# PROJET 3 : Help MacGyver to escape from the Red Keep !

Document texte expliquant ma démarche et comprenant le lien vers mon 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.

Lien Github : <https://github.com/Maximedu13/Help-MacGyver-to-escape-from-the-Red-Keep->

**Les règles du jeu.**

Emprisonné par la reine Cersei Lannister, MacGyver doit s’échapper du Donjon Rouge en réussissant à collecter 3 objets :

* une aiguille
* de l’éther
* un tube en plastique

Ceux-ci permettront la fabrication d’une seringue, visant à endormir la gardienne. Il doit se présenter à celle-ci avec ces 3 objets. Dans le cas contraire, il reste emprisonné et meurt.

**Les fichiers**

* home.py : c’est le menu du jeu ainsi que son point de départ. L’utilisateur lance le terminal dans le dossier du jeu et écrit home.py puis appuie sur entrée. Une interface apparait alors. L’utilisateur peut lancer le jeu en appuyant sur F1, le quitter en appuyant sur F2, et même couper le son avec la touche F3.
* main.py : c’est le fichier central du jeu. Il permet d’afficher les personnages, les items, le labyrinthe, le sac, de collecter les items etc.
* constants.py : il s’agit des constantes du jeu (texte, couleurs, polices etc.)
* maze.py : génère le labyrintheainsi que les objets/items collectables.
* character.py : il permet à Macgyver de se déplacer et lance la méthode victoire ou défaite.
* defeat.py et victory.py : conditionnent la victoire et la défaite.
* map : la carte du jeu sous forme de symboles.
  + ○ est un sprite qui affiche la case de départ.
  + ■ est un sprite qui affiche un mur.
  + – est un sprite qui affiche un chemin
  + ◘ est un sprite qui affiche la case d’arrivée.

Les fichiers main, home, victory, et defeat sont organisés de façon bilatérale : une méthode qui charge les éléments et une méthode qui les affiche. Les fichiers maze et character définissent différentes classes et méthodes.

**Choix des algorithmes**

Define\_position

Deplacement macgyver

loadItems

endgame : conditionne la victoire ou de la defa

Il faut tout d’abord vérifier la structure ou sprite d’arrivée de l’objet, il faut que celui-ci soit collectable, donc sur un chemin puis d’utiliser la fonction random.randint(). Il faut aussi que les différents items ne se superposent pas.

Il faut utiliser with open et lire les différentes lignes du fichier.

**Difficultés rencontrées et solutions trouvées.**

* La première difficulté que j’ai rencontrée est l’appels de fonctions externes.
* La seconde est comment créer une fonction permettant de générer un labyrinthe a part d’un fichier externe (ici map).
* La troisième difficulté a laquelle j’ai été confronté a été définir une position aléatoire pour un objet sur le labyrinthe.
* Refactoring de code : comme je n’ai pas eu de soutenance pendant 2 semaines, mon mentor m’a aidé a comprendre le principe de refactoring de code, bien séparer le code en différentes fonctions.

**Améliorations possibles.**

* Changer les sprites et donner un relief moins « 2D » au jeu.
* Enrichir le jeu (élargir le labyrinthe en dimension, rajouter des objets, éventuellement ajouter des personnages qui pourraient interagir avec le héros etc.)