DOCUMENTATION PROJET PCO PYTHON

BINÔME :

CHARPIAT ARTHUR, LANCIAUX JÉRÔME

Introduction :

Nous avons essayé de répondre au mieux aux exigences demandées pour ce projet, nous avons abordé tous les points de l'étape 2 et un début de l'étape 3.

La version finale du projet est commenté au mieux afin d'y voir plus clair

---

Explications du code et des classes :

La classe SpaceInvaders :

C'est la classe **principale** du jeu, celle qui va porter le canvas et la frame,

Elle contient également la fonction play qui sera appelée à la fin du programme afin de lancer le jeu.

La classe Game :

Celle-ci va construire le frame et gérer tous les éléments relatifs aux différentes classes du jeu.

Elle va utiliser la fonction animation et appeler les fonctions de déplacements des autres classes permettant d'animer en boucle les éléments du canvas, et contient en plus les éléments gérant les différentes façon de terminer la partie.

La fonction start_animation va simplement démarrer l'animation, après avoir été appelée dans par SpaceInvaders.

Cette classe gère également le score et le temps, faute d'avoir pu l'intégrer à une classe externe

---

La classe Score :

Non fonctionnelle, cette classe est censée ajouter les scores et le nom des joueurs dans un fichier score.json, malheureusement elle n'est pas en place

---

La classe Defender :

Cette classe a pour rôle de construire le defender et tous les éléments relatifs à celui-ci

La fonction install_in va permettre de créer le defender et le faire apparaitre sur le canvas.

La fonction move_in permet de limiter ses déplacements à la largeur que nous avons donné au canvas.

La fonction controle est celle qui va permettre à l'utilisateur de déplacer son defender à l'aide des flèches directionnelles Gauche et Droite, mais également de tirer des Bullet grâce à la touche Espace du clavier.

Enfin nous avons la fonction fire, celle-ci est chargée de la gestion des tirs, nous avons notamment fait en sorte que lorsqu'il y a 8 Bullet en même temps sur le canvas, il est impossible d'en tirer plus.

La classe Alien :

Cette classe est responsable de l'affichage d'un Alien. On fera appel à cette classe dans une autre classe (Fleet) afin d'afficher toute une flotte d'Alien.

La fonction install_in d'Alien permet de créer un Alien. Pour cela, nous installons sur le canvas une image. Nous avons récupéré l'image de l'Alien grâce à une méthode de classe.

La fonction move_in va permettre de faire bouger un Alien. Encore une fois, c'est la fonction move_in d'un seul Alien qui va nous permettre de faire bouger toute la flotte d'Alien en même temps, lors d'un appel récursif dans Fleet.

Dans la fonction touched_by, le but est de déterminer si un Alien est mort ou vivant. Pour cela, nous cherchons, dans une variable, s'il existe un élément ayant les mêmes coordonnées que l'Alien, si c'est le cas, cela veut dire qu'une Bullet est rentrée en contact avec l'Alien, il est de ce fait déclaré mort. C'est ce qui va nous permettre par la suite, dans la classe Fleet, de faire disparaître l'Alien du canvas.

La fonction Bullet :

Cette classe est responsable de la création des balles , de leur déplacements et de leur suppression si elles sortent des limites du canvas. Il est possible de modifier leur vitesse et couleur dans l'initialisation.

La fonction install_in est responsable de la création de la balle. Nous avons décidé de la faire ronde. Sa couleur est récupérée depuis sa déclaration dans l'initialisation.

La fonction move_in qui va être responsable de deux choses. Premièrement le déplacement de la balle, nous choisissons la vitesse. Deuxièmement sa suppression. La balle se déplace tant qu'elle est dans les limites du canvas, dès lors qu'elle en sort, celle-ci est supprimée.

La classe Fleet :

Cette classe est responsable de la création d'une flotte d'Aliens, du déplacement de cette flotte et également de la gestions des objets dans le canvas (suppression des bullets et aliens.). Dans l'initialisation de cette classe, nous déterminons le nombre d'Aliens qu'il y aura dans la flotte, l'écart entre chaque Alien ainsi que la vitesse de déplacement.

La fonction install_in est responsable de la création de la flotte d'Alien. Pour cela nous intégrons la fonction créatrice d'un seul Alien, depuis la classe Alien, dans une boucle qui, à partir du nombre d'Alien que nous avons décidé de créer, va afficher une flotte.

La fonction move_in est responsable du déplacement de la flotte. Nous avons fait en sorte qu'elle "rebondisse" contre les bords du canvas, afin qu'elle reste sur le canvas, tout en la faisant descendre à chaque rebond.

Enfin la fonction manage_touched_aliens_by est responsable de l'effacement des bullets et aliens. Contrairement à la fonction move_in

de Bullet, la balle n'est pas supprimée si elle sort du canvas. Ici elle est supprimée si elle touche un Alien(grâce notamment à la fonction touched_by de Alien). Cette fonction supprime également l'Alien au contact de la balle et le cache du canvas.

Possibles améliorations :

L'explosion lors d'une collision entre une Bullet et un Alien, même si étant optionnelle, la réussite de celle-ci aurait amélioré notre programme.

La gestion des scores avec un menu d'accueil, avec un High Score et un nom de joueur dans un fichier json pour chaque partie jouée.

Ainsi que la possibilité de recommencer une partie par un bouton en dessous de l'image annonçant que l'utilisateur a gagné la partie.

Plusieurs vies disponible, ainsi que des défenses pour le joueur, avec des aliens tirant des projectiles

Des obstacles entre les Aliens et le Defenders qui permettraient à la fois au Defender de se cacher lors des tirs des Aliens, et en même temps, rajouterait une difficulté pour le Defender à toucher les Aliens.

Apparition de bonus qui descendrait plus vite que les Aliens et qui permettrait au Defender d'avoir plus de balles pendant une courte période.

Conclusion :

Tout au long de ce projet, nous avons pu améliorer nos connaissances en python, malgré une grande difficulté à commencer. Les classes et leurs fonctions sont devenues beaucoup plus claires, et nous avons trouvé un double intérêt à ce projet: premièrement, approfondir nos connaissances, et deuxièmement, réviser le partiel à venir. Il aurait été possible de pousser le projet plus loin mais le temps nous a limités.

Nous pouvons imaginer une possible intégration de l'IA, en ajoutant par exemple un apprentissage des aliens, avec une possible prédiction des coups, les aliens pourraient se déplacer d'une manière contrôlé par une IA, augmentant la difficulté pour le joueur et rendant le jeu plus dynamique et moins redondant