

Universidad Autónoma de Baja California

Facultad de Ingeniería

Laboratorio de Programación Orientada a Objetos Avanzada
Prof. Maria Luisa Gonzalez Ramirez

Practica No. 3
Interfaz de usuario y lógica del juego

Angelica Figueroa Muñiz

Mexicali, Baja California a 14 de Marzo de 2018

Descripción de las clases

Para la interfaz gráfica del usuario se crearon las clase TablaGrafica e InterfazJugador. Dentro de TablaGrafica se creó el diseño de la tabla del jugador con las 16 imágenes de las cartas aleatorias. Esta clase incluye como atributo un objeto de tipo Tabla, del cual se obtienen las imágenes y las frases que se muestran al iniciar el juego.

Se agregaron 16 jButtons y a cada uno se le asignó como ícono la imagen de la carta correspondiente. Al presionar los botones, se marcan con una X y se deshabilitan, a su vez se van agregando los nombres de los personajes presionados a un ArrayList de tipo String para posteriormente comprobar la lotería.

También se incluyeron jLabels para mostrar la imagen de la carta que van saliendo del bonche (mazo) y para mostrar la frase de la carta. Para mostrar las cartas que salen del bonche se creó un timer en el cual se obtiene una carta del bonche con el método obtenerCarta y se obtiene su imagen y su frase. Por último se tienen dos jButton, uno con la función de iniciar el juego y empezar a mostrar las cartas sacadas y el segundo para que el jugador declare cuando obtuvo lotería y se haga la comprobación de las cartas.

Para la comprobación de la lotería se tiene el método comprobarLoteria, el cual recibe dos ArrayList de tipo String, una con los nombres de las cartas de la tabla (marcadas) y otro con los nombres de las cartas que fueron sacadas del bonche (sacadas), se valida que los ArrayList cumplan con las características necesarias y posteriormente se revisa que el ArrayList sacadas contenga todos los elementos del ArrayList marcadas. El constructor de la clase TablaGrafica recibe como parámetro un objeto de tipo Bonche que es enviado desde la clase InterfazJugador.

La clase InterfazJugador cuenta con un jButton para iniciar y crear una tabla nueva y otro para terminar y salir. Al presionar el botón de iniciar se crea una nueva tabla y se muestra. A esta tabla se le manda un objeto Bonche previamente creado y barajado.

Código de las clase

```
20 public class TablaGrafica extends javax.swing.JFrame {
21
22     private static Bonche bonche;
23     private Tabla tabla = new Tabla();
24     private String numero = Integer.toString(tabla.getNumeroDeTabla());
25     private ArrayList<String> marcadas = new ArrayList<String> ();
26     static private ArrayList<String> sacadas = new ArrayList<String> ();
27     static int count = 0;
28     private Timer timer = new Timer(1800, new ActionListener() {
29         @Override
30         public void actionPerformed(ActionEvent ae) {
31             if (count == 54) {
32                 timer.stop();
33             }else{
34                 Carta c = bonche.obtenerCarta();
35                 sacadas.add(c.getPersonaje().getNombre());
36                 Picture p = new Picture(c.getPersonaje().getImagen());
37                 Image img = p.getImage();
38                 Image newI = img.getScaledInstance(jLabel3.getWidth(), jLabel3.getHeight(), java.awt.Image.SCALE_SMOOTH);
39                 jLabel3.setIcon(new ImageIcon(newI));
40                 frase.setText("<html>\t" + c.getFrase() + "</html>");
41                 count++;
42             }
43         }
44     });
45
46     public TablaGrafica(Bonche bonche) {
47         initComponents();
48         this.bonche = bonche;
49     }
50
51     public ArrayList<String> getMarcadas() {
52         return marcadas;
53     }
54
55     public ArrayList<String> getSacadas() {
56         return sacadas;
57     }
58
59     public boolean comprobarLoteria(ArrayList<String> marcadas, ArrayList<String> sacadas) {
60         if (marcadas == null && sacadas == null) {
61             return false;
62         }
63         else if (marcadas == null && sacadas != null || marcadas != null && sacadas == null || marcadas.size() != 16) {
64             return false;
65         }else
66             return sacadas.containsAll(marcadas);
67     }
68
69     private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
70         // TODO add your handling code here:
71         jButton1.setText("X");
72         jButton1.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 253, 203), 3, true));
73         jButton1.setEnabled(false);
74         marcadas.add(tabla.getPersonajes().get(0).getNombre());
75     }
76
77     private void jButton9ActionPerformed(java.awt.event.ActionEvent evt) {
78         // TODO add your handling code here:
79         jButton9.setText("X");
80         jButton9.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 253, 203), 3, true));
81         jButton9.setEnabled(false);
82         marcadas.add(tabla.getPersonajes().get(8).getNombre());
83     }
84
85     private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
86         // TODO add your handling code here:
87         jButton2.setText("X");
88         jButton2.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 253, 203), 3, true));
89         jButton2.setEnabled(false);
90         marcadas.add(tabla.getPersonajes().get(1).getNombre());
91     }
92
93     private void jButton3ActionPerformed(java.awt.event.ActionEvent evt) {
94         // TODO add your handling code here:
95         jButton3.setText("X");
96         jButton3.setBorder(new javax.swing.border.LineBorder(new java.awt.Color(0, 253, 203), 3, true));
97         jButton3.setEnabled(false);
98         marcadas.add(tabla.getPersonajes().get(2).getNombre());
99     }
100 }
```

```
571 private void loteriaActionPerformed(java.awt.event.ActionEvent evt) {
572     // TODO add your handling code here:
573     timer.stop();
574     if(comprobarLoteria(marcadas, sacadas))
575         System.out.println("Si hay loteria");
576     else
577         System.out.println("No hay loteria");
578 }
579
580
581 private void iniciarActionPerfomed(java.awt.event.ActionEvent evt) {
582     // TODO add your handling code here:
583     timer.start();
584 }
```



```
17 public class InterfazJugador extends javax.swing.JFrame {
18
19 /**
20 * Creates new form InterfazJugador
21 */
22 public InterfazJugador() {
23     initComponents();
24 }
25
26 Picture p = new Picture("C:\\\\Users\\\\harry\\\\Downloads\\\\loteria-parche.jpg");
27 Image img = p.getImage();
28 static Bonche bonche = new Bonche();
```



```
36 @SuppressWarnings("unchecked")
37 Generated Code
102
103 private void jButton1ActionPerformed(java.awt.event.ActionEvent evt) {
104     // TODO add your handling code here:
105     bonche.barajar();
106
107     TablaGrafica t = new TablaGrafica(bonche);
108     t.setVisible(true);
109 }
110
111 private void jButton2ActionPerformed(java.awt.event.ActionEvent evt) {
112     // TODO add your handling code here:
113     System.exit(0);
114 }
```

Comprobación

```
14 public class LoteriaMexicana {
15
16     /**
17      * @param args the command line arguments
18      */
19     public static void main(String[] args) {
20         // TODO code application logic here
21
22         InterfazJugador j = new InterfazJugador();
23         j.setVisible(true);
24
25     }
26
27 }
```





