

Pinball

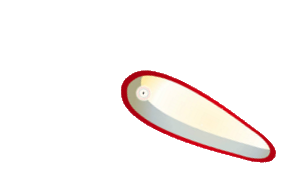
Antonin GUYARD

Pour notre projet final d’ISN (Informatique et Sciences du Numérique), Léo Morin et moi-même, avons décidé de tenter de créer un flipper original sur l’interface de pygame (EduPython).

Logiciels utilisés :

-EduPython (création de l’interface)

-Paint.net (gestion et modification des images)

Objectifs :

-Créer un flipper contenant :

-deux flippers

Q:\Espace d'échange\Projets ISN\pinball\voltball.png

-une balle



-des bumpers

-Assurer les rotations des 2 flippers

-Gérer les collisions de la balle sur les différents objets

-Afficher les scores des joueurs

Répartition des tâches :

* Léo : Gestion des musiques et thèmes nécessaires au flipper

Gestion de l’enregistrement des scores

* Antonin : Mise en place du flipper

Assurer la rotation des flipper

Gérer les collisions de la balle

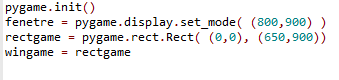
Réalisation

Partie commune :

On commence par importer les différents modules qui nous serviront a créer notre interface



On lance pygame, on  définit la fenêtre de l’interface, puis celle du jeu (en effet on sépare l’affichage du score)



On mesure le temps écoulé depuis le lancement, puis on lance la musique (elle se lance au moment où on ouvre la fenêtre



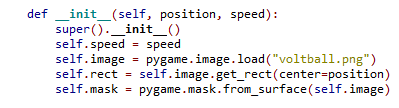
On charge l’image du bumper



Partie Personnelle :

On crée la classe Ball



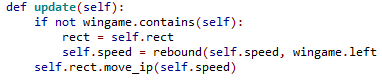


On définit l’Etat initial de la balle



On dessine la balle dans la fenêtre

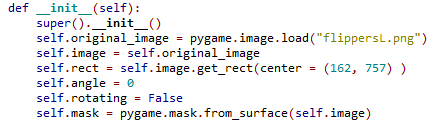
On crée l’objet Ball



A chaque frame les classes s’actualisent. Donc on définit quels paramètres doivent s’actualiser pour la balle

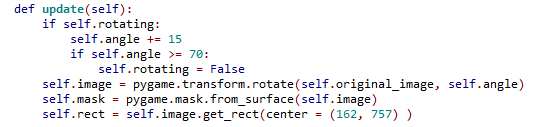
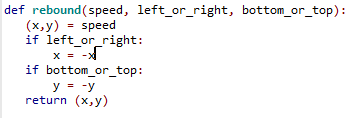
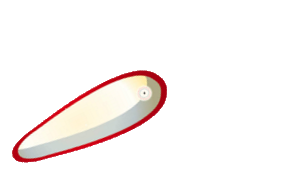
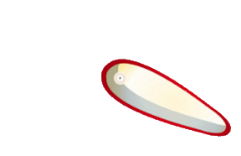


On crée la classe FlipperL



On définit l’état initial

On définit l’update et la fonction rotating pour le flipper



On crée les 2 flippers

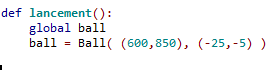
On définit rebound qui gère les collisions sur les bords de la fenêtre

On crée la classe FlipperR

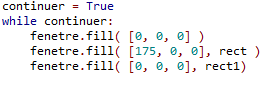


Que l’on place dans un groupe nommé flippers

On crée la fonction qui gère le lancement de la balle dans le jeu du flipper

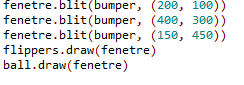


Lancement de la boucle principale



On affiche la fenêtre

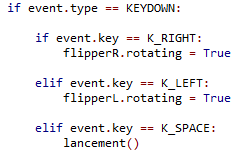
On affiche les différents éléments (flippers + fenêtre + bumper)





On ouvre la boucle gérant les évènements (ici celui de quitter Pygame en fermant la fenêtre)



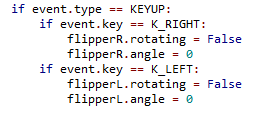


Si on presse la touche :

Fleche droite : on lance la fonction rotating du flipperR

Fleche gauche : on lance la fonction rotating du flipperL

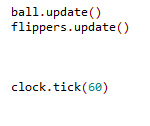
Esapce : on éxécute la fonction lancement



Si on relâche la touche :

Fleche droite : on remet flipperR à 0°

Fleche gauche : on remet flipperL à 0°



On lance les fonctions update de ball et flippers qui sont définies dans leurs classes.

Puis on ajoute une temporisation

On quitte pygame proprement



Améliorations Futures:

Pour l’instant, le programme ne gère pas assez bien les collisions, nous comptons améliorer cela prochainement