計算機実験　2
～　OpenGLの使い方　～
6月24日

高田　雅美@E359
takata@ics.nara-wu.ac.jp

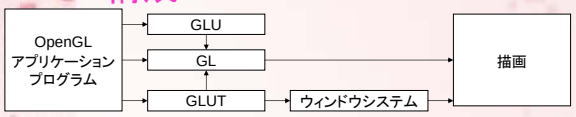
これからの予定

- 6月24日　課題1＆2
- 7月1日　課題3＆4
- 7月8日　課題5＆6
- 7月15日　課題7＆8
- 7月29日　課題9＆10
- 8月5日　課題11＆12＆13
  - 課題は，講義中に確認
  - 8月5日までに，全部を見せてくれればOK
  - 課題1～10：各7点，課題11～13：各10点

OpenGLとは

- Open Graphics Library
- 3次元CGの高度な描画機能を保有
- ライセンス料　無料
- ソースコードの公開
- 利用可能なプラットフォーム
  - Windows, Mac, Unix系

構成



- GL
  - 基本的な描画コマンド
  - ヘッダーファイル：GL/gl.h，オプション：-lGL
- GLU
  - OpenGLユーティリオティライブラリ
  - GLの上位ライブラリ
  - ヘッダーファイル：GL/glu.h，オプション：-lGLU
- GLUT
  - ウィンドウの制御，イベントの処理
  - ヘッダーファイル：GL/glut.h，オプション：-lglut

関数名

（接頭語）（コマンド名）（接尾語）（引数）

- 接頭語
  - ライブラリの種類
- コマンド名
- 接尾語
  - 数字：引数の数
  - アルファベット
    - f:float, d:double, i:int(long), s:Int(short), v:vector

```
#include <GL/glut.h>
#include <GL/gl.h>
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
}
void myKbd(unsigned char key, int x, int y){
  if( key == 27 ) exit(0);
}
void myInit( char *progname){
  int width = 500, height = 500;
  glutInitWindowPosition( 0, 0);
  glutInitWindowSize( width, height );
  glutInitDisplayMode( GLUT_RGBA );
  glutCreateWindow( progname);
  glClearColor( 0.0, 0.0, 0.0, 1.0 );
  glutKeyboardFunc( myKbd );
}
int main( int argc, char **argv ){
  glutInit( &argc, argv );
  myInit( argv[0] );
  glutDisplayFunc( display );
  glutMainLoop();
  return( 0 );
}
```

ウィンドウの表示

```
#include <GL/glut.h>
#include <GL/gl.h>
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
}
void myKbd(unsigned char key, int x, int y){
  if( key == 27 ) exit(0);
}
void myInit( char *progname){
  int width = 500, height = 500;
  glutInitWindowPosition( 0, 0);
  glutInitWindowSize( width, height );
  glutInitDisplayMode( GLUT_RGBA );
  glutCreateWindow( progname);
  glClearColor( 0.0, 0.0, 0.0, 1.0 );
  glutKeyboardFunc( myKbd );
}
int main( int argc, char **argv ){
  glutInit( &argc, argv );
  myInit( argv[0] );
  glutDisplayFunc( display );
  glutMainLoop();
  return( 0 );
}
```

main関数

- glutの初期化
- 描画ウィンドウの設定
- 描画方法の指定
  - 描画する物体
- イベント待ち
  - ウィンドウを維持

```
#include <GL/glut.h>
#include <GL/gl.h>
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
}
void myKbd(unsigned char key, int x, int y){
  if( key == 27 ) exit(0);
}
void myInit( char *progname){
  int width = 500, height = 500;
  glutInitWindowPosition( 0, 0);
  glutInitWindowSize( width, height );
  glutInitDisplayMode( GLUT_RGBA );
  glutCreateWindow( progname);
  glClearColor( 0.0, 0.0, 0.0, 1.0 );
  glutKeyboardFunc( myKbd );
}
int main( int argc, char **argv ){
  glutInit( &argc, argv );
  myInit( argv[0] );
  glutDisplayFunc( display );
  glutMainLoop();
  return( 0 );
}
```

ウィンドウの設定

- 左上の位置
- サイズ
  - ドット数
- 描画方法
  - 色のつけ方（赤緑青）
- ウィンドウの名前
- 背景色
  - （赤，緑，青，透明度）
- イベントの設定
  - キー入力

ウィンドウの表示位置の計算

- 横のドット
  - int glutGet(GLUT_SCREEN_WIDTH);
- 縦のドット
  - int glutGet(GLUT_SCREEN_HEIGHT);

```
#include <GL/glut.h>
#include <GL/gl.h>
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
}
void myKbd(unsigned char key, int x, int y){
  if( key == 27 ) exit(0);
}
void myInit( char *progname){
  int width = 500, height = 500;
  glutInitWindowPosition( 0, 0);
  glutInitWindowSize( width, height );
  glutInitDisplayMode( GLUT_RGBA );
  glutCreateWindow( progname);
  glClearColor( 0.0, 0.0, 0.0, 1.0 );
  glutKeyboardFunc( myKbd );
}
int main( int argc, char **argv ){
  glutInit( &argc, argv );
  myInit( argv[0] );
  glutDisplayFunc( display );
  glutMainLoop();
  return( 0 );
}
```

標準キー入力

- key:キー情報
- x, y:マウスポインタ
- 文字キー
- ASCIIコード（10進数）
  - ESC: 27
  - space: 32
  - return(enter): 13
  - TAB: 9
  - DEL: 16
  - BS: 8

```
#include <GL/glut.h>
#include <GL/gl.h>
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glFlush();
}
void myKbd(unsigned char key, int x, int y){
  if( key == 27 ) exit(0);
}
void myInit( char *progname){
  int width = 500, height = 500;
  glutInitWindowPosition( 0, 0);
  glutInitWindowSize( width, height );
  glutInitDisplayMode( GLUT_RGBA );
  glutCreateWindow( progname);
  glClearColor( 0.0, 0.0, 0.0, 1.0 );
  glutKeyboardFunc( myKbd );
}
int main( int argc, char **argv ){
  glutInit( &argc, argv );
  myInit( argv[0] );
  glutDisplayFunc( display );
  glutMainLoop();
  return( 0 );
}
```

描画物体

- ウィンドウの初期化
  - 背景色を塗る
- 描画終了

課題1

- 縦　280ドット
- 横　700ドット
- 画面の中央に表示
- 背景色　すみれ色
- 画面の終了　DELキー

```
        :
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glLineWidth( 8.0 );
  glColor3f( 1.0, 1.0, 1.0);
  glBegin(GL_POINTS);
    glVertex3f( 0.0, 0.0, 0.0 );
  glEnd();
  glFlush();
}
        :
void myInit( char *progname){
        :
  glutKeyboardFunc( myKbd );
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();
  glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 1.0 );
}
        :
```

ウィンドウの表示

基本図形の描画

```
        :
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glLineWidth( 8.0 );
  glColor3f( 1.0, 1.0, 1.0);
  glBegin(GL_POINTS);
    glVertex3f( 0.0, 0.0, 0.0 );
  glEnd();
  glFlush();
}
        :
void myInit( char *progname){
        :
  glutKeyboardFunc( myKbd );
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();
  glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 1.0 );
}
        :
```

glMatrixMode(・・・)

- スクリーン上の座標系の設定
  - GL_PROJECTION
    - 投影変換を利用
  - GL_MODELVIEW
    - 幾何変換を利用

glLoadIdentity()

- 座標の初期化

glOrtho(・・・)

- 平行投影
- 表示する座標の範囲
  - xyzの最小，最大

```
        :
void display(void){
  glClear(GL_COLOR_BUFFER_BIT);
  glLineWidth( 8.0 );
  glColor3f( 1.0, 1.0, 1.0);
  glBegin(GL_POINTS);
    glVertex3f( 0.0, 0.0, 0.0 );
  glEnd();
  glFlush();
}
        :
void myInit( char *progname){
        :
  glutKeyboardFunc( myKbd );
  glMatrixMode( GL_PROJECTION );
  glLoadIdentity();
  glOrtho( -1.0, 1.0, -1.0, 1.0, -1.0, 1.0 );
}
        :
```

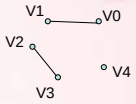
図形の描画

- glLineWidth
  - 線分幅
- glColor3f(・・・)
  - 図形の頂点の色指定
  - 各頂点ごとに指定可能
    - 直前の呼び出しが有効
    - 頂点間の色は補間
- glBegin(・・・)
  - 描画開始
  - 引数：描画図形の指定
- glVertex・・・
  - 頂点座標

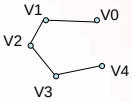
描画図形（点と直線）

- GL_POINTS

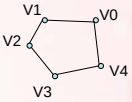
- GL_LINES



- GL_LINE_STRIP

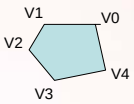


- GL_LOOP

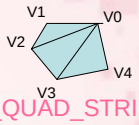


描画図形（面と直線）

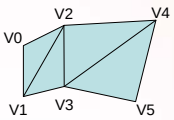
- GL_POLYGON

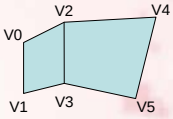


- GL_TRIANGLE_FAN



- GL_QUAD_STRIP
- GL_TRIANGLE_STRIP





課題2

- 背景は白色
- X軸とY軸を描く（黒色）
- 各軸の最小値　-100
- 各軸の最大値　100
- $Y_1=X^2/10$　-30　を描く（青）
- $Y_2=2/(X+2)$　+60　を描く（赤）
- $Y_2 > Y_1$ の範囲を塗りつぶす（緑）