

Course code : **CSE2005**

Course title : **Object Oriented Programming**

JavaFX Canvas

Objectives

This session will give the knowledge about

- JavaFX Canvas

JavaFX Canvas

- Through the `javafx.scene.canvas` package, JavaFX provides the **Canvas API** that offers a drawing surface to draw shapes, images, and text using drawing commands.
- The API also gives pixel-level access to the drawing surface where you can write any pixels on the surface. **The API consists of only two classes:**
 - **Canvas**
 - **GraphicsContext**

What is a Canvas?

A canvas is a bitmap image, which is used as a drawing surface. An instance of the Canvas class represents a canvas.

It inherits from the Node class. Therefore, a **Canvas is a Node**.

It can be added to a Scene Graph, and effects and transformations can be applied to it.

A Canvas has a graphics context associated with it that is used to issue drawing commands to the Canvas. **An instance of the GraphicsContext class represents a graphics context.**

2D Shapes in Canvas

The GraphicsContext class provides two types of methods to draw the basic shapes. The method **fillXxx()** draws a shape Xxx and fills it with the current fill paint.

The method **strokeXxx()** draws a shape Xxx with the current stroke. Use the following methods for drawing shapes:

- fillArc()
- fillOval()
- fillPolygon()

2D Shapes in Canvas

- fillRect()
- fillRoundRect()
- strokeArc()
- strokeLine()
- strokeOval()
- strokePolygon()
- strokePolyline()
- strokeRect() and strokeRoundRect()

How to draw Shapes in Canvas

Step-1: Create a Canvas area by instantiating the Canvas class

- `Canvas()`: Creates a new canvas object.
- `Canvas(double w, double h)`: Creates a new canvas object with specified width and height.

Step-2: Create GraphicsContext object to draw 2D shapes

- `GraphicsContext gc = canvas.getGraphicsContext2D();`

Step-3: Apply common methods to set Stroke and Fill colors

How to draw Shapes in Canvas

Step-4: Use fillXXX() and strokeXXX() methods to draw 2D shape using GraphicsContext object

- gc.fillOval(120, 20, 70, 70);
- gc.strokeRect(0, 0, 300, 200);

Step-5: Create Group object instead of Layout to add all shapes.

- Group group = new Group(canvas);

Step-6: Add Group object into Scene

Step-7: If you are adding other controls like buttons, you must create a layout and add Group object into the layout as controls

Commonly Used Methods in Canvas

- `getGraphicsContext2D()` Returns the graphics context associated with the canvas.
- `getHeight()` Returns the height of the canvas.
- `getWidth()` Returns the width of the canvas.
- `setHeight(double v)` Sets the height of the canvas.
- `setWidth(double d)` Sets the width of the canvas.

2D Shape demo-1

```
public void start(Stage primaryStage) throws Exception {  
    Canvas canvas = new Canvas();  
    canvas.setWidth(300);  
    canvas.setHeight(200);  
  
    GraphicsContext gc = canvas.getGraphicsContext2D();  
  
    gc.setFill(Color.RED);  
    gc.fillRect(20, 20, 70, 70);  
    gc.setFill(Color.BLUE);  
    gc.fillOval(120, 20, 70, 70);  
}
```

2D Shape demo-1

```
gc.strokeRect(0, 0, 300, 200);
Group group = new Group(canvas);
Button b=new Button("click");
VBox root=new VBox();
root.getChildren().addAll(group,b);
Scene scene = new Scene(root, 400, 400);
// set the scene
primaryStage.setScene(scene);
primaryStage.setTitle("creating canvas");
primaryStage.show();
}
```

2D Shapes in Canvas

fillRect - Fills a rectangle using the current fill paint.

```
public void fillRect(double x, double y, double w, double h)
```

strokeRect - strokes a rectangle using the current stroke paint.

```
public void strokeRect(double x, double y, double w, double h)
```

Parameters:

- x - the X position of the left corner of the rectangle.
- y - the Y position of the upper corner of the rectangle.
- w - the width of the rectangle.
- h - the height of the rectangle.

2D Shapes in Canvas

fillRoundRect - Fills a rounded rectangle using the current fill paint.

```
public void fillRoundRect(double x, double y, double w, double h, double  
arcWidth, double arcHeight)
```

strokeRoundRect - Strokes a rounded rectangle using the current stroke paint.

```
public void strokeRoundRect(double x, double y, double w, double h,  
double arcWidth, double arcHeight)
```

Parameters:

- x - the X coordinate of the upper left bound of the oval.

2D Shapes in Canvas

- `y` - the Y coordinate of the upper left bound of the oval.
- `w` - the width at the center of the oval.
- `h` - the height at the center of the oval.
- `arcWidth` - the arc width of the rectangle corners.
- `arcHeight` - the arc height of the rectangle corners.

2D Shapes in Canvas

fillOval - Fills an oval using the current fill paint.

```
public void fillOval(double x,double y,double w,double h)
```

strokeOval - Strokes an oval using the current stroke paint.

```
public void strokeOval(double x,double y,double w,double h)
```

Parameters:

- x - the X coordinate of the upper left bound of the oval.
- y - the Y coordinate of the upper left bound of the oval.
- w - the width at the center of the oval.
- h - the height at the center of the oval.

2D Shapes in Canvas

fillArc (strokeArc) - Fills an oval using the current fill paint.

```
public void fillArc(double x,double y,double w,  
    double h,double startAngle,  
    double arcExtent, ArcType closure)
```

Fills an arc using the current fill paint. A null ArcType or non positive width or height will cause the render command to be ignored.

This method will be affected by any of the global common or fill attributes as specified in the Rendering Attributes Table.

2D Shapes in Canvas

Parameters:

- x - the X coordinate of the arc.
- y - the Y coordinate of the arc.
- w - the width of the arc.
- h - the height of the arc.
- startAngle - the starting angle of the arc in degrees.
- arcExtent - the angular extent of the arc in degrees.
- closure - closure type (Round, Chord, Open) or null.

2D Shapes in Canvas

fillPolygon (strokePolygon) - Fills a polygon with the given points using the currently set fill paint.

```
public void fillPolygon(double[] xPoints,  
                        double[] yPoints,  
                        int nPoints)
```

Fills a polygon with the given points using the currently set fill paint. A null value for any of the arrays will be ignored and nothing will be drawn.

This method will be affected by any of the global common, fill, or Fill Rule attributes as specified in the Rendering Attributes Table.

2D Shapes in Canvas

Parameters:

- xPoints - array containing the x coordinates of the polygon's points or null.
- yPoints - array containing the y coordinates of the polygon's points or null.
- nPoints - the number of points that make the polygon.

2D Shapes in Canvas

strokeLine (no method for fillLine) - Strokes a line using the current stroke paint.

```
public void strokeLine(double x1, double y1, double x2, double y2)
```

Parameters:

- x1 - the X coordinate of the starting point of the line.
- y1 - the Y coordinate of the starting point of the line.
- x2 - the X coordinate of the ending point of the line.
- y2 - the Y coordinate of the ending point of the line.

2D Shape demo-2

```
public void start(Stage primaryStage) throws Exception {  
    Canvas canvas = new Canvas();  
    canvas.setWidth(400);  
    canvas.setHeight(200);  
  
    GraphicsContext gc = canvas.getGraphicsContext2D();  
    gc.setLineWidth(2.0);  
    gc.setFill(Color.RED);  
    gc.setStroke(Color.GREEN);  
  
    gc.strokeRect(0, 0, canvas.getWidth(), canvas.getHeight());  
}
```

2D Shape demo-2

// Draw a rounded Rectangle

```
gc.strokeRoundRect(10, 10, 50, 50, 10, 10);
```

// Draw a filled rounded Rectangle

```
gc.fillRoundRect(100, 10, 50, 50, 10, 10);
```

// Change the fill color

```
gc.setFill(Color.BLUE);
```

// Draw an Oval

```
gc.strokeOval(10, 70, 50, 30);
```

// Draw a filled Oval

```
gc.fillOval(100, 70, 50, 30);
```

2D Shape demo-2

// Draw a Line

```
gc.strokeLine(200, 50, 300, 50);
```

// Draw an Arc

```
gc.strokeArc(320, 10, 50, 50, 40, 80, ArcType.OPEN);
```

// Draw a filled Arc

```
gc.fillArc(320, 70, 50, 50, 00, 120, ArcType.OPEN);
```

```
Group group = new Group(canvas);
```

```
Scene scene = new Scene(group, 400, 400);
```

```
primaryStage.setScene(scene);
```

```
primaryStage.setTitle("creating canvas");
```

```
primaryStage.show();      }
```

2D Shapes in Canvas

fillText (strokeText) - Fills the given string of text at position x, y with the current fill paint attribute. A null text value will be ignored.

```
public void fillText(String text, double x, double y)
```

Parameters:

- text - the string of text or null.
- x - position on the x axis.
- y - position on the y axis.

2D Shapes in Canvas

fillText (strokeText) - Fills text and includes a maximum width of the string. If the width of the text extends past max width, then it will be sized to fit. A null text value will be ignored.

```
public void fillText(String text, double x, double y, double maxWidth)
```

Parameters:

- text - the string of text or null.
- x - position on the x axis.
- y - position on the y axis.
- maxWidth - maximum width the text string can have.

2D Shape demo-3

```
public void start(Stage primaryStage) throws Exception {  
    Canvas canvas = new Canvas();  
    canvas.setWidth(400);  
    canvas.setHeight(200);  
  
    GraphicsContext gc = canvas.getGraphicsContext2D();  
    gc.setLineWidth(1.0);  
    gc.setFill(Color.RED);  
  
    // Draw a Text  
    gc.strokeText("This is a stroked Text", 10, 50);  
}
```

2D Shape demo-3

```
gc.strokeText("This is a stroked Text with Max Width 300 px", 10, 100, 300);  
    // Draw a filled Text  
gc.fillText("This is a filled Text", 10, 150);  
gc.fillText("This is a filled Text with Max Width 400 px", 10, 200, 400);  
  
Group group = new Group(canvas);  
Scene scene = new Scene(group, 400, 400);  
primaryStage.setScene(scene);  
primaryStage.setTitle("creating canvas");  
primaryStage.show();  
}
```

2D Shapes in Canvas

drawImage - Draws an image at the given x, y position using the width and height of the given image. A null image value or an image still in progress will be ignored.

```
public void drawImage(Image img, double x, double y)
```

Parameters:

- img - the image to be drawn or null.
- x - the X coordinate on the destination for the upper left of the image.
- y - the Y coordinate on the destination for the upper left of the image.

2D Shapes in Canvas

drawImage - Draws an image into the given destination rectangle of the canvas. The image is scaled to fit into the destination rectangle. A null image value or an image still in progress will be ignored.

```
public void drawImage(Image img, double x, double y, double w, double h)
```

Parameters:

- img - the image to be drawn or null.
- x - the X coordinate on the destination for the upper left of the image.
- y - the Y coordinate on the destination for the upper left of the image.
- w - the width of the destination rectangle.
- h - the height of the destination rectangle.

2D Shapes in Canvas

drawImage - Draws the specified source rectangle of the given image to the given destination rectangle of the Canvas. A null image value or an image still in progress will be ignored.

```
public void drawImage(Image img, double sx, double sy, double  
sw, double sh, double dx, double dy, double dw, double dh)
```

Parameters:

- img - the image to be drawn or null.
- sx - the source rectangle's X coordinate position.
- sy - the source rectangle's Y coordinate position.

2D Shapes in Canvas

Parameters:

- sw - the source rectangle's width.
- sh - the source rectangle's height.
- dx - the destination rectangle's X coordinate position.
- dy - the destination rectangle's Y coordinate position.
- dw - the destination rectangle's width.
- dh - the destination rectangle's height.

2D Shape demo-4

```
public void start(Stage primaryStage) throws Exception {  
    Canvas canvas = new Canvas();  
    canvas.setWidth(400);  
    canvas.setHeight(200);  
    GraphicsContext gc = canvas.getGraphicsContext2D();  
    gc.setLineWidth(1.0);  
    gc.setFill(Color.RED);  
    FileInputStream file=null;  
    try {  
        file = new FileInputStream("images/logo.jpg");  
    } catch (FileNotFoundException e) {
```


2D Shape demo-4

```
        e.printStackTrace();
    }
    Image image=new Image(file);
    // Draw the Image
    gc.drawImage(image, 10, 10, 200, 200);
    gc.drawImage(image, 220, 50, 100, 70);
    Group group = new Group(canvas);
    Scene scene = new Scene(group, 400, 400);
    primaryStage.setScene(scene);
    primaryStage.setTitle("creating canvas");
    primaryStage.show(); }
```

Summary

We have discussed about

- JavaFX Canvas