**Code:**

package graphicproject;

import javafx.scene.image.Image;

import javafx.application.Application;

import javafx.scene.Camera;

import javafx.scene.Node;

import javafx.scene.PerspectiveCamera;

import javafx.scene.PointLight;

import javafx.scene.Scene;

import javafx.scene.input.KeyEvent;

import javafx.scene.layout.Background;

import javafx.scene.layout.BackgroundImage;

import javafx.scene.layout.BackgroundPosition;

import javafx.scene.layout.BackgroundRepeat;

import javafx.scene.layout.BackgroundSize;

import javafx.scene.layout.StackPane;

import javafx.scene.paint.Color;

import javafx.scene.paint.PhongMaterial;

import javafx.scene.shape.Cylinder;

import javafx.scene.shape.Sphere;

import javafx.scene.text.Text;

import javafx.scene.transform.Rotate;

import javafx.scene.transform.Transform;

import javafx.stage.Stage;

public class

GraphicProject extends Application {

private static final int WIDTH = 1;

private static final int HEIGHT = 1;

@Override

public void start(Stage primaryStage) {

//for objects//

SmartGroup group = new SmartGroup();

//for sun and background and text//

SmartGroup g= new SmartGroup();

group.getChildren().addAll(prepareLightSource());

Background b=preparb();

g.setBackground(b);

Cylinder cylinder = preparc();

Sphere sphere = preparsphere();

Sphere sphere2 = preparsphere2();

Sphere sphere3 = preparsphere3();

Sphere sphere4 = preparsphere4();

Sphere sphere5 = preparsphere5();

Sphere sphere6 = preparsphere6();

Sphere sphere7 = preparsphere7();

Sphere sphere8 = preparsphere8();

Sphere sphere9 = preparsphere9();

group.getChildren().add(sphere2);

group.getChildren().add(sphere4);

group.getChildren().add(sphere5);

group.getChildren().add(sphere3);

group.getChildren().add(sphere7);

group.getChildren().add(sphere6);

group.getChildren().add(sphere8);

group.getChildren().add(sphere9);

group.getChildren().add(cylinder);

g.getChildren().add(group);

Camera camera = new PerspectiveCamera();

Scene scene = new Scene(g, WIDTH,HEIGHT);

Text text = new Text("Milkey way");

text.setFill(Color.RED);

text.setTranslateX(10);

text.setTranslateY(-300);

g.getChildren().add(text);

g.getChildren().add(sphere);

primaryStage.setScene(scene);

primaryStage.show();

scene.setCamera(camera);

sphere.translateXProperty().set(10 );

sphere.translateYProperty().set(50);

sphere2.translateXProperty().set(300 );

sphere2.translateYProperty().set(100);

sphere3.translateXProperty().set(-30 );

sphere3.translateYProperty().set(270);

sphere4.translateXProperty().set(-300 );

sphere4.translateYProperty().set(100);

sphere5.translateXProperty().set(-100);

sphere5.translateYProperty().set(-70);

sphere6.translateXProperty().set(400);

sphere6.translateYProperty().set(-200);

sphere7.translateXProperty().set(550);

sphere7.translateYProperty().set(-60);

sphere8.translateXProperty().set(-550);

sphere8.translateYProperty().set(-200);

sphere9.translateXProperty().set(-300);

sphere9.translateYProperty().set(-200);

group.translateXProperty().set(WIDTH );

group.translateYProperty().set(HEIGHT);

primaryStage.addEventHandler(KeyEvent.KEY\_PRESSED, event -> {

switch (event.getCode()) {

case W:

sphere.translateZProperty().set(sphere.getTranslateZ() + 100);

break;

case S:

sphere.translateZProperty().set(sphere.getTranslateZ() - 100);

break;

case P:

group.rotateByY(10);

break;

case L:

group.rotateByY(-10);

break;

case F:

cylinder.translateZProperty().set(cylinder.getTranslateZ() + 100);

break;

case N:

cylinder.translateZProperty().set(cylinder.getTranslateZ() - 100);

break;

}});

primaryStage.setTitle("galxy");

primaryStage.setScene(scene);

primaryStage.show();

} private Node[] prepareLightSource() {

PointLight pointLight = new PointLight();

pointLight.setColor(Color.WHITE);

Sphere sphere = new Sphere(10);

sphere.getTransforms().setAll(pointLight.getTransforms());

return new Node[]{pointLight, sphere};

}

private Background preparb(){

Image img = new Image("/Resources/red.jpg");

BackgroundImage bImg = new BackgroundImage(img,

BackgroundRepeat.NO\_REPEAT,

BackgroundRepeat.NO\_REPEAT,

BackgroundPosition.DEFAULT,

BackgroundSize.DEFAULT);

Background bGround = new Background(bImg);

return bGround;

}

private Sphere preparsphere() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/sun.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(150);

s.setMaterial(material);

return s;

}

private Sphere preparsphere3() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/mars.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(50);

s.setMaterial(material);

return s;

}

private Sphere preparsphere2() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/venos.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(70);

s.setMaterial(material);

return s;

}

private Sphere preparsphere4() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/earth.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(100);

s.setMaterial(material);

return s;

}

private Sphere preparsphere5() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/Mercury.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(40);

s.setMaterial(material);

return s;

}

private Sphere preparsphere6() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/saturn.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(110);

s.setMaterial(material);

return s;

}

private Sphere preparsphere7() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/urans.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(60);

s.setMaterial(material);

return s;

}

private Sphere preparsphere8() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/neptun.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(50);

s.setMaterial(material);

return s;

}

private Sphere preparsphere9() {

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new

Image(getClass().getResourceAsStream("/Resources/jupter.jpg")));

material.setSpecularColor(Color.WHITE);

Sphere s = new Sphere(115);

s.setMaterial(material);

return s;

}

private Cylinder preparc() {

Cylinder cylinder = new Cylinder();

cylinder.setHeight(150);

cylinder.setRadius(50);

cylinder.setTranslateX(-500);

cylinder.setTranslateY(20);

PhongMaterial material = new PhongMaterial();

material.setDiffuseMap(new Image

("/Resources/sat.jpg"));

material.setSpecularColor(Color.WHITE);

cylinder.setMaterial(material);

return cylinder;

}

class SmartGroup extends StackPane {

Rotate r;

Transform t = new Rotate();

void rotateByY(int ang) {

r = new Rotate(ang, Rotate.Y\_AXIS);

t = t.createConcatenation(r);

this.getTransforms().clear();

this.getTransforms().addAll(t);

}

}

public static void main(String[] args) {

launch(args);

}

}

**Output:**

Without lighting

**صورة تحتوي على داخلي

تم إنشاء الوصف تلقائياً**

With lightingصورة تحتوي على داخلي, أسود

تم إنشاء الوصف تلقائياً