



Nom de votre projet	X-quisse
Membre de l'équipe n°1	Guillaume Jonvel
Membre de l'équipe n°2	Ian Legrand
Membre de l'équipe N°3	Léane Terry
Niveau d'étude	Terminale
Établissement scolaire	International French School (Singapore) (IFS)
Responsable du dépôt (professeur de NSI)	Tantely Rahobisoa

1 / PRÉSENTATION GÉNÉRALE

- Le but de ce projet était de créer un jeu qui rentre dans le thème de **"l'art"**.
- Jeux vidéo roguelike: combat dans des niveaux faits par génération avec la mort permanente
- Histoire: le joueur incarne un artiste qui doit battre ses brouillons à l'aide d'outils et d'objets utilisés lorsqu'on fait de l'art, tel qu'un pinceau ou une gomme.
- Le jeu utilise Python avec les modules `pygame`, `sys`, `time`, `os` et `random`.
- Le joueur utilise **WASD** pour se déplacer dans une carte de jeu où la disposition des salles est générée aléatoirement. Dans cette carte le joueur peut trouver des ennemis qui lui enlèvent des cœurs et qu'il peut combattre en utilisant **clicque gauche** pour les attaques normales, ou **E** pour les attaques spéciales. Une fois morts, les ennemis produisent de l'encre, que le joueur absorbe et peut dépenser en tenant la touche **F** pour se soigner. On peut aussi trouver des salles trésor, qui contiennent un item aléatoire améliorant les caractéristiques du joueur (vitesse, points de vie). Pour pouvoir progresser au prochain niveau, le joueur doit tuer suffisamment d'ennemis pour débloquer une porte.

2 / ORGANISATION DU TRAVAIL

Guillaume	Génération de carte, combat, ennemis, débogage, effet visuels et GUI, montage vidéo	A déjà travaillé sur différents projets de jeux, alors il pouvait réunir et implémenter les différents éléments du code ensemble pour faire un jeu fonctionnel
Ian	Classe affichage, menu principal, classe sprite	Avait déjà un peu d'expérience avec la librairie Pygame, donc il a été logique de le laisser prendre charge de tout ce qui utilisait cette librairie
Léane	art, classe projectile, classe item, classe son, organisation et suivie du projet	Avait de l'expérience dans la création visuelle et l'animation, donc tout ce qui était direction artistique était sous sa responsabilité

- Le game design et le travail de réflexion sur les mécaniques du jeu a été principalement fait par Guillaume et Léane. L'implémentation des classes entre elles et le débogage ont été faits principalement par Ian et Guillaume.
- Le brainstorming du projet a été fait sur **Milanote**, le suivi et la répartition progressive des tâches sur un fichier **Google Sheets**, et tous les fichiers (code, art, sons) ont été mis dans un **Shared Drive** disponible aux membres du groupe. Pour communiquer, nous avons utilisé un groupechats **WhatsApp** et **Discord**.
- Le projet a été commencé en décembre, avec 2 heures de travail par semaine par personne.

3 / ÉTAPES DU PROJET

- Les sprites ont été faits progressivement, quand ils devenaient nécessaires pour faciliter la visualisation pour le débogage du code.

Chronogramme du projet :

Taches	Séance 1	Séance 2	Séance 3	Séance 4	Séance 5	Séance 6	Séance 7	Séance 8	Séance 9	Séance 10	Séance 11	Séance 12
classe affichage												
génération de salle												
classe sprite												
class projectile												
collision												
Ecran de menu												
Layers												
class item												
ennemis												
classe musique												
classe personnage												

- phase de brainstorming: créer un jeu qui rentre dans le thème de "l'art"
 - Direction artistique: trouver différentes manières d'incorporer des objets utilisés dans la création artistique et d'avoir un style qui évoque des dessins 'amateurs' ou 'brouillons'.
- Créer le "moteur de jeu" pour avoir les fondations des autres mécaniques:
 - Génération d'une carte de jeu avec une disposition aléatoire
 - Cette étape a pris beaucoup de temps à cause de sa complexité. En effet, il a fallu créer deux classes parfaitement synchronisées entre elles ainsi qu'un algorithme de génération fiable. Néanmoins le problème le plus chronophage a été un simple bug qui inversait le haut et le bas.
 - Classe pour l'affichage de sprites sur l'écran
- Collision:
 - Méthode pour détecter la collision: puisque tous les sprites ont été initialisés par la même classe, je peux donner un "mask" a tous les sprites dans le jeu. En utilisant ce masque, il est possible de calculer une collision. Le problème c'est que les masques ne se déplacent pas. Donc il faut déplacer le point de collision par la localisation du sprite, me donnant exactement si et où il y a une collision.
- Implémentation du joueur:
 - Un objet de la classe `Personnage` ayant plein de statistique modifiable
 - Cette étape a également consommé beaucoup de temps à cause de l'implémentation des animations.
- Implémentation des items, des coeurs et de l'encre

- Création d'une classe `Item` qui modifie les caractéristiques du joueur (nombre de cœurs, vitesse de mouvement, etc) et qui affiche une description de ce que l'item fait.
- Création d'une classe `Coeur`
- Création d'une classe `Blob` (goutte d'encre)
- effets visuels: création d'une classe particule
- Implémentation des ennemis
 - Programmation des comportements des différents ennemis et leurs modes d'attaques
 - Implémentation de la classe `Projectile`
 - Placement des ennemis selon les types de salles
- implémentation de la musique et des effets sonores pas la classe `Musique`
- Création d'un menu principal pour lancer le jeu, régler les paramètres (volume), voir les crédits et quitter le jeu

4 / FONCTIONNEMENT ET OPÉRATIONNALITÉ

- À présent, le jeu consiste d'un seul thème pour les niveaux, avec 4 types d'armes possibles, 6 items que le joueur peut rencontrer, 6 types d'ennemis différents et un ennemi boss.
 - Il y a 6 niveaux, avec un boss tous les 2 niveaux. Pour progresser au prochain niveau, il faut tuer un certain nombre d'ennemis et battre le boss s'il y en a un. Plus on progresse dans le jeu, plus la carte de jeu est grande. En début de partie, il y a un mini-niveau "tutoriel", qui explique les mécaniques du jeu.
- Les bugs ont été trouvés en faisant des prints pour vérifier si les comportements étaient corrects, en envoyant un excès d'inputs pour détecter des comportements inattendus, et en faisant du playtesting.
- Un problème qu'on a rencontré était de choisir quels sprites afficher en-dessous d'autres sprites. La solution était simple: créer une liste ordonnée des sprites dans l'ordre qu'ils seront affichés..
- Un problème rencontré pendant la création de la classe `carte` fut de coordonner la position des objets et ennemis à la salle dans lequel ils sont. Pour répondre à ce problème de nombreuses importations à travers les paramètres de la fonction ont été fait, ce qui alourdit de manière importante le code .

5 / OUVERTURE

Compétences développées:

- Travail en groupe: Avec un projet sur une durée aussi longue et d'une envergure assez ambitieuse par rapport à notre expérience, on a pu apprendre et s'entraîner à nous organiser, à déléguer des tâches, et à communiquer entre nous pour savoir ce qui devait être fait et quand.

Analyse critique:

- Nous avons manqué de temps pour intégrer certaines mécaniques que nous avions eu envie d'inclure, par exemple d'avoir deux armes disponibles au joueur à tout moment et qu'il peut alterner au choix.
- Au niveau de la gestion du temps on pourrait être. Si c'était à refaire, il pourrait être mieux de se fixer des dates limites internes pour la complétion des différentes parties du jeu, au lieu de s'adapter à ce qui a été déjà complété et attendre.
- On aurait dû, dès le début du projet, rendre l'adaptabilité de la taille de la fenêtre de jeu et des sprites possible. On a réalisé trop tardivement qu'il fallait prendre en compte la taille de l'écran du joueur et il aurait été trop chronophage de complètement recréer la classe `Affichage`. Il aurait été aussi préférable de créer le menu principal avant le jeu lui-même pour que la modification des paramètres soit présente dès le début.

Améliorations possible:

- Il pourrait être intéressant d'incorporer des cinématiques pour que l'histoire et la thématique du jeu soient plus évidentes au joueur.
- Pour faire que la progression du jeu soit plus intéressante, il pourrait avoir différents thèmes de niveaux, basés sur d'autres formes d'art pour rester dans l'idée d'X-quisse (comme un thème inspiré des darkrooms utilisés pour la photographie argentique).
- On pourrait inclure plus d'item et d'armes (argile qui agit comme une bombe, arme avec un projectile rebondissant, item qui donnerait une invincibilité temporaire). Idéalement, il y aurait aussi un système où le joueur peut choisir d'échanger ou non les armes dans son arsenal avec une salle de dépôt d'armes par niveau.