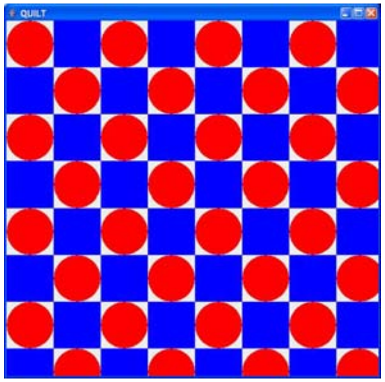
CompSci 4
Java 4
Nov 29 , 2007

Prof. Susan Rodger



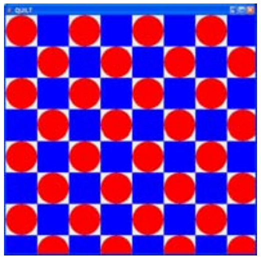
Announcements

- Assignment 7 due Friday night
- Demos on Tuesday and some Thursday
  - You must attend on both Tues and Thurs
- Review for final next Thursday

- Classwork Today: Quilt

Problem: Draw a 2-item Quilt

- Quilt has two items: red circle, blue square
- Each row alternate items across
- Each col alternate items down



Write a complete program

- Project is CPS4Sec1Quilt or CPS4Sec2Quilt
- Also doing graphics today with Java
- Two classes in the project
  - Main.java - DON’T MODIFY THIS FILE
  - Canvas.java – ONLY MODIFY ONE METHOD - paintComponent

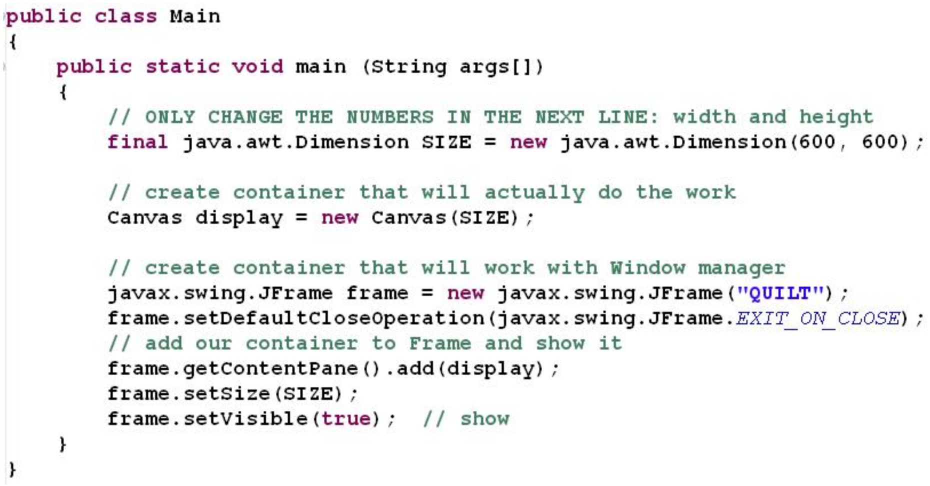
Java Program

- Has lots of classes
- One class has a method named “main”
  - Starting point, like world.myfirstmethod in Alice

main method of Quilt program

- Creates a canvas of size 600 x 600 pixels you can draw on
- Lots of Graphics setup stuff – ignore for now

Quilt main method

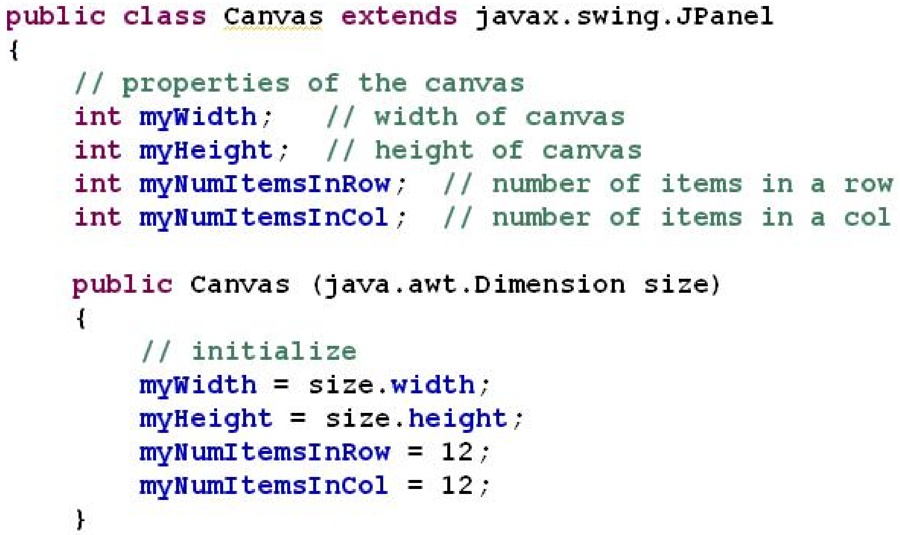


```
public class Main
{
    public static void main (String args[])
    {
        // ONLY CHANGE THE NUMBERS IN THE NEXT LINE: width and height
        final java.awt.Dimension SIZE = new java.awt.Dimension(600, 600);

        // create container that will actually do the work
        Canvas display = new Canvas(SIZE);

        // create container that will work with Window manager
        javax.swing.JFrame frame = new javax.swing.JFrame("QUILT");
        frame.setDefaultCloseOperation(javax.swing.JFrame.EXIT_ON_CLOSE);
        // add our container to Frame and show it
        frame.getContentPane().add(display);
        frame.setSize(SIZE);
        frame.setVisible(true);  // show
    }
}
```

Canvas Class– Part 1



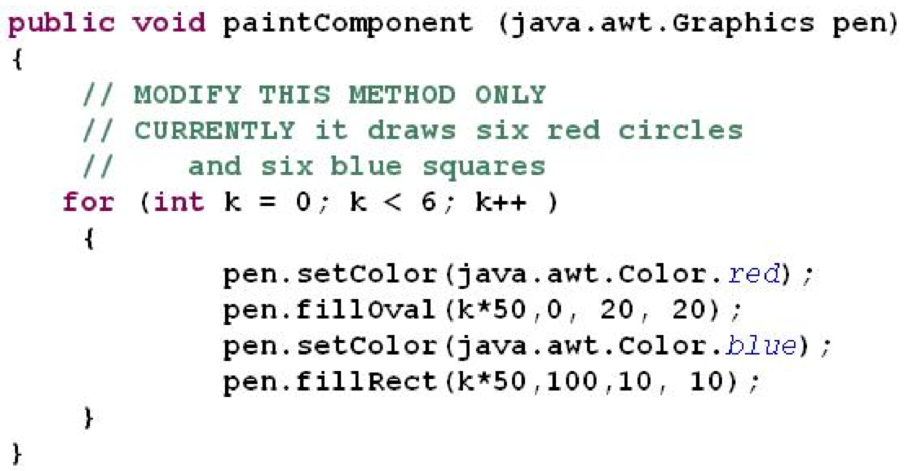
```
public class Canvas extends javax.swing.JPanel
{
    // properties of the canvas
    int myWidth;   // width of canvas
    int myHeight;  // height of canvas
    int myNumItemsInRow;  // number of items in a row
    int myNumItemsInCol;  // number of items in a col

    public Canvas (java.awt.Dimension size)
    {
        // initialize
        myWidth = size.width;
        myHeight = size.height;
        myNumItemsInRow = 12;
        myNumItemsInCol = 12;
    }
```

Canvas Class – Part 1

- First list class variables (like properties in Alice) and give their types
- Special method with **same name as class**
  - called a constructor
  - Give initial values to all class variables here
  - (see previous slide: `public Canvas(…)`)

Canvas Class - Part 2



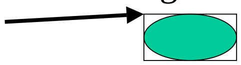
```
public void paintComponent (java.awt.Graphics pen)
{
    // MODIFY THIS METHOD ONLY
    // CURRENTLY it draws six red circles
    //    and six blue squares
   for (int k = 0; k < 6; k++ )
    {
            pen.setColor(java.awt.Color.red);
            pen.fillOval(k*50,0, 20, 20);
            pen.setColor(java.awt.Color.blue);
            pen.fillRect(k*50,100,10, 10);
    }
}
```

Canvas Class - Part 2

- paintComponent method – special method for graphics – automatically called
- Pass in a pen that draws the picture
- Use “pen.setColor(java.awt.Color.red)” to change the color of the pen to a red pen
  - Doesn’t draw anything, just sets the color
- Other Colors: blue, green, yellow, black, cyan, pink, orange, magenta

Canvas Class – Part 2 (cont)

- Draw an oval with the pen using pen.fillOval(x,y,width, height);
  - Draws a filled oval with top left corner at position (x,y) and with width and height

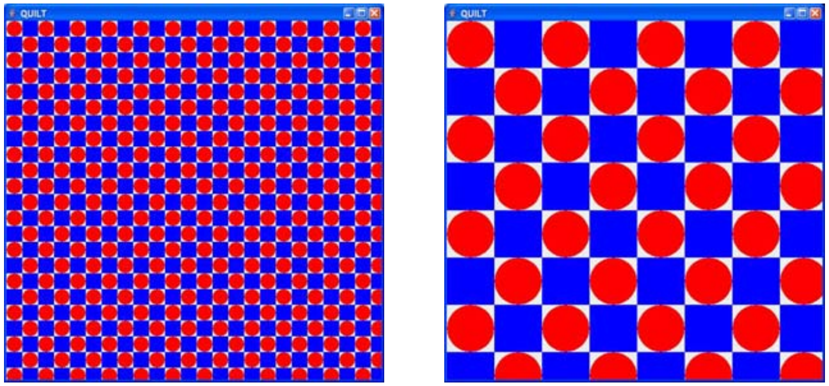


- Draw a rectangle with the pen using pen.fillRect(x,y,width,height);

Classwork: Draw a Quilt

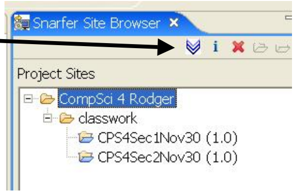
- Pick two shapes and two colors and draw them alternating on a quilt
- Have the code flexible so you can change the number of shapes per line and the quilt will adjust to fit them on the line (it is ok if they don't fit exactly in the canvas)
- myNumItemsInRow = ?
- myNumItemsInCol = ?

Example



Getting Started

- Project started for you, "snarf" it and just modify paintComponent to draw the quilt
- In Eclipse, select "ambient", then "Download (snarf) a project"
- Must connect to CompSci 4 Snarf site
- Click here and add this snarf site


Snarfer Site Browser

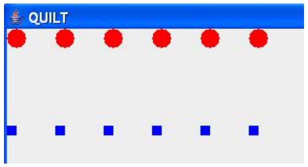
Project Sites

CompSci 4 Rodger
- classwork
  - CPS4Sec1Nov30 (1.0)
  - CPS4Sec2Nov30 (1.0)

http://www.cs.duke.edu/courses/fall07/cps004/snarf/

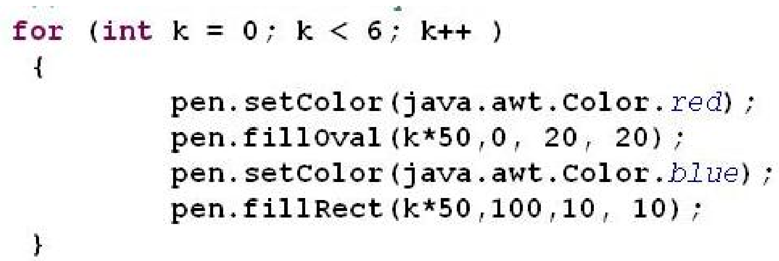
Getting Started (cont)

- Download folder CPS4SecXQuilt
- Run program to see what it does
  - Right Click on Main.java (has main method)
  - Select "Run As" then "Java Application"
  - Should see:



Look at Loop again

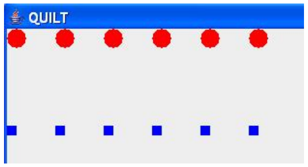
- This prints circles with top left corner at: (0,0), (50, 0), (100, 0), (150, 0) …
- This prints rectangles with top left corner at: (0,100), (50, 100), (100, 100), (150, 100) …



```
for (int k = 0; k < 6; k++ )
 {
          pen.setColor(java.awt.Color.red);
          pen.fillOval(k*50,0, 20, 20);
          pen.setColor(java.awt.Color.blue);
          pen.fillRect(k*50,100,10, 10);
 }
```

Step 1:

- Look at the PaintComponent Method
- There are two copies of the loop, one is commented out so you can refer back to it
- Pick two shapes and colors (can be the same shape if you want) and **modify the code to print out those shapes and colors.**
- I picked red circle and blue rectangle

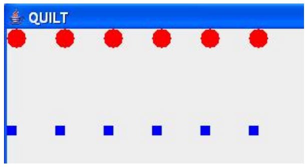


Note

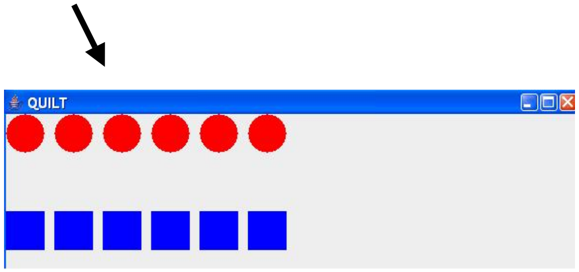
- After each step, make a copy of your loop code and comment one copy out. Add a comment to say what the loop does. Then you can go back and see how you did a previous step….

Step 2:

- Change the size of your two objects to make them bigger (with height and width set to 40)

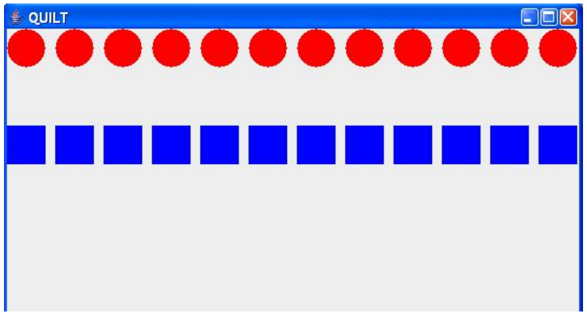


Note we are printing them with 10 units of blank space between two



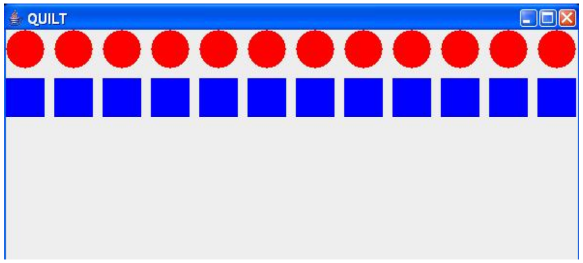
Step 3:

- Now print circles and rectangles all the way across the canvas. The width of the canvas is `myWidth`



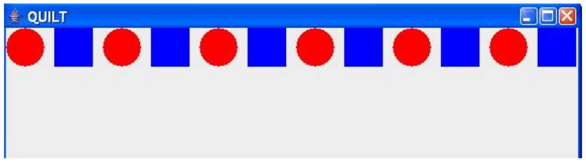
Step 4:

- Now move the 2cd object up so it is 10 units away from the 1st object.



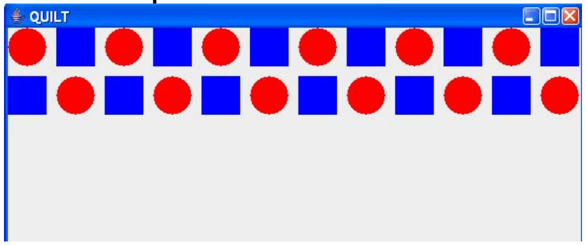
Step 5:

- Change so that the two objects are both printed on the same line. Print one object first every 100 units. Print the second object starting at position (50,0) and then one every 100 units.
- Note: use the same loop, you only have to change the (x,y) value for each



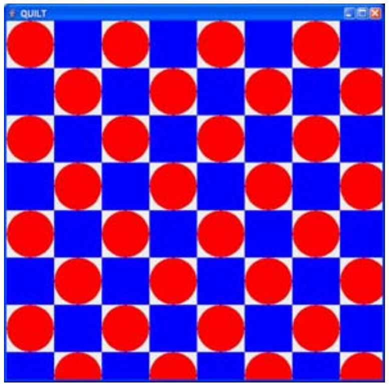
Step 6

- Print two rows, one starting with one object and the other row starting with the other object.
- Copy the four lines inside your for loop so you have 8 lines in your for loop, **fix** the last four lines to print the second row



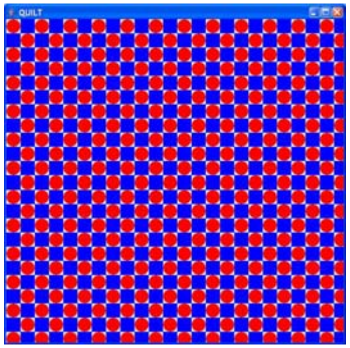
Step 7

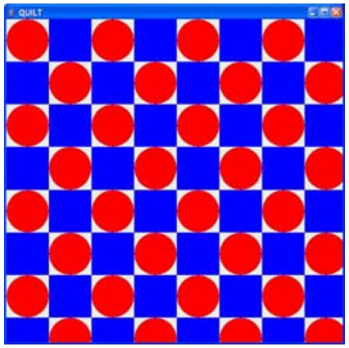
- Add another loop around the whole loop to repeat these two rows to complete the quilt
- (for int j=0; j < ? ; j++)
- How do you use “j”?
- Think about the y part of (x,y) position
- Once that works, then tighten it up to get rid of the white space.



Step 8

- Make quilt shapes adjustable….
- When you modify myNumItemsInRow and myNumItemsInCol, quilt should adjust to fill the canvas so you get different sizes….
- But you must use those variables somewhere!





Quilt Requirements

- Must use two shapes (two circles, two squares, or one circle and one square)
- Must use two colors
- Must make quilt adjustable. When you modify myNumItemsInRow and myNumItemsInCol, quilt should adjust to fill the canvas
- Ok if edge of quilt is not correct
- Submit work in classNov29 folder