

Objectives for this class meeting

- Cover further topics on 2D Graphics
 - we will need this for designing our game



Basic Graphics

- The Java Tutorials, Trail: 2D Graphics
- http://docs.oracle.com/javase/tutorial/2d/index.html



Shape Primitives

4



Shape Primitives

5

ref: http://java.sun.com/developer/technicalArticles/GUI/java2d/java2dpart1.html







Instantiating a Shape object

The name of the class is weird – there is a dot in the middle of it.

Rectangle2D.Double is a subclass of the class Rectangle2D

Rectangle2D is a subclass of the class Shape

Neither Shape nor Rectangle2D have constructors



"When a parent is expected, a child is accepted" (Ch 9)

This is the "substitutability principle"



Rounded Rectangle =? Ellipse

L08Ex1

what value of arcX and arcY would make the rectangle so rounded that it becomes an ellipse?

use GradientPaint (from last lecture) L08Ex2



Predict the Result

```
LO8Ex3

Point anchor1 = new Point(0, dimY / 2);

Point anchor2 = new Point(dimX, dimY / 2);

Point ctrl1 = new Point(dimX / 4, 0);

Point ctrl2 = new Point(3 * dimX / 4, dimY );

Shape s = new CubicCurve2D.Double(

anchor1.x, anchor1.y,

ctrl1.x, ctrl1.y,

ctrl2.x, ctrl2.y,

anchor2.x, anchor2.y);
```



Text

L08Ex4

```
text: drawString(String, int, int)
Point anchor1 = new Point(0, dimY / 2);
String str = "Hello";
graphicsObject.drawString(str, anchor1.x, anchor1.y);
```



Text

11

12

L08Ex5

```
Font font = new Font("Serif", Font.PLAIN, 48);
graphicsObject.setFont(font);
graphicsObject.drawString(str, anchor1.x, anchor1.y);
```

the font family, the font face and the font size.



Size of Rendered Text?

Suppose we need to know the space that will be occupied by some text.

This will depend on the font family, font face and the font size.

We can query the graphics object to determine the logical bounds of string.

The logical bounds contain the **origin**, **ascent**, **advance**, and **height**, which includes the **leading**.



Font Concepts

13

- *Ascent* is the distance from the baseline to the ascender line.
- *Descent* is the distance from the baseline to the descender line.
- *Leading* is the recommended distance from the bottom of the descender line to the top of the next line.



Size of Rendered Text?

// ... assigned already: graphicsObject

```
String str = "Hello";
Font font = new Font("Serif", Font.PLAIN, 48);
FontRenderContext frc = graphicsObject.getFontRenderContext();
Rectangle2D dim = font.getStringBounds(str, frc);
```



For next class

Read:

15

Lesson: Working with Images http://docs.oracle.com/javase/tutorial/2d/images/index.html

