



Бесплатная электронная книга

УЧУСЬ processing

Free unaffiliated eBook created from
Stack Overflow contributors.

#processing

.....	1
1:	2
.....	2
.....	3
Examples.....	3
.....	3
,	4
2: , Sublime	7
Examples.....	7
Sublime Text.....	7
Atom.....	7
Eclipse.....	8
3: P3D	11
.....	11
Examples.....	11
.....	11
4: Android	12
.....	12
Examples.....	12
.....	12
5: P3D	16
.....	16
.....	16
Examples.....	16
3D-.....	16
3D-.....	17
.....	19
6:	21
.....	21
.....	21
.....	21

.....	22
Examples.....	22
.....	22
.....	22
.....	23
.....	23
.....	24
7:	25
.....	25
.....	25
.....	25
.....	25
Examples.....	26
.....	26
.....	27

Около

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.

The content is released under Creative Commons BY-SA, and the list of contributors to each chapter are provided in the credits section at the end of this book. Images may be copyright of their respective owners unless otherwise specified. All trademarks and registered trademarks are the property of their respective company owners.

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

глава 1: Начало работы с обработкой

замечания

Обработка - это язык программирования с открытым исходным кодом и среда для людей, которые хотят создавать образы, анимации и взаимодействия.

[Обработка](#) относится к языку, построенному поверх Java и минимальной IDE, с которой он поставляется. Он бесплатный и с открытым исходным кодом работает под управлением Linux, Mac OS X и Windows и может выводить на экран, печать, 3D-пакеты и печать с ЧПУ.

Язык упрощает множество сложных понятий и облегчает вход дизайнеров, художников и не-программистов в мир программирования.

На протяжении многих лет он использовался для создания ряда проектов, начиная от [визуализации данных](#) , заканчивая [физическими вычислениями](#) , [играми](#) , [3D](#) , [звуком](#) , [живым исполнением](#) и т . Д.

Благодаря своему активному сообществу, Processing не только имеет вклад более 100 библиотек, но также присутствует на основных мобильных платформах, таких как [Android](#) и [iOS](#) .

Для совместного использования контента, например [OpenProcessing](#), существуют онлайн-сообщества.

Некоторые веб-сайты даже позволяют пользователям изучать и использовать Обработка непосредственно в браузере, например, [SketchPatch](#) с флеш- [памятью](#) и JavaScript- [движок](#) [HasCanvas](#) , [Sketchpad](#) и [p5.js](#) (чистый JS).

Существуют также порты обработки на следующих языках:

- JavaScript с использованием [ProcessingJS](#) или [p5js](#)
- [ActionScript](#)
- Python (см. [NodeBox](#) , [Field](#) , [pyProcessing](#) или новый официальный [режим Python](#))
- [Scala](#)
- [Clojure](#)
- [Рубин](#)

Режим [Android](#) позволяет запускать эскизы обработки в виде приложений Android с небольшими или никакими изменениями в коде, автоматизируя задачи от настройки проекта до экспорта файла `.apk` . Эскизы обработки Android также имеют доступ к базовым сенсорным устройствам и устройствам Android.

Продвинутые пользователи не ограничены обработкой IDE; они могут создавать [проекты](#)

обработки в Eclipse ; используйте [прокрутку](#) или альтернативно используйте [Sublime Text](#) для создания и запуска эскиза через пакет [обработки-возвышенности](#) .

Версии

Версия	Дата выхода
1.5.1	2011-05-15
2.2.1	2014-05-19
3.1.2	2016-07-29
3.2.1	2016-08-19

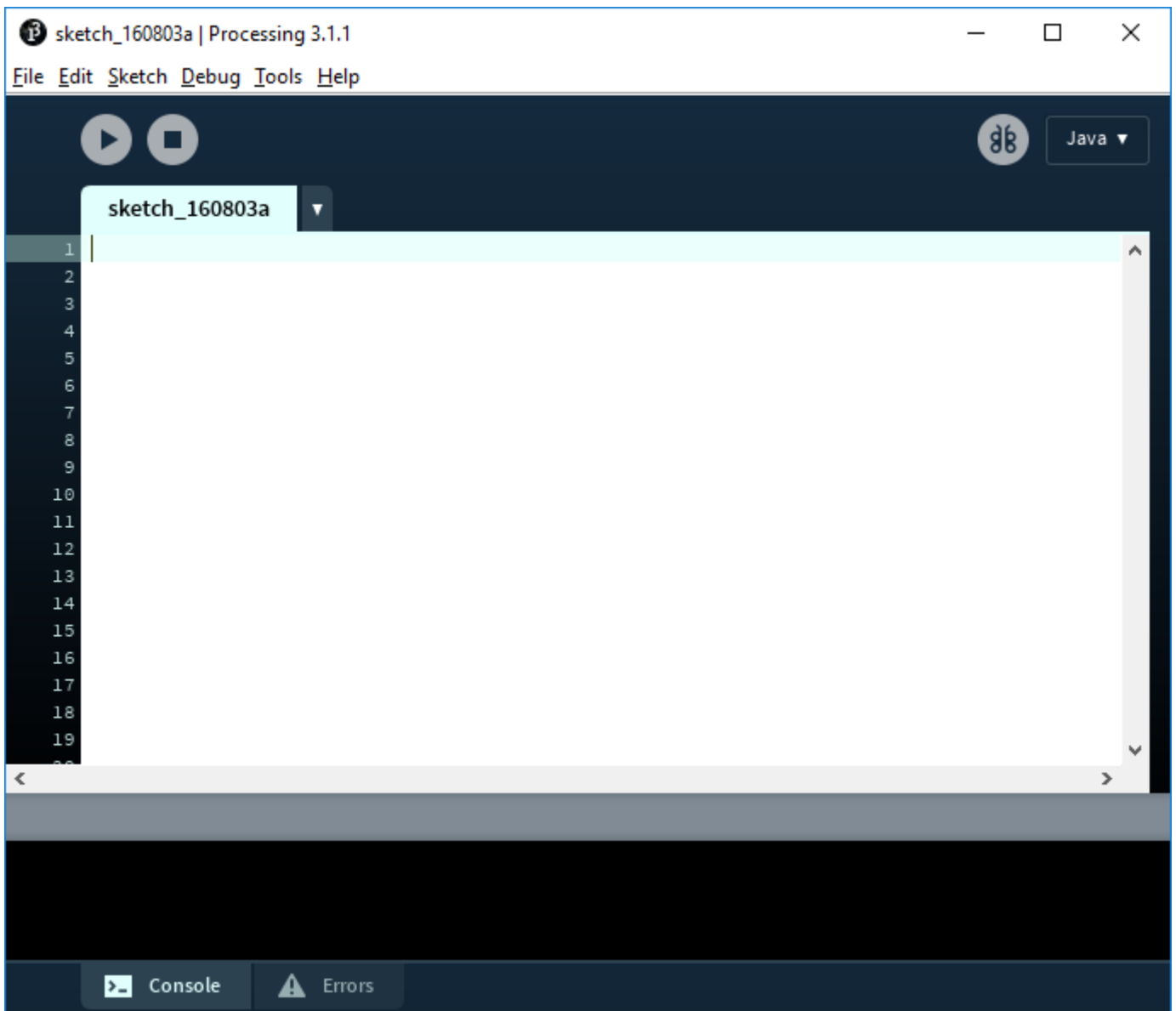
Examples

Установка и настройка

Самый простой способ использования обработки - загрузить редактор [обработки со страницы загрузки](#) .

Это происходит как zip-файл. Разархивируйте этот файл в любом месте, и у вас будет каталог, содержащий исполняемый файл (в Windows, это `processing.exe`).

Запуск этого исполняемого файла открывает редактор обработки:



Редактор обработки (также называемый средой разработки разработки или PDE) содержит множество инструментов, которые много работают для вас. Он позволяет вам писать код обработки, который он автоматически преобразует в Java, а затем компилирует и запускает для вас.

PDE содержит много функций, но пока просто напишите свой код обработки в белом разделе редактора, а затем нажмите кнопку воспроизведения, чтобы запустить код. См. Раздел Hello World ниже для некоторого примера кода.

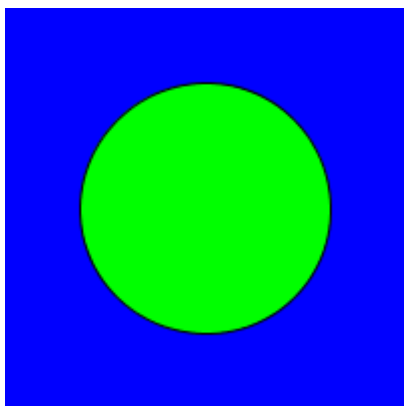
Вы также можете написать код обработки, используя другие базовые редакторы кода, такие как [Atom](#) или [Sublime Text](#) , или с более продвинутой IDE, такой как [eclipse](#) .

Привет, мир

Самый простой способ написать код обработки - просто вызвать ряд функций. Нажмите кнопку запуска в редакторе обработки, и обработка будет запускать ваш код. Вот пример:

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

Этот код создает окно 200×200 , рисует синий фон, меняет цвет заливки на зеленый, а затем рисует круг посередине экрана.



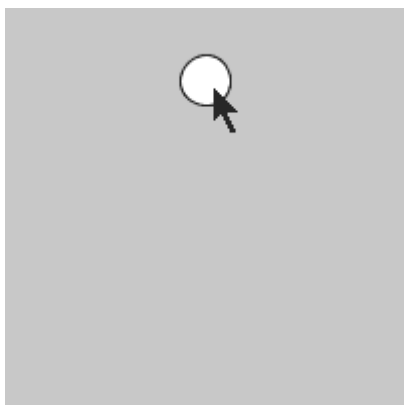
Однако большинство обработок эскизов будут использовать predetermined функции `setup()` и `draw()`.

- Функция `setup()` вызывается автоматически обработкой один раз в самом начале эскиза. Эта функция используется для начальной настройки, такой как `size` и загрузка таких ресурсов, как изображения и звуковые файлы.
- Функция `draw()` вызывается автоматически обработкой 60 раз в секунду. Эта функция используется для рисования и ввода пользовательского ввода.

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

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

Этот код создает окно 200×200 и затем рисует круг в текущей позиции мыши.



Прочитайте Начало работы с обработкой онлайн:

<https://riptutorial.com/ru/processing/topic/3538/начало-работы-с-обработкой>

глава 2: Использование обработки с альтернативными редакторами кода, такими как Sublime и Atom

Examples

Использование обработки с помощью Sublime Text

Чтобы добавить язык в Sublime Text, вы используете [Control Package](#) . Вы можете сделать это, вставив правильный код Python (доступный на сайте управления пакетами, связанный выше), или загрузив файл `.sublime-package` (также доступный для загрузки на сайте). После настройки управления пакетами перезапустите Sublime Text.

Чтобы установить пакет обработки из управления пакетами, выполните следующие действия:

- Выполнить инструменты | Установите обработку-java, чтобы установить инструмент обработки-java. Этот инструмент необходим для создания эскиза обработки в командной строке и редакторах, таких как Sublime Text и Atom.
- Откройте командную палитру (Ctrl + Shift + P или Cmd + Shift + P)
- Поиск «Управление пакетами: установка пакета»
- Найдите «Обработка» и установите пакет
- Перезапустите Sublime Text

После выполнения этих шагов вы должны выбрать «Обработка» в качестве языка. Это облегчит кодирование в обработке в Sublime Text.

Использование обработки с помощью редактора Atom

Существует несколько пакетов, которые могут запускать Обработку эскизов в редакторе Atom. Эти инструкции используют пакет [Script](#) . Также доступны пакеты для [подсветки синтаксиса](#) и [автозаполнения](#) , которые требуются для сценария для определения типов файлов обработки.

Установите плагин Script. Либо запустить `apm install script` из командной строки, либо `apm install script` поиск «скрипта» пакета `rgbkrk` на вкладке « Install » в настройках «Atom» (`command + ,` быстрого доступа `command + ,` , или `ctrl + ,`).

После установки скрипта вам потребуется установить `processing-java` . Этот инструмент поставляется с основным программным обеспечением обработки и необходим для создания эскизов обработки в командной строке и в редакторах:

- **MacOS:** Запустить `Tools > Install "processing-java"` .
- **Linux:** добавьте каталог обработки в `PATH` среды `PATH` (замените `/path/to/processing` `C` помощью пути, в котором установлена обработка):

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

- **Windows:** добавьте каталог обработки в `PATH` среды `PATH` :
 - Откройте расширенные системные настройки, запустив `sysdm.cpl` или выполнив поиск в панели управления.
 - Нажмите кнопку «Переменная окружения» на вкладке «Дополнительно».
 - Измените переменную `PATH` чтобы включить каталог обработки в переменные пользователя (только для вашей учетной записи) или системные переменные (для всех пользователей).

Теперь вы можете запустить Обработку эскизов, выполнив `Packages > Script > Run Script` . Ярлык по умолчанию - `command + shift + b` или `ctrl + shift + b` , но для дальнейшего уменьшения боли перехода вы можете связать ярлык Processing IDE для запуска эскиза. Для этого:

- Откройте файл Atom Keуmap, запустив `File > Keуmap`
- Вставьте следующие строки в конце файла (не стесняйтесь изменять привязку к тому, что вы хотите).

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

Использование обработки с помощью Eclipse

Чтобы использовать Обработку в Eclipse, начните с создания нового Java-проекта. Затем выберите `« File > Import а затем « General > File System` чтобы найти файл `core.jar` . Его можно найти в `PATH_TO_PROCESSING/core/library/` для Windows или `/Applications/Processing 3.app/Contents/Java/core/library/` для Mac. Как только это будет завершено, щелкните правой кнопкой мыши на `core.jar` и добавьте его в путь сборки.

Шаблон для использования Обработка в Eclipse выглядит следующим образом:

```
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 Processing IDE
public void draw(){

}

}

```

Метод `settings()` используется для установки размера окна. Например, чтобы создать окно 400x400, напишите следующее:

```

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

```

Все остальное, как указано в [документации Hello World](#), с точки зрения использования `setup()` и `draw()` применяется здесь.

В качестве окончательного примера, вот код из [примера Drawing a Line](#), он должен быть записан в 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 Processing IDE
    public void draw(){
        line(0, 0, 500, 500);
    }

}

```

Прочитайте Использование обработки с альтернативными редакторами кода, такими как Sublime и Atom онлайн: <https://riptutorial.com/ru/processing/topic/5594/использование-обработки-с-альтернативными-редакторами-кода--такими-как-sublime-и-atom>

глава 3: Использование обработчика P3D для обработки

замечания

Ссылка на использование P3D находится в <https://processing.org/tutorials/p3d/> в учебнике Даниэля Шиффмана.

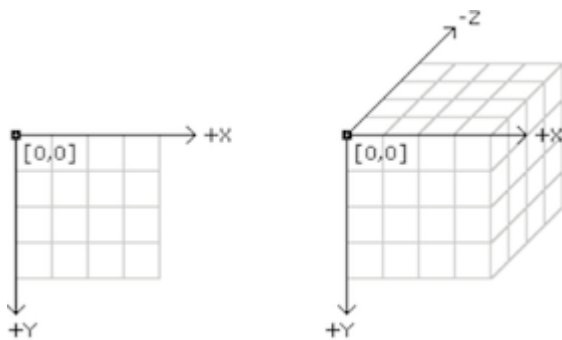
Examples

Начиная

Чтобы иметь возможность использовать средство визуализации P3D, вы должны указать его в функции `size()`

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

Теперь есть три оси: x , y и z .



Теперь вы можете продолжить рисование в 3D.

Прочитайте [Использование обработчика P3D для обработки онлайн](https://riptutorial.com/ru/processing/topic/7589/использование-обработчика-p3d-для-обработки-онлайн):

<https://riptutorial.com/ru/processing/topic/7589/использование-обработчика-p3d-для-обработки>

глава 4: Начало работы с обработкой для Android

замечания

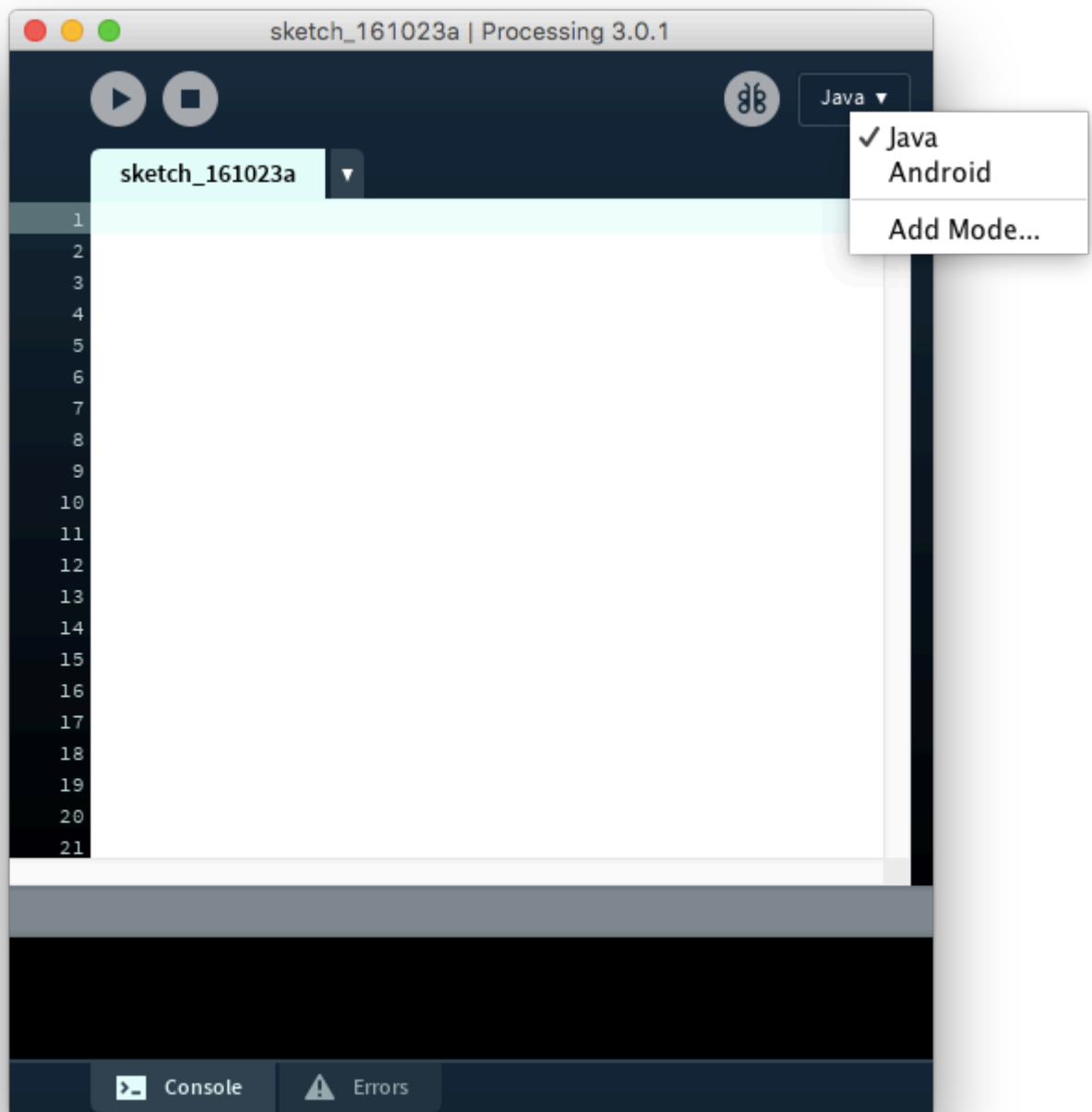
Дополнительную информацию об обработке для Android можно найти по адресу <http://android.processing.org/>

Examples

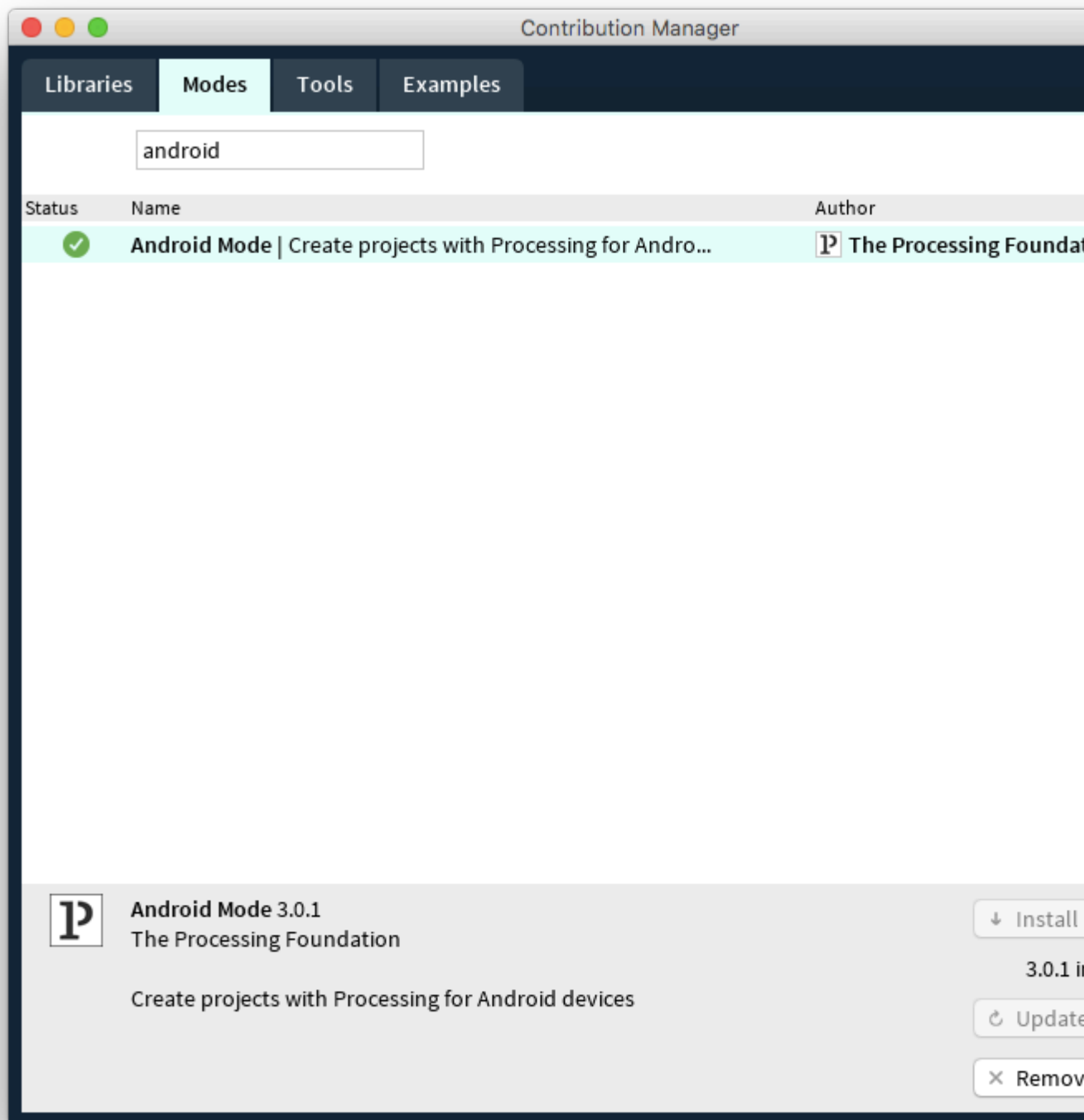
Монтаж

Прежде чем приступить к обработке для Android, убедитесь, что у вас установлена обработка.

1. Открытая обработка
2. Нажмите «Добавить режим», расположенный в правом верхнем углу окна:



3. Найдите «android» в диспетчере вкладок и выберите вариант, который имеет автора «The Processing Foundation» и нажмите «Установить»



4. Как только установка будет завершена, переключите режим на Android
5. Это окно должно появиться. Если у вас нет Android SDK, нажмите «Загрузить SDK автоматически», чтобы установить его. Если вы уже установили Android SDK, нажмите «Найти путь SDK вручную».



- 5,5. Для меня я нашел путь к SDK в `/Users/my-username/Library/Android/sdk`
6. Все готово!

Прочитайте [Начало работы с обработкой для Android онлайн](#):

<https://riptutorial.com/ru/processing/topic/7586/начало-работы-с-обработкой-для-android>

глава 5: Основные формы и функции с использованием P3D

Синтаксис

- translate (float x, float y, float z)
- rotateX (поплавок)
- rotateY (угол поплавка)
- rotateZ (угол поплавка)
- box (размер поплавка)
- box (float w, float h, float d)

параметры

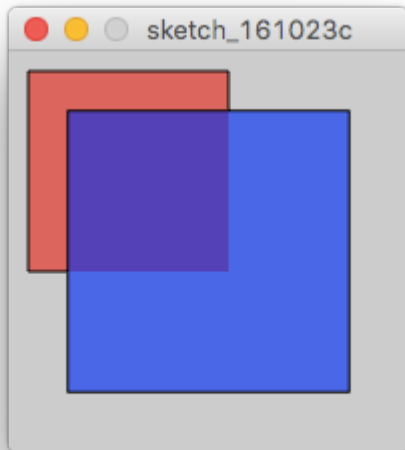
параметры	подробности
угол	угол в радианах
размер	размер окна, который будет использоваться для всех его размеров
вес	размер окна по <code>x-axis</code>
час	размер окна по <code>y-axis</code>
d	размер окна по <code>z-axis</code>

Examples

3D-перевод

Вот как перевести объекты в 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)
```



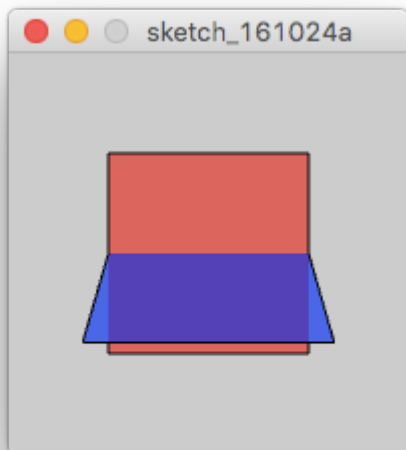
Красный: первый прямоугольник Синий: второй прямоугольник

Как видно из приведенного выше эскиза, второй прямоугольник только кажется более крупным, чем первый, когда на самом деле он «ближе» к экрану в результате перевода прямоугольника на 50 пикселей вдоль z -axis (и Конечно, прямоугольник был переведен вдоль осей x и y).

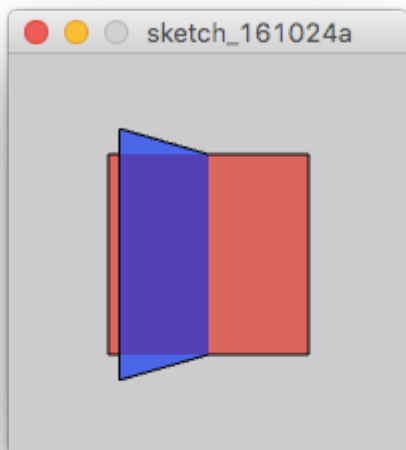
3D-ротация

Для трехмерного вращения есть три функции: `rotateX(angle)` , `rotateY(angle)` И `rotateZ(angle)` для вращения в соответствующих осях, где `angle` находится в радианах.

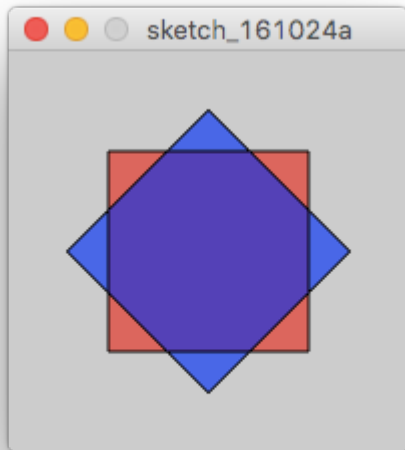
```
size(200, 200, P3D); //Starting P3D renderer
fill(255, 0, 0, 150); //transparent red
translate(width/2, height/2); //translate to centre, ie (100, 100)
rectMode(CENTER); //This makes the rectangle centre in (100, 100)
rect(0, 0, 100, 100); //first rectangle
fill(0, 0, 255, 150); //transparent blue
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)
```

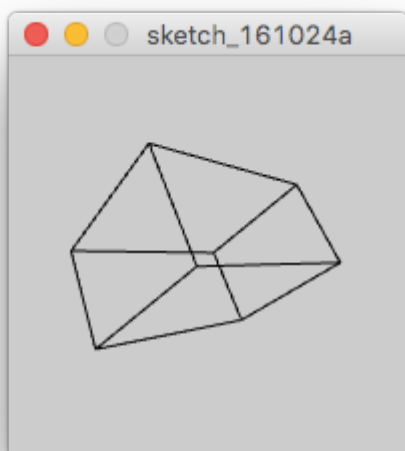


Примечание. Преобразования (такие как переводы и вращения) добавляют к предыдущему преобразованию.

Рисование кубоида

Чтобы нарисовать кубоид, вы должны использовать функцию `box()`, указав ее размеры в качестве параметров.

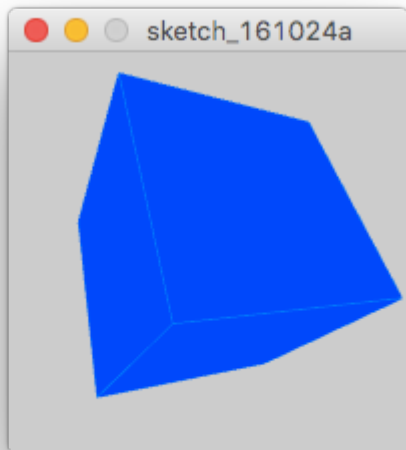
```
size(200, 200, P3D); //Starting the P3D renderer
translate(width/2, height/2); //Translating to the centre of the sketch
rotateY(PI/4); //rotate so that...
rotateX(PI/6); //... it will be easy to see the box
noFill(); //disabling the box's fill, so that we will be able to see its edges
box(100, 50,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



Обратите внимание, что функция `box()` *не* принимает свою позицию в качестве параметров

Также есть способ вызвать функцию `box()` только с одним параметром. В этом случае это будет куб.

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



Прочитайте [Основные формы и функции с использованием P3D онлайн:](https://riptutorial.com/ru/processing/topic/7591/основные-формы-и-функции-с-использованием-p3d)

<https://riptutorial.com/ru/processing/topic/7591/основные-формы-и-функции-с-использованием-p3d>

глава 6: Рисование основных фигур

Вступление

В обработке формы чертежа являются ключевыми для программы. В противном случае на экране ничего не появится. В этом разделе рассказывается, как рисуются основные фигуры.

Синтаксис

- line (float x1, float y1, float x2, float y2)
- строка (float x1, float y1, float z1, float x2, float y2, float z2)
- эллипс (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)

параметры

параметр	подробности
x1	x-координата первой точки
y1	y-координата первой точки
z1	z-координата первой точки
x2	x-координата второй точки
y2	y-координата второй точки
z2	z-координата второй точки
x3	x-координата третьей точки
y3	y-координата третьей точки
Икс	x-координата
Y	y-координата
вес	ширина
час	рост

замечания

Вы можете найти ссылку на основание Процесса.

[Домашняя страница обработки](#)

Examples

Рисование линии

Обработка предоставляет метод с именем `line()` для рисования строки на экране. Этот код рисует белую 10-пиксельную линию на черном фоне.

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

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

Подпись метода `line()` - это.

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

`x1` и `y1` - координата начальной точки. `x2` и `y2` - координата конечной точки.

Метод `stroke()` используется для указания цвета линии, которую вы рисуете.

Метод `strokeWeight()` используется для указания толщины линии, которую вы рисуете. (в пикселях)

Рисование прямоугольника

Обработка предоставляет метод `rect()` для рисования прямоугольника. Этот код рисует белый прямоугольник 50 X 50 на черном фоне.

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

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

```
}
```

Подпись метода `rect()` - это.

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

`x` и `y` - координата прямоугольника. `w` и `h` - ширина и высота прямоугольника.

Метод `fill()` используется для указания цвета заполнения прямоугольника и других фигур, таких как эллипс, треугольник, многоугольник.

Метод `noStroke()` используется для указания того, что вокруг прямоугольника нет штрихов. Этот метод также влияет на другие формы, такие как эллипс, треугольник, многоугольник.

Рисование эллипса

Обработка обеспечивает `ellipse` метода для рисования эллипса. Этот код рисует белый круг с радиусом 25 пикселей.

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

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

Подпись метода `ellipse()` - это.

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

`x` и `y` - координата эллипса. `w` и `h` - ширина и высота эллипса.

Рисование треугольника

Обработка обеспечивает `triangle` метода для рисования треугольника. Приведенный ниже код рисует почти равносторонний треугольник в 25 пикселей между каждой определяющей точкой.

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

void draw() {
  triangle(0, 0, 25, 0, 12, 12);
}
```

Подпись `triangle` такова:

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

Каждая x точка соответствует оси x точки, а y - оси y . Три точки будут соединены, чтобы сформировать треугольник.

Рисование треугольника

Обработка предоставляет метод `triangle()` для рисования треугольника. Этот код рисует белый треугольник на черном фоне.

```
void setup() {  
  size(500, 500);  
  background(0);  
  fill(255);  
  noStroke();  
}  
  
void draw() {  
  triangle(250, 225, 225, 275, 275, 275);  
}
```

Прочитайте [Рисование основных фигур онлайн](https://riptutorial.com/ru/processing/topic/7277/рисование-основных-фигур):

<https://riptutorial.com/ru/processing/topic/7277/рисование-основных-фигур>

глава 7: Цвета в обработке

Вступление

В этой статье будут показаны различные форматы цветов в разделе «Обработка» и способы их использования.

Синтаксис

- цвет (r, g, b);
- цвет (r, g, b, альфа);
- цвет (серый);
- цвет (серый, альфа);
- цвет (h, s, l); // Режим должен быть HSB. Вы можете изменить это с помощью colorMode.

параметры

параметры	подробности
р	Красный цвет, когда режим <code>RGB</code> .
г	Является зеленым цветом, когда режим <code>RGB</code> .
б	Является синим цветом, когда режим <code>RGB</code> .
альфа	Является непрозрачностью цвета.
час	Цветовая гамма цвета, когда режим <code>HSB</code> .
s	Насыщенность цвета, когда режим <code>HSB</code> .
L	Яркость / яркость цвета, когда режим <code>HSB</code> .
серый	Значение между черным (0) и белым (255).

замечания

Хотя он не упоминается в официальной документации для обработки, есть режим `CMYK` который вы можете использовать.

Examples

Обозначение цвета

Существуют различные способы использования цветов в обработке, поскольку обработка очень гибкая с цветовыми форматами.

RGB и RGBA

Это стандартная нотация RGB (A) и цветной режим по умолчанию. Первые три значения цвета (красный, зеленый, синий) варьируются от 0 до 255. Например, приведенный ниже пример является красным цветом, так как красное максимальное значение равно 255 а остальные цвета - 0. Белый находится на (255, 255, 255) а черный - (0, 0, 0).

Необязательный 4-й параметр указывает альфа-значение, то есть прозрачность. Как и в других компонентах, диапазон значений снова [0-255]; 0 является полностью прозрачным и 255 полностью твердым.

```
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

HSB-нотация похожа на нотацию RGB, за исключением того факта, что красный, зеленый и синий заменяются оттенком, насыщенностью и яркостью соответственно. Вы можете переключиться на HSB с помощью `colorMode(HSB)`.

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

Как и в случае с RGB, HSB также имеет значение альфа в качестве четвертого параметра.

Серые значения

Если один параметр указан для цветовой функции, он будет интерпретироваться как количество между черным и белым. Белый представлен как 255, а черный - как 0. Он совпадает с `color(param1, param1, param1)` в режиме RGB. Если указаны два параметра, то первый параметр будет интерпретироваться так, как указано выше, а второй будет альфа-значением.

Прочитайте Цвета в обработке онлайн: <https://riptutorial.com/ru/processing/topic/8283/цвета-в-обработке>

кредиты

S. No	Главы	Contributors
1	Начало работы с обработкой	aspiring_sarge , Community , Cows quack , Kevin Workman , MasterBob , Michael Andersen , user2314737
2	Использование обработки с альтернативными редакторами кода, такими как Sublime и Atom	bakahoe , CodeMacabre , Kevin Workman , Mert Toka , Peter , sohnryang
3	Использование обработчика P3D для обработки	Cows quack
4	Начало работы с обработкой для Android	Cows quack
5	Основные формы и функции с использованием P3D	Cows quack
6	Рисование основных фигур	Cows quack , MasterBob , sohnryang
7	Цвета в обработке	Cows quack , MasterBob , Mert Toka