

# 컴퓨터그래픽스

Lab #7 – SolarSystem 2 -

제 출 일	2017년 11월 27일
분 반	00반
담당교수	남병규
학 과	컴퓨터공학과
학 번	201302423
이 름	신종욱

## 1. Purpose of the lab

태양계 구현 후 텍스처 매핑

## 2. Source code

```
void SetupTexture()
{
    AUX_RGBImageRec *tex[13]; /* structure to contain a texture image */
    tex[0] = auxDIBImageLoad("Sun.bmp");
    tex[1] = auxDIBImageLoad("Mercury.bmp");
    tex[2] = auxDIBImageLoad("Venus.bmp");
    tex[3] = auxDIBImageLoad("Earth.bmp");
    tex[4] = auxDIBImageLoad("Moon.bmp");
    tex[5] = auxDIBImageLoad("Mars.bmp");
    tex[6] = auxDIBImageLoad("Jupiter.bmp");
    tex[7] = auxDIBImageLoad("Io.bmp");
    tex[8] = auxDIBImageLoad("Europa.bmp");
    tex[9] = auxDIBImageLoad("Saturn.bmp");
    tex[10] = auxDIBImageLoad("Uranus.bmp");
    tex[11] = auxDIBImageLoad("Neptune.bmp");
    tex[12] = auxDIBImageLoad("Saturn_Ring.bmp"); //tex배열에 bmp파일 저장
    glGenTextures(13, ids);
    for (int k = 0; k < 13; k++) {
        sphere[k] = gluNewQuadric();
        gluQuadricTexture(sphere[k], GL_TRUE);
        glBindTexture(GL_TEXTURE_2D, ids[k]);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
        glTexImage2D(GL_TEXTURE_2D, 0, 3, tex[k]->sizeX, tex[k]->sizeY, 0, GL_RGB,
                     GL_UNSIGNED_BYTE, tex[k]->data);
        glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
    } //각각 bmp를 텍스처에 저장하면서 사이즈를 변경
}
```

저번에 구현한거에서 매핑을 하기전에 bmp파일을 저장하고 사이즈를 변경해서 매핑할 텍스처를 준비해야한다  
sphere,ids 배열은 전역으로 선언후 초기화를 해주고 bmp파일을 불러온 tex에 저장된값의 사이즈에 맞게  
준비한다.

```

glLoadIdentity();
glEnable(GL_LIGHTING);
glLightfv(GL_LIGHT0, GL_AMBIENT, ambientLight0);
glLightfv(GL_LIGHT0, GL_DIFFUSE, diffuseLight0);
glLightfv(GL_LIGHT0, GL_SPECULAR, specularLight0);
glEnable(GL_LIGHT0);
glLightfv(GL_LIGHT0, GL_POSITION, light0pos);

GLfloat ambientref[] = { 0.2,0.2,0.2,1.0 };
GLfloat diffuseref[] = { 0.6,0.6,0.8,1.0 };
GLfloat specref[] = { 1.0,1.0,1.0,1.0 };

glEnable(GL_COLOR_MATERIAL);
glColorMaterial(GL_FRONT, GL_AMBIENT_AND_DIFFUSE);
glMaterialfv(GL_FRONT, GL_SPECULAR, specref);
glMaterialfv(GL_FRONT, GL_DIFFUSE, diffuseref);
glMaterialfv(GL_FRONT, GL_AMBIENT, ambientref);
glMaterialf(GL_FRONT, GL_SHININESS, 128);
//행성들의 표면 성질을 설정하여 빛에 대하여 좀더 잘보이도록 설정

glRotated(XAngle, 1.0, 0.0, 0.0); //시점 변경을 위한 x축 회전
glRotated(YAngle, 0.0, 1.0, 0.0); //시점 변경을 위한 y축 회전

glPushMatrix();
glRotated(S, 0, 0, 1.0);
S += 0.15; //태양 자전
glBindTexture(GL_TEXTURE_2D, ids[0]);
GLfloat plane_coef_s[] = { 1.0, 0.0, 0.0, 1.0 };
GLfloat plane_coef_t[] = { 0.0, 1.0, 0.0, 1.0 };
glEnable(GL_TEXTURE_GEN_S);
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_S, GL_OBJECT_PLANE, plane_coef_s);
glEnable(GL_TEXTURE_GEN_T);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_T, GL_OBJECT_PLANE, plane_coef_t);
glBindTexture(GL_TEXTURE_2D, ids[0]);
gluQuadricTexture(sphere[0], 1);
gluSphere(sphere[0], 0.1, 30, 30); //태양에 텍스처 매핑
glDisable(GL_TEXTURE_GEN_S);
glDisable(GL_TEXTURE_GEN_T);
glPopMatrix();

```

광원을 일단 설정후 행성들의 빛반사 성질을 설정하였.

그런다음 pushMatrix를 하여서 태양을 그리는데

```
glBindTexture(GL_TEXTURE_2D, ids[k]);
```

```
gluQuadricTexture(sphere[k], 1);
```

```
gluSphere(sphere[k], 0.1, 30, 30);
```

이런식으로 저장한 bmp파일에 맞게 인덱스를 사용하여서 텍스처 매핑을하면된다.

그 외에 행성들에서도 동일하게 하면되는데 위성이 있는 달과 고리를 그리는 토성은 특별한점이 있으니 추가 설명 하겠다.

```
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 + 0.15;
glBindTexture(GL_TEXTURE_2D, ids[3]);
glEnable(GL_TEXTURE_GEN_S);
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_S, GL_OBJECT_PLANE, plane_coef_s);
glEnable(GL_TEXTURE_GEN_T);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_T, GL_OBJECT_PLANE, plane_coef_t);
glBindTexture(GL_TEXTURE_2D, ids[3]);
gluQuadricTexture(sphere[3], 1);
gluSphere(sphere[3], 0.02, 30, 30);
glDisable(GL_TEXTURE_GEN_S);
glDisable(GL_TEXTURE_GEN_T);
glutSolidSphere(0.02, 50000, 1000);

glPushMatrix(); //달그리기
glRotated(k3, 0, 0, 1.0);
k3 = k3 + 0.01;
glTranslatef(0.0, 0.08, 0.0); //y축 이동
glRotated(k3, 0, 0, 1.0);
glBindTexture(GL_TEXTURE_2D, ids[4]);
glEnable(GL_TEXTURE_GEN_S);
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_S, GL_OBJECT_PLANE, plane_coef_s);
glEnable(GL_TEXTURE_GEN_T);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_T, GL_OBJECT_PLANE, plane_coef_t);
glBindTexture(GL_TEXTURE_2D, ids[4]);
gluQuadricTexture(sphere[4], 1);
gluSphere(sphere[4], 0.01, 30, 30);
glDisable(GL_TEXTURE_GEN_S);
glDisable(GL_TEXTURE_GEN_T);
glPopMatrix();
glPopMatrix();
```

태양과 같이 지구도 똑같이 그려준다음 Pop을 하기전에 PushMatrix를하여서 지구의 좌표계에서 달을 그려줘야한다 달도 마찬가지로 매핑을 해준다음 Pop을 2번 하면된다.

```

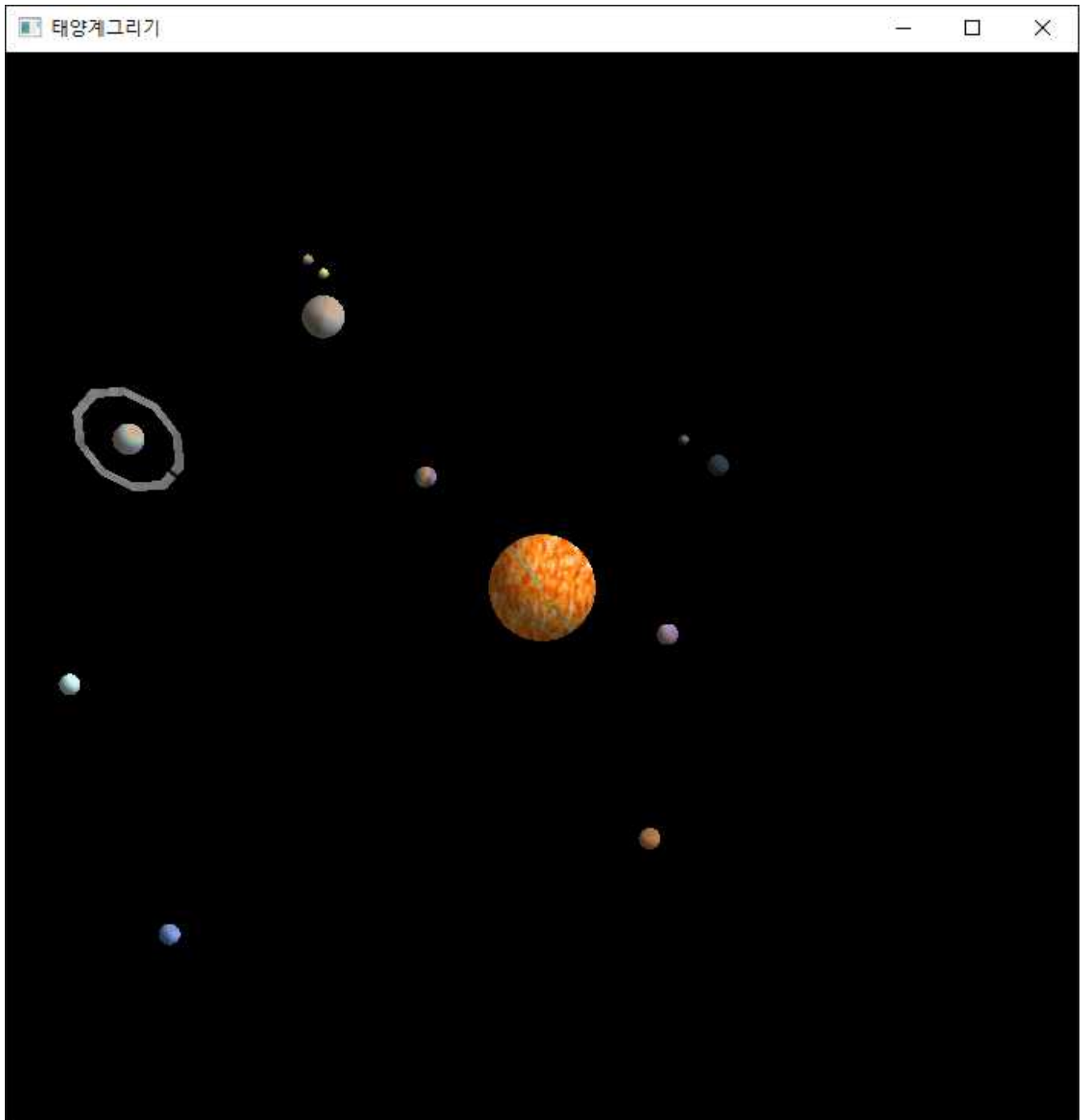
glPushMatrix(); //토성 그리기
glRotated(k8, 0, 0, 1.0);
glTranslatef(0.0, 0.82, 0.0); //y축 이동
glRotated(k8, 0, 0, 1.0);
glRotated(45, 1.0, 0.0, 0.0); //자전축을 x축으로 45도 만큼 기울임
k8 = k8 + 0.07;
glBindTexture(GL_TEXTURE_2D, ids[9]);
glEnable(GL_TEXTURE_GEN_S);
glTexGeni(GL_S, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_S, GL_OBJECT_PLANE, plane_coef_s);
glEnable(GL_TEXTURE_GEN_T);
glTexGeni(GL_T, GL_TEXTURE_GEN_MODE, GL_OBJECT_LINEAR);
glTexGenfv(GL_T, GL_OBJECT_PLANE, plane_coef_t);
glBindTexture(GL_TEXTURE_2D, ids[9]);
gluQuadricTexture(sphere[9], 1);
gluSphere(sphere[1], 0.03, 30, 30);

glPushMatrix(); //토성의 고리 그리기
glBindTexture(GL_TEXTURE_2D, ids[12]);
gluQuadricTexture(sphere[12], 1);
gluDisk(sphere[12], 0.1, 0.12, 10, 8); //gluDisk로 텍스처매핑하며 고리형태를 구현
glDisable(GL_TEXTURE_GEN_S);
glDisable(GL_TEXTURE_GEN_T);
glPopMatrix();
glPopMatrix();

```

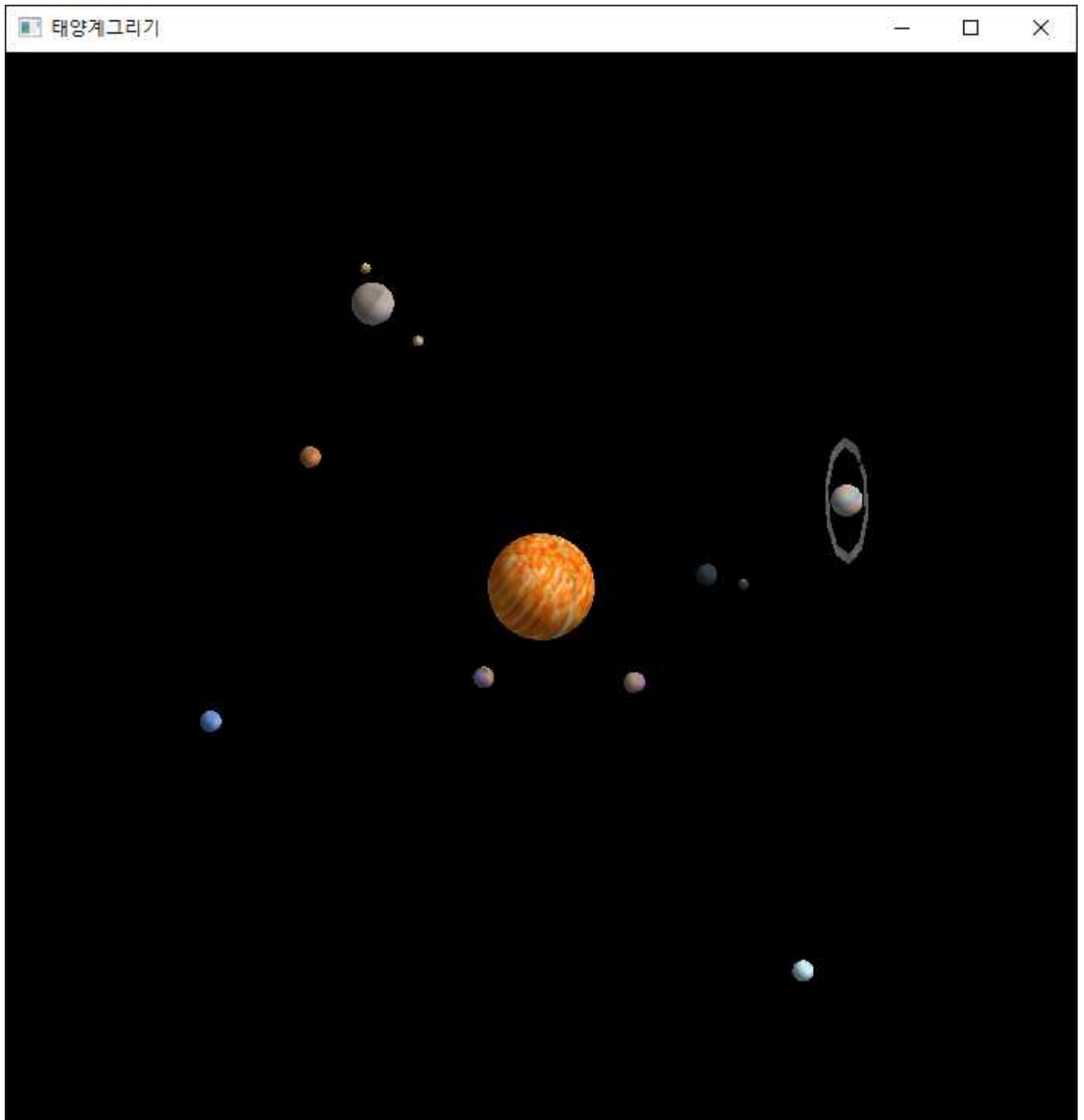
토성은 자전축을 45도 회전해야하고 그리고 토성의 고리를 따로 그려줘야한다.  
달을 그릴때와 마찬가지로 토성을 그리고 토성 좌표계에서 push를해준다음  
고리를 그리기위해 gluDisk로 텍스처 매핑해서 구현해준다 이렇게그리면 토성과 같은속도로  
자전하게된다.

### 3. Results



태양계 z축위에서 본화면 달과 위성 2개 포함

방향키를 눌러서 시점 변경을한 모습



#### 4. Discussions

bmp포맷 밖에 인식을 못한다는 점에서 아쉬웠다 그래서 고화질을 사진을 쓰고싶어도 용량이 커서 매핑이 아예 불가능하였다. png나 jpg도 가능한 매핑법이 있는지 배우고 싶다.