

Lien GITHUB/

<https://github.com/Talanoc/macgyver.git>

Etapes :

1 - Créer le cadre de départ

- Creation d'un fichier excel labyrinthe représentant le labyrinthe avec des cases numérotées.
- Import d'un onglet du fichier excel labyrinthe1 ou labyrinthe2 (maze_sample.pdf)
- Conversion du fichier xlsx en liste « wall_display » contenant la liste des cases mur
- Extraction des positions du gardien et de macgyver de la liste wall_display
- Création de la liste data_maze contenant (case , x , y , sprite , name) wall et floor
- Création et placement des items dans la liste data_maze
- Initialisation de la fenetre de jeux et affichage des sprites dans celle-ci

2 – Animer le personnage

- Détection de l'appui d'une touche (choix du relachement de la touche)
- Appui sur la touche ESC (sortie de la boucle loop)
- Appui sur les touches directionnelles (ajoute ou soustrait un step en fonction de la direction)
- Sauvegarde des positions de départ (old_position) et d'arrivée (position_macgyver)
- Affichage des sprites sur la fenetre (old_position et position_macgyver)

3 – Récupérer les objets

- initialisation des variables ether_in_bag ,syringe_in_bag et needle_in_bag à False
- détection de la présence de l'objet et passage de la variable à True
- création de la variable win qui devient True quand les 3 variables _in_bag sont passées à True

4 – Gagner

- Détection de la présence du gardien sur la case
- Si win=True le joueur gagne
- Si win = False le joueur perd

5 –Difficultés

- Compréhension de la programmation orientée objet pas complète
- Passage du programme de départ maze_macgyver à la version POO

Conclusions : la réalisation de ce projet m'a posé beaucoup de problèmes.J e ne suis pas sur d'avoir acquis toutes les subtilités de la POO .Les modalités de ce style de programmation semblent nombreuses.

