Graphics with Processing

2008-07 3次元描画の基礎

http://vilab.org

塩澤秀和

7.1 3D図形の描画

3D基本設定

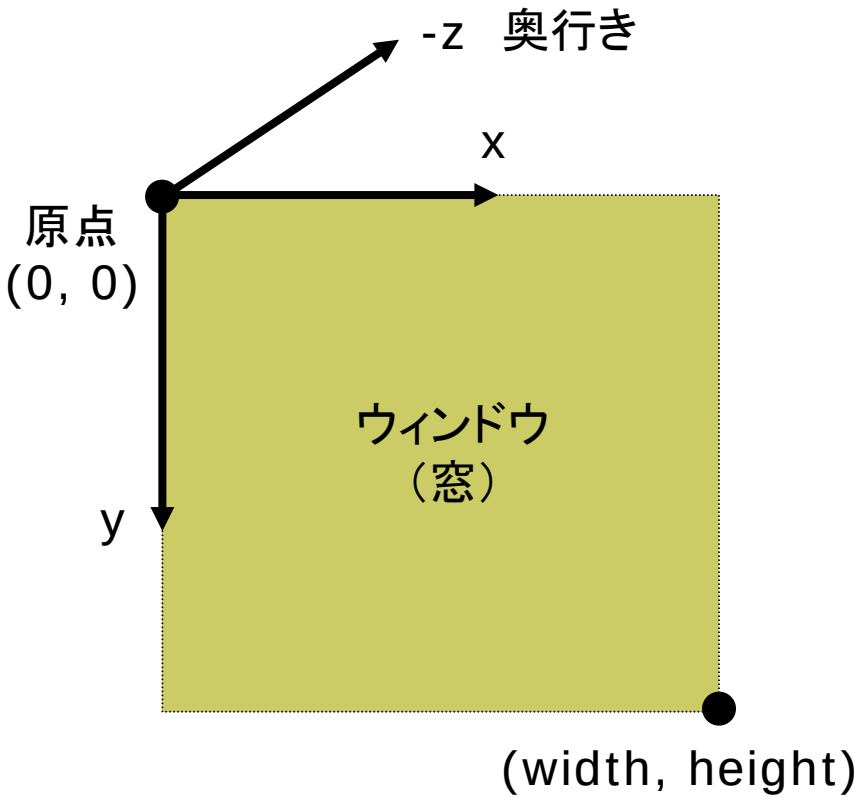
- size(幅, 高さ, P3D)
  - ウィンドウを3D用で開く
  - 次回以降にOPENGLも紹介
- lights()
  - 標準の照明を設定
  - draw()のなかで最初に書く

3D基本形状

- box(辺の長さ)
- box(幅, 高さ, 奥行き)
  - 原点に立方体/直方体を描画
- sphere(半径)
  - 原点に球を描画
  - 通常は noStroke() で描く
- サンプル
  - 3D (and OpenGL) → Forms → Primitives3D

3次元座標系（無指定時）

- Processingではz軸は手前方向



7.2 平行投影と透視投影

平行投影(直交投影)

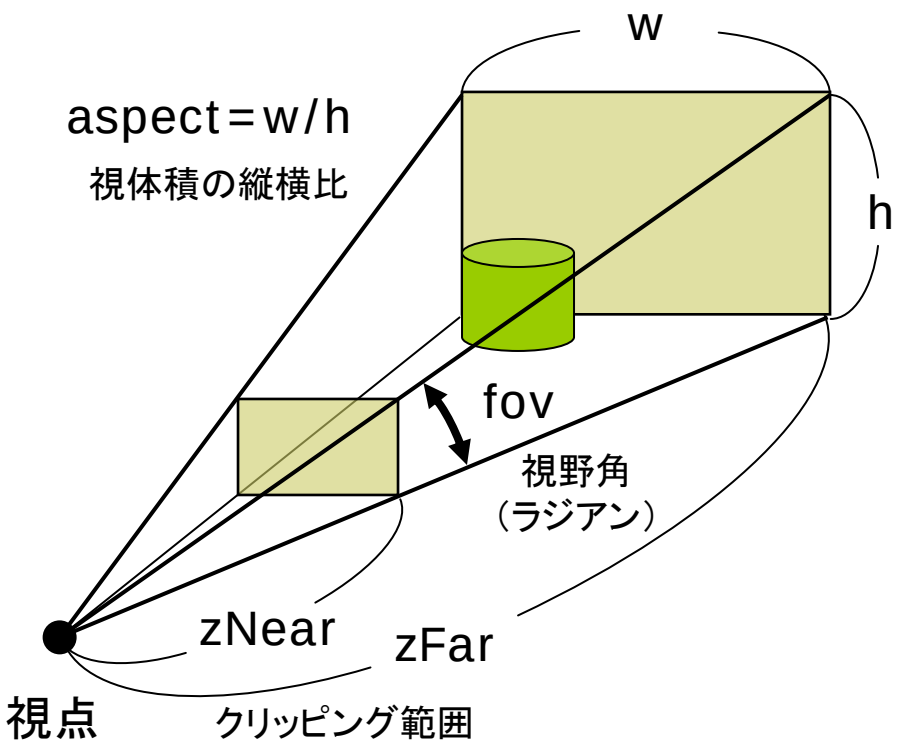
- ortho(x_{min}, x_{max}, y_{min}, y_{max}, z_{min}, z_{max})
  - 遠近感をつけない投影方法
  - 画面に表示するx, y, z座標の範囲(視体積)を設定

透視投影(透視図法)

- pserspective()
  - 近くのものを大きく、遠くのものを小さく、遠近法を使って描画する
- perspective(fov, aspect, zNear, zFar)
  - 視野角などを指定して視体積(右図)を設定(第9回で説明)

透視投影の視体積

- 視体積＝view volume



7.3 3Dでの位置設定

3次元幾何変換

- translate(tx, ty, tz)
  - 3次元平行移動
  - 最初に translate(width/2, height/2) として画面の中心を原点をすると分かりやすい
- scale(sx, sy, sz)
  - 3次元拡大・縮小
- rotateX(θx)
  - x軸まわりの回転
- rotateY(θy)
  - y軸まわりの回転
- rotateZ(θz)
  - z軸まわりの回転
  - 2次元のrotate(θz) と同じ

変換行列の操作

- 2次元とまったく同様
  - 4.7の説明および図を参照
  - pushとpopは必ず対にして、カッコのように対応させること
- pushMatrix()
  - 描画座標系を一時退避する
- popMatrix()
  - 最近保存した描画座標系を戻す
- 使用例

```
pushMatrix();
  translate(150, 100, -100);
  rotateY(radians(30));
  box(150, 50, 100);
popMatrix();
```

7.4 演習課題

課題

- 3D図形（立方体や球でよい）を，画面の中心に見えるように描画するプログラムを作成しなさい
- キーボードかマウスのボタンで、平行投影と透視投影を切り替えられるようにするとよい
- 画面の中心を原点にするとよい

参考サンプル

- 3D (and OpenGL) → Transform → Translate
- 3D (and OpenGL) → Camera → OrthoVsPerspective

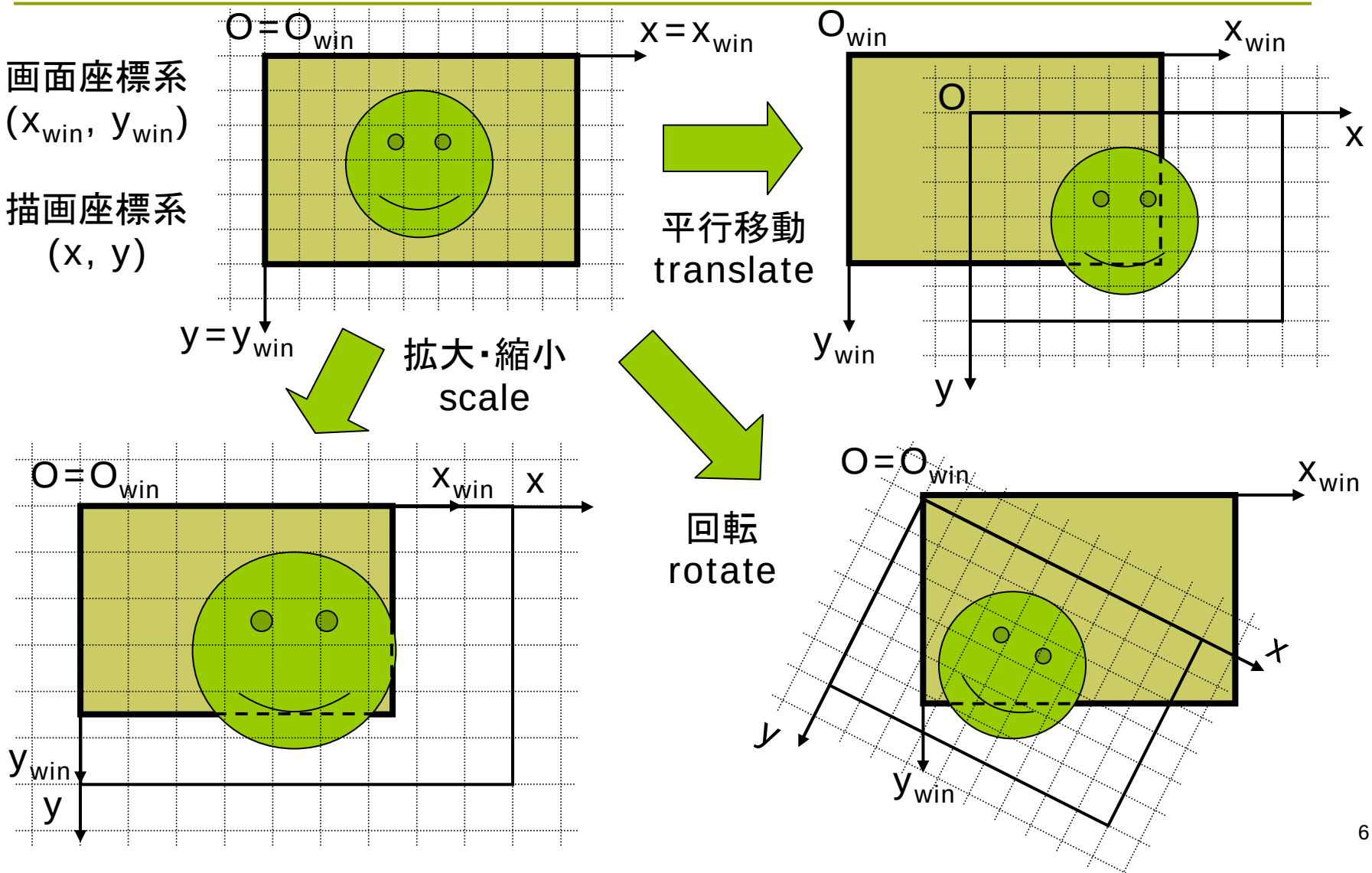
今までの課題について

- 今までの課題で未提出のものがある人は，必ず提出すること
  - ここからがCGの本番です
  - 後半はどんどん難しくなります
  - 今追いつかないとマズイです

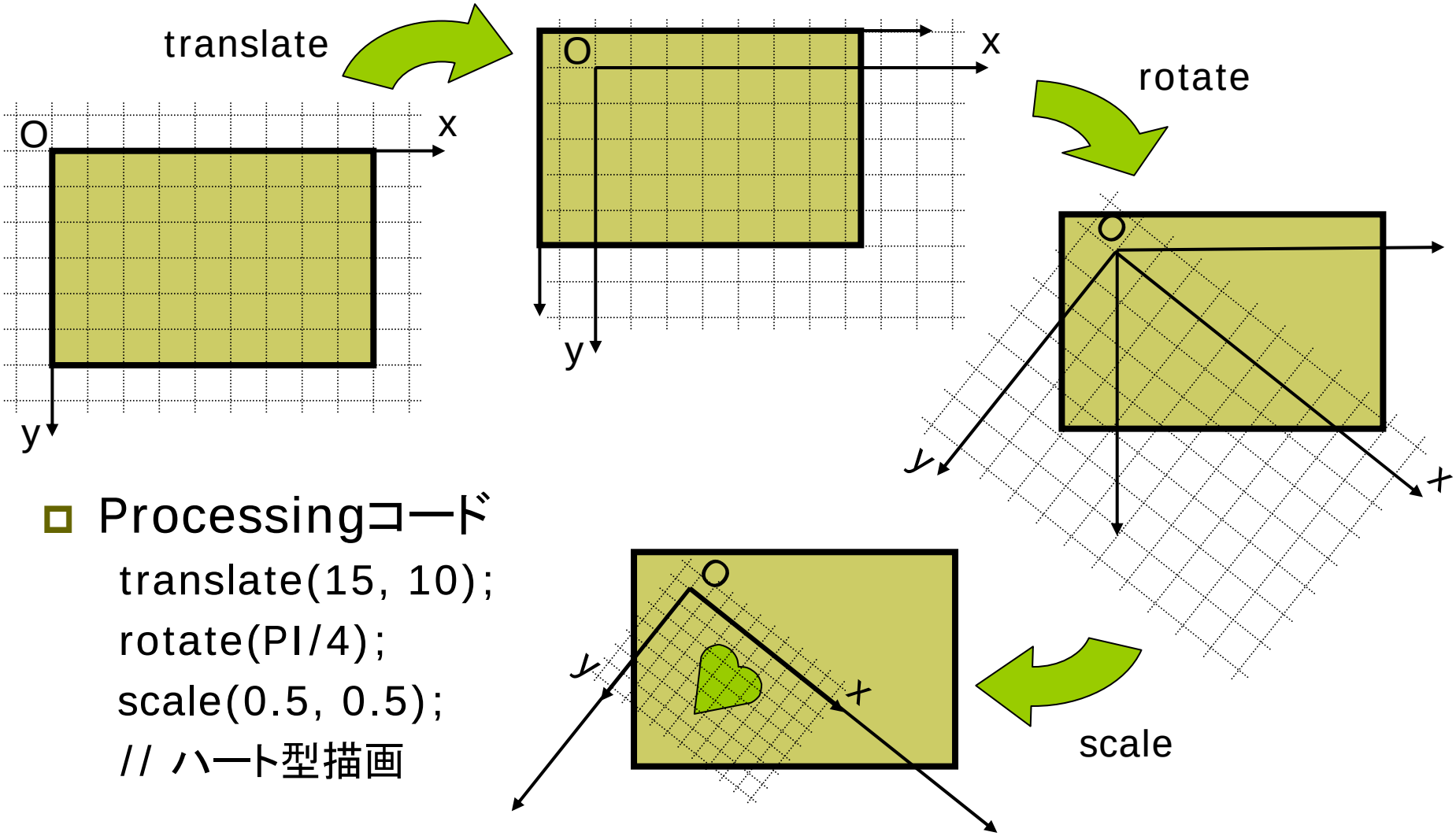
プログラムは分かりやすく

- プログラムは，ただ「動けばいい」ものじゃありません。分かりやすく**きれいに**書かないといけません
- 会社に入ったら，1つのソフトで1000行ぐらいは1人で分担して，しかもちゃんと他人が見ても分かるように書かないといけません

4.2 幾何変換の効果


画面座標系
(xwin, ywin)
描画座標系
(x, y)
O=Owin
x=xwin
y=ywin
平行移動
translate
Owin
xwin
O
x
ywin
y
拡大・縮小
scale
O=Owin
xwin
x
ywin
y
回転
rotate
O=Owin
xwin
x
y
ywin

4.5 幾何変換の合成



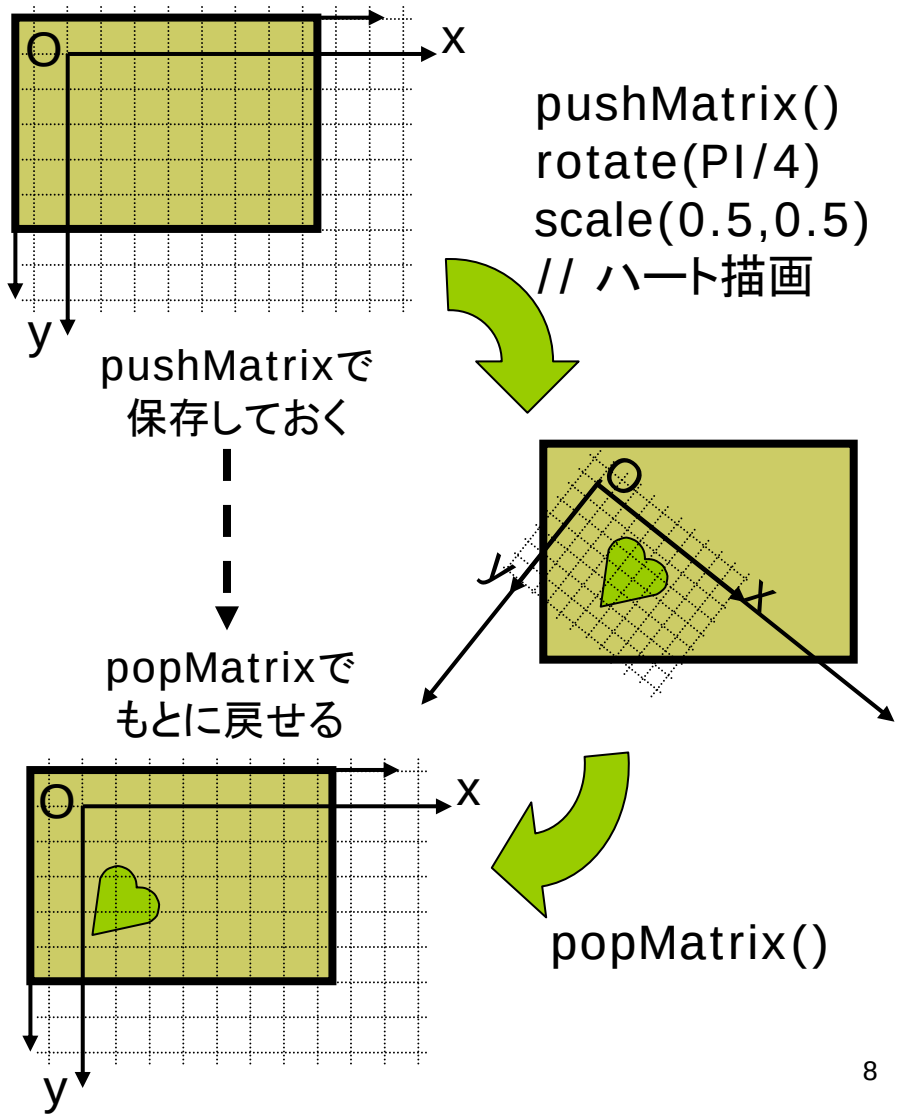
- Processingコード

```
translate(15, 10);
rotate(PI/4);
scale(0.5, 0.5);
// ハート型描画
```

4.7 行列操作

変換行列の操作

- 変換行列
  - 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
  - 変換行列＝描画座標系
- pushMatrix()
  - システム変換行列（描画座標系）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
  - 描画座標系＝画面座標系



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
  - 押されたマウスボタン
  - LEFT, RIGHT, CENTER
- keyPressed
  - キーが押されていればtrue
- key
  - 押された文字
- 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自動で実行される処理を登録
- 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