

멀티미디어 과제 2

이번 과제에서는 PyOpenGL을 활용하여 텍스처 매핑, 노멀 매핑, 폰 라이팅 모델을 구현합니다.

개발 환경 세팅

다음 명령어를 이용해 필요한 라이브러리를 설치하세요. **참고:** 과제 1과 달리 **trimesh** 라이브러리가 추가되었습니다.

```
pip install numpy pyopengl glfw pyglm
pip install trimesh
```

과제 구현 사항

main.py 파일에서 OpenGL 렌더링 코드를 작성하고, **shader.vert** 및 **shader.frag** 파일에서 각각 Vertex Shader와 Fragment Shader를 구현합니다.

0. 모델 로드

- **utils.py**의 **load_object** 함수는 **IridescenceAbalone** 디렉토리에 포함된 메시, 텍스처 맵, 노멀 맵을 로드합니다.
 - 이 함수는 다음 정보를 반환합니다.
 - vertex attribute 데이터 (**vertex_data**)
 - 텍스처 이미지 (**texture_map**)
 - 노멀 맵 이미지 (**normal_map**)
 - 모델 로드는 직접 구현할 필요는 없지만, 아래 vertex attribute 구성을 참고하세요.
 - 위치(pos): 3개 element
 - 노멀: 3개 element
 - 텍스처 좌표: 2개 element
 - tangent: 3개 element
 - 총 11개 float로 구성됨
-

1. 모델 렌더링 (10점)

1. **var.py**에 정의된 값을 참고하여 모델을 렌더링합니다.

- 이번 과제에서 변환 행렬은 **glm**을 사용해도 되고, 과제 1에서 작성한 함수를 재사용해도 됩니다.
 - 단, **glm**을 사용할 경우 행렬을 넘길 때 ****GL_FALSE****를 설정해야 합니다.
 - (이전에 작성한) **numpy** 기반 행렬이라면 ****GL_TRUE****를 설정해야 합니다.
- View 및 projection 행렬은 **var.py**의 값을 그대로 사용합니다.
- World 변환은 아래 설정을 사용합니다.
 - 1. x, y, z 방향 모두 10배로 스케일링
 - 2. 시간에 따라 시계 반대 방향으로 y축 회전 (속도는 너무 빠르거나 느리지 않게 설정)

2. 텍스처 매핑 (20점)

1. `PIL.Image` 타입의 텍스처 맵(`texture_map`)을 받아 OpenGL 텍스처 객체를 생성하는 `create_texture` 함수를 작성하세요.
 2. `main.py`와 셰이더 코드를 수정하여 텍스처 매핑을 구현하세요.
 - Texture filtering 및 wrapping 방식은 자유이나, `LINEAR` 및 `MIPMAP_LINEAR` 사용을 권장합니다.
-

3. 폰 라이팅 모델 (30점)

1. 폰 라이팅 모델을 폰 셰이딩 방식으로 구현합니다.
 - ambient, diffuse, specular 조명 요소를 포함해야 하고, emissive 요소는 제외합니다.
 - 라이팅 계산 시 필요한 material 파라미터 (`m_s` 등)는 텍스처 색상을 기준으로 가정합니다.
 - 라이팅 계산 시 필요한 라이팅 파라미터는 `var.py`를 참고하세요.
 - **Gouraud** 셰이딩 방식으로 구현할 경우 0점 처리됩니다.
-

4. 노멀 매핑 (총 40점)

1. 노멀 매핑 기능 구현 – 30점
 - Tangent space 노멀 매핑을 구현합니다 (Object space는 0점)
 - Vertex attribute의 tangent를 사용합니다.
 2. 키보드 입력으로 노멀 매핑 On/Off 토글 – 10점
 - 프로그램 시작 시 노멀 매핑은 비활성화 상태입니다.
 - 숫자 **2번 키**를 누르면 노멀 매핑이 적용됩니다.
 - 숫자 **1번 키**를 누르면 노멀 매핑이 해제됩니다.
 - **힌트:** 셰이더 코드에서도 C와 유사하게 `if`문을 사용하여 분기 처리가 가능합니다.
-

참고

Shader 코드는 아래와 같은 built-in 함수들이 있습니다.

- `dot`
 - `pow`
 - `normalize`
 - `texture`
 - `reflect`
 - `max`
 - `cross`
-

제출 방법

1. 제출 파일: `main.py`, `shader.vert`, `shader.frag` 파일 3개만 압축(zip)하여 제출
 - 두 파일 외 다른 파일 수정 금지
 - 압축 파일 이름은 `학번_이름.zip`
2. 제출 기한: **2025-04-20 (일) 23:59:59까지**

주의 사항

- 제출 기한 초과 시 받은 점수의 **50%**만 인정합니다.
- README에서 명시하지 않은 추가 라이브러리를 사용하면 **0점** 처리합니다.
 - 단, 만약 필요하다면 python 기본 라이브러리는 (pip install로 설치하지 않아도 되는) 사용해도 무방합니다.
- **zip** 이외의 다른 압축 형식을 사용하면 **0점** 처리합니다.
- 압축 파일 이름을 다른 형식으로 네이밍할 시 감점 처리합니다.