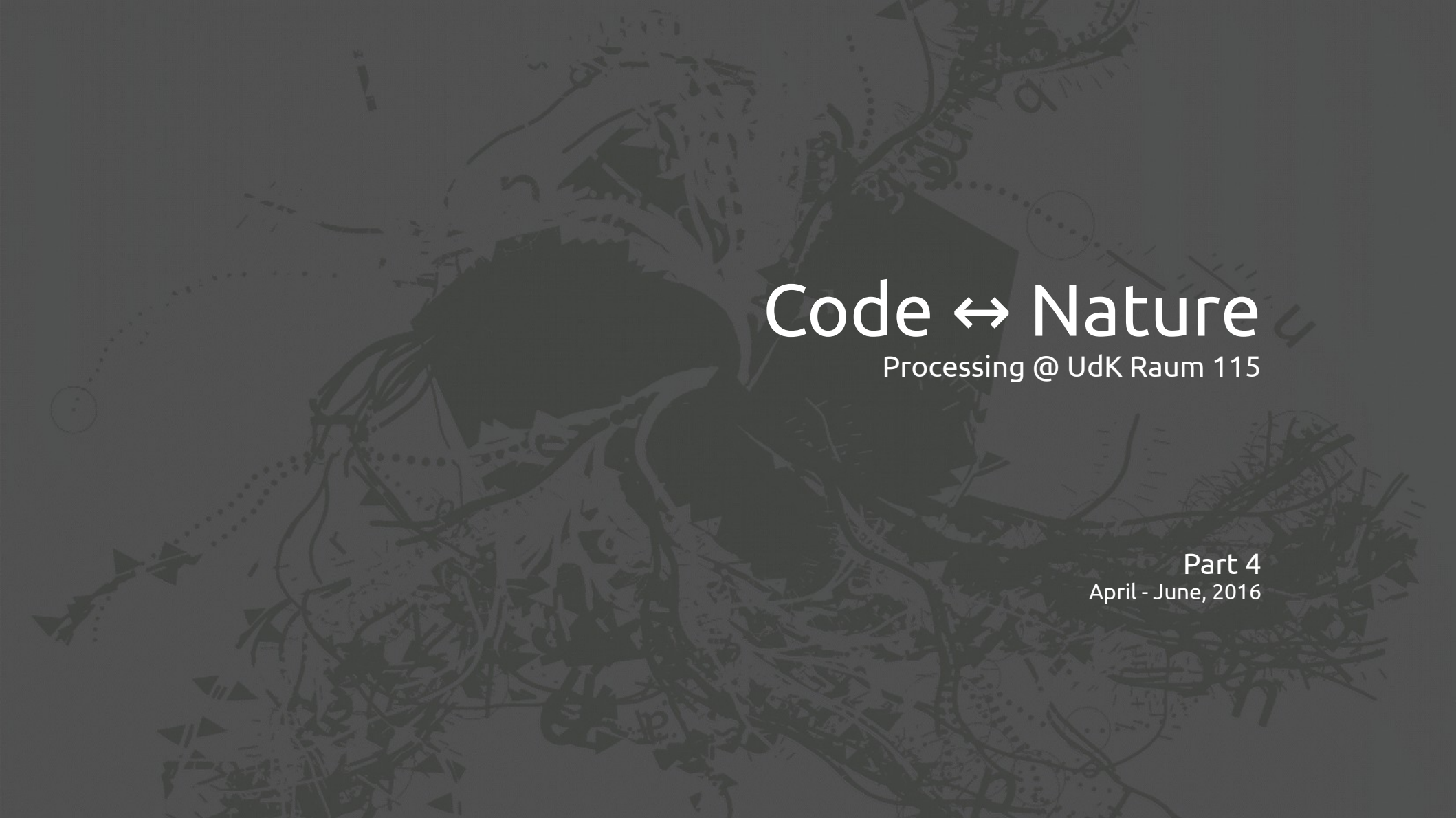


Code ↔ Nature

Processing @ UdK Raum 115

Part 4

April - June, 2016

Exporting vector PDF files

Open the Processing examples, and study the file:

```
Mode Examples > pdf > OneFrame
```

What is new in that program?

Exercise: create a PDF file using for loops, then observe the result in a design program like Inkscape, Illustrator. Then try opening the file in an image editor like Gimp or Photoshop.

mapping numbers to a range

map() is a helper function that simplifies calculations. It lets us convert numbers from one range to a different range. We can achieve the same result just by adding (+) and muliplying (*), but map makes things easier to understand.

To convert temperatures from Celsius to Farenheit, we could use this formula:

```
float celsiusTemp = 25;
println(celsiusTemp * 1.8 + 32);
```

With map it's clearer what we are trying to achieve:

```
float celsiusTemp = 25;
println( map(celsiusTemp,  0, 100,  32, 212) );
```

0 .. 100 is the source range (celsius temperatures), and 32 .. 212 is the target range (fahrenheit).

Syntax: map(value, start1, end1, start2, end2);

map examples

To convert the mouse position to a color:

```
fill(map(mouseX, 0, width, 0, 255));
```

mousex can only be a number between 0 and width. Colors are normally defined by values between 0 and 255.

Convert a random number between 0 and 1 to a position on the screen:

```
float rnd = random(0, 1);
float x = map(rnd, 0, 1, 0, width);
float y = map(rnd, 0, 1, 0, height);
ellipse(x, y, 20, 20);
```

You can think of this as stretching the number range 0..1 to cover the whole width and the whole height of the screen.

Why use map?

Different functions and variables work in different ranges of values. For instance, if your window is 800x600 pixels in size, mouseX will contain a value between 0 and 800. Colors are defined with values between 0 and 255. Trigonometric functions like sin() and cos() return numbers between -1 and 1.

If we want to convert the mouse position into a color, we need to adjust the range somehow. That's what map() makes easy for us.

```
expression    min       max
mouseX         0        width-1
mouseY         0        height-1
sin(x)        -1        1
red            0        255
green          0        255
blue           0        255
random(100)    0        99.99999
noise()        0        1         // generates "smooth" random values
```

Exercise: think of value ranges in "real life": age, temperature, height, speed...

Using map() example

```
// Put 10 ellipses uniformly distributed in a horizontal line.
// The leftmost should be 50 pixels from the left border of the screen.
// The rightmost 200 pixels away from the left border.

for(int i=0; i<10; i++) {
    float x = map(i, 0, 9, 50, 200);
    ellipse(x, 100, 10, 10);
}
```

One of the advantages is how easy it is to update your design.

mapping a counter into different properties

```
// Put 30 ellipses uniformly distributed.
// The first one should be black, at position (80, 80), radius 100.
// The last one white, at position (444, 222), radius 10.

size(600, 400);

for(int i=0; i<30; i++) {
  float x = map(i, 0, 29, 80, 444);
  float y = map(i, 0, 29, 80, 222);
  float sz = map(i, 0, 29, 100, 10);
  fill(map(i, 0, 29, 0, 255));
  ellipse(x, y, sz, sz);
}
```

Note how we map one value into different ranges

Different ways of generating numbers

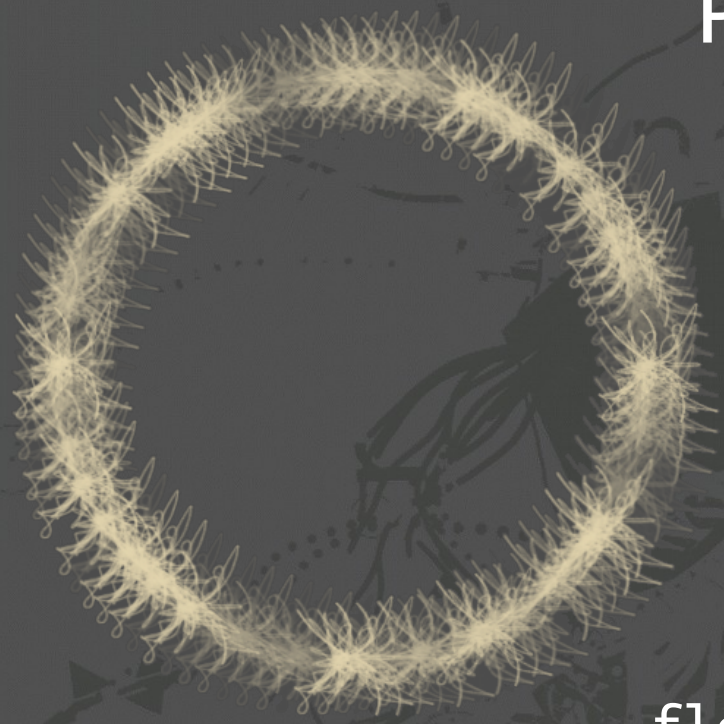
```
// A counter
for(int i=0; i<123; i=i+3) { ... }

// Random
random(1992, 2015);

// Oscillating
sin(x);

// Smooth random values
noise(x);
```

Polar coordinates

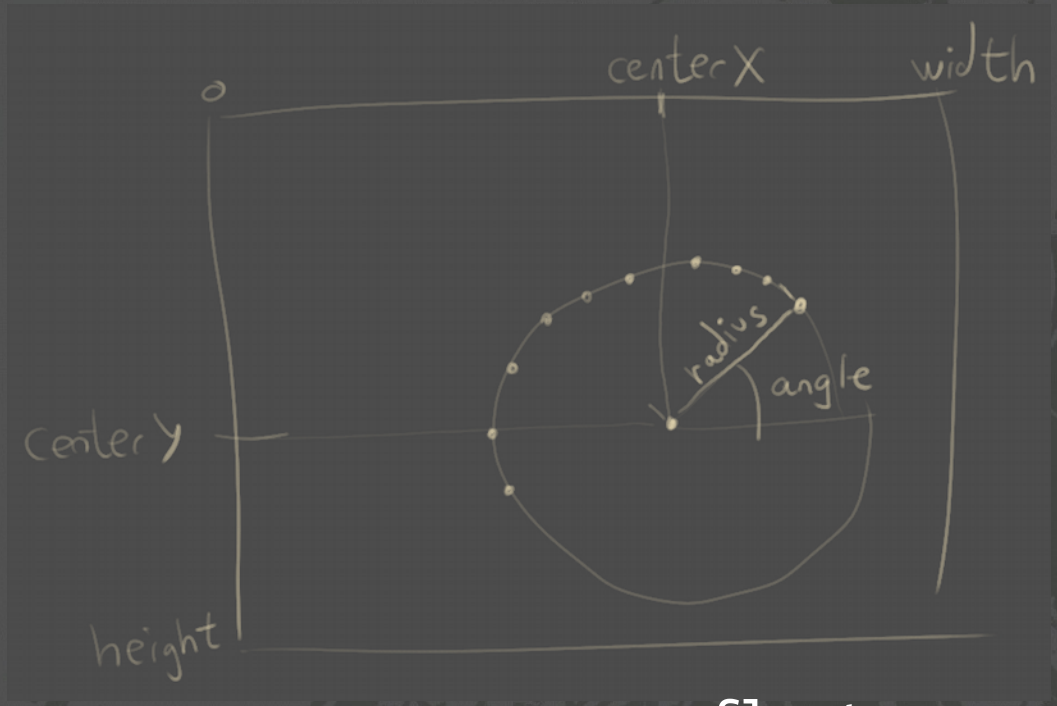


This formula lets us convert
a radius and an angle into
cartesian (x, y) coordinates.

```
float x = centerX + radius * cos(angle);
float y = centerY + radius * sin(angle);
```

http://www.openprocessing.org/sketch/183592

Polar coordinates



By increasing or decreasing "angle", we can get different points on the circle.

We can change the radius to specify the size of the circle.

```
float x = centerX + radius * cos(angle);
float y = centerY + radius * sin(angle);
```

Periodic & circular motion

Examples:

use sin() to produce periodic motion
circular motion
draw a circle
draw a spiral
join two rotating points with a line

Code ↔ Nature

April - June 2016 // UdK - Berlin

Abe Pazos
@fun_pro | fun@funprogramming.org
http://hamoid.com | http://funprogramming.org