

PROJET 3 :

Aidez MacGyver à s'échapper !

Lien GitHub : https://github.com/Ph4nt00m/MacGyver_P3.git

CADRE DE DEPART :

Après la création d'un repo Git et d'un environnement virtuel pour développer ce projet, j'ai créé la structure du labyrinthe dans un fichier txt en écrivant 15 lignes comprenant chacune 15 caractères de manière à créer un labyrinthe carré. Chaque caractère représentant une cellule du labyrinthe et chaque cellule ayant une hauteur et une largeur égales. Les caractères utilisés sont :

« X » pour les murs, « O » pour les cases vides, « D » pour la case de départ et « A » pour la case d'arrivée. Ces caractères, ainsi que leur position, seront plus tard enregistrés dans des listes.

La seconde étape a été de créer le fichier main.py qui démarrera notre programme et initialiser une fenêtre pygame qui sera notre écran de jeu. Pour cela, il faut créer une boucle qui maintiendra la fenêtre du jeu ouverte en y ajoutant une commande de rafraîchissement de la fenêtre (pygame.display.flip()).

J'ai ensuite créé le fichier maze.py, dans lequel se trouve la classe Maze. Cette classe possède les caractéristiques de notre labyrinthe comme les images des murs et du sol ou encore une liste de listes qui comprendront la position de chaque élément du labyrinthe selon des coordonnées sur un axe abscisses et ordonnées.

Cette classe possède deux méthodes :

- load_structure() qui représentera le labyrinthe contenu dans le fichier structure.txt.
- maze(argument) qui matérialisera le labyrinthe dans notre fenêtre de jeu.

ANIMATION DU PERSONNAGE :

Création du fichier player.py qui contient la classe Player. C'est la classe du personnage de MacGyver.

Elle contient les attributs relatifs au personnage comme par exemple sa position dans le labyrinthe ou la liste d'objets qu'il collecte sur son passage afin de construire la seringue.

Cette classe dispose d'une méthode qui lui permettra de se déplacer (move(argument)).

Dans la boucle main, j'ai défini les touches multidirectionnelles du clavier pour le déplacement du personnage. En créant un dictionnaire qui gardera ces touches dès que l'on appuie dessus, il est possible de se déplacer en maintenant la touche enfoncée.

La liste de la position des murs, permettra à MacGyver de se déplacer partout où les murs ne sont pas présents.

La vitesse de déplacement du personnage correspond à la taille des cases. Il se déplace ainsi de case en case.

Cela permet aussi qu'il se déplace de case en case si la taille des cases était changée.

LES OBJETS ET LE GARDIEN :

J'ai créé un fichier items.py qui contient une classe Items. Cette classe créera les trois objets du jeu (bouteille d'éther, tube et aiguille) ainsi que le gardien. J'ai mis le gardien dans cette classe car ce dernier n'a aucune fonction particulière. Il est donc un objet inerte au même titre que les trois autres objets.

La classe Items possède une méthode fondamentale qui est de donner aux objets une position aléatoire dans le labyrinthe. J'ai d'abord donné une position aléatoire, grâce à la fonction random de pygame, à la bouteille d'éther selon la liste des cases disponibles. J'ai supprimé cette position de la liste puis donné une position aléatoire au tube. Même opération pour l'aiguille.

Pour le gardien, sa position étant prédéterminée dans la structure du labyrinthe, ce dernier est absent de cette méthode.

LA FIN :

Pour gagner, il faut que MacGyver ramasse tous les objets et se présente devant le gardien.

Un dictionnaire est créé pour sauvegarder les objets que MacGyver a ramassés. Il fonctionne aussi comme un compteur selon la longueur de son contenu.

Si MacGyver fait face au gardien sans avoir ramassé tous les objets nécessaires, la partie est perdue. S'il a tous les objets, la partie est gagnée. Pour cela il suffira de consulter la longueur du dictionnaire contenant les objets.

CONCLUSION :

Les algorithmes utilisés dans ce projet sont de type boucle while, boucle for, et condition if/elif/else.

Il peut exister deux problèmes majeurs pour ce projet.

Le premier est le déplacement du personnage. Étant donné que notre image de fond est un labyrinthe construit au fur et à mesure, et non une image fixe, il faut afficher l'image du personnage à chaque tour de boucle, ainsi que sa case précédente, et non afficher le labyrinthe entier.

Le deuxième problème pourrait être la position aléatoire des objets. La manière utilisée est mentionnée plus haut.

Pour conclure, il est évidemment possible d'implémenter le jeu en y ajoutant un menu, un système de chronométrage ou un système de points par exemple. On pourrait aussi imaginer un système de combat entre MacGyver et le gardien pour rendre le jeu plus passionnant.

Pour ma part, j'ai préféré m'en tenir au cahier des charges du projet.