Projet 3: aidez McGyver à s'échapper

1: Introduction

réalisation du projet 3 du parcours développeur d'application python,
il est demandé de réaliser un jeux vidéo en utilisant le module pygame,
dans ce dernier le personnage « MacGyver » doit rassembler les éléments défini afin de neutraliser le gardien est finir le niveau,

2: Algorithme

1 : Environnent : J'ai en grande parti rédiger mon code sur notepad++ j'ai utilisé tardivement visual studio code dans l'avancée du projet en grande partie à cause de soucis d'indentation causé par notepad++,

2 : Code du jeu
j'ai découpé mon code en trois fichiers .py,
Un fichier Lab.py qui contient les fonctions principales du jeu qui initialise la fenêtre du jeu et ses éléments graphiques, un fichier position.py ou sont présente l'ensemble des classes, et un fichier constant.py existant par facilité afin d'importer les paramètres de la fenêtre pour le position.py,

3 : Importation des librairies,
Au début de mon fichier Lab.py j'importe la librairie pygame, mon fichier position contenant les classes du code ainsi que le module random,
S'en suit l'initialisation du module Pygame et de la fenêtre de jeu.
La variable « cote » défini la taille de la fenêtre, celle ci fait appel aux variables « number_sprite_inline et sprite_size » contenues dans le constant.py.
Je blit ensuite le fond, ainsi que les murs,
ensuite la génération du niveau et l'affichage des éléments => personnages, objets,

4 : Les classes
Trois classes composent le position.py,
la classe Level, la classe Perso et la classe Items,

La classe Level, elle à pour fonction principale de définir le contenu du labyrinthe, pour ce faire j'ai créée une fonction generate et display, la fonction generate avec la fonction « with open » va lire le fichier « Level,txt » dans lequel se trouve le plan du labyrinthe.
Les vides « , » représente les emplacements libre où le personnage peut circuler, les «# » représente les murs.
Le programme stock sous forme de liste les éléments puisé dans le fichier Level,txt, dans une variable structure.
La fonction display, va remplacer les éléments de la variable structure précédemment créé par leur équivalent visuel, le script va ensuite attribuer des coordonnées X et Y aux éléments en fonction de l'ordre d'apparition des caractères dans la liste « structure » et affiche les images des murs et du gardien dans la fenêtre du jeu,

La classe Perso, c'est la classe qui contient les fonctions utiles à Mac pour se déplacer, les déplacements se font case par case à l'aide des touches fléchées appelé par la fonction pygame,event,get()
c'est une structure conditionnelle en « if »,elle vérifie que les conditions sont remplis avant d'autoriser le mouvement, elle compare les données en x et y (self,case.x, self.case.y) de Mac avec les coordonnées des éléments de level.structure pour voir si les nouvelles coordonnes de Mac correspondent avec une zone libre et non un mur, elle contient également une condition relative à la direction voulue interdisant le déplacement hors de la zone de jeu. >
if self.case_x < (number_sprite_inline – 1).

La classe Items, elle a une fonction qui va répartir les objets au sein du labyrinthe dans les zones libres de level.structure, l'aspect aléatoire est gérer par la fonction random qui génère une valeur entre 0 et 14 pour définir les variables rand_x et rand_y qui seront transformé en coordonnées dans les variables self.x et self.y.

3: Compteur d'objets et fin du jeu

Pour indiquer que Mac ramasse les objets lorsqu'il passe dessus, je compare dans mon ficher Lab.py la position de Mac avec celles des objets répartis dans le labyrinthe, l'affichage des objets dans le labyrinthe est dépendante de booléens,(NotPicked) si Mac marche sur un objet, ce booléens devient False, et la position de l'objet et modifié pour être affiché en haut de l'écran.
Quand Mac atteint la case du gardien, les deux variables booléennes sont comparé,
si les trois objets du jeu son ramassé alors l'état des objets étant en False la variable GAME_WON devient True et l'écran annonce alors la victoire,
dans le cas contraire si les trois conditions ne sont pas remplis alors la variable GAME_LOOSE devient True et annonce la fin négative de la partie.