

手軽にプログラミング

黎明期のパソコンには BASIC 言語のインタープリタが内蔵されているのが普通で、パソコンを買った人の多くが BASIC のプログラミングを楽しんでいました。ディスクやファイルのないパソコンでもプログラミング環境はしっかりと整備されていたわけですから、現在とは状況がかなり違っていたといえるでしょう。これらのパソコンでは、誰でも画面に絵を描くようなプログラムを簡単に作れましたが、最近のほとんどのパソコンにはプログラム開発環境が入っていません。別途、開発環境を導入しても、画面に絵を描くようなプログラムを作るにはいろいろな知識が必要になるため、プログラミングの敷居はずいぶん高くなってしまいました。

たとえば、Flash を使いたければ、Macromedia のソフトウェアを購入して ActionScript 言語の勉強をしなければなりません。Java でプログラムを書く場合は、Eclipse などの統合開発環境を使うか、コンパイラとエディタを組み合わせて利用するのが普通です。いずれにせよ、描画システムや Java の言語仕様を深く理解する必要があり、開発環境についての知識も欠かせないため、プログラミングに不慣れな人が使いこなすのは難しいでしょう。プログラミングが得意な人にしても、たとえばシミュレーションの結果を画面に表示するには、描画ライブラリやイベントの仕様などに関する知識も必要とされるため、それなりの苦勞を強いられます。

こうしてみると、現在のプログラミング・システムの多くは、プログラミングの専門家以外を排除する傾向が強いのではないかと思えてきます。計算機を使ってちょっとし

たシミュレーションをおこなったり、画面に絵を描いたりする人の数は、パソコンの黎明期より減っているかもしれませんが。

計算機の力を借りてダイナミックな映像表現を作りたいと考えるアーティストはたくさんいますが、プログラミングや開発環境に詳しい人はごく少数です。そんななかで、プログラミングについて最小限の知識しかなくても自由に映像表現が作りだせる「Processing」というシステムが注目を集めています¹。Processing は、MIT の Ben Fry 氏と Casey Reas 氏によって開発された Java ベースの言語/プログラム開発環境です。エディタとコンパイラ、ライブラリが 1 つの環境に統合されているため、インストールにも手間がかからず、簡単なプログラムを手軽に作成できます。

Java でプログラムを書くには、クラスやメソッドに関する知識が必要です。一方、Processing ではこれらがユーザーから隠蔽されているため、かつての BASIC 並みの手軽さでプログラムを作って実行することができます。Processing の現在のバージョンは Linux (x86)、Windows、Mac OS 上で動き、アートやホビー、教育などにひろく活用できます(執筆時の最新安定版は、2004 年 2 月 2 日にリリースされた 0068 ALPHA です)。

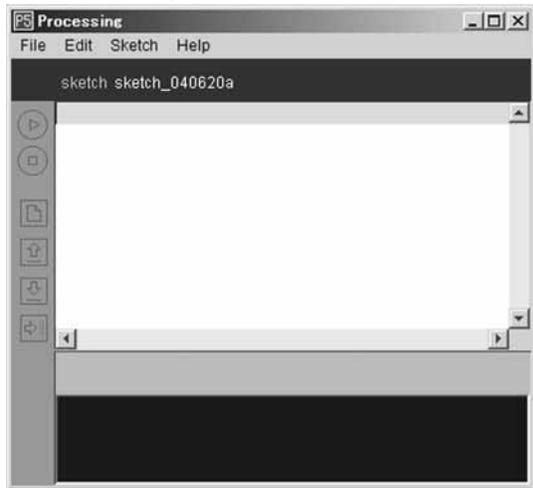
Processing の入手と実行

ダウンロード申込みページ²からメールを送ると、ダウン

1 <http://processing.org/> (Processing という単語はあまりにも一般的で、Google などの検索でみづかりにくいいため、“Proce55ing”や“P5”などと表現されることもあります)

2 <http://processing.org/download/>

図 1 Processing の起動



ロード先の URL を記したメールが返送されてきます。入手したパッケージを適当なディレクトリで展開して Processing を起動すると³、図 1 のようなウィンドウが表示されます。

ウィンドウの中央がプログラムを編集する領域で、下側の黒い部分にはメッセージが表示されます。ウィンドウの左側には、プログラムの実行やファイル操作のためのアイコンが並んでいます。

ここで図 2-a の簡単なプログラムを編集領域に入力し、プログラム実行を指示する横向きの三角形のアイコンをクリックします。すると、別のウィンドウが現れ、図 2-b のように直線が表示されます。

BASIC では、

```
10 LINE (10,10)-(90,90)
```

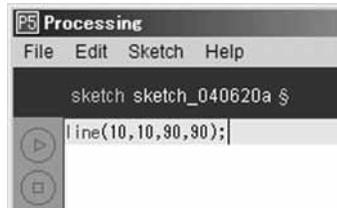
のようなプログラムで画面に直線を描けましたが、Processing でも同じくらいの手間で表示できることが分かります。もちろん、裏側では Java プログラムの作成、コンパイル、実行がおこなわれているわけですが、その様子はユーザーから隠されているため、ボタンを押すとすぐ実行されるようにみえます。

1 フレームごとに実行される "loop" メソッドを定義すると、一定のフレームレートでそのメソッドが実行されます。また、マウスの座標は mouseX と mouseY の 2 つ

³ 起動方法はプラットフォームごとに異なるので、展開したディレクトリの下にある readme.txt などのファイルを参照してください。

図 2 直線を描くプログラム

(a) プログラムを入力



(b) 実行結果

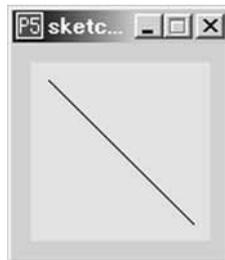
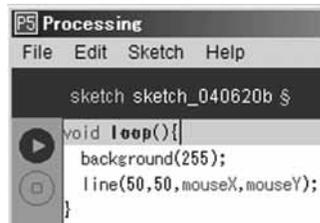


図 3 直線を描画する関数

(a) プログラム



(b) 実行結果



の変数に自動的に格納されるため、図 3-a のように書けば、"(50,50)" の点からマウス位置に向かって線を描くプログラムが作れます(図 3-b)。

通常の Java でこのような処理をおこなうプログラムを作るうとすると、図 4 に示したように、クラスと main メソッドの定義、マウスリスナーの実装や登録が必要になってしまいます。

これは汎用言語としては正しい方針といえますが、Java

脚注追加

図4 線描画プログラム(通常の Java 版)

```
import java.awt.*;
import java.awt.geom.*;
import java.awt.event.*;
import javax.swing.*;

class LineTestFrame extends JFrame{
    double x,y;
    LineTestFrame(){
        LineTestMouseListener ml = new LineTestMouseListener();
        addMouseListener(ml);
    }
    class LineTestMouseListener implements MouseMotionListener {
        public void mouseDragged (MouseEvent me) {
        }
        public void mouseMoved (MouseEvent me) {
            x = (double)(me.getX());
            y = (double)(me.getY());
            repaint();
        }
    }
    public void paint(Graphics g) {
        Graphics2D g2 = (Graphics2D) g;
        g2.setColor(Color.white);
        g2.fill(new Rectangle2D.Double(0,0,200,200));
        g2.setColor(Color.black);
        g2.draw(new Line2D.Double(100,100,x,y));
    }
}

public class LineTest {
    public static void main(String[] args){
        LineTestFrame test = new LineTestFrame();
        test.addWindowListener(new WindowAdapter(){
            public void windowClosing(WindowEvent e){System.exit(0);}
        });
        test.setBounds(0,0,200,200);
        test.setVisible(true);
    }
}
```

プログラミングの敷居を高くしている点は否めません。ちょっと線を描くといった単純なプログラムであっても、クラスやメソッドをはじめとするオブジェクト指向の基本概念やイベントモデルなど、たくさんのことを勉強しなければならないからです。Processing では、この程度の処理であれば、プログラミングに関する一般的な知識がほとんどなくてもすぐに実現できます。

Processing で作成したプログラムは、File メニューから“Export to Web”を選ぶだけで、アプレットとして Web で公開できるようになります。図3のプログラムの場合、図5のページが自動的に生成されます。

このページには、“Built with Processing”という文字列が付加されます。したがって、この文字列をキーに

図5 Web ページで公開

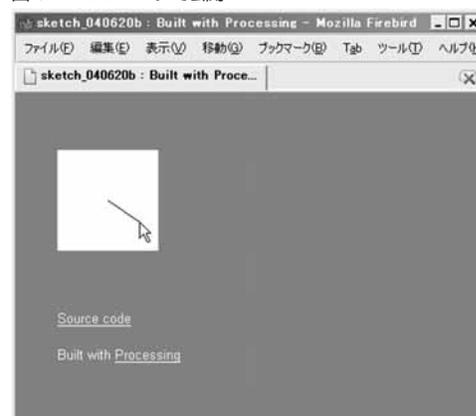
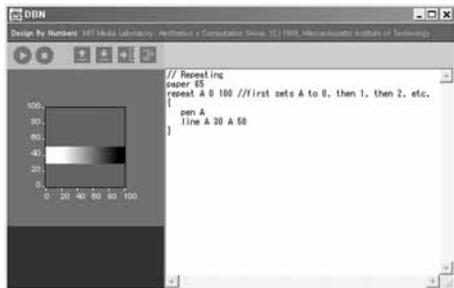


図 6 Design by Numbers



Google などで検索すると、Processing で作られた Web ページを簡単にみつけることができます。

Processing の歴史

さきほど述べたように、Processing は簡単なプログラミングで映像表現を可能にすることを目的に開発されたものですが、一般的なプログラミングの教育にも有用でしょう。

プログラミング教育のための簡単な言語としては、旧いところでは LOGO が有名ですし、新しいものでは MIT の John Maeda 氏が作った DBN (Design by Numbers)⁴ というシステムがあります。Processing は DBN の後継システムといってもよく、その外観も DBN (図 6) とよく似ています。

Processing の機能

Processing は Java に皮をかぶせて使いやすくしたものですから、int や float のような Java の基本データ型や for や while などの基本制御構造が使えます。また、基本ライブラリとして直線や多角形などの 2D/3D 描画、文字やフォントの操作、画像データの処理、ファイル操作をおこなう関数のほか、各種の数学関数も用意されています。これら以外に、拡張ライブラリとしてシリアル制御、オーディオ/ビデオカメラ用のライブラリなどもあります⁵。

簡単なプログラムの例

さきほどはごくごく簡単なプログラムを紹介しましたが、もうすこし複雑な例をみてみましょう。

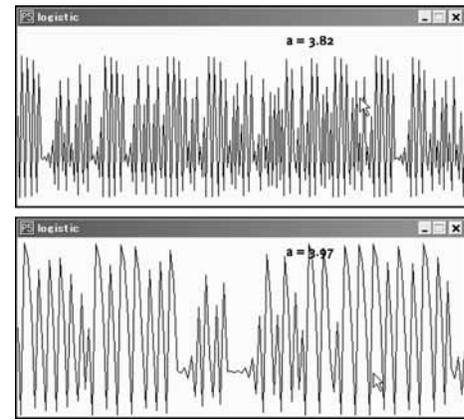
4 <http://dbn.media.mit.edu/>
 5 近藤哲也氏が、Processing の仕様の日本語訳を公開しています (http://stage.itp.tsoa.nyu.edu/~tk403/proce55ing_reference.jp/)。

図 7 カオス計算プログラム

```
void setup(){
  size(500,200);
  BFont fontA = loadFont("Meta-Bold.vlw.gz");
  textFont(fontA, 20);
  fill(0);
}

void loop(){
  float a = mouseX / 100.0;
  float w = mouseY / 40.0 + 0.2;
  float val = 0.5;
  float newval;
  background(250);
  text("a = "+a,300,20);
  for(int i=0;i*w < 500.0;i++){
    newval = a * val * (1.0-val);
    line(i*w,val*200,(i+1)*w,newval*200);
    val = newval;
  }
}
```

実行結果



マウス操作をおこなうプログラム

図 7 は、 $v_{n+1} = a * v_n * (1 - v_n)$ という計算の繰返しによって出現するカオスパターンを表示するプログラムです。マウスの位置に応じて a の値と表示パラメータを変えることができます。setup() は、プログラム実行時に 1 回だけ呼ばれるメソッドです。

フォントの置き場所はなんらかの変数で指定するのでしょうか？

3D 表示プログラム

図 8 は、回転する正四面体を表示するプログラムです。Processing では直線、ベジエ曲線、楕円、矩形などを描画するメソッドが用意されていますが、この例のように beginShape() と endShape() のあいだで頂点を順番に指定して描画をおこなうこともできます。この例では、begin-

図 8 回転する正四面体

```

float rot=0;

void setup() {
  size(300,300);
  framerate(30);
  noStroke();
}

void loop(){
  background(240);
  translate(150,150,100);
  rot += PI/90;
  rotateX(1.0);
  rotateZ(rot);
  beginShape(TRIANGLES);
  fill(255,255,255); vertex(40,40,40);
  fill(255,0,0); vertex(40,-40,-40);
  fill(0,255,0); vertex(-40,40,-40);

  fill(255,255,255); vertex(40,40,40);
  fill(255,0,0); vertex(40,-40,-40);
  fill(0,0,255); vertex(-40,-40,40);

  fill(255,255,255); vertex(40,40,40);
  fill(0,255,0); vertex(-40,40,-40);
  fill(0,0,255); vertex(-40,-40,40);

  fill(255,0,0); vertex(40,-40,-40);
  fill(0,255,0); vertex(-40,40,-40);
  fill(0,0,255); vertex(-40,-40,40);
  endShape();
}

```

実行結果



Shape の引数として "TRIANGLES" を指定しているので、色と座標を指定した頂点を 3 点ずつ組にした三角形が 3 次元空間に描画されます。

カメラの処理

図 9 は、PC に接続されたビデオカメラの映像をマウス位置に表示するプログラムです。カメラ内蔵の PC や USB カメラはかなり普及していますが、カメラを利用する

図 9 カメラ処理プログラム

```

void setup() {
  size(320, 240);
  beginVideo(320, 240, 24);
}

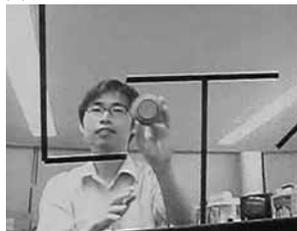
void loop() {
  image(video,mouseX,mouseY);
}

```

実行例



図 10 イライラボール



ためのライブラリは複雑なものが多いため、カメラを制御するプログラムを作った経験のある人は少ないのではないのでしょうか。

しかし、Processing を利用すると、ビデオカメラの映像を扱うプログラムが簡単に作れます。

カメラで撮影した映像にフィルタをかけたり画像認識をおこないたい場合は、Myron⁶ というライブラリが使えます。

UNIX でも使えるのでしょうか？

慶應大学の神原啓介氏は、Processing のビデオ画像処理機能を利用したイライラボール⁷ というゲームを公開しています。

Processing によるアート作品

Processing の Web ページでは、Processing で作られた素晴らしい作品がたくさん公開されています。⁸

図 11 は、Mario Klingemann 氏の "Burning Liquid

6 <http://webcamxtra.sourceforge.net/>

7 <http://sappari.org/iraira.html>

8 <http://processing.org/exhibition/>

図 11 Burning
Liquid Sky

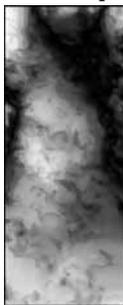


図 12 NudememGarden



Sky”という作品で、炎のような画像がリアルタイムに動くものです。これは、400行弱の Processing プログラムで実現されています。

また、図 12 は Francis Lam と Henry Chu の両氏による “NudememGarden” という作品で、画面の上から裸の人間が落ちてきてビデオカメラで撮った映像に含まれる黒っぽい部分にひっかかったり歩いたりするようになっています。

Processing の拡張

さきほど触れた Myron のようなものがあることから分かるように、Processing ではライブラリの拡張も可能です。また、Processing でメニューやスライドなどの GUI 部品を使えるようにするツールキットも公開されています⁹。

Processing の限界

Processing は導入もプログラミングもいたって簡単ですが、必要最低限の機能は備えているので、手軽に使えるプログラミング環境としてたいへん優れています。BASIC では再帰すら使えませんでした。Processing は Java をベースとしているため、本格的なオブジェクト指向的プログラミングも可能です。

その反面、単純化に重点が置かれているため、通常の Java にくらべると汎用性に劣り、あまり複雑なことはできません。

映像表現をおこなう環境として考えた場合、次のような限界が問題になることもあるでしょう。

⁹ <http://www.processing.co.il/toolkit/controller/>

- Java2 に対応していない
アプレットとして Web で公開できるようにするため、Java2 の機能は使えません。
- 漢字が使えない
現時点では、漢字などの 2 バイト文字を表示したい場合は、ビットマップ画像などを利用するしかないようです。
- 3 次元表示の視点を変更することができない

もうすこし説明を

おわりに

Java のような言語を使いこなすには、やはり言語仕様や環境についての相応の知識が必要になります。Java が登場したとき、往年の BASIC のような存在になるかもしれないと期待したのですが、残念ながら、プログラミングが趣味という人や職業的なプログラマー以外にはほとんど使われていないのが実情です。また、動作が重いという印象があるためか、期待されたほどには Web 上でアプレットとして活用されていることも少ないようです。

Processing の場合は、プログラミングに関する知識がほとんどなくても、ちょっとした絵を描いたり計算したりすることができます。結果を Web ページ上で公開するのも簡単ですから、Web ページにちょっと動きをつけるといった用途でも十分に使えるでしょう。アプレットとして公開しても、通常はそれほど重いと感ずることはありません。今後、日曜プログラマーや学生、デザイナーなどにひろく活用されることを期待しています。

(ますい・としゆき 産業技術総合研究所)