

# Rapport Wordle

Groupe 3

**Mohamed Adnane**

**Maciej Budner**

**Mattéo Callo**

**Sofian Labrak**

**17/04/2025**

Licence Informatique

Ingénierie logiciel

NEAMS Projet IL

ECUE Nom de l'unité constitutive d'enseignement

**Responsables**

Aurelien Pelerin

Ludovic Bonnefoy

## Sommaire

Titre	1
Sommaire	2
1 Introduction	3
2 Groupe	3
2.1 Description générale	3
2.2 Planification des tâches	3
2.3 Liste des tâches individuelle	3
2.4 Travail personnel	5
2.4.1 Mohamed Adnane	5
2.4.2 Maciej Budner	5
2.4.3 Matteo Callo	6
2.4.4 Sofan Labrak	7
3 Retours du semestre	7

## 1 Introduction

Le but du projet est de créer le jeu Wordle, dans un groupe de quatre personnes. L'objectif est la mise en place des outils de production et l'élaboration d'un logiciel. Diverses tâches doivent être réparties au sein du groupe, en prenant compte des contraintes spécifiques qui devront être respectées.

## 2 Groupe

### 2.1 Description générale

L'entièreté du projet devra être déposé **sur notre gitLab**. De plus, un rapport devra être rédigé, détaillant l'avancement globale du projet, ainsi que le travail effectué et les difficultés rencontrées par chaque personne du groupe. Pour cela, nous avons à notre disposition le site Overleaf pour écrire notre rapport, ainsi que le site Trello pour la répartition des tâches.

### 2.2 Planification des tâches

Lors de notre seconde séance, nous avons pris le temps de planifier la réalisation de notre projet en découpant le projet en diverses tâches et en les attribuant à chacune des personnes de notre groupe à l'aide de **Trello**

### 2.3 Liste des tâches individuelle

Voici la répartition des tâches telle que planifiée sur Trello :

Tâches	Description	Personne affectée
Sauvegarder des informations relatives à la partie	Une fois la partie terminée, écrire les informations concernant cette partie dans un fichier data.txt	Mohamed Adnane
Système d'indice : Valider le placement des lettres	Colorier les cases des lettres selon si ces dernières : - sont bien placées - existent dans le mot, mais mal placées	Maciej Budner
Word2Vec	Appelle le script Python via un serveur HTTP (Flask). Transmet les informations du serveur flask vers le java	Matteo Callo
Afficher diverses informations au joueur via un Set-Text.	Remplacer "winText" par un texte global indiquant au joueur : - s'il a gagné - s'il a perdu - si le mot saisi n'existe pas dans le dictionnaire - s'il manque des lettres lors de la saisie	Matteo Callo

Victoire	Si le mot validé est le bon : - Arrêter la partie - Préciser au joueur qu'il a gagné	Mattéo Callo
Défaite	Si la dernière ligne est validée, mais que le mot n'est pas le bon : - Arrêter la partie - Préciser au joueur qu'il a perdu	Mohamed Adnane
Parseur	Créer un script permettant de récupérer les 1750 mots du dictionnaire fourni par Wiktionary (tous les langages de programmation sont permis).	Maciej Budner
Lancer le Wordle	- Affichage de l'interface - Déclaration d'une matrice pour stocker les lettres - Choisir aléatoirement un mot parmi les mots parsés - Laisser la main au joueur	Sofan Labrak
Ajouter et supprimer les lettres saisies	dans l'interface graphique, pouvoir enlever ou rajouter une lettre	Sofan Labrak
Interface du Wordle	Création de l'interface du Wordle, avec lequel le joueur pourra interagir	Sofan Labrak
Passer à la ligne suivante	Après la vérification, si le mot n'est pas bon : - la ligne actuelle doit être bloquée - passer à la ligne suivante	Mohamed Adnane
Vérification du mot	- Permettre au joueur de valider son choix - Vérifier si le mot est complet - Vérifier si le mot existe - S'il n'existe pas, le signaler au joueur - Sinon, passer à la validation des lettres	Mattéo Callo, Sofan Labrak

Remarque : cette liste n'est qu'une liste exhaustive qui nous a permis d'avoir un minimum d'organisation et ne démontre pas la totalité du travail de chacun. Certaines libertés ont été prises par chacun d'entre nous lors de l'avancement du projet, notamment dans des cas d'entraide, de correction du travail d'autrui, voire d'amélioration de fonctionnalités

## 2.4 Travail personnel

### 2.4.1 Mohamed Adnane

Lors du développement de mon projet, j'ai commencé par connecter l'application à une base de données pour afficher un tableau résumant les victoires et l'historique des parties. Bien que la récupération des données ait initialement rencontré des erreurs, des ajustements répétés ont permis de résoudre le problème.

Ensuite, j'ai travaillé sur la coloration des cases, avec des défis liés à l'effacement intempestif des couleurs et à une mauvaise sélection des indices, corrigés par une meilleure gestion des boucles.

J'ai également implémenté une logique de coloration conditionnelle des lettres : orange pour une lettre présente mais mal placée, verte pour une lettre bien placée et gris pour une lettre qui n'est pas dans le mot.

Cependant, la dernière case ne se colorait pas correctement, un problème réglé en ajustant les conditions de traitement.

Par ailleurs, j'ai intégré une gestion des entrées clavier, avec une difficulté liée à la touche "Entrée" fonctionnant sur Mac mais pas sur les ordinateurs de la Fac, résolue en utilisant la touche F1.

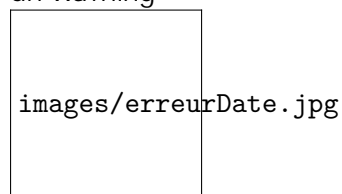
Enfin, j'ai ajouté une fonctionnalité permettant une réinitialisation automatique du parser après 30 jours, finalisant ainsi le projet avec succès malgré ces divers obstacles.

### 2.4.2 Maciej Budner

Le 15/10/2024, on a créé notre gitlab, Chacun est parti dans son boulot, j'ai travaillé sur le Parser. j'ai continué sur le parser chez moi

16/10, j'ai continué le parser pour le finir, cela m'a pris 3-4h pour le finir

23/10 j'ai passé 3h sur la mise à jour du fichier texte dictionary.txt car les fonctions de la variable Date étaient bizarres : sur eclipse, les fonctions étaient barrées et affichaient un warning



Mais l'affichage marchait très bien

donc je suis parti sur autre chose pour éviter tout problème, j'ai donc compté la différence de millisecondes entre aujourd'hui et la date de modification du fichier puis je le convertis en jour en divisant par  $1,157 \times 10^8$

25/10 j'ai fait une classe qui a pour but de prendre un mot dans le dictionnaire et de vérifier les mots saisis pour valider le placement des lettres. donc j'ai fait un système d'indices. Le code ne m'a pas pris beaucoup de temps car on avait déjà une idée de comment le réaliser. cela m'a pris 1h pour tout faire.

30/10 j'ai fait une fonction qui donne un indice sur le mot en donnant une définition. Cela m'a pris 2h, car j'ai dû chercher un bon site qui donnait des définitions sans dévoiler le mot à

deviner.

Cependant, l'indice demandé par la consigne était de donner des mots similaires à la réponse via Word2Vec, une tâche que nous avons affecté à Mattéo, mais nous avons tout de même gardé ma fonction parmi les fichiers du projet, au cas où nous souhaiterions l'ajouter comme fonctionnalité supplémentaire.

13/11 J'ai écrit un code java permettant de tester l'exécution d'un code bash et de récupérer les informations pour pouvoir lancer le code python utilisant Word2Vec. Cela m'a pris 1h pour pouvoir lancer la commande 'cat doc.txt' et afficher le nombre de ligne que je récupère. Le plus gros problème était de comprendre comment ProcessBuilder fonctionnait, mais mes efforts n'ont pas payé lors du test avec le programme python, qui a malheureusement échoué. Nous avons donc essayé de trouver une autre méthode, ce dont Mattéo s'est chargé de faire.

19/11 Nous nous sommes essentiellement entraînés. J'ai voulu aider les autres dans leur travail en leur proposant des solutions ou en cherchant la source du problème. Une fois fini, j'ai commencé à écrire le ReadMe en respectant le modèle proposé par GitLab.

### 2.4.3 Matteo Callo

La première chose que j'ai faite dans le cadre de ce projet a été de commencer à découper le projet en tâches simples à assigner aux membres de mon groupe sur Trello, permettant ainsi d'avoir un minimum d'organisation au sein du groupe.

La prochaine tâche que j'ai effectué était d'afficher des messages au joueur correspondant la saisie de son mot, c'est à dire s'il a gagné, perdu, si le mot est inférieur à 5 lettres ou encore s'il n'est pas dans notre dictionnaire. Sofan ayant déjà fait l'interface utilisateur, ainsi que quelques fonctions dont notamment la vérification de la victoire, j'ai repris cette dernière fonction pour y ajouter tous les cas de figure cités précédemment, créant ainsi la fonction de vérification de la validité du mot saisi. J'ai repris la balise "winText" créée par Sofan qui avait pour seul but à ce moment d'afficher le message de victoire. Désormais, j'ai fait en sorte que cette balise affiche en plus le message de défaite, et les messages concernant l'invalidité du mot saisi.

Ensuite vient la tâche qui fût la plus intéressante à mon avis, le système d'indice via Word2Vec avec l'utilisation de la librairie Gensim. Maciej avait déjà fait son propre système d'indice, mais nous avons été informé qu'il ne s'agissait pas de ce qui était demandé dans la consigne.

Gensim étant une librairie Python, cela signifiait qu'il fallait exécuter un code Python depuis du code Java puis récupérer le retour.

J'ai accepté cette tâche car j'ai déjà fait un programme similaire lors d'un projet de BTS, où j'effectuais en Java une requête vers un serveur PHP puis récupérais le retour grâce la librairie org.json.

Pour accomplir cette tâche, j'ai d'abord écrit un programme python avec Word2Vec permettant de trouver des mots similaires au mot entré en paramètre (qui, dans le cas du projet, sera la réponse). Ces mots similaires servent d'indices pour l'utilisateur. Avec mon groupe, nous nous sommes mis d'accord pour afficher trois mots comme indices, mon programme choisit donc 3 mots au hasard parmi les dix proposés par la fonction de Word2Vec. J'ai pour cela téléchargé différents modèles Word2Vec et les ai comparés en testant mon

programme afin de choisir celui qui proposait les mots similaires les plus cohérents par rapport avec le mot entré en paramètre.

Ensuite, il fallait lancer ce programme et récupérer les indices retournés avec du code Java. Maciej a voulu m'aider en recherchant un moyen d'exécuter du code Python en Java, et j'ai donc commencé à m'appuyer sur une fonction qu'il avait créée. Cependant, nous avons été informé qu'il fallait créer un serveur Flask, rendant cette fonction hors sujet. J'ai donc installé Flask, créé un fichier `app.py` ou j'ai adapté mon programme python à la structure de Flask : le paramètre est désormais reçu via la méthode GET et le retour sera l'affichage des indices sous la syntaxe d'un JSON sur la page web. L'appel de cette requête via Java se fait par la classe "Hints.java" que j'ai créée. Elle possède une fonction statique `getHints()` qui envoie une requête HTTP avec la réponse dans le lien (pour la récupérer via méthode GET), puis lit et renvoie la chaîne de caractères affichée sur la page (le JSON contenant les trois indices, ou un message d'erreur si aucun indice a été trouvé).

Par la suite, j'ai modifié le fichier `Game.fxml` (l'interface du jeu) pour implémenter un bouton permettant à l'utilisateur d'obtenir les indices. J'ai profité de ce moment pour redimensionner des éléments de l'interface, car cette dernière avait une plus grande longueur que la dimension de l'écran de mon ordinateur personnel, ce qui posait un problème d'accessibilité.

Enfin, il me restait plus qu'à modifier la classe `GameController.java`, qui gère le déroulement du jeu, pour implémenter la fonction `Hints.getHints()` et permettre le système d'indices dans les parties de Wordle, concluant ma tâche.

Par ailleurs, j'ai aussi contribué à ce projet en corrigeant des fautes (orthographe ou écriture de code) et en assistant les membres de mon groupe tout au long de ce projet.

#### 2.4.4 Sofian Labrak

Bien que Sofan ait quitté le groupe dû à son alternance, il a fortement contribué au début du projet en créant l'interface du jeu avec JavaFX, la classe `GameController.java`, ainsi que diverses fonctionnalités telles que l'utilisation du clavier virtuel.

### 3 Retours du semestre

Pour conclure, nous avons pu réussir à produire une application Wordle fonctionnelle, malgré les difficultés et les mécompréhensions de certaines consignes que nous avons pu surmonter. Ce projet nous a permis d'acquérir une meilleure coordination au sein d'un groupe de travail, par exemple en s'efforçant à faire en sorte que la répartition des tâches soit équitable pour chaque membre de notre groupe. De ce fait, nous avons pu produire un travail qui récompense les efforts fournis par chacun, tout en enrichissant nos compétences à travers les diverses solutions que nous avons dû apporter face aux divers défis imposés par la réalisation de ce projet.