Mobile Devices in Software Engineering

Lab 3

Objective

The objective of this lab is to:

1. Test various GUI components on your device
2. Continue to develop application on mobile devices

Experiment 1

In this experiment you will utilize several GUI components for a mobile application. It is wise to test GUI components on multiple devices.

Exercise 1

First copy the code below into a file called GUITests.java.

GUITests.java

```
import javax.microedition.lcdui.*;
import javax.microedition.midlet.*;

public class GuiTests extends MIDlet
implements CommandListener {
    // display manager
    Display display;

    // a menu with items
    // main menu
    List menu;

    // list of choices
    List choose;

    // textbox
    TextBox input;

    // ticker
    Ticker ticker = new Ticker(
    "Test GUI Components");

    // alerts
    final Alert soundAlert = new Alert("sound Alert");
```

```
// date
    DateField date = new DateField("Today's date: ",
DateField.DATE);

    // form
    Form form = new Form("Form for Stuff");

    // today's form
    Form today = new Form("Today's date");

    // gauge
    Gauge gauge = new Gauge("Progress Bar", false, 20, 9);

    // text field
    TextField textfield = new TextField(
    "TextField Label", "abc", 50, 0);

    // command
    static final Command backCommand =
     new Command("Back", Command.BACK, 0);
    static final Command mainMenuCommand =
    new Command("Main", Command.SCREEN, 1);
    static final Command exitCommand =
     new Command("Exit", Command.STOP, 2);
    String currentMenu;

    // constructor.
    public GuiTests() {
    }

    /**
     * Start the MIDlet by creating a list of
     * items and associating the
     * exit command with it.
     */
    public void startApp() throws
    MIDletStateChangeException {
      display = Display.getDisplay(this);
      menu = new List(
      "Test Components", Choice.IMPLICIT);
      menu.append("Test TextBox", null);
      menu.append("Test List", null);
      menu.append("Test Alert", null);
      menu.append("Test Date", null);
      menu.append("Test Form", null);
      menu.addCommand(exitCommand);
      menu.setCommandListener(this);
      menu.setTicker(ticker);
      mainMenu();
```

```
// form
      form.append(gauge);
      form.append(textfield);
      // today
      today.append(date);
    }

    public void pauseApp() {
      display = null;
      choose = null;
      menu = null;
      ticker = null;
      form = null;
      today = null;
      input = null;
      gauge = null;
      textfield = null;
    }

    public void destroyApp(boolean unconditional) {
      notifyDestroyed();
    }

    // main menu
    void mainMenu() {
      display.setCurrent(menu);
      currentMenu = "Main";
    }

    /**
     * Test the TextBox component.
     */
    public void testTextBox() {
      input = new TextBox
      ("Enter Some Text:", "", 10, TextField.ANY);
      input.setTicker(new Ticker(
      "Testing TextBox"));
      input.addCommand(backCommand);
      input.setCommandListener(this);
      input.setString("");
      display.setCurrent(input);
      currentMenu = "input";
    }

    /**
     * Test the List component.
     */
    public void testList() {
        choose = new List("Choose Items",
```

```
      Choice.MULTIPLE);
        choose.setTicker(new Ticker(
        "Testing List"));
        choose.addCommand(backCommand);
        choose.setCommandListener(this);
        choose.append("Item 1", null);
        choose.append("Item 2", null);
        choose.append("Item 3", null);
        display.setCurrent(choose);
        currentMenu = "list";
   }

   /**
    * Test the Alert component.
    */
   public void testAlert() {
      soundAlert.setType(AlertType.ERROR);
      //soundAlert.setTimeout(20);
      soundAlert.setString("** ERROR **");
      display.setCurrent(soundAlert);
   }

   /**
    * Test the DateField component.
    */
   public void testDate() {
      java.util.Date now = new java.util.Date();
      date.setDate(now);
      today.addCommand(backCommand);
      today.setCommandListener(this);
      display.setCurrent(today);
      currentMenu = "date";
   }

   /**
    * Test the Form component.
    */
   public void testForm() {
      form.addCommand(backCommand);
      form.setCommandListener(this);
      display.setCurrent(form);
      currentMenu = "form";
   }

   /**
    * Handle events.
    */
   public void commandAction(Command c,
   Displayable d) {
```

```
String label = c.getLabel();
      if (label.equals("Exit")) {
         destroyApp(true);
      } else if (label.equals("Back")) {
          if(currentMenu.equals("list")
          || currentMenu.equals("input") ||
               currentMenu.equals("date")
               || currentMenu.equals("form")) {
            // go back to menu
            mainMenu();
          }

      } else {
         List down = (List)display.getCurrent();
         switch(down.getSelectedIndex()) {
           case 0: testTextBox();break;
           case 1: testList();break;
           case 2: testAlert();break;
           case 3: testDate();break;
           case 4: testForm();break;
         }

      }
  }
}      String label = c.getLabel();
      if (label.equals("Exit")) {
         destroyApp(true);
      } else if (label.equals("Back")) {
          if(currentMenu.equals("list")
          || currentMenu.equals("input") ||
               currentMenu.equals("date")
               || currentMenu.equals("form")) {
            // go back to menu
            mainMenu();
          }

      } else {
         List down = (List)display.getCurrent();
         switch(down.getSelectedIndex()) {
           case 0: testTextBox();break;
           case 1: testList();break;
           case 2: testAlert();break;
           case 3: testDate();break;
           case 4: testForm();break;
         }

      }
  }
}
```

Take some time to read the above code and become familiar with the actions it takes.

Let us now add this code to our application.

1. Run the Sun Java Wireless Toolkit for CLDC.
2. Click on **New Project...**
3. Name the Project Name and MIDlet class name 'GUITests'
4. Proceed to the directory displayed in the Wireless Toolkit console instructing you the location to place the source files
5. Copy the above code into GUITests.java in the given folder
6. Now click on **Build** in the Wireless Toolkit and then **Run**

Explore the application and the different GUI components.

Exercise 2

Take the above code, and as done in Mobile Devices in Software Engineering – Lab I test out the application in various emulators and on a real Blackberry device. Note the differences in display styles and interaction abilities on the components.