

Développeur d'application Python : Projet 3

« Aidez MacGyver à sortir du labyrinthe »

Sommaire :

- Contexte
- Méthode
- Choix d'architecture
- Démonstration
- Rétrospective
- Conclusion

Contexte :

Je suis chef de cuisine, je travail a temps plein et j'effectue la formation OpenClassroom sur mon temps personnel.

Je prends donc plus de temps que la plupart des élèves pour réaliser les projets.

Je n'avais jamais fait de programmation avant de débiter cette formation.

Le but de ce projet est de réaliser un mini jeu vidéo avec Python et d'effectuer l'affichage graphique avec Pygame.

Méthode :

J'ai beaucoup fait de recherches sur internet (StackOverflow, Developpez, GeekForGeeks, Docs Python, cours OpenClassroom, cours Python de l'université de Paris Diderot...) pour tenter de comprendre le langage et sa mise en application, afin d'ébaucher l'architecture de mon programme. Ces sources et bien d'autres ont été d'une grande aide et parfois d'un grand secours.

Mon architecture logiciel a évolué au fur et a mesure du développement pour s'adapter aux contraintes et difficultés que j'ai rencontrées. Elle s'est nourrie des nombreuses manières de coder que j'ai pu voir, des nombreuses manières différentes d'aborder et d'expliquer les mêmes sujets.

Choix d'architecture :

J'ai débuté par la réalisation d'un diagramme de flux avec l'aide du site Draw.io. Ils se trouvent en pièce jointe du présent document.

Je me suis très vite aperçu qu'un seul fichier ne conviendrait pas pour réaliser mon projet. J'avais au moins besoin d'un fichier appelant les fonctions, les classes... et d'un autre ou celles-ci seraient déclarées. J'ai donc souhaité faire plusieurs fichiers .py pour les faire interagir entre eux. Au fur et a mesure du développement j'ai déplacé mes constantes préalablement déclarées dans les divers

fichiers dans un seul fichier.py dédié. Cela m'a permis d'y avoir accès dans chacun de mes fichiers en les importants ce qui a vraiment facilité le développement.

J'ai fait le choix d'avoir trois classes : « Game » où se déroulent les processus du jeu, « Maze » qui sert au chargement et à l'affichage du labyrinthe, et « Objet » qui gère les fonctionnalités liées aux objets. Ces classes sont appelées dans le fichier « Main » où se trouve la structure du programme.

J'ai donc en tout cinq fichier Python (Game, Maze, Object, Main et Variables), un fichier texte où se trouve le labyrinthe, et un dossier contenant les images pour l'affichage avec Pygame.

Démo :

Voici le lien vers mon projet, il suffit de lancer le fichier « Main ».

Le fonctionnement du jeu est expliqué dans la fenêtre d'accueil.

Lien GitHub :

https://github.com/OttaPowa/Project_3_Maze_J_C/tree/Maze-PyGame-Version/P3%20Collet%20J%C3%A9r%C3%A9my/Programme%20P3%20Collet%20J%C3%A9r%C3%A9my

Le programme est également en pièce jointe à ce document.

Rétrospective :

J'ai rencontré de très nombreuses difficultés lors du développement de mon programme. J'ai dû d'abord me familiariser avec Python, son environnement, sa syntaxe, son utilisation.

J'ai heureusement pu compter sur l'aide sans faille de mon tuteur pour me débloquer ou me mettre sur la voie que je cherchais à atteindre.

J'ai, au début beaucoup codé en « dur ». Même si j'avais l'intuition qu'il était possible de synthétiser mes fonctions pour les rendre plus élégantes, plus adaptatives, plus « simples » j'ai eu des difficultés à trouver comment le faire, à saisir la logique et encore plus à la mettre en pratique.

Lors du développement je me suis assez éparpillé, m'éloignant de mon diagramme, ne savant plus quoi coder, ni où le mettre dans mon programme, comment regrouper des fonctionnalités ou les séparées pour les appeler de manière distincte.

Cela se ressent le plus selon moi dans la classe Game avec une méthode de classe qui regroupe beaucoup de choses... j'aurais pu séparer la gestion des fins du jeu dans une autre fonction par exemple. Ou faire un constructeur pour instancier les directions dans le personnage de MacGyver. Et ainsi rendre plus simple le codage des déplacements.

J'aurais aussi pu regrouper mes classes dans le même fichier ce qui aurait permis de n'en avoir que trois au lieu de cinq : Main, Classes et Variables.

Le programme pourrait aussi être complexifié :

Il serait aussi envisageable par exemple de créer plusieurs niveaux ou encore d'introduire un garde supplémentaire se déplaçant à travers le labyrinthe et en ajoutant des cases « d'ombre » ou

MacGyver pourrait se réfugier pour ne pas se faire repérer par celui-ci. Il serait également possible de mettre l'unique garde en mouvement, la sortie serait alors verrouillée et le joueur devrait passer derrière le garde, après avoir récupéré les objets, pour l'endormir et ainsi lui prendre la clé de la porte et s'enfuir...

J'avais d'ailleurs, sur la version console de mon programme développé une fonction donnant le choix à l'utilisateur de ramasser l'objet ou non. Cela n'entrant pas dans les consignes du projet ou il est stipulé qu'un objet se ramasse seul lorsque le personnage se déplace dessus je l'ai abandonné dans la version finale.

Conclusion :

Au final j'ai énormément appris avec ce projet. Si je devais reprendre le développement depuis le début je crois que mon programme serait en grande partie différent. Même si les choix que j'ai fait sur l'architecture de mon code semblaient me correspondre, j'aurais envie tester d'autres manières de faire en utilisant l'expérience que j'ai acquise au cours de ces longs mois de développement, de ces difficultés surmontées, de ces heures passées à lire, à tester, à recommencer.

J'ai hâte de passer au projet 4 d'utiliser ce que j'ai appris et d'apprendre de nouvelles choses !