



Nom de votre projet	Echoes of Ruin
Membre de l'équipe n°1 (prénom/nom)	William Godin
Membre de l'équipe n°2 (prénom/nom)	Kilyan Rhein
Membre de l'équipe n°3 (prénom/nom)	Louis Corbeil
Membre de l'équipe n°4 (prénom/nom)	Luc Magnaudeix
Membre de l'équipe n°5 (prénom/nom)	
Niveau d'étude (première ou terminale)	Terminal
Établissement scolaire	Edmond-Perrier
Responsable du dépôt (professeur de NSI)	Mr Madrias

1 / PRÉSENTATION GÉNÉRALE

Nous avons comme problématique de départ de créer un jeu basé sur un labyrinthe aléatoire. Alors, il nous est venu l'idée de concevoir, plutôt qu'un simple labyrinthe de couloirs, un enchevêtrement de pièces disposées aléatoirement sur un plan, reliées par des couloirs eux aussi générés aléatoirement. Se rapprochant d'un jeu de style **Rogue**, l'objectif est d'aller le plus loin possible dans une suite de niveaux tous différents. Et si le joueur vient à mourir, alors il perd tout ce qu'il avait dans son inventaire et doit recommencer de zéro ou avec l'équipement qu'il a stocké dans sa planque.

2 / ORGANISATION DU TRAVAIL

Les tâches ont été réparties comme suit :

- Kilian s'est chargé de créer la boucle permettant le jeu, ainsi que des différentes entités présentes et du placement des pièces les unes par rapport aux autres sur le plan.
- William, lui, s'est occupé des menus et de l'inventaire du jeu.
- Louis a fait le constructeur de salles et a aussi aidé sur de nombreux autres aspects du jeu pour leur conception.
- Luc a réalisé la gestion des salles et des couloirs sous forme d'objets Python, ainsi que la gestion des rayons.

Cependant, nous avons tous travaillé sur un peu tous les points du jeu, que ce soit en aidant à la programmation ou bien au principe de fonctionnement du code lorsque l'un de nous était en difficulté.

Le projet nous a pris au final, selon nous, plus de 150 heures.

Quant au partage et à la communication, nous avons majoritairement utilisé GitHub pour le partage du code et Discord pour les discussions et la répartition des tâches.

3 / ÉTAPES DU PROJET

Tout d'abord, nous avons eu un projet devant être centré sur un labyrinthe généré aléatoirement, cela nous a immédiatement fait penser au jeu vidéo *Rogue*. Il nous est donc venu à l'esprit de faire un Roguelike.

Pour cela, nous devions modifier la façon dont le labyrinthe était généré. Cependant, nous avons décidé de reprendre tout depuis le début. Nous avons généré un enchevêtrement de pièces placées aléatoirement. Ensuite, nous avons créé le joueur et les ennemis. Puis, nous nous sommes occupés des liaisons entre les pièces.

Finalement, nous avons rajouté un inventaire ainsi qu'un menu principal, une planque et des rayons.

4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ

Au moment du dépôt, le projet possède un menu principal d'où on peut accéder à toutes les autres options, un bouton « Quitter » qui permet de fermer la fenêtre facilement, un menu paramètres qui permet de choisir entre le format fenêtré et un format plein écran, un éditeur de pièces qui est accessible depuis le menu principal à partir d'un bouton et qui permet de créer et modifier des pièces qui pourront être générées dans le jeu, et bien sûr, un bouton « Jouer » qui ouvre en premier lieu la planque où le joueur peut récupérer de l'équipement qu'il y aurait stocké dans une partie précédente.

Puis, quand il appuie sur la touche « E », la génération de la carte de jeu se lance avant de s'afficher et de permettre au joueur de jouer.

En jeu, le joueur et les entités ne peuvent pas voir à travers les murs grâce à des rayons fictifs qui permettent de savoir s'il y a ou non un mur entre deux cases souhaitées.

Nous avons tous testé le programme chacun de notre côté au fur et à mesure de l'avancement sur des configurations différentes (OS, IDE, ...). Et nous avons lancé le programme de nombreuses fois afin de minimiser le risque de ne pas voir des bugs, notamment à cause de l'aléatoire présent.

Tout d'abord, il nous a fallu apprendre le fonctionnement de Pygame, avec l'aide de Kilyan qui avait quelques notions de cette bibliothèque. Ensuite, nous nous sommes heurtés au placement des pièces sur un plan, car celles-ci se superposaient. Nous avons eu recours à la récursivité pour en venir à bout.

La génération des couloirs a elle aussi été problématique puisqu' nous devions faire des couloirs sans pouvoir prédire où se trouvaient les pièces. Nous avons donc instauré de nombreuses vérifications empêchant les erreurs de génération (telles que des vérifications de type, la vérification qu'un chemin existe entre toutes les pièces), bien que cela ait pour effet de ralentir grandement la génération, qui peut prendre de très nombreuses secondes.

5 / OUVERTURE

Nous prévoyons d'ajouter des environnements plus riches et variés, d'améliorer l'immersion du joueur avec de meilleurs graphismes, une interface plus épurée, etc. ... Un système plus poussé d'équipement (enchantements, capacités, etc.), des animations plus poussées, une histoire implicite, une personnalisation de l'avatar. Et aussi d'améliorer le temps de génération de la carte.

Si c'était à refaire, nous retravaillerions la gestion du projet, en définissant mieux les tâches à faire et le rôle de chacun.

Ce projet nous a apporté une meilleure connaissance des outils de développement (*VS Code*, GitHub), mais aussi une meilleure maîtrise du langage Python et des connaissances sur la manière de coder.