

컴퓨터그래픽스

Lab #2 - MyTransform -

제 출 일	2017년 10월 24일
분 반	00반
담당교수	남병규
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. Purpose of the lab

OpenGL에서 사용되는 함수 구현

2. Source code

```
double getRadian(int num)
{
    return num*(PI / 180);
}

struct Matrix
{
    double trix[4][4];
};

struct Vertex
{
    double trix[4];
};

Matrix Stack[100];
int Spoint=0;

struct Matrix Matrix_4x4mul(struct Matrix a, struct Matrix b) {
    struct Matrix result;
    int i, j;
    for (i = 0; i < 4; i++) {
        result.trix[i][0] = a.trix[i][0] * b.trix[0][0] + a.trix[i][1] * b.trix[1][0] + a.trix[i][2] * b.trix[2][0] + a.trix[i][3] * b.trix[3][0];
        result.trix[i][1] = a.trix[i][0] * b.trix[0][1] + a.trix[i][1] * b.trix[1][1] + a.trix[i][2] * b.trix[2][1] + a.trix[i][3] * b.trix[3][1];
        result.trix[i][2] = a.trix[i][0] * b.trix[0][2] + a.trix[i][1] * b.trix[1][2] + a.trix[i][2] * b.trix[2][2] + a.trix[i][3] * b.trix[3][2];
        result.trix[i][3] = a.trix[i][0] * b.trix[0][3] + a.trix[i][1] * b.trix[1][3] + a.trix[i][2] * b.trix[2][3] + a.trix[i][3] * b.trix[3][3];
    }
    return result;
}
```

사용되는 Matrix와 Vertex를 구조체로 선언 그리고 삼각함수값에 필요한 라디안 선언

```
void mySetMatrixMode(int m) {
    if (m == 0) {
        mode = 0;
    }
    else {
        mode = 1;
    }
}

void myLoadIdentity() {
    Matrix newMatrix;
    int i = 0, j = 0;
    for(i=0; i<4; i++)
        for(j=0; j<4; j++){
            if (i == j) {
                newMatrix.trix[i][j] = 1;
            }
            else
                newMatrix.trix[i][j] = 0;
        }
    Stack[Spoint] = newMatrix;
}

struct Matrix myPopMatrix() {
    if (Spoint == 0) {
        printf("Stack is empty");
    }
    return Stack[Spoint--];
}

void myPushMatrix() {
    Stack[Spoint + 1] = Stack[Spoint];
    Spoint++;
}

void myMatrixMul(int m) {
    int i = 0, j = 0;
    for (i = 0; i<4; i++)
        for (j = 0; j<4; j++) {
            Stack[Spoint].trix[i][j] *= m;
        }
}

struct Vertex myRunMatrix(Vertex v) {
    struct Vertex result;
    int i, j;
    for (i = 0; i < 4; i++) {
        result.trix[i] = v.trix[0] * Stack[Spoint].trix[0][i] + v.trix[1] * Stack[Spoint].trix[1][i] + v.trix[2] * Stack[Spoint].trix[2][i] + v.trix[3] * Stack[Spoint].trix[3][i];
    }
    return result;
}
```

기본 셋팅에 사용되는 setMatrixmode 일단 0일땐 모델뷰 그 외는 프로젝션이다 추후 구현예정

LoadIdentity와 push pop는 의미에 맞게 스택을 이용하여 구현하였다.

mul은 현재 스택탑을 모두 곱하는 식으로 했다

runMatrix는 vertex를 받아서 스택탑과 곱해서 결과를 리턴하도록했다.

$$\begin{pmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

glTranslatef(x, y, z)

$$\begin{pmatrix} x & 0 & 0 & 0 \\ 0 & y & 0 & 0 \\ 0 & 0 & z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

glScalef(x, y, z)

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) & 0 \\ 0 & \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

glRotatef(θ , 1, 0, 0)

X축 중심

$$\begin{pmatrix} \cos(\theta) & 0 & \sin(\theta) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

glRotatef(θ , 0, 1, 0)

Y축 중심

$$\begin{pmatrix} \cos(\theta) & -\sin(\theta) & 0 & 0 \\ \sin(\theta) & \cos(\theta) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

glRotatef(θ , 0, 0, 1)

Z축 중심

Translate Rotate Scale를 나타낸 매트릭스이다.

이 값을 곱하면 Matrix를 구하는 것이다.

```

void myTranslateX(int d) {
    Matrix newMatrix;
    int i = 0, j = 0;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 4; j++) {
            if (i == j) {
                newMatrix.trix[i][j] = 1;
            }
            else
                newMatrix.trix[i][j] = 0;
        }
    newMatrix.trix[0][3] = d;
    Stack[Spoint] = Matrix_4x4mul(Stack[Spoint], newMatrix);
    if (mode == 1) { ; }
}

void myRotateX(int r) {
    double num = getRadian(r);
    double coss = cos(num);
    double sinn = sin(num);
    Matrix newMatrix;
    int i = 0, j = 0;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 4; j++) {
            if (i == j) {
                newMatrix.trix[i][j] = 1;
            }
            else
                newMatrix.trix[i][j] = 0;
        }
    newMatrix.trix[1][1] = coss;
    newMatrix.trix[1][2] = -sinn;
    newMatrix.trix[2][1] = sinn;
    newMatrix.trix[2][2] = coss;
    Stack[Spoint] = Matrix_4x4mul(Stack[Spoint], newMatrix);
    if (mode == 1) { ; }
}

```

```

void myScaleX(int s) {
    Matrix newMatrix;
    int i = 0, j = 0;
    for (i = 0; i < 4; i++)
        for (j = 0; j < 4; j++) {
            if (i == j) {
                newMatrix.trix[i][j] = 1;
            }
            else
                newMatrix.trix[i][j] = 0;
        }
    newMatrix.trix[0][0] = s;
    Stack[Spoint] = Matrix_4x4mul(Stack[Spoint], newMatrix);
    if (mode == 1) { ; }
}

```

X축에 대한 것들의 코드이다.

일단 identity matrix를 만든 후 필요한 부분에 값을 대입한 식으로 구현하였다.

추후 projection 모드일 경우를 대비해서 mode에 대한 if문을 삽입해놨다.

3. Results

ㄱ. Scale을 적용하기전

```
void main(int argc)
{
    mySetMatrixMode(0);
    Vertex test;
    test.trix[0] = 10;
    test.trix[1] = 60;
    test.trix[2] = 10;
    test.trix[3] = 1;
    Vertex test2;
    test2.trix[0] = 30;
    test2.trix[1] = 60;
    test2.trix[2] = 20;
    test2.trix[3] = 1;

    printf("Xf Xf Xf Xf\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("Xf Xf Xf Xf\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("Xf\n\n", pow(test2.trix[0]- test.trix[0],2)+pow(test2.trix[1]- test.trix[1],2)+pow(test2.trix[2]- test.trix[2],2));
    mySetMatrixMode(0);
    myLoadIdentity();
    myPushMatrix();
    myTranslateX(3);
    myRotateZ(45);
    test = myRunMatrix(test);
    test2 = myRunMatrix(test2);
    myPopMatrix();

    printf("Xf Xf Xf Xf\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("Xf Xf Xf Xf\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("Xf\n\n", pow(test2.trix[0] - test.trix[0], 2) + pow(test2.trix[1] - test.trix[1], 2) + pow(test2.trix[2] - test.trix[2], 2));
    system("pause");
}

//[CG]HW02_201302423_신종욱
```



값이 변했지만 거리는 그대로 500인걸 볼 수 있다.

ㄴ. X Y Z에 스케일 2를 적용

```
void main(int argc)
{
    mySetMatrixMode(0);
    Vertex test;
    test.trix[0] = 10;
    test.trix[1] = 60;
    test.trix[2] = 10;
    test.trix[3] = 1;
    Vertex test2;
    test2.trix[0] = 30;
    test2.trix[1] = 60;
    test2.trix[2] = 20;
    test2.trix[3] = 1;

    printf("Xf Xf Xf Xf\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("Xf Xf Xf Xf\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("Xf\n\n", pow(test2.trix[0]- test.trix[0],2)+pow(test2.trix[1]- test.trix[1],2)+pow(test2.trix[2]- test.trix[2],2));
    mySetMatrixMode(0);
    myLoadIdentity();
    myPushMatrix();
    myTranslateX(3);
    myRotateZ(45);
    myScaleZ(2);
    myScaleX(2);
    myScaleY(2);
    test = myRunMatrix(test);
    test2 = myRunMatrix(test2);
    myPopMatrix();

    printf("Xf Xf Xf Xf\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("Xf Xf Xf Xf\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("Xf\n\n", pow(test2.trix[0] - test.trix[0], 2) + pow(test2.trix[1] - test.trix[1], 2) + pow(test2.trix[2] - test.trix[2], 2));
    system("pause");
}

//[CG]HW02_201302423_신종욱
```



거리가 2의 제곱인 4배로 늘어난걸 볼 수 있다.

ㄷ. 첫번째 점은 TRS하고 두 번째 점은 TR만 했을 경우

```
void main(int argc)
{
    mySetMatrixMode(0);
    Vertex test;
    test.trix[0] = 10;
    test.trix[1] = 60;
    test.trix[2] = 10;
    test.trix[3] = 1;
    Vertex test2;
    test2.trix[0] = 30;
    test2.trix[1] = 60;
    test2.trix[2] = 20;
    test2.trix[3] = 1;
    printf("%f %f %f %f\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("%f %f %f %f\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("%f\n", pow(test2.trix[0] - test.trix[0], 2) + pow(test2.trix[1] - test.trix[1], 2) + pow(test2.trix[2] - test.trix[2], 2));
    mySetMatrixMode(0);
    myLoadIdentity();
    myPushMatrix();
    myTranslateX(3);
    myRotateZ(45);
    myScaleZ(2);
    myScaleX(2);
    myScaleY(2);
    test = myRunMatrix(test);
    myPopMatrix();

    myPushMatrix();
    myTranslateX(3);
    myRotateZ(45);
    test2 = myRunMatrix(test2);
    myPopMatrix();

    printf("%f %f %f %f\n", test.trix[0], test.trix[1], test.trix[2], test.trix[3]);
    printf("%f %f %f %f\n", test2.trix[0], test2.trix[1], test2.trix[2], test2.trix[3]);
    printf("%f\n", pow(test2.trix[0] - test.trix[0], 2) + pow(test2.trix[1] - test.trix[1], 2));
    system("pause");
}
```



pop이 잘 적용되어 스택탑 값이 사라져서 2번째값은 scale 되지 않은걸 확인할 수 있다.

4. Discussions

아직 setmode에 대해선 잘모르겠어서 어떻게 처리할지는 모르겠다.

나머지 gkgatn를 구현하면서 어떻게 작동하는지 찾아보게됨으로 좀더 OpenGL을 알게되었다.