SphereAndBox.java

```
/*
 * Copyright (c) 2013, Oracle and/or its affiliates. All rights reserved.
 * DO NOT ALTER OR REMOVE COPYRIGHT NOTICES OR THIS FILE HEADER.
 *
 * This code is free software; you can redistribute it and/or modify it
 * under the terms of the GNU General Public License version 2 only, as
 * published by the Free Software Foundation.  Oracle designates this
 * particular file as subject to the "Classpath" exception as provided
 * by Oracle in the LICENSE file that accompanied this code.
 *
 * This code is distributed in the hope that it will be useful, but WITHOUT
 * ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or
 * FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License
 * version 2 for more details (a copy is included in the LICENSE file that
 * accompanied this code).
 *
 * You should have received a copy of the GNU General Public License version
 * 2 along with this work; if not, write to the Free Software Foundation,
 * Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110-1301 USA.
 *
 * Please contact Oracle, 500 Oracle Parkway, Redwood Shores, CA 94065 USA
 * or visit www.oracle.com if you need additional information or have any
 * questions.
 */

import javafx.application.Application;
import javafx.event.EventHandler;
import javafx.scene.Group;
import javafx.scene.PerspectiveCamera;
import javafx.scene.PointLight;
import javafx.scene.Scene;
import javafx.scene.input.MouseEvent;
import javafx.scene.paint.Color;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.shape.Box;
import javafx.scene.shape.Sphere;
import javafx.scene.transform.Rotate;
import javafx.stage.Stage;

public class SphereAndBox extends Application {

    double anchorX, anchorY, anchorAngle;

    private PerspectiveCamera addCamera(Scene scene) {
        PerspectiveCamera perspectiveCamera = new PerspectiveCamera(false);
        scene.setCamera(perspectiveCamera);
        return perspectiveCamera;
    }

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // Remove this line once dirtyopts bug is fixed for 3D primitive
        System.setProperty("prism.dirtyopts", "false");
        launch(args);
    }

    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("SphereAndBox");

        final PhongMaterial redMaterial = new PhongMaterial();
        redMaterial.setSpecularColor(Color.ORANGE);
        redMaterial.setDiffuseColor(Color.RED);

        final PhongMaterial blueMaterial = new PhongMaterial();
```

```
        blueMaterial.setDiffuseColor(Color.BLUE);
        blueMaterial.setSpecularColor(Color.LIGHTBLUE);

        final Box red = new Box(400, 400, 400);
        red.setMaterial(redMaterial);

        final Sphere blue = new Sphere(200);
        blue.setMaterial(blueMaterial);

        blue.setTranslateX(250);
        blue.setTranslateY(250);
        blue.setTranslateZ(50);
        red.setTranslateX(250);
        red.setTranslateY(250);
        red.setTranslateZ(450);

        final Group parent = new Group(red, blue);
        parent.setTranslateZ(500);
        parent.setRotationAxis(Rotate.Y_AXIS);


        final Group root = new Group(parent);

        final Scene scene = new Scene(root, 500, 500, true);

        scene.setOnMousePressed(new EventHandler<MouseEvent>() {
            @Override public void handle(MouseEvent event) {
                anchorX = event.getSceneX();
                anchorY = event.getSceneY();
                anchorAngle = parent.getRotate();
            }
        });

        scene.setOnMouseDragged(new EventHandler<MouseEvent>() {
            @Override public void handle(MouseEvent event) {
                parent.setRotate(anchorAngle + anchorX -  event.getSceneX());
            }
        });

        PointLight pointLight = new PointLight(Color.ANTIQUEWHITE);
        pointLight.setTranslateX(15);
        pointLight.setTranslateY(-10);
        pointLight.setTranslateZ(-100);

        root.getChildren().add(pointLight);

        addCamera(scene);
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```