Structure Shoot em up test

Main script du jeu (shoot em up.py):

☼ Class Game

(bibliothèques et modules importées : pygame as pg, os, Settings, Sprites)

**► \_\_init\_\_(self)**

Définit les champs initiaux de l’objet *Game().* C’est-à-dire, touts les paramètres fondamentaux pour faire tourner le software, on y initalise également le GUI et le mixer de pygame. Listes des champs :

* screen : configuration de la fenêtre
* clock : initalisation de l’horloge pour gérer les framerates
* running : booléen valant True si le jeu tourne et False le cas contraire
* font\_name : chargement de la police par défaut

► new(self)

Créer les differents objets qui intéragirons dans le jeu. On réunit tout les sprites dans une même listes, ce qui permettra de les afficher tous en même temps. Cette méthode est ce qui créer une nouvelle partie en appelant la fonction *run(*). Liste des objets crées :

* all\_sprites : listes incluant tout les sprites
* player : objet représentant le joueur
* pannel : objet représentant le panneau de navigation du joueur
* meteor : objet représentant les météores, obstacle pour le joueur
* meteor\_list : liste incluant les sprites des météores afain de simplifier la gestion de collisions

► run(self)

Contient les mécaniques de fonctionnement du software, c’est ce qui permet de l’executer. Listes des champs :

* playing : booléen valant True si le software tourne et False dans le cas contraire
* clock.tick(fps) : définit la frame per seconde utilisée durant tout le jeu (image/s)

► update(self)

► events(self)

► draw(self)

► show\_splashscreen(self)

► show\_endscreen(self)

► menu(self)

► press\_key(self)

► draw\_text(self, text, size, color, x, y)

► fadeout\_text(self,background, text, size, color, x , y, delay)

► fadeout\_img(self,image, x , y, delay)