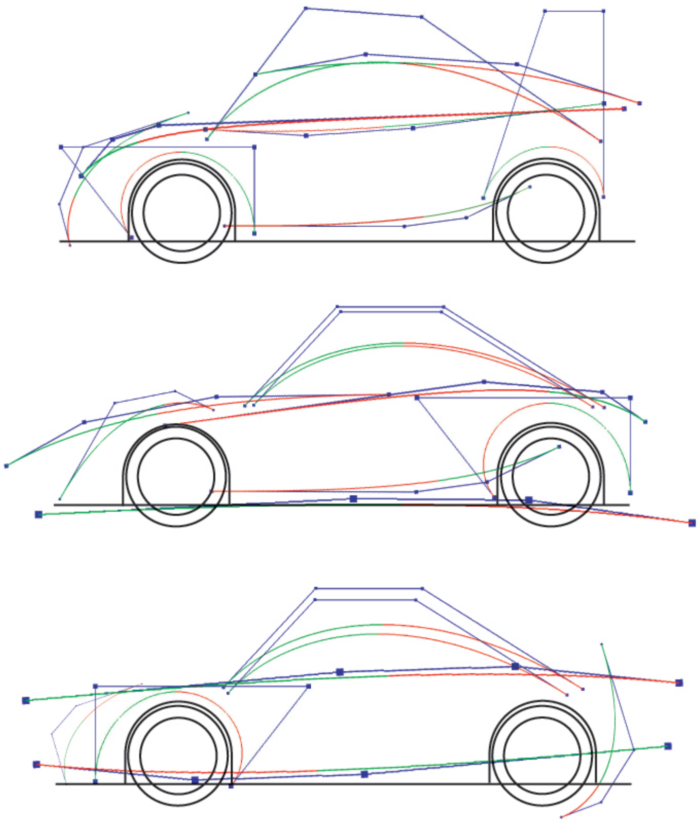
情報工学

2020年度後期 第2回 [10月14日]



静岡大学

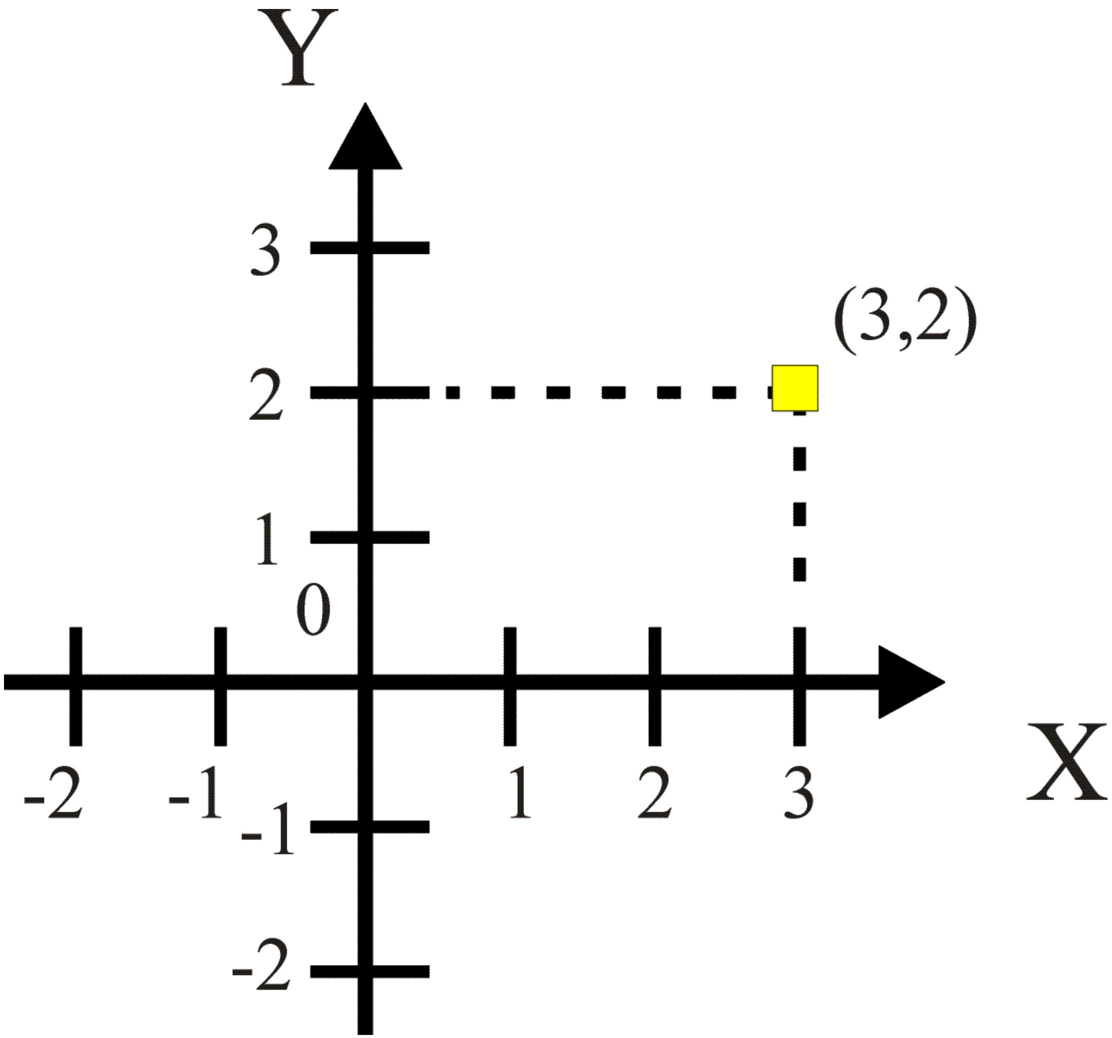
大学院工学研究科機械工学専攻
ロボット・計測情報分野
創造科学技術大学院
情報科学専攻

三浦 憲二郎

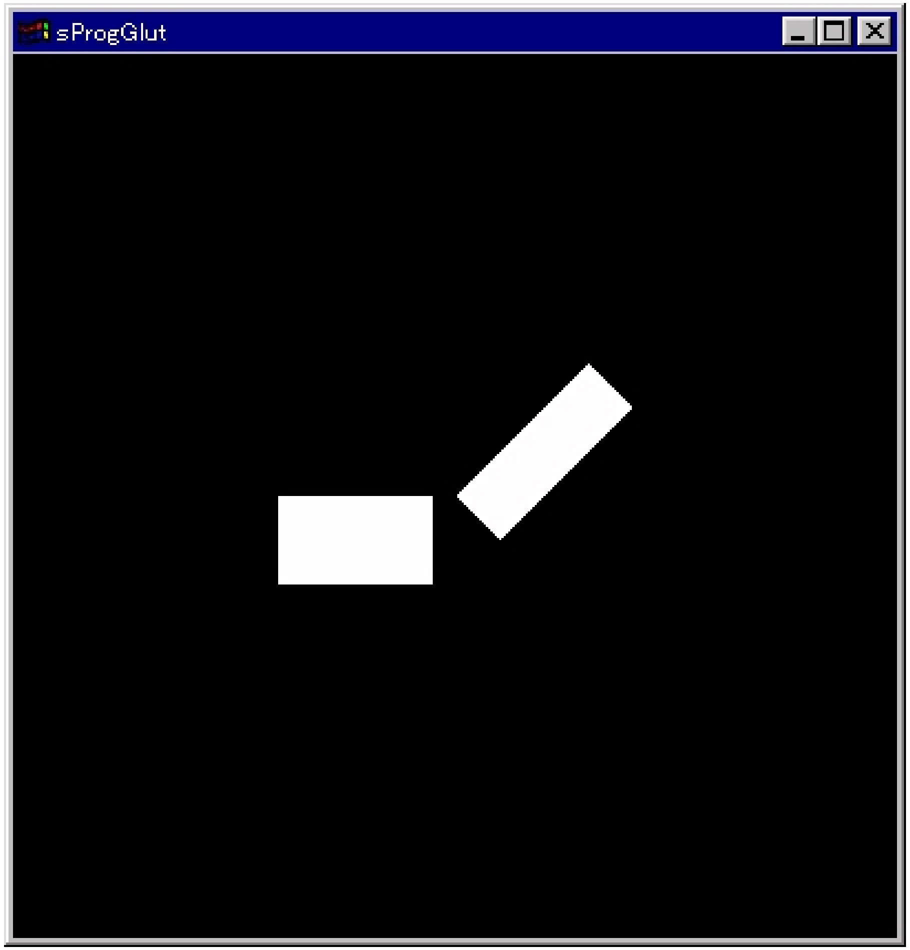
講義アウトライン [10月14日]

- ビジュアル情報処理
  - 1.2 座標系とモデリング
  - 1.3 ビジュアル情報処理の幾何学的モデル
- OpenGL
  - 2D座標系
  - OpenGLによる線の描画

2D座標系



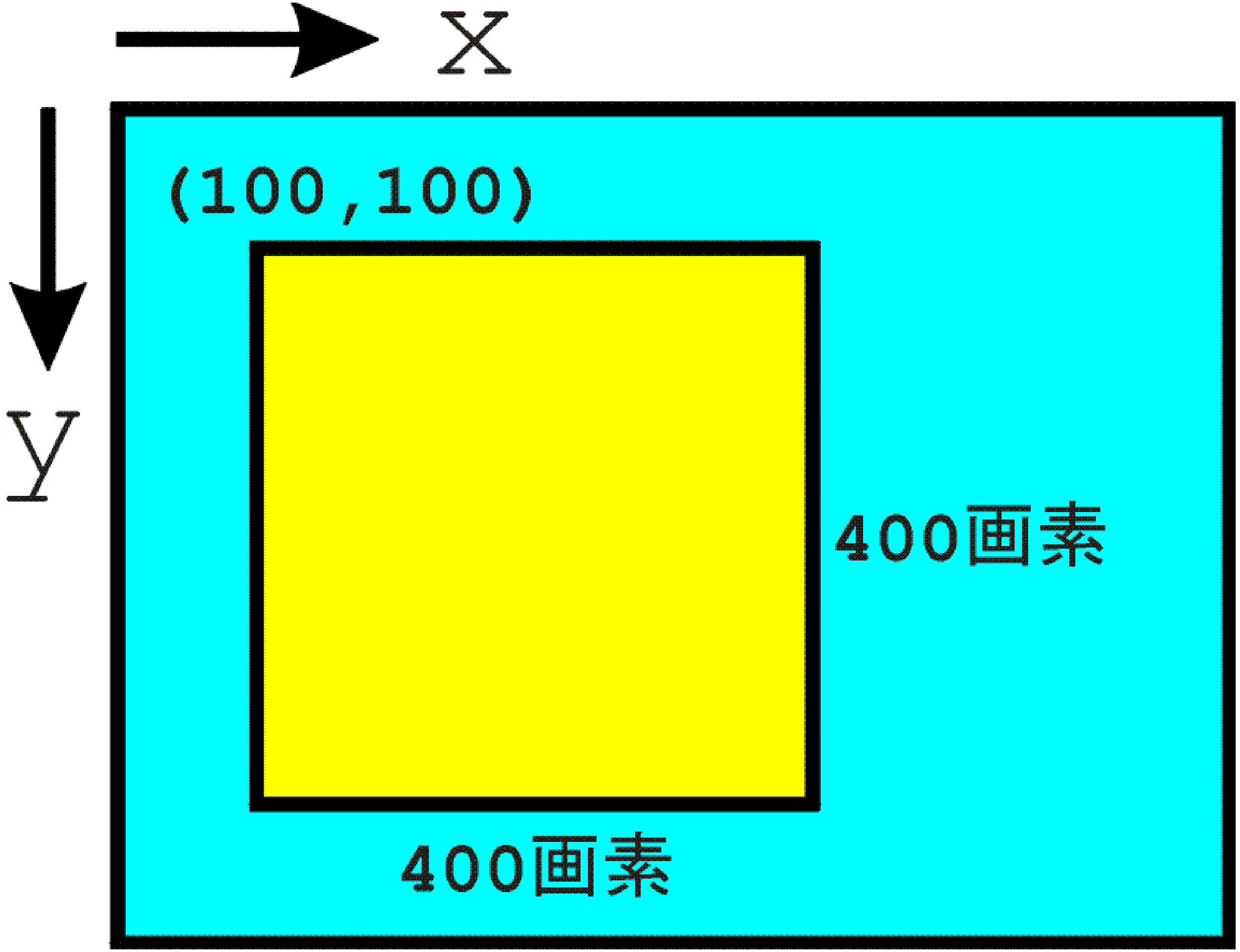
simpleProg.c(実行結果)



simpleProg.c(main())

```
#include <GL/glut.h>
int main(int argc, char **argv)
{
  glutInit(&argc, argv);                    /* GLUTの初期化 */
  glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
                                            /*表示モードの指定*/
  glutInitWindowSize (400,400);             /*画面の大きさの指定*/
  glutInitWindowPosition(100,100);          /*画面の位置の指定 */
  glutCreateWindow (“sProgGlut” );          /*ウインドウのオープン*/
  init();                                   /*初期化処理*/
  glutDisplayFunc(display);                 /*描画関数の指定*/
  gluMainLoop();
  return 0;
}
```

画面の大きさ·位置の指定



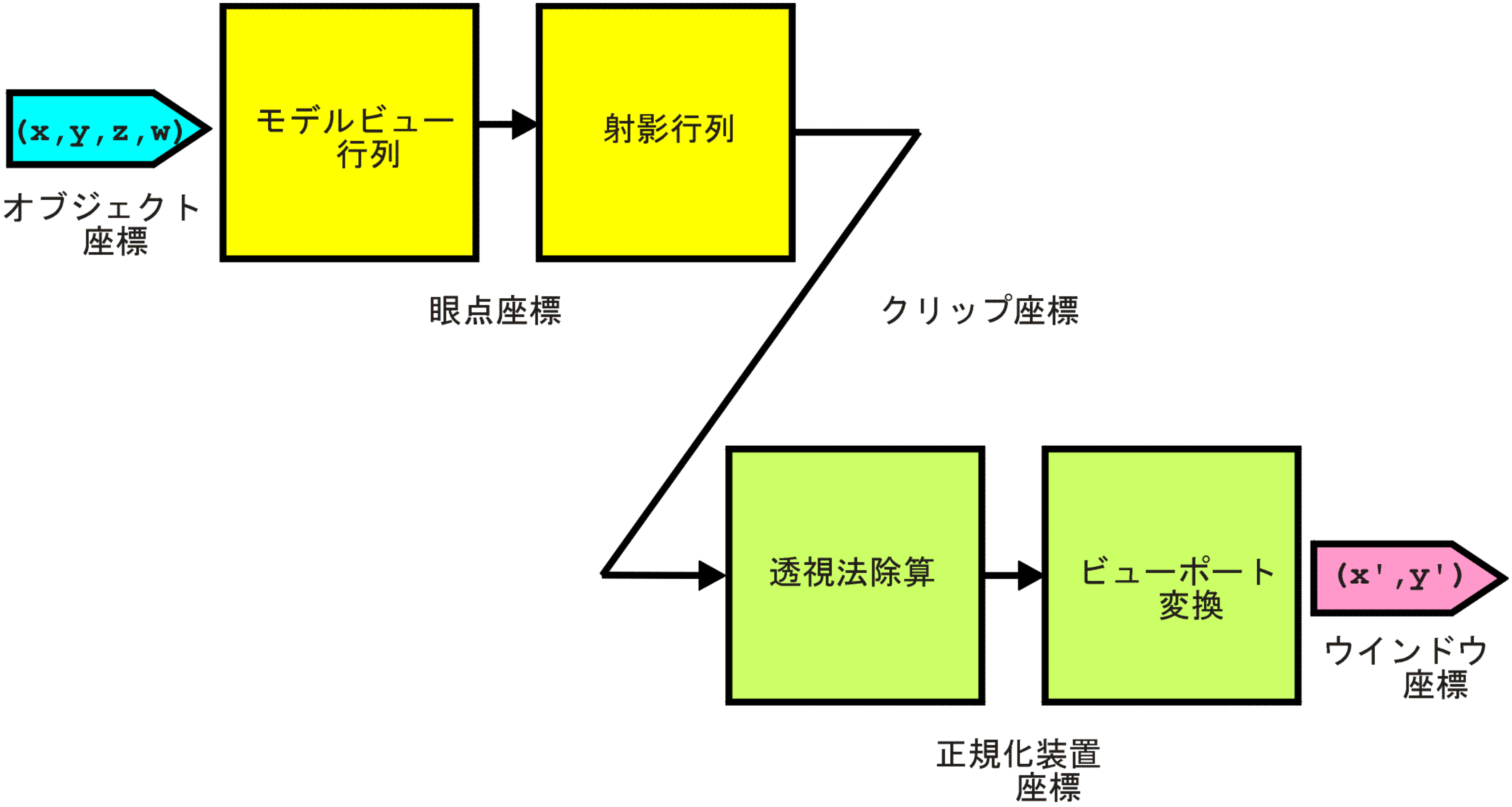
simpleProg.c(init())

```
void init(void)
{
  glClearColor ( 0.0, 0.0, 0.0, 0.0 ); /*背景色の指定*/
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
                                     /*描画のための投影法の指定*/
                                     /*正射影投影法          */
}
```

*初期状態ではカメラ位置は (0,0,0),

z軸の負の向きを向く. y軸が上方向.

頂点変換の手順

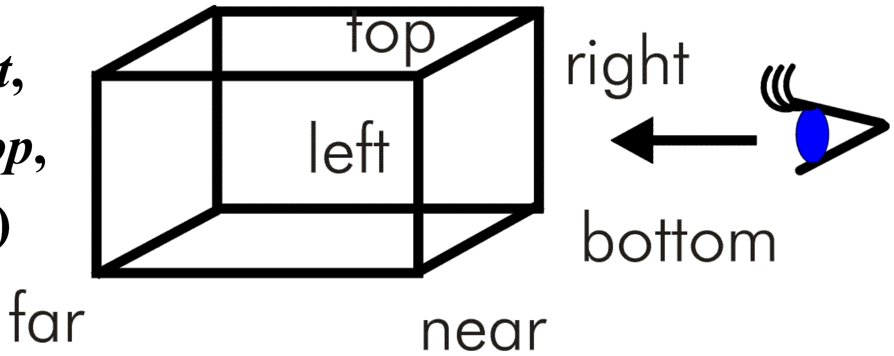

(x,y,z,w)
オブジェクト
座標
モデルビュー
行列
射影行列
眼点座標
クリップ座標
透視法除算
ビューポート
変換
(x',y')
ウインドウ
座標
正規化装置
座標

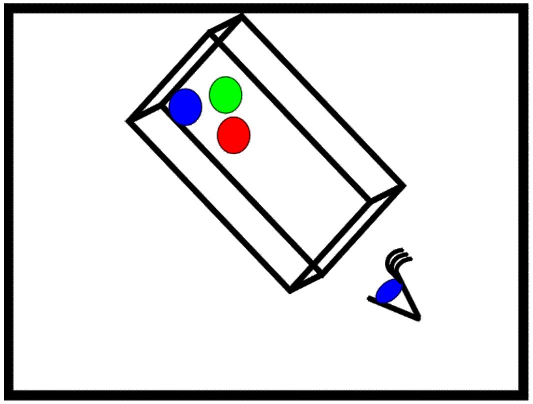
正射影変換

コマンド **glOrtho()**

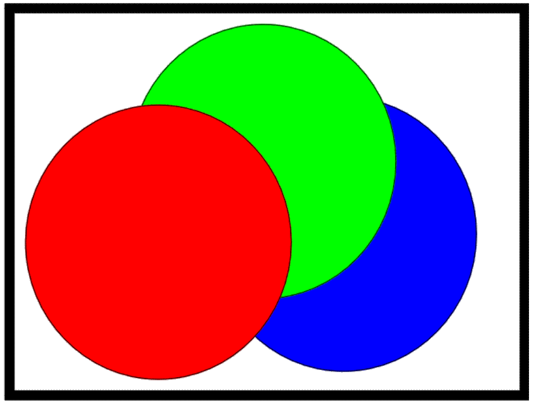
void

glOrtho(GLdouble *left*, Gldouble *right*,
Gldouble *bottom*, Gldouble *top*,
Gldouble *near*, Gldouble *far*)



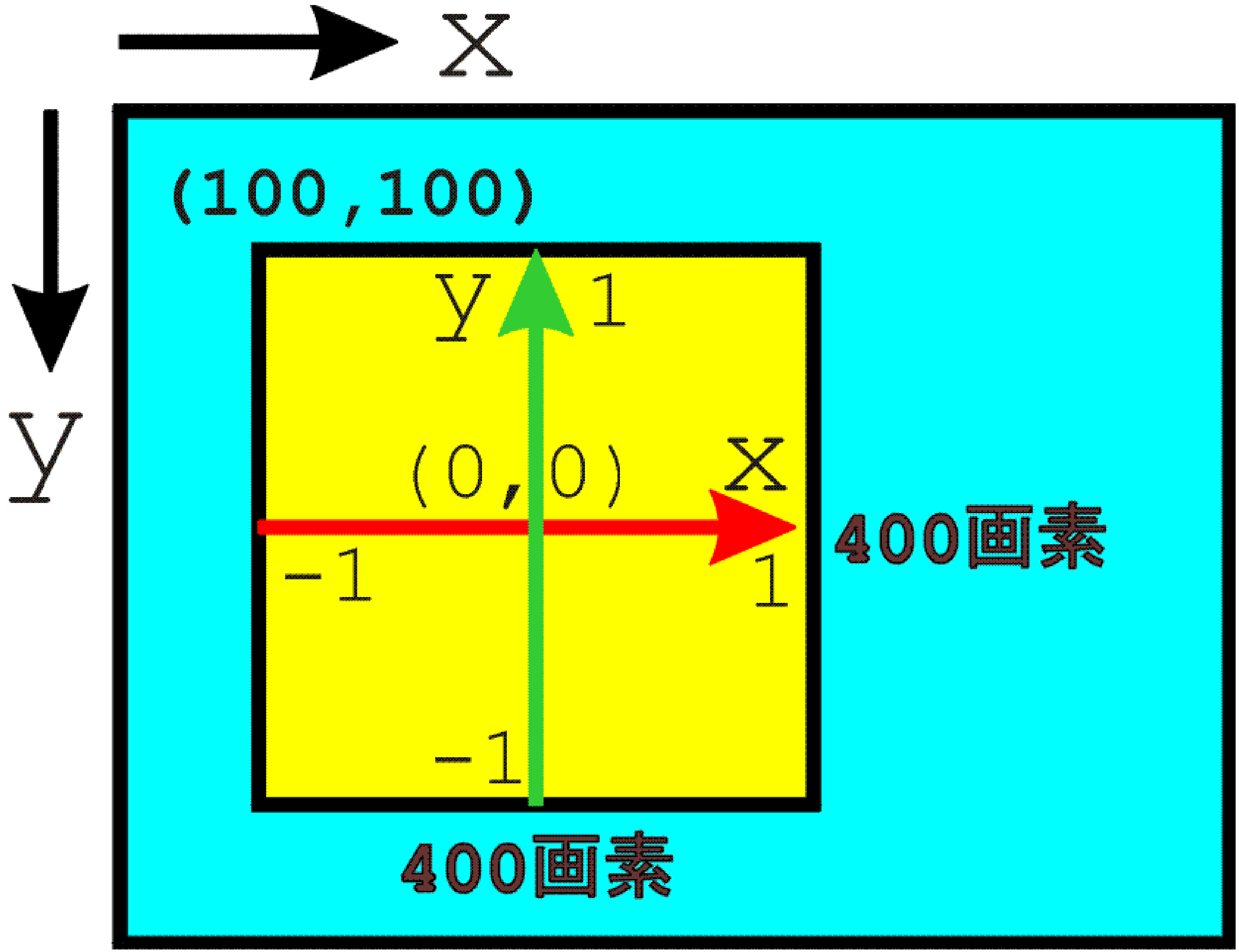


視体積の位置



視点からの図

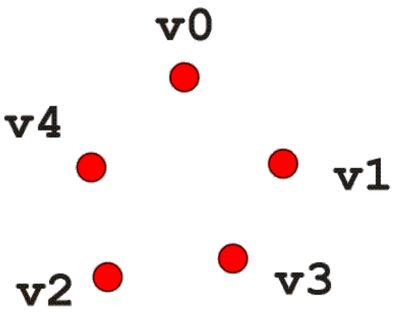
オブジェクト座標



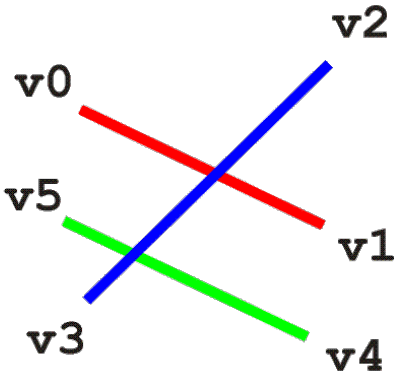
simpleProg.c(display())

```
void display(void) {
  glClear(GL_COLOR_BUFFER_BIT);/*背景のクリア          */
  glColor3f(1.0, 1.0, 1.0);     /*オブジェクトの色の指定*/
  glBegin(GL_POLYGON);          /*長方形の描画          */
    glVertex2f(-0.4, -0.2);
    glVertex2f(-0.4, 0.0);
    glVertex2f(-0.05, 0.0);
    glVertex2f(-0.05, -0.2);
  glEnd();
  glFlush();                    /*描画の強制*/
}
```

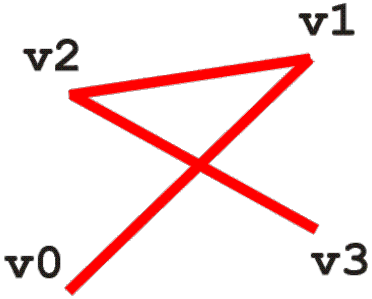
プリミティブの描画(その1)



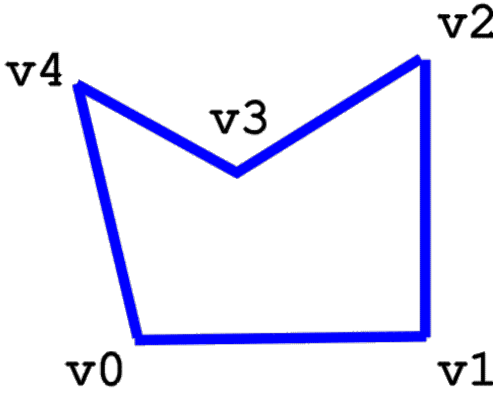
GL_POINTS



GL_LINES

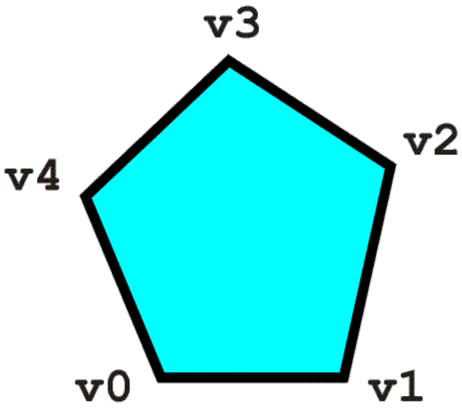


GL_LINE_STRIP

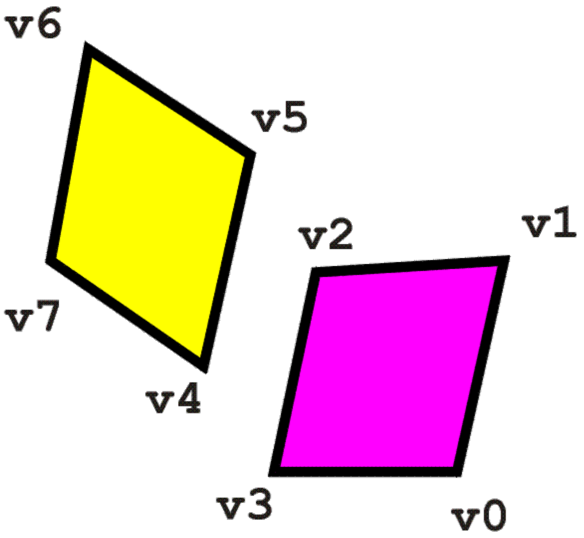


GL_LINE_LOOP

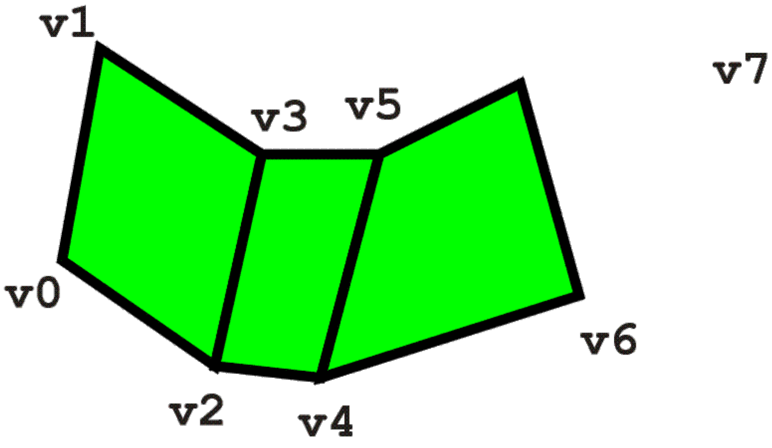
プリミティブの描画(その2)



GL_POLYGON

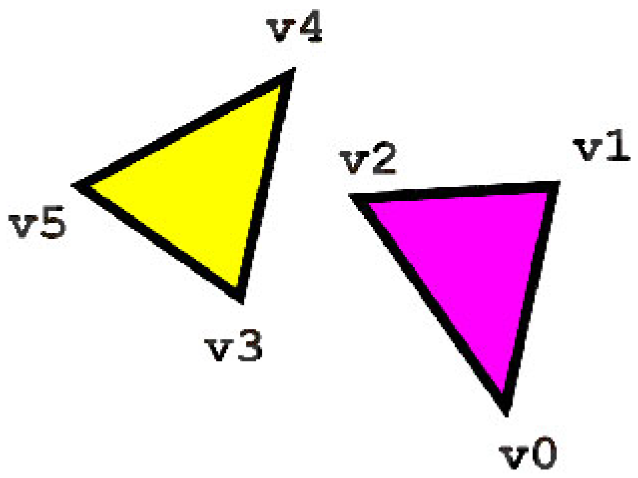


GL_QUADS

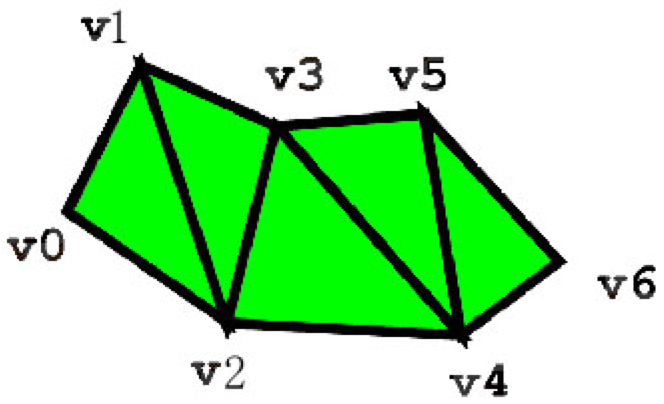


GL_QUAD_STRIP

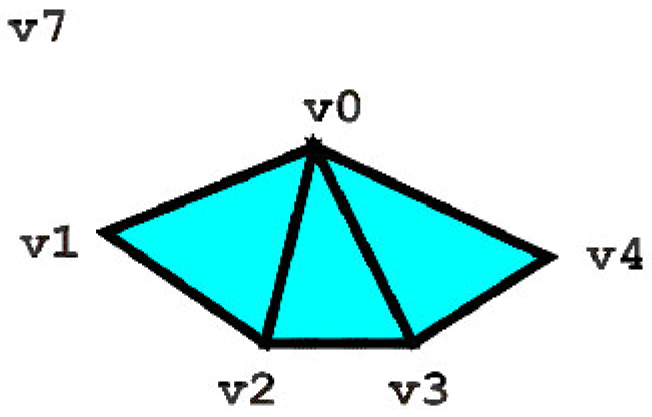
プリミティブの描画(その3)



GL_TRIANGLES



GL_TRIANGLE_STRIP



GL_TRIANGLE_FAN

ポリゴンの描画モード

コマンド glPolygonMode()

void

glPolygonMode(GLenum *face*, GLenum *mode*)

face **GL_FRONT_AND_BACK, GL_FRONT, GL_BACK**

mode **GL_POINT, GL_LINE, GL_FILL**

コマンド glFrontFace()

void

glFrontFace(GLenum *mode*)

mode **GL_CCW, GL_CW**

課題2 線の描画と四則演算

グラフィック

·OpenGLの利用

·線の描画(GL_LINES)

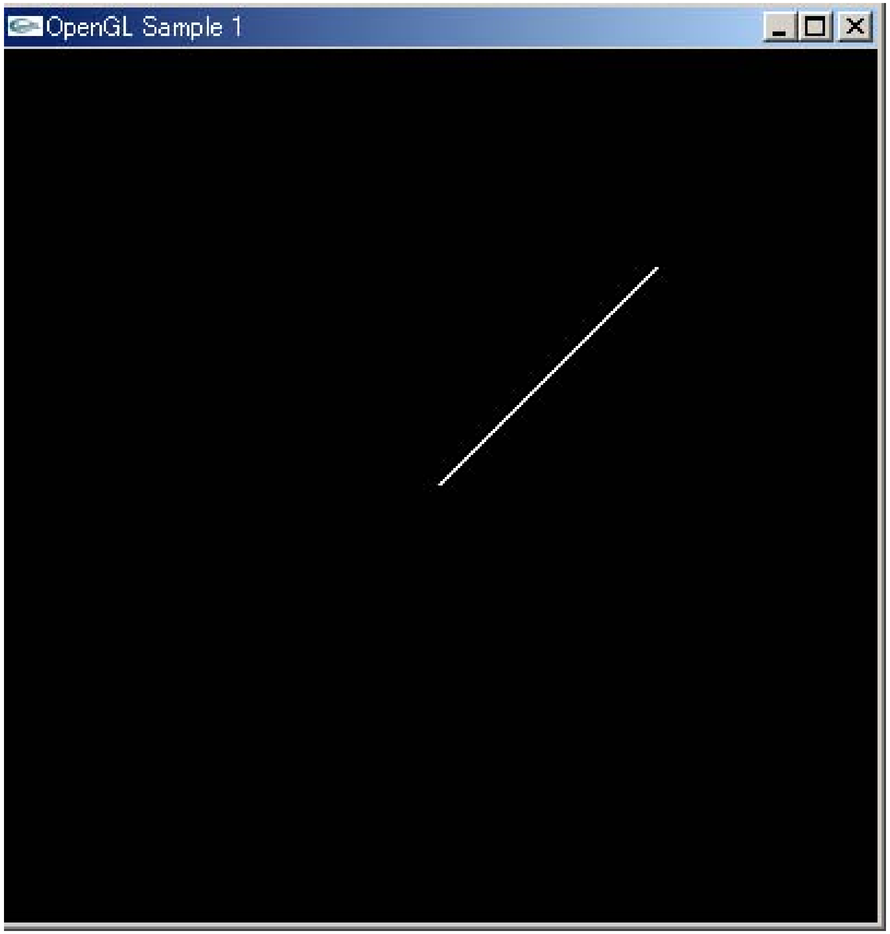
·ウィンドウ座標

·色の指定(glClearColor(), glColor())

·太さの指定(glLineWidth())

·四則演算
·和, 差, 積, 商, 余りの計算
·浮動小数型変数 (double 型)

課題2サンプルプログラム (実行結果)



exer2.c(main())

```
int main(int argc, char **argv){
  glutInit(&argc, argv);  /* GLUTの初期化  */
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
                              /* 表示モードの指定 */
  glutInitWindowSize(400, 400);
                              /* ウインドウサイズの指定  */
  glutInitWindowPosition(100, 100);
                              /* ウインドウの位置の指定  */
  glutCreateWindow("OpenGL Sample 2");
                              /* ウインドウのオープン    */
  init();                      /* 初期化処理 */
  glutDisplayFunc(display);
                               /* 表示の関数の指定 */
  glutMainLoop();              /* GLUTのメインループ    */
  return 0;
}
```

exer2.c(init())

```
void init(void)
{
  glClearColor ( 0.0, 0.0, 0.0, 0.0 ); /*背景色の指定*/
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glOrtho(-1.0, 1.0, -1.0, 1.0, -1.0, 1.0);
                                /*描画のための投影法の指定*/
                                /*正射影投影法           */
}
```

*初期状態ではカメラ位置は (0,0,0),

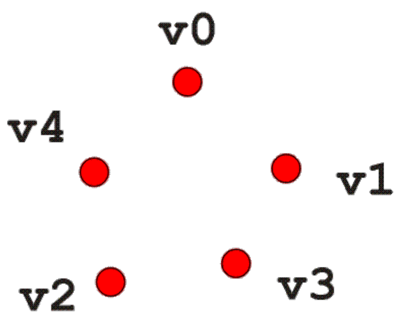
z軸の負の向きを向く. y軸が上方向.

exer2.c(display())

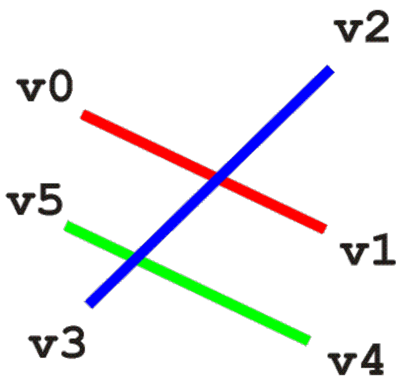
```
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);/* 背景のクリア */
  glBegin(GL_LINES);            /* 線分を描画する */
    glVertex2f(0.0, 0.0);       /* 始点の指定 */
  glVertex2f(0.5, 0.5);         /* 終点の指定 */
  glEnd();                       /* 線分の描画終了*/
  glFlush();                    /* 画面を再描画する */
}
```

線分の描画

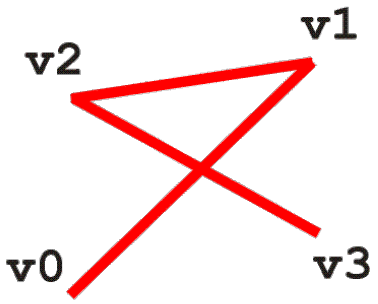
プリミティブの描画(その1)　再掲



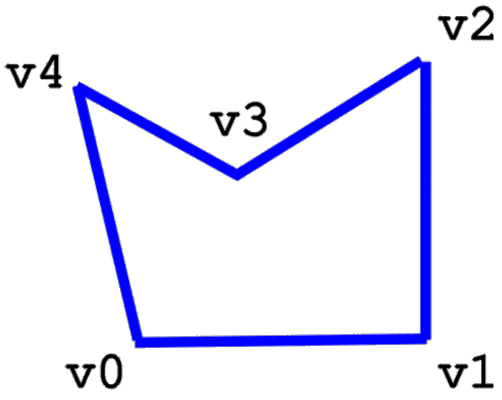
GL_POINTS



GL_LINES



GL_LINE_STRIP



GL_LINE_LOOP

まとめ

- グラフィックス
  - 線の描画
  - ウインドウ座標
  - 色の指定
  - 太さの指定

- 四則演算
- 和, 差, 積, 商, 余りの計算
- 浮動小数型変数 (double 型)