

# Soutenance du PROJET 03

Antoine THIBAUD

Jeu du labyrinthe : « Aidez MacGyver à s'échapper »

*Le projet 03 est un jeu de labyrinthe réalisé en 2D avec le langage de programmation Python avec le module Pygame pour l'interface graphique.*

## **A) Environnement de travail :**



Ce projet a été réalisé avec le langage Python (version 3.6) et le module Pygame depuis un environnement virtuel installé avec le gestionnaire de paquets PIP (package installer for python). Son code source est disponible sur le site internet Github à l'adresse suivante :

<https://github.com/athd33/MacGyver-Openclassroom>

Le dossier contient l'ensemble des fichiers, images et sons nécessaires au bon fonctionnement du programme. Un fichier 'requirements.txt' qui indique la liste des modules utilisés ainsi que leurs versions.

## **B) Présentation du jeu :**

Mac-Gyver est enfermé dans un labyrinthe. Le garde se trouve devant la sortie et empêche notre héros de retrouver la liberté. Pour s'échapper, Mac-Gyver envisage de récolter les objets présents dans le labyrinthe pour se fabriquer une seringue améliorée avec laquelle il pourra endormir le garde et sortir sereinement.

Le premier objectif du joueur est de parcourir la carte afin de récolter les objets dispersés. Pour se déplacer, le joueur utilise les flèches directionnelles du clavier. Une fois cette étape accomplie, il doit se diriger vers la sortie représentée par un personnage (le gardien du labyrinthe).

Si l'ensemble des objets présents sur la carte ont été récoltés avant d'arriver à la sortie, la seringue va fonctionner et le joueur gagne. À défaut, au contact du gardien du labyrinthe il perdra la partie.

## **C) Difficultés rencontrées :**

Pour la réalisation de ce projet, comprendre le système de tableau à deux dimensions a probablement été ma plus grande difficulté.

Du reste, la manipulation de classes m'a permis d'appréhender les concepts généraux de la programmation orientée objet. Le passage d'informations de classes en classes m'a rapidement paru très intéressant et utile.

#### **D) Choix algorithmiques :**

Le jeu se lance depuis le fichier *main.py*. Ce dernier contient l'initialisation du module Pygame ainsi que les boucles principales du jeu (affichage du menu, jeu en cours, menu de victoire et menu de défaite).

Au lancement du jeu, Pygame est initié et les différents modules créés sont importés. La configuration de la surface de Pygame est définie et 3 classes sontinstanciées :

##### **a) La classe *MappToDisplay* :**

Cette classe sert au rendu de la carte. Elle décompose le contenu du fichier *mapp.txt* et génère une liste de liste. L'indexation de chaque élément de la carte est réalisé, ils sont accessibles avec des index X et Y.

##### **b) La classe *Player* :**

Cette classe prend en paramètre la classe *MappToDisplay* précédemment instanciée. Avec sa méthode *init()*, la classe *Player* récupère la position du joueur avec ses coordonnées X et Y. Sa méthode *moveMac()* prend en paramètre la touche directionnelle utilisée par le joueur. En fonction de la direction donnée, elle modifie la position du joueur et renvoie à la classe *MappToDisplay* les nouvelles coordonnées du personnage que cette dernière sera chargée de rendre à l'écran grâce à Pygame.

##### **c) La classe *ObjectsItems* :**

Cette classe, à l'instar de la classe *Player*, prend en paramètre la classe *MappToDisplay* précédemment instanciée. Avec sa méthode *getFreeCases()*, elle crée une liste qui contient l'ensemble des positions libres sur la carte (non occupées par un mur ou personnage). Le module *Random* et sa méthode *choice()* sont utilisés pour générer depuis cette liste, une position aléatoire. Enfin, sa méthode *displayObject()* attribue à chaque item du jeu, une position aléatoire qui sera retournée à la classe *MappToDisplay* pour l'affichage.

*Nota : La classe *ObjectsItems* sert à placer de façon aléatoire, les item du jeu à chaque nouvelle partie.*

Une fois ces classes instanciées, le menu se lance. Pygame affiche une image et les commandes disponibles (quitter, lancer le jeu, activer et désactiver le son).

Si le joueur appui sur la touche ENTRÉE, le menu se ferme, le jeu se lance et la carte s'affiche. La classe ***Loader*** est instanciée, elle affiche l'image d'un loader en haut de la carte. Quand le joueur récupère un item sur la carte, la classe ***Player*** indique à la classe ***Loader*** que le joueur a marqué 1 point supplémentaire. Il change alors d'image à afficher.

Une fois que le joueur a récupéré l'ensemble des items (4 points), les conditions sont réunies pour qu'il sorte vainqueur. Au contact du gardien, le jeu s'arrête et la page de fin de jeu s'affiche. Le joueur peut quitter le programme avec la touche Q ou recommencer avec la touche ENTRÉE.

Si le joueur n'a pas ses 4 points au contact du gardien, le jeu s'arrête. La fenêtre affiche le message de fin de partie (perdue), Le joueur peut quitter le programme avec la touche Q ou recommencer avec la touche ENTRÉE.