Graphics with Processing

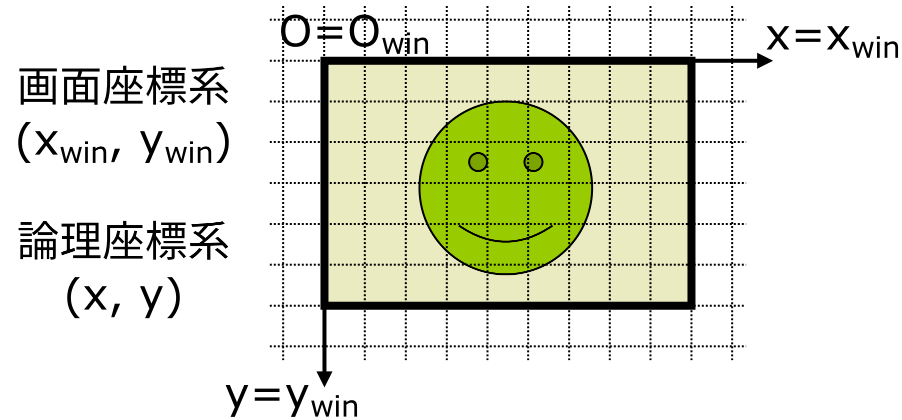
2022-06 座標変換と同次座標

http://vilab.org
塩澤秀和

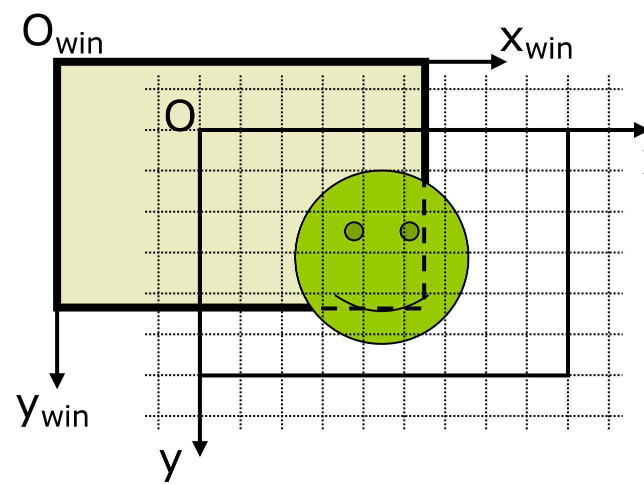
6.1* 座標系

座標系の利用

- 座標系＝目盛りのつけ方
  - （画面上の）原点の位置は？
  - x軸とy軸の方向は？
  - x軸とy軸の目盛りの刻み幅は？
- 画面座標系
  - 左上隅を(0,0)とする初期状態
  - ウィンドウ内の各画素に対応
- 論理座標系
  - 描画命令で使うためのxy座標
  - 画面座標系とは違う描きやすい目盛りのつけ方に変更できる
- 座標系を使う理由
  - “土台”となる座標系を動かせば図形の点をまとめて移動できる
  - コードやデータはそのまま使える



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.22)

座標変換

- 座標変換
  - 任意の座標値を別の座標値に対応させる(写像する)計算

$$(x, y) \longrightarrow (x', y')$$

  - 論理座標系で指示された図形の座標は,画面座標系における表示位置に座標変換される
- 座標変換の合成
  - CGでは基本的な「幾何変換」を多段階に「合成」して利用する

論理座標

$$(x, y) \rightarrow (x', y') \rightarrow \cdots \rightarrow (x_{win}, y_{win})$$

画面座標

幾何変換(幾何学的変換)

- 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

$$x' = \alpha\, x$$
$$y' = \beta\, y$$

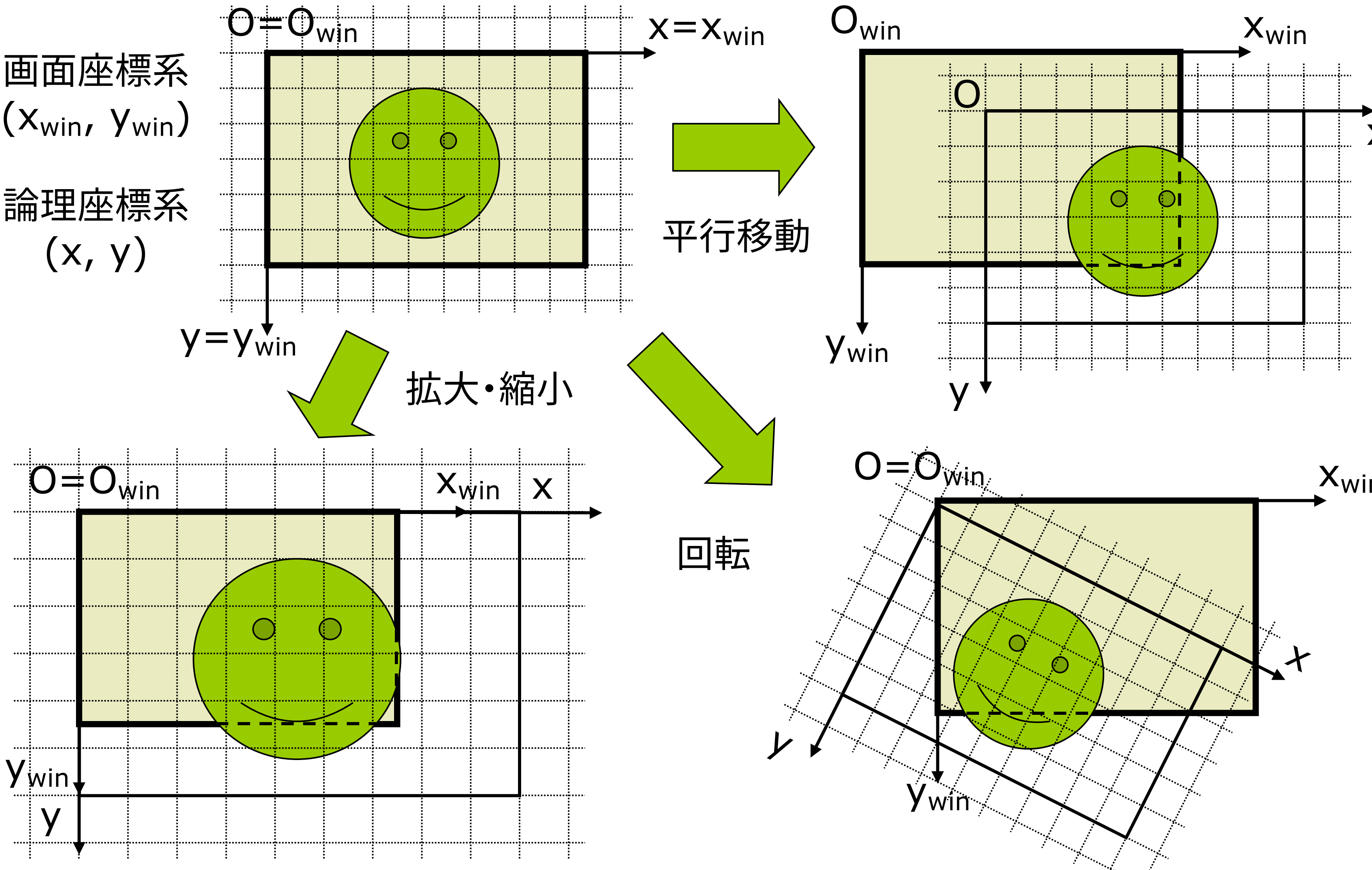
例) 目盛りを横2倍,縦3倍に拡大して,座標 (5, 3) に点を打つと,画面では(10, 9) に表示

- 回転

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

導出方法は6.16参照

6.3* 幾何変換の効果


画面座標系
(xwin, ywin)
論理座標系
(x, y)
O=Owin
x=xwin
y=ywin
平行移動
Owin
xwin
O
x
ywin
y
拡大・縮小
O=Owin
xwin
x
ywin
y
回転
O=Owin
xwin
x
y
ywin

6.4 幾何変換関数

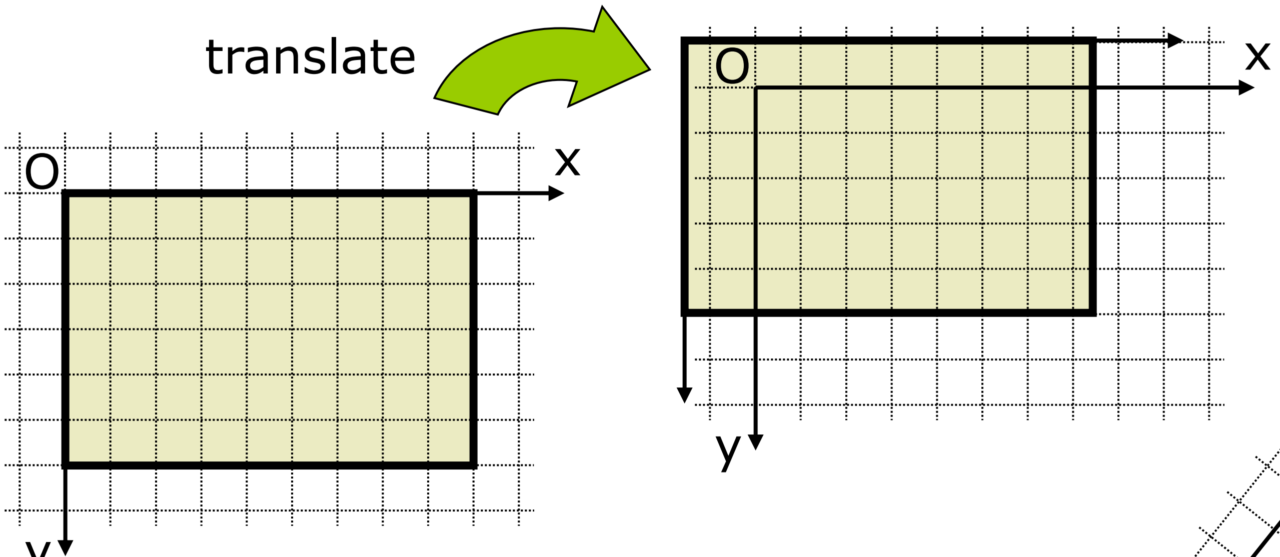
幾何変換関数

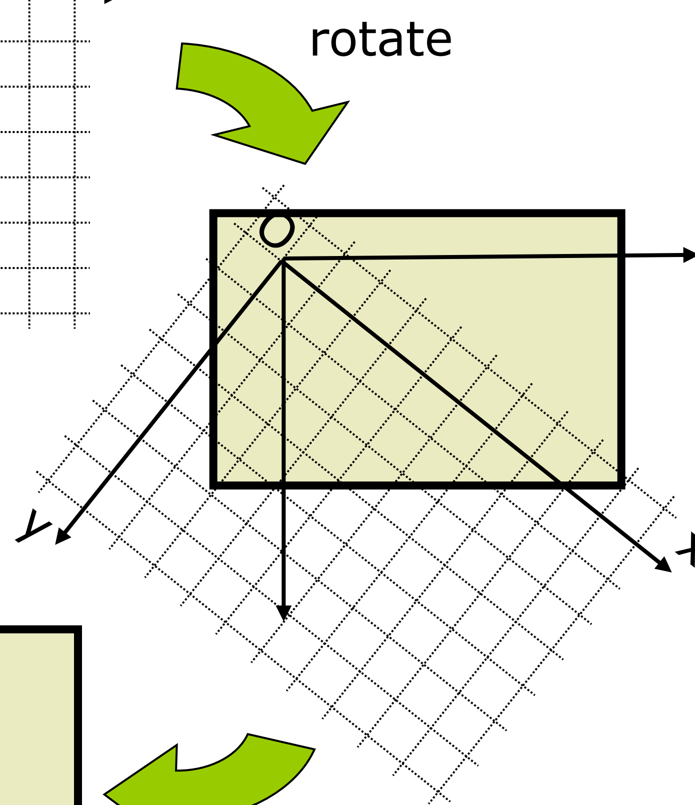
- translate(x_0, y_0)
  - 座標系を平行移動(原点を移動)
  - x軸方向に x_0 移動
  - y軸方向に y_0 移動
  - Processingではy軸は下向き
- scale(α, β)
  - 座標系を拡大または縮小
  - x軸方向(左右)に α 倍
  - y軸方向(上下)に β 倍
  - 原点を中心に全体を拡大
- rotate(θ)
  - 座標系を回転
  - 原点を中心に θ ラジアン回転
  - Processingで+方向は時計回り

```
void setup() {
  size(400, 400);
  noLoop();
}

void draw() {
  background(0);
  // translate(150, 100);
  // rotate(PI/4);
  // scale(0.5, 0.5);
  drawHeart();
}

void drawHeart() {
  beginShape();
  vertex(150, 180);
  bezierVertex(50, 110, 120, 50,
                      150,100);
  bezierVertex(180, 50, 250, 110,
                      150,180);
 endShape(CLOSE);
}
```

1つずつ有効にして
効果を確認してみよう

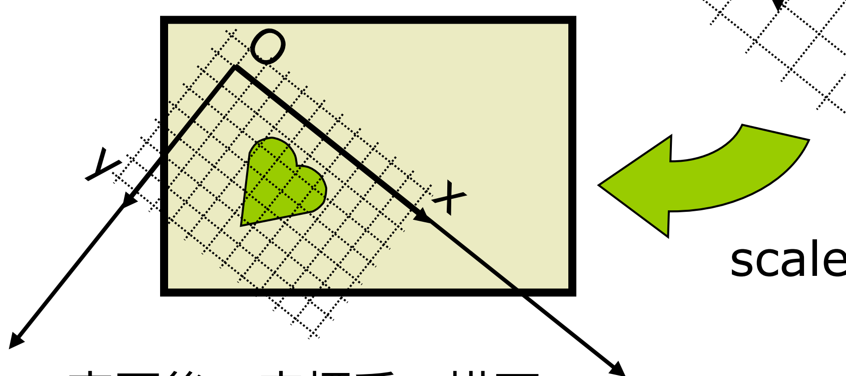
6.5* 幾何変換の合成(p.22)





論理座標系が順に更新されていき…

```
void draw() {
  background(0);
  translate(150, 100);
  rotate(PI/4);
  scale(0.5, 0.5);
  drawHeart();
}
```



変更後の座標系で描画

6.6* 同次座標表現(p.19)

それぞれ展開し，
6.2の式と比較し
て確かめよう

同次(斉次)座標表現

- CGの内部処理で使われる形式

2次元直交座標　　2次元同次座標

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

同じ座標の別の表現

- 同次座標表現による座標変換

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

すべての幾何変換を行列の掛け算で表せる!

幾何変換行列

- 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

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

6.7* 合成変換行列(p.28)

合成変換の数学的表現

- 行列の積の繰り返しになる

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

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

何段階もの変換をまとめた行列

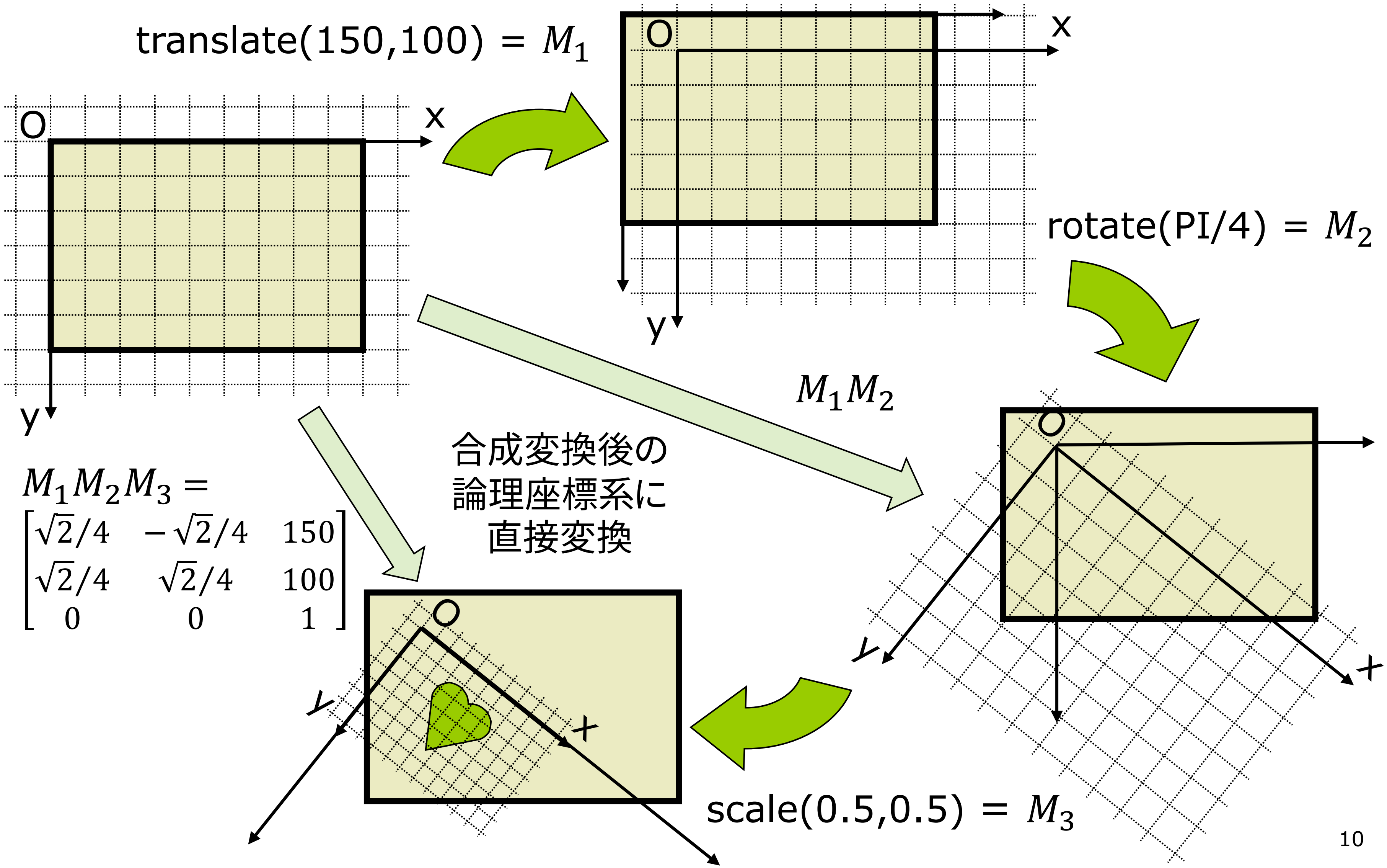
```
void draw() {
  background(0);
  translate(150, 100); // 変換 M1
  rotate(PI/4);        // 変換 M2
  scale(0.5, 0.5);     // 変換 M3
  drawHeart();
}
```

- 右上の例の合成変換行列

$$\begin{bmatrix} x_{win} \\ y_{win} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 150 \\ 0 & 1 & 100 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\pi/4) & -\sin(\pi/4) & 0 \\ \sin(\pi/4) & \cos(\pi/4) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_{win} \\ y_{win} \\ 1 \end{bmatrix} = \begin{bmatrix} \sqrt{2}/4 & -\sqrt{2}/4 & 150 \\ \sqrt{2}/4 & \sqrt{2}/4 & 100 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \therefore M = \begin{bmatrix} \sqrt{2}/4 & -\sqrt{2}/4 & 150 \\ \sqrt{2}/4 & \sqrt{2}/4 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$

6.8* 合成変換行列の意味

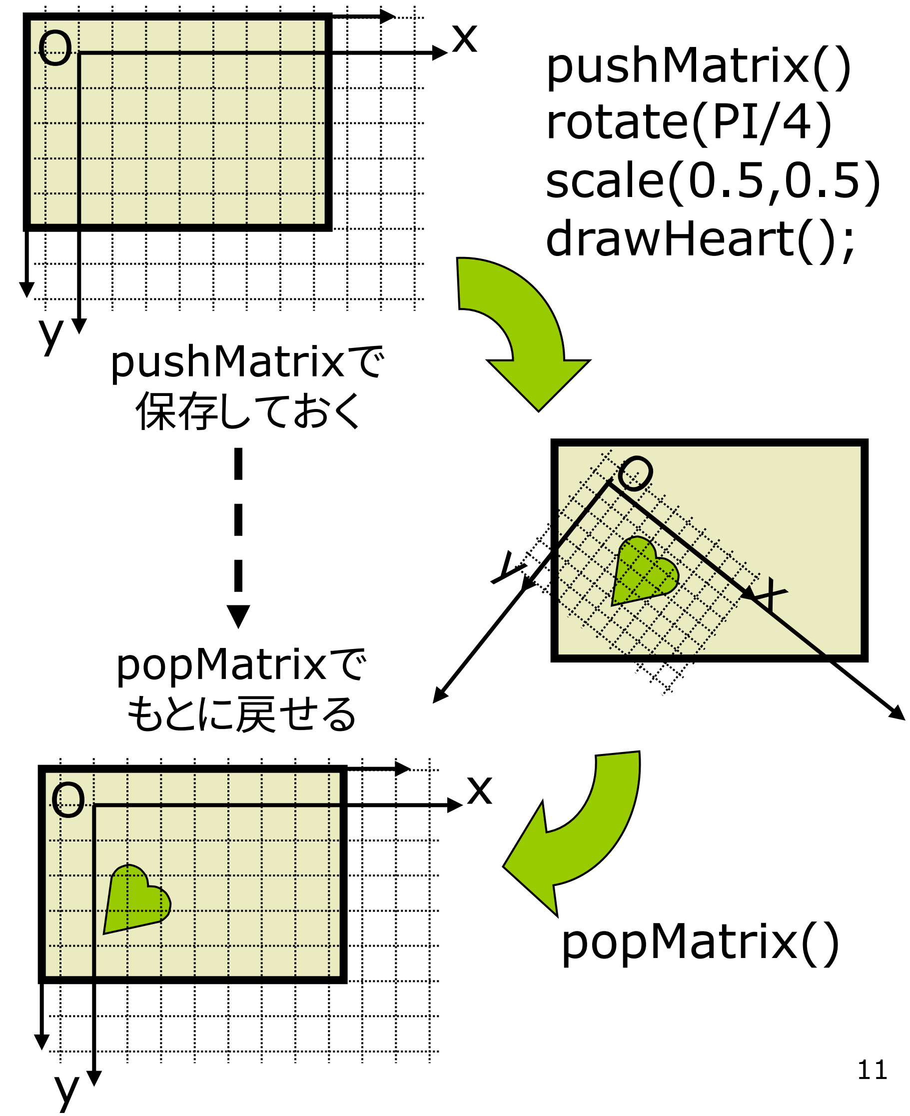


$$M_1 M_2 M_3 = \begin{bmatrix} \sqrt{2}/4 & -\sqrt{2}/4 & 150 \\ \sqrt{2}/4 & \sqrt{2}/4 & 100 \\ 0 & 0 & 1 \end{bmatrix}$$

6.9 変換行列の保存と利用（p.54）

行列スタックの利用

- システム変換行列
  - システムは現在の論理座標系を表す合成変換行列を持つ
  - 幾何変換関数の実行のたびに，行列の掛け算で更新されていく
- pushMatrix()
  - システム変換行列（現在の論理座標系）を一時的に保存する
- popMatrix()
  - 最近保存した変換行列を戻す
  - pushMatrix()と必ず対にする
- resetMatrix()
  - システム変換行列を初期化する（論理座標系＝画面座標系）
  - 保存した行列も全て破棄する



6.10 幾何変換と行列スタックの例

```
PFont font;

void setup() {
  size(400, 400);
  font = createFont("Serif", 48);
  noLoop();
}

void draw() {
  background(0);
  fill(255);

  textFont(font);
  textAlign(CENTER, CENTER);

  translate(200, 100);
  text("壱", 0, 0);
  printMatrix();
```

システム変換行列の表示

```
  pushMatrix();
  translate(0, 100);

  pushMatrix();
    scale(1.5);
    text("弐", 0, 0);
    printMatrix();
  popMatrix();

  translate(0, 100);
  rotate(PI/4);
  text("参", 0, 0);
  printMatrix();

 popMatrix();
}
```

入れ子(多重)にできる

6.11* 演習課題

課題

- 6.12はスマイリー(ニコちゃん)を2つ描画するプログラムである

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
  - A4用紙で次回開始時に提出

問2) これに幾何変換を2つ加えて,外側の顔の向きを回転しないようにして,大きさは半分にしなさい
  - 顔以外の図形に変更してもよい
  - ただし,中心と外側の図形の描画に,必ず同じ関数を使うこと
  - 他にも図形を追加して,様々な動きや変形を試してみるとよい

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

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

かっこ内のマイナスは外す(6.13)

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

- $M_{中心}$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);
}

// 中心と外側で共通の関数を使うこと!
void drawSmiley() {
    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);
    // ★
    pushMatrix();
        rotate(-a);
        drawSmiley();
    popMatrix();
    // ★
    pushMatrix();
        rotate(-a);
        translate(130, 0);
        // ここに2つ幾何変換を追加する
        drawSmiley();
    popMatrix();
    // ★
}
```

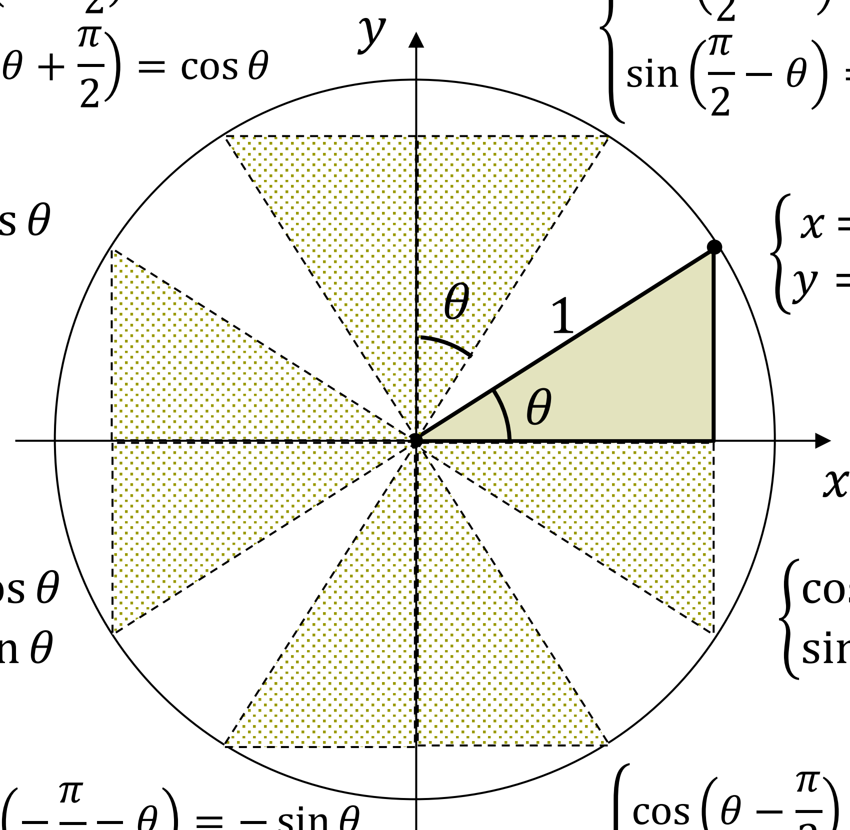
★のところの座標系は同じになる

6.13 三角関数の関係式

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

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

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



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

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

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

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

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

6.14 参考：座標変換の数学的表現

1次変換とアフィン変換

- 1次変換

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

  - 拡大・縮小と回転は表現可能
  - 平行移動は表現できない
- アフィン変換
  - すべての幾何変換を表現可能

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

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

定数項を付加

- 同次（斉次）座標系の利用

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

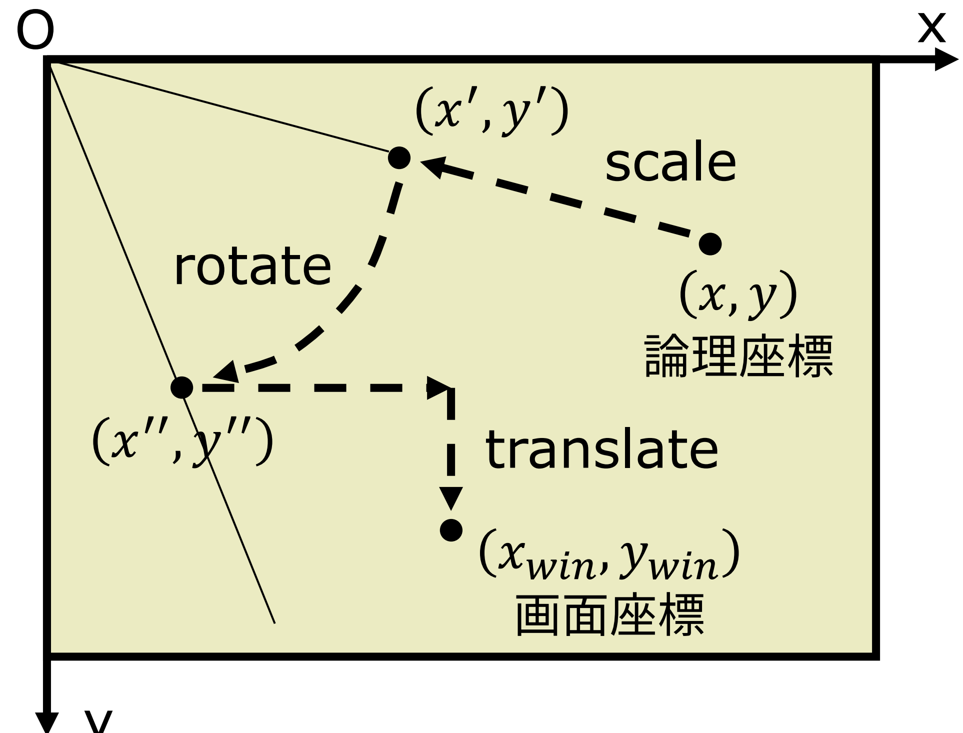
$$\Leftrightarrow \begin{cases} x' = ax + by + e \\ y' = cx + dy + f \\ 1 = 1 \end{cases}$$

  - 1つの行列で表現できる
  - 合成変換の扱いが容易になる
- アフィン変換の特徴
  - 直線は直線に変換される
  - 直線上の線分比が維持される
  - 面積比が維持される

6.15 参考：合成変換の解釈(p.29)

合成変換(6.7)の2通りの解釈

- 座標系が更新される(6.5)
  - 左から計算することに対応
    $P_{win} = \left((M_1 M_2) M_3\right) P$
  - 座標系を表す行列に，変換を表す行列が適用されていき，新しい座標系の中でPの位置に図形が描画される
- 図形の位置が移動する(右図)
  - 右から計算することに対応
    $P_{win} = M_1\left(M_2(M_3 P)\right)$
  - 座標Pに変換を表す行列が適用されていき，(座標系はそのままで)計算後の座標の位置に図形が描画される
- 数学的には等価(解釈の問題)




```
translate(150, 100);
rotate(PI/4);
scale(0.5, 0.5);
point(x, y);
```

変換を下から考える

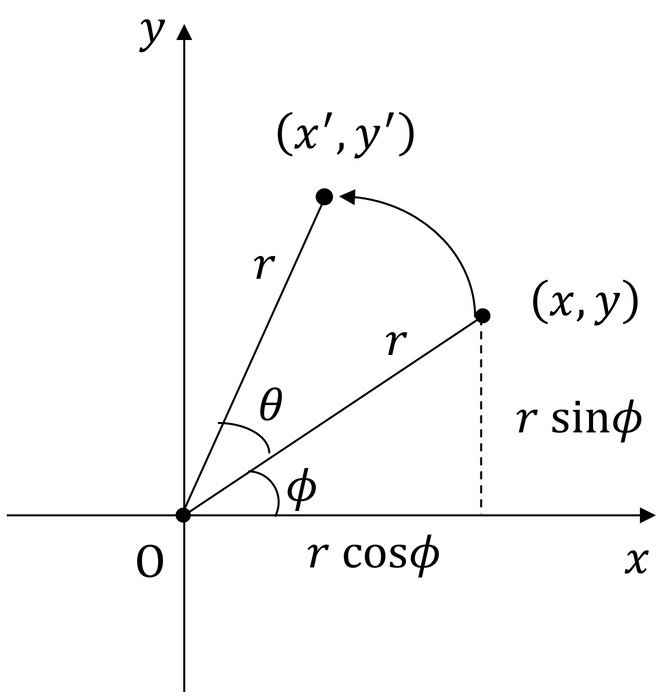
6.16 参考：回転行列の導出

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）

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

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

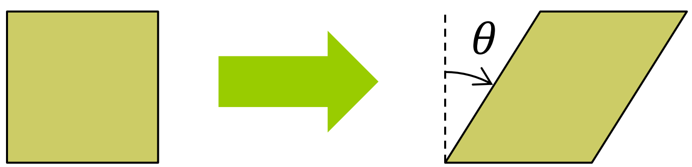
行列形式

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

6.17 参考：せん断と鏡映（p.26）

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

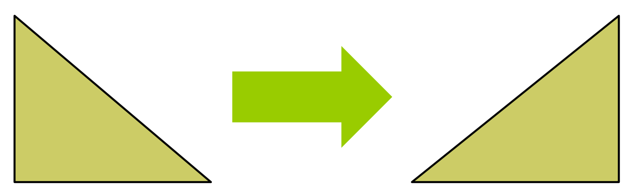
$$x' = x + y\tan\theta$$
$$y' = y$$

- shearY(角度)
  - y軸方向のせん断

$$x' = x$$
$$y' = x\tan\theta + y$$

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

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

6.18 参考：行列計算の確認

$$\begin{bmatrix} a & b & e \\ c & d & f \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} ax+by+e \\ cx+dy+f \\ 1 \end{bmatrix}$$

i行目 j列目

$$\begin{bmatrix} \boxed{a_1 \quad b_1 \quad e_1} \\ c_1 \quad d_1 \quad f_1 \\ 0 \quad 0 \quad 1 \end{bmatrix}\begin{bmatrix} a_2 & b_2 & e_2 \\ c_2 & d_2 & f_2 \\ 0 & 0 & 1 \end{bmatrix}$$

積の行列のi行j列の値
= 左の行列のi行目と
右の行列のj列目の内積

i行j列

$$= \begin{bmatrix} \boxed{a_1a_2+b_1c_2} & a_1b_2+b_1d_2 & a_1e_2+b_1f_2+e_1 \\ c_1a_2+d_1c_2 & c_1b_2+d_1d_2 & c_1e_2+d_1f_2+f_1 \\ 0 & 0 & 1 \end{bmatrix}$$