POLITECNICO
MILANO 1863

Digital Art

2021-2022

Introduction to processing

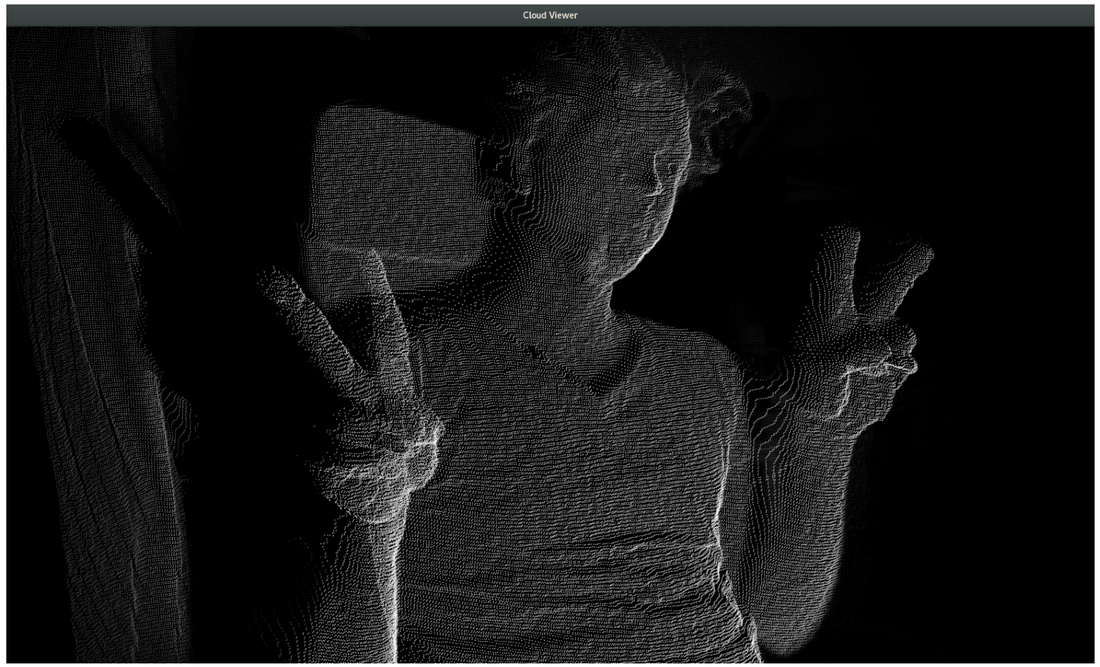
Processing?

POLITECNICO
MILANO 1863

When they started Processing in 2001, the goal was to **bring ideas and technologies** out of MIT and **into the larger world**.

They called this **sketching with code**.

Processing emerged directly from the Aesthetics and Computation Group (ACG), a research group started at the Media Lab by John Maeda in 1996.



https://medium.com/processing-foundation/a-modern-prometheus-59aed94abe85

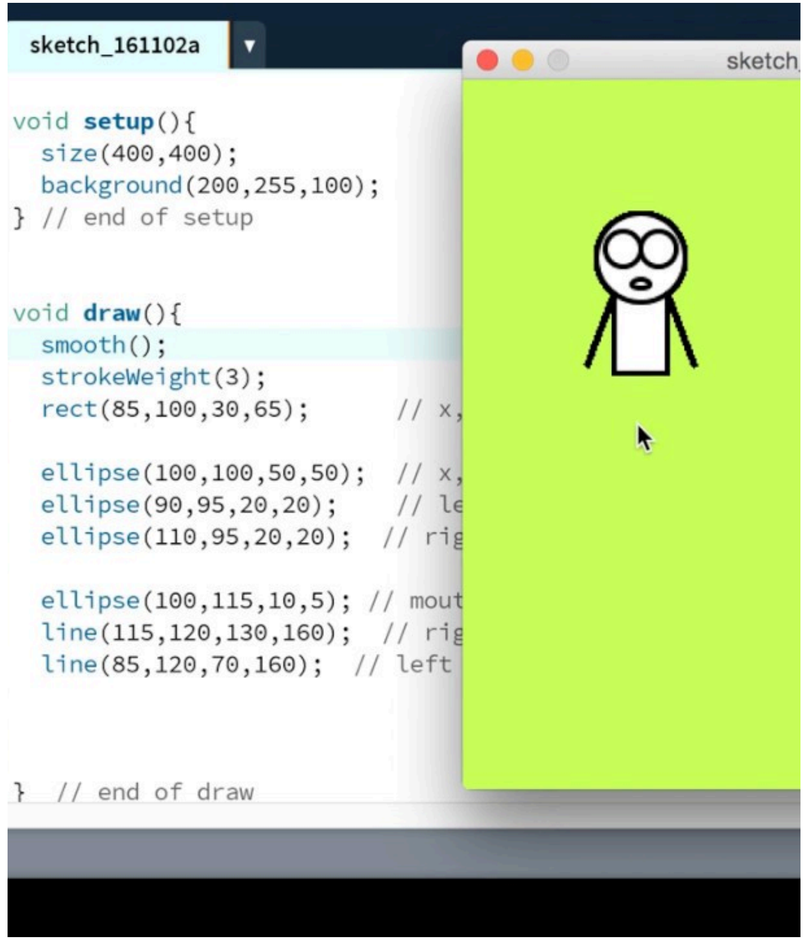
Sketching?

A Processing program is called a sketch. This is more than a change in nomenclature, **it's a different approach to coding.**

The more traditional method is to resolve the entire plan for the software before the first line of code is written.

This approach can work well for well-defined domains, but when **the goal is exploration and invention**, it prematurely cuts off possible outcomes.

Through sketching with code, unexpected paths are discovered and followed. Unique outcomes often emerge through the process.



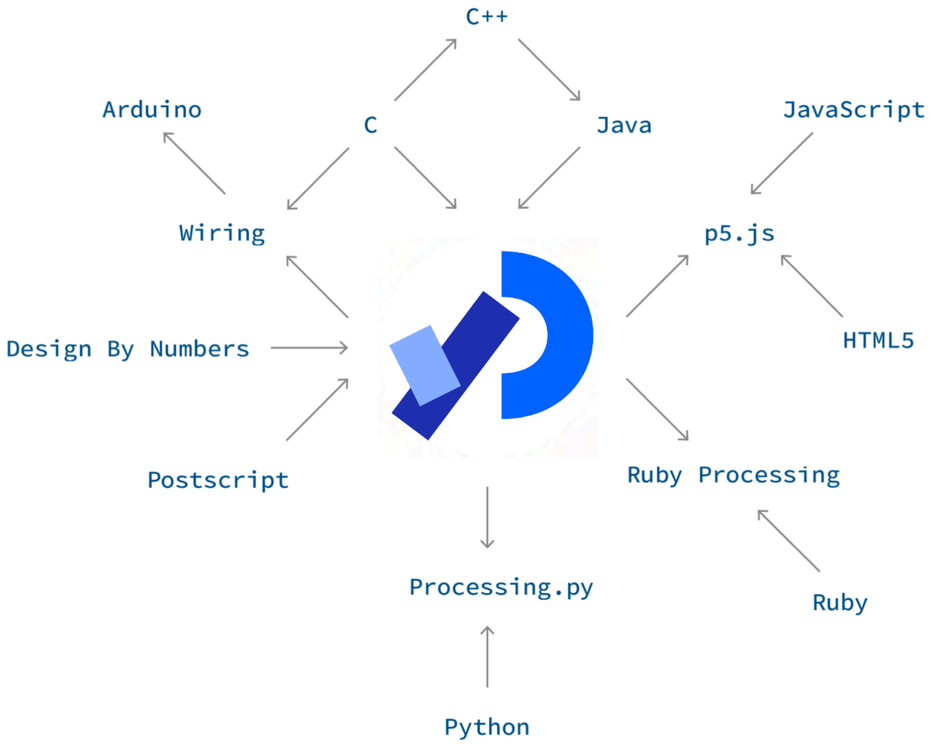
What’s for?

POLITECNICO
MILANO 1863

The original mission of Processing was to create software that made learning to code accessible for visual people (designers, artists, architects) and to help a more technical audience work fluidly with graphics.

From the original Processing software, the Foundation is now supporting a range of different projects.

The **p5.js** project is a JavaScript reimagining of Processing within the context of contemporary web browsers.
Processing.py it’s now a Mode for the Processing 3 editor. Additionally, **Processing for Android** as a Mode for Processing 3, Processing 3 running well on **Raspberry Pi** and CHIP hardware, and there is a library to read and write directly to the I/O pins.



Download

POLITECNICO
MILANO 1863

_On **Windows**, you'll have a .zip file.

Double-click it, and drag the folder inside to a location on your hard disk. It could be Program Files or simply the desktop, but the important thing is for the processing folder to be pulled out of that .zip file. Then double-click processing.exe to start.

_The **Mac OS X** version is also a .zip file.

Double-click it and drag the Processing icon to the Applications folder.
If you're using someone else's machine and can't modify the Applications folder, just drag the application to the desktop. Then double-click the Processing icon to start.

https://processing.org/download/

Download

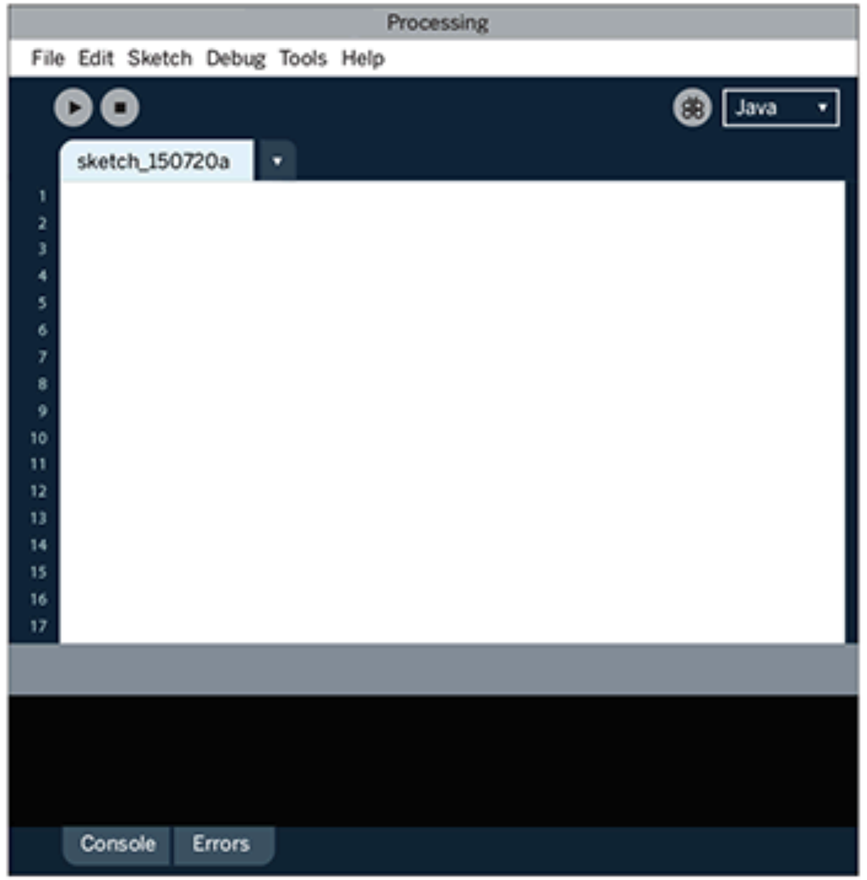
Download

This Git with slides and examples

https://github.com/giulioriot/digitalArt

Hello, I’m Processing

Display Window



Menu

Toolbar

Tabs

Text Editor

Message Area

Console

How it works?

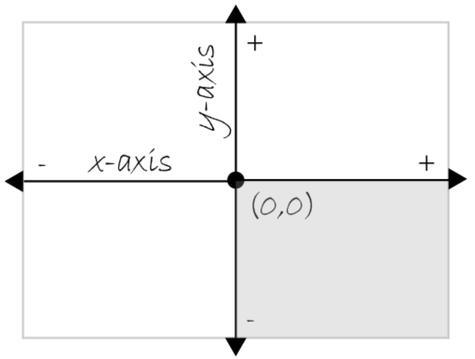
A journey of a thousand miles begins with a single step.

—Lao-tzu

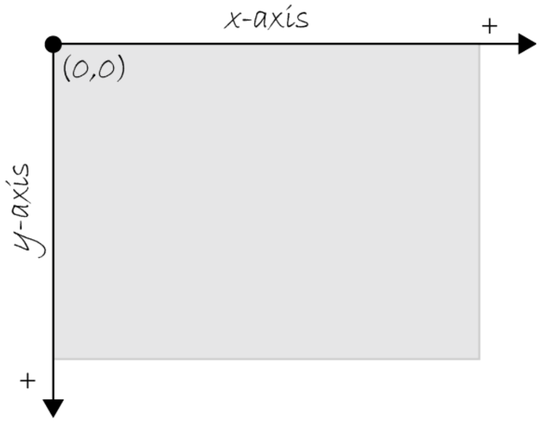
PIXELS

Digital is made by pixels, when you have to create something on a display you have to specify where you want it…

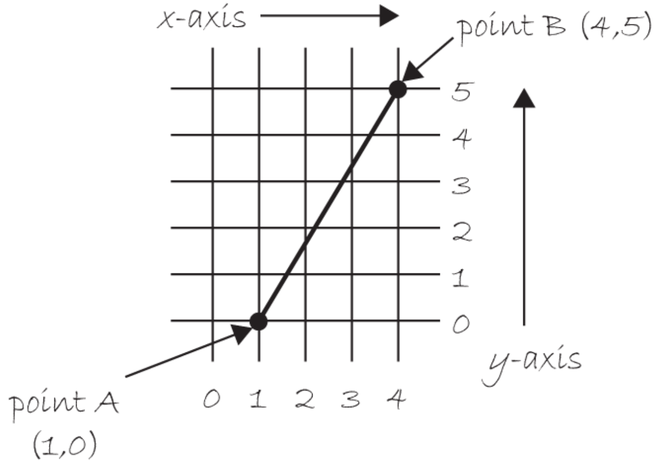
Computers thinks by pixels



graph paper



computer



This figure shows a line between point A (1,0) and point B (4,5). If you wanted to direct a friend of yours to draw that same line, you would say “draw a line from the point one-zero to the point four-five, please.”

How it works?

'I try to apply colors like words that shape poems, like notes that shape music'

—Joan Miró

COLORS

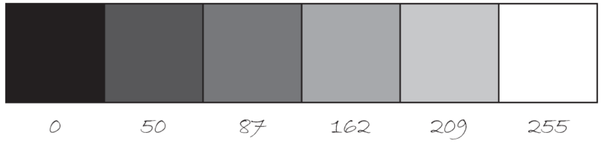
Color is defined with a range of numbers.

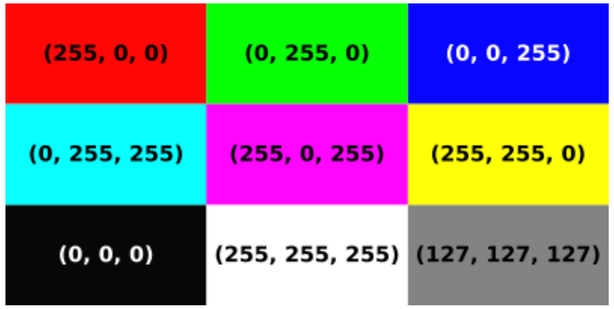
The simplest case: *black and white* or *grayscale*.

To specify a value for grayscale, use the following:

0 means black, 255 means white.

In between, every other number — 50, 87, 162, 209, and so on — is a shade of gray ranging from black to white.





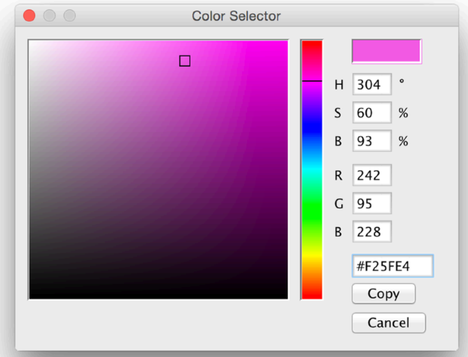

```
fill(0);
fill(255,0,0);
fill(0,255,0);
fill(0,0,255);
```

Syntax

```
fill(rgb)
fill(rgb, alpha)
fill(gray)
fill(gray, alpha)
fill(v1, v2, v3)
fill(v1, v2, v3, alpha)
```

Processing also has a color selector to aid in choosing colors.

Access this via "Tools" (from the menu bar) → "Color Selector."

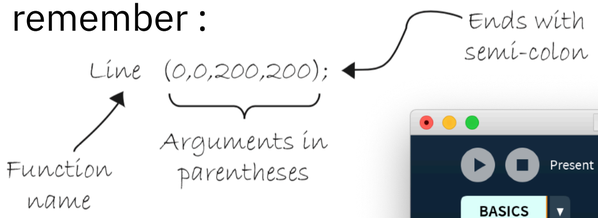


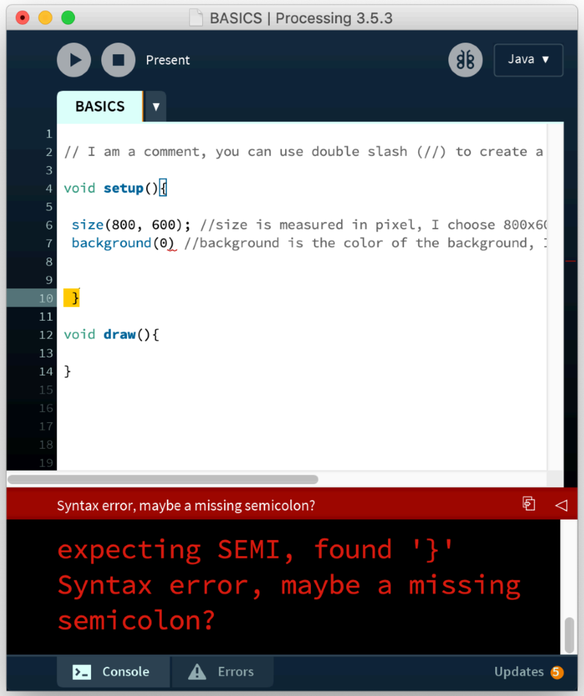
Always;

The console in the lower part shows you errors

the most frequent error is a missing semicolon one, so

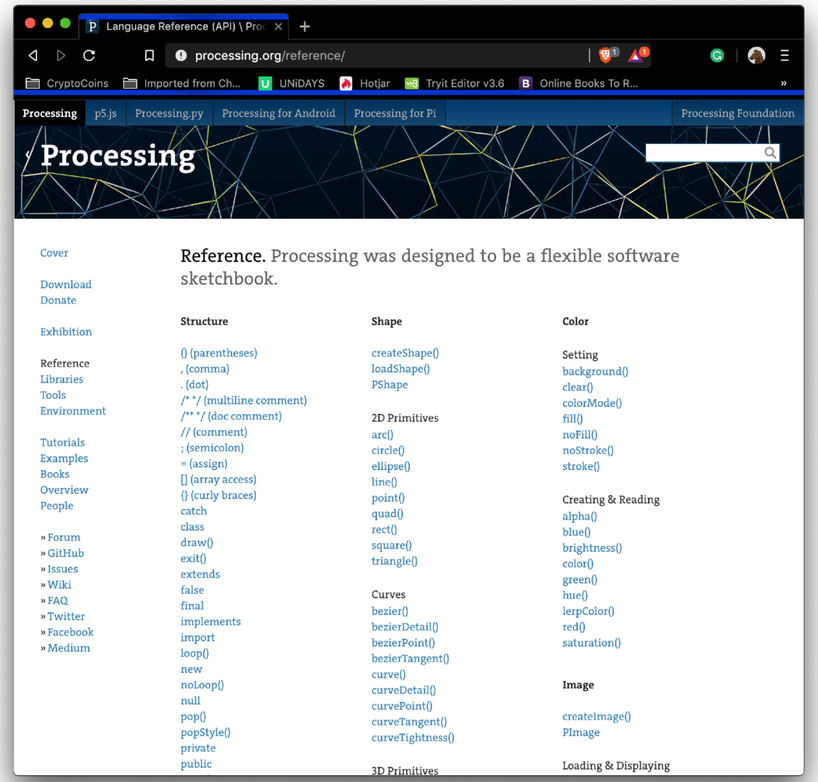
remember :





everything you need is here:

https://processing.org/reference/



Let’s create a sketch

POLITECNICO MILANO 1863

The keyword **void** indicates a function with no value.

If you don't know what a function is...

void setup(){

is the space in which I setup my sketch

Everything you write in the SETUP will run just once

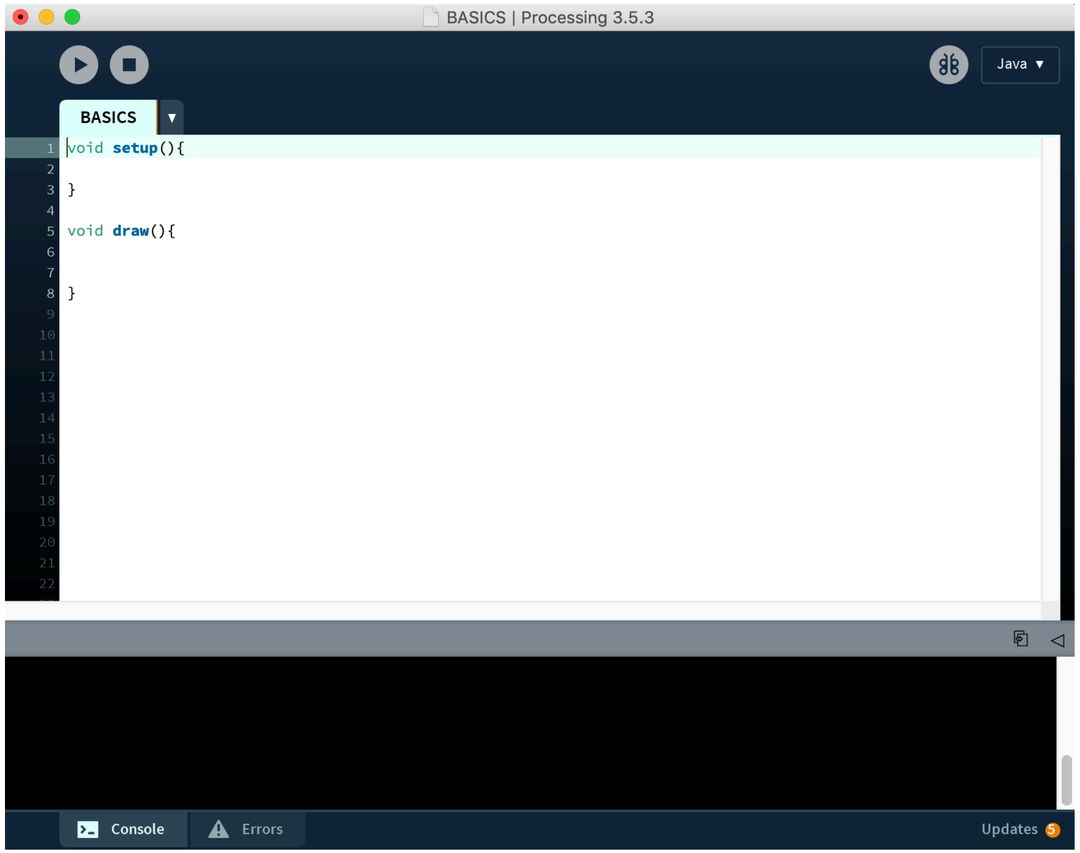
}

void draw(){

is the space in which I draw in my sketch

Everything you write in the DRAW, will run in an infinite loop

}



Basics

POLITECNICO
MILANO 1863

```
// open the folder of examples

sketch_1BASICS
```

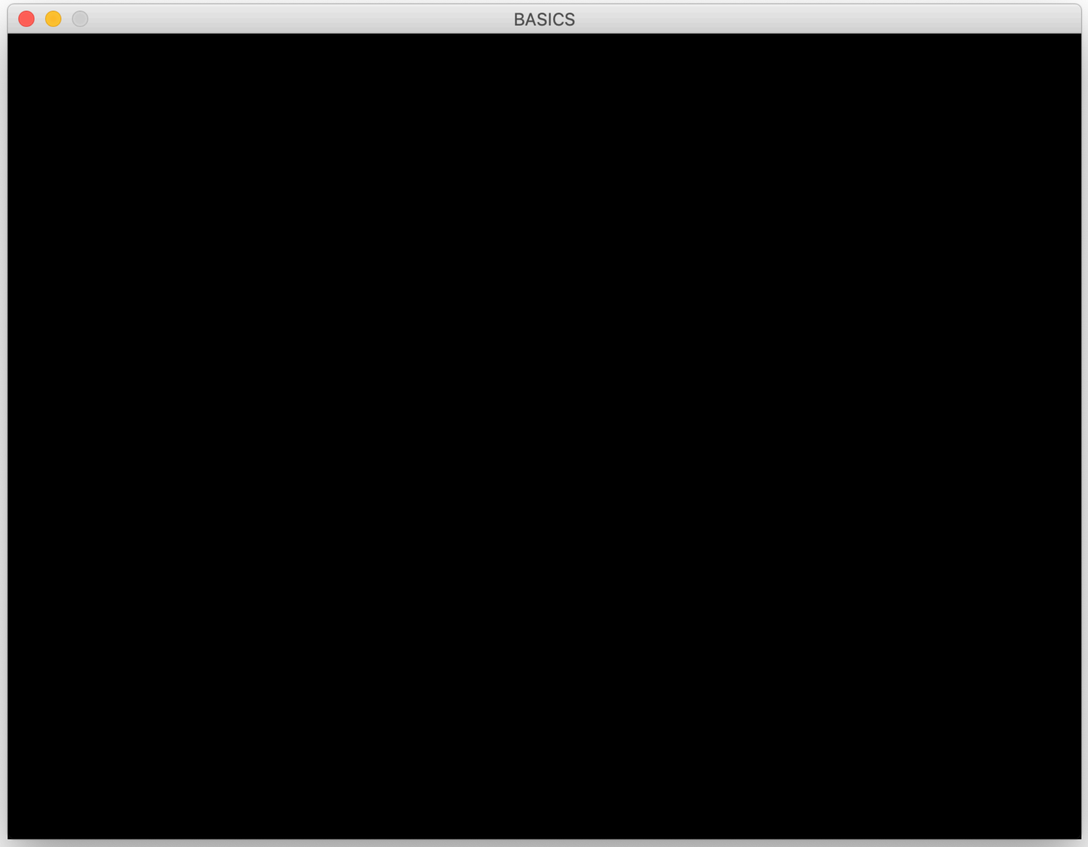
here you find

A black blank page

our hello world

`tips:`

try to change the size of the sketch and the background color at line 10 and 12



Shapes

POLITECNICO
MILANO 1863

```
// open the folder of examples

sketch_2BASIC_SHAPES
```

here you find

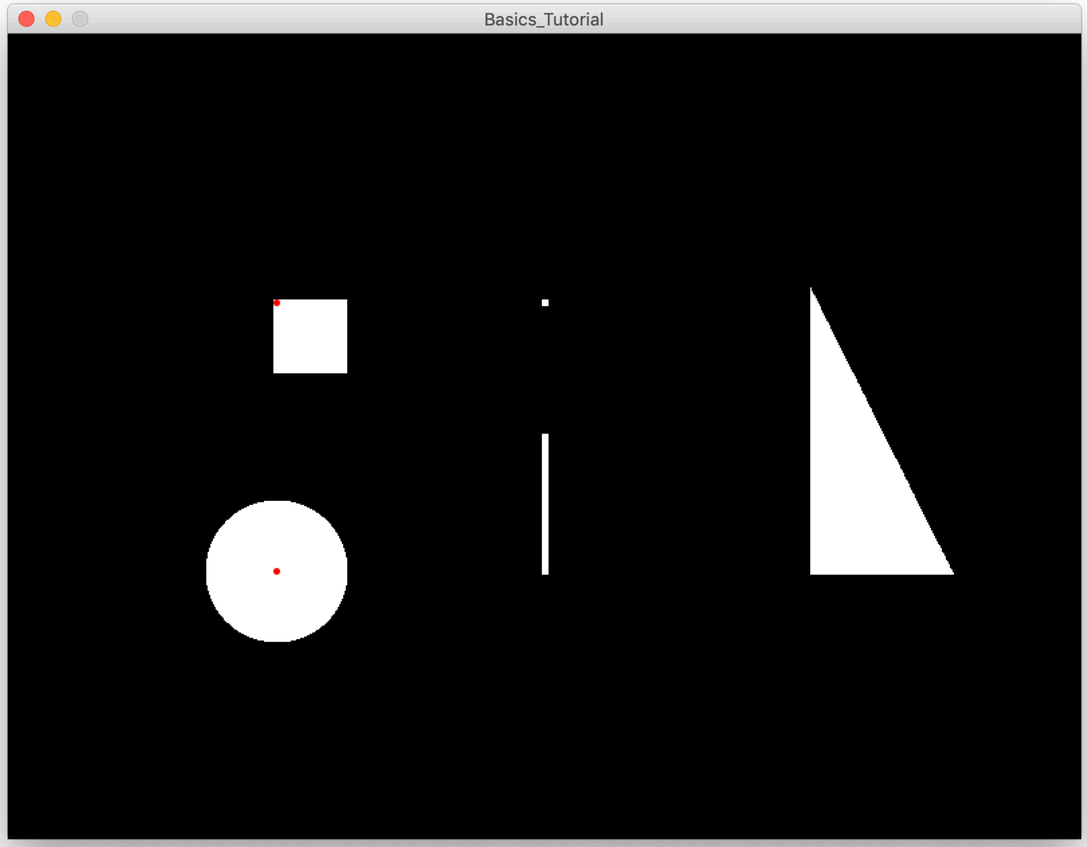
Primitive shapes

to understand how to draw by code

tips:

change from a grayscale color to rgb at lines 8 and 14

try to modify the triangle shape at line 24



Stickman

POLITECNICO MILANO 1863

```
// open the folder of examples

sketch_3STICK_MAN
```

here you find

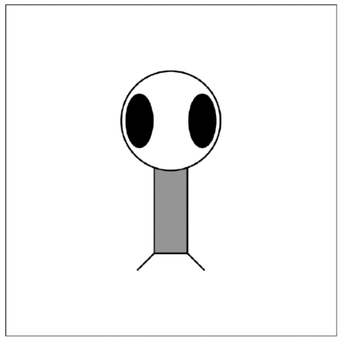
A simple stickman

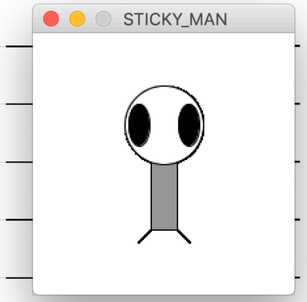
hack and modify code to obtain a personal output

tips:

try to add arms and details, be creative!

```
background(255);
ellipseMode(CENTER);
rectMode(CENTER);
stroke(0);
fill(150);
rect(100, 100, 20, 100);
fill(255);
ellipse(100, 70, 60, 60);
fill(0);
ellipse(81, 70, 16, 32);
ellipse(119, 70, 16, 32);
stroke(0);
line(90, 150, 80, 160);
line(110, 150, 120, 160);
```



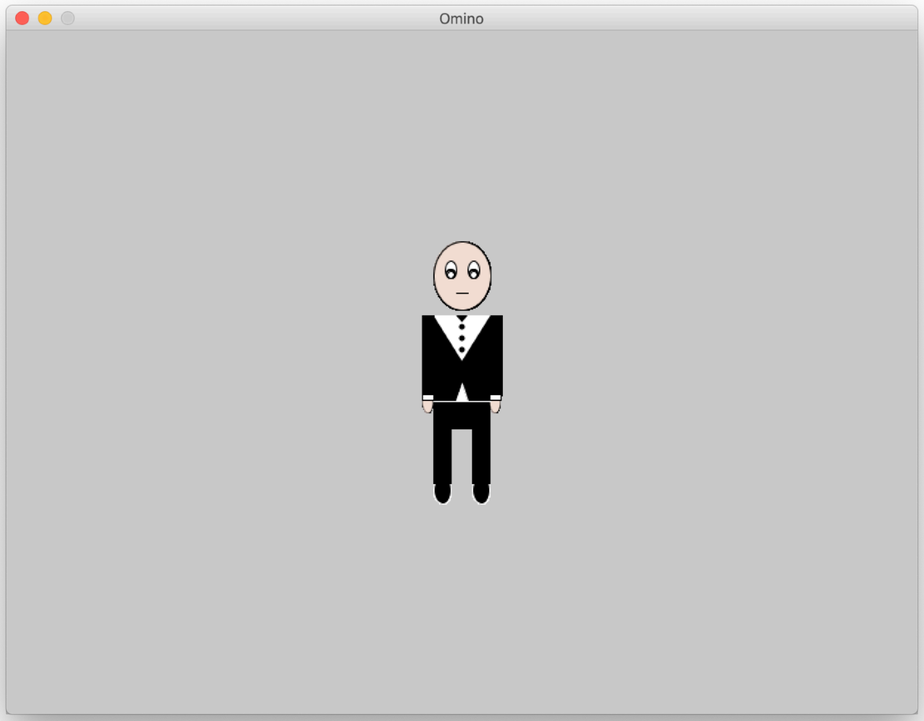


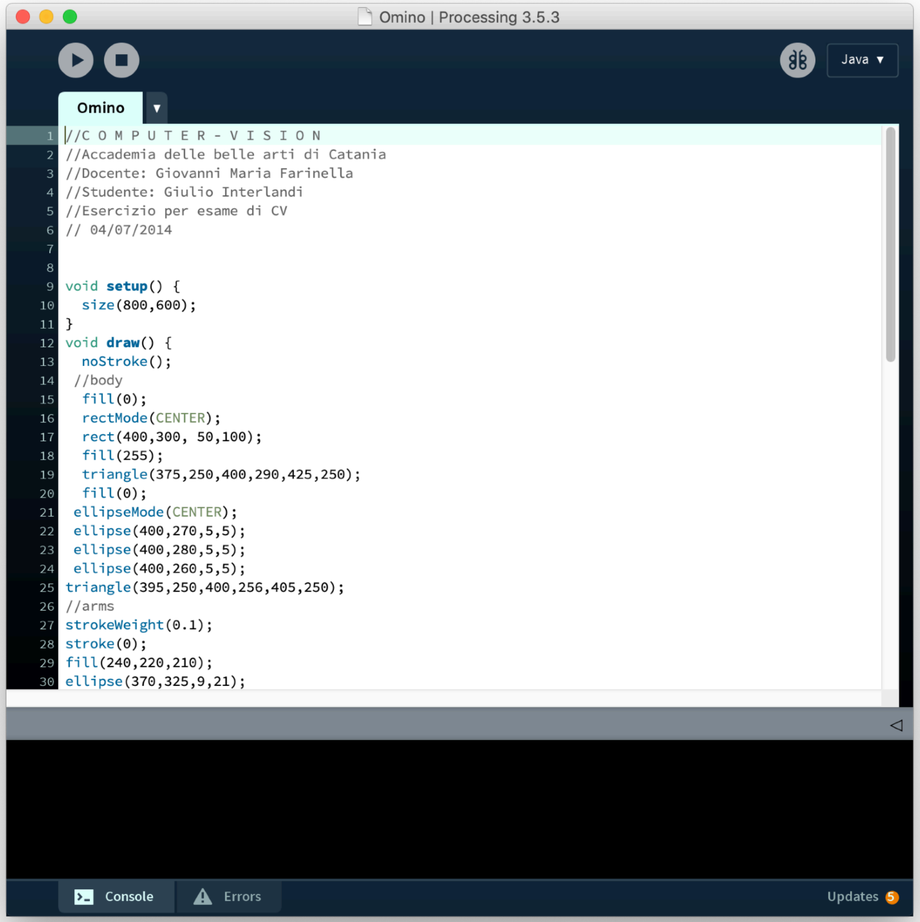
Be creative!

POLITECNICO MILANO 1863

Let’s create your own

that was my exercise some years ago



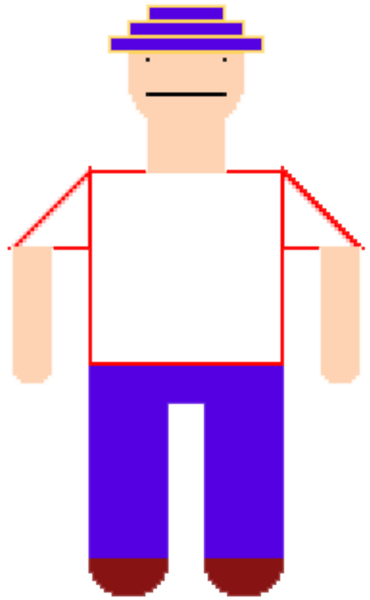


Be creative!

POLITECNICO
MILANO 1863

Let’s create your own

this is a funny sketch done by a student last year.



```
// feet
stroke(133, 21, 24);
fill(133, 21, 24);
ellipse(385, 475, 20, 20); // left foot (center X, center Y, width, height)
ellipse(415, 475, 20, 20); // right foot (center X, center Y, width, height)
// trousers
stroke(84, 21, 222);
fill(84, 21, 222);
rect(375, 425, 50, 10); // pacco (starting X, starting Y, width, height)
rect(375, 435, 20, 40); // left leg (starting X, starting Y, width, height)
rect(405, 435, 20, 40); // right leg (starting X, starting Y, width, height)

// face
stroke(252, 211, 181);
fill(252, 211, 181);
rect(390, 355, 20, 20); // neck (starting X, starting Y, width, height)
ellipse(400, 350, 30, 30); // face (center X, center Y, width, height)
stroke(0);
fill(255);
point(390, 346);    // left eye
point(410, 346);    // right eye
line(390, 355, 410, 355);  // mouth (starting X, starting Y, ending X, ending Y)
// cap
stroke(252, 211, 102);
fill(84, 21, 222);
rect(380, 340, 40, 4); // cap down (starting X, starting Y, width, height)
rect(385, 336, 30, 4); // cap middle (starting X, starting Y, width, height)
rect(390, 332, 20, 4); // cap middle (starting X, starting Y, width, height)
```

Credits: Luca Ghezzi

Let’s create a sketch

POLITECNICO
MILANO 1863

```
// open the folder of examples

sketch_4DRAWING TOOL
```

here you find

Drawing principles

to understand:

•how to connect and draw by mouse input

• if (key pressed) case

`tips:`

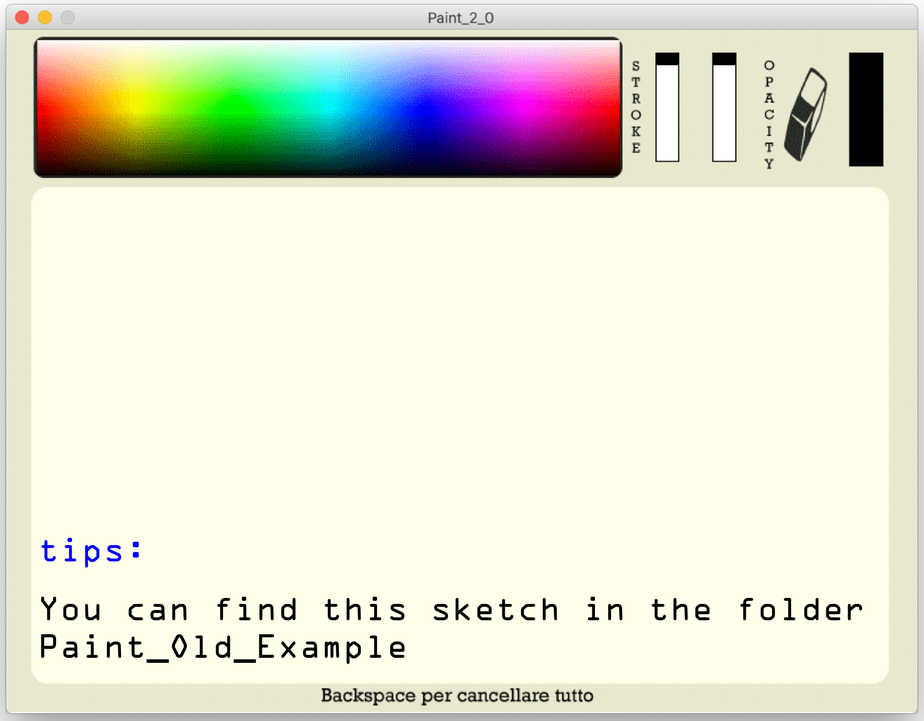
delete or comment the background at line 16 and the rectangle at line 17, uncomment line 19 and the “if” at line 22 to 24

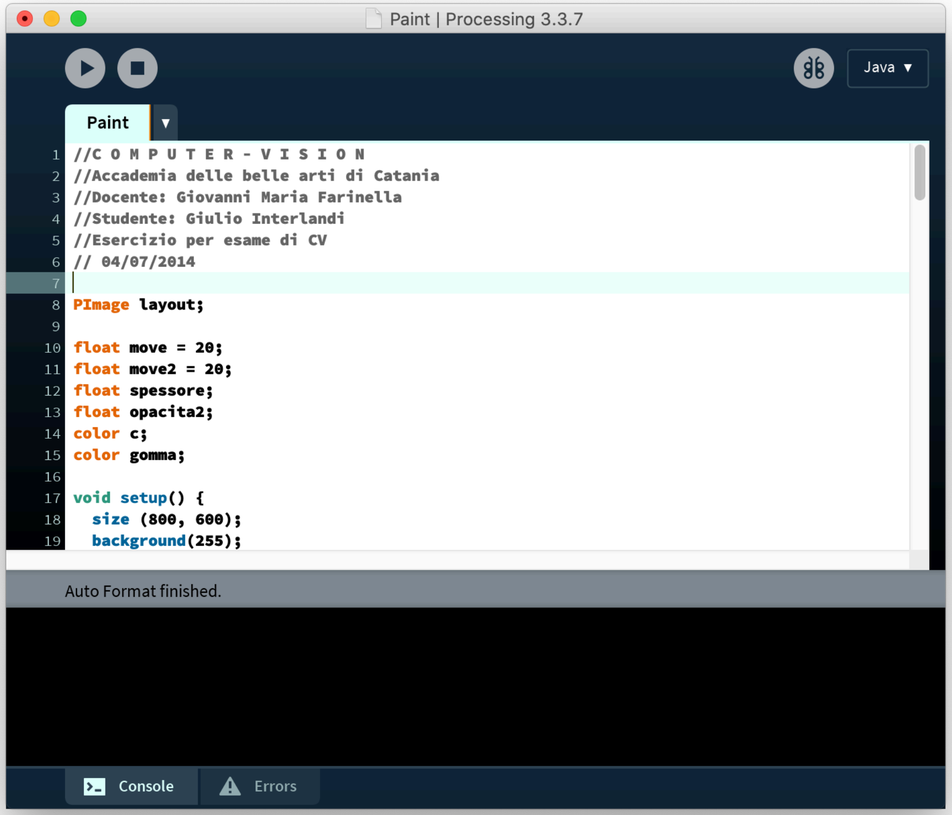
Be creative!

POLITECNICO MILANO 1863

Let’s create your own

that was my exercise some years ago





Let’s create a sketch

POLITECNICO
MILANO 1863

```
// open the folder of examples

sketch_5RANDOM_RELATIVE
```

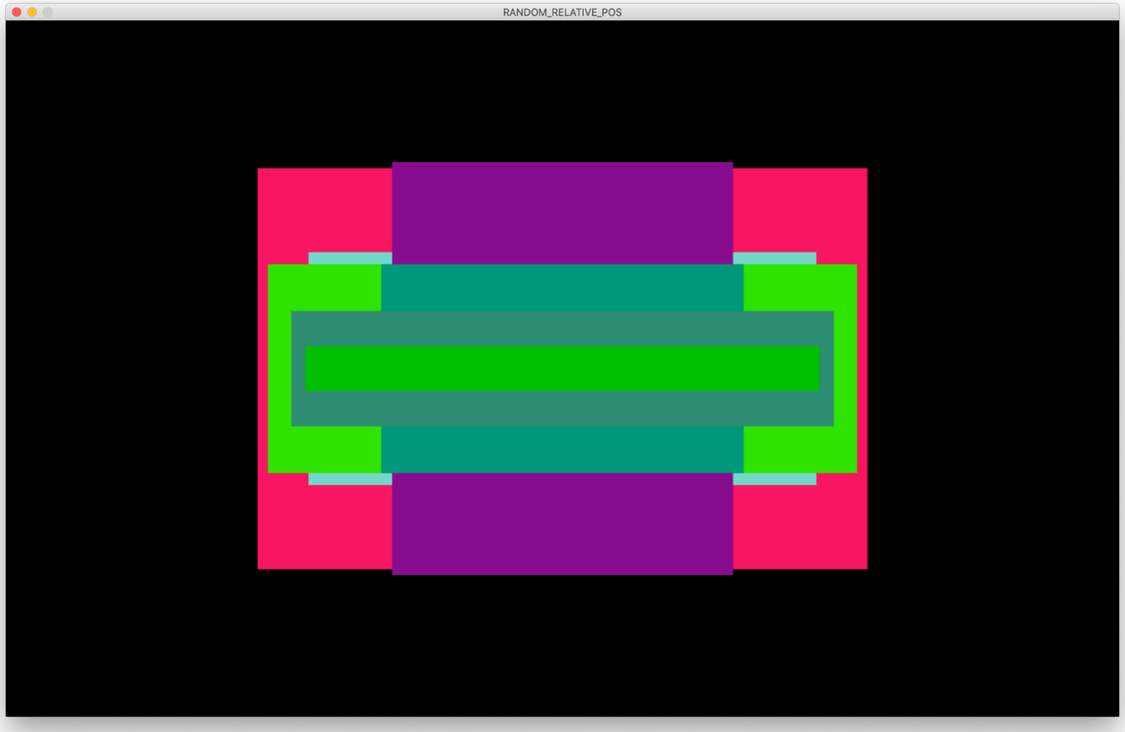
here you find

Random principles and coordinates

to understand randomness and positioning shapes without writing pixel

`tips:`

try to comment the filter at line 15 and delay at line 16



Frieder Nake's remixes

POLITECNICO MILANO 1863

```
// open the folder of examples

NAKE_REMIX
```

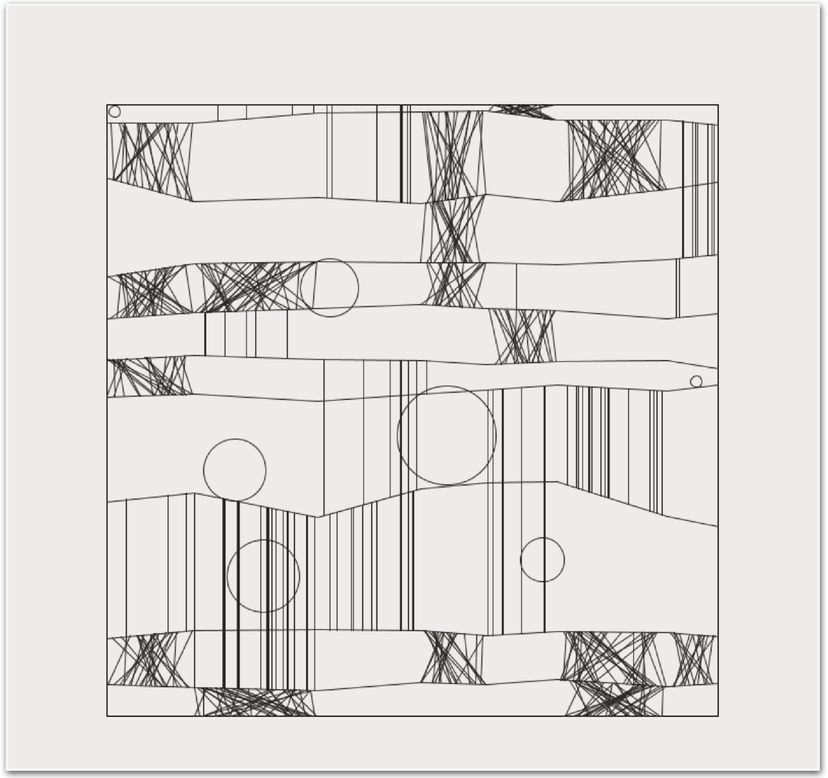
here you find

Remixes of Nake’s works and open processing platform

to understand how to manage more complex sketches and discover the open processing platform

`tips:`

try to understand how these sketches work, and make your own remix



Credits: Max Cura find it at Open Processing page

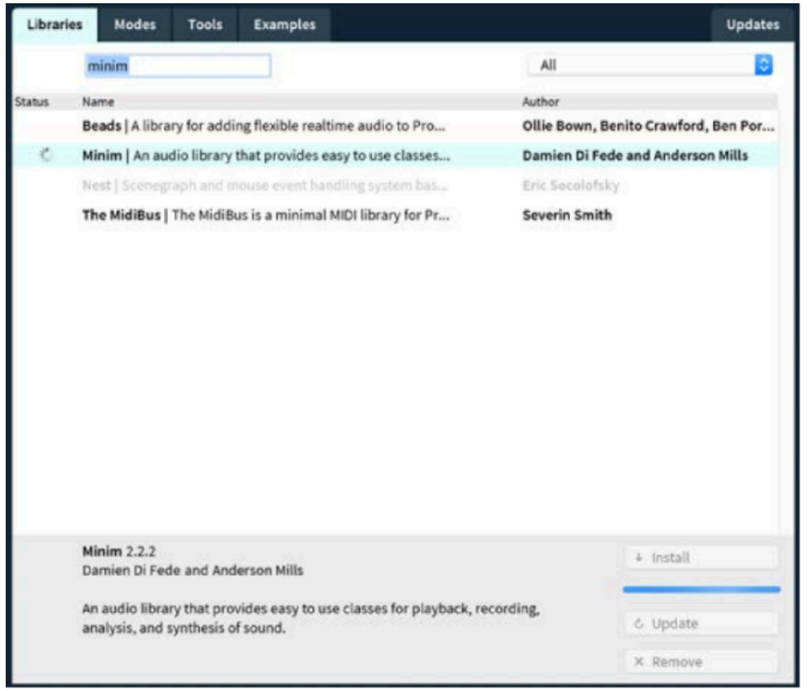
Sound Reactive

POLITECNICO
MILANO 1863

In order to work with sound information in Processing, we first need to **install an extension library** since Processing doesn't natively support this feature.

It can be simply installed by opening:

Sketch
➜
Import Library
➜
Add Library, typing 'minim' into the search field and then clicking on 'Install'.



You can find more in the document:

Processing-Generative_Design_Tutorial_soundmapping

Sound Reactive

POLITECNICO MILANO 1863

Use this code before void setup:

import ddf.minim.*; //import the library

Minim minim; //declare we are using minim

AudioInput in; //choose audio input mode

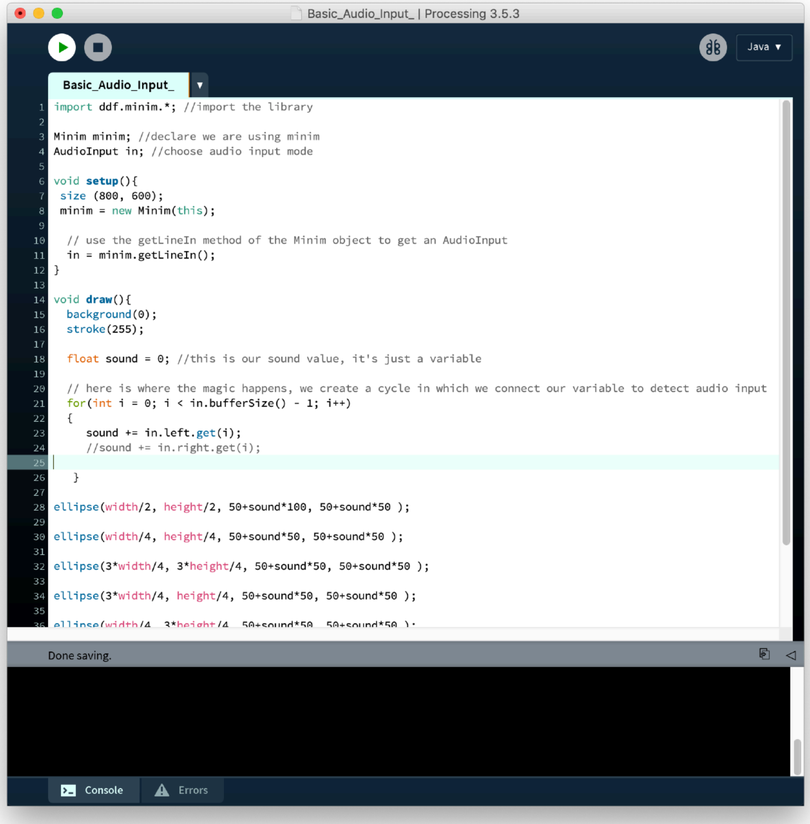
Use this code inside void setup:

minim = new Minim(this);

in = minim.getLineIn();

Use this code inside void draw:

```
float sound = 0;
 for(int i = 0; i < in.bufferSize() - 1; i++) {
   sound += in.left.get(i);
}
```



Connect audio input

now you can connect the variable sound you just created to your shapes

Equalizer

POLITECNICO
MILANO 1863

```
// open the folder of examples

sketch_6BASIC_AUDIO_INPUT
```

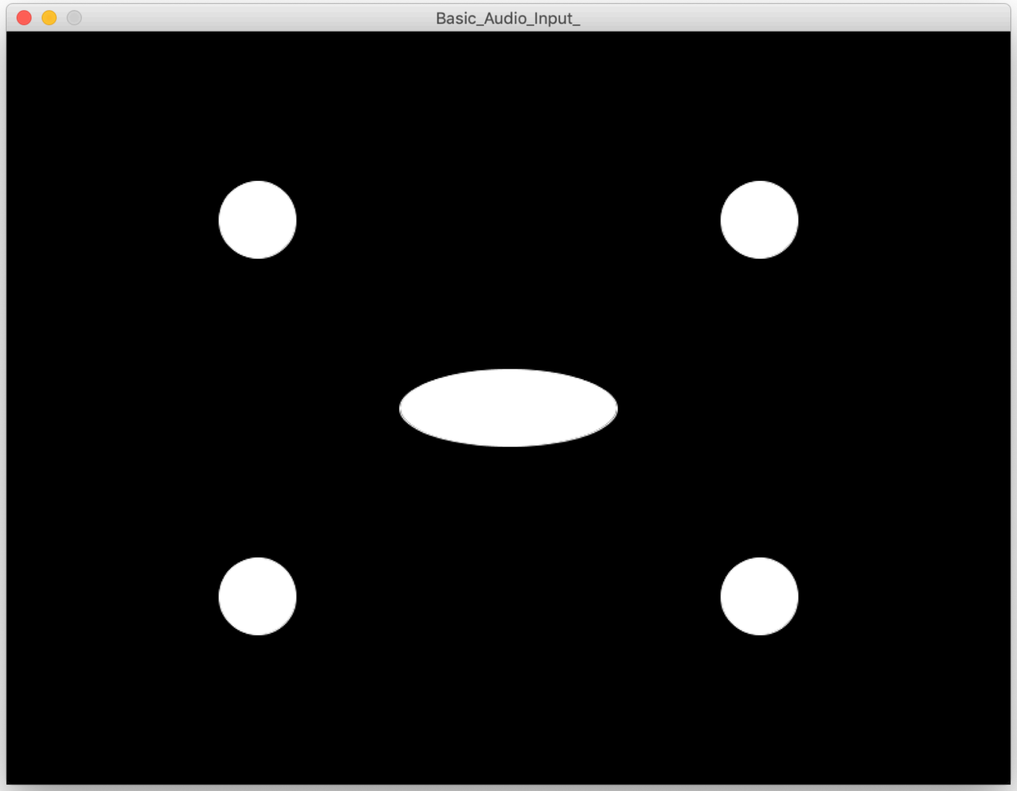
here you can find

Basic principles of sound reactivity

to understand how to import a library and how to implement sound reaction in shapes

`tips:`

try to have different reactions for different shapes, using math to adjust sensibility

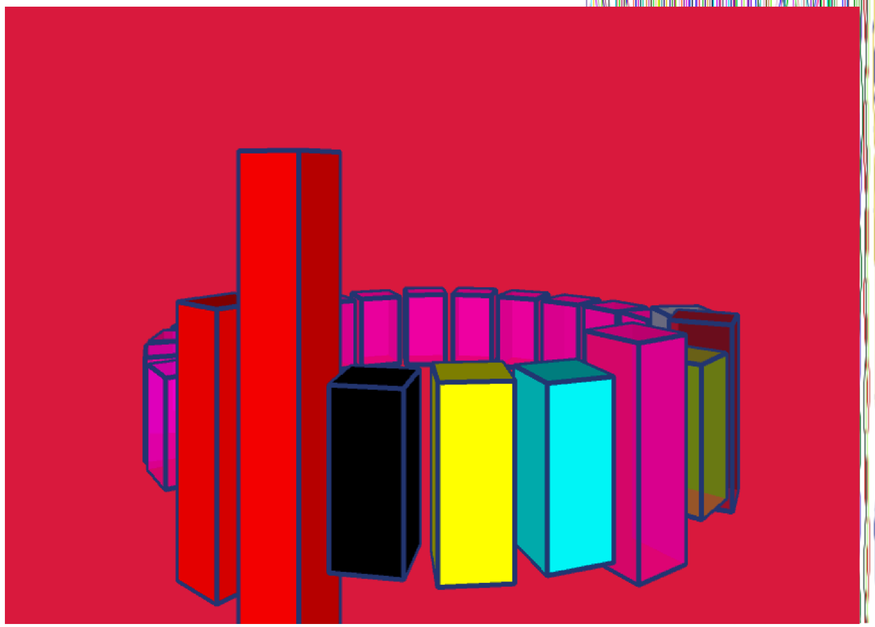


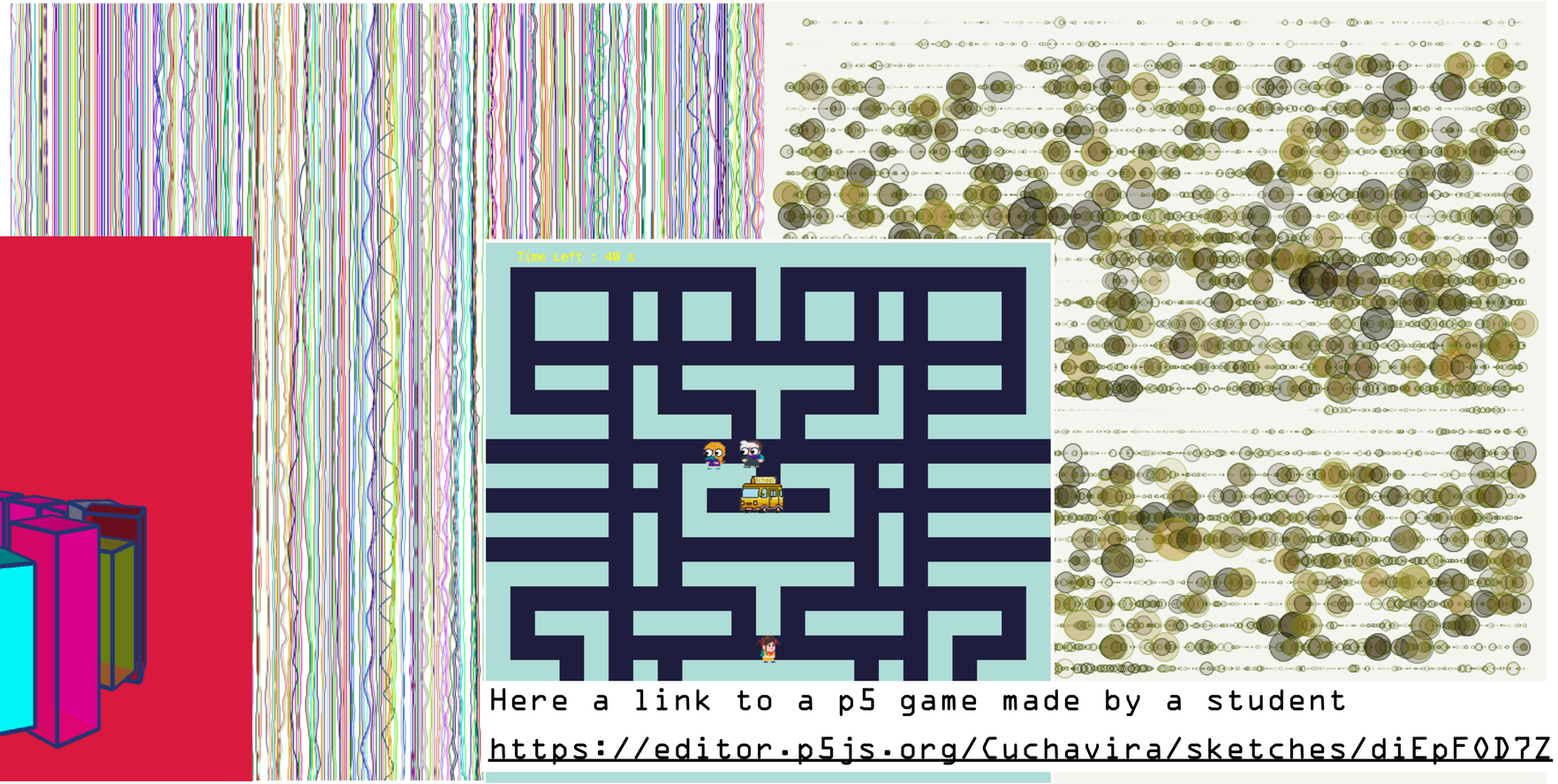
Be creative!

Let’s create your own!

here some sketches done by

student from past years





Here a link to a p5 game made by a student

https://editor.p5js.org/Cuchavira/sketches/diEpF0D7Z

// open the folder past years' examples

Credits: Luca Ghezzi, Andrea Zito, Hongni Ye

Kinect

Kinect® is a **motion sensing input device** produced by Microsoft. It was initially developed as a **gaming accessory**, but artists, third-party developers and researches found several after-market uses because of its **low-cost and advanced features**. There are 2 versions of Kinect commercially available, and a Kinect Azure Development Kit recently became available on the market.

OpenKinect is one of the libraries for interfacing the Kinect to Processing.
It can be simply installed by opening:

Sketch
➜
Import Library
➜
Add Library, typing ‘kinect’ into the search field and then clicking on ‘Install’.

You can find out more on:

https://shiffman.net/p5/kinect/

Kinect

How it works:

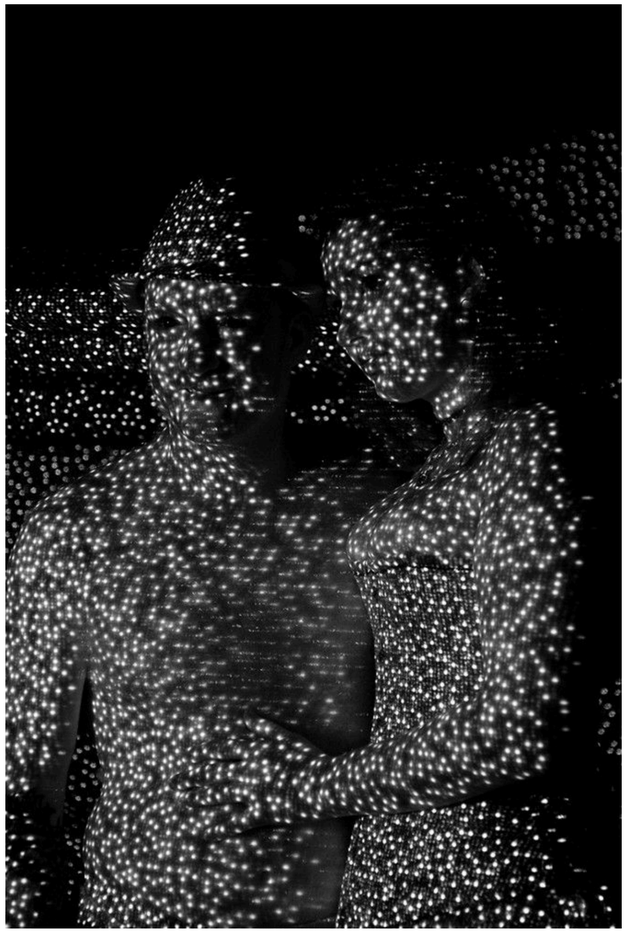
Kinect sensors can detect a three dimensional area thanks to its **infrared** technology. It works by projecting infrared dots on the scene, and calculating their distortion on the projected surface. The Kinect has 1 RGB camera, 1 infrared camera, one infrared emitter, and a microphone array.

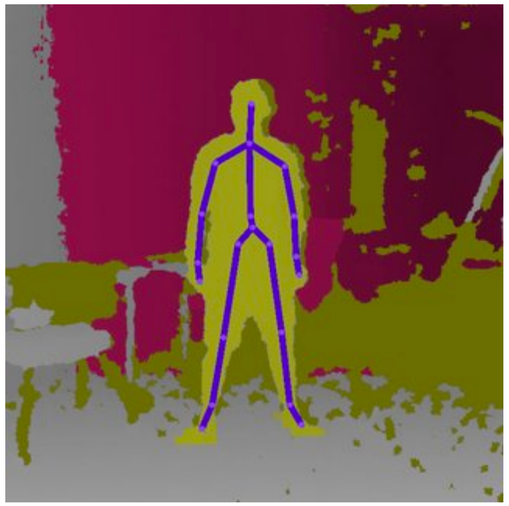
Features:

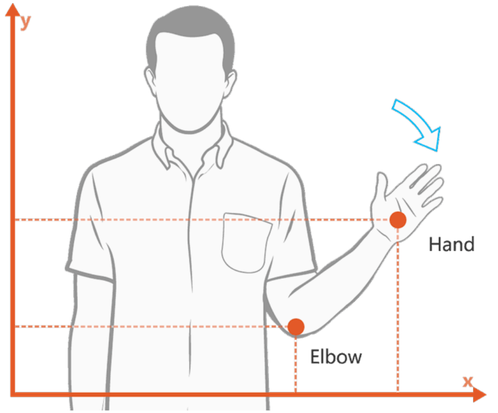
Depth map, it's possible to detect the depth of a scene in order to visualise silhouettes at a given distance.

Skeleton feature, is available for up to six users depending on the Kinect version.

Hands and Gestures recognition.







Kinect to Syphon

POLITECNICO
MILANO 1863

`// extra example for kinect v1`

`KINECT_SYPHON_2020`

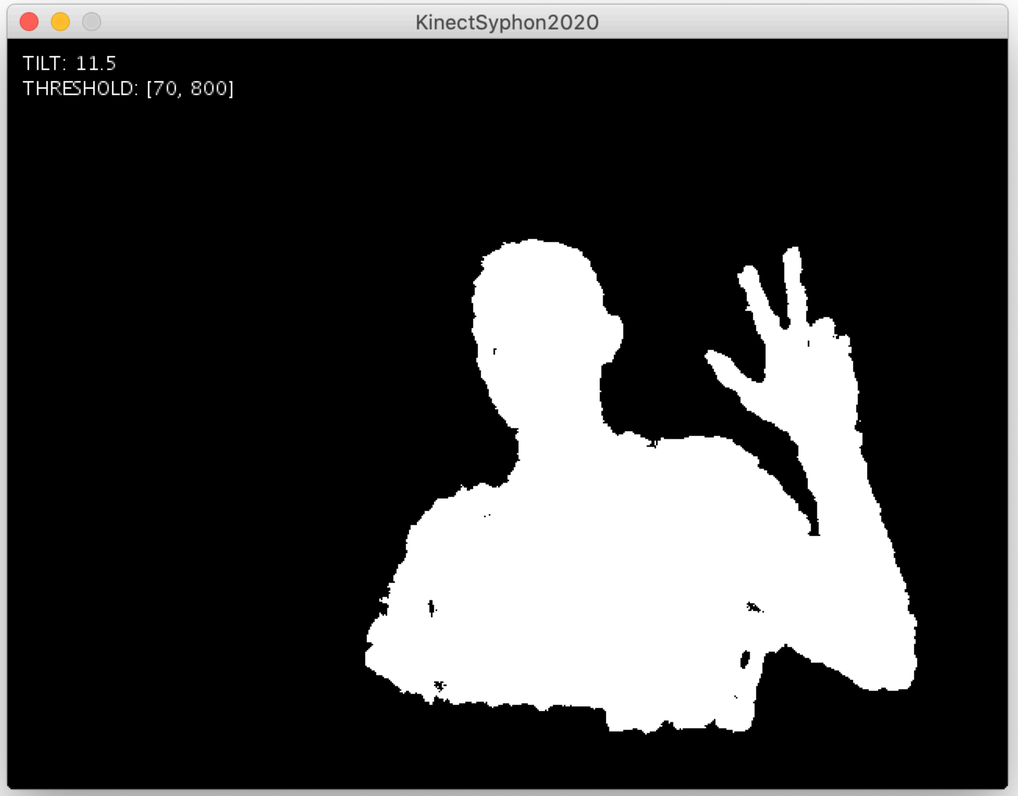
here you can find

Kinect depth feed to Syphon

to understand how to use the Kinect Depth Map feature and how to output it through a video sharing framework, in our case Syphon, and use it in a media server or mapping software (MadMapper, Resolume Arena).

`tips:`

try and change the minimum and maximum tilt and threshold values; these are very useful in the calibration process.



POLITECNICO
MILANO 1863

Thank You

giulio.interlandi@mail.polimi.it