

JDORFX

Providing 3D Graphics to ooRexx

Philipp Schaller

36th International Rexx Language Symposium 7.May.2025

Agenda

- Background
- Development
- Nutshell Examples



Background

ooRexx 1

- Easy to learn syntax
- No strict types
- Setting Environment

ooRexx 2

- Instruction Types
 - Assignment Instructions
 - Keyword Instructions (e.g. DO)
 - Command Instructions
 - Will be sent to shell for execution
 - Or addressing other Rexx Command Handler with ADDRESS keyword
 - Directive Instructions (since ooRexx)
 - :: requires
 - Executed in setup phase

Java

- One of the most used programming languages today
- Java virtual machine (JVM) allows to run Java program everywhere once compiled
- Extensive libraries
- Strictly typed language
- Case sensitive
- Classes organized in packages

BSF4ooRexx850

- Bean Scripting Framework for ooRexx
- ooRexx function and class package
- Allows ooRexx to use all of Java
- Combines ooRexx human oriented design and Java functionalities
- Camouflaging Java objects as ooRexx objects
- Since BSF4ooRexx850: ability to implement Rexx command handlers in Java

JDOR

- Runtime Library written in Java
- Command Handler as part of BSF4ooRexx850
- Implements awt based Java2D
- No knowledge of Java necessary
- JavaDrawingHandler

Objectives of JDORFX

- Developing framework based on JavaFX
- Maintain syntax compatibility with JDOR / seamless switch
- Provide 3D graphics to ooRexx programmers

Java awt vs JavaFX

Java awt

- First Release 1996
- Basic GUI toolkit
- Several Packages
- Uses Java Native Interface (JNI)
- GUI control and event management in a separate AWT/GUI thread

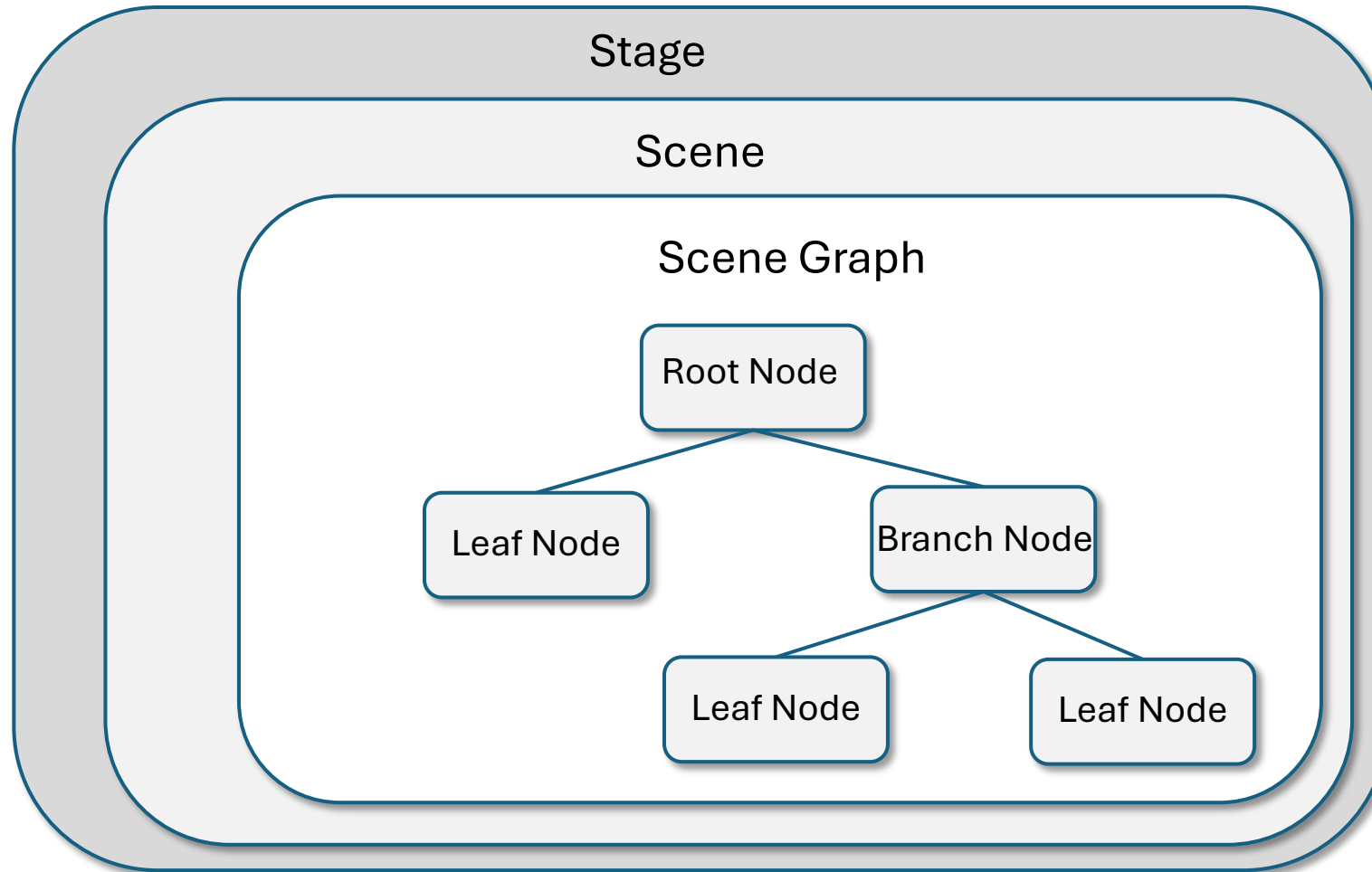
JavaFX

- First Release 2008
- Standalone GUI toolkit meant to replace AWT and Swing
- FX Thread
- Multiplatform classes: eg Audio, Video, 3D Graphics
- Needs Java version with JavaFX package, eg Full JDK!

JavaFX

- Modern Java GUI toolkit
- Structure: Stage -> Scene -> Scene Graph -> Nodes
- Application Class
- public void start
 - Starts a new Application Thread
 - All changes to the scene graph must happen within this thread
 - stage.show(); to show window

JavaFX Structure



Source: https://www.tutorialspoint.com/javafx/javafx_application.htm

JavaFX Sample Code

```
1  import javafx.application.Application;
2  import javafx.scene.Group;
3  import javafx.scene.Scene;
4  import javafx.stage.Stage;
5
6  ▶ public class Example extends Application {
7
8      @Override
9  Ⓢ @ public void start(Stage stage) throws Exception {
10
11      Group root = new Group();
12      Scene scene = new Scene(root, width: 500, height: 500);
13
14      stage.setScene(scene);
15      stage.show();
16
17  }
18 }
```



Development

Requirements

- Java 8+ with JavaFX modules
- Open Object Rexx 5.0.0+
- BSF4ooRexx850 bridge

JavaDrawingHandler (JDOR)

- Class Constructor for custom JFrame
- Implements REXXRedirectingCommandHandler
- Hashmaps for storing variables
- Callback Method handleCommand
 - Iterates through redirected REXX commands
- processCommand
 - Processes redirected REXX commands
 - Switch Statement

Differences 1

JavaDrawingFrame

- JFrame class
- Embedded JPanel
- BufferedImage

JavaFXDrawingFrame

- Scene class
- Pane as Root Node
- Canvas
- Group Nodes
 - shapeGroup
 - shape3DGroup
 - lightGroup

Differences 2

```
243 class JavaFXDrawingFrame extends Scene no usages
244 {
245     /* static definitions */
246     static final private int prefWidth = 500; no usages
247     static final private int prefHeight = 500; no usages
248     static final private boolean prefResizable=false; no usages
249     /* instance definitions */
250     boolean bDebug= false; // true no usages
251     Pane root = new Pane(); no usages
252     Canvas canvas = new Canvas(); no usages
253     Group shapeGroup = new Group(); no usages
254     Group shape3DGroup = new Group(); no usages
255     Group lightGroup = new Group(); no usages
256
257     int currWidth = prefWidth; no usages
258     int currHeight = prefHeight; no usages
259     boolean currFrameVisible = true; // cf. command "winFrame [.true|.false]" no usages
260     boolean currFrameResizable = prefResizable; no usages
261
262     /** Constructor.
263      * @param canvas to set size and display on scene
264      */
265     public JavaFXDrawingFrame(Canvas can) no usages
266     {
267         super(new Group());
268         this.setRoot(root);
269         currWidth = (int) can.getWidth();
270         currHeight = (int) can.getHeight();
271         root.setPrefSize(currWidth,currHeight);
272         canvas = can;
273         root.getChildren().add(canvas);
274         root.getChildren().add(shapeGroup);
275         root.getChildren().add(shape3DGroup);
276         root.getChildren().add(lightGroup);
277     }
278
279     /** Resizes frame.
```

```
234 class JavaDrawingFrame extends JFrame no usages
235 {
236     /* static definitions */
237     static final private int prefWidth = 500; no usages
238     static final private int prefHeight = 500; no usages
239     static final private boolean prefResizable=false; no usages
240     /* instance definitions */
241     boolean bDebug= false; // true no usages
242     Surface sf = null; no usages
243     int currWidth = prefWidth; no usages
244     int currHeight = prefHeight; no usages
245     boolean currFrameVisible = true; // cf. command "winFrame [.true|.false]" no usages
246     boolean currFrameResizable = prefResizable; no usages
247     BufferedImage bufImage = null; no usages
248     /** Constructor.
249      * @param img the BufferedImage to display*/
250     public JavaDrawingFrame(BufferedImage img) no usages
251     {
252         bufImage=img;
253         initUI();
254     }
255
256     /** Set up the UI using predefined width and height for content pane. */
257     private void initUI() 1 usage
258     {
259         sf=new Surface();
260         add(sf);
261         setTitle("JavaDrawingFrame");
262         // needs to be followed by pack() to take effect immediately
263         currWidth=bufImage.getWidth();
264         currHeight=bufImage.getHeight();
265         getContentPane().setPreferredSize(new Dimension(currWidth,currHeight));
266         setResizable(currFrameResizable);
267         pack();
268         setLocationRelativeTo(null);
269         setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
270     }
271 }
```

JavaFXDrawingHandler (JDORFX) 1

JavaFXDrawingHandler

- Extends Application Class
- New Thread to Launch FX Thread
- Start Method
 - Stage that embeds custom Scene
 - New Runnable for GUI updates every 10 milliseconds

JavaFXDrawingHandler (JDORFX) 2

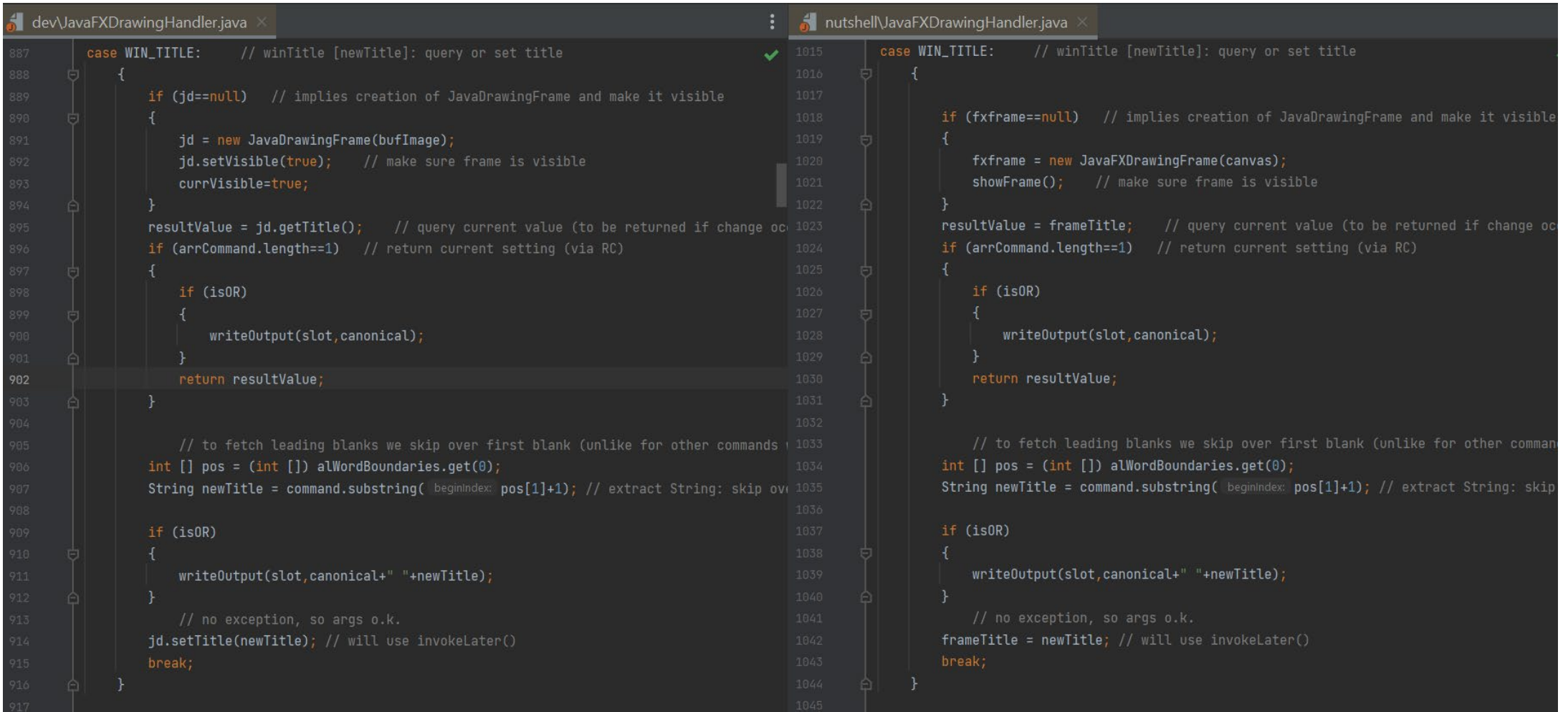
```
5176      @Override
5177      public void start(Stage stage) throws Exception {
5178
5179          // prevents closing of Thread when stage is hidden
5180          Platform.setImplicitExit(false);
5181
5182          // close application thread when window is closed
5183          stage.setOnCloseRequest(event -> {
5184              Platform.exit();
5185          });
5186
5187      >      Thread thread = new Thread(run() -> {
5190      >          Runnable updater = run() -> {
5193
5194          // check if a new scene is available and an update should be executed
5195          if (!deque.isEmpty() && changeScene && fxWinUpdate) {
5196
5197              // set first element of deque to stage
5198              Scene scene = deque.getFirst();
5199              stage.setScene(scene);
5200              stage.setTitle(frameTitle);
5201              stage.sizeToScene();          // in case of resize
```

JavaFXDrawingHandler (JDORFX) 3

JavaFXDrawingHandler

- ConcurrentLinkedDeque for communication between threads
 - Main Thread puts in changed Scene object
 - FX Thread reads Scene object
- Synchronized methods for changing Stage properties
 - Window show
 - Window Location
 - Window Resizable

Comparison of command processing



The image displays a side-by-side comparison of the `case WIN_TITLE:` block in `JavaFXDrawingHandler.java` across two different environments: a development environment (left) and a Nutshell environment (right).

Left Panel (dev\JavaFXDrawingHandler.java):

```
887 case WIN_TITLE: // winTitle [newTitle]: query or set title
888 {
889     if (jd==null) // implies creation of JavaDrawingFrame and make it visible
890     {
891         jd = new JavaDrawingFrame(bufImage);
892         jd.setVisible(true); // make sure frame is visible
893         currVisible=true;
894     }
895     resultValue = jd.getTitle(); // query current value (to be returned if change occurs)
896     if (arrCommand.length==1) // return current setting (via RC)
897     {
898         if (isOR)
899         {
900             writeOutput(slot,canonical);
901         }
902         return resultValue;
903     }
904
905     // to fetch leading blanks we skip over first blank (unlike for other commands)
906     int [] pos = (int []) alWordBoundaries.get(0);
907     String newTitle = command.substring( beginindex: pos[1]+1); // extract String: skip over first blank
908
909     if (isOR)
910     {
911         writeOutput(slot,canonical+" "+newTitle);
912     }
913
914     // no exception, so args o.k.
915     jd.setTitle(newTitle); // will use invokeLater()
916     break;
917 }
```

Right Panel (nutshell\JavaFXDrawingHandler.java):

```
1015 case WIN_TITLE: // winTitle [newTitle]: query or set title
1016 {
1017
1018     if (fxframe==null) // implies creation of JavaDrawingFrame and make it visible
1019     {
1020         fxframe = new JavaFXDrawingFrame(canvas);
1021         showFrame(); // make sure frame is visible
1022     }
1023     resultValue = frameTitle; // query current value (to be returned if change occurs)
1024     if (arrCommand.length==1) // return current setting (via RC)
1025     {
1026         if (isOR)
1027         {
1028             writeOutput(slot,canonical);
1029         }
1030         return resultValue;
1031     }
1032
1033     // to fetch leading blanks we skip over first blank (unlike for other commands)
1034     int [] pos = (int []) alWordBoundaries.get(0);
1035     String newTitle = command.substring( beginindex: pos[1]+1); // extract String: skip over first blank
1036
1037     if (isOR)
1038     {
1039         writeOutput(slot,canonical+" "+newTitle);
1040     }
1041
1042     // no exception, so args o.k.
1043     frameTitle = newTitle; // will use invokeLater()
1044     break;
1045 }
```

The comparison highlights differences in variable names (e.g., `jd` vs `fxframe`, `currVisible` vs `showFrame`, `jd.setTitle` vs `frameTitle`) and the handling of the `isOR` flag between the two environments.

Challenges: Stroke Property

Setting Stroke Properties

JavaDrawingHandler

- awt Stroke class

JavaFXDrawingHandler

- JavaFX Stroke: only color value
- Storing properties in HashMap
 - Color
 - Width
 - LineJoin
 - DashArray

Challenges: PathAppend

Appending a named Shape to a named Path

JavaDrawingHandler

- Append Shape to Path

JavaFXDrawingHandler

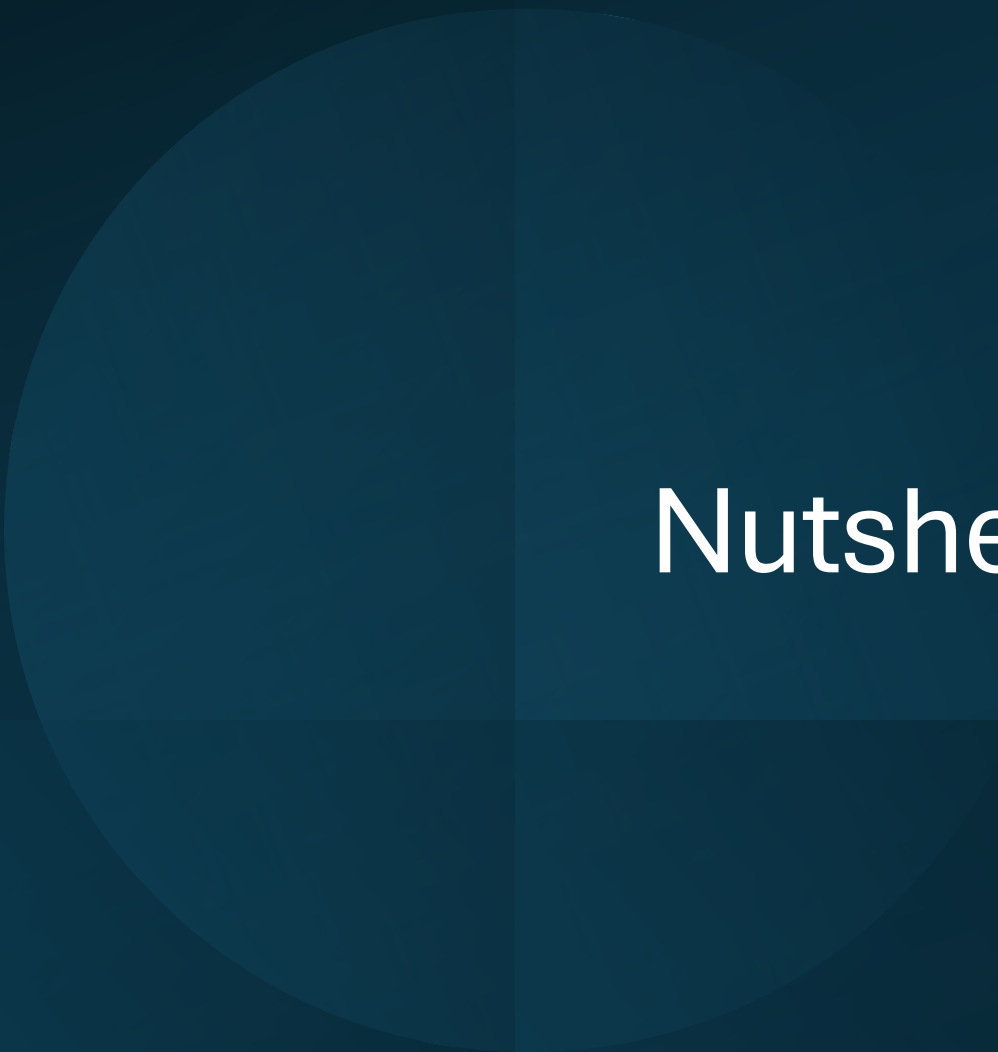
- Turn named Shape into Path
- Iterate through PathElements
- Append new PathElements to named Path
- (Errors when joining shapes)

JDORFX Environment

```
36  /** Creates a new command handler that will serve the environment by the name
37      * of the optional environmentName argument.
38      * @param environmentName the address environment name to use, if omitted defaults to "JDORFX"
39  */
40  ::routine addJdorFXHandler    public
41      use strict arg environmentName="JDORFX"
42
43      call BsfcCommandHandler "add", -
44      environmentName, -
45      .bsf~new("org.oorexx.handlers.jdorfx.JavaFXDrawingHandler")
46
47  ::requires "BSF.CLS"    -- get ooRexx-Java bridge
```

Address JDORX in Rexx Programs

```
25  -- create JDORFX handler
26  -- load and add the Java Rexx command handler, using default name: JDORFX
27  call addJdorFXHandler
28  -- set default environment to JDORFX
29  address jdorfx
30
31  -- get ooRexx-Java bridge, contains JDORFX Rexx command handler
32  ::requires "jdorfx.CLS"
33
```

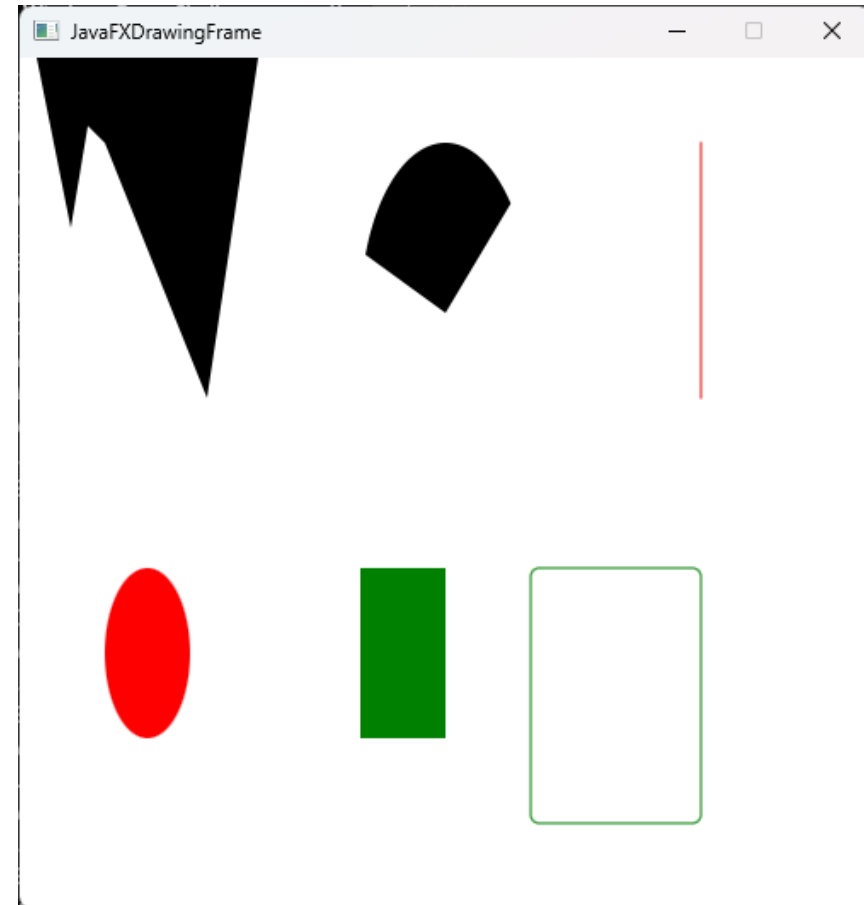
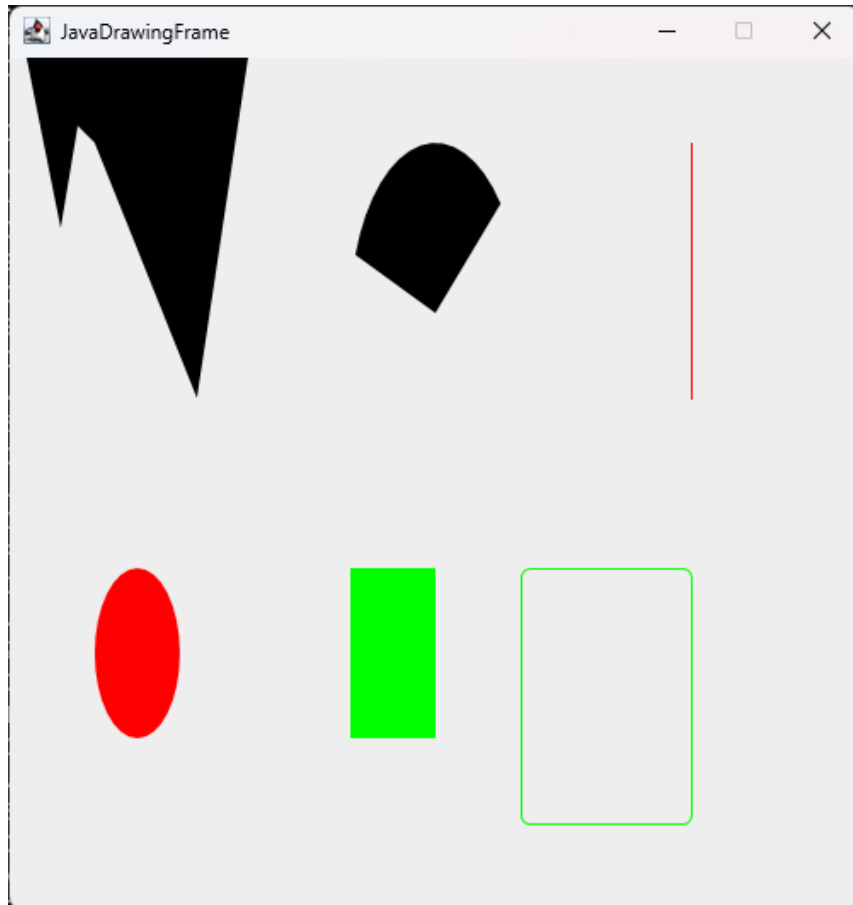


Nutshell Examples

Nutshell: JDOR vs JDORFX Drawing

```
31 say "Create a new scene and draw on canvas"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- draw filled polygon with nPoints taken from xPoints-array and yPoints-array
35 fillpolygon "(10,30,40,50,110,140)" "(0,100,40,50,200,0)" 6
36 -- move current x and y points
37 moveto 200 50
38 -- draw filled arc with width height start angle and arc angle
39 fillarc 100 200 40 120
40 -- move current x and y points
41 moveto 400 50
42 -- set draw and fill color to red
43 color red
44 -- draw line from current x and y points to new x and y
45 drawline 400 200
46 -- move current x and y points
47 moveto 50 300
48 -- draw filled oval with width and height
49 filloval 50 100
50 -- set draw and fill color to green
51 color green
52 -- move current x and y points
53 moveto 200 300
54 -- draw filled rectangle with width and height
55 fillrect 50 100
56 -- move current x and y points
57 moveto 300 300
58 -- draw round rectangle with width height arcwidth and archheight
59 drawroundrect 100 150 10 10
60 -- show created window
61 winshow
62 say
63 say "Sleep for 5 seconds and end program..."
64 parse pull
65
66 -- get ooRexx-Java bridge, contains JDORFX Rexx command handler
67 ::requires "jdorfx.CLS"
```

Nutshell: JDOR vs JDORFX Drawing

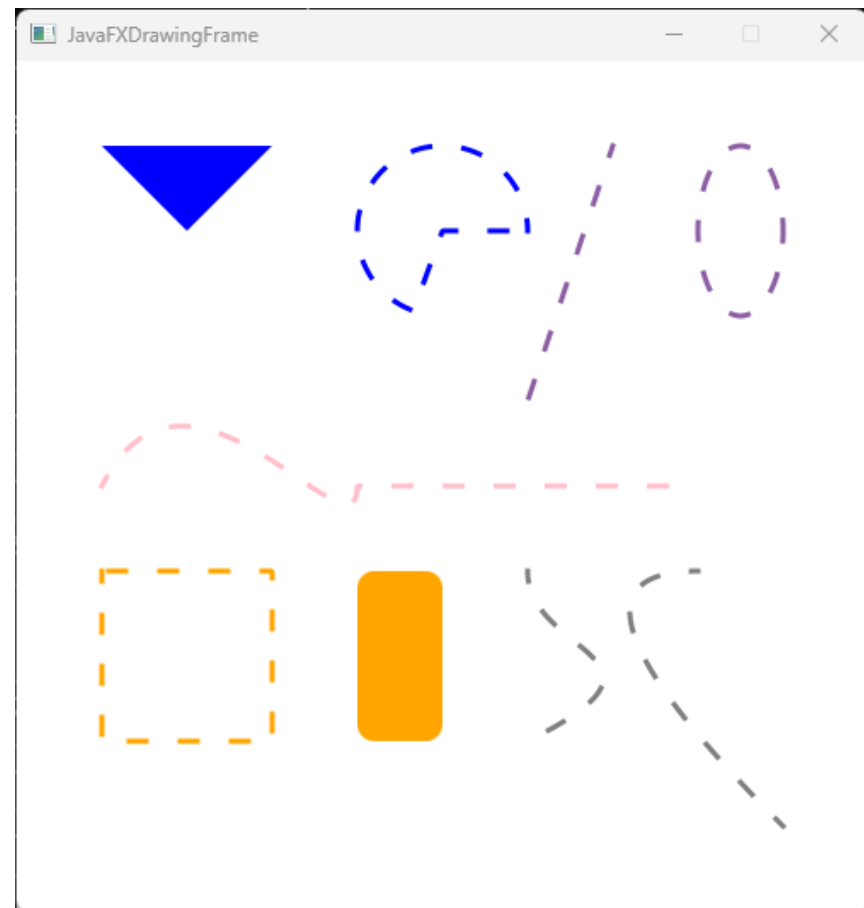
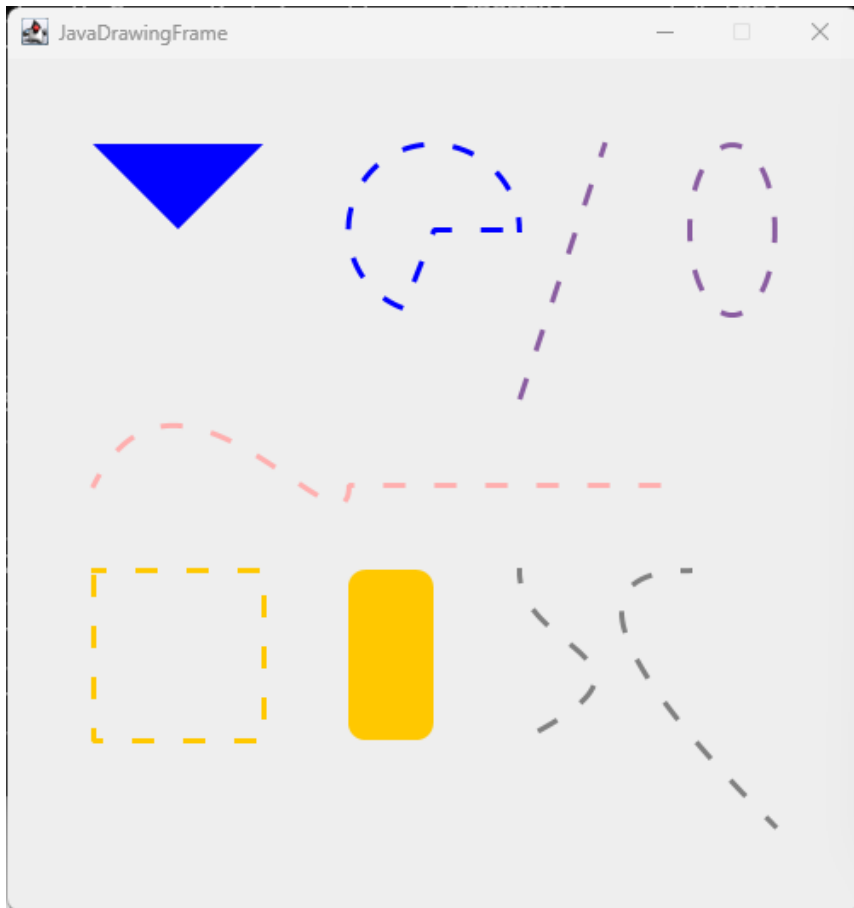


Nutshell: JDOR vs JDORFX 2D Shapes

```
31 say "Create a scene and add 2D shapes"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- set draw and fill color to blue
35 color blue
36 -- set stroke with width=3 cap=2 join=2 miterLimit=10 dashArray=(10,20) dashOffset=4
37 stroke myStroke 3 2 2 10 "(10,20)" 4
38 stroke myStroke
39 -- draw filled polygon with nPoints taken from xPoints-array and yPoints-array
40 shape myPoly polygon "(50,150,100)" "(50,50,100)" 3
41 fillshape myPoly
42 -- draw arc with x y width height start extend type
43 shape myArc arc 200 50 100 100 0 250 2
44 drawshape myArc
45 -- set draw and fill color to R=0.5 G=0.3 B=0.6 Alpha=0.9
46 color myColor 0.5 0.3 0.6 0.9
47 color myColor
48 -- draw line from x y to newX and newY
49 shape myLine line 350 50 300 200
50 drawshape myLine
51 -- draw ellipse with x y width height
52 shape myEllipse ellipse 400 50 50 100
53 drawshape myEllipse
54 -- set draw and fill color to orange
55 color orange
56 -- draw rectangle with x y width height
57 shape rec rectangle2d 50 300 100 100
58 drawshape rec
59 -- draw filled round rectangle with x y width height arcWidth arcHeight
60 shape myRoundrect roundrect 200 300 50 100 20 20
61 fillshape myRoundrect
62 -- set draw and fill color to gray
63 color gray
64 -- draw cubic curve with x1 y1 ctrlx1 ctrlx1 ctrlx2 ctrlx2 x2 y2
65 shape myCubic cubic 300 300 300 350 400 350 300 400
66 drawshape myCubic
67 -- draw cubic curve with x1 y1 ctrlx ctrlx x2 y2
```

```
68 shape myQuadCurve quadcurve 450 450 300 300 400 300
69 drawshape myQuadCurve
70 -- set draw and fill color to pink
71 color pink
72 -- create a path and add elements
73 shape myPath path 1
74 pathmoveto myPath 50 250
75 -- add curve to path with ctrlx1 ctrlx1 ctrlx2 ctrlx2 x y
76 pathcurveto myPath 100 150 200 300 200 250
77 pathlineto myPath 400 250
78 --draw path
79 drawshape myPath
80 -- show created window
81 winshow
82 say
83 say "Press enter to end program..."
84 parse pull
```

Nutshell: JDOR vs JDORFX 2D Shapes

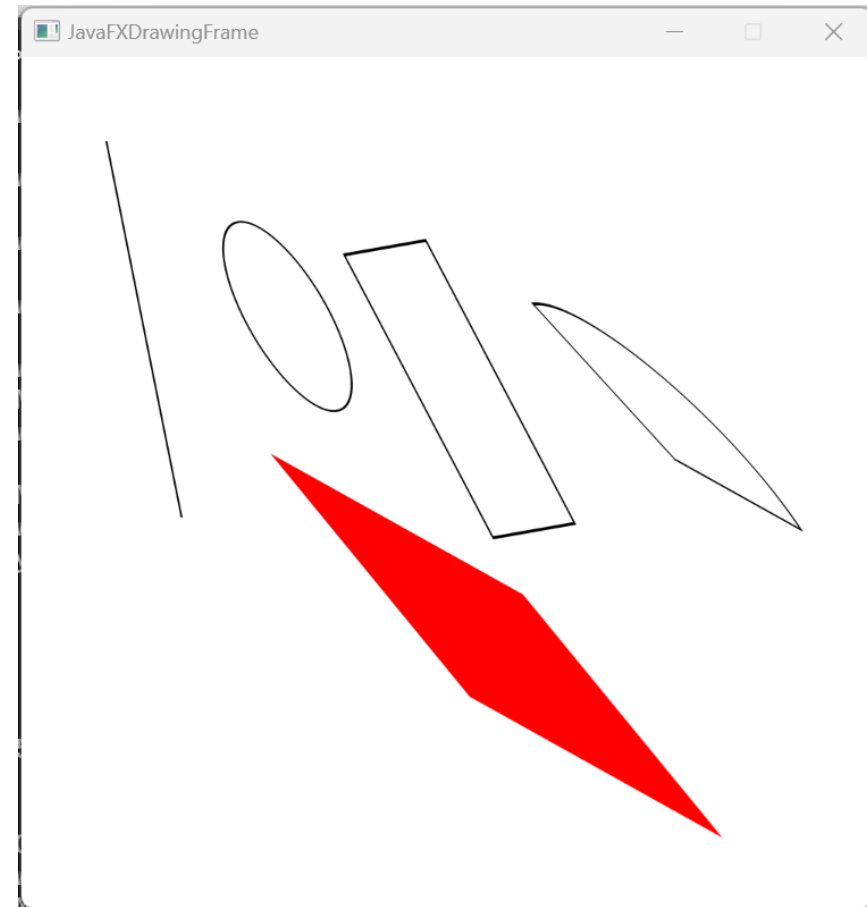
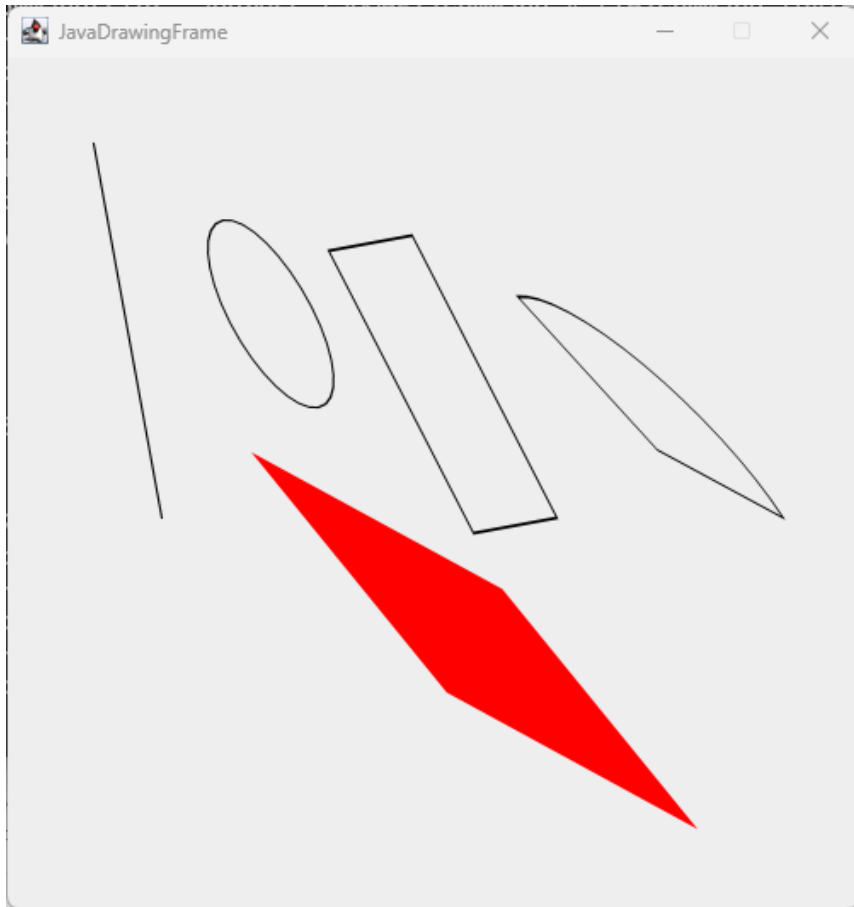


Nutshell: JDOR vs JDORFX 2D Transform

```
31 say "Create a scene and add and transform 2D shapes"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- transform following shapes and drawing elements with
35 -- translateX translateY scaleX scaleY shearX shearY
36 transform 50 50 1.1 1.1 0.2 0.2
37 -- move current x and y points
38 moveto 0 0
39 -- draw line from current x and y points to new x and y
40 drawline 0 200
41 -- rotate following shapes and drawing elements by angle
42 rotate (-20)
43 -- draw ellipse with x y width height
44 shape myEllipse ellipse 25 50 50 100
45 drawshape myEllipse
46 -- scale following shapes and drawing elements by x=1.5
47 scale 1 1.5
48 -- draw rectangle with x y width height
49 shape myRect rectangle2d 100 50 50 100
50 drawshape myRect
51 -- shear following shapes and drawing elements by x=0.2 y=0.6
52 shear 0.2 0.6
53 -- draw arc with x y width height start length type
54 shape myArc arc 175 (-50) 100 100 0 120 2
55 drawshape myArc
56 -- set draw and fill color to red
57 color red
58 -- create a path myPath and add elements
59 shape myPath path 1
60 pathmoveto myPath 300 50
61 pathlineto myPath 400 50
62 pathlineto myPath 400 100
63 pathlineto myPath 300 100
64 pathclose myPath
65 -- create transformation with name myTransform
66 transform myTransform (-350) 50 1 1.5 0.5 0
67 -- use transformation myTransform on path myPath
```

```
68 pathtransform myPath myTransform
69 --draw path
70 fillshape myPath
71 -- show created window
72 winshow
73 say
74 say "Press enter end program..."
75 parse pull
```

Nutshell: JDOR vs JDORFX 2D Transform



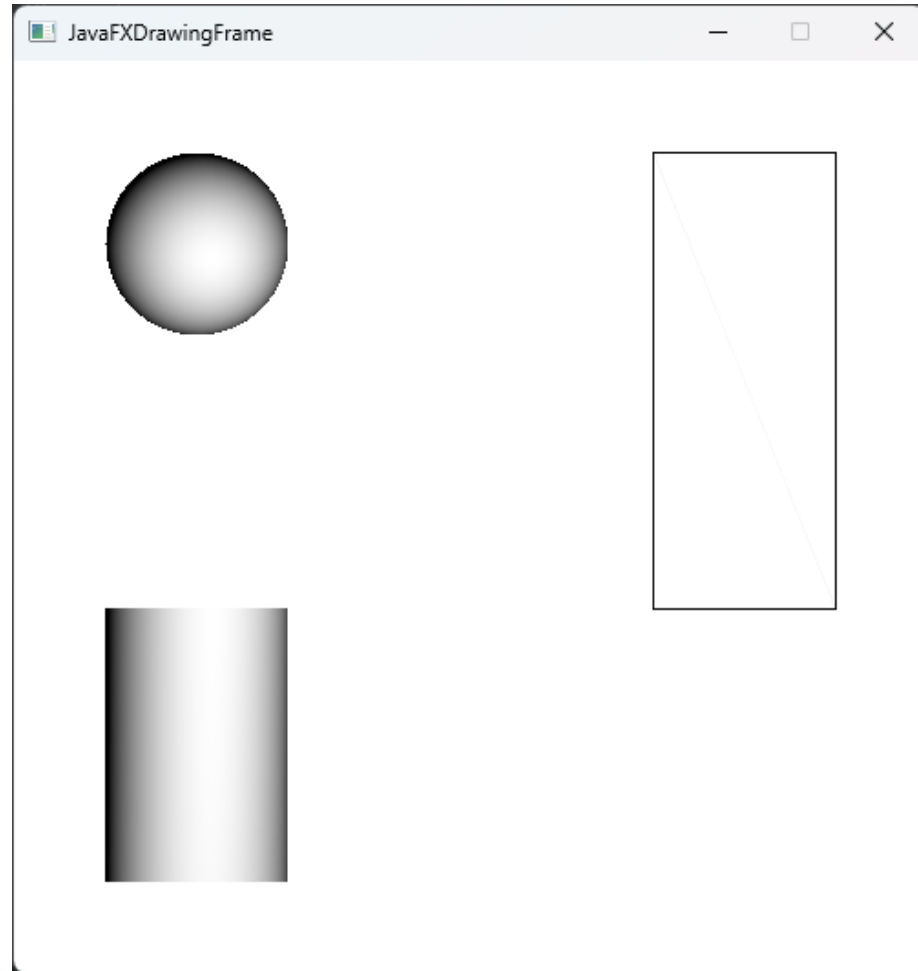
Nutshell: 3D Shapes

- 3DShape classes
 - Box
 - Cylinder
 - Sphere
- Draw3dShape / DrawShape3D: wireframe model
- Fill3dShape / FillShape3D: model with filled vertecies

Nutshell: 3D Shapes

```
31 say "Create a scene and add 3D shapes"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- create new sphere with the name "mySphere" at location x=100, y=100, z=0
35 -- and a radius=50
36 shape3d mySphere sphere 100 100 0 50
37 -- create new box with the name "myBox" at location x=400, y=175, z=0
38 -- and size: width=100, height=250, depth=200
39 shape3d myBox box 400 175 0 100 250 200
40 -- create new cylinder with the name "myCylinder" at location x=100, y=350, z=0
41 -- and size: radius=50, height=150
42 shape3d myCylinder cylinder 100 350 0 50 150
43 -- place mySphere onto scene with filled vertices; also: fillshape3d
44 fill3dshape mySphere
45 -- place myBox onto scene as line / wireframe model; also: drawshape3d
46 draw3dshape myBox
47 -- place myCylinder onto scene with filled vertices; also: fill3dshape
48 fillshape3d myCylinder
49 -- show created window
50 winshow
51 say
52 say "Press Enter to end program..."
53 parse pull
```

Nutshell: 3D Shapes



Nutshell: Camera

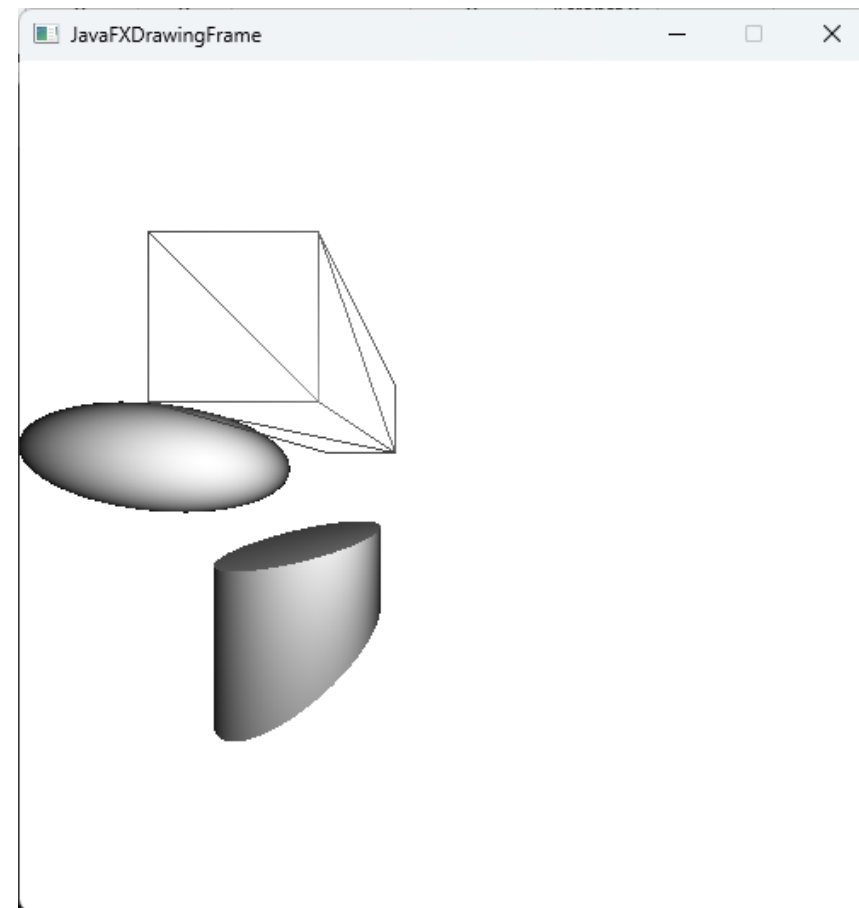
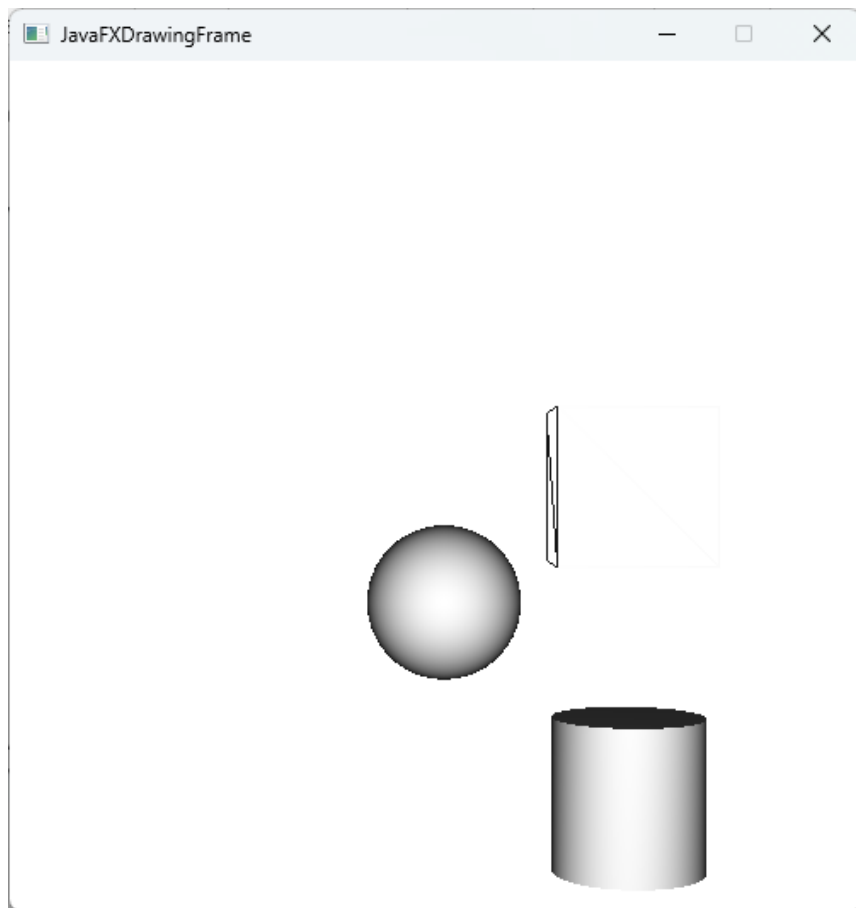
- Parallel Camera
 - Default Scene Camera
 - No Perspective Distortion
- Perspective Camera
 - Perspective Distortion
 - FieldOfView: defines the viewing volume for a perspective projection

Nutshell: Camera

```
31 say "Create a scene with the default parallel camera"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- show created window
35 winshow
36 -- create new sphere with name "mySphere" at location x=50, y=200, z=0 and a radius=50
37 shape3d mySphere sphere 205 225 0 50
38 -- create new box with name "myBox" at location x=200, y=200, z=0
39 -- and size: width=100, height=100, depth=100
40 shape3d myBox box 325 150 0 100 100 100
41 -- create new cylinder with name "myCylinder" at location x=350, y=200, z=0
42 -- and size: radius=50, height=200
43 shape3d myCylinder cylinder 325 350 0 50 100
44 -- place mySphere onto scene with filled vertices; also: fillshape3d
45 fill3dshape mySphere
46 -- place myBox onto scene as line / wireframe model; also: drawshape3d
47 draw3dshape myBox
48 -- place myCylinder onto scene with filled vertices; also: fillshape3d
49 fill3dshape myCylinder
50 say
51 say "Sleep for 5 seconds and then create a new perspective camera"
52 say "Location: x=-50 y=-100 z=-100, default fieldOfView=30"
53 parse pull
54 --create perspectiveCamera1
55 camera perspectiveCamera1 perspective (-50) (-100) (-100)
56 -- set view to perspectiveCamera1
57 setcamera perspectiveCamera1
58 say
59 say "Sleep for 5 seconds and then create a new perspective camera"
60 say "Location: x=200 y=0 z=-50, fieldOfView=150"
61 parse pull
62 --create perspectiveCamera2
63 camera perspectiveCamera2 perspective 200 0 (-50) 150
64 -- set view to perspectiveCamera1
65 setcamera perspectiveCamera2
66 say
67 say "Sleep for 5 seconds and then create a new parallel camera"
```

```
68 say "Location: x=0 y=100 z=-200"
69 say "Note: The z variable has no impact on a parallel camera"
70 parse pull
71 --create parallelCamera
72 camera parallelCamera parallel 0 100 (-200)
73 -- set view to parallelCamera
74 setcamera parallelCamera
75 say
76 say "Press Enter to end program..."
77 parse pull
```

Nutshell: Camera



Nutshell: Ambient Light vs Point Light

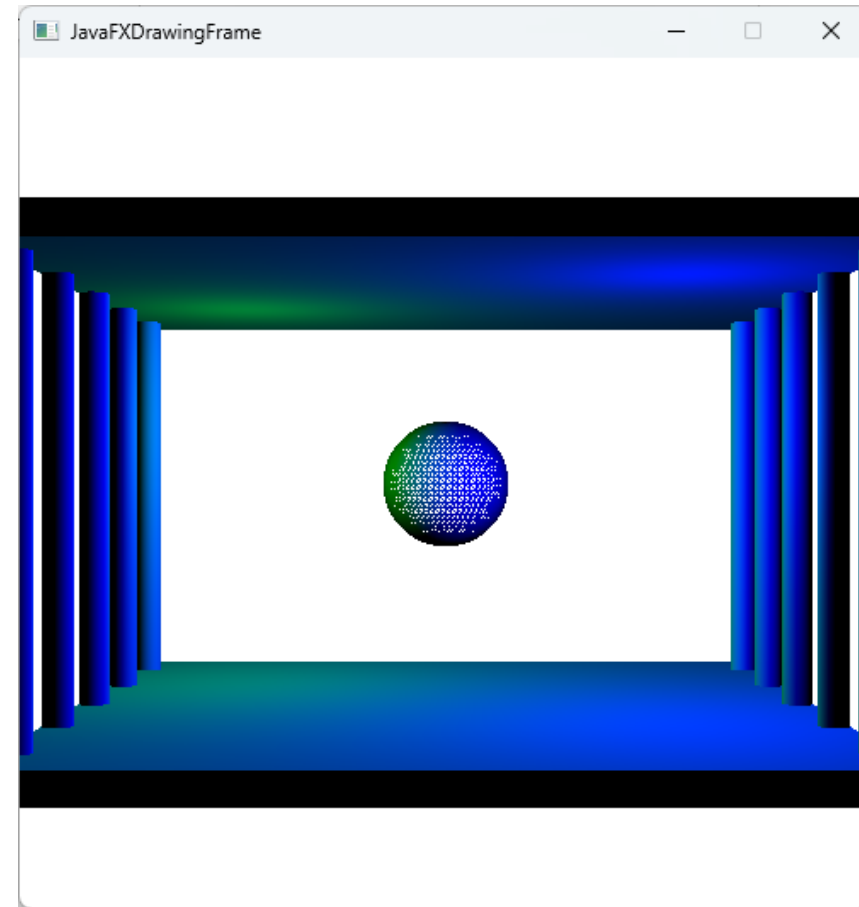
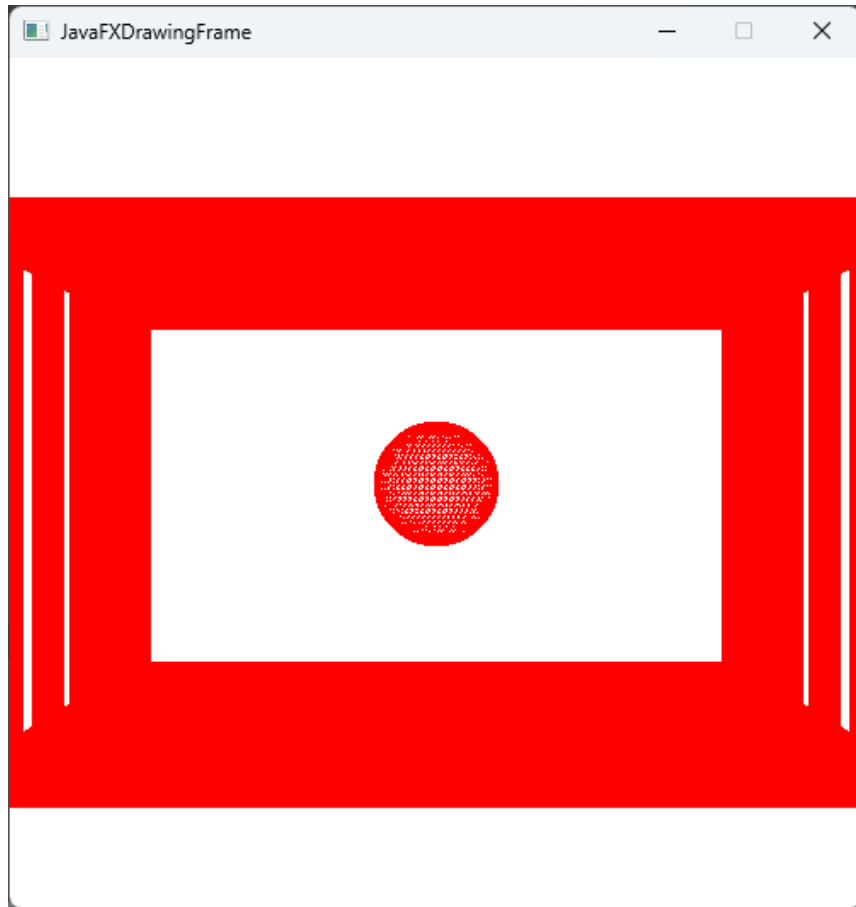
- Ambient Light
 - Light source that appears to come from all directions
- Point Light
 - Radiates light away from its position

Nutshell: Ambient Light vs Point Light

```
31 say "Create a new scene and set ambient light to red"
32 -- creating a new window with size 500 x 500
33 newImage 500 500
34 -- show created window
35 winshow
36 -- create new box shapes
37 shape3d ceiling box 250 110 0 500 20 500
38 shape3d floor box 250 410 0 500 20 500
39 -- place box shapes onto scene
40 fill3dshape ceiling
41 fill3dshape floor
42 -- create new cylinder shapes
43 shape3d leftPillar1 cylinder 10 260 (-200) 10 280
44 shape3d leftPillar2 cylinder 10 260 (-100) 10 280
45 shape3d leftPillar3 cylinder 10 260 0 10 280
46 shape3d leftPillar4 cylinder 10 260 100 10 280
47 shape3d leftPillar5 cylinder 10 260 200 10 280
48 shape3d rightPillar1 cylinder 490 260 (-200) 10 280
49 shape3d rightPillar2 cylinder 490 260 (-100) 10 280
50 shape3d rightPillar3 cylinder 490 260 0 10 280
51 shape3d rightPillar4 cylinder 490 260 100 10 280
52 shape3d rightPillar5 cylinder 490 260 200 10 280
53 -- place cylinder shapes onto scene
54 fill3dshape leftPillar1
55 fill3dshape leftPillar2
56 fill3dshape leftPillar3
57 fill3dshape leftPillar4
58 fill3dshape leftPillar5
59 fill3dshape rightPillar1
60 fill3dshape rightPillar2
61 fill3dshape rightPillar3
62 fill3dshape rightPillar4
63 fill3dshape rightPillar5
64 -- create and add new sphere to scene
65 shape3d mySphere sphere 250 250 200 50
66 draw3dShape mySphere
67 -- create new perspective camera and set scene camera
```

```
68 camera perspectiveCamera perspective 0 0 (-150)
69 setcamera perspectiveCamera
70 -- create and add new ambient light to scene
71 light ambientLight ambient red
72 -- turn ambientLight on
73 setlight ambientLight
74 -- create and add new point lights to scene
75 light blueLight point 400 200 (-100) blue
76 light greenLight point 100 200 100 green
77 say
78 say "Sleep for 5 seconds and turn on point lights"
79 say "Location point light blue: x=400 y=200 z=-100"
80 say "Location point light green: x=100 y=200 z=100"
81 parse pull
82 -- turn point lights on
83 setlight blueLight
84 setlight greenLight
85 say
86 say "Sleep for 5 seconds and turn off ambient light"
87 parse pull
88 -- turn ambientLight off
89 setlight ambientLight turnoff
90 say
91 say "Press Enter to end program..."
92 parse pull
```

Nutshell: Ambient Light vs Point Light



Nutshell: 3D Transformations

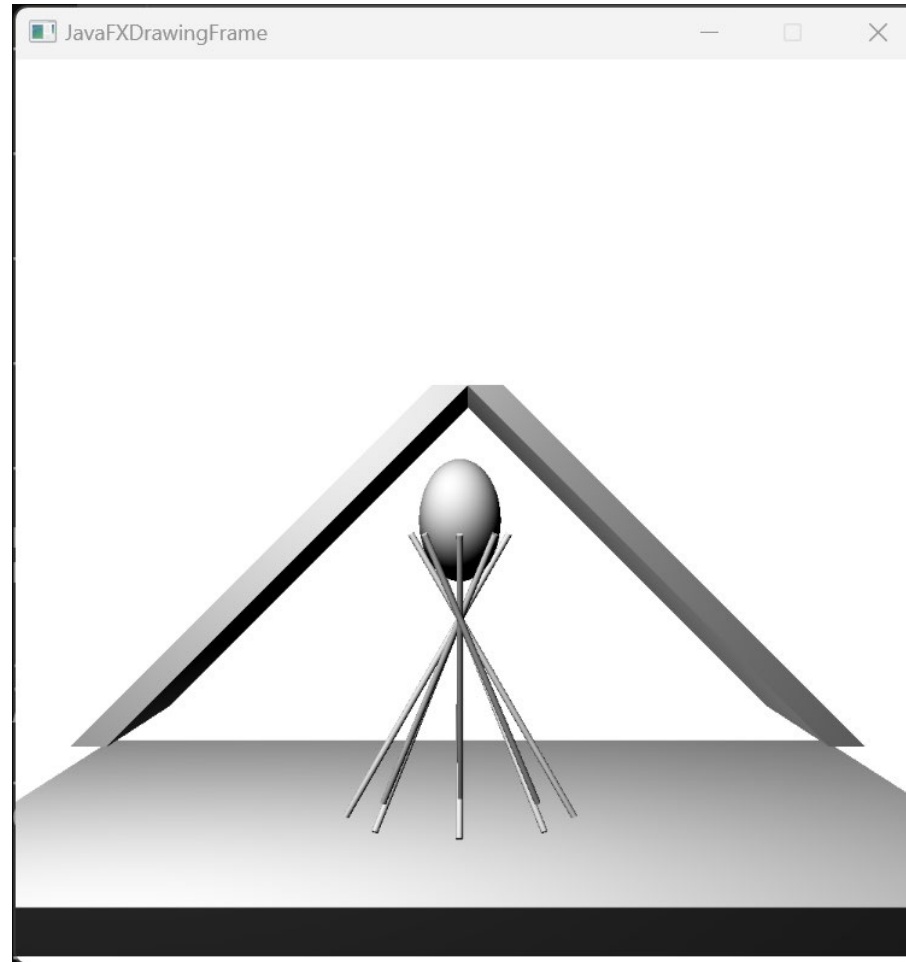
- 3D Transformations
 - Translate
 - Rotate
 - Scale
 - Shear

Nutshell: 3D Transformations

```
31 say "Create a scene and add and transform 3D shapes"
32 -- creating a new window with size 500 x 500
33 newImage 500 500
34 -- create new boxes
35 shape3d box1 box 360 280 0 200 20 200
36 shape3d box2 box 140 280 0 200 20 200
37 shape3d floor box 250 0 (-100) 500 20 500
38 --move box "floor" by x, y, z
39 translate3dshape floor 0 420 0
40 -- shear boxes with shx shy pivotX pivotY
41 shear3dshape box1 1 0 0 0
42 shear3dshape box2 (-1) 0 0 0
43 -- rotate shape 90 degrees on the z axis with its center as pivot point
44 rotate3dshape box1 90 0 0 0 0 1
45 -- rotate shape -90 degrees on the z axis with its center as pivot point
46 rotate3dshape box2 (-90) 0 0 0 0 1
47 -- create new cylinder
48 shape3d mySphere sphere 245 255 0 25
49 --scale the y axis of shape, pivot point is center of shape
50 scale3dshape mySphere 1 1.5 1 0 0 0
51 -- create new cylinder
52 shape3d cylinder1 cylinder 225 350 0 2 200
53 --rotate shape 45 degrees on the y axis, pivot point is 20px left of shape center
54 rotate3dshape cylinder1 45 20 0 0 0 1
55 --rotate shape 30 degrees on the z axis with its center as pivot point
56 rotate3dshape cylinder1 30 0 0 0 0 1
57 -- create new cylinder and apply rotation
58 shape3d cylinder2 cylinder 225 350 0 2 200
59 rotate3dshape cylinder2 90 20 0 0 0 1
60 rotate3dshape cylinder2 30 0 0 0 0 1
61 -- create new cylinder and apply rotation
62 shape3d cylinder3 cylinder 225 350 0 2 200
63 rotate3dshape cylinder3 135 20 0 0 0 1
64 rotate3dshape cylinder3 30 0 0 0 0 1
65 -- create new cylinder and apply rotation
66 shape3d cylinder4 cylinder 225 350 0 2 200
67 rotate3dshape cylinder4 180 20 0 0 0 1
```

```
68 rotate3dshape cylinder4 30 0 0 0 0 1
69 -- create new cylinder and apply rotation
70 shape3d cylinder5 cylinder 225 350 0 2 200
71 rotate3dshape cylinder5 225 20 0 0 0 1
72 rotate3dshape cylinder5 30 0 0 0 0 1
73 -- create new cylinder and apply rotation
74 shape3d cylinder6 cylinder 225 350 0 2 200
75 rotate3dshape cylinder6 270 20 0 0 0 1
76 rotate3dshape cylinder6 30 0 0 0 0 1
77 -- create new cylinder and apply rotation
78 shape3d cylinder7 cylinder 225 350 0 2 200
79 rotate3dshape cylinder7 315 20 0 0 0 1
80 rotate3dshape cylinder7 30 0 0 0 0 1
81 -- create new cylinder and apply rotation
82 shape3d cylinder8 cylinder 225 350 0 2 200
83 rotate3dshape cylinder8 360 20 0 0 0 1
84 rotate3dshape cylinder8 30 0 0 0 0 1
85 -- add shapes to scene
86 -- the order of placing shapes onto scene matters
87 -- later added shapes will cover previous ones, regardless of its distance to the camera
88 fillshape3d floor
89 fillshape3d box1
90 fillshape3d box2
91 fillshape3d cylinder4
92 fillshape3d cylinder5
93 fillshape3d cylinder6
94 fillshape3d cylinder7
95 fillshape3d cylinder8
96 fillshape3d mySphere
97 fillshape3d cylinder1
98 fillshape3d cylinder2
99 fillshape3d cylinder3
100 -- create new perspective camera and point light and add to scene
101 camera perspectiveCamera perspective 0 0 (-100)
102 setcamera perspectiveCamera
103 light myLight point 100 50 (-400) white
104 setlight myLight
```

Nutshell: 3D Transformations



Nutshell: Materials

- Maps

- BumpMap
- DiffuseMap
- SelfIlluminationMap
- SpecularMap
 - Specular Power: level of smoothness, The smaller the value, the narrower the reflections and the smoother the surface appears.

- Color

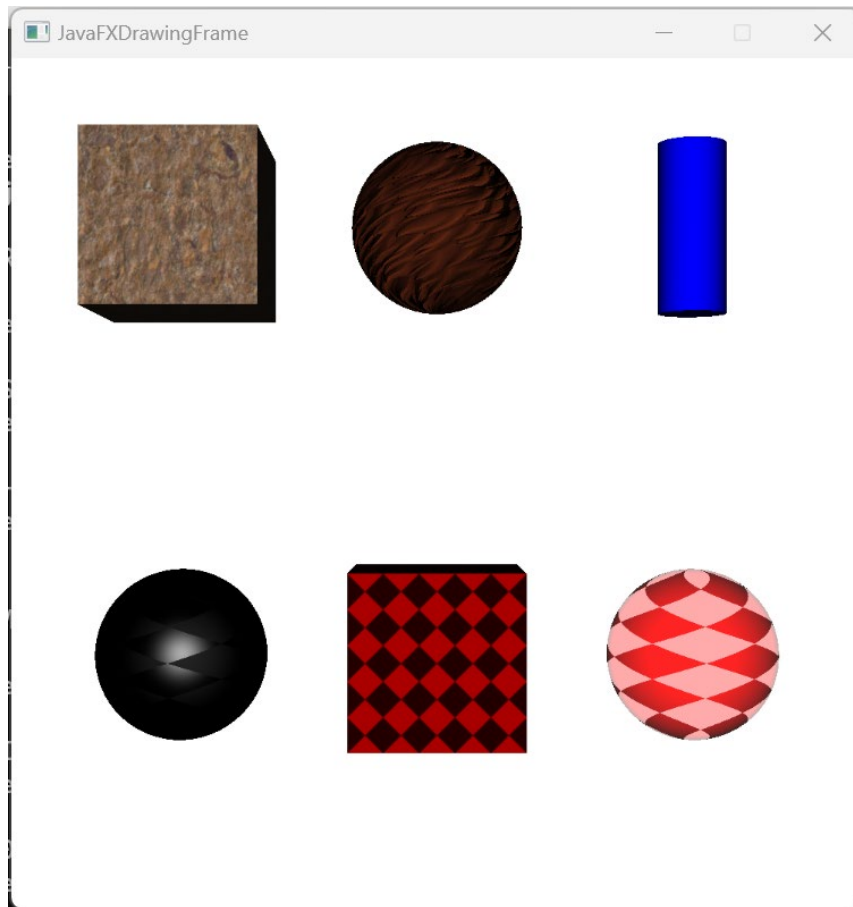
- DiffuseColor
- SpecularColor

Nutshell: Materials

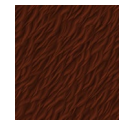
```
39 say "Create a new scene with 3D shapes and materials"
40 -- creating a new window with size 500 x 500
41 newimage 500 500
42 -- create different shapes
43 shape3d rockBox box 100 100 0 100 100 100
44 shape3d furSphere sphere 250 100 0 50
45 shape3d colCylinder cylinder 400 100 0 20 100
46 shape3d tileSphere1 sphere 100 350 0 50
47 shape3d tileBox box 250 350 0 100 100 100
48 shape3d tileSphere2 sphere 400 350 0 50
49 -- add shapes to scene
50 fill3dshape rockBox
51 fill3dshape furSphere
52 fill3dshape colCylinder
53 fill3dshape tileSphere1
54 fill3dshape tileBox
55 fill3dshape tileSphere2
56 -- create new material named rock
57 material rock
58 -- add image with pathname "rock_diffuse.jpg" as diffuse map to material rock
59 materialmap rock diffuse "rock_diffuse.jpg"
60 -- add material rock to rockBox
61 setmaterial rockBox rock
62 --create new material fur
63 material fur
64 -- add image with pathname "fur_bump.jpg" as bump map to material fur
65 materialmap fur bump "fur_bump.jpg"
66 -- add image with pathname "fur_diffuse.jpg" as diffuse map to material rock
67 materialmap fur diffuse "fur_diffuse.jpg"
68 -- add material fur to furSphere
69 setmaterial furSphere fur
70 -- create new material colBlue and set color to blue
71 material colBlue blue
72 -- add material colBlue to colCylinder
73 setmaterial colCylinder colBlue
74 -- create new material tileDiff and set color to red
75 material tileDiff red
```

```
76 -- add image with pathname "tiles.jpg" as diffuse map to material tileDiff
77 materialmap tileDiff diffuse "tiles.jpg"
78 -- add material tileDiff to tileBox
79 setmaterial tileBox tileDiff
80 -- create new material tileSpec and set color to black
81 material tileSpec black
82 -- set material specularColor to white with specularPower of 10
83 materialcolor tileSpec specular white 10
84 -- add image with pathname "tiles.jpg" as specular map to material rock
85 materialmap tileSpec specular "tiles.jpg"
86 -- add material tileSpec to tileSphere1
87 setmaterial tileSphere1 tileSpec
88 -- create new material tileIll and set color to red
89 material tileIll red
90 -- add image with pathname "tiles.jpg" as selfillumination map to material tileIll
91 materialmap tileIll selfillumination "tiles.jpg"
92 -- add material tileIll to tileSphere2
93 setmaterial tileSphere2 tileIll
94 -- create perspective camera and add to scene
95 camera camera perspective 0 0 0
96 setcamera camera
97 -- show created window
98 winshow
99 say
100 say "Press Enter to end program..."
101 parse pull
```

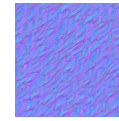
Nutshell: Materials



DiffuseMap



DiffuseMap



BumpMap



DiffuseColor



SpecularMap, DiffuseMap, SelfIlluminationMap

"Texture Images" by <https://3dtextures.me/> are licensed under "CC BY 1.0"
<https://creativecommons.org/publicdomain/zero/1.0/>

Nutshell: Unedited Maps

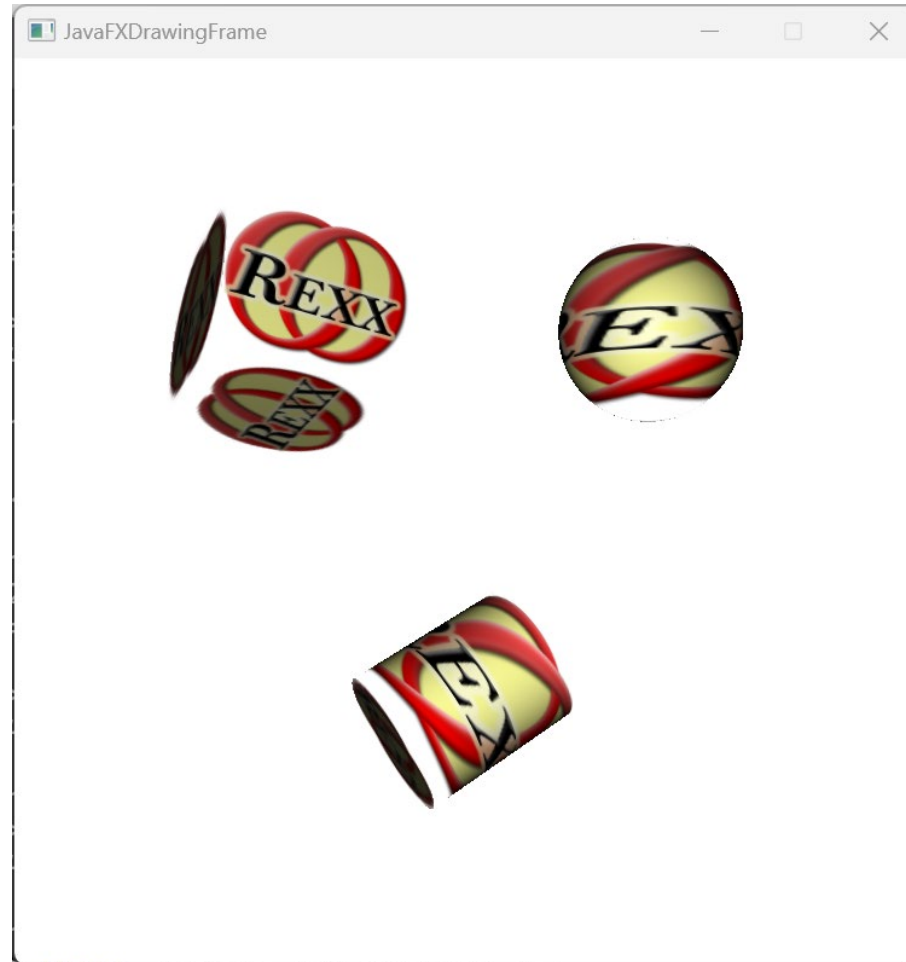
- Problems:
 - Distortion of Image when wrapped
 - Transparent Pixels of PNGs



Nutshell: 3D Transformations

```
31 say "Create a new scene with 3D shapes and png maps"
32 -- creating a new window with size 500 x 500
33 newimage 500 500
34 -- create different 3D shapes
35 shape3d myBox box 150 150 0 100 100 100
36 shape3d myCylinder cylinder 250 350 0 40 100
37 shape3d mySphere sphere 350 150 0 50
38 -- add shapes to scene
39 fill3dshape myBox
40 fill3dshape mySphere
41 fill3dshape myCylinder
42 -- rotate myBox and myCylinder
43 rotate3dshape myBox (-20) 0 0 0 1 0 0
44 rotate3dshape myBox 60 0 0 0 0 1 0
45 rotate3dshape myCylinder 60 0 0 0 0 0 1
46 rotate3dshape myCylinder (-20) 0 0 0 1 0 0
47 -- create perspective camera and add to scene
48 camera camera perspective 0 0 0 50
49 setcamera camera
50 -- create new material myMaterial and add PNG as diffuse map
51 material myMaterial
52 materialmap myMaterial diffuse "oorexx_256.png"
53 -- add myMaterial to shapes
54 setmaterial myBox myMaterial
55 setmaterial mySphere myMaterial
56 setmaterial myCylinder myMaterial
57 -- show created window
58 winshow
59 say
60 say "Press Enter to end program..."
61 parse pull
```

Nutshell: Unedited Maps



Nutshell: Edited Maps

- Edit Map properties:
 - Add Width
 - Add Height
 - Rotate Image
 - Add Color

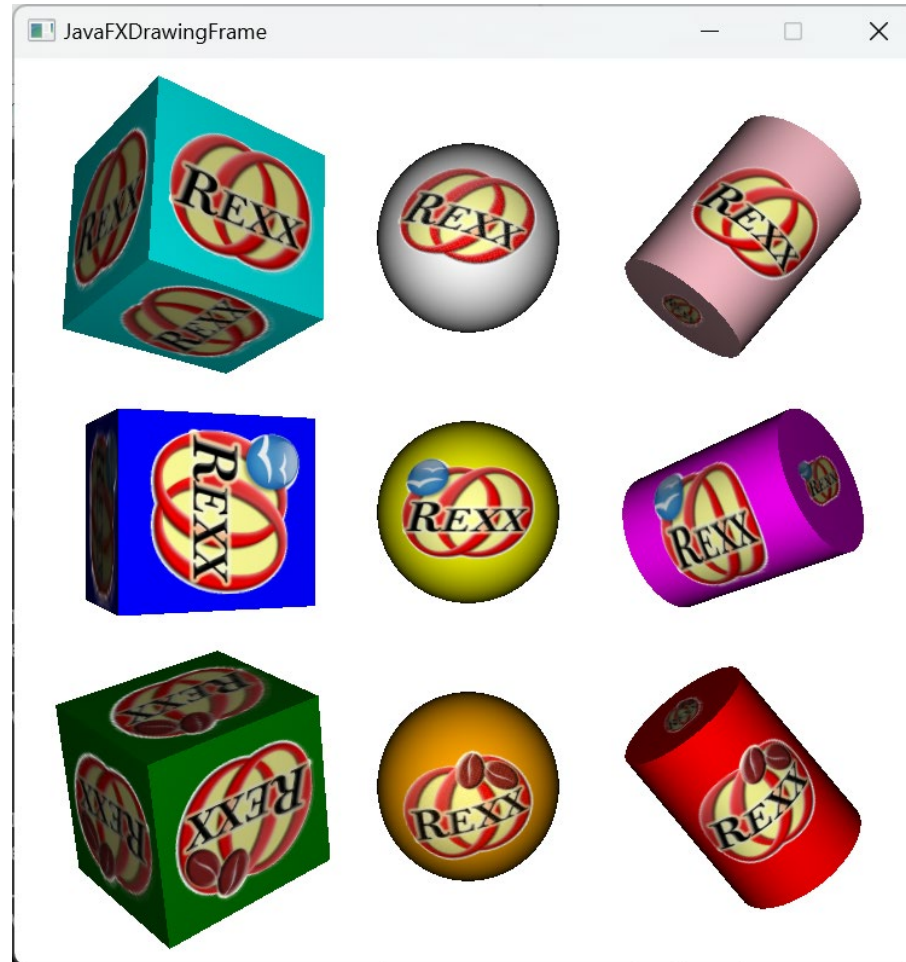
Nutshell: Edited Maps

```
31 say "Create a new scene with 3D shapes and edited png maps"
32 -- creating a new window with size 500 x 500
33 newImage 500 500
34 -- create different 3D shapes
35 shape3d box1 box 100 100 0 100 100 100
36 shape3d box2 box 100 250 0 100 100 100
37 shape3d box3 box 100 400 0 100 100 100
38 shape3d sphere1 sphere 250 100 0 50
39 shape3d sphere2 sphere 250 250 0 50
40 shape3d sphere3 sphere 250 400 0 50
41 shape3d cylinder1 cylinder 400 100 0 40 100
42 shape3d cylinder2 cylinder 400 250 0 40 100
43 shape3d cylinder3 cylinder 400 400 0 40 100
44 -- add shapes to scene
45 fill3dshape box1
46 fill3dshape box2
47 fill3dshape box3
48 fill3dshape sphere1
49 fill3dshape sphere2
50 fill3dshape sphere3
51 fill3dshape cylinder1
52 fill3dshape cylinder2
53 fill3dshape cylinder3
54 -- rotate shapes
55 rotate3dshape box1 (-20) 0 0 0 1 1 0
56 rotate3dshape box1 60 0 0 0 0 1 0
57 rotate3dshape box2 60 0 0 0 0 1 0
58 rotate3dshape box3 20 0 0 0 1 1 0
59 rotate3dshape box3 (-60) 0 0 0 0 1 0
60 rotate3dshape cylinder1 40 0 0 0 0 1
61 rotate3dshape cylinder2 70 0 0 0 1 0 1
62 rotate3dshape cylinder2 30 0 0 0 0 1 0
63 rotate3dshape cylinder3 (-40) 0 0 0 0 0 1
64 -- create perspective camera and add to scene
65 camera camera perspective 0 0 0 50
66 setcamera camera
67 -- create new materials for each shape
```

```
68 material boxMaterial1
69 material boxMaterial2
70 material boxMaterial3
71 material sphereMaterial1
72 material sphereMaterial2
73 material sphereMaterial3
74 material cylinderMaterial1
75 material cylinderMaterial2
76 material cylinderMaterial3
77 -- create new images with name "boxMap1" based on image with image path "oorexx_256.png"
78 -- add width=50 and height=50 to image, set rotation angle=0 and set a color="cyan"
79 map boxMap1 "oorexx_256.png" 50 50 0 cyan
80 map boxMap2 "oorexx4000_256.png" 50 50 90 blue
81 map boxMap3 "bsf4oorexx_256.png" 50 50 180 green
82 map sphereMap1 "oorexx_256.png" 800 200 20 white
83 map sphereMap2 "oorexx4000_256.png" 800 200 0 yellow
84 map sphereMap3 "bsf4oorexx_256.png" 800 200 340 orange
85 map cylinderMap1 "oorexx_256.png" 600 200 0 pink
86 map cylinderMap2 "oorexx4000_256.png" 500 200 270 magenta
87 map cylinderMap3 "bsf4oorexx_256.png" 600 200 0 red
88 -- set maps as diffuse maps to each material
89 materialmap boxMaterial1 diffuse boxMap1
90 materialmap boxMaterial2 diffuse boxMap2
91 materialmap boxMaterial3 diffuse boxMap3
92 materialmap sphereMaterial1 diffuse sphereMap1
93 materialmap sphereMaterial2 diffuse sphereMap2
94 materialmap sphereMaterial3 diffuse sphereMap3
95 materialmap cylinderMaterial1 diffuse cylinderMap1
96 materialmap cylinderMaterial2 diffuse cylinderMap2
97 materialmap cylinderMaterial3 diffuse cylinderMap3
98 --set materials to shapes
99 setmaterial box1 boxMaterial1
100 setmaterial box2 boxMaterial2
101 setmaterial box3 boxMaterial3
102 setmaterial sphere1 sphereMaterial1
103 setmaterial sphere2 sphereMaterial2
104 setmaterial sphere3 sphereMaterial3
```

```
105 setmaterial cylinder1 cylinderMaterial1
106 setmaterial cylinder2 cylinderMaterial2
107 setmaterial cylinder3 cylinderMaterial3
108 -- show created window
109 winshow
110 say
111 say "Press Enter to end program..."
112 parse pull
```

Nutshell: Edited Maps



Bonus: Nutshell RGF

```
115   clzDuration=bsf.importClass("javafx.util.Duration")
116   duration=clzDuration~seconds(secs)
117
118   clzRotationTransition=bsf.importClass("javafx.animation.RotateTransition")
119
120   shapes=box1, box2, box3, sphere1, sphere2, sphere3, cylinder1, cylinder2, cylinder3
121   startAngle=0      --
122   diffAngle=35
123
124   getState  -- get all Java collections, we need access to "shapes3d" to access the shape objects directly
125   shapes3d=rc~shapes3d  -- get the shapes3d Map
126
127   do counter c1 shape over shapes
128     currShape=shapes3d~get(shape)  -- get shape3d from Map
129     rt=clzRotationTransition~new(duration, currShape)  -- create a RotationTransition
130     startAngle+=diffAngle*c1
131     rt~ByAngle  =startAngle -- +diffAngle*c1
132     rt~CycleCount =4
133     rt~AutoReverse=.true
134     |  -- send PLAY on the FX GUI thread
135     .fxGuiThread~runLater(rt, "play")
136   end
137   say "press <enter> to end program"
138   parse pull .
139   say "Done."
```

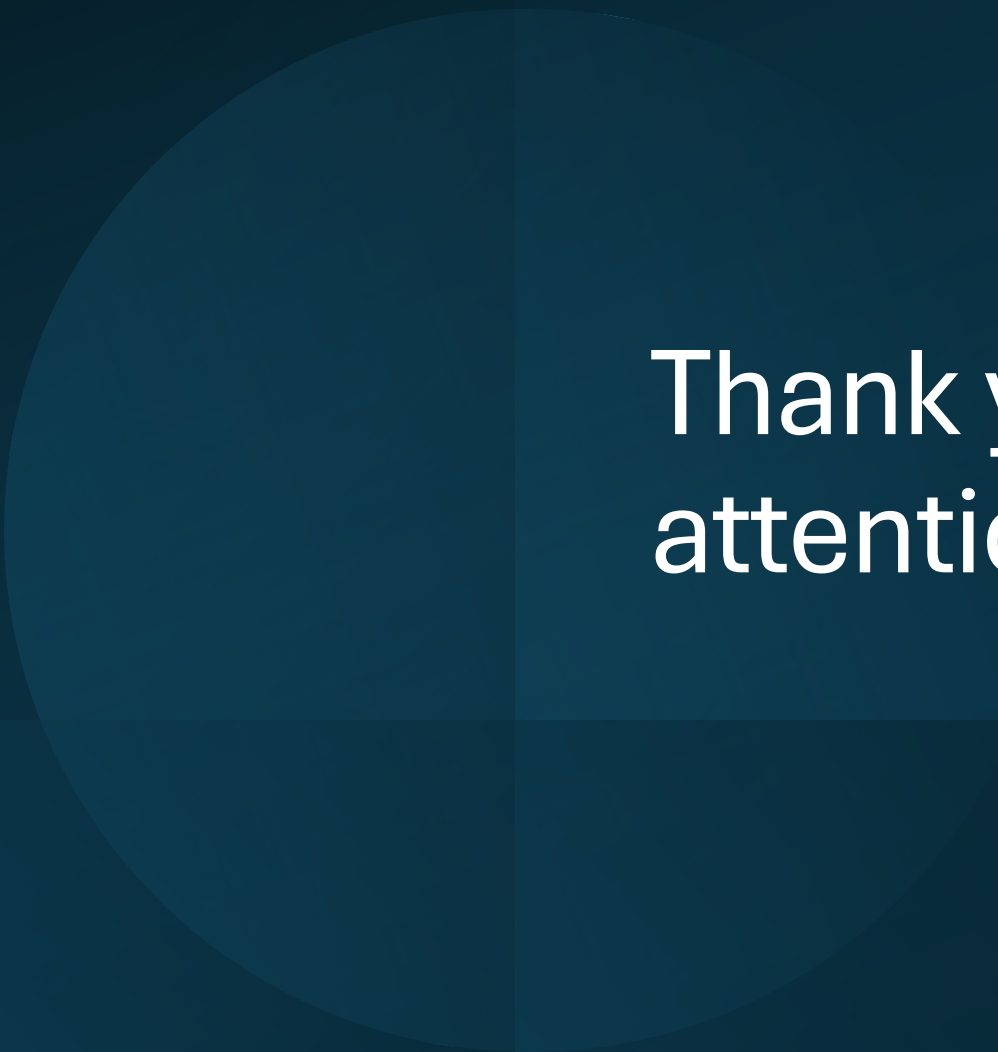
Java vs JDORFX

```
1 import javafx.application.Application;
2 import javafx.scene.Group;
3 import javafx.scene.Scene;
4 import javafx.scene.shape.*;
5 import javafx.stage.Stage;
6
7 public class Example extends Application {
8
9     @Override
10    public void start(Stage stage) throws Exception {
11
12        Group root = new Group();
13        Scene scene = new Scene(root, width: 500, height: 500);
14
15        Box box = new Box( width: 100, height: 100, depth: 100);
16        box.setTranslateX(100);
17        box.setTranslateY(100);
18        box.setTranslateZ(100);
19
20        root.getChildren().add(box);
21
22        stage.setScene(scene);
23        stage.show();
24    }
25 }
26
27
```

```
1 call addJdorFXHandler
2 address jdorfx
3
4 newimage 500 500
5
6 shape3d myBox box 100 100 100 100 100 100
7
8 draw3dshape myBox
9
10 winshow
11
12 -- get ooRexx-Java bridge, contains JDORFX Rexx command handler
13 ::requires "jdorfx.CLS"
```

Outlook

- Implement all remaining JDOR commands
- Modifying existing implementations
 - Transformations of camera objects
 - Transformations of light objects
- Implementing new classes
 - Animations
 - Effects (eg Shadow)
 - Charts
 - MediaView (video / audio)
 - EventHandler



Thank you for your
attention!