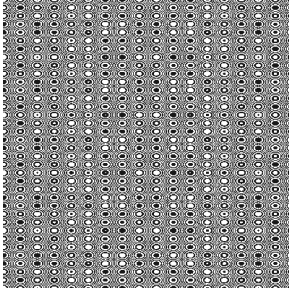


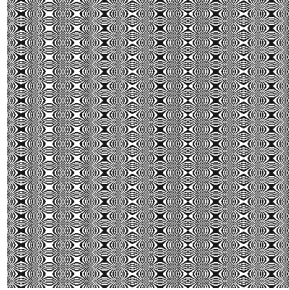
## CMSC 157: Object-Oriented Programming Workshop

### Assignment 1: Moire

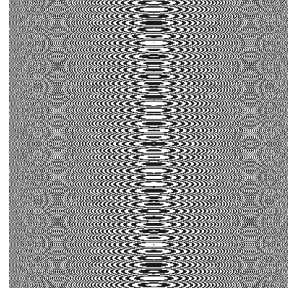
Due by Class (1:30pm) September 5, 2016



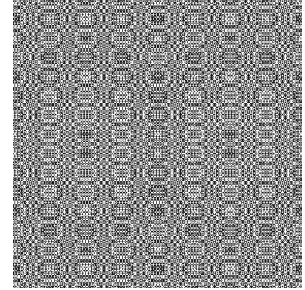
$s = 0.04250$   
 $t = 0.08750$



$s = 0.97400$   
 $t = 0.05000$



$s = 0.99750$   
 $t = 0.89250$



$s = 0.25500$   
 $t = 0.40750$

This assignment serves as a warm-up for programming with Processing and Java. The goal is to create a program that creates varying digital moiré images. Our approach is taken from Dewdney's 1976 "Computer Recreations" column in **Scientific American**.

#### Connett's Algorithm

Assuming  $x, y$  are the pixel coordinates, and  $s, t$  are scaling parameters that range from 0–1:

```
for all pixels in image do
   $z \leftarrow \lfloor s \cdot x^2 + t \cdot y^2 \rfloor$ 
  if  $z$  is even then paint pixel white
  else paint pixel black
end if
end for
```

#### Processing Tips

<code>color(r,g,b)</code>	Create a color with specific red, green, blue values (0–255)
<code>color(g)</code>	Create a color with a specific gray-scale value (0–255)
<code>set(x,y,c)</code>	Set the pixel at location (x, y) to a specific color
<code>width, height</code>	Variables that hold the width & height of the window
<code>x % y</code>	Compute the remainder of dividing $x$ by $y$
<code>int(x)</code>	Create an Integer of $x$ by truncating the fractional part

#### Challenges

Try and accomplish the following items:

- translate Connett's algorithm to Processing;
- render the whole image at once (i.e., not animated);
- vary scale parameters ( $s$  &  $t$ ) using the mouse, either via `random()` function or `mouseX/Y` variables;
- save a screenshot (i.e., `saveFrame`) when a key is pressed;
- use more than two colors;
- use the entire screen (e.g. `displayWidth/Height`).

#### Learning Objectives

- Review Processing's graphics facilities.
- Review loops.

## Processing Tips

A small Processing program to get you started:

```
1 int y = 0;
2
3 void setup() {
4   size(800, 600);
5   background(224);
6 }
7
8 void draw() {
9   for (int x = 0; x < width; x++) {
10     color c = color(y%255);
11     set(x, y, c);
12   }
13   y = y + 1;
14 }
```

## Deliverable

Submitting your assignment:

1. Put a comment at the top of your programs with your name, date assignment description, and collaboration statement.

```
/**
 * Name <bard email>
 * date
 * CMSC 157
 * Project 1: Moire (And any usage information)
 * I (worked alone | collaborated) on this assignment
 *   [with (student | tutor) X]
 *   [with assistance from (website link | book | student | tutor) Z].
 */
```

2. Bring a hardcopy of your program (i.e., the source code) to class.
3. Submit a zip file of your program via Moodle. The zip file should expand into a folder named cmsc157-project1-lastname-firstname with the Processing sketch inside of that folder.