P3 - MAC GYVER

Lien GitHub : https://github.com/msadour/MacGyver

I – règles du jeu

Le but du jeu est d'aider Mac Gyver à s'évader du labyrinthe. Mais il faut d'abord tuer le gardien (en récupérant les 3 éléments qui sont le tube, l'éther et l'aiguille) qui surveille la porte de sortie. Si Mac Gyver se retrouve devant le garde sans ces 3 éléments, celui ci meurt et la partie est terminée. Autrement le garde s'endort et Mac Gyver pourra empreinte la porte de sortie et le joueur remporte la partie. Une fois terminé, le joueur à la possibilité de recommencer ou de quitter le jeu via un menu.

II – algorithme

J'ai développé l'algorithme selon cette logique :

- Initialisation du labyrinthe selon le fichier labyrinthe.txt ('0' représente les murs, les espace représentant le sol, 'M' correspondant à Mac Gyver, 'G' au garde et 'S' la porte de sortie). Chaque items du fichier a une classe lui permettant d'être instancier afin de le placer sur le labyrinthe;

- Placement des 3 éléments nécessaire à l'endormissement du garde aléatoirement sur le labyrinthe. Dans le parcours du fichier, chaque espace libre est stocké dans une liste nommé «list_free_place_for_element» (sous forme de tuple : (position_horizontale, position_verticale)). On pioche ensuite 3 positions au hasard (on prend 3 éléments au hasard de la liste «list_free_place_for_element» via la méthode choice.random) puis on les affectes à chaque élément pour les placer sur le labyrinthe.

- Démarrage du jeu (dans une boucle «while»). Cette boucle vérifie que la variable «in_game» est a «True» ;

- Écoute de chaque événement : seul les déplacement sont permis, sur chaque déplacement, on regarde si mac gyver se trouve devant rien, le gardien, un des éléments ainsi que la porte. A chaque déplacement, on créer un rectangle blanc sur l'ancienne position de Mac gyver (étant donné que le sol est composé uniquement de rectangle blanc). Cela évite la répétition de l'affichage du personnage sur l'écran lorsqu'il se déplace sur le labyrinthe ;

- Lorsque le jeu se fini (accès à la porte de sortie ou présence devant le garde sans les 3 éléments), la variable «in_game» se met à «False» et la boucle s'arrête;

- Ouverture du menu de fin demandant à l'utilisateur s'il souhaite rejouer ou non. Pour cela j'ai utilisé un écouteur d'événement par rapport à chaque clique de la souris. Si celui ci clique sur la zone où se situe «recommencer», le jeu se relance en appelant la méthode «run» de la classe Main. Si celui si clique sur la zone où il y a «quitter», la boucle se casse sans relancer la méthode «run».

III – difficultés rencontrées

Comprendre le fonctionnement des Sprites (le tutoriel de OPC n'avais pas traité de cela), j'ai donc du chercher des tutoriels en anglais sur le web. Autrement je n'ai pas eu de difficulté à prendre en main «Pygame».

IV – Le jeu