

```
/*
 * PropertiesViewPanel.java
 *
 * Created on January 22, 2007, 2:10 PM
 */

package Vista2;

import Controlador2.DefaultController;
import Utilidades.JFontChooserDialog;
import java.awt.Dialog;
import java.awt.Font;
import java.beans.PropertyChangeEvent;
import javax.swing.DefaultBoundedRangeModel;
import javax.swing.SpinnerNumberModel;
import javax.swing.event.DocumentEvent;
import javax.swing.event.DocumentListener;
import javax.swing.text.BadLocationException;
import javax.swing.text.Document;

/**
 * This is a custom view panel that display properties for both the document
 * and text element. Both the document and the text element respond to changes
 * in the model state.
 *
 * @author Robert Eckstein
 */
public class PropertiesViewPanel extends AbstractViewPanel
{
    // The controller used by this view

    private DefaultController controller;

    // A local copy of the current Font which can be passed to the
    // constructor of JFontChooserDialog.

    private Font currentFont;

    /**
     * Creates new form PropertiesViewPanel
     * @param controller An object implenting the controller interface that
     * this view can use to process GUI actions
     */
    public PropertiesViewPanel(DefaultController controller) {
        this.controller = controller;

        initComponents();
        localInitialization();
    }

    // <editor-fold defaultstate="collapsed" desc=" Local Initialization ">

    /**
     * Used to provide local initialization of Swing components outside of the
     * NetBeans auto code generator.
     */
    public void localInitialization() {
        opacitySpinner.setModel(new SpinnerNumberModel(100, 0, 100, 1));
        opacitySlider.setModel(new DefaultBoundedRangeModel(100, 0, 0, 100));

        rotationSpinner.setModel(new SpinnerNumberModel(0, -180, 180, 1));
        rotationSlider.setModel(new DefaultBoundedRangeModel(0, 0, -180, 180));

        text.getDocument().addDocumentListener(new DocumentListener() {

            public void insertUpdate(DocumentEvent e) {
```

```
        textDocumentChanged(e);
    }

    public void removeUpdate(DocumentEvent e) {
        textDocumentChanged(e);
    }

    public void changedUpdate(DocumentEvent e) {
        textDocumentChanged(e);
    }

});

}

// </editor-fold>

// <editor-fold defaultstate="collapsed" desc=" Model Event Handling Code ">

/**
 * Called by the controller when it needs to pass along a property change
 * from a model. Note that the method checks each GUI parameter to determine
 * if the current value is already equal to the incoming value. If it is,
 * the method will not reset the value. This is done to prevent looping from
 * occurring when a model property is reset.
 * @param evt The property change event
 */
public void modelPropertyChange(final PropertyChangeEvent evt) {

    if (evt.getPropertyName().equals(DefaultController.ELEMENT_X_PROPERTY)) {

        String newStringValue = evt.getNewValue().toString();

        if (!xPositionTextField.getText().equals(newStringValue))
            xPositionTextField.setText(newStringValue);

    }

    else if (evt.getPropertyName().equals(DefaultController.ELEMENT_Y_PROPERTY)) {

        String newStringValue = evt.getNewValue().toString();

        if (!yPositionTextField.getText().equals(newStringValue))
            yPositionTextField.setText(newStringValue);

    }

    else if (evt.getPropertyName().equals(DefaultController.ELEMENT_OPACITY_PROPERTY)) {

        int newIntegerValue = (Integer)evt.getNewValue();

        if ((Integer)opacitySpinner.getValue() != newIntegerValue)
            opacitySpinner.setValue(newIntegerValue);

        if ((Integer)opacitySlider.getValue() != newIntegerValue)
            opacitySlider.setValue(newIntegerValue);

    }

    else if (evt.getPropertyName().equals(DefaultController.ELEMENT_ROTATION_PROPERTY)) {

        int newIntegerValue = (Integer)evt.getNewValue();

        if ((Integer)rotationSpinner.getValue() != newIntegerValue)
            rotationSpinner.setValue(newIntegerValue);

        if ((Integer)rotationSlider.getValue() != newIntegerValue)
            rotationSlider.setValue(newIntegerValue);

    }

    else if (evt.getPropertyName().equals(DefaultController.ELEMENT_TEXT_PROPERTY)) {
```

```

        String newStringValue = evt.getNewValue().toString();

        if (!text.getText().equals(newStringValue))
            text.setText(newStringValue);
    }

    else if (evt.getPropertyName().equals(DefaultController.ELEMENT_FONT_PROPERTY)) {
        Font f = (Font)evt.getNewValue();
        String fontString = f.getFontName() + " " + f.getSize();
        font.setText(fontString);
        currentFont = f;
    }

    else if (evt.getPropertyName().equals(DefaultController.DOCUMENT_NAME_PROPERTY)) {
        String newStringValue = evt.getNewValue().toString();

        if (!nameTextField.getText().equals(newStringValue))
            nameTextField.setText(newStringValue);
    }

    else if (evt.getPropertyName().equals(DefaultController.DOCUMENT_WIDTH_PROPERTY)) {
        String newStringValue = evt.getNewValue().toString();

        if (!widthTextField.getText().equals(newStringValue))
            widthTextField.setText(newStringValue);
    }

    else if (evt.getPropertyName().equals(DefaultController.DOCUMENT_HEIGHT_PROPERTY)) {
        String newStringValue = evt.getNewValue().toString();

        if (!heightTextField.getText().equals(newStringValue))
            heightTextField.setText(newStringValue);
    }
}

// </editor-fold>

/** This method is called from within the constructor to
 * initialize the form.
 * WARNING: Do NOT modify this code. The content of this method is
 * always regenerated by the Form Editor.
 */
// <editor-fold defaultstate="collapsed" desc=" Generated Code "> //GEN-BEGIN:initComponents
private void initComponents() {
    jSlider2 = new javax.swing.JSlider();
    jSeparator2 = new javax.swing.JSeparator();
    propertiesViewTabbedPane = new javax.swing.JTabbedPane();
    documentTabPanel = new javax.swing.JPanel();
    nameTextField = new javax.swing.JTextField();
    widthTextField = new javax.swing.JTextField();
    heightTextField = new javax.swing.JTextField();
    nameLabel = new javax.swing.JLabel();
    widthLabel = new javax.swing.JLabel();
    heightLabel = new javax.swing.JLabel();
    textElementTabPanel = new javax.swing.JPanel();
    xPositionTextField = new javax.swing.JTextField();
    yPositionTextField = new javax.swing.JTextField();
    rotationSpinner = new javax.swing.JSpinner();
    rotationSlider = new javax.swing.JSlider();
    opacitySpinner = new javax.swing.JSpinner();
    opacitySlider = new javax.swing.JSlider();
    xPositionLabel = new javax.swing.JLabel();

```

```

yPositionLabel = new javax.swing.JLabel();
rotationLabel = new javax.swing.JLabel();
opacityLabel = new javax.swing.JLabel();
changeFontButton = new javax.swing.JButton();
font = new javax.swing.JLabel();
fontLabel = new javax.swing.JLabel();
textLabel = new javax.swing.JLabel();
textScrollPane = new javax.swing.JScrollPane();
text = new javax.swing.JTextArea();

propertiesViewTabbedPane.setFont(new java.awt.Font("Dialog", 0, 11));
nameTextField.setFont(new java.awt.Font("Dialog", 0, 11));
nameTextField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        nameTextFieldActionPerformed(evt);
    }
});
nameTextField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        nameTextFieldFocusLost(evt);
    }
});

widthTextField.setFont(new java.awt.Font("Dialog", 0, 11));
widthTextField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        widthTextFieldActionPerformed(evt);
    }
});
widthTextField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        widthTextFieldFocusLost(evt);
    }
});

heightTextField.setFont(new java.awt.Font("Dialog", 0, 11));
heightTextField.addActionListener(new java.awt.event.ActionListener() {
    public void actionPerformed(java.awt.event.ActionEvent evt) {
        heightTextFieldActionPerformed(evt);
    }
});
heightTextField.addFocusListener(new java.awt.event.FocusAdapter() {
    public void focusLost(java.awt.event.FocusEvent evt) {
        heightTextFieldFocusLost(evt);
    }
});

nameLabel.setFont(new java.awt.Font("Dialog", 0, 11));
nameLabel.setText("Name:");

widthLabel.setFont(new java.awt.Font("Dialog", 0, 11));
widthLabel.setText("Width:");

heightLabel.setFont(new java.awt.Font("Dialog", 0, 11));
heightLabel.setText("Height:");

javax.swing.GroupLayout documentTabPanelLayout = new javax.swing.GroupLayout(documentTabPanel);
documentTabPanel.setLayout(documentTabPanelLayout);
documentTabPanelLayout.setHorizontalGroup(
    documentTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
            documentTabPanelLayout.createSequentialGroup()
                .addContainerGap()
                .addComponent(heightLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 77,
                    Short.MAX_VALUE)
                .addComponent(widthLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 77,
                    Short.MAX_VALUE)
                .addComponent(nameLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 77,
                    Short.MAX_VALUE)
                .addGap(10, 10, 10)
        )
);

```

```

        .addGroup(documentTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(heightTextField, javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(widthTextField, javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(nameTextField, javax.swing.GroupLayout.Alignment.TRAILING,
javax.swing.GroupLayout.DEFAULT_SIZE, 300, Short.MAX_VALUE))
        .addContainerGap()
    );
    documentTabPanelLayout.setVerticalGroup(
        documentTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(documentTabPanelLayout.createSequentialGroup()
            .addContainerGap()
            .addGroup(documentTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(nameTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(nameLabel))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(documentTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(widthTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(widthLabel))
            .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
            .addGroup(documentTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
                .addComponent(heightTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
                .addComponent(heightLabel))
            .addContainerGap(304, Short.MAX_VALUE))
        );
    propertiesViewTabbedPane.addTab("Document", documentTabPanel);

    xPositionTextField.setFont(new java.awt.Font("Dialog", 0, 11));
    xPositionTextField.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            xPositionTextFieldActionPerformed(evt);
        }
    });
    xPositionTextField.addFocusListener(new java.awt.event.FocusAdapter() {
        public void focusLost(java.awt.event.FocusEvent evt) {
            xPositionTextFieldFocusLost(evt);
        }
    });

    yPositionTextField.setFont(new java.awt.Font("Dialog", 0, 11));
    yPositionTextField.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            yPositionTextFieldActionPerformed(evt);
        }
    });
    yPositionTextField.addFocusListener(new java.awt.event.FocusAdapter() {
        public void focusLost(java.awt.event.FocusEvent evt) {
            yPositionTextFieldFocusLost(evt);
        }
    });

    rotationSpinner.setFont(new java.awt.Font("Dialog", 0, 11));
    rotationSpinner.addChangeListener(new javax.swing.event.ChangeListener() {
        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            rotationSpinnerStateChanged(evt);
        }
    });

    rotationSlider.addChangeListener(new javax.swing.event.ChangeListener() {
        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            rotationSliderStateChanged(evt);
        }
    });

    opacitySpinner.setFont(new java.awt.Font("Dialog", 0, 11));
    opacitySpinner.addChangeListener(new javax.swing.event.ChangeListener() {

```

```

        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            opacitySpinnerStateChanged(evt);
        }
    });

    opacitySlider.addChangeListener(new javax.swing.event.ChangeListener() {
        public void stateChanged(javax.swing.event.ChangeEvent evt) {
            opacitySliderStateChanged(evt);
        }
    });

    xPositionLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    xPositionLabel.setText("X Position:");

    yPositionLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    yPositionLabel.setText("Y Position:");

    rotationLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    rotationLabel.setText("Rotation:");

    opacityLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    opacityLabel.setText("Opacity:");

    changeFontButton.setFont(new java.awt.Font("Dialog", 0, 11));
    changeFontButton.setText("Change");
    changeFontButton.addActionListener(new java.awt.event.ActionListener() {
        public void actionPerformed(java.awt.event.ActionEvent evt) {
            changeFontButtonActionPerformed(evt);
        }
    });

    font.setFont(new java.awt.Font("Dialog", 0, 11));
    font.setText("Arial Bold 12");

    fontLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    fontLabel.setText("Font:");

    textLabel.setFont(new java.awt.Font("Dialog", 0, 11));
    textLabel.setText("Text:");

    text.setColumns(20);
    text.setFont(new java.awt.Font("DialogInput", 0, 13));
    text.setRows(5);
    textScrollPane.setViewportView(text);

    javax.swing.GroupLayout textElementTabPanelLayout = new javax.swing.GroupLayout
(textElementTabPanel);
    textElementTabPanel.setLayout(textElementTabPanelLayout);
    textElementTabPanelLayout.setHorizontalGroup(
        textElementTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addGroup(textElementTabPanelLayout.createSequentialGroup()
                .add(textElementTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                    .add(textElementTabPanelLayout.createSequentialGroup()
                        .addContainerGap()
                        .addGroup(textElementTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
                            .addComponent(textScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 381,
Short.MAX_VALUE)
                            .addGroup(textElementTabPanelLayout.createSequentialGroup()
                                .addGroup(textElementTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.TRAILING, false)
                                    .addComponent(textLabel, javax.swing.GroupLayout.Alignment.LEADING,
javax.swing.GroupLayout.PREFERRED_SIZE, 74, javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(opacityLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(rotationLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 74,
Short.MAX_VALUE)
                                    .addComponent(fontLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 74,
javax.swing.GroupLayout.PREFERRED_SIZE)
                                    .addComponent(xPositionLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 74,
Short.MAX_VALUE)
                                    .addComponent(yPositionLabel, javax.swing.GroupLayout.DEFAULT_SIZE, 74,
Short.MAX_VALUE))
                                .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)

```



```

        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
textElementTabPanelLayout.createSequentialGroup())
        .addComponent(font, javax.swing.GroupLayout.DEFAULT_SIZE, 226,
Short.MAX_VALUE)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(changeFontButton)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED))
        .addComponent(xPositionTextField,
javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE, 301, Short.MAX_VALUE)
        .addComponent(yPositionTextField,
javax.swing.GroupLayout.Alignment.TRAILING, javax.swing.GroupLayout.DEFAULT_SIZE, 301, Short.MAX_VALUE)
        .addGroup(javax.swing.GroupLayout.Alignment.TRAILING,
textElementTabPanelLayout.createSequentialGroup())
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(opacitySlider, javax.swing.GroupLayout.DEFAULT_SIZE,
245, Short.MAX_VALUE)
        .addComponent(rotationSlider,
javax.swing.GroupLayout.DEFAULT_SIZE, 245, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.TRAILING)
        .addComponent(opacitySpinner,
javax.swing.GroupLayout.DEFAULT_SIZE, 50, Short.MAX_VALUE)
        .addComponent(rotationSpinner,
javax.swing.GroupLayout.PREFERRED_SIZE, 50, javax.swing.GroupLayout.PREFERRED_SIZE))))))
        .addContainerGap()
    );

    textElementTabPanelLayout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {opacitySpinner, rotationSpinner});

    textElementTabPanelLayout.linkSize(javax.swing.SwingConstants.HORIZONTAL, new
java.awt.Component[] {fontLabel, opacityLabel, rotationLabel, textLabel, xPositionLabel,
yPositionLabel});

    textElementTabPanelLayout.setVerticalGroup(
textElementTabPanelLayout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
        .addGroup(textElementTabPanelLayout.createSequentialGroup())
        .addContainerGap()
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(xPositionTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(xPositionLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)
        .addComponent(yPositionTextField, javax.swing.GroupLayout.PREFERRED_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(yPositionLabel))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(rotationLabel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(rotationSpinner)
        .addComponent(rotationSlider, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.LEADING, false)
        .addComponent(opacityLabel, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE)
        .addComponent(opacitySpinner)
        .addComponent(opacitySlider, javax.swing.GroupLayout.DEFAULT_SIZE,
javax.swing.GroupLayout.DEFAULT_SIZE, Short.MAX_VALUE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addGroup(textElementTabPanelLayout.createParallelGroup
(javax.swing.GroupLayout.Alignment.BASELINE)

```

```

        .addComponent(changeFontButton)
        .addComponent(font, javax.swing.GroupLayout.PREFERRED_SIZE, 21,
javax.swing.GroupLayout.PREFERRED_SIZE)
        .addComponent(fontLabel, javax.swing.GroupLayout.PREFERRED_SIZE, 22,
javax.swing.GroupLayout.PREFERRED_SIZE))
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(textLabel)
        .addPreferredGap(javax.swing.LayoutStyle.ComponentPlacement.RELATED)
        .addComponent(textScrollPane, javax.swing.GroupLayout.DEFAULT_SIZE, 202,
Short.MAX_VALUE)
        .addContainerGap()
    );
    propertiesViewTabbedPane.addTab("Text Element", textElementTabPanel);

    javax.swing.GroupLayout layout = new javax.swing.GroupLayout(this);
    this.setLayout(layout);
    layout.setHorizontalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(propertiesViewTabbedPane, javax.swing.GroupLayout.DEFAULT_SIZE, 406,
Short.MAX_VALUE)
    );
    layout.setVerticalGroup(
        layout.createParallelGroup(javax.swing.GroupLayout.Alignment.LEADING)
            .addComponent(propertiesViewTabbedPane, javax.swing.GroupLayout.DEFAULT_SIZE, 416,
Short.MAX_VALUE)
    );
} // </editor-fold> // GEN-END: initComponents

// <editor-fold defaultstate="collapsed" desc=" GUI Event Handling Code ">

private void heightTextFieldFocusLost(java.awt.event.FocusEvent evt) { // GEN-
FIRST:event_heightTextFieldFocusLost

    try {
        controller.changeDocumentHeight(
            Integer.parseInt(heightTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} // GEN-LAST:event_heightTextFieldFocusLost

private void heightTextFieldActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_heightTextFieldActionPerformed

    try {
        controller.changeDocumentHeight(
            Integer.parseInt(heightTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} // GEN-LAST:event_heightTextFieldActionPerformed

private void widthTextFieldFocusLost(java.awt.event.FocusEvent evt) { // GEN-
FIRST:event_widthTextFieldFocusLost

    try {
        controller.changeDocumentWidth(
            Integer.parseInt(widthTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} // GEN-LAST:event_widthTextFieldFocusLost

private void widthTextFieldActionPerformed(java.awt.event.ActionEvent evt) { // GEN-
FIRST:event_widthTextFieldActionPerformed

    try {
        controller.changeDocumentWidth(
            Integer.parseInt(widthTextField.getText()));
    }

```



```
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_widthTextFieldActionPerformed

private void nameTextFieldFocusLost(java.awt.event.FocusEvent evt) { //GEN-FIRST:event_nameTextFieldFocusLost

    try {
        controller.changeDocumentName(nameTextField.getText());
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_nameTextFieldFocusLost

private void nameTextFieldActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_nameTextFieldActionPerformed

    try {
        controller.changeDocumentName(nameTextField.getText());
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_nameTextFieldActionPerformed

private void yPositionTextFieldFocusLost(java.awt.event.FocusEvent evt) { //GEN-FIRST:event_yPositionTextFieldFocusLost

    try {
        controller.changeElementYPosition(
            Integer.parseInt(yPositionTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_yPositionTextFieldFocusLost

private void yPositionTextFieldActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_yPositionTextFieldActionPerformed

    try {
        controller.changeElementYPosition(
            Integer.parseInt(yPositionTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_yPositionTextFieldActionPerformed

private void xPositionTextFieldFocusLost(java.awt.event.FocusEvent evt) { //GEN-FIRST:event_xPositionTextFieldFocusLost

    try {
        controller.changeElementXPosition(
            Integer.parseInt(xPositionTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

} //GEN-LAST:event_xPositionTextFieldFocusLost

private void xPositionTextFieldActionPerformed(java.awt.event.ActionEvent evt) { //GEN-FIRST:event_xPositionTextFieldActionPerformed

    try {
        controller.changeElementXPosition(
            Integer.parseInt(xPositionTextField.getText()));
    } catch (Exception e) {
        // Handle exception
    }

}
```

```

    }

    }//GEN-LAST:event_xPositionTextFieldActionPerformed

    private void changeFontButtonActionPerformed(java.awt.event.ActionEvent evt) {//GEN-FIRST:event_changeFontButtonActionPerformed

        JFontChooserDialog fontChooser = new JFontChooserDialog((Dialog)this.getTopLevelAncestor());
        fontChooser.setSelectedFont(currentFont);
        fontChooser.setVisible(true);

        Font returnedFont = fontChooser.getSelectedFont();
        if (returnedFont != null) {
            controller.changeElementFont(returnedFont);
        }

    }//GEN-LAST:event_changeFontButtonActionPerformed

    private void opacitySliderStateChanged(javax.swing.event.ChangeEvent evt) {//GEN-FIRST:event_opacitySliderStateChanged

        controller.changeElementOpacity((int)opacitySlider.getValue());

    }//GEN-LAST:event_opacitySliderStateChanged

    private void rotationSliderStateChanged(javax.swing.event.ChangeEvent evt) {//GEN-FIRST:event_rotationSliderStateChanged

        controller.changeElementRotation((int)rotationSlider.getValue());

    }//GEN-LAST:event_rotationSliderStateChanged

    private void opacitySpinnerStateChanged(javax.swing.event.ChangeEvent evt) {//GEN-FIRST:event_opacitySpinnerStateChanged

        controller.changeElementOpacity((Integer)opacitySpinner.getValue());

    }//GEN-LAST:event_opacitySpinnerStateChanged

    private void rotationSpinnerStateChanged(javax.swing.event.ChangeEvent evt) {//GEN-FIRST:event_rotationSpinnerStateChanged

        controller.changeElementRotation((Integer)rotationSpinner.getValue());

    }//GEN-LAST:event_rotationSpinnerStateChanged

    private void textDocumentChanged(DocumentEvent evt) {

        Document document = evt.getDocument();

        try {
            controller.changeElementText(document.getText(0, document.getLength()));
        } catch (BadLocationException ex) {
            // Handle exception
        }

    }

}

// </editor-fold>

// Variables declaration - do not modify//GEN-BEGIN:variables
private javax.swing.JButton changeFontButton;
private javax.swing.JPanel documentTabPanel;
private javax.swing.JLabel font;
private javax.swing.JLabel fontLabel;
private javax.swing.JLabel heightLabel;
private javax.swing.JTextField heightTextField;
private javax.swing.JSeparator jSeparator2;
private javax.swing.JSlider jSlider2;
private javax.swing.JLabel nameLabel;

```

```
private javax.swing.JTextField nameTextField;
private javax.swing.JLabel opacityLabel;
private javax.swing.JSlider opacitySlider;
private javax.swing.JSpinner opacitySpinner;
private javax.swing.JTabbedPane propertiesViewTabbedPane;
private javax.swing.JLabel rotationLabel;
private javax.swing.JSlider rotationSlider;
private javax.swing.JSpinner rotationSpinner;
private javax.swing.JTextArea text;
private javax.swing.JPanel textElementTabPanel;
private javax.swing.JLabel textLabel;
private javax.swing.JScrollPane textScrollPane;
private javax.swing.JLabel widthLabel;
private javax.swing.JTextField widthTextField;
private javax.swing.JLabel xPositionLabel;
private javax.swing.JTextField xPositionTextField;
private javax.swing.JLabel yPositionLabel;
private javax.swing.JTextField yPositionTextField;
// End of variables declaration//GEN-END:variables
```

```
}
```