Projet n°3

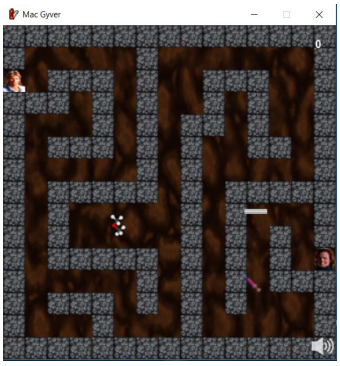
Zoé Belleton - https://github.com/Zoeinfp/macgyver/tree/p3

Aidez MacGyver à s'échapper !

2 juillet 2018

Présentation du projet

Réalisation d'une application développée avec le framework Python intitulé Pygame .

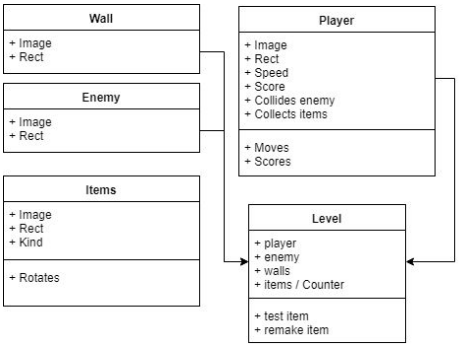


Set de modules dédié aux applications GUI reposant sur TKinter

Le but ici est de diffuser un jeu standalone qui a pour thème l'évasion d'un labyrinthe et pour sujet MacGyver qui devra se déplacer afin de récupérer trois objets lui permettant d'atteindre la sortie du niveau en se plaçant sur les objets pour les récupérer et sur le garde pour interagir avec lui.

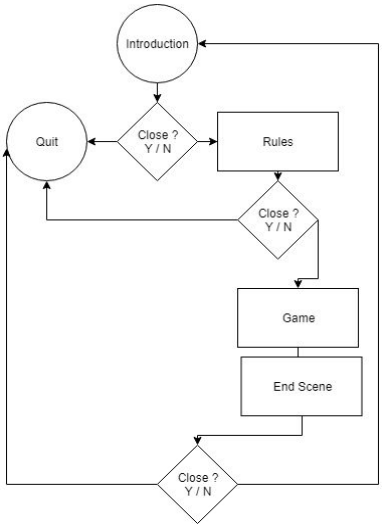
Le jeu en fonctionnement

Tous les objets constituant le niveau (Mac Gyver, le garde et les murs) sont stockés dans un fichier que le jeu utilise ensuite afin de l'afficher à l'écran. Les différents objets, tube en plastique, aiguille et flacon d'éther sont eux générés de manière complètement aléatoire. Si Mac Gyver n'a pas réuni les dits objets alors il ne pourra pas endormir le garde et perd la partie. Si le jeu est fermé avant la fin de celui-ci alors le jeu est terminé. Il s'agit donc d'un "Game Over". Les objets sont comptés à l'écran afin de vérifier si le score a bien atteint les 3 objets requis.



Une classe pour chaque élément du niveau

L'architecture du jeu est simplifié par la séparation de chaque élément constitutif de l'application.



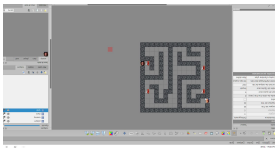
La boucle de jeu

Le principe de la boucle de jeu est de permettre l'interruption immédiat de celui-ci avec pour seule condition de ne pas être déjà en jeu. Si l'on souhaite fermer le jeu pendant la partie alors nous arrivons sur la scène affichant le score ou bien le message prévenant l'interruption du jeu. A ce stade là vous pouvez soit valider la fermeture en fermant la fenêtre ou bien rejouer avec toute autre action de la part du joueur. Ce qui est intéressant pour un jeu qui ne possède qu'un niveau qui n'est jamais identique puisque la génération des objets est aléatoire.

Le déplacement dans le labyrinthe

Le principe de notre algorithme est de demander au joueur uniquement de savoir marcher. C'est le jeu lui même qui permet ou non au joueur d'avancer si il ne rencontre pas de mur et si le jeu n'est pas en pause.

L'utilisation d'un éditeur de niveau et d'un chargeur de carte (Tiled et PyTMX)



La force de notre jeu est de permettre un level design extrêmement simplifié par l'utilisation d'un éditeur de niveau et d'un chargeur de carte à l'aide de Tiled et PyTMX

Pistes d'évolutions.

- Un inventaire qui pourrait présenter les objets au lieu de les compter
- La création de niveau supplémentaires (comme réalisé dans le code contenu sur la branche "master" du dépôt GIT)
- Une vue à la première personne. On pourrait afficher sur un deuxième écran ce qui est visible par le joueur.