



Enunciado Proyecto 1

Algoritmos

1 Introducción

En este proyecto deberán desarrollar un jugador para Wordie. La evaluación será competitiva, y estará determinada por la menor cantidad de jugadas necesarias para adivinar las palabras.

2 Detalles

2.1 Juego

Wordie es un juego donde el jugador debe adivinar una palabra oculta, para lo cual entrega una palabra y el juego le responde uno de tres resultados posibles para cada una de las letras: correcta y en posición, correcta en posición equivocada, incorrecta. El jugador tiene varios intentos hasta adivinar la palabra. Tanto la palabra oculta como las palabras que entrega el jugador deben ser válidas dentro del diccionario del idioma que se está jugando.

Este juego se hizo famoso el año pasado por el desarrollo de Josh Wardle que luego fue comprado por el New York Times. Pueden jugar a la versión en inglés en:

<https://www.nytimes.com/games/wordle>

En este caso el juego es siempre de palabras de 5 letras, en inglés y permite un máximo de 6 intentos. Estas son las instrucciones publicadas en la página:





Después de esto han aparecido nuevas versiones del juego. Este es el caso de **Quordle** en el cual el jugador debe adivinar 4 palabras ocultas en paralelo, entregando una palabra y recibiendo retroalimentación para las 4. El jugador tiene 9 intentos para acertar todas las palabras. Pueden jugar en:

<https://www.quordle.com/>

Las instrucciones que publica el juego son las siguientes:

Guess all four Quordle words in 9 tries.

Each guess must be a valid 5 letter word. Hit the enter button to submit. After each guess, the color of the tiles will change to show how close your guess was to the word.

Examples

C R O W N

The letter C is in the word and in the correct spot.

B A D G E

The letter A is in the word but in the wrong spot.

C O M F Y

The letters C, O, M, F, Y are not in the word in any spot. When you type a guess in Quordle, you will guess that word for all four words that you are solving. All four words you are solving will be different.

W O R L D W O R L D

W O R L D W O R L D

For the guess WORLD:

1. The top left word has none of the letters.
2. The top right word has the R in the wrong spot and the D in the correct spot.
3. The bottom left word has the L in the wrong spot.
4. The bottom right word has the O in the right spot and the D in the wrong spot.

You have 9 guesses to get all 4 words correct. Good luck!

2.2 Aplicación

Su aplicación deberá ser un jugador que juegue al juego a través de una API que será provista por el profesor. Así su jugador deberá enviar palabras para un juego, recibiendo el resultado para cada palabra oculta.



Cada juego estará determinado por:

- **Lenguaje** → su juego deberá soportar distintos lenguajes. La API le permitirá acceder al diccionario con todas las palabras válidas para el lenguaje de cada juego. A su vez, cada lenguaje tiene caracteres propios, por ejemplo, en el caso del español están las vocales con tilde.
- **Cantidad de palabras** → cada juego definirá la cantidad de palabras ocultas que deberá adivinar su jugador. Este valor podrá ser entre 1 a 12.
- **Cantidad de letras** → la cantidad de letras también será una variable para cada juego, aunque en ese juego todas las palabras tendrán el mismo largo. Este valor podrá ser entre 4 y 14.

El objetivo del juego es acertar en la menor cantidad de turnos, no se trata de ser el más rápido en tiempo. De todas formas, para las pruebas oficiales (en clase) los juegos tendrán un tiempo máximo de 3 minutos para ser completados.

2.3 API

La URL base de la API es <https://pds-wordie.herokuapp.com>

La API proporciona las siguientes llamadas:

- **GET /api/games/** → permite obtener la lista de juegos activos en formato JSON.
- **POST /api/play/** → permite enviar una jugada, se requiere indicar el id del juego, la key del jugador y la palabra. Retornará error error BAD REQUEST (400) si no se cumple las restricciones. Si se cumplen las restricciones retorna el resultado para cada palabra oculta como un string de 0 (incorrecta), 1 (correcta mal ubicada), 2 (correcta y en posición). Además indica que palabras ya fueron adivinadas y el estado del jugador en el juego.
- **POST /api/reset/** → permite eliminar las jugadas de un jugador permitiendo que vuelva a comenzar el juego.

Cada estudiante recibirá una **key** la cual deberá utilizar para todas sus jugadas. Habrán juegos activos para que puedan ir probando mientras desarrollan su juego.

Pueden revisar ejemplo de uso de la API con python en este colab:

<https://colab.research.google.com/drive/1-lqwh3VwyWIDp-pRi3f7pspZtBRhUP5N>

3 Entregas

3.1 Entrega parcial [1.5 ptos.]

Lunes 8 de agosto en Canvas hasta las 13.00 horas. Durante la clase se realizará un conjunto de juegos entre todos.

Para esta entrega su jugador deberá jugar juegos con una sola palabra. El puntaje se asignará de la siguiente forma:

- **0.5** → entrega por Canvas de un diagrama que explique el algoritmo utilizado pensando en el juego con una sola palabra. No hay un formato, pero debe ser un diagrama que permita a otro desarrollador implementar la misma estrategia (no debe tener código).



- 1.0 \rightarrow habrán 8 juegos con dos diccionarios (inglés y español) y con 4 tamaños de palabra para cada uno: 4, 5, 6, 8. Para obtener todo el puntaje su jugador no debe tomar más de 40 turnos totales entre los 8 juegos. Luego se descuenta una décima por cada turno adicional.
- Bono: 0.3 \rightarrow el o los con menos intentos tendrán esta bonificación.

3.2 Entrega final

Lunes 22 de agosto en Canvas hasta las 13.00 horas. Durante la clase se realizará la competencia donde se ampliará la cantidad de lenguajes, será con múltiples palabras y número de caracteres. La nota será competitiva, obteniendo todo el puntaje (4.5) aquel que utilice menos turnos en todos los juegos. El profesor desarrollará 2 jugadores, uno simple y uno avanzado, los cuáles también competirán. El jugador simple del profesor definirá el corte para obtener 2 puntos en la entrega (los que tengan más intentos tendrán menos que eso).

4 Consideraciones

No hay restricciones de ningún tipo respecto al desarrollo de su solución, pueden usar cualquier lenguaje, aplicación, programa, infraestructura, algoritmo, estrategia, ...

El problema es simple, lo pueden realizar en unas cuantas horas, pero recuerden que la base está en la competitividad por ser el más eficiente (menor cantidad de jugadas).