|  |
| --- |
| **REPORT** |

****

|  |  |  |  |
| --- | --- | --- | --- |
| **과목명** | **컴퓨터그래픽스(SW)-1분반** | **담당교수** | **송인식** |
| **학 번** | **32221741** | **전 공** | **소프트웨어** |
| **이 름** | **박성현** | **제 출 일** | **2023-11-19(일)** |

**과제 3: 조명과 관찰자 제어**

내용

[1. 개발환경 3](#_Toc151246613)

[2. 설계 3](#_Toc151246614)

[3. 구현 4](#_Toc151246615)

[4. 소스 코드 9](#_Toc151246616)

1. 개발환경

**OS** : Windows 10/11

**Editor** : Visual Studio Code

**Browser** : Microsoft Edge

1. 설계
2. 과제 2로 작성한 ‘rotatingPyramid.html’, ‘rotatingPyramid.js’를 기반으로 과제 3을 수행한다.
3. [cs.unm.edu/~angel/BOOK/INTERACTIVE\_COMPUTER\_GRAPHICS/SEVENTH\_EDITION/CODE/06/shadedCube.html](https://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SEVENTH_EDITION/CODE/06/shadedCube.html)와[cs.unm.edu/~angel/BOOK/INTERACTIVE\_COMPUTER\_GRAPHICS/SEVENTH\_EDITION/CODE/06/shadedSphere3.html](https://www.cs.unm.edu/~angel/BOOK/INTERACTIVE_COMPUTER_GRAPHICS/SEVENTH_EDITION/CODE/06/shadedSphere3.html)를 참고하여 코드를 수정한다.
   1. rotatingPyramid와 구조적으로 비슷한 shadedCube을 참고하여 조명을 추가한다.
   2. shadedSphere3를 참고하여 관찰자를 추가한다.
   3. 제어하는 버튼을 더 추가한다.
4. 구현

**rotatingPyramid.html**

텍스트, 스크린샷, 번호, 문서이(가) 표시된 사진

자동 생성된 설명

shadedCube.html에서 참고하여 작성한 조명을 구현하는 vertex-shader 스크립트 부분이다. 원래 여기에서 회전 행렬으로 vPosition을 회전시켰으나 js의 render()에서 회전시키도록 변경하였다. 그래서 조명은 고정하고 피라미드만 회전시킬 수 있었다. 여기서 포인트는 gl\_Position은 프로젝션 \* 모델뷰 \* vPosition 순으로 곱해야 한다는 것이다. 그리고 fColor를 ambient + diffuse + specular에 vColor를 곱하여 구하는 것이다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

기존 rotatingPyramid.html에 있던 버튼에서 회전을 토글하는 tButton, shadedSphere3.html에서 가져온 Button0~5를 추가하였다.

**rotatingPyramid.js**

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

원래 함수 안에 있던 vertices와 vertexColors의 선언을 전역으로 끄집어내어 리팩토링하였다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

관찰자에 관련한 변수를 예제에서 가져왔다. theta는 기존과 이름이 겹치기 때문에 기존 theta를 rotTheta로 이름을 변경하였다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

조명 관련 변수도 예제에서 가져와 살짝 변경하였다. 내가 작성한 피라미드 코드의 경우 각 면마다 색상을 정하였으므로 벡터의 한 요소가 0.0이 되면 나중에 vColor와 ambientProduct, diffuseProduct 등을 결합하여 최종 색상을 결정할 때 어떤 색상만 조명 효과를 못받게된다. (0을 곱하면 무조건 0이되는 성질로 인하여)

텍스트, 폰트, 스크린샷, 라인이(가) 표시된 사진

자동 생성된 설명

모델-뷰 매트릭스와 프로젝션 매트릭스를 위한 변수도 만들어준다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

colorPyramid() 함수는 그대로지만 triple() 함수는 수정이 필요하였다. 일단 법선 벡터를 계산하여 nomals 배열에 push해주었다. 그리고 정점 배열에 정점들을, 색상 배열에 색상을 push 해 주었다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

법선 벡터에 관한 버퍼를 만들어준다. 이는 예제 코드를 참고하여 작성하였다.



모델-뷰 행렬, 프로젝션 행렬 uniform 변수들의 위치를 저장해준다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

앞서 전역으로 만들어준 조명의 요소들로 행렬곱을 해 ambient, diffuse, specular의 product 값을 저장한다.

텍스트, 폰트, 번호, 스크린샷이(가) 표시된 사진

자동 생성된 설명

추가한 버튼들의 onclick 이벤트 리스너를 달아준다. 이도 역시 예제에서 가지고 왔다.

텍스트, 스크린샷, 폰트, 라인이(가) 표시된 사진

자동 생성된 설명

앞서 행렬곱을 마친 ambient, diffuse, specular와 lightPosition, materialShininess를 uniform 변수에 전달한다.

텍스트, 스크린샷, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

render() 내부에 추가한 코드들이다. eye를 계산하고 앞서 저장한 at, up과 같이 lookAt()으로 카메라 행렬을 만들어 모델-뷰 매트릭스에 저장한다. 그리고 앞서 말했듯 행렬 곱을 통해 원래 vertex shader에서 수행했던 회전을 수행한다. 프로젝션 매트릭스에는 ortho()로 직교투영하도록 하였다. left, right 등의 인자는 앞서 전역으로 만들어 둔 것이다. 그리고 모델-뷰 매트릭스와 프로젝션 매트릭스를 uniform 변수로 전달한다.

1. 실행 결과 캡쳐

삼각형이(가) 표시된 사진

자동 생성된 설명

조명이 제대로 적용되었고, 관측자를 제어하는 버튼도 잘 작동함을 알 수 있었다.

1. 소스 코드

**소스코드는 GitHub 레포지토리에 있습니다.**

[**https://github.com/ParkSeonghyeon2003/DKU\_ComputerGraphics**](https://github.com/ParkSeonghyeon2003/DKU_ComputerGraphics)