

컴퓨터그래픽스

Lab #6 – SolarSystem -

제 출 일	2017년 11월 20일
분 반	00반
담당교수	남병규
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

1. Purpose of the lab

Light할 때 쓰이는 함수들 직접 구현해보기

2. Source code

```
void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glMatrixMode(GL_MODELVIEW);

    glLoadIdentity();
    glRotated(XAngle, 1.0, 0.0, 0.0); //x축 이동
    glRotated(YAngle, 0.0, 1.0, 0.0); //y축 이동

    glPushMatrix(); //태양그리기
    glRotated(S, 0, 0, 1.0);
    S += 1.5; //자전

    GLfloat ambient0[] = { 0.4, 0.3, 0.3, 1.0 };
    GLfloat diffuse0[] = { 0.4, 0.3, 0.3, 1.0 };
    GLfloat specular0[] = { 0.4, 0.3, 0.3, 1.0 };
    GLfloat emission0[] = { 0.8, 0.23, 0.23, 1.0 };
    GLfloat shine0 = 120.0;
    glMaterialfv(GL_FRONT, GL_AMBIENT, ambient0);
    glMaterialfv(GL_FRONT, GL_DIFFUSE, diffuse0);
    glMaterialfv(GL_FRONT, GL_SPECULAR, specular0);
    glMaterialf(GL_FRONT, GL_SHININESS, shine0); //반사율 물체 설정
    glMaterialfv(GL_FRONT, GL_EMISSION, emission0); //광원 설정
    glutSolidSphere(0.10, 50000, 1000);
    glPopMatrix();
}
```

일단 방향키에 의한 시점 이동을 위해서 XAngle과 YAngle을 아이덴티티바로 밑에 선언해준후에 태양과 행성들을 그렸다.

태양은 빛을 내야하기 때문에 emission을 추가하여서 그렸다.

공전과 자전은 전역선언된 상수를 이용하여서 계속 더하는식으로 구현하였다.

```

glPushMatrix(); //지구
glRotated(k2, 0, 0, 1.0);
glTranslatef(0.0, 0.4, 0.0); //y축 이동
glRotated(k2, 0, 0, 1.0);
k2 = k2 + 0.5;
GLfloat diffuse3[] = { 0.1, 0.6, 0.8, 1.0 };
GLfloat ambient3[] = { 0.2, 0.5, 0.8, 1.0 };
GLfloat specular3[] = { 0.4, 0.5, 0.1, 1.0 };
GLfloat light_3[] = { 0.5, 0.5, 0.5, 1.0 };
GLfloat shine3 = 80.0;
glLightfv(GL_LIGHT0, GL_POSITION, light_3);
glLightfv(GL_LIGHT0, GL_AMBIENT, ambient3);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse3);
glLightfv(GL_LIGHT0, GL_SPECULAR, specular3);
glMaterialf(GL_FRONT, GL_SHININESS, shine3);
glutSolidSphere(0.02, 50000, 1000);

glPushMatrix(); //달그리기
glRotated(k3, 0, 0, 1.0);
k3 = k3 + 1;
glTranslatef(0.0, 0.08, 0.0); //y축 이동
glRotated(k3, 0, 0, 1.0);
GLfloat diffuse4[] = { 0.1, 0.2, 0.7, 1.0 };
GLfloat ambient4[] = { 0.8, 0.3, 0.4, 1.0 };
GLfloat specular4[] = { 0.15, 0.2, 0.1, 1.0 };
GLfloat light_4[] = { 0.45, 0.5, 0.5, 1.0 };
GLfloat shine4 = 80.0;
glLightfv(GL_LIGHT0, GL_POSITION, light_4);
glLightfv(GL_LIGHT0, GL_AMBIENT, ambient4);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse4);
glLightfv(GL_LIGHT0, GL_SPECULAR, specular4);
glMaterialf(GL_FRONT, GL_SHININESS, shine4);
glutSolidSphere(0.01, 50000, 1000);
glPopMatrix();
glPopMatrix();

```

행성들 중에 지구와 목성은 위성을 가져야하기 때문에 일단 지구를 먼저 그려준뒤에 Pop을 하지않고 한번더 Push를 해서 달을 그리면 지구를 기준으로 다시 좌표계를 설정할수있어서 위성을 그릴 수 있다.

```

glPushMatrix(); //토성 그리기
glRotated(k8, 0, 0, 1.0);
glTranslatef(0.0, 0.82, 0.0); //y축 이동
glRotated(k8, 0, 0, 1.0);
glRotated(45, 1.0, 0.0, 0.0); //자전축을 x축으로 45도 만큼 기울임
k8 = k8 + 1.4;
GLfloat diffuse9[] = { 0.25, 0.45, 0.35, 1.0 };
GLfloat ambient9[] = { 0.25, 0.5, 0.35, 1.0 };
GLfloat specular9[] = { 0.25, 0.45, 0.5, 1.0 };
GLfloat light_9[] = { 0.25, 0.45, 0.35, 1.0 };
GLfloat shine9 = 80.0;
glLightfv(GL_LIGHT0, GL_POSITION, light_9);
glLightfv(GL_LIGHT0, GL_AMBIENT, ambient9);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuse9);
glLightfv(GL_LIGHT0, GL_SPECULAR, specular9);
glMaterialf(GL_FRONT, GL_SHININESS, shine9);
glutSolidSphere(0.02, 50000, 1000);
glPopMatrix();

```

목성은 자전축을 기울여야 하기 때문에 x축기준으로 기울인다음 자전시키고 y축 이동후 공전시켰다.

```

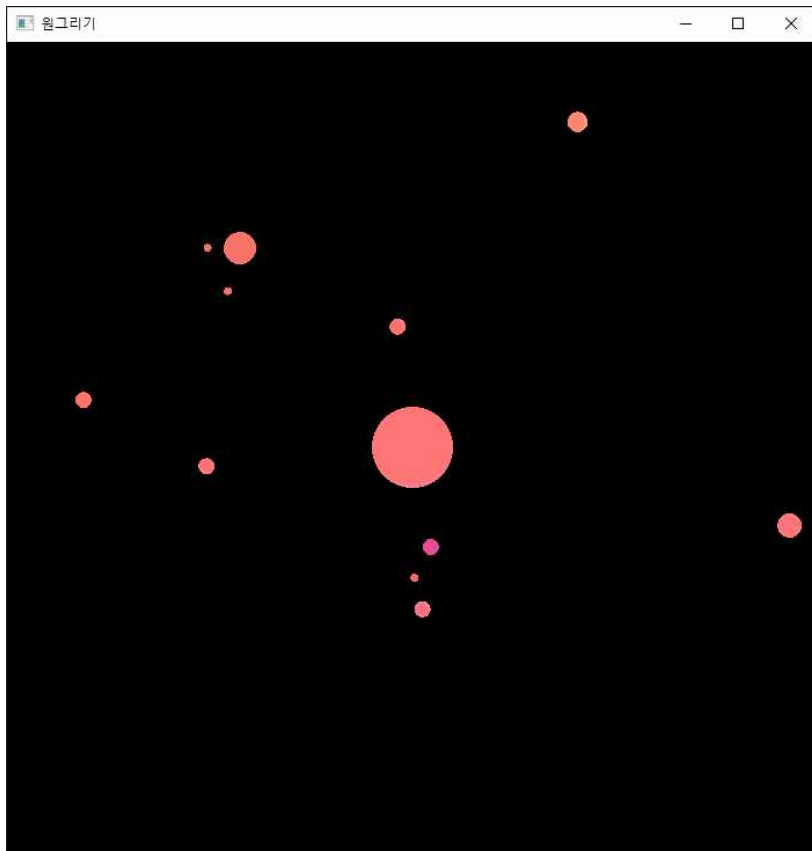
L,
void DoSpecial(int key, int x, int y)
{
    switch (key) { //방향키 조작에 따라 x,y축 변경
    case GLUT_KEY_RIGHT:
        XAngle += 10.0;
        break;
    case GLUT_KEY_LEFT:
        XAngle -= 10.0;
        break;
    case GLUT_KEY_UP:
        YAngle += + 10.0;

        break;
    case GLUT_KEY_DOWN:
        YAngle -= 10.0;;
        break;
    }
    glutPostRedisplay();
}

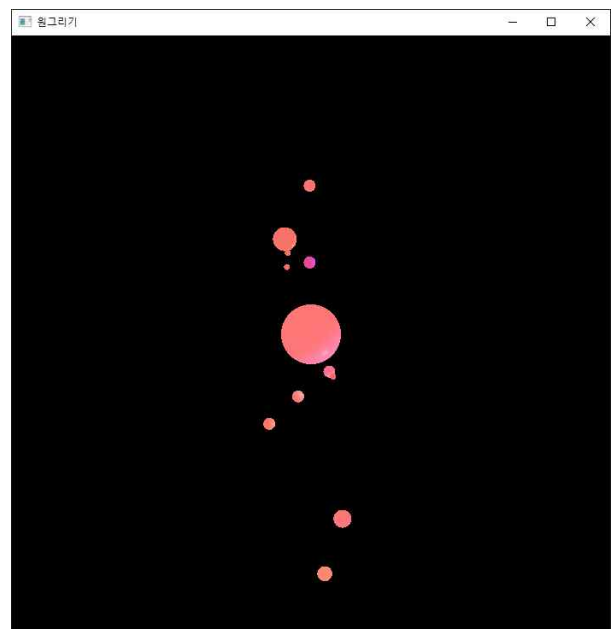
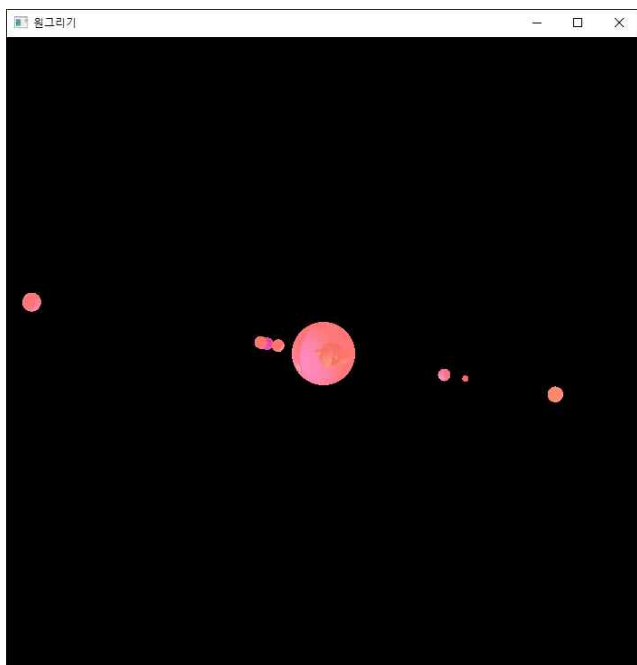
```

방향키의 입력을 받고 어떻게 처리할지에대해서 구현하였다
좌우는 x축 위아래는 y축으로 회전하는 식으로 구현하였다.

3. Results



태양계 z축위에서 본화면 달과 위성 2개 포함



방향키를 눌러서 시점 변경을한 모습

4. Discussions

사실 태양계를 그린다고해서 엄청 어려울줄 알았는데 매핑을 안하고 그림만 나타내어서 그나마 쉬웠다 거기다가 주요함수를 알려줘서 쉽게 구현하였다.