Lien du projet : https://github.com/Claers/ProjectMacGyver

- Présentation globale

Le jeu est assez simple, un personnage contrôlé par le joueur se déplace dans un labyrinthe case par case. Son but est de récupérer des objets qui sont disposé aléatoirement et d'arriver devant un garde, si le joueur a récupéré tout les objets il gagne, sinon il perd.

Pour réaliser ce projet j'ai opté pour une approche orienté objet. La fenêtre du jeu est donc un objet qui à différentes méthodes qui se rapproche d'un GameEngine. En effet a chaque boucle du jeu on à : La récupérations des événements, une boucle qui va effectuer différents tests, la création de l'affichage GUI et finalement l'affichage. Tout ce qui est affiché a l'écran, à par le GUI, sont créé en tant qu'objet et plus spécifiquement en Sprite, un objet hérité de pygame.

- Création graphique du labyrinthe

Pour la création du labyrinthe graphiquement, le programme va lire une liste qui contient un chiffre indiquant la nature de la case : 0 pour un mur, 1 pour un chemin, 2 pour le départ et 3 pour l'arrivée.

- Collisions avec les murs

Le jeu requiert un système de détection de collision, j'ai donc créé une liste qui stocke les objets mur et chemin. A chaque mouvement du joueur un simple test va vérifier si la case sur laquelle se dirige le joueur est une case 1, une case chemin, et si oui le mouvement sera possible.

- Objets

Les objets sont créé au démarrage du programme, un algorithme va tirer aléatoirement une case chemin de la liste des collisions, sans prendre en compte les cases du joueur et du gardien.

Pour détecter une collision des objets, le programme va a chaque nouvelle position du joueur voir si il n'est pas sur un objet. Cette façon de faire permet une bonne optimisation du programme.

- Affichage

Cette partie à été la plus difficile pour moi vu que je n'ai pas pensé au système de Sprite dès le début. J'avais donc commencé par une méthode d'affichage qui m'a posé problème lors de la création du joueur. J'ai du donc tout repensé pour que mon code fonctionne avec les Sprites. A partir de ça j'ai pu afficher

les objets par groupe, les murs et les chemin en premier, le personnage et les objets en second pour qu'ils apparaissent au dessus.

- Environnement Virtuel

Le principe d'environnement virtuel m'était inconnu avant ce projet, j'ai du donc me plonger sur la question et il est vrai que c'est pratique. J'ai donc créé mon environnement virtuel avec virtualenv et virtualenvwrapper-win qui est une version pour windows. J'ai ensuite installé ce qu'il me fallait donc pygame et cx_freeze pour créer un executable.