Roguelike

Contents

APPADOO APOORVA SRINIVAS

1. Presentation

Ce jeu vidéo a été réalisé en Python 3.7 en utilisant le module *pygame* pour l'interface graphique. L'objectif du jeu est de progresser dans les différents niveaux, chacun ayant un boss dont la difficulté augmente au fur et à mesure. Le seul moyen de passer au niveau suivant est de battre le boss du niveau actuel.

Le héros doit se servir d'armes récupérées dans des coffres pour augmenter sa force de frappe face aux ennemis. Il peut aussi utiliser des potions pour augmenter sa vie, son endurance ou ses points de magie. La magie joue un grand rôle dans le jeu car elle permet d'avoir des avantages significatifs contre les ennemis.

2. Contrôles

Mouvement Générale :

- W,A,S,D ou Z,Q,S,D ou les flèches pour bouger
- Clique gauche pour tirer
- Espace pour accélération

Inventaire :

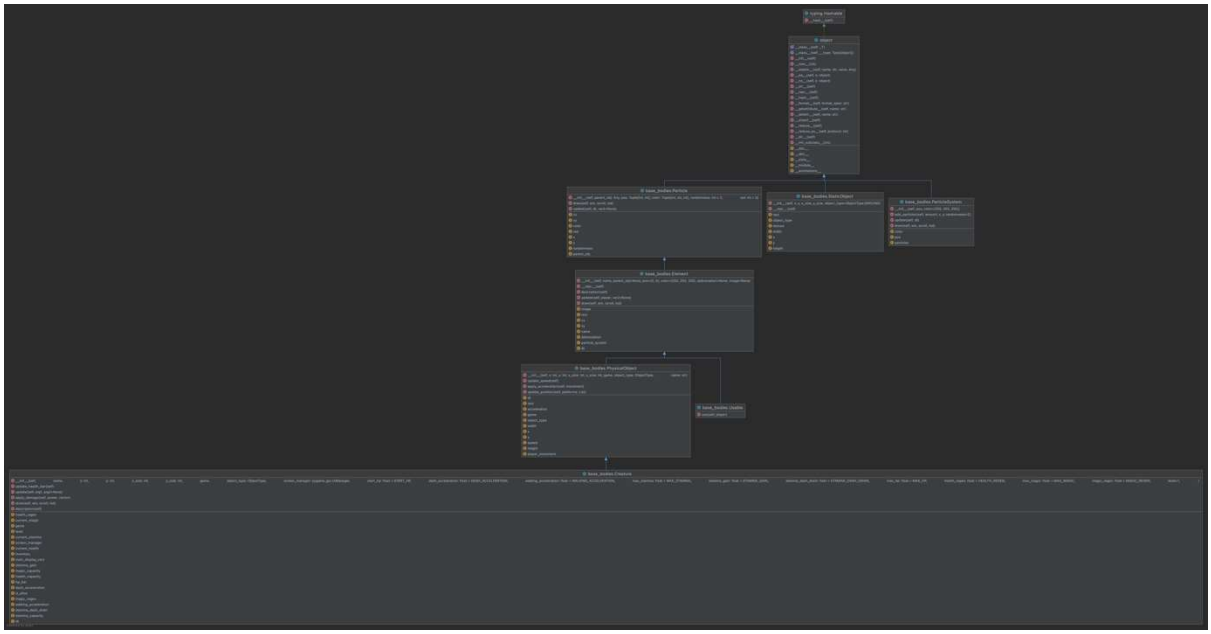
- Clique gauche pour utiliser l'élément
- Clique droit pour supprimer l'élément

Skills :

- On utilise les chiffres du clavier pour activer les skills

3. Description des modules

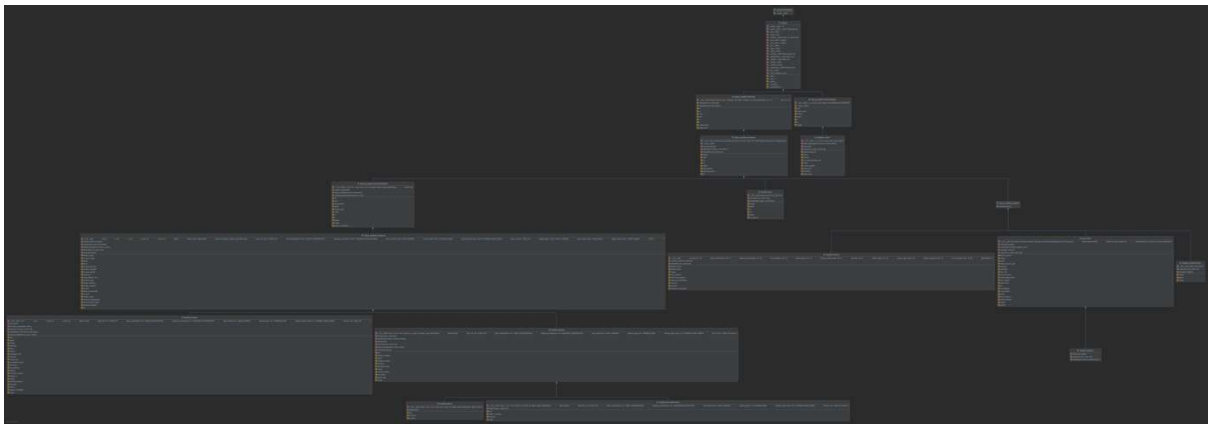
Base_bodies.py:



- Contient les classes de bases pour les éléments

- Particle : Classe d'une particule avec les infos de base accélération, vitesse, position et des méthodes de bases (update qui bouge la particule et draw qui la dessine)
- Particle system : Classe d'un système de particule pour contrôler plusieurs particules à la fois
- StaticObject : Classe des objets statiques
- Element : même classe que le prof mais qui hérite de Particle
- PhysicalObject : Classe pour les objets dynamiques
- Creature : même classe que le prof mais qui hérite du nouveau Element et qui a le ui des créatures
- Usable : Classe abstraite pour les objets qui sont utilisables

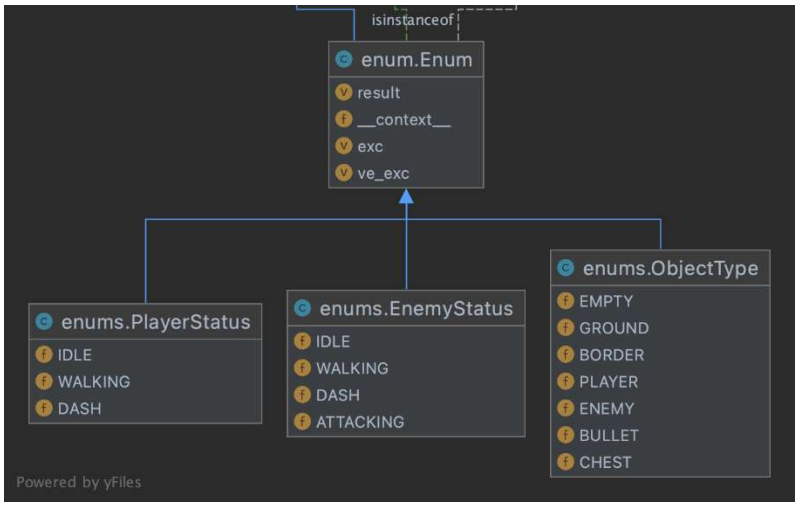
Bodies.py



- Player : classe du joueur
- Enemy : classe de l'ennemi
- Potion : classe de base pour les potions
- Chest : classe de base pour les coffres

Enums.py

- Contient tous les Enum pour la gestion d'états

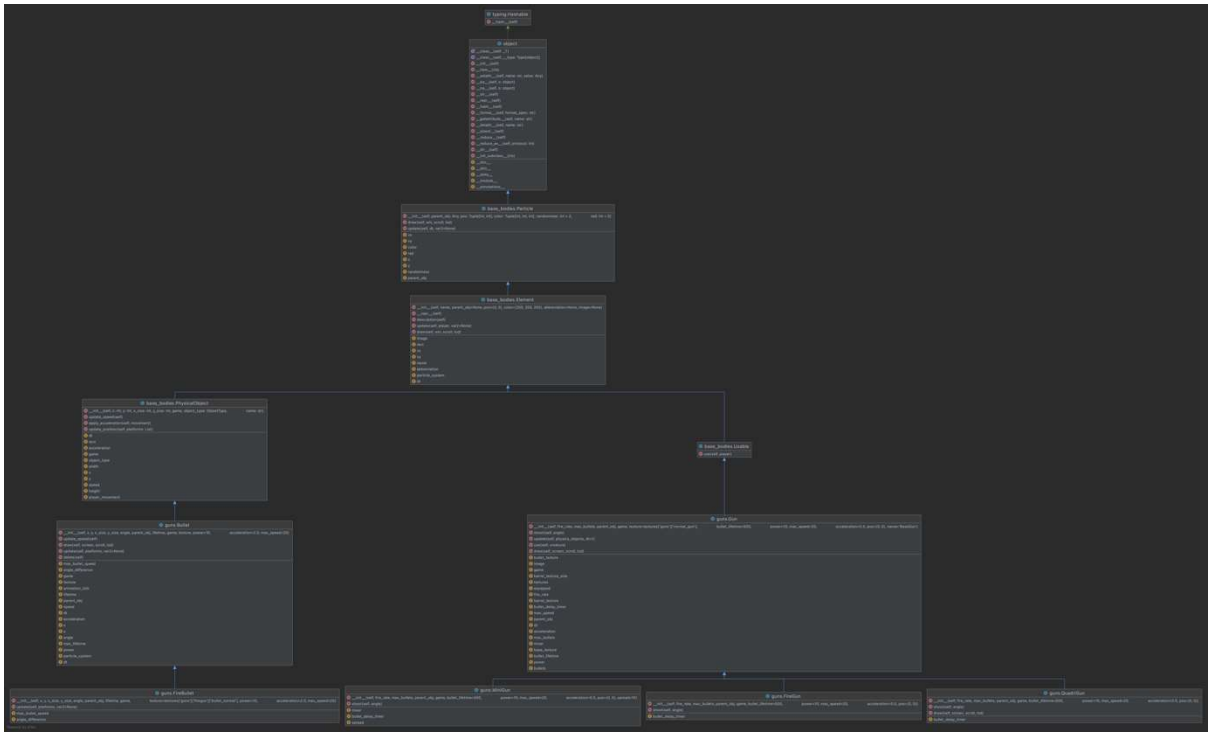


Game_class.py

- Contient la classe Game qui contrôle tout le jeu

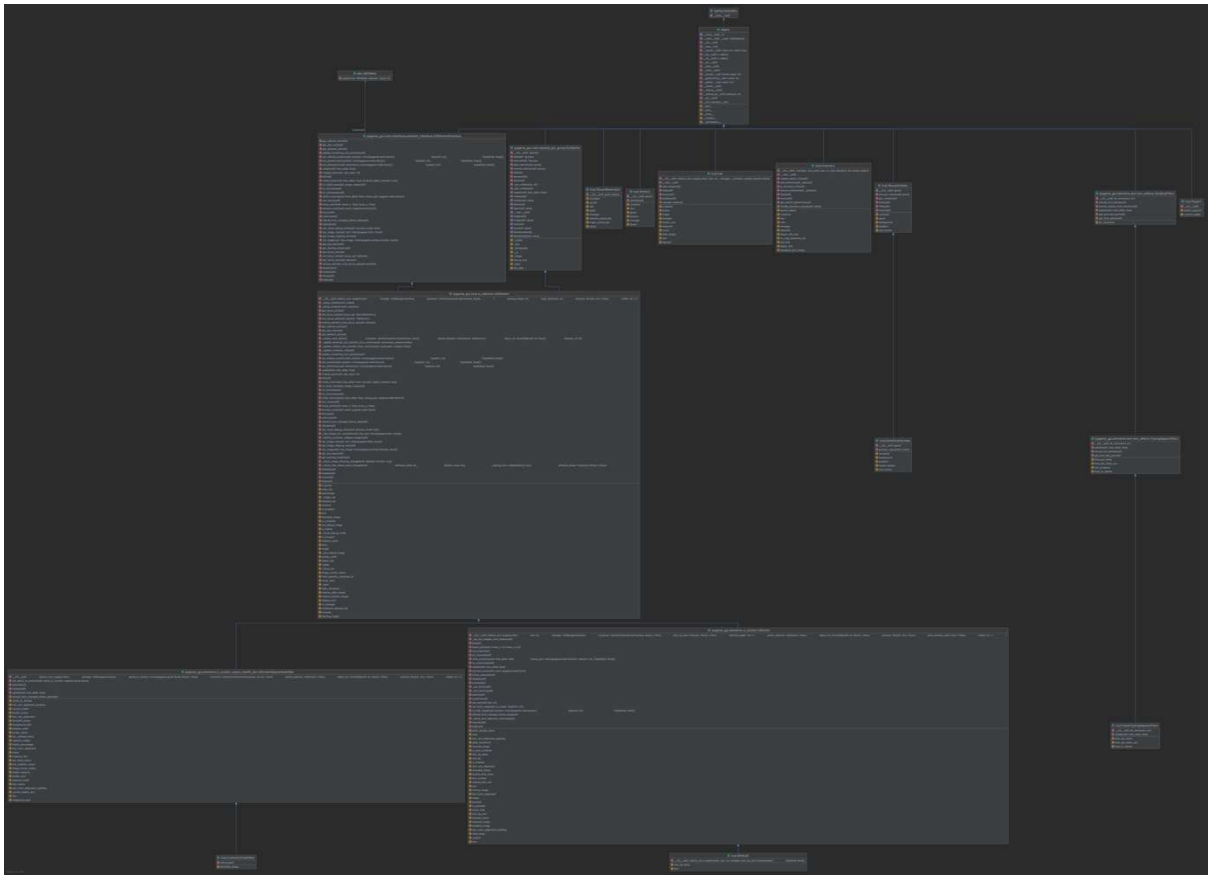
Guns

- Contient toutes les classes en rapport avec les fusils dans le jeu (Bullets et ses sous-classes ainsi que Gun et ses sous-classes

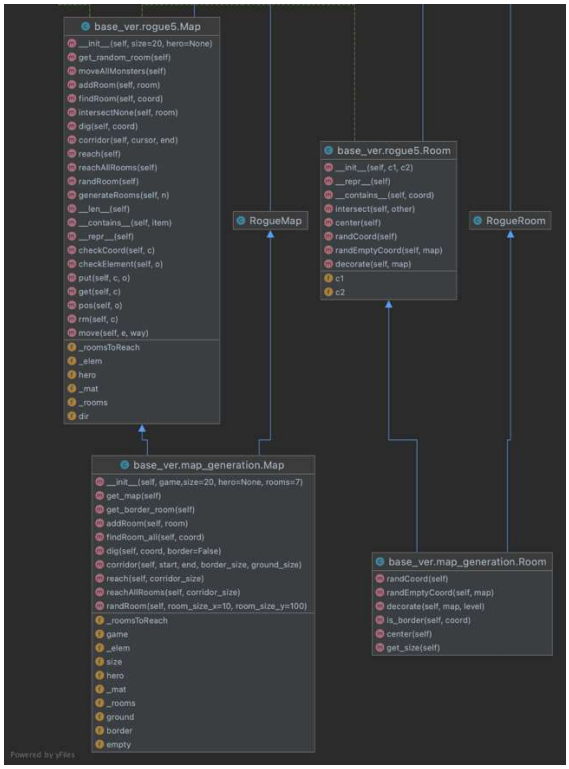


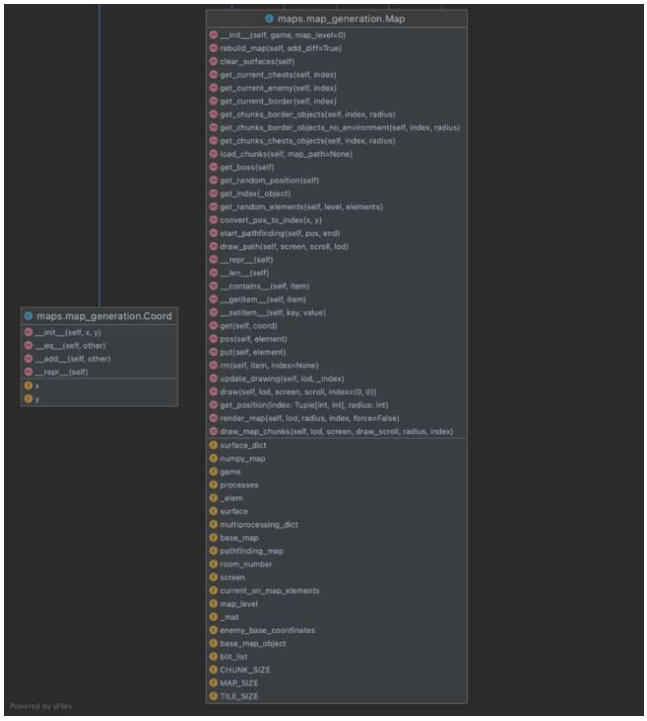
Hud.py

- Contient les classes pour le hud



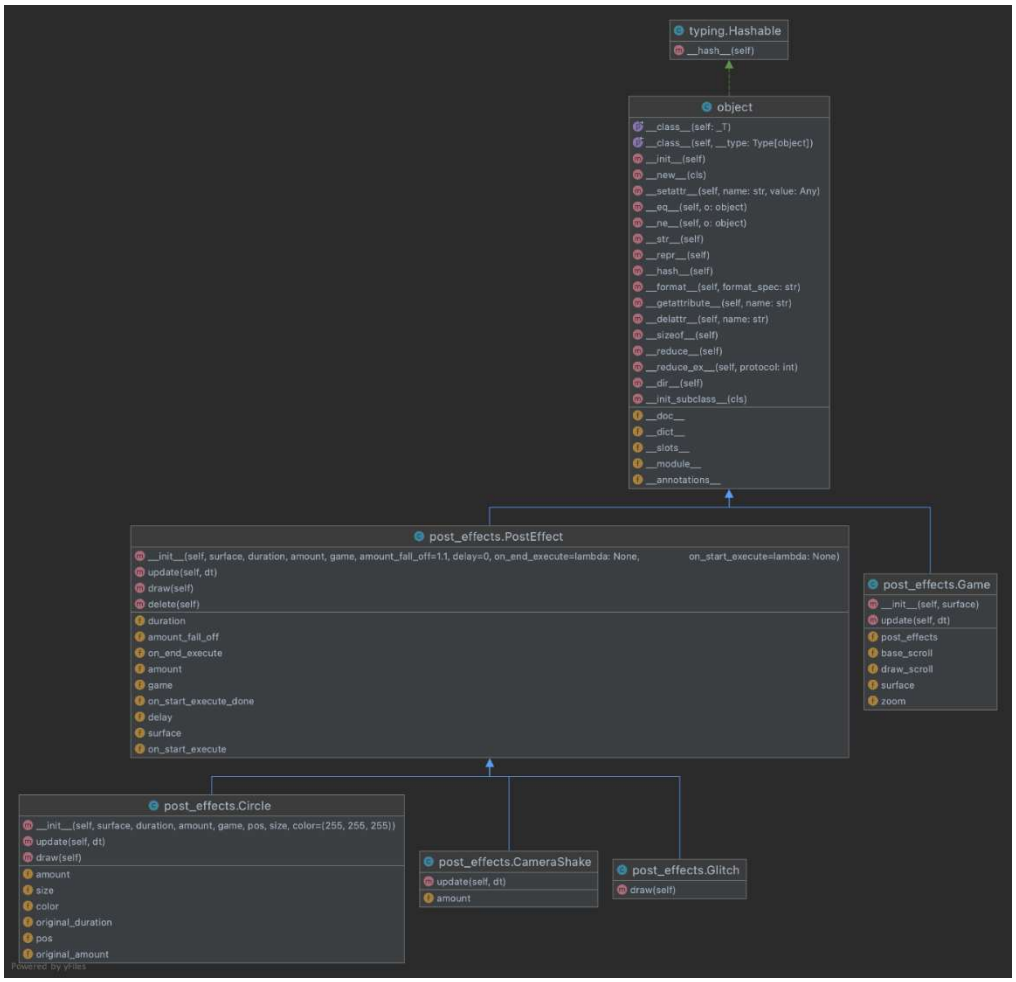
map_generation.py



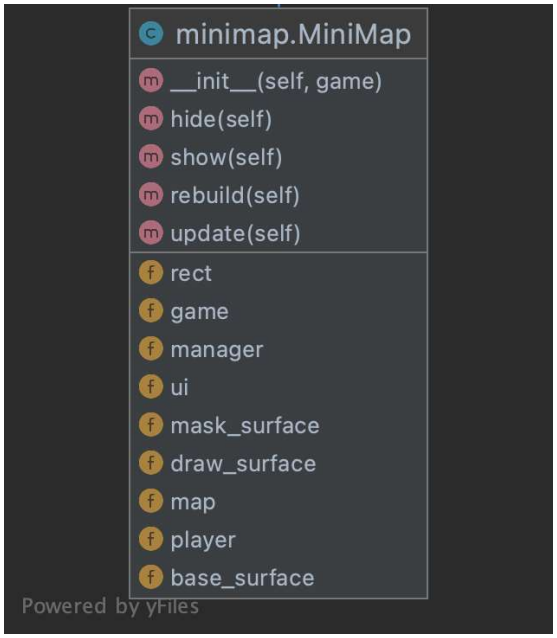


Post_effect.py

- Contient les classes qui font les effets spéciaux



minimap.py



APPADOO APOORVA SRINIVAS

Skills.py

Contient tous les skills

Get_keyboard_layout.py et input_dictionary.py trouvent les contrôles que l'on doit utiliser

Loading.py charge tous les éléments du jeu

Pathfinding.py contient l'algorithme de pathfinding

Dans maps il y a la génération de map qui marche de la même façon que la version de base mais où y est ajoutée des éléments pour pouvoir la dessiner.

Points

a. Gameplay

i. Une interface graphique :

Classe map dans map_generation

On dessine la map dans map_generation avec le lod correspondant et puis on dessine les ennemis et le joueur dessus

ii. Etages

Dans bodies.GoNextFloor

On passe au prochain étage en utilisant l'élément GoNextFloor qui refait la map du jeu et ajoute un certain nombre au niveau de difficulté du jeu.

iii. Point d'expérience (XP)

Dans bodies.Player

On collecte de l'expérience qui sont des instances de la classe Exp(dans bodies) en tuant des ennemis. L'expérience nous permet d'avoir des meilleures armes dans les coffres.

iv. Inventaire limité

Dans hud.py

La classe Inventory gère l'inventaire du joueur ainsi que le rendering de celui-ci sur l'écran

v. Nuage de visibilité.

Classe game_classe methode : draw_night_mode(self)

On crée un masque vide (une surface noire) puis on dessine dessus des gradients noir et blancs correspondant à la lumière émisse par les objets.

Par la suite on multiplie la surface du jeu par ce masque ce qui donne l'effet de lumière.

vi. Diagonales

Dans bodies.Player dans le __init__

Il y a un dictionnaire qui gère les forces appliqués sur le joueur en fonction des inputs.

b. Actions

i. Magie :

- ZaWarudo : arrête le temps dans le jeu

- InvisibleSkill : rend le joueur invisible
- DoubleDmg : double les dégâts du fusil
- DoubleFireRate : double la vitesse de tir
- DoubleHealthRegen : double la régénération de la vie du joueur

Dans Skills.py

c. Monstres

i. Monstre avec fusil

Dans Bodies.Ennemy

La méthode ai dans la classe Enemy gere l'intelligence du monstre En effet il se rapporche du joueur lorsqu'il est a une certaine distance de celui-ci et tire sur le joueur, Mais quand le joueur ce rapproche trop il a peur et s'enfuit.

ii. Monstre suicidaire

Dans Bodies.EnnemySuicide

L'enemi suicidaire a la même, a la place juste fonce sur le joueur et explose.

iii. Boss

Dans Bodies.Boss