Java Class 7

Intro to JavaFX

The programs we've explored thus far have been text-based

They are called *command-line applications*, which interact with the user using simple text prompts

We’ll now begin to explore programs that use graphics and graphical user interfaces (GUIs)

Support for these programs will come from the JavaFX API

JavaFX has replaced older approaches (AWT and Swing)

Intro to JavaFX

JavaFX programs extend the `Application` class, inheriting core graphical functionality

A JavaFX program has a `start` method

The `main` method is only needed to launch the JavaFX application

The `start` method accepts the primary stage (window) used by the program as a parameter

JavaFX embraces a theatre analogy

```java
//********************************************************************
//  HelloJavaFX.java       Author: Lewis/Loftus
//
//  Demonstrates a basic JavaFX application.
//********************************************************************

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class HelloJavaFX extends Application
{
    //--------------------------------------------------------------
    //  Creates and displays two Text objects in a JavaFX window.
    //--------------------------------------------------------------
    public void start(Stage primaryStage)
    {
        Text hello = new Text(50, 50, "Hello, JavaFX!");
        Text question = new Text(120, 80, "How's it going?");

        Group root = new Group(hello, question);
        Scene scene = new Scene(root, 300, 120, Color.LIGHTGREEN);
```

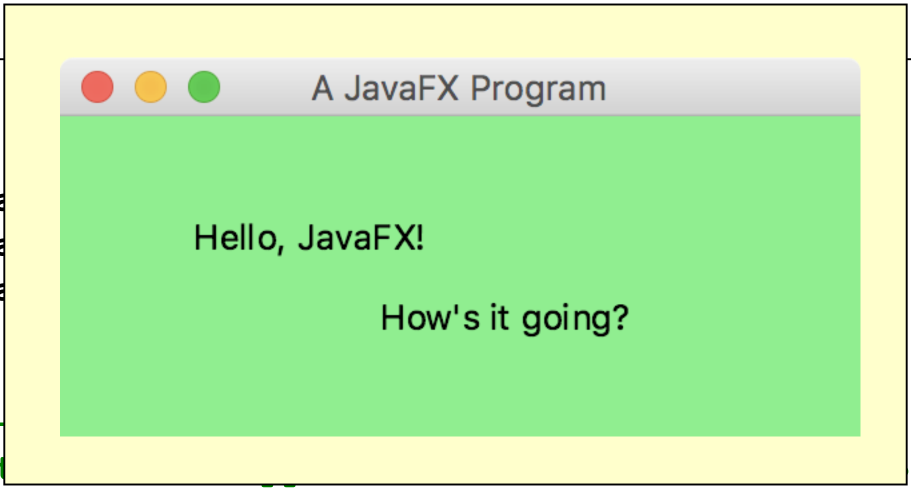
continued

continued

```
        primaryStage.setTitle("A JavaFX Program");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    //-----------------------------------------------------------------------
    //  Launches the JavaFX application. This method is not required
    //  in IDEs that launch JavaFX applications automatically.
    //-----------------------------------------------------------------------
    public static void main(String[] args)
    {
        launch(args);
    }
}
```

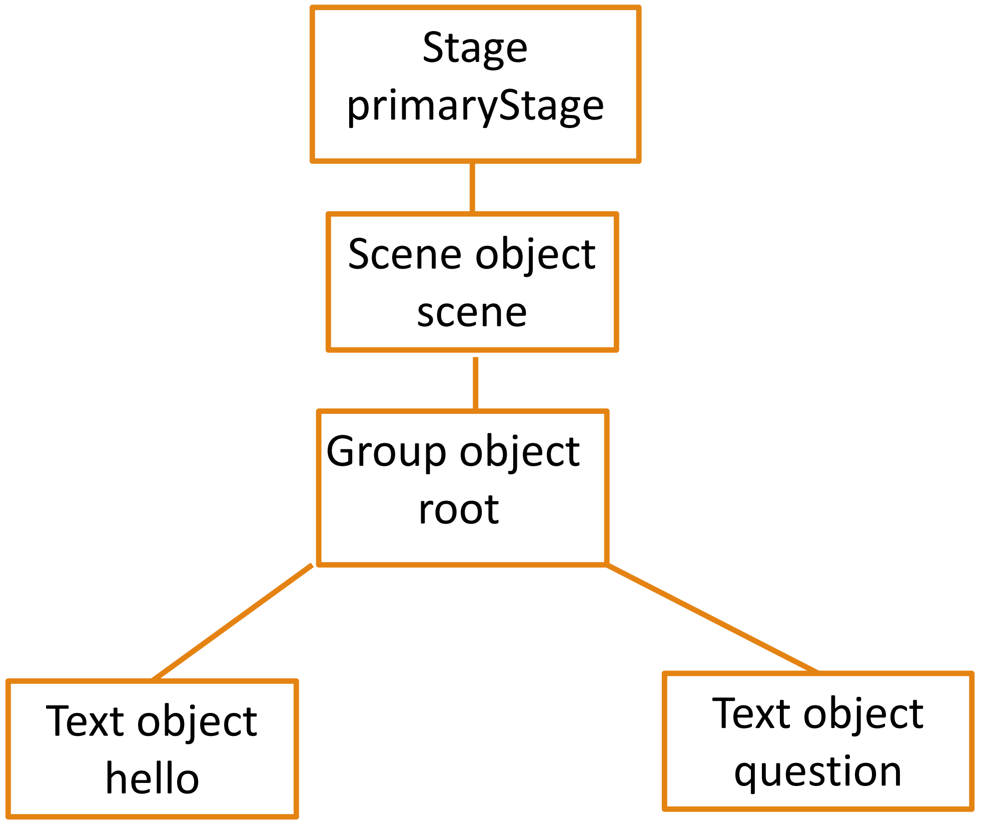
continued

```
        primarySta
        primarySta
        primarySta
    }

    //-----------                                    -----------------
    //  Launches t                                   t required
    //  in IDEs that launch JavaFX applications automatically.
    //----------------------------------------------------------------
    public static void main(String[] args)
    {
        launch(args);
    }
}
```



```
Hello, JavaFX!

        How's it going?
```

HelloJavaFX



Basic Shapes

JavaFX shapes are represented by classes in the javafx.scene.shape package

A line segment is defined by the `Line` class, whose constructor accepts the coordinates of the two endpoints:

```
Line(startX, startY, endX, endY)
```

For example:

```
Line myLine = new Line(10, 20, 300, 80);
```

Basic Shapes

A rectangle is specified by its upper left corner and its width and height:

```
Rectangle(x, y, width, height)
```

```
Rectangle r = new Rectangle(30, 50, 200, 70);
```

A circle is specified by its center point and radius:

```
Circle(centerX, centerY, radius)
```

```
Circle c = new Circle(100, 150, 40);
```

Basic Shapes

An ellipse is specified by its center point and its radius along the x and y axis:

```
Ellipse(centerX, centerY, radiusX, radiusY)
```

```
Ellipse e = new Ellipse(100, 50, 80, 30);
```

Shapes are drawn in the order in which they are added to the group

The stroke and fill of each shape can be explicitly set

```java
//********************************************************************
//  Einstein.java       Author: Lewis/Loftus
//
//  Demonstrates the use of various shape classes.
//********************************************************************

import javafx.application.Application;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;
import javafx.scene.text.Text;
import javafx.stage.Stage;

public class Einstein extends Application
{
    //--------------------------------------------------------------------
    //  Creates and displays several shapes.
    //--------------------------------------------------------------------
    public void start(Stage primaryStage)
    {
        Line line = new Line(35, 60, 150, 170);

        Circle circle = new Circle(100, 65, 20);
        circle.setFill(Color.BLUE);
```

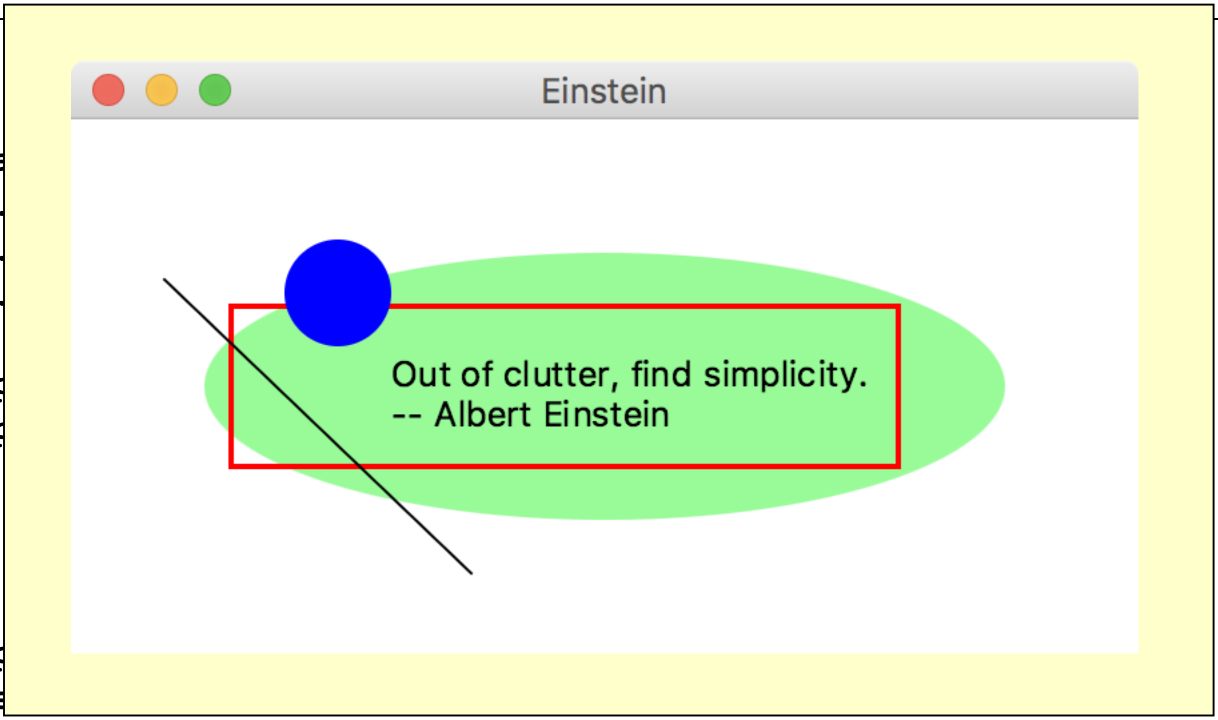
continued

continued

```
        Rectangle rect = new Rectangle(60, 70, 250, 60);
        rect.setStroke(Color.RED);
        rect.setStrokeWidth(2);
        rect.setFill(null);

        Ellipse ellipse = new Ellipse(200, 100, 150, 50);
        ellipse.setFill(Color.PALEGREEN);

        Text quote = new Text(120, 100, "Out of clutter, find " +
                "simplicity.\n-- Albert Einstein");

        Group root = new Group(ellipse, rect, circle, line, quote);
        Scene scene = new Scene(root, 400, 200);

        primaryStage.setTitle("Einstein");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    //  We will typically exclude the main method. Use it to launch
    //  the application if needed.
}
```

continued

```
        Recta
        rect.
        rect.
        rect.

        Ellip
        ellip

        Text


        Group
        Scene
```



```
        primaryStage.setTitle("Einstein");
        primaryStage.setScene(scene);
        primaryStage.show();
    }

    //  We will typically exclude the main method. Use it to launch
    //  the application if needed.
}
```

More about Shapes

Groups can be nested within groups

Translating a shape or group shifts its position along the x or y axis

A shape or group can be rotated using the `setRotate` method

```java
//********************************************************************
//  Snowman.java       Author: Lewis/Loftus
//
//  Demonstrates the translation of a set of shapes.
//********************************************************************

import javafx.application.Application;
import javafx.stage.Stage;
import javafx.scene.Group;
import javafx.scene.Scene;
import javafx.scene.paint.Color;
import javafx.scene.shape.*;

public class Snowman extends Application
{
    //-----------------------------------------------------------------
    //  Presents a snowman scene.
    //-----------------------------------------------------------------
    public void start(Stage primaryStage)
    {
        Ellipse base = new Ellipse(80, 210, 80, 60);
        base.setFill(Color.WHITE);

        Ellipse middle = new Ellipse(80, 130, 50, 40);
        middle.setFill(Color.WHITE);
```

continued

continued

```
        Circle head = new Circle(80, 70, 30);
        head.setFill(Color.WHITE);

        Circle rightEye = new Circle(70, 60, 5);
        Circle leftEye = new Circle(90, 60, 5);
        Line mouth = new Line(70, 80, 90, 80);

        Circle topButton = new Circle(80, 120, 6);
        topButton.setFill(Color.RED);
        Circle bottomButton = new Circle(80, 140, 6);
        bottomButton.setFill(Color.RED);

        Line leftArm = new Line(110, 130, 160, 130);
        leftArm.setStrokeWidth(3);
        Line rightArm = new Line(50, 130, 0, 100);
        rightArm.setStrokeWidth(3);

        Rectangle stovepipe = new Rectangle(60, 0, 40, 50);
        Rectangle brim = new Rectangle(50, 45, 60, 5);
        Group hat = new Group(stovepipe, brim);
        hat.setTranslateX(10);
        hat.setRotate(15);
```

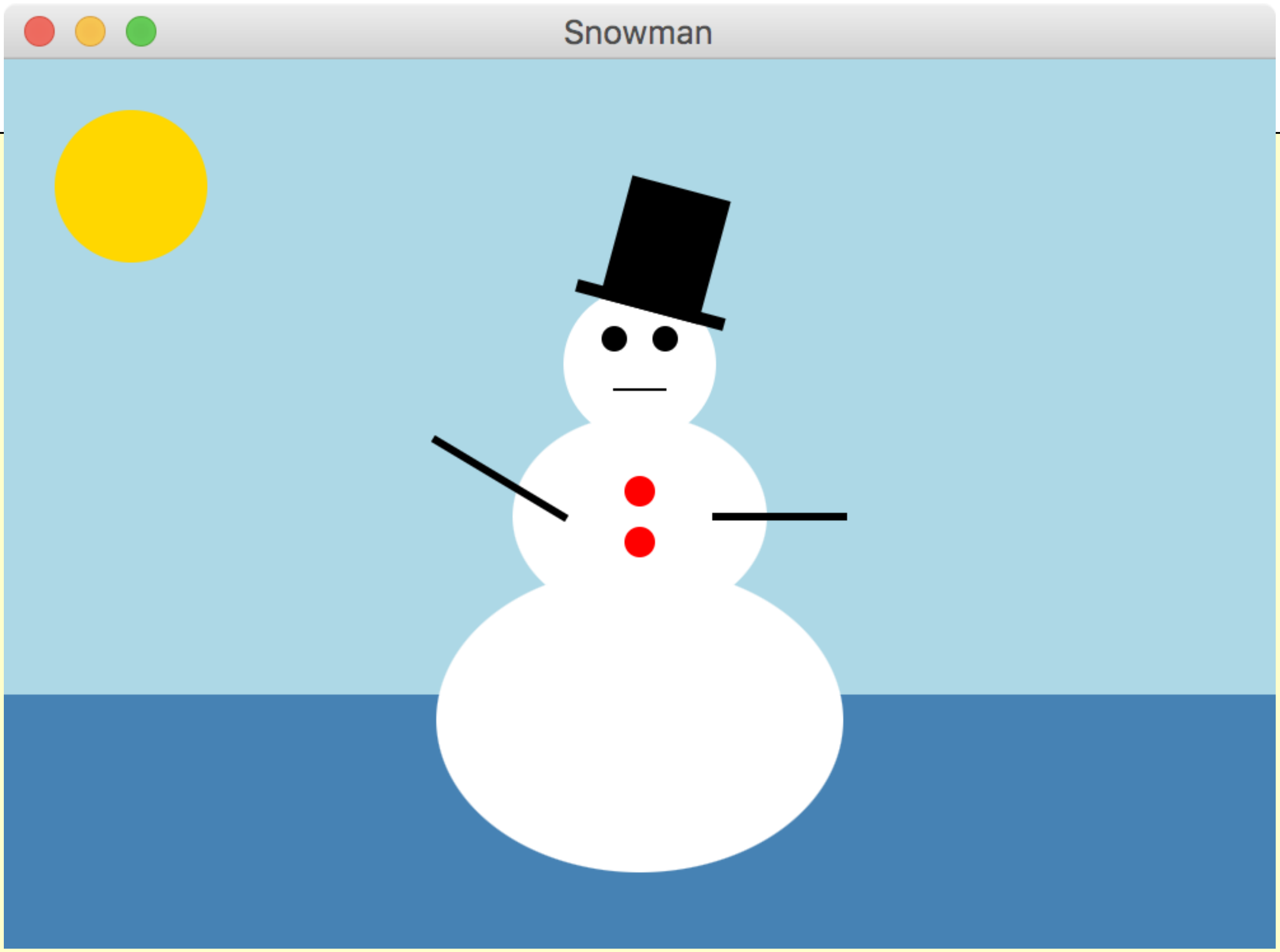
continued

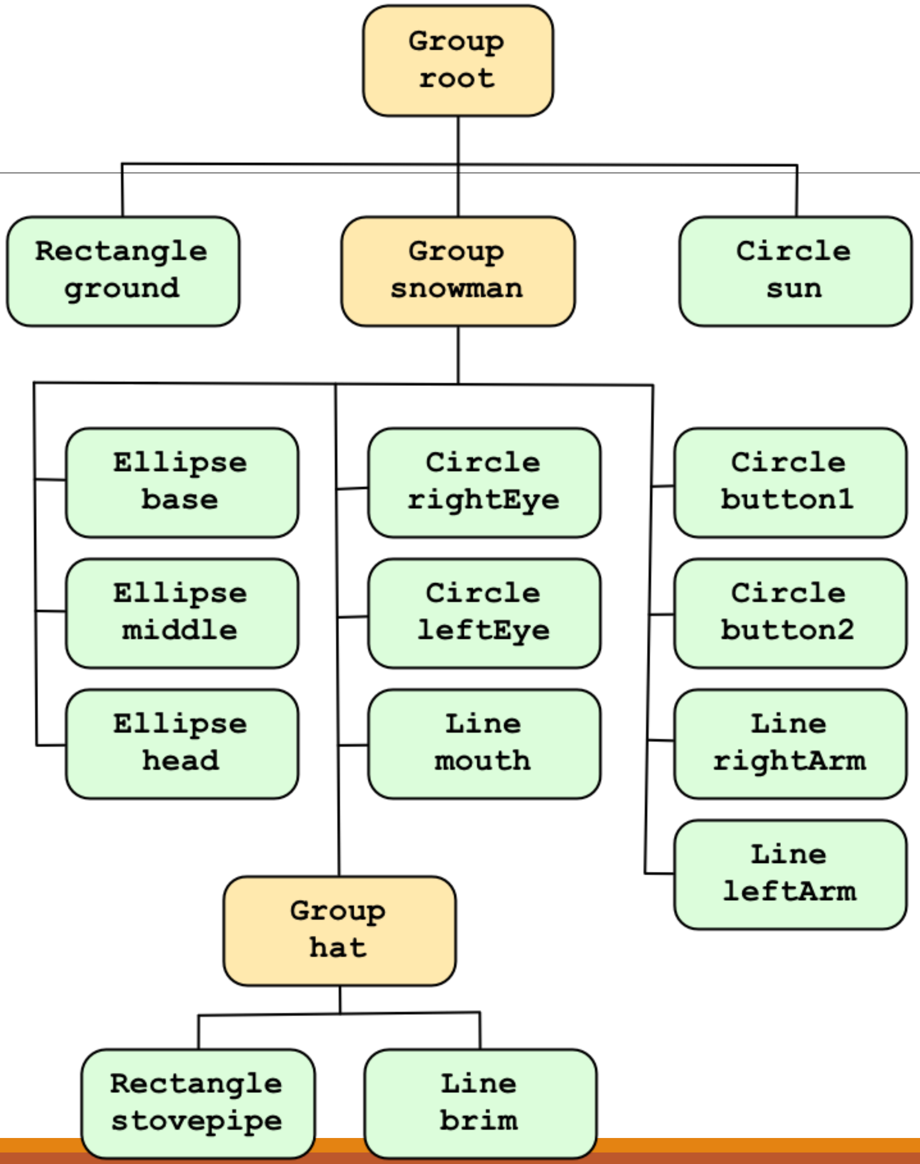
continued

```
        Group snowman = new Group(base, middle, head, leftEye, rightEye,
            mouth, topButton, bottomButton, leftArm, rightArm, hat);
        snowman.setTranslateX(170);
        snowman.setTranslateY(50);

        Circle sun = new Circle(50, 50, 30);
        sun.setFill(Color.GOLD);

        Rectangle ground = new Rectangle(0, 250, 500, 100);
        ground.setFill(Color.STEELBLUE);

        Group root = new Group(ground, sun, snowman);
        Scene scene = new Scene(root, 500, 350, Color.LIGHTBLUE);

        primaryStage.setTitle("Snowman");
        primaryStage.setScene(scene);
        primaryStage.show();
    }
}
```

continue

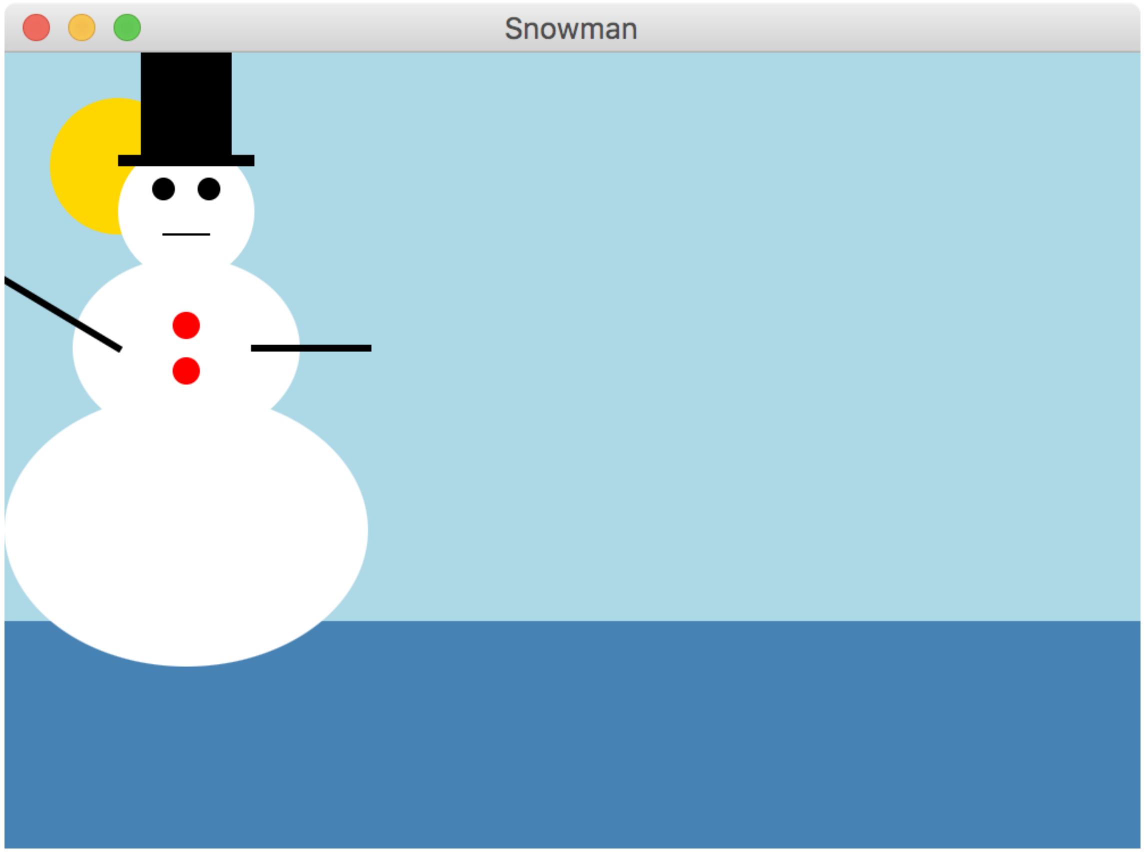
```
        }
}
```




Group root
Rectangle ground
Group snowman
Circle sun
Ellipse base
Ellipse middle
Ellipse head
Circle rightEye
Circle leftEye
Line mouth
Circle button1
Circle button2
Line rightArm
Line leftArm
Group hat
Rectangle stovepipe
Line brim

Snowman

Without translating (shifting) the snowman's position and hat:



Representing Color

A color in Java is represented by a `Color` object

A color object holds three numbers called an RGB value, which stands for Red-Green-Blue

Each number represents the contribution of that color

This is how the human eye works

Each number in an RGB value is in the range 0 to 255

The following call creates a maroon color based on 60% red, 10% green, and 0% blue:

```
    Color maroon = Color.color(0.6, 0.1, 0.0);
```

Representing Color

A color with an RGB value of 255, 255, 0 has a full contribution of red and green, but no blue, which is a shade of yellow

The static `rgb` method in the `Color` class returns a `Color` object with a specific RGB value:

```
Color purple = Color.rgb(183, 44, 150);
```

The `color` method of the Color class uses percentages, so the following call creates a maroon color based on 60% red, 10% green, and 0% blue:

```
 Color maroon = Color.color(0.6, 0.1, 0.0);
```

Representing Color

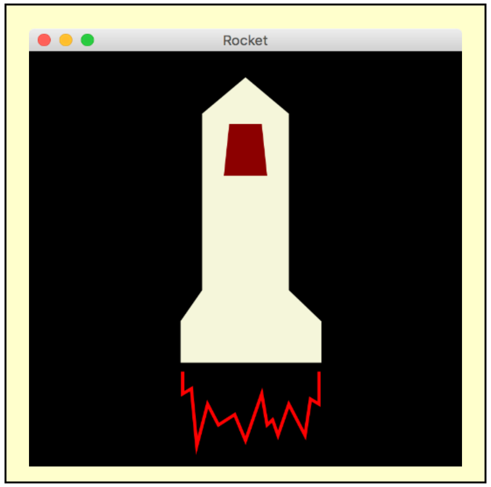
For convenience, several `Color` objects have been predefined, such as:

`Color.BLACK`	0, 0, 0
`Color.WHITE`	255, 255, 255
`Color.CYAN`	0, 255, 255
`Color.PINK`	255, 192, 203
`Color.GRAY`	128, 128, 128

See the online documentation of the `Color` class for a full list of predefined colors

Drawing Polygons and Polylines

```
public class Rocket extends Application
{
    //--------------------------------------------------------------------
    //  Displays a rocket lifting off. The rocket and hatch are polygons
    //  and the flame is a polyline.
    //--------------------------------------------------------------------
    public void start(Stage primaryStage)
    {
        double[] hullPoints = {200, 25, 240, 60, 240, 230, 270, 260,
            270, 300, 140, 300, 140, 260, 160, 230, 160, 60};

        Polygon rocket = new Polygon(hullPoints);
        rocket.setFill(Color.BEIGE);

        double[] hatchPoints = {185, 70, 215, 70, 220, 120, 180, 120};
        Polygon hatch = new Polygon(hatchPoints);
        hatch.setFill(Color.MAROON);
        double[] flamePoints = {142, 310, 142, 330, 150, 325, 155, 380,
            165, 340, 175, 360, 190, 350, 200, 375, 215, 330, 220, 360,
            225, 355, 230, 370, 240, 340, 255, 370, 260, 335, 268, 340,
            268, 310};

        Polyline flame = new Polyline(flamePoints);
        flame.setStroke(Color.RED);
        flame.setStrokeWidth(3);
```

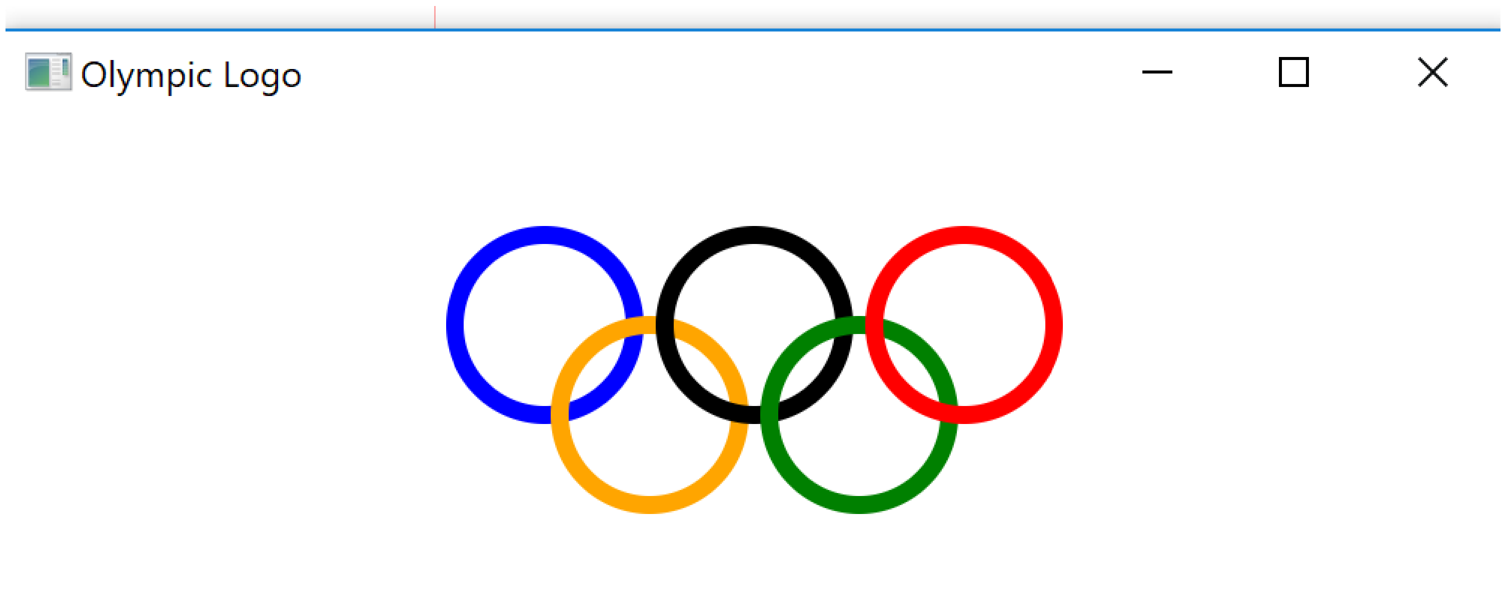


Group Exercises

EX: 3.16

EX: 3.17

PP: 3.11


Olympic Logo

Assignments for Classes 8 and 10

Class 8

- Study HelloJavaFX, Einstein, Snowman
- Programming Project Problem #3
- (This is an individual assignment!)
- Study for Quiz #1

Class 10

- Read 4.1, 4.2, 4.3