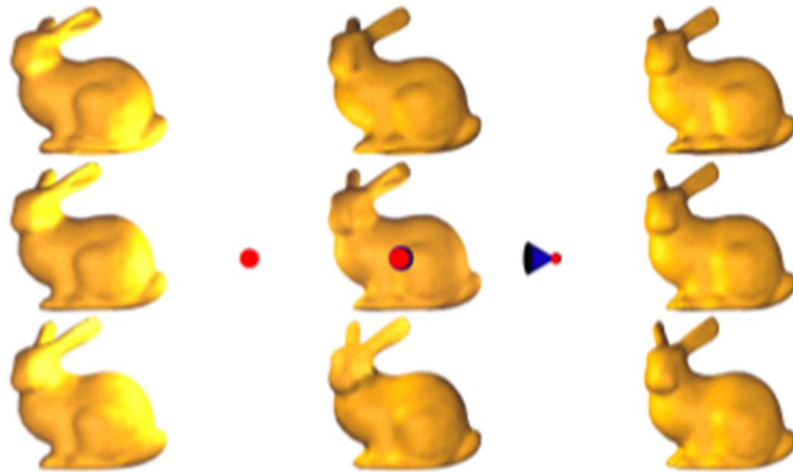


# Computer Graphics

---



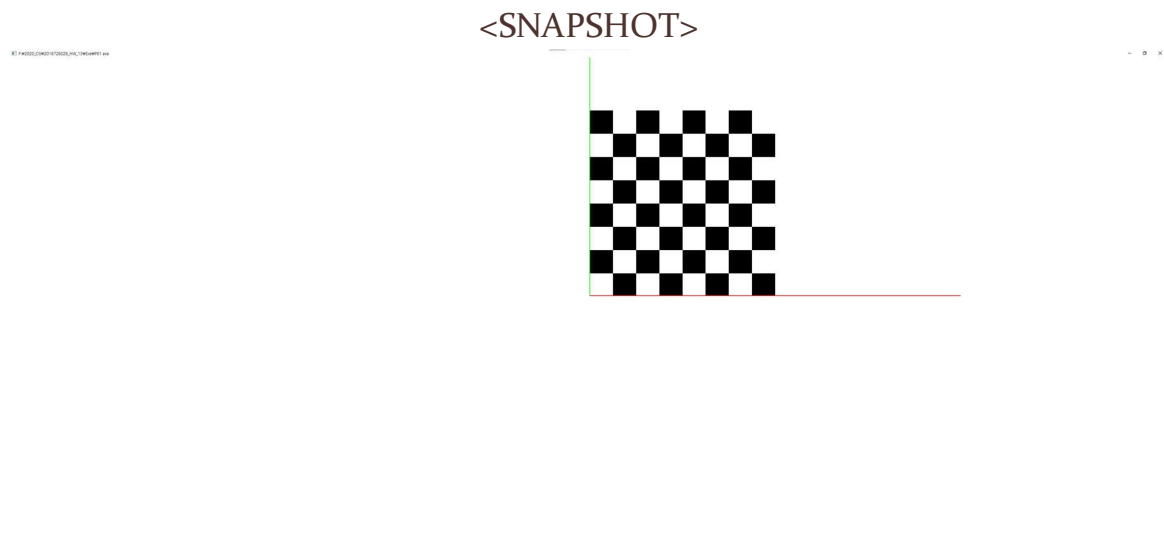
## Texture Mapping

HW13

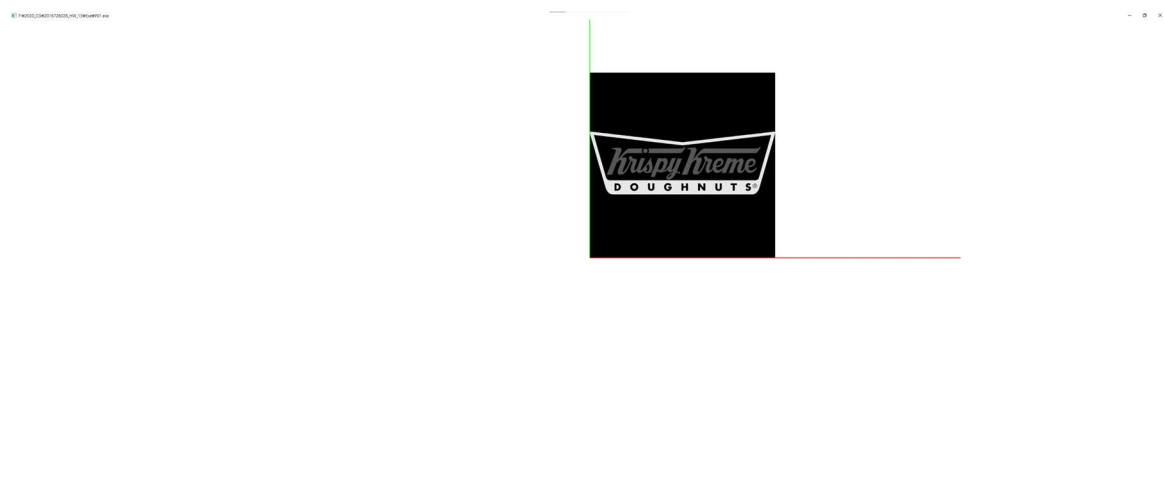
이민재 | Computer Graphics [심화전공실습 1] | 2020/11/29

	P01	P02	P03	P04	E01	Total
SCORE	1	1	1	1	1	5

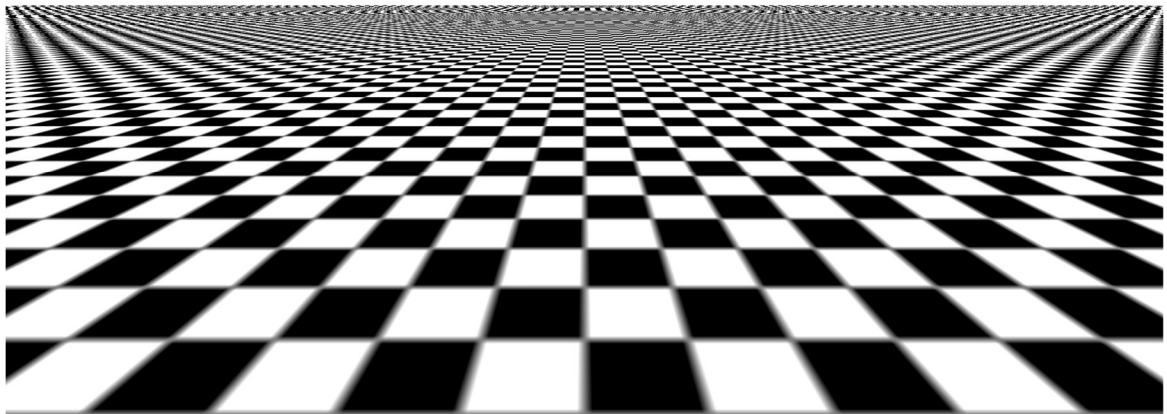
## Po1 (Checkerboard texture)



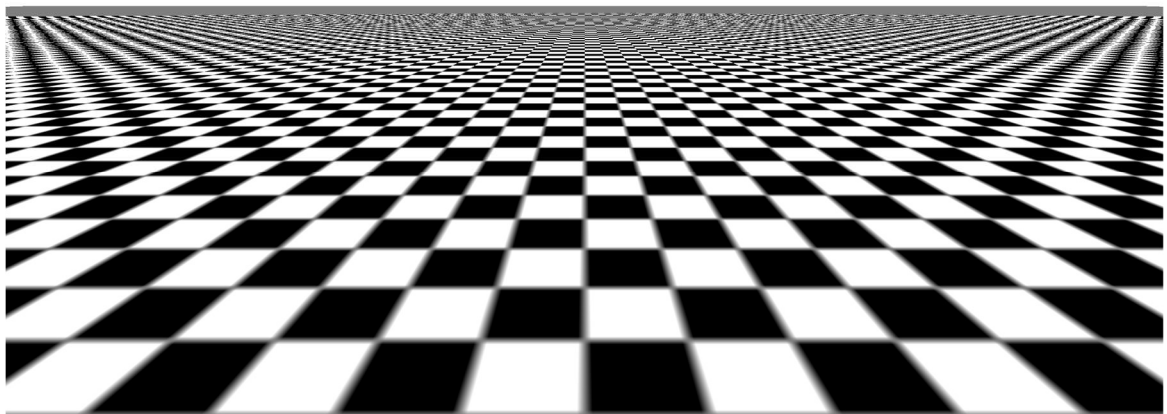
## Po2 (Texture files in the raw format (marble and logo example))



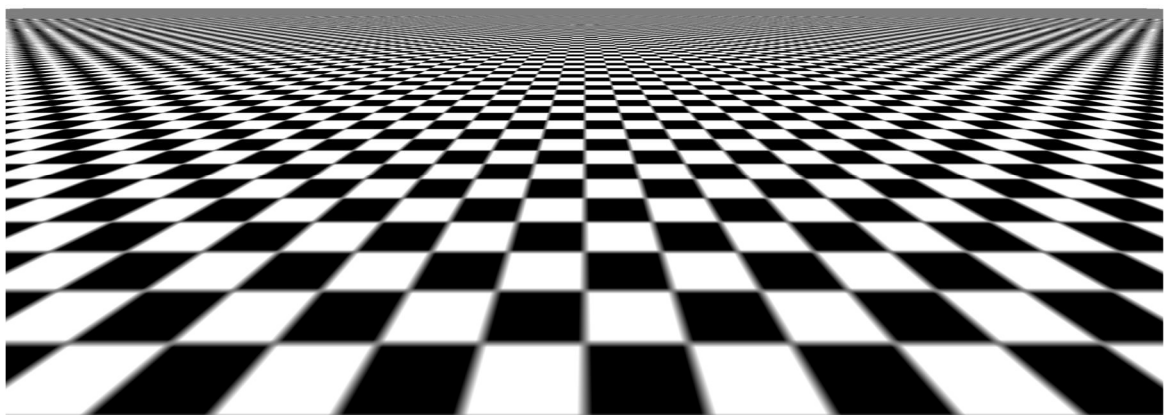
minification - nearest



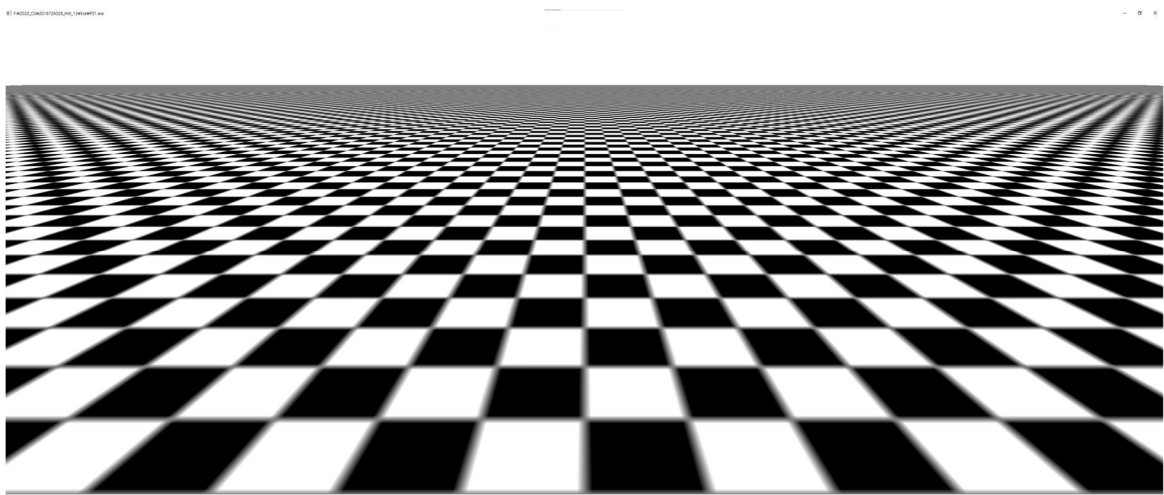
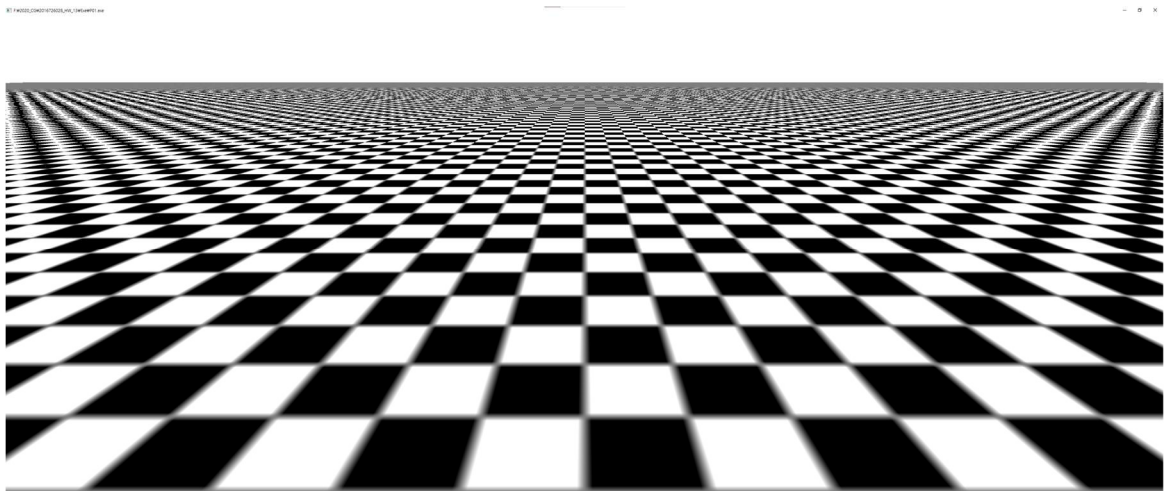
minification - linear



minification - nearest mipmap nearest



minification - linear mipmap nearest

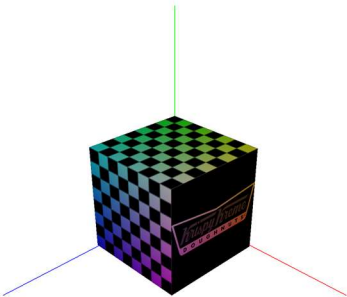


Po4 (Texture mapping to a cube)

<SNAPSHOT>

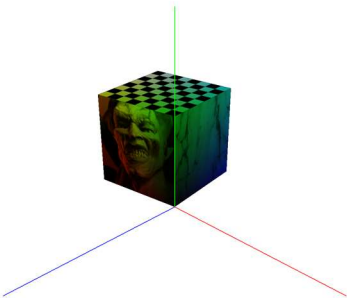
87 F:\MODEL\3D\OBJ\TEXTURE\4\1\TEXTURE01.jpg

- 0 X



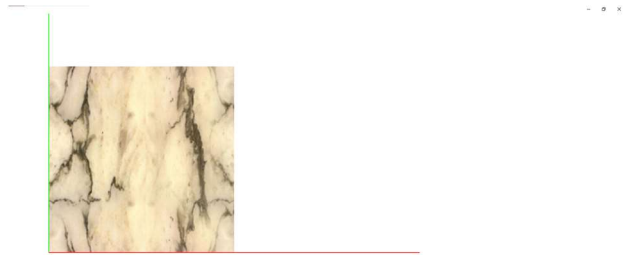
87 F:\MODEL\3D\OBJ\TEXTURE\4\1\TEXTURE01.jpg

- 0 X



### Eo1 (Texture wrapping using a marble texture)

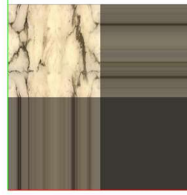
<SNAPSHOT>



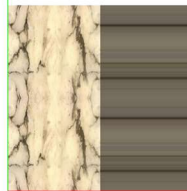
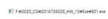
Original (textureNumRepeat = 1)



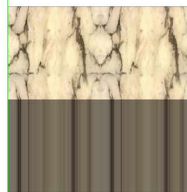
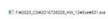
S: repeat, T: repeat (textureNumRepeat = 2)



S: clamp, T: clamp (textureNumRepeat = 2)



S: clamp, T: repeat (textureNumRepeat = 2)



S: repeat, T: clamp (textureNumRepeat = 2)



### <EXPLANATION>

Marble texture 을 로드 는 과정에서 raw texture 을 마블 텍스처로 통일하고, 원하는 래핑 방향을 각기 설정하여 다른 texID 를 부여하여 init() 시에 로드 되도록 설정하였다.

```
// Raw texture
glBindTexture(GL_TEXTURE_2D, texID[1]);
loadRawTexture("m02_marble.raw", 512, 512, 3, 1);

glBindTexture(GL_TEXTURE_2D, texID[2]);
loadRawTexture("m02_marble.raw", 512, 512, 3, 2);

glBindTexture(GL_TEXTURE_2D, texID[3]);
loadRawTexture("m02_marble.raw", 512, 512, 3, 3);

glBindTexture(GL_TEXTURE_2D, texID[0]);
loadRawTexture("m02_marble.raw", 512, 512, 3, 0);
```

추가로 전달하는 인자로 id 값과 일치하게 integer 를 넘겨, loadRawTexture 함수 안에서 S 와 T 방향으로의 clamp 와 repeat 를 결정하게 하는 트리거로서 작동하게 하였다.

```
// Only 3 and 1
if (l == 3)
{
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB8, w, h, 0,
        GL_RGB, GL_UNSIGNED_BYTE, raw);
    //glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    //glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);

    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
}
else if (l == 2)
{
    glTexImage2D(GL_TEXTURE_2D, 0, GL_RGB8, w, h, 0,
        GL_RGB, GL_UNSIGNED_BYTE, raw);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
    // glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
    // glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP);
    glTexParameterf(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT);

    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MIN_FILTER, GL_LINEAR);
    glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_MAG_FILTER, GL_LINEAR);
}
```

첨부한 이미지 하단 주석의 내용과 같은 방향으로 GL\_CLAMP 와 GL\_REPEAT 를 주었으며 통일성을 위해 각 텍스처 반복횟수는 2(repeat 시 화면상에 4 개 텍스처) 로 고정하였다.