

計算機実験 2

～ OpenGLの使い方 ～

7月15日

高田 雅美@E359
takata@ics.nara-wu.ac.jp

```

float dist = 0.0;
void display(void){
    glClearColor( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glPushMatrix();
    gluLookAt( 0.0, 1.0, dist, 0.0, 1.0, dist+1.0, 0.0, 1.0, 0.0 );
    glPushMatrix();
    :
    glPopMatrix();
    glPopMatrix();
    glutSwapBuffers();
}
void idle( void){
    theta = fmod( theta + 0.5, 360 );
    dist = -fmod( -dist +0.05, 32.0);
    glutPostRedisplay();
}
:
:
:
    
```

アニメーション

ウォークスルー

```

float dist = 0.0;
void display(void){
    glClearColor( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glPushMatrix();
    gluLookAt( 0.0, 1.0, dist, 0.0, 1.0, dist+1.0, 0.0, 1.0, 0.0 );
    glPushMatrix();
    :
    glPopMatrix();
    glPopMatrix();
    glutSwapBuffers();
}
void idle( void){
    theta = fmod( theta + 0.5, 360 );
    dist = -fmod( -dist +0.05, 32.0);
    glutPostRedisplay();
}
:
:
:
    
```

ウォークスルー用変数
➤画面の移動

```

float dist = 0.0;
void display(void){
    glClearColor( GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT );
    glPushMatrix();
    gluLookAt( 0.0, 1.0, dist, 0.0, 1.0, dist+1.0, 0.0, 1.0, 0.0 );
    glPushMatrix();
    :
    glPopMatrix();
    glPopMatrix();
    glutSwapBuffers();
}
void idle( void){
    theta = fmod( theta + 0.5, 360 );
    dist = -fmod( -dist +0.05, 32.0);
    glutPostRedisplay();
}
:
:
:
    
```

ウォークスルー

- 一番最初にスタックへ
- 最後にPopされる
- 他の描画と独立
- gluLookAt(・・・)
- 視点位置座標(3次元)
- 注視点座標(3次元)
- 画面の上方向ベクトル

課題7

- 雪だるまの作成
- 前後に振動
- 近づく遠くのの繰り返し

```

float zOrg = -5.0;
void display(void){
    glPushMatrix();
    glTranslatef(0.0, 0.0, zOrg );
    glEnable( GL_DEPTH_TEST );
    :
    :
}
void myKey( int key, int x, int y ){
    switch( key ){
        case GLUT_KEY_UP:    zOrg += 0.2;
            break;
        case GLUT_KEY_DOWN: zOrg -= 0.2;
            break;
    }
    glutPostRedisplay();
}
:
:
:
int main( int argc, char **argv ){
    :
    :
    glutKeyboardFunc( myKbd );
    glutSpecialFunc( myKey );
    glMatrixMode( GL_PROJECTION );
}
    
```

アニメーション

特殊キー割り込み

```
float zOrg = -5.0;
void display(void){
    glPushMatrix();
    glTranslatef(0.0, 0.0, zOrg );
    glEnable( GL_DEPTH_TEST );
}

void mySkey( int key, int x, int y ){
    switch( key ){
        case GLUT_KEY_UP:    zOrg += 0.2;
            break;
        case GLUT_KEY_DOWN:  zOrg -= 0.2;
            break;
    }
    glutPostRedisplay();
}

int main( int argc, char **argv ){
    glutKeyboardFunc( myKbd );
    glutSpecialFunc( mySkey );
    glMatrixMode( GL_PROJECTION );
}
```

キー入力に対する変数
 > 大域変数

```
float zOrg = -5.0;
void display(void){
    glPushMatrix();
    glTranslatef(0.0, 0.0, zOrg );
    glEnable( GL_DEPTH_TEST );
}

void mySkey( int key, int x, int y ){
    switch( key ){
        case GLUT_KEY_UP:    zOrg += 0.2;
            break;
        case GLUT_KEY_DOWN:  zOrg -= 0.2;
            break;
    }
    glutPostRedisplay();
}

int main( int argc, char **argv ){
    glutKeyboardFunc( myKbd );
    glutSpecialFunc( mySkey );
    glMatrixMode( GL_PROJECTION );
}
```

イベントの設定
 > 特殊キー入力
 ● 引数を実行

```
float zOrg = -5.0;
void display(void){
    glPushMatrix();
    glTranslatef(0.0, 0.0, zOrg );
    glEnable( GL_DEPTH_TEST );
}

void mySkey( int key, int x, int y ){
    switch( key ){
        case GLUT_KEY_UP:    zOrg += 0.2;
            break;
        case GLUT_KEY_DOWN:  zOrg -= 0.2;
            break;
    }
    glutPostRedisplay();
}

int main( int argc, char **argv ){
    glutKeyboardFunc( myKbd );
    glutSpecialFunc( mySkey );
    glMatrixMode( GL_PROJECTION );
}
```

> 引数
 ● key: 押されたキー
 ● x, y: マウスポインタ
 > 特殊キー
 ● 矢印キー
 ◆ GLUT_KEY_LEFT
 ◆ GLUT_KEY_RIGHT
 ◆ GLUT_KEY_UP
 ◆ GLUT_KEY_DOWN
 ● ファンクションキー
 ◆ GLUT_KEY_F*
 > glutPostRedisplay
 ● キー操作を反映

課題8

- > 「R」入力で回転速度を速める
- > 「r」入力で回転速度を遅くする
- > 「F」入力で前進
- > 「B」入力で後退
- > 矢印キーで移動