Sema ERDOGAN
Mathilde MERLANGE
Marine BEGOUEN-DEMEAUX
Gr 18

COLOR VALLEY

RAPPORT

Année : 2017-2018

Tout d’abord le but de ce projet consiste à programmer une version minimaliste du jeu Color Valley . Après avoir bien lu le sujet et bien compris le fonctionnement du jeu, nous nous sommes mises d’accord sur le langage de programmation et nous avons choisi “python 3” et "pygame" comme bibliothèque pour créer les interfaces graphiques.

Après avoir mis en place l’architecture “mvc”, nous avons codé dans le contrôleur une fonction main() qui fait appel à la fonction menu().

La fonction menu() : nous affiche une fenêtre avec le titre du jeu et un bouton play qui permet de lancer le jeu (appel de la fonction jeu() (ainsi que les events).

La fonction jeu() : nous initialise le score, le défilement, les positions des obstacles, et de la balle ainsi que tous les événements de la souris ou du clavier. On fait appel au modèle pour les rotations des obstacles, pour les collisions entre la balle et les obstacles et pour la gestion du score.

Finalement, s'il y a collision la fonction fait appel à game_over(score) avec en paramètre le score obtenu.

La fonction game_over(score) : nous affiche un game Over et un bouton replay qui nous lance une nouvelle partie. Et nous affiche le score final.

Vue :

Dans la vue se trouve l’affichage de chaque obstacle, de la balle, d’un cercle qui permet de changer la couleur de la balle, le bouton replay, le bouton play.

Model :

Dans le model se trouve :

Une fonction qui se charge de la gravité de la balle, le changement de couleur de la balle, et une fonction qui retourne une couleur au hasard.

Les fonctions couleur_arc_bas(i), couleur_arc_haut(i), couleur_carre_bas(i), couleur_carre_haut(i) nous renvoient la couleur de l’obstacle à la position de la balle. Tandis que la fonction collision nous renvoie un booléen si il y a collision entre deux objets (rect). Ainsi les fonctions précédemment citées sont utilisées dans d'autres fonctions.

coll_balle_obst_bas(balle,rect_obst,coul_obst,colorball), et coll_balle_obst_haut(balle,rect_obst,coul_obst,colorball) qui nous retournent vrai s'il y a collision entre l’obstacle et la balle et que la couleur de la balle est différente de celle de l’obstacle sinon retournent faux.

Les fonctions rotate_carre(i) et rotate_cercle(i) permettent d’incrémenter i pour faire la rotation des obstacles et sont aussi utilisées pour retourner les couleurs des obstacles à une position y donnée.

Difficultés rencontrés :

Les difficultés rencontrées durant ce projet étaient tout d’abord de découvrir l’interface pygame, une interface qui ne nous était pas familière, mais la plus grande difficulté était de faire les collisions entre la balle et les obstacles.