22-3-2017

Erick Efrain Vargas Romero

Prof. Tecla Parra Roberto

Asignatura: Programación Orientada a Objetos

Grupo: 2CM4

Práctica 2

Arreglos de Objetos

Fecha: 23/03/2017

Arreglos de Objetos

Práctica 2

# **Objetivo**

Al igual que podemos crear arreglos de elementos tipo entero, booleanos, flotantes, etc. Es posible crear arreglos de objetos de cualquier tipo, pero un objeto común es el botón. Siendo así, es posible reducir considerablemente código si hacemos uso de arreglos de objetos tipo button que nos faciliten la comprensión y manejo del código.

# **Desarrollo**

Para esta práctica se eligió el programa de golpear al topo o bien “Whack a mole” la cual consiste en colocar imágenes de topos en un tablero de 5x4 de manera aleatoria, al hacer click sobre un botón que tiene a un topo se incrementará la puntuación del jugador.

Primeramente, cabe destacar que para esta práctica se hace uso de swing para manejar la interfaz. Como es evidente se ha creado una clase llamada WhackAMole la cual extiende de un Applet e implementa la interfaz ActionListener para el manejo de eventos. Posteriormente tenemos nuestras variables de instancia las cuales son privadas, teniendo así, dos objetos tipo Label con los cuales mostraremos en uno la puntuación del jugador y en el otro si es que ya ha ganado, también tenemos dos variables de tipo entero en una de estas tenemos la puntuación del jugador y en la otra el número de topos que hay en el tablero, finalmente, como el título de esta práctica lo indica se ha creado un arreglo de objetos tipo JButton el cual utilizaremos para el tablero.

A continuación tenemos el método init() con el cual inicializaremos el applet, y crearemos todo lo necesario para que nuestro programa funcione como debe ser, pero a diferencia de la práctica anterior se hace uso de un GridLayout para tener en cuadricula diferentes elementos, esta cuadricula es de 5x4 elementos, tal como se muestra en la siguiente línea de código:

setLayout**(new** GridLayout**(**5**,**4**));**

Con esta línea tenemos cuadriculado nuestro applet y podemos colocar todos los botones que serán creados. Hay que hacer hincapié en que primeramente se debe apartar memoria para nuestro arreglo de botones, y posterior mente crear cada uno de los botones y manipularles si es necesario, para hacer esto, se ha utilizado un ciclo for() y dentro de él se han creado cada uno de los botones y añadido al applet, pero no solo eso, se ha colocado el ícono de un agujero en todos los botones y finalmente se ha evaluado con un random si se colocará un topo o no, si esta sentencia es verdadera nuestra variable que lleva el conteo de los topos que hay en el tablero se autoincrementa en una unidad, además la imagen de nuestro botón es cambiada por la de un topo y finalmente se le añade al botón el ActionListener, este trozo de código se muestra a continuación

**for** **(**int i**=**0**;** i**<**buttons**.**length**;** i**++)** **{**

add**(**buttons**[**i**]** **=** **new** JButton**(**""**));**

buttons**[**i**].**setIcon**(new** ImageIcon**(**"Hole.jpg"**));**

buttons**[**i**].**setPreferredSize**(new** Dimension**(**20**,**20**));**

**if(**Math**.**random**()** **<** 0.5**){**

moles**++;**

buttons**[**i**].**setIcon**(new** ImageIcon**(**"Mole.jpg"**));**

**}**

buttons**[**i**].**addActionListener**(this);**

**}**

En el método init() también son añadidos nuestros Label.

Continuando tenemos el método actionPerformed(ActionEvent e) en el cual realizaremos lo debido para cada ocasión en la que el usuario presiona un botón del tablero, esto primeramente lo hacemos obteniendo la fuente del evento, y para saber si el botón que fue presionado es de un topo simplemente obtenemos el ícono que fue colocado en el botón y le transformamos a un String y finalmente comparamos el nombre, si el nombre es “Mole.jpg” entonces cambiamos el ícono al de un topo golpeado y removemos el ActionListener() del botón, además la puntuación se autoincrementa en uno, tal y como se muestra a continuación.

**if(**button**.**getIcon**().**toString**().**equals**(**"Mole.jpg"**)){**

button**.**setIcon**(new** ImageIcon**(**"DeadMole.jpg"**));**

button**.**removeActionListener**(this);**

punctuation**++;**

**}**

Para finalizar, se hace una última evaluación, si nuestros contadores de topos y el de la puntuación son iguales se muestra el mensaje has ganado, salimos de esta evaluación y se actualiza el Label que mostrará la puntuación.

# **Conclusión**

Como es apreciable, es mucho más sencillo el crear arreglos de objetos ya que reducimos de manera significativa nuestro código, además es más sencillo crear un arreglo, y después cada botón utilizando un ciclo de repetición que estar creando botón por botón, lo cual nos facilita de manera abismal el trabajo.