

"Aidez MacGyver à s'échapper !"

OpenClassrooms parcours DA Python - projet 3

Projet sur GitHub : <https://github.com/beSyl/Projet3>

Le programme

Le fichier 'jeu.py' correspond au programme dont les modules sont 'contantes.py', 'Labyrinth.py' (classe), 'Hero.py' (classe).

Le fichier 'labyrinth.txt' représente la structure du jeu à l'aide de lignes de caractères.

Le répertoire 'images' contient les images utiles au jeu (fond, murs, personnages, etc.).

Une fichier README.txt détaille les règles du jeu.

Le présent document correspond au livrable « *Document texte expliquant votre démarche et comprenant le lien vers votre code source (sur Github)* », attendu dans le cadre du projet 3 OpenClassrooms.

Le jeu

Le labyrinthe est vu comme un plateau de jeu de 15 lignes (numérotées de 0 à 14) et 15 colonnes (numérotées de 0 à 14).

Pour le matérialiser, on rédige un fichier 'labyrinth.txt' de 15 lignes composées de 15 caractères chacune. Ces caractères symbolisent :

- une case 'chemin' via laquelle MacGyver peut se déplacer s'il s'agit d'un caractère espace « »
- une case 'mur' empêchant MacGyver de se déplacer s'il s'agit d'un caractère « X »
- la fin du jeu, donc le gardien, s'il s'agit d'un caractère « e »

La classe Labyrinth contient les méthodes :

- init : initialiser un labyrinthe en déterminant le fichier txt utile à sa structure, et la variable permettant de stocker la structure.
- create : lire le fichier texte et structurer le labyrinthe.
- place_objects : positionner les objets à ramasser.
- show : afficher le labyrinthe à l'aide des images stockées dans le répertoire 'images', en déterminant la position des éléments (fond, murs, objets, etc.) et leur dimension dans la fenêtre Pygame.

On instancie un labyrinthe à partir de la classe 'Labyrinth' :

```
labyrinth = Labyrinth()
```

La classe Hero contient les méthodes :

- init : initialiser un personnage MacGyver avec son image, ses positions dans le plateau de jeu et un compteur d'objets ramassés.
- move : gérer les effets d'un déplacement.
- run_pygame : gérer les déplacements au clavier.
- check_victory : vérifier si les conditions nécessaires à la victoire sont remplies.

On instancie le personnage MacGyver (instance 'macGyver') à partir de la classe 'Hero' :

```
macGyver = Hero()
```

Une position se détermine selon les axes x et y (respectivement, abscisse et ordonnée).

Le personnage débute en haut à gauche du labyrinthe ($x = 0$, $y = 0$).

La fin du labyrinthe se situe en bas à droite du labyrinthe ($x = 14$, $y = 14$). Elle est matérialisée par la lettre « e » dans le fichier 'labyrinth.txt'.

On détermine 3 positions aléatoires à l'aide de la méthode 'random.randint()' en s'assurant qu'il s'agit de cases 'chemin' que MacGyver peut parcourir. Les objets à ramasser y sont ainsi positionnés, et matérialisés par les lettres :

- « E » (ether)
- « N » (needle, l'aiguille)
- « T » (tube)

dans la structure du labyrinthe, à savoir la variable de la classe Labyrinth :

`labyrinth.structure`

Un objet est ramassé lorsque la position du personnage est identique à celle de l'objet. La lettre «E », « N » ou « T » est alors modifiée en un caractère espace « » et on incrémente le compteur d'objets ramassés : `macGyver.cart`

Les déplacements sont gérés en ajoutant / retranchant 1 à la position de MacGyver :

- en abscisse (x) s'il se déplace de manière horizontale
- en ordonnée (y) s'il se déplace de manière verticale

Dans la fenêtre de jeu Pygame, les positions en abscisse et en ordonnée sont multipliées par la dimension d'une case en pixels afin que chaque élément (case, objet, MacGyver, etc.) soit correctement positionné à l'écran

Le joueur gagne s'il parvient à conduire MacGyver jusqu'à la case d'arrivée après avoir ramassé les 3 objets. S'il n'a pas ramassé tous les objets en atteignant la fin, il perd : MacGyver ne peut gagner que si la méthode `check_victory` renvoie 'True' lorsqu'il a atteint la dernière case du jeu où se trouve le gardien.