BERRADI Ilyass, PADJA Bryan

Rapport : Dragon Killer

Sommaire

Introduction

Objectifs du projet

Le but de ce projet **consiste à créer une application graphique en temps réel pour un jeu de type "Shoot'em up", où le joueur contrôle un avion et doit éliminer tous les ennemis.**
L'application doit gérer le taux de rafraîchissement de l'écran, mesuré en **"Frames per second"** (images par seconde).

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.
Le jeu se déroule dans un environnement deux dimensions, notre thème c'est des affrontement de dragon en reference au film Godzilla vs King monster. Le joueur a le contrôle d'un dragon rouge qui peut se déplacer librement dans toutes les directions. Son objectif est d'esquiver les ennemis et leurs projectiles tout en les éliminant.
Le jeu prend fin lorsque le joueur perd toutes ses vies.

Conception de l'interface graphique

L'interface graphique est composée de plusieurs éléments :

- Un **fond d'écran** qui représente l'ocean et qui défile de bas en haut.
- Un **Dragon rouge** 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 et de gauche à droite.
- Des **tirs** qui sont lancés par le player et les ennemis.

Implémentation

Organisation du code source

Dans le cadre de ce projet, nous avons adopté l'utilisation de fichiers d'en-tête (".h") pour la déclaration et la documentation des fonctions et des structures. Ces fichiers d'en-tête sont regroupés dans un dossier dédié nommé "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 la police d'écriture du menu.

Développement des fonctionnalités principales

- le création du joueur
- la création des ennemis
- les mouvements du joueur
- les mouvements des ennemis
- création des tirs
- création d'animations (mouvement des tirs de feu et battement des ailes des dragons)

- gestion des collisions (joueur-ennemis, joueur-tirs, ennemis-tirs)
- creation menu et score

Gestion des interactions utilisateur

- Réponses aux interactions : Lorsqu'une touche est pressée, le jeu réagit en conséquence. Par exemple, le mouvement du dragon doit être mis à jour à chaque frame en fonction des touches directionnelles enfoncées.
- Gestion des événements utilisateurs : Outre les mouvements et les tirs, il faut également gérer d'autres types d'interactions, comme la mise en pause du jeu lorsque l'utilisateur appuie sur la touche "P", ou la fermeture du jeu lorsque l'utilisateur clique sur le bouton "Q" du clavier.

Gestion des événements graphiques

- Rafraîchissement de l'écran (FPS) : on a apprit à mettre en place une boucle de jeu qui actualise l'écran à un taux défini par les FPS. Ceci assure que le jeu se déroule à un rythme constant, quelles que soient les performances de l'ordinateur de l'utilisateur.
- Dessin des entités du jeu : À chaque rafraîchissement, nous devons redessiner le joueur, les ennemis et les tirs à leurs nouvelles positions.
- Gestion des collisions : Une partie importante de la gestion des événements graphiques a été la détection et le traitement des collisions. Ceci nous l'avons vérifié à chaque frame si un tir a touché un ennemi, si le joueur a été touché, etc.
- Affichage des informations : Avec la libMLV, nous avons afficher du texte à l'écran pour montrer des informations pertinentes, comme le score et la quantité de vie restante du joueur.

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/IlyassBERRADI/Projet_C.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 :

- **fleche haut** pour se déplacer vers le haut.
- **fleche gauche** pour se déplacer vers la gauche.
- **fleche bas** pour se déplacer vers le bas.
- **fleche droite** pour se déplacer vers la droite.
- **Espace** pour tirer.
- **p** pour mettre le jeu en pause.(utiliser avec modération)
- **q** pour quitter la partie.

regle du jeu, score

- Un ennemie tué vaut +50 de score
- La collision joueur-ennemie enlève -10 au score mais détruit l'enemie rencontré
- Bonus coeur vaut +1 une vie
- La jeu est un survival, donc tenir le plus longtemps aux rangés ennemies et obtenir un meilleur score sont l'objectifs des parties.
- Calcul du score final : Ton score + nombre d'ennemies tué * 100

Documentation du code source

files : DocTX.pdf

Conclusion

Problèmes

- Durant certain partie, lorsqu'il y a une collision du tir joueur et de l'ennemie, defois il arrive que l'autre dragon disparaisse aussi.
- L'option pause à du mal durant une partie.

Bilan du projet

En résumé, ce projet a été une occasion pour nous d'appliquer les connaissances que nous avons acquises lors de notre formation en programmation en langage C
Autres competence acquises :

- Nous avons intégré la librairie MLV dans notre projet pour créer une interface graphique et gérer les événements de manière efficace.
- Travailler sur les pointeurs dans ce contexte, nous a permit de mieux comprendre et d'etre d'avantage rigoureux
- Apprendre à construire un jeu video
- Meilleur connaissance des frames
- Développer des fonctions qui ont des paramètres aléatoires
- Développer des entités qui font plusieurs actions en même temps
- La gestion des vitesses
- On a apprit à faire un travail complexe et regulier sur plusieurs mois

Perspectives d'amélioration

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

- Ajouter un Boss à la fin du niveau
- Ajouter davantage de bonus
- Ajouter des niveaux
- Augmenter la qualité de nos parties
- Meilleur gestion de fin de partie

Remerciements

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