Curso 2023 – 2024

PARAULOGIC

Objetivo

El alumno debe demostrar haber adquirido los conocimientos de programación desarrollados durante el curso. Desarrollo de algoritmos de búsqueda y recorrido, desarrollo de subprogramas, utilización de arrays, solventar problemas usando la librería STL y tratamiento de ficheros secuenciales.

Enunciado

Implementar un programa que permita jugar al PARAULOGIC, el jugador debe formar una serie de palabras a partir de un conjunto de letras dadas. El conjunto de letras es seleccionado de forma aleatoria a partir de un fichero de texto. El juego esta descrito con mayor detalle posteriormente. Las prácticas se entregan en Aula Digital, la fecha límite de entrega es el 17 de Enero a las 23:55 para la convocatoria ordinaria, y el 04 de Febrero para la evaluación extraordinaria.

Dicha entrega estará conformada por:

1. Proyecto correspondiente junto con el código fuente.
2. Documentación en PDF con los siguientes apartados:

    a) Portada con el título de la práctica, el nombre del autor, nombre de la asignatura.

    b) Introducción que sintetice el enunciado de la práctica.

    c) Diseño. Donde se describa el diseño descendente que ha conducido a la solución propuesta.

    d) Juego de pruebas con la descripción de los ensayos, pruebas y datos utilizados para comprobar el buen funcionamiento del programa.

    e) Conclusiones que sinteticen la experiencia alcanzada, describan los conceptos asimilados y resalten los puntos que han resultado más difíciles de resolver.

Detalles del programa

El programa a desarrollar consiste en implementar el juego PARAULOGIC. El programa cargará un fichero de texto llamado Palabras.txt, o bien lo recibirá como parámetro, en tal caso debe usar el fichero indicado (es decir, ambos casos deben estar implementados). El fichero contendrá palabras en minúsculas, una por línea y sin espacios en blanco. Sólo se juega con las palabras menores o igual a 7 letras, las demás palabras son ignoradas.
Al usuario se le pide que forme una palabra con las letras dadas. Y tendrá tantos intentos como se quiera.

- Si la palabra **no se encuentra** en el diccionario → debe indicarlo al usuario.
- Si la palabra **se encuentra** en el diccionario:
    - pero no contiene la letra central → debe indicarlo al usuario.
- Si la palabra se encuentra en el diccionario y contiene la letra central:
    - pero no contiene algunas de las letras dadas → debe indicarlo al usuario.
    - y si contiene las letras dadas → debe indicarlo al usuario y sumar 5 puntos a la puntuación.
- Si el usuario repite una palabra, ésta no debe sumar puntos. En tal caso se le indicará al usuario que ya ha sido introducida.
- Si el usuario encuentra todas las palabras que puede formar con ese conjunto de letras, se le indica al usuario y termina la partida.

Además, de proponer palabras puede pedir ayudas mediante comandos:

- @ayuda: Debe explicar el juego y las ayudas posibles
- @salir: Sale del programa.
- @pista: Dado el conjunto de letras con las que formar una palabra, mostrar todas las posibles combinaciones de las dos primeras letras y el número de veces que aparecen en el diccionario teniendo en cuenta el máximo tamaño de la palabra.

  Ejemplo:

  Empiezan por: al --> aparece: 7 veces
  Empiezan por: az --> aparece: 3 veces
  Empiezan por: ca --> aparece: 12 veces
  Empiezan por: cl --> aparece: 1 veces
  Empiezan por: co --> aparece: 3 veces
  Empiezan por: cu --> aparece: 4 veces
  Empiezan por: ja --> aparece: 1 veces
  Empiezan por: la --> aparece: 1 veces
  Empiezan por: lo --> aparece: 2 veces
  Empiezan por: lu --> aparece: 1 veces
  Empiezan por: oc --> aparece: 2 veces
  Empiezan por: uz --> aparece: 1 veces
  Empiezan por: za --> aparece: 3 veces
  Empiezan por: zo --> aparece: 5 veces
  Empiezan por: zu --> aparece: 4 veces
- @encontradas: Muestra todas las palabras encontradas/acertadas con ese conjunto de letras.

- @mostrar: Muestra la cantidad de palabras posibles que existen con ese conjunto de letras, teniendo en cuenta que siempre contendrá la letra central. Además de mostrar también el número de combinaciones que faltan.
- @rendicion: Muestra la todas las posibles palabras que se pueden formar con ese conjunto de letras y contengan la letra central.
- @puntuacion: Muestra la puntuación obtenida al usuario. Tener en cuenta que, si se repite una palabra, ésta no debe sumar puntos.

Opciones adicionales

Cualquier opción adicional se valorará de forma extra. Cumplir únicamente con los requisitos mínimos implica que la nota máxima alcanzable es 7. Para obtener mejor nota es necesario realizar opciones adicionales.
Algunas posibilidades son:

- Permitir al usuario realizar una partida tras otra.
- Tratar las letras ñ/ç y las mayúsculas de forma correcta.
- Permitir jugar dos jugadores (o más jugadores), cada jugador va indicando una palabra de forma alternativa. Al terminar la partida muestra el ganador (palabras más acertadas).
- Permitir elegir al usuario el tamaño máximo de letras con las que va a poder formar palabras, y por tanto indicar el tamaño con el cuál se va a jugar.

Añadir ayudas adicionales, como:

- @ranking: muestra la puntuación de la partida actual y partidas anteriores, ordenadas de mejor a peor puntuación.
- @shufle: reordena las letras manteniendo la posición de la letra central.
- @reiniciar: terminar partida actual y cargar nueva partida con un nuevo conjunto de letras.
- @recuperar: muestra el conjunto de letras a jugar.
- @idioma: el usuario puede elegir el idioma con el que va a jugar. Tener en cuenta que el diccionario debe ser coherente con el idioma escogido.
- @dificultad: permitir al usuario diferentes niveles de dificultad. Como, por ejemplo, incluir un nivel donde las palabras no acertadas resten puntuación.
- @intentos: limitar el número de intentos, teniendo en cuenta siempre el total de palabras que se pueden formar con ese conjunto de letras dado.
- Cualquier otro comando de ayuda propuesto por el alumno.

Recordatorio Importante

- El juego PARAULOGIC debe tener tamaño 7, y debe ser configurable con una constante llamada TAM_PALABRA. **const int TAM_PALABRA = 7;**
- El juego debe ser capaz de cargar cualquier fichero en cualquier ordenador, independientemente del nombre.
- Para la implementación el alumno debe de ser capaz de usar strings, vectores y maps.