

## Using the Game GUI Interface

Add the GameGUI class to your project.

Add these instance declarations:

```
private GameGUI gui;  
private GameGUI scanner;
```

Add these initialization statements:

```
gui = new GameGUI();  
scanner = gui; // new Scanner(System.in);
```

Modify your display and displayLine methods:

```
public void display(String phrase) {  
    gui.addCommand(phrase);  
    //System.out.print(phrase);  
}  
  
public void displayLine(String phrase) {  
    gui.addCommand(phrase+"\n");  
    //System.out.println(phrase);  
}
```

To set the icon use:

```
ImageIcon icon;  
icon = new ImageIcon("P:\\CS 241\\...\\game\\sun.gif");  
  
gui.setIcon(icon);
```

To set the status use:

```
String status;  
status = "Game Status" + "\n\nUser: Biggest Loser\n" + "Health: 65";  
gui.setStatus(status);
```

# GameGUI.java

```
package game;

import java.awt.BorderLayout;

import javax.swing.BorderFactory;
import javax.swing.Icon;
import javax.swing.ImageIcon;
import javax.swing.JPanel;
import javax.swing.JFrame;
import java.awt.GridLayout;
import java.awt.GridBagLayout;
import java.awt.Dimension;
import javax.swing.JTextArea;
import javax.swing.JScrollPane;
import java.awt.GridBagConstraints;
import java.awt.image.BufferedImage;

import javax.swing.JLabel;
import java.awt.Font;
import java.awt.Color;
import java.awt.SystemColor;
import javax.swing.SwingConstants;
import javax.swing.border.Border;
import javax.swing.border.TitledBorder;

public class GameGUI extends JFrame {

    private static final long serialVersionUID = 1L;

    private JPanel jContentPane = null;

    private JPanel mainPanel = null;

    private JTextArea commandTextArea = null;

    private JPanel informationPanel = null;

    private JScrollPane commandScrollPane = null;

    private StringBuilder command = new StringBuilder(); // @jve:decl-index=0:
    private boolean waitingForResponse;

    private JLabel iconLabel = null;
```

```

private JTextArea statusTextArea = null;

/**
 * This is the default constructor
 */
public GameGUI() {
    super();
    initialize();
}

/**
 * This method initializes this
 *
 * @return void
 */
private void initialize() {
    this.setSize(589, 468);
    this.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    this.setContentPane(getJContentPane());
    this.setTitle("JFrame");
    this.setVisible(true);

    waitingForResponse = true;
}

/**
 * This method initializes jContentPane
 *
 * @return javax.swing.JPanel
 */
private JPanel getJContentPane() {
    if (jContentPane == null) {
        jContentPane = new JPanel();
        jContentPane.setLayout(new BorderLayout());
        jContentPane.add(getMainPanel(), BorderLayout.CENTER);
    }
    return jContentPane;
}

/**
 * This method initializes mainPanel
 *
 * @return javax.swing.JPanel
 */
private JPanel getMainPanel() {

```

```

        if (mainPanel == null) {
            GridLayout gridLayout = new GridLayout();
            gridLayout.setRows(1);
            gridLayout.setColumns(2);
            mainPanel = new JPanel();
            mainPanel.setLayout(gridLayout);
            mainPanel.add(getCommandScrollPane(), null);
            mainPanel.add(getInformationPanel(), null);
        }
        return mainPanel;
    }

/**
 * This method initializes commandTextArea
 *
 * @return javax.swing.JTextArea
 */
private JTextArea getCommandTextArea() {
    if (commandTextArea == null) {
        commandTextArea = new JTextArea();
        commandTextArea.setLineWrap(true);
        commandTextArea.setFont(new Font("Comic Sans MS", Font.BOLD,
14));

        commandTextArea.setWrapStyleWord(true);
        commandTextArea.addKeyListener(new java.awt.event.KeyAdapter() {
            public void keyTyped(java.awt.event.KeyEvent e) {
                command.append(e.getKeyChar());
                if(e.getKeyChar()=='\n') {
                    waitingForResponse = false;
                }
            }
        });
    }
    return commandTextArea;
}

/**
 * This method initializes informationPanel
 *
 * @return javax.swing.JPanel
 */
private JPanel getInformationPanel() {
    if (informationPanel == null) {
        iconLabel = new JLabel();
        iconLabel.setHorizontalAlignment(SwingConstants.CENTER);
        iconLabel.setHorizontalTextPosition(SwingConstants.CENTER);

```

```

        GridLayout gridLayout1 = new GridLayout();
        gridLayout1.setRows(2);
        gridLayout1.setHgap(3);
        gridLayout1.setColumns(1);
        informationPanel = new JPanel();
        informationPanel.setLayout(gridLayout1);
        informationPanel.add(iconLabel, null);
        informationPanel.add(getStatusTextArea(), null);
    }
    return informationPanel;
}

/**
 * This method initializes commandScrollPane
 *
 * @return javax.swing.JScrollPane
 */
private JScrollPane getCommandScrollPane() {
    if (commandScrollPane == null) {
        commandScrollPane = new JScrollPane();
        commandScrollPane.setFont(new Font("Comic Sans MS", Font.BOLD,
18));
        commandScrollPane.setViewportView(getCommandTextArea());
    }
    return commandScrollPane;
}

public void addCommand(String command) {
    getCommandTextArea().append(command);

    getCommandTextArea().setCaretPosition(getCommandTextArea().getText().length());
}

public String nextLine() {
    int start;
    String result;

    start = getCommandTextArea().getCaretPosition();
    while(waitingForResponse) {
        // wait
    }
    waitingForResponse = true;
    result =
getCommandTextArea().getText().substring(start, getCommandTextArea().getText().length());
    command.delete(0, command.length());
    return result;
}

```

```
}

/**
 * This method initializes statusTextArea
 *
 * @return javax.swing.JTextArea
 */
private JTextArea getStatusTextArea() {
    if (statusTextArea == null) {
        statusTextArea = new JTextArea();
        statusTextArea.setLineWrap(true);
        statusTextArea.setFont(new Font("Lucida Sans", Font.BOLD, 12));
        statusTextArea.setEditable(false);
        statusTextArea.setBackground(SystemColor.controlHighlight);
        statusTextArea.setWrapStyleWord(true);

    }
    return statusTextArea;
}

public void setIcon(ImageIcon icon) {
    iconLabel.setIcon(icon);
}

public void setStatus(String status) {
    getStatusTextArea().setText(status);
}

} // @jve:decl-index=0:visual-constraint="10,10"
```