



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 : SpaceShooter

> PRÉSENTATION GÉNÉRALE :

- **PRÉSENTATION**

SpaceShooter est un jeu de survie en vue 2D, le joueur doit y vaincre des monstres tous plus nombreux au fil des vagues. Pour cela deux modes lui sont mis à disposition, un mode Campagne où il devra traverser plusieurs mondes, il y rencontrera des boss et des monstres en tous genres. Ainsi qu'un mode infini dans lequel, il fera face à des vagues toujours plus difficiles, dans le but d'atteindre des vagues toujours plus imposantes.

Lors de ces parties le joueur accumule des pièces d'or lui permettant d'acquérir des armes toujours plus fortes, mais aussi de nouveaux personnages. Ces personnages pourront eux aussi être améliorés à l'aide de ces pièces. Elle permet aussi l'achat d'équipements qui vise à faciliter la progression du joueur, comme un bouclier, des grenades, ou des potions.

- **IDÉE ET OBJECTIFS**

Le projet a été débuté dans un but de progression en code, plus particulièrement en python, et plus particulièrement en programmation orientée objet. Le projet devait donc contenir de nombreuses classes qui interagissent en elles-mêmes.

Un jeu était parfaitement adapté à cela, de plus le jeu rajoute une difficulté, le nombre d'interactions qui est fortement élevé. L'idée était aussi de trouver un projet qui nous donne envie, le développement d'un jeu répondait donc parfaitement à nos attentes.

- **INTÉRÊTS ET ORIGINE DU PROJET**

Comme dit ci-dessus l'intérêt du projet était de progresser en python, découvrir de nouvelles choses comme la gestion des fichiers, la gestion des évènements, la sauvegarde de données dans d'autres fichiers. Tout en prenant du plaisir à développer.

Dans une recherche de type de jeux, nous avons fait le choix de rendre un ancien petit projet jeu texte, en graphique. Tout était donc prêt il ne restait plus qu'à commencer.

> ORGANISATION DU TRAVAIL :

- **PRÉSENTATION DE L'ÉQUIPE**

Heilmann Jonathan : Programmation du projet, crée les différentes fonctionnalités gère le bon fonctionnement du jeu, intègre les éléments apportés par le reste de l'équipe.

Schmitt Nicolas : Graphismes et équilibrage du projet, crée et recherche les décors du jeu, comme les personnages, les armes, les équipements... Equilibre le jeu, en définissant le nombre de dégâts du joueur, la vie du monstre, le nombre de monstres par vague...

- **ORGANISATION DU TRAVAIL**

L'équipe travaille sur le projet quotidiennement, la communication se fait en partie au lycée. Mais aussi à travers certains logiciels, tels que :

- Discord : Pour communiquer rapidement et envoyer des petits fichiers
- Trello : Pour répartir les tâches. Noter les idées et voir l'avancée du projet. Notre Trello : <https://trello.com/b/uS65OQof/id%C3%A9e-a-faire>
- Git et GitHub : Partager le code, suivre l'avancée, GitHub nous a permis de garder une structure commune pour tous, et de rapidement partager ou récupérer l'avancement. GitHub du Projet : <https://github.com/Miniquoinquin/SpaceShooter>
- Photo pea : Création des modèles graphiques

> LES ÉTAPES DU PROJET :

- **PLANIFICATION**

Structuration et recherche des idées, création d'une structure des fichiers/ documents. Tests de premières interactions entre objet python.

- **DÉVELOPPEMENT DU CŒUR DU JEU**

Création de la répétition du fond, des déplacements et de l'animation du personnage. Ajout des monstres, création d'effets avec les sorciers, ajout des projectiles avec l'arme du joueur et les monstres qui tirent.

- **AJOUT DES MENUS**

Implémentation de nouvelles fonctionnalités telles que de nouveaux mondes, de nouveaux personnages, de nouveaux modes. Création de menus pour simplifier et améliorer l'expérience de jeu.

- **SAUVEGARDE DE L'AVANCEMENT**

Sauvegarde des informations (pièces d'or, personnages, son, avancée dans la campagne...) dans différents fichiers texte et csv.

- **FONCTIONNALITÉ TIERCE**

La possibilité d'améliorer son personnage, ses armes, d'acquérir et d'améliorer des équipements (bouclier, grenade, potion). L'ajout d'une carte pour rendre le mode campagne, ajout de sons et de musiques pour améliorer l'immersion du joueur.

Les étapes détaillées de l'avancement du projet son disponible sur GitHub !

> **FONCTIONNEMENT ET OPÉRATIONNALITÉ :**

- **AVANCEMENT ET APPROCHES DU PROJET**

Le jeu est jouable, on peut y faire beaucoup de choses, on y retrouve de nombreux aspects qui permettent d'améliorer la diversité du jeu.

Le jeu est actuellement en phase de test, par plusieurs personnes l'objectif est de détecter et corriger un maximum de bugs.

Le jeu est à première vue terminé, il ne reste rien d'élémentaire à ajouter. L'ajout de nouvelles fonctionnalités que nous verrons plus tard pourrait être envisagé.

- **DIFFICULTÉS RENCONTRÉES ET SOLUTION APPORTÉES**

Faire marcher le Joueur :

On n'y pense pas forcément tout de suite, mais simplement déplacer le personnage ne suffit pas. Il faut aussi lui donner un effet de marche, pour cela nous nous sommes inspirés d'autres jeux vidéo, et avons utilisé plusieurs images qui affichées de manière successive font marcher le joueur.

Saut et trajectoire de lancer :

La modélisation d'un saut, est particulièrement complexe, en effet la vitesse Y du joueur doit varier en fonction de l'avancée du saut mais aussi de la gravité. Une simple formule de mathématique avait permis de résoudre le problème du saut. Mais celle-ci ne pouvait pas être utilisée pour les lancers d'objets qui devaient aussi dépendre de X, nous avons donc modélisé le lancer par un ovale et utilisé le cercle trigonométrique.

Sauvegarder des informations à long terme :

Sauvegarder des informations dans python c'est facile grâce aux variables, mais comment faire pour ne pas tout perdre lorsqu'on stoppe le programme. Pour régler ce problème nous avons choisi d'utiliser des fichiers csv, qui peuvent stocker une grande quantité de données séparées par un caractère défini comme le point-virgule par exemple et forme ainsi un tableau.

Création des animations :

Crée une animation avec EZ, c'est plus ou moins facile tant qu'on utilise uniquement des formes directement traçables avec une fonction EZ. Mais quand l'animation demande des formes plus complexes telles que nos coins rouges pour illustrer un coup reçu par le joueur. L'utilisation d'outils

mathématiques devient alors nécessaire, dans le cas de notre animation de dégâts nous avons utilisé le cercle trigonométrique pour calculer l'ensemble des points présents sur notre ovale dont la taille varie à chaque instant.

> OUVERTURE :

- **IDÉES D'AMÉLIORATION**

- Créer des bordures variées en fonction des mondes :**

- Les bordures sont représentées par des caisses empilées que vous soyez sur Terre, sur Mars ou sur Forestia. L'idée serait de créer une bordure propre à chaque monde, afin de la rendre plus esthétique.

- Créer des effets dans les mondes :**

- Vous avez probablement remarqué que la principale différence entre les mondes hormis le décor, ce sont les monstres. L'idée serait d'accentuer ces différences en créant des particularités à chaque monde, cela peut être la gravité, le vent (pousse aléatoirement le joueur dans une direction), des zones empoisonnées ...

Plus d'idées d'amélioration sont disponible sur notre Trello.

- **STRATÉGIE DE DIFFUSION**

- Nous n'avions pas prévu de rendre notre jeu célèbre, étant donné que le but était de progresser en python, ainsi qu'en logiciel dessin.

- Si nous avons voulu diffuser notre jeu, nous aurions probablement organisé un concours sur les réseaux sociaux, où le joueur qui atteindrait la plus haute vague emporterait un Cash Prize. Mais pour cela nous devrions crypter, ou délocaliser les données pour éviter la modification de certains éléments comme le nombre de pièces d'or.

- **ANALYSE CRITIQUE DU RÉSULTAT**

- Nous trouvons le résultat convaincant, mais si c'était à refaire :

- Heilmann Jonathan : J'organiserais mieux les classes fondamentales des menus, ainsi que les déplacements du joueur pour permettre un mode multi-joueur local.

- Schmitt Nicolas : J'essaierais de créer plus de modèles moi-même, plutôt que de partir de modèles libres sur google.

DOCUMENTATION

- *Spécifications fonctionnelles (guide d'utilisation, déroulé des étapes d'exécution, description des fonctionnalités et des paramètres)*
- *Spécifications techniques (architecture, langages et bibliothèques utilisés, matériel, choix techniques, format de stockage des données, etc)*
- *Illustrations, captures d'écran, etc*

La documentation se trouve dans **doc.html** ou **doc.md**