Arch497i

Week 1

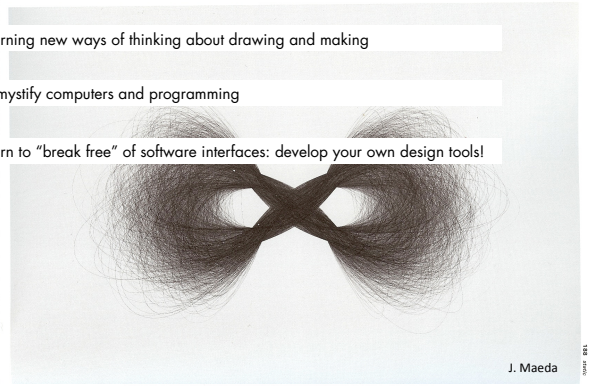
Introduction
Syllabus
Examples
Screen basics
Processing basics
Interaction basics
First Assignment
Submissions

A foundation for exploring visual and spatial design through computer programming

Learning new ways of thinking about drawing and making

Demystify computers and programming

Learn to "break free" of software interfaces: develop your own design tools!



PROCESSING

•Open Source

•Free

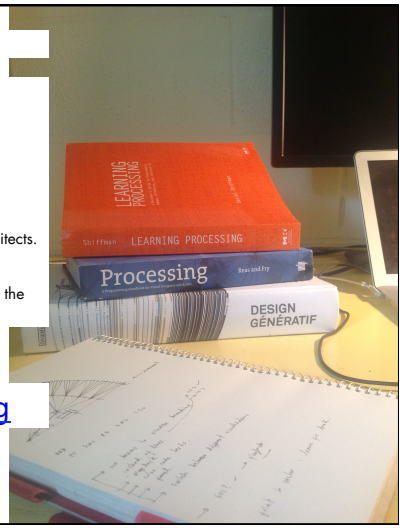
•Visual

•Well documented

•Widely used by artists, designers, architects.

•If you learn it, you will understand the foundations of programming (which are the same in all languages).

www.processing.org



SOME EXAMPLES

Simulation
http://www.processing.org/exhibition/works/sodaprocessing/index_link.html (E. Burton)
http://markmckeague.com/work/city-symphonies/ (M. McKeague)

Visual/spatial exploration
http://processing.org/exhibition/works/versionb/index_link.html (M. Jogan)
http://www.mos-office.net/uploaded_files/project_file/path/84585/Collapse2sm.mov (MOS)
https://vimeo.com/35664267 (Design generatif)
https://vimeo.com/35664620 (Design generatif)
http://vimeo.com/42144061 (Autonomous drawing system: N. Fischer)
http://vimeo.com/43848831 (Glitch-Mesh: M. Plummer-Fernandez)

Installation
http://worthersoriginal.com/viki/#page=shadowmonsters (P. Worther)

Fabrication
http://n-e-r-v-o-u-s.com/ (nervous system)
https://vimeo.com/6975570# (Boom + me)
http://www.creativeapplications.net/android/stonespray-3d-printing-with-sand/ (P. Novikov, I. Shergill and A. Kulik)

Visualization
http://www.wefeelfine.org/ (J. Harris)

So what?

-Computational design (and I don't mean people making renderings) is an expanding area of practice.
-Not all-purpose design tools, but focused computational tools for your (studio, thesis, artistic practice…) project.
- Other forms of reasoning and data informing a design project.

Now the basics

The philosophy of incremental development: Take it one step at a time. AKA: Divide and conquer.

What is an algorithm?

Algorithms are step-by-step recipes for accomplishing a certain task.

In a piece of paper, write an algorithm for something you (presumably) do every day, such as brushing your teeth.

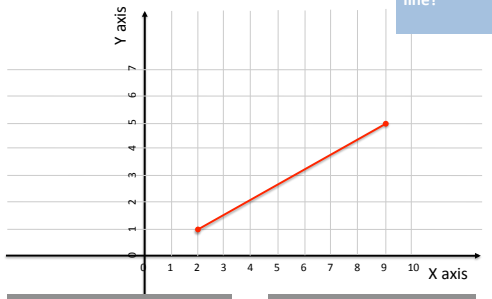
Try to be as explicit (and dumb) as possible.

Step 0.
Step 1.
Step 2.
.
.
.

Screen basics

-Coordinates

How would you tell someone to draw this line?



To a person:
Please draw a line from point 2, 1, to point 9, 5.

To a computer:
`line(2, 1, 9, 5);`

Natural and formal languages are different

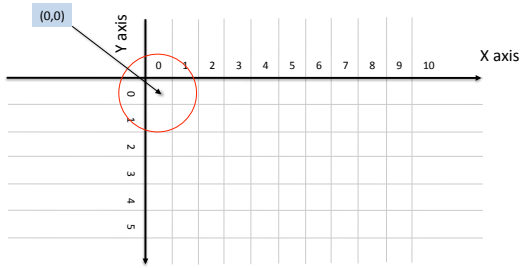
Natural language:
Draw a line from 2, 1, to 9, 5.

Verb — Object — Object

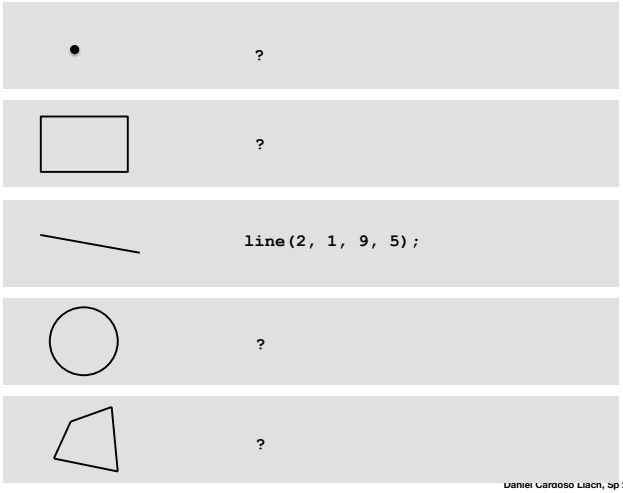
Formal language:
line(2, 1, 9, 5);

Function — Arguments

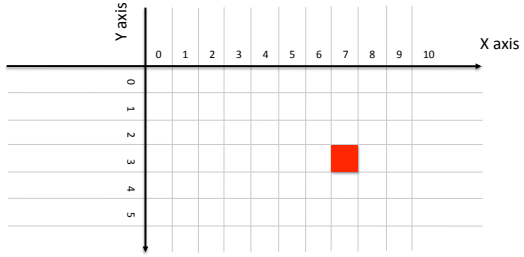
Note: The orientation of the axis on the screen is different from graph paper.



Screen axis orientation

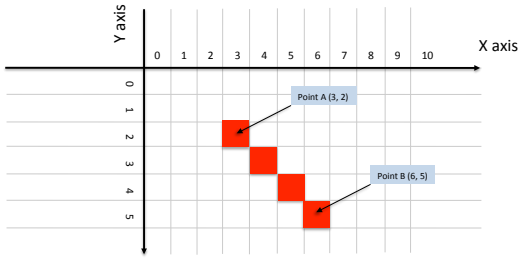


Simple shapes: Point



Point(7, 3);

Simple shapes: Line

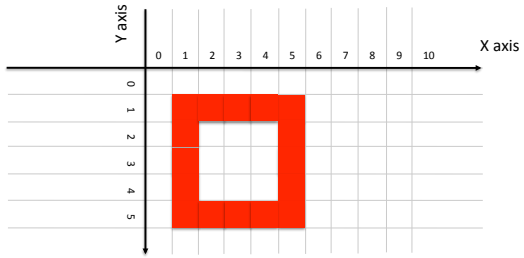



```
Line(3, 2, 6, 5);
```

Point A Point B

Daniel Cardoso Llach, Sp 2015

Simple shapes: Rectangle (Corner)



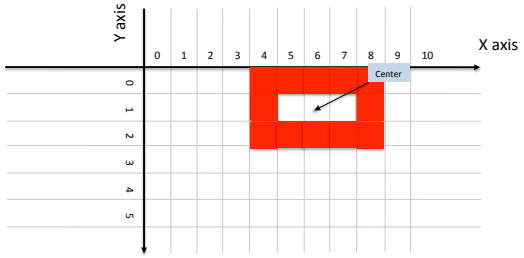
Top left X Top left Y

```
rect(1, 1, 5, 5);
```

Width Height

Daniel Cardoso Llach, Sp 2015

Simple shapes: Rectangle (Center)

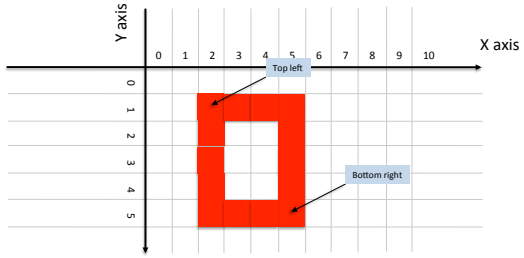



```
rectMode(CENTER);
Rect(6, 1, 5, 3);
```

Center Width Height

Daniel Cardoso Llach, Sp 2015

Simple shapes: Rectangle (Corners)

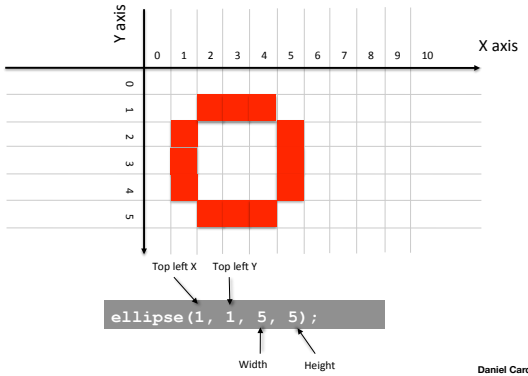



```
rectMode(CORNERS);
Rect(2, 1, 5, 5);
```

Top left Bottom right

Daniel Cardoso Llach, Sp 2015

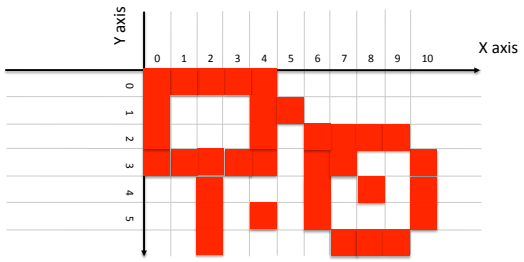
Simple shapes: Ellipse (also has CENTER and CORNERS modes)



In a piece of paper, draw the shapes specified in the following code:

```
line(0, 0, 9, 6);
point(0, 2);
point(0, 4);
rectMode(CORNER);
rect(5, 0, 4, 3);
ellipseMode(CENTER);
ellipse(3, 7, 4, 4);
```

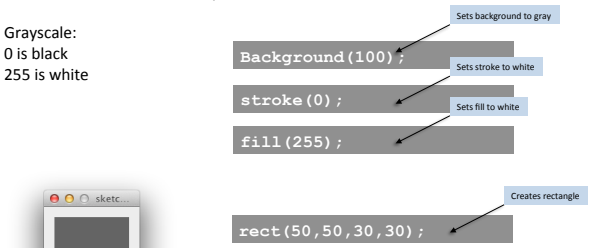

"Reverse engineer" the following picture:



What else can a pixel do? Color

Colors are defined numerically

Grayscale:
0 is black
255 is white



What else can a pixel do? Color

Colors are defined numerically

Grayscale:
0 is black
255 is white


```
background(100);
stroke(0);
fill(255);
rect(50, 50, 30, 30);
noStroke();
ellipse(30, 30, 20, 40);
stroke(255);
line(0,0,100,100);
noFill();
stroke(0);
ellipse(50,50,100,100);
```

Turns off stroke

Turns off fill

Daniel Cardoso Llach, Sp 2015

What else can a pixel do? Color

Colors are defined numerically

RGB (Red, Green, Blue)
0 – 255 range


```
background(255);
stroke(0);

fill(255, 0, 0);
rect(15,30, 15, 30);

fill(0, 255, 0);
rect(40, 30, 15, 30);

fill(0, 0, 255);
rect(65, 30, 15, 30);
```

Bright Red

Bright Green

Bright Blue

Daniel Cardoso Llach, Sp 2015

Make a drawing with simple shapes and colors (a building, an object, an animal… doesn't matter)

-Draw it by hand (just lines, points, rectangles and ellipses)
- Write the code for it

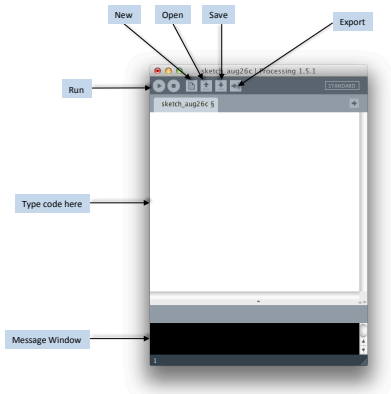
Daniel Cardoso Llach, Sp 2015

Install Processing

http://processing.org/download

Program files (Win) or Applications (Mac)
Locate executable file and run

Daniel Cardoso Llach, Sp 2015



Explore File>Examples, choose one sketch, and run it

Processing

Programs are called "sketches"
No hyphens or spaces in names (always "save as")
Default sketches folder in MyDocuments (or Documents in OSX)

```
size(350, 350);
```
Changes the output window size

```
println("Hello");
```
Prints to the message window

```
Smooth();
```
Enables Anti Aliasing

These are function calls

```
// this is a comment
```
Any text after "//" will be ignored by processing. This is useful for making notes that make the code more readable.

Try coding your drawing adding comments, changing the window size, and printing a line to the message window.)

Exporting

File > Export Applet
Produces:
- html (for web view)
- filename.jar (compiled applet)
- filename.java (the P. code translated into java)
- filename.pde (the processing source code)
- a gif loader image

Interaction basics

Programs are not like movies or animations. Many times they interact with the users.

Code blocks: setup and draw.

```
void setup(){
// initialization code goes here

}

void draw(){
// code that repeats goes here
}
```

Interaction basics: setup and draw

Programs are not like movies or animations. Many times they interact with the users.

```
void setup(){
// step 1a
// step 1b
// step 1c

}

void draw(){
// step 2a
// step 2b
// step 2c

}
```

Runs once

Repeats forever

Interaction basics: setup and draw

Moving with the mouse

```
void setup(){
    size(300, 300);
    smooth();
}

void draw(){
    background(255);
    rect(mouseX, mouseY, 30, 30);
}
```

Runs once

Repeats forever

mouseX and mouseY are keywords for the horizontal and vertical location of the mouse in the window.

Interaction basics: setup and draw

Simple drawing tool!

```
void setup(){
    size(300, 300);
    background(255);
    smooth();
}

void draw(){
    stroke(0);
    line(pmouseX, pmouseY, mouseX, mouseY);
}
```

Runs once

Repeats forever

pmouseX and pmouseY contain the location of the mouse in the previous iteration Of the "draw" code.

Interaction basics: Mouse clicks and key strokes

```
void setup(){
    size(300, 300);
    background(255);
}

void draw(){

}

void mousePressed(){
    stroke(0);
    fill(175);
    rectMode(CENTER);
    rect(mouseX, mouseY, 25, 25);
}

void keyPressed(){
    background(255);
}
```

Runs once

Runs once when the mouse is pressed.

Runs once when a key is pressed.

Summary

- Algorithm = Recipe
- Computer code is different from human language (needs to be more precise)
- Simple shapes: rect(), ellipse(), line(), point()
- Color control: Grayscale and RGB: stroke(), fill(), background(). Ranges 0 – 255.
- Processing functions: size(), smooth().
- To export as an applet: File > Export applet
- void() {} and draw() {}
- mousePressed(){} and keyPressed(){}
- // Comments

In Class Exercise

- Use primitive shapes and RGB colors to design a static sketch (400px X 400px size). Feel free to use parts of the code you wrote today.
- Make the sketch **dynamic** by including some kind of interaction (either mouse, or keyboard).
- Don't forget to include comments in your code (//).
- Export as an applet
- Save a JPEG image of the sketch.
-Write a short text in Notepad (.txt) about your sketch. What was difficult? What was your design intent? (200 words max.)
-Submit a folder including a) the sketch folder, b) the JPEG, and c) the text.
- Name the folder "week1_[your name]" and submit to ANGEL.

Extra credit:
If you are inspired, base your sketch on a painting by Mondrian, or Klee, of your choice.

Week 1 Assignment

Assignments are due before the next class.

Learn to export a high-res image from Processing

```
void keyPressed() {
  if (key==' ') {
    save("this.tif");
  }
}
```

This function saves a TIF image of the current window to the sketch folder when the space bar is pressed.