



Développer sur **Android** avec **Processing**

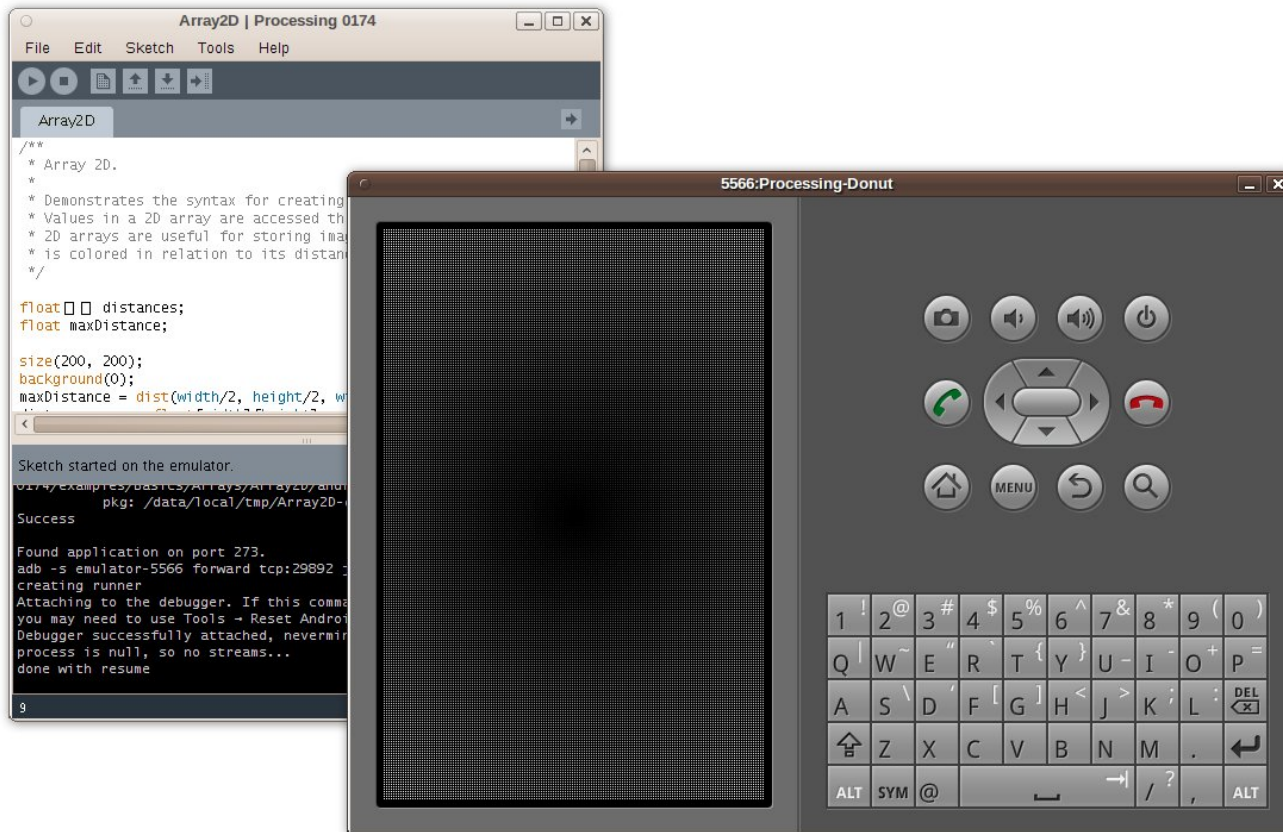


Auteur : jerome.cantaloube@gmail.com

Processing pour Android



Processing permet de créer facilement ses propres applications sur
Android



Processing pour Android



CC BY-NC-SA

SOMMAIRE :

- 1) Présentation de la plate-forme **Android**.
- 2) Présentation de l'outil de programmation **Processing**.
- 3) Configurer **Processing** et **Android**.
- 4) Initiation à **Processing**.
- 5) Créer ma première application.



L'objectif de ce cours est de fournir assez de matière et de motivation pour susciter l'envie de continuer l'exploration de la plate-forme **Android** et du langage de programmation **Processing**.



1. Présentation de la plate-forme Android :

Android est un système d'exploitation “**open source**” à destination des mobiles.
Il est basé sur Linux avec une interface de programmation Java.

La kit de développement (SDK) fourni est composé :

- d'un compilateur
- d'un debugger
- d'un émulateur.

Les applications (micro-logiciel) Android sont contenues dans des fichiers .apk (Android Package)

Pour en savoir plus : <http://www.tbray.org/ongoing/When/201x/2010/11/14/What-Android-Is>

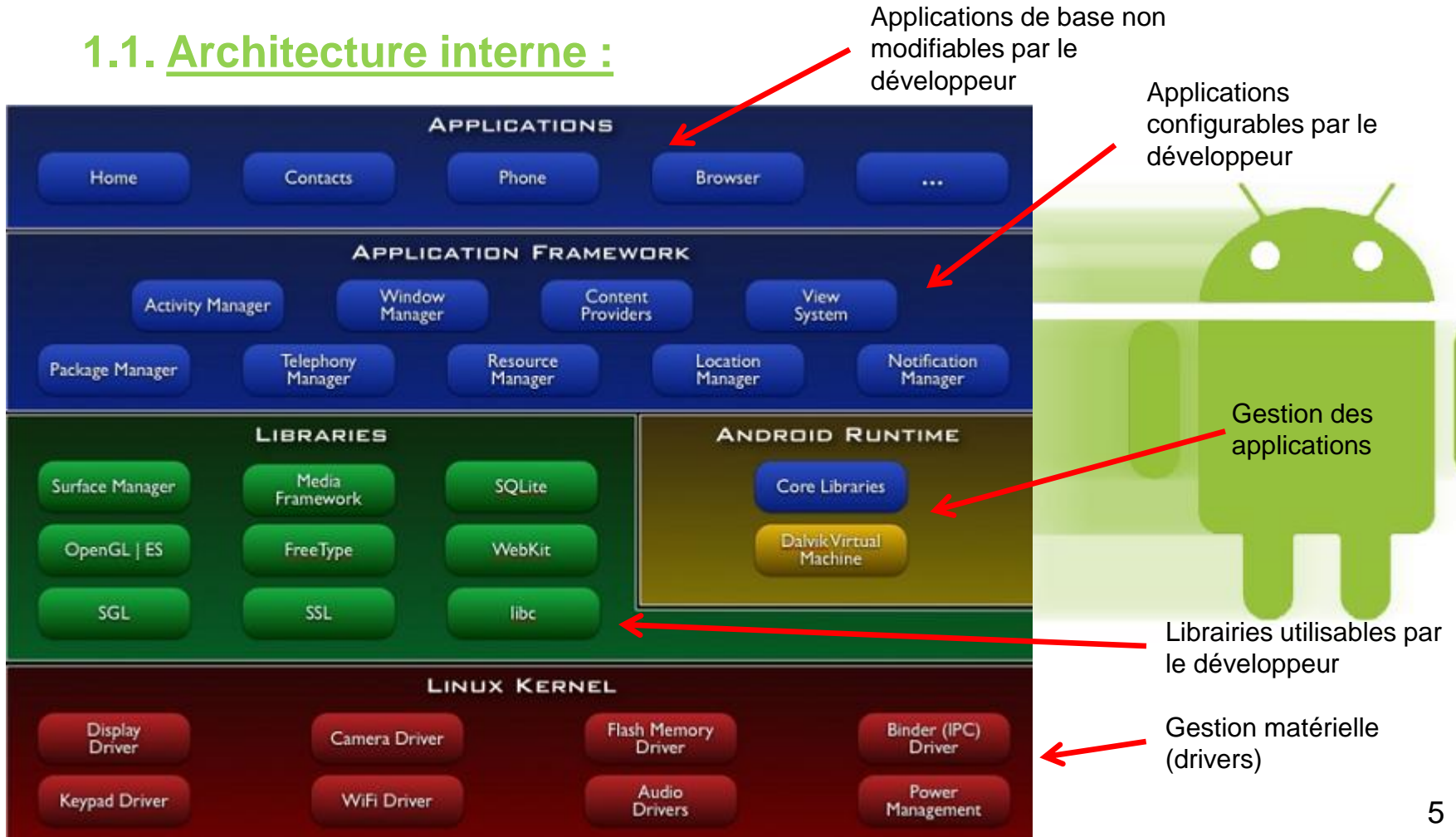


Processing pour Android



CC BY-NC-SA

1.1. Architecture interne :





1.2. Dalvik Virtual Machine :

Dalvik est une machine virtuelle (VM)

Dalvik est conçue pour faciliter l'exécution simultanée de plusieurs applications.

Avec Dalvik, 1 processus = 1 VM :

- Ainsi on garanti l'isolation des processus entre eux et avec le système
- une activité ou un service plante ? un seul processus plante
- la VM plante ? un seul processus disparaît



Pour en savoir plus : <http://www.youtube.com/watch?v=ptjedOZEXPM>



1.4. Compatibilité matérielle :

Pour être compatible avec l'OS **Android 2.1**, l'appareil doit être équipé :

- Affichage (Minimum QVGA 240x320 avec orientation portrait et paysage)
- Clavier (virtuel) et écran tactile (pas nécessairement multitouche).
- Touches de navigation (Home, menu and retour)
- USB (USB-A) et WIFI (minimum 200Kbit/sec)
- Caméra, Accéléromètre (3-axes), Gyroscope (3-axes) et GPS
- Téléphone
- Mémoire et stockage (mini 92Mb pour le kernel, 150Mb non-volatile pour les données utilisateurs)
- Stockage des applications (mini 2GB).



Processing pour Android



CC BY-NC-SA

1.5. Les fabricants :

HTC, Samsung, Motorola, LG, Sony Ericsson...

Pour avoir la liste complète : <http://www.androphones.com/all-android-phones.php>



Processing pour Android

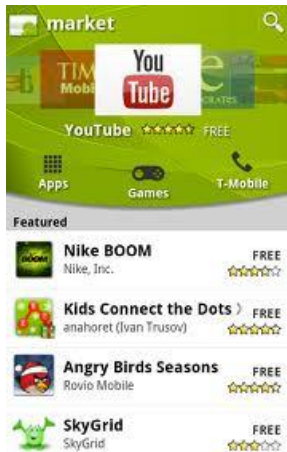


CC BY-NC-SA

1.6. Android Market :



Android Market est un site web développé par Google à destination des appareils Android.



Une application appelée "Market" est pré-installée sur tous les appareils Android et permet aux utilisateurs/développeurs de fournir ou de télécharger des applications publiées par d'autres développeurs et stockées sur Android Market.





2. Présentation de Processing :

Processing est un environnement de programmation « open source » pour ceux qui veulent créer des images, des animations et des interactions.

Initialement développé pour servir de portfolio informatique et enseigner les principes de base de la programmation dans un contexte visuel, Processing s'est développé pour devenir un véritable outil professionnel.

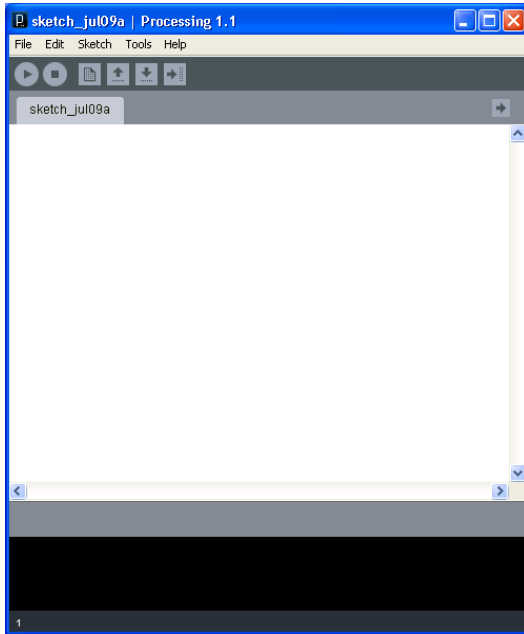
Aujourd'hui, des dizaines de milliers d'étudiants, des artistes, des designers, des chercheurs et les amateurs utilisent Processing.



Processing pour Android



2.1. Processing c'est léger !



Un Environnement de Développement minimaliste qui facilite la prise en main



Un langage de programmation proche du Java

```
void setup()
{
  size(200, 200);
  img = createImage(120, 120, ARGB);
  for(int i=0; i < img.pixels.length; i++) {
    img.pixels[i] = color(0, 90, 102, i%img.width * 2);
  }
}

void draw()
{
  background(204);
  image(img, 33, 33);
}
```

Processing pour Android



CC BY-NC-SA

2.2. Processing c'est facile !

Processing a été créé avec pour but de faire de la programmation accessible à tous, pour créer des graphismes, des interactions.



```
ColorWheel | Processing 1.2.1

ColorWheel

/**
 * Subtractive Color Wheel
 * by Ivo Greenberg.
 *
 * The primaries are red, yellow, and blue. The secondaries are green,
 * purple, and orange. The tertiaries are yellow-orange, red-orange,
 * red-purple, blue-purple, blue-green, and yellow-green.
 *
 * Create a shade or tint of the subtractive color wheel using
 * SHADE or TINT parameters.
 *
 * Updated 26 February 2010.
 */

int segs = 12;
int steps = 61;
float rotAdjust = TWO_PI / segs / 2;
float radius;
float segWidth;
float interval = TWO_PI / segs;

void setup() {
  size(200, 200);
  background(127);
  smooth();
  ellipseMode(RADIUS);
}
```

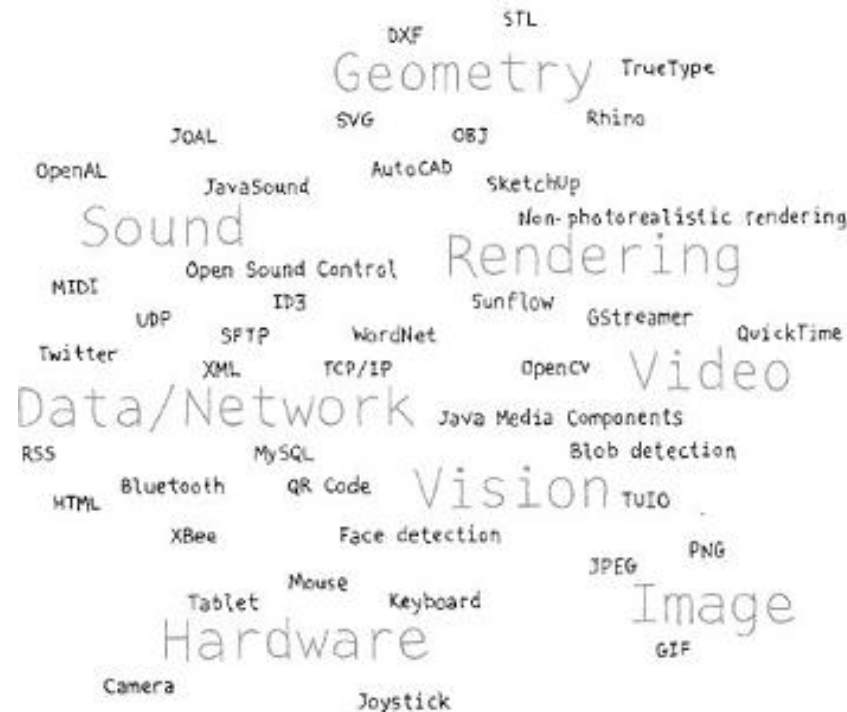
Processing pour Android



CC BY-NC-SA

2.3. Processing c'est flexible !

Processing peut traiter n'importe quel type d'information.



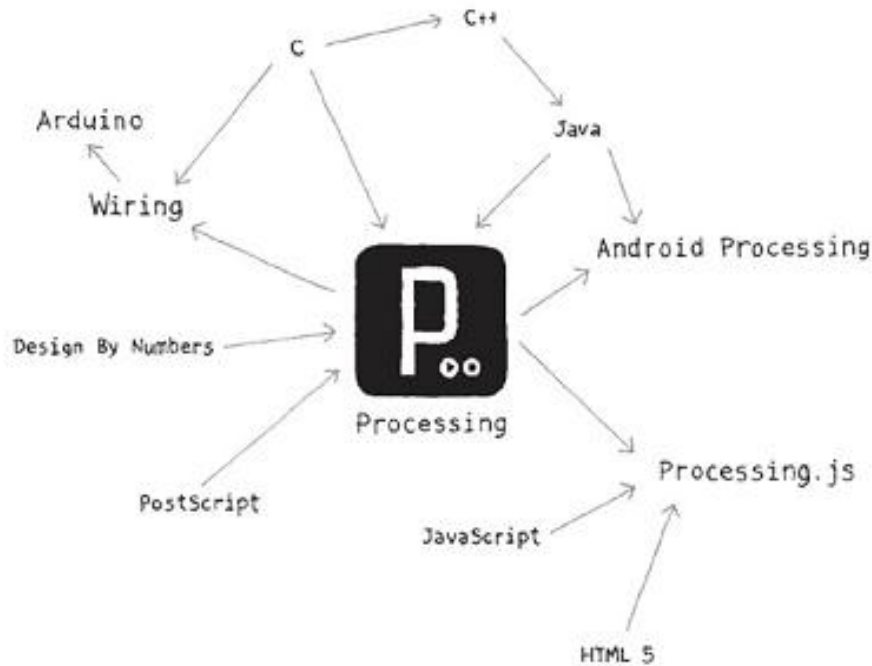
Processing pour Android



CC BY-NC-SA

2.4. Processing c'est modulaire !

Processing c'est aussi pour le développement Web, hardware, ...



Processing pour Android



2.5. Processing pour l'éducation, l'art, ...

OpenProcessing

Classrooms

Collections

sketches from classes on OpenProcessing.
<http://openprocessing.org/collections/>



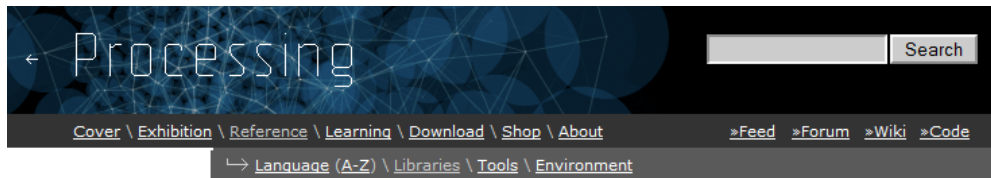
<Shadow Monsters>, 2005, Philip Worthington
<http://worthersoriginal.com/>

Processing pour Android



2.6. Libraries Processing

Plus de 100 bibliothèques externes développées par la communauté :



Libraries. Extending Processing beyond graphics and image, libraries enable audio, video, and communicating with other devices.

The following libraries are included with the Processing software. To add one to a project, select its name from the "Import Library..." option in the Sketch menu. These libraries are open source; the code is distributed with Processing.

Video

Interface to Apple's QuickTime for using a camera, playing movie files, and creating movies.

Network

Sending and receiving data via the Internet through the creation of simple clients and servers.

Serial

Supports sending data between Processing and external hardware via serial communication (RS-232).

PDF Export

Generates PDF files.

OpenGL

Support for exporting OpenGL accelerated sketches. Utilizes the JOGL library.

» Minim

Uses the JavaSound API to provide an easy-to-use audio library. A simple API while still providing a reasonable amount of flexibility for more advanced users.

DXF Export

Lines and triangles from P3D or OPENGL rendering modes can be sent directly to a DXF file.

» Arduino

Allows direct control of an Arduino board through Processing.

» Netscape.JavaScript

Methods for interfacing between Javascript and Java Applets exported from Processing.

Candy SVG Import

This library has been moved to the Processing core code as of version 149. To load SVG files, check out the reference for PShape(), loadShape(), and shape().

XML Import

This library has been moved to the



3. Configuration de Processing et d'Android :

L'Android **SDK** (Software Development Kit) inclut un jeu complet d'outils :

- Debugger
- Bibliothèques
- Emulateur
- Documentations, exemples de codes et tutoriels.

Multiplateforme : Linux, MACOS, Windows.

Officiellement, l'environnement de développement intégré (IDE) est **Eclipse**

=> **Processing** est nettement plus facile à prendre en main qu'Eclipse



Processing pour Android



3.1 Installer Android SDK

Installer dans un premier temps le JDK (Java Development Kit) :

<http://www.oracle.com/technetwork/java/javase/downloads/jdk-7u3-download-1501626.html>

Télécharger le Android SDK à partir site officiel.

Nous allons traité uniquement la version pour Windows.

Télécharger le fichier : [installer_r16-windows.exe](#).

android developers

Home SDK Dev Guide Reference Resources Videos Blog

Android SDK Starter Package

Download

Installing the SDK

Downloadable SDK Components

Adding SDK Components

- Android 4.0.x Platform **new!**
- Android 3.2 Platform
- Android 3.1 Platform
- Android 3.0 Platform
- Android 2.3.4 Platform
- Android 2.3.3 Platform
- Android 2.2 Platform
- Android 2.1 Platform
- Other Platforms

SDK Tools, r16 **new!**

Google USB Driver, r4

Support Package, r6 **new!**

Download the Android SDK

Welcome Developers! If you are new to the Android SDK, please read the steps below, for an overview of how to set up the SDK. If you're already using the Android SDK, you should update to the latest tools or platform using the Android SDK and AVD Manager starter package. See [Adding SDK Components](#).

Platform	Package	Size	MD5 Checksum
Windows	android-sdk_r16-windows.zip	29562413 bytes	6b926d0c0a871f1a9466525984701a
	installer_r16-windows.exe (Recommended)	29561554 bytes	3521dda4904886b0598059083c34469
Mac OS X (intel)	android-sdk_r16-macosx.zip	26158334 bytes	d1dc2b6f13eed5e3ce5c26c4e4c47aa
Linux (386)	android-sdk_r16-linux.tgz	22048174 bytes	3ba457731d51da3741c29c8930a4583

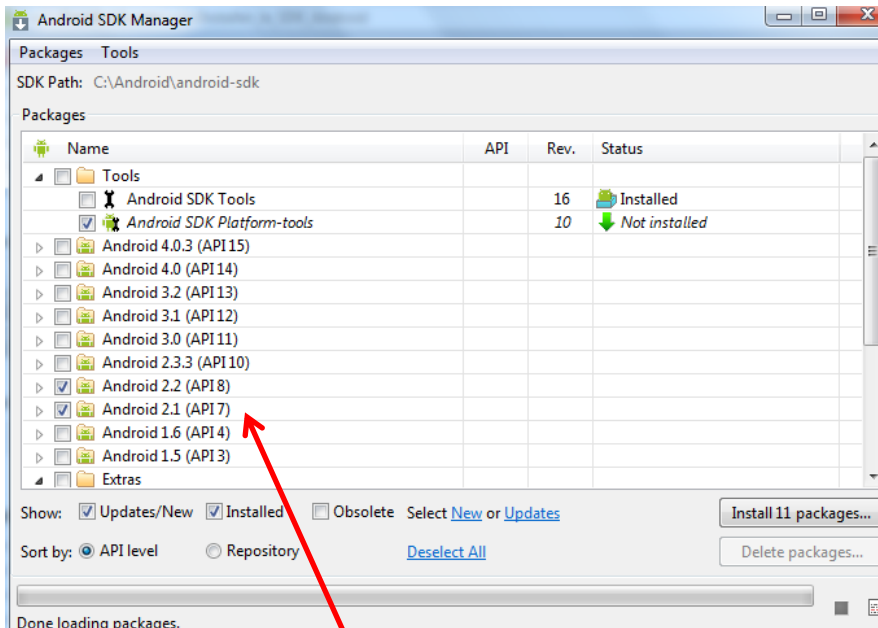
Pour plus d'informations :

<http://developer.android.com/sdk/installing.html>

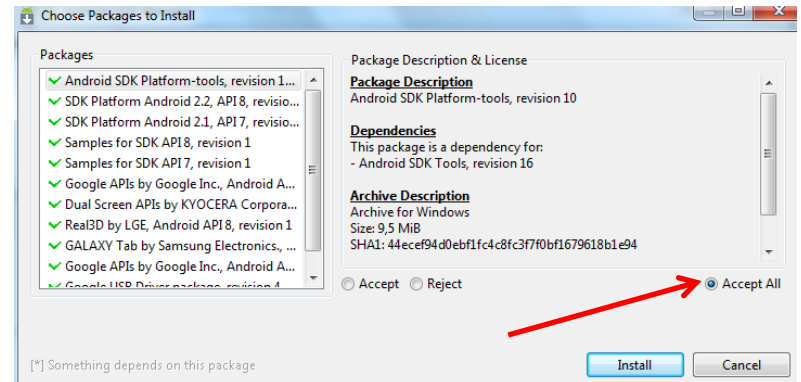
Processing pour Android



Lancer l'Android SDK et télécharger les modules suivants :



En fonction de votre appareil Android



Installer aussi le module USB driver (attention à liste des appareils supportés, voir détail : <http://developer.android.com/sdk/win-usb.html>)

Télécharger, si besoin, le driver de votre appareil : <http://developer.android.com/sdk/oem-usb.html>

Processing pour Android



CC BY-NC-SA

Ajouter une variable d'environnement :

Sous Windows, clic droit sur Mon Ordinateur (Mon poste de travail), et sélectionner Propriétés.

Sur l'onglet avancé, appuyez sur le bouton variable d'environnement. Dans la boîte de dialogue qui arrive, double-clic sur la variable système Path.

Allez à la fin de la ligne, ajoutez un point virgule ";" suivi du chemin complet du répertoire exemple : C:\Android\android-sdk\tools" à la variable système Path.

(conseil : rebooter Windows)

The screenshot shows the Windows 7 System Properties dialog box, Environment tab. The 'Variables d'environnement' section is active, showing a list of system variables. A 'Nouvelle...' button is clicked, opening a 'Nouvelle variable utilisateur' dialog box. The 'Nom de la variable' is 'PATH' and the 'Valeur de la variable' is 'C:\Android\android-sdk\tools'.

Variable	Valeur
MOZ_PLUGIN_P...	C:\Program Files\Foxit Software\Foxit R...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

Variable	Valeur
AMDAPPSDKROOT	C:\Program Files\AMD APP\
CLASSPATH	.;C:\Program Files\Java\jre7\lib\ext\QT...
ComSpec	C:\Windows\system32\cmd.exe
COSMOSM	C:\Program Files\SolidWorks Corp\COS...

Nom de la variable :	Valeur :
PATH	C:\Android\android-sdk\tools

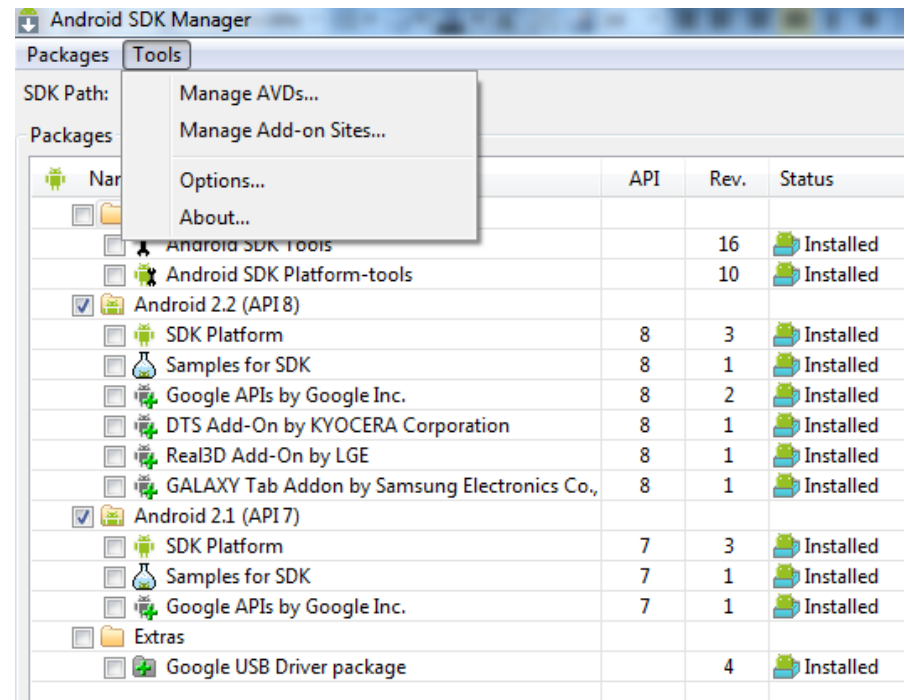
Processing pour Android



3.2 Configurer Android AVD

Android AVD (Android Virtual Device) est l'émulateur qui va nous servir par la suite à tester nos programmes.

Lancer l'outil "Manage AVDs..."



Processing pour Android



Créer (New...) votre propre AVD en fonction de votre appareil puis lancer l'émulation (Start...)

Android Virtual Device Manager

List of existing Android Virtual Devices located at C:\Users\jerome\.android\avd

AVD Name	Target Name	Platform	API Level	CPU/ABI
✓ Xperia_X10_mini	Android 2.1	2.1	7	ARM (armeabi)

Edit Android Virtual Device (AVD)

Name: Xperia_X10_mini

Target: Android 2.1 - API Level 7

CPU/ABI: ARM (armeabi)

SD Card:

- Size: 1 GiB
- File: Browse...

Snapshot: Enabled

Skin: Built-in: QVGA

Resolution: x

Hardware:

Property	Value
Abstracted LCD density	120
Max VM application hea...	24

Override the existing AVD with the same name

Buttons: New..., Edit..., Delete..., Repair..., Details..., Start..., Refresh, Edit AVD, Cancel

Legend: ✓ A valid Android Virtual Device. ✎ A repairable Android. ✗ An Android Virtual Device that failed to load. Click 'Details' for more information.

Processing pour Android



CC BY-NC-SA



Attention cela prend quelques minutes avant que l'émulateur soit complètement lancé.

Processing pour Android



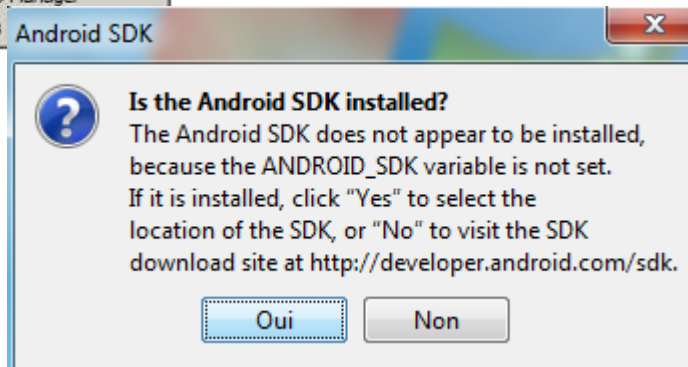
3.3 Installer Processing

A l'heure actuelle, la version de Processing qui prend en charge Android est la 2.0b3 (<http://processing.org/download/>)

Une fois installé mettre Processing en mode Android :



Un message apparait pour créer une variable d'environnement ANDROID_SDK, cliquer sur oui et indiquer l'emplacement du fichier demandé



Download Processing: Processing is available for Linux, Mac OS X, and Windows. Select your choice below to download the software.

THE PROCESSING SOFTWARE IS PROVIDED TO YOU "AS IS," AND WE MAKE NO EXPRESS OR IMPLIED WARRANTIES WHATSOEVER WITH RESPECT TO ITS FUNCTIONALITY, OPERABILITY, OR USE, INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. WE EXPRESSLY DISCLAIM ANY LIABILITY WHATSOEVER FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL OR SPECIAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST REVENUES, LOST PROFITS, LOSSES RESULTING FROM BUSINESS INTERRUPTION OR LOSS OF DATA, REGARDLESS OF THE FORM OF ACTION OR LEGAL THEORY UNDER WHICH THE LIABILITY MAY BE ASSERTED, EVEN IF ADVISED OF THE POSSIBILITY OR LIKELIHOOD OF SUCH DAMAGES.

By downloading the software from this page, you agree to the specified terms.

1.5.1 | 15 May 2011

↓ Linux ↓ Windows
↓ Mac OS X ↓ Windows (Without Java)*

The [list of revisions](#) covers the differences between releases in detail. Please read the [changes](#) if you're new to the 1.0 series. Also check the [known problems](#) for this release.

* The Windows version without Java is for users who can take care of installing a JDK (not JRE) themselves. It should only be downloaded by advanced users who are familiar with Java.

Processing is Open Source Software. The IDE (Processing Development Environment) is released under the GNU GPL (General Public License). The export libraries (also known as 'core') are released under the GNU LGPL (Lesser General Public License). There's more information about Processing and Open Source in the [FAQ](#), and more information about the [GNU GPL](#) and [GNU LGPL](#) at opensource.org. Please contribute to Processing!

[About the releases and their numbering](#)

Pre-Releases

The 2.0 alpha releases contain significant changes, be sure to [read about them](#).
2.0e4 | 2011 11 10 [Windows \(without Java\)](#) | [Mac OS X](#) | [Linux x86](#) | [bug fixes](#)
2.0e3 | 2011 11 05 [Windows \(without Java\)](#) | [Mac OS X](#) | [Linux x86](#) | [pre-release](#)

Resources

- [Tutorials](#)
- [Examples](#)
- [FAQ](#)
- [Troubleshooting](#)
- [Supported Platforms](#)
- [Processing Wiki](#)
- [Processing Forum](#)
- [Report a bug](#)
- [Download Source](#)

Announcements

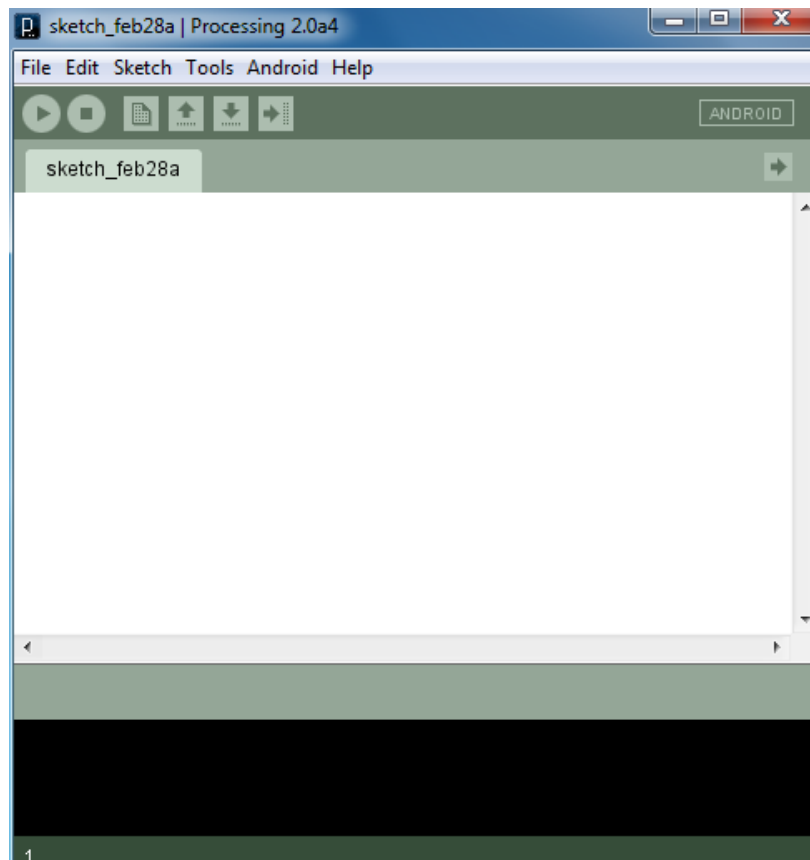
Email address

If you are interested in receiving updates about Processing, submit your email through this form. Your email will only be used to send infrequent updates about Processing. It will not be sold or shared.

Processing pour Android



Processing en mode Android (changement de couleur !)



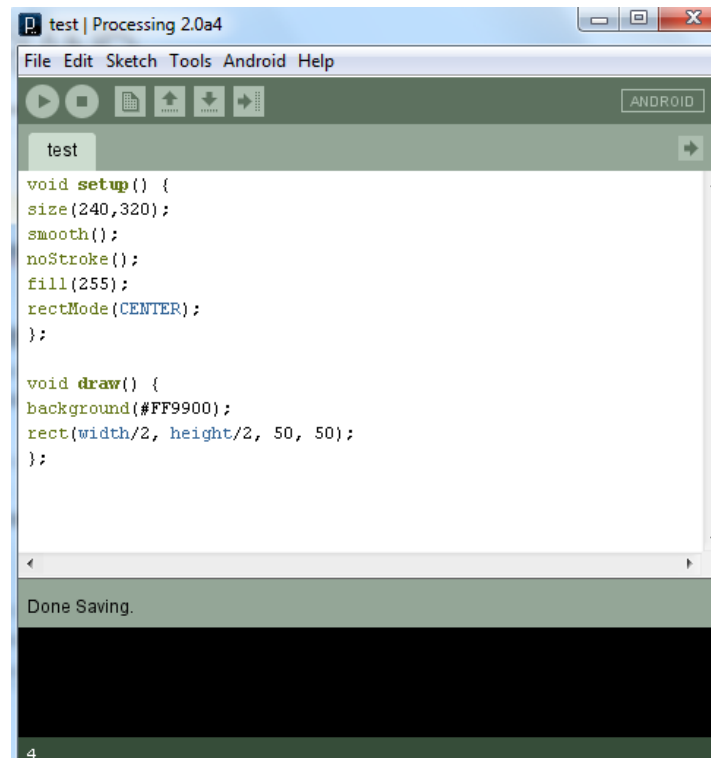


3.4 Tester l'émulateur Android

Nous allons réaliser notre premier programme ("sketch") afin de tester l'émulateur.

Code de test :

```
void setup() {  
  size(240,320);  
  smooth();  
  noStroke();  
  fill(255);  
  rectMode(CENTER);  
};  
  
void draw() {  
  background(#FF9900);  
  rect(width/2, height/2, 50, 50);  
};
```



Processing pour Android



CC BY-NC-SA

Lancer l'émulation depuis Processing (Ctrl+R) (l'émulateur doit être lancé avant, voir diapo 23)

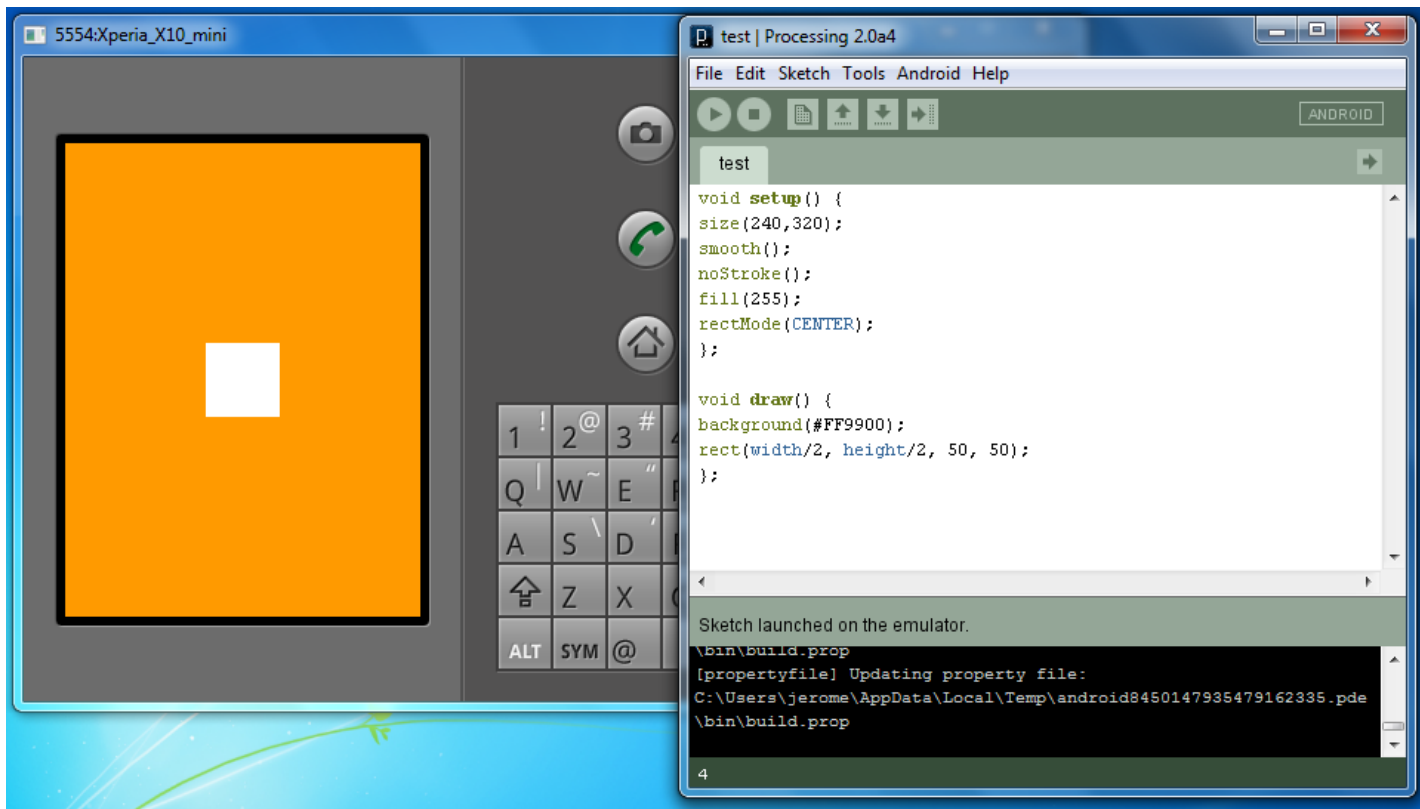


Processing pour Android



CC BY-NC-SA

Résultat de l'émulation



Processing pour Android



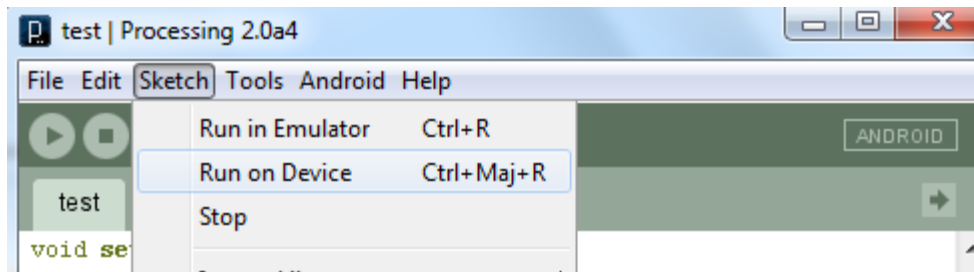
3.5 Tester votre appareil Android

Après avoir installer le **driver** de votre appareil

Connecter votre appareil au port **USB** de votre PC

Paramétrer votre téléphone pour qu'il autorise le mode "**débogage USB**"
(Paramètres\Paramètres des applications\Développement\Débogage USB)

Puis exécuter le programme vers l'appareil : Ctrl+Maj+R



Processing pour Android



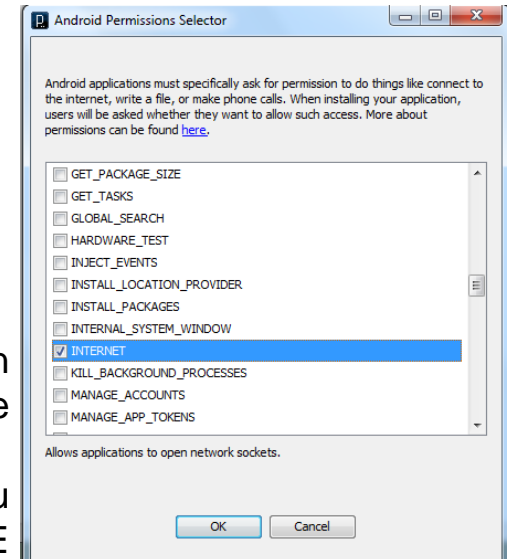
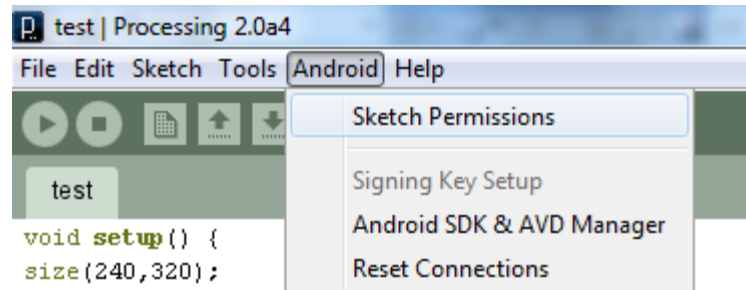
CC BY-NC-SA



Processing pour Android



Les permissions :



Si vous voulez charger des données d'Internet, ou se connecter à un serveur, vous devrez autoriser la permission INTERNET pour votre Sketch.

Si vous voulez utiliser des méthodes comme `saveStrings()` ou `createWriter()`, vous devrez permettre `WRITE_EXTERNAL_STORAGE` pour que vous puissiez sauver des choses à la mémoire flash incorporée.

Il existe des permissions pour accéder au téléphone, à la boussole, etc.

Pour en savoir plus :

<http://developer.android.com/reference/android/Manifest.permission.html>



4. Initiation à Processing:

Il existe de nombreux tutoriels sur le web :

- <http://processing.org/learning>
- <http://www.ecole-art-aix.fr/rubrique81.html>
- http://fr.flossmanuals.net/processing/ch043_processing-dans-lenseignement
- http://www.mon-club-elec.fr/pmwiki_mon_club_elec/pmwiki.php?n=MAIN.OUTILSProcessing

Dont certains dédié à l'utilisation d'Android :

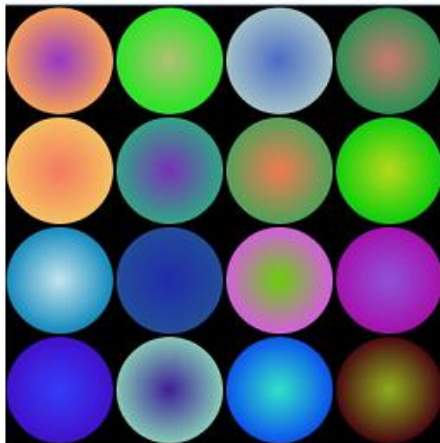
- <http://wiki.processing.org/w/Android>
- <http://forum.processing.org/android-processing>

Processing pour Android



CC BY-NC-SA

4.1 Dessiner en 2D et gérer les couleurs



<http://processing.org/learning/basics/radialgradient.html>

Functions and Parameters

```
size(), point(), line(), Triangle(),  
quad(), rect(), ellipse(), arc(), vertex()
```

Color

```
Color(), Colormode(), fill(), stroke(), Background()
```

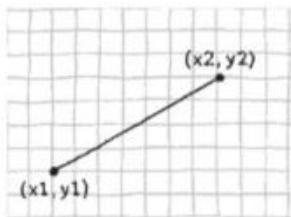
fill() : remplissage
stroke() : contour
Background() : arrière plan

Processing pour Android

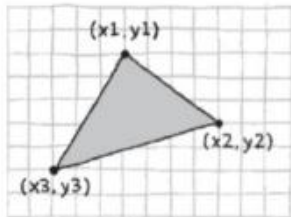


CC BY-NC-SA

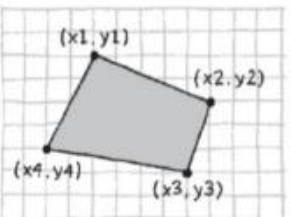
4.1 Dessiner en 2D et gérer les couleurs



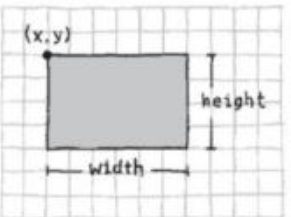
```
line(x1, y1, x2, y2)
```



```
triangle(x1, y1, x2, y2, x3, y3)
```



```
quad(x1, y1, x2, y2, x3, y3, x4, y4)
```



```
rect(x, y, width, height)
```

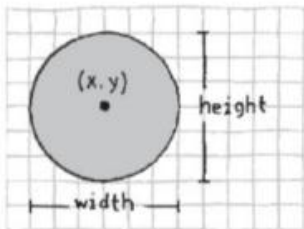
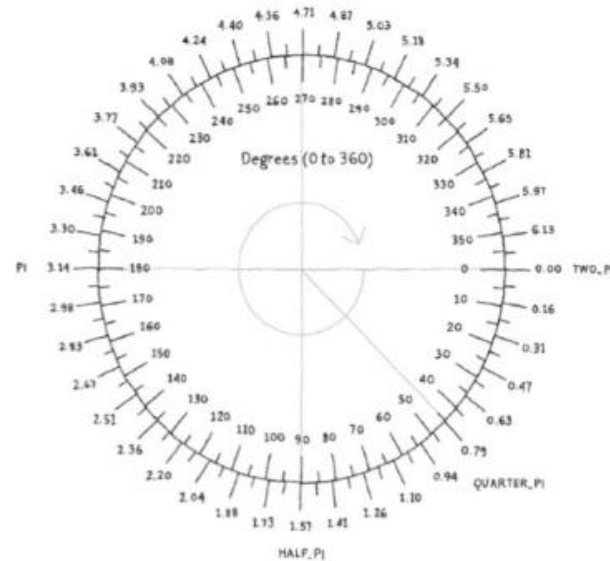
Casey Reas and Ben Fry.
<Getting Started with Processing>.
O'Really Media, 2010

Processing pour Android

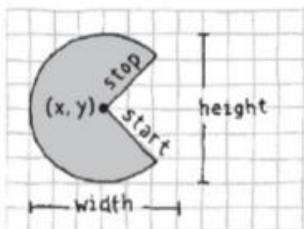


CC BY-NC-SA

4.1 Dessiner en 2D et gérer les couleurs



`ellipse(x, y, width, height)`



`arc(x, y, width, height, start, stop)`

Casey Reas and Ben Fry.
<Getting Started with Processing>.
O'Really Media, 2010

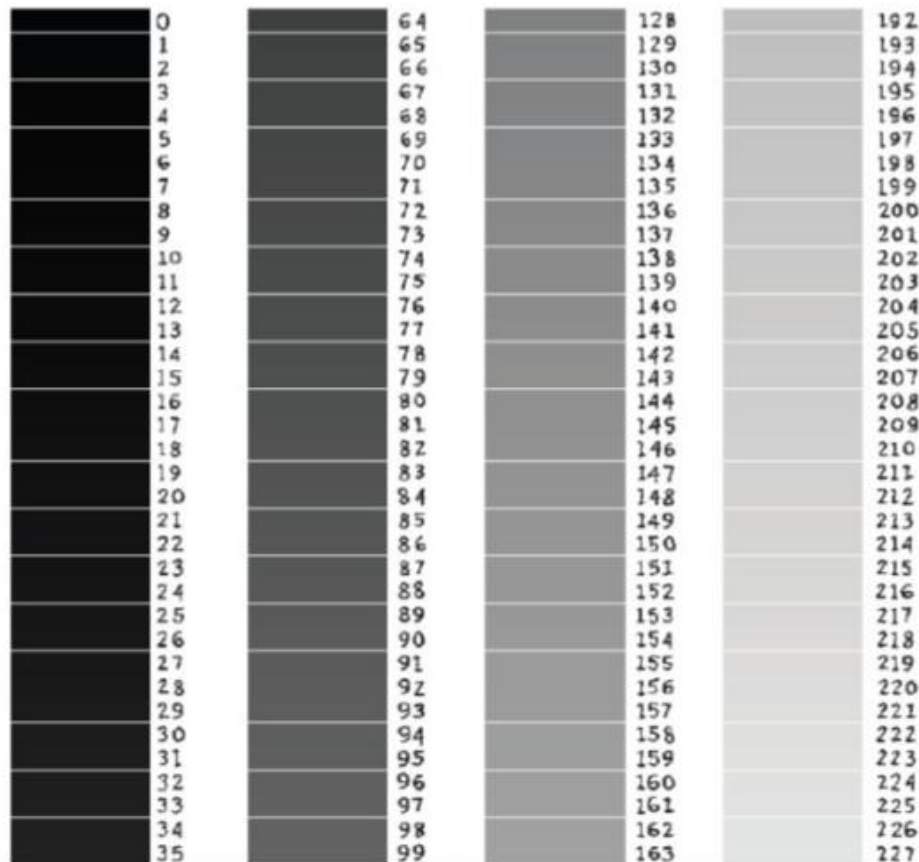
Processing pour Android



CC BY-NC-SA

4.1 Dessiner en 2D et gérer les couleurs

Grayscale



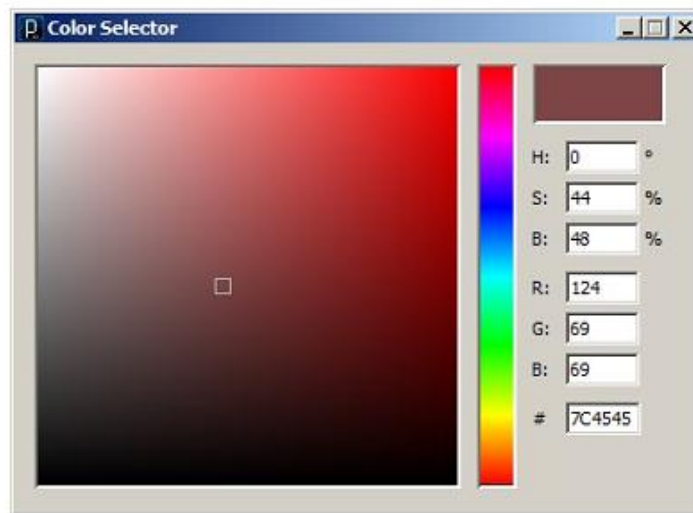
0(black) – 255(white)

Casey Reas and Ben Fry.
<Getting Started with Processing>.
O'Really Media, 2010



4.1 Dessiner en 2D et gérer les couleurs

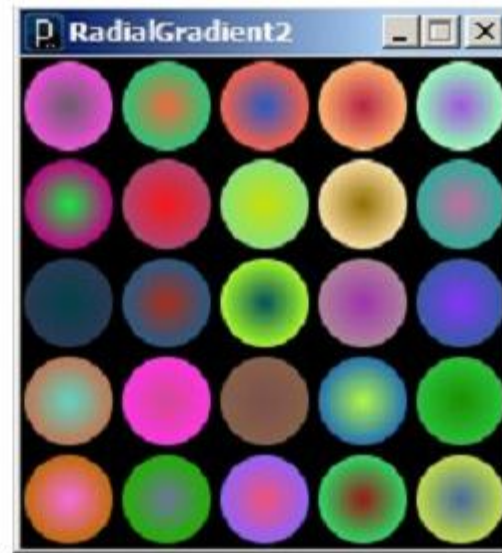
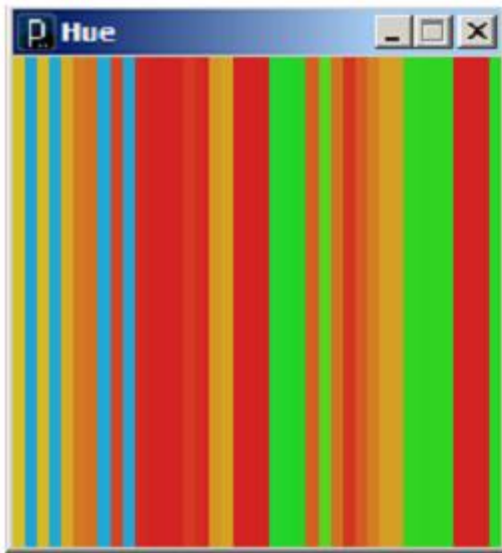
Color



```
color(gray)
color(gray, alpha)
color(value1, value2, value3)
color(value1, value2, value3, alpha) color(hex)
color(hex, alpha)
```



4.1 Dessiner en 2D et gérer les couleurs



fill() : couleur de remplissage
stroke() : contour

<Examples from processing>

```
fill(gray), fill(gray, alpha), fill(value1, value2, value3),  
fill(value1, value2, value3, alpha) fill(color),  
fill(color, alpha), fill(hex) fill(hex, alpha)
```



4.2 Animation et mouvement

setup()

Appelée une seule fois quand le programme démarre. Utiliser pour initialiser l'environnement de travail (taille de l'écran, couleur de l'arrière plan) avant d'exécuter le **draw()**.

Les variables déclarées dans le setup ne sont pas accessible aux autres fonctions.

draw()

Appelée directement après setup(), cette fonction exécute en continue les lignes de codes qu'elle contient tant que le programme n'est pas stopé ou que la fonction noLoop() est appelée.

```
Void setup() {  
  println("Setup: démarrage");  
}  
Void draw() {  
  println("En cours");  
}
```

Quand ce code est en cours d'exécution, les messages suivants sont écrits dans la console:

```
Setup: démarrage  
En cours  
En cours  
...
```

Processing pour Android



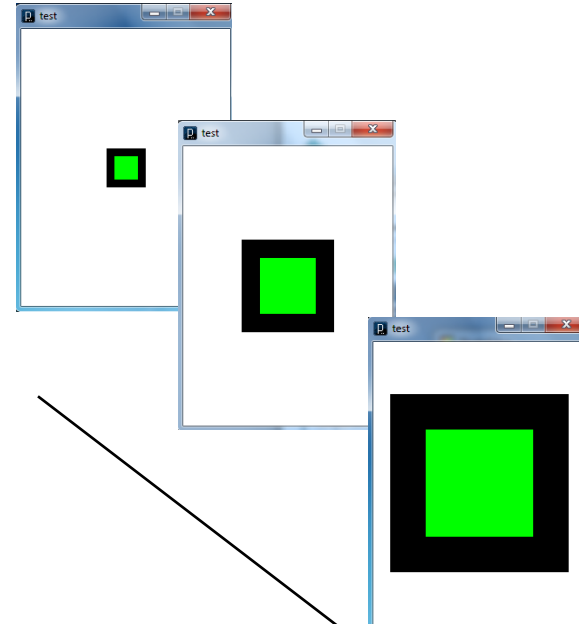
(CC) BY-NC-SA

4.2 Animation et mouvement

```
test | Processing 2.0a4
File Edit Sketch Tools Help
test$
int i=0; //déclaration d'une variable de type entier

void setup() { // fonction d'initialisation
  size(240,320); //taille de l'ecran
  stroke(0); //contour en noir
  rectMode(CENTER); //dessiner un rectangle par son milieu
  frameRate(150); //vitesse
}

void draw() { // fonction exécutée en continué
  i= i+1;
  if (i > 320) {
    i=0;
  }
  background(255); //arrière-plan en blanc
  strokeWeight(i/4); //épaisseur du contour
  fill(0,255,0); //couleur de remplissage du dessin
  rect(width/2, height/2, i, i); //dessiner un rectangle
}
```




Processing pour Android



CC BY-NC-SA

4.3 Média (images et polices de caractères)



mage(image, x, y)
image(image, x, y, c, d)

Image	PImage: l'image à afficher
x	float: coordonnée en x de l'image
y	float: coordonnée en y de l'image
c	float: largeur de l'image
d	float: hauteur de l'image

```
test | Processing 2.0a4
File Edit Sketch Tools Help
test$
PImage img1;

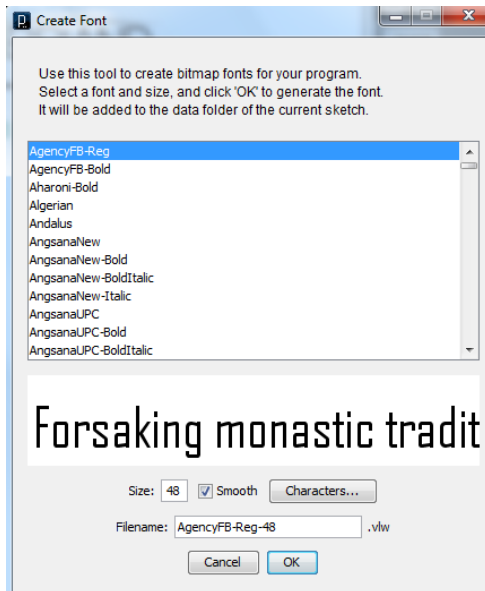
void setup() {
  size(500, 500);
  img1 = loadImage("PR5_1.jpg");
}

void draw() {
  background(255);
  image(img1, -150, 0);
  image(img1, 100, 100);
  image(img1, 270, 250, 100, 100);
}
```

Processing pour Android



4.3 Média (images et polices de caractères)



PFont
loadFont()
textFont()
text(data, x, y) text(data, x, y, z)
text(stringdata, x, y, width, height)
text(stringdata, x, y, width, height, z)

```
size(200,100);  
background(0);
```

```
PFont font;  
// The font must be located in the sketch's  
// "data" directory to load successfully  
font = loadFont("Arial-Black-48.vlw");  
textFont(font, 47);  
text("Hello!", 10, height/2);  
textSize(14);  
text("Hello!", 10, 70);
```



4.4 Interaction (clavier et écran tactile)

mousePressed :

La fonction mousePressed détecte l'appui ou non sur le clic de la souris (ou de l'écran tactile).

Voir aussi :

mousePressed()	http://processing.org/reference/mousePressed_.html
mouseReleased()	http://processing.org/reference/mouseReleased_.html
mouseMoved()	http://processing.org/reference/mouseMoved_.html
mouseDragged()	http://processing.org/reference/mouseDragged_.html
mouseX	http://processing.org/reference/mouseX.html
mouseY	http://processing.org/reference/mouseY.html
keyPressed	http://processing.org/reference/keyPressed.html
key	http://processing.org/reference/key.html

```
void draw() {  
  background(255);  
  if (mousePressed == true)  
  {  
    fill(255,0,0);  
  }  
  else  
  {  
    fill(255);  
  }  
  rect(50, 50, 50, 50);  
}
```

Processing pour Android



4.5 Pour aller plus loin

Gestion du **multitouche** de votre Android via Processing :

<http://code.google.com/p/android-multitouch-controller/>

Gestion des **capteurs** et de la **camera** de votre Android via Processing :

<http://code.google.com/p/ketai>

APWidgets: librairie pour l'utilisation des widgets d'Android avec processing :

<http://code.google.com/p/apwidgets>

OscP5: librairie pour envoyer et recevoir des messages OSC (Open Sound Control)

<http://www.sojamo.de/libraries/oscP5/index.html>



CONCLUSION:



NOT EVIL,
JUST HUNGRY