

Documentation du Mini Projet : Bataille Aérienne

1. Introduction

Le projet "**Bataille Aérienne**" est un jeu de tir en 2D développé avec Pygame, un framework Python pour la création de jeux vidéo. Le joueur contrôle un avion qui doit éviter les collisions avec des ennemis et tirer des projectiles pour détruire les ennemis qui tombent du ciel. Le but du jeu est d'obtenir un score élevé en éliminant les ennemis tout en évitant les impacts avec eux.

Ce jeu inclut :

- Un joueur qui peut se déplacer et tirer des projectiles.
- Des ennemis qui tombent du ciel.
- Un système de score basé sur les ennemis abattus.
- Des effets sonores pour le tir, les explosions et la fin du jeu.
- Un écran de fin avec le score final.

2. Prérequis

Avant de pouvoir exécuter le jeu, vous devez avoir Python et la bibliothèque Pygame installés sur votre système.

Installation de Python

1. Télécharger Python depuis python.org.
2. Assurez-vous d'ajouter Python au PATH pendant l'installation (cochez l'option "Add Python to PATH").
3. Vérifiez l'installation en exécutant la commande suivante dans votre terminal :

```
python --version
```

Installation de Pygame

1. Utilisez pip pour installer Pygame :

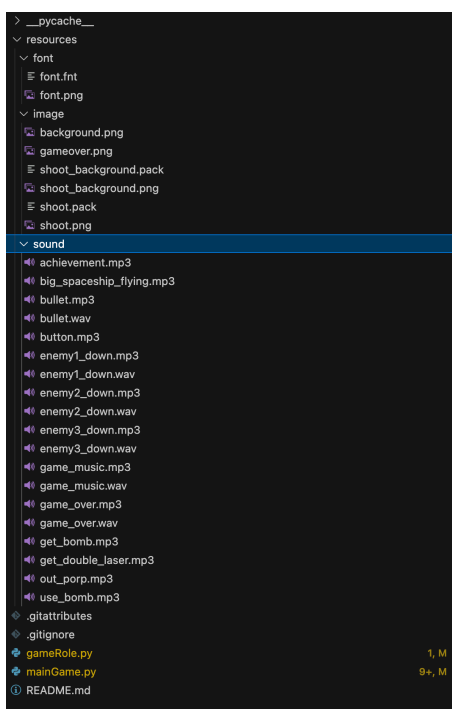
```
pip install pygame
```

Fichiers nécessaires

Assurez-vous d'avoir les fichiers suivants dans votre répertoire de travail pour que le jeu fonctionne correctement :

- **resources/sound/bullet.wav** : Effet sonore pour le tir.
- **resources/sound/enemy1_down.wav** : Effet sonore pour l'explosion des ennemis.
- **resources/sound/game_over.wav** : Effet sonore pour la fin du jeu.
- **resources/sound/game_music.wav** : Musique de fond.
- **resources/image/background.png** : Image de fond du jeu.
- **resources/image/gameover.png** : Image de fin du jeu.
- **resources/image/shoot.png** : Feuille de sprite contenant les images des personnages et ennemis.

3. Structure du Projet: Voici la structure du projet :



4. Description du Code:

4.1. gameRole.py:

Ce fichier contient les classes principales du jeu : **Player** et **Enemy**.

4.1.1. Classe Player:

La classe Player représente le joueur et gère ses actions, telles que le déplacement et le tir.

- **Attributs :**
 - **images** : Liste des images utilisées pour l'animation du joueur.
 - **rect** : Rectangle représentant la position du joueur.
 - **speed** : Vitesse de déplacement du joueur.
 - **bullets** : Groupe de balles tirées par le joueur.
 - **image_index** : Index de l'image actuelle pour l'animation.
 - **is_hit** : Indicateur si le joueur a été touché.
- **Méthodes :**
 - **shoot(bullet_image)** : Permet de tirer une balle.
 - **move_up(), move_down(), move_left(), move_right()** : Permet au joueur de se déplacer sur l'écran.

4.1.2. Classe Enemy

La classe Enemy représente les ennemis du jeu. Elle gère leur mouvement et leur explosion.

- **Attributs :**
 - **image** : Image de l'ennemi.
 - **rect** : Rectangle représentant la position de l'ennemi.
 - **explosion_images** : Liste d'images pour l'animation de l'explosion.
 - **speed** : Vitesse de déplacement de l'ennemi.

- `explosion_index` : Index de l'image actuelle pour l'animation de l'explosion.
- **Méthodes :**
 - `move()` : Déplace l'ennemi vers le bas.

4.2. mainGame.py:

Ce fichier contient le jeu principal, y compris la boucle de jeu et la gestion des événements.

- **Initialisation de Pygame** : Charge les sons, les images et les paramètres du jeu.
- **Boucle de jeu principale** : Cette boucle gère l'affichage, les mouvements du joueur, les tirs et les ennemis.
- **Gestion des collisions** : Vérifie les collisions entre le joueur et les ennemis, et entre les balles et les ennemis.
- **Affichage du score et de la fin du jeu** : Affiche le score et l'écran de fin lorsque le joueur perd.

5. Fonctionnalités:

- **Déplacement du joueur** : Utilisation des touches directionnelles ou des touches W, A, S, D.
- **Tir** : Le joueur peut tirer des projectiles avec la touche Espace.
- **Apparition des ennemis** : Les ennemis apparaissent à intervalles réguliers et tombent du haut de l'écran.
- **Explosions** : Lorsqu'un ennemi est touché, une explosion se produit et l'ennemi est détruit.
- **Fin du jeu** : Le jeu se termine lorsque le joueur entre en collision avec un ennemi.

6. Installation et Exécution du Jeu:

1. Téléchargez tous les fichiers du projet.
2. Ouvrez un terminal et placez-vous dans le répertoire du projet.

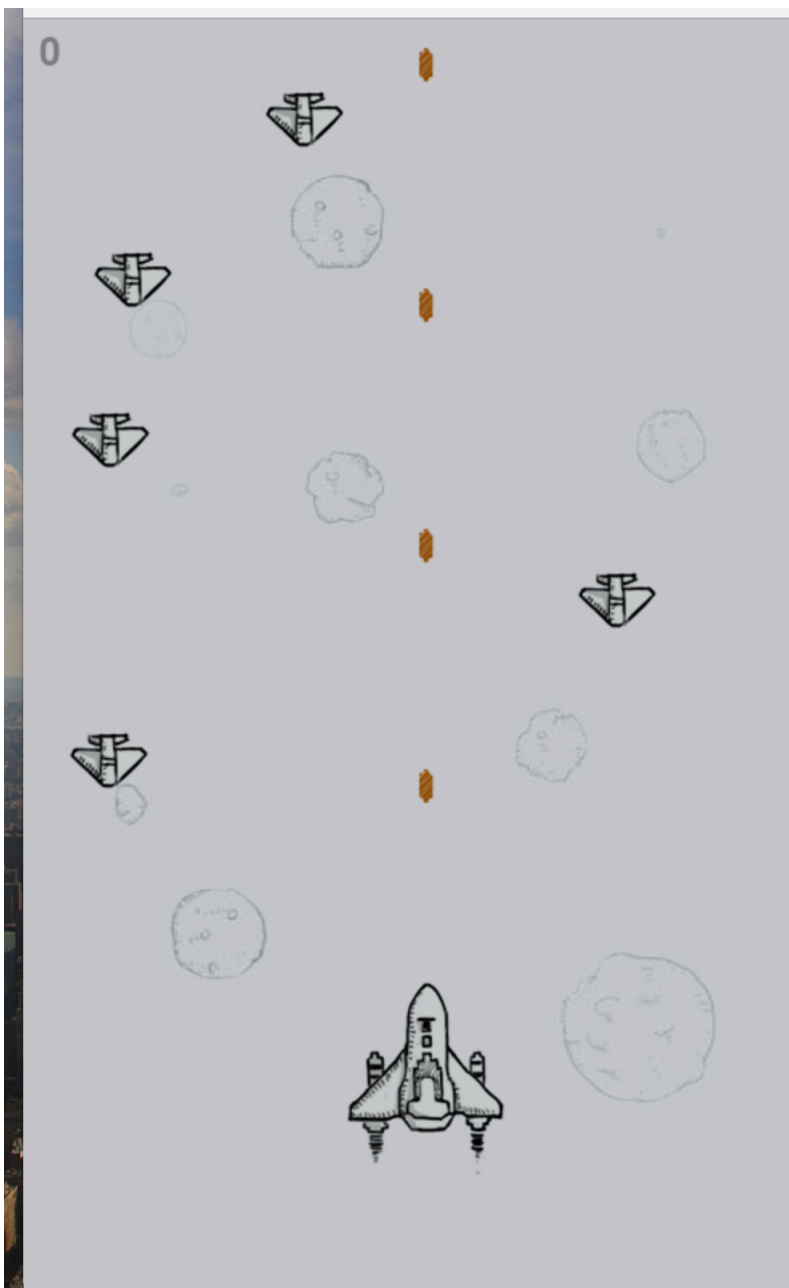
3. Exécutez le jeu avec la commande :

```
python mainGame.py
```

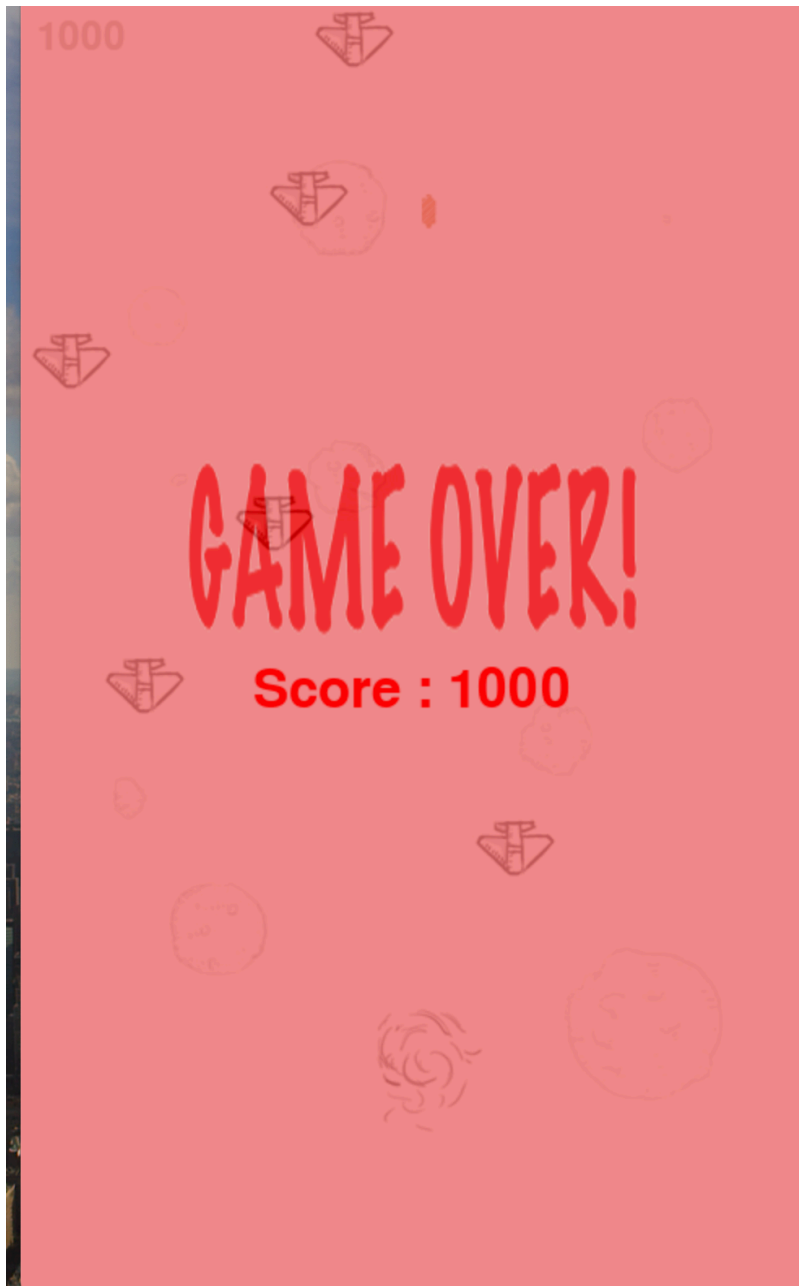
7. Screenshots à Intégrer

Voici les captures d'écran à intégrer dans la documentation pour illustrer le jeu en action.

Capture 1 : Écran de jeu avec le joueur et les ennemis



Capture 2 : Fin du jeu avec le score



8. Conclusion:

Ce projet est un exemple simple de jeu de tir 2D créé avec Pygame. Il permet de mieux comprendre la gestion des événements, des sprites, des collisions et des effets sonores dans un jeu vidéo. Ce jeu peut être étendu avec plus de fonctionnalités, comme des niveaux supplémentaires, des power-ups, ou des ennemis plus variés.

9. Améliorations possibles:

- Ajouter un système de niveaux avec des ennemis de plus en plus rapides.
- Ajouter des armes spéciales ou des power-ups pour le joueur.
- Créer des menus pour démarrer, mettre en pause, et afficher les scores.
- Intégrer un système de vies pour le joueur au lieu d'un seul coup.

BOUNGOU MBIMI Gloire Bryan