**기초 컴퓨터 그래픽스**

**HW3 README**

학번 20171663 이름 이도훈

**1. [환경 명세]**

1) 본인 프로그램의 실제 구동 환경을 명시 할 것 (OS, CPU, GPU, Compiler 등등)

window10 64bit, Intel(R) Core(TM) i5-10400, rtx 2060 super,

visual studio 2019-win64, release

**2. [요구사항]**

   1. Modeling Transformation

1) 먼저 가상의 3차원 세상의 바닥과 좌표의 기준이 되는 세상 좌표계를 그려라 (최대 10점)

- 확인 방법:

Ctrl + F5를 통해 프로그램 실행 시 화면의 중간에 세상의 바닥이 도시되고 바닥 중간에 세상좌표계가 그려진다.

2) 최대 5개까지의 서로 다른 정적인 물체를 서로 다른 모델링 변환을 사용하여 가상의 세상에 배치하라 (물체 당 5점 최대 25점)

a) 사용한 물체: 아이언맨

- 부여한 **서로 다른** 모델링 변환: translate과 rotate, scale을 이용하여 배치하고자 한 위치에 배치했다.

- 확인 방법: 프로그램 실행 시 1시 방향에 위치해 있다

b) 사용한 물체: 고질라

- 부여한 **서로 다른** 모델링 변환: translate과 rotate, scale을 이용하여 배치하고자 한 위치에 배치했다.

- 확인 방법: 프로그램 실행 시 아이언맨에서 -X 방향에 있다

c) 사용한 물체: 버스

- 부여한 **서로 다른** 모델링 변환: translate과 rotate, scale을 이용하여 배치하고자 한 위치에 배치했다.

- 확인 방법: 프로그램 실행 시 고질라의 왼쪽 손 부분에 있다.

d) 사용한 물체: 바이크

- 부여한 **서로 다른** 모델링 변환: translate과 rotate, scale을 이용하여 배치하고자 한 위치에 배치했다.

- 확인 방법: 프로그램 실행 시 고질라의 오른쪽 손 부분에 있다.

e) 사용한 물체: 탱크

- 부여한 **서로 다른** 모델링 변환: translate과 rotate, scale을 이용하여 배치하고자 한 위치에 배치했다.

- 확인 방법: 프로그램 실행 시 아이언맨 밑에 있다.

3) 최대 4개까지의 서로 다른 동적인 물체를 가상의 세상에 배치하라 (물체 당 10점 최대 40점). 각 동적 물체는 이동 변환, 크기 변환, 그리고 회전 변환 등의 기본 기하 변환 중 최소한 두 개 이상을 사용하여 서로 다른 움직임을 표현해야 하며, 각 동적 물체는 키보드 또는 마우스 동작을 통하여 움직임과 멈춤을 조절할 수가 있어야 한다 (자신이 선택한 최대 네 개의 동적인 물체에 대해 이 기능이 구현이 안되어 있으면 물체 당 4점 감점). 요구 사항은 아니나 동적인 물체들 중 최소한 1개의 물체에 대해서는 뉴턴의 운동의 법칙과 같이 물리적으로 충실한 방법을 사용하여 움직임을 표현해볼 것.

‘a’를 누르면 움직임이 멈추고, 다시 ‘a’를 눌러 움직이게 한다.

a) 사용한 물체: 거미

- 부여한 **서로 다른** 모델링 변환: translate과 rotate를 이용하여 롤러코스터를 타는 듯한 효과를 냈다

- 확인 방법: 프로그램 실행 시 화면 중앙에서 9시 방향으로 진행한다

‘9’을 누르면 움직임이 멈추고, ‘8’을 눌러 다시 움직이게 한다,

b) 사용한 물체: 호랑이

- 부여한 **서로 다른** 모델링 변환: translate과 rotate를 이용하여 같은 지역을 배회하는 듯한 효과를 냈다.

- 확인 방법: 프로그램 실행 시 화면 중앙에서 시계방향으로 회전한다.

‘7’을 누르면 움직임이 멈추고, ‘6’을 눌러 다시 움직이게 한다,

c) 사용한 물체: 늑대

- 부여한 **서로 다른** 모델링 변환: translate과 scale을 이용하여 늑대가 점프하며 다가오는 듯한 효과를 냈다.

- 확인 방법: 프로그램 실행 시 1시 방향에서 7시 방향으로 진행한다.

‘l’을 누르면 움직임이 멈추고, ‘k’를 눌러 다시 움직이게 한다,

d) 사용한 물체: 벤(사람)

- 부여한 **서로 다른** 모델링 변환: translate과 rotate를 이용하여 운동장을 막 뛰어다니는 듯한 효과를 냈다.

- 확인 방법: 프로그램 실행 시 화면 중앙에서 반시계방향으로 회전한다.

‘j’를 누르면 움직임이 멈추고, ‘h’를 눌러 다시 움직이게 한다,

2. Viewing Transformation

1) 1번부터 4번까지의 카메라는 CCTV 카메라와 같이 주어진 위치에 고정하여 세상을 바라보는 카메라이다. 적절한 사용자 인터페이스 동작을 통하여 원하는 카메라에서 세상을 바라볼 수 있도록 하라

a) 1번 카메라 확인 방법: 숫자 키 ‘1’을 누른다.

b) 2번 카메라 확인 방법: 숫자 키 ‘2’을 누른다.

c) 3번 카메라 확인 방법: 숫자 키 ‘3’을 누른다.

d) 4번 카메라 확인 방법: 숫자 키 ‘4’을 누른다.

+a 처음 프로그램 실행했을 당시의 카메라 시점을 보고싶으면 숫자 키 ‘0’을 누른다

2) 5번 카메라는 동적인 카메라로서 사용자 인터페이스 동작을 통하여 다음과 같이 움직일 수 있도록 하라

★ ★먼저 숫자 키 ‘5’를 눌러 해당하는 카메라 시점으로 만들어준다. ★ ★

a) 5번 카메라 translation 확인 방법:

‘u’와 ‘t’를 사용하여 u 축으로 이동하는 것을 확인한다.

‘v’와 ‘c’를 이용하여 v 축으로 이동하는 것을 확인한다.

‘n’과 ‘b’를 이용하여 n 축으로 이동하는 것을 확인한다.

b) 5번 카메라 rotation 확인 방법:

‘x’를 누르고 왼쪽 클릭한 뒤 마우스를 좌우로 이동하면 u 축을 둘레로 회전한다.

‘y’를 누르고 왼쪽 클릭한 뒤 마우스를 좌우로 이동하면 v 축을 둘레로 회전한다.

‘z’를 누르고 왼쪽 클릭한 뒤 마우스를 좌우로 이동하면 n 축을 둘레로 회전한다.

(초기 상태는 y이다.)

3) (추가) 1번 카메라에 대하여 고정된 위치를 중심으로 시선의 방향을 바꿀 수 있도록 하라. 어떠한 방식으로 구현할 지는 본인이 결정할 것.

a) 추가 구현 확인 방법:

방향키 상, 하, 좌, 우를 눌러 카메라가 고정된 위치를 중심으로 시선의 방향을

바꾸는 것을 확인한다.

3. Projection Transformation

1) 적절한 사용자 인터페이스 동작을 통하여 5번 카메라에 대하여 줌 인/줌 아웃 기능을 구현하라. 이때, 최대로 줌 인/줌 아웃할 수 있도록 적절히 범위를 설정하라.

★ ★먼저 숫자 키 ‘5’를 눌러 해당하는 카메라 시점으로 만들어준다. ★ ★

- 확인 방법: ‘I’를 눌러 줌 인 되는 것을 확인하고, ‘O’를 눌러 줌 아웃 되는 것을 확인한다.