

Lecture 6 - Java Graphical User Interface (GUI): JavaFX - Part III

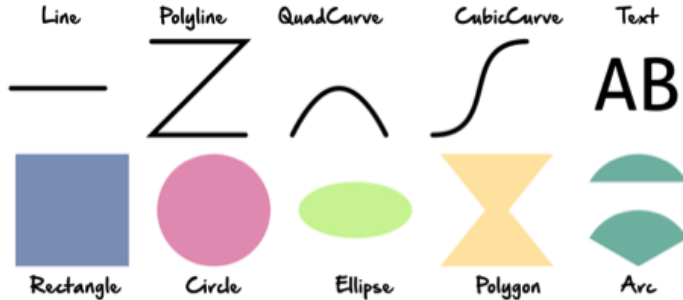
Guiliang Liu

The Chinese University of Hong Kong, Shenzhen

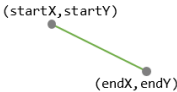
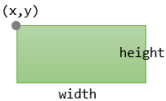

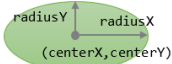
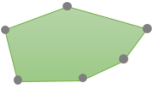

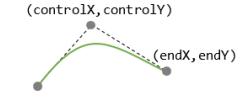
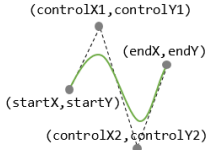
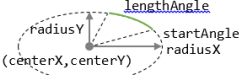
CSC-1004: Computational Laboratory Using Java
Course Page: [\[Click\]](#)

JavaFX 2D Shapes

JavaFX provides the flexibility to create our **2D shapes** on the screen. Various classes can be used to implement 2D shapes in our application. All these classes reside in the `javafx.scene.shape` package.



JavaFX 2D Shapes

 <p>Line</p>	 <p>Rectangle</p>	 <p>Circle</p>
 <p>Ellipse</p>	 <p>Polygon</p>	 <p>Polyline</p>
 <p>QuadCurve</p>		 <p>CubicCurve</p>
 <p>Arc</p>		

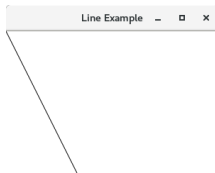
JavaFX 2D Shapes: Line

JavaFX library allows the developers to create a **Line**.

```
public class LineDrawingExamples extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        Line line = new Line(); //instantiating Line class
        line.setStartX(0); //setting starting X point of Line
        line.setStartY(0); //setting starting Y point of Line
        line.setEndX(100); //setting ending X point of Line
        line.setEndY(200); //setting ending Y point of Line
        Group root = new Group(); //Creating a Group
        root.getChildren().add(line); //adding the class object //to the group
        Scene scene = new Scene(root,300,300);
        primaryStage.setScene(scene);
        primaryStage.setTitle("Line Example");
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }
}
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX 2D Shapes: Rectangle

JavaFX library allows the developers to create a **rectangle**.

```
public class Shape_Example extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        primaryStage.setTitle("Rectangle Example");
        Group group = new Group(); //creating Group
        Rectangle rect=new Rectangle(); //instantiating Rectangle
        rect.setX(20); //setting the X coordinate of upper left //corner of rectangle
        rect.setY(20); //setting the Y coordinate of upper left //corner of rectangle
        rect.setWidth(100); //setting the width of rectangle
        rect.setHeight(100); // setting the height of rectangle
        group.getChildren().addAll(rect); //adding rectangle to the //group
        Scene scene = new Scene(group,200,300,Color.GRAY);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX 2D Shapes: Circle

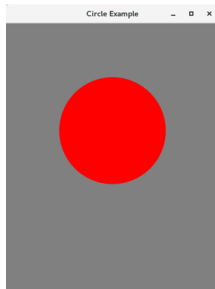
JavaFX library allows the developers to create a **Circle**.

```
public class Shape_Example extends Application{

    @Override
    public void start(Stage primaryStage) throws Exception {
        // TODO Auto-generated method stub
        primaryStage.setTitle("Circle Example");
        Group group = new Group();
        Circle circle = new Circle();
        circle.setCenterX(200);
        circle.setCenterY(200);
        circle.setRadius(100);
        circle.setFill(Color.RED);
        group.getChildren().addAll(circle);
        Scene scene = new Scene(group,400,500,Color.GRAY);
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    public static void main(String[] args) {
        launch(args);
    }

}
```



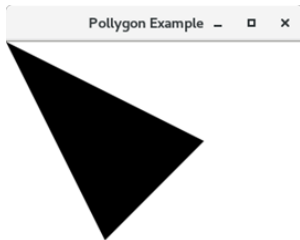
香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX 2D Shapes: Polygons

In JavaFX, **Polygon** can be created by instantiating `javafx.scene.shape.Polygon` class.

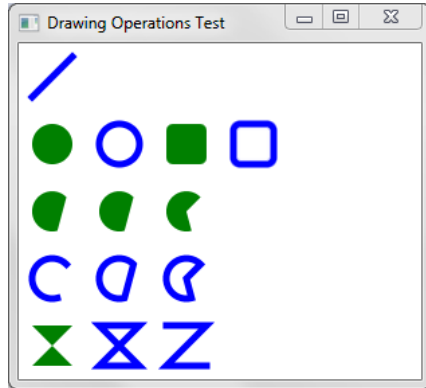
```
public class Shape_Example extends Application {  
  
    @Override  
    public void start(Stage primaryStage) {  
        Group root = new Group();  
        primaryStage.setTitle("Polygon Example");  
  
        Polygon polygon = new Polygon();  
        polygon.getPoints().addAll(new Double[]{  
            0.0, 0.0,  
            100.0, 200.0,  
            200.0, 100.0 });  
  
        root.getChildren().add(polygon);  
        Scene scene = new Scene(root,300,400);  
        primaryStage.setScene(scene);  
        primaryStage.show();  
    }  
  
    public static void main(String[] args) {  
        launch(args);  
    }  
}
```



文 大 學 (深 圳)
The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

Canvas class creates an image that can be drawn on using a set of graphics commands provided by a GraphicsContext. Canvas has a specified height and width and all the drawing operations are clipped to the bounds of the canvas.



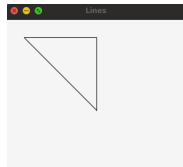
香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

In the first example, we draw **simple lines**.

```
public class CanvasLine extends Application {  
  
    @Override  
    public void start(Stage stage) {  
        initUI(stage);  
    }  
  
    1 usage  
    private void initUI(Stage stage) {  
  
        var root = new Pane();  
  
        var canvas = new Canvas( v: 300, v1: 300);  
        var gc = canvas.getGraphicsContext2D();  
        drawLines(gc);  
  
        root.getChildren().add(canvas);  
  
        var scene = new Scene(root, v: 300, v1: 250, Color.WHITESMOKE);  
  
        stage.setTitle("Lines");  
        stage.setScene(scene);  
        stage.show();  
    }  
}
```

```
1 usage  
private void drawLines(GraphicsContext gc) {  
  
    gc.beginPath();  
    gc.moveTo( v: 30.5, v1: 30.5);  
    gc.lineTo( v: 150.5, v1: 30.5);  
    gc.lineTo( v: 150.5, v1: 150.5);  
    gc.lineTo( v: 30.5, v1: 30.5);  
    gc.stroke();  
}  
  
public static void main(String[] args) {  
    launch(args);  
}
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

- `var canvas = new Canvas(300, 300);` A Canvas is constructed with a width and height that specifies the size of the image into which the canvas drawing commands are rendered.
- `var gc = canvas.getGraphicsContext2D();` The `getGraphicsContext2D` returns a `GraphicsContext` associated with the canvas.
- `gc.beginPath();` A line primitive is represented as a path element. The `beginPath` method starts a new path.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

- `gc.moveTo(30.5, 30.5);` The `moveTo` method moves the starting point of the current path to the specified coordinate.
- `gc.lineTo(150.5, 30.5);` The `lineTo` methods add line segments to the current path.
- `gc.stroke();` The `stroke` method strokes the path with the current stroke paint.



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

A **stroke** is used to draw outlines of shapes. A **fill** is used to paint interiors of shapes.

```
public class CanvasStrokeFill extends Application {
```

```
    @Override
```

```
    public void start(Stage stage) {
```

```
        var root = new Pane();
```

```
        var canvas = new Canvas( w: 300, h: 300);
```

```
        var gc = canvas.getGraphicsContext2D();
```

```
        doDrawing(gc);
```

```
        root.getChildren().add(canvas);
```

```
        var scene = new Scene(root, w: 300, h: 250, Color.WHITESMOKE);
```

```
        stage.setTitle("Stroke and fill");
```

```
        stage.setScene(scene);
```

```
        stage.show();
```

```
    }
```

```
1 usage
```

```
    private void doDrawing(GraphicsContext gc) {
```

```
        gc.setStroke(Color.FORESTGREEN.brighter());
```

```
        gc.setLineWidth(5);
```

```
        gc.strokeOval( w: 30, h: 30, w2: 80, h2: 80);
```

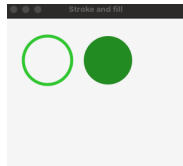
```
        gc.setFill(Color.FORESTGREEN);
```

```
        gc.fillOval( w: 130, h: 30, w2: 80, h2: 80);
```

```
    }
```

```
    public static void main(String[] args) { launch(args); }
```

```
}
```



香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

JavaFX Canvas

- `gc.setStroke(Color.FORESTGREEN.brighter());` The `setStroke` method sets the current stroke paint attribute. The default colour is black. The attribute is used by the stroke methods of the `GraphicsContext`.
- `gc.setLineWidth(5);` The `setLineWidth` sets the current line width.
- `gc.strokeOval(130, 30, 80, 80);` The `strokeOval` method strokes an oval using the current stroke paint.



JavaFX Canvas

- `gc.setFill(Color.FORESTGREEN);` The `setFill` method sets the current fill paint attribute. The default colour is black. The attribute is used by the fill methods of the `GraphicsContext`.
- `gc.fillOval(30, 30, 80, 80);` The `fillOval` fills an oval using the current fill paint.

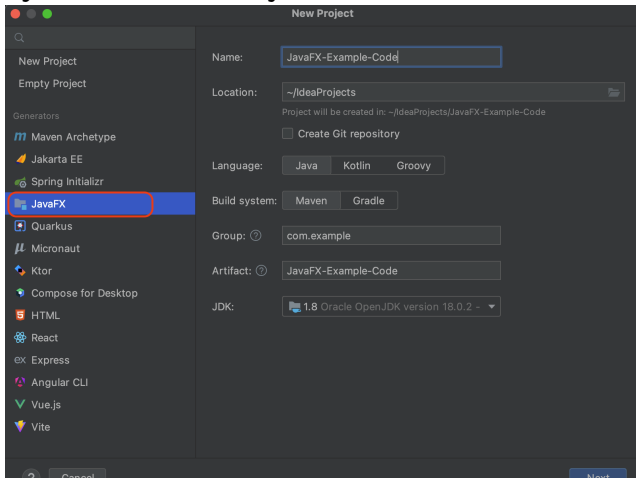


香港中文大學(深圳)

The Chinese University of Hong Kong, Shenzhen

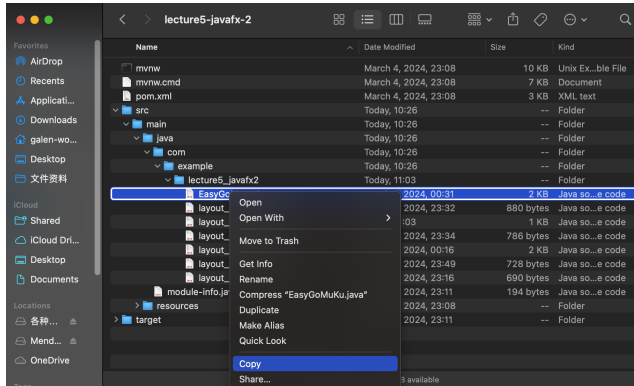
Import JavaFX Example Code

- **Step 1:** Create a new **JavaFX Project** on your PC/Laptop. Try to use *File->New->Project* under the IntelliJ idea.



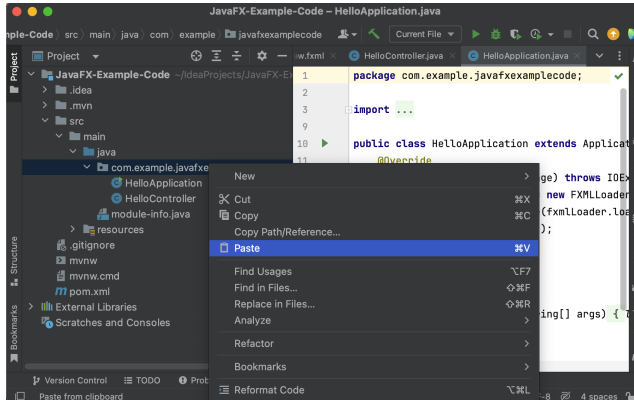
Import JavaFX Example Code

- **Step 2:** Copy the target **.Java File** from the folder of example code.



Import JavaFX Example Code

- **Step 3:** Paste the .Java File to the folder of the right place (e.g., *YourProjectName/src/java/com/example/YourFolderName*) in the **created project**.
The pasting must be done with the **Intellij idea**.

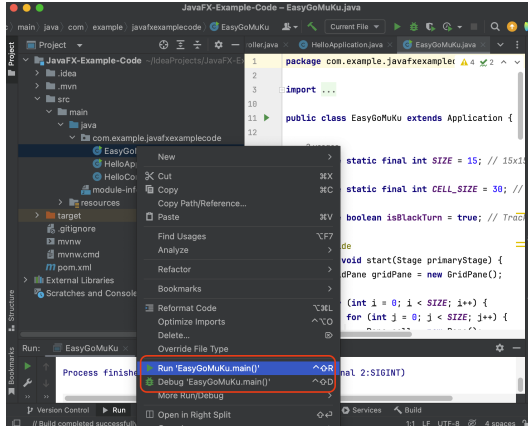


中文大學(深圳)

Chinese University of Hong Kong, Shenzhen

Import JavaFX Example Code

- **Step 4:** Run/Debug the code by right-clicking the Java file (config your JDK if necessary, and please check whether the JavaFX package has been downloaded to the "External Libraries" in your project.).



Question and Answering (Q&A)



香港中文大學(深圳)
The Chinese University of Hong Kong, Shenzhen