Kilian BOIVERT

Rapport Projet 1

Introduction :

Le but de ce projet était de produire un jeu de bataille sur Python. Il fallait utiliser quatre classes différentes pour coder le jeu.
Les règles du jeu de la bataille de notre projet sont les suivantes :

- Le jeu comprend soit 32, soit 52 cartes
- Le jeu est divisé en deux, une moitié pour chaque joueur
- A chaque tour, les joueurs opposent une de leurs cartes
- Le joueur ayant la carte de la plus grande valeur récupère sa carte et celle de l'adversaire
- Si les deux cartes ont la même valeur, alors il y a bataille : les joueurs posent une carte face cachée puis une carte face visible
- Si un joueur n'a plus de cartes, il a perdu

Pour lancer le jeu, il faut exécuter le fichier main.py, la bataille se joue automatiquement. Il y aussi un affichage plus agréable créé avec pygames mais il n'est pas indispensable.

Création des classes:

Personnellement, je me suis occupé des classes Joueur, Cartes, et JeuCarte. Pour faciliter la compréhension du code, toute la spécification a été faite. En bas des fichiers de chaque classes se trouvent des tests qui m'ont permis de vérifier mon code à chaque étape de la création

A - Classe Joueur

Ce n'était pas trop dur de créer le début de cette classe, en effet, les set et les get sont assez simple mais le plus dur était la méthode jouer_carte qui permet de renvoyer la carte en haut de la main du joueur. Si la main du joueur n'est pas vide, alors son nombre de carte diminue de 1 et on renvoi donc la première carte à l'aide du pop(), et s'il n'a plus de cartes, alors on renvoi un None. Clément m'a aidé pour le str en m'apprenant comment faire avec le f"…"

B- Classe Cartes

Cette classe ne m'a pas posé trop de difficultés. Les méthodes get et set sont simples, les méthodes eq, gt et lt étaient un peu plus compliquées mais pas trop dures non plus. Pour le str, il fallait séparer deux cas, étant donnée que le self.get_nom() renvoi le nom de la carte (As, Roi, 1 ou 2 par exemple), si la carte était par exemple le 10 de Cœur, le str renvoyait "1 de Cœur", donc si le nombre de la carte était 10, il fallait renvoyer self.get_nom() et non pas self.get_nom()[0], ce qui est le else de la méthode. Encore une fois, Clément m'a aidé sur le str et on a choisi de renvoyer les caractères des cartes, c'est à dire les piques, cœurs, carreaux et trèfles.

C – Classe JeuCarte

C'est la classe qui m'a posé le plus de difficultés, en effet, la méthode distribuer me permet uniquement de faire marcher la méthode distribuer_jeu. Il fallait donc faire un lien entre elles que je n'ai pas vu, c'est Clément qui m'a mis sur la voie en m'expliquant qu'il était plus simple de séparer la méthode en deux plutôt que de faire une méthode trop compliqué.

Kilian BOIVERT

Conclusion

J’ai trouvé ce projet vraiment intéressant et plaisant à faire, ni trop facile ni trop dur. Il m’a permis d’en apprendre beaucoup, notamment sur les str. Aussi, la communication avec Clément à été simple, on a fait plusieurs séance en appel sur Discord pour faire le point et ça nous a permis d’avancer, assez rapidement, sur les problèmes qu’on rencontrait.