

Graphics with Processing



2009-03 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

配列の作成

- 初期値のある配列の作成
 - `int a[] = { 1, 2, 3, 4, 5 };`
⇒ `a[0]=1~a[4]=5` ができる
- 空の配列の作成
 - `int a[] = new int[10];`
⇒ `a[0]~a[9]` で, 全要素が0

配列の使用

- 配列の添字
 - 添字(番号)は `0~(要素数-1)`
 - **【注意】** `new int [10]` で作成した配列に `a[10]` は存在しない!
- 配列の要素数
 - `a.length` で取得できる

グローバル変数

- `setup()`, `draw()`などの関数の外側で変数を定義すると...
 - 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

- 自動設定されるグローバル変数
- `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) { ...`

3.2 自作関数と組み込み関数

自作関数(メソッド)

- JavaやCと同様

```

戻り値の型 関数名(引数並び) {
    処理手順
    ...
    return 戻り値;
}

```

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(式, 最大, 最小)
 - 式の値を範囲内に収める

三角関数

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

時刻関数

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

乱数関数

- randomSeed(種)
 - 乱数の準備
 - 種は関数 millis() などを使う
- 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(p.162)

- パラパラマンガのように
 - draw()の中で毎回形や位置をずらして描画する
 - 各図形の形や位置(座標等)はグローバル変数で保持する
- 例
 - サンプルのSetupDrawより

```
int y = 100; // 図形の座標位置
...
void draw() {
  background(0); // 毎回消去
  ...
  y = y - 1; // 毎回位置変更
  ...
  line(0, y, width, y); // 描画
}
```

システム変数

- frameCount
 - draw()が呼ばれた回数
- frameRate
 - 現在の実際の毎秒コマ数

関連関数

- frameRate(回数)
 - 1秒あたりの描画(draw)回数
 - 停止・再開: noLoop(), loop()
- redraw()
 - 強制的に再描画させる(おもにアニメーションでないときに使う)
- millis()
 - プログラム開始からのミリ秒

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
 - サンプル Examples → Topics → Motion → Bounce
 - **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
 - ✕切: 来週11時
 - 提出ページ <http://vilab.org/upload/cg-upload.html>

ヒント

- **条件1**(グローバル変数で)

```
float xpos4[] = new float[4];
float ypos4[] = new float[4];
int xdirection4[] = { 1, 1, 1, 1 };
int ydirection4[] = { 1, 1, 1, 1 };
```
- **条件2**(`setup()`の中で)

```
randomSeed(millis());
for (int i = 0; i < 4; i++) {
    xpos4[i] = random(0, width);
    ypos4[i] = random(0, height);
}
```
- **ボールの描画**(`draw()`の中で)

```
for (int i = 0; i < 4; i++) {
    ellipse(xpos4[i]+size/2,
           ypos4[i]+size/2, size, size);
}
```