

## 컴퓨터그래픽스 과제4

컴퓨터공학과 17011599 안정연

### – Create a 3D scene

저번 과제로 3D scene을 다음과 같이 구현하였다. 마우스로 회전, 줌, 이동을 할 수 있고 조명 3개를 키고 끌 수 있다.



### – Read and load images for texture mapping

이미지 파일을 읽어오는 함수를 다음과 같이 작성하였다. 그리고 텍스처를 바인딩 및 초기화하여 opengl에서 텍스처가 생성되도록 하였다.

```
GLuint loadBMP(const char *filename)
{
    FILE *fid = fopen(filename, "rb");
    if (!fid)
        return 0;

    unsigned char header[54];
    if (fread(header, 1, 54, fid) != 54){
        fclose(fid);
        return 0;
    }
}
```

```

    if (header[0] != 'B' || header[1] != 'M'){
        fclose(fid);
        return 0;
    }
    unsigned int width = *(int*)&header[0x12];
    unsigned int height = *(int*)&header[0x16];
    unsigned int imageSize = width * height * 3;

    unsigned char *data;
    data = (unsigned char*)malloc(sizeof(unsigned char)*imageSize);

    if (fread(data, 1, imageSize, fid) != imageSize){
        free(data);
        fclose(fid);
        return 0;
    }

    // Create texture in OpenGL

    GLuint textureID;
    glGenTextures(1, &textureID);

    glBindTexture(GL_TEXTURE_2D, textureID);
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB, width, height, 0, GL_BGR_EXT,
GL_UNSIGNED_BYTE, data);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);

    free(data);
    fclose(fid);
    return textureID;
}

```

메인함수에서 불러올 이미지를 loadBMP함수를 통해 textures[] 배열에 읽어왔다. 그리고 필요할때 glBindTexture(GL\_TEXTURE\_2D, textures[e]);하여 도형을 그릴때 텍스처를 사용하였다. 건물들에 창문, 간판, 입구를 텍스처링하고 집에 벽돌무늬를 텍스처링하였다. 기차에 창문을 텍스처링하고 트럭에도 이미지를 텍스처링하였다.

```
textures[0]= loadBMP("texture/road2.bmp");
```

```
textures[2] = loadBMP("texture/power.bmp");
textures[3] = loadBMP("texture/building2.bmp");
textures[6] = loadBMP("texture/hospital.bmp");
textures[7] = loadBMP("texture/cityhall.bmp");
textures[8] = loadBMP("texture/church.bmp");
textures[9] = loadBMP("texture/school.bmp");
textures[10] = loadBMP("texture/school_surface.bmp");
textures[11] = loadBMP("texture/police2.bmp");
textures[12] = loadBMP("texture/police1.bmp");
textures[13] = loadBMP("texture/police3.bmp");
textures[14] = loadBMP("texture/train.bmp");
textures[15] = loadBMP("texture/door2.bmp");
textures[16] = loadBMP("texture/block.bmp");
textures[17] = loadBMP("texture/cupang.bmp");
textures[18] = loadBMP("texture/hotel_front.bmp");
textures[19] = loadBMP("texture/hotel_up.bmp");
```



### - Texturing

키보드 't'를 누르면 scene에 texturing을 on/off하도록 keyboard함수에서 다음과 같이 구현하였다. 실제로 실행해보니 잘 적용됨을 확인할 수 있었다.

```
case 't':                                // 전체 텍스처 in or out
    if (flag == 0)
    {
        glDisable(GL_TEXTURE_2D);
        flag = 1;
    }
    else
    {
        glEnable(GL_TEXTURE_2D);
        flag = 0;
    }
    break;
```

Texturing-On



Texturing-Off



### – Filtering

필터링은 이미지를 읽어올때 다음과 같이 초기화하였다.

```
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_NEAREST);  
glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_NEAREST);
```

### – Add fog effect

안개 효과를 추가하기 위해 전역변수로 fog\_density와 fogColor를 선언 및 초기화한 후 init함수에서 안개모드, 색상, 힌트값을 설정하고 GL\_FOG를 켜서 안개 효과를 추가하였다.

```
GLfloat fog_density = 0;  
GLfloat fogColor[] = { 1.0f, 1.0f, 1.0f, 1.0f };  
void init()  
{  
    Initlight();  
  
    glFogi(GL_FOG_MODE, GL_EXP); //안개모드  
    glFogfv(GL_FOG_COLOR, fogColor); //안개 색을 설정  
    glHint(GL_FOG_HINT, GL_NICEST); //안개 힌트 값  
    glEnable(GL_FOG); //GL_FOG를 켜다.
```

```
glBlendFunc(GL_SRC_ALPHA, GL_ONE);
glEnable(GL_TEXTURE_2D);
glTexEnvf(GL_TEXTURE_ENV, GL_TEXTURE_ENV_MODE, GL_MODULATE);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();

}
```

mydisplay()에 `glFogf(GL_FOG_DENSITY, fog_density);` 를 추가하여 밀도를 설정하였고 keyboard함수에서 f를 누르면 안개의 밀도가 증가하고 g를 누르면 안개의 밀도가 감소하도록 하여 안개의 양을 조절할 수 있다.

```
case 'f':
    if (fog_density > 1)
        fog_density = 1;
    fog_density += 0.02;
    break;

case 'g':
    if (fog_density < 0)
        fog_density = 0;
    fog_density -= 0.02;
    break;
```

실행해보았더니 f를 누르면 다음과 같이 안개효과가 생기는 것을 볼 수 있었다.

