

Graphics with Processing



2006-6 複雑な図形の描画

<http://vilab.org>

塩澤秀和

6.1 頂点列による図形描画

描画モード(0115 Beta)

- beginShape(図形)
 - 図形描画の開始
 - 省略・POLYGON 多角形
 - POINTS 点
 - LINES 線1本1本
 - LINE_STRIP 線で結ぶ
 - LINE_LOOP 線で囲む
 - TRIANGLES 三角形
 - TRIANGLE_STRIP 連続三角形
 - TRIANGLE_FAN 最初の点を共有した三角形の連続
 - QUADS 四角形
 - QUAD_STRIP 連続四角形
- endShape()
 - 図形描画の終了

頂点の指定

- vertex(x, y)
 - 図形の頂点を加える
 - stroke(色) ⇒ グラデーション
- curveVertex(x, y)
 - 曲線で描画する(途中頂点)
 - bezierVertex(x1, y1, x2, y2, x3, y3)

例(四角形)

```
beginShape(QUADS);
  vertex(30, 20);
  vertex(30, 75);
  vertex(50, 75);
  vertex(50, 20);
endShape();
```

6.1' 頂点列による図形描画

描画モード(新バージョン)

- beginShape(図形)
 - 図形描画の開始
 - 図形: 省略⇒連続線か多角形, POINTS, LINES, TRIANGLES, TRIANGLE_FAN, TRIANGLE_STRIP, QUADS, QUAD_STRIP
- endShape()
 - 図形描画の終了
 - endShape(CLOSE): 最初と最後の点を結んで図形を閉じる (ただし, 塗りつぶさないとき)
- 塗りつぶし
 - fill()/noFill()で指定できる

頂点の指定

- vertex(x, y)
 - 図形の頂点を加える
 - stroke(色) ⇒ グラデーション
- curveVertex(x, y)
 - 曲線で描画する(途中頂点)
 - bezierVertex(x1, y1, x2, y2, x3, y3)

例(塗りつぶさない多角形)

```
noFill();
beginShape();
vertex(30, 20); vertex(30, 75);
vertex(50, 75); vertex(50, 20);
endShape(CLOSE);
```

6.2 対話入力処理

システム変数

- mousePressed
- mouseX, mouseY
 - 既出
- pmouseX, pmouseY
 - 前フレームでのマウス位置
- mouseButton
 - 押されているマウスボタン
 - LEFT, RIGHT, CENTER

- keyPressed
 - キーが押されていればtrue
- key
 - 押されている文字
- keyCode
 - 特殊キーのキーコード
 - 詳しくは, マニュアル参照

コールバック関数

- void mousePressed()
 - マウスボタンが押されたときに自動的に呼ばれる処理を設定
- void mouseReleased()
 - ボタンが離されたとき
- void mouseMoved()
 - マウスが動かされたとき(ただし, ボタンは押されていないとき)
- void mouseDragged()
 - ボタンが押されたまま, マウスが動かされたとき

- void keyPressed()
 - キーが押されたとき
- void keyReleased()
 - キーが離されたとき

6.3 ファイル入出力

ファイル入出力

- loadStrings("ファイル名")
 - ファイルから1行ごとに文字列として読み出して配列に格納
 - String lines[] = loadStrings("data.txt")
 - Sketch → Add File...で、あらかじめ、データフォルダにファイルをコピーしておくこと
- saveStrings("ファイル名", 行配列)
 - ファイルに文字列を保存
 - saveStrings("data.txt", lines)
- 典型的な処理パターン

```
for (int i = 0; i < lines.length; i++) {  
  // lines[i]の処理  
}
```

文字列処理(詳細はマニュアル参照)

- float(文字列)
- int(文字列)
 - 文字列を数値に変換
- str(数値)
 - 数値を文字列に変換
- hex(整数)
 - 整数(int, byte, char, color)を16進文字列に変換
- unhex(文字列)
 - 16進文字列を数値に変換
- trim(文字列)
 - 文字列から空白を除去
- join(文字列配列)
 - 文字列の連結
- split(文字列)
 - 文字列を空白で分割(joinの逆)

6.4 演習課題

課題

- マウスでクリックした位置に点を打ち、それを順に結ぶ“折れ線”を描くプログラムを作成しなさい
 - beginShapeを使うこと
 - 右のプログラムに必要な処理を加えるとよい
 - beginShapeを三角形や四角形にしたらどうなるか？
 - それもできた人は、描ける線の本数を複数にしたり、ファイルに保存できるようにしてみなさい

- プログラムが“汚い”人は、**Tools → Auto Format** で整形してから提出
 - 見やすさも採点対象！

```
int npos = 0;
int x[] = new int[100];
int y[] = new int[100];

void setup() {
    // 省略...
    noLoop(); // アニメーション停止
}

void draw() {
    background(0);
    for (int i = 0; i < npos; i++)
        ellipse(x[i], y[i], 10, 10);
}

void mousePressed() {
    x[npos] = mouseX;
    y[npos] = mouseY;
    npos++;
    redraw(); // 点が増えたら描画
}
```