

Macgyver Projet 3

I Mon raisonnement:

Pour réaliser ce jeu j'ai tout d'abord cherché à ouvrir une fenêtre vierge avec Pygame. Une fois cette étape réalisée, j'ai voulu créer un fichier texte que j'ai par la suite généré sous forme d'affichage graphique en lisant chaque lettre une par une .

Jusqu'ici ça ne m'a pas pris beaucoup de temps. Dans mon raisonnement, il me manquait le fait d'afficher les différents éléments de mon labyrinthe et leurs caractéristiques. Avec surtout, les items affichés dans des positions aléatoires. Ainsi, après cela je pouvais arriver à une étape où tout les éléments de mon labyrinthe pouvaient être affichés mais sans avoir tout les principes du jeu. MacGyver pouvait bouger tout en évitant les murs.

Le programme vérifiait chaque case avant de s'y rendre. Ainsi, si c'était un mur le mouvement était nul. Pour en arriver là j'ai effectuée des captures d'événements. MacGyver pouvait passer sur les objets sans pour autant les ramasser et arriver devant le gardien mais rien ne se passait.

Il ne me manquait plus qu'à ramasser les objets dont MacGyver égalerait les coordonnées et terminer la partie sous une victoire, si tout est ramassé ou sous une défaite si rien ou une partie est ramassée.

Pour conclure, la difficulté principale pour moi c'était la gestion de la création du jeu et surtout la gestion pour faire bouger MacGyver tout en indiquant qu'il fallait ramasser les objets et gagner.

N'ayant pas d'expérience dans ce domaine, ce n'était pas facile de penser à l'avance aux boucles, aux conditions, ainsi qu'à l'exécution de certains événement. Par exemple, pour donner de la mobilité au personnage ce n'est pas de simple mouvement mais une gestion de tout erreur à éviter (*exemple: ne pas éviter les murs, ne pas ramasser les objets*) et à penser au rafraîchissement de la fenêtre pour chaque nouvelle affichage.

Je crois donc, qu'il faut apprendre à penser ainsi pour coder en python.

II Mes fichiers:

Avant de me lancer, il fallait que je me facilite la tâche au plus possible, tout en ayant une organisation, ainsi que de la clarté dans mon programme, j'ai donc, en dehors du fichier texte crée trois fichiers Python:

1. macgyver.py

Ce fichier m'a servi pour l'affichage de la fenêtre avec les éléments du jeu, les boucles

et les mouvement de MacGyver sous plusieurs conditions. C'était en quelque sorte la finalité de mon programme.

2. classes.py

L'utilité de celui-ci était simple, elle servait uniquement à la création de classes, j'ai choisi d'en créer cinq:

- classe MacGyver: Définit le MacGyver
- classe Exit: Définit le Gardien
- classe Square: Définit chaque case
- classe Item: Définit les items
- class Coordinates : Stocke les coordonnées des éléments du jeu (MacGyver, Exit, Item, cases)

3. methods.py

Comme son nom l'indique, c'est pour les méthodes, j'en est 4:

- random_coordinates(): elle génère des positions aléatoires et retourne sous forme d'objet Coordinates.
- generate_laby(): génère le fichier texte sous forme de liste de caractères, crée un objet Square par caractère et donne une valeur selon la lettre lue. Elle retourne le labyrinthe sous forme de liste d'objets de type Square.
- display_laby(): Sert à l'affichage du labyrinthe.
- put_item_in_laby(): Donne des coordonnées aléatoires à l'item, vérifie que la case n'est pas remplie. Si la case est occupée, elle relance la méthode avec des nouvelles coordonnées aléatoires.

Précision:

Les classe “MacGyver” et “Exit” sont identiques, pour faciliter l'organisation du programme j'ai opté pour la création de deux classes.