

Processingによる心理実験プログラミング入門

2013年9月20日

津田裕之

日本心理学会第77回大会 チュートリアルワークショップ

本チュートリアル概要

目的

Processingという言語の紹介

その特徴を知り、ツールの1つとして導入するきっかけに

対象者

- プログラミング言語を用いて実験を作成されている方
- 基本的なプログラミングの知識が既にあると想定

本日の予定

11:30

Processingの概要

基本的な使い方

Processingの特徴と心理実験への応用性

質問

12:50

13:00

質問は話の途中でして頂いてかまいません

チュートリアル資料について

本日のスライド及び関連資料は私のwebサイトに掲載します

<http://hiroyukitsuda.com/>

自己紹介

津田 裕之(つだ ひろゆき)

京都大学 大学院人間・環境学研究科 在学中

Processing(Java)、C++、C#、Matlabなどを使用

最近はUnityがお気に入り

その他

専門は認知心理学

視覚記憶・空間認知・ナビゲーション等に関する実験心理学的研究

プログラミング言語のこと

言語遍歴

MATLAB → C → Java → Processing → C++ → C#

現在はProcessingとC#とMATLABを目的により使い分けて使用

なぜ色々な言語に手を出したのか

最初はMATLABを使っていた。が、これは高価。

将来を考えると無料の環境で研究が行えるようにしておきたかった。

フリーウェアが今後の主流／トレンドに？

統計解析の分野では、**R**がSASやSPSSに取って代わろうとしている

実験用の言語も、今後は**Python**等フリーのものが主流になると思われる

その流れの1つに**Processing**という選択肢もあり得るのではないかな？

実験用の言語に求められること

手軽に実験プログラミングが行なえる

- 環境の導入が容易で、学習の敷居も低いものが良い
- 思いついた実験や視覚刺激のアイデアをすぐさま形にしたい

無料で利用出来る

- お金はケチれるならその方が良い
- 講義など多数のライセンスが必要な場合に有料ソフトの利用を避けたい

様々な環境で使える

- Windows, Mac, Linux
- タブレット端末やWEB上でも

これらのニーズにProcessingは対応できます！

Processingとは何か



Processingとは何か

“Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.”



Processingとは何か

「Processingとは、プログラミング言語であり、開発環境であり、そしてオンラインコミュニティです。2001年以来、Processingはアーティストをソフトウェアに習熟させ、技術者には芸術のセンスをもたらし来しました。Processingは当初は**ソフトウェアにおけるスケッチブック**として、またプログラミングの基礎を視覚的な文脈を通して教える為に作られましたが、現在では専門家の開発ツールとして発展しました。今では、数多くの学生、アーティスト、デザイナー、研究者、趣味人がProcessingを使って学習やプロトタイピングや生産活動を行っています」

公式サイトより



Processingとは何か

Processingとは

- MIT Media Labで2001年にBen FryとCasey Reasにより開発された
- オープンソースソフトウェアとして現在も開発が続けられている

Processingの特徴

- 初心者でも容易に習得可能な言語として作られた
- ビジュアルプログラミング（絵や映像の作成）を主な用途に想定
- Javaベースの言語であり、実行環境とエディタが用意されている
- Windows、MacOSX、Linuxで実行可能
- web上やタブレット上でもコードをJavascriptに変換し実行出来る
- 様々なライブラリが盛んに開発されており、高度な機能も手軽に利用できる（画像処理や各種センサーとの連携等）

Processingの主なユーザー

どのような層がユーザーなのか？

Processingは**アーティストや芸術系の学生**の作品制作のためによく利用されており、芸術大学でプログラミング講義用の言語として採用されている

最近**工学系**でも用いられることが多くなっており、特にインタラクション系の研究ではシステムのプロトタイピングのために使われることが多い

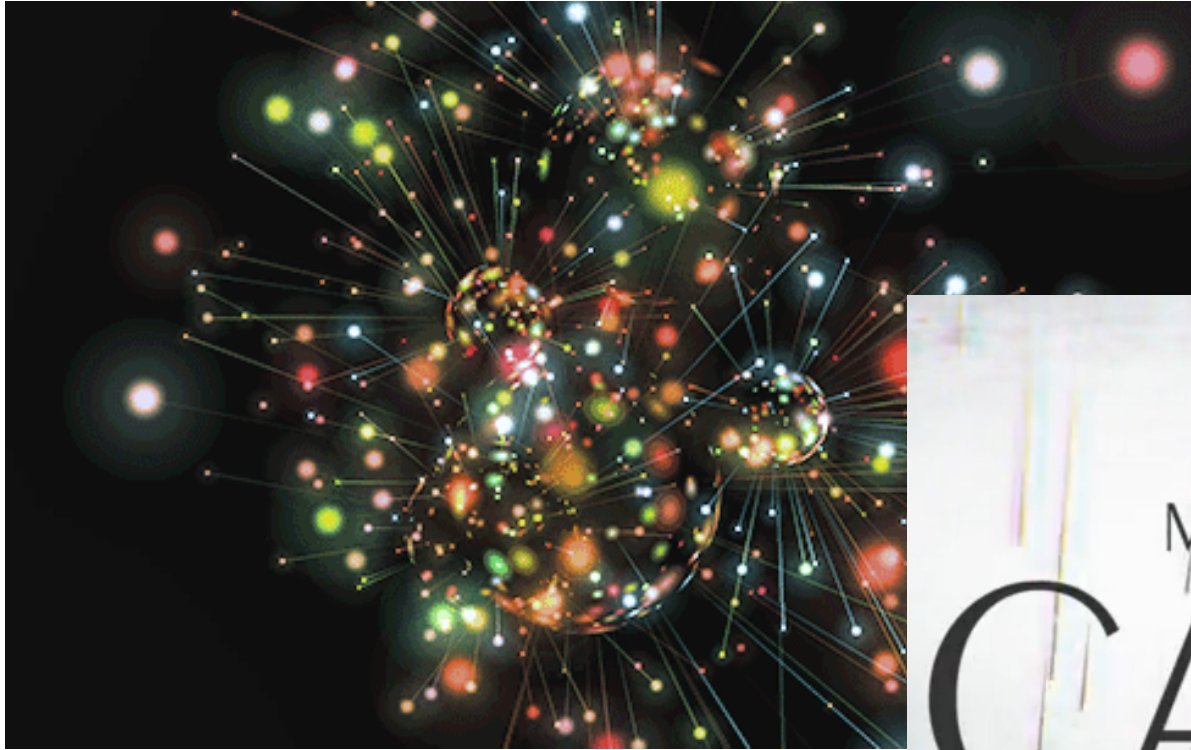
アート作品やアカデミックな研究以外にも、電子工作やデバイス制御のためのツールとして日曜プログラマーに使われている

(これまでの所、心理学研究者に使用された例はほぼ見たことがない)

Processingはどのようなことに使われているか

映像作品

Magnetosphere (iTunes 8のビジュアライザプラグイン)
映像制作のプロトタイピングにProcessingが利用された



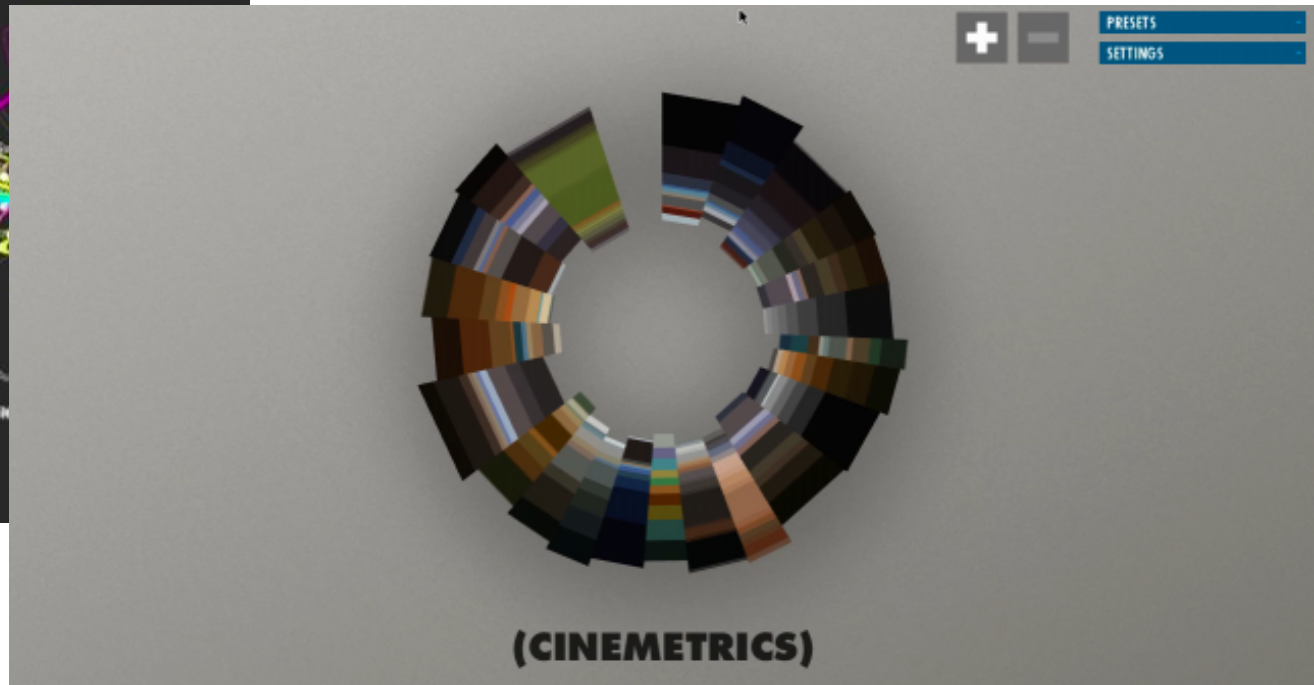
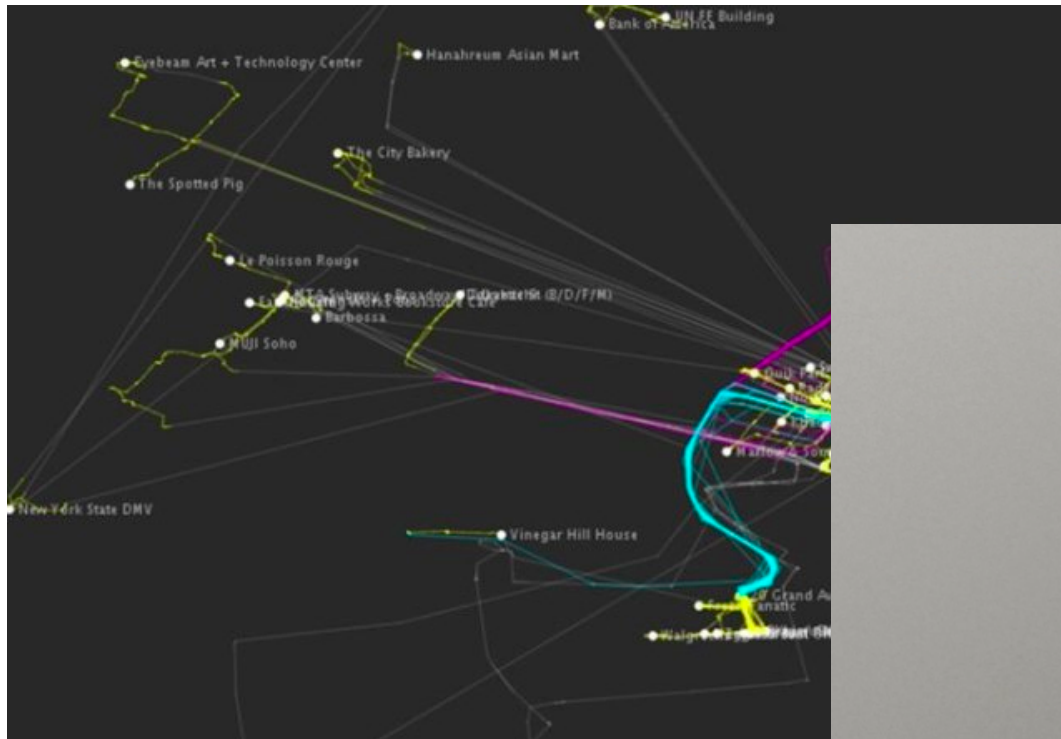
Music video for Catalina, a track off the Chocolat EP
<http://vimeo.com/19723907>

情報可視化

MovesMapper – Processing sketch for rendering location and activity

ユーザーの空間内の移動経路を可視化

<http://bit.ly/15xfJdZ>



Cinematics

映画の特徴量 (場面ごとの色や動き等) をリング状に可視化する

<http://bit.ly/15xhxUn>

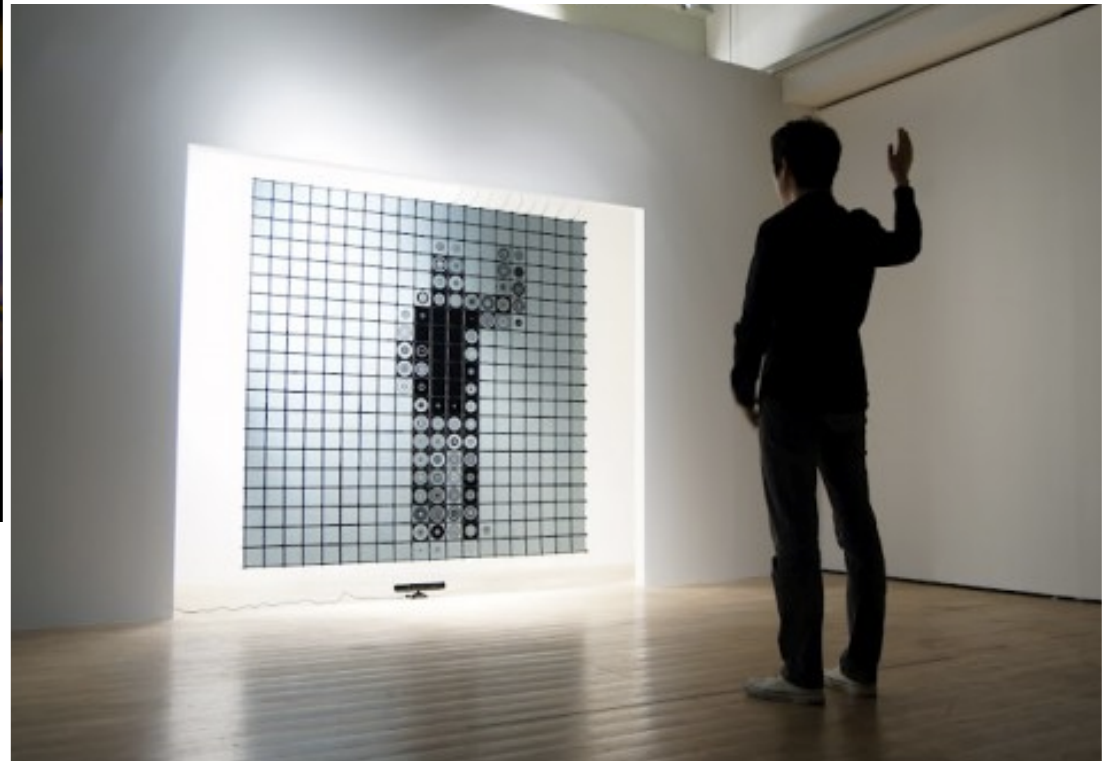
インスタレーション

Firewall – Stretched sheet of spandex as a visual instrument

<http://bit.ly/15xeueD>



インタラクティブな視覚アート

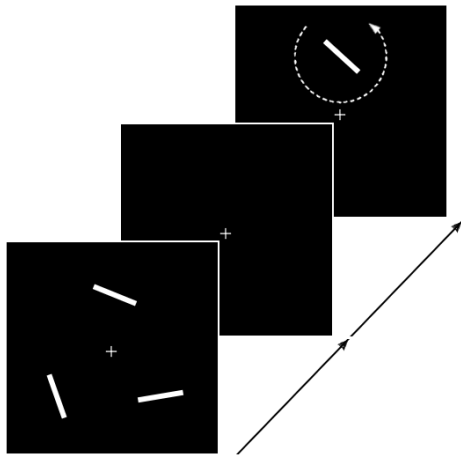


IRIS by HYBE – New kind of monochrome LCD display

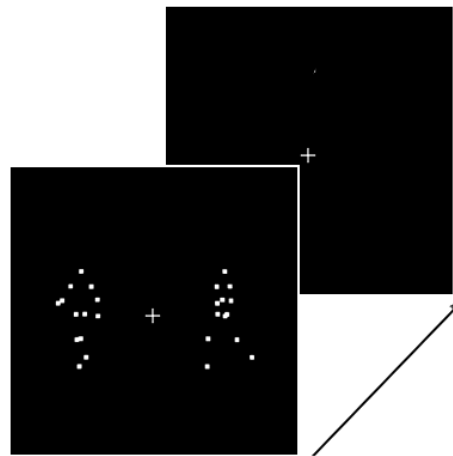
<http://bit.ly/1di9LPT>

心理実験の例

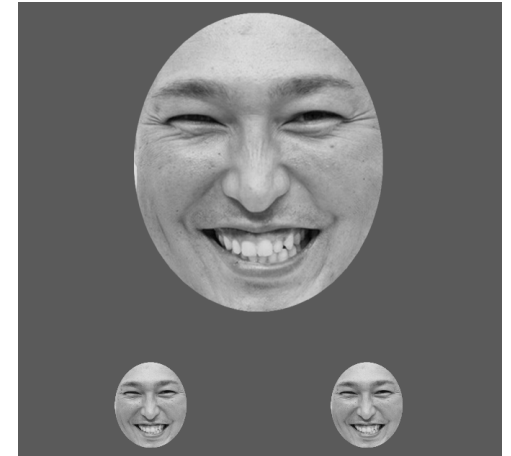
私がこれまでにProcessingを用いて行った実験



短期記憶課題



バイオロジカルモーションの認知



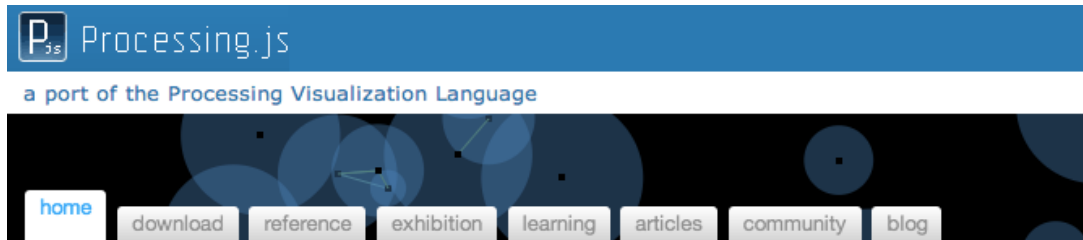
顔の知覚判断課題

ベーシックな知覚課題や認知課題も、もちろん作成可能

webやタブレット上での実行

Processing.js

Processingで書いたコードをJavascriptに変換し、webブラウザ上で実行できる



Processing Android

Android端末でProcessingのコードを実行できる



Procoding

iOS端末でProcessingのコーディングと実行ができる

Processingの利用形態の多様性の背景

豊富なライブラリ

- ビジュアル (画像処理、OpenCV、3D、アニメーション)
- デバイス制御 (Kinect, Leap motion, Arduino, OSC, etc.)
- その他 (サウンド、GUI、数学関数)

ユーザーコミュニティの多様性と活発さ

- 多様なユーザー層 (Artists, Scientists, Hobbyists)
- 面白いデバイス (KinectやLeap motion等)が登場すると、一ヶ月もしないうちに対応したProcessingライブラリを誰かが作る

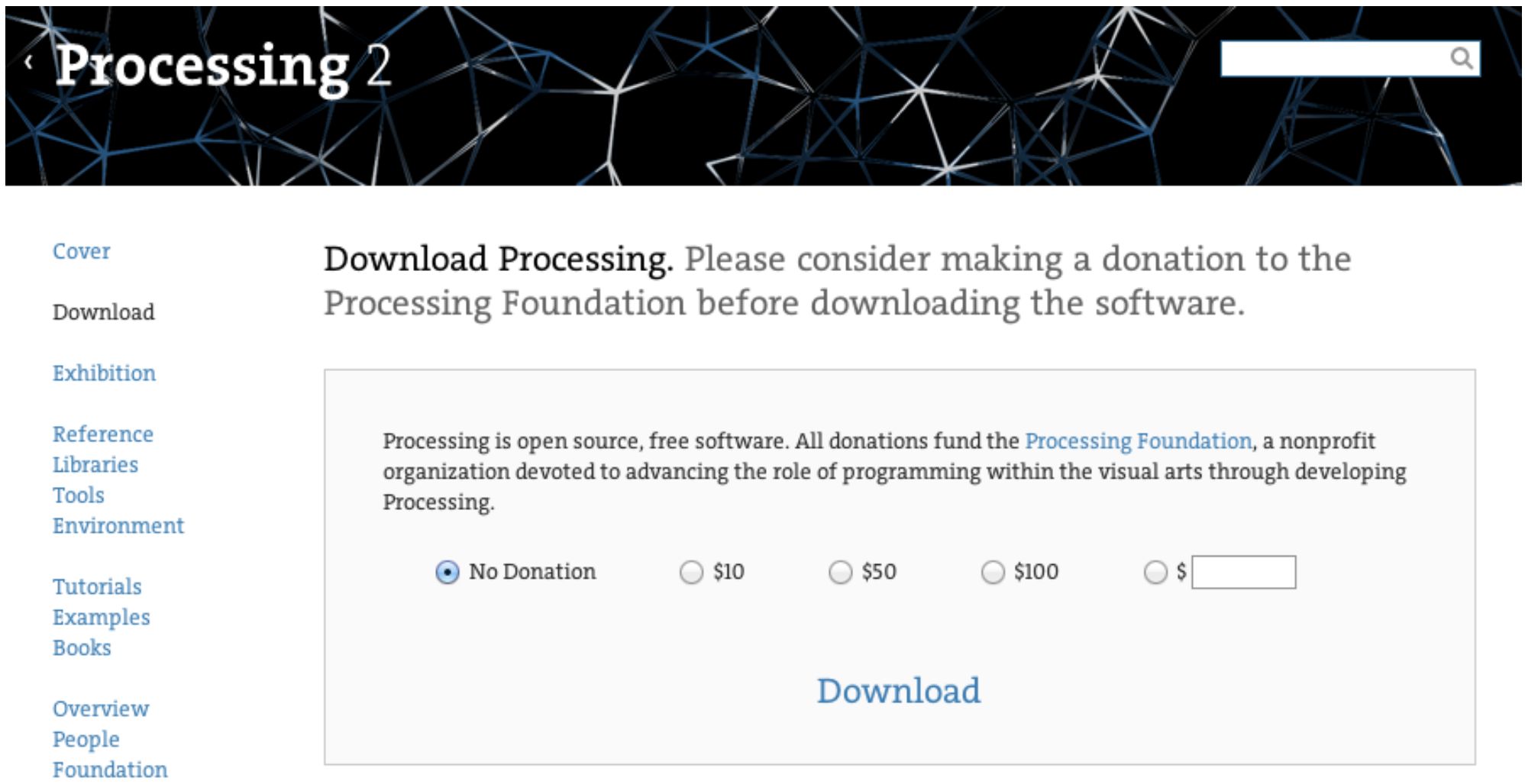
Processingの基本操作



インストール

入手方法

Processingの公式サイト (<http://processing.org/>) で入手可能
左のメニューからDownloadページへ行き、No Donationを選択



The screenshot shows the Processing.org website. The header features the 'Processing 2' logo on the left and a search bar on the right. A left-hand navigation menu lists various links: Cover, Download, Exhibition, Reference, Libraries, Tools, Environment, Tutorials, Examples, Books, Overview, People, and Foundation. The main content area displays the text 'Download Processing. Please consider making a donation to the Processing Foundation before downloading the software.' Below this is a donation form with the text: 'Processing is open source, free software. All donations fund the Processing Foundation, a nonprofit organization devoted to advancing the role of programming within the visual arts through developing Processing.' The form includes five radio button options: 'No Donation' (selected), '\$10', '\$50', '\$100', and '\$' followed by an input field. A large blue 'Download' button is positioned at the bottom of the form.

Processing 2

Cover

Download

Exhibition

Reference

Libraries

Tools

Environment

Tutorials

Examples

Books

Overview

People

Foundation

Download Processing. Please consider making a donation to the Processing Foundation before downloading the software.

Processing is open source, free software. All donations fund the Processing Foundation, a nonprofit organization devoted to advancing the role of programming within the visual arts through developing Processing.

☒ No Donation ☐ \$10 ☐ \$50 ☐ \$100 ☐ \$

Download

インストール

バージョン

上部にある最新版を選択しダウンロードする

ver. 1.5.1は安定版



[Cover](#)

[Download](#)

[Exhibition](#)

[Reference](#)

[Libraries](#)

[Tools](#)

[Environment](#)

[Tutorials](#)

[Examples](#)

[Books](#)

[Overview](#)

[People](#)

[Foundation](#)

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



2.0.3 (5 September 2013)

[Windows](#) 64-bit

[Windows](#) 32-bit

[Linux](#) 64-bit

[Linux](#) 32-bit

[Mac OS X](#)

- » [Github](#)
- » [Report Bugs](#)
- » [Wiki](#)

The [list of revisions](#) covers the differences between releases in detail. Please read the [changes](#) if you're new to the 2.0 series.

起動

アイコンをダブルクリックしてProcessingを起動



Processing.app

画面の見方

 プログラムの実行

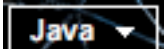
 プログラムの停止

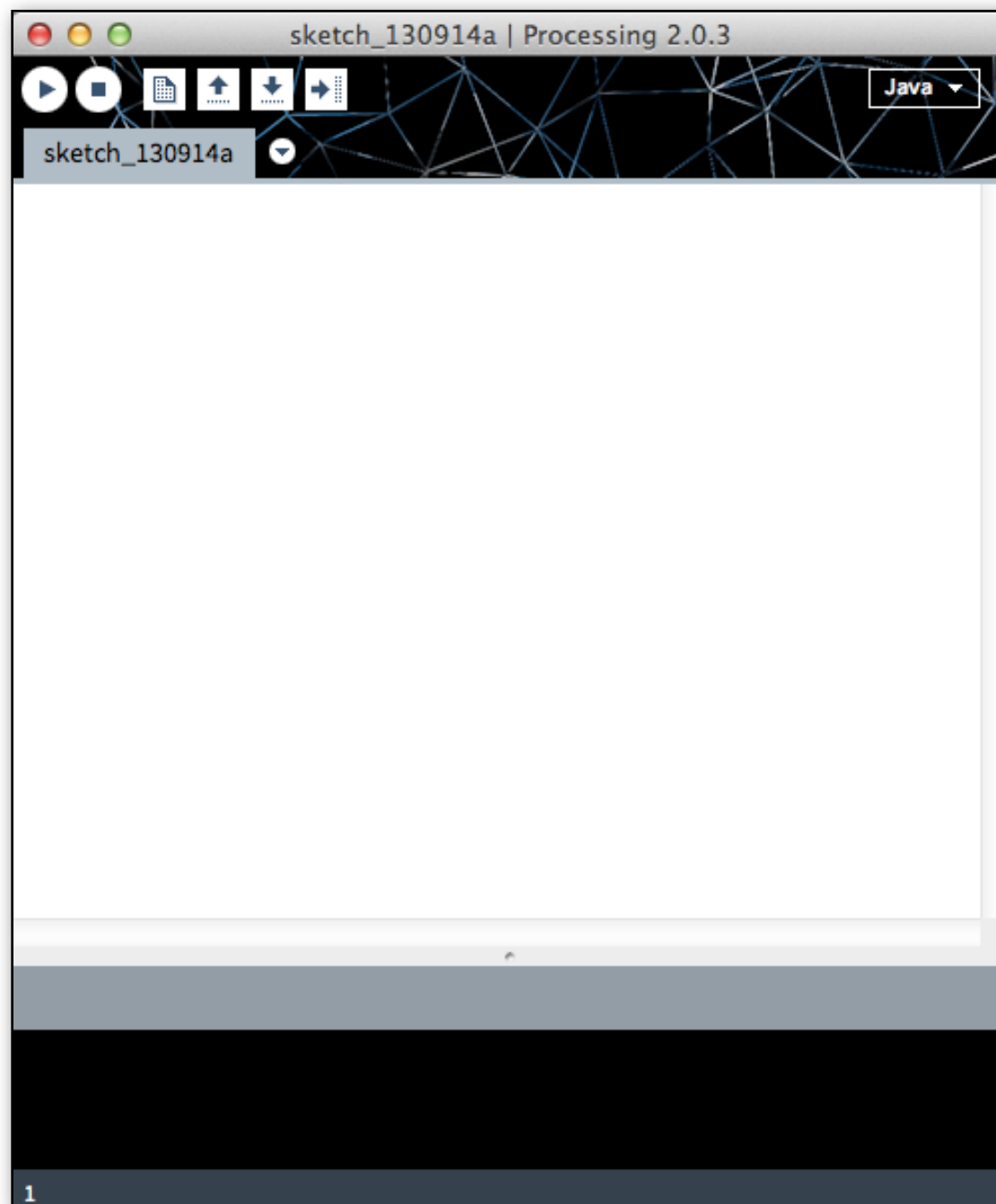
 新規作成

 ファイルの読み込み

 ファイルの保存

 エクスポート

 モード切替



タブ領域

エディタ領域

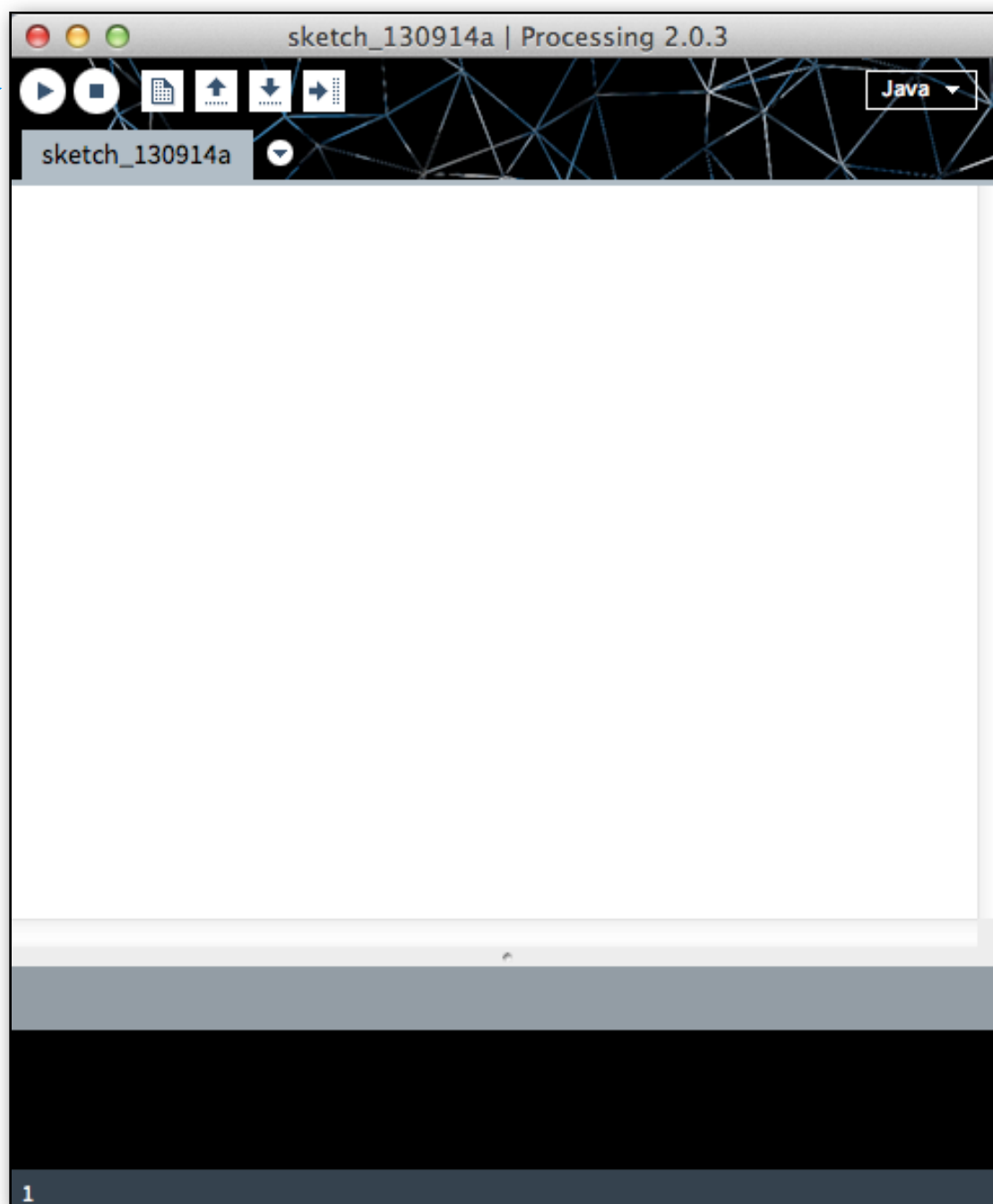
メッセージ領域

コンソール領域

とりあえず動かしてみる

1

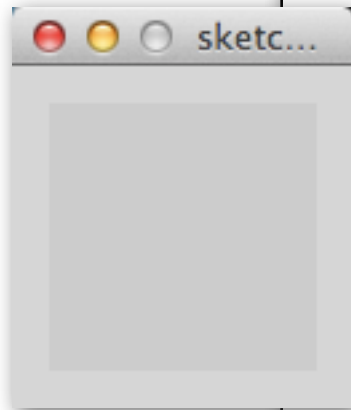
実行ボタン(▶)を
押すと…



とりあえず動かしてみる

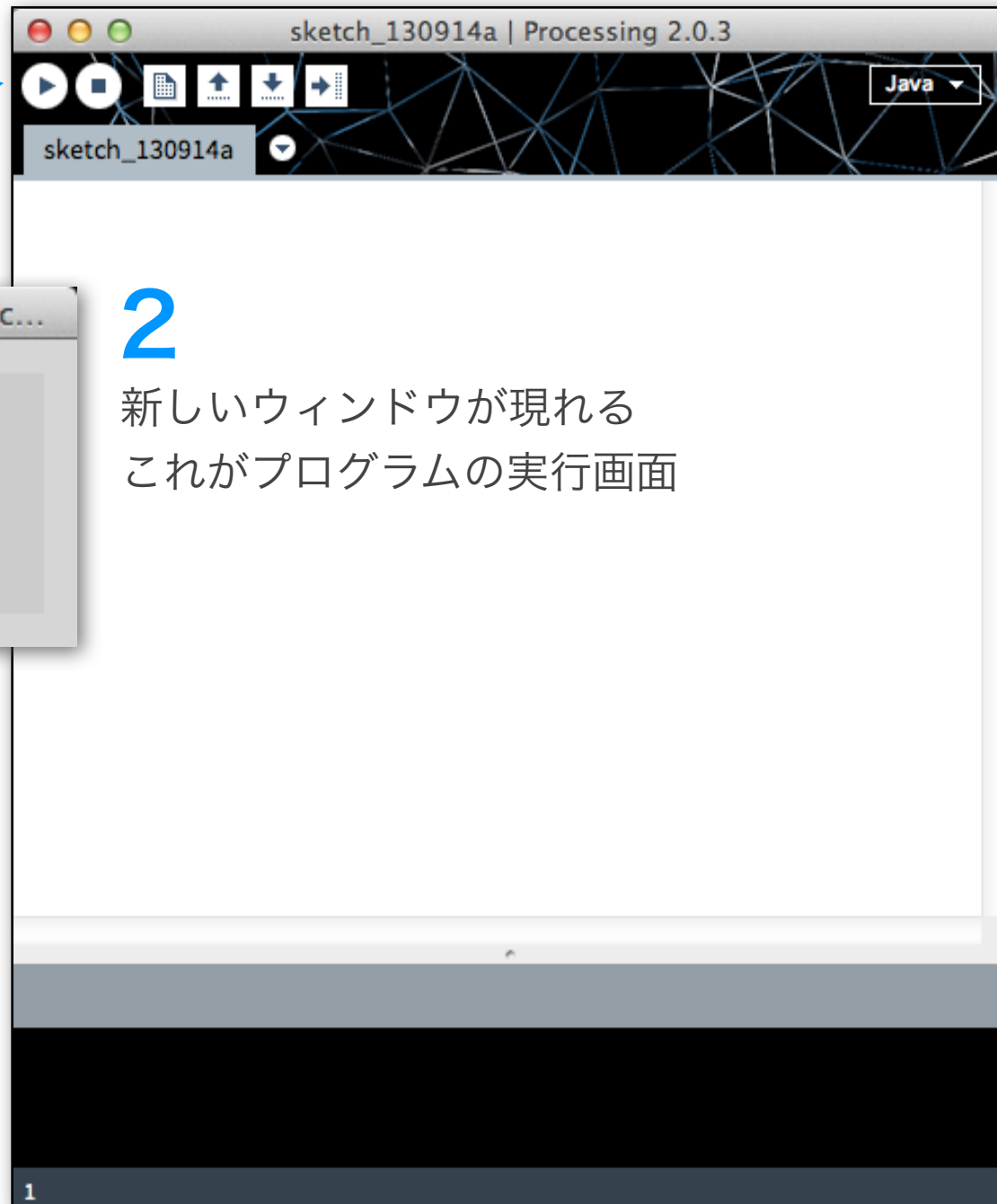
1

実行ボタン(▶)を押すと…



2

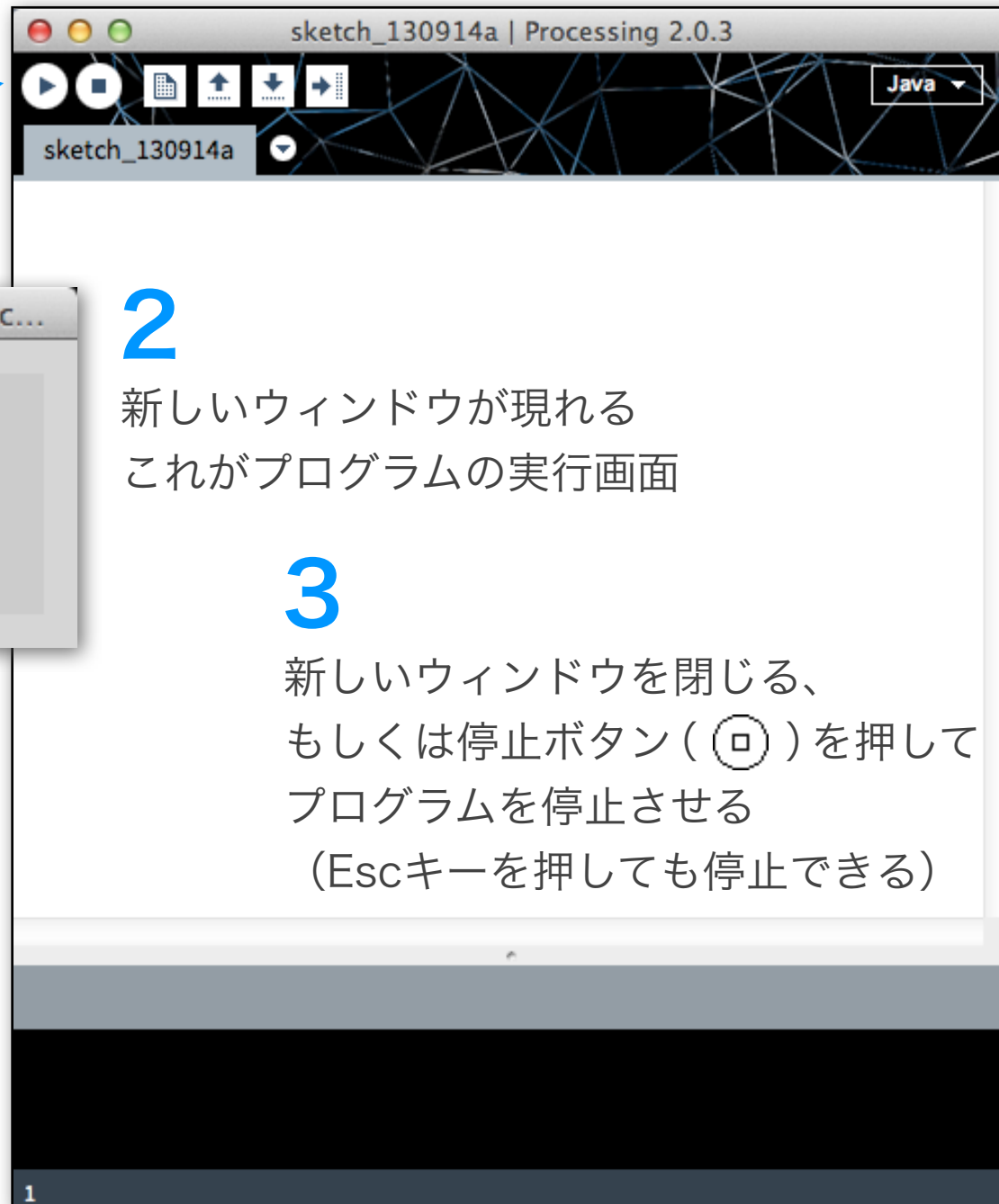
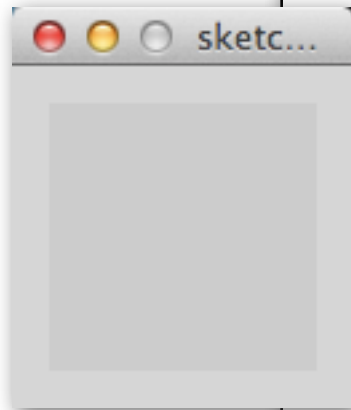
新しいウィンドウが現れる
これがプログラムの実行画面



とりあえず動かしてみる

1

実行ボタン(▶)を押すと…



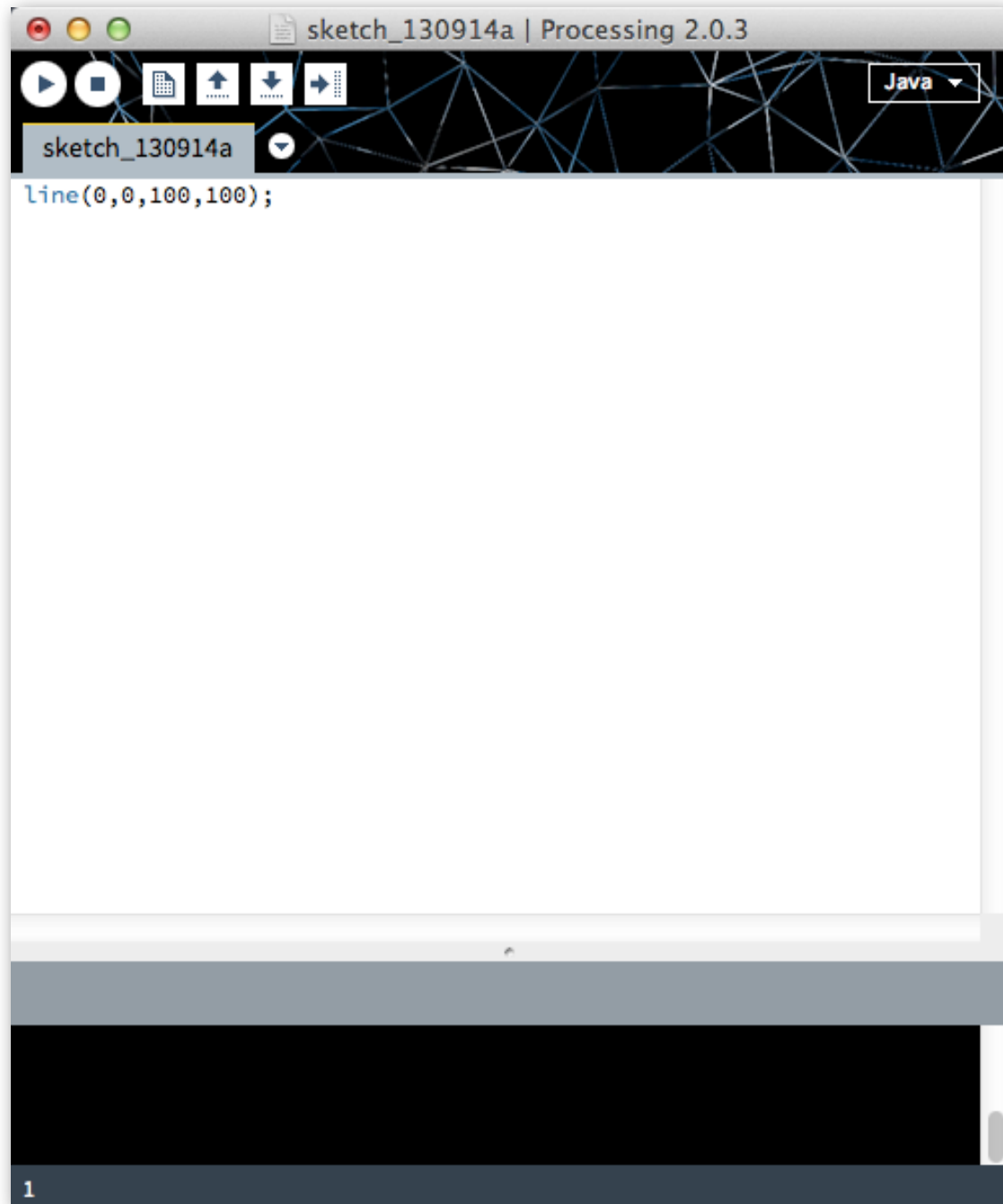
2

新しいウィンドウが現れる
これがプログラムの実行画面

3

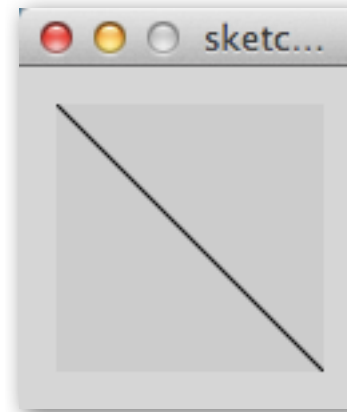
新しいウィンドウを閉じる、
もしくは停止ボタン(◻)を押して
プログラムを停止させる
(Escキーを押しても停止できる)

とりあえず動かしてみる



エディタ領域に以下の内容を記述し、実行する

```
line(0, 0, 100, 100);
```

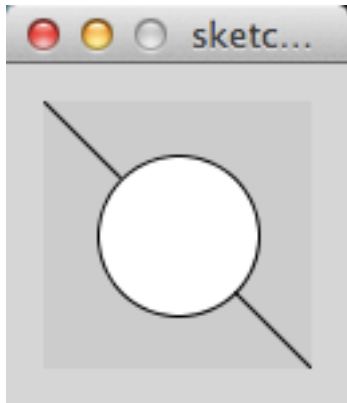


線が描画された

とりあえず動かしてみる

エディタ領域に以下の内容を記述し実行

```
line(0, 0, 100, 100);  
ellipse(50, 50, 60, 60);
```

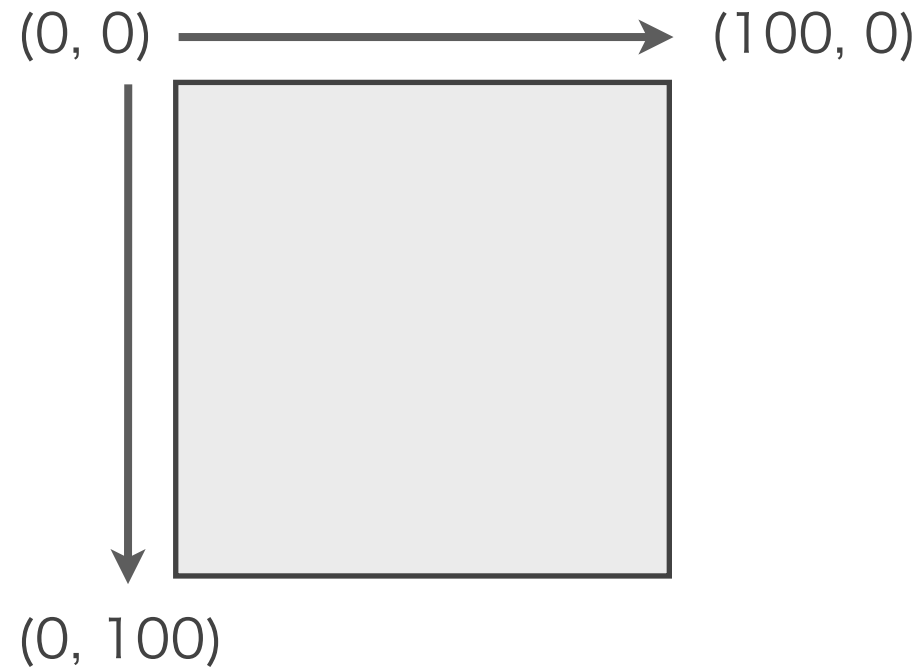


線と円が描画される

Processingの座標系

左上が原点となる

右図は画面サイズが
100 × 100の場合の例



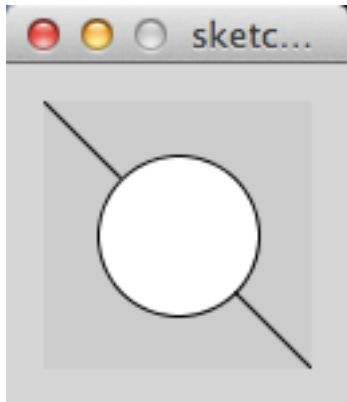
とりあえず動かしてみる

エディタ領域に以下の内容を記述し実行

```
line(0, 0, 100, 100);  
ellipse(50, 50, 60, 60);
```

引数の内容

line(始点のX, 始点のY, 終点のX, 終点のY);
ellipse(中心のX, 中心のY, 円の幅, 円の高さ);



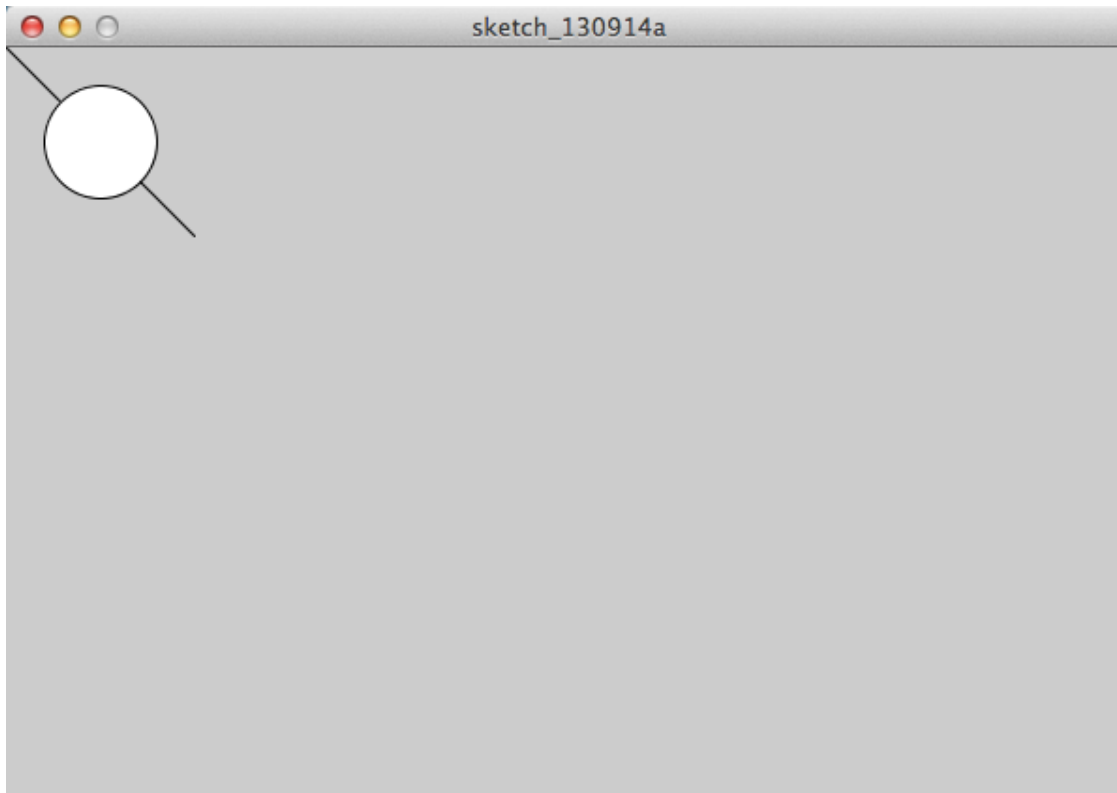
線と円が描画される

とりあえず動かしてみる

実行ウィンドウの大きさを変更する

```
size(600 ,400);  
line(0, 0, 100, 100);  
ellipse(50, 50, 60, 60);
```

size(ウィンドウの幅, ウィンドウの高さ)



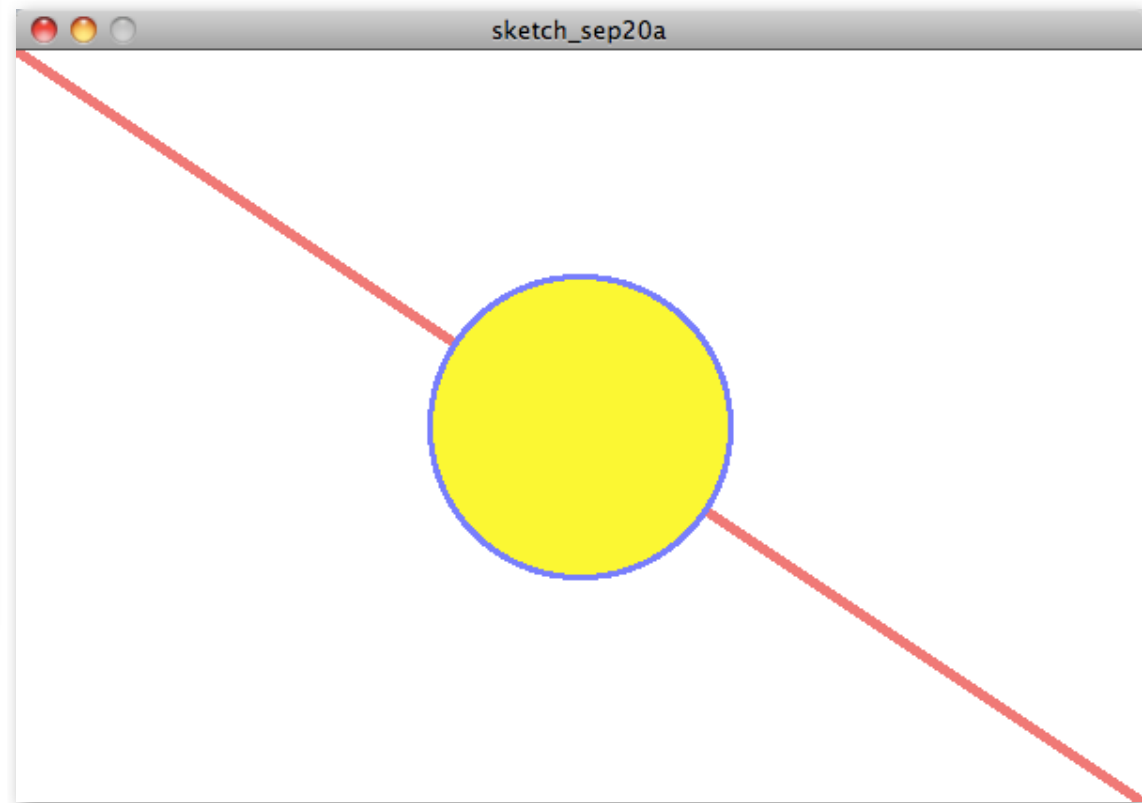
ウィンドウをフルスクリーンにしたい場合
⌘ + Shift + R のショートカットキーで実行
すれば良い

色や線の太さの変更

```
size(600, 400); //画面サイズの指定  
background(255); //背景色
```

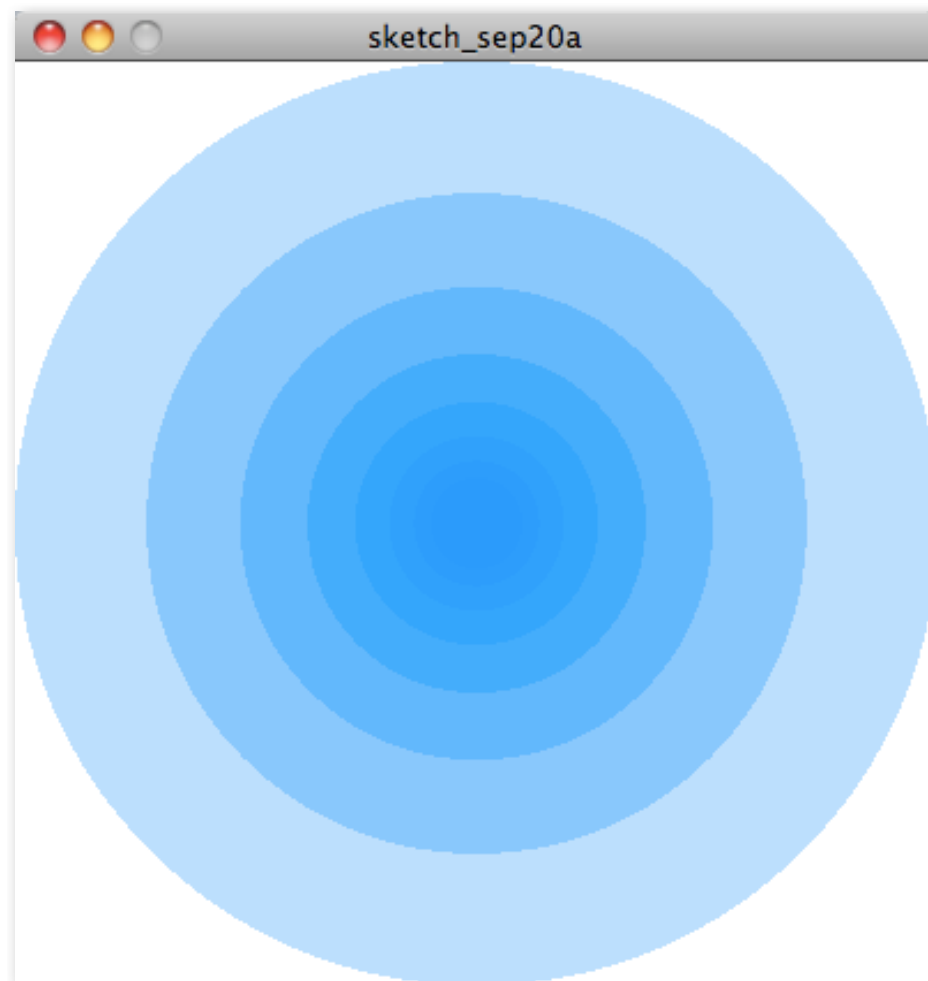
```
stroke(255, 100, 100); //線の色  
strokeWeight(4); //線の太さ  
line(0, 0, 600, 400);
```

```
stroke(100, 100, 255); //線の色  
strokeWeight(2); //線の太さ  
fill(250, 250, 0); //塗りつぶし色  
ellipse(300, 200, 160, 160);
```



繰り返しによる描画

```
size(400,400);  
background(255);  
  
noStroke(); // 輪郭線を描かない  
fill(0, 127, 255, 80); //R,G,B,A  
  
float d=400;  
for(int i=0; i<8; i++) {  
    ellipse(200, 200, d, d);  
    d = d/1.4;  
}
```



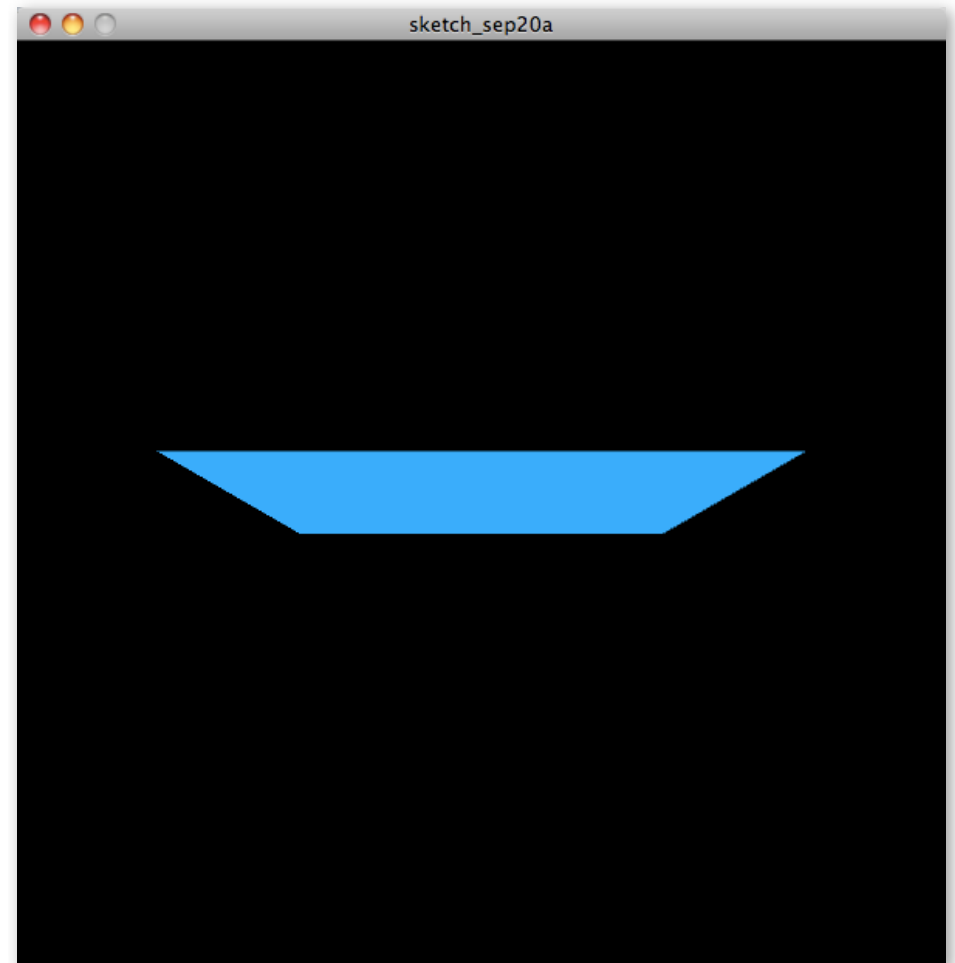
3Dでの描画

```
import processing.opengl.*; // OpenGLライブラリをインポート

float rot=0;

void setup() {
  size(600, 600, OPENGLE); // OpenGLを使用
  frameRate(60); // fpsを60に指定
  fill(200, 100, 100);
  noStroke();
  rectMode(CENTER); // rect関数に関する設定
}

void draw() {
  background(0);
  translate(width/2, height/2);
  rotateX(radians(rot));
  rect(0, 0, width/2, width/2);
  rot += mouseX/50.0;
}
```



心理実験用言語としてのProcessingの可能性



Processingまとめ

Processingは…

簡単です

- 導入の敷居は低く、文法も平易です

無料です

- 個人でも、また集団での利用にも便利です

使い勝手も良い

- マルチプラットフォーム (Win, Mac, Linux, Android, web)
- 描画関数を使いやすく、絵作りがしやすい

高度な機能にも対応

- 画像処理・デバイス連携などのライブラリが豊富

Processingの短所

動作速度

- C (C++/C#) に比べると速度に劣る
- 計算量の多い処理やデータ解析には使いにくい

時間制御

- 描画 (フレーム間隔) のばらつきが大きい (伊丸岡, 2010)
- 呈示時間・反応時間をミリ秒単位の厳密さで見る必要のある課題には不向き

既存の心理用ライブラリが使えない

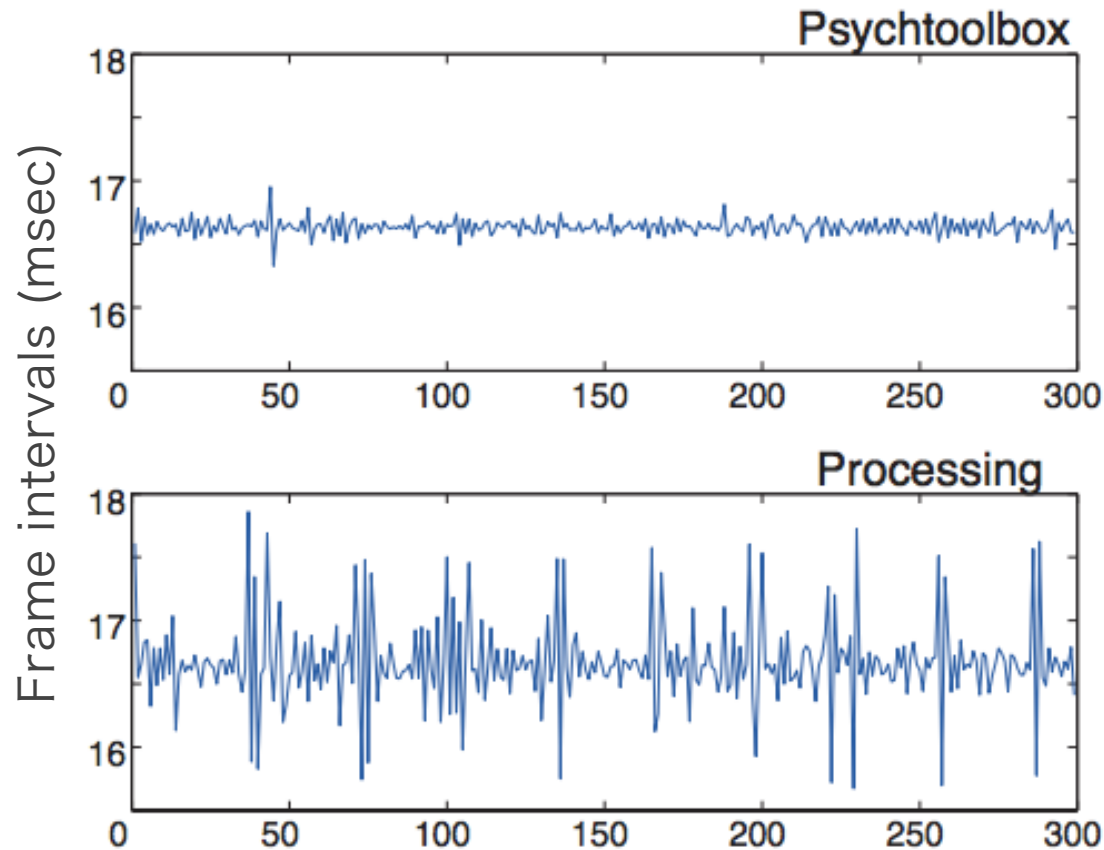
- MATLAB等の他言語向けのライブラリは利用出来ないので、もし何らかのライブラリを用いて実験している場合にはそれをProcessing用に自前で実装する必要がある

その他

- EyeLinkやPhantom等のライブラリは無いので利用は難しい
- web上やタブレット上での実行は動作速度や時間制御に難がある

時間精度の問題

60fpsで全画面への描画を300回繰り返した時のフレーム間隔の変動



平均が16.67ms, 標準偏差が0.41msと設定に近い値になっているものの, 範囲は15.7msから17.7msとフレームによっては1ms 程度のずれが生じてしまう.

図は 伊丸岡 (2010) 心理実験教育用プログラミング言語としての Processing の利用可能性.
Technical Report on Attention and Cognition (2010) No.12 より引用

他の言語との比較

環境の導入しやすさ

- 優れる。CやPython等は実行環境を整えるのは多少面倒だが、Processingはダウンロードしてすぐ使える

学習の容易さ

- CやJavaを使えるなら、1週間で使えるようになるはず

開発環境 (IDE)

- シンプルで使いやすいが、多機能とは言えない (入力補完機能とかは無い)
- 必要があればEclipse (Javaの開発環境) を使うことはできる

ライブラリとユーザーコミュニティ

- ライブラリは豊富 (Processingライブラリ + Javaライブラリも導入可能)
- 活発なコミュニティはあるが、心理学者のユーザはレア。

カバーできる実験の範囲

カバーできる実験の範囲

- 原理的には、どんな実験でも作成可能と思われる
- 画像・映像・テキスト・音声の提示やファイルデータの入出力、マウスやキー反応の取得等、必要な関数は全て揃っている
- 実験心理学分野の多くの知覚・認知課題は特に問題なく作成できるはず

備考

- 他言語で、何か特殊なライブラリを用いて実験を行なっている場合は、それをProcessingに自前で実装する必要があるので面倒が生じる
- データ解析、統計解析やグラフ化にはあまり向かない(その辺はRで)

Processingの有望性

実際の所、Processingをこれから学ぶだけの価値はあるのか

- 状況により一概には言えないが…(プログラミングの習熟度、ニーズ、etc.)
- 使えるようになれば何かと便利 (KinectやAR等の簡単実装等)

将来性

- 実験用言語のフリーウェア化の流れ
- Python、C++あたりが有望？
- Processingは、心理用ライブラリは無く、主流にはなり得ないかもしれない

学習リソース

公式サイト

- Tutorial, Reference, Exampleが充実しており、ここを見るだけで使い方はほぼ把握できる

書籍

『Built with Processing デザイン/アートのためのプログラミング入門』
その他和書、洋書多数

Webサイト

http://yoppa.org/cuc_prog12

http://yoppa.org/geidai_grapha11

ありがとうございました

本日のスライド及び関連資料は私のwebサイトに掲載します

<http://hiroyukitsuda.com/>