Programación de Computadores: Java - Gráficas Interactivas

Juan F. Pérez

Departamento MACC
Matemáticas Aplicadas y Ciencias de la Computación
Universidad del Rosario

juanferna.perez@urosario.edu.co

Segundo Semestre de 2017

Contenidos

Ejemplo Inicial

Un ejemplo

```
package graficas;
import acm.program.GraphicsProgram;
import java.awt.event.MouseEvent;

public class estrellas extends GraphicsProgram{

 private static final double tamEstrella = 10;

 public void init(){
  addMouseListeners();
 }

 public void mouseClicked(MouseEvent e){
  GStar estrella = new GStar(tamEstrella);
  estrella.setFilled(true);
  add(estrella, e.getX(), e.getY());
 }
}
```

Un ejemplo (clase GStar)

```
package graficas;
import acm.graphics.*;

public class GStar extends GPolygon{
 public GStar(double width){
  double dx = width/2;
  double dy = dx*GMath.tanDegrees(18);
  double edge = width/2 - dy*GMath.tanDegrees(36);
  addVertex(-dx, dy);

  int angle = 0;
  for ( int i = 0; i < 5; i++){
   addPolarEdge(edge, angle);
   addPolarEdge(edge, angle+72);
   angle -= 72;
  }
 }
}
```

Eventos y Listeners

Ejecución Sincronizada vs Asíncrona

Consola (ConsoleProgram):

- Interacción con usuario en momentos específicos
- Funcionamiento sincronizado entre aplicación y usuario

Interfaz gráfica (GraphicsProgram):

Ejecución Sincronizada vs Asíncrona

Consola (ConsoleProgram):

- Interacción con usuario en momentos específicos
- Funcionamiento sincronizado entre aplicación y usuario

Interfaz gráfica (GraphicsProgram):

- Interacción dictada por el usuario
- Funcionamiento asíncrono entre aplicación y usuario

Eventos

- `EventObject` (paquete `java.awt`):
  - Objetos que capturan eventos de diversas clases

Eventos

- `EventObject` (paquete `java.awt`):
  - Objetos que capturan eventos de diversas clases
- `MouseEvent`
- `KeyEvent`
- `ActionEvent`

Eventos

- `EventObject` (paquete `java.awt`):
  - Objetos que capturan eventos de diversas clases
- `MouseEvent`
- `KeyEvent`
- `ActionEvent`
  - Eventos **no ejecutan** accionan

Eventos

- `EventObject` (paquete `java.awt`):
  - Objetos que capturan eventos de diversas clases
- `MouseEvent`
- `KeyEvent`
- `ActionEvent`
  - Eventos **no ejecutan** accionan
    - Listeners

Listeners

1. Interfaces de `Java`
2. Conjunto de métodos que se implementar para reaccionar a un evento o tipo de eventos

Listeners

1. Interfaces de `Java`
2. Conjunto de métodos que se implementar para reaccionar a un evento o tipo de eventos
3. `MouseListener`: escucha eventos del ratón (clicks, presión en un botón)

Listeners

1. Interfaces de `Java`
2. Conjunto de métodos que se implementar para reaccionar a un evento o tipo de eventos
3. `MouseListener`: escucha eventos del ratón (clicks, presión en un botón)
4. `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseListener.html`

Listeners

1. Interfaces de `Java`
2. Conjunto de métodos que se implementar para reaccionar a un evento o tipo de eventos
3. `MouseListener`: escucha eventos del ratón (clicks, presión en un botón)
4. `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseListener.html`
5. `public class miPrograma implements MouseListener`

Listeners

1. Interfaces de Java
2. Conjunto de métodos que se implementar para reaccionar a un evento o tipo de eventos
3. `MouseListener`: escucha eventos del ratón (clicks, presión en un botón)
4. `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseListener.html`
5. `public class miPrograma implements MouseListener`
6. `public class miPrograma extends GraphicsProgram implements MouseListener`
7. `Program` ya implementa varios listeners
8. `http://cs.stanford.edu/people/eroberts/jtf/javadoc/student/index.html`

Examinemos la clase DibujarEstrellas

```
package graficas;
import acm.program.GraphicsProgram;
import java.awt.event.MouseEvent;

public class estrellas extends GraphicsProgram{

 private static final double tamEstrella = 10;

 public void init(){
  addMouseListeners();
 }

 public void mouseClicked(MouseEvent e){
  GStar estrella = new GStar(tamEstrella);
  estrella.setFilled(true);
  add(estrella, e.getX(), e.getY());
 }
}
```

Otros Listeners

- `MouseListener`

Otros Listeners

- `MouseListener`
- `MouseMotionListener`

Otros Listeners

- `MouseListener`
- `MouseMotionListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseMotionListener.html`

Otros Listeners

- `MouseListener`
- `MouseMotionListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseMotionListener.html`
- `KeyListener`

Otros Listeners

- `MouseListener`
- `MouseMotionListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseMotionListener.html`
- `KeyListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/KeyListener.html`

Otros Listeners

- `MouseListener`
- `MouseMotionListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/MouseMotionListener.html`
- `KeyListener`
- `https://docs.oracle.com/javase/8/docs/api/java/awt/event/KeyListener.html`
- Más listeners `https://docs.oracle.com/javase/8/docs/api/java/util/EventListener.html`

Examinemos la clase moverObjetos

```
private GObject obj;
private GPoint ultimo;

public void init(){
 GRect rect = new GRect(100, 100, 150, 100);
 rect.setFilled(true);
 rect.setColor(Color.CYAN);
 add(rect);

 GOval oval = new GOval(300, 115, 100, 70);
 oval.setFilled(true);
 oval.setColor(Color.ORANGE);
 add(oval);

 addMouseListeners();
}
```

Examinemos la clase moverObjetos

```
public void mousePressed(MouseEvent e){
 ultimo = new GPoint(e.getPoint());
 obj = getElementAt(ultimo);
}

public void mouseDragged(MouseEvent e){
 if (obj !=null){
  obj.move(e.getX() - ultimo.getX(), e.getY() - ultimo.getY
    ());
  ultimo = new GPoint(e.getPoint());
 }
}

public void mouseClicked(MouseEvent e){
 if(obj !=null){
  obj.sendToFront();
 }
}
```

Otro Ejemplo: DibujarLineas

```
public class DibujarLineas extends GraphicsProgram{

 private GLine linea;

 public void init(){
  addMouseListeners();
 }

 public void mousePressed(MouseEvent e){
  linea = new GLine(e.getX(), e.getY(), e.getX(), e.getY());
  add(linea);
 }

 public void mouseDragged(MouseEvent e){
  linea.setEndPoint(e.getX(), e.getY());
 }
}
```

Otro Ejemplo: MoverObjetosV2

```
 public void init(){
  ...
  addKeyListeners();
 }

public void keyPressed(KeyEvent  e){
 if(obj != null){
  switch(e.getKeyCode()){
   case KeyEvent.VK_UP:    obj.move(0,  -10); break;
   case KeyEvent.VK_DOWN:  obj.move(0,  10);  break;
   case KeyEvent.VK_LEFT:  obj.move(-10, 0);  break;
   case KeyEvent.VK_RIGHT: obj.move(10,  0);  break;
  }
 }
}
```

El paquete de interactores javax.swing

Agregando Botones

Agreguemos un Botón a la clase Extrellas (EstrellasV2)

```
public class EstrellasV2 extends GraphicsProgram{
 ...

 public void init(){
  add(new JButton("Limpiar"), SOUTH);
  addMouseListeners();
  addActionListeners();
 }

 public void actionPerformed(ActionEvent e){
  if(e.getActionCommand().equals("Limpiar")){
   removeAll();
 }
}
```

La clase JButton

- Crear botones

La clase JButton

- Crear botones
- Interacción se captura con `ActionListeners`

La clase JButton

- Crear botones
- Interacción se captura con `ActionListeners`
- Constructor con String

La clase JButton

- Crear botones
- Interacción se captura con `ActionListeners`
- Constructor con String
- Método de acción: `actionPerformed`

La clase JButton

- Crear botones
- Interacción se captura con `ActionListeners`
- Constructor con String
- Método de acción: `actionPerformed`
- Tipo de evento: `ActionEvent`

La clase JButton

- Crear botones
- Interacción se captura con `ActionListeners`
- Constructor con String
- Método de acción: `actionPerformed`
- Tipo de evento: `ActionEvent`
- ¿Cómo saber el elemento que genera la acción? `getActionCommand()`

Otro tipo de botón: JCheckBox

- Extiende `JToggleButton`
- Cuando se hace click, se mantiene presionado hasta que se vuelva a dar click

Otro tipo de botón: JCheckBox

- Extiende `JToggleButton`
- Cuando se hace click, se mantiene presionado hasta que se vuelva a dar click
- `JCheckBox`: al dar click se chequea (selecciona) esta opción hasta que se vuelve a dar click sobre el elemento

Otro tipo de botón: JCheckBox

- Extiende `JToggleButton`
- Cuando se hace click, se mantiene presionado hasta que se vuelva a dar click
- `JCheckBox`: al dar click se chequea (selecciona) esta opción hasta que se vuelve a dar click sobre el elemento
- Ejemplo: `EstrellasV3`

Ejemplo EstrellasV3

```
public class EstrellasV3 extends GraphicsProgram{

 ...

 private JCheckBox llenarEstrellas;

 public void init(){
  add(new JButton("Limpiar"), SOUTH);

  llenarEstrellas = new JCheckBox("Llenar␣Estrellas");
  llenarEstrellas.setSelected(true);
  add(llenarEstrellas, SOUTH);
  addMouseListeners();
  addActionListeners();
 }

}
```

Otros elementos

- `JLabel`

Otros elementos

- `JLabel`
- `JSlider`

Otros elementos

- `JLabel`
- `JSlider`
- `JComboBox`

Otros elementos

- JLabel
- JSlider
- JComboBox
- ButtonGroup

Otros elementos

- JLabel
- JSlider
- JComboBox
- ButtonGroup
- JRadioButton

Otros elementos

- JLabel
- JSlider
- JComboBox
- ButtonGroup
- JRadioButton
- JTextField

Otros elementos

- `JLabel`
- `JSlider`
- `JComboBox`
- `ButtonGroup`
- `JRadioButton`
- `JTextField`
- Muchos mas

Otros elementos

- `JLabel`
- `JSlider`
- `JComboBox`
- `ButtonGroup`
- `JRadioButton`
- `JTextField`
- Muchos mas
- **Taller 9**