

Documentation Space Invader

Edited by: Codron Julien

Date: 10/12/2018

Pour lancer le jeu, utiliser votre compresseur (il est conseillé d'utiliser IDLE 3) avec la bibliothèque pygame pr  -installer (version 1.9.3 utilis   dans le programme) et lancer le fichier main.py.

Le but du jeu est simplement de d  truire la vague de vaisseaux ennemis gr  ce au missile que vous tirez avec le v  tre situ   en bas de l'  cran, les ennemis peuvent   galement tirer et si vous   tes touch  , vous mourez. Vous avez 3 vie pour d  truire tous les vaisseaux ennemis.

Control : AU debut dans le menu du jeu appuyez sur Enter pour lancer la partie.

- D  placement, - Fl  che droite pour aller    droite,
 - Fl  che de gauche pour aller    gauche.
- Tir : - Barre espace
- Quitt   : - Escape (Esp)

Module Main :

Ce module ne contient que la fonction main et utilise tous les modules pour lancer le jeu

- Contient la fonction main(), qui une fois appel  e cr  e une fen  tre pygame, lance une musique et utilise toutes les autres fonctions pr  sentes dans les autres modules pour lancer le jeu.

Module couleur.py :

- Contient simplement des variables qui ont pour nom des couleurs qui contiennent leurs codes RGB afin d'  viter d'entrer un code RGB lorsque cela est n  cessaire mais simplement   crire la couleur exemple : BLANC = (255,255,255)

Module Menu.py :

Ce module contient 2 fonctions qui sont utilis  es dans le main :

- La fonction execut_menu (surface (fen  tre pygame)) qui lance le menu du jeu et demande au joueur s'il veut continuer en g  rant les inputs de celui-ci.
- La fonction dessine_menu (surface (fen  tre pygame)) qui dessine le menu avec Tkinter qui est lanc   dans la premi  re.

Module scénario.py :

Ce module contient les niveaux, les classes des différents éléments, tu jeux, la gestion des déplacements, la mise à jour de la vue lors d'une action

- La fonction init (surface (fenêtre pygame), niveau (int)) qui prend comme argument une fenêtre pygame, et un niveau, il initialise celui-ci en renvoyant un dictionnaire exploitable par la fonction exécute.
- La fonction exécute (surface (fenêtre pygame), niveau (dict)) qui prend comme argument une fenêtre pygame et un dictionnaire de niveau initialisé par la fonction précédente. Elle permet également de dessiner le niveau, et également de gérée des évènements sur pygame. C'est également elle qui met à jour le jeu et la fenêtre.
- La fonction hud (surface, vie(int) , bullets(int),score(int)), qui permet d'afficher l'HUD du jeux, c'est-à-dire le nombre de vie, le score et le nombre de balle disponible.
- Class Vso(self,surface,posx,posy,who): qui est un objet représentant un vaisseau, il peut être allié ou ennemi, et il a une position
 - Méthode update(self) : dessine le vaisseau par rapport à sa position
 - Méthode replace (self, i (int)) : fonction qui permet le déplacement des ennemis
 - Méthode position(self) : Change l'image du personnage à chaque déplacement pour créer une petite animation
 - Méthode superpose(self,x(in),y(int)) : Retourne un booléen qui indique si un vaisseau est superposé à un autre élément en qui est en position x-y
 - Méthode tir(self) : fonction qui créer un tir d'un ennemi selon sa position et le retourne.
- Class Vsojoueur(self,surface,posx,posy,who): qui est un objet représentant un vaisseau, il peut être allié ou ennemi, et il a une position
 - Méthode update(self) : dessine le vaisseau par rapport à sa position
 - Méthode gauche(self) et droite(self) : met à jour la position du vaisseau
 - Méthode replace (self, i (int)) : fonction qui permet le déplacement des ennemis
 - Méthode tir(self) : fonction qui créer un tir du joueur selon sa position et le retourne.

- Class Wall(self,surface,posx,posy,etat='0') : qui est un objet représentant un mur, il a un nombre de point de vie, une position et peut être détruit par des tirs ennemi
 - Méthode update(self) : dessine le vaisseau par rapport à sa position

- Class Tir(self,,surface,posx,posy,who): qui est un objet représentant un tir de l'un des vaisseaux, 'surface' et la surface de pygame ou dessiner, 'posx' et 'posy' les positions ou le dessiner et 'who' est une chaîne de caractère désignant qui tire.
 - Méthode replace(self) fonction qui permet le déplacement des tirs
 - Méthode update(self) méthode qui redessine le tir
 - Méthode surperpose_wall (self,x,y) qui retourne un boolean, True si un tir est à la même position qu'un objet Wall
 - Méthode surperpose_enemies (self,x,y) qui retourne un boolean, True si un tir est à la même position qu'un objet ennemi
 - Méthode surperpose_joueur (self,x,y) qui retourne un boolean, True si un tir est à la même position qu'un objet Wall Joueur

Module end :

Ce module contient 3 fonctions qui gère l'affichage de fin de partie.

- Fonction dessine_go(surface,score) qui dessine l'écran de fin lors ce que le joueur à perdu.
- Fonction dessine_win(surface,score) qui dessine l'écran de fin lors ce que le joueur à gagner.
- Fonction execut_end (surface,WorL,score) fonction qui exécute l'une des deux autre selon la victoire du joueur 'W' ou la défaite du joueur 'L' et qui gère la gestion d'évènement de pygame pour savoir si le joueur veut rejouer ou quitté.