

Projet d'IN104

Imbattable à Wordle !

Contexte :

Apparu en 2021 et rapidement racheté par le New-York Times pour quelques millions de dollars, le jeu wordle est rapidement devenu populaire. Le principe est simple, chaque jour un joueur possède 6 essais pour deviner un mot de 5 lettres. A chaque tentative, le joueur sait si la lettre est bien placée, mal placée mais présente dans le mot ou absente du mot. Le but est bien entendu de deviner le mot en un minimum de tentatives

Le jeu est disponible ici : <https://wordle.louan.me/>

Les règles sont les suivantes :

Comment jouer ?

Ce jeu reprend exactement le même concept que le **Wordle**, mais en français.

Chaque jour, un mot de 5 lettres est choisi aléatoirement. Vous devez le deviner en 6 essais.

À chaque essai, les lettres du mot que vous avez proposé changeront de couleur en fonction de à quel point vous êtes proche de le trouver.

F

R

U

I

T

La lettre **F** est dans le mot, à la bonne place.

P

O

C

H

E

La lettre **C** est dans le mot, mais pas à la bonne place.

S

O

E

U

R

La lettre **R** n'est pas dans le mot.

Projet à réaliser

Le projet se décompose en plusieurs étapes. Les étapes 1 et 2 constituent la base du projet et doivent être réalisées. **Les étapes suivantes sont des bonus**, vous pouvez les implémenter dans l'ordre que vous souhaitez.

1. Développer le jeu en console

Il s'agit de coder le jeu pour qu'un utilisateur puisse y jouer dans la console. Les règles sont les suivantes :

- L'utilisateur est invité à saisir un mot de 5 caractères.
Attention, le mot doit exister. Votre programme doit donc disposer d'un dictionnaire de référence et vous devez vérifier que le mot saisi par le joueur est présent dedans. Sinon un message d'erreur **explicit** doit s'afficher.
De même, si le mot saisi ne fait pas 5 caractères, un message d'erreur doit s'afficher
- Vous devez proposer un code qui permet de savoir pour chaque lettre du mot si elle est présente ou non, et si elle est à la bonne place. Pensez à afficher au début du jeu la légende de votre code.
- La séquence se répète jusqu'à ce que le joueur trouve le bon mot ou arrive au bout de ses 6 essais.
- Si le joueur n'a pas trouvé le mot au bout des 6 essais, celui-ci s'affiche avec un message de fin de partie

2. Programmer un ordinateur pour qu'il joue le meilleur mot possible à chaque étape

L'objectif de cette partie est de faire jouer un ordinateur de la meilleure manière possible.

La théorie de l'information peut par exemple vous aider, mais n'hésitez pas à développer d'autres pistes.

Pensez à détailler votre stratégie pour la restitution finale, et à justifier vos choix.

3. Proposer un affichage plus agréable et proche de celui du jeu original

Vous pouvez utiliser la SDL ou laisser libre cours à votre imagination.

4. Proposer une personnalisation du jeu

Vous pouvez proposer au joueur de choisir : la longueur du mot à trouver, le nombre d'essais, la langue etc.

5. Proposer des statistiques (laisser libre cours à votre imagination)

Vous pouvez par exemple proposer :

- Des statistiques sur les scores et la réussite du joueur
- Regarder quel mot est saisi en premier par les joueurs (dans ce cas, essayez de faire plusieurs jouer un maximum de personnes).
- Regarder les stratégies des différents joueurs (joue-t-il toujours le même mot en premier ? joue-t-il uniquement des mots possibles ou tente-t-il des mots qui comprennent des lettres non disponibles)
- Etudier ce qui vous intrigue (très valorisé pour la restitution).

Critères d'évaluation et consignes

Le rendu du projet se fera par git.

Chaque groupe devra m'envoyer un mail à l'adresse : marine.saintmezard@gmail.com. Ce mail **doit uniquement contenir un lien vers votre répertoire git et la composition de votre groupe**. Je ne tiendrai pas compte des pièces jointes à ce mail.

Dans votre répertoire git vous devez inclure :

- **Un README.txt** : il s'agit d'un fichier txt contenant une brève description du projet, les instructions de compilation, et toute information qui vous semble pertinente.
- **L'ensemble de vos fichiers .c et .h**, le fichier principal s'appellera jeu.c
- Un fichier **pdf qui constituera votre rapport final** d'une longueur de 2/3 pages. L'objectif de ce rapport est de présenter la structure de votre projet. Le plus important est de justifier vos choix d'implémentation. Pourquoi avoir choisi ce type de structure ? Quelle est la logique de l'algorithme ? Vous devrez également commenter les sorties de votre programme. Fait-il ce qui est attendu ? Pourrait-il être amélioré ? Si oui, comment ? Essayez d'avoir d'un esprit critique dans la rédaction de ce rapport.

Le contenu du rapport servira de base pour réaliser vos slides pour la soutenance orale qui aura lieu en fin de projet.

Votre programme doit être autoportant, c'est-à-dire qu'une personne connaissant les règles doit pouvoir jouer sans avoir besoin de lire de la documentation au préalable. Vous devez donc porter un soin particulier aux messages qui s'affichent à l'écran et aux demandes de saisie de la part de l'utilisateur.

La lisibilité et la beauté du code seront prises en compte pour la notation :

- Indentez votre code
- Faites attention à vos noms de variables, ils doivent être explicites
- N'hésitez pas à découper le code en sous fonction voir à faire plusieurs fichiers
- N'hésitez pas à commenter votre code quand cela est nécessaire.

Votre code doit compiler avec les instructions que vous incluez dans le fichier readme. Tout projet qui ne compile pas aura une note en dessous de la moyenne.

Toute prise d'initiative et de créativité sera valorisée ! Amusez-vous !