Synthèse Projet ISN

Introduction

Après de nombreuses réflexions sur le choix de notre projet, nous avons décidé de réaliser un jeu. Le but de ce jeu est de déplacer un personnage afin qu'il évite des bombes qui tombent en continu sur lui. Pour réaliser ce projet nous avons du nous séparer les tâches. La séparation n'a pas été simple, en effet nous pensions qu'il n'était possible de séparer notre en deux parties, le déplacement du joueur et la tombée des bombes, alors que nous étions 3. Finalement nous avons choisi de rajouter une partie, le score et la collision entre les différents blocs. Je me suis occupé durant le projet du déplacement du joueur.

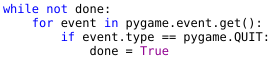
Pour programmer notre jeu, nous avons choisi d'utiliser Pygame, qui utilise le langage de programmation Python. Nous avons du apprendre à utiliser cet outil de programmation et compléter l'enseignement de toute cette année en ISN.

I - Création de la fenêtre d'affichage du jeu

Avant de programmer le jeu il a fallu créer la fenêtre d'affichage du jeu avec Pygame. Pour cela on défini sur Python les paramètres de la fenêtre, la longueur et la largeur. Tout d'abord nous avions choisi d'utiliser une fenêtre de 600 x 400 pixels avec un fond blanc. Pour l'afficher on rentre dans deux variables la valeur de la largeur et de la longueur de la fenêtre.

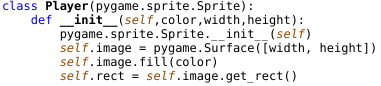
 Mais pour rendre notre jeu plus agréable graphiquement nous avons choisi de remplacer notre fond blanc par une image , un ciel bleu avec des nuages téléchargé sur Internet. Pour cela on utilise on fonctionnalité propre à Pygame qui charge l'image que l'on souhaite en lui indiquant le nom et le chemin d'accès. Nous avons par conséquent modifier les paramètres de la fenêtre afin de l'adapter à la taille de l'image (640 x 438 pixels).

I:\ISN\projet\background.pngI:\ISN\projet\parafenetre.png

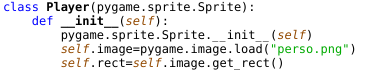
 Désormais, le programme affiche la fenêtre mais n'est pas capable de la fermer. Pygame fonctionne avec des évènements. Pour fermer la fenêtre, il faut effectuer un test qui vérifie si l'utilisateur a oui ou non appuyé "Alt+F4" ou sur la croix en haut à droite de la fenêtre. Si oui, la fenêtre se ferme.

Maintenant que la fenêtre est créée, il ne reste plus qu'à programmer le joueur.

II - Création du joueur

 Afin d'afficher le personnage sur la fenêtre d'affichage Pygame, j'ai du utilisé de la programmation objet, par l'intermédiaire de sprite. Les sprites sont des entités graphiques de Pygame pour lesquelles on peut ajouter différentes propriétés en créant une classe pour chaque type de sprite. J'ai tout d'abord créé une classe "Player", ayant comme paramètres d'afficher un bloc rouge (15 pixels x 20 pixels) à une position prédéfinie. J'ai choisi d'afficher le sprite en bas de la fenêtre, et j'ai par conséquent entré des valeurs pertinentes pour les positions x et y du sprite.

I:\ISN\projet\blockrouge.png J'ai ensuite rajouter à mon programme la fonction des évènements claviers (voir Partie 3) pour contrôler le sprite. Mon programme fait désormais apparaitre un bloc rouge que l'utilisateur peut déplacer.

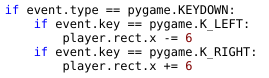
I:\ISN\projet\perso.png Mais pour rendre plus fini le jeu j'ai décidé de remplacer le bloc rouge par une image. J'ai donc télécharger sur Internet une image d'un personnage d'une taille réduite (environ 35 pixels x 20 pixels). L'image possédant un fond blanc, j'ai remplacé ce fond par un fond transparent grâce à l'éditeur d'images GIMP pour un meilleur rendu graphique du jeu.

Afin d'afficher cette image sur le jeu, j'ai du modifié la classe "Player". J'ai retiré tous les paramètres concernant la taille et la couleur du sprite devenus inutiles du fait de l'ajout de l'image et ajouté un nouveau paramètre qui récupère l'image dans un répertoire et permet de l'afficher.

I:\ISN\projet\boucleforplayer.png J'ai du aussi modifier la position de l'image sur la fenêtre puisque la taille de celle-ci est différente de celle du bloc rouge.

III - Evènements claviers

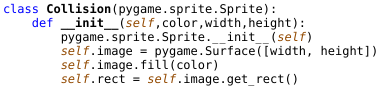
Pour déplacer le personnage, Pygame utilise des événements claviers. Ces évènements attribuent une action à chaque pression sur une touche du clavier ou lors d'un mouvement de la souris. Ne sachant pas comment cette commande fonctionne je me suis renseigné sur Internet afin de découvrir les fonctionnalités et la bonne syntaxe à utiliser.

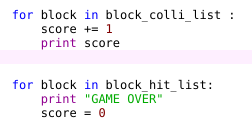
 J'ai commencé par effectuer des tests, ayant pour but de déplacer un rectangle rouge sur une fenêtre d'affichage. Mon objectif était de déplacer le rectangle rouge de tant de pixels sur la gauche en appuyant sur la flèche gauche du clavier et le déplacer d'autant de pixels vers la droite en appuyant sur la flèche de droite.

I:\ISN\projet\ifplayerx.png Après ces tests, je me suis aperçu d'un problème. En effet le personnage se déplaçait indéfiniment sur l'axe x et sortait par conséquent de l'écran. J'ai donc rajouter une condition pour le déplacement du joueur, si la position sur x est inférieure à 2 alors le personnage est envoyé à la position 434 sur x et si la position du joueur est supérieure à 435 alors le personnage est envoyé à la position 12 sur x.

IV - Collisions entre les différents sprites

A l'origine, je ne devais pas m'occuper de cette partie mais nous avons accumulé un léger retard, alors j'ai essayé d'apporter mon aide. Le but était d'insérer un score dans le jeu.

I:\ISN\projet\listcolli.png Pour cela j'ai choisi de créer une autre classe de sprite "Collision" qui fait apparaitre une zone de la largeur de la fenêtre sous le joueur ( non visible sur la fenêtre d'affichage). Dès qu'une des bombes touche cette zone le score augmente d'un point. La collision s'effectue grâce à des listes qui comprennent les différents sprites ainsi qu'une fonction propre à Pygame qui permet la collision entre plusieurs sprites.



Le programme fonctionne, lorsque les bombes touchent le joueur, la console affiche et "GAME OVER" et le score se réinitialise. Et dès qu'une bombe touche la zone de score, on a score = score + 1.

Conclusion

Le projet d'ISN m'a permis d'améliorer mes connaissances en programmation, notamment sur Python, et de prendre conscience que ce n'est pas aussi simple de concevoir un jeu qui visuellement ne parait pas complexe à mettre en œuvre. J'ai aussi appris à mener à bien en projet de groupe et à composer avec les avancements de chacun sur leur propre partie. La difficulté du projet a été d'harmoniser notre travail pour en faire un seul et unique projet. Chaque personne n'avançant pas à la même vitesse pour effectuer sa tâche, nous nous sommes parfois retrouvé bloqué et contraint d'attendre que l'un d'entre nous termine sa partie pour pouvoir continuer. Je suis plutôt satisfait du résultat final de notre travail, même si le projet n'est pas aussi abouti qu'espéré, en effet le score ne s'inscrit pas sur la fenêtre d'affichage mais sur la console du terminal de commande avec lequel nous avons réalisé ce projet.

ANNEXE



