



PPII S6 - Rapport

Armand Londeix, Patrick O'Brien, Lucas Rioux, Serge Téhé

Mars 2022 - Juin 2022

Promotion 2024

PPII S6 : JEU WORDLE



Responsable du projet : Olivier Festor

28 mai 2022

Sommaire

1	Introduction	4
1.1	Objectif du projet	4
1.2	Règles du jeu de WORDLE	4
2	Implémentation de WORDLE	5
2.1	La base de données	5
2.2	L'interface Web	6
2.3	Le solveur	8
2.3.1	Le principe général	9
2.3.2	Les structures de données	9
2.3.3	Les Fonctions	10
3	Tests et performances	11
3.1	Fonction d'évaluation de l'application web	11
3.1.1	Complexité théorique	11
3.1.2	Graphe de performance	11
3.2	Fonction dicoload du solveur	12
3.2.1	Complexité théorique	12
3.3	Fonction evaluation du solveur	12
3.3.1	Complexité théorique	12
3.4	Fonction WordEntropy du solveur	12
3.4.1	Complexité théorique	12
3.5	Fonction getbestword du solveur	12
3.5.1	Complexité théorique	12
3.6	Fonction FindWordsGreen du solveur	12
3.6.1	Complexité théorique	12
3.7	Fonction FindWordsYellow du solveur	12
3.7.1	Complexité théorique	12
3.8	Fonction excludeWords du solveur	12
3.8.1	Complexité théorique	12
3.9	Fonction MotsSuivantsPossibles du solveur	12
3.9.1	Complexité théorique	12
4	Gestion de projet	13
4.1	Membres de l'équipe	13
4.2	Organisation du projet	13
4.3	Outils de travail	13
4.4	Documents relatifs au projet	13
4.5	Volume horaire	14
5	Bilan du projet	15
5.1	Travail réalisé	15
5.2	Bilan de l'équipe projet	15
5.3	Bilan individuel	15
5.3.1	Armand Londeix	15
5.3.2	Patrick O'Brien	15
5.3.3	Lucas Rioux	15
5.3.4	Serge Téhé	16

6	Annexe	17
6.1	Documents de gestion de projet	17
6.2	Comptes-rendus	21
6.2.1	Compte-rendu du 23 Mars 2022	21
6.2.2	Compte-rendu du 28 Mars 2022	22
6.2.3	Compte-rendu du 1er Avril 2022	23
6.2.4	Compte-rendu du 6 Avril 2022	24
6.2.5	Compte-rendu du 13 Avril 2022	26
6.2.6	Compte-rendu du 15 Avril 2022	28
6.2.7	Compte-rendu du 20 Avril 2022	29
6.2.8	Compte-rendu du 22 Avril 2022	30
6.2.9	Compte-rendu du 24 Avril 2022	30
6.2.10	Compte-rendu du 26 Avril 2022	31
6.2.11	Compte-rendu du 28 Avril 2022	32
6.2.12	Compte-rendu du 8 mai 2022	33
6.2.13	Compte-rendu du 13 mai 2022	34
6.2.14	Compte-rendu du 18 mai 2022	35
6.2.15	Compte-rendu du 21 mai 2022	36
6.2.16	Compte-rendu du 28 mai 2022	36
6.2.17	Compte-rendu du 5 juin 2022	37
7	Conception de WORDLE	39
7.1	Présentation de WORDLE	39
7.2	Présentation de l'application Web	39
7.3	Schéma de la base de données et modèle entité-association	39
7.3.1	Agencement des pages HTML et des routes	40
7.3.2	Implémentation du jeu	40
7.3.3	Maquette du site	40
7.3.4	Paramétrage du jeu et profil utilisateur	41
7.3.5	Solveur	41
8	Bibliographie	42

1 Introduction

1.1 Objectif du projet

L'objectif de ce projet est de recréer le jeu Wordle du New York Times tout en y ajoutant de nouvelles fonctionnalités, telles qu'un historique des parties, des succès à débloquer, des statistiques de jeu ainsi que la possibilité de paramétrer les parties en choisissant à la fois la longueur du mot à trouver et le nombre de tentatives autorisées. Cette application sera implémentée selon une architecture trois-tiers (Python, base de données et interface Web).

En parallèle de cette application, un solveur sera développé en C. Ce solveur aura évidemment pour objectif de résoudre le jeu le plus rapidement et efficacement possible. Il sera interactif : le joueur pourra y soumettre les mots qu'il a choisis au cours d'une partie et le solveur renverra alors le meilleur mot possible en fonction de ces informations. Le développement de ce solveur s'appuie notamment sur la théorie de l'information.

1.2 Règles du jeu de WORDLE

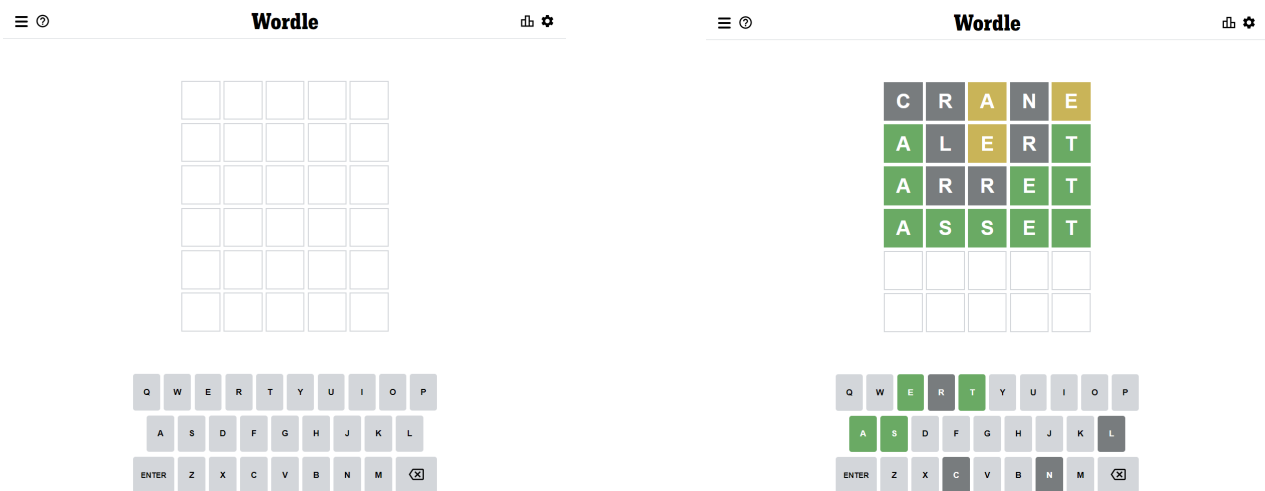


FIGURE 1 – Wordle - The New York Times

Le joueur a jusqu'à six tentatives pour deviner un mot de cinq lettres. Après chaque tentative, le jeu montre les lettres qui correspondent à la réponse ; si l'une des 5 lettres devinées apparaît dans la réponse et a également été placée au bon endroit, elle est mise en évidence en vert. Si la lettre apparaît dans la réponse mais n'a pas été placée au bon endroit, elle est sur-lignée en jaune. Si la lettre est complètement fausse, elle devient grise.

En bas de la page, le joueur peut voir un clavier QWERTY avec les lettres utilisées sur-lignées en vert, jaune ou gris en fonction des essais précédents. Le joueur peut seulement essayer des mots existants dans le dictionnaire de mots possibles, c'est-à-dire que le joueur ne peut pas rentrer des mots aléatoires ou inexistantes dans le but d'obtenir des indices plus rapidement (exemple : AEIOU en français). S'il essaye un mot qui n'est pas dans le dictionnaire, le jeu ne l'accepte pas et le joueur doit en proposer un autre.

2 Implémentation de WORDLE

2.1 La base de données

Notre base de données est constituée de 4 tables :

- Une table **Utilisateur** stockant les informations des joueurs. Ses colonnes sont : **idUser** qui est l'identifiant du joueur de type entier, son nom d'utilisateur **pseudo** de type str, un mot de passe **motDePasse** de type str et un sel **salt** pour la sécurisation du compte utilisateur. La chaîne de caractères stockée dans la colonne **motDePasse** sera le haché de la concaténation du mot de passe et du salt.
- Une table **Partie** qui stocke les données des parties jouées de chaque joueur connecté. Ses colonnes sont : **idPartie** qui représente l'identifiant d'une partie de type int, **idUser** qui est une clé étrangère référençant l'identifiant de l'utilisateur, les points d'expérience **pts_experience** de type int qui représentent les points gagnés lors d'une partie, le mot à deviner **motSecret** associé à la partie de type str, le nombre d'essais utilisés par le joueur pour trouver le mot secret **nb_coups** de type int, le nombre d'essais maximum de la partie **nb_essais_maximum** de type int, la longueur des mots associée au mot de la partie **longueur_mot** de type int et le mode de jeu défini par le joueur **modejeu** (mode sans fin ou mode classique).
- Une table **AchievementDesc** qui stocke les descriptions des achievements. Ses colonnes sont : **idSucces** qui est l'identifiant de l'achievement et **description** la description associée : gagner une partie en 3 coups ou moins, gagner 10 parties, etc. . .
- Une table **Achievements** qui relie les tables **AchievementDesc** et **Partie** dont les colonnes sont : **idPartie**, clé étrangère qui référence l'identifiant d'une partie et **idSucces**, clé étrangère référençant l'identifiant d'un achievement. Ces 2 clés forment la clé primaire de la table **Achievements**.

Notre base de données est en 3ème forme normale.

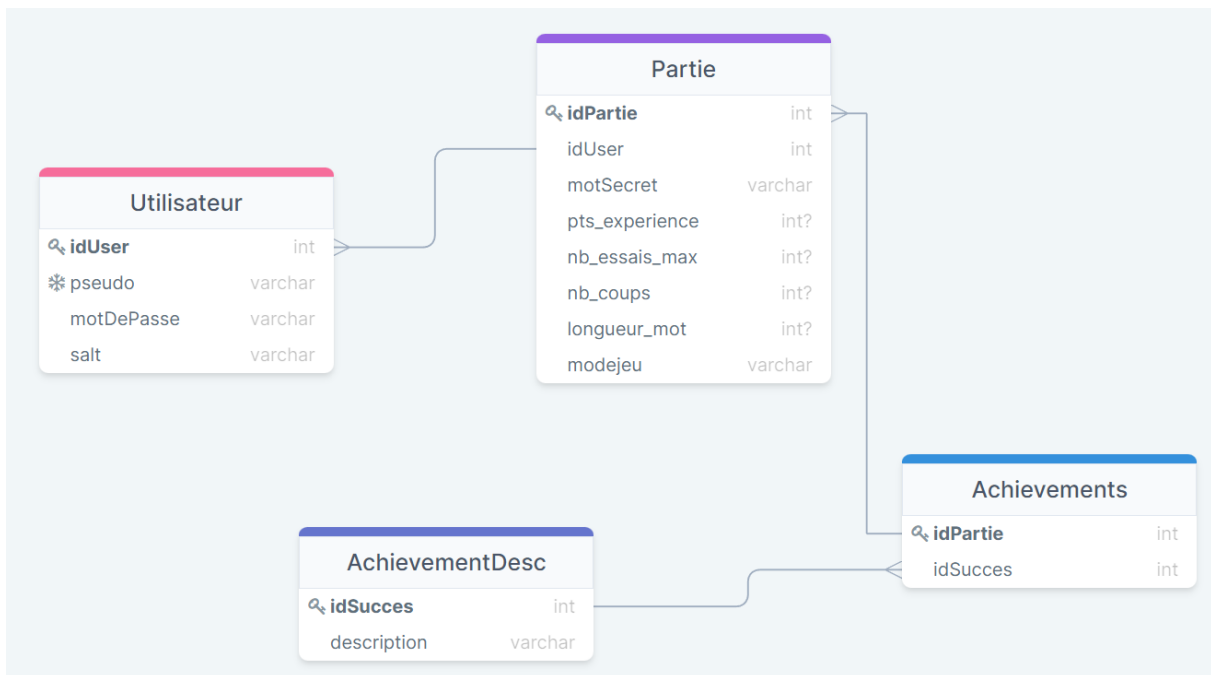


FIGURE 2 – Schéma de la base de données de WORDLE

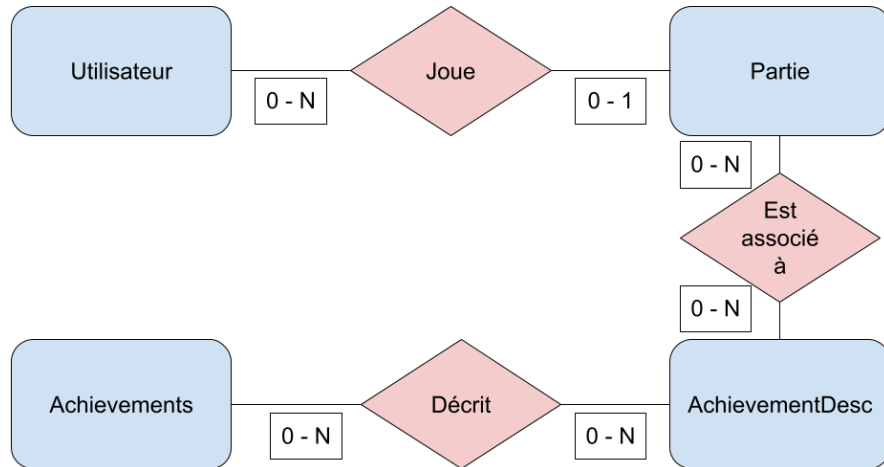


FIGURE 3 – Modèle entité-association de la base de données

2.2 L'interface Web

L'application a été développée en langage Python avec le framework Flask, permettant ainsi de créer et gérer des pages HTML dynamiques. Le fichier contrôleur, *app.py*, permet de rediriger l'utilisateur sur les bonnes routes en fonction de la position sur le site et de la validité de ses réponses aux différents formulaires. Ce fichier gère également les requêtes SQL avec la base de données *database.db*, notamment `INSERT INTO` et `SELECT`. Le jeu WORDLE a, quant à lui, été implémenté en JavaScript.

Ce tableau récapitule le fichier correspondant à chacune des différentes routes :

Route	Page HTML associée	Fonctionnalités
/	<i>index.html</i>	Jeu WORDLE
/register	<i>register.html</i>	Créer un compte
/login	<i>login.html</i>	Se connecter au jeu
/preference	<i>preference.html</i>	Modifier les paramètres du jeu
/profile	<i>profile.html</i>	Consulter son historique de jeu, ses succès et ses statistiques
/logout	Aucune	Se déconnecter
/recup/<userinfo>	Aucune	Récupérer les informations relatives à une partie depuis <i>index.js</i>

Dans la suite de cette partie du rapport (**Interface Web**), on confondra route et page HTML associée pour les 5 premières routes du tableau.

Jouer à WORDLE

La page d'accueil du site est *index.html*, sur la route /. Sur cette page, le joueur peut directement faire une partie et modifier les paramètres de jeu (longueur du mot, nombre de tentatives) s'il souhaite en cliquant sur le bouton **Paramètres de jeu** qui le redirigera sur la route */preference*. Le joueur peut également choisir le mode de jeu de WORDLE : soit classique (le jeu de base), soit sans fin. Dans ce second mode, il faut deviner le plus de mots possibles d'affilée sans se tromper sans quoi le jeu s'arrêtera.

À noter que ces paramètres sont stockés dans la variable globale *session*, ce qui permet même à un joueur non connecté de changer ces paramètres à sa guise. De plus, au lancement du jeu, la fonction *before_first_request()* s'exécute

et déconnecte automatiquement un joueur potentiellement connecté et réinitialise les paramètres de jeu dans le but d'éviter des bugs.

Laisser la possibilité au joueur de créer un compte ou non lui permet de jouer directement et ainsi rend le jeu plus attractif : des joueurs pourraient être repoussés à l'idée d'être forcés de créer un compte. C'est pourquoi il n'est pas obligatoire de créer un compte pour jouer. Cependant, les parties et leurs informations jouées par un joueur non connecté ne sont pas enregistrées, même si les seules informations demandées sont un pseudonyme et un mot de passe.

S'inscrire, se connecter, se déconnecter

Pour s'inscrire, il suffit de cliquer sur le bouton **S'inscrire** en haut à droite de la page d'accueil. Dans ce cas, le joueur est redirigé sur la route `/register` et on lui demande un pseudonyme et un mot de passe, qu'il doit ensuite confirmer. Ces deux informations doivent faire entre 8 et 20 caractères, dans le but de rendre l'application un peu plus sécurisée. Il est impossible de choisir un pseudonyme déjà pris par un autre joueur, et les deux mots de passe renseignés doivent être évidemment identiques.

En cas d'échec d'inscription pour les raisons évoquées ci-dessus, le joueur reste sur la même route et les champs du formulaire sont effacés, sinon les informations du joueur sont ajoutées à la table **Utilisateur** de la base de données, puis le joueur est redirigé sur la route `/login`. Pour plus de sécurité, le mot de passe est concaténé avec un salt de longueur 12 avant d'être haché avec la fonction *SHA256*, puis seulement enregistré.

Sur la page de connexion, il suffit de rentrer ses informations d'authentification pour se connecter. Une fois connecté, le pseudonyme du joueur s'affiche en haut à droite avec la mention "En ligne" (un joueur non connecté verrait la mention "Hors ligne" s'afficher). De plus, son id de joueur (selon la base de données) et son pseudonyme sont ajoutés à session, ce qui permet de savoir que ce joueur est connecté.

Pour se déconnecter, il suffit de cliquer sur le bouton **Se déconnecter** une fois connecté. Le joueur est alors redirigé sur la route `/logout`, où *session* est vidé de l'id du joueur et de son pseudonyme, puis le joueur est redirigé vers la route `/`.

Profil d'un joueur

Un joueur **connecté** a automatiquement accès à son profil via le bouton **Profil**, qui le redirige alors sur la route `/profile` (un joueur non connecté n'a pas accès à cette fonctionnalité). La page *profile.html* recense l'historique de ces parties, les succès débloqués, les statistiques de jeu et l'expérience du joueur. Dès que le joueur se rend sur cette route, toutes ces données sont actualisées grâce aux fonctions *getHistorique()*, *getStatistiques()*, *achQuests()* et *niveau()* respectivement. Aucune de ces fonctions ne prend l'ID du joueur en argument puisque ce dernier est stocké dans *session* avec la clé *idUser*.

Profil d'un joueur - [1] Historique

Pour chaque partie jouée, on enregistre :

- le mot à deviner ;
- sa longueur ;
- le nombre de tentatives autorisées ;
- le nombre de tentatives utilisées pour deviner le mot (-1 en cas de défaite) ;
- l'expérience gagnée à l'issue de la partie (0 en cas de défaite) ;
- le mode de jeu ("Classique" ou "Sans fin").

Ces informations sont récupérées depuis le fichier *dash.js* puis envoyées à *app.py* via la route `/recup/<userinfo>` avec une requête XML HTTP, dans un fichier *.json*. Ces données sont ensuite enregistrées dans la table **Partie**.

L'historique est récupéré en exécutant la fonction *getHistorique()*, puis est ensuite affiché de la partie la plus récente à la plus ancienne. Pour chacune d'entre elles, la couleur informe le joueur de sa victoire (vert) ou de sa défaite (rouge). Les informations enregistrées dans **Partie** sont également affichées pour plus de détails sur chaque partie. L'équipe projet a décidé de n'implémenter que le minimum demandé concernant l'historique pour se concentrer sur d'autres fonctionnalités détaillées plus loin.

Profil d'un joueur - [2] Achievements (succès)

En jouant au jeu, le joueur peut débloquer des achievements en réalisant certaines tâches, comme jouer ou gagner un certain nombre de parties ou en gagnant une partie en 3 essais ou moins. Sur la page *profile.html*, les achievements débloqués sont en vert tandis que ceux pas encore débloqués sont en violet.

Dès que le joueur se rend sur la route */profile*, on exécute la fonction *achQuests(idPartie)* qui récupère en premier lieu la liste des IDs des achievements déjà débloqués avec la fonction *getAchsAlreadyUnlocked()*. Ensuite, pour chacun d'entre eux, on exécute une requête SQL puis on analyse son résultat, dans le but de savoir si on délivre tel achievement au joueur ou non. Si la condition d'obtention de l'achievement est vérifiée et qu'il n'est pas encore débloqué, on exécute la requête SQL `INSERT INTO Achievements VALUES (idPartie, idSucces)` pour donner l'achievement *idSucces* au joueur. *idPartie* est l'ID de la dernière partie jouée par le joueur : ainsi, on s'assure d'effectuer les requêtes SQL sur l'ensemble des parties jouées par le joueur connecté.

Profil d'un joueur - [3] Statistiques

Sur la page *profile.html*, on retrouve un tableau recensant les statistiques de jeu du joueur : moyenne du nombre de coups pour les parties gagnées, nombre de parties jouées et taux de victoires. Les statistiques sont d'abord présentées pour l'ensemble des parties jouées, puis détaillées selon le nombre de lettres. On retrouve aussi la plus grande série de victoires (le plus grand nombre de parties gagnées d'affilée), calculé avec la fonction *getBestStreak()*.

Ces informations sont récupérées grâce à la fonction *getStatistiques()*, qui renvoie un tuple de longueur 6 contenant :

- la moyenne du nombre de coups pour toutes les parties gagnées ;
- une liste contenant la moyenne du nombre de coups pour les parties gagnées avec des mots de 4 à 12 lettres inclus ;
- le nombre de parties jouées ;
- une liste contenant le nombre de parties jouées avec des mots de 4 à 12 lettres inclus ;
- le taux de victoires en % ;
- une liste contenant le taux de victoires en % avec des mots de 4 à 12 lettres inclus.

Profil d'un joueur - [4] Expérience

L'expérience représente le temps passé sur le jeu par le joueur, ainsi que sa capacité à bien jouer. Lorsqu'un joueur crée un compte, il est au niveau 1 et il peut monter jusqu'au niveau 100. Perdre une partie rapporte 0 point d'expérience. Pour chaque partie gagnée, le joueur gagne de l'expérience selon la formule suivante :

$$expGagnee = (nbEssaisMax - nbEssaisUtilises) * 100 + longueurMot * 200 - nbEssaisMax * 20 \quad (1)$$

avec :

- *nbEssaisMax* : le nombre d'essais maximum autorisés pour cette partie ;
- *nbEssaisUtilises* : le nombre d'essais utilisés par le joueur pour deviner le mot ;
- *longueurMot* : la longueur du mot (en lettres).

Pour calculer le nombre de points d'expérience nécessaires pour passer au niveau suivant, on utilise la formule :

$$expNecessaire = \lceil 1.05 * x^{\frac{9}{4}} \rceil \quad (2)$$

où *x* représente le niveau à atteindre (entre 2 et 100).

2.3 Le solveur

Nous avons donc implémenté un solveur codé en C qui nous permettra à partir d'une liste de mots de pouvoir déterminer lequel d'entre eux nous donne le plus d'informations, afin de résoudre en un nombre minimal de coups le problème.

2.3.1 Le principe général

Tout d'abord lorsque l'on donne un mot au jeu WORDLE, ce dernier nous retourne une combinaison de couleurs comme expliqué ci-dessus. La probabilité d'avoir un mot qui a la même combinaison si on le donne au jeu dépend des lettres choisies : plus le mot contiendra de lettres fréquentes et plus la probabilité d'avoir un mot possédant la même combinaison est importante.

On souhaite savoir si un mot possède la même combinaison de couleurs si on l'envoie dans le jeu qu'un autre déjà envoyé dont on connaît la combinaison, pour cela on compare les 2 mots avec la combinaison du second et on regarde s'ils possèdent les mêmes lettres bien placées aux bons endroits, les mêmes lettres mal placées, etc...

Par exemple si nous rentrons le mot "tarie" dans le jeu, et qu'il nous renvoie une combinaison comme VGJGJ, nous savons que le "t" est bien placé, le "r" et le "e" sont mal placés et le "a" et le "i" n'existent pas.

Pour voir si un mot possède la même combinaison, il suffit de regarder s'il possède un "t" placé en première position, un "r" et un "e" à d'autres positions que celles qu'ils occupent dans le mot "tarie" et s'il ne possède pas les lettres "a" et "i". Ainsi le mot "taire" ne conviendrait pas alors que "torre" est compatible avec la combinaison donnée.

Une fois que le jeu nous donne la combinaison il n'est plus utile de considérer les mots qui, comparés au mot choisi ne possèdent pas la même combinaison que celle renvoyée par le jeu, ainsi nous pouvons les supprimer et nous réduisons de fait le champ des possibilités. Ainsi un mot avec peu de lettres communes réduit grandement le nombre de possibilités uniquement s'il possède des lettres bien placées, à l'inverse un mot avec beaucoup de lettres communes a une plus grande probabilité de réduire le champ des possibilités mais le réduit moins efficacement.

De cette manière, il faut choisir un mot avec des lettres suffisamment courantes pour nous donner le maximum de lettres bien placées ou mal placées mais pas trop non plus pour être sûr de réduire le plus possible le champ des possibilités.

Ainsi nous pouvons introduire la formule : $Q_{tinfo} = -p * \log_2(p)$ bits avec p la probabilité d'avoir un mot possédant la même combinaison de couleurs que celle obtenue.

Partant de cette formule, il suffit de choisir le mot qui donne le plus d'informations pour avancer le plus vite dans le jeu.

Maintenant que nous avons introduit ces notions, on introduit une formule qui détermine la quantité d'information moyenne que donne un mot, appelée **Entropie de Shannon**. En adaptant la formule à notre sujet, l'entropie d'un mot devient la somme sur toutes les combinaisons de couleurs possibles de la probabilité p définie ci-dessus multipliée par le logarithme en base 2 de $1/p$. Le solveur marche de la façon suivante : au début de la partie, le solveur va pour chaque mot déterminer son entropie et choisir le mot qui possède la plus grande entropie.

Ensuite, on rentre ce mot dans notre application Web et il en découle une combinaison de couleurs. Cette combinaison de couleurs est identifiée par une suite de chiffres faisant référence aux couleurs données par notre application : 2 pour la couleur verte, 1 pour la couleur jaune, 0 pour la couleur grise. Cette combinaison de couleurs va nous permettre de éliminer les mots du dictionnaire qui ne correspondent pas à cette combinaison de couleurs.

Nous avons maintenant une liste réduite qui contiendrait potentiellement le mot à deviner, nous recommençons ainsi l'opération sur cette liste et ainsi de suite jusqu'à trouver le mot à deviner.

2.3.2 Les structures de données

Nous avons décidé d'utiliser une table de hachage pour stocker notre dictionnaire (l'ensemble des mots que le joueur peut proposer au jeu WORDLE).

Une table de hachage est une liste de listes chaînées pouvant être identifiées par un nombre unique pour faciliter les recherches d'éléments dans les tables.

Ainsi notre structure de données se base sur les listes chaînées, nous avons pour cela implémenté des fonctions pour les créer, les modifier ou y extraire des informations. Aussi tous les tests ont été effectués dans le fichier "test_list.c".

Après avoir créé des fonctions pour les listes, nous en avons conçu d'autres pour les tables en nous appuyant sur ces dernières. Ainsi nous pouvons modifier des tables ou y extraire des informations. Pour les tableaux un nombre de listes *size* est requis, plus cette taille est grande et plus la recherche d'éléments peut être effectuée rapidement. Aussi nous avons commencé par stocker notre dictionnaire dans des tables d'une taille de 300 cellules avant de passer sur des tailles de plusieurs milliers. Les fonctions des tables ont toutes été testées dans le fichier "test_dico.c".

2.3.3 Les Fonctions

Pour réaliser notre solveur nous avons dû faire appel à plusieurs fonctions. Nous avons d'abord créé une fonction appelée *dico_load()* qui permet de créer le dictionnaire depuis un fichier texte contenant tous les mots. Ensuite le solveur cherche le meilleur mot, pour cela il utilise la fonction *wordEntropy* qui calcule l'entropie des mots, elle fonctionne grâce à la fonction *evaluation* qui à partir de 2 mots retourne leur combinaison. Enfin une fonction *getBestWord()* réutilise les 2 fonctions vues précédemment pour donner le meilleur mot au sens de l'entropie.

Toutes ces fonctions s'appuient sur d'autres mineurs tels que *findWordsGreen()* et *findWordsYellow()* qui recherchent tous les mots respectivement avec une lettre bien placée ou une lettre mal placée ; ou encore *motsSuivantsPossibles()* qui ne renvoie que les mots valables pour une combinaison donnée.

3 Tests et performances

Nous avons principalement tester que la fonction d'évaluation au niveau de l'application web et les fonctions utilisées dans notre solveur.

3.1 Fonction d'évaluation de l'application web

3.1.1 Complexité théorique

Soit n la longueur du mot choisi par le joueur ; Soit $k > 0$ est une constante. Cette fonction a une complexité temporelle en $O(k * n)$. La fonction d'évaluation a donc une complexité temporelle linéaire.

3.1.2 Graphe de performance

On sait que, la complexité théorique de la fonction d'évaluation est linéaire. Sur cette figure qui mesure le temps d'exécution en fonction de la longueur des mots ; On peut noter que le temps d'exécution croît linéairement avec une longueur croissante des mots. On peut donc confirmer la complexité théorique de cette fonction qui linéaire.

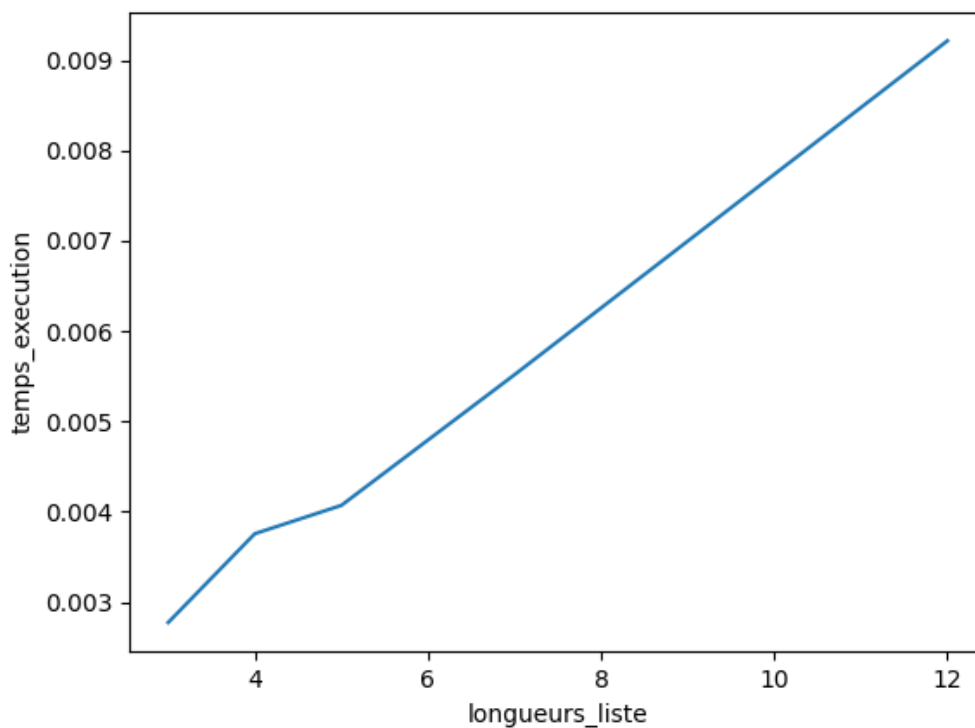


FIGURE 4 – Mesures du temps exécution en fonction de la longueur des mots

3.2 Fonction dicoload du solveur

3.2.1 Complexité théorique

Soit p le nombre de mots du dictionnaire choisi, Cette fonction a une complexité temporelle en $o(p)$. La fonction dicoload du solveur a donc une complexité temporelle linéaire.

3.3 Fonction evaluation du solveur

3.3.1 Complexité théorique

Soit n la longueur des mots du dictionnaire choisi. Cette fonction a une complexité temporelle en $o(n^2)$. La fonction evaluation du solveur a donc une complexité temporelle quadratique.

3.4 Fonction WordEntropy du solveur

3.4.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p * n^2)$.

3.5 Fonction getbestword du solveur

3.5.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p^2 * n^2)$.

3.6 Fonction FindWordsGreen du solveur

3.6.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p * n)$.

3.7 Fonction FindWordsYellow du solveur

3.7.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p * n^2)$.

3.8 Fonction excludeWords du solveur

3.8.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p * n^2)$.

3.9 Fonction MotsSuivantsPossibles du solveur

3.9.1 Complexité théorique

Soit p le nombre de mots du dictionnaire et n la longueur des mots ; Cette fonction a une complexité temporelle en $o(p * n^3)$.

4 Gestion de projet

4.1 Membres de l'équipe

L'équipe projet est composée de 4 élèves : Armand LONDEIX, Lucas RIOUX, Patrick O'BRIEN et Serge TÉHÉ. Lucas est désigné chef de projet et aura la responsabilité de s'assurer de l'avancement global du projet.

4.2 Organisation du projet

Le projet s'est déroulé du 23 mars 2022 au 8 juin 2022. L'équipe projet s'est réunie principalement sur Discord et à TELECOM Nancy pour les réunions, qui se tenaient généralement en début d'après-midi. Le projet a été scindé en deux sous-projets : d'une part, le **jeu WORDLE** (application Web, interface graphique, base de données) et d'autre part le **solveur**.

4.3 Outils de travail

Pour réaliser notre projet, nous avons développé avec l'éditeur de code Visual Studio Code. Nous avons aussi utilisé GanttProject pour réaliser le diagramme de Gantt, Inkscape pour faire une maquette de l'interface graphique et des outils de bureautique tels qu'Excel pour la matrice RACI. Le rapport a été rédigé en \LaTeX . Le travail a été mis en commun sur le gitlab de l'école.

4.4 Documents relatifs au projet

Différents documents de gestion de projet ont été réalisés, en plus des comptes-rendus, afin de retracer son déroulement : une matrice SWOT afin d'analyser les risques du projet, un WBS pour séquencer les tâches, une matrice RACI attribuant les différentes tâches aux membres de l'équipe et un diagramme de Gantt rendant compte de l'avancement du projet. Tous ces documents sont en annexe.

4.5 Volume horaire

Temps de travail	Tâche	Amand	Lucas	Patrick	Serge
Application Python	Fonction d'évaluation d'un mot	3	2	1	7
	Gérer les paramètres d'une partie	4	3	0	2
	Permettre aux utilisateurs de créer un compte	2	3	0	0
	Gérer le profil d'un joueur (historique, statistiques, succès)	3	8	8	0
	Créer les succès	1	1	3	0
	Système d'expérience	2	1	1	1
	Mode de jeu : partie classique	4	3	1	3
Base de données	Mode de jeu : sans fin	0	0	0	7
	Créer les différentes tables	1	1	1	1
	Relier la BD au site	2	1	0	5
	Mettre la BD en 3ème forme normale	0	0	0	3
Développement Web	Modèle entité-association	0	0	0	2
	Page d'inscription	1	2	0	0
	Page de connexion	1	2	0	0
	Page de jeu	4	1	0	1
	Profil d'un joueur - Historique	5	1	1	0
	Profil d'un joueur - Statistiques	4	2	1	0
	Profil d'un joueur - Succès	4	3	5	0
	Profil d'un joueur - Expérience	1	1	1	1
	CSS : mise en page globale	0	7	1	3
Gestion de projet	JavaScript : grille du jeu	0	1	0	20
	Rédaction du rapport	5	7	6	6
	Rédaction du document de conception	2	3	2	2
	Maquette du site	1	2	0	0
	Matrice SWOT	0	0.5	1	0.5
	WBS	0	2	0	0
	Matrice RACI	0	2	0.5	0.5
	Diagramme de Gantt	0	3	1	0
	Compte-rendu 23/03	0	0.5	0.5	0
	Compte-rendu 28/03	0	0.5	0	0
	Compte-rendu 01/04	0.5	0	0	0
	Compte-rendu 06/04	0	0	0	0.5
	Compte-rendu 13/04	0	0.5	0	0
	Compte-rendu 15/04	0.5	0	0	0
	Compte-rendu 20/04	0	0	0.5	0
	Compte-rendu 22/04	0	0	0.5	0
	Compte-rendu 24/04	0	0	0	0.5
	Compte-rendu 26/04	0.5	0	0	0
	Compte-rendu 28/04	0	0	0.5	0
	Compte-rendu 08/05	0	0.5	0	0
	Compte-rendu 13/05	0.5	0	0	0
	Compte-rendu 18/05	0	0	0.5	0
	Compte-rendu 21/05	0	0	0	0.5
	Compte-rendu 28/05	0	0.5	0	0
	Compte-rendu 05/06	0.5	0	0	0
Solveur	Choix et élaboration de la structure de données	8	5	2	6
	Lecture du nombre de lettres au lancement du solveur	2	2	0	0.5
	Garder les mots suivants possibles au fil des itérations	1	4	0	11
	Calculs d'entropie (proposer le meilleur mot possible)	8	3	2	3
	Déroulement du solveur dans le terminal	2	2	3	3
Total	Tests	1	1	1	2
		72	79	42	89

FIGURE 5 – Volume horaire

5 Bilan du projet

5.1 Travail réalisé

Nous avons réussi à développer un jeu WORDLE paramétrable par le joueur (nombre d'essais, taille du mot, mode de jeu) avec un historique des parties et des fonctionnalités secondaires (succès, statistiques, expérience). Pour ce livrable, nous n'avons pas eu assez de temps pour créer un mode contre-la-montre, où le joueur aurait du deviner le maximum de mots en un temps donné.

Le solveur est fonctionnel et réduit très vite l'ensemble des mots possibles pour converger vers la réponse : pour des mots comprenant de 4 à 8 lettres, la réponse est trouvée en 3 coups environ, et même en 2 coups pour des mots plus longs. Les livrables sont donc conformes aux attendus du projet.

5.2 Bilan de l'équipe projet

Points positifs	- Implémentation des fonctions principales - Renforcement des compétences de gestion de projet
Points à améliorer	- Améliorer la communication au sein du groupe - Ne pas se concentrer sur les détails d'implémentation - Être plus efficace pendant les réunions
Expérience collective	- Bonne impulsion de groupe - Utilisation de Gitlab

5.3 Bilan individuel

5.3.1 Armand Londeix

Dans la première partie du projet, je me suis impliqué dans l'interface Web, ainsi que dans la mise en place de certaines fonctions python, aussi cette partie m'a permis de consolider mes connaissances en HTML mais aussi de en m'impliquant dans la gestion du projet de m'améliorer en travail d'équipe, désormais je sais mieux comment renforcer l'organisation, la cohésion d'une équipe.

Dans la partie Solveur, le fait de coder en C m'a permis d'apprendre et de consolider mes connaissances dans un langage qui m'était nouveau, aussi je sais maintenant très bien utiliser les structures de données et coder en C n'est plus vraiment un problème. La collaboration avec mes camarades m'a permis de ne pas m'attarder sur certaines parties qui me posaient problèmes au début pour progresser dans mon travail.

Je garde un bon souvenir de ce projet tant sur le plan des connaissances que j'ai acquises que sur la gestion de projet et les relations avec mes camarades.

5.3.2 Patrick O'Brien

Ce projet m'a permis de développer ma programmation en algorithmes et en Web. J'ai appris à bien travailler en équipe. J'ai eu quelques problèmes dus à une mauvaise communication de ma part mais j'ai vu cela comme une opportunité de créer plus de cohésion dans le groupe. Il était très facile de travailler avec mes camarades et ils ont été très utiles pour résoudre les difficultés que j'ai rencontrées. En ce qui concerne le gestion de projet, je comprend bien l'importance d'une réunion efficace.

5.3.3 Lucas Rioux

Ce projet m'a permis d'améliorer mes compétences en Python et en développement Web, notamment en CSS et en JavaScript ; et également de découvrir le langage C, qui m'a forcé à me montrer plus rigoureux. Certaines fonctions du projet m'ont mis à rude épreuve, comme la fonction éliminant les réponses impossibles dans le solveur. J'ai également pu mettre en pratique mes connaissances en gestion de projet et je commence à comprendre comment travailler en équipe. Dans l'ensemble, je suis satisfait des autres membres de l'équipe ainsi que du travail effectué.

5.3.4 Serge Téhé

Ce projet m'a permis d'acquérir des connaissances plus poussées en algorithmique, en programmation Web et en langage C. J'ai pu développer tout au long du projet certaines aptitudes et réflexes en informatique et en gestion de projet qui me donnent ainsi une expérience de plus sur mon CV. Ce projet m'a aussi permis d'écouter les autres membres du groupe et de trouver un point commun de convergence des idées malgré nos divergences souvent très ostentatoires.

6 Annexe

6.1 Documents de gestion de projet

Matrice SWOT

Matrice SWOT	Positif	Négatif
Origine interne	<ul style="list-style-type: none">- Compétences en programmation (Python, HTML, Flask), en LaTeX et en base de données- Expérience quant à l'utilisation des outils de GdP- Serge est expérimenté en C- Lucas et Armand ont fait le PPII S5 ensemble	<ul style="list-style-type: none">- Méconnaissance du CSS et du JavaScript- Seul Serge est à l'aise en C- Personne ne sait utiliser de logiciel de montage photo afin de créer une maquette du site
Origine externe	<ul style="list-style-type: none">- Les règles de WORDLE sont simples- Les vacances de Pâques nous permettront d'avancer vite	<ul style="list-style-type: none">- Les membres de l'équipe projet ne sont pas dans les mêmes groupes de TD- Partiels

FIGURE 6 – Matrice SWOT

WBS

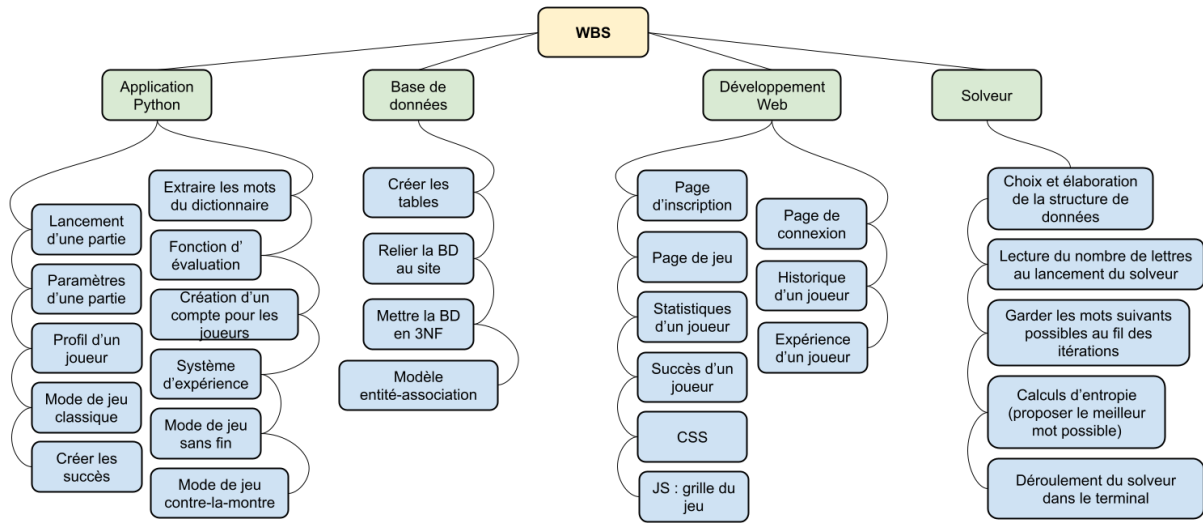


FIGURE 7 – WBS

Matrice RACI

Matrice RACI	Tâche	Armand Londeix			Patrick O'Brien			Lucas Rioux			Serge Téhé		
Application Python	Fonction d'évaluation d'un mot	R						R			R	A	
	Gérer les paramètres d'une partie	R									R	A	
	Permettre aux utilisateurs de créer un compte							R	A				
	Gérer le profil d'un joueur (historique, statistiques, succès)	R	A					R					
	Créer les succès				R	A		R					
	Système d'expérience	R											
	Mode de jeu : partie classique	R			R			R			R	A	
Base de données	Mode de jeu : sans fin										R	A	
	Créer les différentes tables	R			R			R			R	A	
	Relier la BD au site	R			R			R			R		
	Mettre la BD en 3ème forme normale										R	A	
Développement Web	Modèle entité-association										R	A	
	Page d'inscription	R						R	A				
	Page de connexion	R						R	A				
	Page de jeu	R									R	A	
	Profil d'un joueur - Historique	R	A					R					
	Profil d'un joueur - Statistiques	R	A					R					
	Profil d'un joueur - Succès	R			R	A		R					
	Profil d'un joueur - Expérience	R						R	A				
	CSS : mise en page globale							R	A				
	JavaScript : grille du jeu										R	A	
Gestion de projet	Rédaction du rapport	R			R			R	A		R		
	Rédaction du document de conception	R			R			R	A		R		
	Maquette du site	R						R	A				
	Matrice SWOT				R	A							
	WBS							R	A				
	Matrice RACI			I			I	R	A		I		I
	Diagramme de Gantt				R			R					
	Compte-rendu 23/03			I			I	R	A				I
	Compte-rendu 28/03			I			I	R	A				I
	Compte-rendu 01/04	R	A								I		I
	Compte-rendu 06/04			I							I	R	A
	Compte-rendu 13/04			I				I	R	A			I
	Compte-rendu 15/04	R	A								I		I
	Compte-rendu 20/04			I	R	A					I		I
	Compte-rendu 22/04			I	R	A					I		I
	Compte-rendu 24/04			I							I	R	A
	Compte-rendu 26/04	R	A								I		I
	Compte-rendu 28/04			I	R	A					I		I
	Compte-rendu 08/05			I				I	R	A			I
	Compte-rendu 13/05	R	A								I		I
	Compte-rendu 18/05			I	R	A					I		I
	Compte-rendu 21/05			I							I	R	A
	Compte-rendu 28/05			I				I	R	A			I
	Compte-rendu 05/06	R	A								I		I
Solveur	Choix et élaboration de la structure de données	R			R			R			R	A	
	Lecture du nombre de lettres au lancement du solveur	R						R	A				
	Garder les mots suivants possibles au fil des itérations	R	A					R			R		
	Calculs d'entropie (proposer le meilleur mot possible)	R						R			R	A	
	Déroulement du solveur dans le terminal							R	A				
	Tests	R			R			R			R	A	

FIGURE 8 – Matrice RACI

Diagramme de Gantt

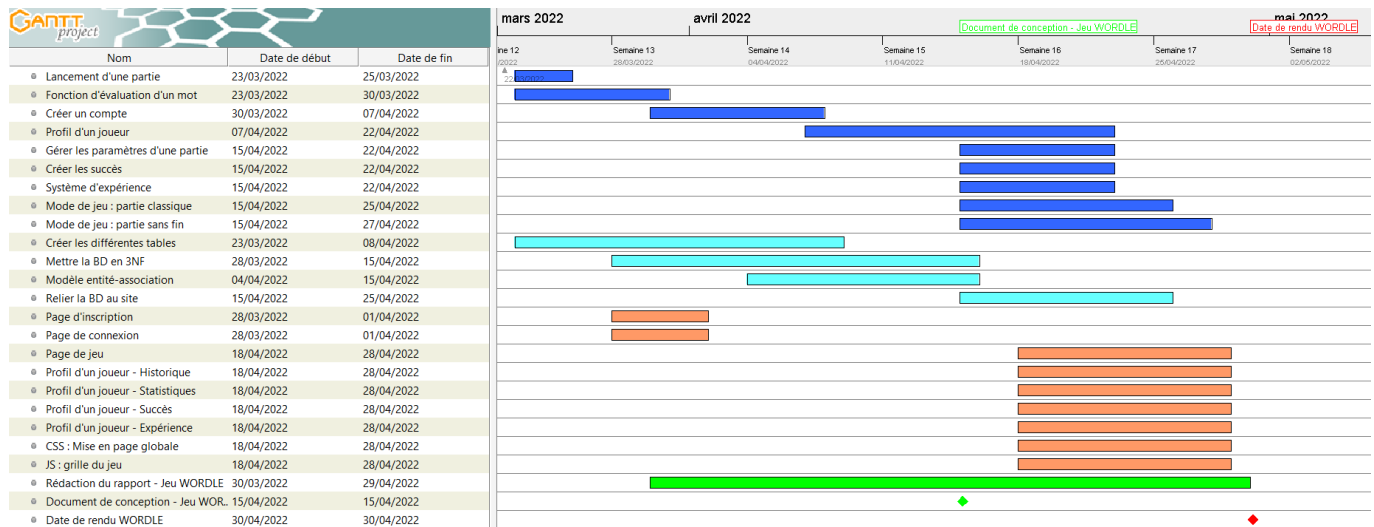


FIGURE 9 – Diagramme de Gantt - Jeu WORDLE

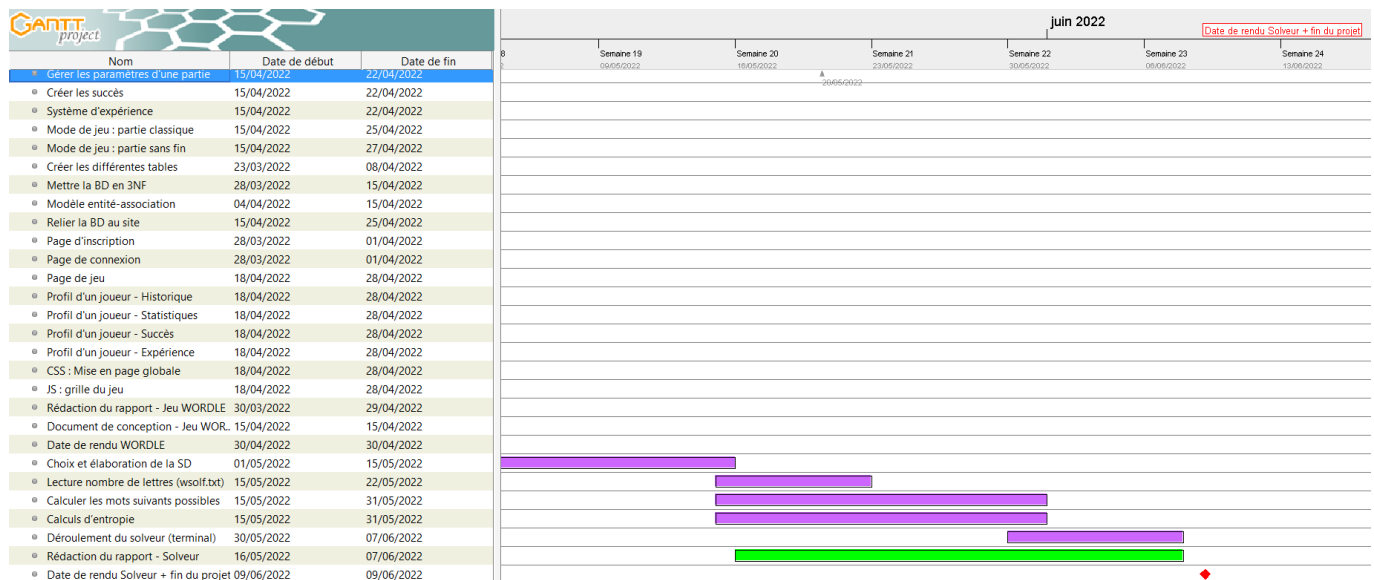


FIGURE 10 – Diagramme de Gantt - Solveur

6.2 Comptes-rendus

6.2.1 Compte-rendu du 23 Mars 2022

Type de réunion	Réunion d'avancement
Lieu	Telecom Nancy
Horaires	16h00 - 17h30
Maître de séance	Lucas Rioux
Secrétaires	Lucas Rioux, Patrick O'Brien
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Atouts et faiblesses des membres du groupe
2. Début du site : premières directives
3. Éléments de gestion de projet

Avant le début de la réunion, le groupe a créé un groupe Messenger, un serveur Discord pour pouvoir communiquer, un Google Drive pour partager les documents relatifs au projet et un document Overleaf pour rédiger le rapport.

1. Atouts et faiblesses des membres du groupe

Lucas prend la parole et questionne les membres du groupe sur leurs compétences en matière de programmation (Python, SQL, HTML, CSS, Flask, C). Tous les membres du groupe affirment maîtriser Python, SQL, HTML et Flask mais moins le CSS. Concernant le C, seul Serge est expérimenté. Serge a d'ailleurs plus d'expérience avec l'utilisation de git car un module y était entièrement consacré l'année scolaire précédente.

Lucas propose d'utiliser les branches de git pour collaborer ensemble sans gêne et ne pas devoir prendre le risque de perdre du travail en commitant directement sur la branche master. Les autres membres du groupe sont d'accord avec cette idée, Lucas propose que tout le monde se renseigne sur ce point d'ici la prochaine réunion (cf. to-do list).

2. Début du site : premières directives

Il est décidé que l'application sera implémentée grâce à la bibliothèque Flask de Python. À première vue, seulement deux pages HTML seront nécessaires : une page de connexion et une page avec le jeu. Patrick évoque la question de la gestion des caractères spéciaux, Lucas précise que ce problème est facilement gérable en Python.

Vient ensuite la question de la récupération des mots du dictionnaire français : ils ont été extraits de la base de données Lexique (version 3.83) (<http://www.lexique.org/>) par Lucas, répertoriant l'intégralité des mots français. L'ensemble des mots a été extrait vers un fichier texte, puis les doublons ont été supprimés. Armand a, quant à lui, récupéré des fichiers texte sur le github de ScienceEtonnante avec des mots déjà sélectionnés et séparés dans différents fichiers selon leur longueur. Il sera décidé plus tard de quel dictionnaire utiliser.

3. Éléments de gestion de projet

Les éléments de gestion de projet à rédiger dès maintenant sont la matrice SWOT et le diagramme de Gantt. Lucas propose aux membres du groupe de télécharger le logiciel gratuit Gantt Project permettant une création et mise à jour facile du diagramme de Gantt. Patrick se désigne pour réaliser la matrice SWOT et le début du diagramme de Gantt pour la prochaine réunion.

TO-DO LIST

Armand Londeix	- Se renseigner sur l'utilisation des branches de git - Mettre en place l'environnement (Flask), implémenter le début de l'application
Patrick O'Brien	- Se renseigner sur l'utilisation des branches de git - Rédiger la matrice SWOT - Commencer le diagramme de Gantt
Lucas Rioux	- Se renseigner sur l'utilisation des branches de git - Extraire les mots de la base de données Lexique, en supprimer les doublons
Serge Téhé	- Se renseigner sur l'utilisation des branches de git - Extraire les mots de la base de données Lexique, en supprimer les doublons

Prochaine réunion : Lundi 28 Mars à 13h00.

6.2.2 Compte-rendu du 28 Mars 2022

Type de réunion	Réunion d'avancement
Lieu	Telecom Nancy
Horaires	13h00 - 14h00
Maître de séance	Lucas Rioux
Secrétaire	Lucas Rioux
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Avancement depuis la dernière réunion

2. Implémentation de l'application

Les membres du groupe ont tous téléchargé GanttProject en amont de la réunion. GanttProject est un logiciel gratuit permettant de créer un diagramme de Gantt.

1. Avancement depuis la dernière réunion

Depuis la précédente réunion, Lucas a obtenu des renseignements concernant d'autres bases de données contenant l'ensemble des mots du dictionnaire français, et notamment l'Officiel du Scrabble (ODS 6) avec presque 400 000 mots. L'équipe projet ne s'est pas encore prononcée sur la question du dictionnaire utilisé. Patrick a mis en place le début du diagramme de Gantt et les membres se sont renseignés sur l'utilisation des branches de git.

2. Implémentation de l'application

Il s'ensuit un débat pour savoir s'il faut ranger les mots dans un seul fichier texte, différents fichiers texte selon leur longueur ou bien dans une base de données (une table plus précisément). Serge et Armand sont partants pour préalablement trier les mots selon leur longueur dans un souci de gain de temps de calcul, ce sur quoi le groupe s'accorde. En ce qui concerne la base de données, elle comporte pour l'instant deux tables :

User	Historique
idUser INTEGER, PRIMARY KEY	idPartie INTEGER, PRIMARY KEY
pseudo VARCHAR(20), UNIQUE	idUser INTEGER, FOREIGN KEY REFERENCES User(idUser)
motDePasse VARCHAR(20)	motSecret VARCHAR(15)
	nbTentatives INTEGER

Lucas a rappelé que la base de données doit être en 3ème forme normale, et rappelle les règles (qui sont dans le cours). Pour l’instant, le groupe se demande s’il faut enregistrer uniquement la solution d’une partie et le nombre de coups joués pour gagner, ou bien s’il faut aussi sauvegarder l’ensemble des mots joués au cours d’une partie. Pour Armand, élaborer la base de données n’est pas une priorité : il faut se concentrer sur le développement Web. Plusieurs pages HTML sont envisagées :

- index, la page d’accueil du site
- register et login, pour s’inscrire et se connecter,
- game, la page avec le jeu,
- history, pour voir l’historique de nos parties,
- success, pour voir nos succès dans le jeu.

Cette liste est évidemment non exhaustive. Aussi, un utilisateur devra impérativement être connecté pour jouer : cette fonctionnalité peut être implémentée avec les bibliothèques Flask-Session et Flask-Login.

TO-DO LIST

Armand Londeix	- Implémenter la page avec le jeu
Patrick O’Brien	- Réaliser une maquette du site sur Inkscape - Commencer à rédiger le document de conception
Lucas Rioux	- Réaliser une maquette du site sur Inkscape
Serge Téhé	- Implémenter la page avec le jeu

Prochaine réunion : Vendredi 1er Avril à 13h00.

6.2.3 Compte-rendu du 1er Avril 2022

Type de réunion	Réunion d’avancement
Lieu	Telecom Nancy
Horaires	13h00 - 14h00
Maître de séance	Lucas Rioux
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Patrick O’Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Avancement depuis la dernière réunion et suppression de fichiers inutiles sur le git
2. Élaboration de la base de données

1. Avancement depuis la dernière réunion et suppression de fichiers inutiles sur le git

On compte prendre un nouveau dictionnaire, soit on garde celui d'Armand qui a l'avantage d'être trié par fréquence et constitue une grosse base de données, soit on prend celui de Lucas qui contient davantage de mots. De toute façon le changement du dictionnaire est relativement aisé et on pourra toujours revenir dessus. On décide donc de garder le dictionnaire d'Armand et de changer au cas où.

2. Élaboration de la base de données

On parle de la façon dont on organise le site, Lucas propose de créer un compte avant de pouvoir jouer, Armand pense qu'on peut jouer sans compte mais que dans ce cas là les achievements, les enregistrements de parties et autres fonctionnalités ne fonctionneraient pas, on va rester sur cette idée.

Lucas pense qu'il faut supprimer la page succes qui ne sert pas à grand chose mais qui ne compromet pas le bon fonctionnement du site, il est décidé de revenir à ça plus tard.

Pour tous se mettre d'accord, on fait sur un tableau le schéma du site ainsi que celui des bases de données.

Pour la page de connexion Lucas propose de mettre juste un pseudo non pris, après réflexion on se demande s'il ne faut pas un mot de passe car certains joueurs pourraient jouer sur le compte d'autres.

Pour la base de données, on créera 3 tables, une table utilisateur avec les noms d'utilisateur et les mots de passe et une table historique (utilisateur, nombre d'essai, mots utilisés,...), enfin une dernière table permettra de stocker les succès (victoires, victoires en moins de coups,...).

TO-DO LIST

Armand Londeix	- Continue la construction du site - Page achievement, historique - Base de données
Patrick O'Brien	- Document de conception
Lucas Rioux	- Mettre à jour la matrice RACI
Serge Téhé	- Gérer la fin du jeu - evaluation de chaque lettre dans un mot pour en associer des couleurs

Prochaine réunion : Mercredi 6 Avril à 12h30.

6.2.4 Compte-rendu du 6 Avril 2022

Type de réunion	Réunion d'avancement
Lieu	Telecom Nancy
Horaires	12h30 - 13h20
Maître de séance	Serge Téhé
Secrétaire	Serge Téhé
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Architecture du site
2. Élaboration de la base de données
3. Présentation du jeu Wordle par Serge Téhé

1. Architecture du site

Au début de la réunion, nous avons commencé par mettre en œuvre l'architecture de notre site Web. Lucas s'est mis au tableau pour écrire toutes les propositions des membres du groupe et nous nous sommes mis d'accord sur plusieurs points.

L'utilisateur non connecté arrivant sur la page d'accueil du site voit le jeu Wordle auquel il peut directement jouer sans être inscrit au préalable. Sur cette même page, cet utilisateur non connecté aperçoit au niveau de la barre de navigation principale un paramétrage pour paramétrer la longueur des mots et le nombre d'essais maximum puis 2 boutons d'inscription et de connexion. Ces boutons d'inscription et de connexion servent à enregistrer un joueur de sorte à ce qu'il puisse voir son historique de parties jouées et les niveaux qu'il a débloqués (hard,medium,easy).

L'utilisateur connecté quant à lui peut voir son profil avec les niveaux débloqués ainsi que des boutons de déconnexion, de statistiques sur les anciennes parties jouées et aussi de paramétrage du jeu.

Certains membres du groupe ont proposé d'autres fonctionnalités telles que le mot du jour, endless, contre-la -montre que nous implémenterons si nous avons le temps.

2. Élaboration de la base de données

La base de données permet d'enregistrer les données des utilisateurs connectés. Nous avons convenu la création de 3 tables :

User	Historique	Succès
idUser INTEGER PK pseudo STR motDePasse STR	idPartie INTEGER PK idUser INTEGER FK nb_essais STR longueurMot INTEGER	idSucces INTEGER PK idUser INTEGER FK estObtenu BOOLEAN

Cette base de données ressort de nos propositions, nous la normaliserons éventuellement en 3ème forme normale.

3. Présentation du jeu Wordle par Serge Téhé

Nous n'avons pas eu le temps pour cette partie, Serge devait présenter et expliquer son implémentation du jeu Wordle.

TO-DO LIST

Armand Londeix	- Rédaction du document de conception
Patrick O'Brien	- Rédaction du document de conception - Présentation textuelle du jeu Wordle
Lucas Rioux	- Maquette du site - WBS - Rédaction du document de conception
Serge Téhé	- Normalisation de la base de données - Rédaction du document de conception

Prochaine réunion : Mercredi 13 Avril à 14h00.

6.2.5 Compte-rendu du 13 Avril 2022

Type de réunion	Réunion d'avancement
Lieu	Discord
Horaires	14h00 - 15h00
Maître de séance	Lucas Rioux
Secrétaire	Lucas Rioux
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Avancement depuis la dernière réunion
2. Fonction `evaluation()`, jeu WORDLE : clarification
3. Base de données de l'application
4. Mise en place du CSS

1. Avancement depuis la dernière réunion

Au début de la réunion, il a été décidé à 3 votes contre un de supprimer la route `/dico` ainsi que la page `dico.html`, jugés inutiles à l'application. Ensuite, chacun des membres de l'équipe projet fait un bilan sur ce qui a été réalisé depuis la dernière réunion :

- Armand a commencé à développer la page `achievements.html` renommée ensuite en `profile.html`. Il a rencontré certains problèmes avec notamment un module non installé (`pytest`) qui ne lui permettait pas de lancer l'application via `flask run`, et aussi avec le dictionnaire `session`. Ces problèmes ont été résolus.
- Patrick a rédigé une partie du document de conception concernant la base de données et le site HTML : Lucas lui fait remarquer qu'il devait s'occuper de la présentation textuelle du jeu, ce que Patrick note et fera pour la prochaine réunion. Il rencontre des soucis avec la commande `git pull`, a priori résolus.
- Serge s'est chargé de concevoir et de normaliser en 3ème forme normale la base de données. Il s'occupera ensuite du modèle entité-association.
- Lucas a réalisé une première maquette du site Web, un schéma de ses routes ainsi qu'un schéma de son architecture. Il a aussi fait le WBS et ajouté le README sur le git.

2. Fonction `evaluation()`, jeu WORDLE : clarification

Lucas a retiré sa fonction `evaluation()` pour laisser celle d'Armand, qui avait implémenté cette fonctionnalité le premier.

Vient ensuite une question cruciale : quelle application utiliser ? Armand,

Patrick et Lucas ont, pour le début du développement, mis en place un formulaire tandis que Serge a développé l'entièreté du jeu de son côté (fonction d'évaluation, mise en forme) en JavaScript, langage que seul lui maîtrise. Armand propose à Serge d'essayer de "remplacer" le code Python sur la route `/` en JavaScript, ou d'essayer d'établir une connexion entre les deux fichiers afin de conserver les fonctions déjà développées. Serge lui répond que c'est impossible.

Concernant le mot secret, le problème ne se pose pas puisqu'il est déclaré dans `app.py`. Afin de relier le fichier JavaScript et la base de données, Serge propose de faire du web scraping. Serge accepte de se charger de la connexion entre le fichier JavaScript et la base de données.

3. Base de données de l'application

Serge nous présente son travail sur la base de données du site. Suite à des discussions ayant eu lieu après la réunion, la base de données retenue est la suivante :



FIGURE 11 – Schéma de la base de données de WORDLE

4. Mise en place du CSS

Enfin, Lucas rappelle au groupe qu'il faudrait commencer à mettre en forme le site. Armand veut bien commencer à s'en charger.

TO-DO LIST

Armand Londeix	- Rédaction du document de conception
Patrick O'Brien	- Rédaction du document de conception - Présentation textuelle du jeu Wordle
Lucas Rioux	- Maquette du site - WBS - Rédaction du document de conception
Serge Téhé	- Normalisation de la base de données - Rédaction du document de conception

Prochaine réunion : Vendredi 15 Avril à 10h00.

6.2.6 Compte-rendu du 15 Avril 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	14h00 - 14h30
Maître de séance	Lucas Rioux
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Lucas Rioux Serge Téhé
Membres absents	Patrick O'Brien

Ordre du jour

1. Ce qui a été fait
2. Validation du document de conception

1. Ce qui a été fait

Armand nous explique que le site a été mis à jour, désormais il est possible de modifier les paramètres de jeu, aussi un clic sur le bouton home permet de remettre à zéro la partie, enfin la page statistique est au point.

Serge et Lucas se sont concentrés sur la base de données et le document de conception, ils conseillent à Armand d'utiliser la propriété "AUTOINCREMENT" dans la base de données pour faciliter le remplissage des clés primaires des différentes tables. Serge a notamment normalisé la base de données.

Lucas s'est concentré sur la base de données ainsi que le Wbs et la conception de la maquette.

2. Validation du document de conception

Le document de conception est au point, Serge remarque qu'on aurait dû le faire plus tôt ceci dit ce n'est pas trop grave : ça nous aurait juste fait perdre du temps. Nous vérifions surtout que tout le monde est d'accord avec la conception du site et la partie CSS avant de pouvoir l'envoyer à M.Festor pour la faire valider, Patrick nous a d'ores et déjà donné son approbation. **On attend la validation du document de conception pour donner quelque chose à faire à Patrick.**

TO-DO LIST

Armand Londeix	- Rédiger le compte rendu
Patrick O'Brien	- Aucune tâche
Lucas Rioux	- Mettre à jour le Gantt
Serge Téhé	- Expliquer comment implémenter le jeu sur le site sur le document de conception

Prochaine réunion : Mercredi 20 Avril à 14h00.

6.2.7 Compte-rendu du 20 Avril 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	14h30 - 14h50
Maître de séance	Lucas Rioux
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Parler du document de conception et de ce qui a été fait

1. Parler du document de conception et de ce qui a été fait

- Tout le monde est d'accord avec ce qu'on a mis dans le document de conception et maintenant on attend la réponse de M. Festor ;
- Lucas a réalisé le Gantt ;
- Lucas pense qu'il faut faire une réunion tous les deux jours. Armand est d'accord mais préférerait des réunions assez courtes avec une réunion d'avancement longue moins fréquente. Tout le monde décide finalement de faire des réunions courtes (stand-up meetings) tous les deux jours de 10-15 minutes ;
- Tout le monde est d'accord que c'est trop tard d'ajouter des nouvelles fonctionnalités ;
- Lucas a commencé à apprendre le JavaScript. Armand pense que JavaScript est moins documenté en ligne du coup il faut minimiser le nombre de fichiers JavaScript. Il demande à Serge, qui a le plus d'expérience en JavaScript, s'il sait faire certains trucs en JavaScript ;
- Lucas demande comment récupérer le nombre de tentatives, Serge dit qu'il a une idée qu'il peut essayer.

TO-DO LIST

Armand Londeix	- Implémenter l'historique des parties et des statistiques
Patrick O'Brien	- Créer les achievements Rédiger les CRs en \LaTeX
Lucas Rioux	- Commencer le CSS
Serge Téhé	- Remplir le fichier .sql (pour créer la BD) Modèle entité-association

Prochaine réunion : Vendredi 22 Avril à 11h00.

6.2.8 Compte-rendu du 22 Avril 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	11h00 - 11h30
Maître de séance	Lucas Rioux
Secrétaire	Patrick O'Brien
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Parler de ce qui a été fait

1. Parler de ce qui a été fait

- Lucas a bien développé le CSS du site ;
- Armand a travaillé sur l'historique et les statistiques. Il avait toujours une petite erreur avec la fonction 'fetchone()' mais on pense qu'il va la résoudre bientôt ;
- Patrick a fait une liste des achievements possibles et il a rédigé le premier compte-rendu en \LaTeX . Il a fait aussi une fonction démo pour calculer les achievements en utilisant les données dans la table 'Partie' mais on a décidé de mettre les achievements dans une autre table 'AchievementDesc' ;
- Serge a mis à jour le fichier database.sql et a réalisé le modèle entité-association.

TO-DO LIST

Armand Londeix	- Implémenter l'historique des parties et des statistiques
Patrick O'Brien	- Implémenter les succès - Mettre leur description dans la BD
Lucas Rioux	- Rédiger les CRs en \LaTeX - Finir le système d'inscription / de connexion
Serge Téhé	- Paramétrage de la grille selon la longueur du mot et le nombre d'essais maximum

Prochaine réunion : Dimanche 24 Avril à 14h00.

6.2.9 Compte-rendu du 24 Avril 2022

Type de réunion	Réunion d'avancement
Lieu	Telecom Nancy
Horaires	12h30 - 13h20
Maître de séance	Serge Téhé
Secrétaire	Serge Téhé
Membres présents	Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Armand Londeix

Ordre du jour

1. Présentation des tâches accomplies
2. Discussion à propos de la page "index.html"

1. Présentation des tâches accomplies

Au début de la réunion, Patrick pose des questions d'éclaircissement sur les tâches qu'il devrait accomplir qui concernent la fonction Achievement et l'insertion des descriptions dans la table. Lucas prend la parole et lui répond en lui faisant la remarque que l'idéesucces en particulier et tous les identifiants devraient être supérieurs à 1. Patrick n'a donc pas terminé ses tâches, il devrait les finir pour la prochaine réunion.

Ensuite Serge prend la parole et présente la tâche qu'il devrait effectuer : le paramétrage de la grille Wordle ; Il explique qu'il a réussi sa tâche et a aussi réglé les bugs au niveau du jeu car Lucas après avoir revérifié le jeu, s'est aperçu d'un bug. En outre, il affirme avoir entamé une autre tâche qui est la liaison des informations des joueurs et la base de donnée qu'il a aussi réussi et qu'il mettra sur le dépôt git après la réunion.

Enfin Lucas explique qu'il a aussi accompli toutes ses tâches à savoir la rédaction de certains comptes rendus en latex, le règlement des bugs au niveau de la page d'inscription.

2. Discussion à propos de la page "index.html"

Serge affirme que le jeu ne fonctionne pas sur la route générant le fichier "index.html" à cause des formulaires qui ont été faits par Armand.

Lucas répond qu'il s'en occupera et va réaménager cette route de sorte à ce que le jeu fonctionne.

Serge propose aussi de permettre qu'un utilisateur non connecté puisse pouvoir paramétrer la grille ; Lucas répond : oui pourquoi pas ? Serge réaffirme : oui ou non ? Lucas se décide finalement et acquiesce cette idée.

TO-DO LIST

Armand Londeix	- Continuer la page Profile
Patrick O'Brien	- Continuer les succès et la fonction Achievement
Lucas Rioux	- permettre à l'utilisateur non connecté de paramétrer la grille - Intégrer le jeu sur la page "index.html"
Serge Téhé	- Prendre en compte les points d'expérience dans la base de donnée - Liaison de la base de donnée avec les informations des joueurs

Prochaine réunion : Mardi 26 Avril à 19h00.

6.2.10 Compte-rendu du 26 Avril 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	19h00 - 19h30
Maître de séance	Lucas Rioux
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Lucas Rioux Serge Téhé Patrick O'Brien
Membres absents	Aucun

Ordre du jour

1. Ce qui a été fait
2. Validation du document de conception

1. Ce qui a été fait

Armand a du mal en JavaScript et n'a pas réussi à implémenter les modes de jeu, cependant il a fait des grades ainsi qu'une barre de progression sur la page profile.html, après discussion on va les remplacer par des niveaux. Enfin l'historique est implémenté mais il reste du travail à faire, ce sera fini dans la soirée.

Patrick a fait la fonction achievements qui est bientôt finie.

Lucas a fait le CSS et a mis en L^AT_EX les comptes-rendus.

Serge a travaillé sur la base de données et leurs liens avec le site sur JavaScript.

TO-DO LIST

Armand Londeix	- Faire les succès et éventuellement trouver des images
Patrick O'Brien	- Finir la fonction achievements
Lucas Rioux	- Continuer le CSS - Finir l'historique
Serge Téhé	- Faire les modes de jeu

Prochaine réunion : Jeudi 28 Avril à 12h00.

6.2.11 Compte-rendu du 28 Avril 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	14h10 - 14h30
Maître de séance	Lucas Rioux
Secrétaire	Patrick O'Brien
Membres présents	Armand Londeix Lucas Rioux Serge Téhé Patrick O'Brien
Membres absents	Aucun

Ordre du jour

1. Avancement depuis la dernière réunion
2. Ce qu'il reste à faire

1. Avancement depuis la dernière réunion

- Serge a fait un mode de jeu "Sans fin" où un nouveau mot est proposé au joueur tant qu'il gagne : la partie s'arrête au premier mot non trouvé ;

- Armand a résolu quelques bugs et a trouvé quelques images pour les achievements ;
- Patrick a complété une fonction qui met à jour les achievements mais il y avait quelques bugs, Lucas a créé une autre fonction qui marche beaucoup plus mieux ;
- Lucas a aussi mis les CR en \LaTeX et fait un peu de CSS pour la page de profil.

2. Ce qu'il reste à faire

Tout le monde est d'accord qu'il ne nous reste pas beaucoup de tâches. Serge pense que c'est important de jouer le jeu plusieurs fois afin de trouver des bugs. Il faut choisir un logo, établir les niveaux d'expérience, tirer les mots les plus fréquents, tester la fonction d'évaluation et rendre le code plus propre.

TO-DO LIST

Armand Londeix	- Compléter la matrice RACI - Tirer le mot à deviner parmi les mots les plus fréquents du dictionnaire (cf. dico science_étonnante)
Patrick O'Brien	- Compléter la matrice RACI - Implémenter le système d'expérience
Lucas Rioux	- Compléter la matrice RACI - Trouver un logo et une icône
Serge Téhé	- Compléter la matrice RACI - Tests en JavaScript de la fonction d'évaluation de <i>dash.js</i>

Oral le Samedi 30 Avril à 10h00.

6.2.12 Compte-rendu du 8 mai 2022

Type de réunion	Réunion d'avancement
Lieu	Telecom Nancy
Horaires	14h00 - 15h00
Maître de séance	Lucas Rioux
Secrétaire	Lucas Rioux
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Mise au point : comment fonctionne le solveur en théorie

1. Mise au point : comment fonctionne le solveur en théorie

Le groupe se met d'accord sur comment fonctionne le solveur en théorie et règle les soucis de compréhension après avoir visionné la vidéo de 3Blue1Brown <https://www.youtube.com/watch?v=v68zYyaEmEA>.

Le joueur devra obligatoirement jouer le mot proposé par le solveur sans quoi il risque d'échouer (conformément au sujet), ce que Armand confirme.

La question de la structure de données utilisée pour stocker le dictionnaire se pose : pour l'instant, seule une liste (chaînée ou contiguë) est envisagée, étant donnée que le groupe n'arrive pas à comprendre comment utiliser un arbre pour résoudre ce problème bien que cette option semble efficace du point de vue de la complexité.

Patrick demande ce qui se passe si l'utilisateur rentre une mauvaise combinaison de couleurs : dans ce cas, le solveur échouera. De toute manière, le solveur ne connaissant pas le mot à trouver, il suffira simplement de le relancer.

Patrick demande aussi s'il sera possible de jouer (sur l'interface Web donc) un autre mot que celui proposé par le solveur : pour l'instant, cette option n'est pas envisagée.

TO-DO LIST

Armand Londeix	- Coder en C pour le solveur - Rapport : rédiger l'historique et les statistiques
Patrick O'Brien	- Rapport : rédiger l'introduction, l'expérience et les achievements
Lucas Rioux	- Mettre les CR dans le rapport - Rapport : agencement du site
Serge Téhé	- Complexité : fonction d'évaluation de index.js - Rapport : rédiger la partie sur la base de données

Prochaine réunion : Vendredi 13 mai à 13h00.

6.2.13 Compte-rendu du 13 mai 2022

Type de réunion	Réunion d'avancement
Lieu	Discord
Horaires	13h-13h30
Maître de séance	Lucas Rioux
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Lucas Rioux Serge Téhé Patrick O'Brien
Membres absents	Aucun

Ordre du jour

1. Ce qui a été fait
2. Todo list

1. Ce qui a été fait

Armand a du mal en c à cause des listes et des structures pour stocker du texte mais est à l'aise avec l'idée du solveur, le rapport n'est pas encore fini. Quelques fonctions ont été créées et sont sur Gitlab cependant certaines sont inachevées.

Patrick a fait une partie du rapport mais avait des doutes sur le solveur, il a profité de la réunion pour poser des questions.

Lucas a avancé sur le solveur en python pour comprendre le principe du solveur en général.

Serge a travaillé sur les structures pour stocker les mots.

TO-DO LIST

Pour la prochaine fois nous allons faire ce qui n'a pas été fait pour la réunion et tous ensemble réfléchir sur le type de structure à implémenter pour les données avant de s'attaquer au solveur en tant que tel.

Aussi nous allons par message nous tenir informé et essayer de déterminer avant ce week end la structure des données afin de commencer le solveur avant la prochaine réunion.

Prochaine réunion : Mercredi 18 mai à 13h.

6.2.14 Compte-rendu du 18 mai 2022

Type de réunion	Réunion d'avancement
Lieu	Discord
Horaires	13h00 - 13h40
Maître de séance	Lucas Rioux
Secrétaire	Patrick O'Brien
Membres présents	Armand Londeix Lucas Rioux Serge Téhé Patrick O'Brien
Membres absents	Aucun

Ordre du jour

1. Ce qui a été fait

1. Ce qui a été fait

- Lucas et Patrick ont bossé sur la rapport ;
- Armand a bossé sur des listes chaînées, il pensait que des listes chaînées ne peuvent pas avoir des mots, Serge n'est pas d'accord, il sait le faire. Lucas dit que on peut stocker les mots dans un arbre binaire mais c'est compliqué ;
- Serge pensait qu'une table des hash codes des mots peut être utile. On met les mots avec le même hashcode dans la même liste. Les autres du groupe avaient un peu du mal à comprendre exactement ce qu'il voulait faire avec cette table mais on est d'accord que c'est la meilleure façon de stocker les mots. ;
- Serge a réussi à récupérer le dictionnaire en faisant ça. Il a fait aussi une fonction (en python) pour calculer l'entropie de chaque mot ;
- Tout le monde a décidé de se renseigner sur le TP de hachage.

TO-DO LIST

Armand Londeix	- Se renseigner sur les tables de hachage, rapport, fichier wself.txt (lire longueur des mots)
Patrick O'Brien	- Se renseigner sur les tables de hachage, rapport
Lucas Rioux	- Se renseigner sur les tables de hachage, rapport
Serge Téhé	- Se renseigner sur les tables de hachage

Prochaine réunion : Samedi 21 Mai à 17h00.

6.2.15 Compte-rendu du 21 mai 2022

Type de réunion	Réunion d'avancement
Lieu	Discord
Horaires	17h00 - 18h00
Maître de séance	Serge Téhé
Secrétaire	Serge Téhé
Membres présents	Patrick O'Brien Lucas Rioux Serge Téhé Armand Londeix

Ordre du jour

1. Présentation des tâches accomplies et Discussion à propos du dictionnaire à choisir
2. Réglages bugs Armand

1. Présentation des tâches accomplies

Au début de la réunion, chaque membre a présenté les tâches qu'il a accomplit.

Lucas a avancé sur le rapport, en particulier sur la partie présentation de l'interface web qu'il devrait finaliser.

Ensuite Patrick prend la parole pour poser une question relative au fonctionnement du solveur, il demande si on utilise le dictionnaire des mots les plus fréquents ou le dictionnaire des mots autorisés. Tous les membres répondent qu'on travaille sur le dictionnaire des mots fréquents. En outre, Patrick propose une stratégie de jeu qui consiste à choisir le mot le plus fréquent parmi les possibilités lorsque le solveur ne trouve pas le mot au bout d'un grand nombre d'essais.

Finalement Serge prend la parole et présente l'implémentation de la structure de données qu'on utilisera dans notre solveur.

2. Réglages bugs Armand

Pendant la réunion, Armand a présenté un bug qu'il rencontrait au niveau de la table de hachage et Serge l'a aidé à trouvé le bug.

TO-DO LIST

Armand Londeix	- Calcul de probabilités
Patrick O'Brien	- Expliquer les règles du jeu dans le rapport
Lucas Rioux	- Fonction d'écriture et de lecture fichier wself.txt
Serge Téhé	- Effectuer le calcul des entropies

Prochaine réunion : Samedi 28 Mai à 14h00.

6.2.16 Compte-rendu du 28 mai 2022

Type de réunion	Stand-up meeting
Lieu	Discord
Horaires	14h00 - 15h00
Maître de séance	Lucas Rioux
Secrétaire	Lucas Rioux
Membres présents	Armand Londeix Lucas Rioux Serge Téhé
Membres absents	Patrick O'Brien

Ordre du jour

1. Avancement depuis la dernière réunion
2. Ce qu'il reste à faire

1. Avancement depuis la dernière réunion

- Armand a aidé Patrick pour le solveur, des erreurs l'empêchaient d'avancer. Il a aussi rédigé la partie **Solveur** du rapport, que Serge a relu.
- Serge a fait tourner son algorithme de recherche de meilleur mot d'ouverture pour 650 mots : il trouve le meilleur mot en 493 secondes, soit un peu plus de 8 minutes. Il a aussi terminé la fonction calculant l'entropie d'un mot.
- Lucas a terminé la fonction lisant dans le fichier texte *wsolf.txt* la longueur du mot.

2. Ce qu'il reste à faire

Le plus important, à l'heure actuelle, est de trouver le mot d'ouverture pour chacune des longueurs du jeu. Il faudra aussi terminer le rapport et tester le code.

TO-DO LIST

Armand Londeix	- Optimiser le solveur : calculer le meilleur mot d'ouverture plus vite - Document de conception : partie Solveur
Patrick O'Brien	- Choix du solveur
Lucas Rioux	- Mettre à jour le schéma BD - Assembler les différentes fonctions écrites pour créer le <i>main()</i>
Serge Téhé	- Optimiser le solveur : calculer le meilleur mot d'ouverture plus vite - Compléter le rapport : partie Solveur

Prochaine réunion : Dimanche 5 Juin à 14h00.

6.2.17 Compte-rendu du 5 juin 2022

Type de réunion	Réunion d'avancement
Lieu	Discord
Horaires	14h-15h
Maître de séance	Armand Londeix
Secrétaire	Armand Londeix
Membres présents	Armand Londeix Patrick O'Brien Lucas Rioux Serge Téhé
Membres absents	Aucun

Ordre du jour

1. Ce qui a été fait depuis la dernière fois
2. Ce qui reste à faire

1. Ce qui a été fait

- Lucas a augmenté le nombre de listes par table afin d'augmenter la rapidité des algorithmes, il a d'autre part travaillé sur le solveur qui est maintenant fonctionnel.
- Serge a travaillé sur le rapport et notamment sur la partie solveur qui est aboutie. Il souhaitait travailler sur l'optimisation des algorithmes mais cette partie a déjà été faite par Lucas.
- Patrick n'a pas encore commencé la partie personnelle mais a commencé à y réfléchir, il souhaitait utiliser le solveur mais n'a pas réussi à l'implémenter.
- Enfin, Armand a travaillé essentiellement sur le rapport, sur la partie solveur ainsi que la partie personnelle et la partie sur le travail réalisé.

2. Ce qui reste à faire

Il faut finaliser le rapport, faire les bilans personnels ainsi que donner le nombre d'heures de travail et enfin faire le bilan du groupe. Aussi, la partie sur les complexités est à finir et des tests doivent être faits sur les algorithmes. Enfin, les meilleurs mots par longueur de mots doivent être trouvés.

TO-DO LIST

À part Lucas et Serge qui peuvent finaliser le solveur, tout le monde doit faire le rapport et tester les algorithmes ainsi que trouver leurs complexités.

Prochaine réunion : Mardi 7 juin à 15h00.

7 Conception de WORDLE

7.1 Présentation de WORDLE

Notre jeu est basé largement sur le jeu Wordle du New York Times. Le principe est d'essayer de deviner un mot secret en le moins de coups possible. Dans le jeu original le mot secret a cinq lettres, pour chaque tentative l'utilisateur doit essayer un mot qui existe dans le dictionnaire du jeu. En utilisant des couleurs différentes, le jeu donne des indices : si une des lettres du mot essayé est à la bonne place dans le mot secret, la lettre devient verte ; si elle est dans le mot mais mal placée elle devient jaune, sinon la lettre n'est pas dans le mot et reste grise. Le joueur a un maximum de 6 essais pour trouver le mot, et le jeu s'arrête lorsque le mot secret est trouvé ou bien si le joueur n'a plus de tentatives. Nous avons préservé les principes fondamentaux du jeu, mais nous avons donné au joueur la liberté de choisir la longueur du mot secret ainsi que le nombre de tentatives maximum.

7.2 Présentation de l'application Web

7.3 Schéma de la base de données et modèle entité-association

Notre base de données est constitué de 4 tables :

- Une table User dont les colonnes sont : idUser qui est l'identifiant du joueur de type entier, un pseudo qui représente le nom du joueur de type str, un mot de passe de type str et une chaîne de caractère salt pour la sécurisation du compte utilisateur. La chaîne de caractères stockée dans la colonne mot_de_passe sera le haché de la concaténation du mot de passe et du salt.
- Une table Partie dont les colonnes sont : idPartie qui représente l'identifiant d'une partie de type int, idUser qui est une clé étrangère, les points d'expérience de type int qui représente les points gagnés lors d'une partie, le mot secret associé à la partie de type str, le nombre de coups mis par le joueur pour trouver le mot secret de type int, le nombre d'essais maximum de la partie de type int et la longueur des mots associé à la partie de type int.
- Une table AchievementDesc dont les colonnes sont : idSucces qui est l'identifiant de l'achievement et description la description associée ("gagner une partie en 2 coups", "gagner 10 parties" etc...).
- Une table Achievements qui relie les tables AchievementDesc et partie dont les colonnes sont : idPartie clé étrangère et idSucces clé étrangère. Ces 2 clés forment la clé primaire de la table Achievements.

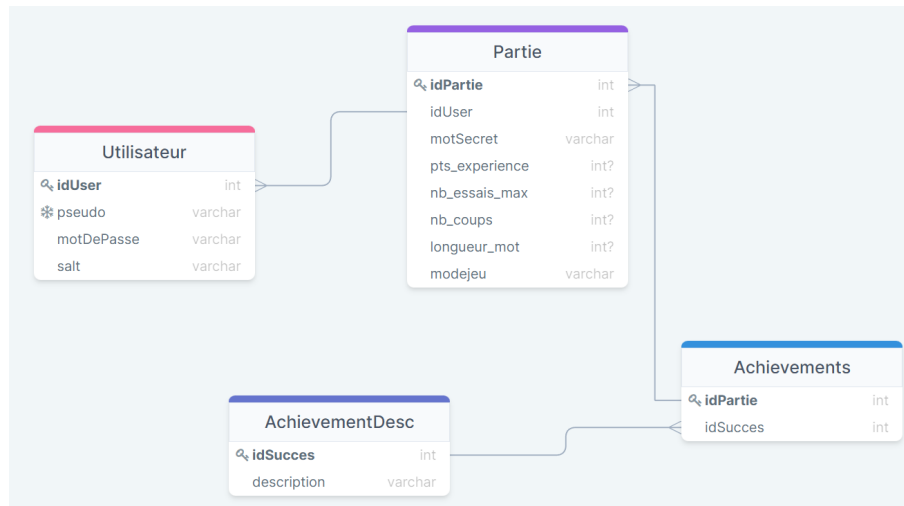


FIGURE 12 – Schéma de la base de données de WORDLE

7.3.1 Agencement des pages HTML et des routes

Le site sera composé de cinq pages HTML :

- index.html, la page d'accueil et la page de jeu. Elle est la page centrale du site : depuis index.html, on peut se créer un compte, se connecter et consulter son profil une fois connecté. À noter que créer un compte n'est pas obligatoire pour jouer mais, dans ce cas, aucune partie ne sera enregistrée et aucun achievement ne sera délivré.
- register.html, où un joueur peut se créer un compte. Après avoir vérifié que le pseudo renseigné est disponible, les données sont enregistrées dans la table User.
- login.html, où le joueur peut se connecter. La connexion réussit si les informations d'authentification (pseudo et mot de passe) renseignées sont valides et échoue sinon.
- profile.html, où le joueur peut consulter les informations relatives à son profil. Ces informations sont l'historique des parties effectuées, les statistiques relatives à ces parties et les succès débloqués par le joueur.
- settings.html, où le joueur peut choisir le mode de jeu ainsi que la longueur du mot à deviner et le nombre maximum de tentatives autorisées.

Quant aux routes, elles seront organisées de la manière suivante :

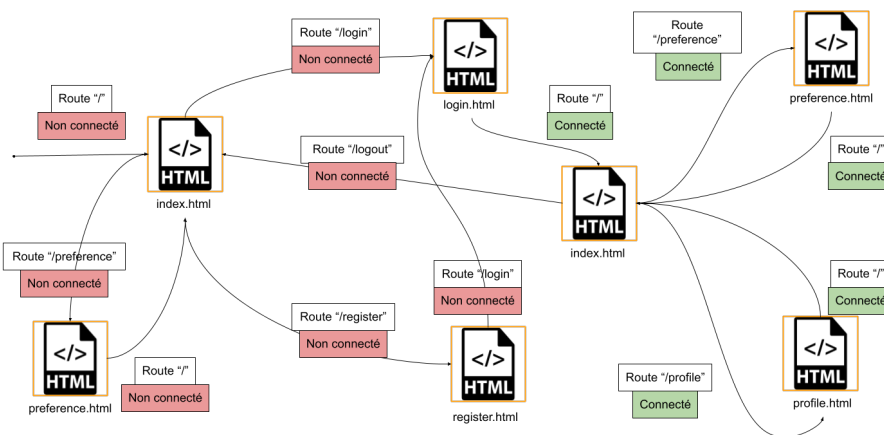


FIGURE 13 – Organisation des routes de WORDLE

7.3.2 Implémentation du jeu

Fonctionnement global

Notre jeu est constitué d'une grille dont les dimensions sont paramétrables par le joueur. Étant donné les paramètres, on crée en JavaScript une grille dont chaque case a un identifiant unique. À l'aide d'un gestionnaire d'événements de clavier, on récupère chaque lettre majuscule tapée par l'utilisateur et on l'insère dans chaque case de la gauche vers la droite jusqu'à ce qu'on atteigne la longueur du mot définie par l'utilisateur. L'utilisateur a également la possibilité d'effacer une lettre rentrée. Aussi, lorsqu'il tape "entrer" et que la longueur du mot entré est égale à celle choisie par l'utilisateur, il verra chaque case contenant chaque lettre se colorer selon la position de la lettre dans le mot secret. Il faut noter que le mot secret est choisi arbitrairement dans le dictionnaire et est caché dans une balise html.

Fin du jeu

Dans une liste, nous récupérerons chaque lettre jusqu'à ce qu'on atteigne la longueur du mot secret. Cette liste est ajoutée dans une autre liste vide définie préalablement et à chaque fois que le mot ne correspond pas au mot secret, on réitère le processus. Lorsqu'on atteint le nombre d'essais maximum ou lorsque le mot est trouvé, la partie se termine et un bouton "rejouer" s'affiche dans le cas où un utilisateur voudrait rejouer le jeu.

7.3.3 Maquette du site

Voici le premier design du site :

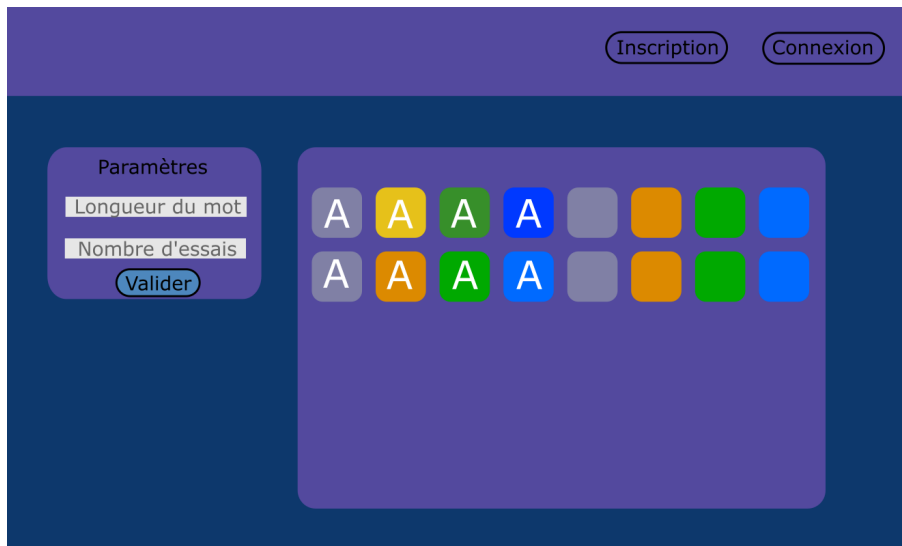


FIGURE 14 – Première maquette de WORDLE

Ce design représente la page `index.html` lorsque le joueur est déconnecté. Le logo se situerait en haut à gauche. Les couleurs sont potentiellement sujettes à modification.

7.3.4 Paramétrage du jeu et profil utilisateur

Le paramétrage du jeu, c'est-à-dire le choix du mode de jeu, de la longueur du mot à deviner ainsi que du nombre de tentatives autorisées est entièrement paramétrable par le joueur sur la page de jeu.

Sur la page `profile.html`, chaque joueur aura accès à l'historique de ses parties (pour chaque partie, voir quel était le mot secret et si la partie a été gagnée ou non), ses statistiques de jeu (le nombre moyen de coups par partie, le pourcentage de victoire et son nombre de parties jouées) et aux achievements (succès) qu'il aura débloquent en jouant. Pour ce qui est des statistiques, elles seront déclinées en fonction de la longueur du mot : pour chaque longueur de mot différente, on aura un volet "nombre moyen de coups par partie" et un volet "pourcentage de victoire" en plus des statistiques globales.

7.3.5 Solveur

Le solveur permet au joueur de penser à un mot que le solveur doit deviner en suivant les règles du jeu. Le solveur est codé exclusivement en langage C et utilise des structures de données les plus appropriées, à savoir une table de hachage permettant au solveur de trouver les mots rapidement.

Au lancement du programme, le solveur explique au joueur comment jouer et le joueur répond dans le terminal. Le joueur donne des indices au solveur de la même façon que le jeu WORDLE donne des indices au joueur. Si le joueur veut arrêter le jeu, il met -1 dans le terminal, sinon il donne une chaîne de la même longueur que le mot avec des chiffres 0, 1 et 2 : 2 représente une lettre bien placée, 1 représente une lettre correcte mal placée et 0 représente une lettre qui ne se trouve pas dans le mot. En fonction des réponses du joueur, le solveur retourne au joueur le mot le plus probable. Par exemple, si le mot recherché est ARBRE et que le solveur propose NAGER, le joueur est obligé de répondre avec '01011'.

```

--++-- Solveur WORDLE --++--

Commandes :

Couleurs :
0 : la lettre n'est pas dans le mot à deviner (gris)
1 : la lettre est dans le mot à deviner, mais elle est MAL placée (orange)
2 : la lettre est dans le mot à deviner et elle est BIEN placée (vert)

Rentrez le mot que vous propose le solveur dans votre partie de WORDLE.
Ensuite, rentrez dans le solveur la combinaison de couleurs que vous avez obtenue.

-1 : arrêt du solveur

Pour cette partie, on joue avec des mots de 5 lettres !

Chargement du dictionnaire...
Dictionnaire chargé !
Nombre d'alvéoles de la table : 300
Taille du dictionnaire : 5070

Meilleur mot d'ouverture : tarie
Combinaison de couleurs obtenue (-1 pour arrêter) :

```

FIGURE 15 – Avant de jouer

```

Meilleur mot d'ouverture : tarie
Combinaison de couleurs obtenue (-1 pour arrêter) :
01220

Temps écoulé : 0 secondes
Meilleur mot : moria avec 1.921928 bits d'information
Réponses restantes possibles : 5
Combinaison de couleurs obtenue (-1 pour arrêter) :
00221

Temps écoulé : 0 secondes
Meilleur mot : abris avec 1.000000 bits d'information
Réponses restantes possibles : 2
Combinaison de couleurs obtenue (-1 pour arrêter) :
20220

Temps écoulé : 0 secondes
Meilleur mot : avril avec 0.000000 bits d'information
Réponses restantes possibles : 1
Combinaison de couleurs obtenue (-1 pour arrêter) :
22222

```

FIGURE 16 – En jouant

8 Bibliographie

- Tutoriel pour jouer à WORDLE (Sean Plays) : <https://www.youtube.com/watch?v=PXUqGT5ySsc>
- Tutoriel sur l'utilisation de la théorie de l'information pour optimiser la résolution d'une session WORDLE (3Blue1Brown) : <https://www.youtube.com/watch?v=v68zYyaEmEA>
- Correction d'un bug dans la vidéo précédente (3Blue1Brown) : <https://www.youtube.com/watch?v=fRed0Xmc2Wg>
- Une autre vidéo sur le même sujet (ScienceEtonnante) : https://www.youtube.com/watch?v=iw4_7ioHWF4