



Processing+JRuby、

ちょっとおさわり

2008年8月

村部@ランカードコム





自己紹介

- (有) ランカードコムの村部淳也です。
- ランカードは売ってません。たぶん。
<http://www.lancard.com> をみて頂けると幸い。
- 執拗に猫ネタをやっていたころもありました。
- 犬を飼っています。





テーマ (おもて)

- 以前「JRubyからJavaのクラス使ってみたらおもしろかったので、にゃんこ」というネタをやりました。
- そのときはJRubyからJavaの標準クラスライブラリだけを使って画像ファイルを変換してみたり、ビューアーを作ったりしました。
- 今回は標準じゃないクラスを使ってみようってことだ。





ところでProcessing

- 本当はProcessing。電子アートとビジュアルデザインのためのプログラミング言語で、Javaを単純化してグラフィック機能を中心にしたモノダキウです。
- Processingだと一般的すぎで検索に難ありなのと、以前、processing.orgというドメインが用いられていたという経緯でprocessingという表記も多かったのでマネしてみました。今の公式サイトは、<http://www.processing.org/>





なぜProcessing?

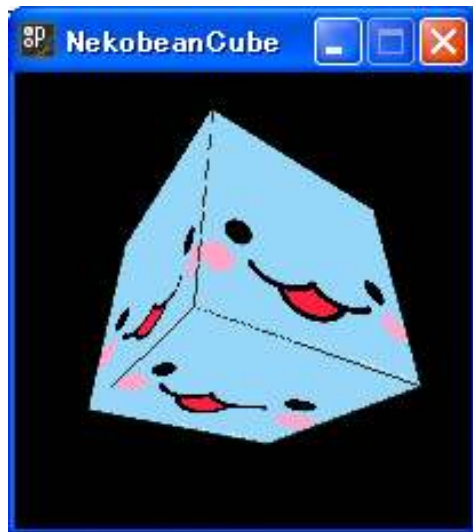
- JRubyのコアデベロッパ、Charles Oliver Nutterさんがデモしてるの見て、おもしろきうだったの。
- コアAPI (processing.core.PApplet) は、当然、標準クラスライブラリではない。けど動くのはわかってる。
- もともとアートやビジュアルデザインのための言語だし、たのしいデモができるはず～





Processingで"あきぼう"

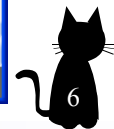
- Examples動かしたり、ちょっといじってみたりするだけでも、かなりたのしい。



```
SP NekobeanCube | Processing 0135 Beta
File Edit Sketch Tools Help
[Run] Run
NekobeanCube $
* by Dave Bollinger.
*
* Drag mouse to rotate cube. Demonstrates use of u/v coords in
* vertex() and effect on texture(). The textures get distorted using
* the P3D renderer as you can see, but they look great using OpenGL.
*/

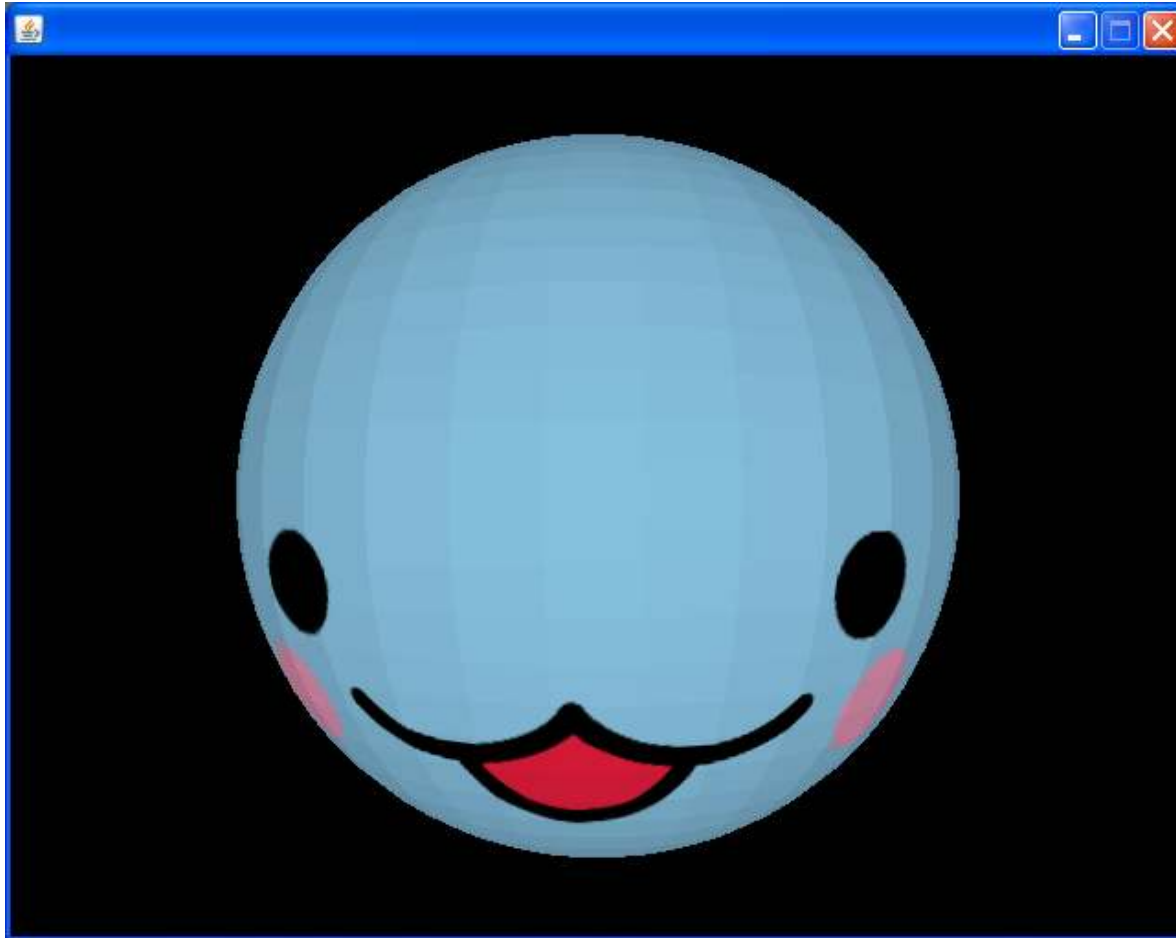
PImage tex;
float rotx = PI/4;
float roty = PI/4;

void setup()
{
  size(200, 200, P3D);
  tex = loadImage("nekobean_face_600x600.png");
  textureMode(NORMALIZED);
  fill(255);
  stroke(color(44,48,32));
}
```





ねこび〜んぽ〜る





ビデオで顔検出

- ビデオ入力して顔検出。カッコイイヤン。やりたいヤン。というわけで、壮大に横道にそれる。
- Bryan Chungさんという方がOpenCV (<http://sourceforge.net/projects/opencvlibrary/>) というフリーの画像処理ライブラリを使用して Processing用に顔検出ライブラリを作成されています (http://www.bryanchung.net/?page_id=251)。これを使ってみる。





あっさり顔検出

- Face Detection Library for Processing (PC)のページを参考にすればOK。
OpenCVに物体検出のサンプルとして付属している顔検出用のデータファイルをProcessingから読めるように配置するのを忘れずに。
- 仕組みについてはパス。

興味がおありでしたら gihyo.jp に『OpenCVで学ぶ画像認識』って特集があります。
<http://gihyo.jp/dev/feature/01/opencv>



検出した顔に定番のアシをかぶせてみたの図。





JRubyがでてきてません

- 実は本題に入っていないという噂。
- さて、JRubyは1.0.0RC1からProcessingをあつかえるようになったようです。去年の5月、ちょうどその時期にチャレンジされた偉大な先人がいらっしゃいます。

<http://d.hatena.ne.jp/dewdrop/20070506/p5r>

<http://d.hatena.ne.jp/dewdrop/20070513/p5r>

そして、月日はめぐり





なぜか靴箱 (shoebox) に

- Ruby-ProcessingというJRubyでProcessingを
あつかうのに便利なフレームワークががが。

<http://the-shoebox.org/apps/44>

今回はコイツをつかいます。

- JRubyでProcessingをあつかう環境としては
Ichitaro Masudaさんのaction-codingなども
あります。ライブコーディングのスクリーン
キャストは必見。

<http://code.google.com/p/action-coding/>





Ruby-Processingの基礎

- 基本は、展開したディレクトリのトップ (READMEがあるディレクトリ) で作業。
- `core.jar`が同梱されておりProcessingのコアAPIを使える。
- `script`ディレクトリには便利なコマンド。
- 実はJRubyをセットアップしなくてもOK。
CRubyが使える環境なら

```
ruby script/open.rb samples/jwishy.rb
```

で同梱されている `jruby-complete.jar` で実行される。詳細は `open.rb` 参照のこと。





テンプレートをジェネする

- せっかくだからジェネレートするぜ。

```
jruby script/generate my/generate_test 400 400
```

- myディレクトリにgenerate_test.rbできた。

```
1 require 'ruby-processing' ↓
2 ↓
3 class GenerateTest < Processing::App ↓
4 ↓
5   def setup ↓
6     ↓
7   end ↓
8   ↓
9   def draw ↓
10  ↓
11  end ↓
12  ↓
13 end ↓
14 ↓
15 GenerateTest.new :title => "Generate Test", :width => 400, :height => 400
```

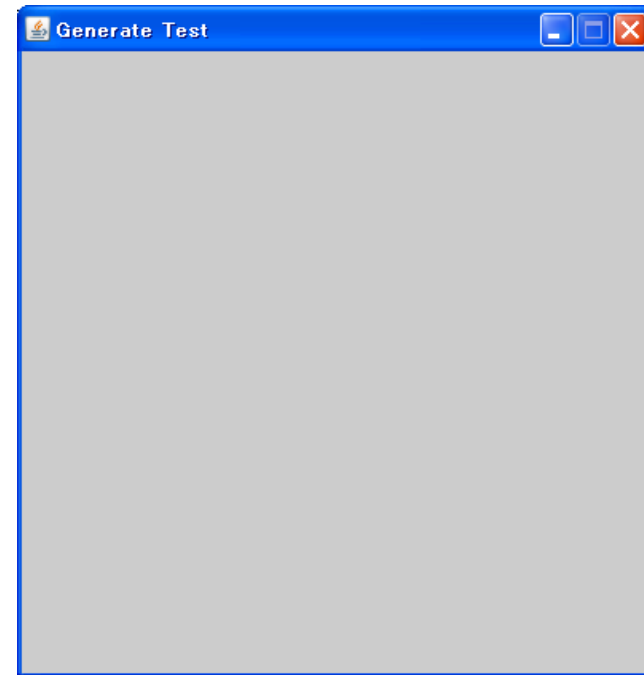
- これが基礎構成。





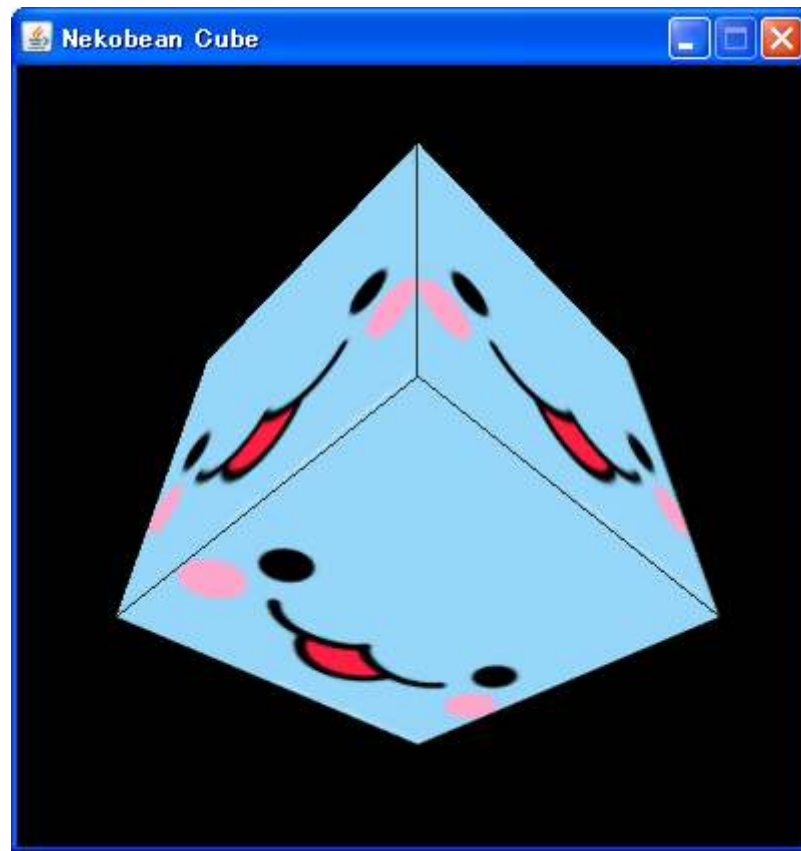
あとはスキにいじるがイイ

- 動かしてみる。
`jruby my/generate_test.rb`
- 灰色の世界。ま、空だし。
- あとはサンプルとかみながら
らいじるといい。
- 最初からサンプルをコピー
していじれば"いいじゃん、
という説もある。
- ここでワンポイント。ProcessingのAPIをう
まく呼べないときはProcessing::Appのクラ
スメソッドとして呼んでみる。





ねこび〜んきゅ〜び





ruby-processing.rbから学ぶ

- Javdのクラスへの扱い方の一例として。
- Processingモジュールで"Processingのcore.jarをrequireして、
include_package "processing.core "
でまとめて取り込んでいる。
- Rubyのprocessing::AppクラスはJavdのprocessing.core.PAppletクラスを継承。
- include_classで特定のJavdのクラスを取り込んでいる。





なぜ JRuby?

- Rubyで書きたいんですもの。
- Rubyのライブラリも使える。

問題は?

- Javaのクラスをあつかう場合、 JRubyは今のところメソッドのオーバーロードをうまくあつかうことができない。Javaとの連携強化を進めるとCharlesさんは話してたので期待しましょう。





ねこび〜んたいま〜

- 発表時間管理用に JRuby+Processing で書いてみました。
- 表示とイベント周りはProcessingのAPI、その他はRubyのライブラリ使ってます。
- コンセプト



時間確認、焦る。ねこび〜ん、なごむ。OK!





jirvbでコントロール

- RubyっぽいネタってことでJRubyに移植した顔検出プログラムをjirvbからリアルタイムでコントロールしてみる。

```
Welcome to the JRuby IRB Console [1.1.3]
```

```
irb(main):001:0> load 'my/face_detect.rb'  
=> true  
irb(main):002:0> @app.mask='asuka_face.png'  
=> ~asuka_face.png~  
irb(main):003:0> @app.mask='laughingman.gif'  
=> ~laughingman.gif~
```





テーマ（おもて）終了

- おつかれさまでした！
- たのしんでいただけでしたか？
- それでは最後に





テーマ (うら)



- NetBeans日本語コミュニティの公式マスコットであらせられる『ねこび〜ん』のサイト、<http://nekobean.net/>が7月末にオープンし、大変喜ばしいので個人的にねこび〜ん祭。





nekobedn.net



<http://nekobean.net/>





ja: ねこび〜ん



<http://ja.netbeans.org/nekobean/>





ねこび〜んWiki



<http://wiki.netbeans.org/NekoBean>





ほんとおわり

- おつかれさまでした！



- ねこび〜んサイコー！



- またね！

