

1. Variables

```
int trans_old = 0;           // 직전에 움직였던 x좌표를 저장함
double amount[3] = { 0,0,0 }; // x, y, z 축으로 움직임을 정해줌
bool rotateOn = false;      // 'r'로 인해 rotate 상태인지 표시

double rotx = 0;            // rotate 할 때 x축
double roty = 0;            // rotate 할 때 y축
double rotz = 0;            // rotate 할 때 z축
```

2. Functions

Rotate일 경우 *glutIdleFunc(idle);* 에 사용하기 위한 *idle* 함수. Cow의 matrix를 (rotx, roty, rotz)를 축으로 5도만큼 rotate 한다. 이때 matrix의 계산은 *MODELVIEW_Matrix*를 Identity matrix로 바꾼 후 Cow matrix를 곱하고 Rotate matrix를 곱한 후, 이 결과를 다시 *Cow matrix*에 저장해준다.

```
void idle() {
    glPushMatrix();
    glLoadIdentity();
    glMultMatrixd(cow2wld.matrix());
    glRotated(5, rotx, roty, rotz);
    glGetDoublev(GL_MODELVIEW_MATRIX, cow2wld.matrix());
    glPopMatrix();
    glutPostRedisplay();
}
```

3. onMouseDrag

Idle() 함수와 비슷하게 cow matrix에 translation 연산을 해주었다. 이때 *trans_old* 변수는 직전의 x좌표이다. 따라서 현재 좌표 x와 이전 좌표 *trans_old*의 차를 이용해 움직이는 정도를 정하였다. *amount[0]*, *amount[1]*, *amount[2]* 는 각각 x, y, z축 중 어느 곳으로 움직일지에 대한 정보가 담겨있다.

```
glPushMatrix();
glLoadIdentity();
glMultMatrixd(cow2wld.matrix());
glTranslated(amount[0] * (x - trans_old), amount[1] * (y - trans_old), amount[2] * (z - trans_old));
glGetDoublev(GL_MODELVIEW_MATRIX, cow2wld.matrix());
glPopMatrix();
trans_old = x;
```

4. onKeyPress

'r'을 누를 때 toggle 되도록 먼저 *rotateOn*의 state를 바꿔준다. ON일 경우 arbitrary line으로 rotate해야하므로, *rand()* 함수를 사용해 random하게 xyz축을 정해준다. 그 후 *glutIdleFunc(idle);*을 실행하여 *idle()* 함수를 callback 한다. 다시 'r'을 눌렀을 경우 callback을 멈추기 위해 *glutIdleFunc(NULL);*을 실행한다.

```
if ((key == 'r') && (rotateOn == false)) {
    rotateOn = true;
    printf("rotate on\n");
    rotx = std::rand();
    roty = std::rand();
    rotz = std::rand();
    glutIdleFunc(idle);
}
else if ((key == 'r') && (rotateOn == true)) {
    rotateOn = false;
    printf("rotate off\n");
    glutIdleFunc(NULL);
}
```

x, y, z를 누렀을 경우 각각의 축으로 소가 움직여야 한다. 이때 *onMouseDown()*에서 cow에 대한 translation을 실행하는데, 움직이는 크기가 *amount* array의 각 element에 움직인 거리를 곱하여 계산된다. 따라서 *amount[0]*은 x축, *amount[1]*은 y축, *amount[2]*은 z축으로 정하였고, 각 축의 버튼이 눌릴 때 그를 제외한 다른 element는 0으로 setting 되고 해당하는 축은 적당한 크기를 정하여 0.02로 setting 하였다.

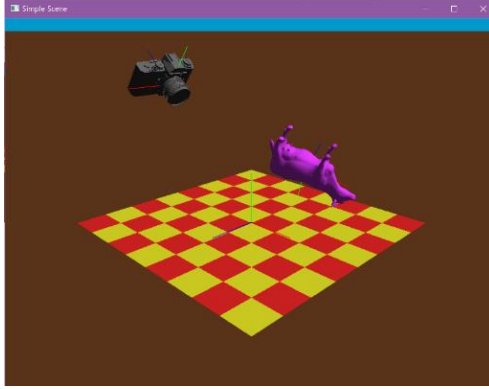
```
if (key == 'x') {
    amount[0] = 0.02;
    amount[1] = 0.0;
    amount[2] = 0.0;
}
if (key == 'y') {
    amount[0] = 0.0;
    amount[1] = 0.02;
    amount[2] = 0.0;
}
if (key == 'z') {
    amount[0] = 0.0;
    amount[1] = 0.0;
    amount[2] = 0.02;
}
```

실행 사진

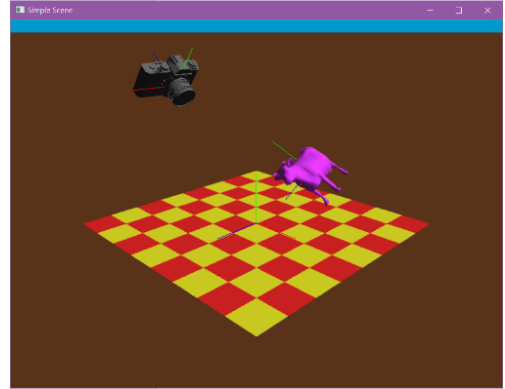
- Spinning cow with arbitrary line



Spin-1

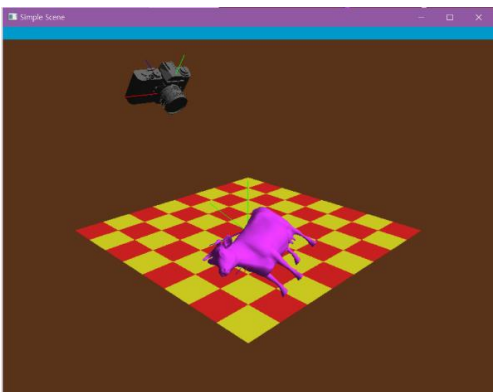


Spin-2

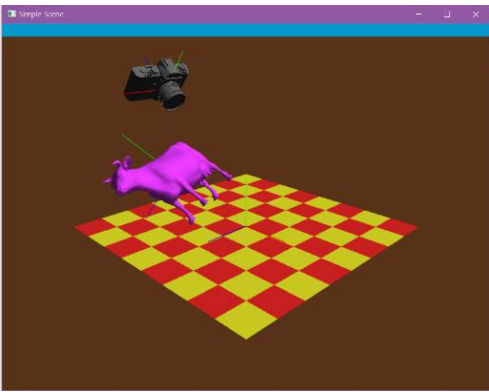


Spin-3

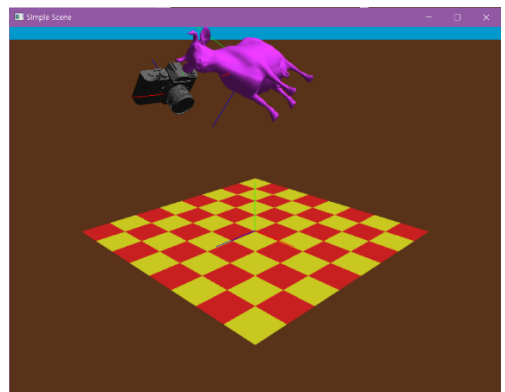
- Cow translated with x, y, z axis (at Spin-3)



X-axis



Y-axis



Z-axis