

Graphics with Processing



2006-2 基本的な描画

<http://vilab.org>

塩澤秀和

2.1 変数と制御構造

変数の型 (Javaと同じ)

- int, float, byte
 - 数値
- boolean
 - 真偽値 (Yes/No)
 - 定数: true (真), false (偽)
- char
 - 文字 `char ch = 'あ'`
- String
 - 文字列
 - `String str = "あいうえお"`
- + 演算子
 - 文字列同士や文字列に文字を“足す”と連結できる

制御構造

- if-else
- switch-case
 - 条件分岐
- for, while
 - 繰り返し
 - do-whileはない

条件式でよく使う演算子

- 関係演算子
 - `==, !=, <, >, >=, <=`
- 論理演算子
 - `||` (OR), `&&` (AND), `!` (NOT)

2.2 基本図形

図形描画関数

- point, line, rect
 - 点, 直線, 長方形(既出)
- triangle(x1, y1, x2, y2, x3, y3)
 - 三角形
- quad(x1, y1, x2, y2, x3, y3, x4, y4)
 - 四角形
- ellipse(x, y, 幅, 高さ)
 - 楕円(円)
- arc(x, y, 幅, 高さ, 開始角, 終了角)
 - 弧(角度はラジアン)
 - 定数PI(π)が使える

描画色

- stroke(色)
 - 線の色を設定(既出)
 - noStroke()で境界線なし
 - strokeWeight(線の太さ)
- fill(色)
 - 塗りつぶし色を設定
 - noFill()で塗りつぶし解除

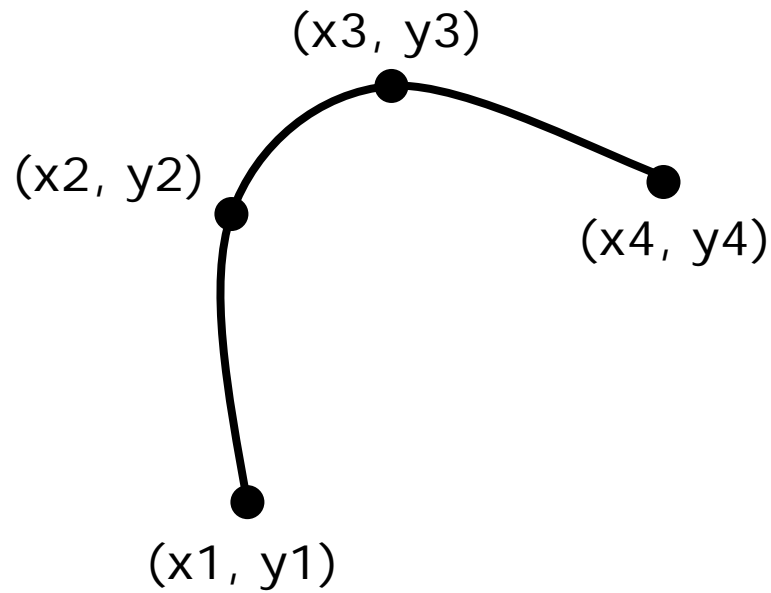
座標指定モード

- rectMode(モード)
- ellipseMode(モード)
 - 左上を指定: CORNER
 - 中心を指定: CENTER

2.3 曲線の描画

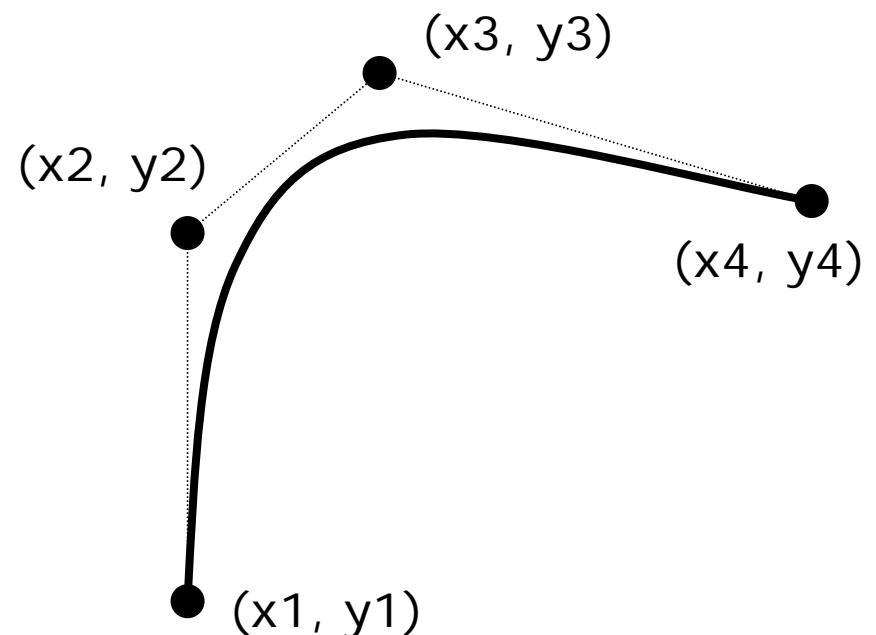
曲線 (Catmull-Rom 曲線)

- `curve(x1, y1, x2, y2, x3, y3, x4, y4)`



ベジエ曲線

- `bezier(x1, y1, x2, y2, x3, y3, x4, y4)`



- スプライン曲線

2.4 パラメトリック曲線

曲線の数式表現

- $y = f(x)$ 型
- $f(x, y) = 0$ 型
- $x = f(t), y = g(t)$ 型
 - パラメータ表現

パラメトリック曲線

- 数式の変数(パラメータ)を変化させて描ける曲線
- 少ない変数で滑らかな曲線
- 例

$$\begin{cases} x = 4t + 3 \\ y = \sin(2\pi t) \end{cases}$$

Catmull-Rom曲線

- 3次スプライン補間曲線

$$q(t) = \frac{1}{2} \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix} \begin{bmatrix} P_0 \\ P_1 \\ P_2 \\ P_3 \end{bmatrix}$$

Bezier曲線

- 2次ベジェ曲線

$$B(t) = (1-t)^2 P_0 + 2t(1-t)P_1 + t^2 P_2$$

- 3次ベジェ曲線

$$B(t) = (1-t)^3 P_0 + 3t(1-t)^2 P_1 + 3t^2(1-t)P_2 + t^3 P_3$$

2.5 演習課題

サンプルプログラム

- File → Sketchbook → Examples
- 文法
 - → Structure
- 曲線
 - → Form → CurvePrimitives

課題

- 基本図形を組み合わせて自由に“絵”を描きなさい
- 必ず曲線を利用すること
- 下記URLから提出
- <http://vilab.org/upload/cg-upload.html>

プログラムの基本構造

```
void setup() {  
    // 初期設定の例  
    size(200, 200);  
    background(0);  
    noLoop(); // ループの停止  
}
```

```
void draw() {  
    // 描画の例  
    stroke(0, 255, 0);  
    fill(0, 255, 0);  
    rect(50, 50, 50, 50);  
}
```