

Comp380 Report

Programming Assignment #2

Due Apr.-5 (Wed.) (before 11:59pm)

Name: In-Young Cho (조인영)

ID : 20150720

Objective: Understand how to perform transformations to the model, idle-based animations, and simple interaction methods.

● 일단 cpp파일에서 //edit ~ //end라고 묶여져 있는 부분은 모두 제가 수정한 코드입니다.

```
#include <cstdlib> // for using rand function
```

```
#include <ctime>
```

```
int onX = 1; int onY = 0; int onZ = 0; // K키가 눌리지면 onK는 1, 아니라면 0. K = X, Y, Z 중 하나. 일단 처음에는 x로 방향으로 움직이는게 default
```

```
int tmpOnX = onX; int tmpOnY = onY; int tmpOnZ = onZ; // r 키가 눌리졌을 때 onK의 값을 tmpOnK에 임시저장하고 onK는 0으로. 그러면 회전도중 드래그를 해도 소는 움직이지 않음. 다시 r키가 눌리지면 저장했던 값을 onK에.
```

```
GLdouble angleX = 0; GLdouble angleY = 0; GLdouble angleZ = 0; // glRotated의 x,y,z 자리에 들어갈 글로벌 변수
```

```
int rCount = 0; // 소가 돌고있으면 1, 멈춰있으면 0
```

```
void drawRotFrame() { // drawFrame()과 흡사하다. 회전하는 방향(angleX, ...)에 대응하는 회전축을 흰색으로 그린다. 길이는 10이고 중심은 소의 모델링 공간 (0,0,0)에 있다.
```

```
    if (rCount != 0) { // 즉, 소가 돌지 않고 멈춰있다면 회전축을 그릴 필요가 없다.
```

```
        GLdouble norm = sqrt(angleX*angleX + angleY*angleY + angleZ*angleZ);
```

```
        glDisable(GL_LIGHTING);
```

```
        glBegin(GL_LINES);
```

```
        glColor3d(1, 1, 1);
```

```
        if (norm != 0.0) {
```

```
            glVertex3d(-5 * angleX / norm, -5 * angleY / norm, -5 * angleZ / norm);
```

```
            glVertex3d(5 * angleX / norm, 5 * angleY / norm, 5 * angleZ / norm);
```

```
        }
```

```
        glEnd();
```

```
    }
```

```
}
```

```
void onMouseDrag( int x, int y ) {
```

```
    y = height - y - 1;
```

```
    printf( "in drag (%d, %d)\n", x - oldX, y - oldY );
```

```
    // (Project 2,3,4) TODO : Implement here to perform properly when drag the mouse on each case, respectively.
```

```
    //edited
```

```
    glPushMatrix();
```

```
    glLoadMatrixd(cow2wld.matrix()); // 소의 변환 행렬을 로드한다
```

```
    glTranslated((x - oldX)*onX*0.1, (x - oldX)*onY*0.1, (x - oldX)*onZ*0.1); // x - oldX는 드래그 보
```

```
    폭, onK는 어떤 방향으로 움직여야 하는지. 0.1은 임의의 속력 상수
```

```
    glGetDoublev(GL_MODELVIEW_MATRIX, cow2wld.matrix());
```

```
    oldX = x; // 위치를 저장한다
```

```
    oldY = y;
```

```
    glPopMatrix();
```

```
    //end
```

```

    glutPostRedisplay();
}

-----

void renderRotation(void) { // IdleAnimation으로 소를 회전시키는 함수
    glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
    glPushMatrix();
        glLoadMatrixd(cow2wld.matrix());
        glRotated(1, angleX, angleY, angleZ); // 정해진 회전축 (angleX, ...) 를 기준으로 1만큼 회전
        glGetDoublev(GL_MODELVIEW_MATRIX, cow2wld.matrix());
    glPopMatrix();
    glutPostRedisplay();
}

-----

//edited
    if (key == 'x'){ // 해당 키를 누르면,
        if (rCount == 1) { // 만약 소가 돌고 있다면, idleFunc(회전)을 멈춘다.
            glutIdleFunc(NULL);
            rCount = 0;
        }
        onX = 1; // x키가 눌러졌으므로 상태를 저장한다.
        onY = 0; onZ = 0;
    }

    if (key == 'y') {
        if (rCount == 1) {
            glutIdleFunc(NULL);
            rCount = 0;
        }
        onY = 1;
        onX = 0; onZ = 0;
    }

    if (key == 'z') {
        if (rCount == 1) {
            glutIdleFunc(NULL);
            rCount = 0;
        }
        onZ = 1;
        onY = 0; onX = 0;
    }

    if (key == 'r') {
        if (rCount == 1) { // 만약 소가 돌고 있다면, idleFunc(회전)을 멈추고, 임시로 저장해뒀던 onX, ...를
        되돌린다.

            glutIdleFunc(NULL);
            rCount = 0;
            onX = tmpOnX; onY = tmpOnY; onZ = tmpOnZ;
        }
        else { // 돌고있지 않다면, 일단 현재의 onK를 tmpOnK에 임시저장하고, 무작위로 회전축을 선택한 뒤
renderRotation idle함수를 실행한다. 회전상태를 rCount에 저장한다.

```

```
tmpOnX = onX; tmpOnY = onY; tmpOnZ = onZ;
onX = 0; onY = 0; onZ = 0;
srand((unsigned int)time(NULL));
angleX = (GLdouble)rand(); angleY = (GLdouble)rand(); angleZ = (GLdouble)rand();
glutIdleFunc(renderRotation);
rCount = 1;
    }
}
//end
```