

Rapport de Projet - Développement d'un Moteur de Jeu 2D avec LibGDX

Équipe et Contributions

- SANTARELLI Maxime : Chef de projet, Architecte logiciel, 2e Game Developer
- DODE Reese : 1er Game Developer, Game Designer
- TEVDORADZE Irakly : Développer des tests

Dans ce rapport vous trouverez un résumé de notre projet. Pour plus de détails sur l'architecture, le code java, les tests, ou toute autre information, nous vous invitons à consulter la documentation sur notre page github :

Lien github du projet : <https://github.com/Xk04/Game-Engine>

Section 1. Introduction

Ce projet a été réalisé dans le but de mettre en application les principes et concepts vus durant le cours de PCOO. Notre objectif principal était de concevoir un moteur de jeu extensible. Nous avons donc essayé de développer une structure capable de servir de base à n'importe quel jeu de plateforme. Si un utilisateur souhaite reprendre notre travail afin de créer son propre jeu, l'architecture du moteur lui permettra de le faire avec une grande facilité.

Un moteur conçu pour l'extensibilité, c'était l'un de nos objectifs avec la création de niveaux via Tiled, tout en ayant une bonne architecture (MVC), ainsi qu'une gestion fluide des interactions physiques.

Section 2. Présentation du projet

Afin de réaliser ce moteur de jeu, nous avons utilisé différentes technologies.

Utilisation du framework LibGDX fournissant une gamme d'outils facilitant la création de fenêtres et de sprites compatibles avec Tiled.

Tiled a été utilisé pour la création de map.

Canva nous a permis d'effectuer le diagramme de classes UML.

Enfin GitHub pour la gestion de version du projet.

Pour toute information complémentaire à l'utilisation de l'un de ces outils, nous vous invitons à consulter la documentation. Celle-ci propose des explications sur l'architecture, l'utilisation de Tiled, l'utilisation du projet, et les tests unitaires.

Lien : <https://github.com/Xk04/Game-Engine/blob/main/README.md>

Compilation et exécution

Une fois le projet téléchargé, pour compiler et exécuter celui-ci:

Prérequis : JDK 21 ou supérieur

1. Ouvrez un terminal et naviguez jusqu'au répertoire où se trouve le projet.
2. Utilisez la commande suivante : `./gradlew lwjgl3:run`
3. Si vous obtenez une erreur concernant les permissions du fichier, vous devez donner les permissions d'exécution au script avec la commande suivante :

`chmod +x gradlew`

Autres informations à cette adresse:

https://github.com/Xk04/Game-Engine/blob/main/docs/doc_files/presentation/continue.md

Section 3. Présentation technique du projet et contributions

Répartition des tâches

SANTARELLI Maxime :

- Chef de projet
- Conception et mise en place de l'architecture globale du projet
- Diagramme UML
- Rédaction de la documentation
- Mise en place des classes structurales pour les dossiers model, controller, view.
- Implémentation de la caméra dynamique
- Implémentation du changement de niveau automatique
- Refactorisation massive du code pour garantir la propreté et maintenabilité du projet

DODE Reese :

- Conception des maps sur Tiled(gestion des calques "Sol", objects "Start", "End" et "Wall")
- Recherche et ajouts des assets du Game Engine
- Diagramme UML
- Rédaction de la documentation et du rapports
- Mise en place des collisions entre le joueurs et le sol
- Récupération du spawn(Start) et de la EndZone
- Implémentation de l'animation du joueurs
- Gestion des inputs pour faire bouger le player

TEVDORADZE Irakly :

- Tests Unitaires
- Rédaction de la documentation sur la partie tests unitaires

Remarque: Certaines classes ont été produites en commun. Plus précisément chacun d'entre nous a proposé différentes versions qui ont permis le développement du moteur.

Section 4. Conclusion et Perspectives

Ce projet nous a permis de mettre en application les concepts et principes de la programmation orienté objets. Nous avions idée de favoriser la maintenabilité afin de faciliter la production de nouvelles fonctionnalités dans le futur pour n'importe quel utilisateur. Notamment grâce à la mise en place de design patterns.

Bien que le moteur soit fonctionnel, plusieurs évolutions pourraient être envisagées pour le rendre encore plus complet.

Par exemple:

- ajouter des effets sonores,
- animations poussées : double saut ou glissade,
- principe de checkpoint pour ne pas revenir au début du niveau à chaque fois (même si on aime la difficulté dans un jeu).

Merci de l'attention que vous portez à notre projet, nous restons à disposition pour toute information complémentaire.

Maxime, Reese, Irakly

Maxime Santarelli : maxime.santarelli@etu.unice.fr

Reese Dode : reese.dode@etu.unice.fr

Irakly Tevdoradze : irkly.tevdoradze@etu.unice.fr