PROJET 3 : Aidez Mac Gyver à s'échapper

Lien Github : **https://github.com/Tepau/MacGyver**

Principes de l'exercice

Créer un labyrinthe 2D dans lequel Mac Gyver est enfermé. La sortie du labyrinthe est protégée par un garde. Mac Gyver, qui se déplace de case en case à l'aide des touches directionnelles du clavier, doit trouver dans le labyrinthe 3 objets placés aléatoirement en se plaçant dessus, qui lui permettront de confectionner une seringue avec laquelle il pourra endormir le garde et donc sortir du labyrinthe.
Si il n'a pas ramassé les 3 objets et qu'il se présente devant le garde, la sortie lui est refusée et le jeu est perdu.

Algorithme

Fichier « n1 » :
Dans ce fichier, je définis la structure de mon labyrinthe grâce à des caractères.
La lettre « d » correspond au point de départ de mon labyrinthe. La lettre « a » correspond à l'arrivée. La lettre « d » correspond au départ. Le chiffre « 0 » correspond aux chemins que peut emprunter Mac Gyver.

Fichier « constantes.py » :
Dans ce fichier, on trouve les données nécessaires au bon fonctionnement des fichiers labyrinthe.py et classes.py, notamment la taille de la fenêtre de jeu, en pixel la taille de la bannière en pixel, le nombre de case de chaque coté du labyrinthe. J'y définis également des lettres qui représenteront les objets que Mac Gyver doit trouver dans le labyrinthe. Enfin je crée des variables qui indiquent le chemin des images nécessaire au jeu.

Fichier « classes.py » : classes nécessaire au bon déroulement du jeu

- **Maze**

Cette classe permet de générer et afficher la structure du labyrinthe. Elle possède deux méthodes. Une première qui ouvre le fichier « n1 », enregistre sa structure dans une liste, et permet de générer les trois objets de façon aléatoire. Ces objets sont représentés par des lettres définies dans le fichier constantes.py. Ils sont générés à partir de la liste de structure du labyrinthe. Elle est enregistrée dans l'attribut « structure_map ». Une seconde qui permet d'afficher le contenu du labyrinthe à l'écran. Elle va remplacer les éléments de la liste de structure par leurs images correspondantes. Elle permet également de définir des coordonnées aux éléments en fonction de leur ordre d'apparition dans la liste « structure_map ». Cette méthode affiche donc les murs, le gardien, et les trois objets dans la fenêtre de jeu.

- **Character**

Cette classe permet de définir les mouvements que doit faire Mac Gyver, tenir un inventaire des objets ramassés, et d'afficher cet inventaire. Elle possède 3 méthodes qui vont permettre de définir les déplacements que MacGyver peut faire en fonction des choix de l'utilisateur et de la structure du labyrinthe. De remplacer l'image d'un objet par celle du chemin une fois que Mac Gyver est passé sur l'objet. D'incrémenter l'inventaire de Mac Gyver lorsqu'il ramasse un objet. D"afficher l'inventaire de Mac Gyver.

Fichier labyrinthe.py :

J'importe la bibliothèque Pygame, permettant l'affichage de la fenêtre, des images et le déplacement à l'aide des touches directionnelles du clavier.J'importe aussi les modules « constantes.py » et « classes.py ». J'initialise Pygame, défini la taille de la fenêtre ainsi que les variables qui permettront l'affichage des images du labyrinthe. Je défini 3 booléens qui serviront à savoir quel objet a été attrapé et quel objet ne l'a pas été.
Mon fichier est composé de 3 boucles :
MAIN_GAME : Boucle principale : c'est la boucle du programme, tant qu'elle tourne, le programme est en marche.
CONTINUE_HOME : Boucle d'accueil : tant qu'elle tourne, l'écran d'accueil est visible.Quand l'utilisateur passe l'accueil en appuyant sur entrée, on ferme la boucle d'accueil.On charge le labyrinthe et le personnage à l'aide des classes.
CONTINUE_GAME : Tant qu'elle tourne, l'utilisateur joue et déplace Mac Gyver. Pour ramasser les objets, le programme va comparer la structure du labyrinthe avec la position de MacGyver. Si le déplacement se fait sur un objet, il appellera les méthodes erase et la valeur du booléen définissant si l'objet est attrapé devient True. Enfin si le joueur possède les 3 objets et se déplace sur le gardien alors il gagne, le programme s'arrête et j'affiche un écran qui lui signifie sa victoire sinon un écran lui signifie sa défaite et l'objet ou les objets qu'il n'a pas trouvé grâce aux booléens.

Difficultés rencontrées

Ma plus grande difficulté a été de comprendre le fonctionnement des classes, j'ai mis pris pas mal de temps à assimiler le concept et à pouvoir l'utiliser. J'ai repris les mêmes cours plusieurs fois.
Placer les objets de façon aléatoire a nécessité beaucoup de recherches sur internet et de tests.
J'ai aussi eu des difficultés à placer un bandeau en haut de la fenêtre et à y déplacer les images. J'ai testé plusieurs méthodes avant de trouver la bonne. Je me suis aidé de l'exercice « DK labyrinthe », et j'ai effectué beaucoup de recherches sur internet. Une autre difficulté a été de bien comprendre l'anglais notamment sur les sites où la plupart des conseils sont donnés en anglais. Enfin j'ai également eu du mal à faire en sorte que mon code corresponde aux convention PEP8.