



Projet Pluridisciplinaire d'Informatique Intégrative 2
1A TELECOM Nancy - 2021-2022

Wordle

Projet réalisé sous la direction de M. Olivier FESTOR

Équipe projet :

- Quentin LENFANT
- Damien SIMON
- Paul SANCHEZ
- Elyas ELAZIZ

Table des matières

1	Introduction	4
1.1	Contexte	4
1.2	Présentation du document	4
2	État de l’art	5
2.1	Le jeu Wordle	5
2.2	Les dérivés du Wordle	5
3	L’application WEB	7
3.1	Les pages WEB du site	7
3.2	La base de données	10
3.3	Fonctionnement interne	11
4	Le solveur	12
4.1	Introduction	12
4.2	Les structures de données	13
4.2.1	La table d’occurrence	13
4.2.2	Le dictionnaire	14
4.2.3	L’arbre	17
4.3	Fonctionnement global du solveur	19
4.3.1	Stratégie 1	20
4.3.2	Stratégie 2	20
4.3.3	Stratégie 3	20
5	Tests et performances	21
5.1	Tests	21
5.1.1	Introduction	21
5.1.2	Tests sur l’application WEB	21
5.1.3	Tests sur la table d’occurrence	21
5.1.4	Tests sur le dictionnaire	22
5.1.5	Tests sur l’arbre	23
5.2	Complexité des fonctions de l’application WEB	23
5.2.1	Notations	23
5.2.2	Fonction generateGrid() (JavaScript)	23
5.2.3	Fonction generateKeyboard() (JavaScript)	23
5.2.4	Fonction couleur() (Python)	23
5.2.5	Fonction testTry() (Python)	24
5.3	Performances	26
5.3.1	Performance de score	26
5.3.2	Performances temporelles	29
6	Gestion de projet	30
6.1	Présentation de l’équipe projet	30
6.2	Caractéristiques du projet	30
6.2.1	Définition des objectifs	30
6.2.2	Gestion des risques	31
6.2.3	Organisation	31
6.2.4	Outils de travail	33

6.3	Comptes-rendus de réunion	35
6.3.1	Réunion du 18 mars 2022	35
6.3.2	Réunion du 27 mars 2022	37
6.3.3	Réunion du 30 mars 2022	39
6.3.4	Réunion du 03 Avril 2022	41
6.3.5	Réunion du 13 Avril 2022	43
6.3.6	Réunion du 20 Avril 2022	45
6.3.7	Réunion du 27 Avril 2022	47
6.3.8	Réunion du 30 Avril 2022	49
6.3.9	Réunion du 08 Mai 2022	51
6.3.10	Réunion du 13 Mai 2022	53
6.3.11	Réunion du 22 Mai 2022	55
6.3.12	Réunion du 26 Mai 2022	57
6.3.13	Réunion du 29 Mai 2022	59
6.3.14	Réunion du 06 Juin 2022	62
7	Bilan du projet	66
7.1	Bilan global	66
7.1.1	Comparaison travail attendu/effectué	66
7.1.2	Points positifs, Points négatifs	67
7.1.3	Volume horaire	67
7.2	Bilans individuels	68
7.2.1	Quentin LENFANT	68
7.2.2	Damien SIMON	68
7.2.3	Paul SANCHEZ	69
7.2.4	Elyas ELAZIZ	69

Chapitre 1

Introduction

1.1 Contexte

Ce projet a été réalisé dans le cadre du module PPII 2 (Projet Pluridisciplinaire d'Informatique Intégrative 2) de première année du cycle ingénieur sous statut étudiant de TELECOM Nancy. Il constitue une application des connaissances vues dans les modules CS54 (Computer Science 54), C/SD (Langage C et Structures de Données), ainsi que le module de Gestion de Projet.

L'objectif était de concevoir en équipe deux applications basées sur le jeu Wordle : d'une part une application WEB où l'utilisateur joue au jeu, et d'autre part un solveur où l'ordinateur joue et l'utilisateur le corrige.

1.2 Présentation du document

Ce premier chapitre constitue une introduction au projet.

Dans le chapitre 2, nous présenterons le jeu Wordle ainsi que les recherches que nous avons effectuées pour la partie État de l'art.

Le chapitre 3 sera consacré à la conception et à l'implémentation de l'application WEB.

Le chapitre 4 traitera lui de la conception et de l'implémentation du solveur.

Nous détaillerons ensuite les tests et la performance de nos programmes dans le chapitre 5.

Toute la partie gestion de projet sera abordée dans le chapitre 6.

Et enfin, le chapitre 7 conclura ce rapport par un bilan du projet d'un point de vue global et du point de vue de chaque membre de l'équipe projet.

Chapitre 2

État de l’art

2.1 Le jeu Wordle

Wordle [1] [2] est un jeu créé par Josh Wardle et qui constitue une adaptation du jeu télévisé américain Lingo. Le but du jeu est de deviner chaque jour un mot anglais de 5 lettres en moins de 6 d’essais. Une partie se termine soit lorsque le mot est trouvé, soit lorsque le nombre maximal d’essais est dépassé. Lorsque le joueur propose un mot, l’ordinateur lui indique, en coloriant les cases de la grille, les lettres qui ne sont pas dans le mot, et celles qui sont bien placées. Un exemple est donné en Figure 2.1.

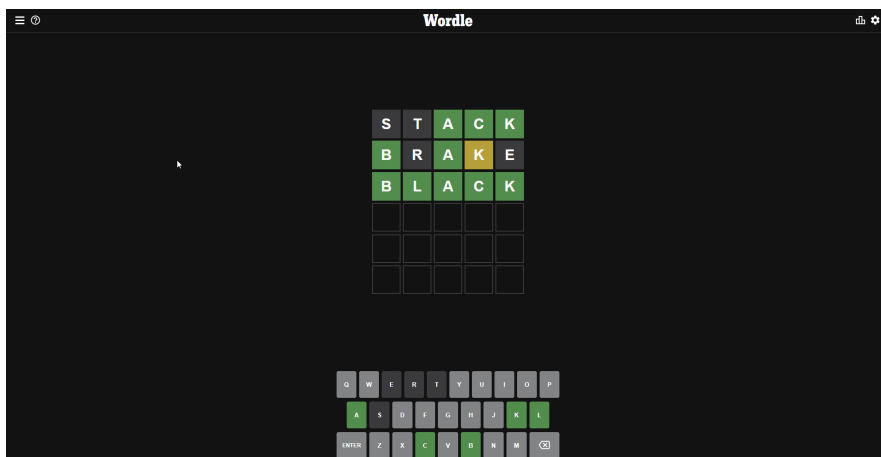


FIGURE 2.1 – Un exemple de partie sur le site du Wordle. En vert les lettres bien positionnées, en jaune, celles qui ne le sont pas, et sans couleur celles qui ne sont pas dans le mot.

2.2 Les dérivés du Wordle

Suite au succès du jeu, de nombreuses variantes du jeu ont été publiées. Les différences portent sur le format d’une part et sur le principe d’autre part. En voici quelques exemples :

- **Sutom** [3] : Il s’agit de la version française du jeu. La charte graphique reprend celle du jeu français Motus.
- **Zutom** [4] : Ce jeu fonctionne par série de 5 mots de longueur variable à trouver en moins de 10 minutes.
- **Quordle** [5] : Dans cette variante disponible dans plusieurs langues, il faut deviner 4 mots en même temps.
- **Weredle** [6] : Dans ce jeu, l’ordinateur peut mentir sur la couleur qu’il affecte à chaque lettre des mots proposés. Le joueur est alors invité à trouver ces lettres trompeuses tout en cherchant le mot voulu.

- **Worldle** [7] : Il s'agit d'une version où le joueur cherche un pays ou un territoire géographique dans le monde.
- **Nerdle** [8] : Il faut deviner dans ce jeu un calcul mathématique contenant des chiffres et des opérations élémentaires.

Chapitre 3

L’application WEB

3.1 Les pages WEB du site

Lorsque l’utilisateur arrive sur le site avec la route / :

- soit il est redirigé sur la route `/newGame` s’il n’a pas de partie en cours,
- soit il recharge cette partie directement sur la route `/currentGame`.

Dans le premier cas, qui est présenté en Figure 3.1, l’utilisateur peut paramétrer la prochaine partie. Il peut choisir la longueur du mot à trouver (entre 5 et 15 lettres), le nombre maximal d’essais possibles (entre 3 et 10) et la difficulté (10 niveaux de difficulté possibles).

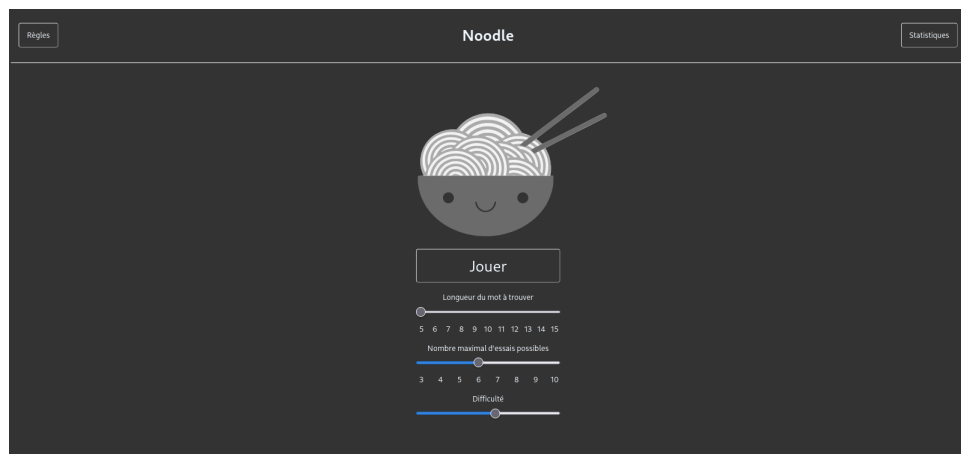


FIGURE 3.1 – Page d’accueil

Le bouton "Règles" est présent sur toutes les pages du site. Il affiche un pop-up expliquant les règles du jeu comme indiqué en Figure 3.2.

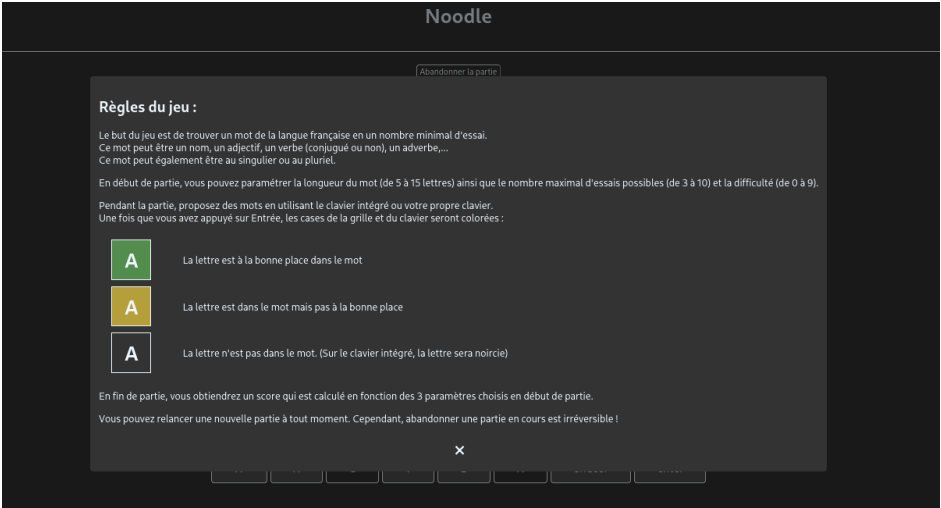


FIGURE 3.2 – Pop-up "Règles"

Une fois les paramètres validés par appui sur le bouton "Jouer" (ou si le joueur arrive sur le site et recharge une partie), le joueur est dirigé sur la route `/currentGame`. Cette route mène à la page principale du jeu sur laquelle se trouve la grille de jeu ainsi qu’un clavier intégré. Cette page est présentée en Figure 3.3. À noter qu’une partie peut être abandonné définitivement à tout moment avec le bouton correspondant situé au-dessus de la grille.

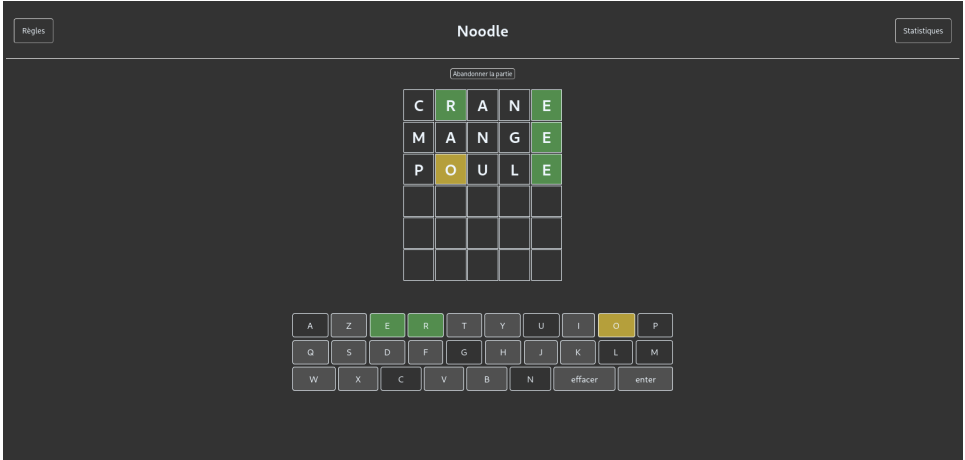


FIGURE 3.3 – Page principale du jeu

En fin de partie, un pop-up affiche les résultats. Il existe 2 variantes du pop-up, une variante en cas de victoire (Figure 3.4), et une en cas de défaite (Figure 3.5). Le score affiché est calculé en fonction des paramètres de la partie selon la formule suivante :

$$Score = 10000 \times Difficulte + \frac{Longueur\ du\ mot}{Nombre\ maximal\ d'essais\ possibles} \times 1000$$

Le résultat de cette formule est un entier compris entre 0 et 105 000.

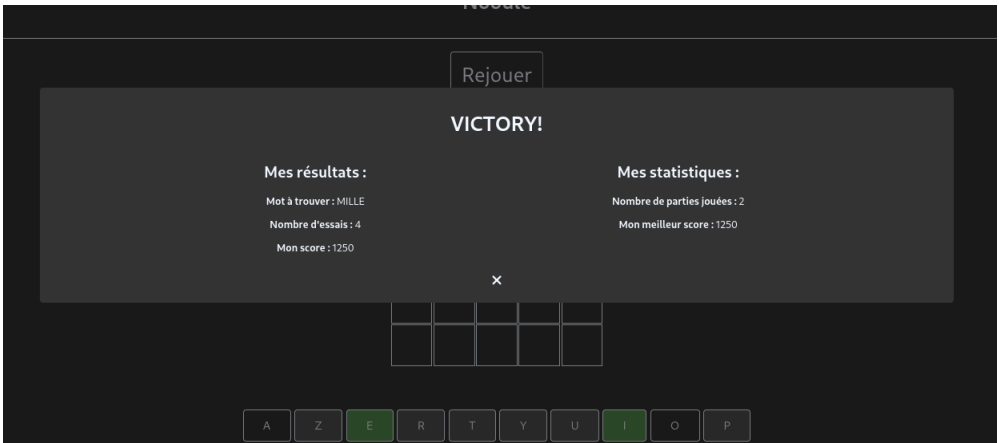


FIGURE 3.4 – Page de fin de partie (victoire)



FIGURE 3.5 – Page de fin de partie (défaite)

Ensuite, lorsque le joueur ferme le pop-up de fin de partie, il peut revoir la grille de sa partie. Il peut également redémarrer une partie en cliquant sur le bouton "Rejouer" (Figure 3.6).

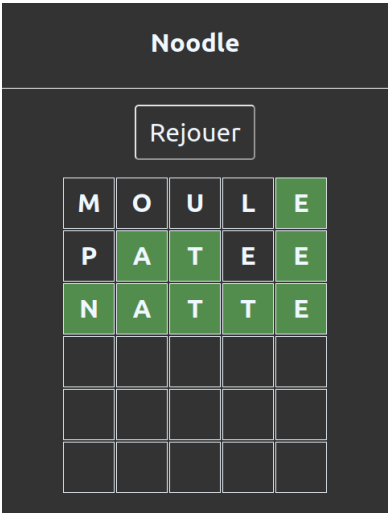


FIGURE 3.6 – Bouton "Rejouer"

Enfin, le bouton "Statistiques" affiche un pop-up avec des statistiques du joueur (Figure 3.7). Sont affichés le nombre de parties jouées du joueur ainsi que son meilleur score.

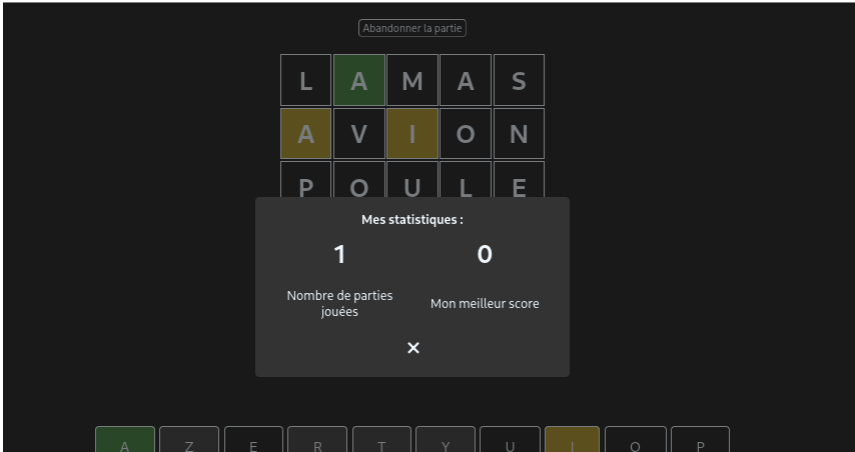


FIGURE 3.7 – Pop-up "Statistiques"

3.2 La base de données

La base de données a été conçue avec l’outil "dbdiagram.io" [9]. Celle-ci est présentée en figure 3.8.

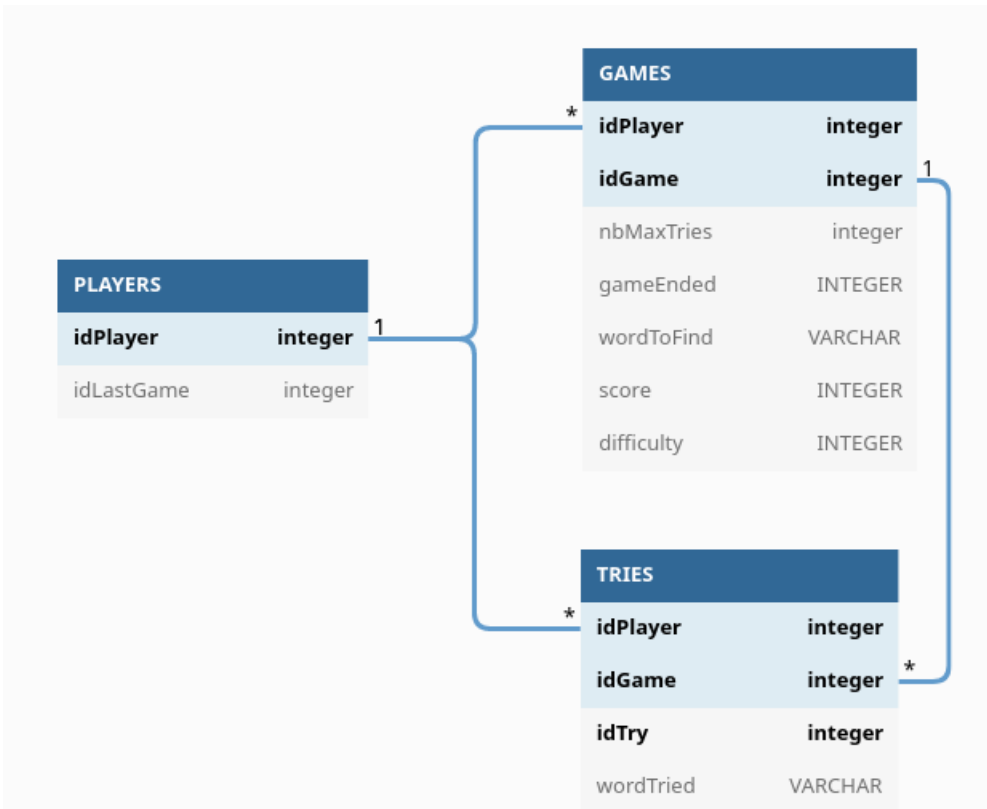


FIGURE 3.8 – Base de données

Cette base de données est en 3^{ème} forme normale. Les attributs des tables sont décrits ci-après :

- `idPlayer` correspond à l'identifiant du joueur. Celui-ci correspond au cookie stocké sur le navigateur de l'utilisateur.
- `idGame` correspond à l'identifiant d'une partie.
- `idLastGame` correspond à l'identifiant de la dernière partie du joueur.
- `nbMaxTries` correspond au nombre maximal d'essais possibles d'une partie.
- `gameEnded` permet de vérifier si une partie est terminée ou non.
- `wordToFind` correspond au mot à trouver de la partie.
- `idTry` correspond à l'identifiant d'un essai.
- `wordTried` correspond au mot proposé par un joueur à un essai donné.
- `score` correspond au score remporté en cas de victoire.
- `difficulty` correspond à la difficulté sélectionnée lors de la création de la partie.

3.3 Fonctionnement interne

Authentification

Lors de l'arrivée sur le site (route `/`), on vérifie si l'utilisateur est authentifié, c'est-à-dire, s'il possède un cookie qui stocke son identifiant (via la variable `session['id']` de Flask) et que ce cookie est cohérent avec la base de données (dans le but d'éviter des erreurs en production).

S'il n'est pas authentifié, on lui assigne un cookie avec un nouvel identifiant et on l'ajoute à la base de données.

Page du jeu (route `/currentGame`)

L'API commence par récupérer toutes les données de la partie en cours dans la base de données. Elle génère ensuite la grille et le clavier virtuel grâce à un script JavaScript - celui-ci crée des tableaux HTML. La coloration de la grille et du clavier en fonction des mots proposés est calculée en Python et est transmise au site via l'API. L'affichage se fait en JavaScript.

Le joueur peut ensuite proposer un mot. Lorsqu'il appuie sur Entrée, une fonction JavaScript va vérifier si le mot est valide. Si ce n'est pas le cas, les cases de la grille vibrent pour lui indiquer l'erreur. Sinon, l'essai est envoyé à l'API pour qu'il soit stocké dans la base de données et que la coloration du mot soit calculée.

L'API vérifie également si le mot entré est le mot recherché ou si le joueur a dépassé le nombre maximal d'essai. Si tel est le cas, elle terminera la partie, mettra à jour la base de données et rechargera la page en affichant le pop-up des résultats de la partie. Sinon, on recommence le processus en rechargeant simplement la page.

En résumé, on exécute les fonctions suivantes dans l'ordre :

1. L'API envoie les données au site.
2. Exécution `generateGrid()` et `generateKeyboard()` (JavaScript).
3. Quand le joueur propose un mot, exécution de `testTry()` (JavaScript).
4. Si le mot est valide, calcul de la coloration avec `couleur()` (Python)
Sinon, retour à l'étape 3.
5. Test de fin de partie avec `testEndGame()` (Python).
6. Si c'est la fin de la partie, rechargement de la page avec affichage de le pop-up de fin.
Sinon retour à l'étape 1.

Chapitre 4

Le solveur

4.1 Introduction

L'objectif de cette partie du projet était de coder en C un solveur Wordle performant en utilisant des structures de données adaptées au problème. La question de la performance de notre solveur prend ici deux aspects :

- En combien de coups le solveur gagne une partie ?
- En combien de temps le solveur calcule un coup ?

On parlera pour la première d'une performance de score, et pour la deuxième d'une performance de temps.

Le problème de la performance de score revient à chercher pour chaque coup le meilleur mot à proposer en fonction des informations déjà acquises au cours de la partie. Pour trouver un tel mot, nous avons choisi d'appliquer un algorithme MinMax en utilisant la formule suivante :

$$\text{Best_word}_i = \min_{\text{mot} \in D_i} \{ \max_{p \in P} \{ \text{nb_matches}(\text{mot}, p, D_i) \} \}$$

Avec :

$$\forall i \in \mathbb{N}, \begin{cases} D_0 = D \\ D_i \subset D_{i+1} \\ |D_{i+1}| = \text{nb_matches}(\text{Best_word}_i, p_i, D_i) \end{cases}$$

où :

- Best_word_i correspond au meilleur mot pour le $i^{\text{ème}}$ coup
- D correspond à l'ensemble des mots possibles de la partie
- P correspond à l'ensemble des patterns possibles
- p_i correspond au pattern renvoyé par l'utilisateur au $i^{\text{ème}}$ coup
- D_i correspond à l'ensemble des mots qui représentent une solution potentielle au $i^{\text{ème}}$ coup
- $\text{nb_matches}(\text{mot}, p, D_i)$ correspond au nombre de mots de D_i qui matchent avec le pattern p associé au mot

L'algorithme MinMax permet d'obtenir le mot qui réduit au maximum le nombre de solutions potentielles pour le coup suivant. Avec cette formule, on prend en compte uniquement le pire des cas, c'est-à-dire le cas où le pattern renvoyé par l'utilisateur donne la valeur maximale de $\text{Nb_matches}()$ possibles pour ce mot.

Toutefois, si cette formule nous permet d'assurer une bonne performance de score pour notre solveur, elle va cependant poser un vrai défi pour la performance de temps. En effet, il s'agit ici d'effectuer pour chaque mot du dictionnaire ($\approx 16\,000$ pour 10 lettres), le calcul de la fonction $\text{nb_matches}()$ pour l'ensemble des patterns possibles $= 3^{\text{nb_letters}}$ ($\approx 59\,000$ pour 10 lettres). La fonction $\text{nb_matches}()$ devant elle même parcourir l'ensemble des mots de D_i ($|D_0| = |D| \approx 16\,000$ pour 10 lettres) pour y trouver les mots qui matchent.

Il s'agit donc de choisir judicieusement les structures de données qui vont nous permettre d'effectuer ce calcul tout en gardant une performance de temps acceptable.

4.2 Les structures de données

4.2.1 La table d'occurrence

Spécifications algébriques

Type Occ_table

Opérations internes :

Init_table	:		→	Occ_table
Change_min	:	Occ_table × Lettre × Entier	→	Occ_table
Change_max	:	Occ_table × Lettre × Entier	→	Occ_table
Fusion_table	:	Occ_table × Occ_table	→	Occ_table

Observateurs :

Get_min	:	Occ_table × Lettre	→	Entier
Get_max	:	Occ_table × Lettre	→	Entier
Trouver	:	Occ_table × Lettre	→	Bool
Sum_min	:	Occ_table	→	Entier

Préconditions :

Get_min(T,a) est défini	↔	$a \in \llbracket 'A', 'Z' \rrbracket$
Get_max(T,a) est défini	↔	$a \in \llbracket 'A', 'Z' \rrbracket$
Trouver(T,a) est défini	↔	$a \in \llbracket 'A', 'Z' \rrbracket$
Change_min(T,a,i) est défini	↔	$a \in \llbracket 'A', 'Z' \rrbracket$ et $i \geq (\text{Get_min}(T,a))$
Change_max(T,a,i) est défini	↔	$a \in \llbracket 'A', 'Z' \rrbracket$ et $i \leq (\text{Get_max}(T,a))$

Axiomes :

Notations compactes :

- $n = \text{nb_letters}$
- $F(T_1, T_2) \leftrightarrow \text{Fusion_table}(T_1, T_2)$
- $\text{GetM}(T, a) \leftrightarrow \text{Get_Max}(T, a)$
- $\text{Getm}(T, a) \leftrightarrow \text{Get_Min}(T, a)$

Sum_min() :

Sum_min(Init_table(),a)	=	0
Sum_min(Change_min(T,a,i)	=	$i - \text{Getm}(T,a) + \text{Sum_min}(T)$
Sum_min(Change_max(T,a,i)	=	$\text{Sum_min}(T)$
Sum_min(F(T ₁ ,T ₂))	=	$\sum_{a='A'}^{'Z'} \text{Getm}(F(T_1, T_2), a)$

Getm() :

Getm(Init_table(),a) = 0

Getm(Change_min(T,a₁,i),a₂) = $\begin{cases} \text{si } a_1 = a_2 \rightarrow i \\ \text{si } a_1 \neq a_2 \rightarrow \text{Getm}(T,a_2) \end{cases}$

Getm(Change_max(T,a₁,i),a₂) = Getm(T,a₂)

Getm(F(T₁,T₂),a) = max(Getm(T₁,a),Getm(T₂,a))

GetM() :

GetM(Init_table(),a) = n

GetM(Change_min(T,a₁,i),a₂) = $\begin{cases} \text{si } a_1 = a_2 \rightarrow \text{Getm}(T,a_2) \\ \text{si } a_1 \neq a_2 \rightarrow \text{Getm}(T,a_2) - (\text{Getm}(T,a_1) - i) \end{cases}$

GetM(Change_max(T,a₁,i),a₂) = $\begin{cases} \text{si } a_1 = a_2 \rightarrow i \\ \text{si } a_1 \neq a_2 \rightarrow \text{Getm}(T,a_2) \end{cases}$

GetM(F(T₁,T₂),a) = $\min \begin{pmatrix} \text{Getm}(T_1,a) \\ \text{Getm}(T_2,a) \\ n - \text{Getm}(F(T_1,T_2),a + \text{Sum_min}(F(T_1,T_2))) \end{pmatrix}$

Trouver() :

Trouver(Init_table(),a) = Faux

Trouver(Change_min(T,a₁,i),a₂) = $\begin{cases} \text{si } a_1 = a_2 \rightarrow i == \text{GetM}(T,a_2) \\ \text{si } a_1 \neq a_2 \rightarrow \text{Getm}(T,a_2) == \text{GetM}(T,a_2) \end{cases}$

Trouver(Change_max(T,a₁,i),a₂) = $\begin{cases} \text{si } a_1 = a_2 \rightarrow \text{Getm}(T,a_2) == i \\ \text{si } a_1 \neq a_2 \rightarrow \text{Getm}(T,a_2) == \text{GetM}(T,a_2) \end{cases}$

Trouver(F(T₁,T₂),a) = Getm(F(T₁,T₂),a) == GetM(F(T₁,T₂),a)

4.2.2 Le dictionnaire

Objectifs

La structure de dictionnaire a pour but de conserver d'une part, l'ensemble des mots d'une taille donnée, contenus dans un fichier au format `.txt` (établi au préalable).

D'autre part, lors du fonctionnement de l'algorithme, son objectif est de rendre accessible la consultation de certains mots uniquement, les autres étant supprimés par le biais de mises à jour, afin de pouvoir calculer efficacement le "score" d'un mot donné, sans avoir à prendre en compte l'ensemble des autres mots existants.

Choix de la structure

Afin de répondre aux objectifs énoncés précédemment, nous avons opté pour l'utilisation d'une **liste doublement chaînée contiguë** pour la structure du dictionnaire.

L'exploitation d'une liste contiguë garantit notamment une complexité d'accès en $O(1)$ aux éléments du contenu du dictionnaire.

Quant à l'utilisation d'une liste doublement chaînée, celle-ci se justifie par son efficacité pour supprimer un élément du contenu du dictionnaire, opération nécessaire lors des mises à jour, pour rendre certains mots inaccessibles.

Spécifications algébriques

La structure du dictionnaire repose sur la structure de cellule, présentée en première dans la spécification algébrique du dictionnaire ci-dessous :

Type C

Opérations internes :

Init_cell : Entier × Mot → C

Observateurs :

getMot : C → Mot

getInt : C → Entier

Préconditions du Type C :

Pas de pré-conditions particulières

Axiomes du Type C :

getMot() :

getMot(Init_cell(i,s)) = s

getInt() :

getInt(Init_cell(i,s)) = i

Type D<C,I>

Opérations internes :

Init_dico : D<C,I> → D<C,I>

Suppr_dico : D<C,I> × Entier → D<C,I>

Observateurs :

Size : D<C,I> → Entier

getNbWord : D<C,I> → Entier

getWord : D<C,I> × Entier → Mot

getCell : D<C,I> × Entier → C

getIsSuppr : D<C,I> × Entier → Bool

Préconditions du Type $D \langle C, I \rangle$:

$\text{Suppr_dico}(D, i)$ est défini $\Leftrightarrow 0 \leq i \leq \text{Size}(D) - 1$ et $\neg(\text{getIsSuppr}(D, i))$
 $\text{getWord}(D, i)$ est défini $\Leftrightarrow 0 \leq i \leq \text{Size}(D) - 1$ et $\neg(\text{getIsSuppr}(D, i))$
 $\text{getCell}(D, i)$ est défini $\Leftrightarrow 0 \leq i \leq \text{Size}(D) - 1$ et $\neg(\text{getIsSuppr}(D, i))$
 $\text{getIsSuppr}(D, i)$ est défini $\Leftrightarrow 0 \leq i \leq \text{Size}(D) - 1$

Axiomes du Type $D \langle C, I \rangle$:

Notations compactes :

- $n = \text{nb_letters}$
- m est la taille du dictionnaire chargé pour les mots de n lettres.

$\text{Size}()$:

$\text{Size}(\text{Init_dico}()) = m$
 $\text{Size}(\text{Suppr_dico}(D, i)) = m$

$\text{getNbWord}()$:

$\text{getNbWord}(\text{Init_dico}()) = m$
 $\text{getNbWord}(\text{Suppr_dico}(D, i)) = \text{getNbWord}(D) - 1$

$\text{getWord}()$:

$\text{getWord}(\text{Init_dico}(), i) = \text{getMot}(\text{getCell}(D, i))$
 $\text{getWord}(\text{Suppr_dico}(D, i), j) = \begin{cases} \text{si } i = j & \rightarrow \text{Non défini} \\ \text{sinon} & \rightarrow \text{getWord}(D, j) \end{cases}$

$\text{getCell}()$:

$\text{getCell}(\text{Init_dico}(), i) = \text{getCell}(D, i)$
 $\text{getCell}(\text{Suppr_dico}(D, i), j) = \begin{cases} \text{si } i = j & \rightarrow \text{Non défini} \\ \text{sinon} & \rightarrow \text{getCell}(D, j) \end{cases}$

$\text{getIsSuppr}()$:

$\text{getIsSuppr}(\text{Init_dico}(), i) = \text{Faux}$
 $\text{getIsSuppr}(\text{Suppr_dico}(D, i), j) = \begin{cases} \text{si } i = j & \rightarrow \text{Vrai} \\ \text{sinon} & \rightarrow \text{Faux} \end{cases}$

4.2.3 L'arbre

Objectifs

L'objectif de la structure d'arbre est de modéliser l'ensemble D_i des solutions potentielles. Cette structure devra évoluer tout au long de la partie pour converger vers le mot voulu d'une part, et faciliter le calcul de `nb_matches()` pour chaque mot d'autre part.

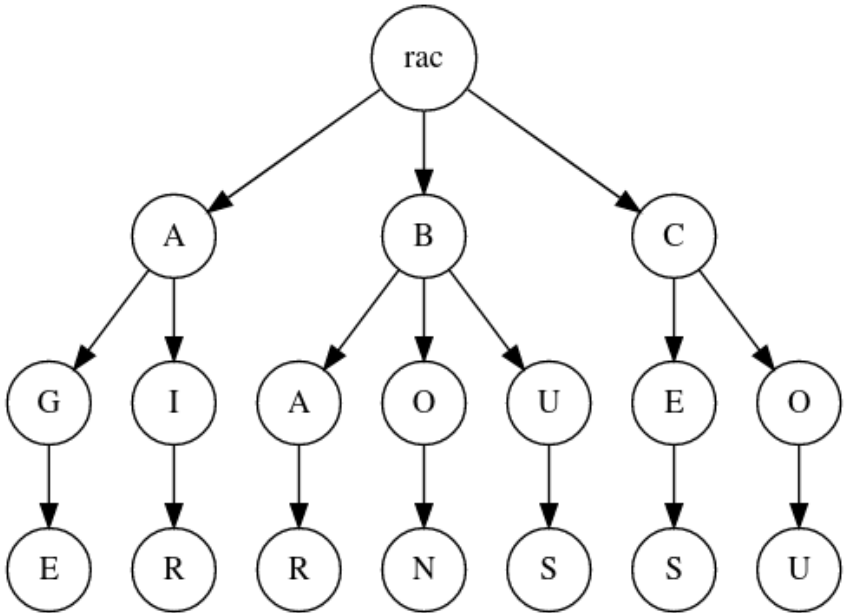


FIGURE 4.1 – Représentation d'un Arbre contenant les mots : AGE, AIR, BAR, BUS, BON, CES et COU

Choix de la structure

Afin de répondre à cet objectif, nous avons choisi d'implémenter un arbre 26-aire, de profondeur égale au nombre de lettres de la partie. Ainsi, chaque branche représente un suffixe des mots enregistrés dans l'arbre. On remarque que la complexité de recherche d'un mot dans l'arbre est en $O(\text{nb_letters})$.

Pour comprendre l'intérêt de la structure d'arbre, on peut comparer son fonctionnement à celui d'une structure plus classique telle que celle d'une liste doublement chaînée (comme celle que l'on utilise pour modéliser D dans la suite de l'algorithme).

Soit $D = \{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z\}^5$ un dictionnaire contenant toutes les combinaisons de 5 lettres possibles.

On pose $n = |D| = 26^5 = 11881376$.

On peut déjà remarquer qu'en terme de complexité sur la recherche d'un mot dans la structure, on a :

- Au mieux $O(\text{Log}(n))$ pour une liste chaînée triée,
- et $O(\text{nb_letters})$, soit du temps constant, pour un arbre.

Toutefois, là où la différence est importante, c'est sur le calcul de la fonction `nb_matches(mot, p_i , D_i)` :

- Pour une liste doublement chaînée contiguë, il faut itérer une vérification sur chaque mot de D_i ,
- Pour un arbre, on peut utiliser le pattern p_i renvoyé par l'utilisateur pour sélectionner les branches à parcourir. Ainsi, on restreint le parcours de l'arbre aux mots qui respectent le positionnement des lettres dans le pattern.

Dès lors, dans le pire des cas, $p_i = 11111$: on peut éviter de parcourir les suffixes commençants par les lettres marqués avec un 1. Le parcours a donc un coût de $(26 - 1)^5 = 9765625 < n$. Tandis que, dans le meilleur des cas, $p_i = 22222$: le nombre de mots étant réduit à 1, le temps de calcul de `nb_matches()` revient exactement à celui de la recherche de l'existence d'un mot dans l'arbre. soit un coût de `nb_letters`.

Spécifications algébriques

Type Abr :

Opérations internes :

<code>Creat_abr</code>	: Entier	→ Abr
<code>Enleve_branche</code>	: Abr × Lettre	→ Abr
<code>Ajout_branche</code>	: Abr × Lettre	→ Abr

Observateurs :

<code>Empty</code>	: Abr	→ Mot
<code>Get_branche</code>	: Abr × Lettre	→ Entier
<code>Nb_mots</code>	: Abr	→ Entier
<code>Profondeur</code>	: Abr	→ Entier

Préconditions :

<code>Creat_abr(i)</code> est défini	⇔	$0 \leq i \leq \text{nb_letters}$
<code>Get_branche(A,a)</code> est défini	⇔	$a \in \llbracket 'A', 'Z' \rrbracket$ et $\neg(\text{Empty}(A))$
<code>Enleve_branche(A,a)</code> est défini	⇔	$a \in \llbracket 'A', 'Z' \rrbracket$ et $\neg(\text{Empty}(\text{Get_branche}(A,a)))$
<code>Ajout_branche(A,a)</code> est défini	⇔	$\text{Profondeur}(A) < \text{nb_letters}$ et $\neg(\text{Empty}(A))$

Axiomes :

Notations compactes :

- $n = \text{nb_letters}$
- $\text{Rm_b}(A,a) \leftrightarrow \text{Enleve_branche}(A,a)$
- $\text{Add_b}(A,a) \leftrightarrow \text{Ajout_branche}(A,a)$
- $\text{Get_b}(A,a) \leftrightarrow \text{Get_branche}(A,a)$
- $P(A) \leftrightarrow \text{Profondeur}(A)$

`Empty()` :

<code>Empty(Creat_abr(i))</code>	= Vrai
<code>Empty(Get_b(Creat_abr(i)),a)</code>	= Vrai $\forall a \in \llbracket 'A', 'Z' \rrbracket$
<code>Empty(Rm_b(A,a))</code>	= Faux
<code>Empty(Add_b(A,a))</code>	= Faux

`Profondeur()` :

<code>Profondeur(Creat_abr(i))</code>	= i
<code>Profondeur(Rm_b(A,a))</code>	= P(A)
<code>Profondeur(Add_b(A,a))</code>	= P(A)

Nb_mots() :

$$\begin{aligned} \text{Nb_mots}(\text{Creat_abr}(i)) &= 0 \\ \text{Nb_mots}(\text{Rm_b}(A,a)) &= \text{Nb_mots}(A) - \text{Nb_mots}(\text{Get_b}(A,a)) \\ \text{Nb_mots}(\text{Add_b}(A,a)) &= \begin{cases} \text{si } P(A) = n - 1 & \rightarrow \text{Nb_mots}(A) + 1 \\ \text{sinon} & \rightarrow \text{Nb_mots}(A) \end{cases} \end{aligned}$$

Get_b() :

$$\begin{aligned} \text{Get_b}(\text{Creat_abr}(i),a) &\rightarrow \text{Non d\'efini} \\ \text{Get_b}(\text{Rm_b}(A,a_1),a_2) &= \begin{cases} \text{si } a_1 \neq a_2 & \rightarrow \text{Get_b}(A,a_2) \\ \text{si } a_1 = a_2 & \rightarrow \text{Non d\'efini} \end{cases} \\ \text{Get_b}(\text{Add_b}(A,a_1),a_2) &= \begin{cases} \text{si } a_1 \neq a_2 & \rightarrow \text{Get_b}(A,a_1) \\ \text{si } a_1 = a_2 & \rightarrow \text{Creat_abr}(P(A) + 1) \end{cases} \end{aligned}$$

4.3 Fonctionnement global du solveur

La fonction `main()` r cup re tout d’abord le nombre de lettres de la partie en lisant le fichier `wsolf.txt`, ainsi que le premier mot que propose le solveur. Ce mot aura  t  pr -calcul  pour les mots entre 1 et 10 lettres dans le fichier `openers.txt`. Au-del  le 10 lettres, on se contente de choisir le premier mot du dictionnaire pour des raisons de complexit .

Ce mot s’affiche en sortie standard. L’utilisateur entre alors la cha ne constitu e de 0, de 1 et/ou de 2.

On initialise ensuite les structures de donn es et on entre dans la boucle principale : tant que la solution n’est pas trouv e ou que l’utilisateur ne d cide pas d’en sortir en tapant "-1", on lance le calcul du prochain mot   proposer.

- On va chercher dans un premier temps   d terminer les possibilit s restantes. Pour cela, on va :
1. Mettre   jour la table d’occurrence :   partir de la nouvelle information, on peut faire  vo-luer le nombre maximum et minimum d’occurrences de chaque lettre dans le mot   trouver.
 2. Mettre   jour l’arbre : on supprime toutes les branches qui m nent vers un mot non valide gr ce   la nouvelle table d’occurrence.
 3. Mettre   jour le dictionnaire : on supprime du dictionnaire les mots que l’on estime inutiles pour le calcul du meilleur mot   proposer au coup suivant.

Enfin, on calcule le meilleur mot   proposer parmi les mots valides et lisibles dans le diction-naire. On le soumet   l’utilisateur qui renseigne la cha ne de 0,1 et 2 correspondante puis on re-commence la boucle.

Pour le calcul de la meilleure proposition, afin d’optimiser le rapport complexit /performance, on utilise une strat gie diff rente selon la taille du mot.

- Jusqu’  6 lettres, on utilise la strat gie pr sent e dans la sous-section 4.3.1,
- Pour un mot de 7   10 lettres, on utilise la strat gie pr sent e dans la sous-section 4.3.2,
- Au-del  de 10 lettres, on utilise la strat gie pr sent e dans la sous-section 4.3.3,

4.3.1 Stratégie 1

La première stratégie est celle qui donne la meilleure performance de score mais est également celle qui demande le plus de temps de calcul. Cette stratégie consiste en une implémentation de l'algorithme MinMax en limitant le parcours de la boucle principale aux mots qui respectent la borne max de la table d'occurrence.

Ainsi si pour une partie à 6 lettres, on a $\text{Get_max}(T, 'A') = 2$, le mot ANANAS ne figurera pas dans le dictionnaire car celui-ci comporte 3 'A'.

En filtrant ainsi le dictionnaire, on réduit déjà le nombre de calculs à effectuer lors de l'exécution de l'algorithme MinMax.

4.3.2 Stratégie 2

Le temps de calcul du solveur pour une partie de 10 lettres dépassant 8 minutes avec la stratégie 1, nous avons décidé de réfléchir à une deuxième stratégie, améliorant la performance de temps, au dépend de la performance de score.

Cette deuxième stratégie diffère de la première dans sa façon de filtrer le dictionnaire. En effet, dans cette stratégie, les mots stockés dans l'arbre et ceux stockés dans le dictionnaire sont les mêmes à toutes les étapes de la partie. Ainsi la recherche du meilleur mot par l'algorithme MinMax se limite aux seuls mots valides.

En d'autres termes, la formule peut se réécrire comme suit :

$$\text{Best_word}_i = \min_{\text{mot} \in D_i} \left\{ \max_{p \in P} \{ \text{nb_matches}(\text{mot}, p, D_i) \} \right\}$$

4.3.3 Stratégie 3

Le temps de calcul pour l'opener des parties à 11 lettres dépassant les 80h, nous avons décider d'arrêter d'utiliser l'algorithme MinMax pour les mots au-delà 10 lettres. A la place, nous choisissons systématiquement le premier mot du dictionnaire, celui-ci étant trié par ordre de fréquence, la stratégie 3 correspond donc à proposer les mots valides du plus fréquent au moins fréquent.

Chapitre 5

Tests et performances

5.1 Tests

5.1.1 Introduction

Afin d'assurer la correction, la fiabilité et la robustesse de nos applications ainsi que l'absence de failles majeures au sein de celui-ci, une batterie de tests a été réalisé.

Chacun des tests a alors été établi en accord avec la méthode du Right-BICEP qui consiste à :

- Vérifier que les résultats obtenus sont corrects.
- Vérifier que les conditions aux limites sont correctes.
- Vérifier les relations inverses de chaque résultat.
- Forcer les conditions qui résulteraient en erreurs.
- Vérifier la performance des différents programmes.

Ces tests nous ont alors conduits à mener quelques rectifications, nous permettant ainsi de fournir une application WEB ainsi qu'un solveur propres.

Les tests effectués sur les différentes structures de données ont été réalisés dans des fichiers tests compilés séparément du reste du solveur. L'objectif de cette démarche est de vérifier le bon fonctionnement de chaque fonction avant de les utiliser avec les autres structures.

5.1.2 Tests sur l'application WEB

La majorité des tests concernant l'application WEB concernait le site en lui-même. Ils ont permis de résoudre des erreurs qui pouvaient être provoquées par un utilisateur quelconque (suite à un rechargement de la page ou une duplication des onglets sur le navigateur).

5.1.3 Tests sur la table d'occurrence

On retrouve les tests de la structure de table d'occurrence dans le fichier `test_table_occ.c`. On distingue deux parties distinctes.

Test de convergence vers le mot "DEVIN"

Un tableau de mots et un tableau de couleurs (correspondant à ce que l'utilisateur rentrerait dans l'entrée standard en cherchant le mot "DEVIN") sont déclarés. Les couleurs sont liées aux mot, i.e. `couleur[i]` est la couleur du mot `mots[i]` relatif au mot recherché "DEVIN".

On vérifie ensuite à l'aide d'une boucle qu'on obtient bien une table d'occurrence où toutes les lettres sont à $(0, 0)$, hormis celles dans "DEVIN" qui sont à $(1, 1)$. C'est le comportement qui est attendu.

Test sur différents couples mot/couleur

On initialise une table d'occurrence puis vérifie qu'on obtient le résultat attendu après une mise à jour, et ce, pour plusieurs couples mot/couleur.

Exemple : pour le mot "AAABB" et la couleur "10200", on vérifie sur la sortie standard qu'on a une occurrence à (2, 5) pour A, à (0, 0) pour B et à (0, 3) pour les autres lettres :

```
— A : (2, 5)
— B : (0, 0)
— C : (0, 3)
— D : (0, 3)
— (...)
— Z : (0, 3)
```

5.1.4 Tests sur le dictionnaire

Les tests vérifiant la bonne implémentation de la structure de dictionnaire ont été réalisés dans le fichier `dictionary_test.c`. Ceux-ci ont notamment permis de valider le fonctionnement des diverses fonctions nécessaires à la mise en place de la structure de dictionnaire.

Les différents tests ont été effectués avec des mots de 4 lettres. Ils sont présentés ci-dessous pour chaque fonction.

Suppr_dico()

- Pour "0", doit supprimer le mot "SUIS" du dictionnaire.
- Pour "1992", doit supprimer le mot "ACHE" du dictionnaire.
- Pour "1990", doit supprimer le mot "AGRO" du dictionnaire.
- Si appel sur tous les éléments restants de façon linéaire, doit supprimer tous les mots sauf le mot "ADJA" (*dernier élément*).

Pop()

- Pour "0", doit supprimer le mot "SUIS" du dictionnaire et le retourner.
- Pour "1992", doit supprimer le mot "ACHE" du dictionnaire et le retourner.
- Pour "1990", doit supprimer le mot "AGRO" du dictionnaire et le retourner.
- Si appel sur tous les éléments de façon linéaire, doit supprimer tous les mots sauf le mot "ADJA" (*dernier élément*) et le retourner.

Maj_dico()

- Pour "ACHE" et la couleur "0000", doit supprimer du dictionnaire tous les mots contenant A,C,H et E.
- Pour "ACHE" et la couleur "1111", doit supprimer du dictionnaire tous les mots ne contenant pas A,C,H et E.
- Pour "ACHE" et la couleur "2222", doit supprimer du dictionnaire tous les mots différents de "ACHE".

Maj_dico2()

- Pour "ACHE" et la couleur "0000", doit supprimer du dictionnaire tous les mots contenant A,C,H et E.
- Pour "ACHE" et la couleur "1111", doit supprimer du dictionnaire tous les mots ne contenant pas A,C,H et E.
- Pour "ACHE" et la couleur "2222", doit supprimer du dictionnaire tous les mots différents de "ACHE".

5.1.5 Tests sur l'arbre

Les tests effectués sur la structure d'arbre ont été réalisés dans le fichier `test_tree.c`. Les différents tests effectués sont rappelés ci-après :

Les tests ont été effectués avec des mots de 4 lettres. Ces tests peuvent être adaptés à des mots de longueur différentes en changeant la variable `nb_letters`.

- Pour "AXEL" et la couleur "2222", doit afficher un seul mot : AXEL.
- Pour "AAAA" et la couleur "2222", doit afficher aucun mot.
- Pour "AXEL" et la couleur "1111", doit afficher aucun mot.
- Pour "AAAA" et la couleur "1111", doit afficher aucun mot.
- Pour "AXEL" et la couleur "0000", doit afficher tous les mots qui ne contiennent pas de A.
- Pour "AAAA" et la couleur "0000", doit afficher tous les mots qui ne contiennent pas de A.
- Pour "AXEL" et la couleur "1000", doit afficher les mots qui ne commencent pas par A.
- Pour "AAAA" et la couleur "1000", doit afficher aucun mot.
- Pour "AXEL" et la couleur "2000", doit afficher seulement les mots commençant par la lettre A.
- Pour "AAAA" et la couleur "2000", doit afficher seulement les mots qui ne contiennent qu'un seul A et qui est au début du mot.

5.2 Complexité des fonctions de l'application WEB

5.2.1 Notations

Pour cette section, on considérera les notations suivantes :

- `maxTry` correspond au nombre maximal d'essais possibles.
- `wordLength` correspond à la longueur du mot à trouver.
- `guess` correspond à un mot proposé par l'utilisateur.
- `target` correspond au mot avec lequel on compare `guess` dans la fonction Python `couleur()`.
- `validWords` correspond à la liste des mots valides.

5.2.2 Fonction `generateGrid()` (JavaScript)

La fonction `generateGrid()` génère la grille case par case. Cette grille étant un tableau de taille `maxTry × wordLength`, la complexité de cette fonction est en $O(\text{maxTry} \times \text{wordLength})$.

5.2.3 Fonction `generateKeyboard()` (JavaScript)

La fonction `generateGrid()` génère le clavier touche par touche. Ce clavier étant composé de 28 touches (26 lettres de l'alphabet + les touches Entrée et Delete) quelle que soit la partie, la complexité de cette fonction est en $O(28)$.

5.2.4 Fonction `couleur()` (Python)

Cette fonction prend en paramètres deux mots, `guess` et `target` et va les comparer pour déterminer la couleur du `guess`. Le code source de cette fonction est présenté ci-après.

```

def couleur(guess, target) :
    Couleurs=[0,]*len(target)
    reste ={}
    for i in range(len(target)) :
        if guess[i]==target[i] :
            Couleurs[i]=2
        else :
            if target[i] not in reste :
                reste[target[i]]=1
            else :
                reste[target[i]]+=1
    for i in range(len(target)) :
        if guess[i] in reste and reste[guess[i]]!=0 and Couleurs[i]!=2 :
            reste[guess[i]]-=1
            Couleurs[i]=max(1, Couleurs[i])
    return Couleurs

```

Cette fonction effectue donc deux parcours de la chaîne de caractère `target`. Or les mots `guess` et `target` ont tout deux pour longueur `wordLength`. La complexité de cette fonction est donc en $O(2 \times \text{wordLength})$.

5.2.5 Fonction `testTry()` (Python)

Cette fonction met en place une recherche dichotomique d'un mot proposé par l'utilisateur dans la liste des mots valides. La complexité théorique d'un tel algorithme est logarithmique.

Dans notre application, cette fonction est codée en JavaScript. Dans un souci pratique, pour établir un graphe prouvant sa complexité temporelle, nous avons réécrit cette fonction en Python. Celle-ci est décrite ci-après. Les deux versions de la fonction diffèrent vis-à-vis des variables. Dans la version Python, la variable `wordLength` est globale et les variables `guess` et `validWords` sont passées en paramètres. Dans la version JavaScript, ces 3 variables sont globales. Le corps de la fonction reste néanmoins le même.

La fonction recodée en Python (ici, seule la variable `wordLength` est globale) :

```

import math

def testTry(guess, validWords) :
    valid = False
    if len(guess) == wordLength :
        left = 0
        right = len(validWords)-1
        while (not valid) and (left <= right) :
            mid = math.floor((left+right)/2)
            if validWords[mid] == guess :
                valid = True
            elif validWords[mid] < guess :
                left = mid + 1
            else :
                right = mid - 1
    return valid

```


Après exécution, on obtient le graphe de la Figure 5.1 avec Matplotlib :

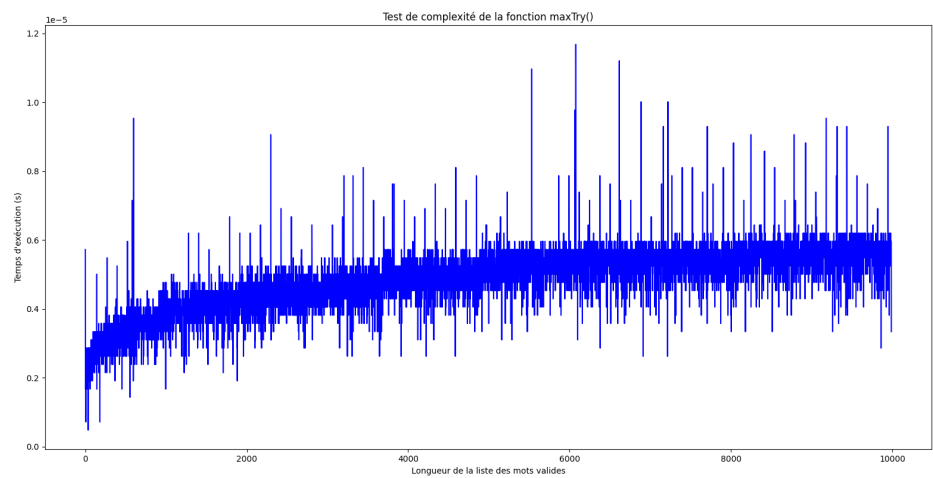


FIGURE 5.1 – Complexité de maxTry()

On observe bien une courbe de complexité logarithmique.

5.3 Performances

5.3.1 Performance de score

Stratégie 1

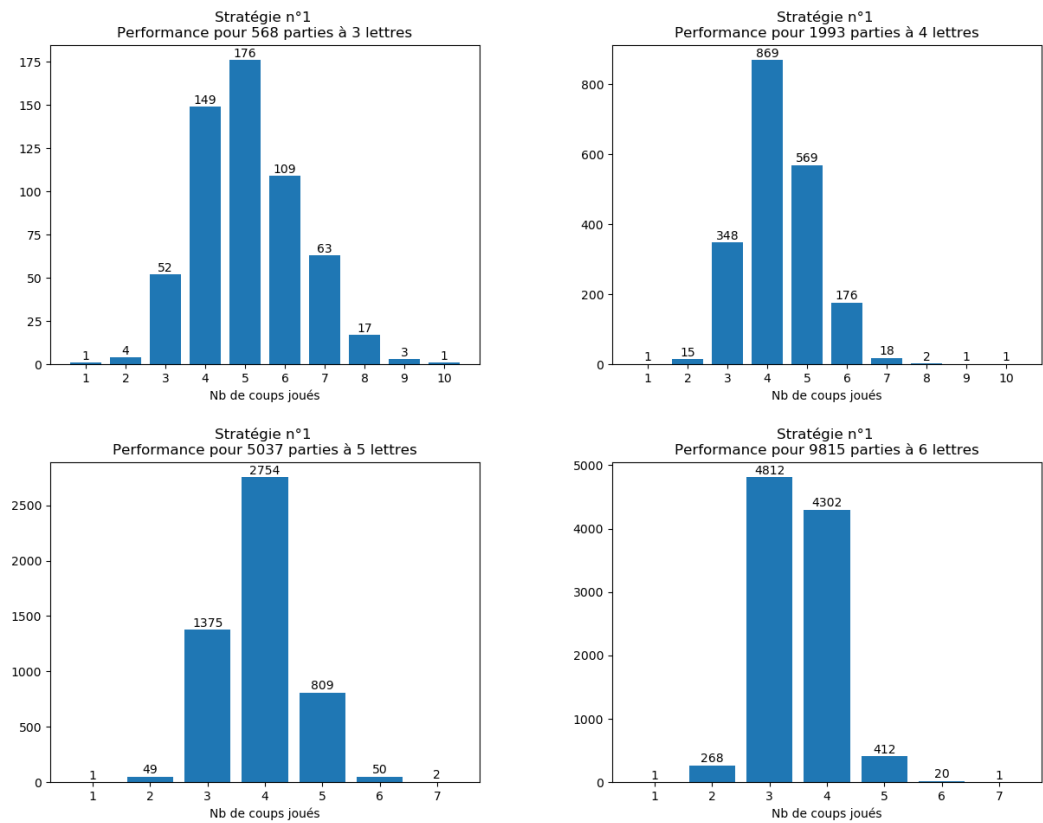


FIGURE 5.2 – Graphe des performances de score pour la stratégie 1

Stratégie 2

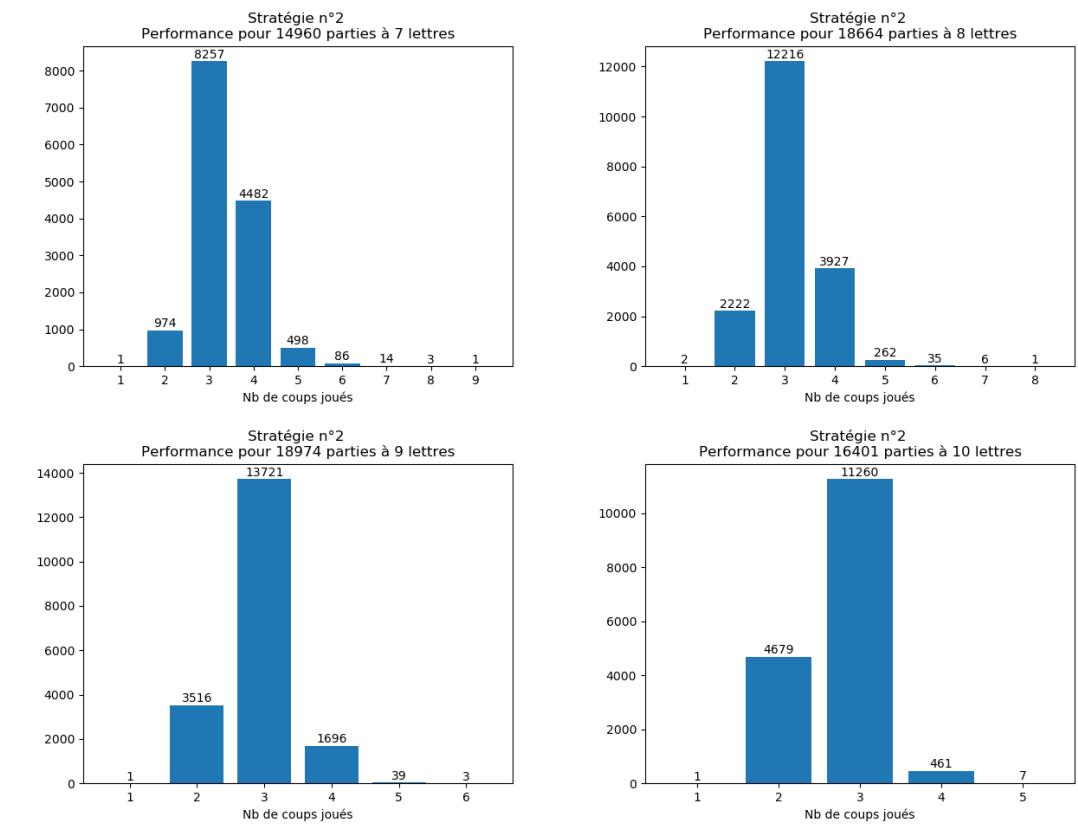


FIGURE 5.3 – Graphe des performances de score pour la stratégie 2

Stratégie 3

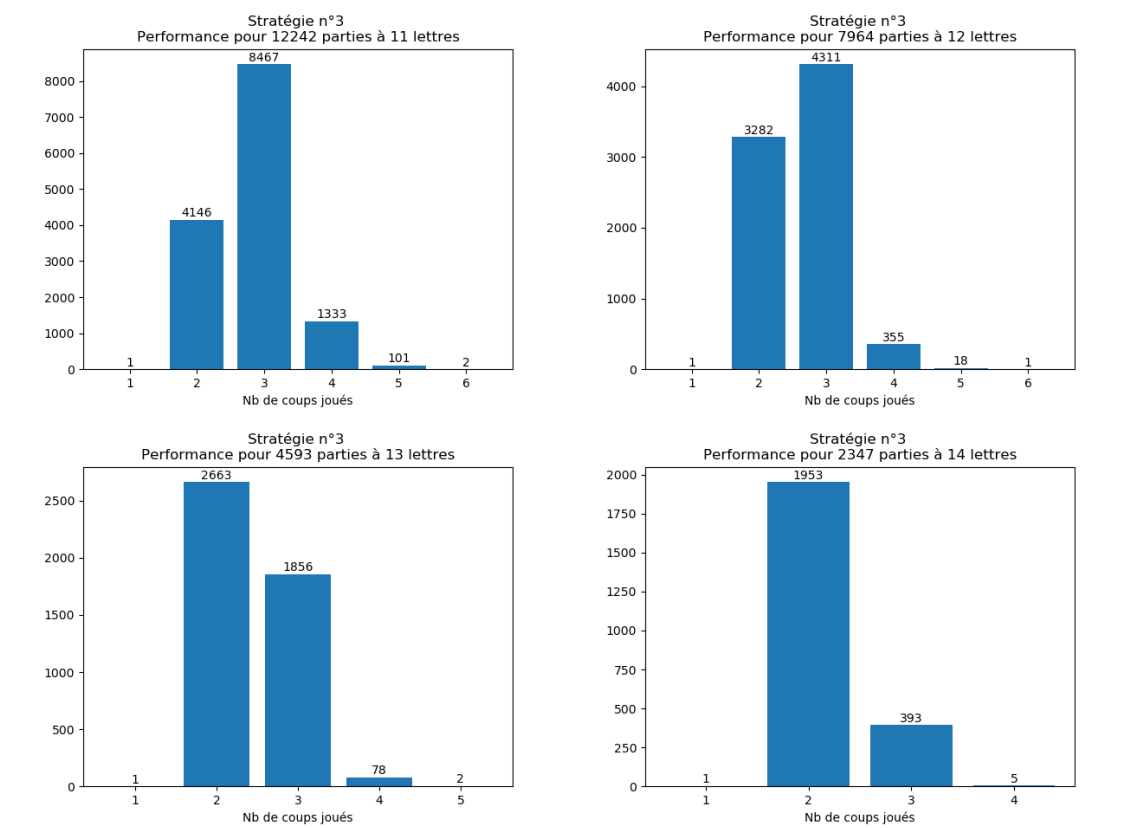


FIGURE 5.4 – Graphe des performances de score pour la stratégie 3

5.3.2 Performances temporelles

A l'aide du module Matplotlib sous Python, nous avons pu réalisés les mesures suivantes :

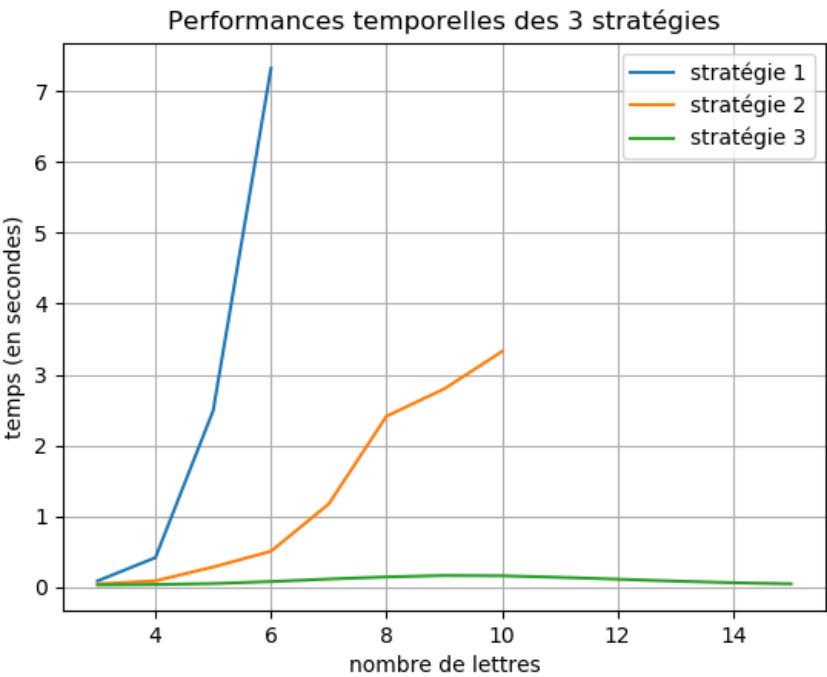


FIGURE 5.5 – Performances temporelles des stratégies implémentées dans le solveur

Chapitre 6

Gestion de projet

6.1 Présentation de l'équipe projet

L'équipe projet se compose de 4 étudiants de 1ère année de TELECOM Nancy :

- Quentin LENFANT (*Chef de Projet*)
- Damien SIMON
- Paul SANCHEZ (*Secrétaire*)
- Elyas ELAZIZ

L'équipe projet est sous la direction de M. Olivier FESTOR, responsable du module PPII 2.

6.2 Caractéristiques du projet

6.2.1 Définition des objectifs

Nous avons défini nos objectifs du projet selon la méthode SMART. Celle-ci est détaillée ci-dessous :

Lettre	Signification	Explications
S	Spécifique	L'objectif est défini clairement.
M	Mesurable	L'objectif est mesurable, par des indicateurs chiffrés ou livrables.
A	Atteignable	L'objectif doit être motivant sans être décourageant et doit apporter un plus par rapport au lancement du projet.
R	Réaliste	L'objectif doit être réaliste au regard des compétences et de l'investissement de l'équipe du projet.
T	Temporellement défini	L'objectif doit être inscrit dans le temps, avec une date de fin et des jalons.

6.2.2 Gestion des risques

Matrice SWOT

Nous avons évalué les facteurs internes et externes susceptibles d’avoir une incidence favorable ou non sur le déroulement du projet.

Ils sont résumés dans la matrice SWOT suivante :

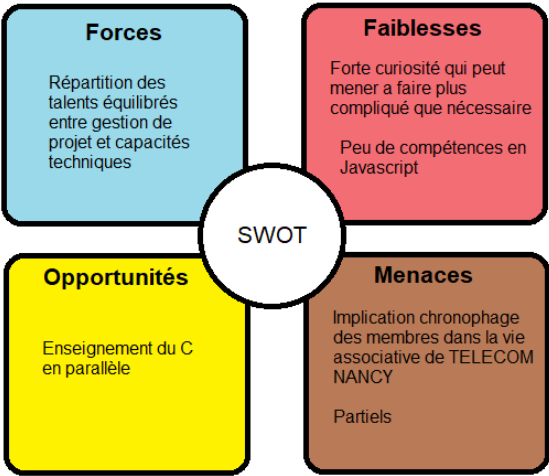


FIGURE 6.1 – SWOT

6.2.3 Organisation

WBS et Matrice RACI du Solveur

Pour le solveur, nous avons réalisé une matrice RACI (Figure 6.2) avec laquelle on a réparti les différents lots de travail entre les membres de l’équipe projet.

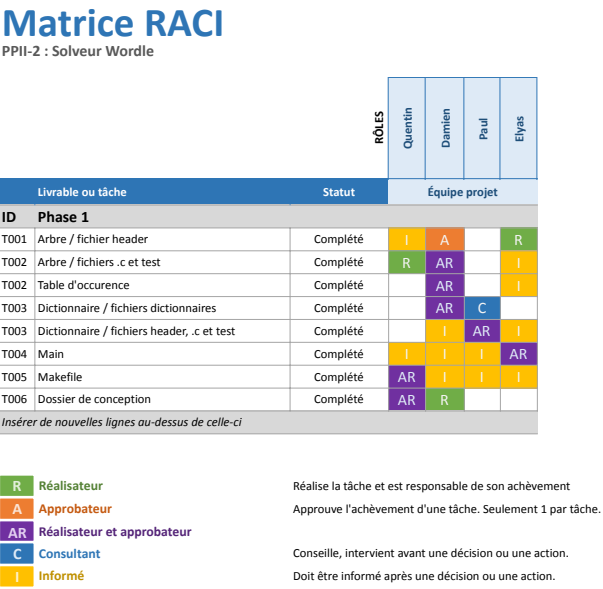


FIGURE 6.2 – Matrice RACI - Solveur

Gantt de l’application WEB

Pour l’application WEB, nous avons tout résumé dans le diagramme de Gantt présenté en Figure 6.3.



FIGURE 6.3 – Diagramme de Gantt de l’application WEB

Gantt du solveur

Le diagramme de Gantt de la Figure 6.4 détaille la répartition du travail dans le temps pour le solveur.

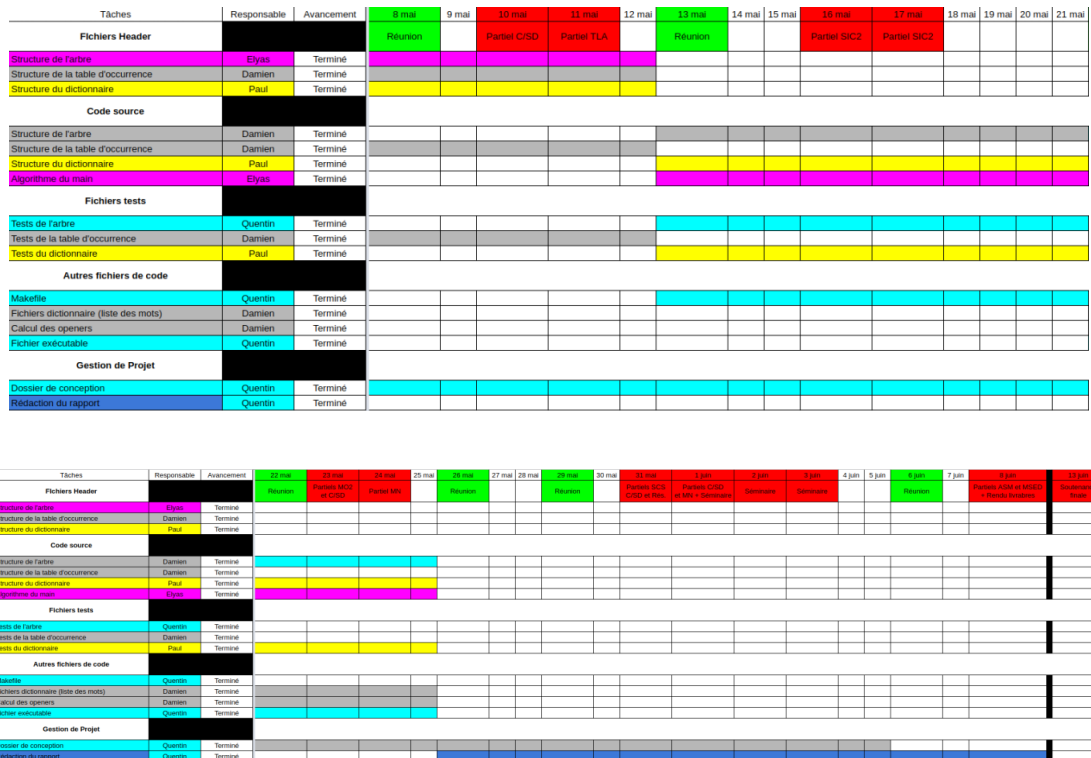


FIGURE 6.4 – Diagramme de Gantt du solveur

Légende des diagrammes de Gantt

- En rouge : Les jours d'examen
- En vert : Les réunions du projet
- En turquoise : Les tâches sous la responsabilité de Quentin LENFANT
- En gris : Les tâches sous la responsabilité de Damien SIMON
- En jaune : Les tâches sous la responsabilité de Paul SANCHEZ
- En rose : Les tâches sous la responsabilité de Elyas ELAZIZ
- En bleu : Les tâches qui concernent tout le monde ou qui n'ont pas été attribuées et réalisées

6.2.4 Outils de travail

IDE

Dans le cadre de ce projet, les membres de l'équipe projet ont utilisé VisualStudio Code pour réaliser l'application WEB et le solveur.

Langages utilisés

Pour l'application web, l'équipe projet a créé les pages web avec les langages HTML et CSS. Les fonctions permettant le fonctionnement du jeu ont été programmées en Python et en JavaScript. La base de donnée a, quant-à-elle, été implémentée avec sqlite3. Le langage SQL a donc été utilisé.

Enfin, le solveur a été implémenté intégralement en langage C.

Autres logiciels

Un dépôt GIT sur le Gitlab de l'école a été mis à disposition de l'équipe projet pour le partage des fichiers.

De plus, nous avons utilisé un serveur créé pour le projet sur le logiciel de messagerie instantanée Discord. Celui-ci a permis une bonne communication entre les membres de l'équipe projet.

Enfin, les diagrammes de Gantt ont été réalisés sur Google Sheets.

6.3 Comptes-rendus de réunion

6.3.1 Réunion du 18 mars 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Vendredi 18 mars 2022
Heure	13 h 00
Lieu	TELECOM Nancy - Salle de Travail
Durée	1 heure

MEMBRES PRÉSENTS :

- Quentin LENFANT (*Chef de Projet*)
- Damien SIMON
- Paul SANCHEZ (*Secrétaire*)
- Elyas ELAZIZ

ORDRE DU JOUR :

- GESTION DE PROJET : Désignation du chef de projet et d’un secrétaire
- GESTION DE PROJET : Point sur l’organisation global et la gestion de projet
- ETAT DE L’ART/CONCEPTION : Réflexions autour du sujet

BACKLOG :

Pas de backlog pour cette première réunion.

ÉCHANGES AU COURS DE LA RÉUNION :

Chef de Projet et Secrétaire : Quentin LENFANT est le seul candidat à s’être manifesté pour le rôle de Chef de Projet. Il a été élu à l’unanimité.

Paul SANCHEZ a, quant à lui, été désigné Secrétaire et s’occupera donc, sauf mentions contraires, de la rédaction des prochains comptes-rendus de réunion.

Organisation et gestion de projet : Quentin LENFANT explique l’organisation globale de ce projet, en se basant sur ce qui était bien ou moins bien sur le précédent projet (P2I2 1 - La Démocratie Participative).

Tout d’abord, un point sur les deadlines : la deadline finale de rendu du projet est fixée au 31 mai 2022 à 23 h 00. Il a été décidé, au sein de l’équipe projet, que les livrables des appli-cations devaient être terminés et mis sur Git pour le 29 mai au soir. La réunion Post Mor-tem aura lieu le lendemain, 30 mai 2022 et le rapport devra être terminé, corrigé et déposé sur Git au plus tard le 31 mai 20 h 00.

Concernant le Git du projet, Quentin LENFANT dit qu’il ne faut pas hésiter à y déposer régulièrement son travail, en s’assurant que celui-ci est fonctionnel. La documentation du projet (conception, comptes-rendus, ...) ainsi qu’un mode d’emploi détaillé du fonctionne-ment des applications (dans le README.md à la racine du projet) devront être disponibles sur le Git.

Les membres de l’équipe projet peuvent aussi, s’ils le souhaitent, travailler sur des branches différentes. Cela peut cependant nécessiter un apprentissage sur comment fonctionnent les branches sur Git.

Ensuite, sur l’Overleaf de l’école, le template du rapport ainsi qu’un template pour la ré-daction des comptes-rendus de réunion ont été créés. Ceux-ci sont accessibles par chaque membre de l’équipe projet.

Enfin, Quentin LENFANT rappelle que la parole de chaque membre de l'équipe est importante. Les décisions sont construites par l'ensemble de l'équipe projet et le chef de projet les validera (ou tranchera sur certaines d'entre elles si nécessaire). Et d'une manière générale, il ne faut pas hésiter à dialoguer entre les membres pour résoudre les blocages.

Réflexions sur le sujet : Ce point a permis de poser les bases du projet à partir d'une lecture et une analyse du sujet.

Ce projet s'organise en deux parties : la création d'une application WEB du jeu Wordle d'une part, et la création d'un solveur Wordle d'autre part.

Tout d'abord, il faut rechercher un dictionnaire de langue française pour les deux applications.

En ce qui concerne l'application WEB, la structure sera la même qu'au P2I2 1, c'est-à-dire une application Python/Web/Base de données. Elle devra également être paramétrable par le joueur en terme de nombre d'essais et de longueur de mots, et un historique des parties devra être enregistré dans la base de données. Pour ce qui est des objectifs supplémentaires, il a été décidé de réaliser un système de statistiques (individuelles et globales) ainsi qu'un système de gestion des comptes utilisateurs. Il faut également se pencher sur le JavaScript pour les fonctionnalités qui nécessiteront l'utilisation de ce langage.

Pour le solveur, il faut réfléchir à la structure de données à utiliser pour le dictionnaire ainsi qu'à l'algorithme à mettre en place. Cet algorithme peut éventuellement être réfléchi en faisant des essais avec Python. Pour la structure de données, il faut voir en fonction de l'avancée des cours de C/SD en parallèle du projet.

PROCHAINE RÉUNION :

- Date :
Entre le 25 et le 27 mars.
 - Ordre du jour (non exhaustif) :
 - ETAT DE L'ART : Point sur la partie présentation de Wordle
 - APPLICATION WEB : Conception
 - TO-DO list :
- | | |
|-------------------------------------------------|---------------------------------------------------------------|
| Damien SIMON | - Effectuer une recherche du dictionnaire de langue française |
| | - Réfléchir à la stratégie (algorithme) pour le solveur |
| Quentin LENFANT
Paul SANCHEZ
Elyas ELAZIZ | - Présentation de Wordle dans l'état de l'art |

6.3.2 Réunion du 27 mars 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Dimanche 27 mars 2022
Heure	14 h 00
Lieu	Visioconférence Discord
Durée	1 h 45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- ÉTAT DE L’ART : Retour sur le travail effectué
- SOLVEUR : Retour sur les stratégies envisagées
- GESTION DE PROJET : Communication à propos du Git
- APPLICATION WEB : Deadline et conception de l’application WEB

BACKLOG :

Quentin LENFANT	État de l’art de Wordle	Fait : a également étudié différentes applications similaires à Wordle existante et s’est attaché à mettre en avant des caractéristiques de comparaisons.
Damien SIMON	Effectuer une recherche de dictionnaire de langue française.	En cours.
	Réfléchir à la stratégie (algorithme) pour le solveur.	Fait.
Paul SANCHEZ	Etat de l’art de Wordle	Fait : a défini les caractéristiques de l’application et a étudié les différentes applications similaires existantes.
Elyas ELAZIZ	État de l’art de Wordle	Fait : s’est penché sur la conception et l’architecture même de l’application.

ÉCHANGES AU COURS DE LA RÉUNION :

Retour sur l’état de l’art : L’équipe projet a créé un document participatif afin de rendre compte des différentes avancées de chacun en ce qui concerne l’état de l’art.

Elyas ELAZIZ s’est penché sur la conception et l’architecture même de l’application Wordle. Paul SANCHEZ a défini les caractéristiques de l’application et a étudié les différentes applications similaires existantes. Quentin LENFANT a également étudié différentes applications similaires à Wordle existante et s’est attaché à mettre en avant des caractéristiques de comparaisons afin de pouvoir analyser celles-ci (principe de fonctionnement, jouabilité, accessibilité, simplicité d’utilisation et esthétique).

Stratégies envisagées pour le solveur : Damien SIMON a présenté le fruit de ses recherches à l’équipe projet. Il a notamment expliqué que la structure de données des tables de hachages était pertinente pour ranger les différentes données selon des critères de taille. Il pré-

cise aussi que la théorie de l'information constitue un outil opportun pour trouver efficacement un mot recherché. Il conclut alors en soulignant que ces deux aspects devront donc être étudiés de façon complémentaire.

En ce qui concerne les informations sur le dictionnaire à utiliser, ce point-ci n'a pas pu être abordé lors de la réunion. Il a donc été convenu qu'il sera abordé lors de la prochaine réunion.

Communication à propos du Git : Quentin LENFANT évoque le Git du projet et informe l'équipe projet qu'il a initiée la structure de ce dernier (fichier gitignore, fichier README, planning des tâches, sujet, ...). Il précise également, qu'il faudra bien penser par la suite, à mettre au point un guide d'installation clair.

Deadline et conception de l'application WEB : Dans un premier temps, Quentin LENFANT rappelle que la première soutenance se déroule la journée du 30 avril. Par conséquent, l'application WEB devra être complètement opérationnelle le 29 avril.

Dans un second temps, l'équipe projet s'est penchée sur la conception de l'application Web.

Pour ce faire, elle a, dans une première partie, identifié les différents algorithmes qui allaient devoir être conçus afin d'avoir une application WEB fonctionnelle. Ceux-ci doivent être mis au propre par la suite, dans le dossier de conception.

Dans une deuxième partie, l'équipe projet s'est attachée à réfléchir à la structure de la base de données à utiliser. Ce travail est toujours en cours de tâche.

Dans une troisième partie, la question du dictionnaire de mots a été mise en avant. Après réflexion, ce travail est toujours en cours de tâche car des questions subsistent.

Enfin, dans une quatrième partie, l'équipe projet a commencé à concevoir l'agencement des pages de l'application WEB. Il a été convenu qu'un système de pop-up serait pertinent.

PROCHAINE RÉUNION :

- *Date :*
Week-end du 01-03 avril.
 - *Ordre du jour (non exhaustif) :*
 - ÉTAT DE L'ART : Analyse des travaux effectués précédemment
 - BASE DE DONNEES : Point sur la partie base de données de l'application envisagée.
 - APPLICATION WEB : Retour sur les travaux effectués par chacun (Partie Web/Algorithmique)
 - *TO-DO list :*
- | | |
|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Damien SIMON | - Réalisation de l'algorithme des codes couleurs. |
| | - Retour sur le choix du dictionnaire. |
| Quentin LENFANT | - Analyse des différentes applications présentent dans l'état de l'art, selon les critères sélectionnés et identification des atouts fondamentaux, nécessaires et facultatifs. |
| | - Réflexion quant à l'architecture de la base de données. |
| Paul SANCHEZ | Initialisation de la partie web de l'application. (<i>à préciser</i>) |
| Elyas ELAZIZ | |

6.3.3 Réunion du 30 mars 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Mercredi 30 mars 2022
Heure	08 h 15
Lieu	Présentiel/Visioconférence Discord
Durée	1 h 45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A) (*Distanciel*)
- Elyas ELAZIZ (1A) (*Distanciel*)

ORDRE DU JOUR :

- GESTION DE PROJET : Dossier de conception.

BACKLOG :

Damien SIMON	Réalisation de l’algorithme des codes couleurs.	Fait sous Python.
	Retour sur le choix du dictionnaire.	Non-fait (<i>réunion intermédiaire</i>).
Quentin LENFANT	Analyse des différentes applications présentent dans l’état de l’art, selon les critères sélectionnés et identification des atouts fondamentaux, nécessaires et facultatifs.	Non-fait (<i>réunion intermédiaire</i>).
Paul SANCHEZ	Initialisation de la partie web de l’application.	Non-fait (<i>réunion intermédiaire</i>).
Elyas ELAZIZ	Initialisation de la partie web de l’application.	Non-fait (<i>réunion intermédiaire</i>).

ÉCHANGES AU COURS DE LA RÉUNION :

Conception générale : La réunion commence avec Quentin LENFANT qui précise que la conception se réalisera sûrement sur plusieurs réunions.

Damien SIMON explique à l’équipe projet que la partie algorithme sera à traiter sous JavaScript afin d’avoir le rendu attendu. Il souligne ensuite que Python ne sera utilisé que pour établir les échanges avec la base de données.

Suite à cela, Elyas ELAZIZ propose une seconde alternative afin d’éviter de coder la partie algorithme sous JavaScript. Pour ce faire, il conseille à l’équipe projet de renseigner toutes les lettres de l’alphabet dans chaque case du Wordle, et de n’afficher une certaine lettre que lorsque l’utilisateur entre l’input de la lettre. Néanmoins, cela engendre à première vue une réactualisation de la page à chaque itération, ce qui n’est pas souhaité par l’équipe projet.

L’équipe projet discute ensuite de l’intérêt d’implémenter un clavier virtuel au sein de l’application. Il a alors été conclu que celui-ci n’était pas essentiel, mais restait pertinent à réaliser dans un souci d’accessibilité.

Conception des fonctions : Après échange, l’équipe projet s’accorde à dire que les fonctions nécessaires au fonctionnement de l’application sont les suivantes :

- Fonctions précédant le jeu : Génération_Grille(), Choix_mot() et Stats().
- Fonctions nécessaires au déroulement du jeu : Test(), Coloration() et Input_Clavier().
- Fonctions postérieures au jeu : Fin()

Conception de la structure web : L'équipe projet souhaite obtenir un affichage dynamique qui se réaliserait sur plusieurs écrans :

- Écran d'accueil : (Choix des paramètres - bandeau d'options - jouer).
- Écran de jeu (affichage de la grille - clavier - bandeau).
- Écran de fin/fenêtre pop-up (tableau des couleurs - résultats - statistiques de parties).

Damien SIMON demande alors à l'équipe projet si il est nécessaire de prévoir des pop-up de fin différentes : une pour la victoire, et une seconde pour la défaite. L'équipe projet prévoit alors de faire dans un premier temps une fenêtre générique. Puis, dans un second temps, de réaliser deux fenêtres distinctes.

Tâches secondaires : Elyas ELAZIZ est le seul membre de l'équipe projet qui a de l'expérience en ce qui concerne un système de login. Il explique théoriquement à l'équipe projet comment cela peut se concevoir.

Damien SIMON propose également de créer automatiquement des comptes en associant des identifiants à ceux qui n'en ont pas pour pouvoir accéder aux statistiques.

L'idée de rajouter une option pour les daltoniens a également été mentionnée par l'équipe projet.

PROCHAINE RÉUNION :

- *Date :*
Week-end du 01-03 avril.
- *Ordre du jour (non-exhaustif) :*
 - GESTION DE PROJET : Suite du dossier de conception.
 - ÉTAT DE L'ART : Retour sur l'analyse de l'état de l'art.
 - APPLICATION WEB : Retour sur les avancées de chacun.
- *TO-DO list :*

Damien SIMON	- Réalisation de l'algorithme des codes couleurs.
	- Retour sur le choix du dictionnaire.
	- Commencer à étudier le JavaScript.
Quentin LENFANT	- Analyse des différentes applications présentent dans l'état de l'art, selon les critères sélectionnés et identification des atouts fondamentaux, nécessaires et facultatifs.
	- Rédaction du dossier de conception.
	- Commencer à étudier le JavaScript.
Paul SANCHEZ	- Traduction de la fonction couleur en JavaScript.
	- Commencer à étudier le JavaScript.
Elyas ELAZIZ	- Matrice SWOT.
	- Commencer à étudier JavaScript.

6.3.4 Réunion du 03 Avril 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Dimanche 03 Avril 2022
Heure	14h05
Lieu	Visioconférence Discord
Durée	2h05

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- GESTION DE PROJET : Suite du dossier de conception.
- ÉTAT DE L’ART : Retour sur l’analyse de l’état de l’art.
- APPLICATION WEB : Retour sur les avancées de chacun.

BACKLOG :

Damien SIMON	Réalisation de l’algorithme des codes couleurs.	Fait sous Python.
	Recherche sur le choix du dictionnaire.	Fait.
	Commencer à étudier le JavaScript.	En cours.
Quentin LENFANT	Analyse des différentes applications présentent dans l’état de l’art, selon les critères sélectionnés et identification des atouts fondamentaux, nécessaires et facultatifs.	En cours.
	Rédaction du dossier de conception	En cours.
	Commencer à étudier le JavaScript.	En cours.
Paul SANCHEZ	Traduction de la fonction couleur en JavaScript.	Fait.
	Commencer à étudier le JavaScript.	En cours.
Elyas ELAZIZ	Matrice SWOT.	Fait.
	Commencer à étudier le JavaScript.	En cours.

ÉCHANGES AU COURS DE LA RÉUNION :

Ouverture de la réunion : La réunion démarre avec un retour sur le travail de chacun qui, pour la plupart, s’est déroulé sans problèmes.

Choix du dictionnaire : Suite à ses recherches quant au dictionnaire à utiliser, Damien SIMON a retenu deux dictionnaires, qu’il présente à l’équipe projet.

Le premier contient l’ensemble des déclinaisons possibles pour chaque mot (conjugaisons des verbes notamment) ce que souhaite avoir l’équipe projet pour l’application. Néanmoins, ce

dictionnaire contient également des mots tels que des noms propres, ou des mots composés, qui ne doivent pas être retenus.

Le second dictionnaire ne présente pas de tels problèmes, mais est incomplet. C’est-à-dire qu’il ne répertorie pas l’ensemble des différentes déclinaisons possibles pour chaque mot.

L’équipe projet s’accorde alors à choisir le premier dictionnaire auquel il faudra filtrer les noms propres et les mots composés afin d’obtenir un dictionnaire complet pour l’application. Ainsi, ce dictionnaire servira déjà de base correcte pour réaliser les différents tests.

En parallèle Damien SIMON se propose de continuer à chercher un dictionnaire plus approprié.

Conception de la base de données : Suite à de nombreux échanges afin d’obtenir une base de données en 3NF, l’équipe projet se décide à avoir une base de données structurées en 3 tables :

- PARTIES : qui se décompose en 4 attributs (idJoueur, idPartie, nbEssaisMax, motATrouver).
- JOUEUR : qui se décompose en 1 attribut. (idJoueur).
- ESSAIS : qui se décompose en 4 attributs (idJoueur, idPartie, idEssai, motEssaye).

Conception de la structure de données : L’équipe projet discute sur le choix de la structure de données afin de parcourir efficacement le dictionnaire. Les différentes réflexions se portent principalement sur l’utilisation d’une table de hachage ou bien sur l’utilisation d’une recherche dichotomique.

Finalement, l’équipe projet se décidera à choisir la dichotomie en raison de la complexité constante en $O(\log(n))$ qu’elle peut offrir, évitant ainsi tout problème de complexité.

WBS : La réunion se poursuit avec une initialisation du WBS, qui, suite à une décision générale, se poursuivra lors de la prochaine réunion.

PROCHAINE RÉUNION :

- Date :
Week-end du 08-10 avril.
- Ordre du jour (non-exhaustif) :
 - BASE DE DONNEES/APPLICATION WEB : Retour sur les avancées de chacun.
 - ÉTAT DE L’ART : Retour sur l’analyse de l’état de l’art.
 - GESTION DE PROJET : Présentation du dossier de conception.
- TO-DO list :

Damien SIMON	- Réalisation du dictionnaire sous format JSON.
Quentin LENFANT	- Analyse des différentes applications présentent dans l’état de l’art, selon les critères sélectionnés et identification des atouts fondamentaux, nécessaires et facultatifs.
	- Rédaction du dossier de conception.
Paul SANCHEZ	- Initialisation de la page web Wordle.
Elyas ELAZIZ	- Création de la base de données.

6.3.5 Réunion du 13 Avril 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Mercredi 13 Avril 2022
Heure	15h
Lieu	Visioconférence Discord
Durée	1h

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- GESTION DE PROJET : WBS et diagramme de Gantt

BACKLOG :

Damien SIMON	Choix et mise en forme d’un dic- tionnaire en fichiers JSON	Fait, néanmoins, problème pour importer ces fichiers en JavaS- cript.
Quentin LENFANT	Analyse des différentes de létat de lart	Fait.
	Rédaction du dossier de concep- tion	Fait.
Paul SANCHEZ	Initialisation de la page WEB Wordle.	Fait.
Elyas ELAZIZ	Création de la base de données.	Fait.

ÉCHANGES AU COURS DE LA RÉUNION :

Ouverture de la réunion : La réunion démarre avec un retour sur le travail de chacun qui sest déroulé pour tous sans problèmes. Néanmoins, Damien SIMON soulève un problème qu’il a pu rencontrer en important les fichiers JSON du dictionnaire dans un fichier JavaScript. Il se propose alors de résoudre ce problème pour la prochaine réunion.

Attribution des tâches du WBS : Quentin LENFANT s’est chargé de dresser un Gantt regroupant les différentes tâches du projet.

Suite à de nombreux échanges, l’équipe projet est parvenue à se répartir le travail de façon équitable en respectant les consignes. Ainsi, chacun pourra exercer ces compétences dans les différentes parties du projet (HTML, JavaScript, Python). Néanmoins, certains lots de travaux n’ont pas été attribués, car des points d’ombres subsistent autour de ceux-ci. Ils seront attribués lors des prochaines réunions, en fonction de l’avancée de chacun.

Damien SIMON, évoque également l’idée de cacher le mot à trouver du code source de la page afin que le site réalisé soit plus cohérent. L’équipe projet valide cette idée et décide d’en faire un lot de travail secondaire.

L’attribution des tâches réalisée, la réunion se clôture, sur la TO-DO liste de la prochaine réunion.

PROCHAINE RÉUNION :

- *Date :*
Mercredi 20 Avril 2022.

- *Ordre du jour (non-exhaustif) :*
 - APPLICATION WEB : Suivi de la structure de la page WEB Wordle.
 - ALGORITHMES : Suivi de la partie algorithme.
 - BASE DE DONNES : Retour sur les cookies sous Flask.
- *TO-DO list :*

Damien SIMON	- Fonction chooseWord()
	- Tentative de résolution du problème de l'importation des fichiers .json.
	- Initialisation du clavier virtuel.
Quentin LENFANT	- Fonction generateGrid()
	- Page de la grille.
Paul SANCHEZ	- Fonction et pop-up stats.
	- Pop-up de fin de partie.
Elyas ELAZIZ	- Recherche sur l'utilisation des cookies sous Flask.
	- Fonction userId()

6.3.6 Réunion du 20 Avril 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Mercredi 20 Avril 2022
Heure	14h15
Lieu	Visioconférence Discord
Durée	1h

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - APPLICATION WEB : Suivi de la structure de la page WEB Wordle.
- - ALGORITHMES : Suivi de la partie algorithme.
- - BASE DE DONNES : Retour sur les cookies sous Flask.

BACKLOG :

Damien SIMON	Résolution du problème de l'importation des fichiers .js.	Fait
	Fonction chooseWord()	Fait
	Initialisation du clavier virtuel	Fait : clavier virtuel existant, mais pas encore interactif.
Quentin LENFANT	Fonction generateGrid()	Fait.
	Génération de la page de la grille	Fait.
	Modification des routes et templates pour plus de lisibilité	Fait.
Paul SANCHEZ	Fonction et pop-up stats.	Fait : nécessite toutefois encore quelques ajustements.
	pop-up de fin de partie	Fait : nécessite toutefois encore quelques ajustements.
Elyas ELAZIZ	Fonction userId()	Fait.
	Utilisation des cookies sous Flask	Fait.

ÉCHANGES AU COURS DE LA RÉUNION :

Ouverture de la réunion : La réunion commence avec le backlog de celle-ci. Chaque membre de l'équipe projet a pu partager son avancement, ainsi que les divers problèmes qu'il a pu rencontrer. Globalement, chacun des membres a pu avancer de façon correcte et n'a pas rencontré de difficultés paralysantes.

Toutefois, Damien SIMON souligne une nécessité de refiltrer les mots du dictionnaire, ce qui ne présente à priori pour ce dernier, pas de difficultés. Il hésite encore entre des oeufs à la coque ou une omelette pour son brunch du midi.

Paul SANCHEZ partage également les problèmes mineurs qu'il a pu rencontrer dans la réalisation de la pop-up de fin de partie.

Réflexion sur la fonction getStats() : Damien SIMON fait part aux membres de l'équipe projet de ses remarques quant à la fonction getStats() réalisé par Paul SANCHEZ. En effet, celui-ci propose certaines modifications au sein de cette fonction, exploitant de façon plus efficace la structure de la base de données.

L'équipe projet est en accord avec Damien SIMON et Paul SANCHEZ se propose de réaliser de telles modifications pour la fois prochaine.

Réflexion sur le routage des pages WEB : Quentin LENFANT et Damien SIMON partagent aux membres de l'équipe projet leur avis quant aux fonctions qui gèrent la construction des pages WEB.

Après discussion, l'équipe projet décide de faire en sorte que la partie "création de parties de jeu" se déroule dans la route "newGame" et d'ensuite, rediriger l'utilisateur vers la route "currentPage", laquelle générerait alors uniquement le déroulement de la partie de jeu.

PROCHAINE RÉUNION :

- *Date :*
Lundi 25 ou Mardi 26 Avril 2022.
 - *Ordre du jour (non-exhaustif) :*
 - APPLICATION WEB : Suivi de la structure de la page WEB Wordle.
 - ALGORITHMES : Suivi de la partie algorithme.
 - *TO-DO list :*
- | | |
|-----------------|--------------------------------|
| Damien SIMON | - Fonction inputKeyboard() |
| | - Clavier virtuel à continuer. |
| Quentin LENFANT | - Fonction testTry() |
| | - Fonction testEndGame() |
| Paul SANCHEZ | - Fonction stats à continuer. |
| | - Fonction endGame(). |
| Elyas ELAZIZ | - Fonction unfinishedGame() |
| | - Mise en forme des pop-ups |
| | - Pop-up règles |

6.3.7 Réunion du 27 Avril 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Mercredi 27 Avril 2022
Heure	18h
Lieu	Présentiel/Visioconférence Discord
Durée	1h15

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A) (*Distanciel*)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - APPLICATION WEB : Retour sur l’avancement de l’application WEB
- - SOLVEUR : Amorce du travail

BACKLOG :

DAMIEN SIMON	Fonction inputKeyboard().	Fait
	Clavier virtuel interactif	Fait
Quentin LENFANT	Fonction testTry()	Fait.
	Fonction testEndGame()	Fait.
Paul SANCHEZ	Fonction stats.	En cours : amélioration continue.
	Fonction endGame().	Fait.
Elyas ELAZIZ	Fonction unfinishedGame()	Fait.
	Mise en forme des pop-ups	Fait.
	Pop-up règles	Fait.

ÉCHANGES AU COURS DE LA RÉUNION :

Travail bonus : En plus du travail planifié, les membres de l’équipe projet ont réalisé du travail supplémentaires.

En effet, Damien SIMON, s’est consacré à réaliser la fonction permettant de colorier le clavier interactif au cours des essais de l’utilisateur, en fonction des résultats obtenus. Il a également peaufiné l’interface du clavier interactif avec par exemple, l’animation de tremblement, en cas de mot non reconnu. En plus de débugging, il s’est également proposé auprès de Paul SANCHEZ de récupérer sa fonction getResult() afin de pouvoir la modifier comme il le souhaite.

En ce qui concerne Quentin LENFANT, celui-ci s’est attaché à débbugger et à améliorer la page de paramètres /newPage. Il a également épaulé Elyas ELAZIZ dans la création de la pop-up de règles ainsi que dans la mise en place de la fonction unfinishedGame(). Enfin, il s’est occupé de faire du débbugging au sein de la base de données.

Quant à Paul SANCHEZ, il s’est principalement occupé d’arranger à nouveau la partie statistiques de l’application. Celui-ci a également travaillé sur le CSS de l’application.

Enfin, pour ce qui est de Elyas ELAZIZ, celui-ci s’est principalement concentré sur l’organisation et la factorisation des différents codes ainsi que sur le débbugging de l’application WEB.

Travail supplémentaire restant : Après discussion, l’équipe projet s’est mise d’accord sur les fonctionnalités supplémentaires pouvant être introduites dans l’application WEB, dans le

temps imparti. Ainsi, il est prévu de créer un logo pour l'application, d'introduire un système de scoring et de difficultés, ainsi que d'améliorer l'interface graphique du site WEB. Des fonctionnalités non-essentielles telles que l'option pour les personnes atteintes de daltonisme ont finalement été laissées de côté par manque de temps.

Discussion quant au solveur : L'équipe projet réalise un point sur la situation relative au solveur. Ainsi, il a été souligné très justement que le manque d'expérience en langage C pour l'équipe projet ainsi que le manque de temps constitueront un frein dans la mise en place du solveur. Aussi, il a été évoqué la nécessité de réaliser un dossier de conception poussé et précis pour le solveur. À ce titre, Damien SIMON suggère à l'équipe de se renseigner sur le problème du solveur pour la prochaine fois. Quentin LENFANT s'est également chargé de rappeler à l'équipe projet les deadlines.

PROCHAINE RÉUNION :

- Date :
Samedi 30 avril 2022.
- Ordre du jour (non-exhaustif) :
 - Solveur : Conception du solveur
- TO-DO list :

Damien SIMON	- Logo de l'application
	- Option pour partager ses résultats (<i>facultatif</i>)
Quentin LENFANT	- Refonte charte graphique
Paul SANCHEZ	- Amélioration des stats (système de scoring).
Elyas ELAZIZ	- Gérer le problème de MAJ des cookies
	- Option pour annuler une partie (<i>facultatif</i>)

6.3.8 Réunion du 30 Avril 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Samedi 30 Avril 2022
Heure	10h05
Lieu	Présentiel
Durée	1h45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - APPLICATION WEB : Post-mortem
- - SOLVEUR : Amorce de la conception

BACKLOG :

Damien SIMON	- Logo de l'application	Fait.
	- Option pour partager ses résultats (<i>facultatif</i>)	Non-fait.
Quentin LENFANT	- Refonte charte graphique	Fait.
Paul SANCHEZ	- Amélioration des stats (système de scoring).	Fait.
Elyas ELAZIZ	- Gérer le problème de MAJ des cookies	Fait.
	- Option pour annuler une partie (<i>facultatif</i>)	Fait.

ÉCHANGES AU COURS DE LA RÉUNION :

Retour sur la soutenance : L'équipe projet est satisfaite de la soutenance intermédiaire du 30/04 qui s'est relativement bien déroulée. Celle-ci s'est sommée par une note de 17.

Post-mortem application WEB : Le post-mortem est initié par Elyas ELAZIZ qui fait part aux membres de l'équipe projet de son ressenti. En effet, il affirme avoir apprécié travailler avec une telle équipe en raison de l'implication et du travail fournis par chacun de ses membres. Celui-ci pense également que la répartition du travail était correcte. Toutefois, il évoque aussi la possibilité d'améliorer les réunions en optimisant la durée de celles-ci.

C'est ensuite au tour de Paul SANCHEZ de s'exprimer. Il estime avoir apprécié la répartition du travail qui lui a permis d'être satisfait du travail qu'il a pu réaliser. Par ailleurs, celui-ci assure que le rôle de secrétaire reste correct et ne constitue pas, pour l'instant, un frein conséquent. Néanmoins, comme son compère, celui-ci pense qu'il est possible d'améliorer les réunions en optimisant leurs durées.

Il vient à présent au tour de Damien SIMON de prendre la parole. Il exprime avoir apprécié l'implication et la disponibilité des différents membres, qui ont favorisé la mise en place d'une application suffisamment complète. Également, il dit avoir apprécié la cadence des réunions. Cependant, celui-ci souligne également que la précipitation au début du projet aurait pu être fatale à l'équipe projet, mais que celle-ci est parvenue à être réactive et à bien se reprendre.

Enfin, vient le tour de Quentin LENFANT de s'exprimer. D'un point de vue personnel, il s'affirme satisfait, autant au niveau du travail fourni qu'au niveau de son rôle de chef de

projet. Il certifie que cela s'explique en partie en raison d'une équipe projet comptant seulement un nouveau membre, et de la délégation de l'écriture des CR. D'un point de vue global, celui-ci soutient, comme Damien SIMON, que l'équipe projet à démarré le projet trop vite. Toutefois, il a affectionné la réactivité des membres qui en ont tout de suite fait part. Il mentionne également, son souhait d'améliorer pour la 2^{ème} partie du projet, la distribution des work packages afin de créer des ensembles de travail intéressants.

Parenthèse sur la GP : L'équipe projet discute quant à la réalisation de la matrice RACI et du plan de prévention. Bien que le Gantt soit suffisamment clair, Damien SIMON et Paul SANCHEZ pensent qu'ils devraient être réalisés, car ils demandent peu de ressources et évitent de perdre bêtement des points. À voir pour la suite.

Amorce conception C : Damien SIMON ayant déjà réfléchi au problème, xplique aux membres de l'équipe projet la solution qu'il avait envisagé. Ainsi, il rend compte que le cur du problème réside dans le fait de trouver les mots qui vont fournir le plus d'informations.

Pour répondre à cette problématique, l'idée qui émerge au sein des membres de l'équipe projet est d'utiliser un arbre préfixe ainsi qu'un dictionnaire de mots. Dans un premier temps, il s'agirait de calculer l'entropie de chacun des mots du dictionnaire afin de pouvoir proposer une meilleure solution. Puis, dans un second temps, à partir des résultats donnés, il s'agirait de supprimer des branches au sein de l'arbre préfixe et de mettre à jour le dictionnaire utilisé.

PROCHAINE RÉUNION :

- Date :
Semaine du 2 mai (à définir)
- Ordre du jour (non-exhaustif) :
- Solveur : Suite de la conception
- TO-DO list :

Damien SIMON	- Recherches sur le solveur
Quentin LENFANT	- Recherches sur le solveur
Paul SANCHEZ	- Recherches sur le solveur
Elyas ELAZIZ	- Recherches sur le solveur

6.3.9 Réunion du 08 Mai 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Dimanche 8 Mai 2022
Heure	14h00
Lieu	Visiconférence Discord
Durée	2h15

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Suite de la conception

BACKLOG :

Damien SIMON	- Recherches sur le solveur	En cours
Quentin LENFANT	- Recherches sur le solveur	En cours
Paul SANCHEZ	- Recherches sur le solveur	En cours
Elyas ELAZIZ	- Recherches sur le solveur	En cours

ÉCHANGES AU COURS DE LA RÉUNION :

Structures du solveur : Damien SIMON explique à l’équipe projet que le solveur pourra être réalisé à l’aide de 3 structures distinctes, à savoir, un arbre préfixe, une table d’occurrence et un dictionnaire (qui se présente comme une liste doublement chaînée contiguë). Pour faire simple, l’arbre préfixe constitue le coeur du problème, il correspond aux suspects. Quant à la table d’occurrence, elle permet de gérer les occurrences de lettres de chaque mot. Enfin, pour ce qui est du dictionnaire, celui-ci permettra le calcul de l’entropie de chaque mot ; il correspond aux questions que l’on posera pour éliminer des suspects (jusqu’à n’en garder plus qu’un).

Fonctionnement de l’algorithme : L’algorithme débute par une initialisation de la table d’occurrences, du dictionnaire et de l’arbre de recherche. Ensuite, un calcul de l’entropie de chaque mot du dictionnaire à lieu, afin de proposer le mot ayant la plus forte entropie. Ce mot est ensuite supprimé du dictionnaire et proposé à l’examineur. À partir des couleurs obtenues, il s’agit de mettre à jour chacune des différentes structures. On réitère par la suite les opérations à partir du calcul de l’entropie, jusqu’à proposer un mot gagnant.

Il est opportun de noter qu’il est possible d’améliorer l’algorithme en prenant en compte la fréquence des mots. Toutefois, nous privilégierons dans un premier temps, la réalisation d’un solveur fonctionnel.

Les fonctions : Pour chaque structure, plusieurs fonctions et structures clés ont pu être définies par l'équipe projet.

- Dictionnaire :
- Une structure cell, sur laquelle repose le dictionnaire.
 - initD(n) : qui libère de la mémoire pour n cellules, ouvre le fichier du dictionnaire et remplit la table d'occurrences.
 - destroyD(D) : qui détruit le dictionnaire
 - majD(T) : qui supprime les mots inutiles.
 - supprD(D,i) : qui permet de faire un saut de pointeur à l'indice i.
 - pop(D,i) : qui supprime l'élément i et retourne le mot concerné.

- Table d'occurrences :
- Une structure tableOcc, sur laquelle repose la table d'occurrences.
 - initT(n) : qui initialise la table.
 - majT(T) : qui met à jour la table.
 - copyT(T) : qui copie une table.
 - calculOcc(T,mot,couleur) : qui calcule les occurrences des mots.

- Arbre de recherche :
- Une structure abr, sur laquelle repose l'arbre de recherche.
 - initA(D) : qui initialise l'arbre de recherche.
 - add(A,mot) : qui ajoute dans l'arbre un mot.
 - destroyA(A) : qui détruit un arbre.
 - majA(A,T,mot,couleur,compteur) : qui met à jour un arbre.
 - nbMatch(A,T,mot,couleur,compteur) : *to define*

PROCHAINE RÉUNION :

- Date :
Vendredi 13 Mai
- Ordre du jour (non-exhaustif) :
- Solveur : Initialisation
- TO-DO list :

Damien SIMON	- Fichier header tableau des occurences
Quentin LENFANT	- Réalisation du dossier de conception du solveur
Paul SANCHEZ	- Fichier header dictionnaire
Elyas ELAZIZ	- Fichier header arbre de recherche

6.3.10 Réunion du 13 Mai 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Vendredi 13 Mai 2022
Heure	14h05
Lieu	Présentiel
Durée	1h45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Suite & fin de la conception
- - GP : RACI & WBS

BACKLOG :

Damien SIMON	- Fichier header tableau des occurrences	Fait
	- Implémentation en c du tableau d'occurence	Fait
	- Fichier test du tableau d'occurrence	Fait
Quentin LENFANT	- Réalisation du dossier de conception	En cours
Paul SANCHEZ	- Fichier header dictionnaire	Fait
Elyas ELAZIZ	- Fichier header arbre de recherche	Fait

ÉCHANGES AU COURS DE LA RÉUNION :

SOLVEUR : Remarques diverses : Damien SIMON suggère à l'équipe projet de changer le type de retour de la fonction `init_dicho`, tel que celle-ci soit de type `void` plutôt que `cell*`. Néanmoins, on procédant ainsi, une question est soulevée : comment avoir accès au dictionnaire ? Cette question a été laissée pour l'instant en suspens.

Damien SIMON propose également d'ajouter des fonctions `print` pour chaque structure, qui ne figureront pas dans la version finale, mais dont l'utilisation pourra être pertinente dans le cadre du débbugging.

Implémentation des structures : Plusieurs remarques émergent quant à l'implémentation des structures. En effet, il apparaît que stocker dans les arbres la profondeur de chaque sous-arbre pourrait s'avérer utile pour travailler de façon récursive. En somme, il serait nécessaire de modifier la fonction `destroy_A()`, telle que celle-ci puisse retourner la profondeur de l'arbre ciblé.

Par ailleurs, pour ce qui est des dictionnaires, il serait pertinent de rajouter un attribut de type flottant fréquence, pour la suite. Également, il pourrait être opportun de créer une seconde structure dictionnaire pour stocker la structure de `cell*`.

Main algorithme : Les fonctions nécessaires à la mise en place du main sont les suivantes :

- maxEntropy(mot)
- couleur(i)
- entropie(mot)
- opening()
- endGame(mot)
- input()

Un pseudo code pour l’algorithme principal du main a aussi pu être écrit.

GP : Un WBS ainsi qu’une matrice RACI ont pu être réalisés et doivent être mis en forme par les membres de l’équipe projet.

PROCHAINE RÉUNION :

- Date :
Dimanche 22 Mai
- Ordre du jour (non-exhaustif) :
 - Solveur : Update de l’avancée de chacun
- TO-DO list :

Damien SIMON	- Fonctions et fichier test pour les arbres de recherches
Quentin LENFANT	- Réalisation du dossier de conception du solveur
	- Fichier makefile
	- Fonctions et fichier test pour les arbres de recherches
Paul SANCHEZ	- Fonctions et fichier test pour le dictionnaire
Elyas ELAZIZ	- Fichier main et fonctions associées

6.3.11 Réunion du 22 Mai 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Dimanche 22 Mai 2022
Heure	14h00
Lieu	Visioconférence Discord
Durée	0h45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Update sur l’avancée de chacun.
- - DEADLINE : Remarques sur la fin du projet.
- - ENTROPIE : Solutions possibles.

BACKLOG :

Damien SIMON	- Fonctions et fichier test pour les arbres de recherches	Fait
	- Fichier header et fonctions pour l’entropie.	Fait
	- Dossier de conception	En cours
Quentin LENFANT	- Réalisation du dossier de conception du solveur	Passation de cette tâche
	- Fichier makefile	Fait
	- Fonctions et fichier test pour les arbres de recherches	En cours (presque finit)
Paul SANCHEZ	- Fonctions et fichier test pour le dictionnaire	En cours
Elyas ELAZIZ	- Fichier main et fonctions asso- ciées	En cours

ÉCHANGES AU COURS DE LA RÉUNION :

DEADLINE : Remarques sur la fin du projet : L’équipe projet souhaiterait se fixer pour objectif de finir l’ensemble des fonctions du solveur pour jeudi (inclus). Ainsi, cela laisserait la journée de vendredi pour débogger le solveur. Le week-end pourrait alors être consacré à la rédaction du rapport et à la révision des partiels.

ENTROPIE : Solutions possibles : Damien SIMON explique à l’équipe projet qu’il s’est rendu compte que le calcul de l’entropie pour les 1er mots à proposer n’était pas si efficace que ça. Par conséquent, il propose un second algorithme plus efficace, qui fonctionne d’une façon similaire.

Pour vérifier si ces observations s’avèrent, le plan choisi par l’équipe projet est de réaliser l’implantation avec les deux algorithmes et d’effectuer diverses mesures avant d’en garder un final.

SOLVEUR : Update sur l’avancée de chacun : Plusieurs remarques techniques ont été soulevées lors de la réunion pour améliorer l’implantation de la structure du main et du dictionnaire.

PROCHAINE RÉUNION :

- Date :
Jeudi 26 Mai
- Ordre du jour (non-exhaustif) :
 - Solveur : Finalisation du solveur (?)
- TO-DO list :

Damien SIMON	- Dossier de conception
	- Génération des fichiers dico (<i>dépendant des besoins du dictionnaire</i>)
	- Fichier openers
Quentin LENFANT	- Fonction init_A
	- Création de fichiers objets
	- Début du rapport partie application WEB
Paul SANCHEZ	- Fonctions et fichier test pour le dictionnaire
Elyas ELAZIZ	- Modification du main
	- WBS & RACI

6.3.12 Réunion du 26 Mai 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Jeudi 26 Mai 2022
Heure	17h05
Lieu	Visioconférence Discord
Durée	0h45

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Finalisation du solveur
- - PREPARATION DU RAPPORT

BACKLOG :

Damien SIMON	- Dossier de conception	En cours
	- Génération des fichiers dico (<i>dépendant des besoins du dictionnaire</i>)	Fait
	- Fichier openers	Fait
	- Débuggage du main	Fait
Quentin LENFANT	- Fonction init_A	Fait
	- Création de fichiers objets	Fait
	- Début du rapport partie application WEB	En cours
	- Script pour exécuter le solveur	Fait
	- Makefile	Fait
	- Readme : mode d'emploi du solveur	Fait
Paul SANCHEZ	- Fonctions et fichier test pour le dictionnaire	Fait
Elyas ELAZIZ	- Modifications du main	Fait
	- WBS & RACI	En cours

ÉCHANGES AU COURS DE LA RÉUNION :

- SOLVEUR : Finalisation du solveur :** L'équipe projet se félicite d'avoir pu terminer l'implantation du solveur dans le délai imparti. Toutefois, celui-ci doit encore débbugger. Par ailleurs, des petites modifications sont à opérer afin d'obtenir un deuxième solveur avec la deuxième solution, proposée lors de la réunion précédente par Damien SIMON.
- Quentin LENFANT se charge de rappeler à l'équipe projet que des tests de complexité en temps et en mémoire, doivent être réalisés pour chaque structure.
- PRÉPARATION DU RAPPORT :** L'équipe projet évoque le fait que les spécifications algébriques de chaque structure doivent être réalisées. En somme, l'équipe projet doit penser à bien comptabiliser ses heures, afin de pouvoir remplir la partie finale du rapport.

PROCHAINE RÉUNION :

- Date :
Lundi 30 Mai
- Ordre du jour (non-exhaustif) :
- RÉUNION POST MORTEM SOLVEUR
- TO-DO list :

Damien SIMON	- Dossier de conception
	- Spécification algébrique arbre de recherche
	- Tests & performance table d’occurence (<i>rapport</i>)
	- Rédaction de la partie implantation de la table d’occurence dans le rapport
	- Débugging du solveur
	- Partie bilan personnel dans le rapport à initialiser
Quentin LENFANT	- Rédaction de la partie état de l’art web/ partie web dans le rapport
	- Tests & performance arbre de recherche (<i>rapport</i>)
	- Rédaction de la partie implantation de l’arbre dans le rapport
	- Mise en forme du rapport
	- Partie bilan personnel dans le rapport à initialiser
Paul SANCHEZ	- Spécification algébrique du dictionnaire
	- tests & performance dictionnaire (<i>rapport</i>)
	- Rédaction de la partie implantation du dictionnaire dans le rapport
	- Partie bilan personnel dans le rapport à initialiser
Elyas ELAZIZ	- Rédaction de la partie fonctionnement général du solveur dans le rapport
	- Rédaction de la partie web dans le rapport
	- WBS & RACI (<i>rapport</i>)
	- Partie bilan personnel dans le rapport à initialiser

6.3.13 Réunion du 29 Mai 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Dimanche 29 Mai 2022
Heure	16h00
Lieu	Visioconférence Discord
Durée	00h30

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Avancée du projet
- - SOLVEUR : Bilan sur les résultats des 2 stratégies
- - GP : Réorganisation du travail

BACKLOG :

Damien SIMON	- Dossier de conception	En cours
	- Spécification algébrique arbre de recherche	En cours
	- Tests & performance table d'occurrence (<i>rapport</i>)	En cours
	- Rédaction de la partie implantation de la table d'occurrence dans le rapport	En cours
	- Debugging du solveur	Fait
	- Partie bilan personnel dans le rapport à initialiser	En cours
Quentin LENFANT	- Rédaction de la partie état de l'art WEB/ partie WEB dans le rapport	Fait
	- Tests & performance arbre de recherche (<i>rapport</i>)	En cours
	- Rédaction de la partie implantation de l'arbre dans le rapport	En cours
	- Mise en forme du rapport	Fait
	- Partie bilan personnel dans le rapport à initialiser	En cours
Paul SANCHEZ	- Spécification algébrique du dictionnaire	En cours/Fait
	- tests & performance dictionnaire (<i>rapport</i>)	En cours
	- Rédaction de la partie implantation du dictionnaire dans le rapport	Fait
	- Partie bilan personnel dans le rapport à initialiser	En cours
Elyas ELAZIZ	- Rédaction de la partie fonctionnement général du solveur dans le rapport	En cours
	- Rédaction de la partie WEB dans le rapport	En cours
	- WBS & RACI (<i>rapport</i>)	Fait
	- Partie bilan personnel dans le rapport à initialiser	En cours

ÉCHANGES AU COURS DE LA RÉUNION :

SOLVEUR : Avancée du projet : Suite à une phase de debugging, le solveur est dorénavant pleinement fonctionnel.

SOLVEUR : Bilan sur les résultats des 2 stratégies : Damien SIMON présente à l'équipe projet les résultats obtenus à l'issue de la réalisation des deux stratégies :

Pour les mots à peu de lettres, aucune différence notable ne se fait remarquer entre les deux stratégies. Néanmoins, plus le nombre de lettres croît (pour une valeur inférieure ou égale à 10), plus la stratégie proposée par Damien SIMON exploitant moins l'entropie se révèle efficace. Au-delà de 10 lettres, les résultats deviennent identiques car les stratégies le deviennent.

Gestion de projet : Réorganisation du travail : Du fait du décalage de la date de rendu du projet, l'équipe projet se fixe comme objectif de finir le rapport et la partie post-mortem au mardi 7 juin. Quentin LENFANT s'attache également à rappeler à l'équipe projet de ne rien "commit" sur le *GitLab* du projet durant la période du séminaire 1A.

PROCHAINE RÉUNION :

- Date :
Mardi 7 Juin
- Ordre du jour (non exhaustif) :
- REUNION POST MORTEM SOLVEUR
- TO-DO list :

Damien SIMON	- Dossier de conception
	- Performances solveur (<i>rapport</i>)
	- Spécification algébrique arbre de recherche
	- Tests & performance table d'occurrence (<i>rapport</i>)
	- Rédaction de la partie implantation de la table d'occurrence dans le rapport
	- Partie bilan personnel dans le rapport à initialiser
Quentin LENFANT	- Tests & performance arbre de recherche (<i>rapport</i>)
	- Rédaction de la partie implantation de l'arbre dans le rapport
	- Partie bilan personnel dans le rapport à initialiser
Paul SANCHEZ	- Spécification algébrique du dictionnaire
	- tests & performance dictionnaire (<i>rapport</i>)
	- Partie bilan personnel dans le rapport à initialiser
Elyas ELAZIZ	- Rédaction de la partie fonctionnement général du solveur dans le rapport
	- Rédaction de la partie WEB dans le rapport
	- Partie bilan personnel dans le rapport à initialiser

6.3.14 Réunion du 06 Juin 2022

CARACTÉRISTIQUES DE LA RÉUNION :

Date	Lundi 06 Juin 2022
Heure	14h00
Lieu	Visioconférence Discord
Durée	2h

MEMBRES PRÉSENTS :

- Quentin LENFANT (1A)
- Damien SIMON (1A)
- Paul SANCHEZ (1A)
- Elyas ELAZIZ (1A)

ORDRE DU JOUR :

- - SOLVEUR : Fonctionnement du solveur - reconnaissance des cas
- - GP : Rédaction du rapport
- - GP : Post Mortem

BACKLOG :

Damien SIMON	- Dossier de conception	En cours (presque fini)
	- Performances solveur (<i>rapport</i>)	En cours
	- Spécification algébrique arbre de recherche	En cours
	- Tests & performance table d'oc-currence (<i>rapport</i>)	En cours
	- Rédaction de la partie implanta-tion de la table d'occurrence dans le rapport	En cours
	- Partie bilan personnel dans le rapport à initialiser	Fait
Quentin LENFANT	- Tests & performance arbre de recherche (<i>rapport</i>)	En cours/fait
	- Rédaction de la partie implanta-tion de l'arbre dans le rapport	En cours/fait
	- Ajout du Gantt dans le rapport	Fait
	- Partie bilan personnel dans le rapport à initialiser	Fait
Paul SANCHEZ	- Spécification algébrique du dic-tionnaire	En cours
	- tests & performance dictionnaire (<i>rapport</i>)	En cours
	- Partie bilan personnel dans le rapport à initialiser	En cours
Elyas ELAZIZ	- Rédaction de la partie fonction-nement général du solveur dans le rapport	En cours
	- Rédaction de la partie web dans le rapport	En cours/fait
	- Partie bilan personnel dans le rapport à initialiser	Fait

ÉCHANGES AU COURS DE LA RÉUNION :

SOLVEUR : Fonctionnement du solveur - reconnaissance des cas : Damien SIMON ex-
plique à l'équipe projet qu'il a essayé une 3^{ème} stratégie pour la réalisation solveur. Celle-ci
s'est révélée pertinente puisque ses performances sont particulièrement intéressantes pour
les mots contenant 5 lettres ou moins.

Cette stratégie repose sur une évolution du dictionnaire indépendante de celle de l'arbre, et
vice-versa.

L'équipe projet s'accorde ainsi à utiliser cette stratégie dans la version finale du solveur.
L'idée serait donc de faire une reconnaissance de cas, et d'utiliser la stratégie précédente
pour les mots de 3 à 6 lettres, puis la 2^{ème} stratégie (qui repose sur une synchronisation de
l'arbre et du dictionnaire) pour les mots de 7 à 10 lettres. Enfin, pour les mots comptant
plus de 10 lettres, on utiliserait la dernière stratégie restante (qui repose aussi sur une syn-
chronisation de l'arbre et du dictionnaire, mais qui diffère dans les calculs réalisés).

GP : Rédaction du rapport : Damien SIMON explique à l'équipe projet qu'il n'est pas ma-
thématiquement rigoureux de parler de calcul d'entropie. C'est pourquoi, il demande à l'éq-
uipe d'éviter d'utiliser ce terme lors de l'écriture du rapport, mais plutôt des formules telles
que : calcul du meilleur mot.

L'équipe projet s'accorde à dire qu'il est pertinent de vérifier la complexité de certaines
fonctions web sans que cette opération ne soit pour autant prioritaire.

GP : Post Mortem : Le post-mortem est initié par Elyas ELAZIZ qui fait part aux membres de l'équipe projet de son ressenti. En effet, il affirme avoir apprécié l'efficacité de la conception de la répartition des tâches du solveur.

C'est ensuite au tour de Paul SANCHEZ de s'exprimer. Il estime avoir apprécié aussi l'efficacité de la conception et de la répartition des tâches. Néanmoins, celui-ci soutient s'être parfois senti un peu perdu dans la conception du solveur du fait principalement d'un manque de temps.

Il vient à présent au tour de Damien SIMON de prendre la parole. Il exprime avoir apprécié l'implication, la rapidité d'exécution, et le travail fourni par chacun des membres du projet. Cependant, celui-ci souligne que du fait de sa maîtrise de la conception et au vu des délais impartis pour la réalisation du solveur, il a eu le sentiment d'imposer sa vision et sa conception au reste des membres de l'équipe projet, et aurait plutôt apprécié concevoir un tel solveur à partir des idées de tout un chacun.

Enfin, vient le tour de Quentin LENFANT. D'un point de vue personnel, il s'affirme satisfait du travail réalisé, de la gestion de projet, et notamment, du déblaiement en avance du problème de réalisation du solveur. Toutefois, celui-ci soutient s'être également senti un peu perdu au commencement de la partie solveur du projet. Il mentionne aussi la problématique de temps donné pour réaliser le solveur, ainsi que le temps qu'il a fallu pour monter et établir le dossier de conception de celui-ci.

PROCHAINE RÉUNION :

- Pas de réunion, fin du projet
- *TO-DO list* :

Damien SIMON	- Dossier de conception
	- Performances solveur (<i>rapport</i>)
	- Spécification algébrique arbre de recherche
	- Tests & performance table d’occurence (<i>rapport</i>)
	- Rédaction de la partie implantation de la table d’occurence dans le rapport
Quentin LENFANT	- Tests & performance arbre de recherche (<i>rapport</i>)
	- Rédaction de la partie implantation de l’arbre dans le rapport
Paul SANCHEZ	- Spécification algébrique du dictionnaire
	- tests & performance dictionnaire (<i>rapport</i>)
	- Partie bilan personnel dans le rapport à initialiser
Elyas ELAZIZ	- Rédaction de la partie fonctionnement général du solveur dans le rapport
	- Rédaction de la partie web dans le rapport
	- Initialiser l’implémentation de la structure du solveur et des stratégies dans le rapport.

Chapitre 7

Bilan du projet

7.1 Bilan global

7.1.1 Comparaison travail attendu/effectué

	Travail attendu	Travail réalisé ?
Application WEB	<ul style="list-style-type: none">- Dossier de conception- Page des paramètres- Page du jeu- Pop-up "Règles"- Pop-up "Statistiques"- Pop-up "Paramètres" avec mode daltonien- Pop-up "Fin de partie"- Clavier intégré- Base de données- Fichiers dictionnaires- Formulaire création de parties- Algorithmes d'exécution de la partie- Algorithmes de statistiques- Requirements- Fichier exécutable	<ul style="list-style-type: none">- Oui- Oui- Oui- Oui- Oui- Non fait- Oui- Oui- Oui- Oui- Oui- Oui- Oui- Oui- Oui
Solveur	<ul style="list-style-type: none">- Dossier de conception- Fichiers du dictionnaire de mots- Structure de "dictionnaire" (liste)- Structure d'arbre- Structure de la table d'occurrence- Algorithme principal- Calcul des openers (Premiers mots proposés par le solveur)- Makefile- Fichier exécutable	<ul style="list-style-type: none">-- Oui- Oui- Oui- Oui- Oui- Oui- Oui- Oui

7.1.2 Points positifs, Points négatifs

	Points positifs	Points négatifs
Gestion de Projet	- Nomination d'un secrétaire, ce qui a allégé le travail du chef de projet en ce qui concerne la gestion de projet	- Réunions parfois trop longues
Application WEB	- Implémentation effectuée rapidement	- Paramétrage général du site (mode daltonien,...) non implémenté - Précipitation dans la conception de l'application au départ, même si cela a été vite résolu
Solveur	- Clarté de la ligne directrice établie lors de la conception - Solveur performant	- Répartition du travail déséquilibrée entre Damien et le reste de l'équipe - Rédaction du dossier de conception réalisée tardivement et difficilement

7.1.3 Volume horaire

	Quentin LENFANT	Damien SIMON	Paul SANCHEZ	Elyas ELAZIZ
ÉTAT DE L'ART	5h	2h	4h	4h30
CONCEPTION ET PROGRAMMATION				
Application WEB	35h	30h	28h	27h30
Solveur	10h	40h	16h	14h
GESTION DE PROJET				
Participation aux réunions	19h30	19h30	19h30	19h30
Comptes-rendus de réunion	1h		11h	
Documentation du projet	4h	7h	5h	4h30
Rédaction du rapport	13h	6h	7h30	11h
TOTAL	87h30	104h30	91h	81h

7.2 Bilans individuels

7.2.1 Quentin LENFANT

Application WEB

Points positifs	- Bonne implication et bonne réactivité de l'équipe projet, que ce soit dans l'application WEB ou dans le solveur - Progrès notables dans le rôle de chef de projet par rapport au précédent projet - Prise de temps pour bien réfléchir à l'algorithme des fonctions implémentées, ce qui a permis de faciliter la résolution des bugs
Difficultés rencontrées	- Répartition des work packages compliquée car cette partie du projet est dimensionnée pour être faite à 3. Néanmoins, cela a convenu à l'équipe projet
Expérience personnelle	- Apprentissage du JavaScript
Axes d'amélioration	- Répartition des work packages

Solveur

Points positifs	- Temps consacré à la conception ayant permis à toute l'équipe projet de bien cerner le fonctionnement du solveur que nous avons implémenté -
Difficultés rencontrées	- Gestion du temps consacré au projet - Réalisation du dossier de conception
Expérience personnelle	- Mise en application des connaissances acquises dans le cadre des modules de C et de Structures de Données
Axes d'amélioration	- Anticipation de la rédaction du dossier de conception pour faciliter le dossier de conception

7.2.2 Damien SIMON

Application WEB

Points positifs	- Très bonne répartition des tâches - Bonne implication de toute l'équipe - Objectifs du projet bien définis
Difficultés rencontrées	- Gérer les events pour des entrées dynamiques du clavier intégré au site. - Communiquer dynamiquement avec l'API Python
Expérience personnelle	- Apprentissage du JavaScript.
Axes d'amélioration	- Moins se précipiter au début du projet - Prendre le temps nécessaire pour la conception.

Solveur

Points positifs	- L'établissement clair des structures et de leur fonctionnement en amont a permis de fluidifier leur implémentation.
Difficultés rencontrées	- Choisir une stratégie optimale en terme de score et de complexité temporelle.
Expérience personnelle	- Mise en pratique de connaissance en C/SD dans le cadre d'un projet concret d'optimisation.
Axes d'amélioration	- Anticiper un peu plus le dossier de conception afin de pouvoir passer plus rapidement sur la partie code.

7.2.3 Paul SANCHEZ

Application WEB

Points positifs	- Répartition du travail appréciée : satisfait du travail réalisé.
Difficultés rencontrées	- Durée pour réaliser les comptes-rendus en début de projets : difficulté qui s’est largement amoindrie par la suite.
Expérience personnelle	- Apprentissage de JavaScript & Rôle de secrétaire.
Axes d’amélioration	- Durée des réunions.

Solveur

Points positifs	- Répartition du travail appréciée : fier et satisfait du travail réalisé.
Difficultés rencontrées	- Conceptualisation du solveur & Spécification algébrique du dictionnaire (Implémentation en C et non en Java comme dans le cours).
Expérience personnelle	- Rôle de secrétaire & Mise en pratique de connaissances et compétences en C&SD.
Axes d’amélioration	- Durée des réunions, bien qu’amélioration notable par rapport à la première partie du projet.

7.2.4 Elyas ELAZIZ

Application WEB

Points positifs	Pas de difficulté majeure rencontrée, capitalisation sur les acquis du précédent projet.
Difficultés rencontrées	Gestion du temps personnel, notamment pendant les deux semaines de vacances.
Expérience personnelle	Découverte d’un nouveau groupe projet qui a déjà une expérience entre eux et une façon de gérer un projet.
Axes d’amélioration	Commencer plus rapidement la partie code.

Solveur

Points positifs	Définition et répartition des tâches avec la participation et l’acceptation de tous les membres projets, menant à des objectifs clairs et réalistes.
Difficultés rencontrées	Appropriation et compréhension des éléments théoriques de la méthode de fonctionnement et d’optimisation du solveur.
Expérience personnelle	Étude approfondie de l’aspect structure de donnée et mise en pratique des compétences C.
Axes d’amélioration	Étaler le travail à faire sur la semaine plutôt que juste avant les réunions.

Bibliographie

- [1] “Wordle - The New York Times.” <https://www.nytimes.com/games/wordle/index.html>.
- [2] “Wordle.” <https://fr.wikipedia.org/wiki/Wordle>.
- [3] “SUTOM.” <https://sutom.nocle.fr/>.
- [4] “ZUTOM.” <https://zutom.z-lan.fr/>.
- [5] “Quordle.” <https://www.quordle.com/>.
- [6] “Weredle.” <https://weredle.netlify.app/>.
- [7] “Worldle.” <https://worldle.teuteuf.fr/>.
- [8] “Nerdle.” <https://nerdlegame.com/>.
- [9] “dbdiagram.io.” <https://dbdiagram.io/home>.