans la fonction avance(). En effet, on met préalablement un caractère représentant la porte dans le fichier bloc note. Quand les coordonnées du personnage sont égales aux coordonnées de la porte, on lit quel est le symbole et le programme cherche dans le répertoire donnée la carte correspondante et l'affiche à la place de l'ancienne. Mais cela va plus loin, en effet lorsque la nouvelle carte est chargé le programme détecte le caractère correspondant à la carte précédente et fait apparaître notre personnage à cette endroit.



Problèmes rencontrés : La reconnaissance des fichiers était très difficile il a donc fallu mettre un nom type à chaque carte et a varié uniquement leurs numéros. De plus, le changement de carte nécessite de supprimer l'ancienne sur l'interfaces graphiques et de tout recréer ce qui peux engendrer de nouvelles erreurs.

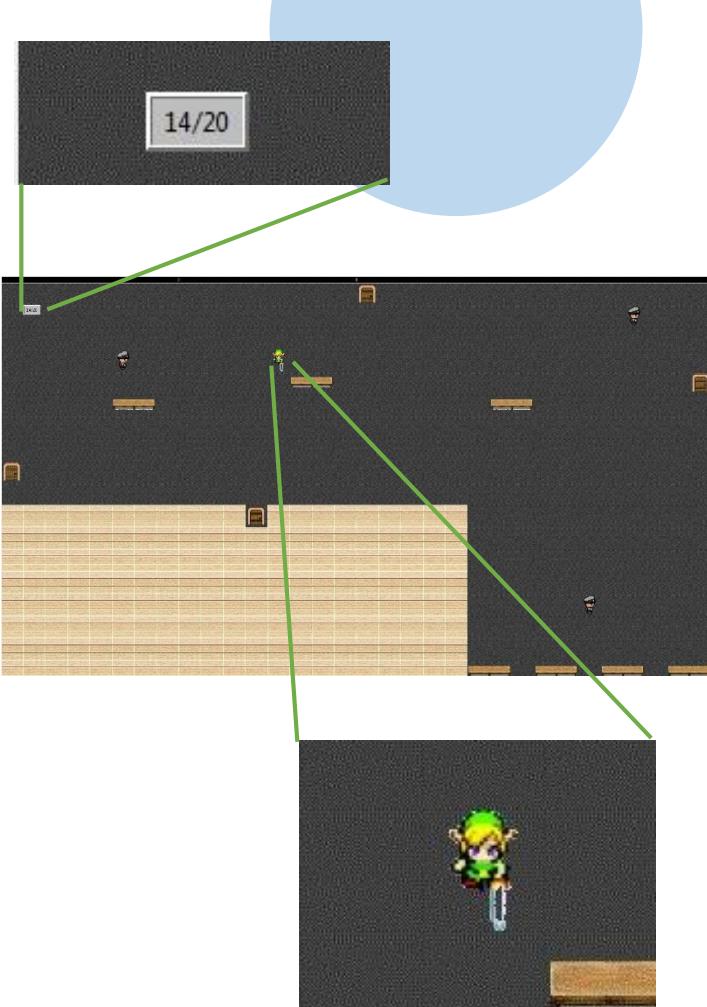
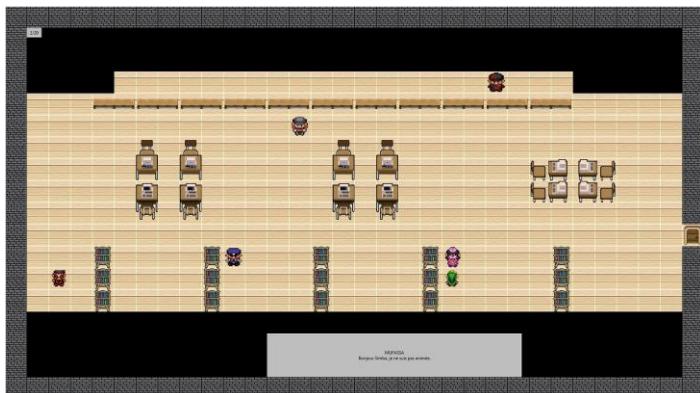
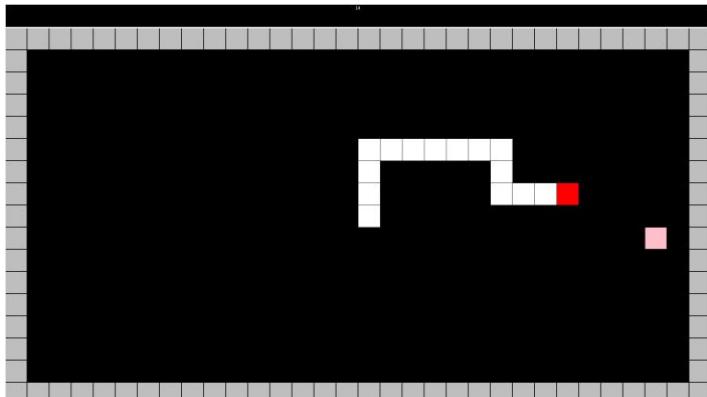
- Beaucoup d'autres fonctions de la production on étaient produites par mon travail mais il serait trop long de tous les définir comme l'intelligence artificiel qui mets en jeux beaucoup d'autre fonctions permettant un déplacement des ennemis intelligent et un combat dynamique(fonctions de déplacements des ennemis, d'attaque du personnage et des ennemis, la gestion de vie etc...) De plus, j'ai coder les épreuves des boss qui sont certains professeurs en expérimentant de nouvelles méthodes de codage avec de nouvelles mécanique comme avec le prof de sport (mini-jeu typé « snake »).

Une très grande partie de mise en commun se fait lors des cours d'ISN que nous avions certes, mais la recherche et l'apprentissage se faisait tout au long de l'année au lycée ou à mon domicile car le domaine du codage est pour moi un sujet très intéressant.



Le travail en groupe était au début difficile mais c'est un apprentissage comme un autre. J'ai appris tout au long de l'année l'entraide et l'organisation d'un groupe. De plus au niveau de la programmation, j'ai appris qu'un problème régler en crée dix mais c'est cela qui fait de cette activité un domaine passionnant.

Intégration du travail



Boss Anglais

Boss Sport

Boss SVT

Intégration des mini-jeux



creamap()

start()

Intégration des plateaux de jeu modulable à partir d'un fichier texte



IA()

ennemie_avance()

tuer()

delete()

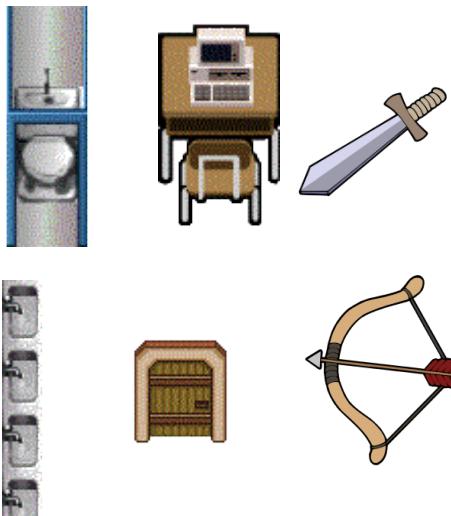
coup()

Intégration de combat contre des ennemis « intelligents » avec gestion de vie



Améliorations possibles

La première amélioration envisageable est de bien sûr améliorer les graphismes avec des dessins plus personnalisé avec un design bien particulier. La participation d'agent extérieur pour le travail de toutes nos images serait conséquente.



Une gestion plus importante de notre personnage avec des objets utilisable ferait variée le gameplay (jouabilité). Rajouter du contenu sera toujours une amélioration possible à notre jeu-vidéo

Enfin une histoire ou une narration rajouterait un fil directeur à notre jeu pour lui donner plus de profondeur. Nous en possédions un mais les cinématiques et les personnages utiles à l'histoire ont été mis à la trappe par manque de temps



BILAN

- Si nous devions diffuser notre production sur internet, nous utiliserions sans hésiter le marché « Steam GreenLight ». En effet, cette plateforme permet au petit programmeur de poster leur programme sur internet et se faire juger par d'autres utilisateurs. Par conséquent, notre jeu serait arbitré par une communauté et aurait une chance de passer sur un marché plus grand (et peut-être payant à partir de 0,99 euro)



Finalement cette année m'aura ouvert la voie de la programmation.

En effet, je voulais confirmer cette année l'envie que j'avais de faire des études en informatique. Le codage nous force à réfléchir différemment pour résoudre chaque problème nouveau pouvant se créer tous les jours.

Par conséquent, je sais maintenant que les langages changent mais moins la réflexion. Ainsi, les notions de variable ou par exemple de fonction « mère » m'ont été apprise.

Finalement, j'ai hâte de continuer cette expérience et éventuellement le jeux-vidéo afin d'en apprendre plus sur le langage utilisé cette année car Python ce n'est pas bidon !