

Computer Graphics - Assignment #1 (15점)

1. 사이버 캠퍼스를 통하여 제출하십시오.

- ① 사이버 캠퍼스의 컴퓨터 그래픽스 강의에 접속하시어 과제 탭에서, 과제 1에 파일을 첨부하여 제출하시면 됩니다.
- ② **제출 파일은 js 파일 하나**입니다. HTML과 클래스 모듈 JS파일들은 레포지토리에서 제공된 것 그대로 사용하시면 됩니다.
- ③ Main.js파일을 만들어 구현한 뒤, 제출 전에 **파일 이름을 “이름_학번.js”로 바꾸어 제출해 주십시오.** (ex, “김형기_202012345.js”)
- ④ 제출 파일은 표절 검사 프로그램을 거칩니다. 표절로 판단될 경우 0점 처리합니다.

2. 제출 기한: **11월 14일 23:55분까지**

- ① **추가 제출(늦은 제출)은 받지 않습니다.** 기한 내에 제출해 주십시오.
- ② 업로드 시간까지 고려하시어 여유 있게 제출해주십시오.

3. 기타 문의사항은 이메일(hk.kim@jbnu.ac.kr) 및 클라썸으로 연락 바랍니다.

<채점 기준>

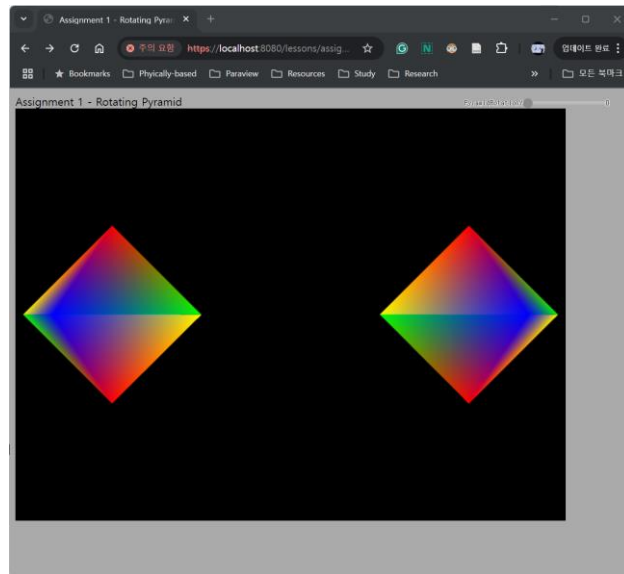
- ① 3차원 각별의 정점 및 인덱스 데이터 올바르게 구현 (6점)
 - ◆ 즉, Indexed Representation을 사용해야 함
- ② 장면을 위한 설정 및 렌더링 과정 구현 (6점)
 - ◆ Model 변환 행렬의 올바른 정의 및 셰이더로의 전달 필요
- ③ UI를 통한 모델 회전 인터랙션 구현 (3점)

<제출 유의 사항>

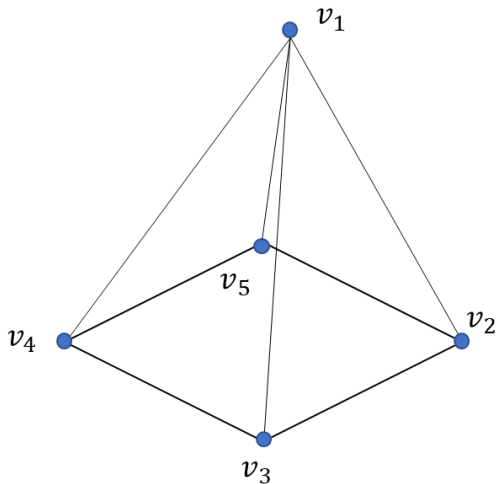
- 제출 파일은 JS파일 하나이므로 셰이더는 JS 파일 안에 포함해서 구현해야 하고, 피라미드 모델의 정점과 인덱스는 코드에서 직접 배열을 사용해 정의해야 합니다.
- 클래스 모듈 import 상대경로 변경하지 마세요
 - Lesson에서처럼 `import Shader from '../_classes/Shader.js';` 경로를 사용하세요.

<문제>

- 그림과 같이 네 개의 각뿔을 사용하여 두 개의 다이아몬드 형상의 물체를 화면에 그리고, Slider UI를 통해 회전시키려 한다.
 - Slider로 값을 변화시키면 두 개의 다이아몬드가 각각 반대 방향으로 회전해야 한다.
 - ◆ 상세한 동작은 동영상 링크 참고(<https://youtu.be/VhQzBMXsLeY>)
 - 네 개의 각뿔의 형상은 완전히 동일하다.



- 주어진 Contents.html 파일과 실습에서 구현한 클래스 모듈을 활용하여 영상과 동일한 화면이 그려지도록 main.js 파일을 작성하라.
 - **Shader, Renderer, Camera, VertexArray, VertexBuffer, IndexBuffer** 클래스 사용
 - 셰이더 소스코드와 각뿔을 정의하는 정점 및 삼각형들은 main.js 파일 내에 모두 구현하며, 각뿔의 정점 위치와 색상은 예를 들어 아래 그림과 같다.



정점	위치	색상
v_1	[0.0, 1.0, 0.0]	[1.0, 0.0, 0.0]
v_2	[1.0, 0.0, 0.0]	[0.0, 1.0, 0.0]
v_3	[0.0, 0.0, 1.0]	[0.0, 0.0, 1.0]
v_4	[-1.0, 0.0, 0.0]	[1.0, 1.0, 0.0]
v_5	[0.0, 0.0, -1.0]	[0.0, 1.0, 1.0]

<기타 유의 사항 (유의 사항 위반시 감점)>

- 정점 배열은 하나면 충분함!! 5개의 정점 좌표(색상까지 포함하여 30개의 float)을 저장한 배열 하나만을 사용해서 구현 하시오.
- 슬라이더의 최소값은 0도 회전, 최대값은 360도 회전이 되도록 하시오.
 - 즉, 슬라이더 왼쪽 끝과 오른쪽 끝에서 렌더링 되는 장면이 동일해야 함
- 첨부한 Contents.html에 대해서 동작해야 합니다.
 - 즉, 위 내용을 구현하는 데 있어서 html 파일의 수정은 필요하지 않습니다.
- `gl.enable(gl.CULL_FACE)` 옵션을 사용하여 back face culling이 수행되도록 하시오.
- 배경은 예시와 같이 검은색으로 그리시오.
- 각뿔의 위치 및 캔버스의 크기는 첨부한 그림과 동일하지 않아도 상관없음.
 - 단, 실행했을 때 카메라 시야 안에 네 개의 각뿔이 모두 보이는 상태여야 함
 - 참고로, 첨부한 그림에서는 fov 60도, 카메라 eye 위치는 [0,1,4]인 상태임
- 왼쪽 위/아래, 오른쪽 위/아래 각뿔의 색상까지 첨부 그림과 같을 필요 없음
 - 단, 각뿔을 구성하는 각각의 정점에 해당하는 색상은 예시와 같이 서로 달라야 함