과제3\_16011024\_주수헌\_보고서

-애니메이션은 제가 과제2에서 했던 군대해병대 만세 체조 애니메이션 바탕으로 했습니다. 외적으로 수정사항이 있다면 좀더 잘 보이는 색깔로 수정하였습니다

\*코드설명

//그리고 소스 코드에서 코드마다 주석을 달아 주었습니다

void MyTimer(int Value) {

time5 = time5 + 0.01; //군대체조 타이머

glutPostRedisplay();

glutTimerFunc(5, MyTimer, 1);

}

void MyIdle() {//아이들 콜백

time5 = time5 + 0.01; //군대체조 타이머

glutPostRedisplay();

}

일단 애니메이션 실행은 아이들 콜백 타이며 콜백 2가지로 준비하였고 제출하는 코드는 아이들 콜백으로 실행되도록 하였습니다

void MyReshape(int w, int h) {

Width = w;

Height = h;

}

이함수로 크기 변환에 대한 값을 수정하도록 하였습니다

void MyDisplay() {

glMatrixMode(GL\_PROJECTION);

glPushMatrix();

glLoadIdentity();

gluPerspective(30, (GLdouble)Width / (GLdouble)Height, 3.0, 50.0);

glMatrixMode(GL\_MODELVIEW);

glPushMatrix();

//gluLookAt(0.0, 0.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

MilitaryExercise();

glPopMatrix();

glMatrixMode(GL\_PROJECTION);

glPopMatrix();

glutSwapBuffers();

}

-이 함수를 통해 보여지는 애니메이션을 실행 하였습니다

void MilitaryExercise() { //로봇이 해병대 만세체조

/\*로봇 움직임 가속도는 sin, cos함수를 이용하였고 관전 제한을 abs()함수를 이용하였다\*/

L\_Arm\_x = -abs(sin(time5) \* 180); //왼쪽,오른쪽 어깨를 180도정도 회전하게 하였고

R\_Arm\_x = L\_Arm\_x; //abs함수를 사용하여 뒤쪽으로는 꺽이지 않게 하였다

R\_Leg\_x = abs(sin(time5) \* 30) - 30; //왼쪽,오른쪽 허벅지는 -30에서 시작으로 30범위를 왔다갔다 움직이게 하였다

L\_Leg\_x = R\_Leg\_x; //물론 뒤로 넘어가지 않게 abs함수를 사용하였으며

R\_Leg\_y = abs(cos(time5) \* 30); //허벅지와 달리, 상단쪽이 30도로 왔다갔다해야 하므로

L\_Leg\_y = R\_Leg\_y; //cos함수를 이용하였으며 반대로 꺽이지 않게 abs함수를 사용하였다

//애니메이션 그리기//

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT); //초기화

//GL\_COLOR\_BUFFER\_BIT: 컬러를 쓰기 위해 현재 활성화된 버퍼

//GL\_DEPTH\_BUFFER\_BIT: 깊이 버퍼

//로봇이 체조를 하면서 상,하 움직이는 것을 표현

float i = 0;

i = abs(sin(time5) \* 0.08); //i변수값 설정

gluLookAt(0.0, 0.0, 7.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);

glViewport(0, 0, Width / 2, Height / 2); //좌하단

DrawGround1(); //움직이지 않은 지면 호출

glPushMatrix();// 처음 저장 좌표를 저장한다

glTranslatef(0.0, i, 0); //변수 i만큼 로봇의 몸체가 Y축기준으로 움직인다

glTranslatef(0.0, 0.5, 0.0);//최초 위치로 다시 옮긴다

glPushMatrix(); // 처음 저장 위치

DrawBody(0, 0, 0, 0); // 몸통 함수 호출

Drawneck(); // 목함수 호출

DrawHead(); // 머리 호출

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//오른쪽 팔부분 호출

DrawR\_Arm(R\_Arm\_x, 1, 0, 0); //오른쪽팔호출

DrawR\_Hand(R\_Arm\_y, 1, 0, 0); //오른쪽팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 팔 호출

DrawL\_Arm(L\_Arm\_x, 1, 0, 0); //왼쪽팔호출

DrawL\_Hand(L\_Arm\_y, 1, 0, 0); //왼족팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 다리 호출

glTranslatef(-0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(-90, 0, 1, 0); //y축 기준으로 회전

DrawL\_Legs(L\_Leg\_x, 1, 0, 0); //왼쪽허벅지

DrawL\_foot(L\_Leg\_y, 1, 0, 0); //왼쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 다리 호출

glTranslatef(0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(90, 0, 1, 0); //y축 기준으로 회전

DrawR\_Legs(R\_Leg\_x, 1, 0, 0); //오른쪽허벅지

DrawR\_foot(R\_Leg\_y, 1, 0, 0); //오른쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity(); //시점좌표를 초기화 후 재설정

gluLookAt(7.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0); //우하단

glViewport(Width / 2, 0, Width / 2, Height / 2);

DrawGround1(); //움직이지 않은 지면 호출

glPushMatrix();// 처음 저장 좌표를 저장한다

glTranslatef(0.0, i, 0); //변수 i만큼 로봇의 몸체가 Y축기준으로 움직인다

glTranslatef(0.0, 0.5, 0.0);//최초 위치로 다시 옮긴다

glPushMatrix(); // 처음 저장 위치

DrawBody(0, 0, 0, 0); // 몸통 함수 호출

Drawneck(); // 목함수 호출

DrawHead(); // 머리 호출

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//오른쪽 팔부분 호출

DrawR\_Arm(R\_Arm\_x, 1, 0, 0); //오른쪽팔호출

DrawR\_Hand(R\_Arm\_y, 1, 0, 0); //오른쪽팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 팔 호출

DrawL\_Arm(L\_Arm\_x, 1, 0, 0); //왼쪽팔호출

DrawL\_Hand(L\_Arm\_y, 1, 0, 0); //왼족팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 다리 호출

glTranslatef(-0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(-90, 0, 1, 0); //y축 기준으로 회전

DrawL\_Legs(L\_Leg\_x, 1, 0, 0); //왼쪽허벅지

DrawL\_foot(L\_Leg\_y, 1, 0, 0); //왼쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//왼쪽 다리 호출

glTranslatef(0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(90, 0, 1, 0); //y축 기준으로 회전

DrawR\_Legs(R\_Leg\_x, 1, 0, 0); //오른쪽허벅지

DrawR\_foot(R\_Leg\_y, 1, 0, 0); //오른쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity(); //시점좌표를 초기화 후 재설정

gluLookAt(0.0, 7.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, -1.0); //좌상단

glViewport(0, Height / 2, Width / 2, Height / 2);

DrawGround1(); //움직이지 않은 지면 호출

glPushMatrix();// 처음 저장 좌표를 저장한다

glTranslatef(0.0, i, 0); //변수 i만큼 로봇의 몸체가 Y축기준으로 움직인다

glTranslatef(0.0, 0.5, 0.0);//최초 위치로 다시 옮긴다

glPushMatrix(); // 처음 저장 위치

DrawBody(0, 0, 0, 0); // 몸통 함수 호출

Drawneck(); // 목함수 호출

DrawHead(); // 머리 호출

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장//오른쪽 팔부분 호출

DrawR\_Arm(R\_Arm\_x, 1, 0, 0); //오른쪽팔호출

DrawR\_Hand(R\_Arm\_y, 1, 0, 0); //오른쪽팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장/

DrawL\_Arm(L\_Arm\_x, 1, 0, 0); //왼쪽팔호출

DrawL\_Hand(L\_Arm\_y, 1, 0, 0); //왼족팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장

glTranslatef(-0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(-90, 0, 1, 0); //y축 기준으로 회전

DrawL\_Legs(L\_Leg\_x, 1, 0, 0); //왼쪽허벅지

DrawL\_foot(L\_Leg\_y, 1, 0, 0); //왼쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix();// 처음 저장 좌표 다시 저장

glTranslatef(0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(90, 0, 1, 0); //y축 기준으로 회전

DrawR\_Legs(R\_Leg\_x, 1, 0, 0); //오른쪽허벅지

DrawR\_foot(R\_Leg\_y, 1, 0, 0); //오른쪽종아리 리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity(); //시점좌표를 초기화 후 재설정

gluLookAt(5.0, 5.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0); //우상단

glViewport(Width / 2, Height / 2, Width / 2, Height / 2);

DrawGround1(); //움직이지 않은 지면 호출

glPushMatrix();// 처음 저장 좌표를 저장한다

glTranslatef(0.0, i, 0); //변수 i만큼 로봇의 몸체가 Y축기준으로 움직인다

glTranslatef(0.0, 0.5, 0.0);//최초 위치로 다시 옮긴다

glPushMatrix(); // 처음 저장 위치

DrawBody(0, 0, 0, 0); // 몸통 함수 호출

Drawneck(); // 목함수 호출

DrawHead(); // 머리 호출

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix(); // 처음 저장 위치

DrawR\_Arm(R\_Arm\_x, 1, 0, 0); //오른쪽팔호출

DrawR\_Hand(R\_Arm\_y, 1, 0, 0); //오른쪽팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix(); // 처음 저장 위치

DrawL\_Arm(L\_Arm\_x, 1, 0, 0); //왼쪽팔호출

DrawL\_Hand(L\_Arm\_y, 1, 0, 0); //왼족팔뚝

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix(); // 처음 저장 위치

glTranslatef(-0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(-90, 0, 1, 0); //y축 기준으로 회전

DrawL\_Legs(L\_Leg\_x, 1, 0, 0); //왼쪽허벅지

DrawL\_foot(L\_Leg\_y, 1, 0, 0); //왼쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPushMatrix(); // 처음 저장 위치

glTranslatef(0.1, 0.0, -0.1);//미리 위치 조정후 회전

glRotatef(90, 0, 1, 0); //y축 기준으로 회전

DrawR\_Legs(R\_Leg\_x, 1, 0, 0); //오른쪽허벅지

DrawR\_foot(R\_Leg\_y, 1, 0, 0); //오른쪽종아리

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

glPopMatrix(); // 처음 저장 좌표로 돌아 간다.

}

-이 함수에서 구체적으로 애니메이션을 구현하였는데

4개의 뷰포트로 나누었었습니다

(1)좌하단에서는 앞면이 보이도록

(2)좌상단에서는 위에서 바라보는 쪽으로

(3)우하단은 옆면에서 바라보도록

(4)우상단에서는 쿼터뷰로 화면이 보이도록 하였습니다

-각 뷰포트 별로 그림을 넣었으며 뷰포트 설정은 바꿀 때 마다

glMatrixMode(GL\_MODELVIEW);

glLoadIdentity(); //시점좌표를 초기화 후 재설정

이 코드를 추가하여 시점좌표를 초기화 하고

다시 gluLookat 함수를 넣었습니다

\*실행결과 화면

실행결과는 애니메이션 실행장면을 캡쳐하였습니다

그리고 윈도우 크기를 바꿀 때도 모델이 왜곡되지 않도록 하였습니다







