**Aidez MacGyver à s'échapper !**

100 heures Mis à jour le lundi 20 janvier 2020

 Pour valider ce projet, vous passerez une soutenance auprès d'un mentor *validateur*.

Votre mentor accompagnateur vous inscrira pour cette soutenance dès que vous aurez défini une date de complétion de ce projet.

Vous venez de faire vos premiers pas avec Python et avez averti tous vos amis. Bravo ! Les réactions s'enchaînent sur votre mur Facebook et vous vous sentez l'âme d'une rock star avant la sortie de son dernier disque. Et là, c'est le drame : un troll a la bonne idée de commenter "Python ça sert à rien. Essaie plutôt Swift, tu pourras faire des jeux pour iOS". Ni une ni deux, vous décidez de créer un jeu pour montrer à votre "ami" de quel bois Python se chauffe.

Étant un grand fan de Richard Dean Anderson, vous imaginez un labyrinthe 2D dans lequel MacGyver aurait été enfermé. La sortie est surveillée par un garde du corps dont la coiffure ferait pâlir Tina Turner. Pour le distraire, il vous faut réunir les éléments suivants (dispersés dans le labyrinthe) : une aiguille, un petit tube en plastique et de l'éther. Ils permettront à MacGyver de créer une seringue et d'endormir notre garde.

Fonctionnalités

* Il n'y a qu'un seul niveau. La structure (départ, emplacement des murs, arrivée), devra être enregistrée dans un fichier pour la modifier facilement au besoin.
* MacGyver sera contrôlé par les touches directionnelles du clavier.
* Les objets seront répartis aléatoirement dans le labyrinthe et changeront d’emplacement si l'utilisateur ferme le jeu et le relance.
* La fenêtre du jeu sera un carré pouvant afficher 15 sprites sur la longueur.
* MacGyver devra donc se déplacer de case en case, avec 15 cases sur la longueur de la fenêtre !
* Il récupèrera un objet simplement en se déplaçant dessus.
* Le programme s'arrête uniquement si MacGyver a bien récupéré tous les objets et trouvé la sortie du labyrinthe. S'il n'a pas tous les objets et qu'il se présente devant le garde, il meurt (la vie est cruelle pour les héros).
* Le programme sera standalone, c'est-à-dire qu'il pourra être exécuté sur n'importe quel ordinateur.

Fichiers nécessaires au programme

Un ami graphiste, soucieux de votre santé, a décidé de vous aider un peu et vous a envoyé certains éléments graphiques nécessaires au jeu. Les voici :

Murs, fonds et accessoires : [Téléchargez [.zip - 387ko]](https://s3-eu-west-1.amazonaws.com/course.oc-static.com/projects/macgyver_ressources.zip)

MacGyver : 

Gardien : 

Ces ressources ont été mises à disposition originellement sous licence [**WTFPL**](https://fr.wikipedia.org/wiki/WTFPL) sur le site [**http://jessefreeman.com/**](http://jessefreeman.com/)

Etapes

**1 - Créer le cadre de départ**

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 !

**Compétences évaluées**

 Lire et comprendre une documentation de module

 Gérer les différentes versions de Python et ses modules en fonction des projets

 Créer des scripts pour le web en utilisant Python

 Utiliser un algorithme pour résoudre un besoin technique

 Coder efficacement en utilisant les outils adéquats

 Conceptualiser l'ensemble de son application en décrivant sa structure (Entités / Domain Objects)