# CSE 215: Programming Language II Lab

## Lab – 18

## JavaFX UI Controls

**Objective:**

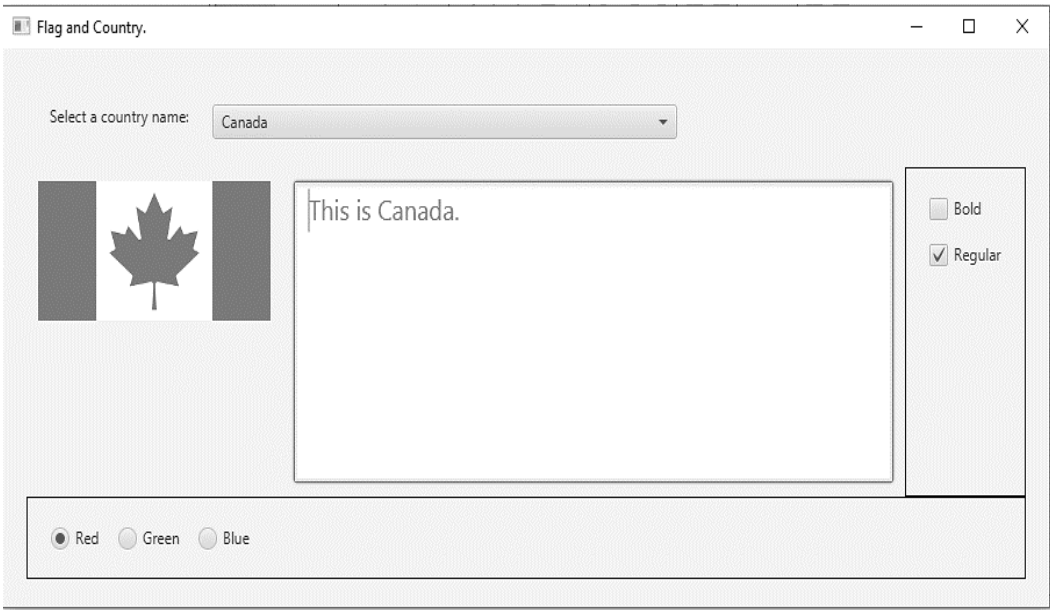
- To learn about Label, CheckBox, RadioButton, TextArea, ListView, ImageView

In the last two lab we saw the designing aspect and action performed based on an event of JavaFX. Today we will be covering on how to develop comprehensive user interface using UI controls provided by JavaFX.

- Button - A button is a control that triggers an action event when clicked. JavaFX provides regular buttons, toggle buttons, check box buttons, and radio buttons. The Labeled class defines the common properties for labels and buttons. A button is just like a label except that the button has the onAction property defined in the ButtonBase class, which sets a handler for handling a button's action.
- CheckBox - A CheckBox is used for the user to make a selection. When a check box is clicked (checked or unchecked), it fires an ActionEvent. To see if a check box is selected, use the isSelected() method.
- RadioButton - Radio buttons, also known as option buttons, enable the user to choose a single item from a group of choices. In appearance radio buttons resemble check boxes, but check boxes display a square that is either checked or blank, whereas radio buttons display a circle that is either filled (if selected) or blank (if not selected).
- TextArea - f you want to let the user enter multiple lines of text; you may create several instances of TextField. A better alternative, however, is to use TextArea, which enables the user to enter multiple lines of text.
- ComcoBox - A combo box, also known as a choice list or drop-down list, contains a list of items from which the user can choose. A combo box is useful for limiting a user's range of choices and avoids the cumbersome validation of data input.
- ListView - A list view is a control that basically performs the same function as a combo box, but it enables the user to choose a single value or multiple values.

Example:

**Write a JavaFX program that will let user to select a nation from a combo box and then display a country's flag and its description in a text area. When user selects the provided radio Button or checkbox appropriate action should have to be performed.**



Select a country name: Canada

This is Canada.

□ Bold

☑ Regular

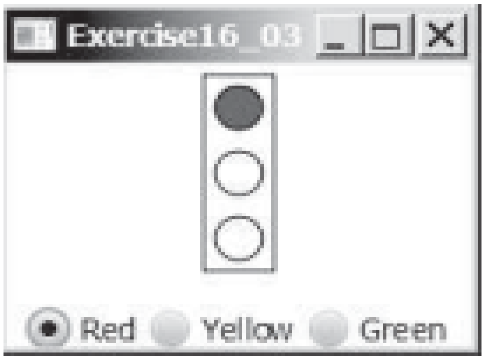
● Red ○ Green ○ Blue

**Code:**

```
public class UIControl extends Application {
  public static BorderPane bPane = new BorderPane();
  public static String title = "Select a country name: ";
  public static String[] item = { "Canada", "USA", "Bangladesh" };
  public static int selectedItem = 0;
  public static ImageView[] imageofCountry = { new
ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png"),
      new ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png"),
      new ImageView("file:\\C:\\Users\\Phantomhive\\Pictures\\739891.png") };
  public static String[] countryDesList = { "This is Canada.", "This USA.", "This is
Bangladesh" };

  public static Font boldFont = Font.font("Times New Roman", FontWeight.BOLD, 20);
  public static Font italicFont = Font.font("Times New Roman", FontPosture.ITALIC,
20);
  public static Font regularFont = Font.font("Times New Roman", FontPosture.REGULAR,
20);
  public static Font boldAndItalicFont = Font.font("Times New Roman", FontWeight.BOLD,
FontPosture.ITALIC, 20);
  public static TextArea textBody;

  @Override
  public void start(Stage primaryStage) {

    bPane.setPadding(new Insets(20, 20, 20, 20));
    setDisplay(selectedItem);
    bPane.setTop(getTopofPane(title, item));
    bPane.setRight(setPanelForCheckBox());
    bPane.setBottom(setPanelForRadioButton());
    BorderPane.setMargin(setPanelForRadioButton(), new Insets(10));

    Scene scene = new Scene(bPane, 900, 400);
    primaryStage.setScene(scene);
    primaryStage.setTitle("Flag and Country.");
    primaryStage.show();

  }

  public static HBox setPanelForRadioButton() {
    HBox panelForRadioButton = new HBox(16);

    final ToggleGroup toggleGroupOfColorRadioButtons = new ToggleGroup();

    RadioButton rbRed = new RadioButton("Red");
    rbRed.setToggleGroup(toggleGroupOfColorRadioButtons);
    RadioButton rbGreen = new RadioButton("Green");
    rbGreen.setToggleGroup(toggleGroupOfColorRadioButtons);
    RadioButton rbBlue = new RadioButton("Blue");
    rbBlue.setToggleGroup(toggleGroupOfColorRadioButtons);

    panelForRadioButton.setPadding(new Insets(20, 20, 20, 20));
    panelForRadioButton.setStyle("-fx-border-color: black;");
    panelForRadioButton.getChildren().addAll(rbRed, rbGreen, rbBlue);
    EventHandler<ActionEvent> handler = e -> {
      if (rbRed.isSelected()) {
        textBody.setStyle("-fx-text-fill: red;");
        bPane.setCenter(textBody);
      } else if (rbGreen.isSelected()) {
        textBody.setStyle("-fx-text-fill: green;");
        bPane.setCenter(textBody);
      } else if (rbBlue.isSelected()) {
        textBody.setStyle("-fx-text-fill: blue;");
        bPane.setCenter(textBody);
      } else {
        textBody.setStyle("-fx-text-fill: black;");
        bPane.setCenter(textBody);
      }
    };

    rbRed.setSelected(true);
    rbRed.setOnAction(handler);
    rbGreen.setOnAction(handler);
    rbBlue.setOnAction(handler);

    return panelForRadioButton;
  }

  public static VBox setPanelForCheckBox() {
    VBox panelForCheckBox = new VBox(16);
    panelForCheckBox.setPadding(new Insets(20, 20, 20, 20));
    panelForCheckBox.setStyle("-fx-border-color: black;");
    CheckBox chkBold = new CheckBox("Bold");
    CheckBox chkItalic = new CheckBox("Italic");
    panelForCheckBox.getChildren().addAll(chkBold, chkItalic);

    EventHandler<ActionEvent> handler = e -> {
      if (chkBold.isSelected() && chkItalic.isSelected()) {
        textBody.setFont(boldAndItalicFont);
        bPane.setCenter(textBody);
      } else if (chkItalic.isSelected()) {
        textBody.setFont(italicFont);
        bPane.setCenter(textBody);
      } else if (chkBold.isSelected()) {
        textBody.setFont(boldFont);
        bPane.setCenter(textBody);
      } else {
        textBody.setFont(regularFont);
        bPane.setCenter(textBody);
      }
    };

    chkBold.setOnAction(handler);
    chkItalic.setOnAction(handler);
    return panelForCheckBox;
  }

  public static TextArea getTextArea(int index) {
    TextArea taNote = new TextArea(countryDesList[selectedItem]);
    taNote.setPrefColumnCount(20);
    taNote.setPrefRowCount(5);
    taNote.setWrapText(true);
    taNote.setStyle("-fx-text-fill: red");
    taNote.setFont(regularFont);
    BorderPane.setMargin(taNote, new Insets(10));
    return taNote;
  }

  public static void setDisplay(int index) {
    ImageView selectedCountryImage = imageofCountry[selectedItem];
    selectedCountryImage.setFitHeight(100);
    selectedCountryImage.setFitWidth(200);
    BorderPane.setMargin(selectedCountryImage, new Insets(10));
    bPane.setLeft(selectedCountryImage);
    textBody = getTextArea(selectedItem);
    bPane.setCenter(textBody);
  }

  public static HBox getTopofPane(String title, String[] listItem) {

    HBox horizontalBox = new HBox(16);
    horizontalBox.setPadding(new Insets(20, 0, 20, 20));

    Label titleLabel = new Label(title);

    ComboBox<String> boxForItem = new ComboBox<>();

    ObservableList<String> boxItem = FXCollections.observableArrayList(listItem);
    boxForItem.getItems().addAll(boxItem);
    boxForItem.setValue(listItem[0]);
    boxForItem.setPrefWidth(400);
    boxForItem.setOnAction(new EventHandler<ActionEvent>() {
      @Override
      public void handle(ActionEvent event) {
        String value = boxForItem.getValue();
        selectedItem = boxItem.indexOf(value);
        setDisplay(selectedItem);
      }
    });

    horizontalBox.getChildren().addAll(titleLabel, boxForItem);
    return horizontalBox;
  }

  public static void main(String[] args) {
    launch(args);
  }
}
```

## Home Task

Write a program that simulates a traffic light. The program lets the user select one of three lights: red, yellow, or green. When a radio button is selected, the light is turned on. Only one light can be on at a time (see Figure). No light is on when the program starts.


Exercise16_03
● Red
○ Yellow
○ Green