

Ahmed Soussi, Enzo Greneche

# Rapport : Shooting Odissey

---

## Sommaire

### Introduction

- **Objectifs du projet**

### Conception du jeu

- **Présentation du jeu**
- **Conception de l'interface graphique**

### Implémentation

- **Organisation du code source**
- **Développement des fonctionnalités principales**
- **Gestion des interactions utilisateur**
- **Gestion des événements graphiques**

### Documentation technique

- **Guide d'installation**
- **Guide d'utilisation**
- **Documentation du code source**

### Conclusion

- **Bilan du projet**
- **Perspectives d'amélioration**
- **Remerciements**

---

## Introduction

### Objectifs du projet

L'objectif de ce projet est de **développer une application graphique d'un jeu d'avion de type Shoot'em up (descendez-les tous)**

en temps réel et donc en gérant le taux de rafraîchissement de l'écran **F.P.S.** (Frames per second).

---

## Conception du jeu

### Présentation du jeu

Le jeu est en **2d** et se déroule dans **l'espace**. Le joueur contrôle un vaisseau spatial qui peut se déplacer dans **toutes les directions**.

Il doit éviter ennemis et leurs tirs tout en les détruisant.

Le jeu se termine lorsque le joueur n'a plus de vie ou lorsque le joueur a détruit tous les ennemis.

### Conception de l'interface graphique

L'interface graphique est composée de plusieurs éléments :

- Un **fond d'écran** qui représente l'espace et qui défile de bas en haut.
  - Un **vaisseau spatial** qui peut se déplacer dans toutes les directions.
  - Des **ennemis** qui apparaissent selon un fichier en haut de l'écran et qui se déplacent de haut en bas.
  - Des **tirs** qui sont lancés par le vaisseau et les ennemis.
- 

## Implémentation

### Organisation du code source

Pour ce projet, nous avons utilisé des fichiers de type '.h' pour déclarer et documenter des fonctions et des structures dans un dossier include

et des fichiers de type '.c' pour implémenter les fonctions et les structures dans un dossier src.

On a également utilisé un dossier data pour stocker les images et les sons utilisés dans le jeu.

### Développement des fonctionnalités principales

- le création du vaisseau
  - la création des ennemis
  - les mouvements du vaisseau
  - les mouvements des ennemis
  - création des tirs
  - mouvement des tirs avec des angles
  - création d'animations (reload et explosion)
  - gestion des collisions (vaisseau-ennemis, vaisseau-tirs, ennemis-tirs)
  - creation menu et score
- 

## Documentation technique

### Guide d'installation

Pour installer le jeu, il faut d'abord cloner le dépôt git sur votre machine en utilisant la commande suivante :

```
git clone https://github.com/ahmedSOUSSI21/shootingOdyssey.git
```

Ensuite, il faut compiler le code source en utilisant la commande suivante :

```
make
```

Et enfin, il faut exécuter le fichier exécutable en utilisant la commande suivante :

```
./main
```

## Guide d'utilisation

Pour jouer, il faut utiliser les touches suivantes :

- **Z** pour se déplacer vers le haut.
- **Q** pour se déplacer vers la gauche.
- **S** pour se déplacer vers le bas.
- **D** pour se déplacer vers la droite.
- **Espace** pour tirer.
- **Echap** pour mettre le jeu en pause.

## Documentation du code source

La documentation du code source est disponible en ouvrant le fichier [doc.html](#) à la source du projet avec un navigateur web.

---

## Conclusion

### Bilan du projet

Pour conclure, ce projet nous a permis de mettre en pratique les connaissances acquises en cours de programmation en C.

Autres compétence acquises :

- Utilisation de la libMLV pour la création d'une interface graphique et la gestion des événements.

### Perspectives d'amélioration

Pour améliorer le jeu, on peut ajouter les fonctionnalités suivantes :

- Ajouter un Boss à la fin du niveau
- Ajouter des bonus

- Ajouter des niveaux

## Remerciements

Nous tenons à remercier notre professeur de programmation en C, **M. Borie**, pour son aide et ses conseils.

---