

PARCOURS OPENCLASSROOMS DEVELOPPEUR D'APPLICATIONS PYTHON

Projet 3 : Aidez MacGyver à s'échapper



<https://github.com/AxlMi/OC> Project3

Introduction :

Durant ma formation Openclassroom , il m'a été demandé de réaliser un labyrinthe doté de plusieurs fonctionnalités et de contraintes. Pour se faire j'ai commencé en respectant l'ordre de fabrication du labyrinthe donné sur le projet 3. J'ai commencé par créer le cadre de départ, d'animer le personnage, de récupérer les objets et ensuite de changer la fin du jeu en fonction des contraintes et conditions indiquées.

Apprentissage :

Pour réaliser ce projet je me suis aidé des cours de départ écrit par Céline Martinet Sanchez , du cours de Python écrit par Vincent Le Goff , du livre Apprendre à programmer avec Python 3 de Gerard Swinnen mais aussi principalement du cours « interface graphique Pygame pour Python » sur lequel je me suis inspiré pour le lancement du projet, sans oublier bien sur les explications et conseils de mon mentor.

Architecture du projet :

Module main : Elle commence par initialiser le jeu avec l'affichage des différentes informations de départ. Elle se compose aussi de la fonction `launch_game()`, qui permet de lancer le jeu. Elle est composée de deux boucles avec une condition de booléen permettant l'affichage du menu et du labyrinthe, ainsi que diverses conditions permettant le déplacement.

Modules constantes : Ce module se compose des différentes constantes du jeu.

Module classes : Dans ce module nous retrouvons principalement deux classes, la première permettant la construction, l'affichage du labyrinthe ainsi que des objets aléatoirement placés via des méthodes, la seconde classe concerne le personnage et ses méthodes pour se déplacer et de prendre des objets.

Algorithme :

Ramasser les items :

Pour ramasser les objets j'ai décidé d'utiliser une méthode nommée « take-obj » , elle indique les conditions de départ pour qu'elle soit exécutée , elle doit correspondre à une lettre différente d'un espace blanc ainsi que de la lettre « A » (correspondant à l'arrivée).

Une fois la condition effectuée, nous allons donner une valeur a la variable `obj_nb` qui déterminera le numéro de l'objet situé dans la liste « `rdm_objet` » si c'est `i` alors l'objet de cette liste correspondra à `i` ainsi de suite.

J'incrmente de 1 ma liste d'inventaire

Je modifie ensuite la case en espace blanc pour indiquer que l'objet a bien été ramassé et pour ne pas réitérer l'opération.

Difficultés :

Trouver une manière d'afficher le nom de l'objet ramassé et de pas répéter cette méthode à chaque fois que l'on passe sur cette case.

Position aléatoire des items :

Pour afficher les objets aléatoirement dans le labyrinthe j'ai effectué une boucle for avec les objets à placer dans le labyrinthe.

- Ensuite dans une boucle while, j'ai créé deux variables « random_x » et « random_y », chacune d'elles affichera un nombre aléatoire compris entre 0 et le nombre de sprite (-1 car on commence à 0).
- J'affiche une condition permettant d'afficher la map du labyrinthe avec les coordonnées x et y (données précédemment aléatoirement), si elles correspondent à un espace qui correspond au carrelage, dans ce cas on effectue la condition.
- La condition comprend une chaîne de caractère avec 3 lettres « ijk » je modifie donc la lettre concernant ma position dans le labyrinthe par une de ces lettres.

Fin de partie :

Pour indiquer une fin de partie, j'ai décidé d'indiquer que si la case correspondait à A et si la taille de l'inventaire correspond à celui des objets placés aléatoirement dans ce cas on gagne et on affiche une image sinon on perd et on affiche une image.

Conclusion :

Pour conclure, étant novice dans le monde du développement, je suis satisfait de ce que j'ai pu produire, même si je suis persuadé qu'il existe un moyen plus rapide et plus simple d'accéder à ce résultat. Ce projet m'a permis de découvrir et de mettre en pratique le fonctionnement de Pygame, la structuration d'un projet, l'utilisation de GitHub et d'utiliser un environnement virtuel.