



Ce document est l'un des livrables à fournir lors du dépôt de votre projet : 4 pages maximum (hors documentation).

Pour accéder à la liste complète des éléments à fournir, consultez la page [Préparer votre participation](#).

Vous avez des questions sur le concours ? Vous souhaitez des informations complémentaires pour déposer un projet ? Contactez-nous à info@trophees-nsi.fr.

NOM DU PROJET : Nobeliane

> PRÉSENTATION GÉNÉRALE :

• *Idee et objectifs*

Le projet est un jeu de rôle façon vieux Zelda. Le joueur incarne un personnage et exécute des actions pour suivre un scénario prévu dans le jeu. L'objectif est de développer un projet très ambitieux sur du long terme. La création d'un jeu donne la possibilité de toujours améliorer son projet tant dans l'univers graphique que dans les fonctionnalités propres au jeu.

• *Origines et intérêts du projet*

Cette idée nous est venue d'un projet bien plus simple effectué en début d'année sur le développement d'un programme utilisant plusieurs classes en Python. Ce premier projet nous avait montré concrètement comment manipuler la programmation orientée objet, la base de notre projet. En parallèle nous jouons à un jeu de rôle réel (le Donjon de Naheulbeuk) qui nous inspire pour l'univers de Nobeliane.

> ORGANISATION DU TRAVAIL :

• *Présentation de l'équipe (prénom de chaque membre et rôle dans le projet)*

Mathéo est chargé de la création des graphismes du jeu : design de la carte, implémentation de différents monstres et du joueur. Il a aussi co-développé le cœur du jeu, notamment les premières versions de la gestion du joueur.

Renan s'occupe du développement de toute l'interface et du menu de démarrage. Il a aussi créé l'ambiance sonore du jeu.

Samuel porte le rôle de coordinateur du projet. Il regroupe les codes et les optimise. Il a aussi développé le cœur du jeu au niveau des animations et de la gestion des cartes.

• *Organisation du travail (répartition par petits groupes, fréquence de réunions, travail en dehors de l'établissement scolaire, outils/logiciels utilisés pour la communication et le partage du code, etc.)*

Nous avons chacun développé des fonctionnalités séparément comme l'interface pour Renan mais certaines parties complexes ont été développées à plusieurs autour du même ordinateur. Chaque semaine, nous réunissons les codes pour repartir sur la même base.

Après chaque cours où nous avançons sur le projet, nous faisons un débrief sur ce que nous venons de faire.

Il nous arrive fréquemment de travailler chacun chez soi pour avancer plus vite ou par exemple Renan pour créer lui-même la musique à partir de ses logiciels.

Nous avons créé un serveur discord nous permettant d'y déposer des fichiers et des sites internet utiles pour notre projet. La majorité des brainstormings se sont faits pendant les pauses au lycée. Discord a surtout permis de valider ou non le travail fait chacun chez soi. La mise en place d'un git sera bientôt faite pour faciliter notre gestion du code.

LES ÉTAPES DU PROJET :

- *Présenter les différentes étapes du projet (de l'idée jusqu'à la finalisation du projet)*

Au tout début, nous avons réfléchi à l'histoire et aux paysages que nous voudrions créer dans notre jeu. Nous avons tout réuni d'une manière « barbare » toutes nos idées sur l'univers de base que nous voudrions créer dans le document 'brainstorming_lieux.odt'.

Ensuite, nous nous sommes inspirés d'un tutoriel pour créer la base d'un petit jeu en Python avec la bibliothèque Pygame. Aujourd'hui, il ne reste plus grand-chose de ce code mais il nous a permis de concrétiser le fonctionnement du jeu. Cela nous a fait gagner beaucoup de temps.

Les graphismes du jeu ont été faits sur le logiciel Tiled à cette étape là. Ce logiciel consiste à disposer des Tuiles (textures en 2D au format PNG) et à créer des zones réutilisables à l'intérieur du code en Python.

> FONCTIONNEMENT ET OPÉRATIONNALITÉ :

- *Avancement du projet (ce qui est terminé, en cours de réalisation, reste à faire)*

La gestion des cartes et des animations est terminée. Le joueur peut se déplacer et se téléporter d'une zone à une autre.

L'interface est très bien avancée mais il manque encore la gestion d'un inventaire.

Nous voyons beaucoup de choses qui sont à faire mais principalement il faut développer une interaction entre le joueur et d'autres personnages. Pour l'instant, l'équipe s'est concentrée sur la base du jeu. Nous n'avons pas encore intégré le concept d'interaction en dehors des changements de zones (d'une carte à une autre).

- *Approches mises en œuvre pour vérifier l'absence de bugs et s'assurer de la facilité d'utilisation du projet*

Un document README.txt a été ajouté à la documentation pour expliquer comment installer les bibliothèques utilisées dans le projet. Pour les bugs, nous avons testé au maximum les actions que peut faire le joueur pour découvrir d'éventuels bugs.

- *Difficultés rencontrées et solutions apportées*

Le code de base qui correspondait à un tutoriel ne nous permettait pas d'avoir plusieurs cartes. Tout le système a dû être repensé ce qui a pris beaucoup de temps.

La gestion des personnages a aussi été difficile car nous n'avons pas assez abordé la notion d'héritage dans les cours de programmation orientée objet. Nous avons réussi à implémenter cette notion en cherchant plusieurs exemples sur internet.

Au niveau de l'interface, Renan a dû penser tout le système pour qu'il s'adapte à la résolution de l'écran.

Nous avons créé un exécutable pour simplifier le démarrage, mais windows le reconnaissait comme un virus et supprimait instantanément le fichier. Nous n'avons pas encore résolu ce problème.

> OUVERTURE :

- *Idées d'améliorations (nouvelles fonctionnalités)*

La prochaine étape serait clairement la gestion des interactions avec l'environnement, soit les combats et par exemple pouvoir actionner des leviers ou autre.

Pour avoir un jeu d'avantage complet, il faudrait enfin créer un inventaire avec des objets récupérés dans des coffres et sur des monstres. Pour gérer les données de l'inventaire il faudrait une base de données, ça permettrait d'intégrer le SQL au projet pour plus de complexité.

Enfin, nous ferons une interface de paramètres pour activer ou non les sons. La touche Échap dans le jeu permet de le quitter avec une confirmation.

Nous envisageons de créer une ambiance sonore pour une meilleure immersion comme des bruits de nature ou encore des changements de musiques en fonction des cartes.

- *Stratégie de diffusion pour toucher un large public (faites preuve d'originalité !)*

Nous souhaitons créer un site Web que nous hébergerons nous-mêmes pour donner accès à notre jeu une fois que toutes les étapes décrites précédemment seront réalisées. L'objectif serait d'attirer l'attention de personnes à la fois nostalgiques des années 90 grâce aux musiques 8-bit et aux graphismes pixelisés. De plus, nous pourrions mettre en avant notre autonomie dans ce projet en tant que lycéens pour présenter notre parcours à d'autres jeunes en quête d'un projet ambitieux. Nous mettrons donc l'accent sur ces deux aspects de notre projet sur le site Web.

- *Analyse critique du résultat (si c'était à refaire, que changeriez-vous dans votre organisation, les fonctionnalités du projet et les choix techniques ?)*

Nous avons perdu beaucoup de temps à adapter le code original que nous avons trouvé pour la base de notre jeu. Maintenant que nous connaissons les méthodes de travail qui nous correspondent et la façon dont nous appréhendons le jeu, le refaire serait bien plus rapide. Au niveau de l'organisation nous passerions moins de temps à coder à plusieurs sur le même ordinateur pour avancer plus efficacement.

Une des premières choses mises en place, si c'était à refaire, ce serait un git. Nous aurions dû en mettre un en place dès le début pour faciliter toute l'organisation et la sécurité de nos données. Nous effectuons actuellement des double sauvegardes mais ça ne garantit évidemment pas de ne perdre aucune donnée par erreur.

DOCUMENTATION

- *Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)*

Nobeliane utilise les bibliothèques pygame, pytmx et pyscroll pour fonctionner.

Pour pouvoir démarrer le fichier main.py dans un éditeur, il faut entrer les trois commandes suivantes dans une console python :

- pip install pygame
- pip install pytmx
- pip install pyscroll

Une fois ces bibliothèques installées, le fichier main.py devrait être exécutable et lancer le jeu.

Le jeu est lancé en plein écran pour une bonne gestion des proportions. Cependant pour le fermer il faut exécuter le raccourci **Alt + F4** ou le fermer pour un autre moyen.

- *Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)*

Au démarrage du main.py, la boucle run() de la classe Game est enclenchée. Elle constitue la base du jeu qui va tourner en boucle pour exécuter tout le programme. Tous les éléments qui auront une animation ou une quelconque utilité dépendent de la classe Entity. Cela nous permet d'ajouter des éléments facilement et rapidement sans multiplier le code.

Nous n'utilisons que le langage Python pour le moment. La bibliothèque pygame nous permet de démarrer la fenêtre de jeu et d'actionner différentes fonctionnalités à l'intérieur. La bibliothèque pytmx sert à faire le lien entre la carte du logiciel Tiled et notre code en python grâce à des accesseurs.

- *Illustrations, captures d'écran, etc*



