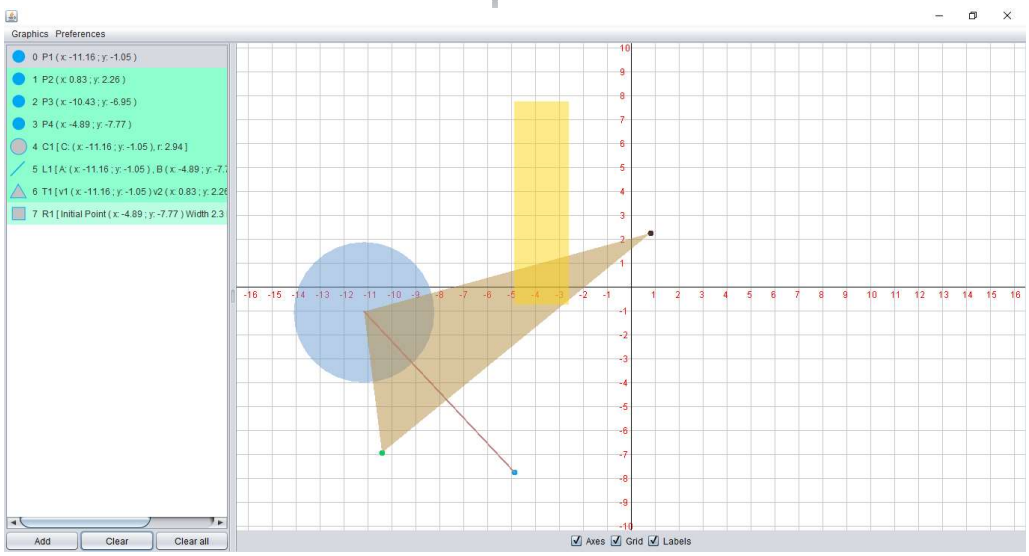
Informe técnico N° 1 pare ser entregado el 21 de febrero

Agustín Cartaya

Parte 1

Graphics



Descripción

Graphics es una aplicación desarrollada en Java cuya funcionalidad es graficar elementos 2D en un plano compuesto por dos ejes de coordenada (X,Y).

La aplicación permite:

- Graficar elementos como:
    - Puntos. Teniendo sus coordenadas X,Y
    - Segmentos. Teniendo las coordenadas de sus extremos
    - Triángulos. Teniendo las coordenadas de sus 3 vértices
    - círculos. Teniendo la coordenada de su centro y el valor de su radio
    - Rectángulos. Teniendo la coordenada de su esquina superior izquierda y los valores de su altura y su anchura

- Obtener los datos de los elementos graficados: la aplicación tiene la opción de mostrar información en la cual se despliega un panel con la información de todos sus elementos, incluyendo sus nombres, su tipo, si ubicación, y su área y perímetro (Si se es posible)

- Gestionar colores: la aplicación cuenta con un panel para gestionar la mayor parte de los colores que se pueden visualizar en la aplicación, incluyendo los colores del espacio grafico y la lista de elementos.

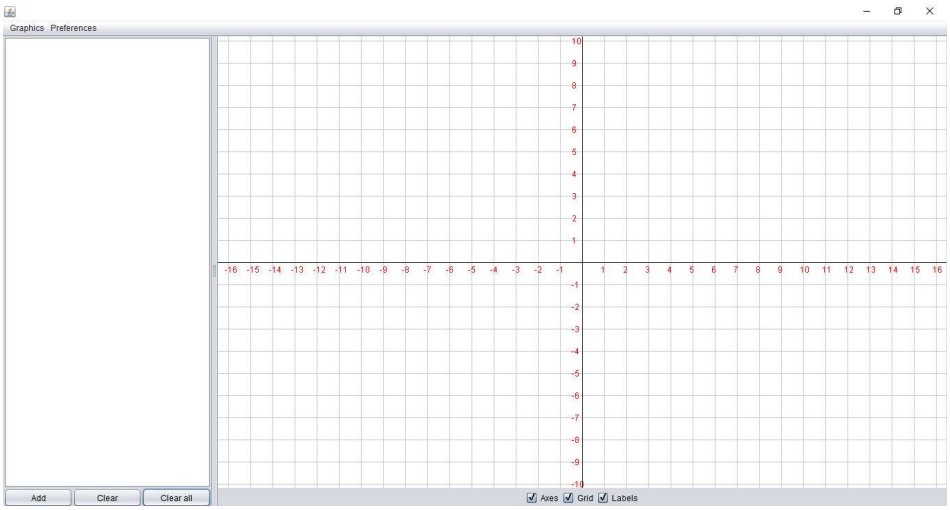
- Gestionar fuentes: la aplicación cuenta con un panel para gestionar fuentes que permite cambiar el tipo de letra, el tamaño y el estilo a algunos elementos de del espacio grafico o de la lista de elementos

- Edición del Espacio gráfico: la aplicación cuenta con distintos checkboxes que permiten mostrar o ocultar elementos guas del espacio gráfico

Ventanas de la aplicación y utilidades

La aplicación cuenta actualmente con una ventana principal (JFrame) que contiene el espacio grafico y la lista de elementos y luego 4 ventanas secundarias (JDialog) destinadas a mostrar, agregar o editar, a continuación, se detallaran más precisamente cada una de ellas.

Ventana Inicial (JFrame)



Esta es la ventana inicial del programa que cuenta con una barra de herramientas y un panel divisor (JSplitPane) el cual contendrá el espacio gráfico y la lista de elementos en cada uno de sus lados.

Árbol

JFrame Grapics

- ✓ (BorderLayout)
- ✓ JMenuBar menuBar
    - JMenu jMenu1 “Graphics”
        - JMenuItem addDefaultElements “Add default Elements”
        - JMenuItem showElementDescriptions “Elements Description”
    - JMenu jMenu2 “Preferences”
        - JMenuItem showColorsForm “Colors”
        - JMenuItem showFontform “Fonts”
- ✓ JSplitPane jSplitPane1, CENTER
    - JPanel jPanel1
        - (BorderLayout)
            - JPanel jPanel3, CENTER
                - (GridLayout [1,1])
                - JScrollPane jScrollPane1
                    - JList elementsList
            - JPanel jPanel4, SOUTH
                - (GridLayout [2,1])
                - JPanel jPanel2
                    - (FlowLayout)
                    - JButton jbAdd “Add”
                    - JButton clear “Clear”
                    - JButton clearAll “Clear All”
                - JPanel jPanel4
                    - (FlowLayout)
                    - JRadioButton lock “Lock”
                    - JRadioButton unlock “Lnlock”
    - JPanel jpanelCenter
        - (BorderLayout)
        - JPanel jPanel5, SOUTH
            - (FlowLayout)
            - JCheckBox axesVisibility “Axes”
            - JCheckBox gridVisibility “Grids”
            - JCheckBox labelsVisibility “Labels”
        - GraphicSpace, CENTER

Elementos

Barra de herramientas (JMenuBar).

Menú Graphics (JMenu)

Contiene las opciones para la edición o información del espacio grafico

Add default Elements (JMenuItem)

Opción que permite añadir los elementos predefinidos al espacio gráfico, estos son:

- 4 puntos con coordenadas aleatorias y 3 de ellos con colores aleatorios, además el primer punto está establecido como no visible,
- 2 círculos C1 y C2 con sus centros en P1 y P2 respectivamente y de radios aleatorios
- 2 líneas L1 y L2 cuyos extremos son P1-P4 y P2-P3 respectivamente
- 1 Triangulo cuyos vértices son P1,P2,P3
- 1 rectángulo con vértice superior izquierdo en P4 y anchura y largura aleatoria

Elements Description (JMenuItem)

Opción que permite mostrar el panel información de los elementos del grafico

Menu Preferences (JMenu)

Contiene las opciones para la personalización de la ventana

Colors (JMenuItem)

Opción que permite mostrar el panel para la edición de colores

Fonts (JMenuItem)

Opción que permite mostrar el panel para la edición de las fuentes

Lista (JList)

Lista que contiene todos los elementos añadidos al espacio gráfico, permite la selección de los elementos y cuando el ratón esta sobre uno de ellos este cambia de color en el espacio grafico

Add (JButton)

Botón que permite añadir un elemento nuevo al espacio gráfico. Despliega el formulario para añadir el nuevo elemento

Clear (JButton)

Botón que permite eliminar los elementos seleccionas en la lista

Clear All (JButton)

Botón que permite eliminar todos los elementos de la lista

Axes (JCheckBox)

Check que permite mostrar o no los ejes principales en el espacio grafico

Grids (JCheckBox)

Check que permite mostrar o las líneas guía en el espacio grafico

Labels (JCheckBox)

Check que permite mostrar la numeración de divisiones en el espacio grafico

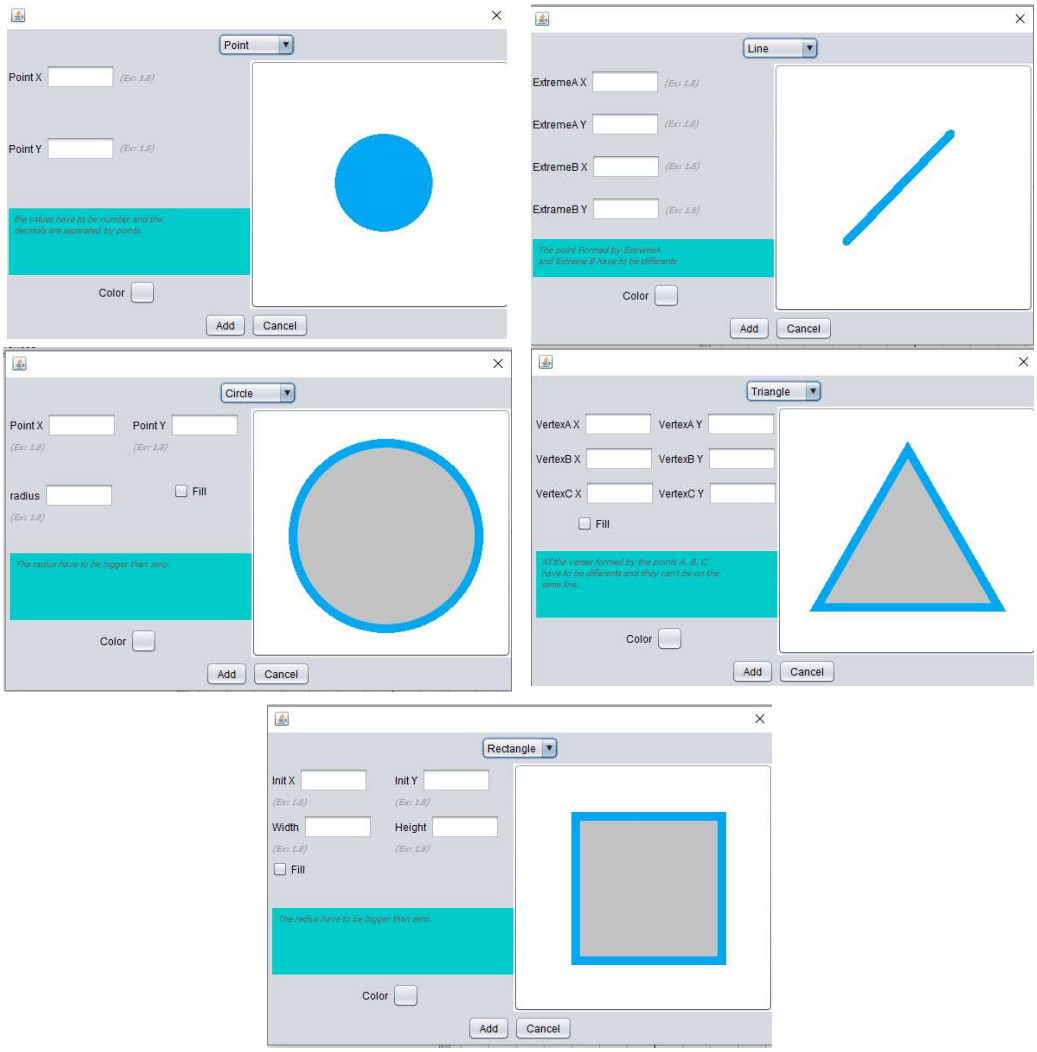
Lock (JRadioButton)

Si este Radiobutton esta seleccionado no se puede agregar o eliminar ningún elemento

Unlock (JRadioButton)

Si este Radiobutton esta seleccionado se pueden agregar y eliminar elementos

Formulario para Anadir elementos (JDialog)



Este formulario permite agregar los distintos elementos disponible al espacio grafico, cuenta con espacio dinamico que cambia dependiendo el elemento que deseemos agregar y con dos botones Cancel y Add que cancelan o agregan el elemento respectivamente

Árbol

- ✓ JDialog FormAddElement (JDialog)
- ✓ (BorderLayout)
- ✓ JPanel jPanel1, NORTH
    - (FlowLayout)
    - JComboBox addElementList
- ✓ JPanel jPanel2, CENTER
    - (BorderLayout)
    - JPanel jPanel7, CENTER
        - (BorderLayout)
        - JPanel jPanel5, SOUTH
            - (FlowLayout)
            - JLabel jLabel1 “Color”
            - JButton colorSelected “ ”
        - JPanel panelForm, CENTER
            - (GridLayout [1,1])
            - (Contenido dinámico)
    - JPanel jPanel4, EAST
        - (GridLayout [1,1])
        - JButton preview “”
- ✓ JPanel jPanel6, SOUTH
    - (FlowLayout)
    - JButton addElement “Add”
    - JButton cancel “Cancel”

Elementos

Lista de elementos para agregar (JComboBox)

Esta lista contiene los nombres de los elementos que se pueden agregar al espacio gráfico, al desplegarla y seleccionar uno de ellos esta modifica el formulario dependiendo los datos que necesite el elemento y muestra una imagen del elemento a agregar en el botón preview

Preview (JButton)

En este botón podemos ver la previsualización del elemento que se quiere agregar

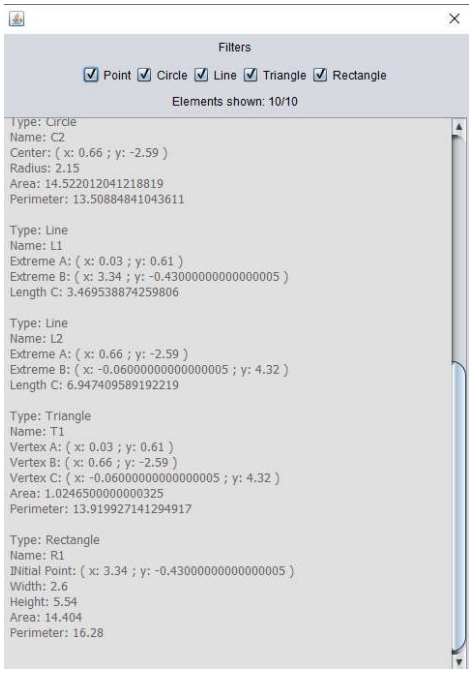
Add (JButton)

Al pulsar este botón se agrega el elemento al espacio gráfico y se cerrará el formulario, en caso de error una ventana emergente se mostrará indicando el error

Cancel (JButton)

Al pulsar este botón se cierra el formulario sin agregar ningún elemento

Lista de información (JDialog)



Esta lista contiene la información de los elementos en el espacio gráfico, tiene también una serie de filtros para visualizar solo los elementos deseado

Árbol

- ✓ JDialog Information
  - (BorderLayout)
  - JPanel jPanel1, NORTH
    - (GridLayout [3,1])
    - JPanel jPanel2
      - (FlowLayout)
      - JLabel jLabel1 “Filters”
    - JPanel jpanelFilters
      - (FlowLayout)
      - (Contenido dinámico)
    - JPanel jPanel3
      - (FlowLayout)
      - JLabel ElementsShown “(Texto dinámico)”
  - JScrollPane jScrollPane1, CENTER
    - JTextArea elementsDescription “(Texto dinámico)”

Elementos

Área de información (JTextArea)

Aquí es donde se mostrará toda la información de los elementos

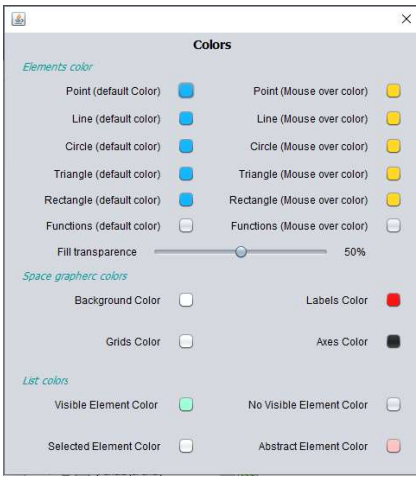
Filtros (JChechBox)

Filtros que aparecerán dinámicamente dependiendo los tipos de elementos que tengamos en nuestro espacio gráfico, estos nos ayudarán a mostrar o no dichos elementos en la lista de información

Contador de elementos mostrados (Jlabel)

Indica la cantidad de elementos mostrados y la cantidad total de elementos

Formulario edición de colores (JDialog)



Este formulario permite editar los colores de la ventana, incluidos los de la lista de elementos y los del espacio gráfico.

Árbol

- ✓ JDialog FormColor
  - (BorderLayout)
  - JPanel jPanel2, NORTH
    - (FlowLayout)
    - JLabel jLabel1 “Colors”
  - JPanel jPanel1, CENTER
    - (GridLayout [1,4])
    - JPanel jPanel3
      - (BorderLayout)
      - JPanel jPanel6, NORTH
        - (FlowLayout)
        - JLabel jLabel1 “Elements colors”
      - JPanel jPanel7, CENTER
        - (GridLayout [2,3])
          - JPanel jPanel4
            - (FlowLayout)
            - JLabel jLabel1 “Point (default Color)”
            - JButton pointDefaultColor

.

.

.

.

Es largo y repetitivo, lo hare luego…

Elementos

Botones de colores (JButton)

Cada botón se encarga de cambiar el color de lo que indica, haciendo el llamado a un JColorChooser para seleccionar el color

Formulario edición de Fuentes (JDialog)

Este formulario permite editar algunas de los tipos de texto de la ventana, incluidos los de la lista de elementos y los del espacio gráfico.

Árbol

JDialog FormFonts

- ✓ (BorderLayout)
- ✓ JPanel jPanel1, NORTH
    - (FlowLayout)
    - JLabel jLabel1, “Font”
- ✓ JPanel jPanel2, CENTER
    - (GridLayout [2,1])
    - JPanel jPanel3
        - (FlowLayout)
        - JLabel jLabel2, “List Font”
        - jButton listFont “(Texto dinamico)”
    - JPanel jPanel4
        - (FlowLayout)
        - JLabel jLabel3, “Axes Font”
        - jButton axesFont “(Texto dinámico)”

Elements

Botones (JButton)

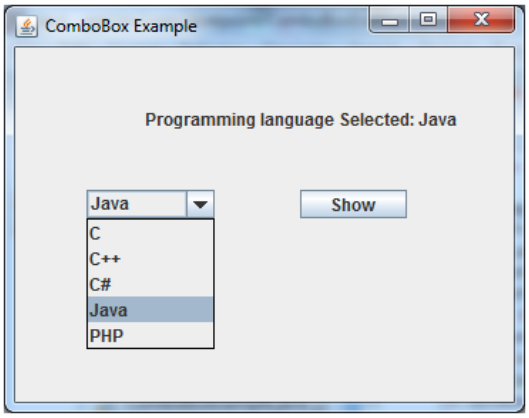
Estos botones muestran un panel que permite elegir la fuente, el estilo y el tamaño de esta para luego agregarlo al componente indicado, el texto del botón cambia indicando la fuente seleccionada

Parte 2:

utilización de JComboBox y JList

JComboBox

Funcionamiento de un JComboBox
Un JComboBox es una lista desplegable de elementos que te permite seleccionar uno de ello, ejemplo:



La creación de un JComboBox se hace con *ayuda de la clase JComboBox perteneciente al* paquete javax.swing. esta clase tiene sobrecarga de constructores pero en esta definición nos enfocaremos solo en el constructor por defecto. La creación de un JComboBox se hace con la siguiente línea de código:

```
JComboBox miComboBox = new JComboBox();
```

Métodos más importantes de un JComboBox:

addItem(E ele): permite agregar un elemento al combo box pide por parámetro un objeto del tipo que guarda el comboBox en nuestro caso será de tipo String.

```
miComboBox.addItem( “Elemento 1” );
```

removeItem(E ele): permite eliminar un elemento del comboBox pide por parámetro un objeto del tipo que guarda el comboBox.

```
miComboBox.removeItem ( “Elemento 1” );
```

removeItemAt(int index): permite eliminar un elemento del combo box pide por parámetro la posición del objeto .

```
miComboBox.removeItem ( 0 );
```

removeAllItems(): permite eliminar todos los elementos del comboBox

```
miComboBox.removeAllItems ();
```

getSelectedItem(): permite obtener el elemento seleccionado, este método devuelve un abjeto del tipo que guarda el comboBox en nuestro caso será un String, pero hay que hacerle un casting

```
String itemSelected = (String)miComboBox.getSelectedItem ();
```

getSelectedIndex(): permite obtener la posición del elemento seleccionado,

```
int indexSelected = miComboBox.getSelectedIndex ();
```

setSelectedItem (E obj): permite seleccionar un elemento.

```
miComboBox. setSelectedItem (“Elemento 1”);
```

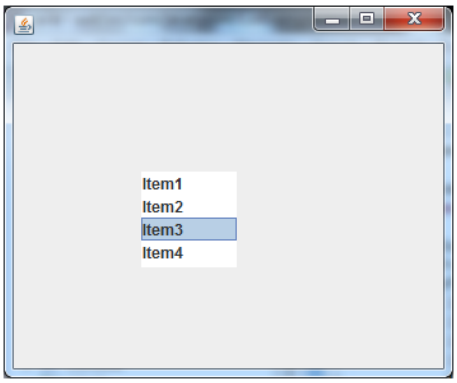
setSelectedIndex(int index): permite seleccionar un elemento teniendo pasandole su posicion por parámetro.

```
miComboBox.setSelectedIndex (0);
```

getItemCount(): devuelve la cantidad de elementos contenida en el comboBox

```
int indexSelected = miComboBox.getItemCount (0);
```

JList



Una JList es una lista de elementos en forma de menú, los cuales se pueden seleccionar (permite selección múltiple) e interactuar con ellos.

Para la crecion de un JList se necesita un ListModel que es el encargado de guardar los elementos de la JList

La creación de un JList y su respectivo ListModel se hace con las siguientes líneas de código (utilizaremos el modelo por defecto):

```
DefaultListModel dlm = new DefaultListModel();
JList miLista = new JList ( dlm );
```

Métodos Mas importantes de un JComboBox:

setModel(ListModel mod): establece el modelo de lista

```
miLista.setModel( dlm );
```

addElement(E ele): agrega un elemento al listModel

```
dlm.addElement ( "Element 0" );
```

remove(int index): elimina el elemento del listModel en la posicion "index"

```
dlm.remove ( 0 );
```

removeAll(int index): elimina todos los elementos del

```
dlm.removeAll ( 0 );
```

getSelectedIndex(): recupera la posición del elemento seleccionado de la lista

```
int pos = miLista.getSelectedIndex ();
```

getSelectedIndeces(): recupera las posiciones de los elementos seleccionados de la lista

```
int[] pos = miLista.getSelectedIndices ();
```