

# **PROJET 3 : Aidez Mac Gyver à s'échapper !**

## **SOUTENANCE DE PROJET**

**Bastien Renaud**

(Exodrum : GitHub & Open Classrooms)

Code Source : [https://github.com/Exodrum/PROJET\\_3\\_MAC\\_GYVER](https://github.com/Exodrum/PROJET_3_MAC_GYVER)

### **Introduction :**

Cette documentation présentera le « **Projet n°3 : Aidez Mac Gyver à s'échapper !** » réalisé dans le cadre de ma formation de **développeur d'application Python** sur **Open Classrooms**.

Ce document aura pour objectifs d'expliquer les contraintes, difficultés rencontrées, choix technique et modules choisis durant le projet.

### **Structure Programme :**

Ci-dessous un aperçu de l'organisation du programme.

#### **Fichiers :**

- lvl.txt ; main.py ; classe.py ; constants.py ; readme.txt ; requirements.txt

#### **Modules :**

- Random ; Pygame

#### **Classes :**

- Level ; MacGyver ; Guardian ; Item

### **Analyse de programmation :**

Premièrement, j'ai créé la structure logique du labyrinthe dans un fichier texte.

Je me suis intéressé à la structure du programme que j'ai programmé en trois parties, cela me semblait le plus approprié pour éviter d'avoir trop de scripts pour un programme simple.

### **Voici les scripts :**

**constants.py** : contient les images utilisées dans le jeu, les dimensions de la fenêtre d'affichage et les données de positionnement de certaines variables.

**classe.py** : contient les classes des objets du jeu (objets, personnages, labyrinthe).

**main.py** : contient les paramètres principaux du jeu (boucles et gestion d'affichage du jeu).

Je me suis penché sur le calcul des dimensions de la fenêtre, j'ai choisi un rectangle de 15x25 (étant un projet personnel, je me suis permis de modifier les dimensions pour afficher un background plus agréable à l'œil).

### Vient après la partie des constants, la création des classes :

#### ➤ Ecriture de la classe Level :

Initialise une liste vide et ajoute une condition pour parcourir le fichier « lvl ».

Ecriture d'une méthode **generate**, crée une liste vide et ajoute une condition pour parcourir le fichier « lvl » et gère l'affichage du labyrinthe.

Créer les lignes et les cases où seront placées les images dans la fenêtre et enregistre le labyrinthe.

Ecriture d'une méthode **toggle** pour ajouter la texture des murs et le garde au bon endroit grâce à la structure précédemment chargée.

#### ➤ Ecriture de la classe MacGyver :

Initialise le personnage Mac Gyver et lui donne ses caractéristiques (images, position, nombre d'objets).

Ecriture d'une méthode « **move** » pour définir ses mouvements et gère les collisions.

Ecriture d'une méthode « **take\_items** » pour prendre un objet quand Mac Gyver rentre collision avec celui-ci, le rajoute à son inventaire. Si la position de MacGyver est égale à la position de l'objet, supprime l'image de l'objet et ajoute à la méthode « **inventory** » +=1.

Ecriture d'une méthode « **inventory** » pour qu'à chaque objet que Mac Gyver obtient rajoute le nombre d'items en sa possession grâce à la méthode « **take\_items** ».

#### ➤ Ecriture de la classe Guardian :

Initialise le personnage guardian dans labyrinthe et lui donne 1 point de vie.

Ecriture d'une méthode « **damage** » pour si dommage sur gardien enlève 1 au point de vie.

#### ➤ Ecriture de la classe Item :

Initialise les objets, leur donne des caractéristiques et leurs positions random.

Ecriture d'une méthode « **randpos** » pour donner aux objets une position au hasard à chaque fois que la boucle du jeu est relancée. J'ai utilisé la fonction « **randrange** » en donnant le nombre de case max (largeur et longueur) entre  $1 < X$ , lui donner une position à condition si les objets apparaissent sur les cases « 0 ».

Ecriture d'une méthode « **damage** » et « **display** » si Mac Gyver se situe sur la même case que l'objet, enlève 1 point de vie et donc enlève l'objet de sa position « 0 » si **health** > 0.

Pour finir j'ai écrit le script principal où sont importés les constantes des caractéristiques de la fenêtre du jeu, les classes. Puis programmer les variables : **continu\_game**, **continu\_home**, **continu** pour activer ou désactiver les boucles principales : du menu principal et du jeu avec leurs conditions. Définition du contrôle des mouvements de Mac Gyver avec les touches directionnelles et conditions WIN/LOOSE à l'aide du système d'inventaire.

### **Création d'un environnement virtuel avec virtualenv :**

Nécessaire pour avoir des installations de python isolées du système d'exploitation donc de pouvoir réutiliser le programme avec la bonne version des modules.

Installation via la CMD \$ pip install virtualenv

Création de l'environnement virtuel \$ python -m venv C://vers/projet/env\_nom\_projet

Puis dans le dossier \$ mkvirtualenv nom\_env

Isolation de mon dossier avec \$ python -m venv C://vers/projet/env\_nom\_projet/activate.bat

### **Vérification PEP 8 avec pylint :**

Installation de pylint via CMD.

### **Difficultés rencontrées :**

J'ai eu quelques problèmes principaux lors de la programmation :

#### **1) La génération d'objet random :**

J'ai mis beaucoup de temps avant de trouver comment générer les objets de manière aléatoire dans la structure du labyrinthe, j'ai fini par utiliser **random.range** qui permet à la méthode de parcourir les cases de la fenêtre, y déposer un objet de manière aléatoire, donc sur l'emplacement « 0 » et de retourner sa position pour permettre à la classe de l'afficher.

#### **2) Les méthodes pour récupérer la valeur de l'objet et l'ajouter à l'inventaire :**

Ce problème-là n'a pas duré très longtemps, j'ai cherché quelques indices sur le net, il m'a suffi de définir une classe pour prendre un item lorsque qu'il perd sa vie et de l'ajouter via une autre méthode inventaire « inventory » qui ajoute +1 à celui-ci.

### **Pour l'amélioration du jeu :**

- Musiques d'ambiance et sonores pour l'immersion.
- Différent niveau dans la structure du labyrinthe, par exemple passer sur une autre zone via un portail, pour ajouter de la difficulté à trouver les objets et rendre le temps jeu plus long. Salle secrète
- Personnages Non Joueur avec une direction prédéfinie, qu'il faut esquiver pour éviter que Mac Gyver meure.
- Rajouter une barre de vie et une barre armure pour avoir plus de chance face au gardien.

