

Parcours: Développeur D'application Python

Projet 3: Aidez MacGyver à s'échapper

lien du projet 3 sur le compte Pianistealb sur Github:

https://github.com/Pianistealb/Projet_3.git

I-Présentation du jeu et de ses contraintes

a) Quel est le but du jeu

Macgyver doit récupérer 3 objets (une aiguille, un tube en plastique et de l'éther) afin de confectionner un anesthésiant pour endormir le gardien « Murdoc » et s'échapper du labyrinthe.

b) Les contraintes

-Macgyver contrôlé par les touches directionnelles avance de case en case pour récupérer les objets, en se déplaçant dessus. Les items doivent être affichés dans un inventaire. Enfin s'il se présente devant le garde sans avoir tout ramassé, il meurt.

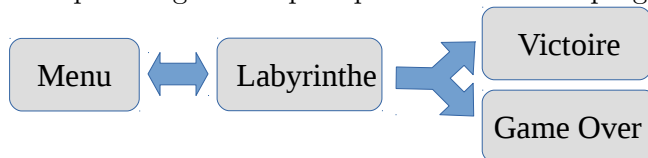
-Dans un labyrinthe de 15 sprites sur la longueur, les objets doivent être générés aléatoirement à chaque ouverture de jeu. Celui-ci doit être enregistré dans un fichier.

-Le programme doit être standalone (dans un environnement virtuel), respecter les bonnes pratiques de la PEP 8, utiliser python 3 et rédiger en anglais.

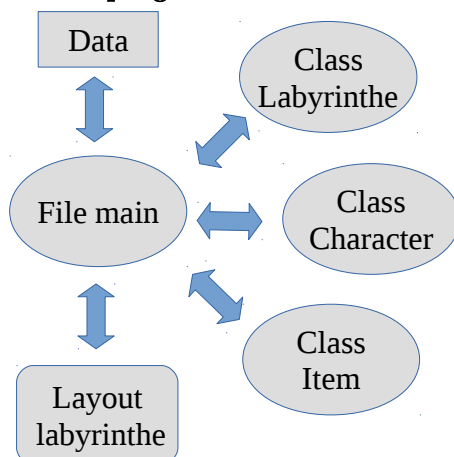
II-Conception du projet

a) Schématisation du labyrinthe:

J'ai commencé par imaginer un plan pour structurer le programme:



Gestion du programme:



Légende:

File main: gère les boucles et des chargement du jeu
Layout labyrinthe: plan lettré du labyrinthe
Class Labyrinth: charge les éléments du labyrinthe
Class Character: gère les déplacements de Macgyver
Class Item: attribut un algo pour le placement de l'objet sur le layout du labyrinthe
Data: images et sons

b) Préparer les ressources:

les images et les sons:

J'ai cherché des images sur internet pour chaque élément puis j'ai utilisé et découvert gimp pour les calibrer et les modifier à volonté. j'ai ensuite eu le plaisir de pouvoir utiliser audacity pour enregistrer des sons et les arranger. J'étais maintenant parée pour commencer à coder.

III-Réalisation du projet

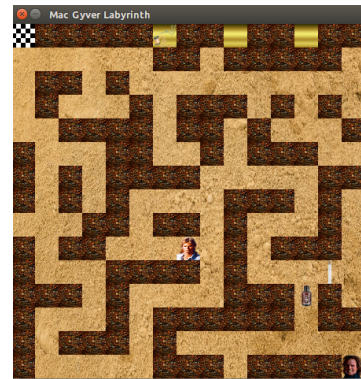
a) Plan du Labyrinthe:

```
1 | dmmmmmmmmmm
2 | 000000mmmmmm
3 | 0mm0m00000000
4 | 0m000m0mm0m0m
5 | 00mmmm0mm0m0m
6 | m00mm0m0mm0m0
7 | m0m0mmmm0000m
8 | m0m0m000mm0m0
9 | 000mm0mm0m0m0
10| m0mm0000m0mm0
11| m00mmmm0m0000
12| mmm0m000mm0m0
13| m000m0mmmmmmmm
14| m0mmmm0m000000
15| m0000mmmmmmma
16|
```

Plan structuré du labyrinthe

Légende :

- d : position de départ de Macgyver
- m : les murs du labyrinthe
- 0 : les chemins du labyrinthe
- i : l'inventaire des objets récupérer
- a : l'arrivée où se trouve Murdoc



Résultat après chargement des caractères en image

b) Chargement du labyrinthe :

Dans la `class Labyrinth` chaque élément du plan est chargé sur la fenêtre en 2D. Grâce à la fonction `enumerate` les images sont extraites dans un fichier référençant les constantes graphiques. Cette fonction respecte les principes d'itération du langage python. Elle permet en effet, de récupérer d'une manière efficace une liste de tuples en fonction du contenu de la liste. On itère ainsi directement sur les valeurs de sortie d'un générateur au lieu d'utiliser un indice. (comme on aurait pu le pratiquer dans un langage tel que le C). Cette `class` interagira en fonction des déplacements du personnage sur les objets pour les charger dans les cases de l'inventaire symbolisé par 'i'.

c) Déplacement du personnage :

Les déplacements de MacGyver sont gérés dans la `class Caractere`. Chaque mouvement haut, bas, droit et gauche est conditionné à tour de rôle par l'instruction `if`. Un booléen sur « short-circuit evaluation », c'est-à-dire sur une affirmation « False » permettra d'autoriser le déplacement si les cases sont différentes de 'm' et 'i'. De plus si notre héros arrive sur une case objet celle-ci obtiendra une nouvelle position pour être chargé dans la `class Labyrinth`.

d) Position aléatoire des objets :

La `class Item` est chargée de définir l'emplacement des objets de manière aléatoire. Pour cela j'importe le module `random` qui permet de générer facilement des positions tirées au sort. J'applique la fonction `randint` sur les abscisses et les ordonnées du plan lettré du labyrinthe. Ainsi elle génère une position au hasard. Si la case obtenue est '0' soit un chemin, l'objet obtiendra une position '01' pour l'aiguille '02' pour le tube en plastique et '03' pour le flacon d'éther. Sinon, si la position obtenue est différente de '0' alors il réitéra la fonction grâce à `pass` jusqu'à obtenir une case '0'.

e) Condition de Victoire :

Le déplacement de MacGyver sur un objet déclenche un compteur. Si le compteur est égal à trois alors la case où se trouve Murdoc devient 'v' et donc si notre héros se déplace face au gardien, il accède à la boucle de victoire. Si au contraire Mac Gyver n'a pas récupéré les trois objets l'attribution demeure 'a' et charge la boucle de Game Over.

IV-Les difficultés rencontrées

N'ayant jamais fait de programmation en python, j'ai mis un temps pour assimiler les compétences nécessaires pour débiter ce projet. Je ressentais une certaine frustration de ne pas pouvoir coder rapidement. Échafauder une organisation pour la réalisation et entrevoir une vision globale représente de réelles difficultés. Grâce au cours sur pygame, j'ai pu mieux imaginer comment concrétiser ce jeu. MacGyver est un personnage qui applique les sciences naturelles et d'ingénieur, pour combiner l'utilisation d'objets du quotidien avec ingéniosité. Il reste une source d'inspiration pour la résolution de ces problèmes !