Lecture

Images and 3D shapes

4 3D Shapes

In general, a 3D shape is a geometrical figure that can be drawn on the XYZ plane. These include a **Cylinder**, **Sphere** and a **Box**. Each of the above mentioned 3D shape is represented by a class and all these classes belong to the package javafx.scene.shape. The class named Shape3D is the base class of all the 3-Dimensional shapes in JavaFX. The following table gives you the list of various 3D shapes provided by JavaFX.

S.No	Shape & Description
1	Box: A cuboid is a three-dimensional shape with a length (depth), width, and a height. In JavaFX a three-dimensional box is represented by a class named Box. This class belongs to the package javafx.scene.shape. This class has 3 properties of the double datatype namely – width – The width of the box. height – The height of the box. depth – The depth of the box.
	Box box = new Box(width, height, depth);
	Or, by using their respective setter methods as follows –
	Box box = new Box(); setWidth(value); setHeight(value); setDepth(value);
2	Cylinder: is a closed solid that has two parallel (mostly circular) bases connected by a curved surface. It is described by two parameters, namely, the radius of its circular base and the height of the cylinder. In JavaFX, a cylinder is represented by a class named Cylinder. This class package javafx.scene.shape. This class has 2 properties of the double datatype namely – height – The height of the Cylinder. radius – The radius of the Cylinder.
	Cylinder cylinder = new Cylinder(radius, height);
	Or, by using their respective setter methods as follows –
	Cylinder cylinder = new Cylinder(); setRadius(value); setHeight(value);

Images and 3D shapes

3Lectusphere: is defined as the set of points that are all at the same distance r from a given point in a 3D space.

This distance r is the radius of the sphere and the given point is the centre of the sphere. In JavaFX, a sphere is represented by a class named Sphere. This class belongs to the package javafx.scene.shape. This class has a property named radius of double datatype. It represents the radius of a Sphere.



```
Sphere sphere = new Sphere(radius);

Or, by using a method named setRadius() as follows –

Sphere sphere = new Sphere();
setRadius(value);

ge javafxapplication21;
```

```
package javafxapplication21;
import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Scene;
import javafx.scene.Group;
import javafx.scene.image.Image;
import javafx.scene.PerspectiveCamera;
import javafx.scene.paint.Color;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.shape.Shape3D;
import javafx.scene.shape.*;
import java.io.FileInputStream;
import java.io.FileNotFoundException;
public class JavaFXApplication21 extends Application {
  @Override
  public void start(Stage primaryStage) throws FileNotFoundException{
    final PhongMaterial REDMaterial = new PhongMaterial();
                                                               اللون المنتشر Diffuse color //
    REDMaterial.setDiffuseColor(Color.RED);
    REDMaterial.setSpecularColor(Color.LIGHTPINK);
                                                              اللون البراق specular color //
    final PhongMaterial GRMaterial = new PhongMaterial();
    GRMaterial.setDiffuseColor(Color.GREEN);
    GRMaterial.setSpecularColor(Color.LIGHTGREEN);
    final PhongMaterial blueMaterial = new PhongMaterial();
    blueMaterial.setDiffuseColor(Color.BLUE);
    blueMaterial.setSpecularColor(Color.GOLDENROD);
```

Cechong Material material 4 new Prioring Material (1) Shapes setDiffuseMap(new Image(new FileInputStream("C:\\\\Users\\\\D. Salma\\\\ Desktop\\\\ s3.jpg"))); Box b = new Box(150f, 250f, 200f); b.setRotate(-345); // b.setWidth(250); //b.setHeight(250); //b.setDepth(150); //b.setDrawMode(DrawMode.LINE); b.setCullFace(CullFace.BACK); b.setLayoutX(300); b.setLayoutY(300); b.setMaterial(material4); //Setting the drawing mode of the box b.setDrawMode(DrawMode.FILL); Shape3D c = new Cylinder(50, 300); c.setLayoutX(100); c.setLayoutY(400); c.setMaterial(REDMaterial); //c.setRotate(135); Shape3D c1 = new Cylinder(50, 150); c1.setLayoutX(100); c1.setLayoutY(250); c1.setMaterial(GRMaterial); Shape3D s = new Sphere(50, 7); s.setMaterial(GRMaterial); s.setLayoutX(100);

3

s.setLayoutY(100);

s1.setLayoutX(300);
s1.setLayoutY(150);
s1.setTranslateX(300);

s1.setTranslateY(150);

s1.setCullFace(CullFace.BACK);

Group root = new Group();

Shape3D s1 = new Sphere(50, 130); s1.setMaterial(blueMaterial);

//Setting the cull face of the sphere to back

root.getChildren().addAll(b, s1, c, c1, s);

= new Scene (rob) and 3D shapes

veCamera camera = new PerspectiveCamera(false); camera.setTranslateX(50);

```
camera.setTranslateY(50);
camera.setTranslateZ(25);
scene.setCamera(camera);
```

```
primaryStage.setTitle("Hello World!");
primaryStage.setScene(scene);
primaryStage.show();
```

} }

Properties of 3D Objects

Cull Face

culling is the removal of improperly oriented parts of a shape (which are not visible in the view area). You can set the Cull Face of a shape using the method setCullFace() as shown below –

box.setCullFace(CullFace.NONE);

The stroke type of a shape can be -

None – No culling is performed (CullFace.NONE).

Front - All the front facing polygons are culled. (CullFace.FRONT).

Back – All the back facing polygons are culled. (StrokeType.BACK).

By default, the cull face of a 3-Dimensional shape is Back.

Drawing Modes

It is the property is of the type DrawMode and it represents the drawing mode used to draw the current 3D shape. You can choose the draw mode to draw a 3D shape using the method setDrawMode () as follows –

box.setDrawMode(DrawMode.FILL);

In JavaFX, you can choose two draw modes to draw a 3D shape, which are -**Fill** – This mode draws and fills a 2D shape (DrawMode.FILL). **Line** – This mode draws a 3D shape using lines (DrawMode.LINE).

By default, the drawing mode of a 3Dimensional shape is fill.

```
you must don't forget to open library
  import javafx.scene.shape.DrawMode;
```

Material

The cull Face property is of the type Material and it is used to choose the surface of the material of a 3D shape. You can set the material of a 3D shape using the method setCullFace() as follows -

Images and 3D shapes

aterial(material);

As mentioned above for this method, you need to pass an object of the type Material. The PhongMaterial class of the package javafx.scene.paint is a sub class of this class and provides 7 properties that represent a Phong shaded material. You can apply all these type of materials to the surface of a 3D shape using the setter methods of these properties. Following are the type of materials that are available in JavaFX -

import javafx.scene.paint.PhongMaterial;

bumpMap – This represents a normal map stored as a RGB Image.

diffuseMap - This represents a diffuse map.

selfilluminationMap - This represents a self-illumination map of this PhongMaterial.

specularMap – This represents a specular map of this PhongMaterial.

diffuseColor - This represents a diffuse color of this PhongMaterial.

specularColor – This represents a specular color of this PhongMaterial.

specularPower – This represents a specular power of this PhongMaterial.

By default, the material of a 3-Dimensional shape is a PhongMaterial with a diffuse color of light gray.

```
// you must don't forget to open library
package javafxapplication48;
import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.PerspectiveCamera;
import javafx.scene.Scene;
import javafx.scene.image.Image;
import javafx.scene.paint.Color;
import javafx.scene.paint.PhongMaterial;
import javafx.scene.shape.Cylinder;
import javafx.stage.Stage;
public class JavaFXApplication48 extends Application {
  @Override
  public void start(Stage Stg) {
   //Drawing Cylinder1
   Cylinder cylinder1 = new Cylinder();
   //Setting the properties of the Cylinder
   cylinder1.setHeight(130.0f);
   cylinder1.setRadius(30.0f);
   //Setting the position of the Cylinder
   cylinder1.setTranslateX(100);
   cylinder1.setTranslateY(75);
```

Preparing the phong material of type bump map 3D shapes

```
ngMaterial material1 = new PhongMaterial();
material1.setBumpMap(new Image ("http://livedoor.blogimg.jp/jin115/imgs/3/2/324041bc.jpg"));
//Setting the bump map material to Cylinder1
cylinder1.setMaterial(material1);
//Drawing Cylinder2
Cylinder cylinder2 = new Cylinder();
//Setting the properties of the Cylinder
cylinder2.setHeight(130.0f);
cylinder2.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder2.setTranslateX(200);
cylinder2.setTranslateY(75);
//Preparing the phong material of type diffuse map
PhongMaterial material2 = new PhongMaterial();
material2.setDiffuseMap(new Image ("http://livedoor.blogimg.jp/jin115/imgs/3/2/324041bc.jpg"));
//Setting the diffuse map material to Cylinder2
cylinder2.setMaterial(material2);
//Drawing Cylinder3
Cylinder cylinder3 = new Cylinder();
//Setting the properties of the Cylinder
cylinder3.setHeight(130.0f);
cylinder3.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder3.setTranslateX(300);
cylinder3.setTranslateY(75);
//Preparing the phong material of type Self Illumination Map
PhongMaterial material3 = new PhongMaterial();
material3.setSelfIlluminationMap(new Image ("http://livedoor.blogimg.jp/jin115/imgs/3/2/324041bc.jpg"));
//Setting the Self Illumination Map material to Cylinder3
cylinder3.setMaterial(material3);
//Drawing Cylinder4
```

Lectioner cylinder4 = new Cylinder(); and 3D shapes

```
//Setting the properties of the Cylinder
cylinder4.setHeight(130.0f);
cylinder4.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder4.setTranslateX(400);
cylinder4.setTranslateY(75);
//Preparing the phong material of type Specular Map
PhongMaterial material4 = new PhongMaterial();
material4.setSpecularMap(new Image ("http://livedoor.blogimg.jp/jin115/imgs/3/2/324041bc.jpg"));
//Setting the Specular Map material to Cylinder4
cylinder4.setMaterial(material4);
//Drawing Cylinder5
Cylinder cylinder5 = new Cylinder();
//Setting the properties of the Cylinder
cylinder5.setHeight(130.0f);
cylinder5.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder5.setTranslateX(100);
cylinder5.setTranslateY(300);
//Preparing the phong material of type diffuse color
PhongMaterial material5 = new PhongMaterial();
material5.setDiffuseColor(Color.BLANCHEDALMOND);
//Setting the diffuse color material to Cylinder5
cylinder5.setMaterial(material5);
//Drawing Cylinder6
Cylinder cylinder6 = new Cylinder();
//Setting the properties of the Cylinder
cylinder6.setHeight(130.0f);
cylinder6.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder6.setTranslateX(200);
cylinder6.setTranslateY(300);
```

paring the phong material of type specifian color 3D shapes

PhongMaterial material6 = new PhongMaterial();

```
//setting the specular color map to the material
material6.setSpecularColor(Color.BLANCHEDALMOND);
//Setting the specular color material to Cylinder6
cylinder6.setMaterial(material6);
//Drawing Cylinder7
Cylinder cylinder7 = new Cylinder();
//Setting the properties of the Cylinder
cylinder7.setHeight(130.0f);
cylinder7.setRadius(30.0f);
//Setting the position of the Cylinder
cylinder7.setTranslateX(300);
cylinder7.setTranslateY(300);
//Preparing the phong material of type Specular Power
PhongMaterial material7 = new PhongMaterial();
material7.setSpecularPower(0.1);
//Setting the Specular Power material to the Cylinder
cylinder7.setMaterial(material7);
//Creating a Group object
Group root = new Group(cylinder1, cylinder2, cylinder3,
cylinder4, cylinder5, cylinder6, cylinder7);
//Creating a scene object
Scene scene = new Scene(root, 600, 400);
//Setting camera
PerspectiveCamera camera = new PerspectiveCamera(false);
camera.setTranslateX(0);
camera.setTranslateY(0);
camera.setTranslateZ(-10);
scene.setCamera(camera);
//Setting title to the Stage
Stg.setTitle("Drawing a cylinder");
//Adding scene to the stage
Stg.setScene(scene);
```

Second

Images and 3D shapes

}

```
Lecture //Displaying the contents of the stage
  Stg.show();
}
```

