

Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Estructura de Datos
Sección B
Ing. Carlos Alonzo
Aux. Marco Galindo



PRACTICA #1

Pseudo-Scrabble

Objetivos

- Crear una implementación de memoria dinámica a través de objetos en Java.
- Comprender y elaborar listas dinámicas, colas y matrices
- Definir algoritmos de búsqueda y recorridos para las listas

Descripción

Una empresa local ha decidido crear su propia versión del famoso juego Scrabble y para esto ha decidido contactarlo a usted, ingeniero de la famosa universidad de San Carlos de Guatemala, para realizar dicho juego.

Implementación

Se deberá crear una aplicación Java de escritorio que cumpla la descripción. Esta aplicación cuenta con 3 módulos principales, la lectura del archivo de entrada, el juego y la muestra de reportes. Se utilizarán listas simples, circulares, colas y una matriz para la lógica del juego.

El menu principal

La primer pantalla que debemos de ver será el menú principal. Este tendrá 2 opciones, leer el archivo de entrada y jugar. Si no hemos leído ningún archivo de entrada, no podremos seleccionar la opción de entrar al juego.



Archivo de Entrada

Al seleccionar la opción “LEER ARCHIVO” deberemos de seleccionar un archivo .xml.

El archivo de entrada sera un XML con los siguientes datos:

- Las dimensiones del tablero de juego, el cual será un cuadrado.
- Las casillas que cuentan con doble y triple puntuación.
- El diccionario de palabras válidas del juego

Las palabras del diccionario deben de guardarse en una **lista simple**.

Ejemplo de un XML de entrada

```
<scrabble>
  <dimension>20</dimension>
  <dobles>
    <casilla>
      <x>2</x>
      <y>5</y>
    </casilla>
    <casilla>
      <x>7</x>
      <y>9</y>
    </casilla>
    ...
  </dobles>
  <triples>
    <casilla>
      <x>20</x>
      <y>20</y>
```

```

        </casilla>
        <casilla>
            <x>1</x>
            <y>1</y>
        </casilla>
        ...
    </triples>
    <diccionario>
        <palabra>hola</palabra>
        <palabra>mundo</palabra>
        <palabra>como</palabra>
        <palabra>estas</palabra>
        <palabra>uno</palabra>
        <palabra>dos</palabra>
        <palabra>tres</palabra>
        <palabra>universidad</palabra>
        <palabra>practica</palabra>
        <palabra>primera</palabra>
        <palabra>correo</palabra>
        <palabra>juego</palabra>
        ...
    </diccionario>
</scrabble>

```

NOTA: Las secciones <scrabble>, <dimensión>, <dobles>, <triples> y <diccionario> solo van a aparecer **UNA** vez, así que no se deben de preocupar de varias ocurrencias de estas.

Ingreso de Jugadores

Ya cuando podamos jugar, nos deberá salir la opción de agregar jugadores, los cuales únicamente tendrán nombre de usuario, el cual no debe de estar repetido.



Los jugadores deben de guardarse en una **lista circular**.

El Juego

Scrabble es un juego de mesa en el cual cada jugador intenta ganar más puntos mediante la construcción de palabras sobre un tablero de cuantas casillas indique el archivo de entrada. Las palabras pueden formarse horizontalmente o verticalmente siempre y cuando aparezcan en el diccionario ingresado anteriormente.

Las fichas de juego

El juego cuenta con diferentes fichas representando letras del alfabeto para formar palabras. Cada una de estas fichas tiene una puntuación diferente.

1 punto: A ×12, E ×12, O ×9, I ×6, S ×6, N ×5, L ×4, R ×5, U ×5, T ×4

2 puntos: D ×5, G ×2

3 puntos: C ×4, B ×2, M ×2, P ×2

4 puntos: H ×2, F ×1, V ×1, Y ×1

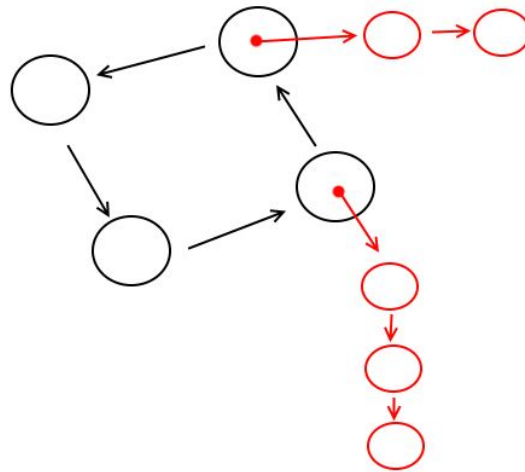
5 puntos: Q ×1

8 puntos: J ×1, Ñ ×1, X ×1

10 puntos: Z ×1,

Todas las fichas estaran dentro de **una cola**. Las fichas serán ingresadas al azar.

Cada jugador tendrá 7 fichas en su mano, las cuales deberán estar en una **lista simple**, dentro de los nodos de la **lista circular** de usuarios.



Mecánica del juego

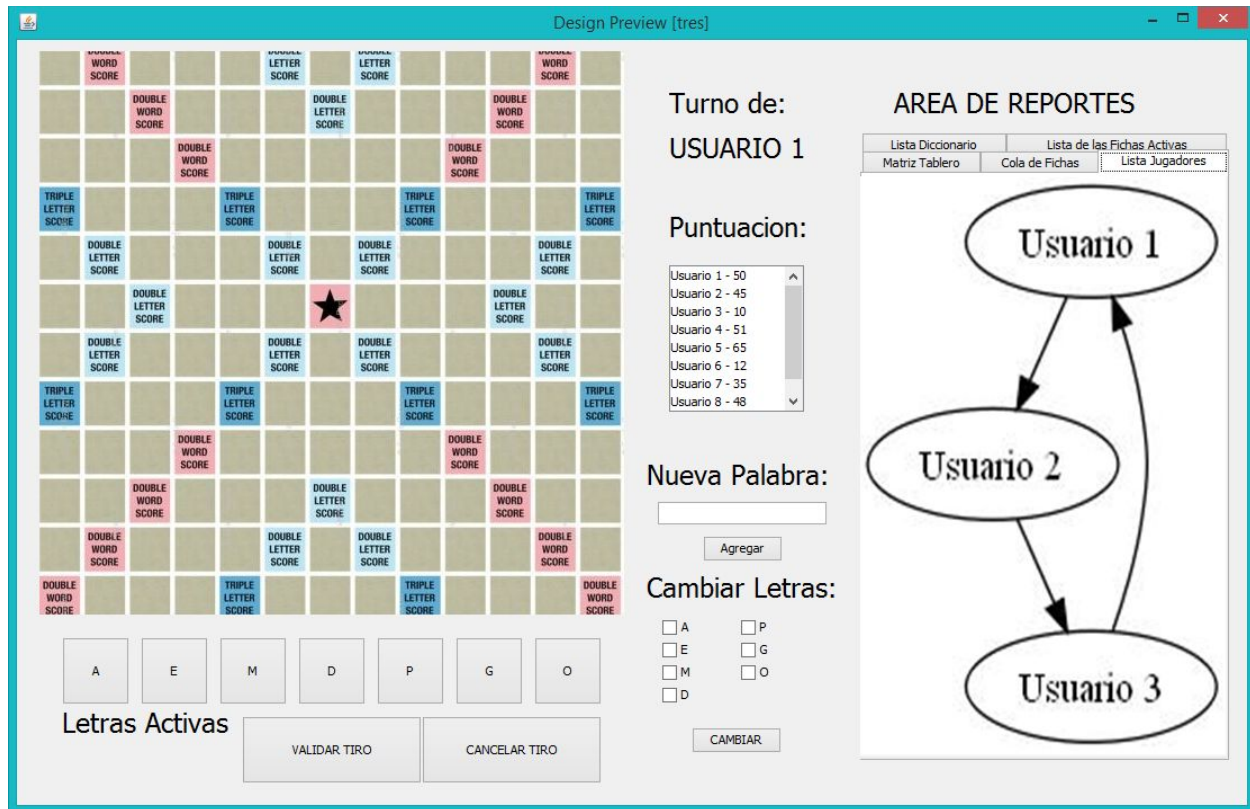
El objetivo del juego es crear palabras para obtener puntos. El valor de una palabra es igual a la suma de los valores individuales de las letras que la componen. Si se utiliza alguna casilla que tenga doble o triple puntuación, el multiplicador será aplicado en el **valor individual de la letra** que esté en la casilla, **no al valor total de la palabra**.

Se comienza el juego con un jugador al azar, y este solo puede formar 1 palabra por turno. Si el jugador no puede formar ninguna palabra puede intercambiar algunas de sus fichas por otras, esto también acabara su turno y las fichas que él tenía, regresaran a la cola de fichas.

El Tablero

Interfaz de usuario sugerida.

La **matriz** del juego debe de ser una clase creada por el estudiante.



- Para colocar una letra en el tablero se deberá de usar la funcionalidad “**drag and drop**”.
- Las “letras activas” representan las 7 fichas del usuario con el turno activo.
- Para hacer más fácil la calificación, se tendrá una opción de “Nueva Palabra”, la cual agregara una palabra nueva al diccionario.
- Al dar click en “validar tiro”, se deberá de comparar el tiro actual con las palabras existentes en el diccionario. Si el tiro es válido, se suman los puntos, se llena su mano con más letras de la cola y se cambia de turno. Si el tiro es inválido, deberá mostrar un error.
- Al dar click en “cancelar tiro”, se deberán de cancelar todos los movimientos actuales en el tablero, es decir, si ya se colocaron letras, estan deberán de regresar a la mano del jugador.
- Se pueden cambiar letras por otras de la cola de letras, seleccionando por medio de checkboxes las letras que se desean cambiar. Estas opciones también cambian dependiendo del jugador activo.

Fin del Juego

El juego debe de finalizar cuando se acaben las fichas en la cola o se escoja finalizar el juego (con un botón o de la manera que el estudiante vea conveniente). Se deberá de

mostrar una ventana con el ganador (la persona que tenga más puntos) y el tiempo total que duró el juego.

Reportes

El área de reportes debe ser dinámica, en tiempo real. Los reportes deben estar hechos en Graphviz y se deben de mostrar como se ve en la interfaz gráfica. Para que los reportes se muestran en tiempo real, se deberá utilizar hilos que están refrescando la imagen que nosotros seleccionemos en las pestañas.

Las gráficas que se deben de mostrar son:

- La cola de fichas pendientes por sacar
- La lista circular de los usuarios
- La lista simple de las fichas del jugador activo
- La lista simple del diccionario de palabras
- La matriz

Ejemplos de los reportes serán dados en clase.

Restricciones

Los reportes son esenciales para verificar si se trabajó correctamente la estructura, así que **sin reportes, no se calificará el proyecto.**

Observaciones

- Las estructuras deben de ser realizadas por el estudiante. No está permitido el uso de ArrayLists, librerías de colas, librerías de matrices ni de arreglos estáticos.
- Lenguaje a utilizar: Java, cualquier IDE.
- Se calificará **únicamente** archivos .jar
- Se debe de crear un repositorio en GitHub con el nombre de Practica1s12017_#carnet.
- Los documentos deben estar sincronizados don GitHub así mismo teniendo un historial de que trabajo en dicha plataforma
- Forma de entrega: vía GitHub
- Fecha de Entrega: 20 de Febrero del 2017 a las 11:59 P.M. (Última sincronización de GitHub).
- **COPIAS SERÁN PENALIZADAS con una nota de 0.**
- **COPIAS SERÁN PENALIZADAS conforme al reglamento.**

