

第9回 卒業研究 1

2010.11.23

田中弘美研究室 研究紹介

研究グループ

- デジタルミュージアム
- ボリュームビジュアライゼーション
- ITプラント
- ハプティックビジョン

DM班の紹介

VV班の紹介

その他の班紹介

OpenGL課題

OpenGL課題について

- 目的
 - OpenGLを使い、画像処理技術について学び、プログラミング能力の向上と卒業研究を行うための基礎知識を身につける。
- 評価
 - 講義の全出席と全2回のレポートで評価
- サポートページ
 - <http://www.cv.ci.ritsumei.ac.jp/haptic/support.html>

OpenGL課題スケジュール

- 今後のスケジュール（予定）
 - 第 9回（11月23日）班紹介・OpenGL課題の導入
 - 第10回（11月30日）マウス・キーボード入力
 - 第11回（12月 7日）三次元図形の描画
 - 第12回（12月14日）アニメーション
 - 第13回（12月21日）隠面消去処理 ☆
 - 第14回（ 1月11日）陰影付け
 - 第15回（ 1月18日）メタセコイアの概要 ☆

☆レポート提出日

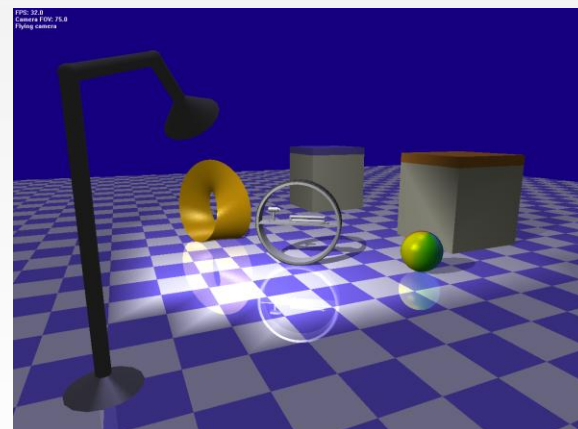
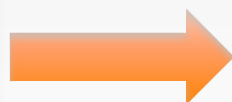
OpenGLとは？

- OpenGL (Open Graphics Library)
 - 3Dグラフィックスのためのライブラリ
 - OpenGLの拡張ライブラリ
 - OpenGL ES : iOSやAndroidのアプリ作成



```
Terminal — zsh — 56x13
s089%
s089%
s089%
s089% ./a.out
<分布1のモデルパラメータ>
平均(x,y) : ( 107.009, 1149.85 )
分散(x,y,xy) : ( 8.84695, 884.279, -0.490067 )
重み(m) : ( 0.59693 )
<分布2のモデルパラメータ>
平均(x,y) : ( 111.968, 1100.26 )
分散(x,y,xy) : ( 0.984497, 97.3801, 0.298754 )
重み(m) : ( 0.40307 )
s089%
```

CUIベースのプログラム



OpenGLプログラム

OpenGLを使えば簡単にCGを作成することができる！

OpenGLを使ってみる

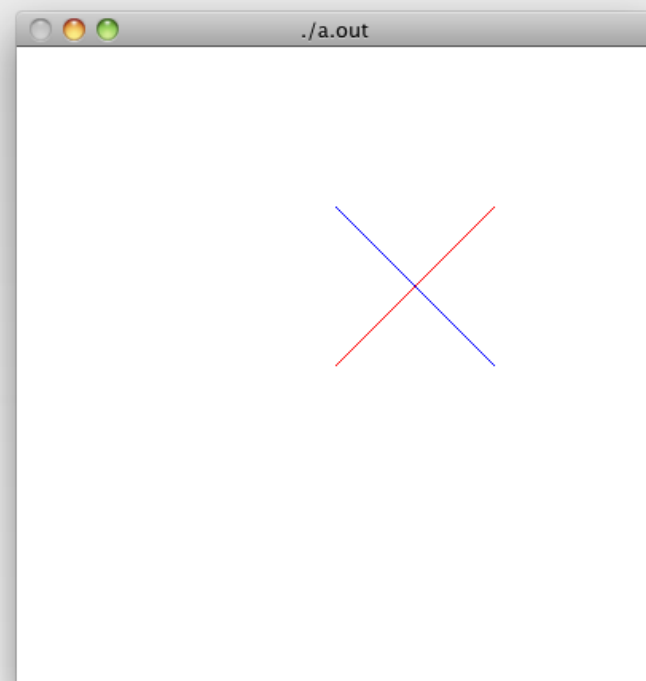
1. サポートページからプログラムをダウンロード
<http://www.cv.ci.ritsumei.ac.jp/haptic/support.html>
2. ソリューションファイル（sln）を開き、サポートページに記述してあるプロジェクトの設定を行う
3. プログラムのビルド、実行

二次元図形の描画例①

●例：2本の線を描画

```
//描画処理関数
void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT); //ウィンドウ
    を背景色で塗りつぶし

    //1本目
    glColor3d(1.0,0.0,0.0); //赤色
    glBegin(GL_LINES);
    glVertex2d(0.0,0.0); //始点
    glVertex2d(50.0,50.0); //終点
    glEnd();
    //2本目
    glColor3d(0.0,0.0,1.0); //青色
    glBegin(GL_LINES);
    glVertex2d(0.0,50.0); //始点
    glVertex2d(50.0,0.0); //終点
    glEnd();
    glFlush();
}
```



プログラム実行画面

二次元図形描画の基本(1/2)

```
void display(void){
```

```
    . . .
```

```
    glColor3d(1.0,0.0,0.0); //赤色
```

```
    glBegin(GL_LINES);
```

```
    glVertex2d(0.0,0.0); //始点
```

```
    glVertex2d(50.0,50.0); //終点
```

```
    glEnd();
```

```
    . . .
```

```
}
```

display() :

描画処理のメインとなる関数

glColor3d(R, G ,B) :

描画する物の色をRGBで指定

glBegin(Type) ~ glEnd() :

引数Typeで図形のタイプを指定し、glBegin()~glEnd() の間に描画する図形の各頂点の座標値を設定

glVertex2d(X, Y) :

描画する図形の二次元の座標値を指定

二次元図形描画の基本(2/2)

- 図形のタイプ
 - glBegin(GL_POINTS) ~ glEnd()
指定した座標(複数可)に点を描画
 - glBegin(GL_LINES) ~ glEnd()
指定した2点を結ぶ直線を描画
 - glBegin(GL_LINE_STRIP) ~ glEnd()
指定した点を結ぶ折れ線を描画
 - glBegin(GL_TRIANGLES) ~ glEnd()
指定した3点を頂点にもつ三角形を描画
 - glBegin(GL_POLYGON) ~ glEnd()
指定した点を頂点にもつ凸多角形を描画

二次元図形の描画例②

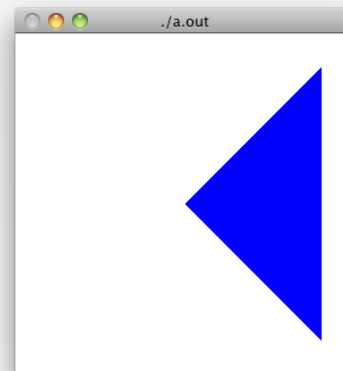
- 折れ線の描画

```
glColor3d(1.0,0.0,0.0); //赤
glBegin(GL_LINE_STRIP);
glVertex2d(0.0,0.0); //点1
glVertex2d(20.0,20.0); //点2
glVertex2d(40.0,-20.0); //点3
glVertex2d(60.0,40.0); //点4
glVertex2d(80.0,-80.0); //点5
glEnd();
```



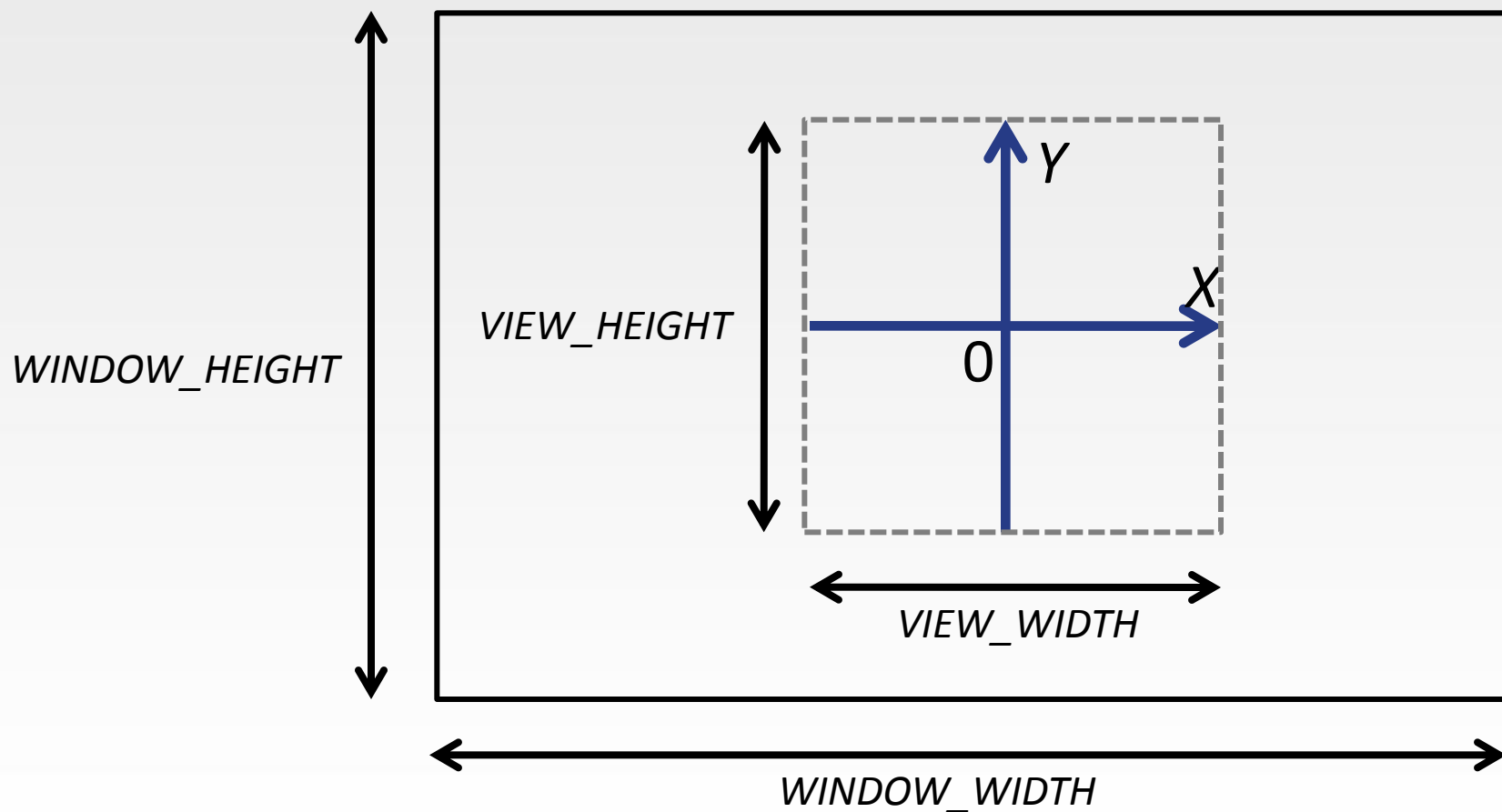
- 三角形の描画

```
glColor3d(0.0,0.0,1.0); //青
glBegin(GL_TRIANGLES);
glVertex2d(0.0,0.0); //点1
glVertex2d(80,80); //点2
glVertex2d(80,-80); //点3
glEnd();
```



OpenGLの座標系（２次元）

ウィンドウ



実習課題

- 課題 1

- 座標軸を描画する関数drawAxis()を作成し、display関数で呼び出さない

- 課題 2

- GL_POLYGONを使って多角形の図形を生成しない

- 課題 3

- 生成した図形の拡大・縮小を行いなさい。また、生成した図形を原点中心に回転を行いなさい

* 第13回（12月21日）に第9回から第12回までの課題をレポートとして提出してもらいます

課題の実行例

