

# <Computer Graphics Final Project>

소프트웨어학과 201620986 채성희

## Texture Mapping (multiple texture mapping)

### 1. Basic Texture Mapping

- 3D 객체에 고양이 사진을 mapping한 간단한 예제이다.

- 3D 객체는 다음과 같은 과정을 통해 만들어졌다.

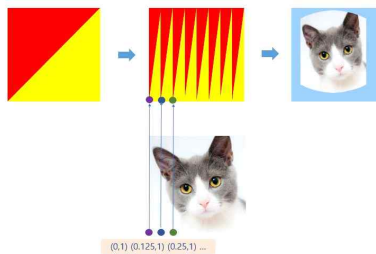
1. 삼각형 두 개로 정사각형을 그린다.

2. 정사각형을 다시 여러 개의 삼각형으로 쪼갬다.

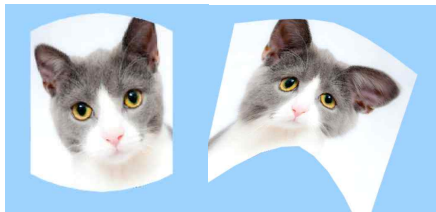
- 예제에서는 총 16개의 삼각형을 이용하여 정사각형을 이용하였다.

3. 각 꼭짓점의 좌표를 계산하여 정사각형을 그릴 수 있도록 한다.

4. 정사각형을 그린 다음 자연스러운 굴곡이 생길 수 있도록 z 값을 조정해준다.



- 3D객체를 구성하는 삼각형이 많이 있어 각 삼각형의 vertex에 해당하는 값이 texture의 어느 부분인지 계산하는 것이 중요하다. texture에서의 좌표 값을통해 uv vector를 형성하고, 형성한 vector를 통해 texture를 mapping 시킬 수 있다.



- 완성한 3D객체를 보면 rotate하는 각도에 따라, 고양이의 표정이 변화하는 것을 볼 수 있다. 만약, 삼각형을 더 잘게 자른다면, 굴곡이 더욱 자연스러워 질 것이다.

### - Vertex Shader Source Code

```
attribute highp vec3 myVertex;
attribute highp vec4 myColor;
attribute highp vec2 myUV;
uniform mediump mat4 Pmatrix;
uniform mediump mat4 Vmatrix;
uniform mediump mat4 Mmatrix;
varying mediump vec4 color;
varying mediump vec2 texCoord;
void main(void)
{
    gl_Position = Pmatrix*Vmatrix*Mmatrix*vec4(myVertex, 1.0);
    color = myColor;
    texCoord = myUV;
}
```

attribute로 현재 vertex의 정보와 각 vertex에서의 color 및 texture의 uv vector 정보가 있다. 또한, vertex들의 이동을 통해 object를 transformation 할 수 있는데, 이때 matrix의 곱의 형태로 사용된다. texture의 uv vector는 varying으로 fragment에 전해진다.

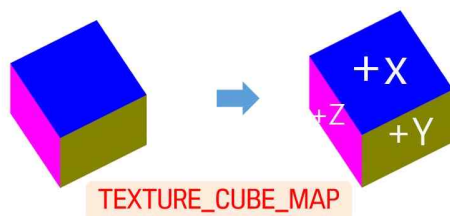
## - Fragment Shader Source Code

```
varying mediump vec2 texCoord;
uniform sampler2D sampler2d;
void main(void)
{
    gl_FragColor = texture2D(sampler2d, texCoord);
    gl_FragColor.a = 1.0 ;
}
```

vertex fragment로부터 varying을 통해 전달받은 texture의 uv vector를 통해 uniform 으로 선언된 texture를 mapping 할 수 있게 된다.

## 2. Cube Multiple Texture Mapping

- 대표적인 3D 객체 중 하나인 cube의 각면에 서로 다른 texture를 mapping하는 예제이다.
- cube의 각 면에 동일한 texture를 mapping해주게 된다면, 정육면체의 한면에 mapping할 때 이전에 보았던 texture mapping 방법과 동일하게 진행해주면 되지만, 정육면체의 각면에 서로 다른 texture를 mapping해주기 위해서는 조금 다른 방법을 사용한다.
- 예제에서는 3개의 이미지를 가지고 서로 다른 위치에 mapping 하였다.



1. 정육면체를 그릴 각 면을 삼각형들로 정의한다.
2. 정육면체의 각 면을 positive +x, negative -x, positive +y, negative -y, positive +z, negative -z로 나눈다.
3. 각 면에 위치시킬 texture들을 정의한다.
4. 정의한 texture들을 해당위치에 mapping한다. 이 때, 정육면체에 texture를 mapping 할 때, texture\_cube\_map을 이용하여 cube에 여러가지 texture들을 mapping 할 수 있다.



- 완성된 cube를 보면, 회전하면서 각 면에 고양이, 토끼, 강아지가 다른 면에 위치해 있다.

## - texture 이미지 정의와 바인딩

```
var animal_images = [
  { location: gl.TEXTURE_CUBE_MAP_POSITIVE_X, src: cuty_cat_src },
  { location: gl.TEXTURE_CUBE_MAP_NEGATIVE_X, src: cuty_cat_src },
  { location: gl.TEXTURE_CUBE_MAP_POSITIVE_Y, src: cuty_dog_src },
  { location: gl.TEXTURE_CUBE_MAP_NEGATIVE_Y, src: cuty_dog_src },
  { location: gl.TEXTURE_CUBE_MAP_POSITIVE_Z, src: cuty_rabbit_src },
  { location: gl.TEXTURE_CUBE_MAP_NEGATIVE_Z, src: cuty_rabbit_src },
]; <br>

animal_images.forEach(function (animal_image) {
  const location = animal_image.location;
  const src = animal_image.src;

  gl.texImage2D(location, 0, gl.RGBA, 1, 1, 0, gl.RGBA, gl.UNSIGNED_BYTE, new Uint8Array([0, 0, 255, 255]));

  const image = new Image();
  image.src = src;
  image.addEventListener('load', function () {
    gl.bindTexture(gl.TEXTURE_CUBE_MAP, animal_texture);
    gl.texImage2D(location, 0, gl.RGBA, gl.RGBA, gl.UNSIGNED_BYTE, image);
    gl.generateMipmap(gl.TEXTURE_CUBE_MAP);
  });
});
```

각 이미지를 positive x, negative x, positive y, negative y, positive z, negative z 의 location으로 정의해 놓고 for문을 통해 한꺼번에 이미지를 로딩 시켜준다. 이때, TEXTURE\_CUBE\_MAP을 이용하여 쉽게 texture를 binding할 수 있다.

## - VerTex Shader Source code

```
attribute highp vec3 myVertex;
attribute highp vec4 myColor;
attribute highp vec2 myUV;
uniform mediump mat4 Pmatrix; \
    uniform mediump mat4 Vmatrix; \
    uniform mediump mat4 Mmatrix; \
varying highp vec3 texCoord;
void main(void)
{
    texCoord = normalize(myVertex.xyz);
    gl_Position = Pmatrix*Vmatrix*Mmatrix*vec4(myVertex, 1.0);
};
```

나머지는 이전에 보았던 texture mapping과 같지만, texture의 vector가 vec3로 dimension 이 늘어난 것을 확인할 수 있다. x,y,z 가 모두 필요하기 때문에 vec3를 사용하여야한다.

## - fragment shader source code

```
precision mediump float;
varying highp vec3 texCoord;
uniform samplerCube Cube;
void main(void)
{
    gl_FragColor = textureCube(Cube, normalize(texCoord));
    gl_FragColor.a = 1.0 ;
}
```

fragment shader에서는 vertex shader로 부터 받은 texture coordinate를 이용하여 texture mapping을 한다. 이때, sample2D로 정의하지않고, sampleCube로 정의해 주어 cube에 mapping하기 쉽도록 해준다.