eBook Gratuit

APPRENEZ processing

eBook gratuit non affilié créé à partir des **contributeurs de Stack Overflow.**

#processing

Table des matières

À propos

You can share this PDF with anyone you feel could benefit from it, downloaded the latest version from: processing

It is an unofficial and free processing ebook created for educational purposes. All the content is extracted from Stack Overflow Documentation, which is written by many hardworking individuals at Stack Overflow. It is neither affiliated with Stack Overflow nor official processing.

Use the content presented in this book at your own risk; it is not guaranteed to be correct nor accurate, please send your feedback and corrections to info@zzzprojects.com

Chapitre 1: Démarrer avec le traitement

Remarques

Processing est un environnement et un langage de programmation open source pour les personnes qui souhaitent créer des images, des animations et des interactions.

Le traitement fait référence au langage intégré à Java et à l'EDI minimal fourni avec. Il est gratuit et open-source, fonctionne sous Linux, Mac OS X et Windows et peut produire des écrans, des impressions, des packages 3D et des impressions CNC.

Le langage simplifie beaucoup de concepts complexes et facilite l'entrée des concepteurs, des artistes et des non-programmeurs dans le monde de la programmation.

Au fil des ans , il a été utilisé pour produire un certain nombre de projets allant de la visualisation des données , à l' informatique physique , jeux , 3D , son , perfomance en direct , et plus .

Grâce à sa communauté dynamique, Processing bénéficie non seulement d'une contribution de plus de 100 bibliothèques, mais est également présente sur les principales plates-formes mobiles telles que Android et iOS .

Il existe des communautés en ligne pour partager du contenu de traitement, comme OpenProcessing .

Certains sites Web permettent même aux utilisateurs d'apprendre et d'utiliser Processing directement dans le navigateur, comme SketchPatch piloté par Flash et HasCanvas , Sketchpad et p5.js (pur JS) pilotés par JavaScript.

Il existe également des ports de traitement dans les langues suivantes:

- JavaScript en utilisant ProcessingJS ou p5js
- ActionScript
- Python (voir NodeBox , Field , pyProcessing ou le nouveau mode Python officiel)
- Scala
- Clojure
- Rubis

Le mode Android permet d'exécuter des esquisses de traitement en tant qu'applications Android avec peu ou pas de modifications du code en automatisant les tâches depuis la configuration du projet vers l'exportation de fichiers *.apk* . Les esquisses de traitement Android ont également accès aux capteurs et périphériques Android sous-jacents.

Les utilisateurs avancés ne sont pas contraints à l'EDI de traitement; ils peuvent configurer des projets de traitement dans Eclipse ; utiliser proclipsing ou utiliser Sublime Text pour créer et exécuter des esquisses via le package processing-sublime .

Versions

Version	Date de sortie
1.5.1	2011-05-15
2.2.1	2014-05-19
3.1.2	2016-07-29
3.2.1	2016-08-19

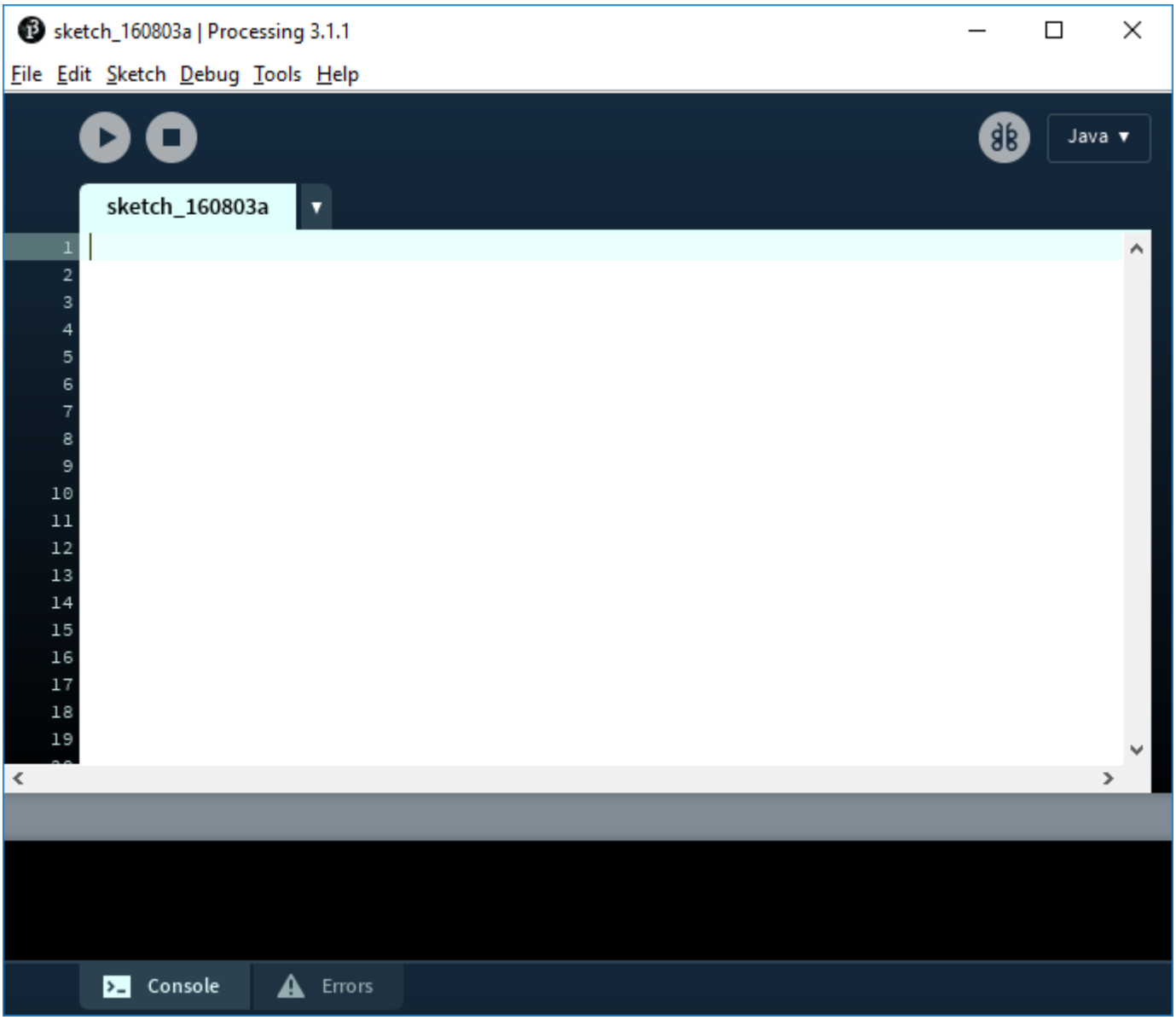
Examples

Installation et configuration

La manière la plus simple d'utiliser le traitement est de télécharger l'éditeur de traitement à partir de la page de téléchargement du traitement .

Cela vient comme un fichier zip. Décompressez ce fichier n'importe où et vous aurez un répertoire contenant un exécutable (sous Windows, `processing.exe`).

L'exécution de cet exécutable ouvre l'éditeur de traitement:



L'éditeur de traitement (également appelé environnement de développement de traitement ou PDE) contient de nombreux outils qui vous apportent beaucoup de travail. Il vous permet d'écrire du code de traitement, qu'il convertit automatiquement en Java, puis compile et exécute pour vous.

Le PDE contient de nombreuses fonctionnalités, mais pour l'instant, écrivez simplement votre code de traitement dans la section blanche de l'éditeur, puis appuyez sur le bouton de lecture pour exécuter votre code. Voir la section Hello World ci-dessous pour un exemple de code.

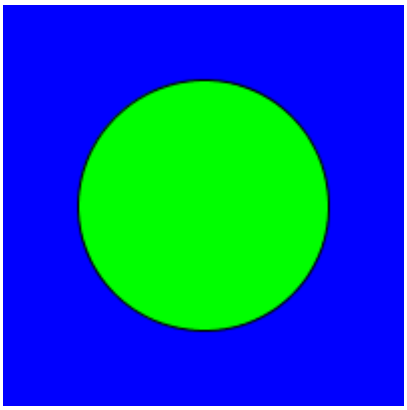
Vous pouvez également écrire du code de traitement en utilisant d'autres éditeurs de code de base tels que Atom ou Sublime Text , ou avec un IDE plus avancé comme eclipse .

Bonjour le monde

Le moyen le plus simple d'écrire du code de traitement consiste à appeler simplement une série de fonctions. Appuyez sur le bouton Exécuter dans l'éditeur de traitement et Traitement exécutera votre code. Voici un exemple:

```
size(200, 200);
background(0, 0, 255);
fill(0, 255, 0);
ellipse(100, 100, 100, 100);
```

Ce code crée une fenêtre `200x200` , dessine un arrière-plan bleu, change la couleur de remplissage en vert, puis dessine un cercle au milieu de l'écran.



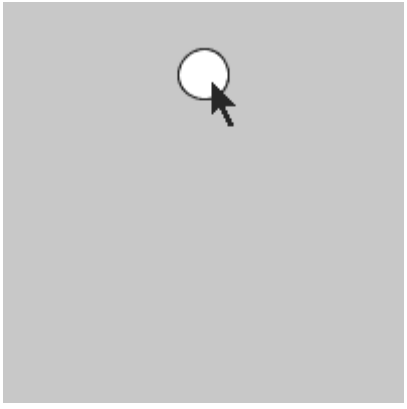
Cependant, la plupart des esquisses de traitement utiliseront les fonctions prédéfinies `setup()` et `draw()` .

- La fonction `setup()` est appelée automatiquement par Processing, une fois au tout début de l'esquisse. Cette fonction est utilisée pour effectuer la configuration initiale, telle que la `size` et le chargement de ressources telles que des fichiers image et audio.
- La fonction `draw()` est appelée automatiquement par Traitement 60 fois par seconde. Cette fonction est utilisée pour dessiner et obtenir une entrée utilisateur.

```
void setup() {
  size(200, 200);
}

void draw(){
  background(0);
  ellipse(mouseX, mouseY, 25, 25);
}
```

Ce code crée une fenêtre `200x200` puis dessine un cercle à la position actuelle de la souris.



Lire Démarrer avec le traitement en ligne: https://riptutorial.com/fr/processing/topic/3538/demarrer-avec-le-traitement

Chapitre 2: Couleurs en traitement

Introduction

Cet article va vous montrer les différents formats de couleurs dans le traitement et les manières dont ils peuvent être utilisés.

Syntaxe

- couleur (r, g, b);
- couleur (r, g, b, alpha);
- couleur (gris);
- couleur (gris, alpha);
- couleur (h, s, l); // Le mode doit être HSB. Vous pouvez changer cela en utilisant colorMode.

Paramètres

Paramètres	Détails
r	Est le rouge de la couleur lorsque le mode est `RGB` .
g	Est le vert de la couleur lorsque le mode est `RGB` .
b	Est le bleu de la couleur lorsque le mode est `RGB` .
alpha	Est l'opacité de la couleur.
h	La teinte de la couleur lorsque le mode est `HSB` .
s	La saturation de la couleur lorsque le mode est `HSB` .
l	La luminosité / luminosité de la couleur lorsque le mode est `HSB` .
gris	La valeur entre le noir (0) et le blanc (255).

Remarques

Bien que cela n'ait pas été mentionné dans la documentation officielle sur le traitement, il existe un mode `CMYK` que vous pouvez utiliser.

Examples

Notation de couleur

Il existe différentes manières d'utiliser les couleurs dans le traitement, car le traitement est très flexible avec les formats de couleur.

RGB et RGBA

C'est la notation RVB (A) standard et le mode couleur par défaut. Les trois premières valeurs de couleur (rouge, vert, bleu) vont de `0` à `255` . Par exemple, l'exemple ci-dessous est la couleur rouge puisque le rouge est maximum à `255` tandis que les autres couleurs sont à `0` . Le blanc est à `(255, 255, 255)` et le noir est à `(0, 0, 0)` . Le 4ème paramètre facultatif indique la valeur alpha - c'est-à-dire la transparence. Comme dans les autres composants, la plage de valeurs est à nouveau [0-255]; `0` étant complètement transparent et `255` étant complètement solide.

```
color(255, 0, 0) // This is red

color(0, 255, 0, 255) // This is opaque green, and is the same as color(0, 255, 0)

color(255, 255, 0, 10) // This is almost transparent yellow
```

HSB

La notation HSB est similaire à la notation RVB, sauf que le rouge, le vert et le bleu sont remplacés respectivement par la teinte, la saturation et la luminosité. Vous pouvez basculer dans HSB en utilisant `colorMode(HSB)` .

```
color(0, 0, 255) //This is white
```

Comme pour RGB, HSB a également la valeur alpha comme quatrième paramètre.

Valeurs grises

Si un paramètre est spécifié pour une fonction de couleur, il sera interprété comme la quantité entre noir et blanc. Le blanc est représenté par 255 et le noir par 0. Il est identique à la `color(param1, param1, param1)` en mode RVB. Si deux paramètres sont spécifiés, le premier paramètre sera interprété comme ci-dessus et le second sera la valeur alpha.

Lire Couleurs en traitement en ligne: https://riptutorial.com/fr/processing/topic/8283/couleurs-en-traitement

Chapitre 3: Démarrer avec Processing pour Android

Remarques

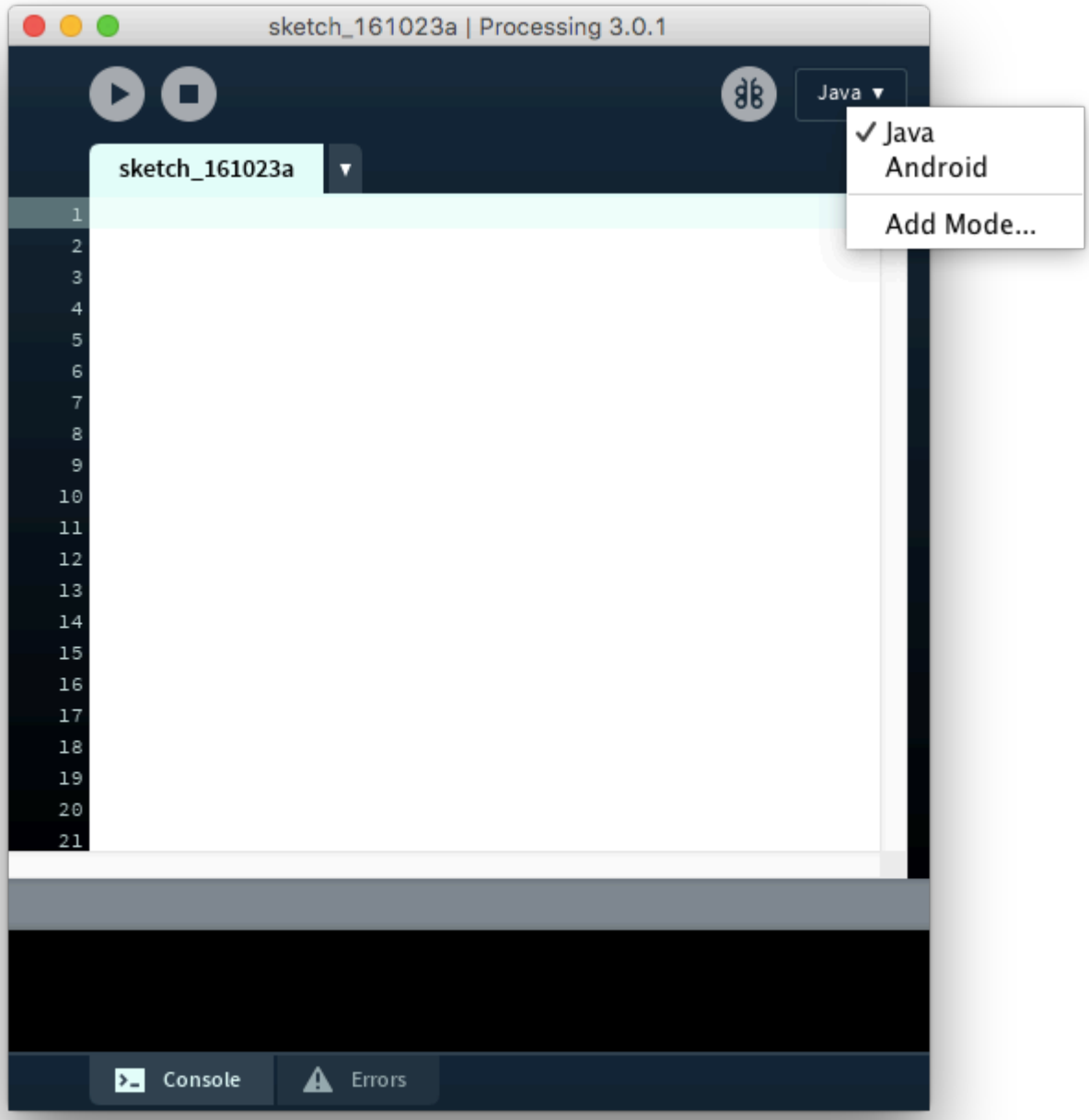
Vous pouvez trouver plus d'informations sur Processing for Android sur http://android.processing.org/

Examples

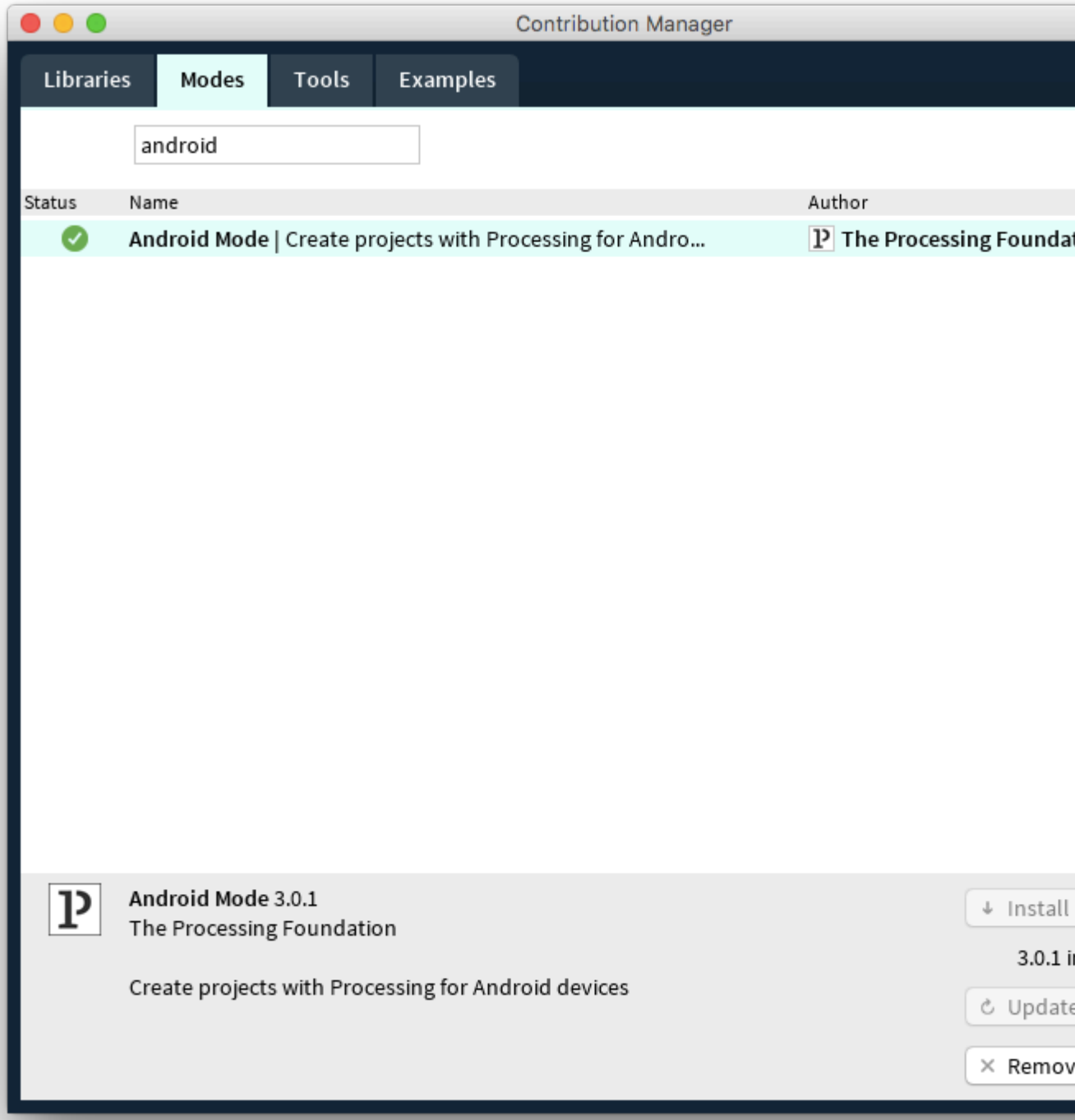
Installation

Avant de commencer le traitement pour Android, assurez-vous que le traitement est installé.

1. Traitement ouvert
2. Cliquez sur Ajouter Mode ... situé en haut à droite de la fenêtre:



3. Recherchez "android" dans le Gestionnaire de contributions et choisissez l'option qui a l'auteur "The Processing Foundation" et cliquez sur Installer



4. Une fois l'installation terminée, basculez le mode sur Android
5. Cette fenêtre ci-dessous devrait apparaître. Si vous ne possédez pas le SDK Android, cliquez sur "Télécharger le SDK automatiquement" pour l'installer. Si vous avez déjà installé le SDK Android, cliquez sur "Localiser le chemin SDK manuellement".

5,5. Pour moi, j'ai localisé mon chemin SDK dans `/Users/my-username/Library/Android/sdk`

6. Vous avez terminé!

Lire Démarrer avec Processing pour Android en ligne:
https://riptutorial.com/fr/processing/topic/7586/demarrer-avec-processing-pour-android

Chapitre 4: Dessin de formes de base

Introduction

En traitement, dessiner des formes est la clé du programme. Sinon, rien n'apparaît à l'écran. Cette section vous indiquera comment les formes de base sont dessinées.

Syntaxe

- line (float x1, float y1, float x2, float y2)
- line (float x1, float y1, float z1, float x2, float y2, float z2)
- ellipse (float x, float y, float w, float h)
- rect (float x, float y, float w, float h)
- triangle (float x1, float y1, float x2, float y2, float x3, float y3)

Paramètres

Paramètre	Détails
x1	coordonnée x du premier point
y1	coordonnée y du premier point
z1	coordonnée z du premier point
x2	Coordonnée x du deuxième point
y2	coordonnée y du deuxième point
z2	coordonnée z du deuxième point
x3	Coordonnée x du troisième point
y3	coordonnée y du troisième point
X	coordonnée x
y	coordonnée y
w	largeur
h	la taille

Remarques

Vous pouvez trouver une référence sur la base de traitement.

Traitement de la page d'accueil

Examples

Dessiner une ligne

Processing fournit une méthode nommée `line()` pour tracer une ligne à l'écran. Ce code dessine une ligne blanche de 10 pixels sur fond noir.

```
void setup() {
    size(500, 500);
    background(0);
    stroke(255);
    strokeWeight(10);
}

void draw() {
    line(0, 0, 500, 500);
}
```

La signature de la méthode `line()` est la suivante.

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

`x1` et `y1` sont les coordonnées du point de départ. `x2` et `y2` sont les coordonnées du point final.

Method `stroke()` est utilisé pour spécifier la couleur de la ligne que vous allez dessiner.

La méthode `strokeWeight()` est utilisée pour spécifier l'épaisseur de la ligne que vous allez dessiner. (en pixels)

Dessiner un rectangle

Le traitement fournit la méthode `rect()` pour dessiner un rectangle. Ce code dessine un rectangle blanc de 50 X 50 sur fond noir.

```
void setup() {
    size(500, 500);
    background(0);
    fill(255);
    noStroke();
}

void draw() {
    rect(225, 225, 50, 50);
}
```

La signature de la méthode `rect()` est la suivante.

```
rect(x, y, w, h);
```

`x` et `y` sont les coordonnées du rectangle. `w` et `h` sont la largeur et la hauteur du rectangle.

La méthode `fill()` permet de spécifier la couleur de remplissage du rectangle et d'autres formes telles que l'ellipse, le triangle ou le polygone.

La méthode `noStroke()` est utilisée pour spécifier qu'il n'y a pas de trait autour du rectangle. Cette méthode affecte également d'autres formes telles que l'ellipse, le triangle, le polygone.

Dessiner une Ellipse

Le traitement fournit une `ellipse` méthode afin de dessiner une ellipse. Ce code dessine un cercle blanc qui a un rayon de 25 pixels.

```
void setup() {
    size(500, 500);
    background(0);
    fill(255);
    noStroke();
}

void draw() {
    ellipse(225, 225, 50, 50);
}
```

La signature de la méthode `ellipse()` est la suivante.

```
ellipse(x, y, w, h);
```

`x` et `y` sont les coordonnées de l'ellipse. `w` et `h` sont la largeur et la hauteur de l'ellipse.

Dessiner un triangle

Le traitement fournit le `triangle` méthode afin de tracer un triangle. Le code ci-dessous trace un triangle presque équilatéral de 25 pixels entre chaque point de définition.

```
void setup() {
    size(500, 500);
    background(0);
}
void draw() {
    triangle(0, 0, 25, 0, 12, 12);
}
```

La signature du `triangle` est la suivante:

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

Chaque point `x` correspond à l'axe x du point et `y` à l'axe y. Les trois points seront joints pour former un triangle.

Dessiner un triangle

Le traitement fournit la méthode `triangle()` pour dessiner un triangle. Ce code dessine un triangle blanc sur fond noir.

```
void setup() {
    size(500, 500);
    background(0);
    fill(255);
    noStroke();
}

void draw() {
    triangle(250, 225, 225, 275, 275, 275);
}
```

Lire Dessin de formes de base en ligne: https://riptutorial.com/fr/processing/topic/7277/dessin-de-formes-de-base

Chapitre 5: Formes et fonctions de base utilisant P3D

Syntaxe

- translate (float x, float y, float z)
- rotateX (angle de flottaison)
- rotateY (angle de flottaison)
- tournerZ (angle flottant)
- boîte (taille du flotteur)
- box (float w, float h, float d)

Paramètres

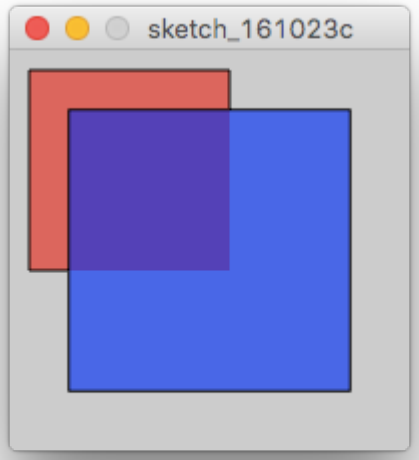
Paramètres	Détails
angle	l'angle est en radians
Taille	la dimension de la boîte à utiliser pour toutes ses dimensions
w	la dimension de la boîte en `x-axis`
h	la dimension de la boîte dans l' `y-axis`
ré	la dimension de la boîte dans l' `z-axis` des `z-axis`

Examples

Traduction 3D

Voici comment traduire des objets dans P3D:

```
size(200, 200, P3D); //Starting P3D renderer
fill(255, 0, 0, 150); //transparent red
rect(10, 10, 100, 100); //first rectangle
fill(0, 0, 255, 150); //transparent blue
translate(50, 50, 50); //translate x, y and z by 50 pixels
rect(0, 0, 100, 100); //second rectangle (same dimensions as the first one)
```



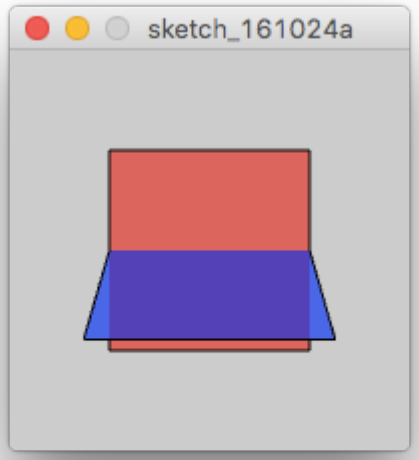
Rouge: premier rectangle Bleu: deuxième rectangle

Comme on peut le voir sur l'esquisse ci-dessus, le second rectangle ne semble être plus grand que le premier, alors qu'en réalité il est «plus proche» de l'écran en traduisant le rectangle de `50` pixels le long de l' `z-axis` (et de Bien entendu, le rectangle a été traduit selon les axes `x` et `y`).

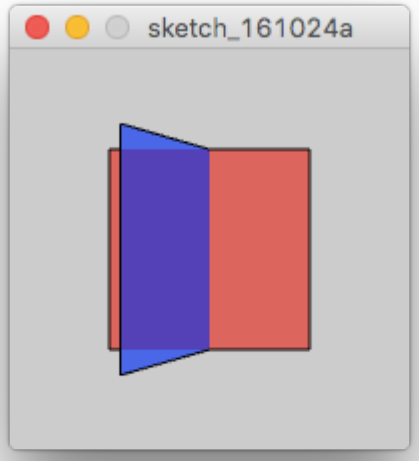
Rotation 3D

Il y a trois fonctions pour la rotation 3D: `rotateX(angle)` , `rotateY(angle)` et `rotateZ(angle)` pour la rotation dans leurs axes respectifs où l' `angle` est en radians.

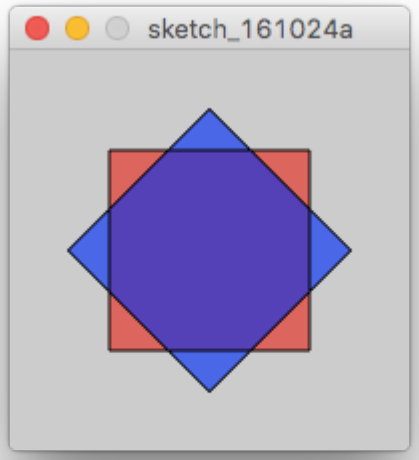
```
size(200, 200, P3D); //Starting P3D renderer
fill(255, 0, 0, 150); //transparent red
translate(width/2, height/2);//translate to centre, ie (100, 100)
rectMode(CENTER);//This makes the rectangle centre in (100, 100)
rect(0, 0, 100, 100); //first rectangle
fill(0, 0, 255, 150); //transparent blue
rotateX(PI/4); //rotate in the x-axis by PI/4 radians (45 degrees)
rect(0, 0, 100, 100); //second rectangle (same dimensions as the first one)
```




```
rotateY(radians(45)); //rotate in the y-axis by passing the radians conversion of 45 degrees
```




```
rotateZ(3*PI/4); //rotate in the z-axis by 3*PI/4 radians (270 degrees)
```

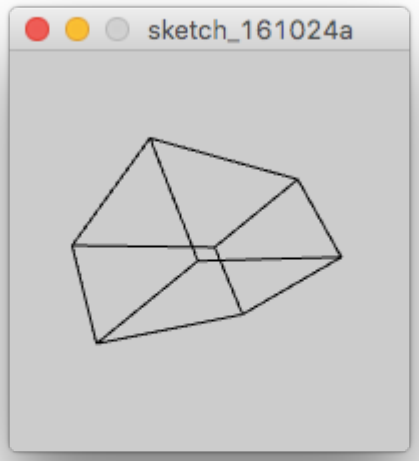


Remarque: les transformations (telles que les traductions et les rotations) s'ajoutent à la transformation précédente.

Dessiner un cuboïde

Pour dessiner un cuboïde, vous devez utiliser la fonction `box()` en donnant ses dimensions comme paramètres.

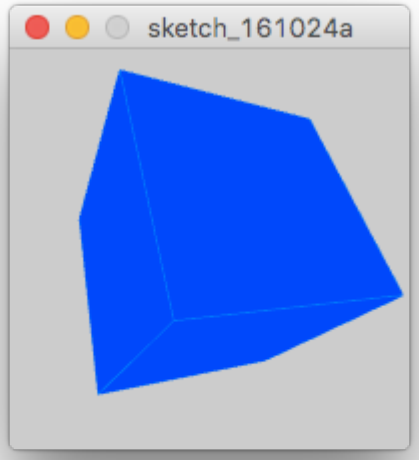
```
size(200, 200, P3D); //Starting the P3D renderer
translate(width/2, height/2); //Translating to the centre of the sketch
rotateY(PI/4); //rotate so that...
rotateX(PI/6); //... it will be easy to see the box
noFill(); //disabling the box's fill, so that we will be able to see its edges
box(100, 50, 75); //the box function requires its dimensions as its parameters
```



Notez que la fonction `box()` n'accepte *pas* sa position comme paramètre

Il y a aussi un moyen d'appeler la fonction `box()` avec un seul paramètre. Dans ce cas, ce sera un cube.

```
stroke(0, 100, 255); //change the edges' colour
fill(0, 0, 255); //fill the `box` in a blue colour
box(100); //draw a cube
```



Lire Formes et fonctions de base utilisant P3D en ligne:
https://riptutorial.com/fr/processing/topic/7591/formes-et-fonctions-de-base-utilisant-p3d

Chapitre 6: Utilisation du moteur de rendu P3D de Processing

Remarques

Une référence pour l'utilisation de P3D se trouve sur https://processing.org/tutorials/p3d/ dans un tutoriel de Daniel Shiffman.

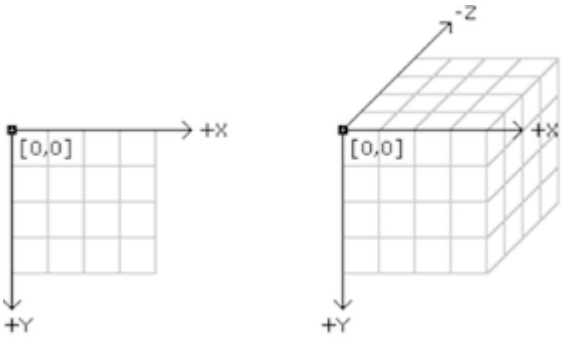
Examples

Commencer

Pour pouvoir utiliser le moteur de rendu P3D, vous devez le spécifier dans la fonction `size()`

```
size(200, 200, P3D);
```

Maintenant, il y a trois axes: `x` , `y` et `z` .



Maintenant, vous pouvez procéder au dessin en 3D.

Lire Utilisation du moteur de rendu P3D de Processing en ligne: https://riptutorial.com/fr/processing/topic/7589/utilisation-du-moteur-de-rendu-p3d-de-processing

Chapitre 7: Utiliser le traitement avec des éditeurs de code alternatifs comme Sublime et Atom

Examples

Utilisation du traitement avec du texte sublime

Pour ajouter une langue à Sublime Text, vous utilisez le contrôle de package . Vous pouvez le faire en collant le code Python correct (disponible sur le site de contrôle de package lié ci-dessus) ou en téléchargeant le fichier .sublime-package (également disponible en téléchargement sur le site). Une fois que vous avez configuré le contrôle des packages, redémarrez Sublime Text.

Pour installer le package de traitement à partir du contrôle de package, procédez comme suit:

- Outils d'exécution | Installez processing-java pour installer l'outil processing-java. Cet outil est nécessaire pour créer des esquisses de traitement en ligne de commande et des éditeurs tels que Sublime Text et Atom.
- Ouvrez la palette de commandes (Ctrl + Maj + P ou Cmd + Maj + P)
- Rechercher 'Contrôle de package: Installer le package'
- Recherchez 'Processing' et installez le package
- Redémarrer le texte sublime

Après avoir suivi ces étapes, vous devriez pouvoir sélectionner Traitement en tant que langue. Cela facilitera le codage dans Processing dans Sublime Text.

Utilisation du traitement avec l'éditeur Atom

Plusieurs packages peuvent exécuter des schémas de traitement dans l'éditeur Atom. Ces instructions utilisent le package Script . Il existe également des packages disponibles pour la mise en évidence de la syntaxe et la saisie semi - automatique , nécessaires pour que Script puisse identifier les types de fichiers de traitement.

Installez le plugin Script. Soit en exécutant `apm install script` partir de la ligne de commande, soit en recherchant le package 'script' par *rgbkrk* dans l'onglet `Install` dans les paramètres Atom (`command + ,` raccourci `command + ,` ou `ctrl + ,`).

Une fois le script installé, vous devrez installer `processing-java` . Cet outil est fourni avec le logiciel de traitement principal et est nécessaire pour créer des esquisses de traitement sur la ligne de commande et dans les éditeurs:

- *MacOS:* Exécuter `Tools > Install "processing-java"` .
- *Linux:* Ajoutez le répertoire de traitement à votre variable d'environnement `PATH` (remplacez `/path/to/processing`

avec le chemin d'accès à Processing):

```
sudo ln -s /path/to/processing/processing-java /usr/local/bin/
```

- *Windows:* Ajoutez le répertoire de traitement à votre variable d'environnement `PATH` :
    - Ouvrez les paramètres système avancés en exécutant `sysdm.cpl` ou en `sysdm.cpl` une recherche dans le Panneau de configuration.
    - Cliquez sur le bouton Variable d'environnement dans l'onglet Avancé.
    - Modifiez la variable `PATH` pour inclure le répertoire de traitement dans les variables utilisateur (pour votre compte uniquement) ou les variables système (pour tous les utilisateurs).

Vous pouvez maintenant exécuter les esquisses de traitement en exécutant `Packages > Script > Run Script` . Le raccourci par défaut est `command + shift + b` ou `ctrl + shift + b` , mais pour réduire davantage la douleur de la transition, vous pouvez lier le raccourci Processing IDE pour exécuter l'esquisse. Pour pouvoir faire ça:

- Ouvrez le fichier Keymap Atom en exécutant `File > Keymap`
- Collez les lignes suivantes à la fin du fichier (n'hésitez pas à changer la liaison à ce que vous voulez).

```
'atom-text-editor':
  'ctrl-r': 'script:run'
```

Utilisation du traitement avec Eclipse

Pour utiliser Processing dans Eclipse, commencez par créer un nouveau projet Java. Ensuite, sélectionnez `File > Import` , puis choisissez `General > File System` pour localiser le fichier `core.jar` . On peut le trouver dans `PATH_TO_PROCESSING/core/library/` pour Windows ou `/Applications/Processing 3.app/Contents/Java/core/library/` pour Mac. Une fois cette opération terminée, cliquez avec le bouton droit sur `core.jar` et ajoutez-le au chemin de génération.

La procédure à suivre pour utiliser Processing dans Eclipse est la suivante:

```
import processing.core.PApplet;

public class UsingProcessing extends PApplet {

    // The argument passed to main must match the class name
    public static void main(String[] args) {
        PApplet.main("UsingProcessing");
    }

    // method used only for setting the size of the window
    public void settings(){

    }

    // identical use to setup in Processing IDE except for size()
```

```
    public void setup(){

    }

    // identical use to draw in Prcessing IDE
    public void draw(){

    }
}
```

La méthode `settings()` permet de définir la taille de la fenêtre. Par exemple, pour créer une fenêtre 400x400, écrivez ce qui suit:

```
 public void settings(){
    size(400,400);
 }
```

Tout ce qui est décrit dans la documentation de Hello World en termes d'utilisation de `setup()` et `draw()` s'applique ici.

Comme dernier exemple, voici le code de l' exemple Drawing a Line s'il était écrit en Eclipse:

```
import processing.core.PApplet;

public class UsingProcessing extends PApplet {

    // The argument passed to main must match the class name
    public static void main(String[] args) {
        PApplet.main("UsingProcessing");
    }

    // method for setting the size of the window
    public void settings(){
        size(500, 500);
    }

    // identical use to setup in Processing IDE except for size()
    public void setup(){
        background(0);
        stroke(255);
        strokeWeight(10);
    }

    // identical use to draw in Prcessing IDE
    public void draw(){
        line(0, 0, 500, 500);
    }
}
```

Lire Utiliser le traitement avec des éditeurs de code alternatifs comme Sublime et Atom en ligne: https://riptutorial.com/fr/processing/topic/5594/utiliser-le-traitement-avec-des-editeurs-de-code-alternatifs-comme-sublime-et-atom

Crédits

S. No	Chapitres	Contributeurs
1	Démarrer avec le traitement	aspiring_sarge, Community, Cows quack, Kevin Workman, MasterBob, Michael Andersen, user2314737
2	Couleurs en traitement	Cows quack, MasterBob, Mert Toka
3	Démarrer avec Processing pour Android	Cows quack
4	Dessin de formes de base	Cows quack, MasterBob, sohnryang
5	Formes et fonctions de base utilisant P3D	Cows quack
6	Utilisation du moteur de rendu P3D de Processing	Cows quack
7	Utiliser le traitement avec des éditeurs de code alternatifs comme Sublime et Atom	bakahoe, CodeMacabre, Kevin Workman, Mert Toka, Peter, sohnryang