



TELECOM Nancy

Projet Pluridisciplinaire d'Informatique Intégrative 2

WORDLE

Mathilde BARD
Lucie BOUCHER
Rémi BOURDAIS
Lola MONTIGNIER

Responsables de module :
Olivier FESTOR
Gérald OSTER
Sébastien DA SILVA



8 juin 2022

Déclaration sur l'honneur de non-plagiat

Nous soussignés,

Nom, prénom : **BARD** Mathilde - **BOUCHER** Lucie - **BOURDAIS** Rémi-MONTIGNIER Lola

Élèves-ingénieurs régulièrement inscrits en 1^e année à TELECOM Nancy

Année universitaire : 2021-2022

Auteurs du document, mémoire, rapport ou code informatique intitulé :

Projet de programmation et d'informatique intégrative

Par la présente, nous déclarons nous être informés sur les différentes formes de plagiat existantes et sur les techniques et normes de citation et référence.

Nous déclarons en outre que le travail rendu est un travail original, issu de notre réflexion, et qu'il a été rédigé entièrement par nos soins. Nous affirmons n'avoir ni contrefait, ni falsifié, ni copié tout ou partie de l'œuvre d'autrui, en particulier texte ou code informatique, dans le but de nous l'accaparer.

Nous certifions donc que toutes formulations, idées, recherches, raisonnements, analyses, programmes, schémas ou autre créations, figurant dans le document et empruntés à un tiers, sont clairement signalés comme tels, selon les usages en vigueur.

Nous sommes conscients que le fait de ne pas citer une source ou de ne pas la citer clairement et complètement est constitutif de plagiat, que le plagiat est considéré comme une faute grave au sein de l'Université, et qu'en cas de manquement aux règles en la matière, nous encourrons des poursuites non seulement devant la commission de discipline de l'établissement mais également devant les tribunaux de la République Française.

Fait à Nancy, le 18 Mai 2022

Signatures :

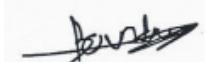
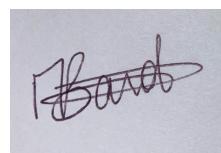


Table des matières

1	Introduction	4
2	Gestion de Projet	5
2.1	Équipe Projet	5
2.1.1	Composition du groupe	5
2.1.2	Organisation du groupe	5
2.1.3	Organisation des réunions	5
2.1.4	Finalités	5
2.2	Diagrammes de Gantt	6
2.2.1	Diagramme de Gantt prévisionnel	7
2.2.2	Diagramme de Gantt réel	8
2.3	Matrice SWOT	9
2.4	Work Breakdown Structure	9
3	Notre Application	10
3.1	État de l'art du Wordle	10
3.1.1	Présentation de Wordle	10
3.1.2	Actualités	10
3.1.3	Comparaison entre différentes applications	10
3.2	Présentation Générale	10
3.3	Présentation page par page	12
3.3.1	Mock-up	12
3.3.2	Page d'Inscription	12
3.3.3	Page de Connexion	12
3.3.4	Page de Configuration	12
3.3.5	Page des Règles du jeu	12
3.3.6	Page de Jeu	13
3.3.7	Page Profil	13
3.3.8	Page Historique	13
3.3.9	Page du Classement	13
3.4	Notre base de Données	14
3.4.1	Diagramme entité-association	14
3.4.2	Modèle relationnel	14
3.5	Notre algorithme de coloration	14
3.5.1	Problématique	14
3.5.2	Solution algorithmique	15
3.5.3	Les tests et les performances	15
3.6	Points d'améliorations	15
4	Notre Solveur	17
4.1	État de l'art du Solveur	17
4.1.1	Présentation du Solveur	17
4.1.2	Comparaison entre des solveurs existants	17
4.2	Présentation Générale	18
4.3	Première version du Solveur	19
4.3.1	Structure de données	19
4.3.2	Algorithme d'épuration du dictionnaire	19
4.3.3	Choix de la meilleure proposition	19
4.4	Deuxième version du Solveur	19
4.4.1	Structure de données	19
4.4.2	Algorithme d'épuration du graphe	21
4.4.3	Choix de la meilleure proposition	22
4.5	Les tests et les performances	26
4.5.1	Les tests	26

4.5.2	Les performances de nos algorithmes et leur complexité (Partie Solveur)	26
4.6	Points d'améliorations	27
5	Difficultés rencontrées	28
5.1	Difficultés lors de la phase Application Web	28
5.2	Difficultés lors de la phase Solveur	28
6	Bilan du Projet	29
6.1	Conclusion	29
6.2	Conclusions Personnelles	30
6.2.1	Mathilde BARD	30
6.2.2	Lucie BOUCHER	30
6.2.3	Rémi BOURDAIS	30
6.2.4	Lola MONTIGNIER	31
6.3	Travail réalisé	31
7	Annexes	33
7.1	Comptes Rendus de Réunions	33
7.1.1	Réunion du 18 Mars	34
7.1.2	Réunion du 24 Mars	36
7.1.3	Réunion du 27 Mars	38
7.1.4	Réunion du 3 Avril	40
7.1.5	Réunion du 7 Avril	42
7.1.6	Réunion du 18 Avril	44
7.1.7	Réunion du 25 Avril	46
7.1.8	Réunion du 28 Avril	48
7.1.9	Réunion du 8 Mai	50
7.1.10	Réunion du 14 Mai	51
7.1.11	Réunion du 18 Mai	53
7.1.12	Réunion du 22 Mai	55
7.1.13	Réunion du 26 Mai	57
7.1.14	Réunion du 28 Mai	59
7.1.15	Réunion du 29 Mai	61
7.2	Illustrations de l'application web	63
7.2.1	Page d'Inscription	63
7.2.2	Page de Connexion	63
7.2.3	Page de Configuration	64
7.2.4	Page des Règles du Jeu	64
7.2.5	Page de Jeu	65
7.2.6	Page Profil	65
7.2.7	Page Historique	66
7.2.8	Page du Classement	66

Chapitre 1

Introduction

Wordle est un jeu de devinette où, tous les jours, un mot de 5 lettres doit être deviné en 6 essais. Des indices sont donnés à chaque essai : les lettres bien placées sont colorées en vert, les lettres mal placées sont colorées en jaune et celles qui ne sont pas présentes dans le mot n'ont pas de couleur.

Cette application a été créé par Josh Wardle en 2019, puis a récemment été racheté par le célèbre journal américain The New York Times. Ce jeu, à l'origine, créé aux États-Unis a été repris dans le monde entier. Il existe, à l'heure actuelle, de nombreuses variantes du jeu en France.

Nous avons décidé dans le cadre du projet, de créer un Wordle où il y aura un mot à deviner chaque jour. De plus, notre Wordle pourra être modulable : le nombre de lettres et d'essais pourront être modifiés.

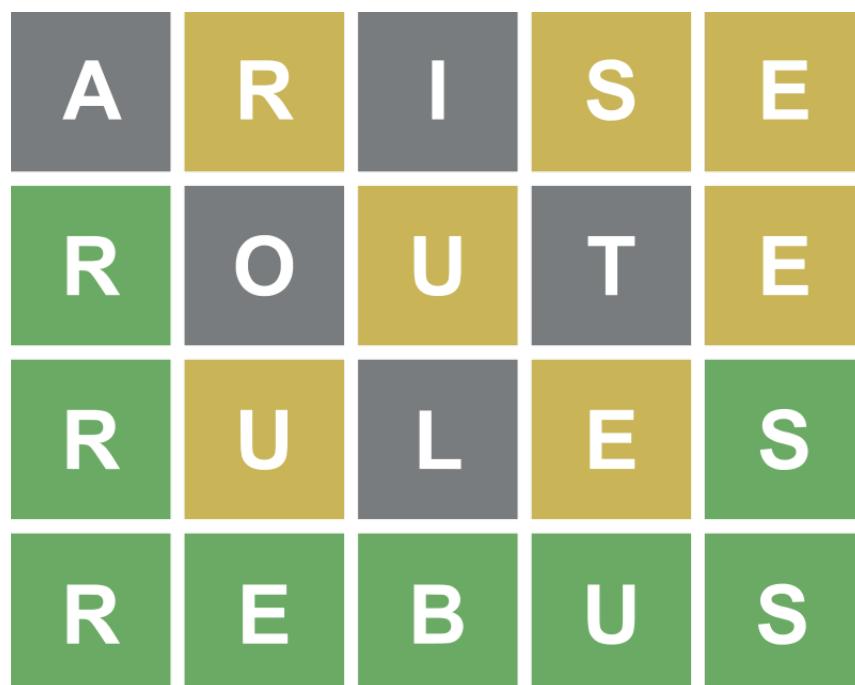


FIGURE 1.1: Exemple d'une partie sur l'interface Wordle

Chapitre 2

Gestion de Projet

2.1 Équipe Projet

2.1.1 Composition du groupe

Le groupe est constitué de quatre personnes :

Mathilde BARD,
Lucie BOUCHER,
Rémi BOURDAIS,
Lola MONTIGNIER.

2.1.2 Organisation du groupe

Afin de nous organiser au mieux, lors de notre réunion de lancement nous avons décidé de désigner un chef de projet, Lola MONTIGNIER. Elle avait pour mission d'assurer la communication au sein de l'équipe, de déterminer les jalons de travail à atteindre, de préparer les ordres du jour des réunions et également de diriger ces dernières.

De plus, nous avons nommé une secrétaire pour toute la durée du projet, Lucie BOUCHER, afin que des notes soient prises pendant les réunions et pour plus d'organisation. Lorsque les membres du groupe étaient à distance, la communication se faisait via un groupe de messages. Lorsqu'un membre avançait sur le projet, ou se posait une question, il prévenait les autres. La communication était de rigueur afin que chacun n'empêche pas sur le travail des autres, qu'un problème ne le reste pas trop longtemps et donc de préserver un bon rythme d'avancement sur le projet.

2.1.3 Organisation des réunions

Le chef de projet préparait l'ordre du jour chaque veille de réunion et l'envoyait aux autres membres du groupe afin que chacun puisse y rajouter un point si voulu. Notre secrétaire prenait des notes à chaque réunion et se faisait relayer par un autre membre du groupe quand elle prenait la parole.

A chaque réunion hebdomadaire, un retour sur la TODO liste de la précédente réunion était effectué afin de voir au mieux l'avancement de chacun. Tour à tour, nous évoquions nos difficultés, s'il y en avait. Par la suite, nous discutions des prochaines étapes à accomplir afin d'avancer au mieux dans le projet.

Enfin, chacun des membres du groupe se voyait affecter des tâches à réaliser pour la prochaine réunion. Les réunions se tenaient environ une fois par semaine, et une fois tous les 2/3 jours à la fin du projet. Elles se déroulaient soit sur Nancy soit sur Messenger.

Les différents comptes rendus de réunion sont en annexe à partir de la page ??.

2.1.4 Finalités

L'objectif de ce projet respecte la méthode SMART :

- Spécifique : L'objectif est spécifique car il est clairement détaillé dans le sujet.

- Mesurable : Les résultats de ce projet sont mesurables du fait que l'application Worldle fonctionne, ainsi que le solveur et que ces deux parties respectent les objectifs énoncés par le sujet.
- Accepté : L'objectif est atteignable car ayant déjà été réalisé par le passé et suivi par les encadrants du projet.
- Réaliste : Le projet est réalisable car toutes les connaissances nécessaires à sa réalisation sont à notre disposition. Il est également ambitieux, car aucun membre de l'équipe n'a déjà réalisé ce genre de projet auparavant et demande une importante quantité de travail.
- Temps : Le projet possède une date limite fixée au 13/06/2022 à laquelle nous devrons présenter notre travail. Le rapport devra avoir été rendu le 08/06/2022.

2.2 Diagrammes de Gantt

Au lancement du projet, nous avons décidé de réaliser un diagramme de Gantt prévisionnel afin d'anticiper les différents jalons à atteindre et pour s'organiser dans le temps afin que leur date limite de réalisation soit atteinte. Afin de visualiser notre avancement, nous avons produit un diagramme de Gantt réel que nous complétons au fur et à mesure. Dans son ensemble, le Gantt prévisionnel a été respecté à l'exception de certaines tâches où nous avons dû retarder les jalons.

2.2.1 Diagramme de Gantt prévisionnel

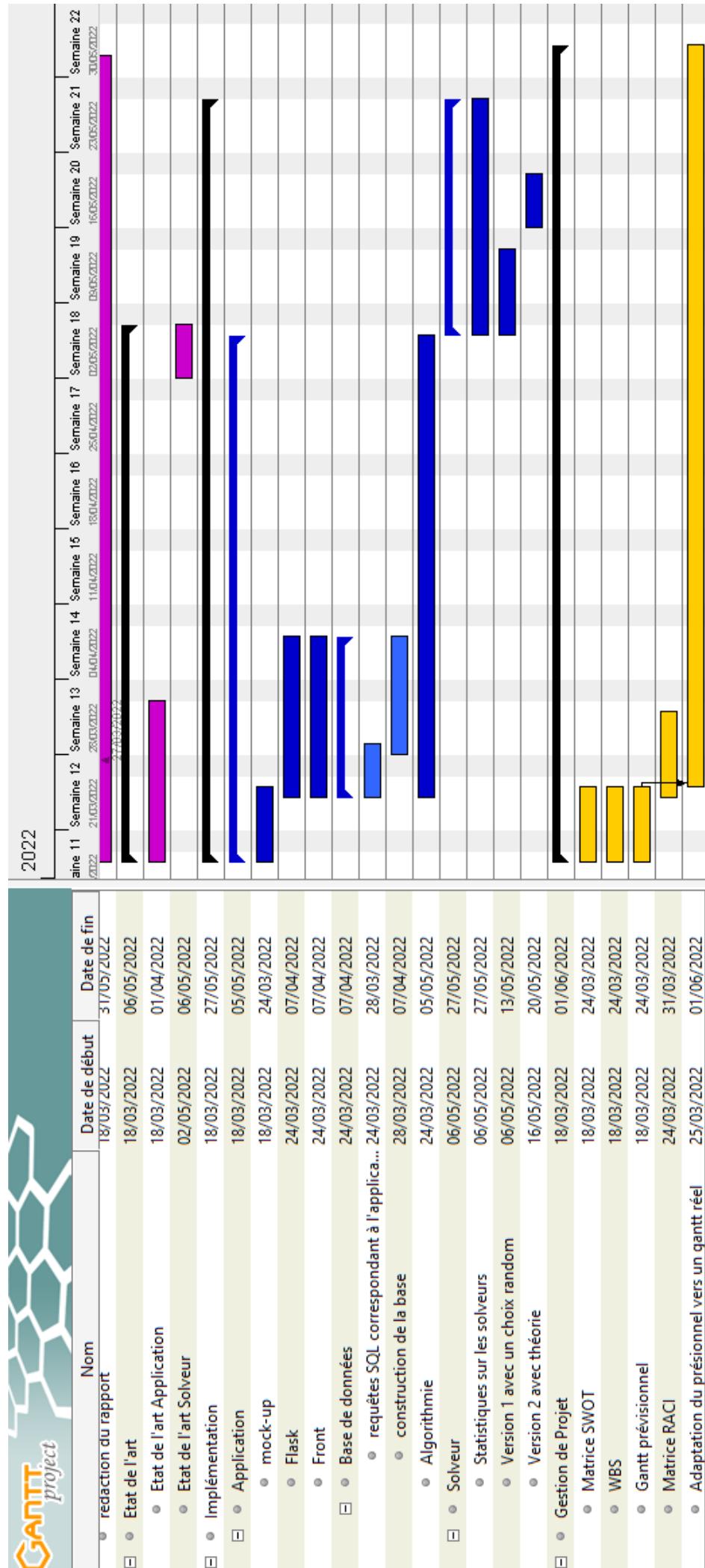


FIGURE 2.1: Diagramme de Gantt Prévisionnel

2.2.2 Diagramme de Gantt réel

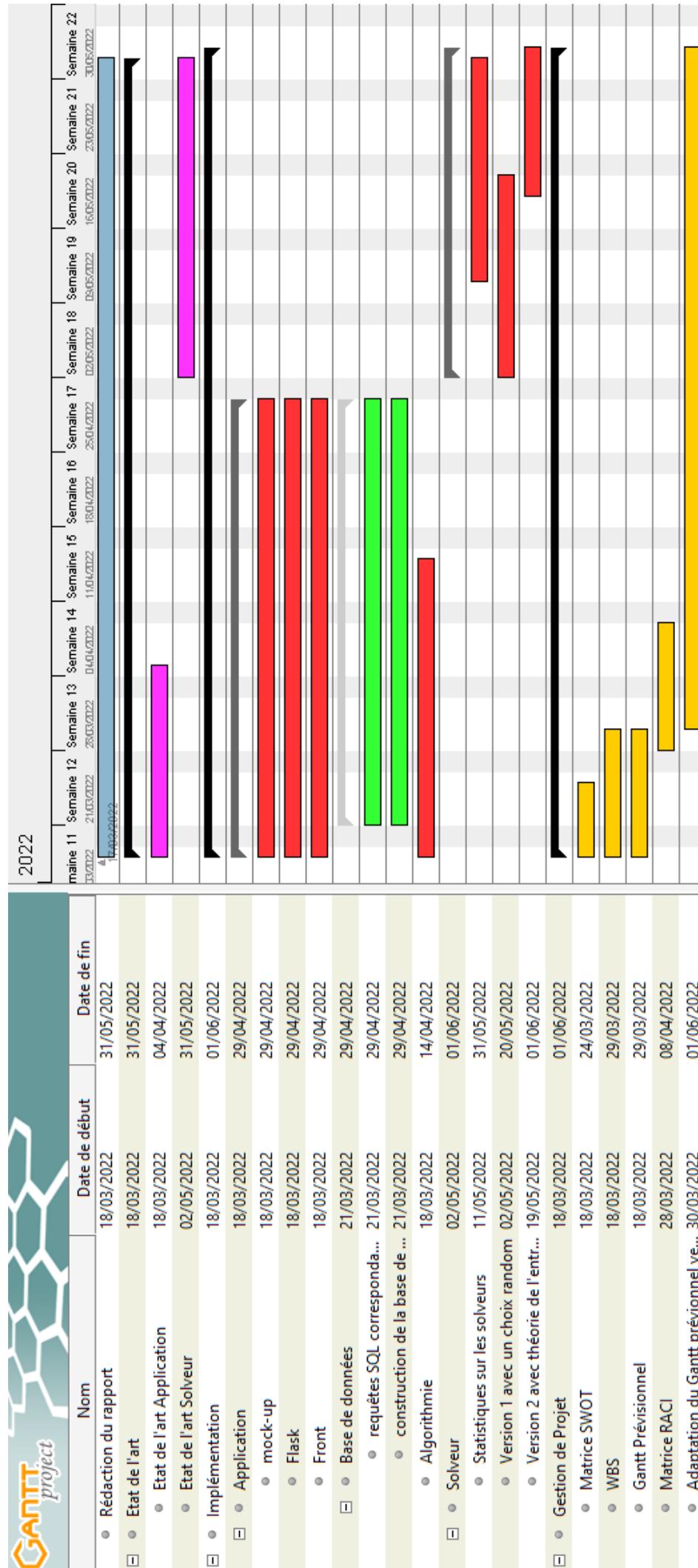


FIGURE 2.2: Diagramme de Gantt Réel

2.3 Matrice SWOT



FIGURE 2.3: Matrice SWOT du projet PP2I

2.4 Work Breakdown Structure

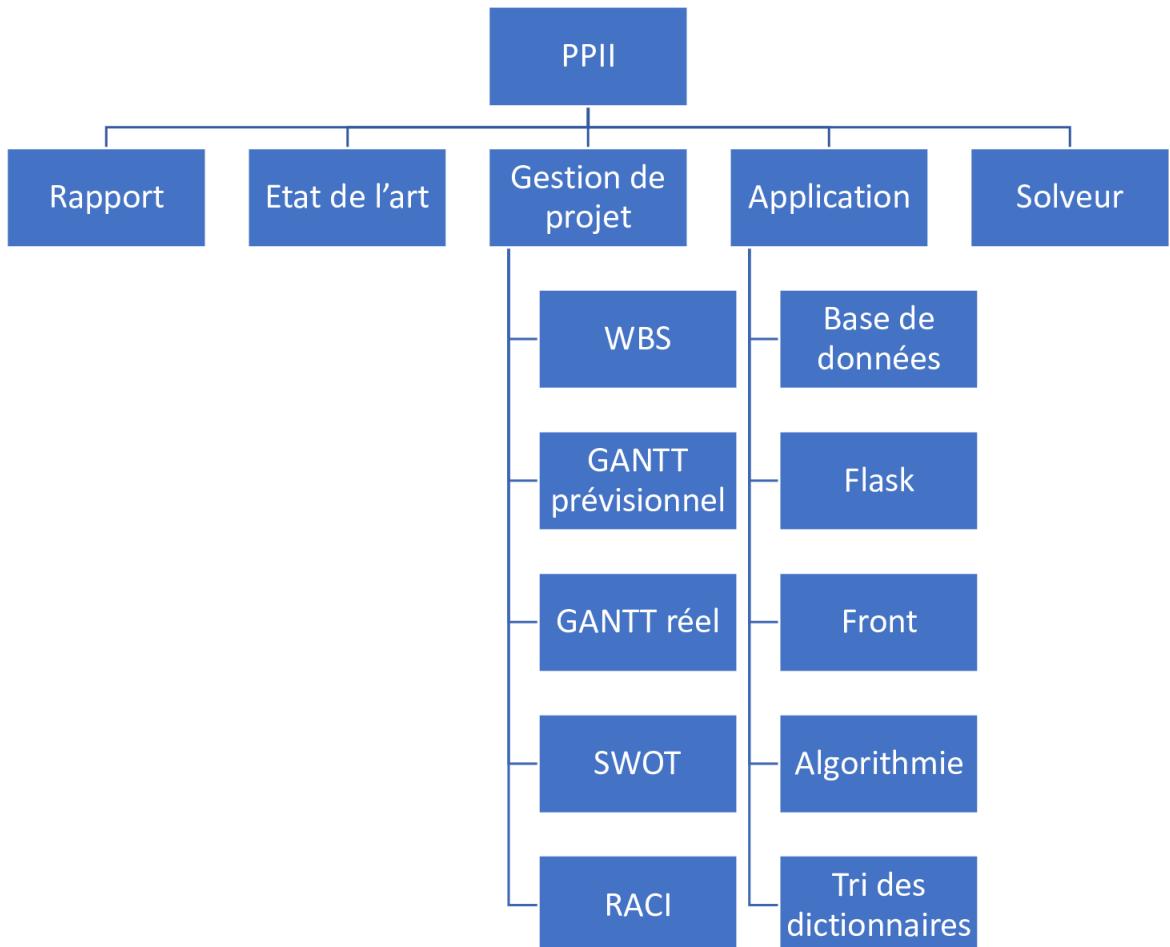


FIGURE 2.4: WBS du projet PP2I

Chapitre 3

Notre Application

3.1 État de l'art du Wordle

3.1.1 Présentation de Wordle

WORDLE est un jeu en ligne consistant à deviner un mot par jour. Pour cela, le joueur dispose seulement de la taille du mot. A chaque tentative, il obtient des indications : si une lettre est dans le mot recherché mais pas au bon endroit, elle se colore en jaune et si une lettre est bien placée, elle se colore en vert.

3.1.2 Actualités

Le jeu Wordle fait beaucoup parler de lui depuis ces dernières semaines. D'une part, le jeu officiel a été racheté par le célèbre journal The New York Times. D'autre part, de nombreuses variantes du jeu voient le jour. Le développeur du Sutom, une des variantes françaises du jeu, a déclaré le jeudi 24 Mars devoir fermer son site suite à des pressions du groupe France Télévisions car ressemblant excessivement, par son code couleur et son principe, au jeu télévisé Motus. Un revirement de situation a eu lieu grâce aux nombreux utilisateurs qui se sont indignés sur les réseaux sociaux que leur jeu préféré soit supprimé.

3.1.3 Comparaison entre différentes applications

Application	Mode Multi-joueur	Nombre de mots à deviner	Nombre d'essais maximum	Indices	Statistiques	Langues	Niveau de difficulté
Sutom	Non	1	6	Oui : première lettre	Oui	Français	Non
WordleGame	Oui	mode infini ou mode mot du jour	6	Au choix : nombre d'occurrences d'une lettre	Oui	Au choix	Oui
Tusmo	Oui	1 ou une suite de mots	6	Non	Oui	Français ou Anglais	Non
Nerdle	Non	1	6	Non	Non	Opération mathématique	Oui
Wordle	Non	1	6	Non	Oui	Anglais	Oui

3.2 Présentation Générale

L'objectif de notre application Wordle est d'offrir deux modes de jeu : le premier mode est le mode classique du Wordle, avec un mot de 5 lettres à deviner en 6 essais maximum chaque jour. Le deuxième mode, un mode où l'on peut configurer sa partie en choisissant le nombre d'essai maximal et la taille du mot à deviner ainsi que la langue et le niveau de difficulté :

- Facile : la première et dernière lettre du mot sont données.
- Moyen : seule la première lettre est donnée.

- Difficile : aucun indice n'est donné.

Notre application insiste sur le côté compétitif du Wordle en proposant des classements en fonction du nombre de mots trouvés dans les différentes catégories de Jeu : Difficile, Moyen, Facile, Mot du jour. Chaque utilisateur peut donc se connecter à l'aide d'un mot de passe qui est stocké de manière cryptée avec un clé césar de 5. Il peut voir ses informations ainsi que ses classements dans l'onglet : Mes informations.

Notre application propose aussi une page explicative des règles du jeu pour les joueurs débutants de façon à ce qu'ils comprennent le but du jeu et ainsi qu'ils puissent en profiter pleinement. Un historique des parties est également disponible permettant à chaque utilisateur de visualiser chaque partie qu'il a déjà jouée. Cet historique lui permet de voir les mots qu'il a tentés durant la partie et si sa partie était victorieuse ou non.

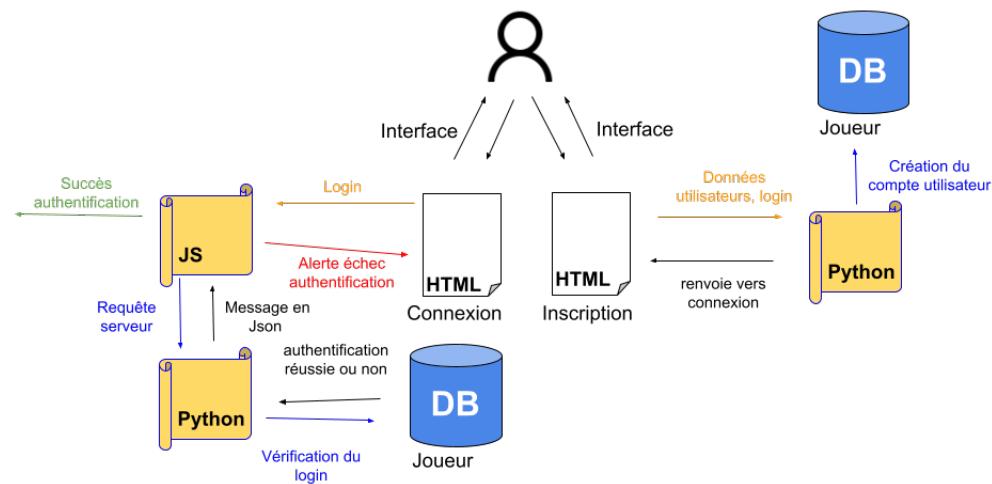


FIGURE 3.1: Schéma de conception des interactions partie connexion

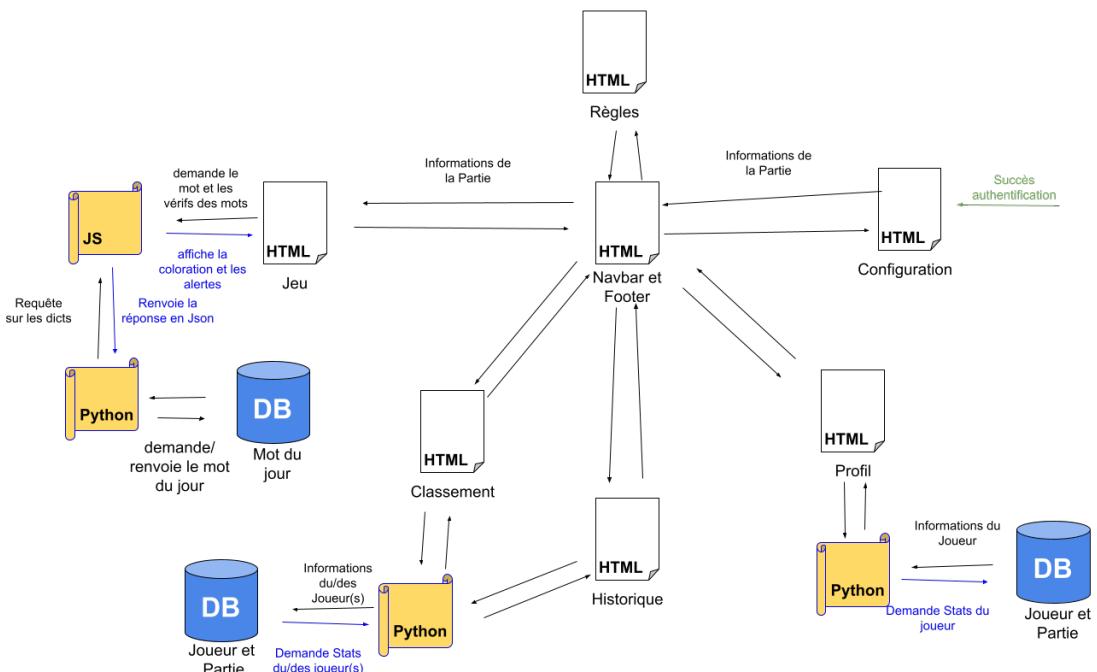


FIGURE 3.2: Schéma de conception des interactions partie jeu

3.3 Présentation page par page

3.3.1 Mock-up

Pour notre application, nous avons décidé de faire un visuel sobre, avec l'essentiel des fonctionnalités qu'une application peut posséder. Le joueur peut :

- Se connecter ou créer un compte.
- Consulter les règles du jeu avant de commencer une partie.
- Consulter l'historique de ses parties.

L'application est très simple d'usage, et a une prise en mains rapide.

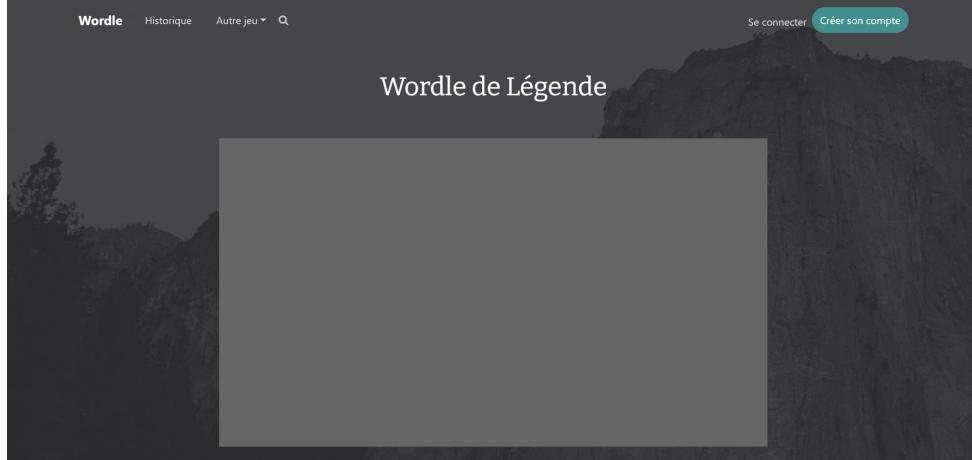


FIGURE 3.3: Mock-up de notre application Wordle

3.3.2 Page d'Inscription

La page d'Inscription va permettre aux nouveaux utilisateurs du site de se créer un compte, c'est-à-dire de renseigner leurs différentes informations (pseudonyme, adresse mail et mot de passe) qui seront alors enregistrées dans la base de données. Une fois inscrits, les utilisateurs pourront ainsi se connecter et donc accéder à la page de Configuration.
(Voir une illustration en annexe page 63)

3.3.3 Page de Connexion

La page de Connexion a pour but de diriger les utilisateurs s'étant précédemment inscrits vers la page de Configuration. Les utilisateurs doivent, pour y accéder, renseigner leur pseudonyme et leur mot de passe. Elle permet aussi de se diriger vers la page d'Inscription si la personne ne possède pas encore de compte.
(Voir une illustration en annexe page 63)

3.3.4 Page de Configuration

Sur la page de Configuration, les joueurs peuvent choisir les paramètres de leur partie. D'une part, ils peuvent jouer au mot du jour : deviner le mot du jour de 5 lettres en 6 essais maximum. D'autre part, les joueurs peuvent choisir le mode libre. Ils pourront alors choisir le niveau de difficulté de leur partie (facile, moyen ou difficile), le nombre d'essais et de lettres du mot à deviner, et la langue (Français ou Anglais).
(Voir une illustration en annexe page 64)

3.3.5 Page des Règles du jeu

Sur la page des Règles du jeu, les joueurs peuvent retrouver les règles qui régissent le jeu. Une explication des différents niveaux est fournie :

- Facile : la première et dernière lettre sont données au joueur tout au long de la partie.
- Moyen : la première lettre est indiquée au joueur.
- Difficile : le joueur ne connaît aucune lettre.

Le code couleur des lettres après un essai dans la partie est donné :

- Lettre en Rouge : la lettre est bien placée dans le mot.
- Lettre en Jaune : la lettre est mal placée dans le mot mais présente.
- Lettre non colorée : la lettre n'est pas dans le mot.

(Voir une illustration en annexe page 4)

3.3.6 Page de Jeu

Après que le joueur ait choisi ses paramètres de jeu sur la page de Configuration, il accède à la page de Jeu. Ainsi, le joueur se retrouve sur une page où il y a une grille de la taille du nombre de lettres du mot multiplié par le nombre d'essais. Le joueur rentre un essai puis le soumet à l'application, le mot est coloré selon les règles du jeu. Le mot soumis doit faire parti du dictionnaire utilisé par l'application sinon un message d'erreur apparaît. La partie se termine si le mot à deviner est trouvé ou si un trop grand nombre d'essais a été soumis sans avoir trouvé le mot.

(Voir une illustration en annexe page 65)

3.3.7 Page Profil

Sur la page Profil, chaque utilisateur peut consulter son pseudonyme, son mot de passe, et son nombre de parties jouées. Cependant, son mot de passe demeure caché sauf s'il clique sur un bouton qui permet de l'afficher en clair. Il peut également visualiser son classement dans les différents mode de jeu.

(Voir une illustration en annexe page 65)

3.3.8 Page Historique

La page Historique permet au joueur de consulter toutes ses anciennes parties jouées. L'historique contient les informations suivantes :

- La date de la partie
- Le mot à deviner
- Le nombre d'essais réalisés par rapport au nombre d'essais autorisés
- Le mode de jeu (mot du jour ou libre)
- Le niveau de difficulté (facile, moyen ou difficile)
- La langue (Français ou Anglais)
- Victoire ou Défaite

(Voir une illustration en annexe page 66)

3.3.9 Page du Classement

Une fois connecté, le joueur peut consulter ses différents classements dans la page Classement. Il existe 4 pages de Classement :

- Une page pour le mode mot du jour
- Une page pour le mode libre en niveau facile
- Une page pour le mode libre en niveau moyen
- Une page pour le mode libre en niveau difficile

(Voir une illustration en annexe page 66)

3.4 Notre base de Données

3.4.1 Diagramme entité-association

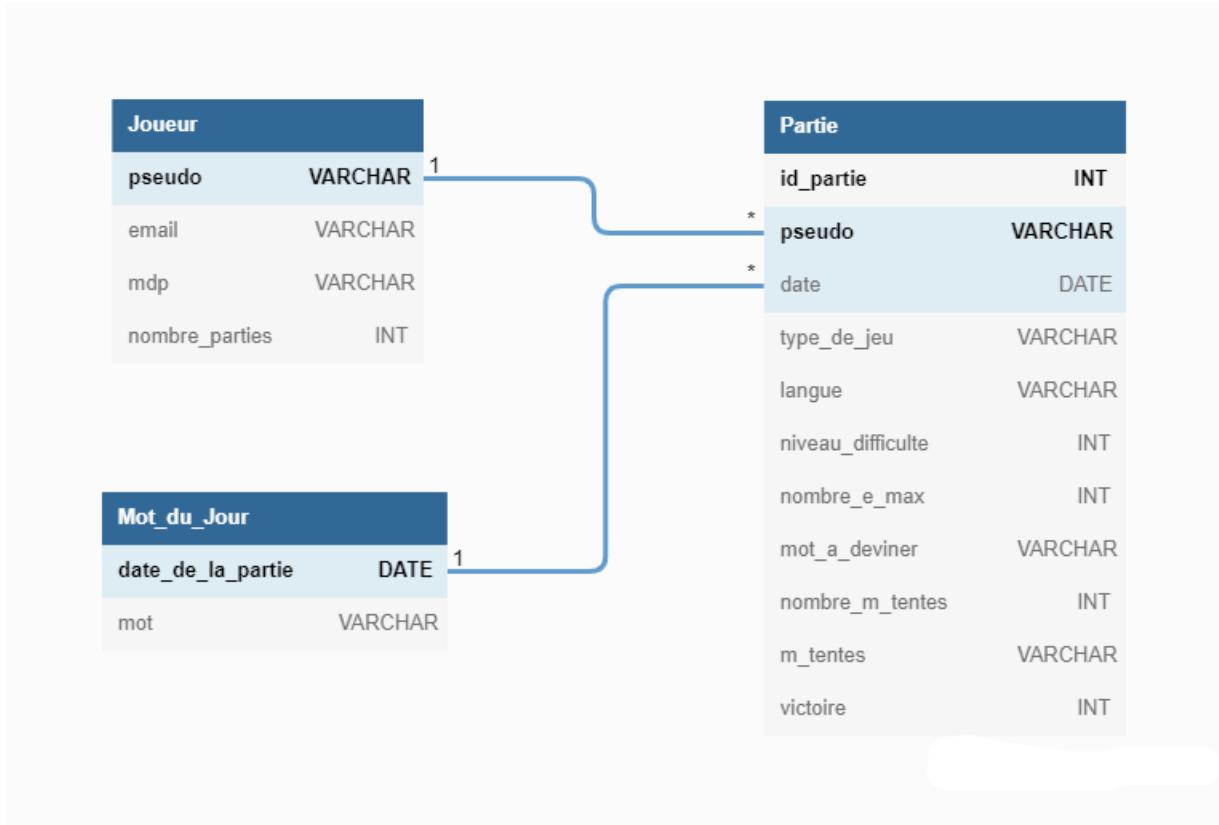


FIGURE 3.4: Diagramme Entité-Association de notre base de données

3.4.2 Modèle relationnel

— : clé primaire

: clé étrangère

Partie(id_partie, pseudo, date, type_de_jeu, langue, niveau_difficulte, nombre_e_max, mot_a_deviner, nombre_m_tentes, m_tentes, victoire)

Joueur(pseudo, email, mdp, nombre_parties)

Mot_du_Jour(date_de_la_partie, mot)

FIGURE 3.5: Modèle relationnel de notre base de données

3.5 Notre algorithme de coloration

3.5.1 Problématique

Le jeu Wordle repose sur l'algorithme dit de "coloration". En effet, l'algorithme permettant de colorier, une case en rouge si la lettre est bien placée, en orange si la lettre est dans le mot mais mal placée et sans couleur si elle ne fait pas partie du mot, est central au jeu. Cet algorithme de coloration doit ainsi comparer deux mots : le mot à deviner et le mot tenté par l'utilisateur. Cet algorithme doit renvoyer une liste de la taille du mot contenant à la position i , 2 si la lettre i du mot essayé est bien placée, 1 si elle est mal placée et 0 si elle n'est pas dans le mot.

3.5.2 Solution algorithmique

Première version

La première version de cet algorithme comparait simplement les lettres du mot tenté avec celle du mot à deviner. Cependant, cette version produisait une coloration erronée dans le cas où le mot tenté contenait deux fois la même lettre. Prenons le cas où le mot à deviner serait PATES et le mot tenté serait PASSE, cette version aurait renvoyé (2,2,1,1,1), suggérant à l'utilisateur que la lettre S apparaît deux fois dans le mot à deviner, ce qui est faux.

Deuxième version

Cette version va comparer le nombre d'occurrences des lettres présentes dans le mot à deviner et dans le mot tenté. Si l'occurrence d'une lettre est supérieure dans le mot tenté que dans le mot à deviner, l'algorithme va colorier cette lettre autant de fois que le nombre d'occurrences de la lettre dans le mot à deviner. Prenons le cas où le mot à deviner est AFIFT et le mot tenté par l'utilisateur est AFFEF. Dans le mot à deviner, la lettre F apparaît deux fois alors que dans le mot tenté cette lettre apparaît trois fois. L'algorithme va donc colorier en jaune ou en rouge que deux des trois F présents dans le mot tenté. L'algorithme va d'abord vérifier si un ou plusieurs des F sont à la bonne place, si c'est le cas, il va colorier la ou les cases correspondantes en rouge. Puis, si seulement une seule case sur les deux a été colorié, l'algorithme va colorier la première occurrence de la lettre non coloriée en jaune. Si aucun des F n'est à la bonne place, l'algorithme va colorier les deux premières de la lettre F en jaune dans le mot tenté.

3.5.3 Les tests et les performances

Nos tests

Les fonctions suivantes ont été testées :

- getclassement
- jeu
- getmot
- crypto
- decrypto
- dictionnaire
- compare
- donne_couleur
- start
- majlettreniveau
- getmotactuel
- majTableau
- gestionEntree
- gestionSupprime

Les performances de nos algorithmes et leur complexité

Pour la partie web, le seul algorithme implémenté en Python non linéaire est celui de coloration. Tous les autres algorithmes de notre application web implémentés en JavaScript, ou en Python, sont linéaires y compris ceux de cryptage.

Notre algorithme de coloration est quadratique en calcul en fonction de la taille du mot et linéaire en stockage. Cela est dû à la manière de fonctionner de l'algorithme qui parcourt plusieurs fois les lettres du mot et stocke les occurrences.

Notre application est, néanmoins, extrêmement fluide de la réponse de l'application après une tentative de mot jusqu'à l'affichage des classements. Cependant, cette fluidité est limitée à l'échange entre le serveur et l'utilisateur car le jeu utilise une requête asynchrone à chaque mot empêchant, en grande partie, la triche.

3.6 Points d'améliorations

Notre application Wordle comporte quelques points que nous aurions souhaité améliorer. Tout d'abord, notre base de données pourrait posséder une table supplémentaire où tous les mots tentés seraient stockés séparément. A l'heure actuelle, nous stockons les mots tentés pour chaque partie dans une chaîne de caractères dans notre table "Partie".

Nous pourrions également réaliser des statistiques sur les parties gagnées, l'occurrence des mots tentés, etc. Ainsi, la modification de notre base de données permettrait de réaliser ces dernières avec une plus grande facilité.

De plus, l'esthétisme de notre mock-up pourrait être amélioré. En effet, l'image en arrière-plan empêche parfois la lecture de certaines informations.

Enfin, nous aurions aimé implémenter des dictionnaires supplémentaires afin d'offrir une configuration encore plus singulière aux joueurs, comme par exemple, un dictionnaire relatif à TELECOM Nancy.

Chapitre 4

Notre Solveur

4.1 État de l'art du Solveur

4.1.1 Présentation du Solveur

Un solveur est un programme logiciel permettant de calculer et de fournir le résultat d'un problème après sa transcription informatique. Le solveur permet de trouver une valeur optimale pour un problème donné. Au fur et à mesure, le solveur affine ses recherches et propose la solution qui lui paraît la plus adéquate. Dans le contexte du Wordle, le solveur nous proposera un premier mot qui constituera le premier essai. Nous rentrerons, par la suite, dans le solveur le résultat produit par le premier mot choisi sous la forme d'une liste de numéros composés uniquement de 0, 1 et 2 (0 : la lettre n'est pas présente dans le mot, 1 : la lettre est dans le mot mais mal placée, 2 : la lettre est bien placée). Après l'entrée de la combinaison, le solveur proposera le mot le plus adapté, selon lui. Puis, nous recommencerons le processus décrit précédemment autant de fois que le nombre d'essais autorisé.

4.1.2 Comparaison entre des solveurs existants

Des sites/applications proposent déjà des solveurs où nous rentrons des informations sur les lettres déjà tentées et, en sortie, nous obtenons les mots qui vérifient les contraintes introduites.

Solveur de Wordle (Dcode) est un site proposant de résoudre le Wordle en plusieurs langues : Français, Anglais, Espagnol,... Sur 10 parties réussies, la moyenne est de 3 essais. Sur 12 parties jouées, 2 ont échoué car le solveur ne proposait plus de mots. Les mots à deviner pour ces deux parties étaient CHAUMER et INDISCUTE.

De plus, ce solveur ne propose pas de mot de départ donc son efficacité dépend du premier mot proposé par le joueur. Additionnellement, il ne propose pas un mot à chaque essai mais une liste de mots, laissant la possibilité à l'utilisateur de choisir (pour réaliser les statistiques précédentes, le premier mot de la liste était utilisé). Un autre défaut de ce solveur est qu'il ne prend pas en compte la position des lettres colorées en jaune, donc il était possible qu'il propose un mot contenant une lettre à une place où nous savions qu'elle ne pouvait pas être.



FIGURE 4.1: Interface du solveur Dcode

Word Find est un solveur pour le Wordle, nous représentons le mot tenté par des points d'interrogation si nous ne savons pas quelle lettre se trouve à cette position et sinon nous plaçons la lettre connue. Nous rentrons les lettres n'appartenant pas au mot dans une case et celles colorées en jaune dans une autre.

Sur les 10 parties réussies, la moyenne du nombre d'essais est de 4,2. Sur 11 parties, une seule a échoué. Cependant, ce solveur présente les mêmes défauts que le précédent : la position d'une lettre colorée en jaune n'est pas prise en compte, le solveur ne propose pas de mot de départ, et il propose une liste de mots à chaque essai.

FIGURE 4.2: Interface du solveur Word Find

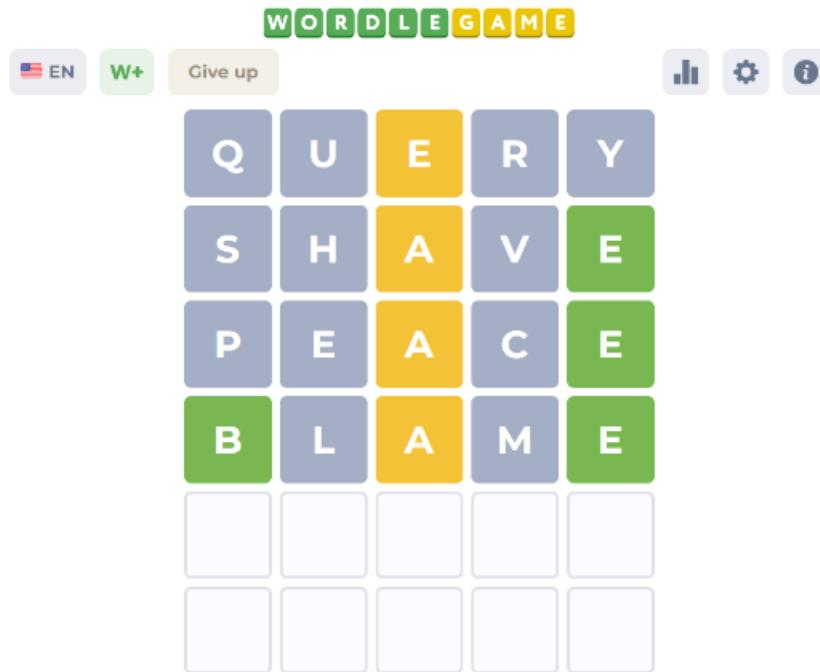


FIGURE 4.3: Interface d'une partie de Wordle en utilisant le solveur Word Find

4.2 Présentation Générale

Pour la deuxième partie du projet, nous devions créer un solveur, afin de trouver en un nombre de tentatives, le plus optimal possible, la solution du Wordle.

Nous avons décidé de créer plusieurs versions de solveur. En commençant par la version la plus simple à coder, c'est-à-dire prendre un mot au hasard dans le dictionnaire. Puis, nous nous sommes dirigés vers une version plus complexe reposant sur la théorie de l'information, tout en réutilisant des fonctions de bases développées pour la première version. Ainsi, nous avons pu développer deux versions du solveur, et l'efficacité des deux versions est très différente.

4.3 Première version du Solveur

4.3.1 Structure de données

Pour la première version de notre solveur, nous avons décidé de stocker les différentes tentatives pour trouver le mot à deviner sous forme d'un tableau. La taille du tableau sera la longueur du mot multiplié par le nombre d'essais. Chaque élément du tableau comporte quatre éléments : une lettre du mot tenté, la place de la lettre dans le mot, la coloration de la lettre et le numéro de l'essai qui est initialisé à 0. Ainsi, toutes les informations concernant les tentatives de mots sont stockées au même endroit et sont accessibles rapidement.

Par exemple, pour le mot *MERCY* avec une coloration de 01201, le tableau aura pour forme : [['M',1,0,0],['E',2,1,0],['R',3,2,0],['C',4,0,0],['T',5,1,0]]

4.3.2 Algorithme d'épuration du dictionnaire

L'algorithme d'épuration se compose de 3 étapes. On parcourt le dictionnaire initial mot par mot.

La première étape consiste à vérifier si ce mot n'est pas composé des lettres colorées en 0. La deuxième étape consiste à vérifier si ce mot est composé aux bonnes positions des lettres colorées en 2.

La troisième étape consiste à vérifier si le mot contient les lettres colorées en 1 sauf aux positions où elles ont été testées.

Si ce mot vérifie ces trois conditions, alors il est inséré dans un nouveau dictionnaire qui servira de dictionnaire initial à la prochaine tentative de l'utilisateur.

4.3.3 Choix de la meilleure proposition

Pour cette version, nous avons décidé de choisir un mot au hasard dans notre dictionnaire. Après chaque essai et donc après épuration du dictionnaire, un mot est choisi aléatoirement dans le nouveau dictionnaire et ainsi de suite.

4.4 Deuxième version du Solveur

4.4.1 Structure de données

Pour cette deuxième version, nous avons décidé d'implémenter une structure originale qui n'avait pas été vue en cours de Structure de Données. Notre choix a changé à de nombreuses reprises afin d'ajuster au mieux notre structure. A l'origine, un simple arbre avait été évoqué puis, nous avons décidé de faire plusieurs graphes où chaque graphe contient un langage de taille de mot différente (ex : graphe des mots de taille 3, graphe des mots de taille 6). Finalement, notre choix s'est porté sur un graphe orienté acyclique (DAWG en anglais). Ce type de graphe a comme spécificité le fait qu'il n'y a pas de chemin pré-défini. Nous avons choisi un graphe, car un arbre où l'on lit des mots correspond finalement à un langage et, pendant les modules de *TLA* et de *MSED*, nous avons étudié des graphes et automates définissant des langages. Dans cette version du solveur, un graphe est créé à chaque essai en fonction du nombre de lettres qu'il y a dans le mot. Après chaque essai, le graphe est épuré. Le graphe est composé d'un noeud racine, de noeuds enfants, d'arêtes reliant les différents noeuds. Sur chaque arête, il y a deux informations : une lettre et la pondération associée.

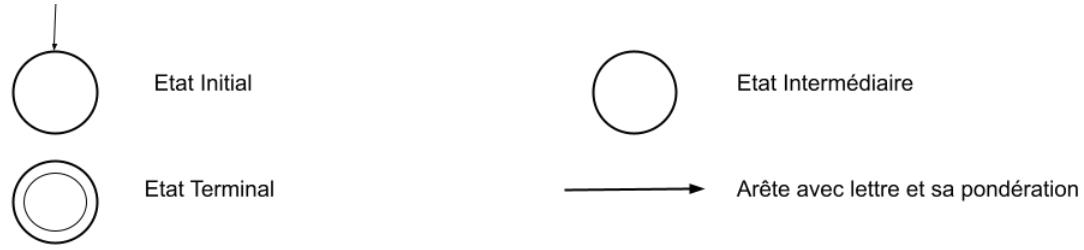


FIGURE 4.4: Légende des graphes

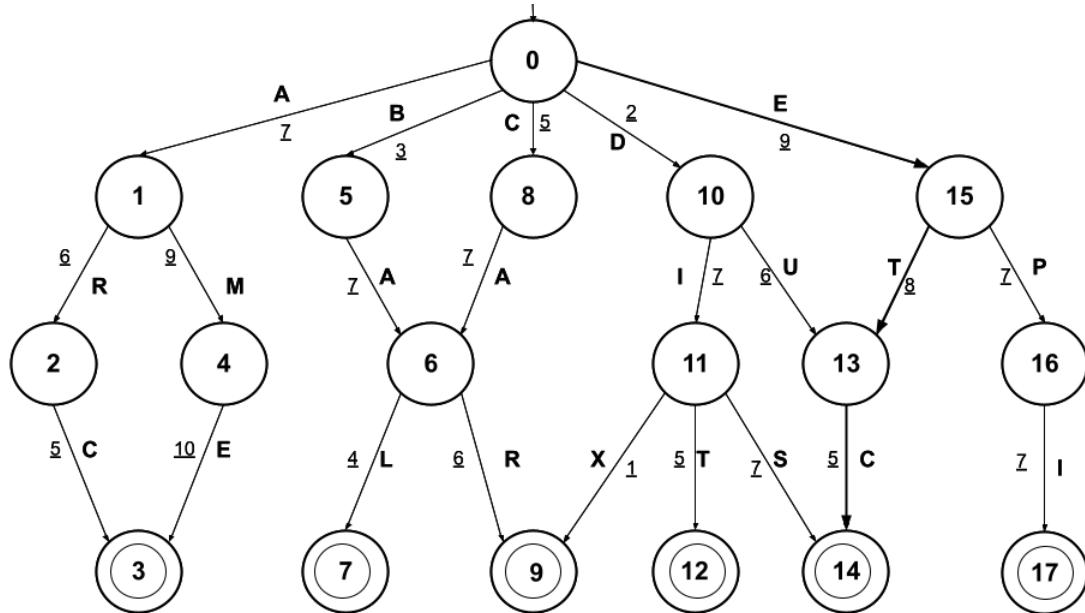


FIGURE 4.5: Exemple de graphe pour des mots de taille 3

La pondération, de chaque lettre, est calculée en fonction de sa place dans le mot. A chaque occurrence de la lettre à une position donnée, la pondération est incrémenté de 1. Nous avons décidé de stocker les différentes pondérations pour chaque lettre sous forme de Hash-map.

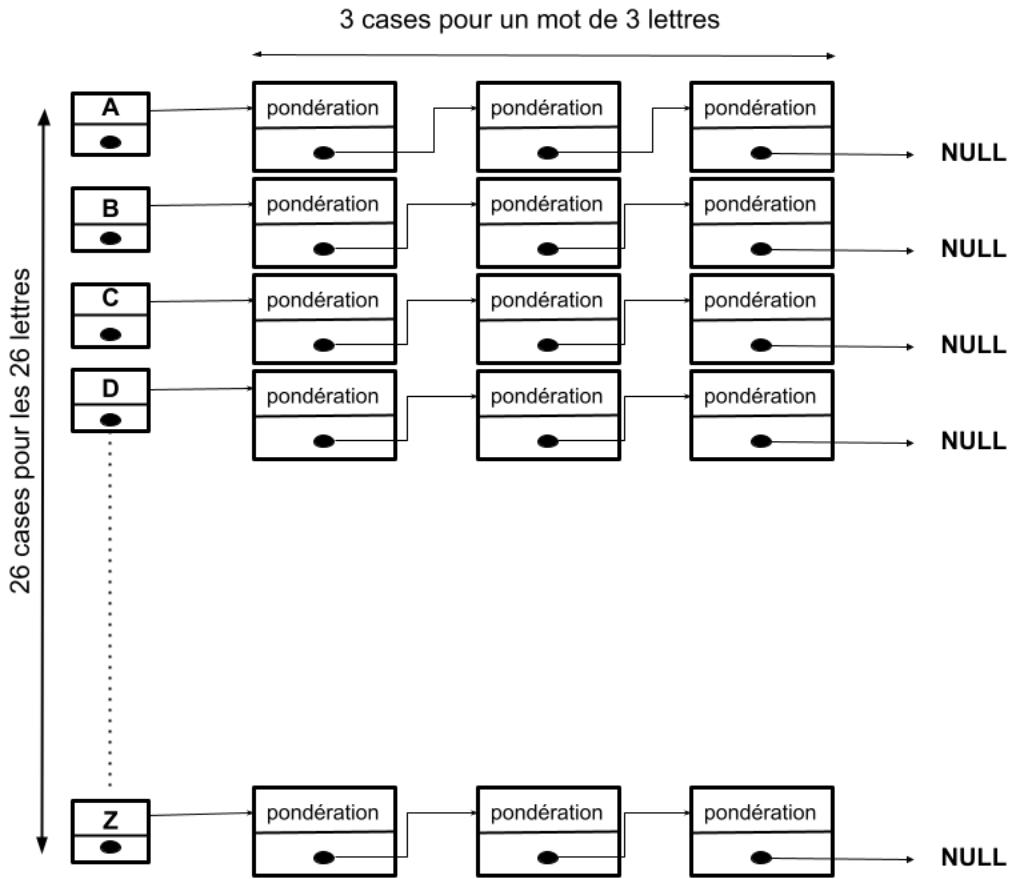


FIGURE 4.6: Schéma de la Hashmap utilisée dans la deuxième version de notre solveur

Nous avons implémenté trois types de structures pour notre graphe :

- node : contient une liste de noeuds enfants, un identifiant, le nombre d'arêtes entrantes sur le noeud et un boolean indiquant si le noeud est terminal.
- edge : contient un noeud, un identifiant, une lettre et la pondération associée.
- list_edge : contient un élément de type edge et un pointeur vers le noeud suivant.

```

struct _node {
    list_edge* listEdge;
    int id;
    bool terminal;
    int nbrAretesEntrantes;
};

struct _edge {
    node* node;
    int id;
    char label;
    int ponderation;
};
struct _list_edge{
    edge e;
    list_edge *next;
};

```

FIGURE 4.7: Extrait de l'implémentation de notre structure de données

4.4.2 Algorithme d'épuration du graphe

Pour notre algorithme d'épuration, nous sommes repartis du même principe que celui pour notre première version. Après chaque essai, la coloration du mot tenté est connue. Ainsi, nous lisons les mots du graphe un par un et on compare ce mot à la coloration obtenue, si ce mot remplit les conditions il est ajouté à un nouveau graphe. Cette action est répétée sur tous les mots. A la fin, le graphe actuel est supprimé et remplacé par le nouveau graphe que nous venons de remplir. Ce mécanisme est répété à chaque essai.

4.4.3 Choix de la meilleure proposition

Pour trouver le mot qui a le plus de chance d'être la bonne solution, nous avons décidé d'attribuer une pondération à chaque lettre. Cette pondération de sa position dans le mot, par exemple la pondération de la lettre **E** à la 1^{ère} position est différente de celle de la lettre **E** à la 3^{ème} position.

Nous avons écrit deux algorithmes de parcours des graphes pour trouver le mot avec la plus grande pondération.

V2.1

Pour le premier algorithme, nous avons décidé que le solveur allait parcourir le graphe de la plus simple des manières : à chaque noeud, la lettre qui a la plus grande pondération est choisie. A la fin, un mot est proposé et il a une probabilité importante d'être la solution.

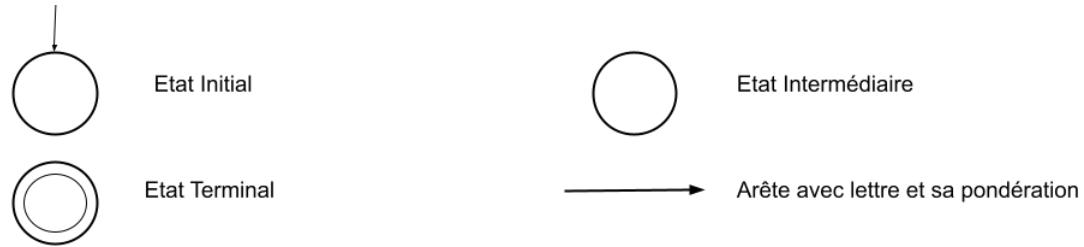


FIGURE 4.8: Légende des graphes

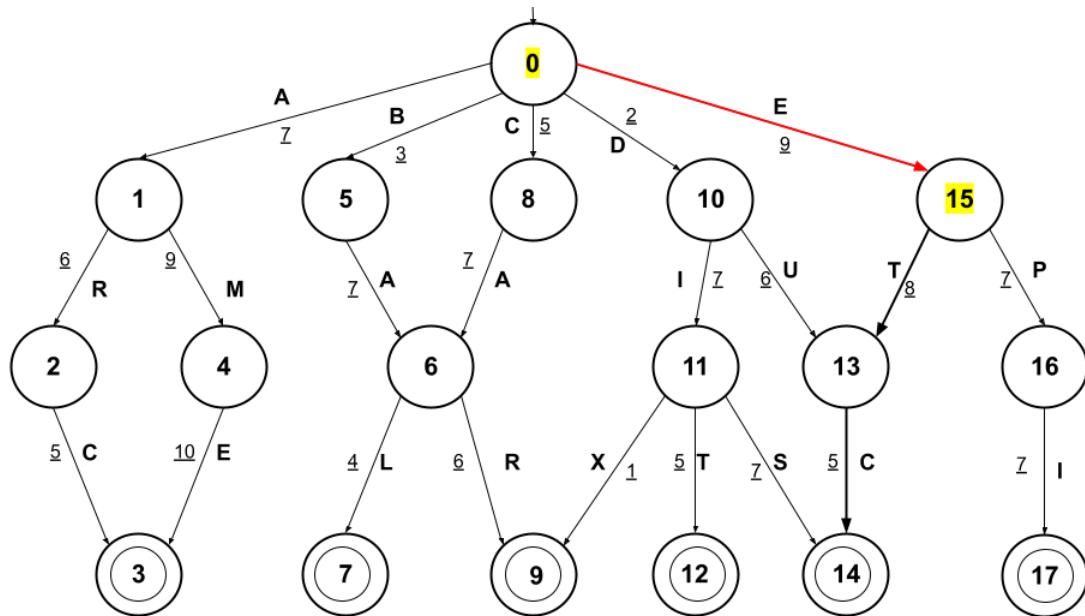


FIGURE 4.9: Première étape de l'algorithme, choix de la plus grande pondération à ce noeud

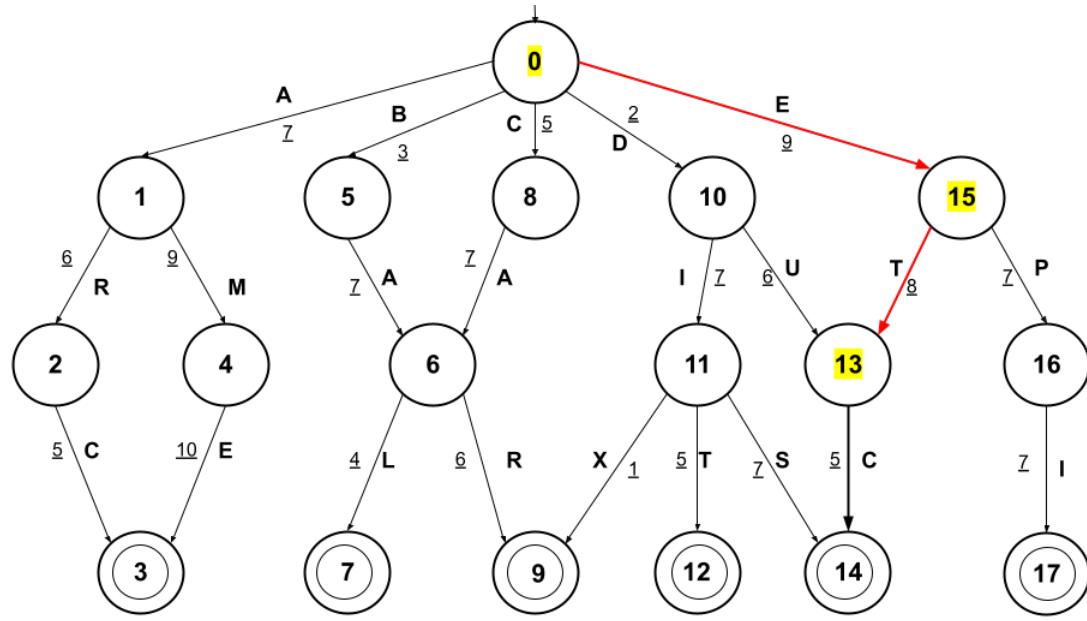


FIGURE 4.10: Deuxième étape de l'algorithme, choix de la plus grande pondération à ce noeud

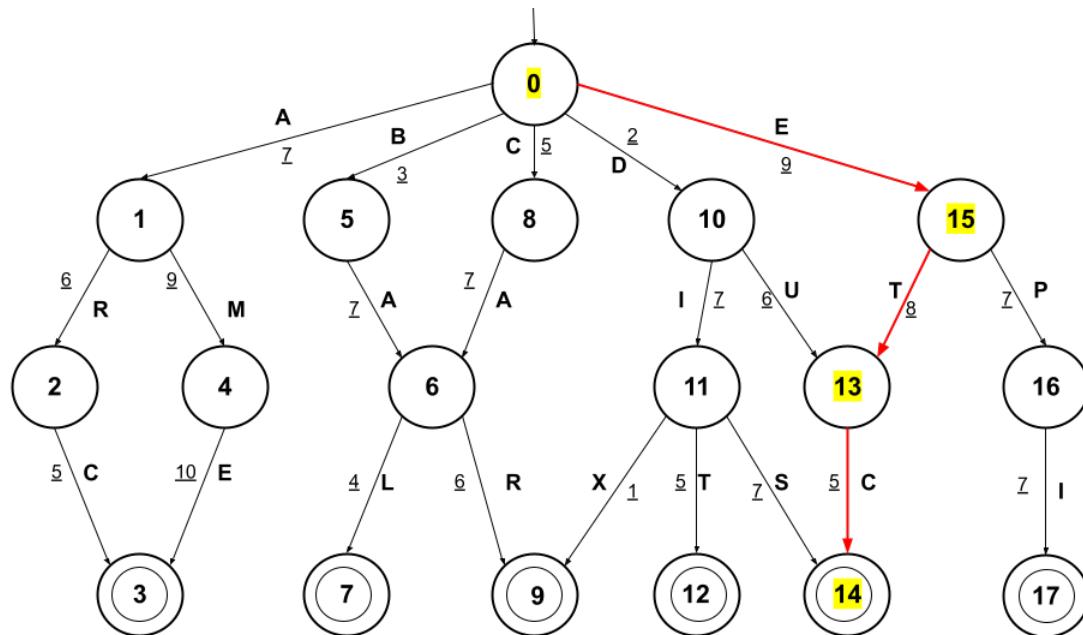


FIGURE 4.11: Troisième étape de l'algorithme, choix de la plus grande pondération à ce noeud

V2.2

Avec l'algorithme précédent, le mot proposé n'était pas celui avec la plus grande probabilité d'être la bonne solution. En effet, l'algorithme ne testait pas toutes les possibilités de pondération. Pour ce second algorithme, nous avons décidé d'utiliser le principe de *backtracking* vu au premier semestre de notre cursus dans le module *CS54*. Avec ce moyen de fonctionnement, le solveur parcourt toutes les combinaisons de pondération possibles. Ainsi, nous sommes sûrs d'obtenir le mot avec la plus grande pondération possible.

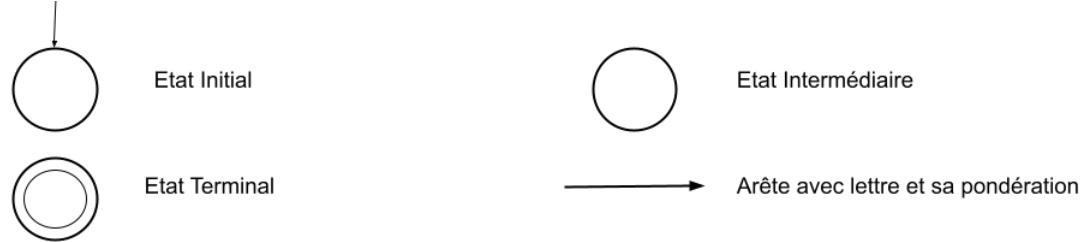
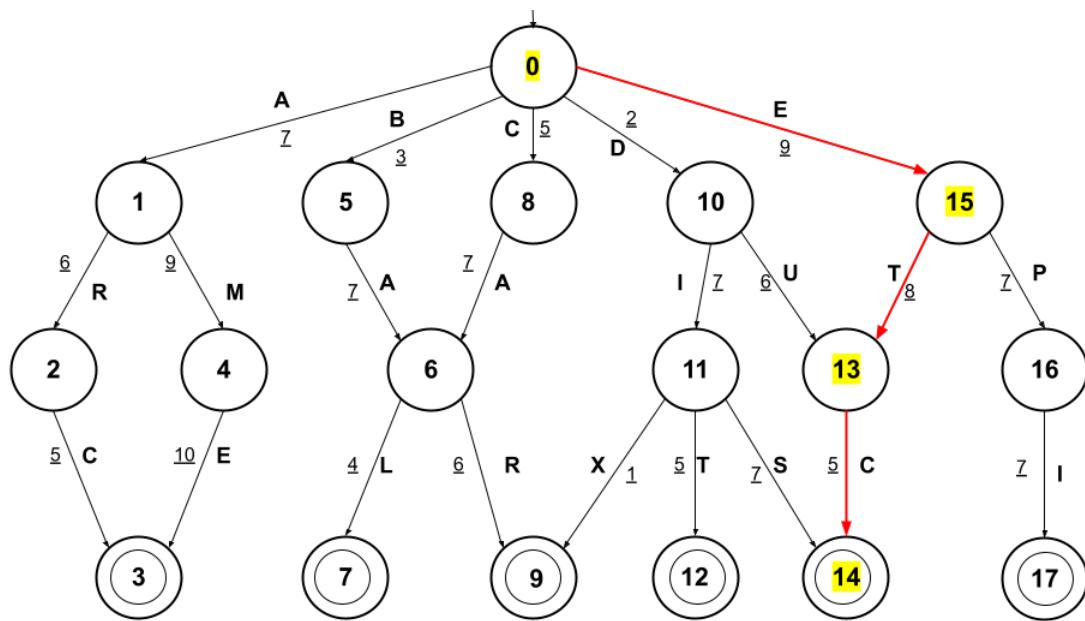


FIGURE 4.12: Légende des graphes

FIGURE 4.13: Exemple de recherche du meilleur choix première étape avec une **pondération= 22**

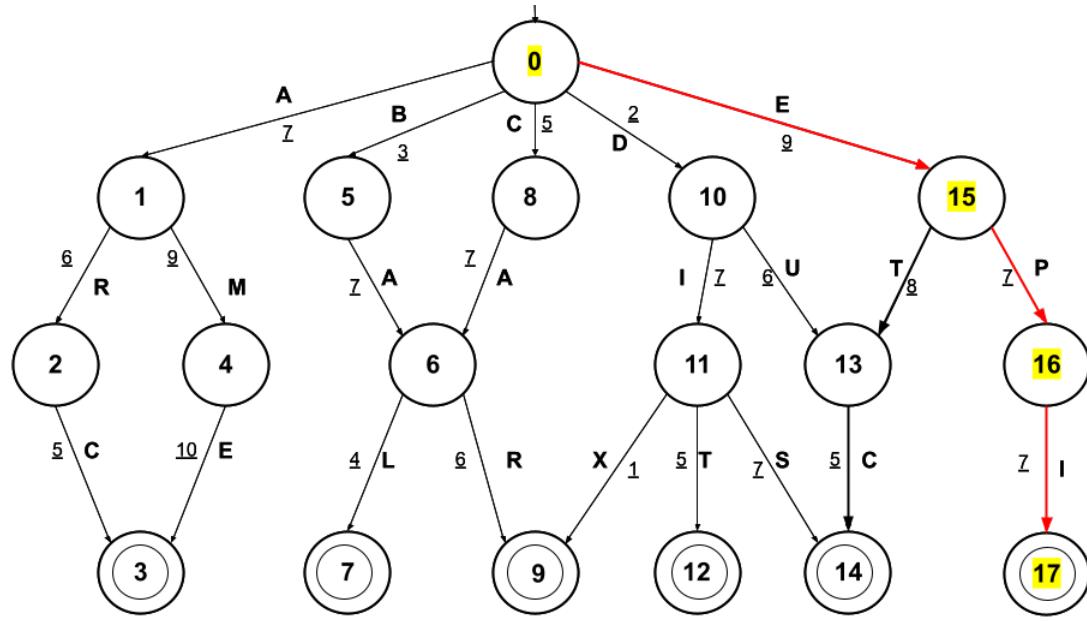


FIGURE 4.14: Exemple de recherche du meilleur choix deuxième étape, parcours des arêtes précédentes avec une **pondération=23**

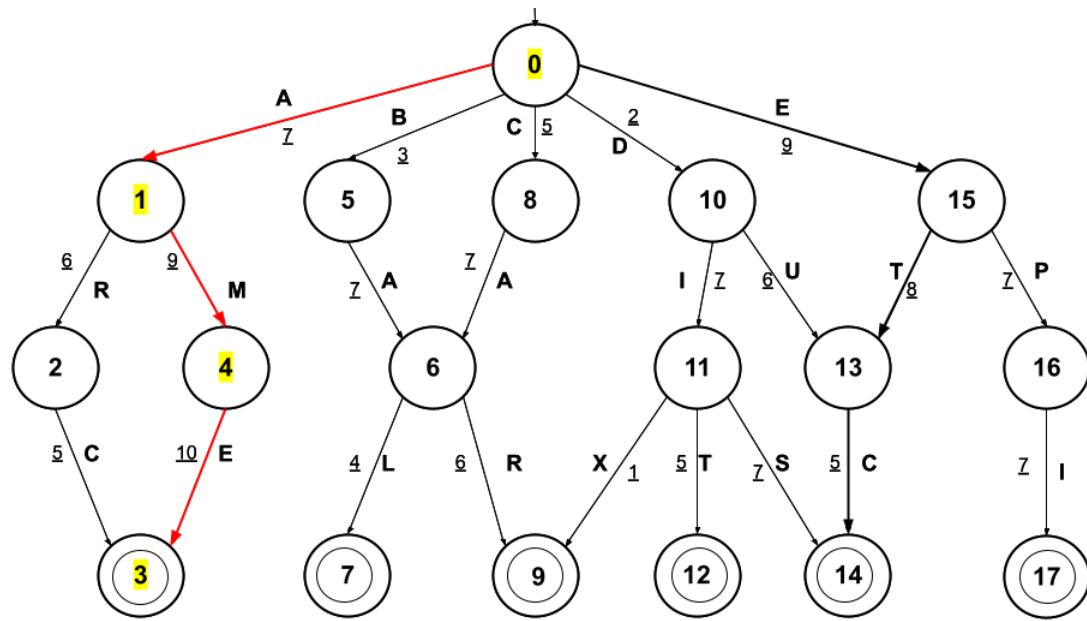


FIGURE 4.15: Exemple de recherche du meilleur choix dernière étape, résultat meilleur combinaison d'arêtes avec une **pondération=26**

4.5 Les tests et les performances

4.5.1 Les tests

Les fonctions suivantes ont été testées :

- contains
- aceteplace
- containsexceptposition
- contentofline
- coloration0
- coloration1
- coloration2
- ajout_mot
- randomInFile
- get_ponderation_hash_list
- get_ponderation
- append_hash_list
- node_add_char
- node_add_word
- node_add_all_words

4.5.2 Les performances de nos algorithmes et leur complexité (Partie Solveur)

Pour analyser les performances et la complexité de notre solveur, nous avons souhaitons les différentes versions de notre solveur. Pour se rendre compte des avantages et des inconvénients de chacune, nous avons implémentés un testeur de solveur en C.

Analyse performance V1 du solveur

Notre première version n'utilise aucune structure de données avancée. Il épure seulement les dictionnaires mot par mot et sélectionne un mot aléatoire dans le fichier texte.

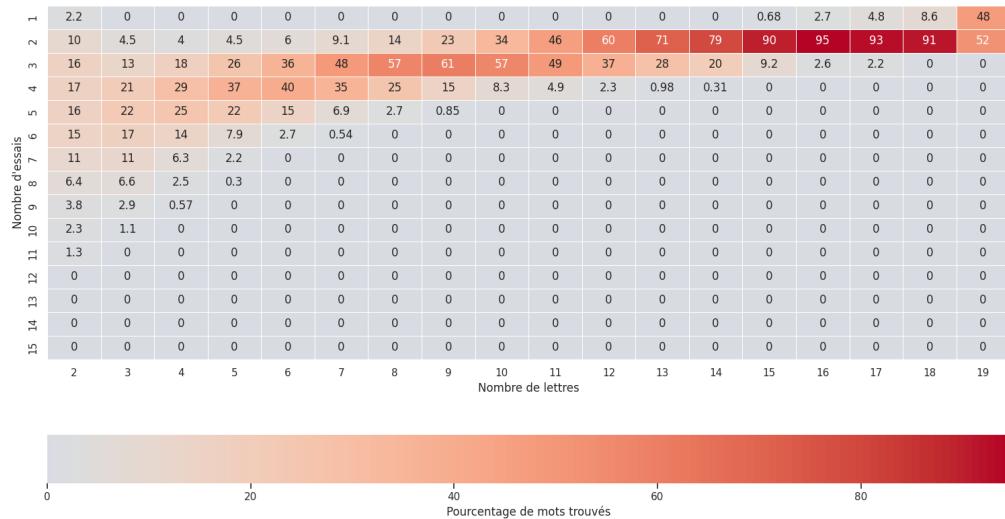


FIGURE 4.16: Heatmap du pourcentage de mots trouvés en fonction du nombre de lettres et du numéro de l'essai

Ses performances sont donc moyennes. On remarque, tout de même, que le solveur trouve tous les mots, peu importe leur taille ou leur nombre de lettres, en moins de 13 essais. On remarque aussi que plus le nombre de lettres est important plus le nombre d'essais nécessaire à notre première version est faible.

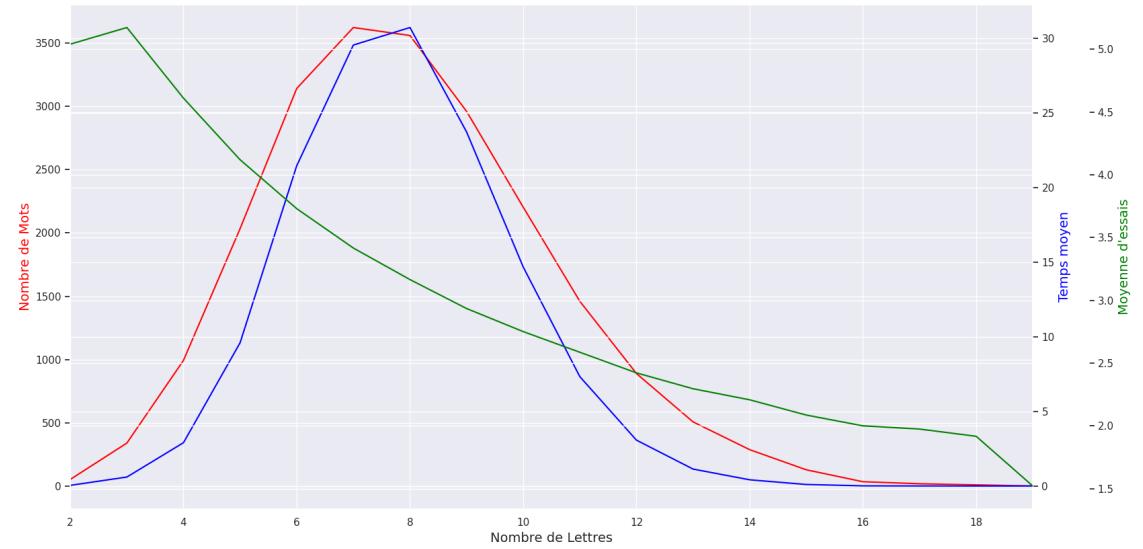


FIGURE 4.17: Graphique représentant la moyenne du nombre d'essais nécessaires, le temps moyen de complétion de la recherche et le nombre de mots en fonction du nombre de lettres

On peut remarquer que le temps mis par le solveur est bien linéaire par rapport au nombre de mots et par rapport à la taille du mot. Pour la complexité en stockage, on stocke 2 dictionnaires de taille maximale la taille du dictionnaire initial. La première version est donc linéaire par rapport au nombre de mots

Analyse performance V2 du solveur

Nous n'avons malheureusement pas eu le temps de faire des tests aussi poussés que pour la première version de notre solveur. Cependant, nous avons testé manuellement notre solveur avec notre application wordle. Pour 10 parties jouées, la bonne réponse est trouvée en moyenne en 4,5 essais. Sur les parties jouées, le mot a été trouvé à chaque fois.

4.6 Points d'améliorations

Notre solveur comporte quelques points que nous aurions voulu améliorer.

Nous en avons actuellement deux versions, nous aurions voulu en implémenter davantage ce qui aurait permis à notre solveur d'être plus performant.

Comme expliqué précédemment, à chaque essai un nouveau graphe est créé. Cependant, nous n'avons pas pu minimiser ces graphes comme prévu initialement. En effet, à l'aide de l'algorithme vu lors du module de TLA, minimiser les graphes aurait permis de réduire le nombre de noeuds total du graphe et ainsi la complexité de nos algorithmes aurait été réduite.

Une autre possibilité d'amélioration de notre solveur aurait été de transformer entièrement notre graphe en binaire. Actuellement, chaque noeud comporte un identifiant, un booléen indiquant s'il est terminal ou non, et une liste de noeuds enfants. Chaque noeud enfant possède un caractère, une pondération et toutes les autres caractéristiques du noeud décrites précédemment. Toutes ces informations auraient, alors, été converti en binaire. En effet, la comparaison entre deux binaires est bien plus simple qu'entre deux caractères.

Chapitre 5

Difficultés rencontrées

5.1 Difficultés lors de la phase Application Web

La phase Application Web s'est déroulé sans problèmes majeurs. En effet, le projet du premier semestre le sujet comportait également le développement d'une application Web, nous avions donc déjà des compétences dans ce domaine. Nous avons donc pu avancer rapidement sur la construction de la structure du site et des principales fonctionnalités. Cependant, la partie la plus importante du projet était l'implémentation du jeu Wordle et nous avons décidé de coder cette partie en JavaScript, langage que nous ne maîtrisons pas. Nous avons procédé à de l'auto-formation sur ce langage afin de pouvoir avancer dans le projet. Finalement, la première partie du projet s'est déroulé sans trop de difficultés malgré la découverte de ce nouveau langage.

5.2 Difficultés lors de la phase Solveur

La création d'un solveur a été une expérience inédite pour deux des membres du groupe. De plus, le sujet du projet demandait explicitement de réaliser le solveur en C. Pour trois des membres du groupe, le C venait d'être découvert quelques semaines auparavant lors de notre cursus scolaire à TELECOM Nancy. Nous nous sommes répartis les tâches rapidement mais les difficultés sont vite apparues dans l'exécution du code (fuites mémoire, erreurs à la compilation, ...). Pour la deuxième version du solveur, nous voulions implémenter une structure de données particulière mais nous avons mis du temps à nous décider. En effet, nous avons changé d'avis à de multiples reprises afin de choisir la structure la plus adaptée au projet et à nos compétences.

Additionnellement, nous étions dans une période charnière de notre première année à TELECOM Nancy. En effet, nous avions de nombreux partiels en parallèle et devions fournir une grande quantité de travail et d'énergie pour les réussir. Ainsi, répartir notre temps, entre projet et partiels, a été un véritable challenge : nous avions peur d'en négliger un au profit de l'autre, ce qui engendrait beaucoup de stress. C'est pourquoi, nous avons remis en question les objectifs qui nous nous étions fixés. Par exemple, nous voulions réaliser plusieurs versions du solveur pour pouvoir l'améliorer, tout en comparant l'efficacité de toutes les versions. Cependant, nous n'avons pu réaliser que deux versions du solveur.

Chapitre 6

Bilan du Projet

6.1 Conclusion

A l'issue de ce projet, nous avons créé une application web sur le principe de fonctionnement du jeu Wordle ainsi qu'un solveur permettant de résoudre automatiquement notre Wordle.

Pour mener à bien ce projet, nous avons utilisé divers outils de management vus lors du premier semestre de notre cursus à TELECOM Nancy. Le diagramme de Gantt prévisionnel, nous a permis de planifier clairement les différentes étapes du projet. La réalisation du diagramme de Gantt réel tout au long du projet, quant à elle, nous a permis de suivre notre progression. De plus, la conception de la matrice SWOT dès le début du projet, nous a permis d'identifier clairement nos forces et nos faiblesses. L'établissement d'un WBS, nous a permis de diviser correctement la charge de travail aux différents moments du projet.

Concernant, la partie application web du projet, nous avons pu implémenter toutes les fonctionnalités prévues initialement et même plus, avec un classement des joueurs qui n'était pas prévu mais que nous trouvions pertinent. Cependant, lors des vacances d'avril, nous avons dû travailler à distance ce qui a ralenti notre rythme de travail. En effet, nous avions l'habitude de travailler en présentiel. Tous nos objectifs ont, tout de même, été remplis à la fin de cette partie du projet. Ainsi, nous avons réalisé une application web s'inspirant du Wordle, où les utilisateurs peuvent jouer en mode mot du jour ou en mode libre, ce qui leur permet de paramétrier à leur souhait le jeu. Ces différents paramètres sont la langue, le niveau de difficulté, le nombre de lettres dans le mot à deviner et le nombre d'essais.

Pour la deuxième partie du projet qui était la conception d'un solveur codé en C, les choses ont été plus compliquées. En effet, cette partie du projet s'est déroulée en même temps que nos partiels du second semestre. Nous avons donc dû repousser, à plusieurs reprises, nos réunions par manque de temps ou parce que personne n'avait eu le temps d'accomplir les tâches qui lui étaient attribuées lors de la précédente réunion. Heureusement, nous avons bénéficié d'une semaine supplémentaire pour la réalisation du projet. Nous avons partiellement atteint nos objectifs. A l'origine, nous souhaitions créer de multiples versions du solveur avec, à chaque version, une amélioration par rapport à la précédente. Cependant, nous n'avons eu le temps d'en concevoir uniquement deux. La première version nous a pris plus de temps que prévu. Le manque de connaissances en C ainsi que les partiels ont été des obstacles conséquents. En ce qui concerne la deuxième version, nous avons mis du temps à nous décider sur la meilleure structure de données à adopter. De plus, nous souhaitions minimiser nos graphes à chaque itération du solveur afin de réduire le temps de parcours pour rechercher la meilleure solution et la place en mémoire que prend le graphe, nous n'avons malheureusement pas eu le temps de coder l'algorithme entier. Enfin, nous n'avons pas pu réaliser tous les tests que nous souhaitions sur la deuxième version de notre solveur par manque de temps.

Dans l'ensemble, le projet s'est très bien déroulé et a été apprécié par l'intégralité du groupe. En effet, le côté concret du projet, la création d'un jeu Wordle dans un premier temps et la conception d'un solveur qui permet de résoudre notre jeu dans un second temps a rendu le projet plus intéressant. De plus, ce projet a permis à tous les membres de monter en compétence sur les différents langages informatiques utilisés (Javascript, C). La motivation au sein du groupe a toujours été de rigueur et les nombreuses séances de travail en présentiel ont permis de garder un rythme soutenu de travail.

6.2 Conclusions Personnelles

6.2.1 Mathilde BARD

Tout d'abord, ce fut un plaisir de travailler avec ce groupe. En effet, il y avait toujours une ambiance agréable mais propice au travail entre nous. De plus, tous les membres du groupe étaient toujours volontaires pour me conseiller ou m'aider.

Notre projet s'est déroulé en deux parties séparées : une partie pour l'application Web et une partie pour le solveur en C. La première partie de notre projet s'est bien déroulée car nous bénéficiions de notre expérience acquise lors du premier projet. Lors du premier PP2I, j'avais, en majorité, travaillé sur l'interface. Ainsi, dans l'optique de développer mes connaissances, j'ai essayé de diversifier mon champ d'action lors de ce projet : j'ai donc implémenté la base de données, réalisé des algorithmes en Python et en JavaScript.

La deuxième partie du projet s'est révélée être plus difficile, autant sur le plan langage informatique que sur le plan disponibilité. En effet, le C était une nouveauté pour moi. Ainsi, réaliser un projet d'une telle ampleur avec si peu de pratique s'avéra compliqué et j'ai eu besoin de l'aide de mes camarades à de nombreuses reprises. Cependant, il fut satisfaisant de constater ma progression entre le début de la deuxième partie et la fin de cette dernière. Additionnellement, cette période était chargée en partiels. Par conséquent, réussir à trouver un équilibre entre révisions et projet fut compliqué. En effet, je ne voulais négliger ni l'un ni l'autre.

Finalement, j'ai apprécié réaliser ce projet avec mon groupe. J'ai particulièrement apprécié le fait qu'on créé nous-même le jeu pour le résoudre avec un solveur.

6.2.2 Lucie BOUCHER

Dans son ensemble, le projet s'est bien déroulé, cela résulte sûrement de l'investissement important de chacun dans ce projet, et je remercie mon équipe pour cela.

Le projet s'est déroulé en deux parties distinctes.

Lors de la première partie du projet, la maîtrise préalable de Python, du framework Flask, HTML et CSS, nous ont permis d'approfondir le développement web grâce au langage JavaScript.

La partie solveur du jeu a été différente car elle était plus complexe. Cependant, nous avons pu utiliser des notions étudiées en cours pour réaliser le solveur. La concrétisation de ces notions théoriques était, pour moi, une grande source de motivation.

Tout au long du projet, j'ai endossé le rôle de secrétaire. Je me suis portée volontaire car la rédaction est un de mes points faibles.

Pour finir, j'ai particulièrement apprécié le caractère concret de ce projet et travailler avec ce groupe toujours prêt à s'entraider.

6.2.3 Rémi BOURDAIS

J'aimerais, dans un premier temps, remercier mes camarades pour la bonne entente qu'il y a eue dans le groupe et qui a rendu agréable la réalisation de ce projet. J'ai grandement apprécié les séances de travail en groupe qui m'ont permis d'être plus efficace et motivé sur toute la durée du projet. Concernant l'organisation du groupe, chacun a su trouver rapidement la place qu'il souhaitait ce qui nous a permis de commencer tôt le projet.

La première partie du sujet m'a permis de découvrir le JavaScript mais aussi de revoir toutes les bases du développement Web vu au premier semestre. Cette première partie s'est très bien déroulée, malgré quelques difficultés rencontrées sur le CSS et sur le JavaScript. Nous avons rapidement eu une application fonctionnelle ce qui m'a motivé à poursuivre.

La seconde partie du projet, elle, a été plus compliquée car je ne maîtrisais que les bases du C. La gestion de la mémoire était, pour moi, une toute nouvelle notion. Mes camarades ont su m'aider sur les difficultés que j'avais et j'ai fait de même avec les leurs, cela nous a permis de ne pas rester bloqués sur une étape de notre solveur trop longtemps. Bien que nous nous sommes répartis les parties du solveur et que nous avons codés chacun dans des fichiers séparés, joindre tous les fichiers a été pour le moins compliqué et a retardé l'arrivée de notre V2.

D'un point de vue personnel, ce projet m'a beaucoup appris sur la partie technique du Web, ainsi que sur la gestion des chaînes de caractères et de la mémoire en C. L'organisation du groupe était différente que celle que j'avais expérimenté lors de mon premier projet PP2I, ce qui m'a permis de voir une nouvelle façon de faire et de gérer un projet.

6.2.4 Lola MONTIGNIER

Premièrement, je voudrais mettre en avant la bonne entente au sein du groupe qui a permis de rendre ce projet agréable. J'ai beaucoup apprécié les séances de travail en groupe. Nos réunions se sont toujours déroulées dans le respect et l'écoute de chacun.

Ce projet fut, en partie, similaire à celui du premier semestre. Cependant, pour ce projet, j'ai exercé le rôle de cheffe de projet ce qui était une nouveauté pour moi. Lors de nos différentes réunions, j'ai dû préparer en amont les ordres du jour et animer les réunions. Ce rôle a, au début, été difficile au début pour moi, car la prendre la direction d'un groupe n'est pas dans mon caractère. Cependant, j'ai su, grâce à la confiance de mes camarades, prendre l'assurance nécessaire à l'accomplissement de ce rôle.

La première partie du projet s'est très bien passée, car très similaire au projet précédent, il a donc été assez facile d'avancer rapidement dans le projet. Seul le JavaScript a été une difficulté à surmonter, en effet, ce langage étant totalement inédit pour moi. J'ai dû me former par moi-même et je remercie mes camarades qui m'ont apporté leurs connaissances.

La seconde partie du projet s'est déroulée de façon diamétralement opposée. En effet, la création d'un solveur en C était un challenge important pour tous les membres du groupe. Pour ma part, je venais de découvrir ce langage quelques semaines auparavant et donc, ne le maîtrisais que très peu. De plus, les partiels se sont accumulés pendant cette période. Ce moment était donc d'une grande importance quant à la réussite de mon année scolaire à TELECOM Nancy. Ceci a généré beaucoup de stress car je ne voyais pas comment il était possible de tout réussir. De plus, je me sentais parfois inutile car, le C étant très compliqué, les autres membres du groupe devaient m'aider à chaque fois que je codais. C'est pourquoi dans la dernière semaine du projet, je me suis consacrée à la rédaction du rapport pour permettre à mes camarades d'avancer plus rapidement sur l'écriture du code.

Finalement, ce projet s'est avéré très formateur, aussi bien sur le plan informatique avec l'apprentissage des langages C et JavaScript, que sur le plan management avec mon rôle de cheffe de projet. Il m'aura permis de savoir m'adapter dans des moments de stress, de me former et de chercher des informations pour pallier à mes lacunes.

Je suis plutôt satisfaite du travail que j'ai effectué et des fonctionnalités que j'ai pu implémenter au sein du projet.

6.3 Travail réalisé

Ce tableau comptabilise le nombre d'heures de chacun passées à travailler sur le projet. Les heures de réunions sont incluses. La charge de travail de ce projet est de l'ordre de 100h par personne.

Étapes	Mathilde	Lucie	Rémi	Lola
Gestion de projet				
Réunions	15h	15h	15h	18h
Compte-rendus	2h	12h	0h	5h
Outils de gestion de projet(Gantt, Swot)	1h	3h	2h	4h
Rédaction du rapport	16h	5h	3h	20h
Recherche d'informations	8h	6h	5h	5h
Partie Conception Wordle				
Schéma relationnel	0h	0h	1h	1h
Application web	8h	10h	4h	8h
Code SQLite	4h	1h	2h	1h
Interface Graphique	4h	3h	8h	3h
Algorithmes de Coloration	15h	6h	7h	6h
Gestion des Dictionnaires	5h	0h	6h	3h
Tests de l'application	1h	0h	4h	0h
Partie Solveur				
Gestion des Données	0h	5h	4h	9h
Algorithme Dictionnaires	10h	3h	5h	0h
Structure de Données	5h	15h	3h	6h
Gestion des Graphes	0h	15h	5h	9h
Algorithme Entropie	1h	0h	5h	0h
Automatisation/Statistiques	0h	0h	10h	0h
Test et Performance du Solveur	0h	0h	10h	0h
Interface	0h	0h	3h	1h
Total				
	95h	101h	104h	98h

TABLE 6.1: Charge de travail

Chapitre 7

Annexes

7.1 Comptes Rendus de Réunions

Nous avons réalisé 15 réunions, elles ont toutes eu le même format : nous revenions sur la TODO liste de la réunion précédente, puis nous évoquions tous les points à l'ordre du jour, enfin nous nous répartissions de nouvelles tâches en fonction de l'avancement du projet.

La secrétaire Lucie BOUCHER prenait des notes pendant les réunions, puis elle mettait en forme au fur et à mesure les comptes-rendus de réunion.

7.1.1 Réunion du 18 Mars

Type : Réunion de lancement	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 18 Mars à 18h

Ordre Du Jour :

- Présentation des idées
- Document Conception
- Gestion de projet : chef de projet, état de l'art, matrices RACI et SWOT

Présentation des idées :

Comment va-t-on procéder ? on va commencer par coder une application simple et qui fonctionne puis l'améliorer.

Améliorations obligatoires :

- Choix du nombre de lettres des mots
- Choix du nombre d'essais maximal

Boite à idées :

- Interface en JS en plus du CSS
- Gestion des utilisateurs
 - Stocker les parties jouées
 - Classement
- Thème avec dictionnaire personnalisé
 - Différentes langues
- Page avec une explication du solveur pédagogique

Document de conception :

Créer un overleaf pour le document de conception.

Commencer par l'état de l'art :

- Présenter Wordle
- Rechercher des solveurs de Wordle existants et les analyser

Faire le mock-up dynamique. Attendre le début du cours de SD. Comment communiquer entre les différents langages ? .txt et terminal

Gestion de Projet :

Cheffe de Projet : Lola Montignier	Secrétaire : Lucie Boucher
------------------------------------	----------------------------

Certains éléments pour remplir la matrice SWOT :

Mathilde :

Force : maîtrise le CSS	Faiblesse : ne sais pas coder en C
Opportunités : Apprendre de nouveaux langages	Menace : associatif

Rémi :

Force : bonnes notions en C et a déjà codé des solveurs	Faiblesse : orthographe
Opportunités : Apprendre le JavaScript	Menace : associatif

Lucie :

Force : bonnes notions en C et a déjà codé des solveurs	Faiblesse :
Opportunités : Apprendre le JavaScript	Menace : associatif

Lola :

Force : expérience du premier projet et des connaissances en base de données	Faiblesse : ne sais pas coder en C
Opportunités : Apprendre le C et le JavaScript	Menace : Virtual Machine très lente

Il faut faire un Gantt prévisionnel pour, à la fin, le comparer avec le Gantt réel.
Section pour le gant prévisionnel :

- Gestion de projet
- Solveur
- Interface
- Base de données
- Algorithme de résolution

TODO List :

To Do	Qui ?	Pour Quand ?
Création fichier overleaf	Tout le monde	24 mars
État de l'art	Tout le monde	24 mars
Gantt prévisionnel	Lucie	24 mars
Matrice SWOT	Mathilde	24 mars
Imaginer la base de données	Lola	24 mars
Mock up	Rémi	24 mars

Prochaine Réunion : 24 mars 2022

7.1.2 Réunion du 24 Mars

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 24 Mars à 14h

Ordre Du Jour :

- Retour sur la TODO de la semaine dernière
- Base de données à fixer
- Etat de l'art
- Gantt prévisionnel
- Séance de code du 24 mars : quoi faire ? Comment répartir le travail ?
- Fonctionnalités du jeu à implémenter

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
Etat de l'art	Tout le monde	Brouillon de l'état de l'art réalisé, à compléter
Gantt prévisionnel	Lucie	Gantt réalisé en partie, à rediscuter (CF : point sur le Gantt)
Matrice SWOT	Mathilde	Matrice complète et validée
Mock up	Rémi	Le mock-up est en png sur le drive et validé par le groupe
Conception de la base de données	Lola	un premier jet de schéma est sur le drive, à rediscuter (CF : point sur la base de données)

Base de données à fixer :

Lola présente au reste du groupe les tables qu'elle a imaginées. Dans la table Joueur, on décide tous ensemble que le pseudonyme sera la clé primaire de la table.

Dans la table Partie Jouée :

- On décide de stocker les mots essayés en une seule chaîne de caractères avec des séparateurs comme "|". Mais attention, il faudra donc bloquer le caractère séparateur lors de l'entrée clavier. On pourra aussi tester le mot dans la base de données.
- On rajoute aussi le nombre d'essais maximum et le nombre de mots tentés.

On affichera les scores en faisant une requête SQL à la base de données

État de l'art :

On s'est demandé sous quelle forme on voulait faire l'état de l'art : un tableau ou un texte.

On a opté pour un texte.

Pour l'instant, on réalise l'état de l'art seulement sur la partie application.

L'état de l'art est à développer.

Gantt :

Lucie a divisé l'implémentation en deux parties : l'application et le solveur.

Pour ce qui est de l'application, on peut la diviser en base de données, site web, front, Algorithmie.

Le Gantt prévisionnel du solveur sera fait plus tard, quand nous aurons plus de bagages en C et en structures de données.

Fonctionnalités du jeu à implémenter :

Obligatoirement, le joueur doit pouvoir choisir la taille du mot et le nombre d'essais maximum qu'il s'accorde de faire.

On peut, en plus, chronométrier les parties, créer plusieurs niveaux de difficultés :

- facile : première et dernière lettres données
- moyen : première lettre donnée
- difficile : sans aucune lettre donnée

Le principe de récompense a été aussi évoqué. On va mettre un système de points à déclencher pour obtenir des indices.

Exemples d'indices :

- éliminer des lettres dans les possibilités
- placer une lettre

On a aussi envie de créer des dictionnaires personnalisés avec différents thèmes.

Séance de code du 24 mars : quoi faire ? Comment répartir le travail ? :

Algorithmie :

- fonction qui vérifie si un mot existe dans le dictionnaire
 @param : mot
 @return : bool
- fonction qui compare deux mots
 @param : essai, mot
 @return : int de la forme “012001” avec :
 0 : lettre n'est pas dans le mot
 1 : lettre mal placée
 2 : lettre bien placée

Réaliser toute la partie front.

Réaliser le squelette flask avec la gestion de session.

Implémenter la base de données.

TODO List :

To Do	Qui ?	Pour Quand ?
WBS	Lucie	27 mars
Matrice RACI	Tout le monde	1 avril
Squelette flask	Lucie	24 mars
Front	Rémi	24 mars
Base de données	Mathilde	24 mars
Algorithmie	Lola	24 mars
Etat de l'art à finaliser	Tout le monde	27 mars

Prochaine Réunion : 27 mars 2022

7.1.3 Réunion du 27 Mars

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 27 Mars à 14h

Ordre Du Jour :

- Retour sur la TODO du 24 mars
- Visuel de l'application
- Implémentation du jeu
- Ajout de pop-ups en javascript sur les pages inscription et connexion
- Requêtes SQL sur le fichier app.py
- Adapter l'interface HTML à flask
- Transférer le projet sur le git dédié

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
WBS	Lucie	Lucie n'a pas eu le temps de le faire
Squelette Flask	Lucie	Fait
Front	Rémi	C'est bien avancé, on va en rediscuter pour voir si cela convient à tout le monde
Base De Données	Mathilde	C'est fait, les lignes de code pour créer les tables sont dans un .txt sur le git
Algorithmie	Lola	Fonctions existe() et compare() codées, le dictionnaire a été traité (toutes les lettres sont en MAJ)
Etat de l'art à finaliser	Tout le monde	Ce n'est pas fait

Le visuel de l'application :

Tout le monde est satisfait par le visuel, il faudra préciser que le mot de passe doit faire un minimum de 8 caractères. Les boutons "se connecter" et "s'inscrire" sont les boutons de base de CSS, on peut aussi les personnaliser un peu plus.

Implémentation du jeu :

On va séparer l'implémentation en 2 parties :

- 1. Le joueur tape le mot dans les cases
- 2. Notre fichier .py récupère le mot, l'analyse et colore les cases.

On peut aller voir le code des applications "type wordle" open source pour s'inspirer.

Pop-ups sur les pages "Inscription" et "Connexion" :

Rémi aimerait que pour envoyer un message d'erreur lors de la connexion ou de l'inscription cela soit sous forme de pop-ups codées en JavaScript. Cela permettra de ne pas recharger la page.

Requêtes SQL sur le fichier app.py :

Lucie a fait les requêtes pour l'inscription et la connexion.

Les autres requêtes par rapport aux parties doivent être faites en parallèle de l'implémentation du jeu.

Adapter l'interface HTML à flask

Il faut le faire mais normalement Rémi et Lucie se sont coordonnés pour les noms des variables dans les forms. Il faut sûrement ajuster quelques lignes de code tout de même.

Transférer le projet sur le git dédié

Lola va s'occuper de tout transférer.

TODO List :

To Do	Qui ?	Pour Quand ?
WBS	Lucie	3 avril
Préciser que le mot de passe doit faire 8 caractères	Rémi	3 avril
Les Pop-ups en JavaScript sur les pages "Connexion" et "Inscription"	Rémi	3 avril
Faire les requêtes pour la pages "Mes Infos"	Lola	3 avril
Faire des dictionnaires par taille de mots	Mathilde	3 avril
Faire les requêtes pour la pages "Historique"	Mathilde	3 avril
Implémentation du jeu	Lucie et Rémi	3 avril
Etat de l'art à finaliser	Lola	3 avril

Prochaine Réunion : 3 avril 2022

7.1.4 Réunion du 3 Avril

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 3 avril à 15h

Ordre Du Jour :

- Retour sur la TODO du 27 mars
- Implémentation du jeu
- Etat de l'art
- Document de conception
- Interface graphique

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
WBS	Lucie	Il est fait, il faut le mettre sur le drive
Préciser que le mot de passe doit faire 8 caractères	Rémi	C'est fait
Les Pop-ups en JavaScript sur les pages "Connexion" et "Inscription"	Rémi	Ce n'est pas encore fait
Faire les requêtes pour la pages "Mes Infos"	Lola	C'est fait mais graphiquement ce n'est pas joli à regarder
Faire des dictionnaires par taille de mots	Mathilde	C'est fait
Faire les requêtes pour la pages "Historique"	Mathilde	C'est fait mais graphiquement ce n'est pas joli à regarder
Implémentation du jeu	Lucie et Rémi	La page "configuration" a été faite et la grille se génère d'une taille différente en fonction des configurations choisies
Etat de l'art à finaliser	Lola	C'est fait, il est prêt pour le document de conception et le rapport

Implémentation du jeu :

Ce qu'il faut faire :

- Réaliser l'écoute clavier
- Récupérer les mots tapés
- Vérifier si le mot existe
- Retourner un tuple du type [0,1,2,0,0] pour les couleurs
- Gérer l'affichage des couleurs en fonction du tuple
- Gérer l'événement "Gagné !!" et "Perdu ..."

Pour les couleurs, on prend le rouge si la lettre est bien placée, le orange ou le jaune si la lettre est dans le mot mais mal placée et aucune couleur sinon.

Etat de l'art :

Lola a fini l'état de l'art et demande si le reste du groupe à des remarques
 Mathilde et Rémi pensent qu'il faut rajouter des colonnes dans le tableau recensant les applications déjà existantes. Nous nous sommes donc mis d'accord sur les colonnes à rajouter

- Si l'application est libre de droit
- Si il y a une option multi-joueurs
- Si le joueur peut choisir la taille du mot

Document de conception :

Ce qu'il faut faire :

- Réaliser un schéma de la base de données
- Réaliser un schéma de l'application
- Réunir l'état de l'art, les deux schémas dans un seul document

Il faut envoyer le mail à M. Festor, M. Da Silva et M. Oster.

Interface graphique :

Pour le moment tout le monde est satisfait de ce qui à été fait, il faut maintenant rajouter du graphisme sur la page "Historique" et la page "Mes Infos"

TODO List :

To Do	Qui ?	Pour Quand ?
Mettre le WBS sur le drive	Lucie	7 avril
Reprendre l'état de l'art	Mathilde	7 avril
Rendre jolies les pages "Historique" et "Mes Infos"	Rémi	7 avril
Écoute clavier	Lucie et Lola	7 avril
Document de conception	Mathilde et Lola	7 avril
Les Pop-ups en JavaScript sur les pages "Connexion" et "Inscription"	Rémi	7 avril

Prochaine Réunion : 7 avril 2022

7.1.5 Réunion du 7 Avril

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 7 avril

Ordre Du Jour :

- Retour sur la TODO du 3 avril
- Document de conception
- Problème de Base de Données sur le git
- Coloration des cases
- Connexion avec les dictionnaires
- Interface graphique

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
Mettre le WBS sur le drive	Lucie	Il est sur le drive
Reprendre l'état de l'art	Mathilde	L'état de l'art est fini
Rendre jolies les pages "Historique" et "Mes Infos"	Rémi	"Mes Infos" est faite mais pas la page "Historique"
Écoute clavier	Lucie et Lola	Les cases du jeu sont remplies avec le clavier physique et/ou virtuel le joueur peut choisir
Document de conception	Mathilde et Lola	C'est fait
Les Pop-ups en JavaScript sur les pages "Connexion" et "Inscription"	Rémi	C'est fait grâce à la programmation asynchrone

Document de conception :

Mathilde et Lola l'ont réalisé. Rémi et Lucie suggèrent qu'il faut plus détailler le schéma de l'application en ajoutant les connexions entre python, la base de données et les pages HTML.

Dans le schéma de la base de données, il manque le type des objets et une légende. Pour le moment il est dans un document word, on a décidé qu'il fallait le rédiger en LaTeX.

Problème de Base de Données sur le git :

Cette semaine nous avons tous eu des problème de merge sur le git à cause de la base de données.

Nous avons donc décidé que chacun garde sa base de données en local. Si jamais il faut modifié la base de données, on change le fichier qui contient le code qui génère la base de données.

Coloration des cases :

Il faut le faire, le code couleur est dans le compte rendu de la réunion du 3 avril.

Connexion avec les dictionnaires :

Pour traiter le mot rentré par le joueur, nous allons devoir faire cela avec de la programmation asynchrone afin de ne pas recharger la page à chaque fois que l'on vérifie si le mot existe.

Interface graphique :

L'image de fond se dilate en fonction de ce que l'on affiche. Afin qu'elle reste jolie, Lucie propose de limiter le nombre d'essais entre 2 et 10 et de limiter le nombre de lettres entre 3 et 20.

Le clavier virtuel n'est pas très joli, il faut améliorer son visuel.

TODO List :

To Do	Qui ?	Pour Quand ?
Gestion des dictionnaires	Mathilde	18 avril
Écrire les règles du jeu	Lola	18 avril
Coloration des cases	Mathilde et Lola	18 avril
Connexion avec le dictionnaire	Rémi	18 avril
Schéma application	Lola	18 avril
Mettre à jour le document de conception	Mathilde	18 avril
Améliorer le visuel du clavier et de l'historique	Lucie	18 avril

Prochaine Réunion : 18 avril 2022

7.1.6 Réunion du 18 Avril

Type : Réunion hebdomadaire	Lieu : à distance
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 18 avril

Ordre Du Jour :

- Retour sur la TODO du 7 avril
- Listage des éléments qui ne fonctionnent pas
- Version en anglais de notre application
- Relier complètement la base de données à l'application
- Gérer les modes facile, moyen difficile
- Validation de l'algorithme de coloration
- Fixer les règles pour le Mot du Jour

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
Gestion des dictionnaires	Mathilde	C'est fait
Écrire les règles du jeu	Lola	Elles sont écrites dans un fichier sur le drive, elles sont aussi sur le site dans le footer mais cela ne rend pas bien donc on va les mettre dans un onglet à part
Coloration des cases	Mathilde et Lola	La coloration n'est pas encore au point
Connexion avec le dictionnaire	Rémi	Réalisé complètement, avec un appel pour sélectionner le mot, puis la partie est générée dans le fichier jeu.js.
Schéma application	Lola	c'est fait
Mettre à jour le document de conception	Mathilde	C'est fait, il a été envoyé à l'équipe pédagogique et nous avons tous reçu le retour plutôt positif
Améliorer le visuel du clavier et de l'historique	Lucie	Le clavier n'a pas été modifié mais l'historique est complètement réalisé

Listage des éléments qui ne fonctionnent pas :

Lola relève le fait qu'on ne peut pas accéder au menu déroulant "mon compte" lorsqu'on est sur la page "historique" et la page "mes infos", Rémi pense que cela vient d'une superposition de feuille de style.

Le dernier essai n'est pas pris en compte car cela renvoie directement "tu as réalisé trop d'essais". C'est un problème d'indices facilement réglable.

Version en anglais de notre application :

Il faut faire un bouton sur la page "config" pour choisir la langue puis gérer les appels aux différents dictionnaires.

Relier complètement la base de données à l'application :

Dans la table partie, il faut rajouter un attribut "mot à deviner". On peut le crypter avec l'algo crypto() pour qu'il ne soit pas lisible à l'oeil nu.

Il faut aussi stocker les parties dans la base de données, en faisant attention aux contraintes de la base de données :

- contrainte sur la langue 'en' ou 'fr'
- contrainte sur le niveau 0, 1 ou 2
- contrainte sur le mode de jeu 'mdj' ou 'libre'

Pour stocker les parties, il faut écrire une ligne dans la base de données à chaque début de partie et incrémenter le nombre de parties jouées dans la table Joueur et mettre à jour cette ligne à chaque nouveau mot tenté.

Gérer les modes facile, moyen difficile :

Il faut faire des boutons sur la page "config" pour choisir le niveau. Puis gérer les niveaux dans le fichier jeu.js en affichant plus ou moins de lettres dans le tableau de cases.

Validation de l'algorithme de coloration :

Les fonctions ne fonctionnent pas, Mathilde ne voit pas pourquoi. Il faut qu'une autre personne regarde.

Fixer les règles pour le Mot du Jour :

niveau : difficile
5 lettres pour le mot
6 essais pour jouer

TODO List :

To Do	Qui ?	Pour Quand ?
Gérer la superposition des fiches de style pour la page historique	Rémi	25 avril
Créer un onglet Règles du jeu et l'enlever du footer	Lola	25 avril
Gérer la version en anglais	Mathilde	25 avril
Gérer les niveaux	Lola	25 avril
Algorithme de coloration	Lucie	25 avril
Gérer les parties "Mot du jour" dont créer un algorithme qui génère les 300 prochains mots du jour.	Rémi	25 avril
Améliorer le visuel du clavier	Lucie	25 avril
Gérer le stockage des parties dans la base de données	Mathilde	25 avril
Stocker le mot à deviner dans partie	Mathilde	25 avril

Prochaine Réunion : 25 avril 2022

7.1.7 Réunion du 25 Avril

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 25 avril

Ordre Du Jour :

- Retour sur la TODO du 18 avril
- Réaliser un bouton "nouvelle partie" à la fin d'un partie
- Trier les dictionnaires car certains mots inappropriés
- Définir quelles requêtes SQL on doit faire
- Gérer le niveau de difficulté et le changement de langue
- Préparer des requêtes SQL pour la soutenance

Retour sur la TODO de la semaine dernière :

TO DO	Qui ?	Quand est-il ?
Gérer la superposition des fiches de style pour la page historique	Rémi	Rémi n'a pas réussi et propose de repartir à zéro pour la page historique
Créer un onglet Règles du jeu et l'enlever du footer	Lola	Il est fait et joli à regarder
Gérer la version en anglais	Mathilde	Mathilde a essayé mais cela ne fonctionne pas totalement
Gérer les niveaux	Lola	Lola a essayé mais cela ne fonctionne pas
Algorithme de coloration	Lucie	C'est fait le case se colorent correctement en fonction du mot à deviner
Gérer les parties "Mot du jour" dont créer un algorithme qui génère les 300 prochains mots du jour.	Rémi	C'est fait
Améliorer le visuel du clavier	Lucie	C'est fait, en plus le clavier se colore en fonction des essais et du mots à deviner
Gérer le stockage des parties dans la base de données	Mathilde	Il fallait attendre que les niveaux et langues soient gérés donc Mathilde n'a pas pu le faire
Stocker le mot à deviner dans partie	Mathilde	Il fallait attendre que les niveaux et langues soient gérés donc Mathilde n'a pas pu le faire

Réaliser un bouton "nouvelle partie" à la fin d'un partie :

Mathilde propose de faire deux boutons :

- un pour rejouer avec les mêmes configurations
- un pour rejouer avec d'autres configuration, il renverrai donc vers la page configuration

Il faut faire ces deux boutons que lorsque le mode libre est choisi car on ne peut pas refaire une partie en mode mot du jour.

Trier les dictionnaires car certains mots inappropriés :

Pour ce qui est du dictionnaire en anglais, Mathilde en a déjà téléchargé un. Il faut juste le trier et le mettre en majuscule.

Pour ce qui est du dictionnaire français, il faut enlever les mots avec des tirets.

Définir quelles requêtes SQL on doit faire :

Il faut enregistrer les parties jouées dans la base de données. Pour cela, dès le début de la partie, on crée une ligne dans la table partie. A chaque mot tenté, on met à jour cette ligne. Il ne faut pas oublier d'indenter le nombre de parties jouées dans la table joueur.

Pour le mode mot du jour, il faut regarder dans la base de données si le joueur a déjà joué la partie "mot du jour" ce jour-là.

Gérer le niveau de difficulté et le changement de langue :

Pour la langue, Mathilde sait comment régler son problème.

Pour la gestion des niveaux, il faut qu'en fonction du niveau, on puisse afficher les indices dans le tableau de cases directement ou dans un encadré à côté.

Préparer des requêtes SQL pour la soutenance :

Lucie et Mathilde avaient déjà fait des requêtes SQL pour remplir la base de données, on peut récupérer celles-ci et y ajouter des SELECT et UPDATE.

TODO List :

To Do	Qui ?	Pour Quand ?
Faire un exemple dans la page règles du jeu	Lola	28 avril
Gérer l'aspect de configuration	Rémi et Mathilde	28 avril
Gérer l'aspect de historique et règles	Rémi et Mathilde	28 avril
Faire un readme.md	Lucie	28 avril
Mettre à jour le fichier qui contient les requêtes SQL en vue de la soutenance	Lucie et Mathilde	28 avril
Deux boutons sur la page jeu “rejouer”	Lucie et Lola	28 avril
Gestion des niveaux	Lucie et Lola	28 avril
Gestion des dictionnaires	Mathilde	28 avril
Relier entièrement l'application à la base de données	Tout le monde	28 avril

Prochaine Réunion : 28 avril 2022

7.1.8 Réunion du 28 Avril

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 28 avril

Ordre Du Jour :

- Retour sur la TODO du 18 avril
- Tour de table si on a détecté des problèmes au niveau du jeu ou si on a des suggestions d'amélioration
- Requêtes SQL pour écrire l'avancée des parties dans la base de données
- Discussion par rapport à la page "historique"
- Comment se démarquer ?

Retour sur la TODO de la semaine dernière :

To Do	Qui ?	Quand est-il ?
Faire un exemple dans la page règles du jeu	Lola	C'est fait, avec des captures d'écran de l'application
Gérer l'aspect de configuration	Rémi et Mathilde	C'est presque terminé
Gérer l'aspect de historique et règles	Rémi et Mathilde	Règles c'est fait, historique c'est en cours
Faire un readme.md	Lucie	Le readme est réalisé, Lola l'a relu
Mettre à jour le fichier qui contient les requêtes SQL en vue de la soutenance	Lucie et Mathilde	C'est fait, il y a toute sorte de requêtes type
Deux boutons sur la page jeu "rejouer"	Lucie et Lola	C'est fait finalement par Mathilde
Gestion des niveaux	Lucie et Lola	C'est fait
Gestion des dictionnaires	Mathilde	C'est fait
Relier entièrement l'application à la base de données	Tout le monde	C'est pas encore totalement fait

Tour de table si on a détecté des problèmes au niveau du jeu ou si on a des suggestions d'amélioration :

Mathilde : pour le niveau facile, au niveau des indices, il y a un problème de cohérence car il faut taper la dernière contrairement à la première.

Solution possible : lorsqu'on tape le mot, les lettres disparaissent.

Lucie : enlever les boutons rejouer sur la page jeu quand mode == mot du jour.

Rémi : Page classement.

Lola : une célébration style confetti est une bonne idée lors de l'événement victoire.

Requêtes SQL pour écrire l'avancée des parties dans la base de données :

- Importer depuis le fichier jeu.js les mots tentés et à deviner : requêtes pour push chaque partie avec les réglages les mots tentés, le mot à deviner et l'id de la partie.
- Récupérer sur app.py les mots tentés quand ils sont testés pour savoir si oui ou non ils existent dans le dictionnaire.
- Incrémenter le nombre de parties jouées dans la table joueur.

Discussion par rapport à la page "historique" :

On peut rendre moins opaque le tableau actuel.

On peut mettre un "victoire" plutôt que d'afficher le mot a deviné.

On peut ajouter dans la base de données le nombre de victoire.

Comment se démarquer ? :

Nous n'avons pas assez de temps pour se démarquer plus. Donc on ajoute rien.

TODO List :

To Do	Qui ?	Pour Quand ?
Connecter entièrement la base de donné au site	Lola et Lucie	29 avril
Confettis en cas de victoire	Mathilde	29 avril
Classement	Rémi	29 avril
Enlever bouton page jeu mot du jour	Mathilde	29 avril
Gérer la cohérence dans les indices	Lucie	29 avril
Aspect de la page historique	Rémi ou Lola	29 avril

Prochaine Réunion : 8 mai 2022

7.1.9 Réunion du 8 Mai

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 8 mai

Ordre Du Jour :

- Etat de l'art sur la partie solveur
- Conception solveur : différentes étapes et tests
- Rédaction Rapport : répartition des premières tâches

Etat de l'art sur la partie solveur :

L'état de l'art de la partie solveur a été commencé par Lola. On peut le retrouver à la racine du drive.
Lucie a lu des articles et les vidéos proposés par l'équipe pédagogique. On peut pondérer les mots en fonction de leur coloration.

Conception solveur : différentes étapes et tests :

Rémi propose de faire des statistiques sur toutes les versions du solveur pour montrer les différentes étapes sur l'avancée du solveur.

Pour la première version du solveur, on peut commencer par choisir un random dans un dictionnaire. Ce dictionnaire sera réduit à chaque tour en fonction de la coloration.

Quelles fonctions doit-on coder ?

- fonction qui met à jour un tableau d'information en fonction des couples (essais,colorations).
- fonction qui trie un dictionnaire en fonction du tableau d'information
- fonction random : elle choisit aléatoirement un mot dans un dictionnaire
- toutes sortes de fonctions de statistiques

Rédaction Rapport : répartition des premières tâches :

Il faut faire le schéma E/A de la base de données, le schéma décrivant la structure du site, décrire l'algorithmie présente derrière notre application et décrire l'application page par page.

TODO List :

To Do	Qui ?	Pour Quand ?
Etat de l'art	Lola	14 mai
Schéma base de données	Lucie	14 mai
Schéma application	Rémi	14 mai
Partie algorithmie sur le rapport	Mathilde	14 mai
Fonction qui update le tableau d'information en fonction du couple (essai,coloration)	Lola	14 mai
Fonction de tri du dictionnaire par rapport au tableau d'information	Mathilde	14 mai
Fonction main interface et Fonction random mot	Lucie	14 mai
Statistiques	Rémi	14 mai

Prochaine Réunion : 14 mai 2022

7.1.10 Réunion du 14 Mai

Type : Réunion hebdomadaire	Lieu : rue de la commanderie
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 14 mai

Ordre Du Jour :

- Retour sur la TODO de la semaine dernière
- Avancement du solveur, les prochaines étapes
- Organisation d'une séance de code
- Rédaction Rapport : avancement

Retour sur la TODO de la semaine dernière :

To Do	Qui ?	Quand est-il ?
Gantt	Lola	Gantt prévisionnel fait mais pas le réel
Etat de l'art	Lola	Pas avancé
Rapport BDD	Lucie	Pas fait
Rapport page par page	Lola	Fait page par page à relire
Rapport structure du site (schema)	Rémi	Pas fait
Rapport algorithmie	Mathilde	Pas fait
fonction algo épuration dictionnaire	Mathilde	Débuté, 3 "sous-fonctions" de prêtes, calcule la taille du dictionnaire, calcule si elle contient un mot, calcule si le mot contient une lettre à une certaine position
Fonction tableau d'information lettres	Lola	Essayé mais ne fonctionne pas
Main interface +Fonction random mot	Lucie	Random fait, main non fait

Avancement du solveur, les prochaines étapes :

Au vu du point sur le retour de la TODO de la dernière réunion, tout le monde est bloqué dans ses tâches. Il faut donc continuer sa partie.

Si une personne est vraiment bloquée, elle demande l'aide aux trois autres qui essaieront de la débloquer.

Tous les membres du groupe confirment qu'il est important d'avancer au vu l'approche de la date de rendu du projet.

Organisation d'une séance de code :

Lola demande quelle date arrange tout le monde et souhaite faire la réunion au plus vite. Après quelques échanges, il est convenu de faire d'une séance de code le soir même à 18h afin de débloquer principalement Mathilde et Lola.

Si possible, une autre séance de code sera programmé avant la prochaine réunion.

Rédaction Rapport, avancement :

Lucie, Rémi et Mathilde n'ont pas eu le temps d'avancer leur parties au vu des examens et du reste du projet. Ils le feront donc pour la prochaine réunion.

Lola propose de rédiger l'introduction du rapport ainsi qu'un préambule du chapitre "Page par page" du rapport.

TODO List :

To Do	Qui ?	Pour Quand ?
Gantt Réel	Lucie	18 mai
Etat de l'art	Lola	18 mai
Partie Base de données du rapport	Rémi	18 mai
Schéma application	Rémi	18 mai
Partie algorithmie sur le rapport	Mathilde	18 mai
Corriger les erreurs du code du tableau des mots	Lola	18 mai
Corriger les erreurs de code de la fonction d'épuration des mots du dictionnaire	Mathilde	18 mai
Fonction main interface	Lucie	18 mai
Statistiques	Rémi	18 mai
Rédaction introductions du rapport et du chapitre Page par Page	Lola	18 mai

Prochaine Réunion : 18 mai 2022

7.1.11 Réunion du 18 Mai

Type : Réunion hebdomadaire	Lieu : Télécom Nancy
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 18 mai

Ordre Du Jour :

- Retour sur la TODO de la dernière réunion
- Choix de la structure de données
- Point sur l'algorithme d'épuration du dictionnaire

Retour sur la TODO de la dernière réunion :

To Do	Qui ?	Quand est-il ?
Gantt Réel	Lucie	Pas fait
Etat de l'art	Lola	Pas fait
Partie Base de données du rapport	Lucie	Fait mais à modifier, et il faut rédiger l'introduction de cette partie
Partie introduction du rapport	Lola	Pas fait
Partie structure du site du rapport	Rémi	Fait, à relire
Partie algorithmie du rapport	Mathilde	Commencé
Fonction épuration du dictionnaire	Mathilde	Toutes les "sous-fonctions" fonctionnent, il reste la fonction principale à faire
Faire fonctionner la fonction tableau	Lola	Fait, mais il reste la fonction update_tableau à faire
Interface principale	Lucie	Lecture du fichier wsolf.txt, on attend les autres fonctions pour pouvoir continuer
Statistiques	Rémi	un algorithme de coloration a été fait pour pouvoir tester sur beaucoup de mots

Choix de la structure de données :

Après de nombreuses recherches, Lucie a trouvé que la meilleure solution serait de réaliser un graphe contenant des mots, ce qui reviendrait à réaliser un automate.

L'automate lirait tous les mots, puis on le minimisera, ce qui permettrait de relier certaines branches entre elles.

Une première ébauche d'implémentation pour le graphe serait une struct liste d'états initiaux et de transitions.

Il faudrait aussi réaliser un algorithme de minimisation qui serait appelé à chaque ajout de mot dans l'automate. A chaque essai, l'automate sera épuré puis à nouveau minimisé.

Lien de l'article concernant les automates : <https://jbp.dev/blog/dawg-basics.html>

Lien vers un exemple d'implémentation en C pour les automates : <https://jbp.dev/blog/dawg-first-implementation.html>

Point sur l'algorithme d'épuration du dictionnaire :

Mathilde se questionne sur comment parcourir le tab[30] car à chaque essai elle n'a pas besoin de parcourir le grand tableau. Mathilde se demande si créé des "sous-tableaux" du grand tableau ne serait pas mieux.

Après discussion, il est décidé qu'on ne créé pas ces "sous-tableaux", et qu'on parcourt le grand tableau à chaque essai.

TODO List :

To Do	Qui ?	Pour Quand ?
Gantt Réel	Lucie	22 mai
Etat de l'art	Lola	22 mai
Partie introduction du rapport	Lola	22 mai
Automatisation du solveur	Rémi	22 mai
Partie algorithmie	Mathilde	22 mai
Corriger les erreurs du code du tableau des mots	Lola	22 mai
Fonction d'épuration des mots du dictionnaire	Mathilde	22 mai
Structure de données et fonctions de base	Lucie et Lola	22 mai

Prochaine Réunion : 22 Mai 2022

7.1.12 Réunion du 22 Mai

Type : Réunion hebdomadaire	Lieu : Rue de la commanderie, Nancy
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 22 mai

Ordre Du Jour :

- Retour sur la TODO de la dernière réunion
- Choix final de la structure de données
- Etapes restantes
- Séances de code avant le rendu final
- Rédaction rapport : partie manquante

Retour sur la TODO de la dernière réunion :

To Do	Qui ?	Quand est-il ?
Gantt Réel	Lucie	Pas fait
Etat de l'art	Lola	Pas fait
Partie introduction du rapport	Lola	22 mai
Automatisation du solveur	Rémi	Rémi a continué à travailler dessus mais n'a pas terminé
Partie algorithmie	Mathilde	Fait, à relire par Lola et Rémi
Corriger les erreurs du code du tableau des mots	Lola	Fait
Fonction d'épuration des mots du dictionnaire	Mathilde	Mathilde a continué à travailler dessus mais la fonction principale reste toujours à réaliser
Structure de données et fonctions de base	Lucie et Lola	Fait mais des changements à venir

Choix final de la structure de données :

Le graphe sera un graphe acyclique dirigé.

Lien vers un code dont on peut s'inspirer : GitHub - chalup/dawggenerator : Fast directed acyclic word graph generator

Etapes restantes :

En C :

- Implémentation structure et minimisation
- Adaptation des fonctions à la nouvelle structure de données
- Interface (main qui communique avec les autres fichiers)

En gestion de projet :

- Gantt réel des 2 parties du projet
- Etat de l'art

Dans la rédaction du rapport :

- Partie ressenti personnel pour chacun
- Partie solveur
- Rédaction du read.me pour le solveur

Séances de code avant le rendu final :

Le week-end de l'ascension sera une superbe opportunité pour coder ensemble.

Rédaction des parties manquantes du rapport :

En C :

- La version 1 du solveur
- La version 2 du solveur

Le tableau des heures de travail. Les compte-rendus de réunion à partir du 8 mai. Les parties personnelles de chacun.

Insérer les signatures de chacun.

TODO List :

To Do	Qui ?	Pour Quand ?
Gantt Réel	Lola	26 mai
Etat de l'art	Mathilde	26 mai
Rédaction du rapport	Lola et Mathilde	26 mai
Relire le rapport	Mathilde	26 mai
Automatisation solveur	Rémi	26 mai
Relecture partie algorithme du rapport	Rémi	26 mai
Fonction d'épuration des mots du dictionnaire	Mathilde	26 mai
Implémentation et réflexion sur la structure de données (elle sous-traitera si besoin)	Lucie	26 mai

Prochaine Réunion : 26 Mai 2022

7.1.13 Réunion du 26 Mai

Type : Réunion hebdomadaire	Lieu : Messenger
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 26 mai

Ordre Du Jour :

- Retour sur la TODO de la dernière réunion
- Point sur la situation
- Discussion sur certains points du rapport
- Organisation pour la fin du projet

Retour sur la TODO de la dernière réunion :

To Do	Qui ?	Quand est-il ?
Gantt Réel	Lola	Fait et mis sur le drive
Etat de l'art	Mathilde	Pas avancé mais une piste sur une partie sur les graphes
Rédaction du rapport	Lola et Mathilde	le tableau des heures de travail est fait ainsi que les comptes rendus
Relecture du rapport	Mathilde	Fait, elle a corrigé des fautes d'orthographe
Automatisation du solveur	Rémi	Presque fini, tout marche il faut juste réussir à vider le tableau
Epuration du dictionnaire	Mathilde	Tout fonctionne
Relecture de la partie algorithmie du rapport	Rémi	Fait, tout lui semble cohérent
Implémentation de la structure	Lucie	Elle a bien avancé, d'ici demain elle pourra passer à autre chose
Gestion de projet	Lola et Mathilde	Pas avancé

Point sur la situation :

Lola souhaite connaître le ressenti de tout le monde concernant l'avancé du projet. Chaque membre travaille beaucoup et ait confiant quant à l'avancé du projet.

Rémi dit qu'il faut adapter la fonction des dictionnaires pour faire marcher la version 1 mais que sinon la v1 du solveur est prête.

Concernant la version 2 de notre solveur, la structure a été bien avancée par Lucie et dans les prochains jours, elle va passer à l'implémentation des graphes. Lola va l'aider à coder l'algorithme de minimisation.

Mathilde et Rémi, quant à eux, vont s'attaquer dès demain à la théorie de l'information, cela consiste à pondérer le poids de chaque lettre en fonction de sa place dans le mot.

Discussion sur certains points du rapport :

Lola aimeraient connaître l'avis des autres membres concernant certains points du rapport : les noms des titres de parties dans le tableau des tâches de travail. Lucie et Mathilde proposent des idées pour améliorer le tableau.

Concernant la partie du rapport sur le solveur, il a été décidé de changer le terme *algorithme de résolution* par solveur car cela englobe plus les parties structure de données et algorithmie du solveur.

Rémi pense qu'il faudrait rajouter dans la partie Application Web, un paragraphe sur la complexité des algorithmes et un paragraphe sur les tests réalisés sur les différentes fonctions.

Organisation pour la fin du projet :

Rémi et Mathilde vont coder ensemble la partie sur la théorie de l'information. Rémi va également faire des graphiques sur différentes statistiques afin d'apporter des éléments à notre rapport.

Lucie conseille à Lola de réaliser des schémas d'hash map, de minimisation de graphe,... afin qu'elle soit prête pour coder ces parties-là quand Lucie aura fini la partie structure.

TODO List :

To Do	Qui ?	Pour Quand ?
CR de réunion	Mathilde	28 mai
Etat de l'art	Mathilde	28 mai
Rédaction rapport, partie solveur	Mathilde et Lola	28 mai
Rédaction rapport, partie schéma	Lola	28 mai
Entropie, hash map	Mathilde et Rémi	28 mai
Structure avancée	Lucie	28 mai
Statistiques	Rémi	28 mai
Interface (relire le sujet du projet)	Lola	28 mai

Prochaine Réunion : 28 mai 2022

7.1.14 Réunion du 28 Mai

Type : Réunion hebdomadaire	Lieu : Messenger
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 28 mai

Ordre Du Jour :

- Retour sur la TODO de la dernière réunion
- Que reste-t-il à faire ? Avancements de la V1 et de la V2
- Points schémas/statistiques : que mettre dans la hashmap et quelles fonctions à schématiser ?
- Travail sur la structure de graphe

Retour sur la TODO de la dernière réunion :

To Do	Qui ?	Quand est il ?
CR de réunion	Mathilde	Fait
Etat de l'art	Mathilde	Pas Fait
Rédaction rapport, partie solveur	Mathilde et Lola	La partie épuration des dictionnaires est faite
Rédaction rapport, partie schéma	Lola	Fait : schéma graphe, partie équipe projet, modèle relationnel, rajout WBS, changer entité-association
Entropie, hash map	Mathilde et Rémi	La structure est faite, il reste une fonction, il faut changer la pondération des lettres
Structure avancée	Lucie	Fonction qui ajoute un mot dans le graphe et fonction qui lit le txt et qui remplit le graphe (reste à minimiser : étapes à faire)
Statistiques	Rémi	Toujours sur l'interface auto, les statistiques sont en cours
Interface (relire le sujet du projet)	Lola	Commencé mais erreurs de code donc Rémi a repris le code.

Que reste-t-il ? Etat de la version 1 et Etat de la version 2 :

Il faut encore rendre l'interface main_auto fonctionnelle.

Il faut reprendre la vidéo sur la théorie de l'information pour choisir la bonne formule pour la pondération de chaque lettre dans la hashmap.

La structure de hash map est créée ainsi que les fonctions de création et de destruction, il faut écrire la fonction qui remplit la hash map.

Points schémas/statistiques : que mettre dans la hashmap et quelles fonctions à schématiser ? :

Lola aimerait connaître plus de détails sur la structure de la hashmap afin de réaliser le schéma qui colle au mieux.

Mathilde et Lucie explique que dans la hashmap de taille $26 * \text{taille_mot}$ chaque élément contient la pondération de la lettre et un pointeur vers la prochaine case. Si on veut la pondération de la lettre 'c' à la troisième position dans un mot, il faut accéder à la troisième colonne de la hashmap et prendre le troisième élément de la liste.

Travail sur la structure de graphe :

Pour le moment, Lucie a fait toute la structure et a créé une fonction qui lit un fichier txt et remplit le graphe.

Il faut, à présent, créer une fonction qui renverra, en fonction d'un mot et d'un noeud, un tableau contenant les id des arêtes du graphe pour créer le mot. Il reste encore toutes les étapes de la minimisation :

- retirer les états inatteignables
- faire un tableau de booléens
- gérer le fait que deux états soient équivalents
- recréer le graphe post minimisation

TODO List :

To Do	Qui ?	Pour Quand ?
Refaire le schéma BDD + le mettre sur le rapport à la place de l'autre et ajouter la case victoire	Rémi	29 mai
Ajouter le mock up au rapport/ ajouter les photos du site dans les annexes avec référence dans page à page	Lola	29 mai
Introduction diagramme de Gantt et différence entre prévisionnel et réel	Lola	29 mai
Changer formule pondération	Mathilde	29 mai
Fonction qui rempli la hash map	Mathilde	29 mai
Fonction get chemin/ minimisation	Lucie et Lola	29 mai
Statistiques et rendre interface main(auto)fonctionnel	Rémi	29 mai
Partie difficultés : le C/ Struct donnée changée plein de fois/ Web : JS algo coloration	Lola	29 mai

Prochaine Réunion : 29 mai 2022

7.1.15 Réunion du 29 Mai

Type : Réunion hebdomadaire	Lieu : Rue de la Commanderie, Nancy
Présents : Lola Montignier, Mathilde Bard, Lucie Boucher et Rémi Bourdais	Date : 29 mai

Ordre Du Jour :

- Retour sur la TODO de la dernière réunion
- Que reste-t-il à faire ?
- Organisation de la semaine

Retour sur la TODO de la dernière réunion :

To Do	Qui ?	Quand est il ?
Refaire le schéma BDD + le mettre sur le rapport à la place de l'autre et ajouter la case victoire	Rémi	Pas fait
Ajouter le mock up au rapport/ ajouter les photos du site dans les annexes avec référence dans page à page	Lola	Fait
Introduction diagramme de Gantt et différence entre prévisionnel et réel	Lola	Fait
Changer formule pondération	Mathilde	Pas fait
Fonction qui rempli la hash map	Mathilde	Fait
Fonction get chemin/ minimisation	Lucie et Lola	Les fonctions get chemin et suppression des états inaccessibles sont faites.
statistiques et rendre interface main(auto)fonctionnel	Rémi	L'interface fonctionne et les statistiques sont en cours de réalisation
Partie difficultés : le C / Struct donnée changée plein de fois/ Web : JS algo coloration	Lola	La partie web est faite uniquement

Que reste il ? :

Concernant les statistiques, Rémi explique qu'il fera 4 catégories qui, selon lui, sont les plus pertinentes :

- le nombre moyen d'essais pour la bonne solution
- le taux d'échec
- histogramme du numéro de l'essai lors de la réussite
- temps pour trouver la solution

La rédaction du rapport est presque achevée, il faut encore compléter le nombre d'heures dans le tableau de travail réalisé, rédiger le compte rendu de la réunion du jour, chacun doit écrire son expérience personnelle, et enfin rédiger les difficultés rencontrées lors de la seconde partie du projet.

Concernant la minimisation des graphes, il faut coder les 4 étapes restantes de l'algorithme de minimisation. Etant donné que la date de rendu se rapproche, le groupe se met d'accord pour dire que la minimisation sera un bonus (dépendant de notre vitesse d'avancement). Ainsi, on se concentre sur la table de hashage et la pondération du graphe. Enfin, concernant la pondération de chaque lettre, elle sera représentée sous forme de poids sur chaque arrête du graphe, et il faudra effectué du backtracking afin de trouver le mot qui enlève le plus de possibilités.

Organisation de la semaine :

Au vu de la semaine, chargée en partiels et occupée par le séminaire, nous allons nous organiser de la manière suivante.

- Lundi et Mardi soirs : Révisions partiels
- Mercredi soir : peut-être une séance de travail pour Mathilde et Lola

- Jeudi soir : Barbecue
- Vendredi soir : Travail pour tout le monde

TODO List :

To Do	Qui ?	Pour Quand ?
Dossier V1 et V2 dans le git	Rémi	4 Juin
Etat de l'art	Mathilde	4 Juin
Fonction prenant un fichier txt et qui remplit le graphe	Mathilde	4 Juin
Pondération sur les lettres + Backtracking	Lucie/Lola	4 Juin
Rédaction rapport : CR du jour, difficultés et V2	Lola	4 Juin
Minimisation si le code est bien avancé	Lucie	4 Juin
Statistiques et interface	Rémi	4 Juin
Si le point précédent fini faire interface de la v2	Rémi	4 Juin

Prochaine Réunion : 4 Juin 2022

7.2 Illustrations de l'application web

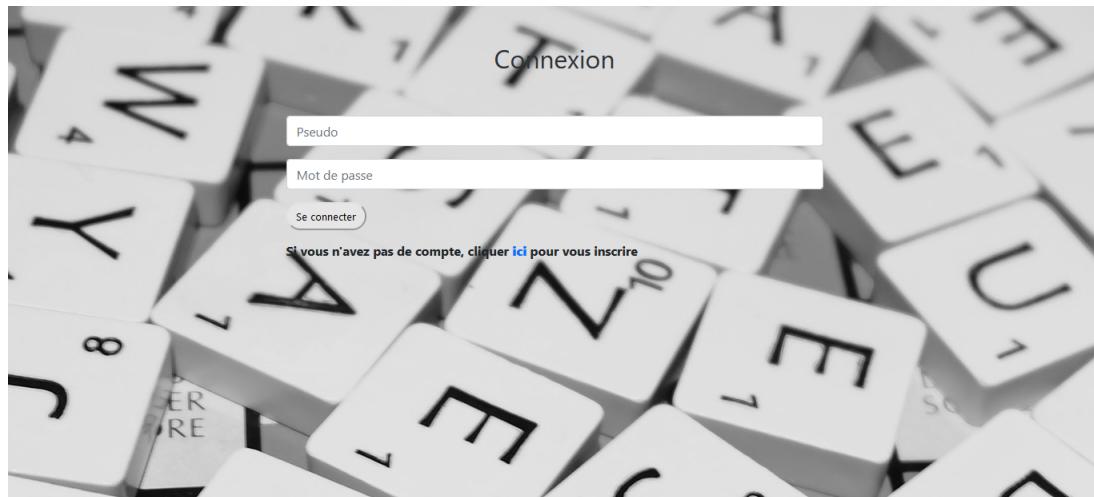
7.2.1 Page d'Inscription



Le formulaire d'inscription est intitulé "Inscription". Il contient les champs suivants : "Pseudo", "Email", "Mot de passe (8 caractères minimum)", et "Confirmation mot de passe". En bas du formulaire, il y a un bouton "S'inscrire" et une ligne de texte indiquant : "Si vous avez déjà un compte, cliquer [ici](#) pour vous connecter."

FIGURE 7.1: Capture d'écran de la page d'Inscription de notre application

7.2.2 Page de Connexion



Le formulaire de connexion est intitulé "Connexion". Il contient les champs suivants : "Pseudo" et "Mot de passe". En bas du formulaire, il y a un bouton "Se connecter" et une ligne de texte indiquant : "Si vous n'avez pas de compte, cliquer [ici](#) pour vous inscrire".

FIGURE 7.2: Capture d'écran de la page de Connexion de notre application

7.2.3 Page de Configuration

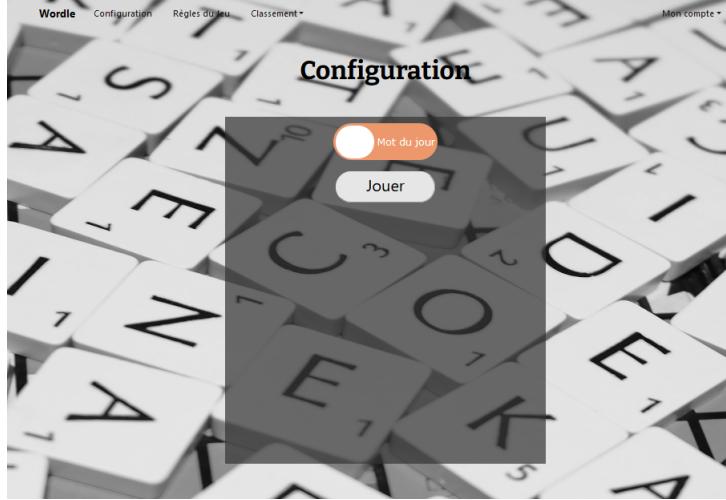


FIGURE 7.3: Capture d'écran de la page Configuration en mode mot du jour de notre application

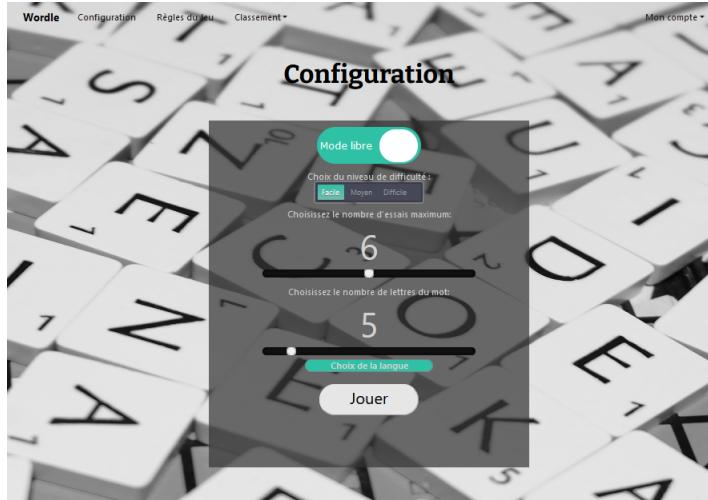


FIGURE 7.4: Capture d'écran de la page Configuration en mode libre de notre application

7.2.4 Page des Règles du Jeu



FIGURE 7.5: Capture d'écran de la page des Règles du Jeu de notre application

7.2.5 Page de Jeu



FIGURE 7.6: Capture d'écran de la page de Jeu, quand une partie est commencée en mode libre et difficulté facile, de notre application

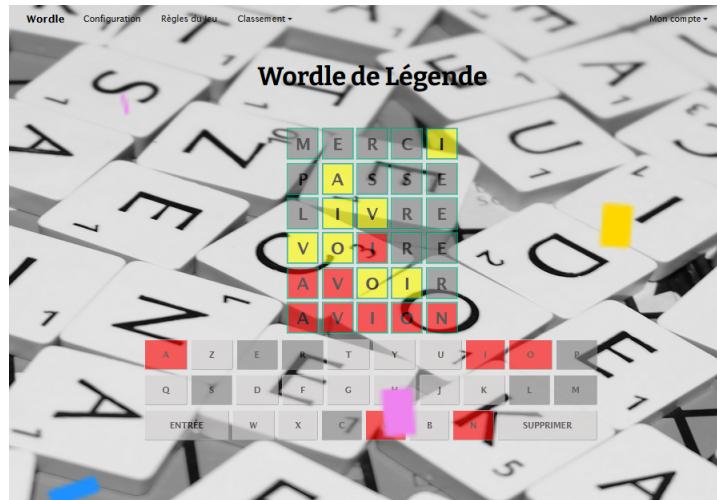


FIGURE 7.7: Capture d'écran de la page de Jeu, quand une partie est gagnée, de notre application

7.2.6 Page Profil

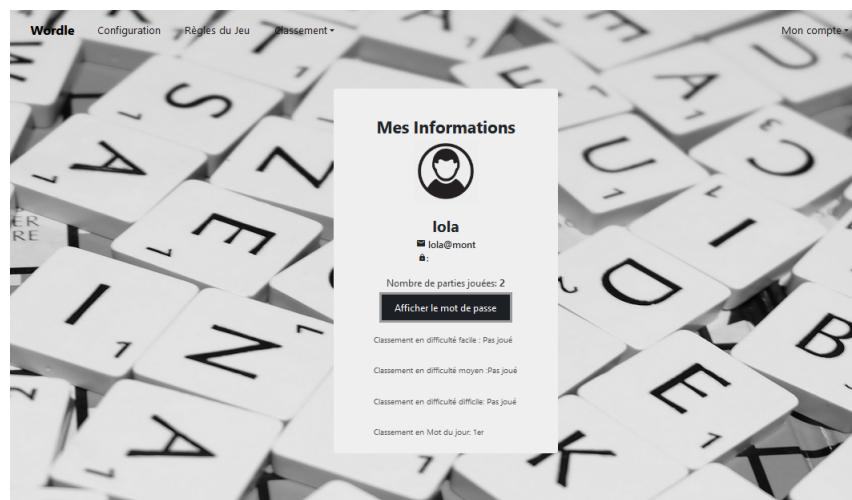


FIGURE 7.8: Capture d'écran de la page Profil de notre application

7.2.7 Page Historique



FIGURE 7.9: Capture d'écran de la page Historique de notre application

7.2.8 Page du Classement

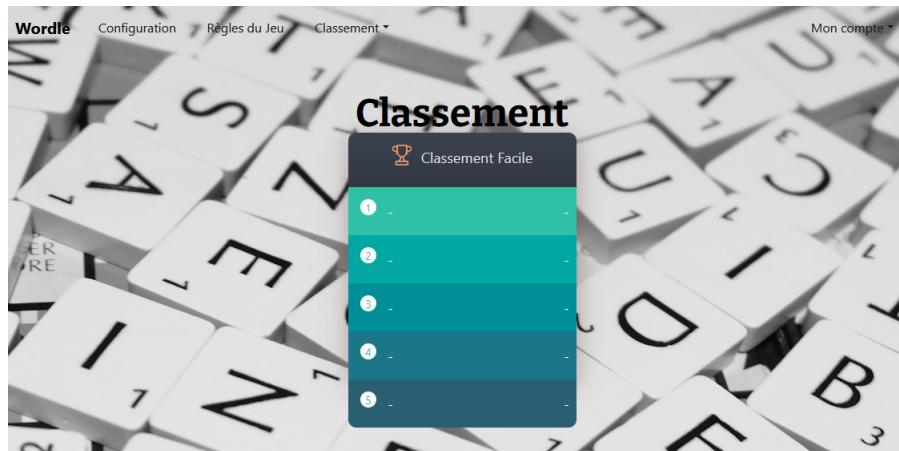


FIGURE 7.10: Capture d'écran de la page du Classement de notre application

Bibliographie

- [1] Jonathan Olson. Optimal wordle solutions. *Jonathanolson.net*, 2022.
- [2] Evan Wallace and Jake Archibald. Directed acyclic word graphs part 1 - the basics. *JB's Blog*, 2019.
- [3] Evan Wallace and Jake Archibald. Directed acyclic word graphs part 2 - first implementation. *JB's Blog*, 2019.