

Documentation Technique

I. Introduction

A. Présentation du jeu

Ce projet est un jeu développé avec le moteur Godot. Le joueur contrôle un vaisseau spatial devant survivre à des vagues de météores et d'ennemis, en utilisant des tirs et des déplacements pour éviter les collisions.

B. Objectifs de la documentation technique

Cette documentation vise à fournir une vue d'ensemble technique du projet, détailler l'architecture logicielle, les choix de conception, et servir de référence pour la maintenance et l'évolution du jeu.

II. Architecture du jeu

A. Moteur de jeu

Le jeu repose sur Godot Engine, qui gère la boucle principale, la physique, les entrées et la gestion des scènes.

B. Système de rendu

Le rendu 2D est assuré par Godot : chaque objet (joueur, météore, projectiles) est une Node2D ou Sprite, affiché via le système de scènes (.tscn).

C. Gestion des ressources (graphismes, sons, etc.)

Les ressources sont organisées dans des dossiers dédiés :

- Sprites pour les images et spritesheets
- Sounds pour les effets sonores et musiques
- Scripts pour les scripts GDScript
- HUD pour l'interface utilisateur

III. Conception du gameplay

A. Mécaniques de jeu

- Déplacement du joueur (tourner avec la souris, tir)
- Apparition aléatoire de météores
- Accélération de l'apparition des météores au fil de la partie
- Gestion des collisions (joueur, projectiles, météores)
- Système de score et de vies

B. Systèmes de contrôle

Gestion des entrées via le système Input de Godot :

- Cliquer pour tirer
- Pointer avec la souris pour viser

C. Interactions joueur-environnement

- Collision avec les météores : perte de vie
- Destruction des météores par les tirs
- Apparition de nouveaux ennemis

IV. Graphismes et Animation

A. Sprites 2D

Sprites trouvés sur internet ou dessinés à la main grâce au logiciel Aseprite.

B. Textures

Utilisation de sprites d'image pour les objets, et de spritesheets pour les animations.

C. Animation des personnages et objets

Animation des explosions via spritesheet.

Déplacement animé des objets (météores) vers le player.

V. Son et Musique

A. Effets sonores

Sons pour tirs, explosions, dégâts, mort

Stockés dans le fichier Sounds

B. Musique d'ambiance

Musiques de fond dans Sounds/bgMusic

C. Intégration sonore dans le gameplay

Déclenchement des sons via signaux Godot dans les scripts (ex : collision, tir)

VI. Optimisation et Performance

A. Optimisation du code

- Utilisation de signaux pour limiter le couplage
- Boucles optimisées pour la gestion des entités

C. Tests de performance

- Tests manuels sur différentes machines
- Surveillance de la consommation mémoire et CPU via Godot

VII. Documentation du Code

A. Structure du code source

Scripts GDScript dans Scripts :

- Un script par entité principale (joueur, météore, gestionnaire de spawn, etc.)

B. Bonnes pratiques de codage

- Respect de la convention GDScript
- Découpage en fonctions courtes et lisibles
- Utilisation des signaux pour la communication entre objets

VIII. Déploiement et Maintenance

A. Configuration requise

Godot Engine 4.5

Système Linux, Windows ou MacOS

B. Procédure d'installation

Cloner le dépôt

Pour le debug :

- Ouvrir le projet avec Godot
- Lancer la scène principale (menu.tscn)

Pour la version compilée :

- Lancer Meteor-Defender.exe pour Windows
- Lancer Meteor-Defender pour Linux

C. Gestion des mises à jour et correctifs

- Utilisation de Git pour le versioning
- Nom de branches claires
- Pull requests