

## **Jeu des mots glissés C#**

### **Objectif et fonctionnement du projet :**

Ce projet consiste à développer en C# un jeu de mots glissés, qui permet à deux joueurs de trouver des mots dans une matrice de lettres issue d'un fichier, ou bien générée de façon aléatoire. Ainsi, les joueurs proposent tour à tour des mots se trouvant sur le plateau, trouvés à la verticale, en horizontal ainsi qu'en diagonale. Lorsqu'un mot est validé, les lettres utilisées disparaissent et les colonnes contenant ces anciennes lettres glissent vers le bas.

Enfin, la partie se termine lorsque le temps est écoulé ou qu'il n'y a plus de lettres disponibles.

### **Structure du projet :**

Ce jeu repose sur une classe principale (Program) qui coordonne le fonctionnement de l'application, et qui délègue les différentes tâches à 4 autres classes :

- Jeu : Elle supervise le déroulement complet de la partie : la boucle de jeu, le chronomètre, l'alternance des tours et la coordination des joueurs.
- Plateau : Crée les lettres en fonction des règles définies, vérifie la validité des mots et permet le glissement des colonnes après chaque bon mot trouvé par un joueur.
- Dictionnaire : Cette classe s'occupe de charger et organiser les mots français, et vérifie qu'un mot proposé par un joueur existe bien dans le lexique.
- Joueur : Stocke en mémoire les caractéristiques des joueurs (leur score et la liste des mots qu'il ont déjà trouvés pendant la partie).

### **Gestion du dictionnaire :**

Nous avons organisé le dictionnaire sous la forme d'un tableau de listes (*List<List string>>*) afin de permettre une recherche plus rapide des mots.

Ainsi, les mots sont répartis en 26 sous listes en fonction de leurs premières lettres, allant de A à Z. Ceci réduit fortement l'espace de recherche d'un mot.

Ensuite, l'algorithme vérifie qu'un mot existe bel et bien grâce à une recherche dichotomique récursive. A chaque étape, l'intervalle de recherche est divisé par deux, ce qui garantit une validation quasiment instantanée malgré un grand nombre de mots.

### **Recherche sur le plateau :**

Pour que l'algorithme puisse valider qu'un existe bien sur le plateau, nous avons utilisé une méthode récursive. Pour se faire, la fonction part en bas du plateau au niveau de la première lettre, et vérifie tout autour de lui si la lettre suivante forme la combinaison ou le début de combinaison d'un mot. Si c'est le cas, le programme avance. Autrement, il se bloque et revient en arrière pour tester un autre chemin. Cette méthode est donc très fiable pour s'assurer qu'un mot existe dans la grille.

### **Expérience utilisateur et robustesse :**

Notre programme propose une flexibilité assez grande. En effet, il permet à l'utilisateur de choisir la taille du plateau, la durée de la partie, le temps par tour, et le nombre de joueurs lors du lancement.

De plus, notre projet ne crash pas. En effet, si un joueur essaie de charger un fichier plateau qui n'existe pas, alors le programme ne plantera pas, mais à la place il le signalera et générera aléatoirement un autre plateau à la place.

Enfin, le jeu se déroule en temps réel puisque lorsque le joueur saisit un mot avec son clavier, cela n'arrête pas le décompte du temps.

