

Projet 3

Aidez MacGyver à s'échapper !

 <https://github.com/Mcflan-7/macgayver-labyrinth>

Introduction au projet :

J'ai effectué une première séance de mentorat avec Thierry, mon mentor, afin d'établir une base départ sur mon projet, il m'a fourni des conseils d'organisation pour bien débuter.

Réalisation du projet :

Le projet m'a demandé beaucoup d'implication, de recherche et d'apprentissage avec les cours de Céline, Vincent Le Goff et des cours indépendants sur Udemey.

Il fallait réussir à représenter non seulement une carte, mais faire travailler les classes entre-elle.

Les terminologies étaient difficiles à saisir, j'ai donc noté les définitions sur un tableau blanc en face de mon bureau pour bien les intégrer.

Avec l'aide de mon mentor j'ai consolidé ma compréhension des classes, comment les utiliser, pourquoi ces dernières à la place de fonction et j'ai commencé à coder progressivement sur mon projet.

Les classes :

J'ai représenté mon labyrinthe via 3 classes principales :

- Laby : Elle s'occupe de lire et représenter la carte, c'est elle qui gère tout ce qui se rapporte au fichier "map" ainsi que son fonctionnement.
- Hero : La classe "hero" gère le déplacement de MacGyver ainsi que son inventaire
- Item : Enfin "item" dispose sur la carte 3 objets à récupérer et les places de manière aléatoire à chaque initialisation du jeu.

L'interface graphique :

Une fois le code principal terminé, je me suis attaqué à la gestion de l'interface graphique avec Pygame, j'ai appris à gérer des sprites, récupérer les valeurs de ma carte les afficher à l'utilisateur etc.

C'est au moment de réaliser le jeu avec Pygame que l'on se rend compte de l'intérêt des classes, ainsi il est très facile de faire évoluer le jeu sans se perdre dans du code spaghetti

Le motif d'architecture :

J'ai choisi de représenter mon package avec une architecture modèle-vue-contrôleur.

Ayant déjà travaillé sur plusieurs projets Django, ce design me paraissait adapté à ce projet et il permet également de commencer à découper un projet proprement.

Les difficultés rencontrées :

Il y a eu une première difficulté évidente, les classes.

J'ai compris assez vite leur intérêt et leur façon d'être initialisée, mais quand il fallait faire travailler plusieurs classes en même temps, leur attribuer des méthodes, passer des paramètres, cela a été tout de suite plus compliqué.

Il a donc fallu réaliser un gros effort de consolidation sur ce sujet-là.

Thierry a pris son temps pour m'expliquer leur fonctionnement et leur spécificité

Un problème que je n'avais pas anticipé et qui m'a posé un problème tout au long du projet c'est la logique.

J'ai eu du mal à résoudre des problèmes assez simples par manque de logique et de méconnaissance des outils et modules Python.

Je participe maintenant assez régulièrement sur des applications de codage pour me perfectionner sur ce sujet.

Conclusion :

Je constate une nette progression dans ma relation vis-à-vis de Python.

J'ai pu apprendre à rédiger un code propre et réutilisable, j'ai appris des fonctions avancées de Python et j'ai surtout appris à préserver sur un projet qui n'était, ma foi, pas si facile que je ne l'avais cru.

Ce projet m'a fait apprécier encore plus le langage Python grâce à sa facilité de rédaction et de compréhension.

Gaëtan GROND