

Universidad de San Carlos de Guatemala

Facultad de Ingeniería

Introducción a la Programación y Computación I

Sección: "D"

Cat. Ing. Herman Igor Veliz Linares

Auxiliar: Jorge Daniel Monterroso Nowell

MANUAL TECNICO JUEGO DE CONSOLA "SOPA DE LETRAS"

Marvin Eduardo Catalán Véliz

201905554

Guatemala, 8 de marzo de 2020

Índice

REQUERIMIENTO TÉCNICO PARA EJECUCION	3
HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO	3
JAVA	3
¿Qué es?	3
Versión	3
IDE	3
¿Qué es?	3
IDE utilizado: NetBeans	3
FUNCIONALIDAD Y EXPLICACION	3
Variables globales	3
Método: menuPrincipal()	4
Método: menu_nuevaPartida()	4
Método: menú_menuPalabras()	4
Función: abc()	5
Método: matrizAleatoria()	5
Método: insertar()	5
Método: llenarVectorPalabras()	5
Función: numAleatorio()	5
Función: num()	5
Método: insertarPalabras()	5
Método: jugar()	6
Método: informaci()	6

REQUERIMIENTO TÉCNICO PARA EJECUCION

- Máquina virtual de Java

HERRAMIENTAS UTILIZADAS PARA EL DESARROLLO

JAVA

¿Qué es?

Java es un lenguaje de programación y una plataforma informática comercializada por primera vez en 1995 por Sun Microsystems. Hay muchas aplicaciones y sitios web que no funcionarán a menos que tenga Java instalado y cada día se crean más. Java es rápido, seguro y fiable. Desde portátiles hasta centros de datos, desde consolas para juegos hasta súper computadoras, desde teléfonos móviles hasta Internet, Java está en todas partes.

Versión

La versión que se utilizó para la creación de este programa es *java version "1.8.0_111"*

IDE

¿Qué es?

en inglés Integrated Development Environment (IDE), es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador el desarrollo de software.

Normalmente, un IDE consiste de un editor de código fuente, herramientas de construcción automáticas y un depurador.

IDE utilizado: NetBeans

El IDE o entorno de desarrollo utilizado en esta practica fue Netbeans versión 8.2

FUNCIONALIDAD Y EXPLICACION

El programa está desarrollado en su mayoría por métodos y una que otra función, se utilizaron para tratar de obtener un código más ordenado y también para el momento de que se encuentre un error, su solución se facilitara más porque se trabaja por bloques.

Variables globales

Se declararon diferentes variables de alcance global, de diferentes tipos, por la misma razón de que se utilizó métodos para escribir el código, entonces se necesito las variables globales para poder comunicar métodos y funciones entre sí, con números, cadenas, datos, etc. Para una ejecución efectiva.

Método: `menuPrincipal()`

En este método se encuentran todos los demás métodos del programa. Principalmente la funcionalidad de `menuPrincipal()` es mostrar el primer menú del juego, que dentro de ese menú se encontraran diferentes menús añadidos.

Solicita al usuario una opción del menú mostrado, y si el número no se encuentra dentro del rango valido se repetirá el menú hasta que se contradiga la condición del `while`.

Con el número ingresado por el usuario el método se dirige hacia un `switch` que consta de 7 casos diferentes de los cuales se hablará más a detalle a continuación.

En el caso 1 solicita al usuario que ingrese su nombre y luego le solicita que ingrese el valor de el número de filas y columnas que desea para su sopa de letras, con la condición de que ese número no sea menor a 5, luego se hace un llamado a `menu_nuevaPartida()`

En el caso 2 se hace llamado a el método `historialDePartidas()`

En el caso 3 se hace llamado a el método `mejoresPuntuaciones()`

En el caso 4 se hace llamado a el método `perdedores()`

En el caso 5 se hace llamado a el método `ganadores()`

En el caso 6 se hace llamado a el método `informacion()`

En el caso 7 se hace llamado a el método `salir()`

Método: `menu_nuevaPartida()`

Como este método esta ingresado en un caso de un método anterior antes de ejecutarse el mismo espera que todo lo que esta encima en ese caso se ejecute, entonces la espera que el usuario ingrese su nombre y número N.

Como su mismo nombre lo dice este método es un menú que muestra 3 posibles elecciones del usuario y solicita que ingresen una de ellas. Al momento de que el usuario ingrese el número de su elección `menu_nuevaPartida()` ingresa a un `switch` con 3 diferentes casos de ejecución.

Caso 1 `menu_menuPalabras()`

Caso 2 `jugar()`

Caso 3 `menuPrincipal()`

Si observamos nos damos cuenta que en este método podemos ingresar a otro menú más dentro del programa y también podemos jugar, pero además si es decisión de usuario también puede retornar.

Método: `menú_menuPalabras()`

Este método, es un tercer menú del programa de sopa de letras , este muestra 5 diferentes opciones a elegir, esta opción es ingresada al usuario a través de un Scanner o lectura, que luego se complementa con su ejecución en la entrada a un `switch` que se encuentra abajo para la ejecución de los diferentes casos del tercer menú mostrado.

Función: abc()

Esta función esta conformada por un vector de tipo carácter que contiene a todas las letras del abecedario, y también por una variable llamada *posición* esta variable a través del comando `Math.random` y `Math.floor` guarda una posición aleatoria del vector de caracteres del abecedario.

Como esta es una función retorna algún valor o algo que sea de un interés de volver a utilizar, retorna un carácter del abecedario al azar generado por un numero aleatorio.

Método: matrizAleatoria()

Este método de entrada crea una nueva matriz del tamaño ingresado por el usuario anteriormente. Por medio de 2 ciclos que recorren cada posición de filas y columnas de la matriz se va llenando la matriz con lo que nos retorna la función `abc()` creada anteriormente.

Método: insertar()

Este método tiene como función preguntarle al usuario con cuantas palabras quiere jugar, y además este mismo método luego de preguntarle eso, le dice al usuario que escriba ese número de palabras.

Método: llenarVectorPalabras()

Al momento de que el usuario ingrese cada palabra se encuentra un ciclo `for` trabajando para convertir esas palabras ingresadas en un vector del tamaño del numero de palabras que el usuario ingreso. Este mismo ciclo evalua la condición de que la palabra sea mayor a 5 .

Cuando finaliza nos retorna a `menu_menuPrincipal`.

Función: numAleatorio()

Este método nos genera un número aleatorio de 1 hasta N(numero tamaño de tablero ingresado por usuario) qué nos servirá en un futuro para diferentes métodos.

Función: num()

Genera un numero aleatorio de 0 a 4 utilizado en un método más adelante para escoger en un switch aleatoriamente un caso.

Método: insertarPalabras()

En insertar palabras se manda a llamar al método de `matrizAleatoria()` porque se sobrescribirá sobre ella las nuevas palabras que el usuario ingreso para poder jugar.

La forma en que se ingresaran las palabras será a travez de un casteo dentro de un ciclo, que analizara cada palabra carácter por carácter y lo colocara dentro de la matriz, pero sin volver la palabra un vector, sino que solamente analizándolo carácter por carácter con el comando `toCharArray()`.

Hay 4 casos distintos de que las palabras ingresen a la matriz. Variando un contador en el numero de filas o columnas y sumándole o restándole a alguna de las dos(dependiendo del caso) se logra llegar a 4 casos que pueden ingresar carácter por carácter de cada letra de izquierda a derecha o viceversa , o también de arriba hacia abajo o viceversa.

Cabe recalcar que la forma en que ingrese cada palabra a la matriz será aleatoria por el numero aleatorio que escoge el caso del switch.

Método: jugar()

Este método funcionara si ya el usuario ha ingresado palabras y además las ha insertado, porque este método hace llamado a matrices y vectores necesarios para la jugabilidad, pero si estos están vacíos no se podrá jugar.

Lo primero que realiza este método es indicarle al usuario que busque palabras en su tablero y que las escriba, entonces ya cuando las escriba tocaría compararlas o tratar de encontrarlas de alguna forma para poder tacharlas con signos de dólar.

Este método consta de 4 casos que están pensados para que barran la matriz de 4 formas distintas en busca de la combinación de caracteres que formen la palabra ingresada anteriormente por el usuario.

Estos métodos básicamente tienen exactamente la misma funcionalidad, que es recorrer la matriz en una dirección en específico comparando carácter por carácter hasta que encuentra una cadena de caracteres colocados uno al lado del otro formando la palabra que ingreso el usuario. Al tachar la palabra hay contadores verificadores, que se suman o restan depende si el usuario está ingresando bien las palabras o no, porque el usuario al cometer 3 errores perderá la partida.

Método:informaci()

Este método imprime información sobre el creador del programa.