

# 심화전공실습

## HW #13



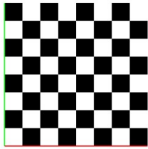

Self-scoring table

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

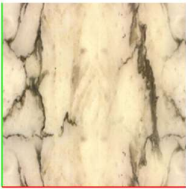

과목명	심화전공실습
학부	소프트웨어학부
학번	2019203010
이름	김민철
제출일자	2024년 12월 01일

## I. Practice

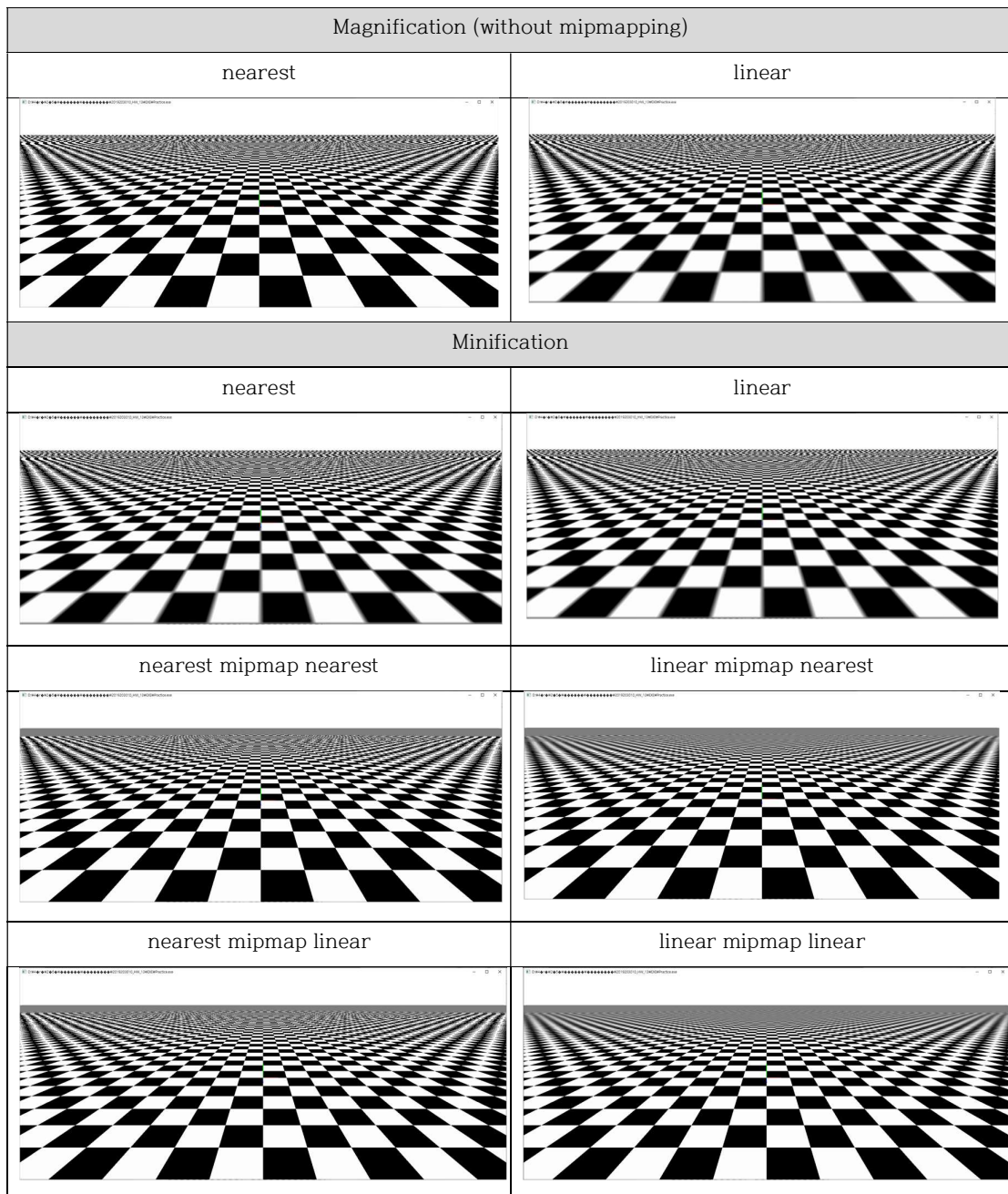
### Practice 01. Checkerboard texture

Checkerboard	+ demon head
 <hr/>	 <hr/>

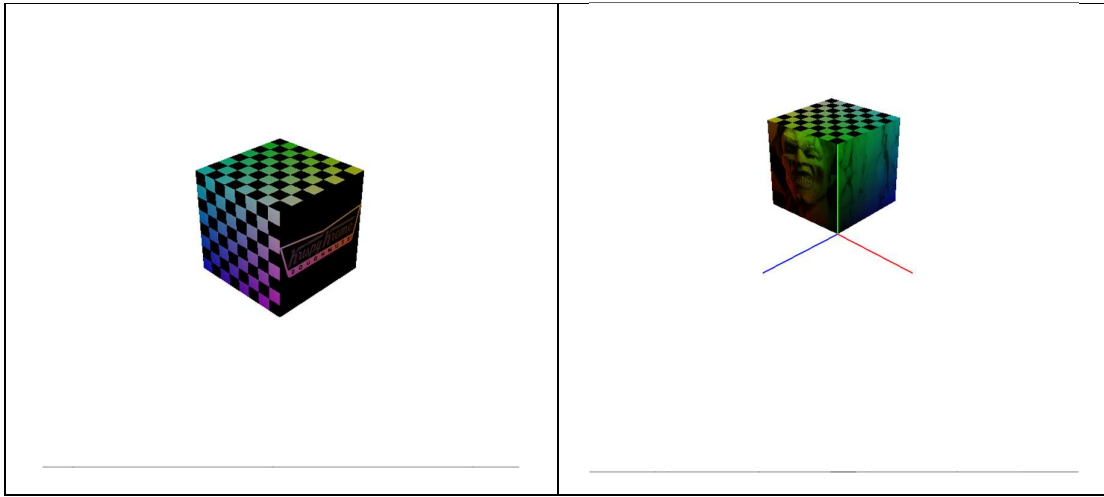
### Practice 02. Texture files in the raw format (marble and logo example)

Marble	logo
 <hr/>	 <hr/>

### Practice 03. Antialiasing with mipmapping (floor example)

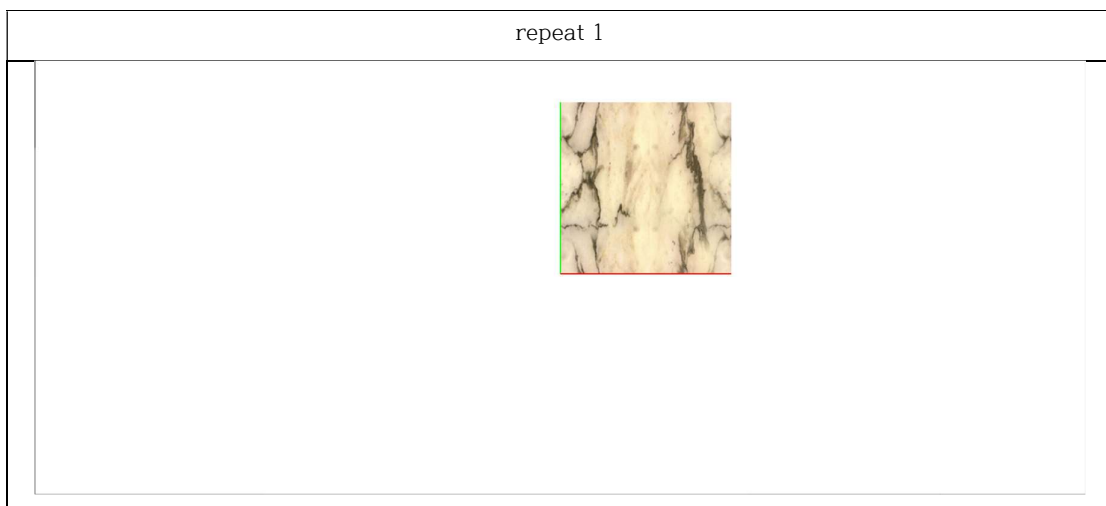


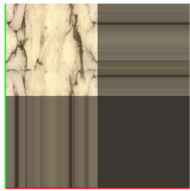
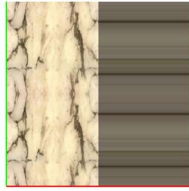
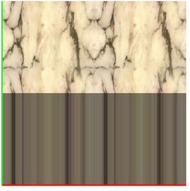
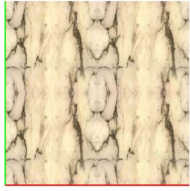
## Practice 04. Texture mapping to a cube



## II. Exercise

### Exercise 01. Texture wrapping using a marble texture



repeat 2 , s: clamp, t: clamp	repeat 2 , s: clamp, t: repeat
	
repeat 2 , s: repeat, t: clamp	repeat 2 , s: repeat, t: repeat
	

#### 텍스처 래핑 방식 설정

```

switch (selection)
{
case 1: glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT); break;
case 2: glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP); break;
case 3: glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_CLAMP);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT); break;
case 4: glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_CLAMP); break;
case 5: glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_S, GL_REPEAT);
        glTexParameteri(GL_TEXTURE_2D, GL_TEXTURE_WRAP_T, GL_REPEAT); break;
}

```

## 텍스처 필터 및 반복 설정

```
// Texture filter
case GLFW_KEY_1:selection = 1; textureNumRepeats = 1; loadRawTexture("m02_marble.raw", 512, 512, 3); break;
case GLFW_KEY_2:selection = 2; textureNumRepeats = 2; loadRawTexture("m02_marble.raw", 512, 512, 3); break;
case GLFW_KEY_3:selection = 3; textureNumRepeats = 2; loadRawTexture("m02_marble.raw", 512, 512, 3); break;
case GLFW_KEY_4:selection = 4; textureNumRepeats = 2; loadRawTexture("m02_marble.raw", 512, 512, 3); break;
case GLFW_KEY_5:selection = 5; textureNumRepeats = 2; loadRawTexture("m02_marble.raw", 512, 512, 3); break;
}
```

### 1. 텍스처 매핑 설정

코드의 핵심은 `glTexParameterf` 를 통해 텍스처의 래핑 및 필터링 방식을 설정하는 부분이다. 아래는 이를 간단히 설명한다.

- **텍스처 래핑 방식** (`GL_TEXTURE_WRAP_S`, `GL_TEXTURE_WRAP_T`)

텍스처 좌표가 범위를 벗어났을 때의 처리 방식을 지정한다. 예제에서는 다음과 같은 옵션을 사용하였다:

- `GL_CLAMP`: 텍스처의 가장자리 색상을 반복 없이 고정.
- `GL_REPEAT`: 텍스처가 범위를 벗어나면 반복적으로 그려짐.

- **코드 분석**

스위치 문에서 `selection` 변수의 값에 따라 래핑 옵션이 달라진다. 예를 들어, `selection = 1` 은 S 좌표는 `GL_CLAMP`, T 좌표는 `GL_REPEAT` 로 설정된다. 이는 다양한 반복 방식의 시각적 차이를 확인하기 위함이다.

### 2. 텍스처 로딩 및 필터링 설정

- **텍스처 데이터 로딩**

텍스처는 `loadRawTexture()` 함수에서 `.raw` 파일을 읽어 OpenGL 의 텍스처 메모리에 업로드된다. 텍스처 크기와 채널 수를 지정해 효율적인 로딩을 구현한다.

- **텍스처 필터링**

필터링 방식으로는 `GL_LINEAR` 가 사용되어 텍스처 확대 및 축소 시 부드러운 변환 효과를 제공한다. 이로 인해 고해상도와 저해상도 텍스처 모두에서 부드러운 품질을 유지할 수 있다.

### 3. 키보드 입력에 따른 텍스처 설정 변경

- **키보드 입력 처리**

`keyboard` 함수는 GLFW 라이브러리를 사용하여 키 입력을

처리한다. 1 에서 5 까지의 키를 통해 텍스처 래핑 방식을 변경할 수 있으며, 이를 통해 다양한 시각적 결과를 얻는다.

- **반복 횟수 조정**

키보드 입력(Shift + 방향키)을 통해 텍스처의 반복 횟수를 동적으로 조정할 수 있다.

### III. 느낀점

OpenGL에서 텍스처 매핑의 다양한 설정 옵션이 렌더링 결과에 미치는 영향을 직접 확인할 수 있었습니다. 텍스처 래핑과 필터링 방식을 자유롭게 변경하면서, 단순한 설정 차이가 결과물의 시각적 품질과 분위기에 큰 차이를 만들어낸다는 점이 인상적이었던 과제였습니다.