**Student ID: 2018008659**

**Name: 연승훈**

1. **Introduction to the program, a brief description of how to run it, and how to use it.**

Introduction:

사람 모양의 main object와 강아지 모양의 object 하나, 눈사람 모양의 object 하나를 rendering 합니다. Main object는 Q,W,A,S,Z,X 키로 움직이며 E,R,D,F,C,V 키로 방향을 바꿉니다. 강아지 object는 목줄을 차고있는 것처럼 main object와 일정거리 이상 멀어지려고 하면 그 일정 거리까지 다가옵니다. 눈사람 object는 main object가 다가가서 충돌하면 충돌 직전의 main object가 움직인 방향으로 멀리 날아갑니다. 날아갈 때에는 main object 기준 y축 방향으로 튀어오르며 날아갑니다. 숫자 0키를 누르면 view mode가 전환됩니다.

How to run it:

압축을 푼 후, main.py를 실행하면 됩니다.

How to use it:

위의 introduction과 아래 B에 대한 설명에 있는 key로 사용하면 됩니다.

1. **Description of the implementation of each requirement:** 1) How you implemented that requirement, 2) How TA can verify that the requirement has been implemented, by performing a program's specific feature.. **Requirement items not implemented should be left blank.**
   1. (15 pts) Draw 3 or more 3D objects.

1) obj viewer code를 추가하여 obj file 정보를 읽어와서 sphere와 cube를 그린다. F-1에 따라, object는 animating hierarchical model이다. 3D triangle mesh를 이용하여 vertex array로 render하였다. object는 main object를 포함하여 총 3개이다.

2) camera mode를 바꾸어가며 3개의 object를 확인할 수 있다. (0,0,0)에 main object, (10,0,0)에 두번째 object, (0,0,15)에 세번째 object가 존재한다.

* 1. (15 pts) One of the objects should be "main object". The user should be able to transform the main object using the mouse or keyboard.

1) Global variable로 transform matrix를 두어서 key가 입력될 때 마다 해당 matrix에 transform에 대한 연산을 수행한 뒤 render시에 glMultMatrixf 함수를 호출하여 해당 transform matrix를 곱해준다.

2) 다음의 key를 누르거나 누르고 있으면 해당 transform이 main object에, local frame에 대해 수행된다.

Q: x축 방향으로 -1.0 만큼 translate

W: x축 방향으로 1.0 만큼 translate

A: y축 방향으로 -1.0 만큼 translate

S: y축 방향으로 1.0 만큼 translate

Z: z축 방향으로 -1.0 만큼 translate

X: z축 방향으로 1.0 만큼 translate

E: x축에 대해 -10° rotate

R: x축에 대해 10° rotate

D: y축에 대해 -10° rotate

F: y축에 대해 10° rotate

C: z축에 대해 -10° rotate

V: z축에 대해 10° rotate

T: x축 방향으로 y에 대해 -0.1만큼 shear

Y: x축 방향으로 y에 대해 0.1만큼 shear

U: x축 방향으로 z에 대해 -0.1만큼 shear

I: x축 방향으로 z에 대해 0.1만큼 shear

G: y축 방향으로 x에 대해 -0.1만큼 shear

H: y축 방향으로 x에 대해 0.1만큼 shear

J: y축 방향으로 z에 대해 -0.1만큼 shear

K: y축 방향으로 z에 대해 0.1만큼 shear

B: z축 방향으로 x에 대해 -0.1만큼 shear

N: z축 방향으로 x에 대해 0.1만큼 shear

M: z축 방향으로 y에 대해 -0.1만큼 shear

,(comma): z축 방향으로 y에 대해 0.1만큼 shear

O: x축 방향으로 0.9배 scale

P: x축 방향으로 1.1배 scale

L: y축 방향으로 0.9배 scale

;(semicolon): y축 방향으로 1.1배 scale

.(period): z축 방향으로 0.9배 scale

/(slash): z축 방향으로 1.1배 scale

1: yz평면에 대해 reflect (x값 반전)

2: xz평면에 대해 reflect (y값 반전)

3: xy평면에 대해 reflect (z값 반전)

* 1. (15 pts) The rest of the objects should be automatically moved in response to the movement of the main object or other objects around it.

1) 두번째 object는 강아지 모양을 표현한 object로, F-1에 의해 시간에 따른 animation이 있다. Movement는 render시에 transform matrix를 이용해 구한 main object의 위치와 이 object의 위치 사이의 거리를 구한다. Point – point의 결과로 나온 vector의 길이를 구하는 방식으로 거리를 구한 후, 해당 거리가 10보다 크다면 현재 main object의 방향으로 이동하여 거리를 10으로 맞춘다. 이동하는 위치는 위에서 구한 vector의 크기를 10으로 변환한 뒤, main object의 좌표에 더하는 방식으로 구한다.

세번째 object는 눈사람을 표현한 object로, F-1에 의한 animation이 있으며 F-2에 의해 curve movement를 가지고있다. main object와의 거리가 5 이하가 되면, main object 기준 y축 방향으로 튀어오르며 가장 최근 main object의 움직임의 반대 방향으로 20배의 거리만큼 튕겨나간다. Curve에 대한 내용은 F-2에 설명되어있다.

2) 두번째 object는 main object와의 거리가 10을 벗어나면 main object 방향으로 거리가10이 될때 까지 따라온다. 세번째 object는 main object가 가까이 가면, main object 기준 y축 방향으로 튀어오르며 main object의 최근 이동 방향으로 멀리 날아간다.

* 1. (15 pts) Use perspective projection. The camera should be able to be switched between two modes:

1)

First-person view: (0,0,0)에 위치한 점과 (0,0,1)에 위치한 점에 main object와 동일한 transformation을 적용시켜서 main object의 frame상의 (0,0,0)과 (0,0,1)을 구한다. 그리고 각각을 eye point와 look-at point로 사용한다. Up vector는 위와 같은 방식으로 구한 local frame상의 <0,1,0> vector를 사용한다. 즉, 카메라는 항상 local frame의 z축 방향을 바라보며 위쪽 방향은 local frame의 y축 방향이다. 이 때, main object가 카메라를 가릴 수 있기 때문에 main object는 rendering 하지 않는다. 그래서 main object의 크기가 커짐에 따른 카메라 위치 변화는 없다.

Quarter view: 위와 같은 방법으로 look-at point를 main object의 local frame 상의 (0,0,0)으로 정한다. Eye point는 look-at point에 global frame에 대해 (20,20,20)만큼 이동한 점으로 정한다. Up vector는 항상 <0,1,0>이다. 즉, 카메라는 main object의 원점 위치에 대해 항상 global frame 상으로 (20,20,20) 만큼 떨어진 위치에서 main object를 바라본다.

2) 숫자 0키를 누르면 camera mode가 switching된다. 기본 값은 First-person view이다.

* 1. (15 pts) Use OpenGL lighting / shading.

1) object마다 다른 색을 이용하고, 2개의 광원을 사용한다. (1, -1, -0.5) directional light, (sin(time), 1, cos(time)) directional light를 이용하였다. Time은 실행시간을 의미한다. Shading은 obj flie에서 읽어온 normal vector을 이용하여 shade 하였다.

2) 시간이 지남에 따라 y축 방향에서 원을 그리며 도는 light의 위치가 변한다.(animated light source) object의 각도에 따라 색이 변함을 확인할 수 있다.

* 1. Extra credit

1)

1. animating hierarchical model: main object는 시간에 따라 팔과 다리의 각도가 rotate하도록 하였고, 두번째 object는 시간에 따라 네 개의 다리가 roatate하며, 세번째 object는 시간에 따라 손이 rotate한다.

2. curve to express the movement: main object와 세번째 object(눈사람)의 거리가 5 이하인 경우, 가장 최근 main object의 movement 방향으로, movement의 20배 더 먼 거리로 눈사람이 튕겨나간다. 이 때 튕겨나가는 경로는 Bezier curve이며 main object의 y축 방향으로 볼록한 curve를 그린다. 충돌지점과 착지지점 사이를 Bezier curve를 이용하여 경로를 계산한 후, 도착 전 까지는 해당 curve 위로 이동한다.

2)

1. 세 object 모두 hierarchical model로 구성된 것을 확인할 수 있으며 시간 경과에 따라 main object의 팔 다리, (10,0,0)에 위치한 두번째 object의 네 다리, (0,0,15)에 위치한 세번째 object의 두 팔이 움직이는 것을 확인할 수 있다.

2. main object가 세번째 눈사람 object에 가까이 가면 main object 기준 y축 방향으로 튀어오르며 반대 방향으로 날아가는 것을 확인할 수 있다.