プログラミング The 心理学実験

PC を使う心理学実験が広く一般的になって，今はプログラミング環境に良い選択肢があります。最新の概要と特徴，デモやサンプルコード（http://odalab.org/pepe/）を通して，硬派で楽しい心理学の一面をお届けします。（小田浩一）

Processing を用いた心理実験プログラミング〈導入編〉

京都大学大学院人間・環境学研究科博士後期課程
津田裕之（つだ　ひろゆき）

Profile ── 津田裕之
京都大学工学部工業化学科卒業。日本学術振興会特別研究員（DC2）。専門は認知心理学（視覚記憶・空間認知）。

概　要

本稿では Processing（プロセシング）の初歩的な紹介を行います。Processing は Java をベースにしたプログラミング言語で，プログラミング初心者でも容易に習得可能な言語として開発されました。グラフィックや音を使ったプログラムが手軽に作成できます。Windows，Mac，Linux で実行可能であり，無料で利用することができます。

なぜ Processing なのか

Processing はプログラミングによるデザインやインタラクティブアートを作成するためのツールとして開発がスタートしました。そのため，動画像や音声の提示が短いコードで実現できるよう作られています。また，プログラミングに馴染みのない美術系の学生でも使いやすいように，文法は Java を簡略化した初心者にとってわかりやすいものとなっています。こうした特徴は，心理実験を作成するうえでもうってつけと言えます。すなわち，動画像や音声の提示が必要となることの多い心理実験プログラミングを素早く手軽に作成することができ，また学部生が卒業研究のためにプログラミングを学習する際などに比較的小さな学習コストで実験プログラムが作成できると考えられます。

導入方法と基本操作

Processing は公式サイト http://processing.org/からダウンロードできます。公式サイトの左側のメニューからダウンロードページへ進んでください。最初に寄付を募る画面が表示されますが，「No Donation」を選択してダウンロードページに進みましょう。Windows と Linux については 32bit と 64bit の二つの選択肢があるので，自分のパソコンに合ったほうをダウンロードしてください。複数のバージョンがありますが，基本的に画面最上部に表示されている最新バージョンを選びます。本稿では執筆時点での最新版である 2.2.1 をもとに説明します。ZIP ファイルを解凍したのち，Windows の場合は Processing.exe を，Mac の場合は Processing.app をダブルクリックして Processing を起動します。

Processing の起動に成功すると図 1 のようなウィンドウが表示されます。これは Processing のプログラムを書くためのエディタで，この画面でプログラムを書いて実行することができます。ウィンドウ上部にはいくつかのボタンが並んでおり，それぞれプログラムの実行や停止，またファイルの読み込みや保存に使用します。詳しくは図 1 内の説明を見てください。白紙の領域は現在編集中のプログラムファイルの内容で，ここにコードを書いていきます。ウィンドウ下部の黒い領域にはエラーメッセージなどが表示されます。

図 1　Processing のエディタ

簡単なプログラムの実行

それでは，簡単なプログラムを使って Processing を実際に動かしてみましょう。以下のコードをエディタに記述し，実行ボタンを押してください。

Example 1

```
size(600, 400);
line(0, 0, 600, 400);
ellipse(300, 200, 100, 100);
```

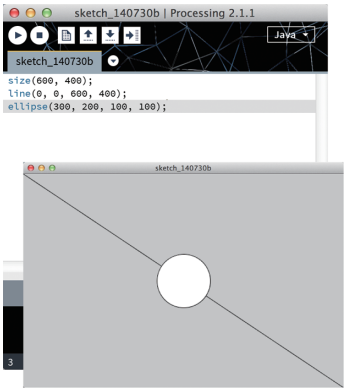

図 2　Example 1 の実行結果

図 2 のような結果が表示されたはずです。プログラムを終了するには実行ウィンドウ（図形の描画されたウィンドウ）を閉じるか，エディタのプログラム停止ボタンを押します。

このプログラムには三つの行があり，それぞれが Processing の持つ関数を記述したものです。size 関数はプログラムの実行ウィンドウの大きさを指定する関数で，line 関数と ellipse 関数はそれぞれ線分や円を描くための関数です。それぞれの関数の引数の意味は以下のとおりです。

```
line(始点の X 座標, 始点の Y 座標, 終点の X 座標, 終点の Y 座標);
ellipse(中心の X 座標, 中心の Y 座標, 円の横直径, 円の縦直径);
```

座標は，ウィンドウの左上を原点とする座標系で指定します。ウィンドウを開いて線分や円といった図形を描くというプログラムが，きわめて簡潔に記述できることがわかると思います。

初期化関数とメインループ関数

Processing によるプログラミングでは，初期化関数とメインループ関数という二つの関数をコードの中に含めてプログラミングすることが通例です。初期化関数とはプログラムの実行時に最初に一度だけ実行される関数のことで，関数名は setup です。メインループ関数はプログラムの実行中に繰り返し実行され続ける関数で，関数名は draw です。

実例を用いて説明しましょう。先の Example 1 のプログラムを，setup 関数と draw 関数を用いて書き直すと次のようになります。

Example 2

```
void setup( ){
  size(600, 400);
}

void draw( ){
  line(0, 0, 600, 400);
  ellipse(300, 200, 100, 100);
}
```

実行すると Example 1 と同様の結果になることを確認してください。このコードは setup 関数と draw 関数のそれぞれの関数を定義している形になっており，それぞれの関数の処理内容の部分に Example 1 の内容が書かれています。size 関数など，最初に一度だけ実行されれば良い命令を setup 関数内に記述します。

さて，この例の場合，プログラムを setup 関数と draw 関数にわざわざ分けて書く意義がわかりにくいと思います。また，setup 関数と draw 関数の挙動の違いも把握しづらいと思います。では次の例はどうでしょうか。

Example 3

```
void setup(){
  size(600, 400);
  background(255);
  frameRate(10);
}

void draw(){
  line(random(600), 0, random(600),
400 );
}
```

はじめに setup 関数内で背景色を白（255= 輝度の最大値）に，フレームレート（draw 関数内の処理が一秒間に繰り返される回数）を 10Hz に指定しています。draw 関数内には line 関数を記述し，始点と終点の X 座標をランダムとした線分を引いています（random 関数の引数は戻り値の最大値）。draw 関数内の処理は繰り返し実行され続けるため，プログラムの実行中に線分が次々と描き重ねられていきます。

もっと詳しく知るために

本稿では Processing のごく初歩的な導入を行いました。ごく限られた内容ではありますが言語の雰囲気は感じ取ってもらえたのではないかと思います。筆者のサイト http://hiroyukitsuda.com にてより網羅的な解説や実際的な心理実験プログラミングの方法について解説記事を用意しているので，本稿を通して Processing プログラミングに興味を持たれた方はぜひそちらに目を通していただければと思います。