

컴퓨터그래픽스 과제2 인간막대 만들기

학과, 학번 : 컴퓨터공학과 21011575

이름 : 박준형

과제 보고서는 코드를 과제 별 실행한 함수 및 실행결과로 나온 화면을 먼저 보여 준 후 혹시나 cpp파일이 열리지 않을 경우를 대비해 소스코드를 뒤에 첨부합니다.

(코드에 관련된 간단한 설명은 코드에 주석으로 표시하였습니다.)

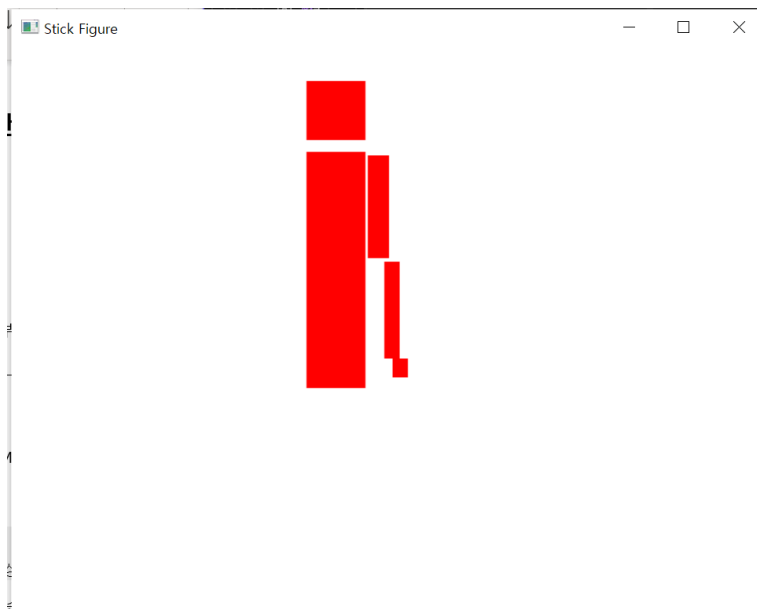
추가적으로 사용한 함수 -> Keyboard 콜백, Reshape 콜백, Menu 콜백을 이용하였습니다.

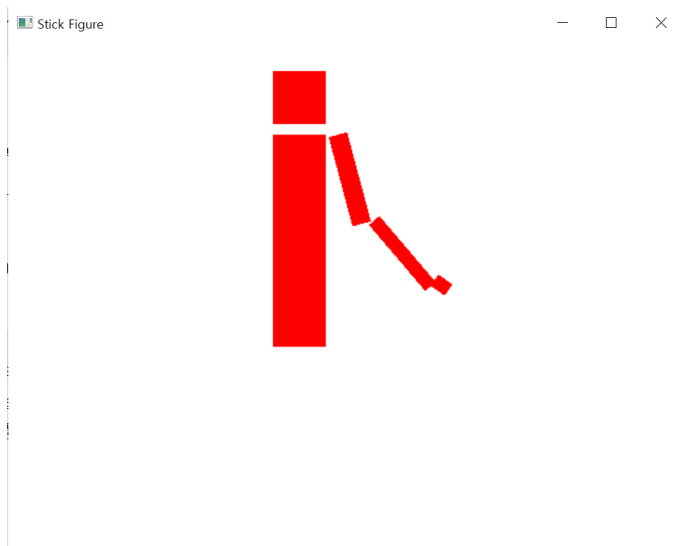
1) 계층구조 모델링을 사용하여 '막대 인간'을 만들고

이 막대 인간을 'a' 또는 'A' 를 누르면 애니메이션 하는 프로그램 작성하기

-> 막대 인간을 머리, 몸통, 윗팔, 아랫팔, 손으로 구성하였습니다. 좌표를 이동시키며 각 신체의 부위를 그렸고, 키보드의 'a' 또는 'A'키를 누르면 윗팔, 아랫팔, 손을 각도를 변화시켜 움직이도록 설정하였습니다.

<A>





<C>



-> 위의 A,B,C 그림에서와 같이 'a'또는 'A'키를 누를 시 팔이 움직이도록 코딩하였습니다.

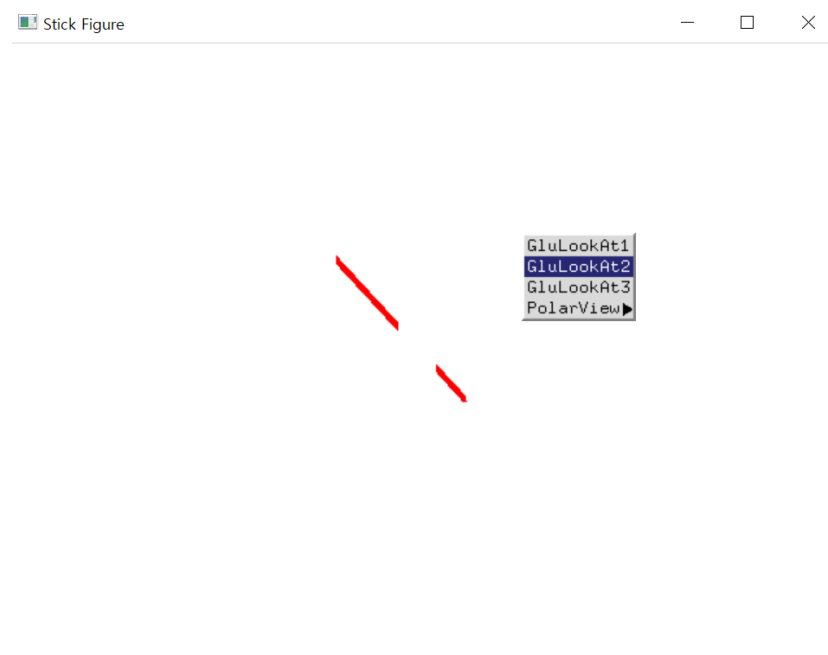
2) 위의 막대 인간을 **gluLookAt** 을 이용하여 시점 설정에 의해 애니메이션하기

-> gluLookAt을 통해 시점 변환을 할 수 있는 것을 공부자료에서 확인할 수 있습니다. 저는 이 부분을 Menu 콜백을 이용하여 본인이 원하는 gluLookAt을 선택하여 원하는 시점으로 볼 수 있도록 설정하였습니다.

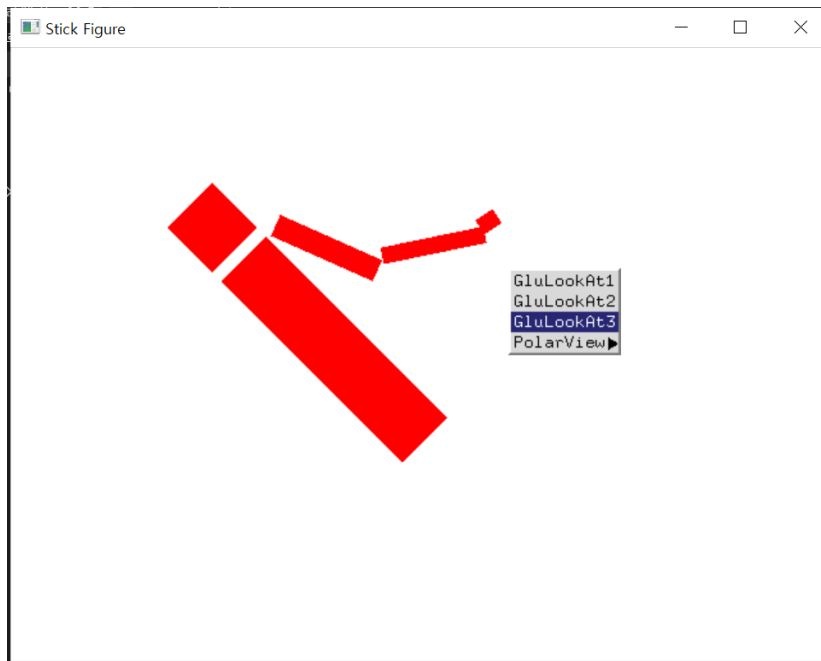
<A> gluLookAt(0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0, 0.0);의 메뉴를 선택한 결과



 gluLookAt(0.3, 0.3, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0, 0.0);의 메뉴를 선택한 결과



<C>gluLookAt(0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 1.0, 1.0, 0.0);의 메뉴를 선택한 결과

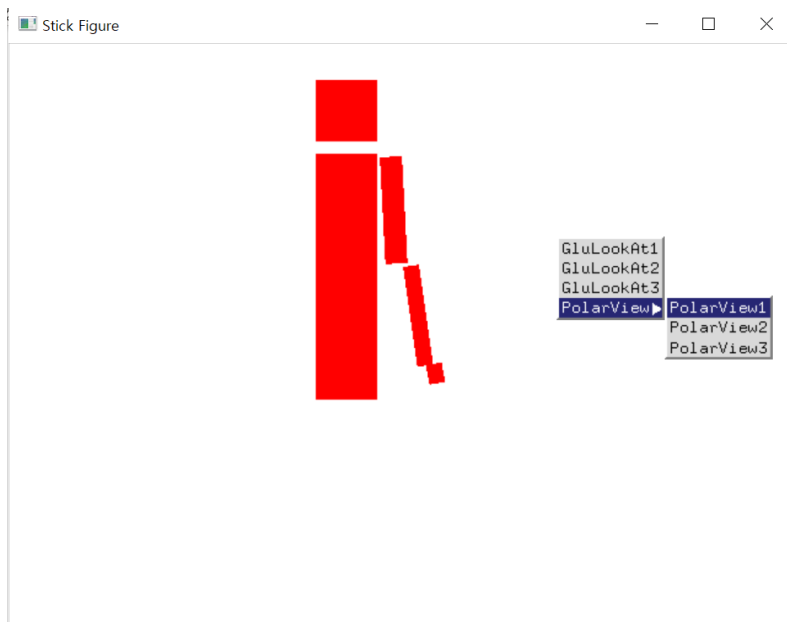


-> 메뉴를 통해 선택한 glulookat시점으로 볼 수 있다.

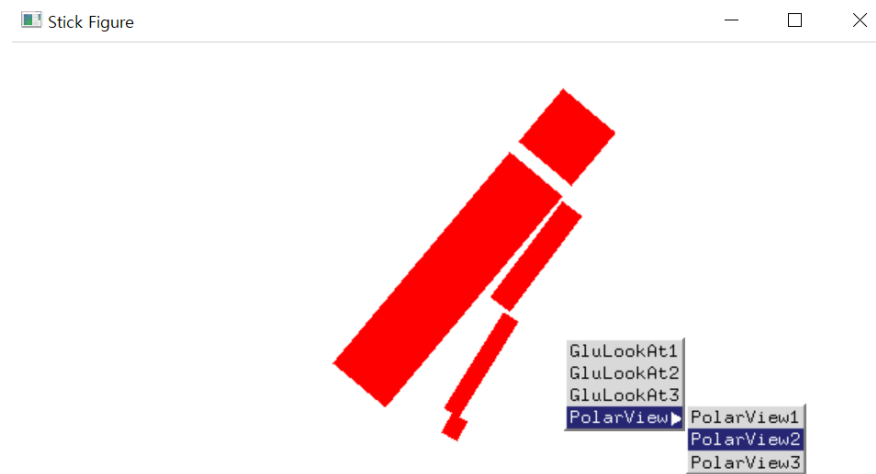
3) 위의 막대 인간을 극좌표를 이용하여 시점 설정에 의해 애니메이션하기

-> PolarView 함수를 사용하였고, 메뉴의 PolarView의 서브메뉴안에서 원하는 PolarView 시점을 선택하여 시점변환을 실시하였다.

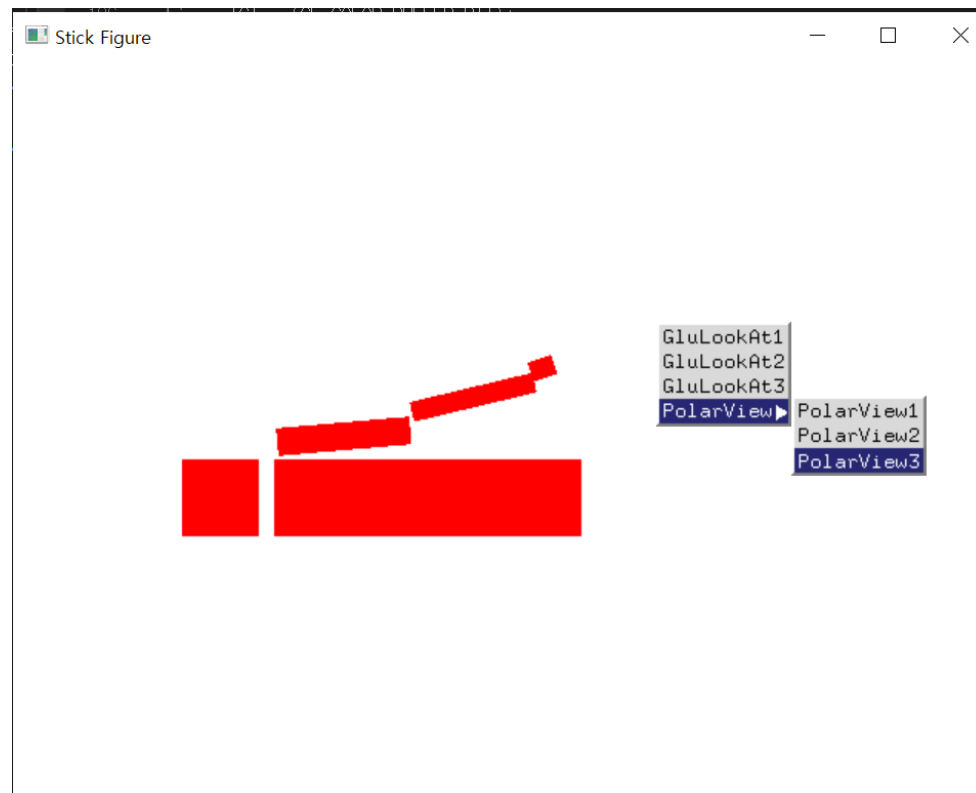
<A> PolarView(0.0, 0.0, 0.0, 0.0);을 사용하였다.



 PolarView(0.0, 0.0, 40.0, 0.0);을 사용하였다.



<C> PolarView(0.0, 0.0, 0.0, -90.0);을 사용하였다.



아래는 최종 소스코드 첨부입니다.

```
#include <iostream>
#include <GL/glut.h>

// 화면 크기
const int SCREEN_WIDTH = 640;
const int SCREEN_HEIGHT = 480;

// 초기 각도
GLfloat angle = 0.0f;

//gluLookAt을 통한 시점 변환을 파악하기 위해 gluLookat을 일단 기본값인 1로 설정한다.
GLint glulookat = 1;
//PolarView를 통한 시점 변환을 파악하기 위해 polarview를 일단 기본값인 0으로 설정한다.
GLint polarview = 0;

//극 좌표를 이용한 시점 설정을 위하여 PolarView함수 선언
void PolarView(GLfloat radius, GLfloat elevation, GLfloat azimuth, GLfloat twist) {
    glTranslatef(0.0, 0.0, -radius);          // 시점좌표계를 전역좌표계로 일치시키기 위함
    glRotatef(-twist, 0.0, 0.0, 1.0);        // 이동된 시점 좌표계의 z축을 기준으로 x-y축
    방향이 회전
    glRotatef(-elevation, 1.0, 0.0, 0.0);    // 두 좌표계의 z 축이 일치
    glRotatef(-azimuth, 0.0, 0.0, 1.0);      // y축이 일치함으로써 나머지 x축은 자동으로 일치
}

// 막대 인간 그리기 함수
//머리 모양 그리기
void Draw_Head()
{
    glBegin(GL_QUADS);
    glVertex2i(-70, 160);
    glVertex2i(-70, 210);
    glVertex2i(-20, 210);
    glVertex2i(-20, 160);
    glEnd();
    glColor3f(1.0, 0.0, 0.0);
}

//몸통 모양 그리기
void Draw_Body()
{
    glBegin(GL_QUADS);
    glVertex2i(-70, -50);
    glVertex2i(-70, 150);
    glVertex2i(-20, 150);
    glVertex2i(-20, -50);
    glEnd();
}
```

```
    glColor3f(1.0, 0.0, 0.0);  
}
```

//윗팔 모양 그리기

```
void Draw_UpperArm()  
{  
    glBegin(GL_QUADS);  
    glVertex2i(2, -3);  
    glVertex2i(2, -90);  
    glVertex2i(20, -90);  
    glVertex2i(20, -3);  
    glEnd();  
    glColor3f(1.0, 0.0, 0.0);  
}
```

//아래팔 모양 그리기

```
void Draw_LowerArm()  
{  
  
    glBegin(GL_QUADS);  
    glVertex2i(0, -3);  
    glVertex2i(0, -85);  
    glVertex2i(13, -85);  
    glVertex2i(13, -3);  
    glEnd();  
    glColor3f(1.0, 0.0, 0.0);  
}
```

//손 그리기

```
void Draw_Hand()  
{  
    glBegin(GL_QUADS);  
    glVertex2i(1, 0);  
    glVertex2i(1, -16);  
    glVertex2i(14, -16);  
    glVertex2i(14, 0);  
    glEnd();  
    glColor3f(1.0, 0.0, 0.0);  
}
```

// 머리 그린 후 몸통 그리기 위해 좌표를 원점으로 이동

```
void GoToOriginCoordinates()  
{  
    glTranslatef(0, 0, 0);  
}
```

// 몸통 그린 후 윗 팔 그리기 위해 어깨로 좌표 이동

```
void GoToShoulderCoordinates()  
{  
    glTranslatef(-20, 150, 0);  
}
```

//윗팔 그린 후 아랫 팔 그리기 위해 팔꿈치로 좌표 이동

```
void GoToElbowCoordinates()
```

```

{
    glTranslatef(16, -90, 0);
}

//아랫 팔 그린 후 손을 그리기 위해 손목으로 좌표 이동
void GoToWristCoordinates()
{
    glTranslatef(6, -85, 0);
}

// 팔 움직임 함수 각도가 1씩 변하게 만든다.
void Move_Arm()
{
    angle += 1.0f;
    if (angle > 60.0f) {
        angle = 0.0f;
    }
    glutPostRedisplay(); // 화면 다시 그리기
}

// 디스플레이 설정 함수
void Display()
{
    glClear(GL_COLOR_BUFFER_BIT);

    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();

    //마우스 오른쪽키 메뉴를 통해서 선택한 메뉴로 gluLookAt을 다르게 설정해 시점을 변환한다.
    if (glulookat == 1)
        gluLookAt(0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0, 0.0);
    else if (glulookat == 2)
        gluLookAt(0.3, 0.3, 0.0, 0.0, 0.0, -1.0, 0.0, 1.0, 0.0);
    else if (glulookat == 3)
        gluLookAt(0.0, 0.0, 0.0, 0.0, 0.0, -1.0, 1.0, 1.0, 0.0);

    //메뉴에서 PolarView의 서브메뉴에서 극좌표에 따라 다르게 설정해 시점을 변환한다.
    if (polarview == 1)
        PolarView(0.0, 0.0, 0.0, 0.0);
    else if (polarview == 2)
        PolarView(0.0, 0.0, 40.0, 0.0);
    else if (polarview == 3)
        PolarView(0.0, 0.0, 0.0, -90.0);

    Draw_Head();
    glPushMatrix();
    GoToOriginCoordinates();

    Draw_Body();
    glPushMatrix();
    GoToShoulderCoordinates();

    glRotatef(angle, 0.0f, 0.0f, 1.0f); // 회전
    Draw_UpperArm();
    glPushMatrix();

```



```

GoToElbowCoordinates();

glRotatef(1.7*angle, 0.0f, 0.0f, 1.0f); // 회전
Draw_LowerArm();
glPushMatrix();
GoToWristCoordinates();

glRotatef(angle, 0.0f, 0.0f, 1.0f); // 회전
Draw_Hand();

glPopMatrix();
glPopMatrix();
glPopMatrix();
glPopMatrix();
glutSwapBuffers();

}

// 초기화 함수
void Init()
{
    glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
}

// 창 크기 변경 시 호출되는 함수
void Reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-320, 320, -240, 240);
}

// 키보드 입력 처리 함수
//a또는 A를 누르면 Move_Arm() 함수를 호출시켜 팔이 움직이도록 한다.
void Keyboard(unsigned char key, int x, int y)
{
    switch (key) {
        case 'a':
        case 'A':
            Move_Arm();
            break;
    }
}

//메뉴를 뭘 설정하느냐에 따라 glulookat변수의 값을 바꿔준다.
void MyMainMenu(int entryID) {
    if (entryID == 1)
        glulookat = 1;
    else if (entryID == 2)
        glulookat = 2;
}

```

```

    else if (entryID == 3)
        glulookat = 3;
    glutPostRedisplay();
}

```

//메뉴에서 PolarView의 서브메뉴를 무엇을 설정하느냐에 따라 polarview변수의 값을 바꿔준다

```

void MySubMenu(int entryID) {
    if (entryID == 1)
        polarview = 1;
    else if (entryID == 2)
        polarview = 2;
    else if (entryID == 3)
        polarview = 3;
    glutPostRedisplay();
}

```

```

int main(int argc, char** argv)
{

```

```

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    //빌드하여 윈도우창이 뜨도록 한다.
    glutCreateWindow("Stick Figure");

```

```

    //마우스 오른쪽키를 누른 후 PolarView메뉴의 산하에 있는 추가 메뉴로
    //극좌표를 통한 시점변환을 가능하게 해준다.
    GLint MySubMenuID = glutCreateMenu(MySubMenu);
    glutAddMenuEntry("PolarView1", 1);
    glutAddMenuEntry("PolarView2", 2);
    glutAddMenuEntry("PolarView3", 3);

```

```

    //마우스 오른쪽키를 눌렀을 때 gluLookAt시점 또는 극좌표를 통한 시점으로
    //바꿔줄 수 있도록 메뉴를 설정한다.
    GLint MyMainMenuID = glutCreateMenu(MyMainMenu);
    glutAddMenuEntry("GluLookAt1", 1);
    glutAddMenuEntry("GluLookAt2", 2);
    glutAddMenuEntry("GluLookAt3", 3);
    glutAddSubMenu("PolarView", MySubMenuID);
    glutAttachMenu(GLUT_RIGHT_BUTTON);

```

```

    //콜백함수 실행
    glutDisplayFunc(Display);
    glutKeyboardFunc(Keyboard);
    glutReshapeFunc(Reshape);

```

```

    Init();
    glutMainLoop();
    return 0;
}

```