

PROJET 3 : AIDEZ MAC GYVER A S'ECHAPPER

Introduction

Ayant débuté le parcours de Développeur d'application Python sans avoir la moindre expérience dans le domaine du développement, le projet qui doit être réalisé me paraît semblable à l'ascension de l'Everest, autant dire que j'ai l'impression que le ciel me tombe sur la tête.

Après avoir validé les cours liés au projet je ne sais toujours pas comment organiser ce programme, j'entreprends de suivre des cours sur d'autres supports (sites web) et d'effectuer des exercices sur « France IOI » pour augmenter ma compréhension du développement, des algorithmes et plus particulièrement du langage Python.

Après avoir changé de mentor accompagnateur nous entreprenons de reprendre l'initiation au développement en Python à son début tout en jetant les bases de notre programme

I. Démarrage du projet

Ne sachant pas par où commencer pour l'élaboration de ce jeu vidéo en 2D je décide de démarrer par la création du labyrinthe qui sera la base sur laquelle viendront se greffer les autres éléments qui seront régis par des boucles, des fonctions et des conditions, car à vrai dire en ces temps troubles c'était la vision que j'avais du développement en Python.

Le premier jet sera donc un fichier texte contenant des chaînes de caractères pour illustrer les murs du labyrinthe ainsi que nos deux protagonistes, viens ensuite la première grande réflexion, comment formater ou plutôt « parser » cette structure composée de caractères grâce à une fonction qui renverra une liste exploitable pour mettre en place la première interface graphique du programme, la création de mon premier algorithme.

Le début de mon programme se met en place et je décide de prendre un virage relativement serré, quitter Windows 10 et Pycharm pour m'initier à Linux et l'éditeur de texte Atom, chose qui n'a pas été aisée dans la mesure où j'étais bien plus à l'aise avec une souris qu'avec un terminal, mais il faut avouer que le choix était judicieux pour travailler sur ce projet.

II. Mise en place de la structure principale

Avant de créer la boucle principale de mon programme je découvre la bibliothèque multimédia Pygame, je me rends vite compte que la plupart des outils dont j'aurais besoin pour l'élaboration de mon programme sont présents dans cette bibliothèque et que la plupart de mes interrogations ont à présent trouvées leur réponse, je peux donc poser les fondations de mon programme.

Affichage de l'interface utilisateur

La première étape consistait à afficher la fenêtre du jeu et les différents éléments (ou sprites) qui la compose, leur affichage est géré grâce aux commandes du module Pygame, qui fait appel à un algorithme pour reconnaître les emplacements des murs, personnages et espaces vides du labyrinthe représentés à la base par des chaînes de caractères qui seront ajoutés à une liste qui sera parsée par notre algorithme.

Affichage des éléments de l'inventaire

Pour afficher les trois éléments nécessaires à la validation de la sortie de notre héros dans l'inventaire en bas de l'écran, une méthode a été créée, méthode qui fait appel à la commande « `get_rect` » du module Pygame pour transformer nos images en rectangles et les positionner.

Déplacement et gestion des collisions

Le déplacement du personnage et les collisions avec les objets qu'il doit ramasser ainsi qu'avec le garde est également géré grâce aux commandes du module Pygame, gestion des événements (`EVENT` & `EVENT_KEY`) conditions et modifications de variables si ces conditions sont remplies et « `Collidect` » (collisions entre deux rectangles).

Conditions de victoire ou d'échec

Idem que pour le déplacement et la gestion des collisions, des conditions ont été établies et sont validées par les valeurs booléennes `True` et `False`, si les trois conditions sont validées par la valeur Booléenne `True` en cas de collision entre les sprites des deux personnages principaux du jeu (Mc Gyver & Murdock) l'écran de succès s'affiche, dans le cas contraire l'écran de défaite s'affiche.

Une musique a été ajoutée, si vous n'avez pas été un grand fan de la série télévisée Mac Gyver, rien ne vous empêche de mettre le volume à zéro sur votre système audio ou de modifier la ligne de code

`self.pygame.mixer.music.set_volume(0.6)` en mettant le volume (0.0) dans les paramètres

cette ligne se trouve dans la méthode « `def init_music` » de la classe `Gui` du fichier `gui.py`