

Nom : BARRY

Prenom : Oumar Sadjo

Programme : Electroniques des systèmes embarqués et communicants

Cours : Programmation Orientée objet en C++

Projet : ShootEmUp 2D

Année Scolaire : 2023 – 2024

Ecole : Polytech Nice Sophia

## Introduction

Le projet "ShootEmUp 2D" est un jeu vidéo développé en langage C++ en utilisant la bibliothèque SFML (Simple and Fast Multimedia Library). Ce jeu s'inspire du genre classique du shoot'em up, où le joueur pilote un vaisseau spatial. Le joueur est chargé de détruire un grand nombre d'ennemis à l'aide de projectiles. À mesure que le joueur progresse à travers les niveaux, le défi augmente, exigeant des réflexes rapides, une grande précision et une stratégie astucieuse pour éviter les projectiles ennemis et survivre.

La complexité de jeu, fait appelle à plusieurs concepts de la programmation orientée objet (POO), qui m'ont motivé d'explorer à travers ce projet.

L'objectif de ce rapport est de présenter en détail le développement du jeu, en mettant en avant les concepts utilisés, le fonctionnement du jeu, les outils et ressources employés, ainsi que les points à améliorer pour de futures versions.

## Le mode fonctionnement

Le jeu "ShootEmUp 2D" suit un modèle de fonctionnement classique des jeux de type shoot'em up, dont les étapes principales du fonctionnement sont :

Le jeu démarre en initialisant la fenêtre, le joueur, les ennemis et les ressources graphiques. Ensuite, il entre dans une boucle principale où il écoute les événements utilisateur, met à jour l'état du jeu et affiche les éléments à l'écran. Pendant cette boucle, le jeu détecte les collisions entre les projectiles et les ennemis, gère la fin de partie selon les conditions définies et attend les interactions de l'utilisateur pour relancer ou quitter le jeu.

# Les concepts de la POO utilisés

Dans ce projet "ShootEmUp 2D", plusieurs concepts de la programmation orientée objet (POO) sont utilisés pour organiser le code de manière modulaire et réutilisable :

**Classes et Objets** : Chaque entité du jeu est représentée par une classe distincte qui sont : *Game*, *Player*, *Enemy*. Ces classes encapsulent à la fois les données et les fonctionnalités associées à chaque entité.

**Encapsulation** : Les données et les fonctionnalités sont encapsulées dans chaque classe, ce qui permet de limiter l'accès aux données et de garantir leur intégrité. Par exemple, les attributs et méthodes de chaque class sont encapsulés pour gérer la logique du jeu, propre à chaque jeu.

**Polymorphisme** : Le polymorphisme est utilisé pour traiter des objets de différentes classes de manière uniforme. Par exemple, la méthode *update()* est définie dans la classe *Game* et est appelée de manière polymorphique pour mettre à jour le joueur, les ennemis et les projectiles.

**Composition** : La composition est utilisée pour assembler les différents éléments du jeu. Par exemple, la classe *Game* utilise la composition pour inclure des objets tels que le joueur, les ennemis et les projectiles, ce qui permet une gestion modulaire et flexible du jeu.

## Utilisation du jeu

**Pour cloner et compiler ce projet :**

```
git clone https://github.com/oumarsadjo/ShootEmUp
```

```
cd ShootEmUp
```

```
mkdir build && cd build
```

```
cmake ..
```

```
make
```

```
./ShootEm
```

- ❖ La touche de la direction de gauche '**<**' pour diriger le joueur vers la gauche ;
- ❖ La touche de la direction de droite '**>**' pour diriger le joueur vers la droite ;
- ❖ La touche '**S**' pour tirer les balles du joueur ;
- ❖ La touche '**R**' pour relancer le jeu ;
- ❖ La touche '**X**' pour quitter le jeu ;

# Résultat du jeu

## Outils utilisés

Dans le projet "ShootEmUp 2D", les outils suivants ont été utilisés pour son développement :

**Environnement Linux** : L'environnement de développement utilisé est basé sur Linux, offrant une plate-forme stable et robuste pour le développement logiciel.

**Langage C++** : Le projet est développé en utilisant le langage de programmation C++, connu pour sa performance, sa flexibilité et son support avancé de la programmation orientée objet.

**Compilation via un Makefile (Makefile.txt)** : Le projet est compilé à l'aide d'un Makefile (Makefile.txt), un script de compilation utilisé pour automatiser le processus de construction du projet.

**Bibliothèque SFML (Simple and Fast Multimedia Library)** : La bibliothèque SFML est utilisée pour la gestion des graphismes, des entrées utilisateur et du son dans le projet. Elle offre une interface conviviale et puissante pour le développement de jeux et d'applications multimédias.

**Gestion de version avec GitHub** : GitHub est utilisé comme plateforme de gestion de versions pour héberger le code source du projet, pour faciliter le suivi et les modifications apportées au code.

## Amélioration du projet

Pour améliorer la performance et la professionnalité du jeu "ShootEmUp 2D":

Dans la première itération du jeu, l'accent est mis sur le gameplay unilatéral où le joueur agit seul en tirant et en restant statique tandis que les ennemis progressent. Pour une seconde version, j'envisage d'incorporer une dynamique différente en permettant aux ennemis de riposter par des tirs tout en introduisant des déplacements pour le joueur, offrant ainsi une expérience de jeu plus immersive et interactive.

## Conclusion

Le cours de Programmation Orientée Objet que vous avez dispensé s'est avéré être une expérience enrichissante. Il m'a permis de plonger dans les profondeurs du langage C++ et d'explorer ses nombreuses bibliothèques. Grâce à ce projet, j'ai eu l'opportunité de mettre en pratique les concepts fondamentaux de la POO que nous avons étudiés, notamment la création de classes, le polymorphisme, l'héritage et l'encapsulation, entre autres. L'alliance de la puissante bibliothèque SFML avec ces principes de la POO m'a offert une approche exceptionnelle pour aborder ce sujet complexe de manière complète et aboutie.