

컴퓨터 그래픽스 과제 #3

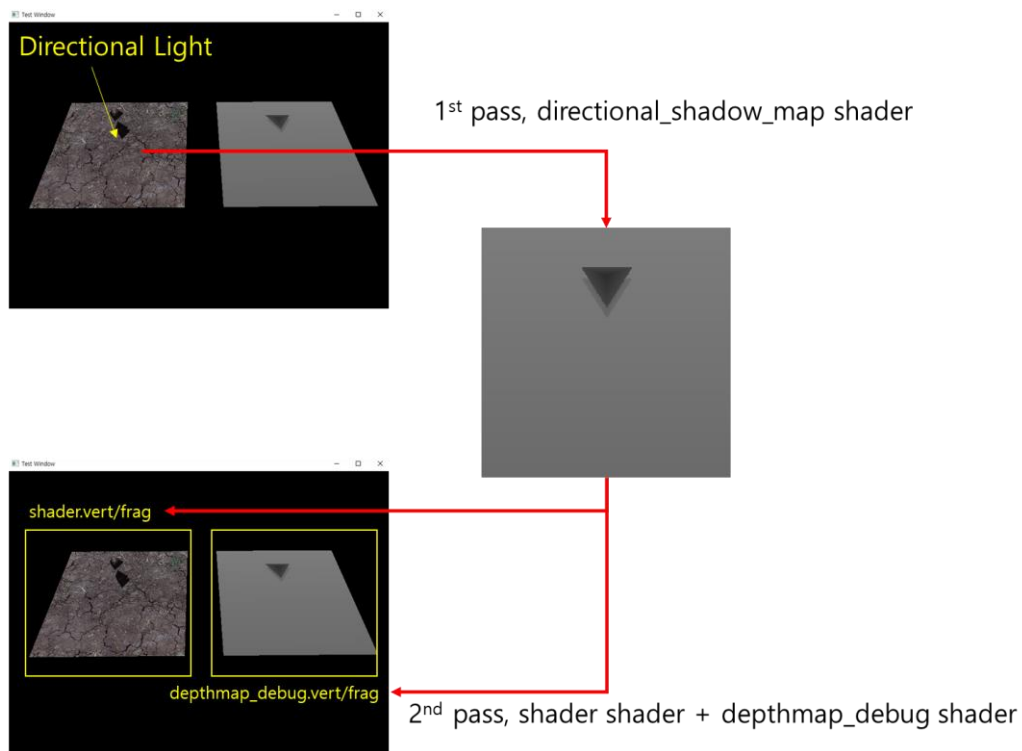
1. DepthMap과 OpenGL 이해를 위한 과제입니다.
2. 제출은 kedilms 페이지의 과제 제출 시스템을 사용하여 제출하십시오.
 - ① 제출 파일은 depthmap_debug.vert/depthmap_debug.frag/main.cpp 파일입니다. 이외 다른 파일을 수정하였을 경우 사유와 함께 수정한 모든 파일을 제출 해주세요.
 - ② 제출 파일은 표절 검사 프로그램을 거칩니다. 표절로 판단될 경우 0점 처리합니다.
3. 첨부한 예제 프로그램을 주의 깊게 테스트 해 보시고, 동일하게 동작하는지 확인 후 제출하십시오.
4. 제출 기한: 시스템에 제출 시간이 자동으로 기록됩니다. **12월 16일 23:55분 이전까지 제출하셔야 합니다.**
 - ① 늦게 제출하실 경우 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)

문제

- 첫 번째 pass에서 렌더링된 depth map을 화면에 보이도록 프로그램을 작성하시오.
- 첨부한 Assignment3 Project 를 기반으로, depthmap_debug.vert 및 depthmap_debug.frag를 구현한다.
- 강의 시간에 배운 바와 같이, shadow mapping은 두 단계 pass로 구현되며, 첫 번째 pass에서는 깊이 값을 저장하는 shadow map이 생성되며, 두 번째 pass에서는 이러한 shadow map에 저장된 깊이 값과 카메라에서 보이는 fragment와의 깊이 값 비교를 통해 그림자를 표현한다.
- Shadow map이 제대로 생성 되었는지 확인하기 위해, 평면을 하나 만들고, 그 평면에 shadow map을 image texturing하여 가시화하고자 한다.



- 프로그램에서는 총 3종류의 셰이더가 사용된다.
 1. Directional_shadow_map.vert/frag (구현/수정 불필요)
 - ◆ Shadow map을 생성하는 첫 번째 pass shader
 2. Shader.vert/frag (구현/수정 불필요)
 - ◆ 그림자 효과를 더하여 물체를 렌더링하는 두 번째 pass shader
 3. **Depthmap_debug.vert/frag (구현 필요)**
 - ◆ Shadow map 데이터를 평면에 texturing하기 위한 shader

- **Depthmap_debug.vert/frag shader를 구현하고, 이 셰이더에 필요한 정보들을 전달할 수 있도록 main.cpp 내 RenderDepthMap() 함수의 본문을 구현하시오.**

1. 위와 같이 shadow map을 이미지 텍스처처럼 활용할 때 필요한 정보가 무엇인지 생각해 보시오.
2. Shadow map을 사용하는 shader.shader/vert로 데이터를 넘겨주는 RenderPass() 함수를 참고하시오.
3. 키보드의 화살표 키를 누르면, directional light의 방향이 변하도록 구현해 두었다. 이러한 방향 변화에 따라 shadow map이 변화하는지 확인하시오.