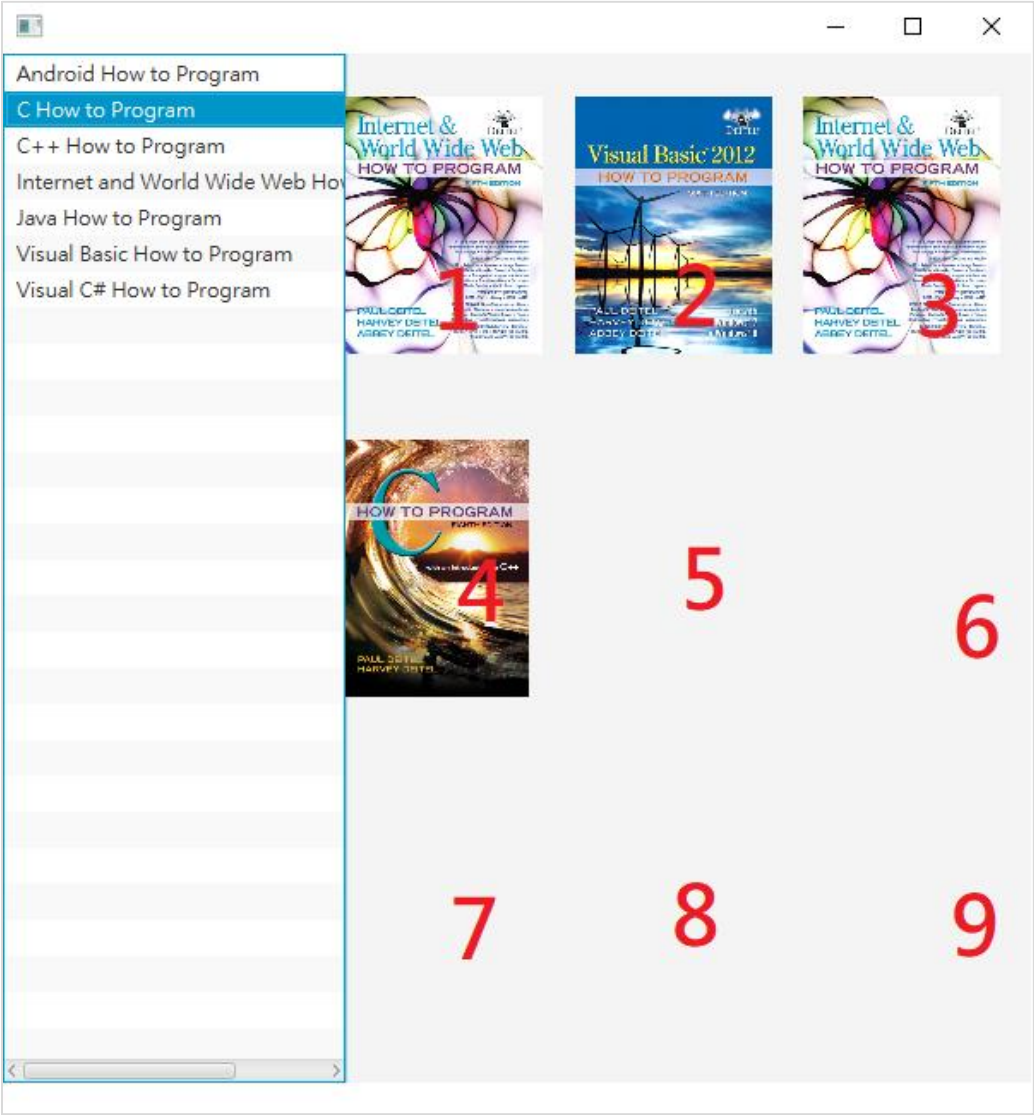
1. Cover Viewer

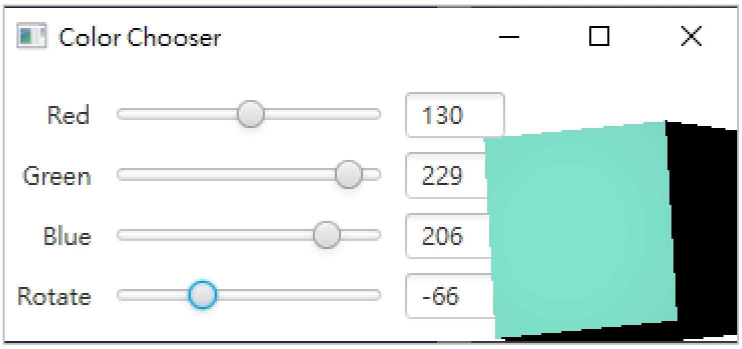
Please modify the code in HW5 problem3. Write a JavaFX program to choose the images to display. The left side is a list of the description of the images. When you choose different description, display the corresponding image on the right side. You need to display your choices in order. There are nine ImageView's on right GridPane as shown in following figure. The first image is displayed on the first ImageView. After choosing the second image, the second one is displayed on the second ImageView. If all the ImageView's are full, clear all the ImageView's.



2. Color Chooser

Write a JavaFX program to change the color and the rotation of the box. This program contains 3 sliders to set color(RGB), and another slider to change the rotation angle.

Sample Input & Output



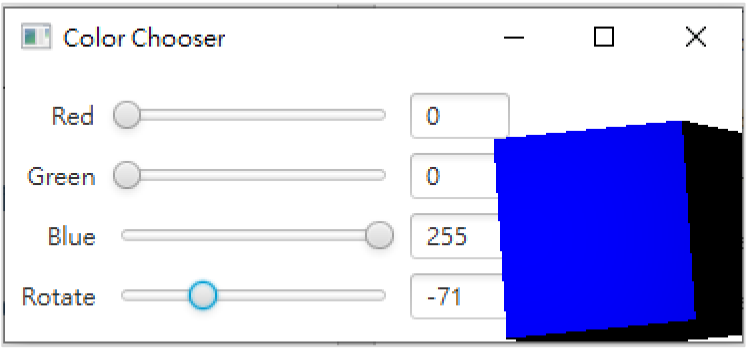
Color Chooser

Red 130

Green 229

Blue 206

Rotate -66



Color Chooser

Red 0

Green 0

Blue 255

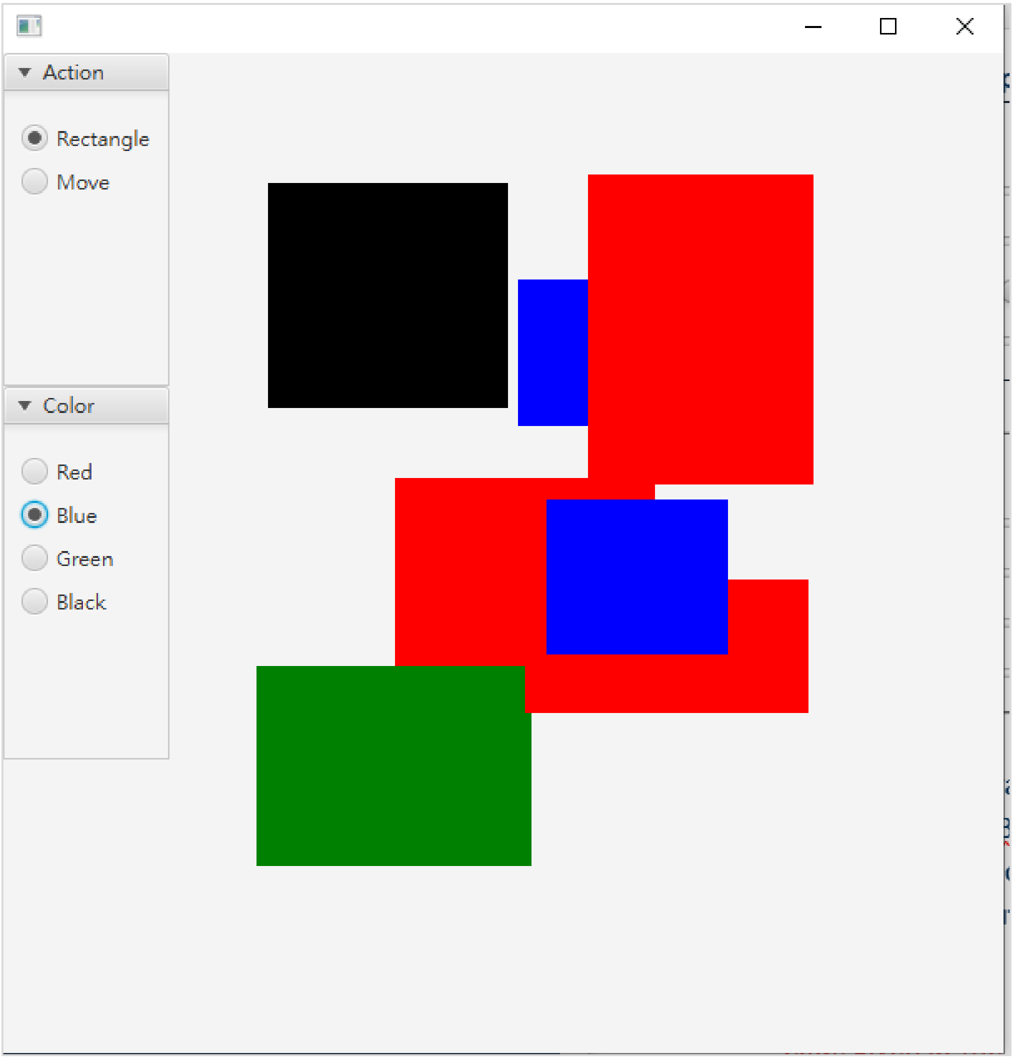
Rotate -71

3. Painting

Write a JavaFX program to draw multiple Rectangles.

- Use four Radiobutton's to determine the color of the rectangle.
- Draw a rectangle when you press and drag the mouse.
- Continue to change the size of the rectangle as you drag the mouse.

Sample Output



Note:

Rectangle function:

```
public final void setLayoutX(double value)
public final void setLayoutY(double value)
public final void setWidth(double value)
public final void setHeight(double value)
public final double getLayoutX()
public final double getLayoutY()
public final void setOnMouseEntered(EventHandler<?
super MouseEvent> value)
```

```
        public final void setOnMouseExited(EventHandler<?
        super MouseEvent> value)
```

MouseEvent function:

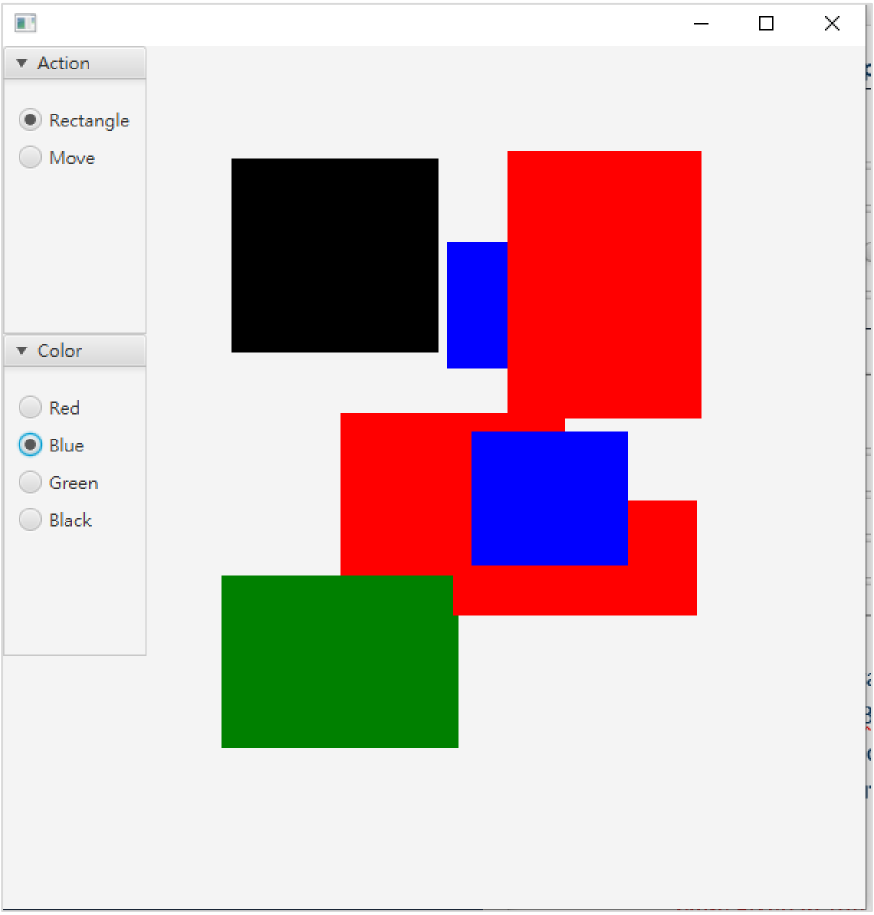
```
        public final double getX()
        public final double getY()
```

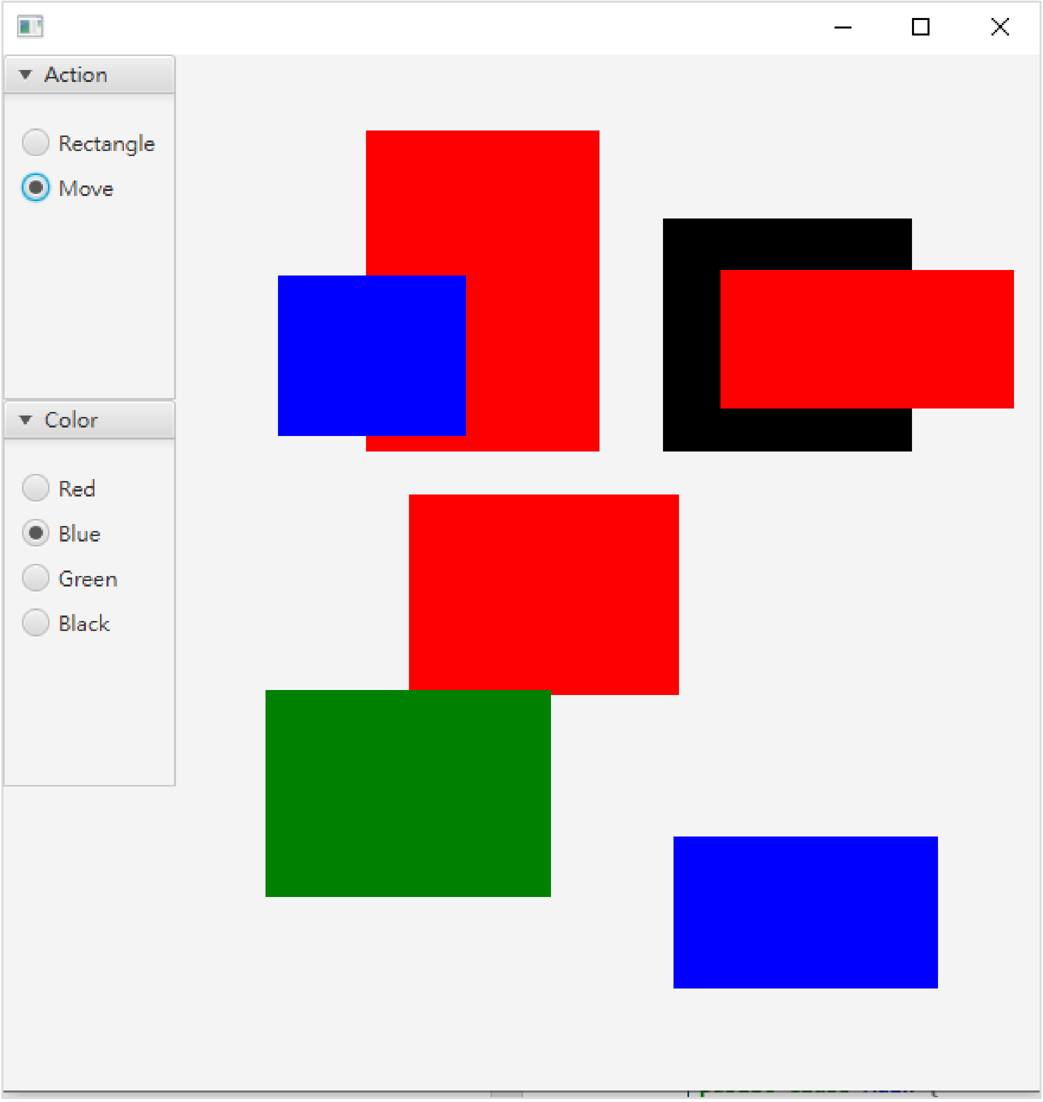
```
EventHandler<MouseEvent>(){
        @Overrdie
        public void handle(MouseEvent event){//……}
}
```

4. Painting (continued)

Please modify your code in problem 3. When you hover your mouse over the rectangle and drag it, the rectangle will move with the mouse.

Sample Output





Remember to change the object name

```
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.fxml.FXML;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.input.MouseEvent;
import javafx.scene.layout.Pane;
import javafx.scene.paint.Color;
import javafx.scene.paint.Paint;
import javafx.scene.shape.*;

public class Controller_forstudent {
        //3
        @FXML
        ToggleGroup color,action;
        @FXML RadioButton redbutton,bluebutton,greenbutton,blackbutton;
        //modify to your fxml
```

```
    @FXML RadioButton movebutton,rectanglebutton;
    @FXML Pane drawarea;
    Color drawcolor=Color.BLACK;
    Rectangle nowdrawrec;
    Rectangle nowmouseonrec;
    double drawx1,drawy1;
    //4
    enum NowAction {Move,Rect};
    double mousestartx,mousestarty;
    double reclayoutx,reclayouty;
    boolean ismove=false;   //Moving the rectangle
    NowAction nowaction=NowAction.Rect;

    public void initialize() {
        //initial some data
    }
    @FXML
    public void colorClick(ActionEvent e) {
            //change rectangle color

    }
    @FXML
    public void actionClick(ActionEvent e) {
        //you can use userdata here
        if(e.getSource()==movebutton)
            nowaction=NowAction.Move;
        else if(e.getSource()==rectanglebutton)
            nowaction=NowAction.Rect;

    }
    @FXML
    public void mousePressed(MouseEvent e) {
        if(nowaction==NowAction.Rect) {
            nowdrawrec=new Rectangle();

            drawarea.getChildren().add(nowdrawrec);
            //......... problem3
        }
        else if(nowaction==NowAction.Move) {
            if(nowmouseonrec!=null) {
                //............ problem4
                ismove=true;
            }
        }
    }
    @FXML
    public void mouseDragged(MouseEvent e) {
        if(nowaction==NowAction.Rect) {
            //...... problem3

        }
        else if(nowaction==NowAction.Move) {
            if(nowmouseonrec!=null) {
                //......... problem4
            }
        }
    }
    @FXML
    public void mouseReleased(MouseEvent e) {
        if(nowaction==NowAction.Rect) {
            //lambda in java
```

```
                  nowdrawrec.setOnMouseEntered((event)->{
                       //use(Rectangle)event.getSource() to get Rectangle
                       //when mouse enter rectangle
                        // .... problem4

                  });
                  nowdrawrec.setOnMouseExited(
                         new EventHandler<MouseEvent>() {
                         //when mouse leave rectangle
                        @Override
                        public void handle(MouseEvent event) {
                              //....problem4
                        }

                  });
            }
            else if(nowaction==NowAction.Move) {
                  ismove=false;
            }
      }

}
```