

Projet 3 - Aidez MacGyver à s'échapper !

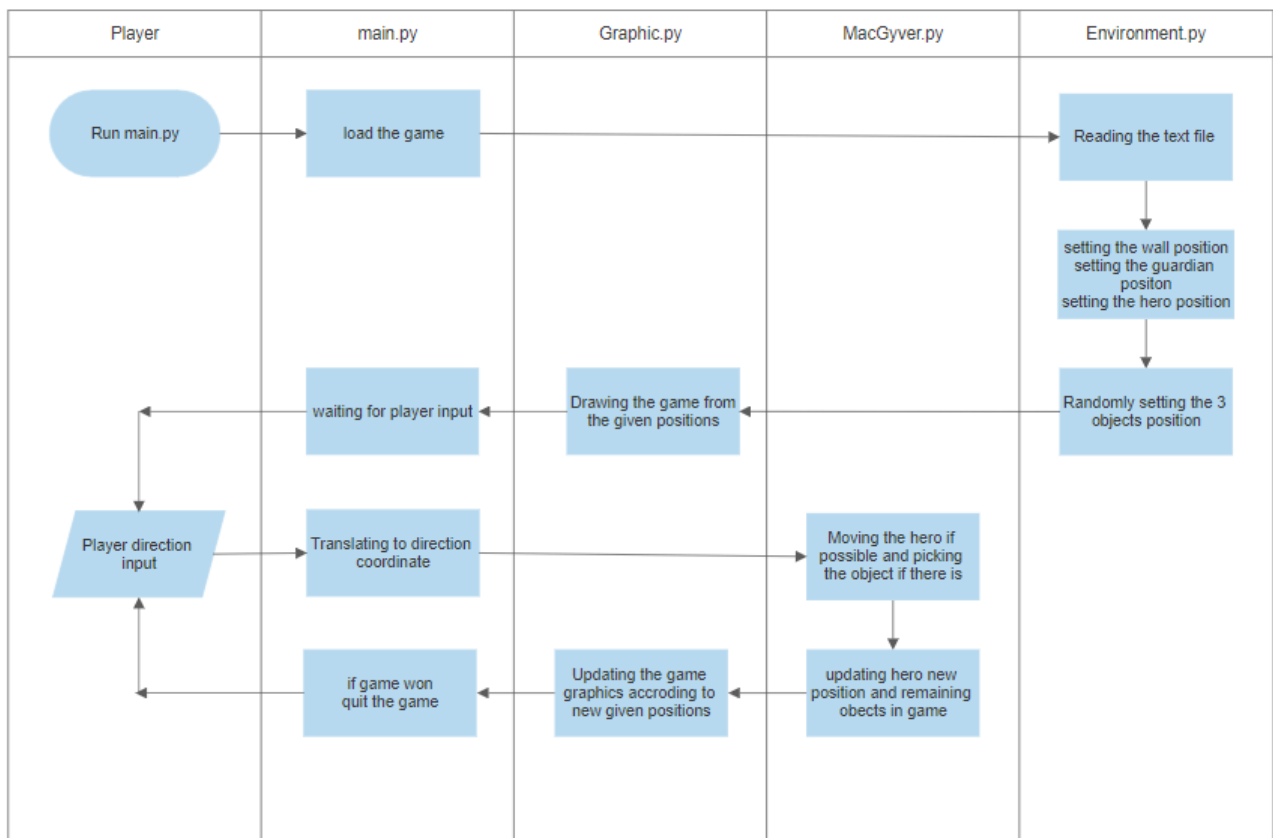
https://github.com/Natnat94/Project_3.git

Afin de mener à bien le projet 3, j'ai procédé à un développement du jeu en plusieurs étapes. La première étape a consisté à bien assimiler l'énoncé et les points importants du projet. Une fois le projet bien assimilé, j'ai mis par écrit les différentes fonctions que doit avoir le jeu, puis j'ai créé un diagramme de déroulement (flowchart) afin d'ordonner dans mon esprit le déroulement du jeu et de cibler les points qui devront être approfondis.

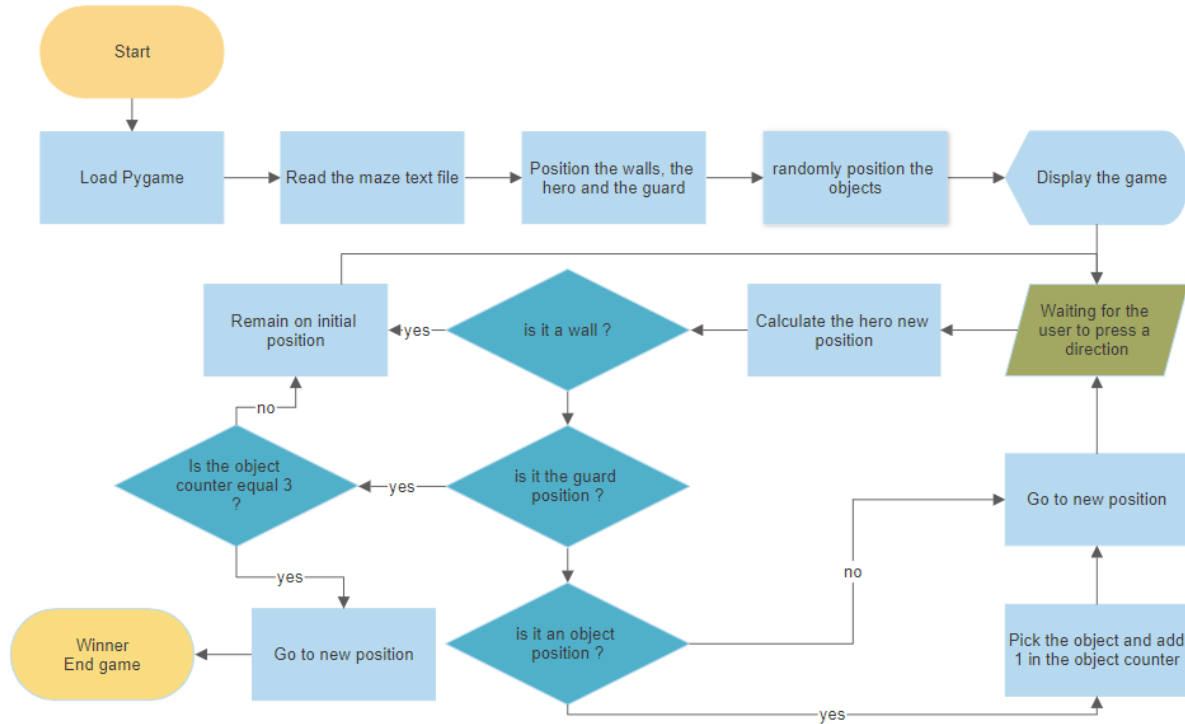
Ensuite, sur les conseils de mon mentor, j'ai mis par écrit les différentes classes qui feront fonctionner le jeu et les méthodes qui leurs seront attribuées. La découpage du jeu que j'ai choisis est le suivant:

J'ai souhaité distinguer la partie graphique, la partie objet inerte, la partie objet mouvante et la partie du fonctionnement du jeu.

- Un premier fichier (main.py) qui contient le déroulement du jeu et qui fait figure de chef d'orchestre et fait intermédiaire entre le joueur et le jeu.
- Un second fichier (environment.py) qui génère la partie back-end du jeu avec la position des objets et des murs à partir d'un fichier texte
- Un troisième fichier (macgyver.py) qui gère le déplacement du héros et son interaction avec son environnement (récupérer un objet, ne pas traverser un mur)
- Un dernier fichier (graphic.py) qui génère les graphismes à partir des informations fournies par environment.py et macgyver.py



L'algorithme choisi est simple, pour chaque déplacement du héros, le jeu vérifie que la nouvelle position n'est pas un mur, puis que ce n'est pas le gardien. Ensuite, si la nouvelle position est celui d'un objet, le héros le récupère (l'objet est retiré du dictionnaire) et le compteur d'objet est incrémenté de 1. Lorsque le héros veut se déplacer sur la position du gardien, si le compteur d'objet n'est pas égal à au moins 1, le gardien est considéré comme un mur, sinon le héros peut se déplacer sur la position du gardien et le jeu est gagné si il a récupéré les trois objets.



J'ai rencontré des difficultés à attribuer pour chaque position d'un objet à récupérer, une image dédiée qui reste fixe car, lorsqu'un objet est récupéré, le jeu réattribue une image à chaque objet. La solution que j'ai trouvée et qui m'a permis d'éviter beaucoup de lignes de code superflues a été de me servir d'un dictionnaire avec comme clé le nom de l'image d'un objet et comme valeur sa position.

J'ai eu des difficultés à maîtriser le principe des classes, mais après avoir vu et lu à nouveau des cours sur les classes et des essais sur mon jeu, j'ai réussi à maîtriser suffisamment le sujet.