

Graphics with Processing



2008-03 配列とアニメーション

<http://vilab.org>

塩澤秀和

3.1 配列とシステム変数

配列

- 初期値のある配列の作成
 - `int a[] = { 1, 2, 3 };`
- 空の配列の作成
 - `int a[] = new int[20];`
- 要素の参照
 - 長さ20なら, `a[0]~a[19]`
- 配列の長さ(要素数)の取得
 - `a.length`

グローバル変数

- 関数(`setup()`, `draw()`など)の外側で定義された変数
- すべての関数から参照できる
- 関数を抜けても値が保持される

システム変数

- `width`, `height`
 - ウィンドウのサイズ
- `mouseX`, `mouseY`
 - マウスのX座標とY座標
- `mousePressed`
 - ボタンが押されているか
 - `if (mousePressed) { ...`

ユーザ定義関数(メソッド)

- `setup()`, `draw()`以外に, 自由な関数をユーザが定義可能
- 定義の文法は, CやJavaと同じ(`return`で値を返せる)
- よく使う描画処理をまとめるなど

3.2 組み込み関数 (その他マニュアル参照)

型変換

- `int(値)`
 - 値をint型に変換
- `float(値)`
 - 値をfloat型に変換
- `str(値)`
 - 値をprint用文字列に変換

数学関数

- `sqrt(値)`
 - 平方根($\sqrt{\quad}$)
- `pow(x, y)`
 - xのy乗
- `dist(x1, y1, x2, y2)`
 - 2点間の距離
- `constrain(値, 最大, 最小)`
 - 値を範囲内に収める

三角関数

- `radians(deg), degrees(rad)`
 - 度 \leftrightarrow ラジアンの変換
- `sin(角度), cos(角度), ...`
- `atan2(x, y)`
 - x軸とベクトル(x, y)の成す角

時間関数

- `year(), month(), day()`
- `hour(), minute(), second()`

乱数関数

- `randomSeed(種)`
 - 乱数の準備(種は`millis()`など)
- `random(最大値)`
- `random(最小値, 最大値)`
 - 乱数の発生(float型)

3.3 アニメーション

アニメーション

- `frameRate(回数)`
 - 1秒あたりの描画(`draw`)回数
 - 停止・再開: `noLoop()`, `loop()`
- `frameCount`
 - `draw()`が呼ばれた回数
- `redraw()`
 - 強制的に再描画させる
- `millis()`
 - プログラム開始からのミリ秒
- パラパラ漫画の要領
 - 各図形の形や位置(座標等)をグローバル変数で保持
 - `draw()`の中で変数を変化させ、毎回形や位置をずらして描画

アニメーションの例

- サンプル Examples → Basics → Structure → Loop

```
...
int y = 100; // グローバル変数
void draw() {
    background(0); // 毎回消去
    ...
    y = y - 1; // 毎回位置変更
    ...
    line(0, y, width, y); // 描画
}
```
- Examples → Topics → Motion → Bounce
 - ボールが四方の壁でバウンド

3.4 演習課題

課題

- サンプルBounceを参考にして、4つ(以上)のボールがはね返るアニメーションを作成しなさい
- 条件1: ボールの座標や方向は配列に格納すること
- 条件2: ボールの最初の座標は乱数で決めること
- (条件ではないが) 各ボールの大きさや速さを変えると面白い

提出

- 〆切: 来週11時
- 提出ページ <http://vilab.org/upload/cg-upload.html>
 - 出席番号等を間違えないように

ヒント

- 条件1(グローバル変数で)

```
float xpos[] = new float[4];
float ypos[] = new float[4];
int xdirection[] = { 1, 1, 1, 1 };
int ydirection[] = { 1, 1, 1, 1 };
```
- 条件2(setup()の中で)

```
randomSeed(millis());
for (int i = 0; i < 4; i++) {
  xpos[i] = random(width);
  ypos[i] = random(height);
}
```
- ボールの描画(draw()の中で)

```
for (int i = 0; i < 4; i++) {
  ellipse(xpos[i]+size/2,
    ypos[i]+size/2, size, size);
}
```