計算機実験　2
～　OpenGLの使い方　～
7月29日

高田　雅美@E359
takata@ics.nara-wu.ac.jp

```
                :
unsigned char revolveFlag = GL_FALSE;
int xBegin, yBegin;
                :
void myMouse( int button, int state, int x, int y ){
  if(state == GLUT_DOWN ) {
    if( button == GLUT_LEFT_BUTTON ){
      revolveFlag = !revolveFlag;
      if( revolveFlag == GL_TRUE )
        glutIdleFunc( idle );
      else
        glutIdleFunc( NULL );
    }
  xBegin = x;
  yBegin = y;
  }
}
                :
Int myInit(char *progname){
                :
  glutSpecialFunc( mySkbd );
  glutMouseFunc( myMouse );
  glMatrixMode( GL_PROJECTION );
                :
```

特殊キー割り込み

クリック

```
                :
unsigned char revolveFlag = GL_FALSE;
int xBegin, yBegin;
                :
void myMouse( int button, int state, int x, int y ){
  if(state == GLUT_DOWN ) {
    if( button == GLUT_LEFT_BUTTON ){
      revolveFlag = !revolveFlag;
      if( revolveFlag == GL_TRUE )
        glutIdleFunc( idle );
      else
        glutIdleFunc( NULL );
    }
  xBegin = x;
  yBegin = y;
  }
}
                :
Int myInit(char *progname){
                :
  glutSpecialFunc( mySkbd );
  glutMouseFunc( myMouse );
  glMatrixMode( GL_PROJECTION );
                :
```

イベントの設定

- マウスの動作

```
                :
unsigned char revolveFlag = GL_FALSE;
int xBegin, yBegin;
                :
void myMouse( int button, int state, int x, int y ){
  if(state == GLUT_DOWN ) {
    if( button == GLUT_LEFT_BUTTON ){
      revolveFlag = !revolveFlag;
      if( revolveFlag == GL_TRUE )
        glutIdleFunc( idle );
      else
        glutIdleFunc( NULL );
    }
  xBegin = x;
  yBegin = y;
  }
}
                :
Int myInit(char *progname){
                :
  glutSpecialFunc( mySkbd );
  glutMouseFunc( myMouse );
  glMatrixMode( GL_PROJECTION );
                :
```

- 引数
  - ボタンの種類
    - 左ボタン　GLUT_LEFT_BUTTON
    - 中ボタン　GLUT_MIDDLE_BUTTON
    - 右ボタン　GLUT_RIGHT_BUTTON
  - 状態
    - オン：GLUT_DOWN
    - オフ：GLUT_UP
  - ポイント
- glutIdleFunc(NULL)
  - 動作をとめる

課題9

- 平行投影
- 左クリック地点を画面中心に変更
  - 物体は，元の位置
    →中心が変わるので，物体が動く
- 右クリックで，描画物体を画面の中心に表示

```
                :
float xDist = 0.0, yDist = 0.0;
void display( void ){
                :
  glPushMatrix();
  glTranslatef( xDist,  yDist , zOrg );
  glEnable( GL_DEPTH_TEST );
                :
}
                :
void myMotion( int x, int y ){
  xDist += (float)( x-xBegin )/250;
  yDist += -(float)( y-yBegin )/250;
  xBegin = x;
  yBegin = y;
  glutPostRedisplay();
}
                :
Int myInit(char *progname){
                :
  glutMouseFunc( myMouse );
  glutMotionFunc( myMotion );
  glMatrixMode( GL_PROJECTION );
                :
```

クリック

ドラッグ

```
                :
Float xDist = 0.0, yDist = 0.0;
void display( void ){
                :
  glPushMatrix();
  glTranslatef( xDist,  yDist , zOrg );
  glEnable( GL_DEPTH_TEST );
                :
}
                :
void myMotion( int x, int y ){
  xDist += (float)( x-xBegin )/250;
  yDist += -(float)( y-yBegin )/250;
  xBegin = x;
  yBegin = y;
  glutPostRedisplay();
}
                :
Int myInit(char *progname){
                :
  glutMouseFunc( myMouse );
  glutMotionFunc( myMotion );
  glMatrixMode( GL_PROJECTION );
                :
```

イベント設定

- ドラッグ
- 引数を実行

```
                :
Float xDist = 0.0, yDist = 0.0;
void display( void ){
                :
  glPushMatrix();
  glTranslatef( xDist,  yDist , zOrg );
  glEnable( GL_DEPTH_TEST );
                :
}
                :
void myMotion( int x, int y ){
  xDist += (float)( x-xBegin )/250;
  yDist += -(float)( y-yBegin )/250;
  xBegin = x;
  yBegin = y;
  glutPostRedisplay();
}
                :
Int myInit(char *progname){
                :
  glutMouseFunc( myMouse );
  glutMotionFunc( myMotion );
  glMatrixMode( GL_PROJECTION );
                :
```

イベント時に変動

- 図の平行移動

```
                :
Float xDist = 0.0, yDist = 0.0;
void display( void ){
                :
  glPushMatrix();
  glTranslatef( xDist,  yDist , zOrg );
  glEnable( GL_DEPTH_TEST );
                :
}
                :
void myMotion( int x, int y ){
  xDist += (float)( x-xBegin )/250;
  yDist += -(float)( y-yBegin )/250;
  xBegin = x;
  yBegin = y;
  glutPostRedisplay();
}
                :
Int myInit(char *progname){
                :
  glutMouseFunc( myMouse );
  glutMotionFunc( myMotion );
  glMatrixMode( GL_PROJECTION );
                :
```

ドラッグ時のイベント

- x-xBegin
  - ドラッグした距離
- /250
  - 軸は-1～1
  - 画面サイズは500ドット
- yDist
  - 画面
    - 上のドット数が小さい
  - y軸
    - 上の方が大きい

課題10

- 平行投影
- 三角と四角を重ならないように描画
- 右にドラッグ→三角が右回転
- 左にドラッグ→三角が左回転
- 上にドラッグ→四角の拡大
- 下にドラッグ→四角の縮小

- 右クリック
  - 三角：回転を止める
  - 四角：元のサイズに戻す

課題11

- ライフゲームの作成
  - 2次元（100*100）
  - 初期値はランダム
  - 左クリック　生命の誕生
  - 右クリック　死

課題12

- 衝突振り子（糸でつるされた連振り子）
  - 振り子の数：10個
  - 左クリック：動く玉の数を増やす
  - 右クリック：停止

課題13

- 拡散方程式を用いたシミュレーション
  - 高須先生＆瀬戸先生に習った知識をフル活用
  - 3次元でシミュレーション
  - タイムステップごとに描画