

## Projet Algorithmique : Document explicatif

Pour commencer notre projet nous avons créé les classes **Joueur** et **Dictionnaire**. La classe **Joueur** possède trois attributs : string name, int score, List<string>mots\_trouvés, et trois méthodes : toString() qui retourne les informations de notre joueur, une méthode Add\_Mots() qui ajoute le mot entré par le joueur dans sa liste de mots trouvés, une méthode Add\_Score() qui ajoute au score du joueur la taille du mot trouvé.

La classe **Dictionnaire** est composé des attributs langue et d'un tableau de list string. Chaque case de tableau est composé d'une liste en fonction de la taille qui est remplie grâce à la méthode readFile. On change de case de tableau avec des while et dès qu'un chiffre est lu on change de case. Pour la méthode dichotomique on a choisi de remettre les elements du tableau de liste dans une seule liste pour faciliter la manipulation. On va récupérer une liste réduite de moitié en taille en fonction de si le mot est plus petit ou plus grand que le mo

La classe **Plateau** possède trois attributs : une matrice de caractères char[,]plateau, un niveau de difficulté, un tableau de mots. Nous avons ensuite définit le plateau en fonction de la difficulté choisie. Nous nous sommes concentré sur la création d'une grille à partir d'un fichier.csv ; nous avons donc créé plusieurs grilles par difficulté, choisies aléatoirement. Avec une méthode Test\_Plateau, nous vérifions que le mot entré par le joueur est bien dans la grille. Nous avons commencé par verifier que les lignes et colonnes entrés par le joueur sont dans la grille puis nous verifions que le mot fait partie de la liste des mots à trouver, enfin nous vérifions la direction donnée.

Dans la classe **Program**, nous avons la mise en page et l'affichage du Jeu. Nous appelons la classe **Jeu** avec la méthode TourdeJeu qui prend en paramètre un joueur, la difficulté et la langue choisie. Nous avons pris en compte le cas où l'utilisateur rentre une langue non valide. Après avoir fini le tour de Jeu, le programme affiche qui est le vainqueur.

Dans la classe **Jeu** nous créons le plateau et le timer. Pour afficher le plateau nous utilisons la méthode toString(). Pour le timer nous avons utilisé la fonction TimeSpan. Puis nous demandons au joueur de rentrer le mot à chercher, la ligne et la colonne où le mot commence et enfin la direction (E,O,N,S,SE,SO,NE,NO). Puis nous vérifions si le mot est bien positionné avec la méthode Test\_Plateau de la classe **Plateau**.