

## 컴퓨터 그래픽스 과제 #1

1. Vertex Shader 및 OpenGL 기본 학습을 위한 과제입니다.
2. 제출은 kedilms 페이지의 과제 제출 시스템을 사용하여 제출하십시오.
  - ① 제출 파일은 main.cpp/shader.cpp/shader.h/실행파일(exe) 의 4가지 파일이 기본입니다. 이외 다른 파일을 수정하였을 경우 사유와 함께 수정한 모든 파일을 제출 해주세요.
  - ② 제출 파일은 표절 검사 프로그램을 거칩니다. 표절로 판단될 경우 0점 처리합니다.
3. 첨부한 예제 프로그램을 주의 깊게 테스트 해 보시고, 동일하게 구현 하십시오. 예제 프로그램은 채점 기준 및 추가점수를 모두 만족하도록 구현되어 있습니다.
4. 제출 기한: 시스템에 제출 시간이 자동으로 기록됩니다. **10월 21일 23:55 분 이전까지 제출하셔야 합니다.**
  - ① 늦게 제출하실경우 1시간당 1점씩 감점합니다.
  - ② 시스템에 기록된 시간은 업로드 완료 시점입니다. 감안하시고 여유 있게 제출 해주십시오.
5. 기타 문의사항은 이메일(hk.kim@jbnu.ac.kr)로 연락 바랍니다.

### 채점 기준

- ① 요구 조건대로 프로그램 정상 동작 (10점)
  - ◆ Exe 파일이 표준적인 조건에서 채점 가능하도록 shader는 상대 경로로써 Shaders/shader.vert 파일을 참조하도록 수정하지 마세요.
- ② (추가점수) 프레임 갱신 속도(fps)에 의해 이동 및 회전 속도가 변하지 않도록 deltaTime을 사용 (+2점)
- ③ (추가점수) Q와 E 버튼을 이용하여 상하좌우 이동이 가능하도록 기능을 구현 (단순 up vector 방향 이동이 아님) (+1점)

## 문제

- 게임 또는 여러 3차원 응용 프로그램에서 카메라의 이동 기능은 필수적이다. Vertex shader를 수정하여 카메라의 이동에 따른 화면 갱신 기능을 추가하려 한다.
- Vertex shader에 view transform matrix를 전달해 카메라의 이동에 따른 화면 변환을 적용할 수 있다.
  - View transform matrix는 glm::lookAt() 함수를 사용하여 계산한다.
  - glm::lookAt() 함수에 EYE, AT, UP 벡터를 인자로 넘겨주면 view transform matrix를 반환한다.
- EYE 벡터 & AT 벡터의 계산
  - EYE 벡터는 카메라의 위치, AT 벡터는 카메라의 위치 + Forward 벡터이다.
  - UP 벡터는 {0,1,0}으로 고정한다.
  - EYE 벡터의 이동은 W,S,A,D키 입력을 사용한다.
    - ◆ W키 : EYE 벡터를 Forward 방향으로 이동
    - ◆ S키 : EYE 벡터를 - Forward 방향으로 이동
    - ◆ A키 : EYE 벡터를 -Right 방향(아래 설명)으로 이동
    - ◆ D키 : EYE 벡터를 Right 방향으로 이동
- Forward 벡터 & Right 벡터의 계산
  - Forward 벡터는 {0,0,-1}이 초기값이며 마우스를 X방향(좌우)으로 움직일 경우  $\varphi$ 만큼, Y방향(상하)으로 움직일 경우  $\theta$ 만큼 회전한다. 이 때,  $\varphi$ 와  $\theta$ 는 구면 좌표계에서 forward 벡터의 polar angle과 azimuth angle이다.
    - ◆ 구면 좌표계 : [https://en.wikipedia.org/wiki/Spherical\\_coordinate\\_system](https://en.wikipedia.org/wiki/Spherical_coordinate_system)
  - 카메라의 UP 벡터는 {0,1,0}으로 고정되어 있으므로, 카메라의 오른쪽 방향인 right 벡터를 아래와 같이 정의할 수 있다.
    - ◆  $\text{right} = \text{forward} \times \text{up}$ , where  $\times$  means cross product