



Aidez MacGyver à s'échapper !

Projet 3 : Parcours Développeur d'application - Python

31/01/2020 - Camille Clarret
<https://github.com/CamClrt/labyrinth>



Sommaire

Cahier des charges

Outils utilisés

Visuels

Structure générale du programme

Elaboration

Développement du projet en mode console et point clé

Développement du projet en mode graphique et points clés

Difficultés rencontrées et solutions trouvées

Enseignements et conclusion



Cahier des charges

Contexte

Concevoir un labyrinthe 2D dans lequel MacGyver aurait été enfermé et dont il doit s'échapper. Pour ce faire, il doit distraire le garde surveillant la sortie.

Il vous faudra alors **réunir les 3 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 le garde**.

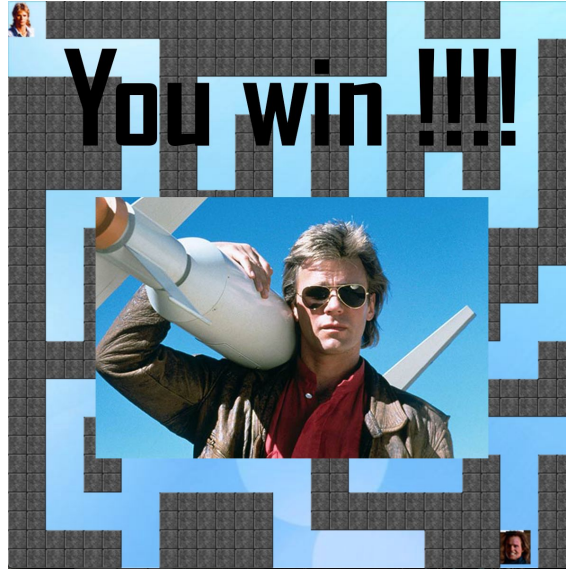
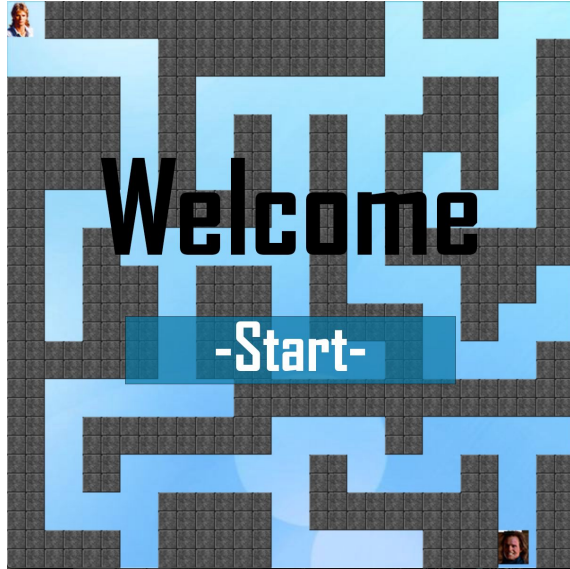
Fonctionnalités :


- **Un seul niveau**, la structure sera enregistrée **dans un fichier** pour une modification rapide.
- MacGyver sera **contrôlé par les touches directionnelles du clavier**.
- Les objets seront **répartis aléatoirement** dans le labyrinthe et ils **changeront d'emplacement à chaque fois que le jeu est relancé**.
- La fenêtre du jeu sera **un carré pouvant afficher 15 sprites sur la longueur**.
- MacGyver devra **se déplacer de case en case**.
- Il récupèrera un objet simplement **en se déplaçant dessus**.
- MacGyver gagne, s'il 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 et il perd.
- Le programme **sera standalone**, c'est-à-dire qu'il pourra être exécuté sur n'importe quel ordinateur.

Outils utilisés :



Visuels





Structure générale du programme

Others resources

map.txt

Graphics resources
folder

matrix_generator.xls

Python files

MAIN

Data

Game (1)

map_list
item_collected
item_tuple
x_wall
y_wall
x_item
y_item
home_page
game_page
end_page_win
end_page_lose

start_game()
init_game()
init_home_page()
init_game_page()
init_win_page()
init_lose_page()

Player (1)

name
x
y

go_right(*maplist)
go_left(*maplist)
go_down(*maplist)
go_up(*maplist)
get_position()

Map (1)

txt_file

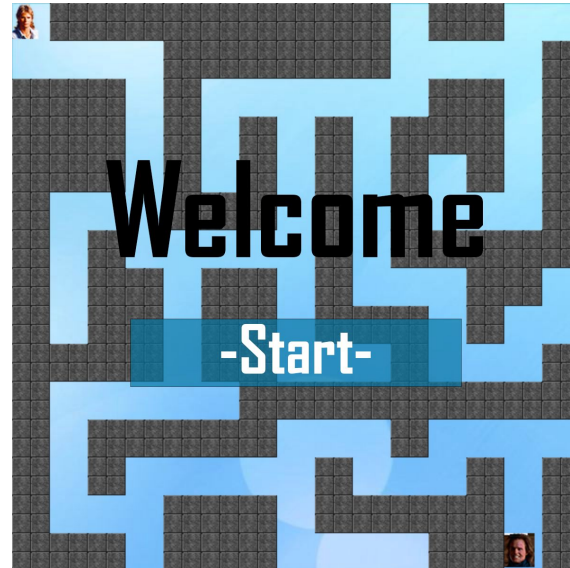
generate_maplist(*maplist)

Item (3)

url
position_x
position_y

put_items()
get_position()

2ème étape : mode graphique




Welcome

-Start-



Développement du projet en mode console

Actions concrètes :

- 
- > création de la map
 - > chargement et affichage de la map
 - > déplacement du joueur dans la map et affichage
 - > placement des 3 éléments aléatoirement
 - > collecte des 3 éléments
 - > détermination de si c'est gagné ou perdu

Programmation :

- > création des classes Map, Player, Item
- > création de data.py et de main.py
- > création des fonctions dans un fichier .py à part



Visuel

```
Welcome into labyrinth, what's your name ? : MacGyver
```

```
MacGyver : will you find the exit ?
```

```
Sxxxxxxxxxx0xx00
0000xxxxxx0000x
xxx0x000000xx0x
xxx0x0x0x0xxx0x
xxx0x0x0x0x0x0x
x000xxx0x0x000x
x0x00000x0xxxxx
x0xx0xx0x000xx0
x0xx0xx0xxx0x00
xxxx0xx000x000x
x00000xxxxxxxxx
x0xxxxx000x0000
x0x00000x000x0x
x000xxx0xxx0x0x
xxx0xxx0000xxFx
```

```
Before finding the exit, you have to collect 3 items located at : [(4, 6), (14, 11), (12, 1)]
```

```
Your actual position is (0, 0))
```

```
Which position do you want to take ? U = Up, D = Down, L = Left and R = Right : d
```

```
Your current position is (0, 1)
```

```
You have collected 0 items
```

```
Which position do you want to take ? U = Up, D = Down, L = Left and R = Right : |
```

mode console : point clé

Création de la map

P3 Labyrinth & générateur de matrice

Fichier

Édition

Afficher

Insertion

Format

Données

Outils

Modules complémentaires

Aide

Toutes les modifications ont été en

100%

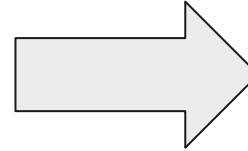
€ % .00 123

Montserrat

12

B I S A

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	px	0	40	80	120	160	200	240	280	320	360	400	440	480	520	560	600			
2		40	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14			Ligne x Colonne
3	0	0	S	X	X	X	X	X	X	X	X	X	X	o	X	X	o	o	\n	Sxxxxxxxxxxo\n
4	40	1	o	o	o	o	X	X	X	X	X	X	o	o	o	o	o	X	\n	oooooooooooo\n
5	80	2	X	X	X	o	X	o	o	o	o	o	o	X	X	o	X	\n	xxxxxxxxoooo\n	
6	120	3	X	X	X	o	X	o	X	o	X	o	X	X	o	X	o	X	\n	xxxxxxxoooo\n
7	160	4	X	X	X	o	X	o	X	o	X	o	X	o	X	o	X	\n	xxxxxxxxoooo\n	
8	200	5	X	o	o	o	X	X	X	o	X	o	X	o	o	o	X	\n	xxxxxxxxoooo\n	
9	240	6	X	o	X	o	o	o	o	o	X	o	X	X	X	X	X	\n	xxxxxxxxoooo\n	
10	280	7	X	o	X	X	o	X	X	o	X	o	o	o	X	X	o	\n	xxxxxxxxoooo\n	
11	320	8	X	o	X	X	o	X	X	o	X	X	X	o	X	o	o	\n	xxxxxxxxoooo\n	
12	360	9	X	X	X	X	o	X	o	X	o	X	o	o	o	o	X	\n	xxxxxxxxoooo\n	
13	400	10	X	o	o	o	o	o	X	X	X	X	X	X	X	X	X	\n	xxxxxxxxoooo\n	
14	440	11	X	o	X	X	X	X	X	o	o	o	X	o	o	o	o	\n	xxxxxxxxoooo\n	
15	480	12	X	o	X	o	o	o	o	o	X	o	o	o	X	o	X	\n	xxxxxxxxoooo\n	
16	520	13	X	o	o	o	X	X	X	o	X	X	X	X	X	o	X	\n	xxxxxxxxoooo\n	
17	560	14	X	X	X	o	X	X	X	o	o	o	o	X	x	F	X	\n	xxxxxxxxoooo\n	
18	600																		Sxxxxxxxxxxo\n	
19																			xxxxxxxxoooo	
20																				




map.txt

1	Sxxxxxxxxxxo
2	oooooooooooo
3	xxxoxxxxxxx
4	xxxoxxxxxxx
5	xxxoxxxxxxx
6	xooooxxxxxx
7	xoxxxxxxx
8	xoxxxxxxx
9	xoxxxxxxx
10	xxxxxxx
11	xooooxxxxxx
12	xoxxxxxxx
13	xoxxxxxxx
14	xooooxxxxxx
15	xxxoxxxxxxx



Adaptation du projet en mode graphique

Actions concrètes :

- 
- > installation et intégration au code de PyGame
 - > adaptation des classes Map, Player, Item
 - > allègement et rationalisation du code général
 - > suppression de function.py pour la classe Game
 - > rédaction du README.md
 - > création d'un package

Programmation :

- > affichage de la fenêtre de jeu, du fond et des murs
- > placement de MacGyver et du garde
- > déplacement de MacGyver dans le labyrinthe
- > placement des 3 éléments dans le labyrinthe
- > gestion de la collecte des 3 éléments par le joueur
- > ajout de la page d'accueil et de son bouton start
- > ajout des 2 pages de fin alternatives
- > packaging du programme

mode graphique : points clés

Reconstruction de la classe Map, Player, Item

Map (1)
txt_file
generate_maplist(*maplist)

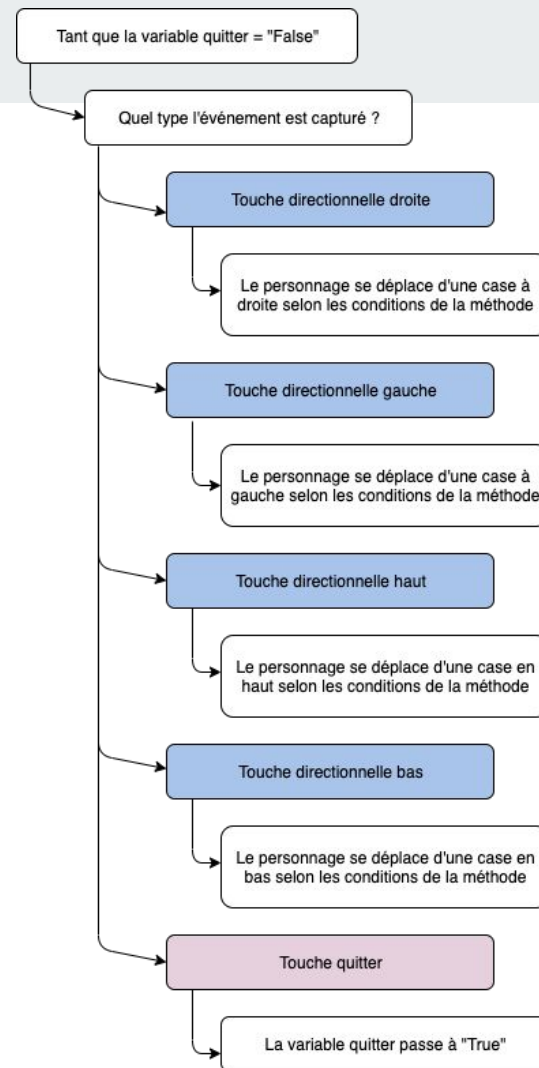
Player (1)
name x y
go_right(*maplist) go_left(*maplist) go_down(*maplist) go_up(*maplist) get_position()

Item (3)
url position_x position_y
put_items() get_position()

PyGame et affichage



PyGame et les événements



mode graphique : points clés

Suppression du fichier de fonctions et création de la classe Game

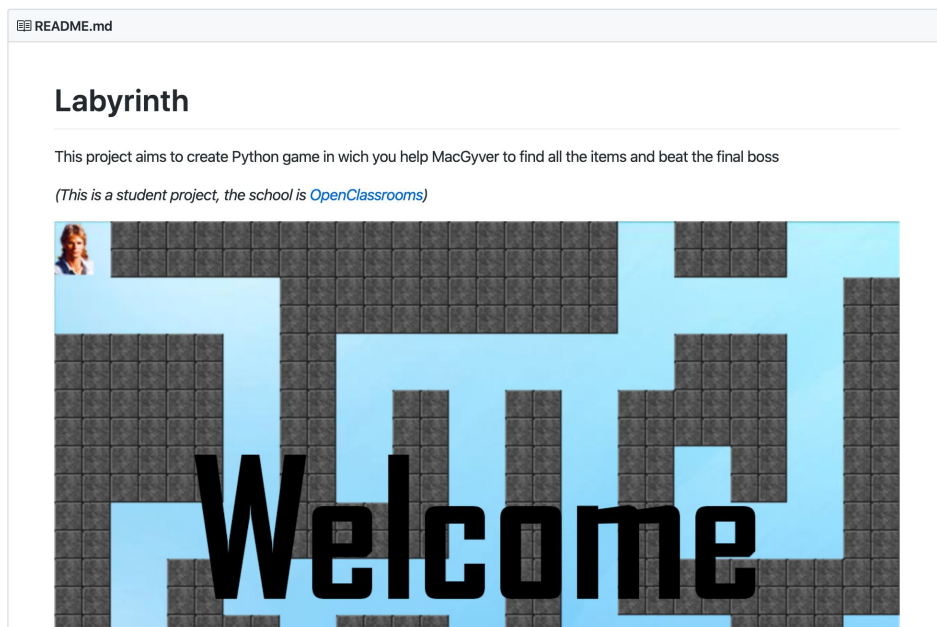


```
1 #!/usr/bin/env python
2 # coding: utf-8
3
4 from cam_labyrinth.Game import Game
5
6 if __name__ == "__main__":
7     game = Game()
8     # Start Labyrinth
9     game.start_game()
```

Game (1)
map_list item_collected item_tuple x_wall y_wall x_item y_item home_page game_page end_page_win end_page_lose
start_game() init_game() init_home_page() init_game_page() init_win_page() init_lose_page()

mode graphique : points clés

Redaction du README.md

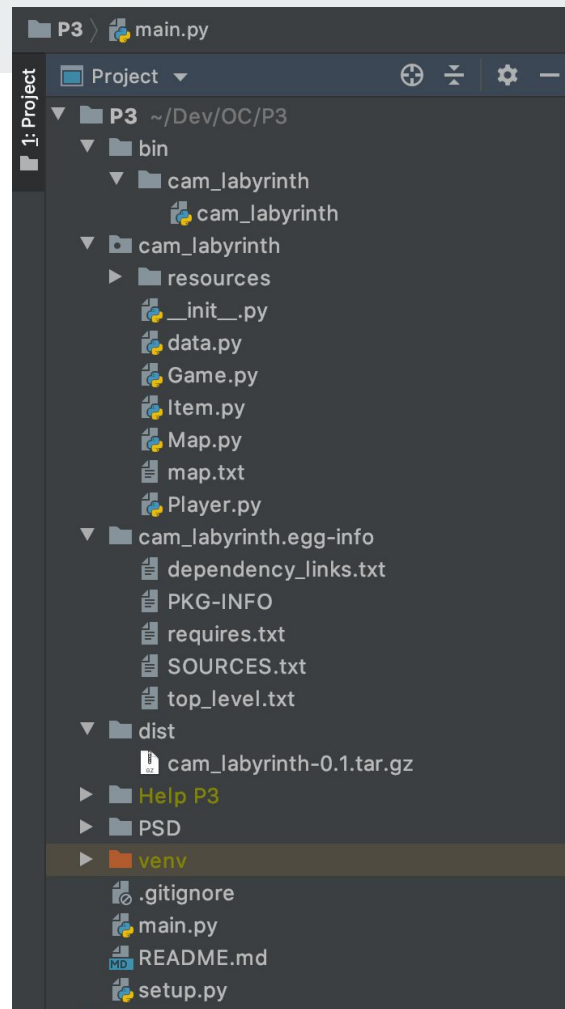


mode graphique : points clés

Finalisation du projet : packaging

Commande pour installer le package :

```
git clone https://github.com/CamClrt/labyrinth.git  
cd labyrinth  
pip install .
```





Difficultés rencontrées et solutions trouvées

> Démarrer le projet, comprendre la POO et l'appliquer dans un cadre concret.

Solution: acquérir les bases en Python en terminant le Mooc commencé avant de m'inscrire sur le parcours. Suivre les cours OC et les croiser à d'autres ressources trouvées sur Internet dont le cours de [Jason Champagne](#) et mêler le tout à la pratique par des exercices et TP.

> Installer PyGame sur mon Mac.

Solution: demander de l'aide auprès d'un pair.

> Vouloir mêler développement de la logique du jeu avec le développement de la surcouche graphique.

Solution: développer le jeu en mode console pour bien appréhender la POO (création de classe, attributs, méthodes) puis une fois le jeu en mode console finalisé, intégrer la partie graphique au code avec Pygame.



Conclusion

> **Acquérir de solide bases en Python** : forte du Mooc suivi avant de démarrer mon entrée sur ce parcours ainsi que de la réalisation de ce projet.

> **Appréhender la complexité et la rigueur que demande la programmation**. Cela passe par l'application d'une méthodologie simple mais précise : aborder le projet en petites étapes, réfléchir point par point à chacun des fonctionnalités, puis une fois réalisées revoir le projet dans son ensemble afin de s'assurer de sa cohérence.

> **Programmer ce n'est pas "que maîtriser un langage de programmation"**. Nous avons aussi besoin d'adopter son écosystème, outils de versionning (Git), plateforme collaborative pour les projets à plusieurs (Github), connaissance de sa propre machine et de l'utilisation des lignes de commandes etc.

> **Aller chercher de l'information et faire appel à la communauté**. Confronter les points de vue, les sources d'informations et s'adosser aux sources faisant références.



Merci de votre attention