Etapes

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

1 - Créer le cadre de départ

- Creation d'un tableau excel représentant le labyrinthe avec des cases numérotées.
- Import du fichier excel (labyrinthe ou labyrinthe2)
- Conversion du fichier xlsx en liste « wall_display »
- Affectation des positions du gardien et de macgyver
- Création de la liste floor_display
- Ouverture de la fenetre de jeu size*size
- Changement de nom et d'icone
- Chargement et mise en forme des décors
- Creation de la liste (numero de case,localisation x,localisation y) et remplissage de la fenetre avec le sol
- Placement des murs
- Création de la liste de position des objets

- Placement des objets et des personnages Initialisez un repo Git et envoyez-le sur Github.

Commencez par créer le labyrinthe sans l'interface graphique. Quand la logique de votre labyrinthe est faite, utilisez le module PyGame pour dessiner l'interface graphique.

Puis intéressez-vous aux trois éléments principaux du jeu : le gardien, MacGyver et les objets. Comment les représenter dans votre programme ? Où sont-ils placés au commencement du jeu ?

2 - Animer le personnage

Le seul élément mouvant est MacGyver. Créez les méthodes de classe qui permettent de l'animer et de trouver la sortie. Pour l'instant, faites une version simplifiée du jeu dans laquelle MacGyver gagne en arrivant face au gardien.

3 - Récupérer les objets

Ajoutez la gestion des objets. Comment MacGyver les ramasse-t-il ? Ajoutez également un compteur qui les listera.

4 - Gagner !

Enfin, changez la fin du jeu : MacGyver gagne s'il a bien ramassé tous les objets et endormi le garde. Sinon, il perd.

Livrables

- **Programme** hébergé par Github,
- Document texte expliquant votre **démarche** et comprenant le lien vers votre code source (sur Github). Développez notamment le choix de l'algorithme. Expliquez également les difficultés rencontrées et les solutions trouvées. Le document doit être en format pdf et ne pas excéder 2 pages A4.

Pour faciliter votre passage au jury, déposez sur la plateforme, dans un dossier nommé "*P3_nom_prenom*", tous les livrables du projet. Chaque livrable doit être nommé avec le numéro du projet et selon l'ordre dans lequel il apparaît, par exemple "*P3_01_programme*", "*P3_02_démarche*", et ainsi de suite.

Contraintes

- Vous versionnerez votre code en utilisant Git et le publierez sur Github pour que votre mentor puisse le commenter,
- Vous respecterez les bonnes pratiques de la PEP 8 et développerez dans un environnement virtuel utilisant Python 3,
- Votre code devra être écrit en anglais : nom des variables, commentaires, fonctions...

Pensez à mettre à jour votre page de profil quand votre mentor aura validé votre projet !