

Projet 3 : Aidez MacGyver à s'échapper

Lien Github : <https://github.com/Kuru-chan/MacGyver>

I - Principe de l'exercice

Nous devons créer un labyrinthe en 2D, dans lequel se trouvent MacGyver, trois objets et un gardien. MacGyver doit se déplacer à l'aide des touches directionnelles afin de récupérer les trois objets afin de pouvoir sortir. Si il n'a pas récupéré les trois objets, il ne parviendra pas à passer le gardien.

II - Explications des programmes

- Le fichier "stage1"

Ce fichier correspond à la structure du labyrinthe. Il est composé de "0", "m", "a" et "d". Ces lettres correspondent à une partie de la structure :

le "d" est l'emplacement de MacGyver, le point de départ

le "a" est l'emplacement du gardien, le point d'arrivée

le "m" est un mur, infranchissable

le "0" est un couloir dans lequel MacGyver peut se déplacer

- Le fichier "constantes.py"

Dans ce fichier sont contenues les différentes constantes nécessaires au bon fonctionnement du jeu. Cela va donc des images des personnages aux dimensions des objets en pixels en passant par les trois objets à récupérer.

- Le fichier "classes.py"

Dans ce fichier sont contenues deux classes : la classe "Maze" et la classe "Character"

La classe "Maze" est la classe qui s'occupe du labyrinthe, de la génération à l'affichage. Au niveau de la génération, on prend le fichier "stage1", on enregistre la structure des différentes lignes dans une liste et on place aléatoirement les trois objets. La deuxième partie de la classe va quand à elle, se charger de replacer les éléments de la liste par les éléments qui vont être affichés à l'écran. Cette méthode va donc afficher les murs, MacGyver, le gardien et les trois objets à récupérer.

La classe "Character" me permet de définir les différentes interactions que va effectuer MacGyver. Il y a donc une partie qui s'occupe de définir comment fonctionnent ses déplacements en fonction du choix de l'utilisateur et une autre partie qui s'occupe de ses interactions avec les trois objets qu'il doit récupérer. Dans cette dernière, lorsque MacGyver ramasse un objet on incrémente un compteur et on remplace la case sur laquelle était l'objet par une case de type "couloir".

- Le fichier “main.py”

Ce fichier est l'élément central du jeu dans lequel on fait appel aux différentes classes que nous avons créées auparavant. On importe donc PyGame, élément essentiel qui nous permet de créer la fenêtre de jeu, d'importer nos images, et qui s'occupe de gérer tous les déplacements avec les touches directionnelles.

On initialise PyGame et on définit toutes les variables nécessaires au bon fonctionnement du jeu. Le programme est composé de trois boucles :

La première, la boucle “MAIN” est la boucle principale qui sert à faire fonctionner le programme. Tant qu'elle est active, le programme reste fonctionnel.

La deuxième, la boucle “LOOP” est la boucle qui affiche l'écran d'accueil avec les règles du jeu. Tant que l'utilisateur n'appuie pas sur la touche “Entrée”, elle reste affichée à l'écran. Si l'utilisateur appuie sur “Échap”, il ferme le programme. Si l'utilisateur appuie sur la touche “Entrée”, la boucle se ferme et le programme charge le labyrinthe et les personnages puis passe à la dernière boucle.

La troisième boucle, la boucle “GAME” est la boucle dans laquelle se déroule le jeu.

L'utilisateur va pouvoir contrôler MacGyver à l'aide des touches directionnelles.

Si l'utilisateur passe sur une case avec un objet, il remplacera la case par une case vide et ajoute “1” au nombre d'objets trouvés. Si le joueur se trouve sur la case du gardien et à en sa possession la seringue (obtenue après avoir ramassé les trois objets), il gagne et une fenêtre affichant sa victoire s'affiche. Si il lui manque un ou des objets, il perd et une fenêtre affichant les objets manquants s'affiche.

III - Difficultés rencontrées

Ma plus grande difficulté à été de me motiver après avoir dû recommencer mon travail plusieurs fois à cause d'éléments extérieurs. J'ai aussi dû faire quelques recherches afin de pouvoir placer les objets aléatoires correctement. Sinon en soi l'exercice ne m'a pas semblé si dur, j'avais déjà faire le labyrinthe DK donc je savais à quoi m'attendre. Mettre le code aux normes PEP8 était embêtant, c'est une convention de nommage qui m'était inconnue et totalement différente de ce que je faisais habituellement, surtout dans le fait de limiter le nombre de caractères par ligne.