

# Fichier technique

---

# Sommaire

**# SOMMAIRE**

**# LIEN VERS LE GITHUB**

**# INTRODUCTION**

**# STRUCTURE DU DOSSIER**

**# DIFFICULTES RENCONTRES**

**# AMELIORATIONS DU JEU POSSIBLE**

## # Lien vers le GitHub

[>> Cliquez ici pour accéder au GitHub <<](#)

## # Introduction

"Il faut sauver MacGyver" est un jeu développé en Python3.6.5 avec la bibliothèque libre multiplateforme nommée Pygame1.9.4.

## # Structure du dossier

Le fichier permettant de lancer le jeu est nommé « *main.py* ».

Dans le dossier « *images* » se trouve toutes les images nécessaires au fonctionnement du jeu.

Le dossier « *levels* » contient la structure du niveau, il y a 2 niveaux.

Le fichier nommé « *constants.py* » contient toutes les constantes nécessaires (chemin vers les différentes images, taille des sprites, taille de la fenêtre du jeu etc..), à noter que l'image des murs, du sol ainsi que le fichier qui sera utilisé pour la structure du jeu sont sélectionnés de manière aléatoire.

« *utils* » est un module contenant les classes et les fonctions utiles au bon fonctionnement du jeu.

## # Difficultés rencontrés

Le jeu a été développé dans un environnement virtuel avec Pipenv.

Durant le développement, j'ai rencontré un problème de version entre Pipenv (précisément la version de Python dans Pipenv) et Pygame.

En effet Pygame ne pouvait pas se lancer sur une version Python3.7 en étant dans Pipenv.

J'ai donc dû revenir à une version inférieure de Python (la version 3.6.5).

J'ai résolu ce problème en lisant les tickets déjà émis sur le GitHub de Pipenv à ce sujet.

Je n'arrivais pas à gérer les collisions dans le jeu, après plusieurs dizaines de minutes de réflexion, je suis venu à la conclusion que dès que le joueur essaierait d'aller sur une case je vérifierai s'il n'y a pas la présence d'un mur ou qu'il sortait de la limite du jeu, si oui le joueur serait remis à sa précédente position.

## # Améliorations du jeu possible

Le jeu peut encore être amélioré, par exemple :

- Je pourrais ajouter la fonctionnalité qui permettrait à l'ennemi de se déplacer seul.
- L'ennemi pourrait chercher lui aussi les objets.
- Les murs pourraient être placés de façon aléatoire (mais de manière cohérente avec un algorithme adéquat) à la conception du niveau.
- Tous les objets ainsi que les murs pourraient aussi se déplacer pendant le jeu ce qui rendrait la tâche plus difficile.