TAIBAH UNIVERSITY جـامـعة طيبة

Computer System

Name: Waseem Zahr Aljumah.

ID: 4206111

Section: M6

Develop a JavaFX application that utilizes radial gradient fill to create visually appealing shapes. Allow the user to choose different colors, stops, and radii for the radial gradient. Display the shapes with radial gradient fill on the canvas and ensure the user can interactively modify the parameters.

In your implementation consider the following:

- Set up your application with appropriate packages and classes.

- Implement various JavaFX shapes (e.g., Circle, Rectangle) that will be filled with radial gradients.

- Ensure that each shape can be dynamically filled with the user-selected radial gradient.

- Organize your code into modular functions or methods, enhancing code readability and maintainability.

- Use meaningful variable and method names to convey the purpose of your code.

- Test your program by running the JavaFX application.

- Submit the entire project folder, including source code and any necessary resource files.

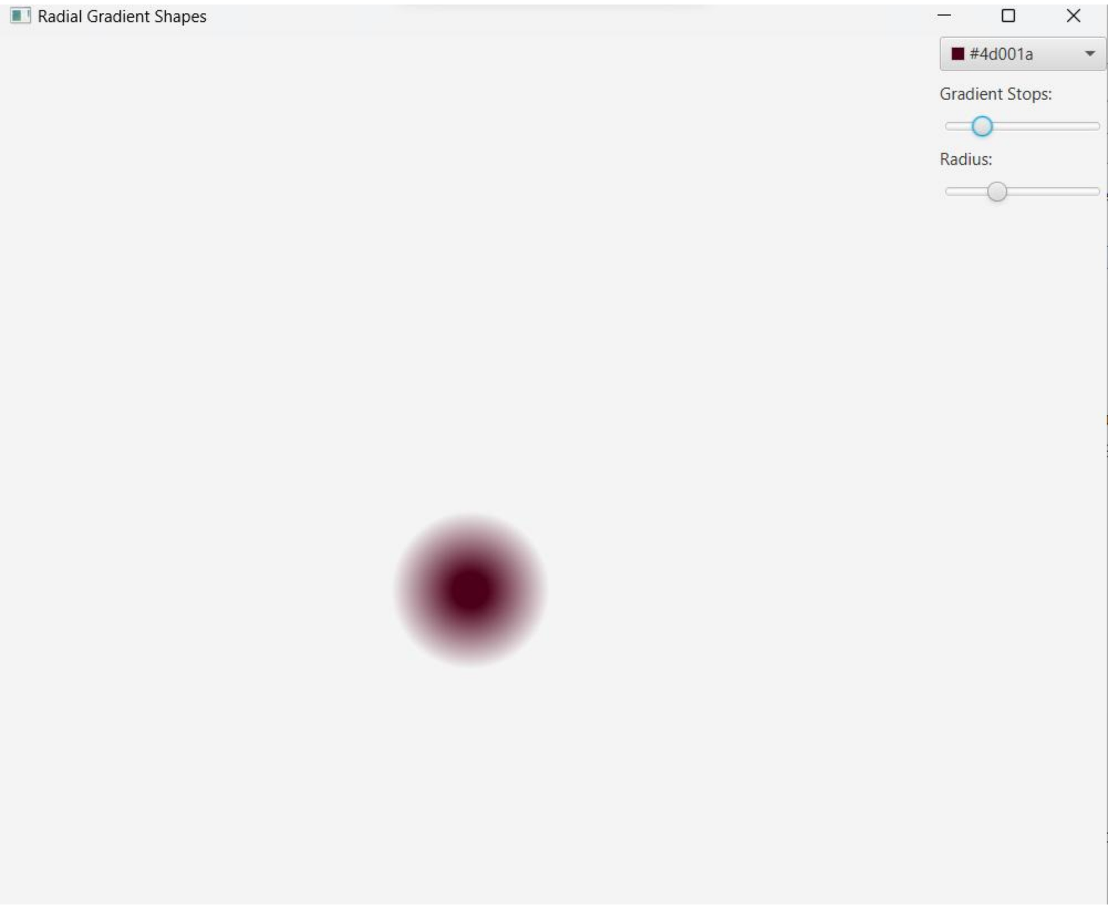
- Submit a screenshot displaying the output of your implementation.

CODE:

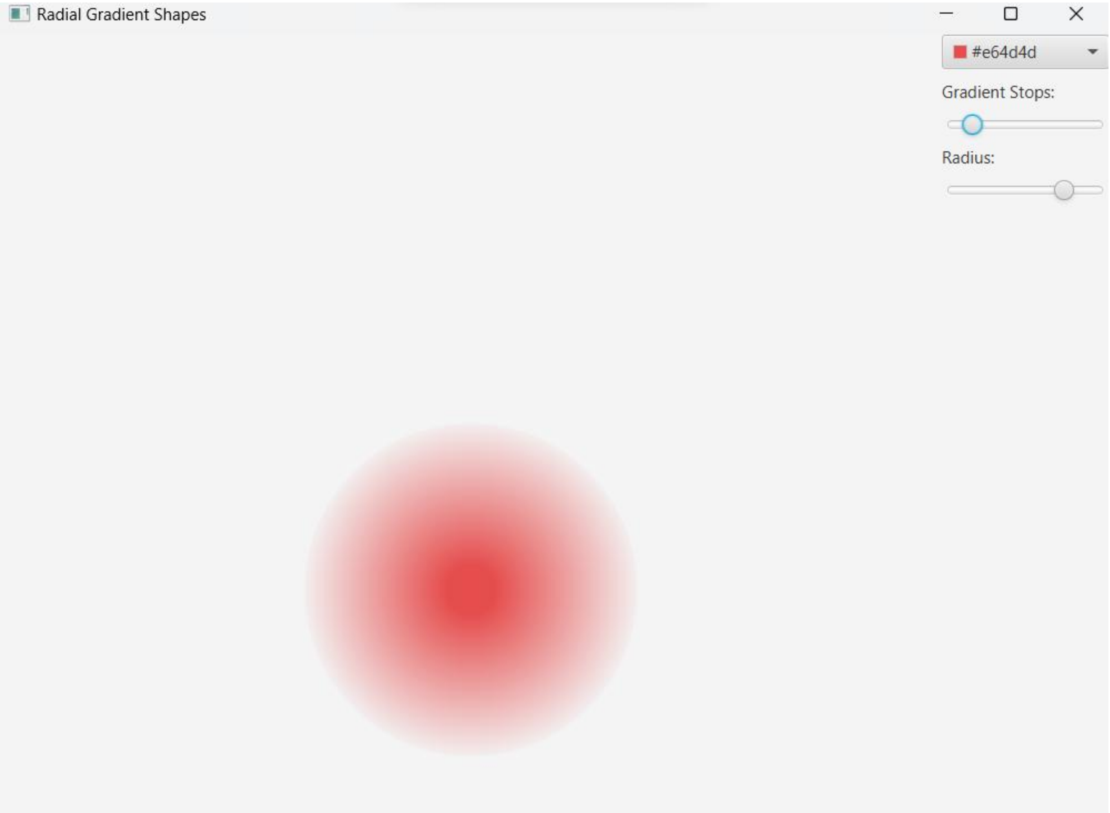
```
package Project;


import javafx.application.Application;
import static javafx.application.Application.launch;
import javafx.scene.Scene;
import javafx.scene.control.ColorPicker;
import javafx.scene.control.Label;
import javafx.scene.control.Slider;
import javafx.scene.layout.BorderPane;
import javafx.scene.paint.Color;
import javafx.scene.paint.CycleMethod;
import javafx.scene.paint.RadialGradient;
import javafx.scene.paint.Stop;
import javafx.scene.shape.Circle;
import javafx.stage.Stage;


public class Project extends Application {


   private Circle shape;
   private RadialGradient radialGradient;


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

```
    @Override
    public void start(Stage primaryStage) {
        primaryStage.setTitle("Radial Gradient Shapes");

        shape = new Circle(150);
        BorderPane root = new BorderPane();
        root.setCenter(shape);

        createControls(root);

        primaryStage.setScene(new Scene(root, 1000, 1000));
        primaryStage.show();

        updateShape();
    }

    private void createControls(BorderPane root) {
        ColorPicker colorPicker = new ColorPicker();
        colorPicker.setOnAction(event -> updateShape());
        colorPicker.setId("colorPicker");

        Label stopsLabel = new Label("Gradient Stops:");
        Slider stopsSlider = new Slider(0, 1, 0.5);
        stopsSlider.valueProperty().addListener((observable, oldValue,
newValue) -> updateShape());
        stopsSlider.setId("stopsSlider");
        Label radiusLabel = new Label("Radius:");
        Slider radiusSlider = new Slider(10, 200, 100);
        radiusSlider.valueProperty().addListener((observable, oldValue,
newValue) -> updateShape());
        radiusSlider.setId("radiusSlider");

        javafx.scene.layout.VBox controls = new
javafx.scene.layout.VBox(10);
        controls.getChildren().addAll(
                colorPicker,
                stopsLabel, stopsSlider,
                radiusLabel, radiusSlider
        );

        root.setRight(controls);
    }

    private Slider createSlider(double min, double max, double
initialValue, javafx.beans.value.ChangeListener<Number> listener) {
        Slider slider = new Slider(min, max, initialValue);
        slider.valueProperty().addListener(listener);
        return slider;
    }

    private void updateShape() {
        double radius = shape.getRadius();
        double centerX = shape.getCenterX();
        double centerY = shape.getCenterY();

        BorderPane parentBorderPane = (BorderPane) shape.getParent();
        javafx.scene.layout.VBox controlsVBox =
(javafx.scene.layout.VBox) parentBorderPane.getRight();

        ColorPicker colorPicker = (ColorPicker)
controlsVBox.lookup("#colorPicker");
        Color color = colorPicker.getValue();

        Slider stopsSlider = (Slider) controlsVBox.lookup("#stopsSlider");
        double stopValue = stopsSlider.getValue();

        Slider radiusSlider = (Slider) controlsVBox.lookup("#radiusSlider");
        double radiusValue = radiusSlider.getValue();

        radialGradient = new RadialGradient(
                0, 0, centerX, centerY, radiusValue, false,
                CycleMethod.NO_CYCLE, new Stop(stopValue, color), new
Stop(1, Color.TRANSPARENT)
        );

        shape.setFill(radialGradient);
    }
}
```

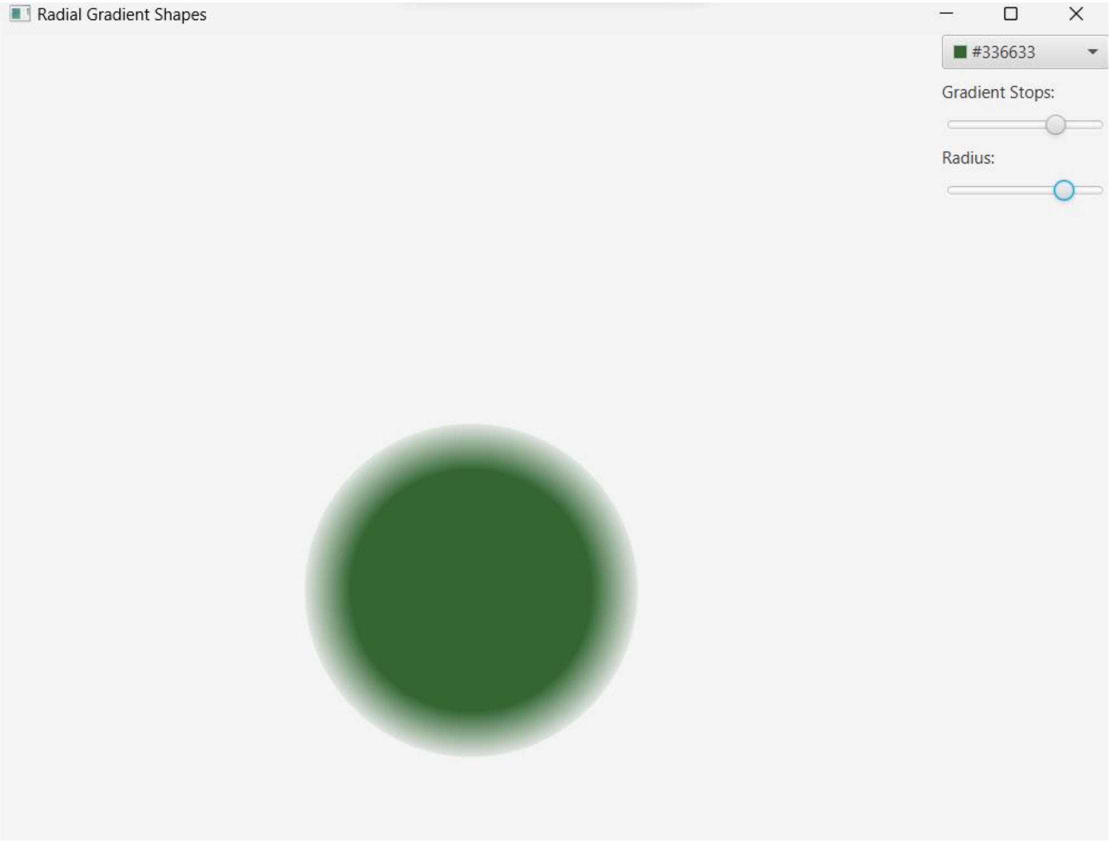
OUTPUT:



Radial Gradient Shapes



Radial Gradient Shapes

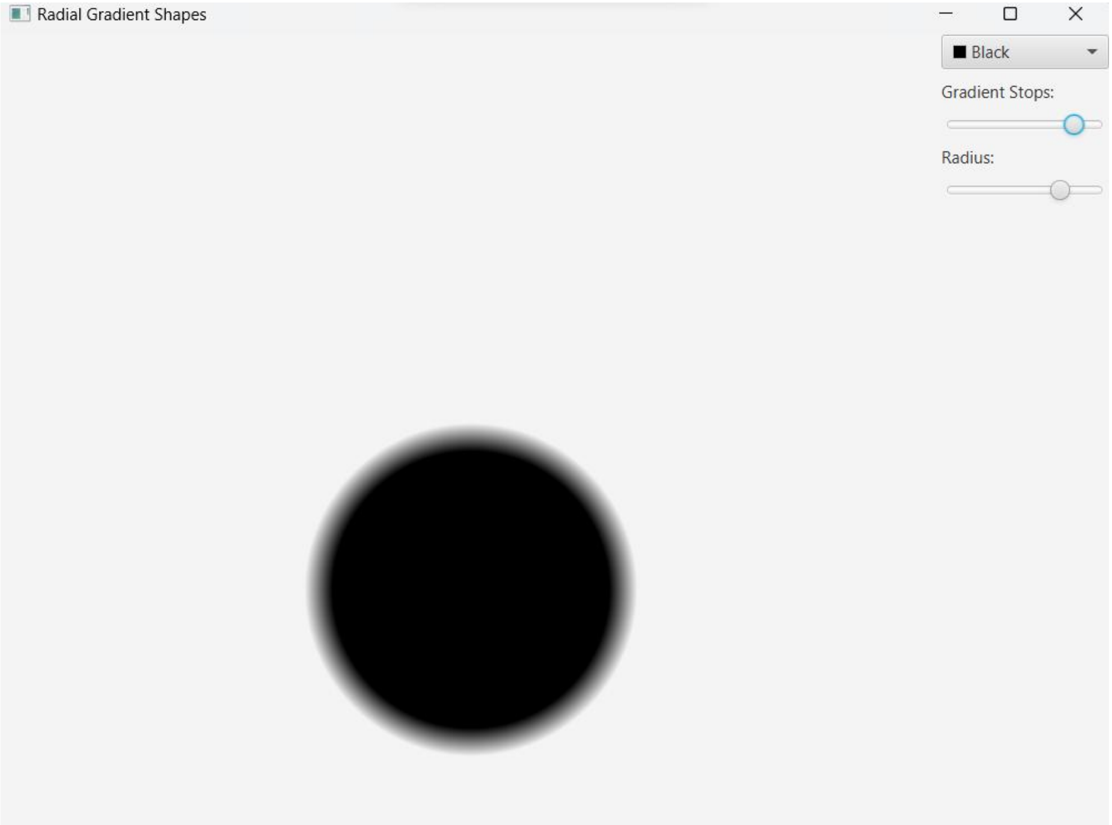


Radial Gradient Shapes

🟩 #336633

Gradient Stops:

Radius:

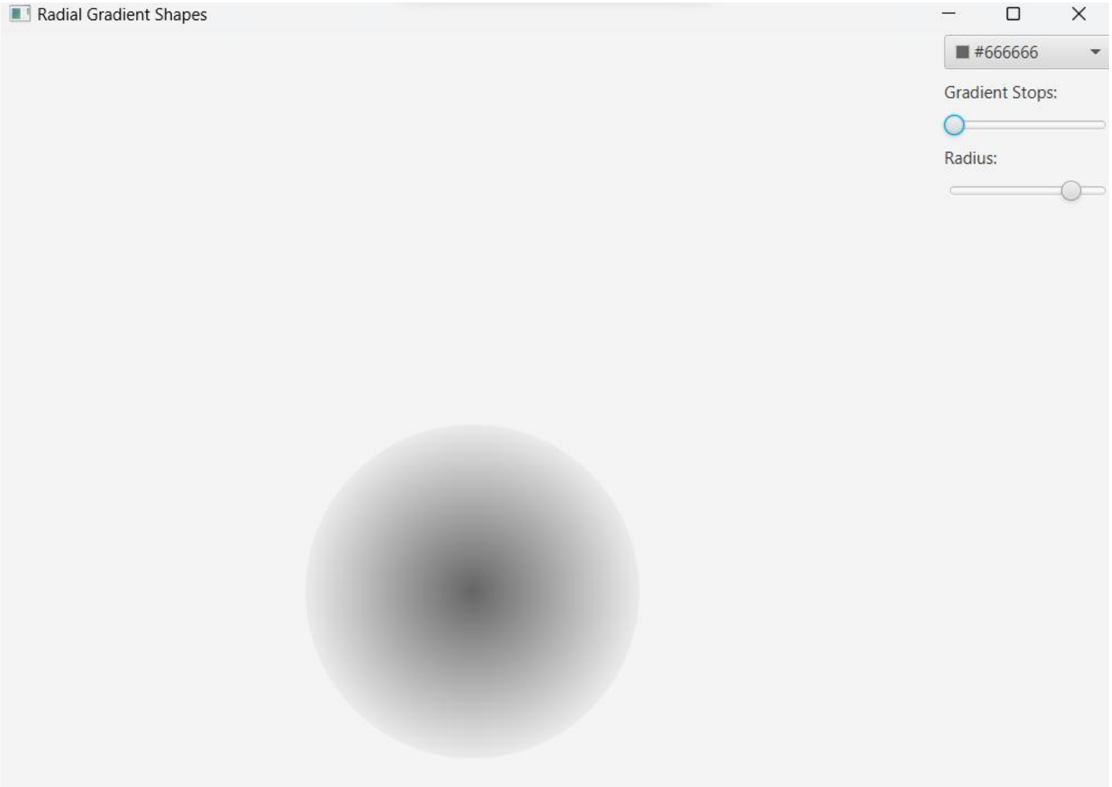


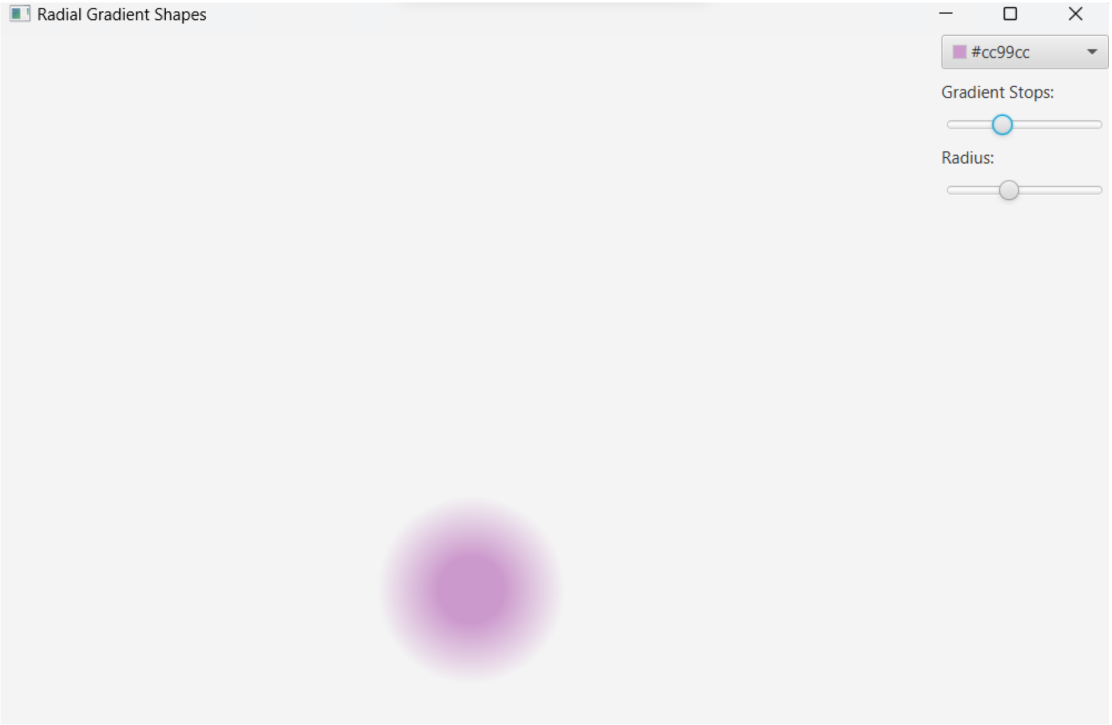
Radial Gradient Shapes

■ Black

Gradient Stops:

Radius:


Radial Gradient Shapes
#666666
Gradient Stops:
Radius:


Radial Gradient Shapes
#cc99cc
Gradient Stops:
Radius: