Processing.org

Ph. Truillet

November 2018 – v. 2.5

1. introduction

Processing (**http://processing.org**) is a language (actually, another layout of the Java language) developed by Ben Fry and Casey Reas (MIT). Processign.org is dedicated to creative programming.
It allows among other things to create images (fixed or animated) and to interact with them. Processing.org also allows you to manipulate, generate texts, shapes, three-dimensional images (using OpenGL), sound, ... and more generally everything!
Processing can finally be extended using libraries or through "close" projects like Arduino (derived from Processing) (**http://www.arduino.cc**) and Wiring (**http://wiring.org.co**) in order to add some sensors.

There are multiple versions of Processing including "p5js" (**http://p5js.org**), "processing JS" (**http://processingjs.org**) ritten in javascript (using the syntax of Processing), Processing Python (**http://py.processing.org**), Processing R, etc ...

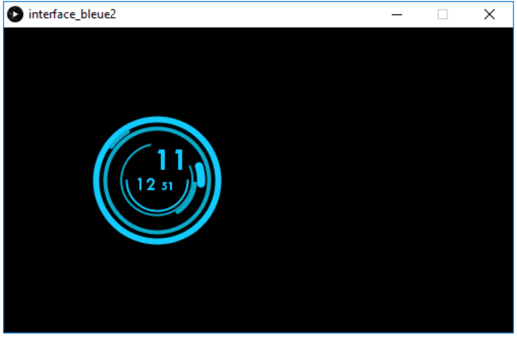


Figure 1: example of Processing sketch

Processing relies on the Java language, which allows it to run on Windows, MacOS X and Linux operating systems (including Raspbian on Raspberry Pi) and **Android** with the appropriate mode proposed with Processing 3).

Despite some shortcomings (more and more corrected), Processing has many advantages including its ease of use which makes it an almost ideal language for rapid prototyping!

2. how to install Processing

<u>Nota</u>: We will assume that Processing 3.4 is installed in the directory "C:/languages"

If this is not the case, the first thing to do is to go to **https://www.processing.org/download/?processing** and download one of the proposed versions (stable version 3.4 of July 26, 2018 in 32 or 64 bits).

Once downloaded, unzip the archive into the right directory of your choice and launch the "processing" executable.
First, determine the directory where your programs will be saved (sketches)
(see File|Preferences), choose the location in the field "sketchbook location" and press ok (see Figure 2).

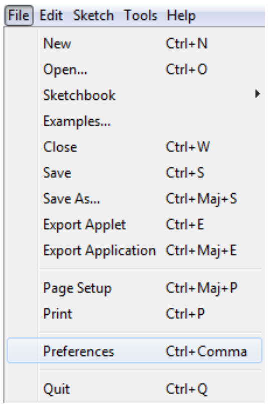
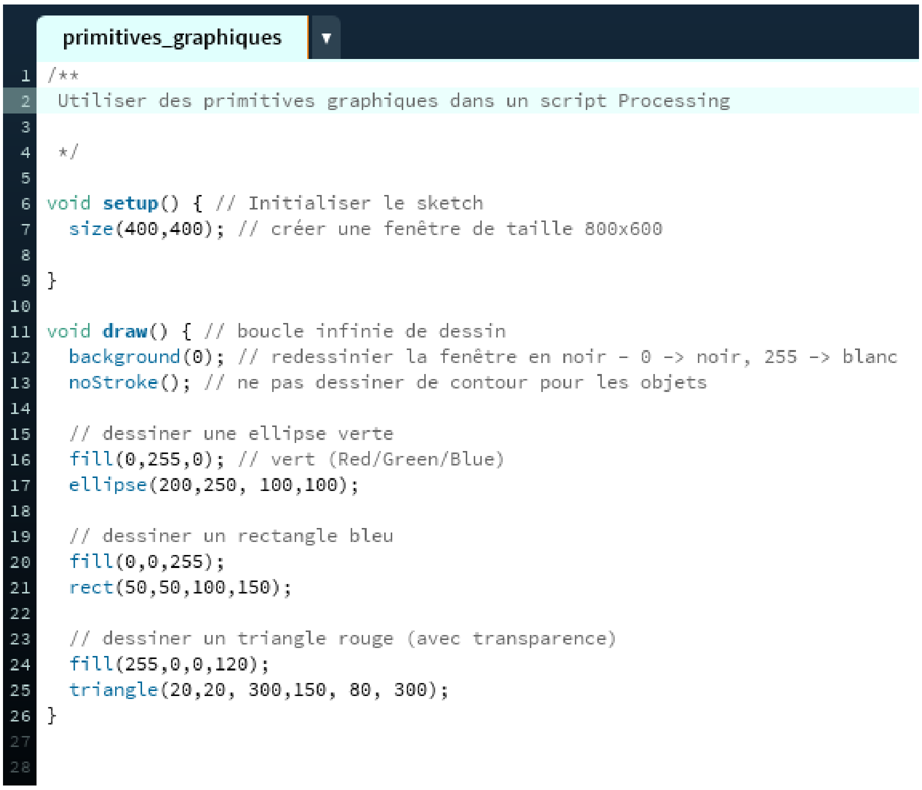


Figure 2: Menu

3. my first sketch

It is possible to use Processing in different ways, especially in "script" mode (sequences of actions executed once), and in the "continuous" mode (with an infinite loop – **draw()** function).
The "continuous" mode requires to implement at least two functions: ***setup()*** which initializes the variables and ***draw()***. This loop is used to display complex graphics animations, react to synchronous events from user actions or system events.

By convention, the reserved words are displayed in blue, green and orange in the IDE (see Figure 3).



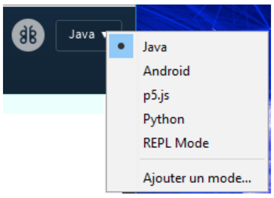
```
/**
 Utiliser des primitives graphiques dans un script Processing

 */

void setup() { // Initialiser le sketch
  size(400,400); // créer une fenêtre de taille 800x600

}

void draw() { // boucle infinie de dessin
  background(0); // redessinier la fenêtre en noir - 0 -> noir, 255 -> blanc
  noStroke(); // ne pas dessiner de contour pour les objets

  // dessiner une ellipse verte
  fill(0,255,0); // vert (Red/Green/Blue)
  ellipse(200,250, 100,100);

  // dessiner un rectangle bleu
  fill(0,0,255);
  rect(50,50,100,150);

  // dessiner un triangle rouge (avec transparence)
  fill(255,0,0,120);
  triangle(20,20, 300,150, 80, 300);
}
```

Figure 3: my first sketch

The possibilities of the Processing language are almost infinite especially with the possibility of using object-oriented programming, to add external libraries and to write them yourself!
Some exercises below will give you an overview of what can be done in a few lines of code.

The structure of a Processing program also makes it easy to implement **a state machine**!

The IDE Processing offers several modes: Java (default) but also Android (ADB must be installed), p5.js, Python, R and REPL (Read Eval Print Loop is in the form of "shell").
Just choose the mode (an installation may be required) that you want on the button to the right of the IDE.



4. exercices (basics)

4.1 some drawings

- Create a grayscale graphical composition using at least one ellipse, one line, and one rectangle
- Draw a simplified painting of the type "Mondrian" (father of the neoplasticism) or Sophie Taeuber-Arp – see examples here: **http://www.wikipaintings.org/en/piet-mondrian/composition-a-1923**
  and here: **http://deetrendss.esy.es/tag/sophie-taeuber-arp**

4.2 images

- Show two images with different shades
- Load a **png** file with a transparency bit and create a graphic composition by superimposing the layers.

4.3 typography

- Display your favorite quote with your favorite font
- Use two different fonts to simulate a dialogue between 2 users (justified sentences on the left for one and on the right for the other)

5. interact with class

5.1 standard inputs

- Use the arrow keys to change the position of a triangle displayed in the window
- Draw two shapes that react differently depending on the mouse actions

5.2 play movie

Download a video on Youtube in mp4 format (you can use "vlc portable" to do this or for example a specialized website such as `https://de.save-from.net`)

Download the **video** project (`https://github.com/truillet/international/blob/master/ostfalia/code/video.zip`) and change the code to manage rewind, pause and start functions by using keys on the keyboard.

5.3 Oriented Objet Programming

Download the **interface** project (`https://github.com/truillet/international/blob/master/ostfalia/code/Gestion_Objets.zip`) and modify the code in the way that the object changes color when clicked and returns to its original color when released (think about using the events *MouseDragged()*, *MousePressed()* and *MouseReleased()*).

6 network

6.1 RSS feed

Download the **RSS** project (`https://github.com/truillet/international/blob/master/ostfalia/code/rss.zip`), install it and open it. Run the code.
Modify the code in order to download the RSS feed of the newspaper "Le Monde" and to display the "front pages" in circles when the user clicks on the title of the newspaper.

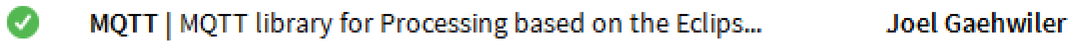
6.2 JSON

Now, download the **JSONWeather** project (`https://github.com/truillet/international/blob/master/ostfalia/code/JSONWeather.zip`), intall it ans open it.
Go to `http://home.openweathermap.org/users/sign_up` website and create a (free) account. After signing up, copy your **API key**.

In the Processing code, locate the line (query) where "**&APPID=**"is located and paste the key there. Run the code. Change the code to display a corresponding icon instead of the description and add a button to reload the weather for a different city.

6.3 MQTT

Install the MQTT library (Menu "**Tools**"|"**Add a tool outil**", tab "**Libraries**" MQTT.



✓ MQTT | MQTT library for Processing based on the Eclips... Joel Gaehwiler

Open the example installed with the library. Open an internet navigator and go the **shiftr.io** website at: **`https://shiftr.io/try`**

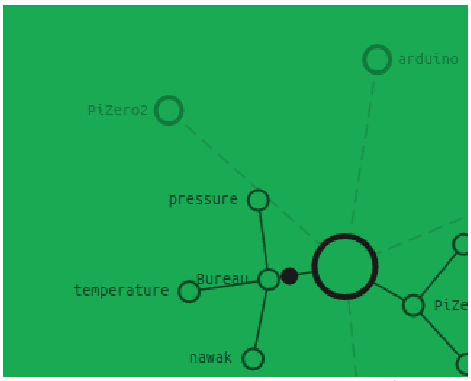


Figure 4: MQTT Architecture "Publish/Subscribe"

Write a Processing program that generates random temperature values every second and sends them to the MQTT broker (**you must determine the right topic**). Each instance launched will be considered as a different "room").

Then write a program that must subscribe to all the temperatures emitted by each instance and display the general average of the temperatures of the "house" (thus, generated by each room) and the different averages by "room".

7 image processing

7.1 "I am … your Father!"

First, install **OpenCV** library (**`https://github.com/atduskgreg/opencv-processing/releases`**) that you can find in menu *"Tools"*|*"Add a tool"*… than tab "Libraries".

Modify the example *LiveCam* that comes from with **OpenCV** library. Replace each face detected by a Dark Vador face picture.
(`https://github.com/truillet/international/blob/master/ostfalia/code/darth_vader.png`)

7.2 QRCode

From the example available here:
`https://github.com/truillet/international/blob/master/ostfalia/code/QRCode.zip`
Create a sketch that displays a 3D object on the detected QR code.

Nota: this sketch will use the **ZXing** library (**https://github.com/zxing/zxing**)
Nota2: to generate a QR Code from Processing, see the code below:
`https://github.com/truillet/international/blob/master/ostfalia/code/QRCode_Generator.zip`

7.3 Mixed reality - TopCodes

From the example downloaded here: `https://github.com/truillet/TopCode`
Create a new sketch which displays some information on physicals items that embed TopCodes when the user clicks on it through a webcam

Nota: this sketch uses the **TopCodes-Tangible Objet Placement Codes** library re-writing for Processing (see `http://users.eecs.northwestern.edu/~mhorn/topcodes` for the original version)

8. sensors and multimodality

8.1 Robot class

The java class Robot (`import java.awt.Robot`) allows to manipulate any item on the whole screen (Windows, MacOS ou Linux) by using key or mouse events.
Write a Processing sketch that takes a **screenshot** of the whole screen and saves it as a jpeg file.

8.2 ivy middleware

Download the example below
`https://github.com/truillet/international/blob/master/ostfalia/code/ivyP5.zip`

Unzip all sketches and launch them both. These two sketches use the ivy middleware to communicate on the local network (see `http://www.eei.cena.fr/products/ivy` for a more general information).

By clicking on the first window (sketch "sender"), a message will be displayed on the other window ("receiver") and a feedback is sent to the first.

Once the principle understood, write an application composed of several sketches (which can communicate on the local network) which can decode a QR-code presented in front of a webcam, displays the decoded information in another window (on another machine for example) and synthetize (via a speech synthesizer) the decoded text (we can use the **ttslib** library for example).

8.3 Arduino

First, download and install the Arduino IDE (**http://www.arduino.cc**) on your computer.

Donwload the example below:

`https://github.com/truillet/international/blob/master/ostfalia/code/arduino.zip`

Compile and upload the **capteur.ino** file on the arduino board plugged on your serial port.
Plug in an infrared LED on Analog pin **A0** and **GND**. (the code will first read a value from the sensor et write it on the serial port)

Execute the Processing code.
Change the code so that the value retrieved from a sensor is displayed as a vertical bar of up to 300 pixels (if the retrieved value is "1024" for instance).

9. useful links

- **Processing**: http://www.processing.org
- **P5.js**: http://p5js.org
- **studioSketchpad**: sri-upssitech-p5.sketchpad.cc

- **Reference**: http://processing.org/reference
- **Learning Processing**: http://www.learningprocessing.com
- **Hello Processing**: http://hello.processing.org

- **Lecture notes** : https://github.com/truillet/international/ostfalia
- **Libraries**: https://processing.org/reference/libraries/

- **OpenCV for Processing**: https://github.com/atduskgreg/opencv-processing/releases
- **TTSlib for Processing**: http://www.local-guru.net/blog/pages/ttslib