Introduction to Creative Coding with Processing

Week 2 - Drawing with algorithms

Thomas Deacon, 2019

Resources for each week available at:
https://github.com/VizRCA/intro-to-creative-coding

Topics

Week 2

- Recap wk1
- Digital Op Art
- Using maths for drawing
- Using algorithms for patterns
- Using randomness in pieces

Recap week 1

Functions

```
// set the size of display window
// width:600 height:300
size (600, 300);

// make the background white
background(255);
// or black
background(0);

// set the stroke to 10 pixels wide
strokeWeight (10);

// Arguments can be other datatypes, like text:
println ("hello"); // outputs to console
// display text at x:10, y:20
text ("hello", 10, 20);
```

Recap week 1

Syntax

```
// SEMI-COLON (;)
// All lines of code end with a semi-colon (;)
// A missed semi-colour will result in an error.
// CURLY BRACKETS ({}):
// Indicates a “block” of code, like a paragraph.
// basic 'blocks' of code are 'set up' and 'draw'.

void setup ( ) {
  // things within 'setup' happen once at startup
  println ("this is setup"); // note the ;
}

void draw ( ) {
  // things within 'draw' happen continually**
  println ("this is draw");
}

// ** unless you tell it not to
```

Recap week 1

Variables

```
// BASIC VARIABLE TYPES:
// 'int' values are whole numbers:
int score = 30;
// 'float' values are numbers that may be decimal numbers
float x_pos = 120.5;
// 'String' values are used to store text characters.
String userName = "Harry";

// Once created, the value of the variable can be used and/or
changed.
// number variables can be easily manipulated using simple
mathematical functions (+ * - /)
float num = 100;
println (num);
num = num + 20; // add 20
println (num);
num = num / 2; // divide 2.
println (num);
num = num * 4; // multiply by 4;
println (num);
num = num - 120; // minus 120
println (num);
```

Recap week 1

Conditionals

```
/* Values of variables are tested using standard comparison
operators:
 >    Greater Than
 >=   Greater Than or Equal to
 <    Less Than
 <=   Less Than or Equal to
 ==   Equal to.
*/

// If/Else statements allow an alternate piece of code to
be processed if the condition is not met.

float score = 75;

if (score < 50) {
  println ("nevermind you did not manage 50 points");
}
else {
  println ("well done you managed to score 50 or above");
}
```

Recap week 1

Loops

```
// the following is an example of a for loop
which loops 10 times.
// the variable 'i' is used to count the loop
// i starts at 0 (i=0)
// the loop is repeated whilst i is less than 10
(i<10)
// after each loop i is increased by 1 (i++)

for (int i=0; i<10; i++) {
  println (i); // outputs the value of 'i'
during each loop
}

// the result is an output of  "0 1 2 3 4 5 6 7
8 9"
```

Recap week 1

User Functions

```
void setup ( ) {
  // call the custom function 'calcSum':
  calcSum (10, 2); // outputs 20
  calcSum (5, 30); // outputs 150
}

// function outputs the sum of 2 numbers
void calcSum (float num1, float num2) {
  float number1 = num1;
  float number2 = num2;
  println (number1 * number2);
}
```

Recap week 1

Drawing Functions

```
strokeWeight (5); // set the width of the outline stroke
fill (13, 184, 216); // blue fill color
stroke (12, 83, 232);  // dark blue outline

// draw a circle at x:150, y:300
ellipse (150, 300, 120, 120);

// draw a square at x:240, y:240
rect (240, 240, 120, 120);

// triangle is 3 points (x1, y1, x2, y2, x3, y3)
triangle (450, 220, 380, 360, 520, 360);

// draw 60 lines.
// the value of i is used to set the x location of each line
// 60 lines are drawn using only a few lines of code.
for (int i=0; i<60; i++) {
  line (i*10, 20, i*10, 100);
}
```

Op Art

Op art was a major development of painting in the 1960s that used geometric forms to create optical effects.

The effects created by op art ranged from the subtle, to the disturbing and disorienting.

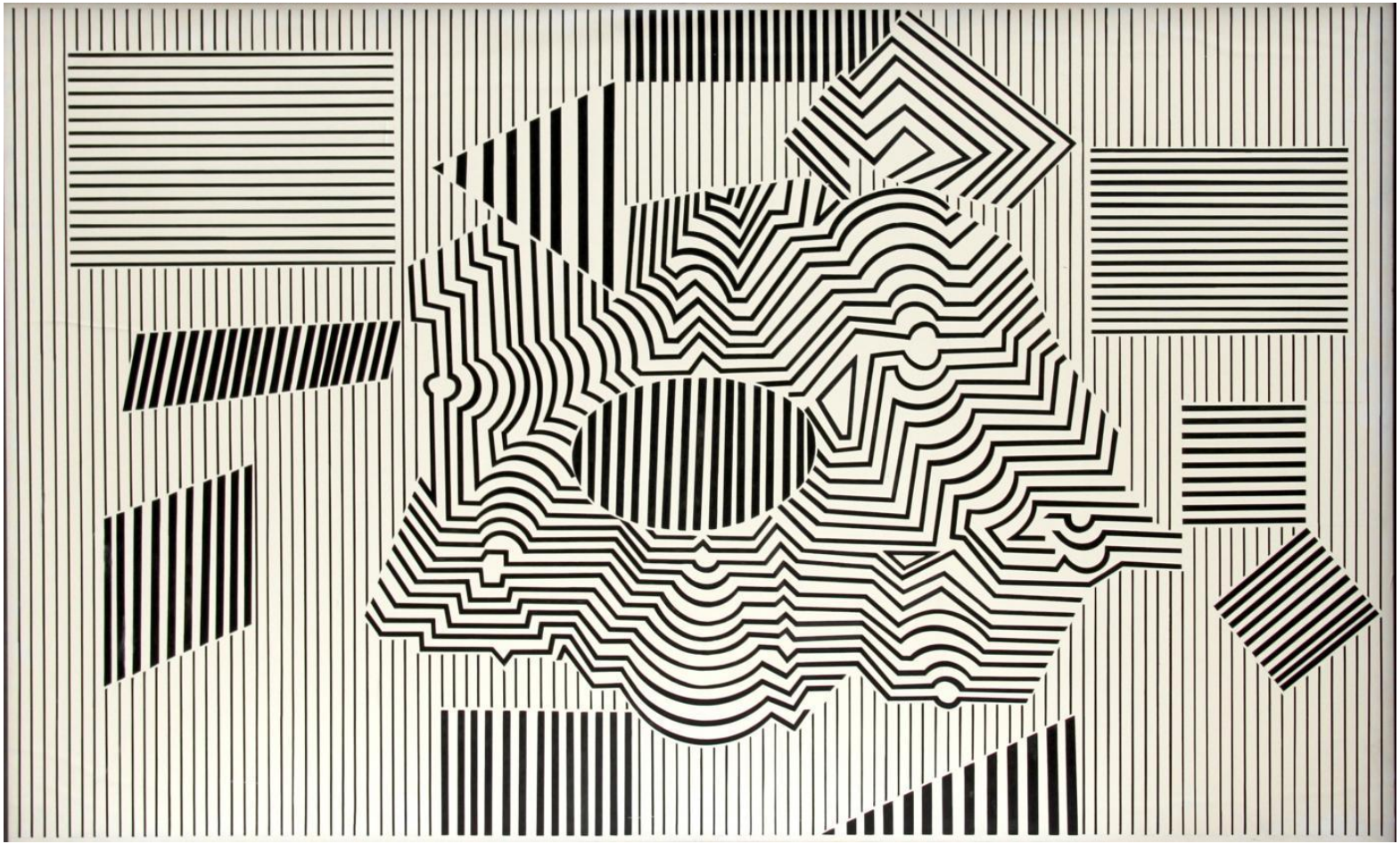
Op painting used a framework of purely geometric forms as the basis for its effects and also drew on colour theory and the physiology and psychology of perception. Leading figures: Bridget Riley, Jesus Rafael Soto, and Victor Vasarely.

Bridget Riley; *Untitled [Fragment 5/8]* 1965; Tate

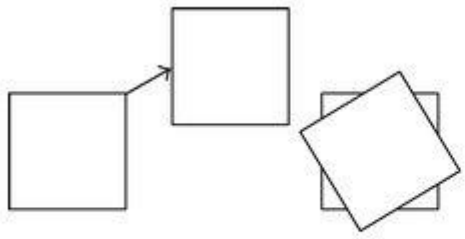
Jesús Rafael Soto

Untitled (Escritura), 1974, wood, wire, paint and nylon strings, 102 x 172 x 30 cm

Victor Vasarely



Operenccia, 1954,
Acrylic on Board,
101.6 cm x 167.64
cm



Tools

translate(x,y) and rotate(ø)

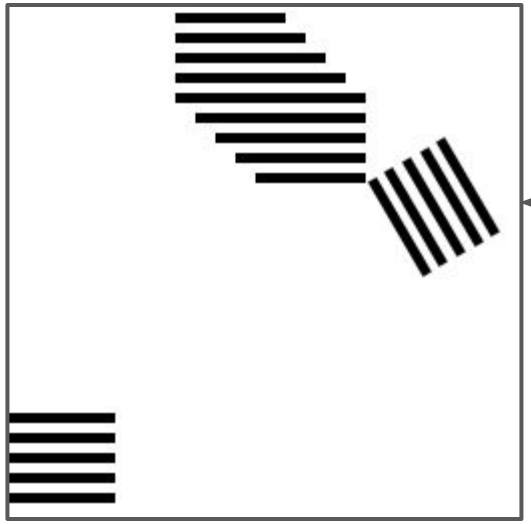
intro-to-creative-coding/w01/
translateRotate/
translateRotate.pde

```
size (300, 300);
background (255);

// draw a rectangle at 0, 0: no translation
fill (200);
rect (0, 0, 60, 60);

// translate the co-ordinate grid and redraw the rectangle
(under translation)
pushMatrix ( );
translate (80, 80);
fill (140);
rect (0, 0, 60, 60);
popMatrix( );

// push/pop Matrix re-set the co-ordinate grid
// now perform another translation and rotation
pushMatrix( );
translate (160, 160); // translate co-ordinates
rotate (radians (45)); // rotate co-ordinate grid, convert
degrees to radians.
fill (100);
rect (0, 0, 60, 60);
popMatrix ( );
```



Challenge

line, translate, rotate, and push/pop matrix

intro-to-creative-coding/wk2 translateGrid/translateGrid.pde

Make this

Open translateRotate.pde, tasks:

1. Fill in lineSegment function
2. Make the first line group appear on bottom left
3. Draw series across the middle
4. Translate, rotate, and draw the final line group

Tools

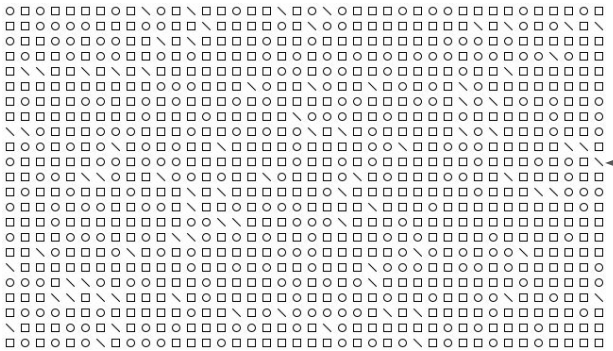
random()

intro-to-creative-coding/wk2
randomBars/randomBars.pde

```
/**
 * Random.
 * Random numbers create the basis of this
image.
 * Each time the program is loaded the result is
different.
 */

size(640, 360);
background(255);
strokeWeight(8);

for (int i = 6; i <= width; i+=16) {
  float r = random(255);
  stroke(r);
  line(i, 0, i, height);
}
```



Challenge

Weighted Probability

intro-to-creative-coding/wk2
weightProbChallenge/
weightProbChallenge.pde

Make something like this

Open weightProbChallenge.pde, tasks:

1. Generate a random number from 1-100;
2. Write an if statement that uses 60:30:10 ratio to draw either a square, circle or line
   a. 60% chance of square
   b. 30% chance of circle
   c. 10% chance of line

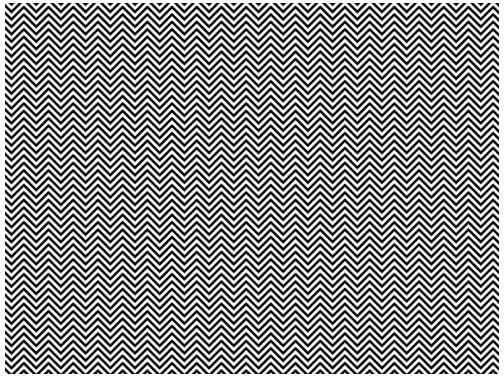
Tools

beginShape, endShape, vertex

intro-to-creative-coding/wk2
simpleShapes/
simpleShapes.pde

```
/** Custom shapes beyond rect and ellipse etc
 *  beginShape, vertex and endShape allow
creation of new geometries
 *  here, z like shapes are made
 */

size(600, 200);
background(255);
noFill();
// Order of vertex position changes the way the
shape is drawn
beginShape();
vertex(30, 20);
vertex(width/2-30, 20);
vertex(30, height-20);
vertex(width/2-30, height-20);
endShape();
...
```



Example

shapes

intro-to-creative-coding/wk2 vertexShapePattern/vertexShapePattern.pde

Ideas to try

1. Make this draw in a section of the screen only
2. Turn this into a single function, so you can use in larger composition
3. Make it a coloured pattern, or form of tiling.

Tools

curves

intro-to-creative-coding/wk2
curves/curves.pde

```
/** Curves
 *  curveVertex connects a series of points
with a curve
 *  the first and last vertices act as control
points
 */
size(600, 300);
smooth();
noFill();
beginShape();
curveVertex(20, 30); // C1
curveVertex(20, 40); // V1
curveVertex(30, 130); // V2
curveVertex(40, 180); // V3
curveVertex(40, 180); // C2
endShape();
...
```

Example

curves

intro-to-creative-coding/wk2
topoCurves/topoCurves.pde

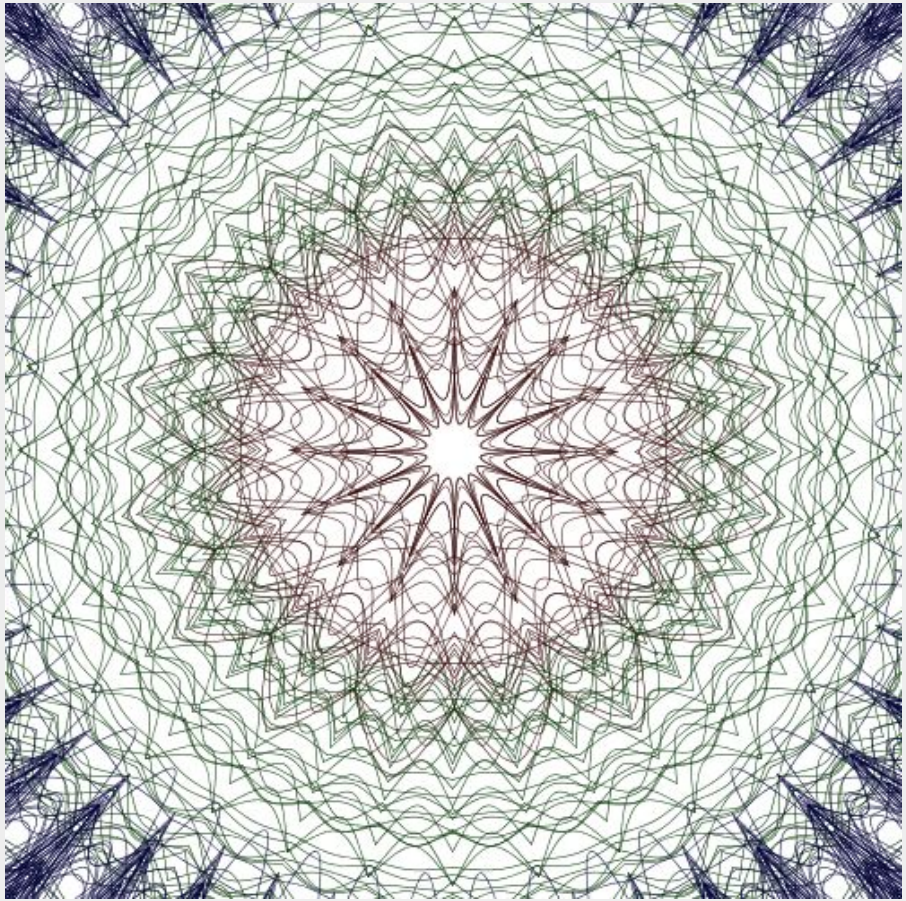
```
/** Complex Curves
 *  Using repetition and change simple curves can become quite
complex patterns
 */

int lineCount = 30;
size(600, 300);
background(255);
smooth();
noFill();
strokeWeight(1);
stroke(140);
for (int i = 0; i < lineCount; i++)
{
  beginShape();
  curveVertex(width-i*20, -20); // C1
  curveVertex(width-i*20, 0); // V1
  curveVertex(width-(i*width/lineCount), height/3); // V2
  curveVertex((i*width/lineCount), (height/3)*2); // V3
  curveVertex(width-i*40, height); // V4
  curveVertex(width-i*40, height+20); // C2
  endShape();
}
...
```

Example

curves

intro-to-creative-coding/wk2
fans/fans.pde



Challenge

Make something yourself!

Using curves, shapes, for loops and conditional statements, make something that interests you. You now have all the pieces to recreate something like Operenccia.