



Rapport de projet

BECQUET Albert, CHRISTIAEN Adrien, LACROIX Thomas, VERMEL Louis

Mai 2022

Promotion 2024

CS54 : RÉALISATION D'UNE APPLICATION DE JEU WORDLE C/SD : RÉALISATION D'UN SOLVEUR WORDLE



UNIVERSITÉ
DE LORRAINE



Responsables du module : Oliver Festor

20 mars 2022 - 8 juin 2022

Sommaire

1	Introduction	4
2	État de l'art	4
2.1	Web	4
2.1.1	Présentation de WORDLE	4
2.1.2	Présentation textuelle de l'application et des paramètres du jeu	4
2.1.3	Présentation argumentée de la base de données	4
2.1.4	Agencement des pages web	6
2.2	Solveur	9
2.2.1	Solveur fréquentiel	9
2.2.2	Solveur entropie	10
2.2.3	Structures de données	10
3	Gestion de Projet	11
3.1	Equipe projet	11
3.2	Organisation de l'équipe projet	11
3.2.1	Organisation immatérielle	11
3.2.2	Organisation matérielle	11
3.3	Analyse du projet	12
3.3.1	Première matrice SWOT	12
3.3.2	Seconde matrice SWOT	13
3.3.3	Matrice RACI	14
3.3.4	Matrice RACI Application Web	14
3.3.5	Matrice RACI Solveur	14
3.3.6	Diagramme de Gantt	15
3.3.7	Diagramme de Gantt Application Wordle	15
3.3.8	Diagramme de Gantt Solveur Wordle	16
3.3.9	BackLog	17
3.3.10	WBS	19
3.4	Comptes-Rendus	20
3.5	Charte Projet	20
3.5.1	Description du client	20
3.5.2	Problématique	20
3.5.3	Objectifs	20
3.5.4	Rédaction du rapport	20
3.6	Relevé horaire	21
4	Composition du projet	22
5	Première partie : Application Wordle	22
5.1	Présentation et attendus	22
5.2	Mise en oeuvre	22
5.3	Base de données	22
5.3.1	Présentation et attendus	22
5.3.2	Mise en oeuvre	22
5.3.3	Explication	23
5.3.4	Résultats	28
5.3.5	Limites du Modèle	28
5.4	Réalisation du serveur Web	29
5.4.1	Présentation et attendus	29
5.4.2	Mise en oeuvre	29
5.5	Réalisation de l'algorithme de jeu Wordle	29
5.5.1	Présentation et attendus	29
5.5.2	Mise en oeuvre	29

5.5.3	Résultats	30
5.5.4	Bonus : JavaScript	33
6	Deuxième partie : Solveur Wordle	36
6.1	Présentation et attendus	36
6.2	Mise en oeuvre	36
6.2.1	Principe du solveur	36
6.2.2	Structures de données utilisées	36
6.2.3	Fonctions utilisées	37
6.2.4	Analyse de la compléxité	38
6.3	Résultats	38
6.3.1	Modèle fréquentiel	38
6.3.2	Modèle entropique	40
6.4	Limites des modèles	42
6.4.1	Limites du modèle fréquentiel	42
6.4.2	Limites du modèle entropique	42
7	Bilan du projet	43
7.1	Remerciements	43
8	Références	44
9	Annexes	45
9.1	Comptes rendus	45
9.1.1	CR1	45
9.1.2	CR2	47
9.1.3	CR3	48
9.1.4	CR4	50
9.1.5	CR5	51
9.1.6	CR7	53
9.1.7	CR8	55
9.1.8	CR9	57
9.1.9	CR10	59
9.1.10	CR11	61
9.1.11	CR12	63
9.1.12	CR13	64
9.1.13	CR14	65
9.1.14	CR15	67
9.2	Sprint Backlog	69

1 Introduction

2 État de l'art

2.1 Web

2.1.1 Présentation de WORDLE

WORDLE est un jeu où le joueur doit deviner un mot mystère d'une certaine taille (de 5 lettres la plupart du temps) avec un nombre de coups donné (6 généralement). Pour cela, le joueur propose un mot qui existe et de bonne taille qui pourrait être le mot mystère. Puis le jeu lui indique si certaines des lettres de son mot sont dans le mot mystère et si elles sont à la bonne place ou non. Le joueur peut ensuite proposer un autre mot. Le jeu s'arrête quand le joueur trouve le mot mystère ou quand il dépasse le nombre de coups défini.

2.1.2 Présentation textuelle de l'application et des paramètres du jeu

L'application permettra de jouer à WORDLE depuis un navigateur web. Le menu principal permettra au joueur de jouer directement sans se connecter, cependant, une connexion sera nécessaire afin d'accéder aux fonctionnalités avancées. Il sera ainsi possible d'accéder à un mode multijoueur, de visionner ses statistiques ainsi que son historique et ses succès accomplis. Le joueur aura également accès aux paramètres du mode jeu, où il pourra non seulement choisir la taille du mot qu'il devra trouver et le nombre de coups qui lui seront donnés, mais aussi choisir d'autres modes de jeu originaux. Il pourra ainsi jouer à un mode “contre la montre”, où l'objectif serait de trouver le plus de mots en un temps donné, ou encore à un mode survie où le but sera de trouver le plus de mots avec un nombre de “vies” prédéfini.

Nous souhaiterions réaliser une application facile à utiliser et à comprendre, tout en gardant la possibilité pour les joueurs aguerris de renforcer leurs connaissances avec des modes plus difficiles. De plus, afin de motiver les joueurs, notre application proposerait un système d'expérience afin de permettre aux utilisateurs de monter en niveau, avec possiblement à la clé des récompenses permettant une customisation de son profil et un classement.

2.1.3 Présentation argumentée de la base de données

Notre application sera en lien avec une base de données afin d'assurer son bon fonctionnement. Afin de stocker les données de l'utilisateur, nous utiliserons une première table (figure 1) permettant de sauvegarder :

- identifiant, mot de passe et email de récupération de l'utilisateur
- nombre de victoires/défaites afin de lui présenter ses statistiques
- nombre de points d'expérience

UTILISATEUR	
- <u>Id</u>	
- Nom_utilisateur	
- Mot_de_passe	
- Email	
- Nb_victoires	
- Nb_défaites	
- Expérience	

FIGURE 1 – Table utilisateur

Une seconde table (figure 2) permettra de sauvegarder l'ensemble des parties réalisées par les joueurs, leur permettant ainsi d'améliorer leur façon de jouer en adoptant différentes stratégies. Elle se présentera sous cette forme :

- un identifiant unique permettant d'identifier chaque partie

- l'identifiant du joueur ayant réalisé cette dernière
- l'état de la partie, gagnée ou perdue
- le nombre de coups réalisés
- la date de la partie
- le mode de jeu et les paramètres utilisés (figure 4)

HISTORIQUE
<ul style="list-style-type: none"> - <u>Id_partie</u> - Id (joueur) - Etat (booléen) - Nb_coups - Date - Mode_de_jeu

FIGURE 2 – Table historique

Les deux tables suivantes (figure 3) serviront de support pour les quêtes proposées au joueurs, apportant de la motivation à ces derniers. Chaque quête aura :

- un nom
- un objectif
- une valeur en expérience

QUÊTES	QUÊTES_REA
<ul style="list-style-type: none"> - <u>Id_quete</u> - nom - objectif - valeur_xp 	<ul style="list-style-type: none"> - <u>Id_quete</u> - Id (joueur) - Etat (Booléen)

FIGURE 3 – Tables quêtes

La dernière table (figure 4) permettra de stocker les modes de jeux utilisés (contre la montre, classique...) ainsi que les paramètres de longueur de mots et nombre d'essais utilisés lors de la partie.

MODES
<ul style="list-style-type: none"> - <u>Mode_de_jeu</u> - Nb_caracteres - Nb_essais

FIGURE 4 – Table modes de jeu

2.1.4 Agencement des pages web

Nous souhaitons que la page d'accueil soit la page de jeu, afin de simplifier l'expérience pour l'utilisateur. La figure 5 montre le modèle sur lequel nous aimerions construire la page web.

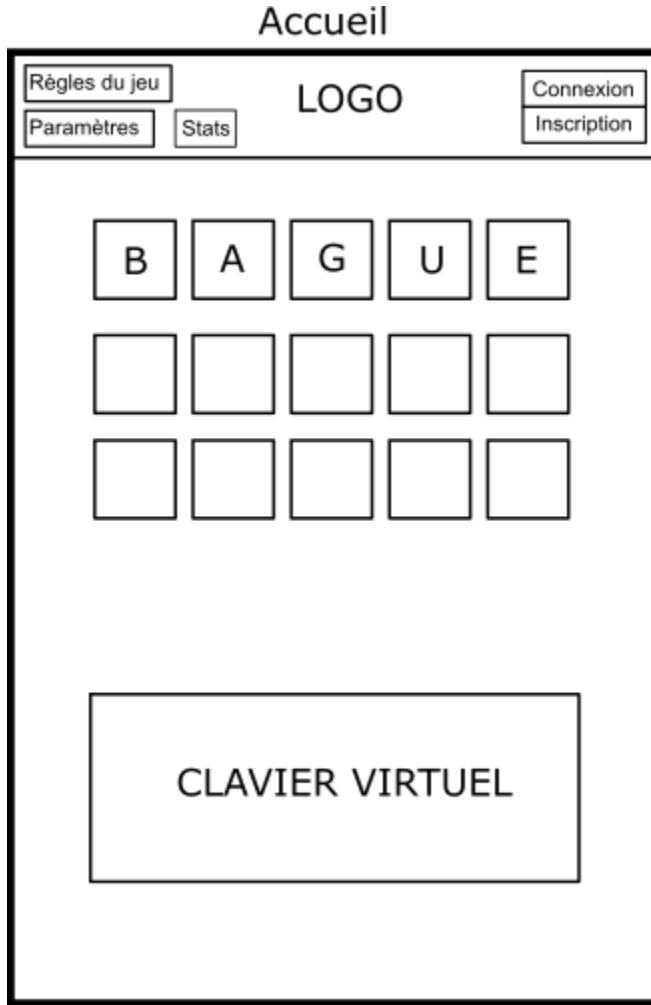


FIGURE 5 – Page d'accueil

La figure 6 montre la page dédiée aux statistiques, où l'accès sera protégé par une authentification de l'utilisateur. Cette page permettra de visualiser ses performances et d'avoir accès aux mots des derniers jours. L'utilisateur pourra également voir dans combien de temps sera déverrouillé le prochain mot.



FIGURE 6 – Page statistiques

La page suivante (figure 7) propose une sélection de paramètres, dont le mode multijoueur que nous souhaiterions développer si nous avons le temps de remplir nos objectifs principaux. L'utilisateur pourra facilement au sein de cette page modifier des paramètres clés afin de personnaliser son expérience de jeu. C'est ici qu'il choisira le nombre de lettres qu'il souhaite deviner, le nombre d'essais qu'il souhaite se voir accorder, le choix du mode jeu (présenté dans les parties 1.2.2 et 1.2.3), etc ...

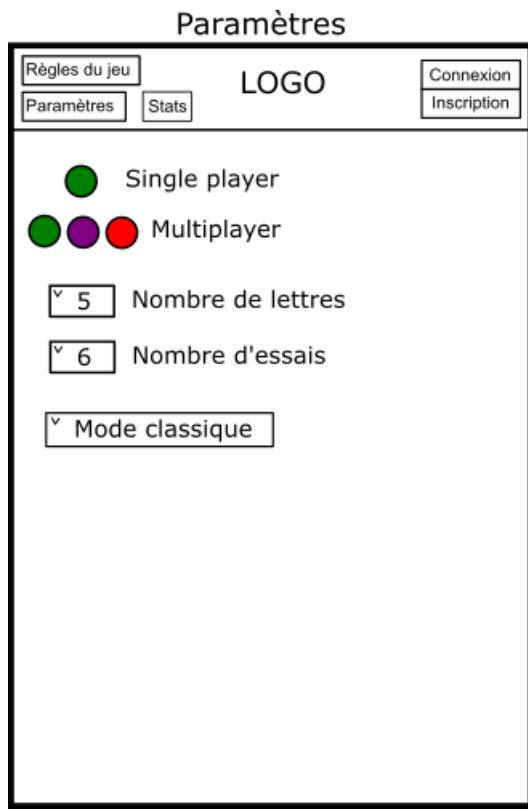


FIGURE 7 – Page paramètres

La dernière page (figure 8) que nous avons choisis de vous présenter est la page d’authentification. Nous mettrons donc en place un système de connexion afin de gérer de manière simple les données des utilisateurs et protéger l’accès à certaines fonctionnalités.

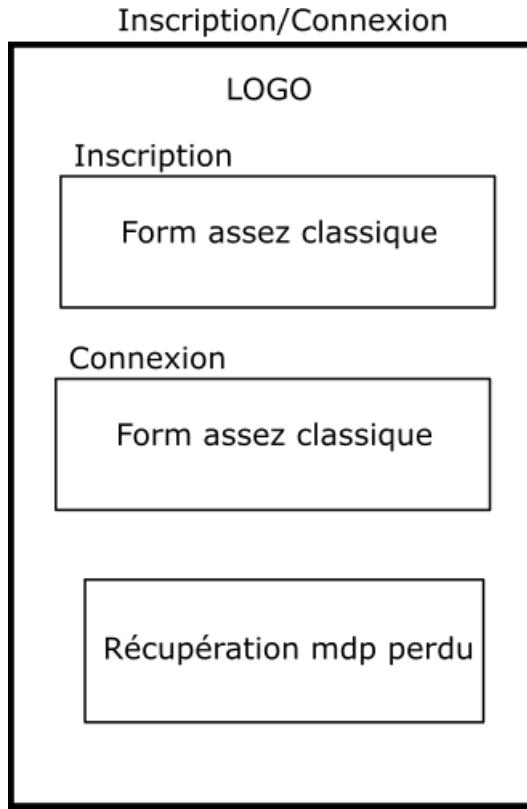


FIGURE 8 – Page authentification

2.2 Solveur

2.2.1 Solveur fréquentiel

La première idée que nous avons eu fut de réaliser un solveur uniquement basé sur la fréquence des lettres des mots dans la liste.

Le fonctionnement est simple :

- on parcourt tout d'abord notre liste de mots possibles afin d'obtenir la fréquence de chaque lettre ;
- un deuxième parcours de la liste permet de calculer ce que nous avons nommé *score fréquence*, calculé de manière très simple ;
- on cherche ensuite le mot ayant le score fréquence le plus élevé afin de choisir le mot qui sera proposé lors de l'essai suivant.

Soit f la fréquence, x une lettre dont on souhaite calculer la fréquence, $n(x)$ le nombre d'apparitions de la lettre x et N le nombre total de lettres, on a :

$$f(x) = \frac{n(x)}{N} \quad (1)$$

Soit S le *score fréquence* et $n(x)$ le nombre d'apparitions d'une lettre, le *score fréquence* est calculé de la manière suivante :

$$S(FRITE) = n(F) + n(R) + n(I) + n(T) + n(E) \quad (2)$$

Le choix du mot se fait alors de la manière suivante :

$$proposition = \{ mot \mid S(mot) = \max(S(mot_1), S(mot_2), \dots, S(mot_n)) \} \quad (3)$$

2.2.2 Solveur entropie

Nous nous sommes ensuite dirigés vers un solveur utilisant la théorie de l'information. Pour cela, nous avons étudié de manière précise les vidéos en annexes du sujet. Voici les informations que nous avons pu en tirer :

- chaque pattern a une probabilité de tomber différente suivant les mots proposés à wordle ;
- il va falloir étudier, pour chaque mot, la probabilité que chaque pattern tombe ;
- globalement, il faut réussir à trouver le mot qui en moyenne va éliminer le plus de mots, c'est à dire celui qui nous rapporte le plus d'information.

Important : Un *pattern* est une combinaison de couleurs, ou informatiquement de 0-1-2, permettant de vérifier certaines règles, définies par le jeu, sur les mots de notre liste.

Voici donc les différentes étapes de la réalisation du solveur par entropie :

Soit *20012* un pattern que nous utiliseront pour l'exemple. Calcul de la probabilité qu'un pattern se réalise :

$$p(20012) = \frac{nbMatches}{nbMots} \quad (4)$$

Calcul de l'information en bits :

$$I(20012) = -\log_2(p(20012)) \quad (5)$$

Calcul de l'entropie d'un mot :

$$E(mot) = \frac{\sum_{pattern} I(pattern)}{nbMots} \quad (6)$$

Le choix du mot se fait alors de la manière suivante :

$$proposition = \{ mot \mid E(mot) = \max(E(mot_1), E(mot_2), \dots, E(mot_n)) \} \quad (7)$$

2.2.3 Structures de données

Le choix de la structure de données a fait l'objet de nombreuses discussions au sein de notre groupe. La structure finale retenue fut la liste chaînée, de cette forme :

Pour le solveur fréquentiel :

Element	Structure
<i>element_{mot}</i>	<i>double score</i> fréquence <i>char mot[20]</i> <i>element_{mot}</i> *suivant
<i>liste_{mot}</i>	<i>element_{mot}</i> *premier

Pour le solveur par entropie :

Element	Structure
<i>element_{info}</i>	<i>int match</i> <i>double bits</i> <i>char resultat[20];</i> <i>element_{info}</i> *suivant
<i>liste_{info}</i>	<i>element_{info}</i> *premier

Nous avons également débattu sur l'utilisation d'arbres pour le stockage des mots, mais lorsque nous avons commencés à coder nous avions encore trop peu de notions pratiques sur l'utilisation de telles structures.

3 Gestión de Projet

3.1 Équipe projet

L'équipe projet se compose de 4 membres :

BECQUET Albert
CHRISTIAEN Adrien
LACROIX Thomas
VERMEL Louis

3.2 Organisation de l'équipe projet

3.2.1 Organisation immatérielle

L'équipe projet s'est réunie en présentiel à l'école ou par vision-conférence durant les vacances de noël. La communication s'est faite en parallèle à travers un système de messagerie instantanée dans lequel un groupe a été créé. Ce groupe était constitué de l'ensemble des membres de l'équipe projet.

3.2.2 Organisation matérielle

Afin de remplir la mission qui nous avait été confiée, nous avons utilisé différents outils mis à notre disposition :

Ordinateurs personnels : Afin de travailler sur le projet de chez nous.

Messenger : Afin de communiquer et transmettre les informations rapidement.

Visual Studio Code, flask, Python : Afin de développer le code du projet.

OverLeaf de Télécom Nancy : Afin de réaliser le rapport du projet ainsi que différents documents de Gestion de Projet.

Gitlab de Télécom Nancy : Afin de partager les fichiers du code et le maintenir à jour pour l'ensemble de l'équipe projet.

Excel : afin de tracer le diagramme de Gantt.

License Microsoft Office fournie par l'Université de Lorraine : Afin de réaliser une prise de notes lors des réunions.

3.3 Analyse du projet

3.3.1 Première matrice SWOT

Cette matrice SWOT concerne essentiellement la première partie du projet, le site du wordle.

Forces <ul style="list-style-type: none">- Connaissances et expériences en création d'application web (PPII-1).- Équipe sérieuse et motivée qui souhaite produire un projet plus réussi que le PPII-1.	Faiblesses <ul style="list-style-type: none">- Premier travail entre les différents membres du groupe.
Opportunités <ul style="list-style-type: none">- Le jeu wordle connaît un effet de mode en 2022 avec le jeu sutom où les joueurs partagent leur résultat sur les réseaux sociaux.- Vacances scolaires pendant les semaines de travail.	Menaces <ul style="list-style-type: none">- Délais assez restreint pour rendre l'application wordle.

FIGURE 9 – Matrice SWOT Application Wordle

Analyse de la matrice SWOT :

Pour gérer les menaces :

- Les différents langages à utilisés sont bien connu des membres du projet
- Ce projet ressemble dans la forme au projet de 1er semestre, on peut donc aisément s'en inspirer pour gagner du temps.

Pour gérer les faiblesses :

- La cohésion du groupe se fait facilement à ce niveau là
- On en tire un avantage en se devant d'être régulier pour faire avancer le travail

Pour profiter de nos opportunités :

- Le jeu Wordle est un jeu auquel on a pu jouer de nombreuses fois et de nombreux tuto explicatif sont disponible sur internet
- Les vacances scolaires permettent d'accélérer le travail

Pour utiliser nos forces :

- Le projet du premier semestre sert de point de départ
- Le groupe possède une bonne connaissance du langage python et SQL, l'apprentissage en CSS et HTML n'est pas difficile.
- L'équipe est motivé pour valider son projet.

3.3.2 Seconde matrice SWOT

Cette matrice SWOT concerne essentiellement la seconde partie du projet, le solveur.

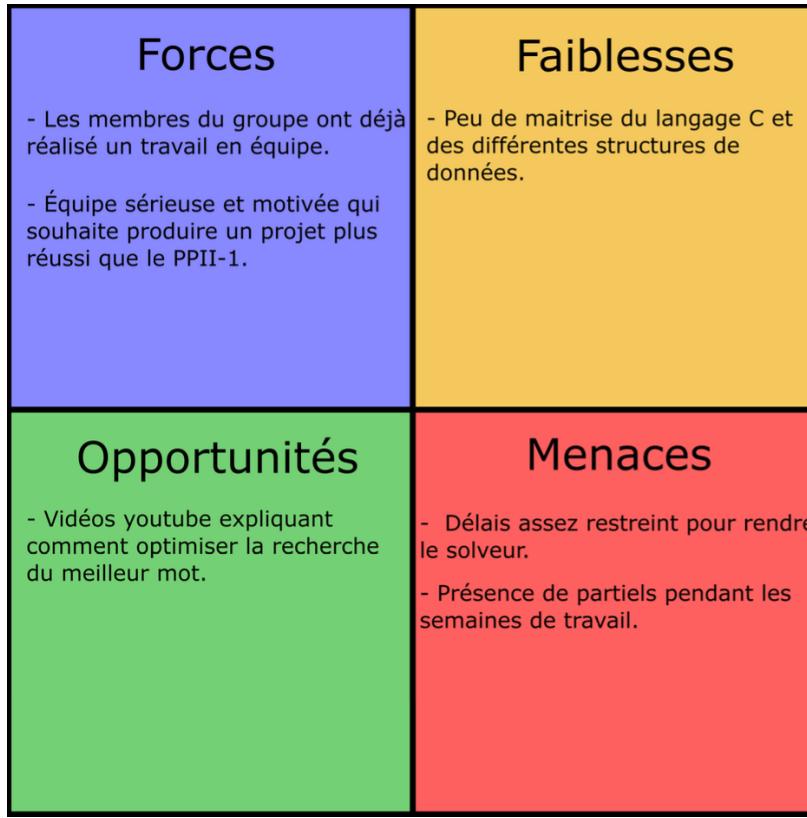


FIGURE 10 – Matrice SWOT Solveur

Analyse de la matrice SWOT :

Pour gérer les menaces :

- L'entraide de travail dans les partiels comme dans le projet permet d'arriver à avoir un résultat sérieux
- Les vidéos présentées dans l'énoncé du projet permettent de gagner du temps dans la compréhension des différents attendus

Pour gérer les faiblesses :

- Les différents TP réalisés permettent de mieux comprendre les structures de données et le code C

Pour profiter de nos opportunités :

- La vidéo youtube donne un code python dont on peut s'inspirer pour tirer un code en C

Pour utiliser nos forces :

- C'est le 3ème travail d'équipe réalisé au total ainsi que le 2ème avec ce même groupe
- La ligne de conduite reste la même : l'équipe est motivée pour valider son projet.

3.3.3 Matrice RACI

3.3.4 Matrice RACI Application Web

Cette matrice SWOT concerne essentiellement la première partie du projet, le site du wordle.

Domaine des tâches	Tâches	Sous-tâches	Sous-sous-tâches	Adrien CHRISTIAËN	Albert BECQUET	Louis VERMEL	Thomas LACROIX
Web	Organisation Page Web	Accueil	HTML/CSS Dynamisme Base de Données Intégration du jeu Implémentation Java Script	R R A	I I I I	I R A R A	I R A I
		Statistiques	HTML/CSS Dynamisme Base de Données	I	I C	R A	I
		Paramètres	HTML/CSS Dynamisme Base de Données	I	I	I R A	I
		Inscription/Connexion	HTML/CSS Dynamisme Base de Données	I	R A R A	I R A	I
		Profil	HTML/CSS Dynamisme Base de Données	I	R A R A	I R	I
		Succès	HTML/CSS Dynamisme Base de Données	I	R A R A	I	I
		Comment jouer	HTML/CSS Schéma des routes Proposition des tables	I R	I R A	I R I R	I R I R
		Base de données	Création/Modification base de données Passage/Vérification des tables en 3NF Modification tables pour les différents modes de jeux Mise à jour schéma 3NF Schéma E/A Explication table Parties	R A R A I R A R A	I I I I I	R R R R C	I I I I I
	Algorithmique	Trier le dictionnaire ODS6		I	I R A	I I R A	I
		Première approche		R	I	I R A	I R A
		Amélioration de l'algorithme				I R A	I
		Système d'expérience		I	R A	I R A	I
	Gestion de projet	Ajout de modes de jeux		I	I	I R A	I
		Testing des principales fonctions		I	R A	I	I
		WBS		I	R A	I R A	I
		Backlog		C	I	I R A	I R A
		Diagramme de Gantt		R A	I	I R A	I
		Matrice SWOT			I R	I R	I
		Document de conception	Réalisation Rédaction LaTeX Compte-Rendu-1	R R A	I R A	I R I R A	I R I R A
		Compte-rendu	Compte-Rendu-2 Compte-Rendu-3 Compte-Rendu-4 Compte-Rendu-5 Compte-Rendu-6 Compte-Rendu-7 Compte-Rendu-8	I I I I R A I I	R A R A R A R A I R A I R A I R A	I R A I R A I R A I R A I R A I R A I R A	I I I I I R A I R A I R A
		Rapport final du projet		R A	I R A	I I R A	I I R A
		Matrice RACI			I R A	I I R A	I I R A

FIGURE 11 – Matrice RACI de l'application WORDLE

3.3.5 Matrice RACI Solveur

Cette matrice SWOT concerne essentiellement la seconde partie du projet, le solveur.

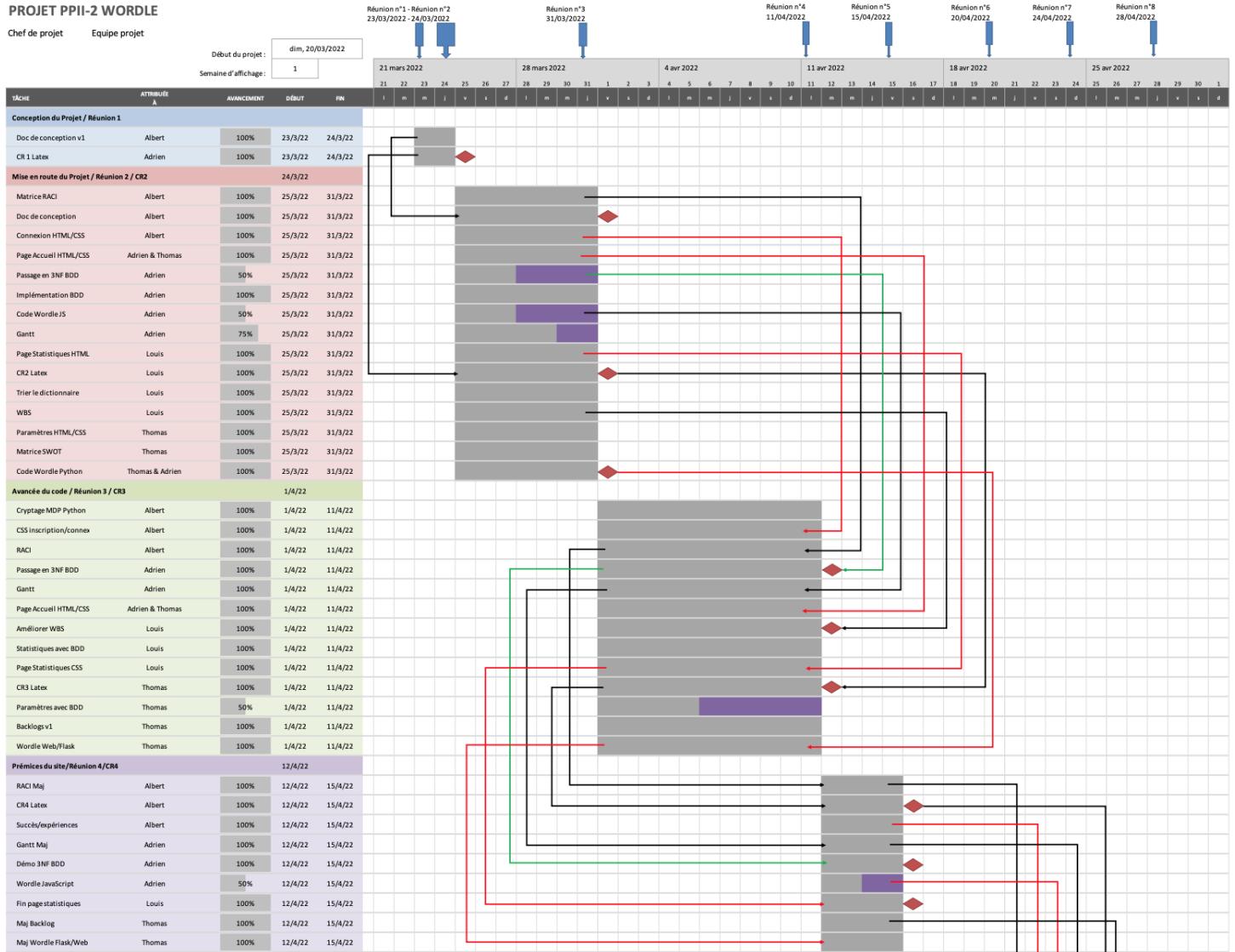
Domaine des tâches	Tâches	Sous-tâches	Sous-sous-tâches	Adrien CHRISTIAËN	Albert BECQUET	Louis VERMEL	Thomas LACROIX
Solveur	Conception	Recherches sur les solveurs		R	I R	I R	I R
		Développement des fonctions		R	C	C	C
		Testing des principales fonctions		R	C	C	C
		Analysé complexité et mémoire		R	I R	I R	I R
	Algorithmique	WBS		I	I R A	I R A	I
		Backlog		C	I	I R A	I R A
		Diagramme de Gantt		R A	I	I R A	I
		Matrice SWOT			I R	I R	I
	Gestion de projet	Document de conception	Réalisation Rédaction LaTeX	R	I R A	I R A	I R A
		Comptes-Rendus	Document récapitulatif fonctions solveur	I	I R A	I R A	I R A
		Compte-Rendu-9		I	I R A	I R A	I R A
		Compte-Rendu-10		R A	I	I R A	I R A
		Compte-Rendu-11		I	I	I R A	I R A
		Compte-Rendu-12		I	R A	I R A	I R A
		Compte-Rendu-13		I	R A	I R A	I R A
		Compte-Rendu-14		R A	I	I R A	I R A
		Compte-Rendu-15		I	I R A	I R A	I R A
		Rapport final du projet		R A	I R A	I R A	I R A
		Matrice RACI			I R A	I I R A	I I R A

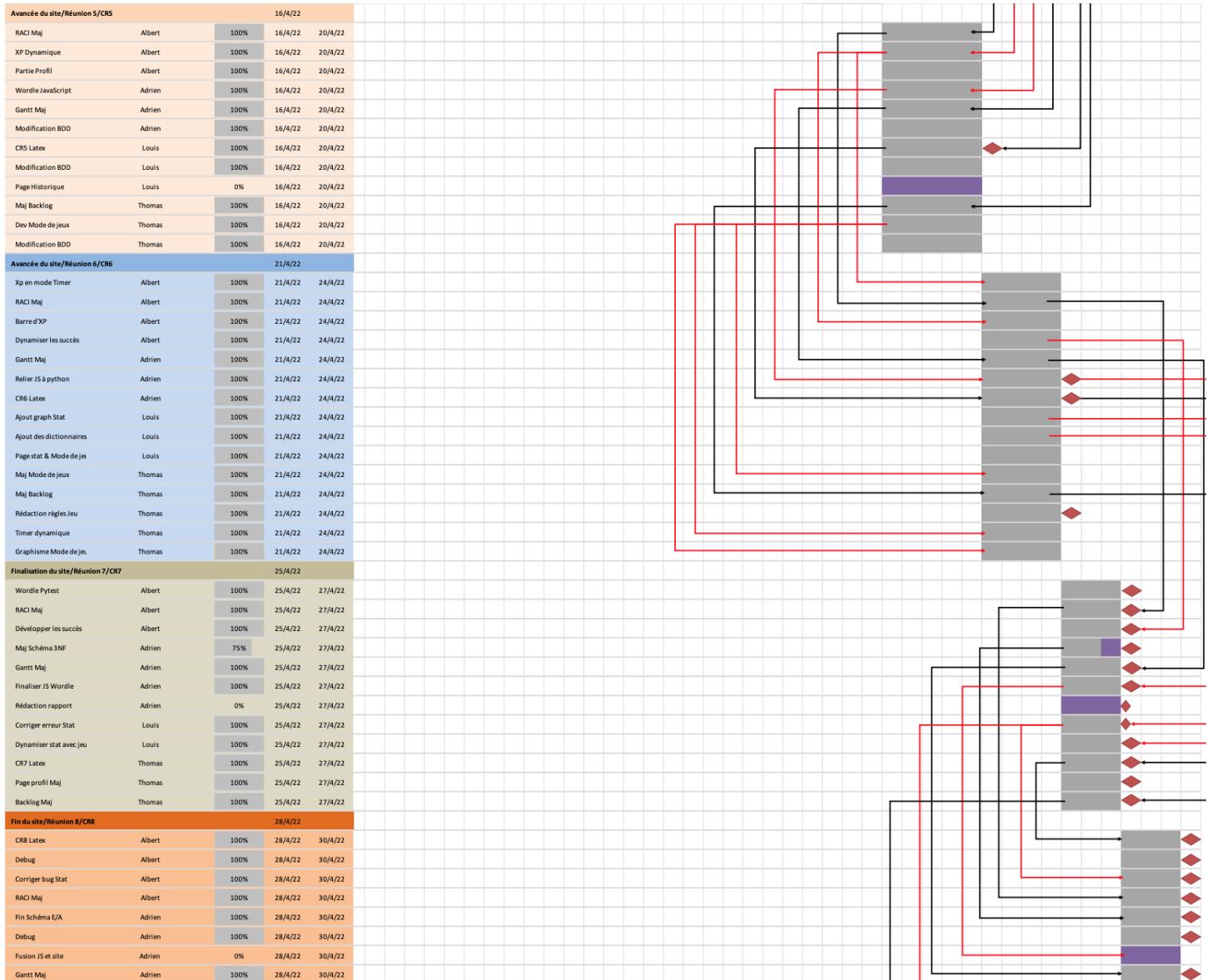
FIGURE 12 – Matrice RACI du Solveur

3.3.6 Diagramme de Gantt

3.3.7 Diagramme de Gantt Application Wordle

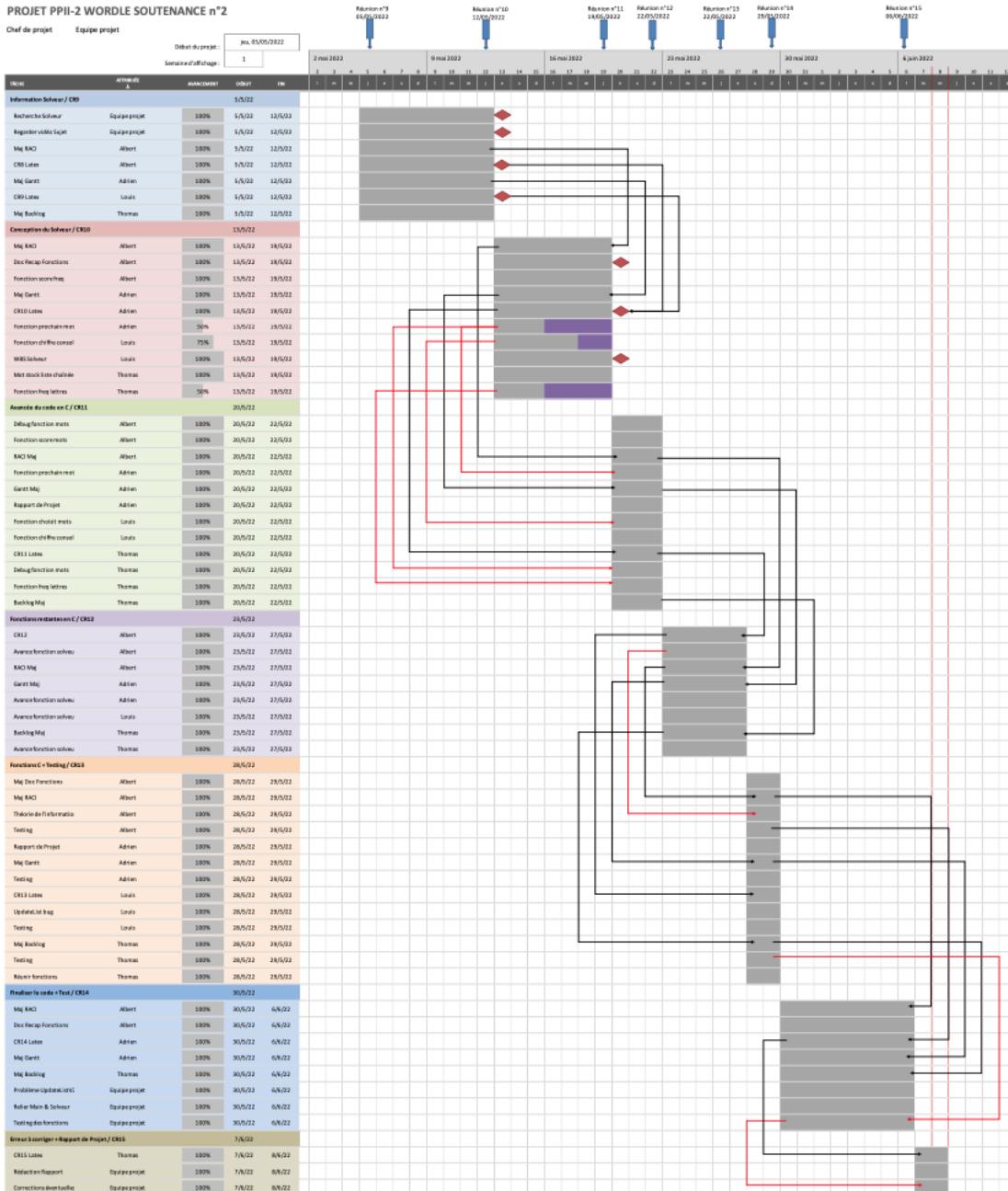
Le diagramme de GANTT ci-dessous représente la version finale du projet Wordle Application.





		Semaine d'affichage :	1	21 mars 2022							28 mars 2022							4 avr 2022							11 avr 2022							18 avr 2022							25 avr 2022								
TÂCHE	ATTRIBUÉE À	AVANCEMENT	DÉBUT	FIN	1	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d	l	m	m	j	v	s	d								
Debug	Louis	100%	28/4/22	30/4/22																																											
Corriger bug Stat & Hist	Louis	100%	28/4/22	30/4/22																																											
Backlog Maj	Thomas	100%	28/4/22	30/4/22																																											
Debug	Thomas	100%	28/4/22	30/4/22																																											
Doc table Partie	Thomas	100%	28/4/22	30/4/22																																											

3.3.8 Diagramme de Gantt Solveur Wordle



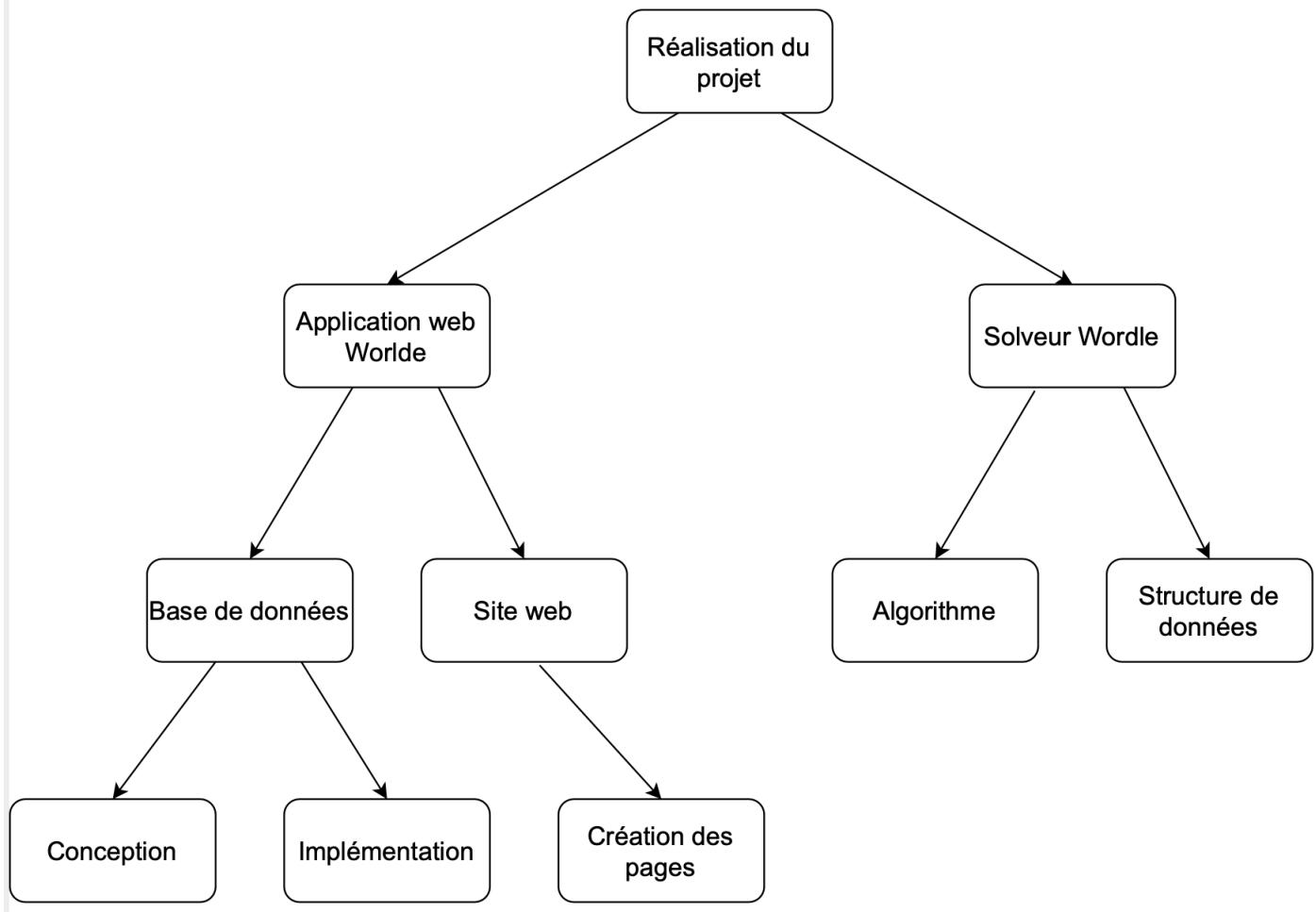
3.3.9 BackLog

On détail ci-après les différentes tâches générales réalisées par l'équipe projet au cours des deux parties du projet. Les Sprint du Backlog seront mis en annexe pour éviter de perdre de la place dans le compte-rendu.

Groupe	Nom	Priorité	Sprints
1A	Page d'accueil/jeu	100	1, 2, 3, 6
1B	Page des paramètres	90	1,2
1C	Page des statistiques	90	1, 2, 3, 5, 6
1D	Page de connexion	90	1, 2, 4, 6
1E	Utilisation de Javascript	70	3, 4, 5, 6
1F	Base de données	100	1, 2
2A	Modes de jeux	80	4, 5
2B	Succès	70	3, 5, 6
2C	Expérience	70	3, 4, 5
2D	Historique	80	4
2E	Base de données	100	3, 4, 5
3A	Algorithme solveur	100	6,7
3B	Résolution fréquentielle	90	6,7
3C	Résolution entropique	80	8
3D	Tests du solveur	100	10

FIGURE 13 – Product Backlog

3.3.10 WBS



Le projet était dès le début scindé en deux parties : la partie application web et la partie solveur. La première partie consistait à implementer une application web qui permettrait de jouer au jeu Wordle en, nous avons découpé cette partie en deux sous partie, la partie web, composer de html et de css. Et en une partie base de donnée, pour gérer les données des différents utilisateurs. La seconde partie consistait à coder en C un solveur, qui pourrait jouer au Wordle et gagner les parties de jeu le plus rapidement possible. Nous avons d'un coté due créer un algorithme pour jouer, mais nous savions aussi due réfléchir à comment stocker et gérer les différentes données nécessaires à l'algorithme. Bien sur nous avons ensuite découpé ces parties en sous-tâches pour permettre à l'ensemble du groupe de toucher à chaque partie du projet.

3.4 Comptes-Rendus

Les comptes-rendus seront mis en annexe afin d'économiser de la place.

3.5 Charte Projet

Présentation du groupe, l'équipe est composée de quatre élèves de première année du cursus ingénieur de Télécom Nancy.

- BECQUET Albert : Coden Gestion de Projet
- CHRISTIAEN Adrien : Code, Gestion de projet
- LACROIX Thomas : Code, Gestion de Projet
- VERMEL Louis : Code, Gestion de Projet

3.5.1 Description du client

Le projet est effectué à la demande de deux membres de l'équipe éducative de Télécom Nancy.
M. Olivier Festor, Professeur en informatique
M. Oster Gerald, Professeur en informatique

3.5.2 Problématique

Problématique principale et prioritaire :

- acquérir de nouvelles connaissances et compétences en algorithmique, en base de données, en web et en gestion de projets
- réaliser une base de données
- réaliser un ensemble de fonctions python
- réaliser un ensemble de fonctions en C
- réaliser une structure de données
- réaliser un code en html et css pour la tenue du site web
- rédiger un rapport en LaTeX synthétisant le travail

3.5.3 Objectifs

- Exploitation correcte des outils de gestion de projet (Gantt, Charte de projet, RACI, SWOT, WBS, BACKLOG)
- Maîtriser les outils mis à notre disposition (gitlab, sharelatex, Python, bibliothèques données en support, flask, Visual Studio Code)
- Maîtriser la prise en main de flask et réaliser un code html, css, python, C et SD viable
- Mettre en œuvre les principes de la programmation dynamique

3.5.4 Rédaction du rapport

Le rapport a été fait au fur et à mesure de l'avancée du projet mais la grande partie a été finalisée lors des derniers jours du projet. Chaque membre s'est occupé des parties qu'il a traitées lors de la réalisation du code et chaque membre a relu l'ensemble du rapport.

3.6 Relevé horaire

Catégorie	Becquet Albert	Vermel Louis	Lacroix Thomas	Christiaen Adrien
Etat de l'art Wordle	3h	3h	3h	3h
HTML/CSS	20h	20h	30h	5h
Python/Flask/BDD	25h	15h	25h	15h
Javascript	0h	0h	5h	20h
Etat de l'art SOLVEUR	6h	4h	4h	4h
C/SD	35h	25h	30h	10h
Gestion de projet	30h	25h	35h	40h
Total	119h	92h	132h	97h

4 Composition du projet

Rappel de l'objectif de notre projet : La première partie consiste en la réalisation d'une application web jouable du jeu WORDLE. La première partie du projet se compose de 3 parties distinctes :

1. Réalisation de la base de données
2. Réalisation des différentes pythons
3. Réalisation des tests de fonctions

La seconde partie consiste en la réalisation d'un solveur en C/SD du jeu WORDLE. La seconde partie du projet se compose de 3 parties distinctes :

1. Réalisation de la structure de données
2. Création du serveur web
3. Réalisation des algorithmes de traitement

5 Première partie : Application Wordle

5.1 Présentation et attendus

L'objectif de cette partie est de réaliser un site web permettant de jouer au jeu Wordle. L'ensemble du site devra être fait en python, css, html et potentiellement javascript.

5.2 Mise en oeuvre

On sépare le travail en 3 points clés :

1. Réalisation de la base de données
2. Réalisation du serveur web
3. Réalisation de l'algorithme de jeu Wordle

5.3 Base de données

5.3.1 Présentation et attendus

Le but de cette première partie est de réaliser la base de données de l'application Wordle. L'objectif final étant de pouvoir jouer au jeu Wordle avec un ensemble de mots à disposition et en ayant un système de connexion propre pour chaque utilisateur. Dans notre projet, nous avons pris la liberté de rajouter des modes des jeux ainsi qu'un système de quête.

5.3.2 Mise en oeuvre

On décante alors la base de données en plusieurs tables :

1. Une table pour les Parties
2. Une table pour les Utilisateurs
3. Une table pour l'Historique des parties
4. Une table pour les Quêtes de l'utilisateur
5. Une table pour quantifier les Quêtes réalisé par l'utilisateur

Le schéma relationnel de la base de données est de la forme suivante :

STATISTIQUES (Id_partie, Id, Nb_essais)

PARTIES (Nb_essais, Nb_caracteres, mot_cherche, mots_proposes, etat_lettres, Mode_de_jeu, Survie_vie, Survie_score, Big50_nb_essais, Big50_score, Clm_depart, Clm_score)

UTILISATEUR (Id, Nom_utilisateur, Mot_de_passe, Email, Nb_victoires_classique, Nb_defaites_classique, Experience, Photo)

HISTORIQUE (Id_partie, Identifiant, Estat, Score, Date, Mode_de_jeu, Mot)

CONNEXION (Id, Id_partie, Id_quete, Id_quete_rea)

ETAT (Id_quete, Id_quete_commensale)

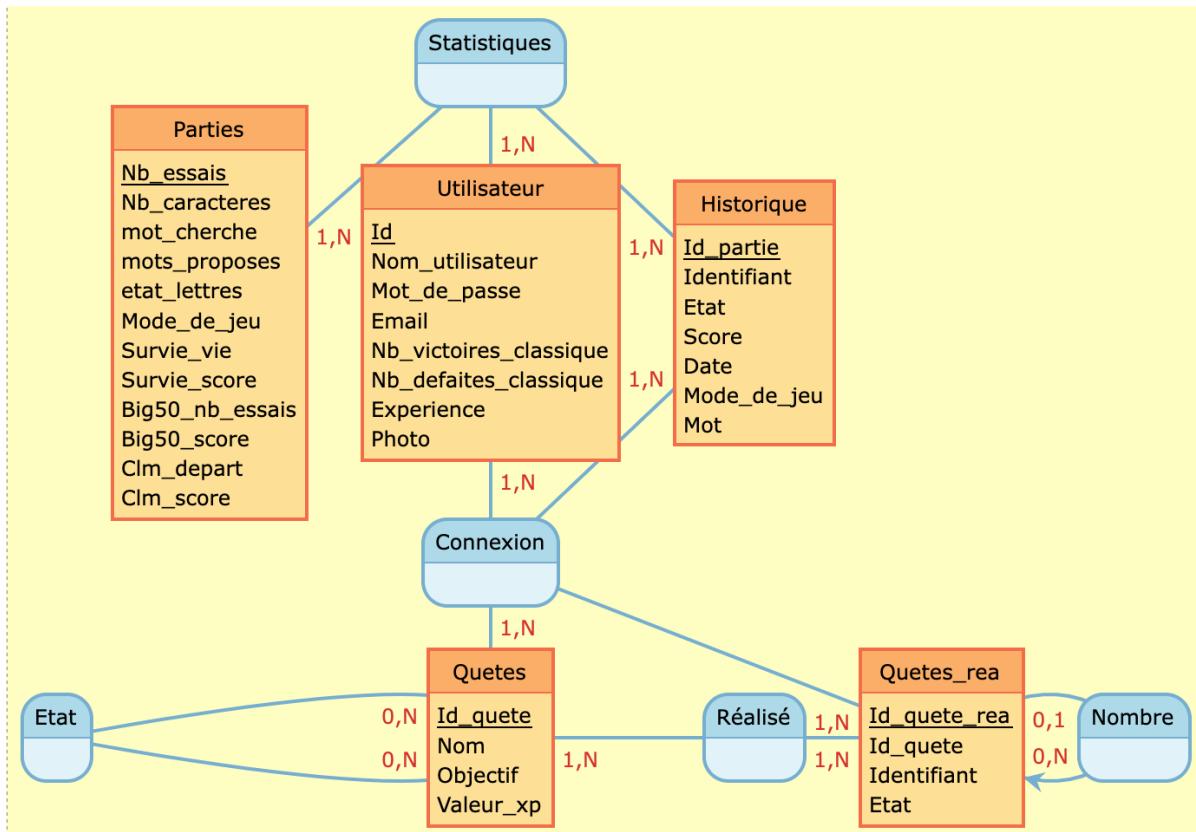
QUETES (Id_quete, Nom, Objectif, Valeur_xp)

RÉALISÉ (Id_quete, Id_quete_rea)

QUETES_REA (Id_quete_rea, Id_quete, Identifiant, Estat, Id_quete_rea_mère)

Comme vue lors de l'état de l'art : La connection de l'utilisateur est géré par la table Utilisateur qui gère également la partie Statistiques lié à l'utilisateur. Un utilisateur non connecté peut avoir accès aux différents modes de jeu mais aucune partie ne sera enregistrées. Un utilisateur connecté aura chacune de ses parties enregistrées et pourra accéder aux statistiques et à l'historique par mode de jeux comme sans mode de jeux. Un système de Quêtes avec expérience a été ajouté dans la table Quêtes, ce qui permet de déduire le nombre de quêtes réalisées par le joueur.

On obtient le modèle entité-association simpliste suivant :



5.3.3 Explication

Explication de la table « Modes » à renommer en « Partie » ou « Jeu »

Lorsque que l'on lance le serveur, le mode de jeu par défaut est classique et voici un exemple de partie.

Quand on initialise le fichier init_database, la table est de la forme :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
6	7	"	"	"	classique	3	0	50	0	"	0

(Nb_essais est la clé primaire car toujours différent à chaque ligne)

Si le mot que l'on cherche n'est pas défini (""), on supprime la première ligne de la table et on ajoute le mot à chercher :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
6	7	'bonjour'	"	"	classique	3	0	50	0	"	0

Le joueur propose un mot, on ajoute donc une ligne à la table :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
6	7	'bonjour'	"	"	'classique'	3	0	50	0	"	0
5	7	'bonjour'	'adoubier'	'00011102'	'classique'	3	0	50	0	"	0

FIGURE 14 – Page d'accueil

Le joueur propose un mot, on ajoute donc une ligne à la table :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
6	7	'bonjour'	"	"	'classique'	3	0	50	0	"	0
5	7	'bonjour'	'adoubier'	'0011102'	'classique'	3	0	50	0	"	0
4	7	'bonjour'	'blouson'	'2011001'	'classique'	3	0	50	0	"	0

Le joueur propose un mot, on ajoute donc une ligne à la table :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
6	7	'bonjour'	"	"	'classique'	3	0	50	0	"	0
5	7	'bonjour'	'adoubier'	'0011102'	'classique'	3	0	50	0	"	0
4	7	'bonjour'	'blouson'	'2011001'	'classique'	3	0	50	0	"	0
3	7	'bonjour'	'bonjour'	'2222222'	'classique'	3	0	50	0	"	0

Il a donc gagné, on efface la table et on repart au premier tableau.

Si le nb_essais est égale à 0 et que etat_lettres != '2'*nb_caracteres alors il a perdu la partie, on efface la table et on repart au premier tableau.

(Pour tous les autres modes de jeu, le nombre d'essai est fixé à 5 mais le nombre de lettres varie

FIGURE 15 – Page d'accueil

Maintenant un exemple pour le mode de jeu survie :

Le principe de la table est le même, je vais détailler lorsque l'utilisateur trouve le mot ou pas.

S'il trouve le mot :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	CIm_depart	CIm_score
...
3	7	'bonjour'	'bonjour'	'2222222'	'survie'	3	3	50	0	"	0

(Survie_score += nb_essais)

On supprime alors la table actuelle et on repart avec la même table de départ en actualisant le score :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	CIm_depart	CIm_score
5	8	"	"	"	'survie'	3	3	50	0	"	0

S'il ne trouve pas le mot, alors on initialise de cette façon :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	CIm_depart	CIm_score
5	6	"	"	"	'survie'	2	0	50	0	"	0

FIGURE 16 – Page d'accueil

Maintenant un exemple pour le mode de jeu Big50 :

S'il trouve le mot :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
...
3	7	'bonjour'	'bonjour'	'2222222'	'big50'	3	0	47	1	"	0

(Big50_score correspond au nombre de mots trouvés)

On supprime alors la table actuelle et on repart avec la même table de départ en actualisant le score :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
5	6	"	"	"	'big50'	3	0	47	1	"	0

La partie s'arrête lorsque Big50_nb_essais=0 :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
...
3	7	'pomme'	'poire'	'22002'	'big50'	3	0	0	8	"	0

FIGURE 17 – Page d'accueil

Maintenant un exemple pour le mode de jeu Contre la montre :

Si le temps de départ n'est pas défini (" "), on supprime la première ligne de la table et on ajoute le temps de départ:

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
5	7	'bonjour'	"	"	'clm'	3	0	50	0	'33 :41'	0

'33 :41' signifie par exemple que l'on a commencé la partie à 14h33min41secondes.

S'il trouve le mot :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
...
3	7	'bonjour'	'bonjour'	'2222222'	'clm'	3	0	50	0	'33 :41'	1

(Clm_score correspond au nombre de mots trouvés)

On supprime alors la table actuelle et on repart avec une table de départ en actualisant le score :

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
5	6	"	"	"	'clm'	3	0	50	0	'33 :41'	1

FIGURE 18 – Page d'accueil

La partie se termine lorsque temps_actuel - temp_depart = 5min (temps_actuel calculer chaque passage dans accueil et temp_depart = Clm_depart)

Nb_essais	Nb_caracteres	mot_cherche	mots_proposes	etat_lettres	Mode_de_jeu	Survie_vie	Survie_score	Big50_nb_essais	Big50_score	Clm_depart	Clm_score
...
3	6	'rigole'	'lourds'	'110100'	'clm'	3	0	50	0	'33 :41'	4

FIGURE 19 – Page d'accueil

5.3.4 Résultats

En appliquant les formules de cours vu en base de données, on obtient par exemple les différentes tables suivantes :

SELECT * FROM 'Utilisateur'

0|Adrien|010c577615874c0f63dfc54974f1878838a45c822aa184faf7d27c44|christiaen.adrien@gmail.com|7|3|0|profil3

SELECT * FROM Partie

6|6|BAVURE|||classique|3|0|50|0||0

SELECT * FROM 'Historique'

0|Adrien|Vrai|6|10-03-2022|Classique|muret
1|Adrien|Faux|0|10-03-2022|Classique|table
2|Adrien|Faux|6|21-04-2022|Classique|flute
3|Adrien|Vrai|4|21-04-2022|Classique|carte
4|Adrien|Vrai|4|21-04-2022|Classique|roues
5|Adrien|Faux|6|22-04-2022|Classique|bouee
...

SELECT * FROM 'Quetes'

0|Stratège|Trouver le mot cherché en 4 coups ou moins|1000
1|Mentaliste|Trouver le mot cherché en 3 coups ou moins|5000
2|Bingo|Trouver le mot cherché en 2 coups ou moins|10000
3|Bouum|Trouver le mot cherché du premier coup|100000
4|Rookie|Gagner 5 parties classiques|1000
5|Rookie doué|Gagner 10 parties classiques|2000
...

SELECT * FROM 'Quetes_rea'

0|0|Adrien|0
1|1|Adrien|0
2|2|Adrien|0
3|3|Adrien|0
4|4|Adrien|0
5|5|Adrien|0
...

5.3.5 Limites du Modèle

La base de données est complexe dans la mesure où elle prend en compte également le stock de mots possible. Certaines relations sont en (1,N), elle n'est donc théoriquement pas en 3NF.

5.4 Réalisation du serveur Web

5.4.1 Présentation et attendus

L'objectif de cette partie est d'implémenter un site web permettant de jouer au jeu Wordle. Il y avait donc toute une partie du projet consacrée à la création de page web en html et en css, ainsi que du python(Flask). De plus il fallait lié cette partie à la base de données.

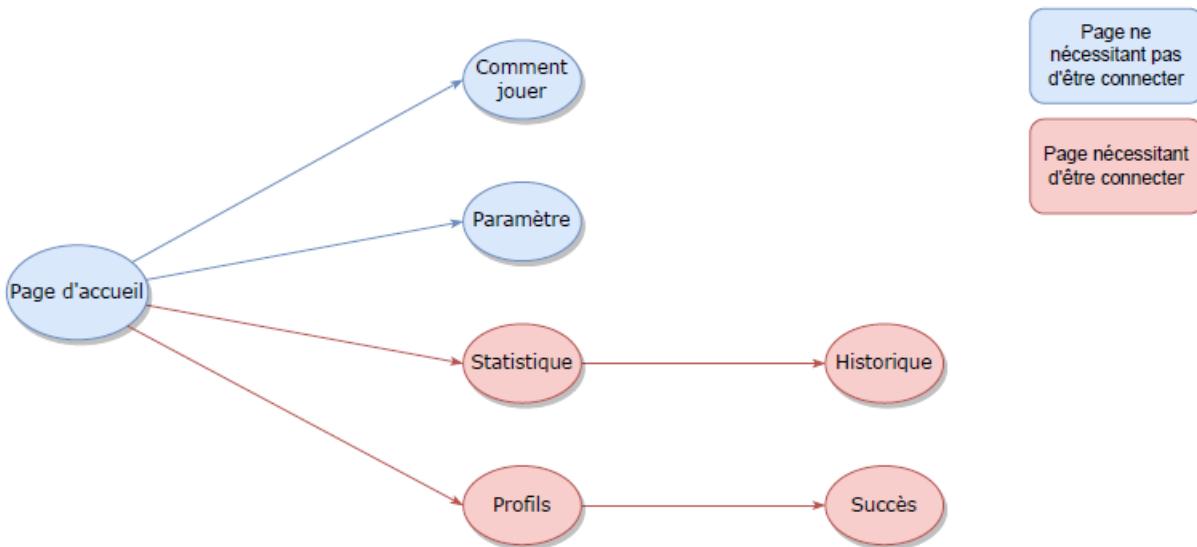
5.4.2 Mise en oeuvre

Nous avions différentes pages web à implémenter :

1. Une page d'accueil où l'utilisateur pourra jouer sa partie
2. Une page connection/inscription
3. Une page statistique/historique
4. Une page paramètre, où l'utilisateur pourra choisir son mode de jeu et la difficulté.
5. Une page expliquant les règles du Wordle.
6. Une page profils et une page succès.

Les pages statistique, historique, profil et succès nécessitant d'avoir créer un compte et d'être connecter.

Le schéma des routes est le suivant :



5.5 Réalisation de l'algorithme de jeu Wordle

5.5.1 Présentation et attendus

Une fois de le site implémenter, il nous fallait implémenter le jeu Wordle en python, pour ensuite le connecter les deux via Flask.

5.5.2 Mise en oeuvre

Pour commencer, nous allons créer une liste des mots proposés de la forme :

$$[('.' * nbLettres) * nbEssais] \quad (8)$$

Ce sont les mots de cette liste qui sont affichés dans le tableau sur la page accueil. A chaque proposition de mot, on remplace donc la suite de points par le mot proposé.

Dans le même temps, on créer une liste contenant l'état de chaque lettre dans le mot proposé, elle est de la forme :

$$[(0' * nbLettres) * nbEssais] \quad (9)$$

. A chaque proposition de mot, on remplace donc la suite de zéros par l'état des lettres dans le mot à trouver. Pour rappel, une lettre est à l'état 2 si elle est au même endroit que dans le mot à trouver, 1 si elle est à un autre emplacement dans le mot à trouver et 0 si elle ne se trouve pas dans le mot à trouver.

Lors d'une partie, ces deux listes vont se mettre à jour à chaque proposition.

Par exemple

$$[[\dots], [\dots], [\dots], [\dots], [\dots], [\dots]] \quad (10)$$

$$[[00000], [00000], [00000], [00000], [00000], [00000]] \quad (11)$$

devient au bout de 3 essais

$$[[PORES], [TETRA], [TRAME], [\dots], [\dots], [\dots]] \quad (12)$$

$$[[00110], [21011], [22222], [00000], [00000], [00000]] \quad (13)$$

5.5.3 Résultats

Voici une présentation des différentes pages de l'application :

- ## 1. Les pages accessibles sans être connecté

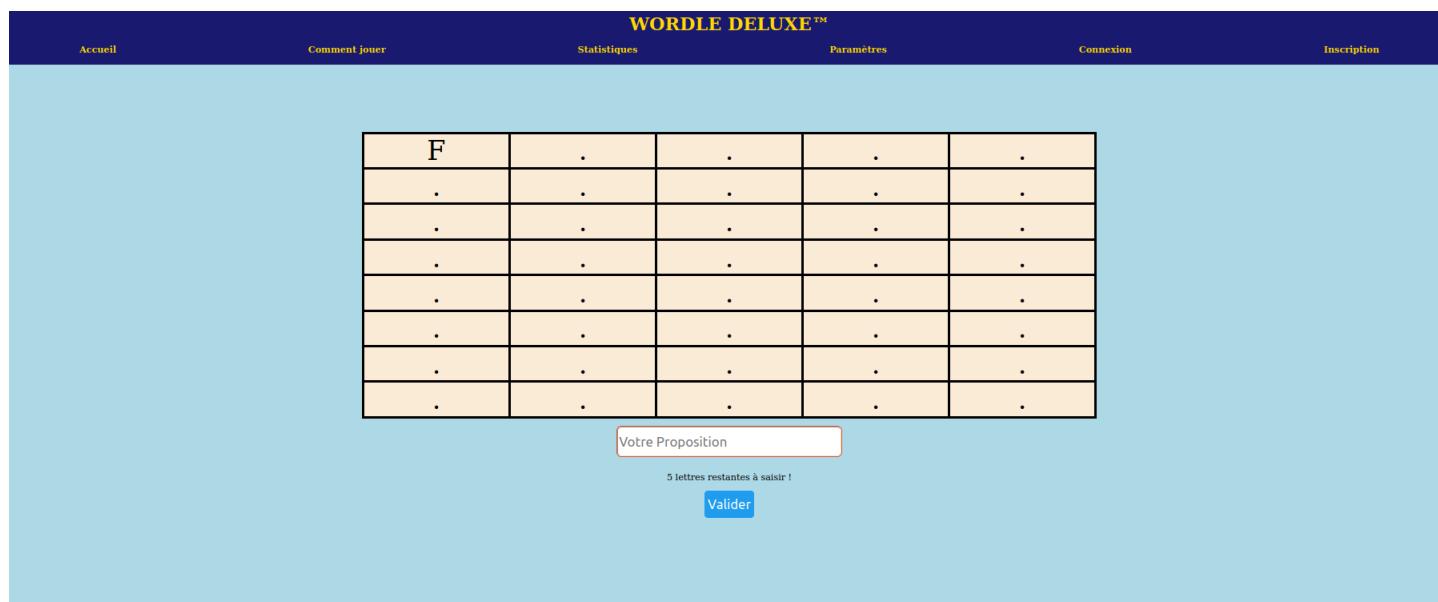


FIGURE 20 – Page d'accueil

WORDLE DELUXE™

[Accueil](#) [Comment jouer](#) [Statistiques](#) [Paramètres](#) [Connexion](#) [Inscription](#)

Comment jouer à Wordle Deluxe

Notre site vous propose en plus du Wordle Classique, différents modes de jeu avec chacun ses propres règles et spécificités pour votre plus grand plaisir.

Règle classique

Vous devez trouver un mot de langue française. Chaque mot que vous proposez doit être de la même longueur que le mot à trouver.

Après chaque proposition, l'ordinateur vous affiche des couleurs pour chacune des lettres proposées dans votre mot :

- rouge si la lettre est au même endroit que dans le mot à deviner
- jaune si la lettre est dans le mot à deviner mais pas au bon endroit
- sinon la lettre n'est pas dans le mot à deviner

Chaque partie est limitée par un nombre d'essais pour trouver le mot. Le nombre d'essais correspond au nombre de lignes du tableau.

La partie se termine si vous trouvez le mot ou si vous ne le trouvez pas après le nombre d'essais maximum.

Vous pouvez changer le nombre de lettres du mot à trouver ainsi que le nombre d'essais à trouver depuis la page paramètres.

Règle Survie

Les règles pour deviner le mot sont les mêmes que dans le mode classique.

Cependant vous avez toujours 6 essais pour trouver le mot et la longueur du mot est aléatoire.

Vous disposez de 3 vies, vous perdez une vie lorsque vous ne trouvez pas le mot à deviner. La partie s'arrête lorsque vous n'avez plus de vie.

Plus vous trouvez le mot rapidement, plus vous gagnez de points de score.

Règle BIG 50

Les règles pour deviner le mot sont les mêmes que dans le mode classique.

Cependant vous avez toujours 6 essais pour trouver le mot et la longueur du mot est aléatoire.

FIGURE 21 – Page expliquant les règles des différents mode de jeux

WORDLE DELUXE™

[Accueil](#) [Comment jouer](#) [Statistiques](#) [Paramètres](#) [Connexion](#) [Inscription](#)

Choix du mode de jeu :

Si vous avez choisi le mode de jeu classique alors choisissez le nombre de lettres du mot à trouver ainsi que le nombre d'essais que vous souhaitez avoir puis cliquer sur Jouer.

Simplement cliquez directement sur Jouer.

Choix du nombre de lettres :

Choix du nombre d'essais :

Singleplayer 

Multiplayer 

FIGURE 22 – Page permettant de choisir le nombre de lettres, le nombre d'essais, ainsi que le mode de jeu

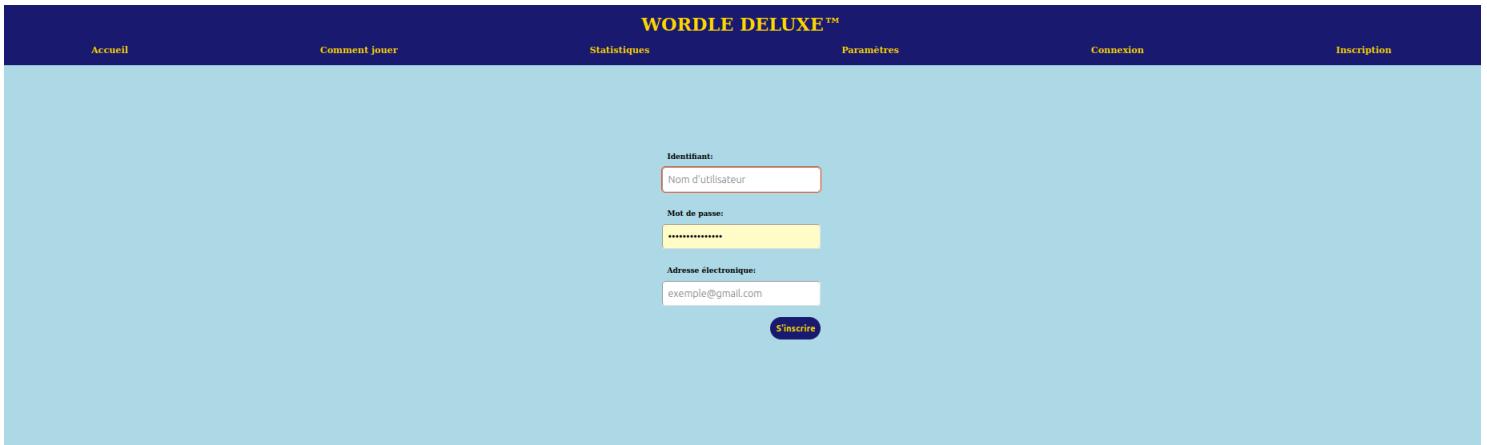


FIGURE 23 – Page où le joueur peut s'inscrire

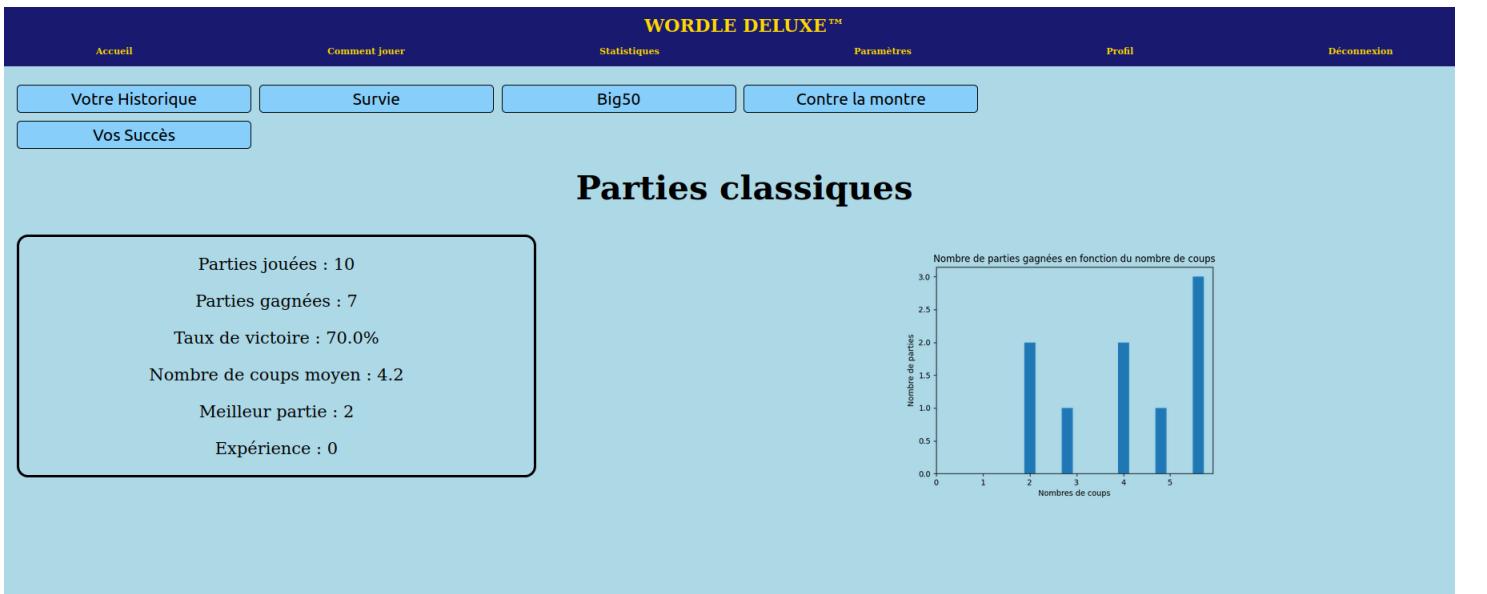


FIGURE 24 – Exemple de statistiques pour le mode classique

5.5.4 Bonus : JavaScript

On a réalisé en parallèle un jeu Wordle en JavaScript qui nous permet d'avoir accès à un clavier numérique et d'éviter le recours aux bases de données.

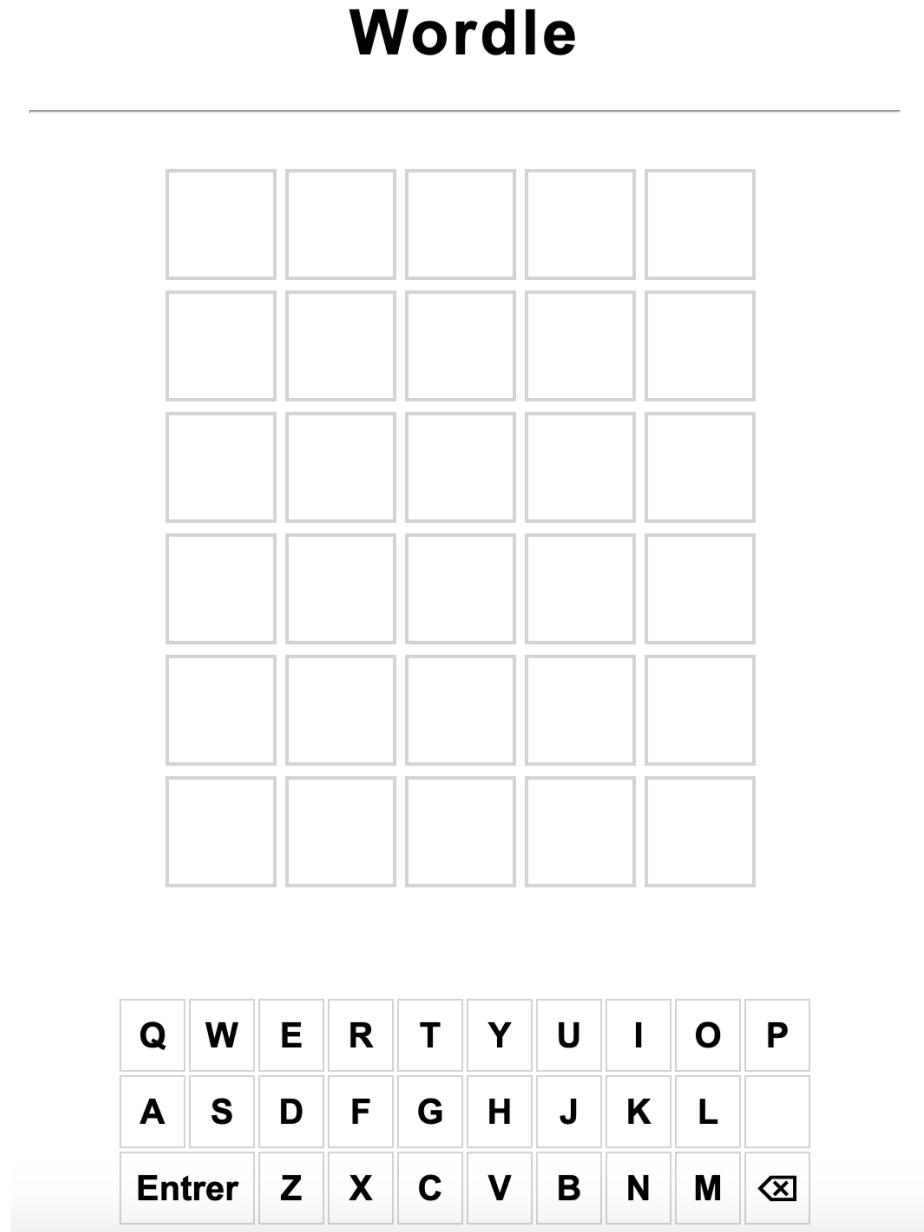


FIGURE 25 – Page d'accueil Wordle JS

Démonstration d'une partie de jeu avec deux listes :

Liste de mots à trouver = ["teste", "boire", "coder"]

Liste de mots dont ont peut s'aider = ["fermer", "ecole", "veste", "doigt"]

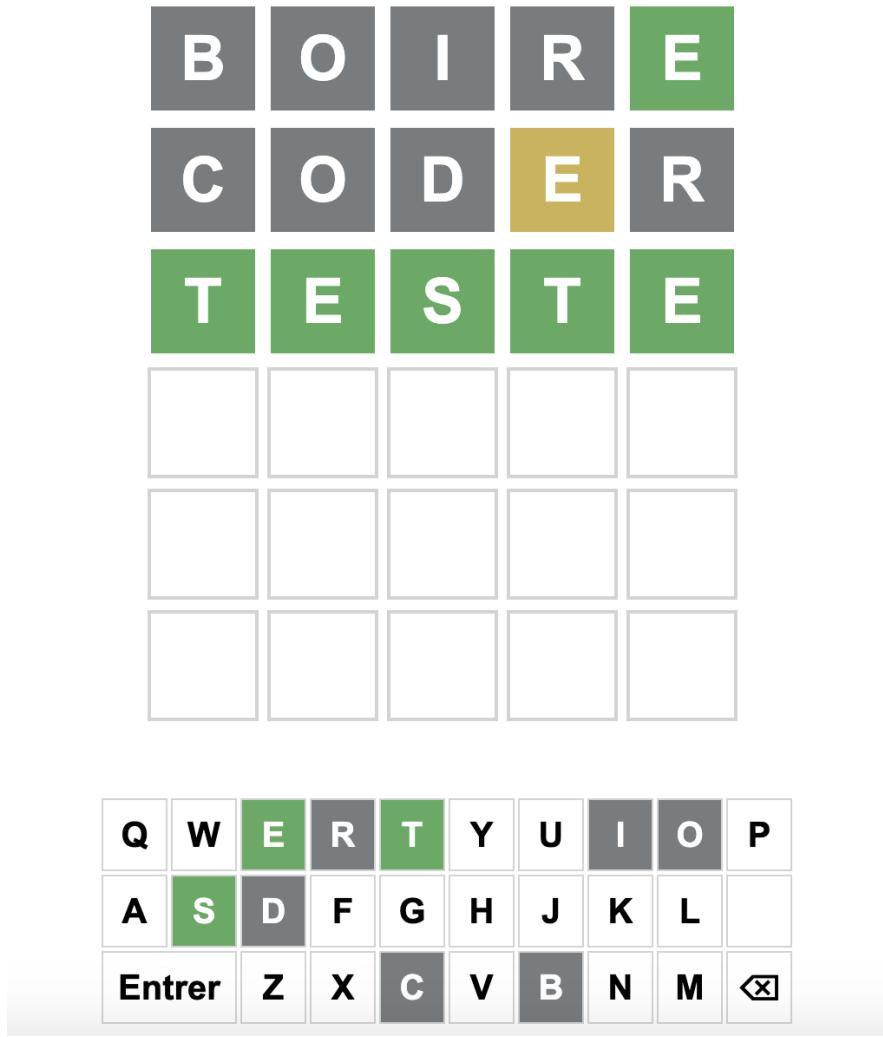


FIGURE 26 – Exemple d'une partie de Jeu Wordle avec JavaScript

6 Deuxième partie : Solveur Wordle

6.1 Présentation et attendus

L'objectif de la seconde partie du projet était d'implémenter en C un solveur de Wordle. Le but étant que le solveur réussisse à trouver le mot attendu en moins de 6 coups. Nous n'avions pas à connecter notre application web au solveur, en effet nous rentrons "à la main" la réponse donnée par le jeu dans le solveur après que ce dernier est proposé un mot.

6.2 Mise en oeuvre

6.2.1 Principe du solveur

Nous avons décidé de ranger tout les mots du dictionnaire dans une liste chaînée, ainsi, cette liste sera la liste des mots qui peuvent être solution de la partie en cours. A chaque tour, le solveur va proposer un mot de cette liste, et en fonction de la réponse, supprimer tous les mots qui ne peuvent plus être solution au vu de cette réponse. Par exemple, si le solveur propose le mot "table", et que la réponse est "20221", tous les mots qui ne commencent pas par un "t" seront supprimé, etc...

Ainsi, la liste va réduire de taille à chaque mot proposé. La problématique étant maintenant le choix du mot à proposer parmi cette liste des mots encore possible.

Nous avons eu deux idées pour cela, une dites fréquentielle, une autre dites entropique. Comme expliqué précédemment, le solveur fréquentiel calcule un score pour chaque mot en fonction des lettres qui le compose et de leurs fréquence d'apparition dans le dictionnaire. Le solveur entropique, lui calcule le mot qui, peu importe la réponse du jeu, nous donne le plus d'information possible. C'est-à-dire le mot qui permettra de réduire le plus possible la liste des mots encore possible. C'est donc ce choix de mot à proposer qui va déterminer la rapidité de notre solveur.

6.2.2 Structures de données utilisées

Pour le solveur fréquentiel :

Element	Structure
$element_{mot}$	$double$ score fréquence $char$ mot[20] $element_{mot}^*$ suivant
$list_{mot}$	$element_{mot}^*$ premier

Pour le solveur par entropie :

Element	Structure
$element_{info}$	int match $double$ bits $char$ resultat[20] ; $element_{info}^*$ suivant
$liste_{info}$	$element_{info}^*$ premier $list_{info}^*$ next $char$ word[20] $double$ meanBits
$allListe_{info}$	$list_{info}^*$ premier

6.2.3 Fonctions utilisées

Voici le tableau résumant toutes les fonctions codées, présentant les développeurs, testeurs et analyseurs de chaque fonction. Nous avons tenu à ce que chaque fonction ait un développeur différent du testeur, afin de trouver des erreurs que le développeur aurait pu ne pas voir.

D	Développeur
T	Testeur
A	Analyseur
DA	Développeur Analyseur
TA	Testeur Analyseur

Groupe	Fonctions	Variables Entrée	Variables Sortie	Description	Becquet Albert	Christiaen Adrien	Lacroix Thomas	Vermel Louis
Structure	<code>list_create()</code>	/	Liste chaînée	Crée la liste chaînée contenant les mots			DA	T
	<code>ajout_mots()</code>	Liste chaînée	/	Lit le fichier texte et choisi les mots à ajouter	D		DA	T
	<code>list_append()</code>	Liste chaînée, mot, score fréquence	/	Ajoute un élément à la liste chaînée	D		DA	T
	<code>list_destroy()</code>	Liste chaînée	/	Libère la mémoire allouée à la liste chaînée	D		DA	T
	<code>list_print()</code>	Liste chaînée	/	Affiche le contenu de la liste chaînée	D		DA	T
	<code>createListInfo()</code>	/	Liste chaînée	Crée la liste chaînée contenant les bits	DA		T	
	<code>info_print()</code>	Elément liste chaînée	/	Affiche un élément de la liste chaînée contenant les bits	DA		T	
	<code>listInfo_print()</code>	Liste chaînée	/	Affiche le contenu de la liste chaînée	DA		T	
	<code>listInfo_append()</code>	Liste chaînée, pattern	/	Ajoute un élément à la liste chaînée	DA		T	
	<code>listInfo_destroy()</code>	Liste chaînée	/	Libère la mémoire allouée à la liste chaînée	DA		T	
	<code>initListInfo()</code>	Liste chaînée	/	REMPLI la liste chaînée contenant les bits avec tous les patterns possibles	DA		T	
	<code>lengthListInfo()</code>	Liste chaînée	Longueur	Taille de la liste chaînée	DA		T	
Théorie Information	<code>createAllInfo(list)</code>	/	Liste chaînée	Crée la liste chaînée contenant toutes les listes chaînées de bits	DA		T	
	<code>destroyAllInfo()</code>	Liste chaînée	/	Libère la mémoire allouée à la liste chaînée	DA		T	
	<code>allResults()</code>	Liste chaînée, pattern, profondeur, taille du pattern	/	Calcule toutes les combinaisons possibles de pattern	D		TA	
	<code>getMatches()</code>	Liste chaînée (bits), Liste chaînée (mots), Mot	/	Calcule le nombre de "match" possibles avec un mot donné pour chaque pattern	DA		T	
	<code>indiceOccurrence()</code>	Mot, caractère	Indice de profondeur	Cherche l'indice où se trouve un caractère dans un mot	DA			T
	<code>getBits()</code>	Nombre de matchs, Nombre de mots	Bits d'information	Converti le nombre de matchs en nombre de bits d'information	DA			T
	<code>getAllInfoForAllWords()</code>	Liste chaînée (mots)	/	Calcule le nombre de bits pour chaque mot	D			TA
Gestion des mots	<code>getAllInfoForOneWord()</code>	Liste chaînée (bits), Liste chaînée (mots), Mot	/	Calcule le nombre de bits pour un seul mot	D			TA
	<code>getMeanBits</code>	Liste chaînée (bits)	Moyenne bits	Calcule la moyenne des bits contenu dans une liste de bits	D			TA
	<code>getBestWord</code>	Liste chaînée (globale)	Mot	Choisi le mot associé à la meilleure valeur moyenne de bits	D		A	T
	<code>updateList()</code>	Liste chaînée (mots), Mot, Pattern	/	Garde uniquement les mots voulus selon certains critères	D	T	D	DA
	<code>presentXfois</code>	Tableau (mot), Caractère, Entier	Entier (0 ou 1)	Renvoie si un caractère est présent X fois dans un mot	T			DA
Maximum	<code>supprimeMot()</code>	Liste chaînée, élément	Liste chaînée	Supprime des éléments de la liste chaînée par critère	T	DA	D	
	<code>getResults()</code>	Longueur	Mot	Récupère le résultat renvoyé par le jeu	T		D	DA
Fréquence	<code>giveProposition()</code>	Liste chaînée (mots)	Mot	Choisi le prochain mot qui sera proposé par le solveur		DA	D	T
	<code>freq_letters()</code>	Liste chaînée, caractère	Fréquence (double)	Calcule la fréquence d'apparition des lettres		TA	D	
Console	<code>freqScore()</code>	Liste chaînée, Liste des fréquences	/	Calcule le score fréquence de chaque mot en fonction de la fréquence d'apparition des lettres	D	TA		
	<code>main()</code>	/	/	Fonction générale faisant fonctionner le solveur	D	D	D	D

FIGURE 27 – Gestion/Explication fonctions

6.2.4 Analyse de la compléxité

Fonction	Complexité
$list_{create}()$	$\Theta(1)$
$ajout_{mot}()$	$\Theta(n^2)$
$list_{append}()$	$\Theta(n)$
$list_{destroy}()$	$\Theta(n)$
$list_{print}()$	$\Theta(1)$
$createListInfo()$	$\Theta(1)$
$info_{print}()$	$\Theta(1)$
$listInfo_{print}()$	$\Theta(1)$
$listInfo_{append}()$	$\Theta(n)$
$listInfo_{destroy}()$	$\Theta(n)$
$initListInfo()$	$\Theta(3^n)$
$lengthListInfo()$	$\Theta(n)$
$createAllInfoList()$	$\Theta(1)$
$destroyAllInfoList()$	$\Theta(n^2)$
$allResults()$	$\Theta(3^n)$
$getMatches()$	$\Theta(n^2)$
$indiceOccurrence()$	$\Theta(1)$
$getBits()$	$\Theta(1)$
$getAllInfoForAllWord()$	$\Theta(n^3)$
$getAllInfoForOneWord()$	$\Theta(n^2)$
$getMeanBits()$	$\Theta(n)$
$getBestWord()$	$\Theta(n)$
$updateList()$	$\Theta(n^2)$
$presentXfois()$	$\Theta(1)$
$supprimeMot()$	$\Theta(n)$
$getResult()$	$\Theta(1)$
$giveProposition()$	$\Theta(n)$
$freqLetters()$	$\Theta(n)$
$freqScore()$	$\Theta(1)$
$main() solveur frequentiel$	$\Theta(n^2)$
$main() solveur entropie$	$\Theta(n^3)$

6.3 Résultats

6.3.1 Modèle fréquentiel

On va jouer une partie classique : un mot de 5 lettres à trouver en moins de 6 coups.

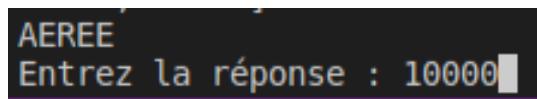


FIGURE 28 – Le premier mot proposé par le solveur est AEREE, le jeu renvoie 10000

```

Supression du mot: AZERI
Mot étudié: AZORA
Non supression.
Mot étudié: AZOTA
Non supression.

```

FIGURE 29 – On voit ici la liste qui s'actualise : certains mots, qui ne peuvent être la solution, sont supprimé de la liste

```

ARASA
Entrez la réponse : 20202

```

FIGURE 30 – Le mot proposé suivant est ARASA. On va sauter les étapes suivantes car c'est le même principe.

```

[ABACA, AGACA, AGADA, ATACA, AVALA]
AGACA
Entrez la réponse : 20222

```

FIGURE 31 – Le solveur finit par trouver le mot solution, ATACA

```

[ABACA, ATACA]
ATACA
Entrez la réponse :

```

FIGURE 32 – La liste est maintenant réduite à 4 éléments et le solveur propose le mot TRAME, qui est la solution

A	E	R	E	E
A	R	A	S	A
A	S	A	N	A
A	G	A	C	A
A	T	A	C	A
.

Votre Proposition

Rejouer

Vous avez gagné

FIGURE 33 – La partie sur l'application web

6.3.2 Modèle entropique

On va jouer une partie classique : un mot de 5 lettres à trouver en moins de 6 coups.

```
PORES  
Entrez la réponse : 00110
```

FIGURE 34 – Le premier mot proposé par le solveur est PORES, le jeu renvoie 00102

```
Mot étudié: RECUP  
Supression du mot: RECUP  
Mot étudié: RECUT  
Non suppression.  
Mot étudié: REDAN  
Non suppression.
```

FIGURE 35 – On voit ici la liste qui s'actualise : certains mots, qui ne peuvent être la solution, sont supprimé de la liste

```
Mot: RIFLE. Bits: 2.868313  
Mot: RINCE. Bits: 2.341564  
Mot: RIOTE. Bits: 2.267490  
Mot: RISSE. Bits: 2.139918  
Mot: RUADE. Bits: 2.983539  
Mot: RUCHE. Bits: 2.395062  
Mot: RUENT. Bits: 2.666667  
Mot: RUERA. Bits: 2.802469  
Mot: RUGIE. Bits: 3.189300  
Mot: RUILE. Bits: 3.193416  
Mot: RUINE. Bits: 2.884774  
Mot: RUSSE. Bits: 2.049383  
Mot: SABRE. Bits: 2.358025
```

FIGURE 36 – On voit ici le calcul de bits pour chaque mots encore possible

```

Best bits: 3.617284.
Best word: TETRA
TETRA
Entrez la réponse : 21011

```

FIGURE 37 – Le solveur retourne le mot qui a le meilleur score : TETRA ici

```

[TRABE, TRACE, TRAIE, TRAME]
Mot: TRABE. Bits: 0.008230
Mot: TRACE. Bits: 0.008230
Mot: TRAIE. Bits: 0.008230
Mot: TRAME. Bits: 0.008230
Best bits: 0.008230.
Best word: TRAME
TRAME
Entrez la réponse :

```

FIGURE 38 – La liste est maintenant réduite à 4 éléments et le solveur propose le mot TRAME, qui est la solution

P	O	R	E	S
T	E	T	R	A
T	R	A	M	E
.
.
.
.
.

Votre Proposition

Rejouer

Vous avez gagné

FIGURE 39 – La partie sur l'application web

6.4 Limites des modèles

6.4.1 Limites du modèle fréquentiel

Le premier mot proposé est toujours AEREE, cela est due au fait que les lettres A et E sont très présentes dans le dictionnaire. Le fait que le mot contiennent trois E réduit sûrement l'efficacité du solveur.

6.4.2 Limites du modèle entropique

Le majeur problème de ce modèle est le temps d'exécution car le calcul des bits est long (surtout pour le premier mot, même si étant toujours le même, on n'a besoin de calculer les bits que pour la liste complète).

7 Bilan du projet

Travail attendu	Un rapport détaillé Réalisation d'une base de données Réalisation d'un serveur web Réalisation d'algorithme du jeu Wordle
Travail accompli	Rapport détaillé Réalisation d'une base de données Réalisation d'un serveur web Réalisation d'algorithme du jeu Wordle
Retours positifs	Projet intéressant Réutilisation de connaissances S'améliorer sur ses compétences
Retours négatifs	Régularité de travail bousculé par les partiels

TABLE 1 – Bilan Wordle Application

Travail attendu	Un rapport détaillé Réalisation d'une structure de données Réalisation de fonctions en C Réalisation de fonctions tests en C
Travail accompli	Rapport détaillé Réalisation d'une structure de données Réalisation de fonctions en C Réalisation de fonctions tests en C
Retours positifs	Projet intéressant Acquisition de nouvelles compétences
Retours négatifs	Régularité de travail bousculé par les partiels Manque de TP sur les arbres (un peu en retard) pour avoir une structure de données performantes Timing assez restreint

TABLE 2 – Bilan Solveur

7.1 Remerciements

Merci à l'équipe pédagogique pour leur disponibilité et rapidité durant ces temps particuliers.

8 Références

- un tutoriel sur l'utilisation de la théorie de l'information pour optimiser la résolution d'une session Wordle
<https://www.youtube.com/watch?v=v68zYyaEmEAt=250s>
- le complément du tutoriel précédent qui y corrige un bug <https://www.youtube.com/watch?v=fRed0Xmc2Wg>

9 Annexes

9.1 Comptes rendus

9.1.1 CR1

Compte-rendu N°1 - Adrien Christiaën - 23/03/2022

Motif :	Lieu :
Réunion de chantier afin de mettre en place le projet PPII-2	Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 23/03/2022 16h00 1h15

Ordre du jour

1. Rencontre entre les membres de l'équipe
2. Prise de connaissance du sujet
3. Liste exhaustives d'idées réalisables dans le cadre du projet

Informations échangées

Brainstorming d'idées pour le projet :

Propositions	Détails
Document de conception à réaliser Faire valider ce document	-Présentation de wordle - Présentation textuelle de l'application - Présentation argumentée de la base de données (Présentation des tables) - Présentation agencement pages web - Paramétrages du jeu
Mode Connexion	-Connexion (login et mdp) -Connexion invité (enregistre pas les parties) -Possibilité d'utiliser les cookies pour ne pas se connecter
Page d'accueil	-Utilisateur peut choisir le nombre de lettre de son mot -Une seule page pour le site web -Implémentation des fonctionnalités en une seule page
Implémentation de la saisie	-Input avec bouton pour tester le mot -Réponse dynamique de l'écran lorsque détecte une touche du clavier -Affichage du clavier sur l'écran
Fonctionnalités possibles	-Possibilité de création de partie multiplayer -Système d'amis -Comparaison des statistiques entre les "amis" -Partage de partie -Débrief de la partie par une IA ? -Système de niveau et d'xp -Système de contre la montre

Tâches effectuées par rapport à la dernière réunion

To Do List

Description	Responsable	Délai	Livrable	Validé par
Réalisation du CR1	Adrien CHRISTIAËN	Prochaine réunion	CR1	Equipe projet
Réalisation du document conception			Document de conception	
Réalisation document conception	Albert BECQUET	Prochaine réunion	Document de conception	Equipe projet
Réalisation document conception	Louis VERMEL	Prochaine réunion	Document de conception	Equipe projet
Réalisation document conception	Thomas LACROIX	Prochaine réunion	Document de conception	Equipe projet

Date de la prochaine réunion

24 Mars à 16h

9.1.2 CR2

Compte-rendu N°2 - Louis Vermel - 24/03/2022

Motif : Réunion de chantier afin de répartir les tâches	Lieu : Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 24/03/2022 16h00 0h45

Ordre du jour

1. Répartir les tâches

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
Python	-Réaliser une première ébauche du code de wordle - Trier le dictionnaire obs6
HTML et CSS	-Implémenter les premières pages (cf. design web conception) -Réaliser un schéma des routes
Latex	-Faire le compte-rendu 2 -Récrire le document de conception
Base de données	-Passer la base de données en 3NF -Créer la base de données (cf. design web conception) -Relier la base de données avec le site
Gestion de projet	-Réaliser le diagramme de Gantt -Réaliser la matrice SWOT -Réaliser le WBS et le RACI

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Réalisation du CR1	Adrien CHRISTIAËN	CR1	Equipe projet
Réalisation du document conception		Document de conception	
Réalisation document conception	Albert BECQUET	Document de conception	Equipe projet
Réalisation document conception	Louis VERMEL	Document de conception	Equipe projet
Réalisation document conception	Thomas LACROIX	Document de conception	Equipe projet

To Do List

Date de la prochaine réunion

31 Mars à 16h00

Description	Responsable	Délai	Livrable	Validé par
Réalisation d'une page en html Passage en 3NF de la BDD Implémentation de la BDD Réalisation du code de wordle Réalisation du diagramme de Gantt	Adrien CHRISTIAËN	Prochaine réunion	Page d'accueil du site BDD implémentée Code de Wordle Diagramme de Gantt	Equipe projet
Réalisation d'une page en html Document de conception Latex Réalisation du RACI	Albert BECQUET	Prochaine réunion	Page connection du site Doc de conception Latex RACI	Equipe projet
Réalisation d'une page en html Réalisationde CR2 Trier le dictionnaire Réalisation du WBS	Louis VERMEL	Prochaine réunion	Page des stats du site CR2 Dictionnaire trié WBS	Equipe projet
Réalisation d'une page en html Réalisation du code de worlde Réalisation de la matrice SWOT	Thomas LACROIX	Prochaine réunion	Page modes de jeu du site Code de Wordle Matrice SWOT	Equipe projet

9.1.3 CR3

Compte-rendu N°2 - Thomas Lacroix - 31/03/2022

Motif : Réunion de chantier pour expliquer présenter l'avancement des objectifs	Lieu : Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 31/03/2022 15h15 1h45

Ordre du jour

1. Présenter l'avancement des tâches
2. Critiquer l'avancement des tâches
3. Répartir l'avancement des prochaines tâches

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
Python	Comparer les codes du wordle pour établir le programme final qui sera utilisé sur la page web
HTML et CSS	Faire la page accueil Relier la page statistique à la base de données Améliorer le css de la page statistique Relier la page paramètres à la base de données Améliorer le style css de l'ensemble du site
Latex	Faire le compte-rendu
Base de données	Terminer le passage la base de données en 3NF Ajouter un "DROP TABLE" pour initialiser les tables
Gestion de projet	Mettre à jour la matrice RACI Corriger la matrice SWOT Améliorer le WBS Commencer les backlogs

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Wordle en python avec interface graphique supportée par pygame Création de la base de données	Adrien CHRISTIAËN	Fichier python Fichier .db	Equipe projet
Première version HTML et CSS des pages d'inscription et de connexion Première version HTML et CSS de la structure du site Rédaction du document de conception Matrice RACI	Albert BECQUET	Page d'inscription et de connexion Page d'accueil du site Document de conception en pdf RACI	Equipe projet
Réalisation de plusieurs dictionnaires pour chaque mots de tailles différentes Réalisation du CR2 Réalisation du WBS Base html de la page statistiques	Louis VERMEL	CR2 en pdf WBS Page statistiques	Equipe projet
Réalisation du code wordle jouable depuis le terminal Version 1 de la page html et css des paramètres Matrice SWOT	Thomas LACROIX	Fichier python Page paramètres SWOT	Equipe projet

To Do List

Date de la prochaine réunion

11 Avril à 16h00

Description	Responsable	Délai	Livrable
Page html de l'accueil Implémentation de la BDD Réalisation du diagramme de Gantt	Adrien CHRISTIAËN	Prochaine réunion	Page d'accueil Fichier.db Gantt
Fonction python pour crypter un mot de passe Améliorer le CSS des pages inscription/connexion Compléter la matrice RACI	Albert BECQUET	Prochaine réunion	Fichier python Pages améliorées RACI à jour
Améliorer le WBS Améliorer le css et faire le lien avec la base de données de la page statistiques	Louis VERMEL	Prochaine réunion	WBS Page statistiques
Faire le CR3 en Latex Lier la page paramètre à la base de données Rédiger les backlogs	Thomas LACROIX	Prochaine réunion	CR3 Pages paramètres Backlogs

9.1.4 CR4

Compte-rendu N°4 - Albert Becquet - 11/04/2022

Motif :	Lieu :
Mise en commun de l'avancement, définition des prochains objectifs	Distanciel
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 11/04/2022 16h00 1h20

Ordre du jour

1. Mise en commun de l'avancement
2. Définition des objectifs suivants

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
HTML et CSS	-Page statistiques à mettre en forme -Finir la page d'accueil
Latex	-Faire le compte-rendu 4 -Démonstration de la 3NF pour la base de donnée
Gestion de projet	-Mettre à jour le diagramme de Gantt -Mettre à jour la matrice RACI -Améliorer et mettre à jour le Backlog

Tâches effectuées par rapport à la dernière réunion

To Do List

Date de la prochaine réunion

14 Avril à 16h00

Description	Responsable	Livrable	Validé par
Code python du wordle Vérification de la base 3NF Diagramme de Gantt	Adrien CHRISTIAËN	Code python Document PDF	Equipe projet
Cryptage du mot de passe (non testé) Mise à jour de la matrice RACI	Albert BECQUET	Code python Document PDF	Equipe projet
Dynamisation de la partie statistiques et tracés des premiers graphiques	Louis VERMEL	Page statistiques	Equipe projet
Dynamisation des paramètres Premier wordle intégré au site web Première version du Backlog	Thomas LACROIX	Code python/HTML Code python/HTML/CSS Document PDF	Equipe projet

Description	Responsable	Délai	Livrable	Validé par
Mettre à jour Gantt Démonstration 3NF BDD Amélioration Backlog	Adrien CHRISTIAËN	Prochaine réunion	Code de Wordle Document PDF Document PDF	Equipe projet
Mettre à jour RACI CR4 Latex Partie succès/expérience	Albert BECQUET	Prochaine réunion	Document PDF Document PDF Code Python/HTML	Equipe projet
Statistiques mise en forme	Louis VERMEL	Prochaine réunion	Code HTML/CSS	Equipe projet
Amélioration code de worlde Amélioration Backlog	Thomas LACROIX	Prochaine réunion	Code Python Document PDF	Equipe projet

9.1.5 CR5

Compte-rendu N°5 - Louis Vermel - 15/04/2022

Compte-rendu N°5

Motif : Réunion de chantier afin de répartir les tâches	Lieu : Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 15/04/2022 14h30 1h15

Ordre du jour

1. Démonstration des différents avancements
2. Détermination des prochains objectifs

Informations échangées

Listing des tâches à réaliser :

Tâches effectuées par rapport à la dernière réunion

To Do List

Date de la prochaine réunion 20 Avril à 14h30

Domaines	Détails
Python	-Implémentation fonctions expériences dynamiques
HTML et CSS	<ul style="list-style-type: none"> -Accueil : Relier JS à Flask -Ajouter une page historique -Modifier/Ajouter graphiques dans statistiques/historique -Partie profil à avancer et dynamiser -Mettre à jour historique après chaque partie -Développement des modes de jeux
Latex	-Faire le compte-rendu 5
Base de données	<ul style="list-style-type: none"> -Mettre valeur par défaut pour certaines données -Ajouter le mot chercher dans la table historique -Changer de nom Identifiant/Nom d'utilisateur
Gestion de projet	<ul style="list-style-type: none"> -Backlog à terminer -Mise à jour Gantt

Description	Responsable	Livrable	Validé par
Mettre à jour Gantt Démonstration 3NF BDD Amélioration Backlog	Adrien CHRISTIAËN	Code de Wordle Document PDF Document PDF	Equipe projet
Mettre à jour RACI CR4 Latex Partie succès/expérience	Albert BECQUET	Document PDF Document PDF Code Python/HTML	Equipe projet
Statistiques mise en forme	Louis VERMEL	Code HTML/CSS	Equipe projet
Amélioration code de worlde Amélioration Backlog	Thomas LACROIX	Code Python Document PDF	Equipe projet

Description	Responsable	Délai	Livrable	Validé par
Backlog à terminer Changer de nom Identifiant Accueil : Relier JS à Flask Mise à jour Gantt	Adrien CHRISTIAËN	Prochaine réunion	Baclog BDD modifiée Page accueil avec du JS Gantt Mis à jour	Equipe projet
Mise à jour RACI Fonctions expériences dynamiques Partie profil à avancer	Albert BECQUET	Prochaine réunion	RACI mis à jour Fonction python Page profil	Equipe projet
Réalisation du CR5 Mettre valeur par défaut BDD Ajouter mot dans la table historique Crée page historique Maj historique après partie	Louis VERMEL	Prochaine réunion	CR5 BDD modifiée BDD modifiée page historique	Equipe projet
Backlog à terminer Dév des modes de jeux	Thomas LACROIX	Prochaine réunion	Backlog Mdj implémenter	Equipe projet

9.1.6 CR7

Compte-rendu N°7 - Thomas Lacroix - 24/04/2022

Motif :	Lieu :
Réunion de chantier pour expliquer présenter l'avancement des objectifs	Distanciel
Présent(s) (retard/excusés/non excusés) :	Date / heure de début / durée :
Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	24/04/2022 15h15 1h45

Ordre du jour

1. Présenter l'avancement des tâches
2. Critiquer l'avancement des tâches
3. Répartir l'avancement des prochaines tâches

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
Python	Effectuer un nouveau tri du dictionnaire pour avoir des mots plus "courants" Essayer de réaliser des tests sur le code avec pytest Terminer la liaison entre flask et le wordle en javascript
HTML et CSS	Corriger les bugs dans la page statistiques Actualiser les statistiques en fonction des modes de jeu Faire la mise en page de la page profil
Latex	Faire le compte-rendu de réunion
Base de données	Mettre au point la table contenant les données de la partie en cours Mise à jour du schéma 3NF de la base de données
Gestion de projet	Mettre à jour la matrice RACI Mettre à jour le Gantt Mettre à jour le backlog Commencer la rédaction du rapport final

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Avancement dans l'implémentation du wordle en javascript avec Flask Réalisation du CR6	Adrien CHRISTIAËN	Fichier python Document pdf	Équipe projet
Expérience pour le mode timer Lien dynamique des succès avec la base de données Barre d'expérience utilisateur Matrice RACI	Albert BECQUET	Code python Code python HTML/CSS RACI	Équipe projet
Mise à jour BDD pour les statistiques Création des statistiques pour les modes de jeu	Louis VERMEL	Code python Code python + HTML	Équipe projet
Mise en page global et page statistiques Rédaction des règles de jeu Graphismes pour les modes de jeux Timer dynamic pour contre la montre Mise à jour du backlog	Thomas LACROIX	Code HTML et CSS Code HTML Code HTML + images Code javascript BACKLOG	Équipe projet

To Do List

Description	Responsable	Délai	Livrable
Mettre à jour schéma 3NF Terminer lien entre flask et le wordle en javascript Mettre à jour et améliorer le Gantt Rédaction du rapport de projet	Adrien CHRISTIAËN	Prochaine réunion	Document pdf Fichier python GANTT Document pdf
Tester le code du wordle avec pytest Développer les succès Mettre à jour la matrice RACI	Albert BECQUET	Prochaine réunion	Code python Code python + HTML RACI
Corriger erreurs page statistiques Dynamisation des statistiques pour les différents modes de jeu	Louis VERMEL	Prochaine réunion	Page statistiques Code python
Faire le CR7 en Latex Mise en page de la page profil Mettre à jour et améliorer le backlog	Thomas LACROIX	Prochaine réunion	CR7 Page profil BACKLOG

Date de la prochaine réunion

28 Avril à 15h00

9.1.7 CR8

Compte-rendu N°8 - Albert Becquet - 28/04/2022

Motif :	Lieu :
Mise en commun de l'avancement, définition des prochains objectifs	Télécom Nancy
Présent(s) (retard/excusés/non excusés) :	Date / heure de début / durée :
Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix (absent car malade)	28/04/2022 15h00 1h00

Ordre du jour

1. Retour sur les éléments de la gestion de projet
2. Présentation des tâches réalisées
3. Discussion et Assignation des tâches futures

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
Python	-Debug intensif pour être en confiance lors de la démonstration
Web	-Correction de bugs sur les pages statistiques et historique -Finaliser la fusion javascript avec la branche principale -Correction bugs mise en page statistiques
Latex	-Compte-Rendu n°8
Base de données	-Schéma E/A à terminer -Document explicatif de la table Partie -Ajout de données pour visualisation des statistiques
Gestion de projet	-Backlog à mettre à jour -RACI à mettre à jour -Gantt à mettre à jour

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Schéma E/A partiellement terminé Diagramme de Gantt MAJ	Adrien CHRISTIAËN	PDF	Equipe projet
Testing des principales fonctions RACI MAJ	Albert BECQUET	Code python PDF	Equipe projet
Amélioration statistiques et correction de bugs Statistiques pour les modes de jeux	Louis VERMEL	Page statistiques/Python	Equipe projet
Mise en page du Profil Backlog partiellement terminé	Thomas LACROIX	HTML/CSS PDF	Equipe projet

Description	Responsable	Délai	Livrable
Schéma E/A à terminer Debug intensif Finaliser la fusion javascript avec la branche principale Gantt à mettre à jour	Adrien CHRISTIAËN	Prochaine réunion	PDF Jouer au Wordle JS/Python/HTML PDF
Compte-Rendu n°8 Debug intensif Correction bugs mise en page statistiques RACI à mettre à jour	Albert BECQUET	Prochaine réunion	PDF Jouer au Wordle HTML/CSS PDF
Debug intensif Correction de bugs sur les pages statistiques et historique	Louis VERMEL	Prochaine réunion	Jouer au Wordle
Backlog à mettre à jour Debug intensif Document explicatif de la table Partie	Thomas LACROIX	Prochaine réunion	PDF Jouer au Wordle PDF

To Do List

Date de la prochaine réunion

5 Mai à 13h00

9.1.8 CR9

Compte-rendu N°9 - Louis Vermel - 05/05/2022

Compte-rendu N°9

Motif :	Lieu :
Réunion de chantier afin de répartir les tâches	Télécom Nancy
Présent(s) (retard/excusés/non excusés) :	Date / heure de début / durée :
Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	05/05/2022 13h30 1h00

Ordre du jour

1. Retour sur des éléments de la gestion de projet
2. Revoir les attendus du solveur

Informations échangées

Listing des tâches à réaliser : Utilisation d'arbres binaires afin de trouver un protocole permettant d'être le plus rapide.

Faire un travail sur les probabilités/fréquences des lettres pour chaque longueurs, pour avoir le solveur le plus optimisé possible

Faire un algorithme qui nous donnerait les statistiques de réussite pour un solveur utilisé Nécessité de traduire l'entrée '21021' pour le solveur

Analyses complexité et mémoire des fonctions à ne pas oublier de faire

Personne développant les tests différente de la personne développant la fonction en question

Personne développant la fonction réalise également l'analyse en complexité et mémoire de la fonction

Faire une première analyse de la fréquence des lettres dans les mots, cette tâche reste cependant non prioritaire par rapport aux autres

Arriver à se mettre d'accord pour la prochaine réunion sur la manière dont va fonctionner la première version du solveur
Bien gérer le temps sur la partie solveur, les examens de fin d'année arrivant très vite

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Mettre à jour Gantt Schéma EA Amélioration Backlog	Adrien CHRISTIAËN	Gantt Mis à jour Document PDF Document PDF	Equipe projet
Mettre à jour RACI CR8 Latex Mise en page différentes pages Ajout de données BDD pour démonstration Test général du site	Albert BECQUET	Document PDF Document PDF Code Python/HTML BDD modifié	Equipe projet
Correction bug Test général du site	Louis VERMEL	Code HTML/CSS	Equipe projet
Document explicatif BDD Test général du site Backlog à mettre à jour	Thomas LACROIX	Document PDF Document PDF	Equipe projet

To Do List

Description	Responsable	Délai	Livrable	Validé par
Mise à jour Backlog Faire des recherches sur des solveurs Regarder les vidéos à la fin du sujet	Adrien CHRISTIAËN	Prochaine réunion	Baclog	Equipe projet
Mise à jour RACI Faire des recherches sur des solveurs Réalisation du CR8 Regarder les vidéos à la fin du sujet	Albert BECQUET	Prochaine réunion	RACI mis à jour CR8	Equipe projet
Réalisation du CR9 Faire des recherches sur des solveurs Regarder les vidéos à la fin du sujet	Louis VERMEL	Prochaine réunion	CR9	Equipe projet
Faire des recherches sur des solveurs Mise à jour Backlog Regarder les vidéos à la fin du sujet	Thomas LACROIX	Prochaine réunion	Backlog	Equipe projet

Date de la prochaine réunion 20 Avril à 14h30

9.1.9 CR10

Compte-rendu N°10 - Adrien Christiaën - 12/05/2022

Motif :	Lieu :
Réunion de chantier afin de mise au point pour le solveur	Télécom Nancy
Présent(s) (retard/excusés/non excusés) :	Date / heure de début / durée :
Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	12/05/2022 15h00 1h20

Ordre du jour

1. Débriefing sur la vidéo à regarder
2. Répartition des différentes fonctions à coder en C et du document SD

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
Fichier main	-Utilisation des fonctions nécessaires
Fonction nécessaires	<ul style="list-style-type: none"> -Fonction qui crée la liste chaînée des mots en x lettres -Fonction qui supprime les mots impossibles de la liste chaînée -Fonction qui choisit le prochain mot a donné -Fonction qui calcul et qui ajoute le poids(fréquence) associé à chaque mot de la liste chaînée -Fonction qui récupère la combinaison de chiffre entrée dans la console
Structure de données	<ul style="list-style-type: none"> Liste_mots_possibles -Premier_element Element -Mot -Score_freq -Pointeur_sur_suivant

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
MAJ GANTT	Adrien CHRISTIAËN	Document PDF	Equipe projet
MAJ RACI (x2) CR8	Albert BECQUET	Document PDF Document PDF	Equipe projet
CR9	Louis VERMEL	Document PDF	Equipe projet
MAJ Backlog	Thomas LACROIX	Document PDF	Equipe projet
Faire des recherches sur les solveurs existants Regarder les vidéos à la fin du sujet		Document PDF Document PDF	
Fonction qui supprime les mots d'une liste selon des critères Tests des fonctions développées	Equipe Projet	Document PDF Document PDF	Equipe projet
Analyse complexité et mémoire des fonctions développées Analyse de la fréquences des lettres		Document PDF Document PDF	

Description	Responsable	Délai	Livrable	Validé par
MAJ GANTT CR10	Adrien CHRISTIAËN	Prochaine réunion	Document PDF Document PDF	Equipe projet
MAJ RACI Document récap des fonctions/tests/structure	Albert BECQUET	Prochaine réunion	Document PDF Document PDF	Equipe projet
MAJ WBS CR9	Louis VERMEL	Prochaine réunion	Document PDF Document PDF	Equipe projet
MAJ BACKLOG	Thomas LACROIX	Prochaine réunion	Document PDF	Equipe projet

To Do List

Date de la prochaine réunion

19 Mai à 14h00

9.1.10 CR11

Compte-rendu N°11 - Thomas Lacroix - 19/05/2022

Motif :	Lieu :
Réunion de chantier pour expliquer présenter l'avancement des objectifs	Télécom Nancy
Présent(s) (retard/excusés/non excusés) :	Date / heure de début / durée :
Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	19/05/2022 14h00 1h30

Ordre du jour

1. Débriefer les fonctions réalisants le solveur
2. Réfléchir sur une nouvelle structure de données

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
C	Expliquer les structures des différentes fonctions Réaliser/Mettre à jour les fonctions nécessaires
SD	Étudier les avantages/inconvénients de la liste chaînée pour la structure de données Réfléchir à l'utilisation d'arbres pour la structure de données

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Début de la fonction qui choisit le prochain mot a donné Réalisation du CR10	Adrien CHRISTIAËN	Code C Document pdf	Équipe projet
Fonction qui calcul le score de fréquence d'un mot Document explication fonctions Matrice RACI	Albert BECQUET	Code C Fichier Excel RACI	Équipe projet
Fonction qui récupère la combinaison de chiffre entrée dans la consoles WBS solveur	Louis VERMEL	Code C WBS	Équipe projet
Récupération des mots pour les stocker dans la liste chainée Début de la fonction qui calcul la fréquence d'apparition des lettres	Thomas LACROIX	Code C Code C	Équipe projet

To Do List

Description	Responsable	Délai	Livrable
Terminer la fonction qui choisit le prochain mot a donné Mettre à jour le Gantt Rédaction du rapport de projet	Adrien CHRISTIAËN	Prochaine réunion	Code C GANTT Document pdf
Débuguer la fonction de récupération des mots Améliorer la fonction du calcul du score des mots Mettre à jour la matrice RACI	Albert BECQUET	Prochaine réunion	Code C Code C RACI
Fonction qui choisit les mots supprimés/garder Améliorer la fonction qui lit l'entrée de la console	Louis VERMEL	Prochaine réunion	Code C Code C
Faire le CR11 en Latex Débuguer la fonction de récupération des mots Terminer la fonction calcul de la fréquence des lettres Mettre à jour du backlog	Thomas LACROIX	Prochaine réunion	CR11 Code C Code C BACKLOG

Date de la prochaine réunion

22 Mai à 17h00

9.1.11 CR12

Compte-rendu N°12 - Albert Becquet - 22/05/2022

Motif : Avancement des fonctions du solveur	Lieu : Distanciel
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 22/05/2022 17h00 30min

Ordre du jour

1. Présentation des fonctions réalisées
2. Établissement des fonctions restantes

Informations échangées

Fonctions du solveur réalisées :

Responsable	Fonction	Statut
Christiaen Adrien	Supprime des éléments de la liste chaînée par critère Choisi le prochain mot qui sera proposé par le solveur	En cours En cours
Lacroix Thomas	Crée la liste chaînée contenant les mots Calcule la fréquence d'apparition des lettres	En cours En cours
Becquet Albert	Calcule le score fréquence de chaque mot	Terminé
Vermel Louis	Garde uniquement les mots voulus selon certains critères Récupère le résultat retourné par le jeu	En cours Terminé

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Avancement Rapport de Projet Diagramme de Gantt MAJ	Adrien CHRISTIAËN	PDF	Equipe projet
RACI Maj	Albert BECQUET	PDF	Equipe projet
WBS	Louis VERMEL	PDF	Equipe projet
Compte Rendu 11 Backlog MAJ	Thomas LACROIX	PDF	Equipe projet

To Do List

Date de la prochaine réunion

27 Mai, 16h00

Description	Responsable	Délai	Livrable
Gantt à mettre à jour Avancer fonctions solveur	Adrien CHRISTIAËN	Prochaine réunion	PDF C
Compte-Rendu n°12 Avancer fonctions solveur RACI à mettre à jour	Albert BECQUET	Prochaine réunion	PDF C PDF
Avancer fonctions solveur	Louis VERMEL	Prochaine réunion	C
Backlog à mettre à jour Avancer fonctions solveur	Thomas LACROIX	Prochaine réunion	PDF C

9.1.12 CR13

Compte-rendu N°13 - Thomas Lacroix - 27/05/2022

Compte-rendu N°13

Motif :	Lieu :
Réunion de chantier afin de répartir les tâches	Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën (absence justifiée due au gala de TN) Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 05/05/2022 13h30 1h00

Ordre du jour

- Debriefing sur les fonctions réalisées

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Gantt à mettre à jour Avancer sur les fonctions du solveur	Adrien CHRISTIAËN	Gantt à jour Fonction C	Equipe projet
CR12 Latex RACI à mettre à jour Avancer sur les fonctions du solveur	Albert BECQUET	Document PDF RACI à jour Fonction C	Equipe projet
Avancer sur les fonctions du solveur	Louis VERMEL	Fonction C	Equipe projet
Backlog à mettre à jour Avancer sur les fonctions du solveur	Thomas LACROIX	Backlog à jour Fonction C	Equipe projet

To Do List

Date de la prochaine réunion 29 mai à 20h00

Description	Responsable	Délai	Livrable	Validé par
Testing Maj Gantt Commencer rapport de projet	Adrien CHRISTIAËN	Prochaine réunion	Gant maj Rapport de projet	Equipe projet
Testing Mise à jour doc fct Mise à jour RACI Continuer fcts sur théorie de l'informaiton	Albert BECQUET	Prochaine réunion	doc fonctions C RACI mis à jour	Equipe projet
Testing Corriger derniers bug updateList Réalisation du CR13	Louis VERMEL	Prochaine réunion	CR13	Equipe projet
Testing Mise à jour Backlog Réunir l'ensemble des fonctions	Thomas LACROIX	Prochaine réunion	Backlog	Equipe projet

9.1.13 CR14

Compte-rendu N°14 - Adrien Christiaën - 29/05/2022

Motif :	Lieu :
Réunion de chantier pour l'avancée du code du solveur	Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 29/05/2022 21h00 1h00

Ordre du jour

1. Avancer et corriger le code du solveur
2. Commencer le rapport final

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
GDP	-SWOT + BACKLOG -RACI + Doc fonctions -Continuer le Rapport de projet -CR14
CODE	-Régler problème de liste vide UpdateListV2 -Relier main avec solveur bits -Testing des fonctions

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
MAJ GANTT Commencer le rapport de projet	Adrien CHRISTIAËN	Document PDF Document PDF	Equipe projet Equipe projet
MAJ RACI MAJ du doc Fonctions	Albert BECQUET	Document PDF Document PDF	Equipe projet
CR13 UpdateList fonctionnelle	Louis VERMEL	Document PDF CODE	Equipe projet Equipe projet
MAJ Backlog	Thomas LACROIX	Document PDF	Equipe projet
Réunir l'ensemble des fonctions Finaliser les deux solveurs Python représentation graphique de la vitesse des solveurs	Equipe Projet	CODE CODE CODE	Equipe projet

To Do List

Description	Responsable	Délai	Livrable	Validé par
MAJ GANTT CR14	Adrien CHRISTIAËN	Prochaine réunion	Document PDF Document PDF	Equipe projet
MAJ RACI Document récap des fonctions/tests/structure	Albert BECQUET	Prochaine réunion	Document PDF Document PDF	Equipe projet
MAJ BACKLOG	Thomas LACROIX	Prochaine réunion	Document PDF	Equipe projet
Régler problème de liste vide UpdateListV2 Relier main avec solveur bits Testing des fonctions	Equipe Projet	Prochaine réunion	CODE CODE CODE	Equipe projet

Date de la prochaine réunion

06 juin à 21h00

9.1.14 CR15

Compte-rendu N°15 - Thomas Lacroix - 06/06/2022

Motif : Dernière réunion de chantier pour décider des dernières tâches à affectuer avant de rendre le projet	Lieu : Télécom Nancy
Présent(s) (retard/excusés/non excusés) : Adrien Christiaën Albert Becquet Louis Vermel Thomas Lacroix	Date / heure de début / durée : 06/06/2022 21h00 1h30

Ordre du jour

1. Débriefer les tests des fonctions réalisants le solveur
2. Affectations des tâches restantes pour terminer le projet

Informations échangées

Listing des tâches à réaliser :

Domaines	Détails
C	Expliquer les éventuels problèmes lors des test des fonctions Corriger les problèmes trouvés
Latex	Terminer le rapport de projet

Tâches effectuées par rapport à la dernière réunion

Description	Responsable	Livrable	Validé par
Tests de fonctions Réalisation du CR14	Adrien CHRISTIAËN	Code C Document pdf	Équipe projet
Tests de fonctions Document explication fonctions Matrice RACI	Albert BECQUET	Code C Fichier Excel RACI	Équipe projet
Tests de fonctions	Louis VERMEL	Code C	Équipe projet
Tests de fonctions BACKLOG terminé	Thomas LACROIX	Code C BACKLOG	Équipe projet

To Do List

Description	Responsable	Délai	Livrable
Corrections éventuelles erreurs Rédaction du rapport de projet	Adrien CHRISTIAËN	Prochaine réunion	Code C Document pdf
Corrections éventuelles erreurs Rédaction du rapport de projet	Albert BECQUET	Prochaine réunion	Code C Document pdf
Corrections éventuelles erreurs Rédaction du rapport de projet	Louis VERMEL	Prochaine réunion	Code C Document pdf
Faire le CR15 en Latex Corrections éventuelles erreurs Rédaction du rapport de projet	Thomas LACROIX	Prochaine réunion	CR15 Code C Document pdf

Dernière réunion du projet

9.2 Sprint Backlog

