

# Projet info

Bonjour, nous allons vous présenter notre projet de développement du jeu de Scrabble.

## Introduction

Nous avons utilisé les notions vues en cours mais également des fonctionnalités supplémentaires que nous avons apporté de nos propres connaissances ou d'internet.

Nous avons ainsi fait le code avec des classes comportant des fonctions permettant de faire fonctionner le jeu

Nous n'avons pas réussi à mettre en place l'affichage graphique avec les couleurs pour les cases bonus.

Notre jeu fonctionne ainsi avec l'interface textuelle uniquement.

Le code a été fait en anglais car il était plus facile pour moi de le faire en anglais.

## Expliquer les classes et les instances

### Partie 1 : Classe Board (/plateau)

Pour créer le plateau nous avons écrit une classe nommée Board qui signifie plateau en anglais. Nous avons affiché le plateau de manière textuelle, car nous n'avons pas réussi à afficher le plateau avec le module pygame.

Dans cette classe nous avons défini les cases bonus MT MD LT et LD, ainsi que les cases vides, ces dernières contenant une chaîne vide dans chaque cellule.

Nous avons fait une petite erreur de frappe dans la définition des cases bonus MD où nous avons écrit LT, et les cases LT où nous avons écrit WD.

Cette classe permet de changer l'affichage du plateau à chaque fois que l'on joue un mot, c'est-à-dire qu'il ajoute un mot sur le plateau après l'avoir placé.

### Partie 2 : Classe Jeton

Cette classe permet la création des jetons contenant les 98 lettres en majuscule et les deux jetons joker, ainsi que le score de chacun des jetons.

Cette classe contient les deux fonctions qui retournent la lettre du jeton et son score correspondant.

### Partie 3 : Classe Bag (/Pioche)

Cette classe crée la pioche avec tous les jetons disponibles pendant une partie du jeu, donc les 98 lettres et les 2 jokers.

Nous avons initialisé l'instance bag comme une liste qui contient tous les jetons ainsi que leurs valeurs. Nous avons utilisé des notions vues en cours pour supprimer les jetons utilisés de la liste, telle que la commande .pop qui permet de retirer un élément de la liste.

Cette classe contient des fonctions qui nous permet d'ajouter et des jetons au sac ou d'en enlever, afin de réapprovisionner la main du joueur à chaque tour.

En dehors des classes, nous avons écrit 3 fonctions.

La première fonction « `start_game` » démarre le jeu en demandant le nombre de joueur et vérifiant qu'il y a bien entre 2 et 4 joueur. Nous n'avons pas pu faire la partie intelligence artificielle permettant de jouer seul ou à plus de 4.

La deuxième fonction nommée « `turn` » permet à chaque joueur de jouer à son tour dans chaque partie.

La troisième fonction « `end_game` » termine le jeu lorsque le sac est à court de jetons. Elle demande aussi à l'utilisateur s'il veut rejouer.

## Partie 4 : Classe Rack

Elle permet de créer la main du joueur avec les 7 jetons, en ajoutant et supprimant les jetons afin de reconstituer le nombre de jetons dans leur main.

Tout d'abord, elle ajoute les 7 jetons dans la main du joueur, puis affiche la main du joueur. A chaque tour, elle retire les lettres jouées et en affiche de nouvelles présentes dans la pioche.

Une fonction présente dans la classe permet de s'assurer qu'il y a toujours 7 jetons dans la main du joueur.

## Partie 5 : Classe Word (/mot)

Cette classe a 5 instances :

- `word` (= mot)
- `player` (= joueur)
- `location` (= emplacement)
- `direction`
- `board` (= plateau)

Nous avons écrit une fonction « `check_word` » dans l'instance `word` afin de vérifier que le mot tapé par le joueur est bien présent dans le fichier `liste_mots` que vous nous avez fourni sur Caséine. Cependant, cette partie la du code n'a pas fonctionné, et notre code permet de jouer au Scrabble mais avec n'importe quel mot, en le plaçant ensuite sur le plateau. Cette fonction est supposée générer un message d'erreur si le mot ne se trouve pas dans la liste.

L'instance `direction` de cette fonction incite à le premier joueur à placer son mot à l'emplacement (7,7). Elle permet d'autre part de placer le mot horizontalement si l'on tape `right`, ou verticalement si l'on tape `down`.

La fonction vérifie également que les mots se superposent correctement. Si le mot n'est pas placé correctement le jeu affiche un message d'erreur invitant le joueur à choisir un autre mot.

Si un mot tapé par un joueur n'est pas lié à l'un des mots présents sur le plateau, le jeu affiche un autre message d'erreur disant « Veuillez lier le mot à une lettre précédemment jouée ».

Si l'emplacement du mot est en dehors des limites du plateau, le jeu indique que l'emplacement est hors limites.

Cette classe permet aussi de sauter notre tour si l'on a pas de jetons permettant de jouer, en tapant sur la touche espace à chaque question posée.

Elle invite le premier joueur à ne pas sauter le premier tour en affichant « Ne sautez pas votre premier tour. Entrez un mot. ».

Dans la suite de la partie, elle demande au joueur s'il veut vraiment sauter son tour, en lui demandant d'écrire « y » pour oui ou « n » pour non.

Si le joueur tape un mot sans avoir les jetons nécessaires pour ce mot, le jeu affiche « Vous n'avez pas de jetons pour ce mot »

Après chaque message d'erreur, le jeu redemande au joueur d'écrire un mot et redemande son emplacement afin de corriger ce qui a pu poser problème.

On a également écrit une fonction `valeur_mot` qui permet de calculer le score du mot joué, en tenant compte des cases bonus.

Nous avons écrit une fonction `set_word` qui permet de réécrire le mot tapé en minuscules par l'utilisateur en majuscule dans le plateau.

## Partie 6 : Class Player

Cette classe `player` = joueur permet d'obtenir les noms de chaque joueur pour lui permettre de jouer. Elle donne également des informations telles que le score ou la main du joueur.

Elle fait appel à plusieurs instances des classes définies tout au long du code afin de créer une instance pour chaque joueur.

## Conclusion

Pour pouvoir travailler en groupe malgré la crise sanitaire nous avons essayé de trouver des solutions telles que le travail en appel sur Discord. Nous avons mis en commun nos connaissances et avons effectué beaucoup de recherches afin de les mettre en pratique pour écrire le code.

Ce projet nous a permis d'enrichir nos connaissances en informatique tel que le module `pygame` qui nous était inconnu jusqu'à maintenant.

Il est également très intéressant pour nous de pouvoir mettre en place un jeu comme celui-ci avec Python.