



# Bilan de Travail

Projet JAVA | Mots Fléchés

**Yohan Davion**

## Table des matières

.....	1
1. Conception & Réalisation .....	3
2. Tests Unitaires .....	4
3. Difficultés rencontrés .....	5
4. Enseignement .....	5

## 1. Conception & Réalisation

Lors de la phase de conception, j'ai commencé par comprendre les contraintes et les exigences liées à la réalisation d'un jeu de mots fléchés. J'ai examiné en détail les fonctionnalités que le jeu devait offrir, notamment la possibilité de placer des mots dans la grille, de les retrouver à partir des définitions et de les remplir de manière aléatoire. J'ai également pris en compte la nécessité de gérer les interactions avec l'utilisateur pour lui permettre de saisir ses réponses.

Pour implémenter ces fonctionnalités, j'ai conçu une classe principale appelée "ModeleGrille" qui serait responsable de la gestion de la grille de mots fléchés. J'ai également identifié les différentes structures de données nécessaires, telles que les tableaux pour représenter la grille et les mots disponibles, ainsi que les listes pour stocker les définitions.

J'ai pris soin de gérer les contraintes relatives à la taille de la grille et des mots, en m'assurant que la grille pouvait accueillir les mots sans dépasser ses dimensions. J'ai également veillé à ce que les mots ajoutés dans la grille ne se chevauchent pas et ne sortent pas de la grille.

Pendant la réalisation, j'ai suivi les principes de conception orientée objet en décomposant le problème en différentes classes et méthodes, chacune ayant une responsabilité claire. J'ai utilisé des méthodes d'ajout, de recherche et de remplissage pour garantir un code modulaire et facilement extensible.

En comprenant les contraintes propres au jeu de mots fléchés, j'ai pu concevoir et réaliser une solution robuste et fonctionnelle. J'ai également pris en compte la convivialité de l'interface utilisateur en permettant à l'utilisateur de remplir les cases de la grille de manière interactive. Tout au long du processus, j'ai veillé à ce que la solution soit efficace en minimisant les opérations redondantes et en utilisant des structures de données appropriées.

## 2. Tests Unitaires

Les tests unitaires sont essentiels dans ce projet car ils permettent de vérifier la fonctionnalité du code, détecter rapidement les erreurs, faciliter le refactoring en toute sécurité, servir de documentation vivante, faciliter la collaboration entre les développeurs et valider les exigences du jeu de mots fléchés. Ils garantissent la qualité et la fiabilité du code tout au long du développement.

1. `TestInitialisationMotsDisponibles()`: Ce test vérifie si la liste des mots disponibles dans le modèle de la grille est correctement initialisée. On s'assure que cette liste est vide à l'initialisation. Cela permet de s'assurer que les mots disponibles sont bien chargés et prêts à être utilisés lors de la création des grilles de mots fléchés.
2. `TestInitialisationGrilleVirtuelle()`: Ce test vérifie si la grille virtuelle du modèle est correctement initialisée avec les dimensions spécifiées (5 lignes et 5 colonnes). Il s'agit de s'assurer que la structure de données qui représente la grille est prête à accueillir les mots et les lettres dans la grille.
3. `TestAjouterMot()`: Dans la classe "GrilleMotsFlechesTest", le test `testAjouterMot()` vérifie si la méthode `ajouterMot()` de la classe `GrilleMotsFleches` fonctionne correctement en ajoutant un mot à la grille. On vérifie si le mot est ajouté correctement aux coordonnées spécifiées avec l'orientation souhaitée. Cela garantit que les mots sont placés correctement dans la grille.
4. `TestInitialisationGrilleMotsFleches()`: Ce test vérifie si la classe `GrilleMotsFleches` est correctement initialisée avec les dimensions spécifiées, si les mots disponibles ont été chargés depuis le fichier et si la grille est correctement initialisée avec des cases vides. Ce test global s'assure que la classe `GrilleMotsFleches` fonctionne comme prévu dès le début.

### 3. Difficultés rencontrés

Durant ce projet plusieurs difficultés ont été rencontrées, tout d'abord la génération de la grille interactive n'a pas fonctionné malgré mes plusieurs essais, des erreurs NPE (Null Pointer Exception) surviennent par moment malgré les vérifications faites avant la lecture ou l'écriture de cases.

Des problèmes avec la librairie Wing (pourtant bien présent dans le JRE 17) m'a fait perdre beaucoup de temps sur la réalisation du projet.

Des problèmes avec la génération de la javadoc m'empêche de la générer.

Je pense que certains de ses problèmes s'explique par des problèmes de conception de ma part.

### 4. Enseignement

Malgré les difficultés j'ai pu monter en compétence sur la gestion d'interface graphique en JAVA, chose que je n'avais jamais fait dans ce langage, étant assez familier avec le JAVA car c'est le langage que j'utilise durant mon alternance j'ai pu correctement (je pense) mettre en place le modèle orienté objet dans la gestion des classes et ainsi que respecter le patron MVC.