Program #2 README

Apr 24, 2014

1. ShapeViewer notes

The ShapeViewer class displays a list of shapes and allows the user to select shapes.

- ➢ Shapeviewer is-a JPanel
- ➢ ShapeViewer has-a Shape... in fact, it has an ArrayList of Shape objects

Q&A:

- Four questions about your shape list:
    1. Where is your shape list declared? **Answer:** It's a class variable.
    2. Where is it created? **Answer:** In the ctor.
    3. How can people using this class add Shape objects to be viewed? **Answer:** You need a method in ShapeViewer: addShape(s)?
    4. Where is the list used to draw the Shape objects? **Answer:** In paintComponent() which Java calls to redraw the JPanel

- How can the user select a Shape? **Answer:** You need a mouse listener to handle it.

- In the listener, how can I tell what Shape is selected? **Answer:** You can get the (x,y) location of the click from the mouse event.

- From the (x,y), how do I select a Shape? **Answer:** Pick the closest one.

- How do I know which is closest? **Answer:** Foreach Shape in the list, measure the distance from the click (x, y) to the center of the shape. Manhattan distance = abs(x1-x2) + abs(y1-y2).

- How do you tell the user a Shape has been selected? **Answer:** An easy way is to draw its name in the middle of the shape. Or, you could give it a red border or something like that.

2. RandomHelper

I have placed a class called RandomHelper on the k: drive to, um, help you generate random things for your shapes. The methods are static, ala Java's Math class:

- `public static Point randomPoint()` - returns a random (x,y) location
- `public static int randomSize()` - create a random shape size
- `public static Color randomColor()` - create a random color

There are default ranges for these random value or you can set min/max values with methods:
Set a range of acceptable locations with:

- `public static void setFrameWidth( int w)`
- `public static void setFrameHeight( int h)`

Set a range of a range of acceptable sizes with:
- `public static void setMinSize( int size)`
- `public static void setMaxSize( int size)`

3. ShapeConsole notes

NOTE: Do your console last. Wait until your viewer is largely working.

The ShapeConsole class allows the user to send commands to the viewer, i.e. add Shapes.
- ➔ ShapeConsole has-a ShapeFactory so that he can create Shape object. In fact, he's got a list of them.
- ➔ ShapeConsole has-a ShapeViewer so that he can tell the viewer about the new Shapes

Console commands are:
- add - adds a shape, more on this later
- clear - clear all shapes from the viewer
- deselect - deselect all shapes in the viewer
- shapes - print a console report listing all shapes in the viewer, sorted by name
- area - print a console report listing all **selected** shapes, sorted by area
- exit - System.exit(0);

The add command is the only one with more user choices (user typing is **bold**):

```
console> add
     What shape? circle
     How many? 3
     Name? Bill
Adding 3 circles: Bill1, Bill2, Bill3
```

This is optional, but I have defaults if the user just hits enter for any of these choices: default shape = randomly-chosen shape, number of shapes = 1, name = the shape type. Here's an example:

```
console> add
     What shape?
     How many?
     Name?
Adding 1 square: Square1
```

We'll cover sorting in lecture next week.