

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

“RPGMAKER avec peu de budget”

I. Introduction

Mon objectif était de créer un moteur me permettant de créer un jeu de type rogue like/RPG dans un monde remplis de cartes interconnecté qui me permettrait de créer plein de monde possible directement via Tiled. Que ce soit les textures ou les monstres, leur position et les interactions entre les cartes.

II. Présentation du projet

A. Technologies et outils utilisés

Seulement libgdx et Tiled.

B. Fonctionnalités implémentées

Passage d'une carte à l'autre par des zones de téléportations, système basique de tir avec la touche “a”, système basique de combat en temps réel contre des monstres, système basique de niveau avec augmentation des statistiques, reset des statistiques en cas de mort et zone de collisions.

C. Configuration et Ajout de Contenu avec Tiled

Il y a trois types de typographie de couche d'objet, importantes à respecter, les rectangles de collisions doivent être appelé “`collisions`”, les téléporteurs “`teleporters`” et les ennemis “`enemies`”.

Pour créer une nouvelle carte il suffit dans Tiled de créer une nouvelle carte avec en taille 30 tuiles de largeur et 20 de hauteur avec des tuiles en 32*32, puis d'en dessiner une avec un tilesset et y rajouter les différents objets souhaités. Les téléporteurs doivent se voir rajouter des propriété personnalisé int targetX et int targetY, les coordonnées de la carte atteinte en prenant le téléporteur.

Les ennemis ont seulement besoin d'un int pour leur point de vie.

Tous les objets doivent être dessinés avec l'outil rectangle.

Pour chaque carte visitée, le moteur récupérera les données de chaque couche d'objet et récupérera les différentes positions et données liées à l'objet, puis les placera dans le monde.

Les collisions deviendront des murs rectangulaire invisible, les ennemis apparaîtront à l'endroit ciblé et les téléporteurs seront aussi des rectangle invisibles il est donc recommandé de leur rajouter une texture sur la couche de dessins des tuiles.

Une fois créer les cartes devront être déposée dans le dossier core/assets/maps et chacunes des cartes devra être nommées map_x_y.tmx où x et y représente ses coordonnées de la carte dans le monde

D. Compilation et exécution

Dans le dossier parent de src : gradlew build pour compiler puis gradlewn lwjgl3:run pour lancer l'exécution.

Version de java : openjdk version "21.0.6" 2025-01-21 LTS.

version libgdx : 1.14.0.3.

III. Présentation technique du projet

A. Architecture générale du moteur de jeu

Les projectiles, les ennemis et le joueur sont des classes héritant de la classe entity qui leur donne texture vitesse position destination,

Le worldManager sert à charger les bonnes cartes dans le mapmanager qui lui sert à récupérer les cartes et les différentes couches d'objets.

Pour finir il y a la gameScreen qui implémente le code reliant tous les éléments.

B. Utiliser et étendre la librairie du moteur de jeu

Les différents types d'objet et de personnage amis ou ennemis hériteront de la classe entity.

Conclusion et perspectives

De base je souhaitais faire un moteur de jeu permettant de créer des jeux de type MMORPG comme dofus, mais pour avoir un minimum de choses à créer dans le moteur la charge de travail était un peu trop haute j'ai donc décidé de prendre la base que j'avais commencé et de rajouter un système de tir.

En terme d'extension de contenu il aurait été intéressant de rajouter une classe sort permettant d'envoyer différent type de projectile. On pourrait aussi rajouter une classe ennemis plus ouverte permettant d'en avoir plusieurs types différents. La montée de niveau pourrait être améliorée pour obtenir de nouveaux sorts / plus de statistique varié.