TD K : Noé Le Yhuelic et Roméo Lhuillier

**DOCUMENT EXPLICATIF**

L’objectif du projet était de créer un jeu de mots mêlés avec plusieurs niveaux de difficulté et de plusieurs manières (plateaux générés aléatoirement et plateaux déjà existants sur des fichiers CVS).

Pour réaliser ce projet, nous avons décidé de créer 4 classes : Joueur, Plateau, Jeu et Dictionnaire.

Voici l’ordre chronologique de l’avancement de notre projet.

Dans un premier temps, nous avons réalisé la classe joueur qui est la plus simple. Cette classe va simplement décrire les joueurs tout au long de la partie par l’intermédiaire de leur nom, leur score et les listes de leurs mots trouvés. Nous avons utilisés une liste pour les mot trouvés car c’est beaucoup plus simple d’ajouter des éléments ou de vouloir en retirer.

Dans un second temps, nous avons crée la classe dictionnaire qui a été le début de l’utilisation des classes File et Stream et de la fonction .Split que l’on a utilisé de nombreuses fois durant ce projet. Dans cette classe, la difficulté a été de comprendre comment prendre les informations d’un fichier CVS et ensuite de réussir à les introduire dans chaque case de notre attribut (mots) qui est un tableau de tableau de string. Nous avons réalisé cela dans notre constructeur. Nous sommes partis sur un [][]string et non une matrice car il n’y a pas le même nombre de mot en fonction du nombre de lettre et non plus une liste de liste car nous avions directement repéré l’utilité de la fonction .Split().

Dans un troisième temps, nous avons créé la classe plateau, la classe jeu et le Main à peu près en même temps. Dans la classe plateau, nous avons utilisé une liste pour les mots à recherchés pour pouvoir retiré des mots de la liste au fur et à mesure qu’ils sont trouvés. Nous avons commencé par réaliser tout ce qui tournait autour de la création de plateau aléatoire. Notamment la fonction **Test\_Plateau** (Plateau) et **Créer\_Tableau** (Plateau). Cette 1ere fonction va retourner un booléen disant si oui ou non un mot peut être placer à une position en fonction de ligne et colonne. Cette 2e fonction va simplement créer le plateau en prenant à chaque fois une longueur de mot aléatoire pouvant rentrer dans le tableau, une direction aléatoire en fonction des différents niveaux de difficultés et enfin une ligne et une colonne aléatoires dépendant de ces 2 premiers paramètres. Suite à cela, nous avons créer la fonction **jouer** (Jeu) qui va faire jouer sur un seul niveau de difficulté tous les joueurs sur leur plateau respectifs. Puis nous avons créer le Main qui va demander toutes les informations utiles aux joueurs puis va les faire jouer aux 5 niveaux de difficulté. La partie principale de notre programme était terminé.

Nous avons eu une idée suite à cela : Nous avons créé un second mode de jeu (voir Main) qui permet de faire jouer tous les joueurs à un seul niveau de difficulté.

Dans un 4ème temps, nous nous sommes occupés de tout ce qui touchait à la sauvegarde des plateaux en utilisant la méthode **toFile** (Plateau) et à la création des plateaux en fonction de fichiers CVS déjà existants par l’intermédiaire de la fonction **toRead** (Plateau). Nous avons donc créé plusieurs plateaux (voir plateaux\_initiaux) et proposons aux joueurs de jouer avec ces plateaux ou bien de jouer avec certains crées aléatoirement (Main).

Pour finir, nous nous sommes lancés dans la sauvegarde totale d’une partie. Pour cela nous avons réfléchi comment faire et nous avons créé un fichier CVS nommé Enregistrement (Jeu) qui va, en utilisant la fonction **Enregistrement** (Jeu), sauvegarder toutes les informations nécessaires à la reprise d’une partie : la langue, le choix, le niveau de difficulté, le nombre de joueurs, le niveau, le chrono, l’index du dernier joueur ayant joué. Nous sauvegardons à chaque fois qu’une de ces informations changent. Ensuite, nous devions simplement demander à l’utilisateur s’il voulait reprendre la partie (cela n’est pas possible si la partie d’avant s’est terminée ou si le joueur n’a pas encore rentré toutes les informations nécessaires pour commencer une partie.) S’il répond oui, toutes les informations sont instanciées en fonction du document Sauvegarde.CVS qui a été créé dans la partie précédente (avant que le joueur quitte).