

Graphics with Processing



2016-01 Processing

<http://vilab.org>

塩澤秀和

1.1 Processing

Processingとは？

□ 特徴

- グラフィックに適した手軽な言語（C/C++でCGをやるより簡単）
- 芸術家などコンピュータの専門家でない人に配慮
- 文法はJavaとほとんど同じ（実体はJava+“便利な機能”）

□ Webサイト

- <http://processing.org>
- Exhibition(展示室)は必見

□ 今後役に立つの？

- OpenGL, DirectX (C/C++)でも考え方は同じ
- Javaのライブラリとして使える

参考資料

□ インストール方法

- 1.4 演習課題の手順に従うこと

□ サンプルプログラム集

- File → Examples

□ マニュアル

- Help → Reference
- コンピュータの世界, 最新情報は英語です。英語を読みましょう。

□ 参考文献

- 『Processingをはじめよう』, Casey Reas, Ben Fry著, オライリー
- 『Nature of Code: Processingで始める自然現象のシミュレーション』, ダニエル・シフマン著, ボーンデジタル
- 最近は, その他多数出ています
Amazon等で「Processing」で検索

1.2 簡単なプログラム

プログラムの基本構造

□ サンプルプログラム

- Examples → Basics → Structure → SetupDraw
- コメントはJavaやC++と同じ

□ void setup()

- 初期設定メソッド(関数)
- プログラム開始時に、自動的に1回だけ実行される

□ void draw()

- 描画メソッド(関数)
- 画面上で描画をする必要があるときに自動的に実行される
- アニメーション(loop)の場合、繰り返し何回も実行される
- 他にも再描画が必要なとき実行

基本的なsetup処理

□ size(幅, 高さ)

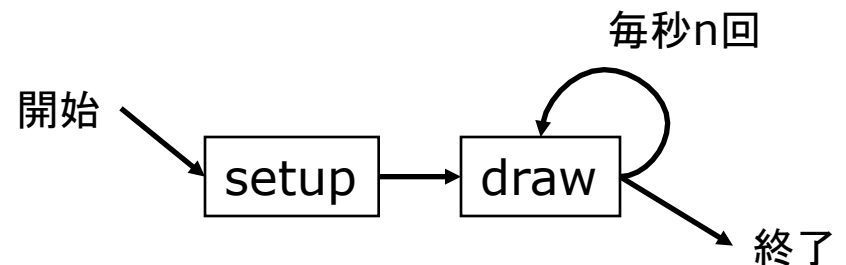
- ウィンドウを開く
- setupの最初に必要

□ frameRate(回数)

- 1秒間あたりの描画コマ数
- アニメーション(動画)に利用
- 省略した場合は毎秒60コマ

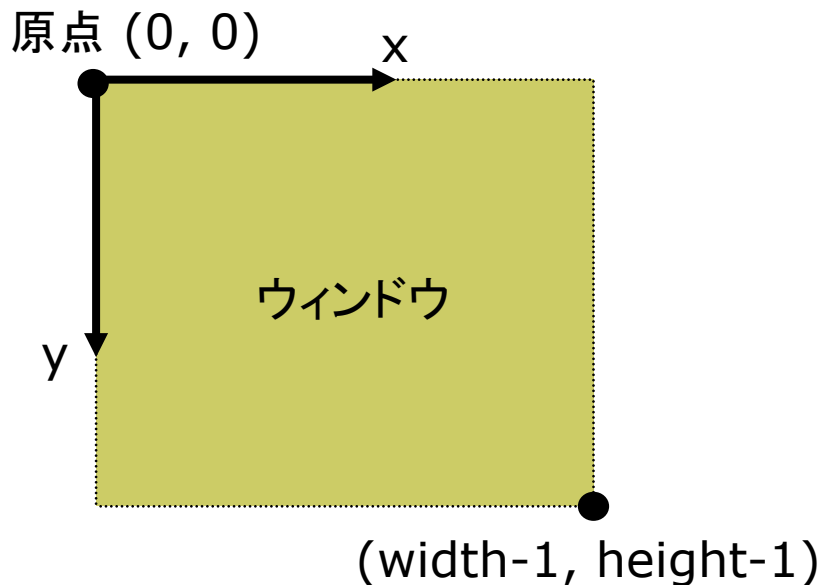
□ noLoop()

- アニメーションを無効化する
- loop()で解除できる



1.3 基本的な描画

Processingの座標系



色の数値表現

- 白黒(グレー)
 - 0~255の整数
- カラー
 - 0~255の整数 × 3個(RGB)
 - 例: `stroke(r, g, b)`

描画の準備

- `background(色)`
 - 背景色でウィンドウを塗りつぶす
 - 通常, `draw`の最初にやる
- `stroke(色)`
 - 線の色を指定する
- `strokeWeight(太さ)`
 - 線の太さを指定する

基本図形

- `point(x座標, y座標)`
 - 点を打つ
- `line(x1, y1, x2, y2)`
 - 直線を引く
- `rect(x, y, 幅, 高さ)`
 - 長方形(矩形)を描く

1.4 第1回課題

Processingのインストール

- ダウンロード
 - <https://processing.org/download/>
 - <http://www2.vilab.org/p5/processing-バージョン.zip>
- インストール
 - ZIPを適当な場所に展開する
(必ず**展開**するのを忘れずに)
 - processing.exe を実行
 - ユーザ名(フォルダ名)に日本語が含まれているとダメかも
- 3Dの動作確認
 - File → Examples → Basics → Form → Primitives3D
 - (>)ボタンでプログラム実行

課題

- サンプルを参考にして自分なりに簡単な図形を描いてみよう
 - SetupDrawなどを参考にする
 - プログラム(スケッチ)を作って、保存できることを確認する
 - ソースコードを印刷できることを確認する(画面コピーはダメ)
- 提出
 - プログラムと実行画面を印刷し、授業開始時に提出する
 - 番号・氏名と第何回の課題かを必ずレポートの最初に書くこと
 - 標準のソースコードの場所:
ドキュメント¥Processing
¥スケッチ名¥スケッチ名.pde
(OSのバージョンによって違う)