

Rapport de projet

Scotto Lucas : Développeur interface menu, gestion sonore et documentation

Fodeba Fofana : Développeur moteur et gameplay

Introduction

Ce projet vise à créer un jeu 2D de type "Space Shooter" en utilisant le framework LibGDX.

- Le joueur contrôle un vaisseau spatial qui combat des ennemis, accumule des points et monte en niveau.
- La difficulté augmente progressivement, offrant une expérience de jeu dynamique et engageante.
- L'intégration de Tiled permet la gestion d'arrière-plans variés et interactifs pour enrichir l'univers graphique.

Présentation du projet

Le jeu consiste à contrôler un vaisseau spatial que l'on peut déplacer dans les quatre directions. L'objectif est de survivre le plus longtemps possible tout en détruisant des ennemis.

Technologies et outils utilisés

- **LibGDX** : Fournit les outils nécessaires pour structurer le moteur de jeu, gérer les graphismes, les entrées utilisateur et les performances globales.
- **Tiled** :
 - Permet de concevoir des arrière-plans détaillés pour chaque niveau.
 - Simplifie l'ajout et la configuration de nouvelles cartes sans modifier le code source.
 - Assure une projection orthographique précise grâce à *OrthogonalTiledMapRenderer*, en s'appuyant sur *TmxMapLoader* pour charger les fichiers .tmx.

Fonctionnalités principales

- **Gameplay** : Le joueur doit détruire un certain nombre d'ennemis pour monter de niveau.
- **Gestion de la difficulté** : La difficulté augmente à mesure que le joueur avance, avec des ennemis de plus en plus nombreux ou rapides.
- **Modularité** : Le code est conçu pour permettre l'ajout futur de nouvelles fonctionnalités ou éléments graphiques (ennemis, niveaux, objets interactifs).
- **Gestion du son** : Dans les paramètres, il est possible de régler la musique du jeu en appuyant sur les touches "increase" et "decrease".
- **Système d'échappatoire** : Il est possible d'arrêter une partie en appuyant sur la touche *Échap* (*revenir au menu*).
- Gestion du son: dans les paramètre il est possible en appuyant sur les touche "increase" et "decrease" de réglé la musique du jeu.
- Systeme d'échappatoire: il est possible d'arrété une partie en appuaynt sur la touche échap.

Configuration et ajout de contenu avec Tiled

- **Modification des arrière-plans** : Pour modifier l'arrière-plan du jeu, ouvrez le fichier FullBackground.tmx avec Tiled et ajoutez autant de tuiles que souhaité, à condition que les tuiles soient dans le dossier *assets*. Vous pouvez ensuite ajuster le fond du jeu à votre guise.
- Si le fichier FullBackground.tmx est manquant ou non lisible, le fond par défaut (background.txt) sera utilisé.

Prérequis :

- Java 17
- LibGDX 1.13.1

Compilation du projet :

Exécutez les commandes suivantes pour compiler le projet :

```
./gradle core:build lwjgl3:build
```

Exécution du projet :

Lancez le projet avec la commande suivante :

```
./gradlew lwjgl3:run
```

Générer un fichier .jar :

Builder le fichier avec la commande suivante :

```
./gradlew build
```

*[Si une commande n'est pas reconnue, utilisez « **gradle** » au lieu de « **./gradlew** ». Si l'erreur persiste, il est probable que la version de Java soit trop ancienne.]*

Réglage de départ:

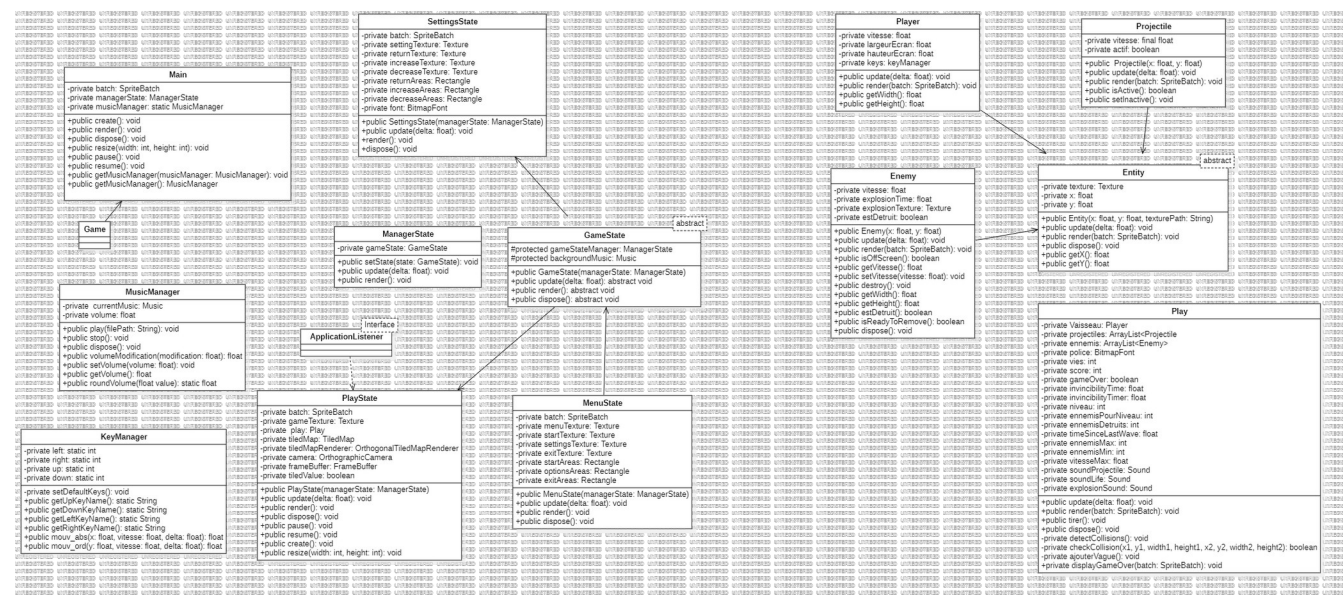
Bien qu'il n'y ait pas de règle précise pour lancer le jeu, une configuration par défaut est fournie. Vous pouvez cependant ajuster le son dans le menu «*Settings*».

Lien vers le dépôt GitHub : <https://github.com/Stratis-LS/Java-Project>

Architecture générale du moteur de jeu

Le projet est organisé en plusieurs packages : «state», «play», «key», «sound».

Diagramme UML:



Utilisation et extension de la librairie du moteur de jeu

L'extension de la librairie est simple. Par exemple, on peut ajouter des ennemis en créant des classes qui étendent la classe Enemy, en modifiant la vitesse pour avoir des ennemis avec des vitesses différentes. Grâce aux setters de la classe Play, il est possible de modifier la configuration du jeu, comme ajouter des objets qui donnent des vies ou ajuster le nombre d'ennemis par niveau.

Répartition des tâches

Scotto Lucas : Développeur interface menu, gestion sonore et documentation

- Codage du fichier *Main*, des fichiers *states*, du *KeyManager* et du *MusicManager*.
- Réalisation du diagramme UML et de la documentation Javadoc pour l'ensemble des fichiers.
- Adaptation de la classe *Play* en fonction de la caméra de *PlayState*.

Fodeba Fofana : Développeur moteur et gameplay

- Codage des classes du dossier *Play*.
- Versification des différentes commandes de compilation et d'exécution
- Intégration de Tiled dans le projet.

Collaboration des deux :

- Collaboration dans l'optimisation du gameplay et la gestion du fichier *Main*.

Conclusion et perspectives

Nous avons largement atteint nos objectifs dans ce projet, bien que certaines fonctionnalités comme le menu de configuration des touches du clavier n'aient pas pu être terminées, faute de temps. Des difficultés ont été rencontrées, notamment :

- **Intégration de Tiled** : La compréhension du fonctionnement des fichiers .tmx avec le moteur de jeu a pris du temps, la documentation n'étant pas toujours claire sur certains points.
- **Gestion des performances** : Certaines configurations graphiques ont entraîné des ralentissements, nécessitant une optimisation du rendu des couches Tiled.
- **Équilibrage du gameplay** : Trouver le bon rythme d'augmentation de la difficulté et la vitesse des ennemis a nécessité plusieurs ajustements.
- **Apprentissage de LibGDX** : Travailler avec cette bibliothèque pour la première fois a impliqué une phase d'apprentissage importante.

Nous aurions aimé implémenter les fonctionnalités suivantes :

- Un menu pour changer les touches du clavier (fonctionnalité partiellement implémentée).
- Un système de "meilleurs scores" enregistrés sur un fichier JSON.
- La possibilité de changer la texture des ennemis avec Tiled.
- L'ajout de plusieurs types d'ennemis.

Annexe: (lien de site de documentation)

- <https://blogue-nboure.profweb.ca/?p=519&cpage=1>
- <http://tuto-libgdx.blogspot.com/?view=classic>
- <https://gamefromscratch.com/libgdx-tutorial-series/>