Graphics2D

3-23-2006

Opening Discussion

- What did we talk about last class?
- Why are so many exceptions unchecked?
- ACM High School programming competition: our practice competition will be a week from today. I will give extra credit for both doing the practice competition and for helping with the competition on April 8th.

Coding with Exceptions

- Last time we finished with you doing some coding for exceptions writing a reverse polish calculator. This used a stack and had to deal with input that was numbers or operators.
- Let's go back to that and look at some solutions.

Custom Drawing in Java

- Today's topic is how to draw things to the screen in Java. For this we will focus on the Java2D libraries and the Graphics2D class.
- There are a few steps to follow to get what we draw on screen though. The general process is to make a class that extends JComponent and then override the paintComponent method of that class.
- This method will be passed Graphics object that you can draw to. That is what we will focus on.
- As you have also seen, we can get a graphics object for an Image as well.
- Let's write code to bring up a frame and add our custom drawing component to it.

Graphics and Graphics2D

- The actual drawing is done to a Graphics or Graphics2D object. Graphics2D is a subtype of Graphics. In Swing applications when you get a Graphics object it will be of the type Graphics2D.
- Let's look in the API that these classes. We will focus on Graphics2D because it has significantly more functionality and allows us to produce cool effects with fairly little effort.

Drawing and Filling

- The two main methods for drawing in the Graphics2D class are draw and fill. Both take an object of type Shape. The draw method only does the outline of the shape while the fill method fills in the entire shape.
- In the Graphics class there are specific methods for drawing certain specific shapes like rectangles and ovals. Sometimes these can be simpler to use and they still work.
- The methods in Graphics work with integer arguments while the Graphics2D has the added flexibility of working with floating point numbers.

Shapes

- A big part of the power of fill and draw comes from the flexibility of the Shape interface.
- Java provides a number of different shapes for you in the java.awt.geom package. Let's go look at those.
- You also have the ability to put together more complex shapes on your own using the GeneralPath and Area classes.
- Let's go in and make our code so it draws a shape or two. Do one draw and one fill to see how they work.

Paint

- By default a new Graphics object will draw in black. We can make it draw with other colors or more complex patterns as well.
- Graphics2D has a method called setPaint that allows us to tell the object how things should be filled in.
- There are 3 main subclasses of Paint. Color is the simplest. There are also GradientPaint and TexturePaint subclasses that can add a lot to your drawing with rather little effort.
- Let's augment our drawings so that they use each of these 3 type of paints.

Stroke

- When you call fill, Java2D actually builds an outline by using a stroke on the shape and then does fill on that outline. As we'll see later, almost every operation in Java2D really comes down to a fill.
- The Graphics2D class has a setStroke method. The only implementation of the Stroke interface that is provided is BasicStroke. However, that class is very capable of doing most anything you need.
- Lets go look at the BasicStroke class then add interesting outlines to our shapes that have different colors than the shapes.

Drawing Programing

- With the rest of the time I want us to write code for our drawing program. We haven't worked on it much, but now we have the ability to actually draw stuff once the foundation for that is set up.
- Next class we'll learn about other abilities of Java2D that will make the drawing program even more interesting.

Minute Essay

- How might you be able to use these custom drawing abilities to enhance your game?
- Remember that design #5 is due today.
- I will be putting up a newer version of “From C to Java” with material related to the topics for today and tomorrow.