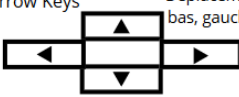



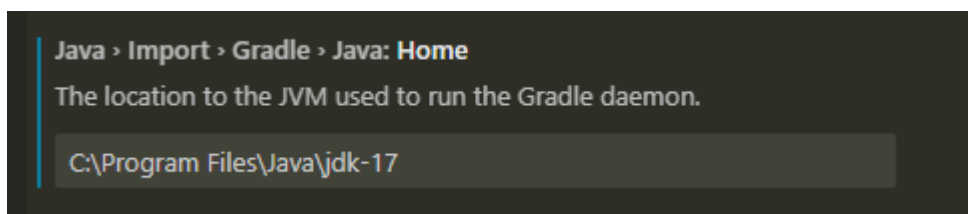
Introduction

Cette application est développée en Java, avec le Framework LibGDX. Il s'agit d'un jeu vidéo de type shoot 'em up en 2D à défilement horizontal. Dans ce jeu, vous contrôlez un avion de combat capable de se déplacer dans toutes les directions et de tirer des projectiles. Sur fond de décor urbain, des dragons apparaissent et vous attaquent, eux aussi armés de projectiles.

Cheat Code	Mapping des commandes
<div style="border: 1px solid black; padding: 5px; width: 50px; margin: 0 auto;">X</div> <p>ajoute des cœurs pour tester le jeu</p>	<div style="display: flex; flex-direction: column; align-items: center;"> <div style="margin-bottom: 20px;"> <p>Arrow Keys</p>  </div> <div style="margin-bottom: 20px;"> <div style="border: 1px solid black; padding: 5px; text-align: center;">Left Shift</div>  <p>Ralenti la vitesse de mouvement</p> </div> <div> <div style="border: 1px solid black; padding: 5px; text-align: center;">SpaceBar</div> <p>Envoyer des projectiles</p> </div> </div>

Dépendances

Pour utiliser le jeu, il est nécessaire d'installer la version 17 de Java. Ensuite, il suffit de lancer l'application. Si vous utilisez Visual Studio Code, vous devrez configurer Gradle dans les paramètres (préférence > Settings) et ajouter le chemin d'installation de Java 17 dans l'option "Home".



Liens de téléchargement de Java : [Page Download](#)

Liens du GitHub : [DragonGaiden](#)

Classe et méthode

Player :

- **Attributs**
 - x - float
 - y - float
 - size - int
 - characterImg - Texture
 - character - Sprite
- **Méthodes**
 - Player(x - float, y - float, size - int)
 - move(x - float, y - float) - void

- setPosition(x - float, y - float) - void
 - get/set - x, y, size
 - draw(batch - SpriteBatch) - void
 - returnSprite() - Sprite
-

Monster :

- **Attributs**
 - x - float
 - y - float
 - size - int
 - type - int
 - monsterSheet - Texture
 - monster - Sprite
 - animation - Animation
 - stateTime - float
 - step - int
 - **Méthodes**
 - Monster(x - float, y - float, size - int, type - int)
 - update() - void
 - draw(batch - SpriteBatch) - void
 - move(x - float, y - float) - void
 - get/set - x, y, size, type, step
 - nextStep() - void
 - resetStep() - void
 - returnSprite() - Sprite
 - dispose() - void
-

PowerUp :

- **Attributs**
 - x - float
 - y - float
 - largeur - float
 - hauteur - float
 - powerImg - Texture
 - power - Sprite
 - type - int
 - **Méthodes**
 - PowerUp(x - float, y - float, largeur - float, hauteur - float, type - int)
 - move(x - float, y - float) - void
 - get/set - x, y, type
 - draw(batch - SpriteBatch) - void
 - returnSprite() - Sprite
-

Shoot :

- **Attributs**
 - x - float
 - y - float
 - largeur - float
 - hauteur - float
 - shootImg - Texture
 - shoot - Sprite
 - **Méthodes**
 - Shoot(x - float, y - float, largeur - float, hauteur - float, type - int)
 - move(x - float, y - float) - void
 - draw(batch - SpriteBatch) - void
 - returnSprite() - Sprite
-

SQLiteManager :

- **Attributs**
 - URL-String"jdbc:sqlite:dragonGaidenDb.db"
 - **Méthodes**
 - SQLiteManager()
 - insererScore(pseudo - String, score - int, date - String) - void
 - recupererTopScores() - ArrayList<String[]>
-

Point d'amélioration

- Introduire des ennemis au comportement plus imprévisible et à la position aléatoire, car dans la version actuelle du jeu, ils apparaissent toujours au même endroit.
- Ajouter divers environnements pour les niveaux (par exemple, une ville, une montagne, etc.) afin de varier l'expérience de jeu.
- Intégrer une page d'options. Toutefois, cela nécessiterait une refonte du code, avec une organisation par classes pour chaque page, car l'optimisation actuelle ne permet pas cette implémentation.
- Inclure différents types de projectiles pour enrichir les mécaniques de combat.

Expérience personnelle

La création de ce jeu en un peu plus d'un mois m'a permis de mieux appréhender le développement de jeux en Java. Bien que j'aie déjà expérimenté cet exercice avec Processing, cette nouvelle expérience m'a donné envie de concevoir prochainement un jeu plus sophistiqué et mieux structuré, en tirant parti des connaissances acquises durant ce projet. J'ai également pris beaucoup de plaisir à concevoir ce jeu, ce qui a renforcé ma motivation et mon intérêt pour le développement.