Graphics with Processing

2014-06 座標変換と同次座標

http://vilab.org

塩澤秀和

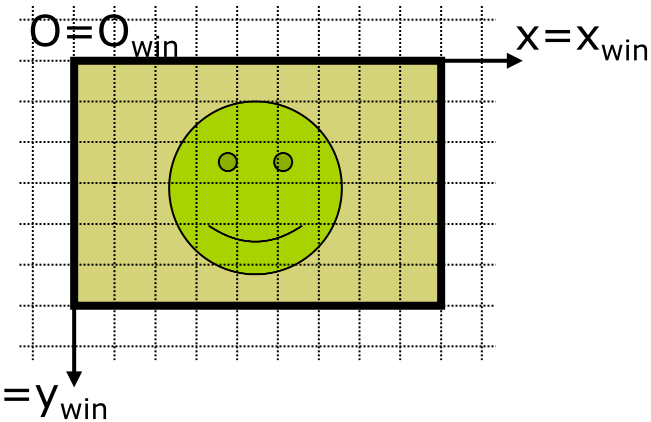
6.1 座標系

座標系の変換

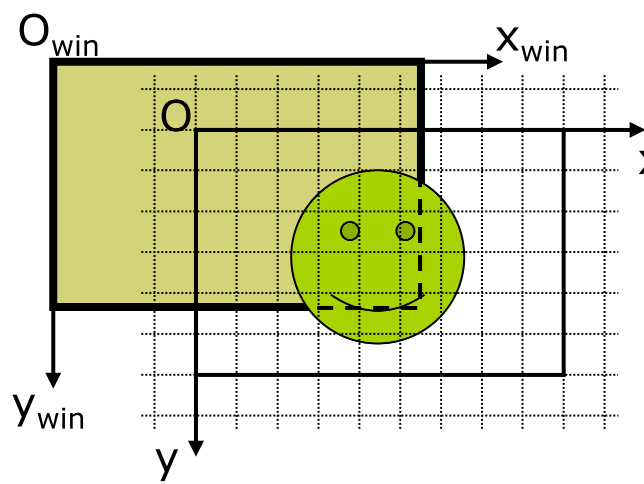
- 座標系＝目盛りのつけかた
  - 原点の位置
  - x軸とy軸の方向
  - x軸とy軸の目盛りの刻み
- 論理座標系
  - 描画命令で使う目盛り（座標系）をつけかえることができる
  - 論理座標系
    ⇒ 描画命令で使うxy座標
  - 画面座標系
    ⇒ ウィンドウでのピクセル位置
- 座標系の変換
  - 論理座標での描画命令
    → 画面座標でのピクセル設定
  - 数学的に計算して自動的に変換

画面座標系
(x_{win}, y_{win})

論理座標系
(x, y)



初期状態（画面座標系＝論理座標系）



描画用の原点をずらした例

6.2 座標変換(p.16)

座標変換と幾何変換

- 座標変換

$$\begin{pmatrix} x \\ y \end{pmatrix} \xrightarrow{f} \begin{pmatrix} x' \\ y' \end{pmatrix}$$

- 座標変換の合成
  - 論理座標系から，何回かの座標変換を経て，画面座標系へ

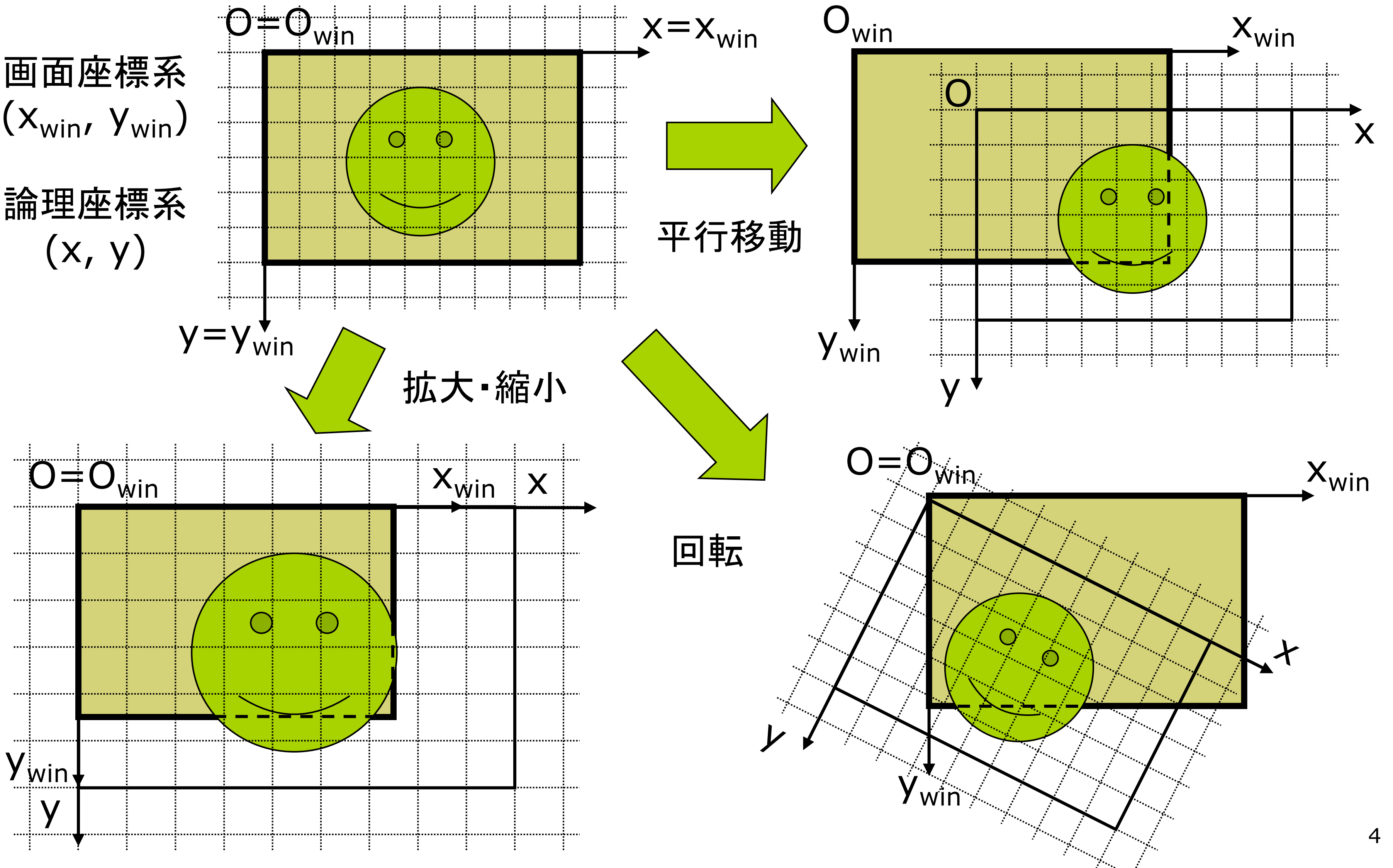
$$\begin{pmatrix} x \\ y \end{pmatrix} \to \begin{pmatrix} x' \\ y' \end{pmatrix} \to \dots \to \begin{pmatrix} x_{win} \\ y_{win} \end{pmatrix}$$

- 幾何変換(幾何学的変換)
  - 平行移動
  - 拡大・縮小
  - 回転

幾何変換関数

- translate(x_0, y_0)
  - 座標系を平行移動(原点を移動)
  - x軸方向に x_0 移動
  - y軸方向に y_0 移動
  - Processingではy軸は**下向き**
- scale(α, β)
  - 座標系を拡大または縮小
  - x軸方向(左右)に α 倍
  - y軸方向(上下)に β 倍
  - 原点が中心に全体が拡大
- rotate(θ)
  - 座標系を回転
  - 原点中心に θ 回転
  - Processingで+方向は**時計回り**

6.3 幾何変換の効果


画面座標系
(x_{win}, y_{win})
論理座標系
(x, y)
O=O$_{win}$
x=x$_{win}$
y=y$_{win}$
平行移動
O$_{win}$
x$_{win}$
O
x
y$_{win}$
y
拡大・縮小
O=O$_{win}$
x$_{win}$
x
y$_{win}$
y
回転
O=O$_{win}$
x$_{win}$
x
y
y$_{win}$

6.4 幾何変換の数学表現

数式による表現

- 平行移動

$$x' = x + x_0$$

$$y' = y + y_0$$

例）原点を画面の(10, 20) に移動し，座標 (5, 7) に点を打つと，画面では(15, 27) に表示

- 拡大・縮小

$$x' = \alpha\, x$$

$$y' = \beta\, y$$

例）目盛りを横2倍，縦3倍に拡大して，座標 (5, 3) に点を打つと，画面では(10, 9) に表示

- 回転

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

1次変換と座標の回転は高校では「数学C」の範囲

ベクトルと行列による表現

- 1次変換

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

なぜ平行移動は表現できないか？

  - 拡大・縮小と回転が表現可能

  ⇒ 各自，対応する行列を求めよ

- アフィン変換
  - すべての幾何変換を表現可能

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}$$

$$\Leftrightarrow \begin{cases} x' = ax + by + e \\ y' = cx + by + f \end{cases}$$

定数項を付加

6.5 同次座標表現(p.19)

同次座標表現

- 座標計算をしやすい形式

2次元直交座標　　2次元同次座標

$$(x,\ y) \Leftrightarrow (x,\ y,\ 1)$$

同じ座標

- 同次座標表現による変換行列

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

行列1つですべてのアフィン変換を表せる

同次座標表現による幾何変換

- 平行移動

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

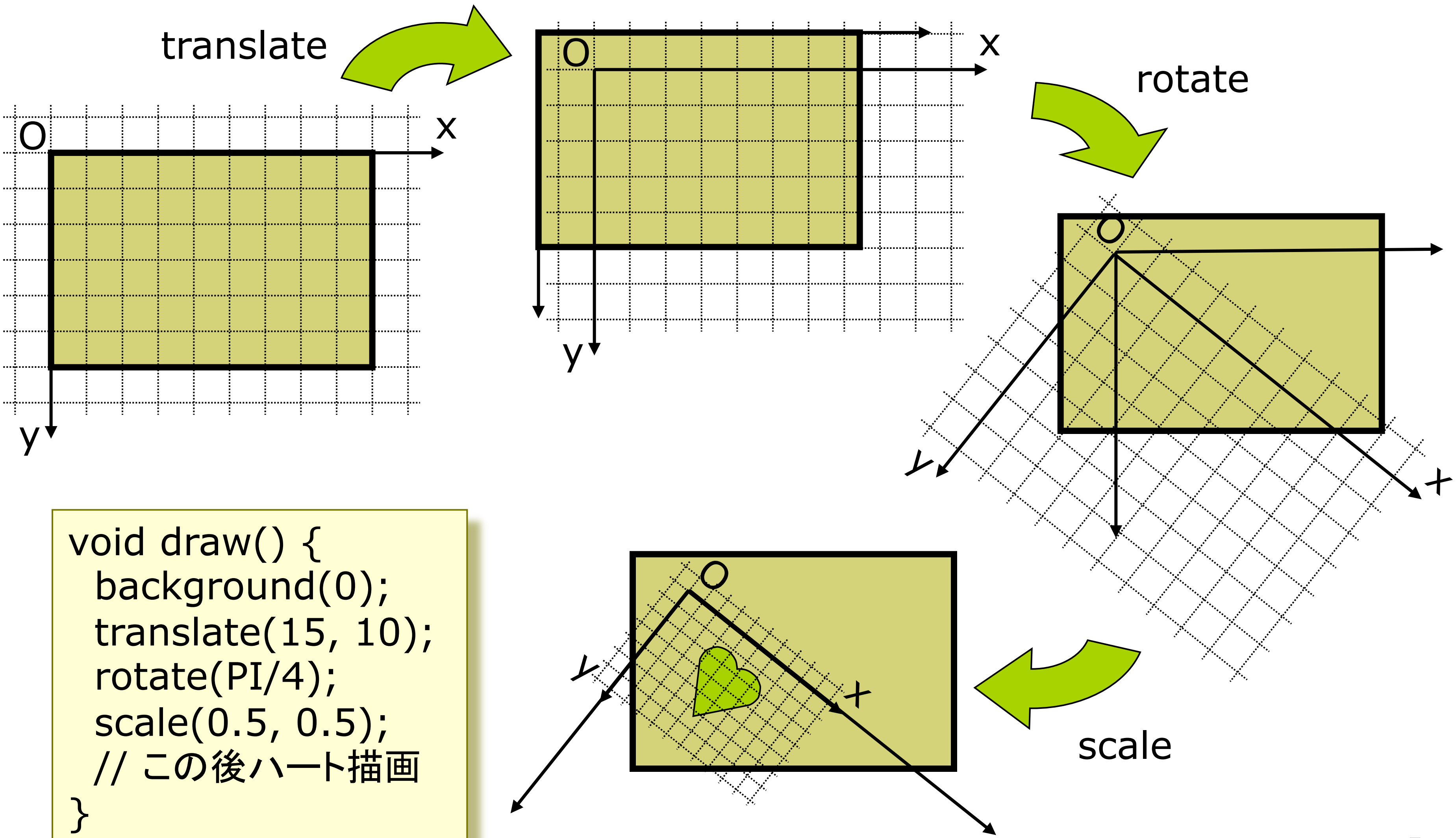
- 拡大・縮小

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha & 0 & 0 \\ 0 & \beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- 回転

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

6.6 幾何変換の合成(p.22)




```
void draw() {
  background(0);
  translate(15, 10);
  rotate(PI/4);
  scale(0.5, 0.5);
  // この後ハート描画
}
```

6.7 合成変換行列(p.22)

合成変換の数学表現

- 同次変換行列の積になる

$$P_{win} = M_1 M_2 M_3 \cdots M_n P$$

$$M = M_1 M_2 M_3 \cdots M_n$$

```
void draw() {
  background(0);
  translate(15, 10);  // 変換 M1
  rotate(PI/4);       // 変換 M2
  scale(0.5, 0.5);    // 変換 M3
  // 図形描画…
}
```

- 右上の例の行列表現

$$\begin{bmatrix} x_{win} \\ y_{win} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 15 \\ 0 & 1 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_{win} \\ y_{win} \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/4 & -\sqrt{2}/4 & 15 \\ \sqrt{2}/4 & \sqrt{2}/4 & 10 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \therefore\ M = \begin{bmatrix} \sqrt{2}/4 & -\sqrt{2}/4 & 15 \\ \sqrt{2}/4 & \sqrt{2}/4 & 10 \\ 0 & 0 & 1 \end{bmatrix}$$

6.8 図形移動での考え方(p.23)

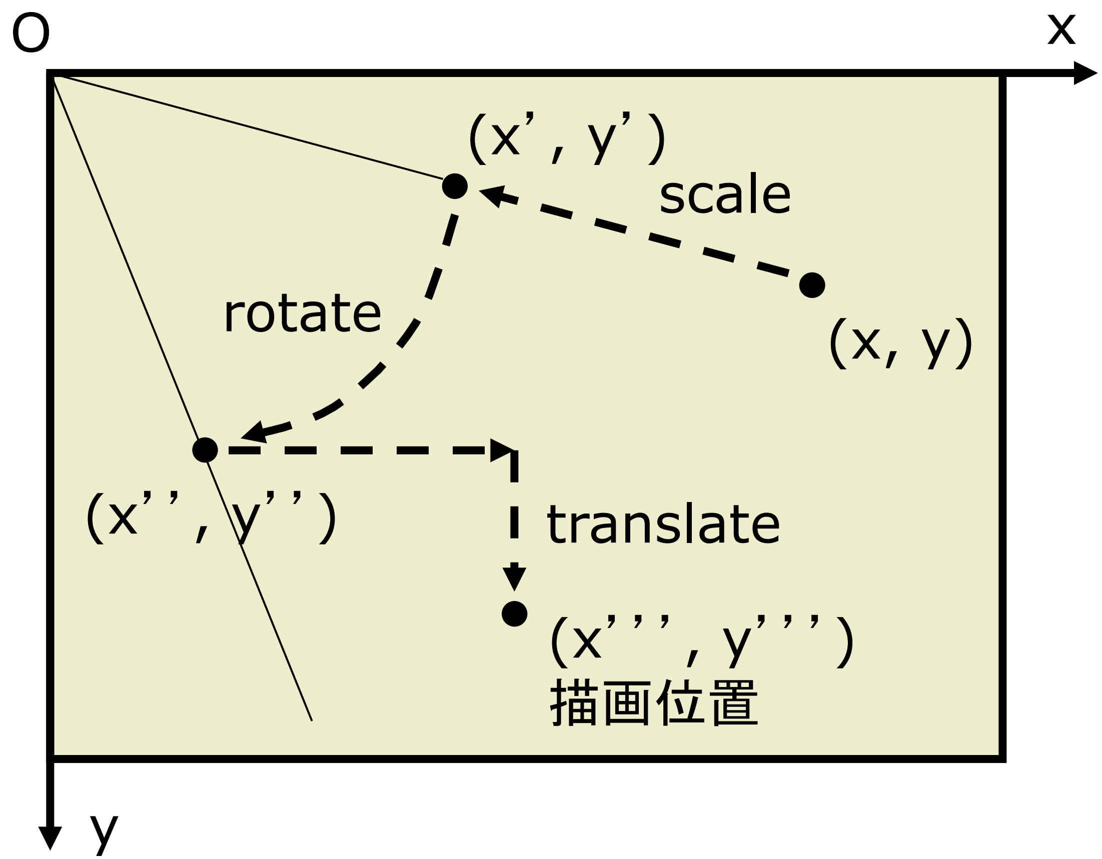
座標変換の別の解釈

- 座標系の移動ではなく，同じ画面座標系上での図形の移動として考えることもできる
- その場合，描画からさかのぼって，図形に命令の逆順で変換を作用させる
- 数学的に同じこと
  ＝結果はどちらも同じ

- 右図の例

```
translate(15, 10);
rotate(PI/4);
scale(0.5, 0.5);
point(x, y);
```

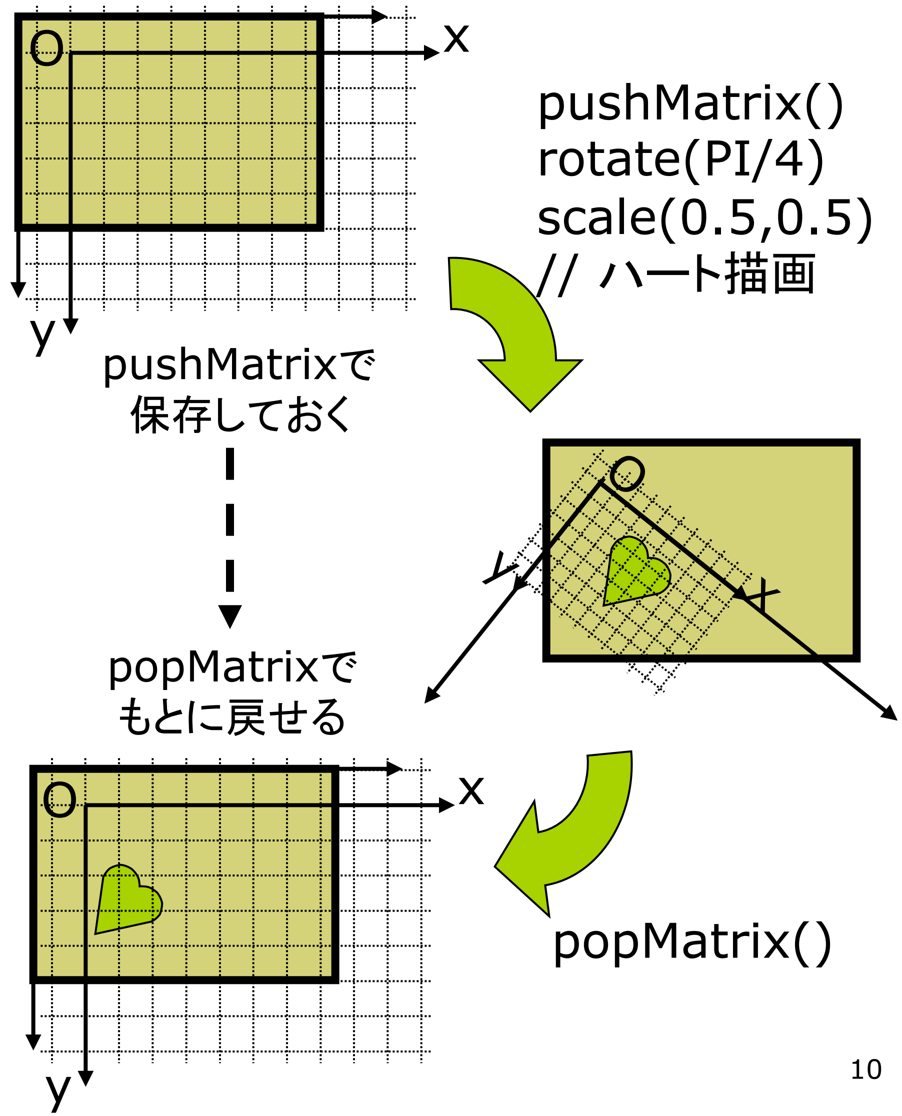
下から順に作用させる



6.9 変換行列の操作(p.45)

行列スタックの操作

- システム変換行列
  - 現在の座標系を示す行列
  - システム変換行列は幾何変換(translate, rotate, scale)の処理のたびに合成されていく
- pushMatrix()
  - システム変換行列(現在の座標系)を一時待避する
- popMatrix()
  - 最近保存した変換行列を戻す
  - pushMatrix()と必ず対にする
- resetMatrix()
  - 変換行列をリセットする
  - 画面座標系 ＝ 論理座標系の初期状態に戻す



6.10 幾何変換と行列操作の例

```
// 描画の原点を移動する例
float bai = 1.0;

void setup() {
    size(400, 400);
    rectMode(CENTER);
}

void draw() {
    background(255);
    translate(200, 200);
    scale(bai);

    rect(0, 0, 50, 50);
    bai += 0.02;
    if (bai > 8.0) bai = 1.0;
}
```

```
// 行列のpushとpopの例
void setup() {
  size(600, 400);
  rectMode(CENTER);
  noLoop();
}

void draw() {
  background(#8080e0);
  pushMatrix();
    translate(200, 200);
    fill(#ffd0d0); rect(0,0, 50, 50);
  popMatrix();
  pushMatrix();
    translate(400, 200);
    rotate(radians(45));
    fill(#ffff00); rect(0,0, 50, 50);
  popMatrix();
}
```

6.11 演習課題

課題

- 6.12のプログラムは，スマイリー（にこちゃんマーク）を2つ描画するものである

問1) 中心と外側の顔の描画位置を決めている**合成変換行列**（M$_{中心}$とM$_{外側}$）の**両方を**求めなさい

- M$_{中心}$は右のヒント参照
- 次回，**A4レポート用紙**で提出

問2) このプログラムに幾何変換の関数を2つ加えて，外側の顔の大きさを半分にして，顔の向きは回転しないようにしなさい

- ただし，顔を描画する関数は，変更したり追加したりしないこと
- プログラム（.pde）をWeb提出

問1のM$_{中心}$のヒント

- M$_{中心}$は次の2つの変換の合成
  - M$_1$ = translate(200, 200)
  - M$_2$ = rotate(-a)
- それぞれの行列表現は

$$M_1 = \begin{bmatrix} 1 & 0 & 200 \\ 0 & 1 & 200 \\ 0 & 0 & 1 \end{bmatrix}$$

$$M_2 = \begin{bmatrix} \cos(-a) & -\sin(-a) & 0 \\ \sin(-a) & \cos(-a) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- M$_{中心}$はこの2つの合成なので

$$M = \begin{bmatrix} 1 & 0 & 200 \\ 0 & 1 & 200 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(-a) & -\sin(-a) & 0 \\ \sin(-a) & \cos(-a) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

6.12 演習課題（続き）

```
void setup() {
    size(400, 400);
    frameRate(30);
}

void draw_smiley() {
    ellipseMode(CENTER);
    strokeWeight(3);
    stroke(0);  fill(#ffff00);
    ellipse(0, 0, 100, 100);
    noStroke();  fill(0);
    ellipse(-15, -15, 12, 12);
    ellipse( 15, -15, 12, 12);
    stroke(#ff0000);  noFill();
    bezier(-25, 20, -10, 35,
              10, 35,  25, 20);
}
```

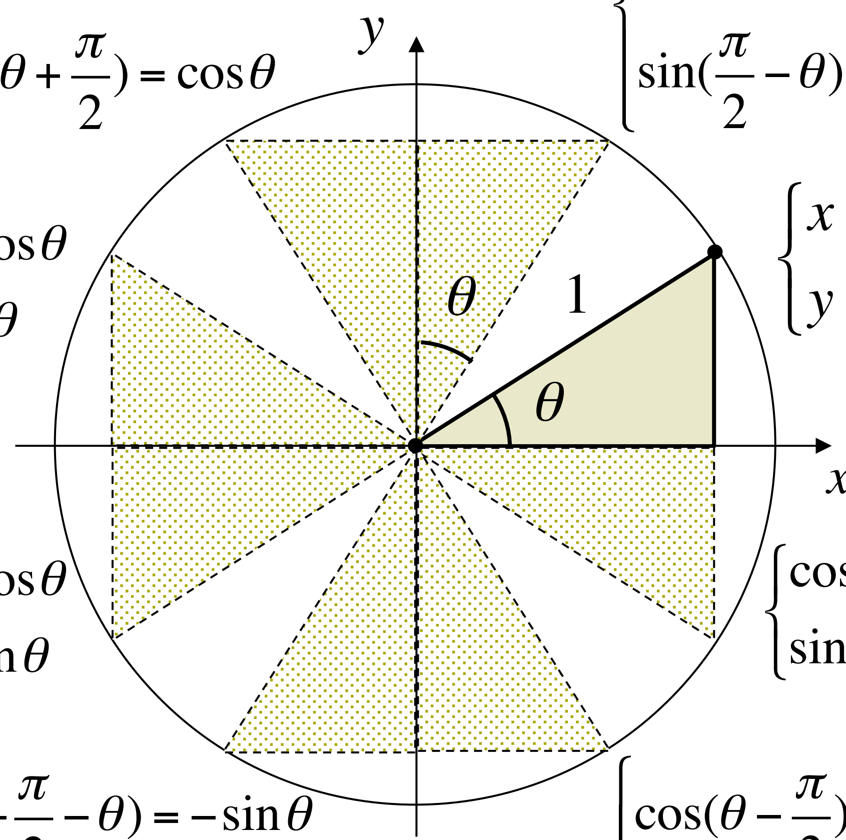
```
void draw() {
    float a = radians(frameCount);
    background(255);
    translate(200, 200);  // 原点移動
    // ★
    pushMatrix();
        rotate(-a);
        draw_smiley();
    popMatrix();
    // ★
    pushMatrix();
        rotate(-a);
        translate(130, 0);
        // ここに2つ幾何変換を追加する
        draw_smiley();
    popMatrix();
    // ★
}
```

★のところ
の座標系は
同じになる

6.13 参考：三角関数の関係式

$$\begin{cases}\cos(\theta+\frac{\pi}{2})=-\sin\theta\\ \sin(\theta+\frac{\pi}{2})=\cos\theta\end{cases}$$

$$\begin{cases}\cos(\frac{\pi}{2}-\theta)=\sin\theta\\ \sin(\frac{\pi}{2}-\theta)=\cos\theta\end{cases}$$



$$\begin{cases}x=\cos\theta\\ y=\sin\theta\end{cases}$$

$$\begin{cases}\cos(\pi-\theta)=-\cos\theta\\ \sin(\pi-\theta)=\sin\theta\end{cases}$$

$$\begin{cases}\cos(\theta\pm\pi)=-\cos\theta\\ \sin(\theta\pm\pi)=-\sin\theta\end{cases}$$

$$\begin{cases}\cos(-\theta)=\cos\theta\\ \sin(-\theta)=-\sin\theta\end{cases}$$

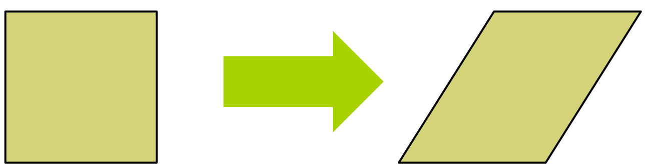
$$\begin{cases}\cos(\frac{3\pi}{2}-\theta)=\cos(-\frac{\pi}{2}-\theta)=-\sin\theta\\ \sin(\frac{3\pi}{2}-\theta)=\sin(-\frac{\pi}{2}-\theta)=-\cos\theta\end{cases}$$

$$\begin{cases}\cos(\theta-\frac{\pi}{2})=\sin\theta\\ \sin(\theta-\frac{\pi}{2})=-\cos\theta\end{cases}$$

6.14 参考：せん断と鏡映

せん（剪）断/スキュー/シアー

- 斜めにゆがめる変換
  - 座標系を平行四辺形にゆがめる
  - 変換後も平行関係は保たれる



- shearX(角度)
  - x軸方向のせん断
  - x軸より上は左に，x軸より下は右にずれていくように歪める
  - y軸を指定の角度だけ傾ける

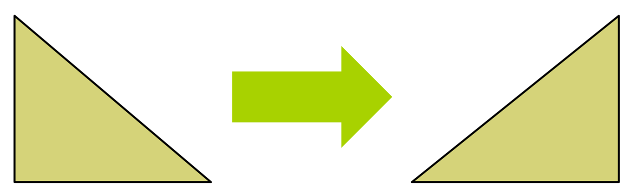
$$x' = x + a\,y \qquad (a = \tan\theta)$$
$$y' = y$$

- shearY(角度)
  - y軸方向のせん断

$$x' = x \qquad (b = \tan\theta)$$
$$y' = b\,x + y$$

鏡映（反転）

- 負の拡大縮小変換
  - x軸またはy軸を基準に反転
  - 例）scale(-1, 1)



図の例の変換式

$$x' = (-1)\cdot x$$
$$y' = y$$

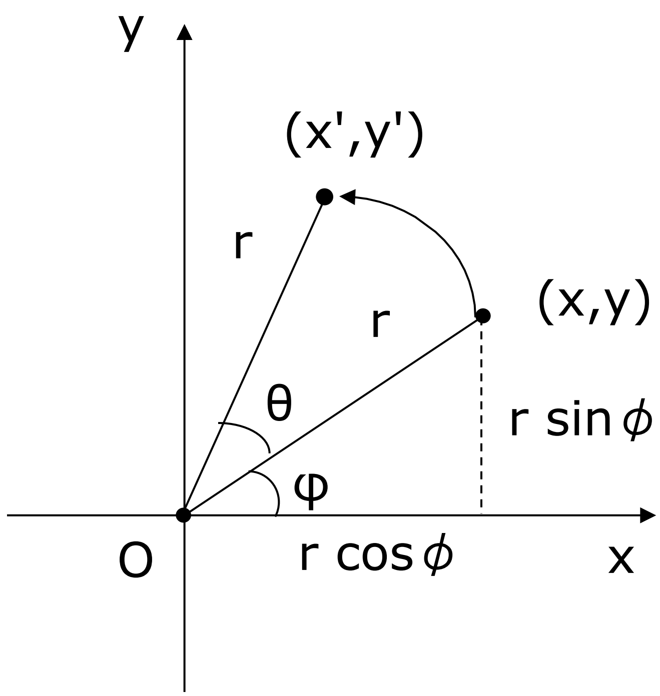
6.15 参考：回転行列の導出

初期位置 (x, y)

$$\begin{cases} x = r\cos\phi \\ y = r\sin\phi \end{cases}$$

θ回転後 (x', y')

$$\begin{cases} x' = r\cos(\phi+\theta) \\ y' = r\sin(\phi+\theta) \end{cases}$$



展開計算（加法定理）

$$\begin{aligned} x' &= r\left(\cos\phi\cos\theta - \sin\phi\sin\theta\right) \\ &= r\cos\phi\cdot\cos\theta - r\sin\phi\cdot\sin\theta \\ &= x\cos\theta - y\sin\theta \end{aligned}$$

$$\begin{aligned} y' &= r\left(\sin\phi\cos\theta + \cos\phi\sin\theta\right) \\ &= r\sin\phi\cdot\cos\theta + r\cos\phi\cdot\sin\theta \\ &= y\cos\theta + x\sin\theta \\ &= x\sin\theta + y\cos\theta \end{aligned}$$

行列形式

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

0.7 行列計算の復習(2)

行列同士の積

$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix} = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{12} \\ a_{21}b_{11} + a_{22}b_{12} \end{bmatrix}$$

$$\left[\begin{array}{cc} a_{11} & a_{12} \\ \hdashline a_{21} & a_{22} \end{array}\right] \left[\begin{array}{c:c} b_{11} & b_{12} \\ b_{21} & b_{22} \end{array}\right] = \begin{bmatrix} a_{11}b_{11} + a_{12}b_{21} & a_{11}b_{12} + a_{12}b_{22} \\ a_{21}b_{11} + a_{22}b_{21} & a_{21}b_{12} + a_{22}b_{22} \end{bmatrix}$$

$$\begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix} =$$