

```
import javafx.scene.layout.StackPane;
import java.io.FileInputStream;                // io for input and out put operations
import java.io.FileNotFoundException;
import javafx.application.Application;
import javafx.beans.value.ChangeListener;
import javafx.beans.value.ObservableValue;
import javafx.event.ActionEvent;
import javafx.event.EventHandler;
import javafx.scene.Scene;                  //For creating a scene

import javafx.scene.layout.GridPane;        // layout organises GUI components
import javafx.scene.layout.ColumnConstraints;
import javafx.scene.control.Label;
import javafx.scene.control.Button;
import javafx.scene.control.RadioButton;
import javafx.scene.control.ToggleGroup;
import javafx.scene.control.Toggle;
import javafx.scene.control.Slider;
import javafx.scene.control.CheckBox;
import javafx.scene.layout.HBox;
import javafx.scene.layout.VBox;
import javafx.scene.image.Image;
import javafx.scene.image.ImageView;
import javafx.scene.image.WritableImage;
import javafx.scene.image.PixelWriter;
import javafx.scene.paint.Color;
import javafx.stage.Stage;

import java.lang.reflect.Array;
import java.util.ArrayList;
import java.io.*;
```

```

import java.lang.Math.*;

import javafx.geometry.HPos;

import java.lang.Math;

public class Main extends Application { //Declares a public class named "Main", that extends
"Application" class

//main class for all javaFX applications

    int Width = 700; //Declare 2 integer variables "width" & "height.
Representing the dimensions of the output image

    int Height = 700;

    int green_col = 255; //just for the test example //Declares an integer variable named "green_col".
Initialized to 255. Not used in rest of code.

    double r_value; //Declares double variables

    double g_value; //Store values of red, green, blue
components of colour, to render the spheres.

    double b_value;

    ArrayList<Sphere> spheres = new ArrayList<>(); //Creates an "ArrayList" object named
"spheres". Will store the "Sphere" objects to be rendered.

    @Override //Override
annotation.

    public void start(Stage stage) throws FileNotFoundException { // Method named "Start" is
overriding a method of the same name, from the superclass "Application".

//Method initialises GUI.

Starts the program.

    stage.setTitle("Ray Tracing"); //Set title of the main window to "Ray
Tracing".

//We need 3 things to see an image

//1. We create an image we can write to

```

WritableImage image = new WritableImage(Width, Height); //Create new "WritableImage" object named "image". Has dimensions of "width" and "height".

//2. We create a view of that image

ImageView view = new ImageView(image); //Create "ImageView" object named "view".
It will display an "image".

//3. Add to the pane (below)

//Create the simple GUI

Slider r_slider = new Slider(0, 1, 0.2); //These create 3 "slider" objects, that allow
the user to adjust the red, green and blue colors.

Slider g_slider = new Slider(0, 1, 0.3);

Slider b_slider = new Slider(0, 1, 0.4);

Button One = new Button("Sphere 1");

Button Two = new Button("Sphere 2");

Button Three = new Button("Sphere3");

r_value = r_slider.getValue();

g_value = g_slider.getValue();

b_value = b_slider.getValue();

//Add all the event handlers

```
r_slider.valueProperty().addListener(  
    new ChangeListener<Number>() {  
        public void changed(ObservableValue<? extends Number>  
            observable, Number oldValue, Number newValue) {  
            r_value = newValue.intValue();  
            Render(image);  
        }  
    });
```

```

g_slider.valueProperty().addListener(
    new ChangeListener<Number>() {
        public void changed(ObservableValue<? extends Number>
            observable, Number oldValue, Number newValue) {

            g_value = newValue.intValue();
            Render(image);
        }
    });

```

```

b_slider.valueProperty().addListener(
    new ChangeListener<Number>() {
        public void changed(ObservableValue<? extends Number>
            observable, Number oldValue, Number newValue) {

            b_value = newValue.intValue();
            Render(image);
        }
    });

```

```

//The following is in case you want to interact with the image in any way
//e.g., for user interaction, or you can find out the pixel position for debugging
view.addEventHandler(javafx.scene.input.MouseEvent.MOUSE_PRESSED, event -> {
    System.out.println(event.getX() + " " + event.getY());
    event.consume();
});

```

```

Render(image);

```

```

GridPane root = new GridPane();
root.setVgap(12);

```

```

root.setHgap(12);

//3. (referring to the 3 things we need to display an image)
//we need to add it to the pane
root.add(view, 0, 0);
root.add(r_slider, 0, 1);
root.add(g_slider, 0, 2);
root.add(b_slider, 0, 3);

// create a stack pane
StackPane r = new StackPane();

// add button
r.getChildren().add(One);
r.getChildren().add(Two);
r.getChildren().add(Three);


//Display to user
Scene scene = new Scene(root, 1200, 1200);
stage.setScene(scene);
stage.show();
}

public void Render(WritableImage image) {
    //Get image dimensions, and declare loop variables
    int w = (int) image.getWidth(), h = (int) image.getHeight();
    int i = 0;
    int j = 0;
    PixelWriter image_writer = image.getPixelWriter();

```

```
int r = 50;
```

```
Vector cs1 =new Vector(200, 350, 100);
```

```
Vector cs2 =new Vector(100, 250, 300);
```

```
Vector cs3 =new Vector(50, 100, 300);
```

```
Vector col1 =new Vector(r_value,g_value,b_value);
```

```
Vector col2 =new Vector(r_value,g_value,b_value);
```

```
Vector col3 =new Vector(r_value,g_value,b_value);
```

```
Sphere FirstSphere = new Sphere(20,cs1 , col1);
```

```
Sphere SecondSphere = new Sphere(100,cs2 , col2);
```

```
Sphere ThirdSphere = new Sphere(90,cs3 , col3);
```

```
spheres.add(FirstSphere);
```

```
spheres.add(SecondSphere);
```

```
spheres.add(ThirdSphere);
```

```
for (j = 0; j < h; j++) {
```

```
    for (i = 0; i < w; i++) {
```

```
        Vector color = new Vector(0, 0, 0);
```

```
        for (Sphere sphere : spheres) {
```

```
            Vector sphereColor = sphere.intersection(i, j, h, w);
```

```
            color = color.add(sphereColor);
```

```
        }
```

```
        image_writer.setColor(i, j, Color.color(color.getX(), color.getY(), color.getZ(), 1));
```

```
    }
```

```
}
```

```
}
```

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

```
    launch();
```

```
}
```

```
}
```