컴퓨터그래픽스

과제 2:

로봇팔 애니메이션

201903868

컴퓨터전자시스템공학과

황가은

<소스코드>

```
#include <GL/glut.h>
#include <GL/gl.h>
#include <GL/glu.h>
#include <math.h>
#include <stdio.h>

GLfloat theta = -60;
GLfloat sh_theta = -60;
GLfloat max_T = 60;
GLfloat min_T = -60;
GLint check = 0;
GLint sh_mv = 0;

void Draw_Body()
{
	glBegin(GL_POLYGON);
	glColor3f(1.0f, 0.0f, 0.0f);
	glVertex3f(0.5, 0.5, 0.0);
	glVertex3f(0.5, -0.5, 0.0);
	glVertex3f(-0.5, -0.5, 0.0);
	glVertex3f(-0.5, 0.5, 0.0);
	glEnd();
}

void Draw_upperArm()
{
	glBegin(GL_POLYGON);
	glColor3f(1.0f, 1.0f, 0.0f);
```

```
        glVertex3f(0.0, 0.0, 0.0);
        glVertex3f(0.0, 0.3, 0.0);
        glVertex3f(0.7, 0.3, 0.0);
        glVertex3f(0.7, 0.0, 0.0);
        glEnd();
}

void Draw_Left_upperArm()
{
        glBegin(GL_POLYGON);
        glColor3f(1.0f, 1.0f, 0.0f);
        glVertex3f(0.0, 0.0, 0.0);
        glVertex3f(0.0, 0.3, 0.0);
        glVertex3f(-0.7, 0.3, 0.0);
        glVertex3f(-0.7, 0.0, 0.0);
        glEnd();
}

void GoToShoulderCoordinates()
{
        glTranslatef(0.5, 0.5, 0.0);
        glRotatef(sh_theta, 0.0, 0.0, 1.0);                //sh_theta로 함으로써 팔뚝을 움직일
때 회전되지 않도록 하기 위함.
}

void GoToShoulderCoordinates_L()
{
        glTranslatef(-0.5, 0.5, 0.0);
        glRotatef(sh_theta * -1, 0.0, 0.0, 1.0);

}

void Draw_LowerArm()
{
        glBegin(GL_POLYGON);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex3f(0.0, 0.0, 0.0);
        glVertex3f(0.0, 0.3, 0.0);
        glVertex3f(0.7, 0.3, 0.0);
        glVertex3f(0.7, 0.0, 0.0);
        glEnd();
}

void Draw_Left_LowerArm()
{
        glBegin(GL_POLYGON);
        glColor3f(0.0f, 0.0f, 1.0f);
        glVertex3f(0.0, 0.0, 0.0);
        glVertex3f(0.0, 0.3, 0.0);
        glVertex3f(-0.7, 0.3, 0.0);
        glVertex3f(-0.7, 0.0, 0.0);
        glEnd();
}

void GoToElbowCoordinates()
```

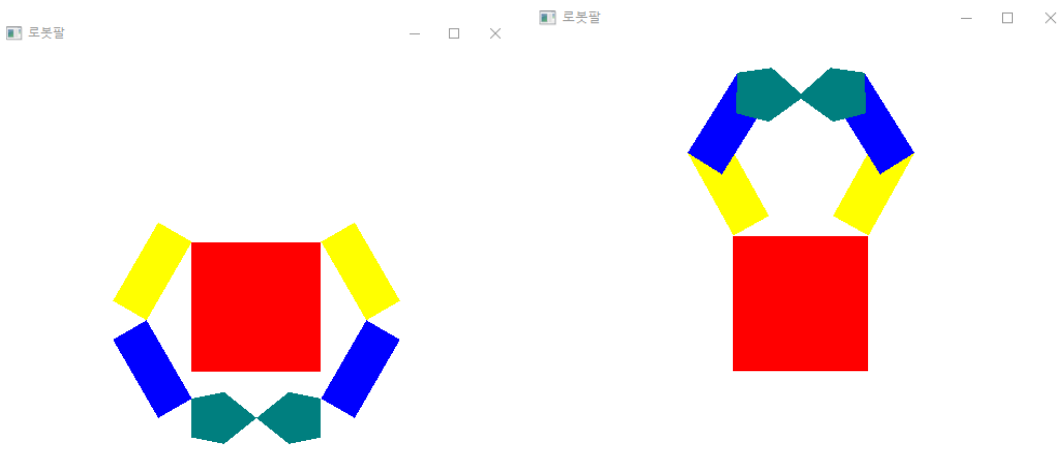
```
{
	glTranslatef(0.7, 0.0, 0.0);
	glRotatef(theta, 0.0, 0.0, 1.0);
}

void GoToElbowCoordinates_L()
{
	glTranslatef(-0.7, 0.0, 0.0);
	glRotatef(theta * -1, 0.0, 0.0, 1.0);
}

void Draw_Hand()
{
	glBegin(GL_POLYGON);
	glColor3f(0.0f, 0.5f, 0.5f);
	glVertex3f(0.0, 0.0, 0.0);
	glVertex3f(0.0, 0.3, 0.0);
	glVertex3f(0.25, 0.35, 0.0);
	glVertex3f(0.5, 0.15, 0.0);
	glVertex3f(0.25, -0.05, 0.0);
	glEnd();
}

void Draw_Left_Hand()
{
	glBegin(GL_POLYGON);
	glColor3f(0.0f, 0.5f, 0.5f);
	glVertex3f(0.0, 0.0, 0.0);
	glVertex3f(0.0, 0.3, 0.0);
	glVertex3f(-0.25, 0.35, 0.0);
	glVertex3f(-0.5, 0.15, 0.0);
	glVertex3f(-0.25, -0.05, 0.0);
	glEnd();
}

void GoToWristCoordinates()
{
	glTranslatef(0.7, 0.0, 0.0);
	glRotatef(theta, 0.0, 0.0, 1.0);
}

void GoToWristCoordinates_L()
{
	glTranslatef(-0.7, 0.0, 0.0);
	glRotatef(theta * -1, 0.0, 0.0, 1.0);
}

void create_R()
{
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	Draw_Body();
	glPushMatrix();
	GoToShoulderCoordinates();
	Draw_upperArm();
```

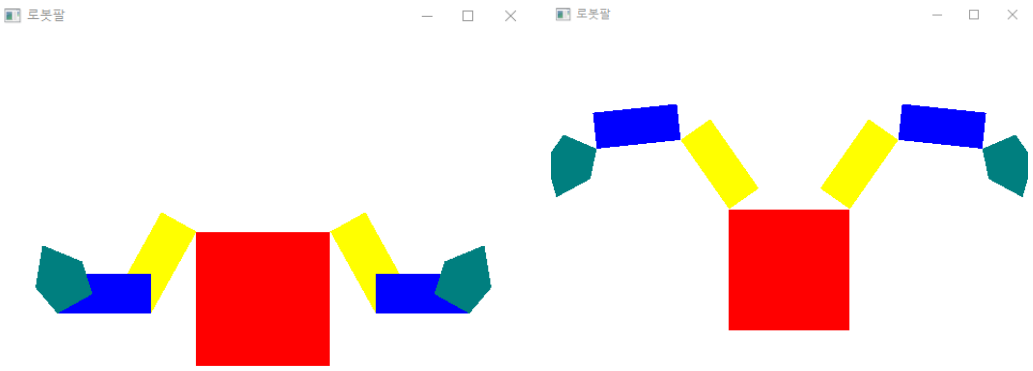
```
	glPushMatrix();
	GoToElbowCoordinates();
	Draw_LowerArm();
	glPushMatrix();
	GoToWristCoordinates();
	Draw_Hand();
	glPopMatrix();
	glPopMatrix();
	glPopMatrix();
}

void create_L()
{
	glMatrixMode(GL_MODELVIEW);
	glLoadIdentity();
	Draw_Body();
	glPushMatrix();
	GoToShoulderCoordinates_L();
	Draw_Left_upperArm();
	glPushMatrix();
	GoToElbowCoordinates_L();
	Draw_Left_LowerArm();
	glPushMatrix();
	GoToWristCoordinates_L();
	Draw_Left_Hand();
	glPopMatrix();
	glPopMatrix();
	glPopMatrix();
}


void move_up(GLint sh_mv)
{
	if (sh_mv == 0)
	{
		if (theta <= max_T && sh_theta <= max_T)
		{
			theta++;
			sh_theta++;
		}
	}
	else if (sh_mv == 1)
	{
		if (theta <= max_T)
		{
			theta++;
		}
	}
}

void move_down(GLint sh_mv)
{
	if (sh_mv == 0)
	{
		if (theta >= min_T && sh_theta >= min_T) {
```

```
                        theta--;
                        sh_theta--;
                }
        }
        else if (sh_mv == 1)
        {
                if (theta >= min_T)
                {
                        theta--;
                }
        }

}

void MyDisplay()
{
        glClear(GL_COLOR_BUFFER_BIT);
        create_R();
        create_L();
        if (check == 1)
        {
                sh_mv = 0;
                move_up(sh_mv);
        }
        else if (check == 2)
        {
                sh_mv = 0;
                move_down(sh_mv);
        }
        else if (check == 3)
        {
                sh_mv = 1;
                move_up(sh_mv);
        }
        else if (check == 4)
        {
                sh_mv = 1;
                move_down(sh_mv);
        }
        glEnd();
        glutSwapBuffers();
}

void MyTimer(int Value)
{
        glutPostRedisplay();
        glutTimerFunc(10, MyTimer, 1);
}

void key(unsigned char key, int x, int y)
{
        if (key == 'q' || key == 'Q')
        {
                check = 1;
        }
```

```
        else if (key == 'a' || key == 'A')
        {
                check = 2;
        }
        else if (key == 'p' || key == 'P')
        {
                check = 3;
        }
        else if (key == 'l' || key == 'L')
        {
                check = 4;
        }
}

int main(int argc, char** argv)
{
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(0, 0);
        glutCreateWindow("로봇팔");
        glClearColor(1.0, 1.0, 1.0, 1.0);
        glMatrixMode(GL_PROJECTION);
        glLoadIdentity();
        glOrtho(-2.0, 2.0, -2.0, 2.0, 2.0, -.0);
        glutDisplayFunc(MyDisplay);
        glutKeyboardFunc(key);
        glutTimerFunc(40, MyTimer, 1);
        glutMainLoop();
        return 0;
}
```

<화면 DUMP>





<소감>

만드는 도중 팔뚝만 움직여진 상태에서 q와 a를 누르면 그 상태로도 움직여야 하는지 고민을 했었습니다. 다행히 과제를 설명해주실 땐 따로 언급을 해주진 않으셔서 그대로 했습니다. 팔뚝만 움직여야하는데 어깨쪽도 같이 움직여서 구분을 하기위해 시간이 좀 걸렸던 것 같습니다.