

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

FACULTAD DE INGENIERÍA EN ELECTRICIDAD Y COMPUTACIÓN

PROGRAMACION ORIENTADA A OBJETOS

SEGUNDA EVALUACIÓN – 2012-II - 2013-01-28

Nombre: _____ Matrícula _____ Paralelo _____

(100 PUNTOS) Usted está aplicando para trabajar en una compañía dedicada al desarrollo de videojuegos y le solicitan implementar el juego "Buscaminas POO" el mismo consiste de una matriz de celdas N filas por M columnas, en la cual se colocará de manera aleatoria las minas en las celdas; es decir, hay celdas que contienen minas y celdas sin minas.

Cuando la ventana del juego se abre al usuario se le muestra un cuadro de dialogo en el cual el jugador debe escoger el tipo de nivel: Básico, Intermedio y Experto. Una vez que el jugador selecciona el nivel el juego genera el tablero de acuerdo al nivel:

- Para nivel Básico el tablero será de 9 x 9 y 10 minas distribuidas de forma aleatoria.
- Para nivel Intermedio el tablero será de 16 x 16 y 40 minas distribuidas de forma aleatoria.
- Para nivel Avanzado el tablero será de 16 x 30 y 99 minas distribuidas de forma aleatoria.

El jugador descubrirá las celdas haciendo uso del ratón (evento clic) luego de lo cual el juego verifica si la celda está vacía, contiene una mina o contiene un número que indica la cantidad de minas cercanas.

- Quando la celda descubierta contiene una mina se le presenta al jugador una "x" en el lugar donde dio clic. Adicionalmente, se le muestra al jugador un cuadro de dialogo indicándole que perdió el juego presentando el mensaje "BOOM, acaba de pisar una mina...!".
- Si la celda descubierta contiene un número el usuario puede seguir jugando.
- Si la celda descubierta se encuentra vacía, es decir no contiene ni un número ni una mina, se descubren de forma recursiva todas las celdas adyacentes a ella que están vacías o contienen números; las celdas con minas no se descubren. Un celda tiene adyacencia con las 8 celdas a su alrededor.

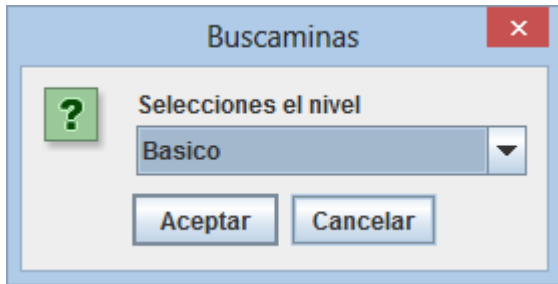
1	2	3
8		4
7	6	5

El juego finalizará cuando todos los casilleros que no contienen minas han sido descubiertos y se le presentara un mensaje al jugador diciendo "Ganaste".

Su supervisor quiere saber si usted cuenta con todos los conocimientos de la Programación Orientada a Objetos requeridos para estar en el cargo, por lo cual le entrega el siguiente diagrama de clases del juego y le pide que su implementación sea acorde a los requerimientos que se detallan más adelante.

Clase Buscaminas

- a) (5pts.) Cree la clase Buscaminas y su constructor en el cual se establecen las filas y columnas del tablero así como la cantidad de minas que estarán ocultas. Adicionalmente, establezca el título de la ventana a "Buscaminas POO" y que por defecto cierre la aplicación al salir.



- b) (5pts.) Cree el método "main" en el cual deberá solicitarle al usuario mediante un cuadro de dialogo el nivel en el que desea jugar y generara el tablero de celdas al construir un objeto Buscaminas. Según el nivel seleccionado se instanciara la ventana del juego para luego hacerla visible.

- c) (5 pts.) Crear el método "private void inicializarComponentes()" el mismo que en base a las variables de instancia genera el tablero del juego.

La representación lógica de la colección de celdas se la realiza mediante la implementación de la variable de instancia **HashMap<Point,CeldaUI> celdas;** en la cual el tipo Point representa la coordenada de la celda en la matriz y a la vez es un atributo de la celda.

Usted cuenta con el método "**private static setMinas(Map<Point,CeldaUI> celdas, int nMinas)**" el cual llena de minas y números al tablero. No olvide que el tamaño del tablero debe ser acorde a los parámetros de inicialización.

- d) (1 pt.) Invoque el método del literal (c) desde el constructor de la clase.
- e) (5 pts.) Cree el método "private CeldaUI getCelda(HashMap<Point, CeldaUI> tablero, int x, int y)" el cual retorna la celda ubicada en la coordenada (x,y) del tablero.
- f) (20 pts.) Cree el método "private void setMinas(Map<Point,CeldaUI> tablero, int nMinas)" el cual puebla el tablero con minas en posiciones aleatorias según la cantidad de minas pasadas por parámetro. Considere el literal (h) en su implementación.
- g) (15 pts.) Cree el método "private void setObservadores(Map<Point,CeldaUI> tablero)" el cual puebla el tablero con minas en posiciones aleatorias según la cantidad de minas pasadas por parámetro. Considere el literal (h) en su implementación.

Clase CeldaUI

- h) (5 pts.) Cree la clase CeldaUI la cual en el constructor solamente recibe la coordenada respecto a la matriz en una de sus variables de instancia. Adicionalmente, contiene una referencia a un objeto Celda. No olvide establecer el tamaño de la CeldaUI.
- i) (5 pts.) Cuando la celda no ha sido descubierta visualmente el jugador vera un recuadro de color negro con un borde blanco. En caso de que la celda esta descubierta se le deberá mostrar un recuadro de color negro con fondo blanco ya sea vacío, con un número o con un 'X' según sea el caso.
- j) (3 pt.) Cree el setter y el getter para exponer la variable de instancia del objeto Celda asociado que indica si es una mina o no. Adicionalmente, cree un getter para exponer la celda asociada.

Clase Celda

- k) (5 pts.) Cree la clase Celda como clase interna a CeldaUI y utilizando el patrón Observador visto en clase permita que esta clase Observe y sea Observada. Adicionalmente, una variable de instancia que indica si es una mina o no, otra si esta descubierta o no y una en la que almacene un valor que puede ser un numero o una 'x'.
- l) (10 pts.) Cree el método descubrirCelda el cual descubrirá la celda:
- En caso de que sea una mina mostrara un cuadro de dialogo indicando que ha perdido y cerrara el juego.

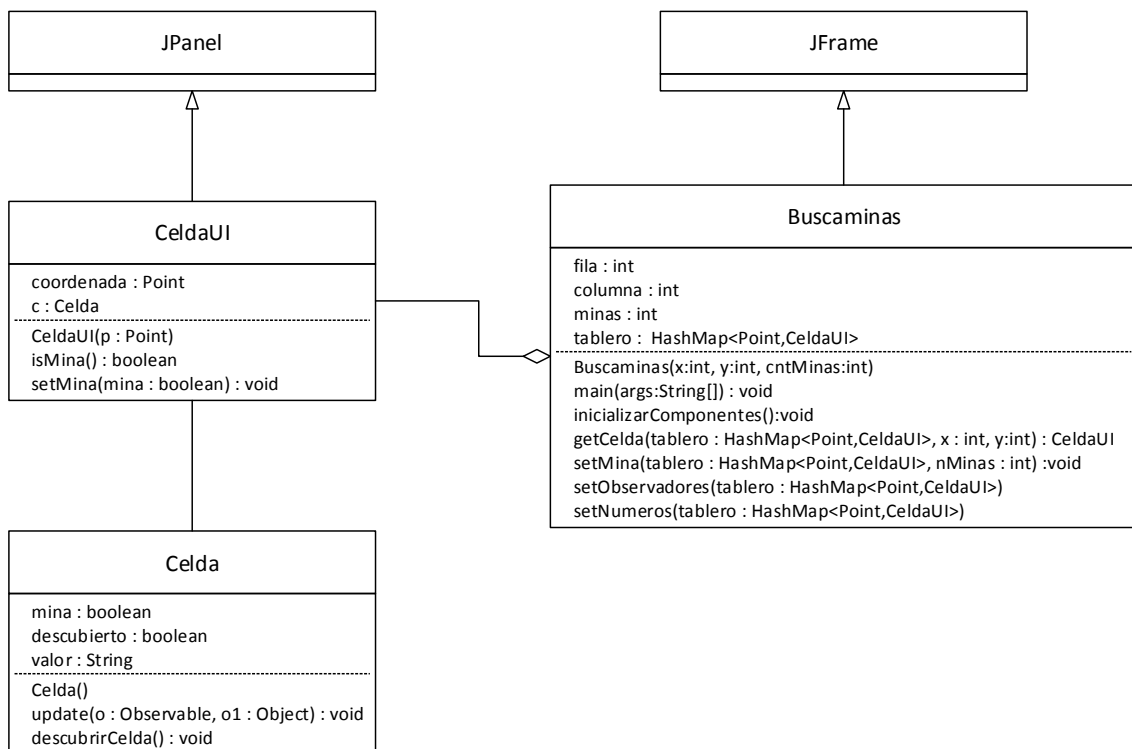
- Caso contrario se descubre la mina, se actualiza visualmente, se establece que ha cambiado y se notifica a las celdas adyacentes que están observando algún cambio en la celda que se está descubriendo.
- En caso de que la celda ya se encuentre descubierta se deber lanzar la excepción CeldaDescubiertaException en la cual el mensaje indicara la celda "(x,y)" ya se encuentra descubierta y no se hace nada.

m) (5 pts.) Cree una excepción llamada CeldaDescubiertaException.

n) (3 pts.) Implemente el método update del patrón Observador

Clase CeldaUI

o) (10 pts.) Haciendo uso de clases anónimas maneje el evento de dar clic sobre un objeto de tipo CeldaUI en el cual se procederá a descubrir el objeto Celda asociado a la CeldaUI. En caso de ser lanzada la excepción CeldaDescubiertaException manéjela y muestre el mensaje haciendo uso de un cuadro de dialogo.



Referencias

JOptionPane

- static void showMessageDialog(Component c, Object texto, String titulo, int tipoMensaje)
- static String showInputDialog(Component c, Object texto, String titulo, int tipoMensaje, Icon icono, Object[] lista, Object seleccionado)

Clase java.util.Observable

- void addObserver(Observer o)
Adds an observer to the set of observers for this object, provided that it is not the same as some observer already in the set.
- boolean hasChanged()
Tests if this object has changed.
- void notifyObservers()
If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed.
- void notifyObservers(Object arg)
If this object has changed, as indicated by the hasChanged method, then notify all of its observers and then call the clearChanged method to indicate that this object has no longer changed.
- protected void setChanged()
Marks this Observable object as having been changed; the hasChanged method will now return true.

Interface java.util.Observer

- void update (Observable o, Object data)
This method is called whenever the observed object is changed. An application calls an Observable object's notifyObservers method to have all the object's observers notified of the change.
Parameters: o - the observable object. arg - an argument passed to the notifyObservers method.

Clase Random

- int nextInt(int n)
Returns a pseudorandom, uniformly distributed int value between 0 (inclusive) and the specified value (exclusive), drawn from this random number generator's sequence.