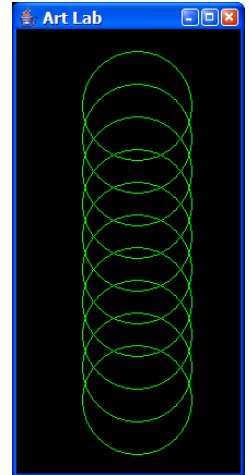


while loops in ArtLab

pattern of circles

```
1. int counter;  
2. g2.setPaint ( green );  
3. width = 100;  
4. height = 100;  
5. counter = 0;  
6. while (counter < 10)  
7. {  
8.   newShape();  
9.   x = 200;  
10.  y = 20 + counter * 30;  
11.  g2.drawOval(x, y,  
               width, height);  
12.  counter = counter + 1;  
13. }
```



exercises

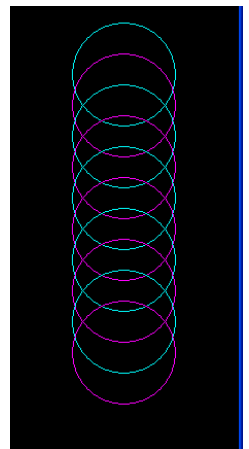
what would happen if:

- line 6 were changed to: `while (counter < 8)`
- line 5 were changed to: `counter = 5;`

ifs and whiles combined

Here is what happens if line 11 is replaced with:

```
11a. if (counter%2==0)  
11b. { g2.setPaint(cyan); }  
11c. else  
11d. { g2.setPaint(magenta); }  
11e. g2.drawOval(x, y, width, height);
```

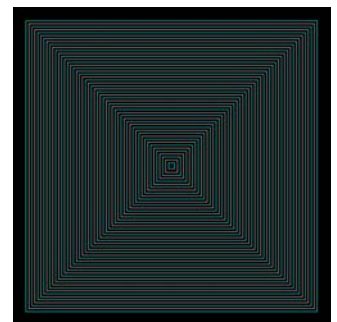
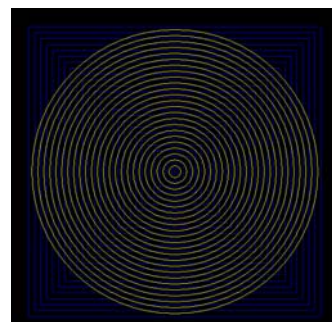
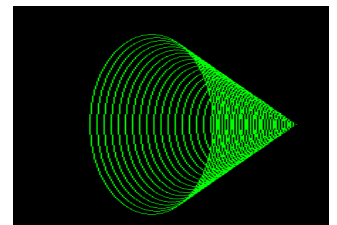


more exercises

how might you:

- arrange the circles in a *horizontal* line?
- arrange the circles along a *diagonal* line?
- arrange the circles to be *nested* in each other?

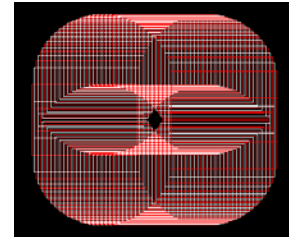
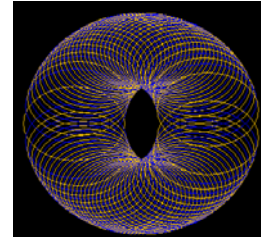
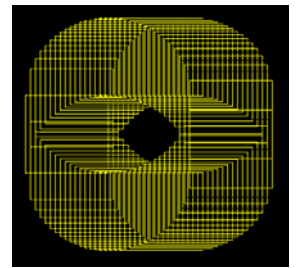
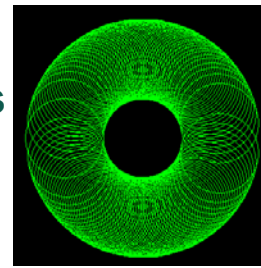
nested shapes and more...



sample code (top right picture)

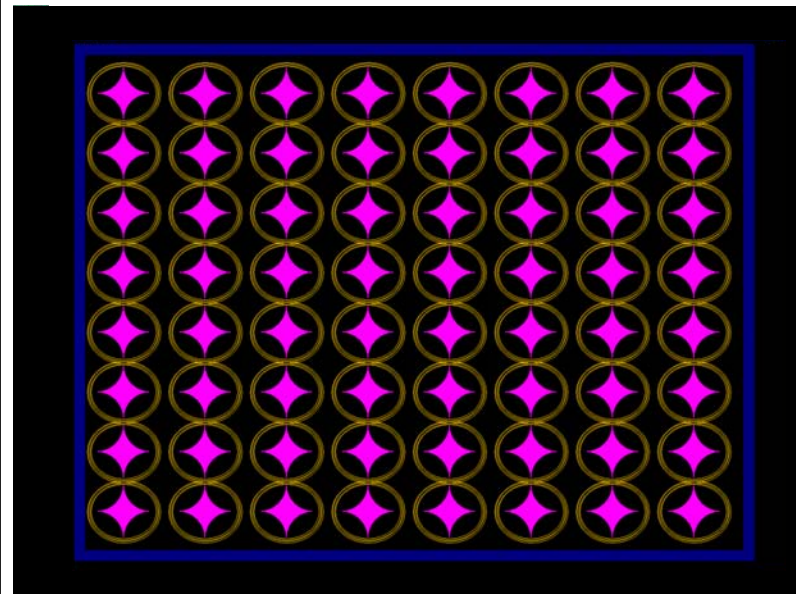
```
counter = 0;
while (counter < 30)
{
    newShape ();
    x = 300 - counter * 5 ;
    y = 280 - counter * 5;
    width = counter * 3;
    height = counter * 10;
    g2.drawOval(x, y, width, height);
    counter = counter + 1;
}
```

shapes in circles



sample code (top right picture)

```
counter = -100;
while (counter < 100)
{
    newShape();
    x = 150 + counter;
    y = (int) Math.sqrt(10000 - (x-150)*(x-150));
    width = 100;
    height = 100;
    if ( (counter % 2) == 0) { g2.setPaint (yellow); }
    else { g2.setPaint (black); }
    g2.drawRect(x, 120 + y, width, height);
    g2.drawRect(x, 120 - y, width, height);
    counter = counter + 3;
}
```



```
int counter, counter2;
counter = 0;
while (counter < 8)
{ counter2 = 0;
  x = 50 + counter*150;
  while (counter2 < 8)
  { y = 50 + counter2*110;
    g2.setPaint (magenta);
    g2.fillRect(x, y, 100, 100);
    g2.setPaint (black);
    g2.fillOval(x-50, y-50, 100, 100);
    g2.fillOval(x+50, y-50, 100, 100);
    g2.fillOval(x-50, y+50, 100, 100);
    g2.fillOval(x+50, y+50, 100, 100);
    counter2 = counter2 + 1; }
  counter = counter+1; }
```

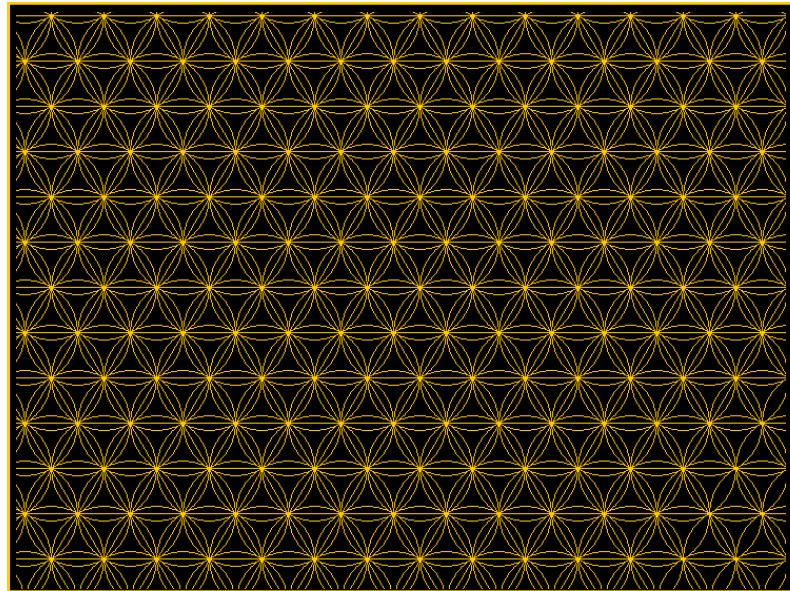
```
g2.setPaint (orange); counter = 0;
while (counter < 8)
{ counter2 = 0;
  x = 50 + counter*150;
  while (counter2 < 8)
  { y = 50 + counter2*110;
    g2.drawOval(x-10, y, 120, 100);
    g2.drawOval(x-12, y-2, 125, 105);
    g2.drawOval(x-15, y-5, 130, 110);
    g2.drawOval(x-17, y-7, 135, 115);
    counter2 = counter2 + 1; }
  counter = counter+1; }
g2.setPaint (blue); counter=0;
while (counter < 10)
{ g2.drawRect (10+counter*2,
  10+counter*2, 1250-4*counter,
  950-4*counter);
  counter = counter+1; }
```

resources

- cs015 web page at Brown University
 - <http://www.cs.brown.edu/courses/cs015/>
 - click on "lectures" to see course slides and draft of book by Kate Sanders and Andries van Dam
- java.sun.com
 - java compilers (free) and tutorials
 - 2D graphics tutorial at <http://java.sun.com/docs/books/tutorial/2d/>

more resources

- note from class at Dartmouth on art and math:
<http://www.dartmouth.edu/~matc/math5.pattern/syllabus.html>
- simple lessons on geometry in Islamic art:
<http://www.askasia.org/frclasrm/lessplan/I000030.htm>



project idea

Using tools of the java ArtLab, figure out how to generate traditional geometric designs; explain your goals and approach.

resources

- programming environment designed by the MIT media lab for artists: <http://processing.org/>
- article about René Jodoin:
<http://www.awn.com/mag/issue5.07/5.07pages/robinsonjodoin.php3>