



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

---

# Application web Wordle

Document de conception

---

## Table des matières

<b>1</b>	<b>Présentation de Wordle</b>	<b>3</b>
1.1	Le jeu Wordle . . . . .	3
1.2	Les dérivés du Wordle . . . . .	3
<b>2</b>	<b>Présentation de l'application web</b>	<b>5</b>
2.1	Rappel des consignes . . . . .	5
2.2	Les langages utilisés . . . . .	5
2.3	Les pages web . . . . .	5
2.4	La base de données . . . . .	7
2.5	Les algorithmes Python et JavaScript . . . . .	8

# 1 Présentation de Wordle

## 1.1 Le jeu Wordle

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

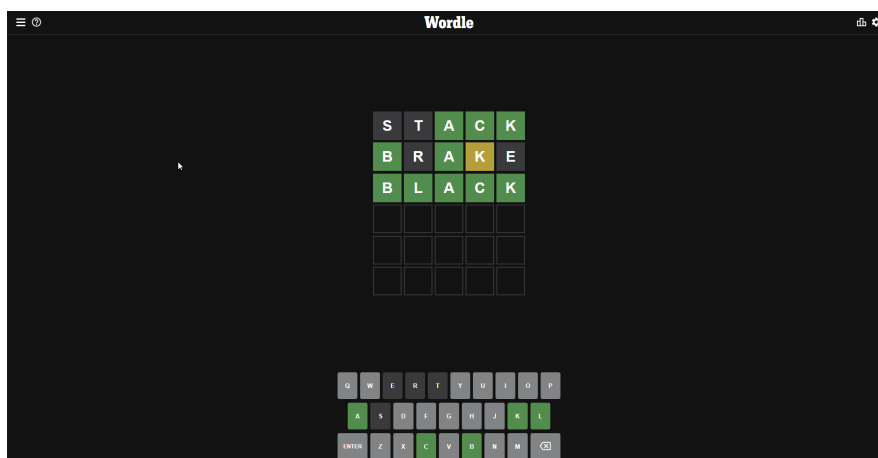


FIGURE 1 – Un exemple de partie sur le site du Wordle. En vert les lettres bien positionnées, en jaune, celles qui ne le sont pas, et sans couleur celles qui ne sont pas dans le mot.

## 1.2 Les dérivés du Wordle

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

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

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

## 2 Présentation de l'application web

### 2.1 Rappel des consignes

Dans le cadre de ce projet, nous devons réaliser une application web du jeu Wordle sur une architecture Python/Web/Base de données. Les fonctionnalités obligatoires en plus du jeu en lui-même sont le paramétrage des parties (longueur du mot à trouver et nombre maximal d'essais possibles) ainsi qu'un système d'enregistrement des parties jouées. D'autres fonctionnalités peuvent également être implémentées.

### 2.2 Les langages utilisés

Nous utiliserons les langages HTML, CSS, JavaScript, Python et SQL pour réaliser notre application.

### 2.3 Les pages web

Les pages de notre application web seront construites selon le template présentée en figure 2.

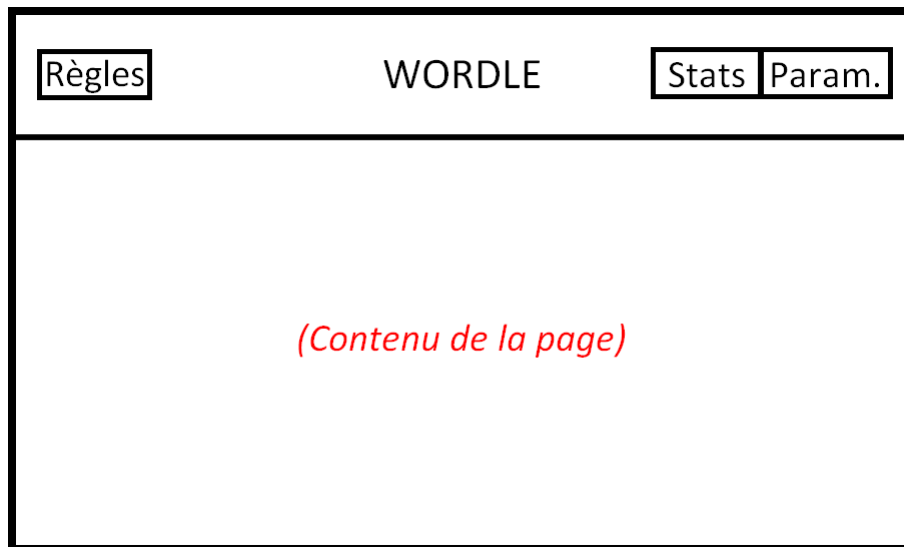
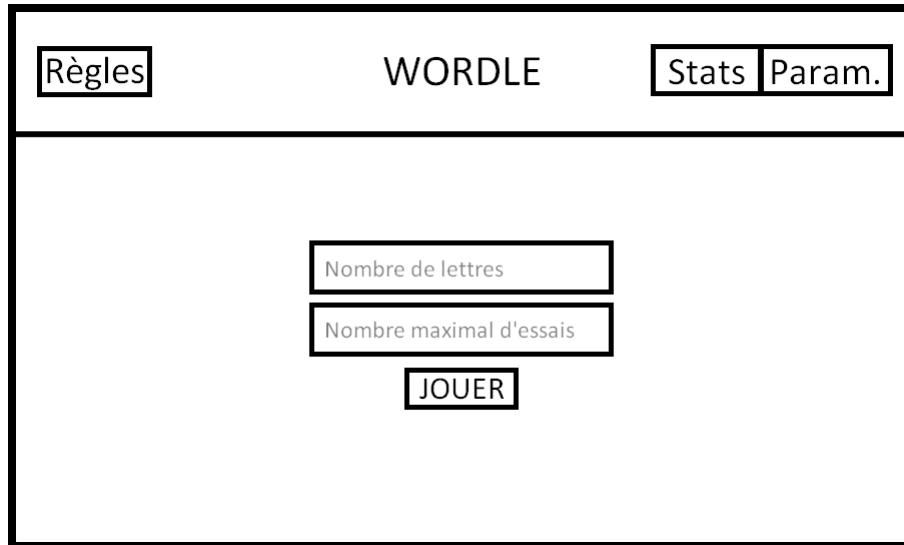


FIGURE 2 – Template général de la page Web

Les boutons "Règles", "Stats" et "Param." seront des fenêtres pop up qui apparaîtront sur la page lorsque l'utilisateur cliquera dessus. Le bouton "Règles" affichera les règles du jeu, le bouton "Stats" les statistiques de l'utilisateur et le bouton "Param." les paramètres de la page dont une possibilité de changer les couleurs. Ce dernier point permet une plus grande accessibilité du jeu, notamment pour les datltoniens, mais ne sera implémenté que si cela est possible au vu des contraintes imposées par ce projet.

La partie "Contenu de la page" contiendra toutes les fonctionnalités nécessaires au jeu Wordle. Tout d'abord, il y aura la page de paramétrage de la partie, présentée en figure 3. Elle sera constituée d'un formulaire dont les valeurs rentrées seront limitées pour éviter des problèmes avec la génération de la grille (mot qui n'existe pas pour la longueur donnée, nombre d'essais trop grand, ...). Cette page sera chargée si l'utilisateur n'a pas de partie en cours. Dans le cas contraire, la page suivante sera directement chargée.



The image shows a web interface for the Wordle game. At the top, there is a header bar with three buttons: "Règles", "WORDLE", and "Stats Param.". Below the header, the main content area is a large white rectangle. Inside this rectangle, there is a form with two input fields: "Nombre de lettres" and "Nombre maximal d'essais". Below these fields is a button labeled "JOUER".

FIGURE 3 – Page de paramétrage de la partie

La grille sera donc générée une nouvelle page. La structure de celle-ci est indiquée en figure 4. Si aucune partie n'était en cours, les dimensions de la grille seront celles indiquées dans le formulaire de la page précédente. Dans le cas contraire, la grille sera créée selon les paramètres de la partie en cours et sera remplie avec les mots déjà essayés par l'utilisateur.

Une fois la partie terminée, une fenêtre pop up apparaîtra sur la page. Elle contiendra les résultats de la partie, les statistiques de l'utilisateur ainsi qu'un bouton "Rejouer".

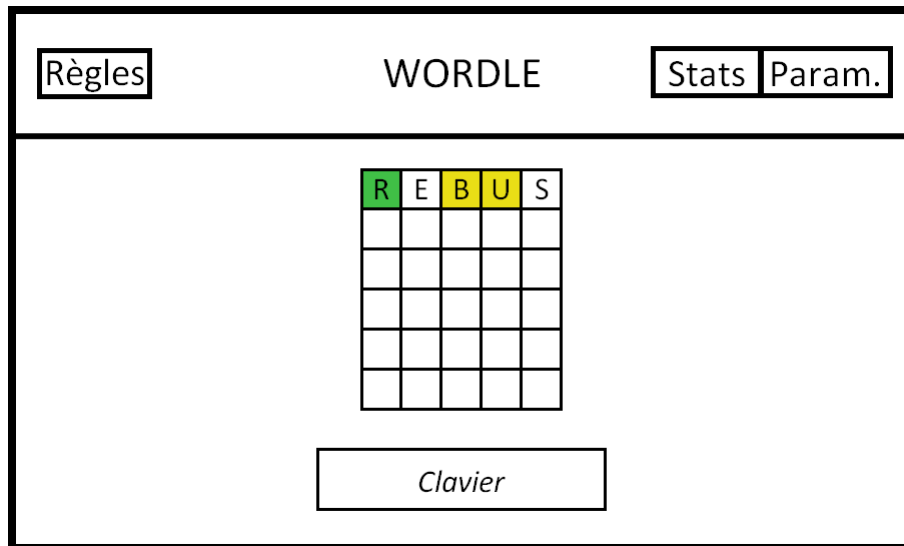


FIGURE 4 – Page contenant la grille. Les couleurs sont ici celles du jeu de base mais ne représentent pas celles qui seront choisies lors de l’implémentation. Elles seront choisies en fonction du web design mis en place.

## 2.4 La base de données

La base de données utilisée dans le cadre de notre projet a été construite sur le logiciel en ligne dbdiagramm.io [9]. Celle-ci a été pensée pour être en 3<sup>ème</sup> forme normale. Elle est décrite dans la figure 5.

Pour implémenter notre base de données, nous utiliserons SQLite3 et la bibliothèque Python du même nom. Elle sera donc constituée de 3 tables :

- Une table PLAYERS dans laquelle seront stockés pour chaque joueur son identifiant et l’identifiant de la dernière partie qu’il a jouée.
- Une table GAMES qui stocke les informations générales sur chaque partie.
- Une table TRIES qui stocke tous les essais de chaque joueur pour chaque partie.

L’attribut PLAYERS.idLastGame permet de vérifier à l’aide de l’attribut GAMES.gameEnded si une partie est en cours pour un joueur donné. Enfin, la longueur du mot à trouver dans une partie n’a pas été ajoutée comme attribut de la table GAMES. Elle peut être aisément récupérée en prenant la longueur de la chaîne de caractère stockée dans l’attribut GAMES.wordToFind.

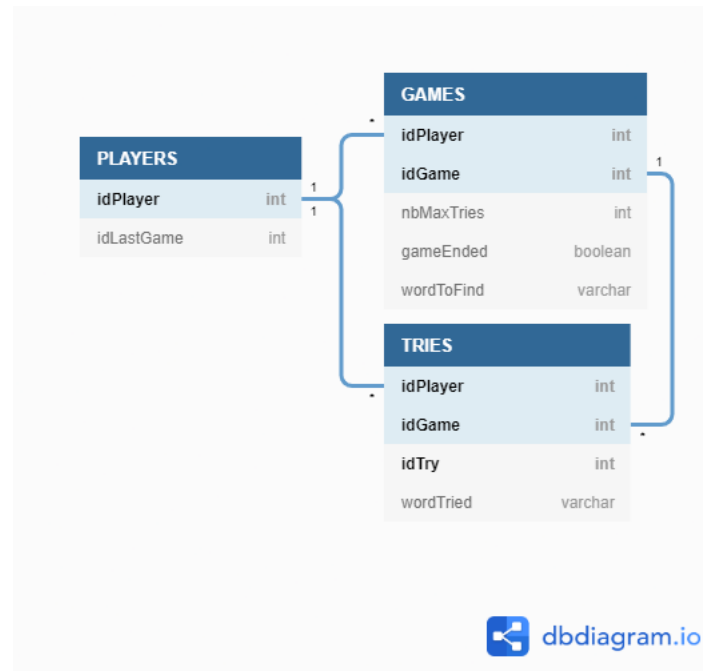


FIGURE 5 – Schéma de la base de données

Pour ce qui est du dictionnaire, les mots seront rangés par ordre alphabétique dans des fichiers JSON. Tous les mots de la même longueur seront rangés dans le même fichier.

## 2.5 Les algorithmes Python et JavaScript

Toutes les fonctions de notre application seront codées en JavaScript et en Python. Python sera utilisé pour l'API Flask du site ainsi que pour toutes les interactions avec la base de données. JavaScript sera quant à lui utilisé pour les parties. Les fonctions qui seront implémentées sont détaillées ci-après :

- **(Python)** `userId()` qui récupère l'identifiant de l'utilisateur stocké en cookie. Si le cookie n'existe pas, elle affecte un identifiant à l'utilisateur.
- **(Python)** `unfinishedGame()` qui vérifie au chargement de la page si l'utilisateur a une partie en cours ou non. Si c'est le cas, elle recharge la partie.
- **(JavaScript)** `generateGrid()` qui générera la grille avec les paramètres rentrés par l'utilisateur (longueur du mot et nombre d'essais).
- **(JavaScript)** `chooseWord()` qui importe le fichier JSON pour la partie correspondant à la longueur du mot à trouver entrée par l'utilisateur et qui choisira ce mot.



- **(JavaScript)** `inputKeyboard()` qui détecte les entrées du clavier de l'utilisateur.
- **(JavaScript)** `testTry()` qui vérifie si le mot proposé a la bonne longueur (en particulier qu'il ne manque pas de lettres) et s'il est dans le dictionnaire. Elle renvoie un booléen.
- **(JavaScript + Python)** `coloration()` qui s'exécute si `testTry()` renvoie Vrai et qui colorie les cases du mot proposé par l'utilisateur. Pour cela, elle retourne une liste de longueur égale à celle du mot où les valeurs sont soit 0 si la lettre correspondante n'est pas dans le mot, soit 1 si elle n'est pas à la bonne place, soit 2 si elle est à la bonne place. Elle enregistre aussi l'essai dans la base de données.
- **(JavaScript)** `testEndGame()` qui met fin à la partie si le nombre d'essais maximal est dépassé ou si le mot est trouvé. Dans le cas contraire, la partie continue sur la ligne suivante de la grille.
- **(JavaScript + Python)** `endGame()` qui sauvegarde la partie dans la base de données et affiche ses résultats.
- **(Python)** `stats()` qui affiche les statistiques de l'utilisateur.

## Références

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