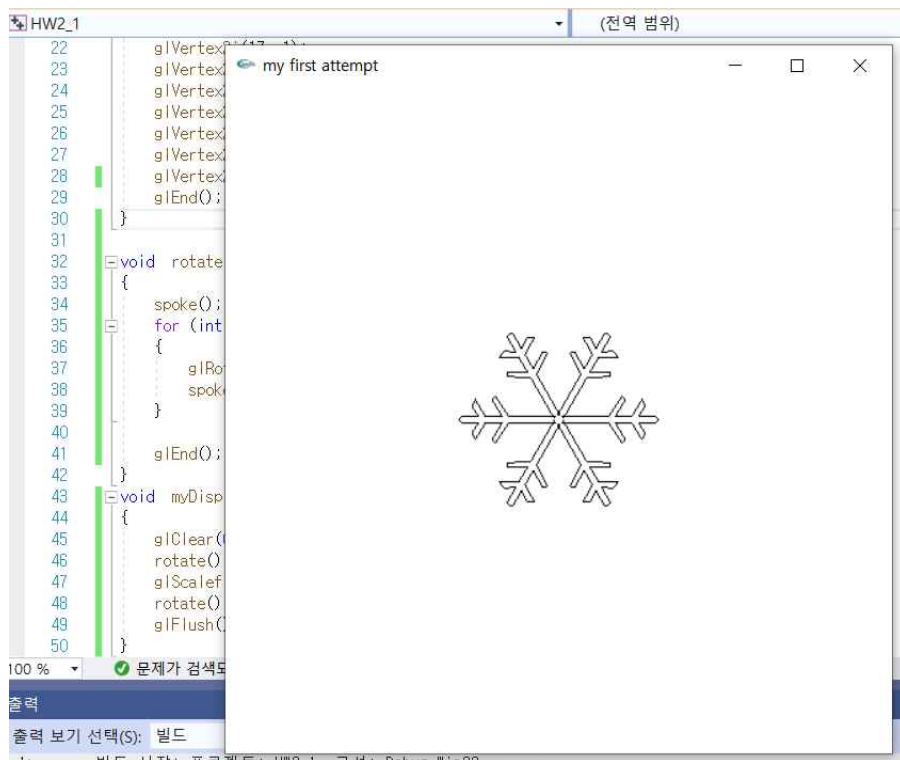


1.

(1) 실행화면



<결과>

(2) 코드설명

임의로 vertex를 계산하여 지정하여 spoke의 위쪽 반을 만들고 6번 회전시켜서 불완전한 눈송이를 만든다. 이 눈송이를 x축 대칭시키면 각자 필요한 부분이 채워져 6개의 완전한 가지를 만들어낸다.

▶가시공간 : (-100,100,-100,100)

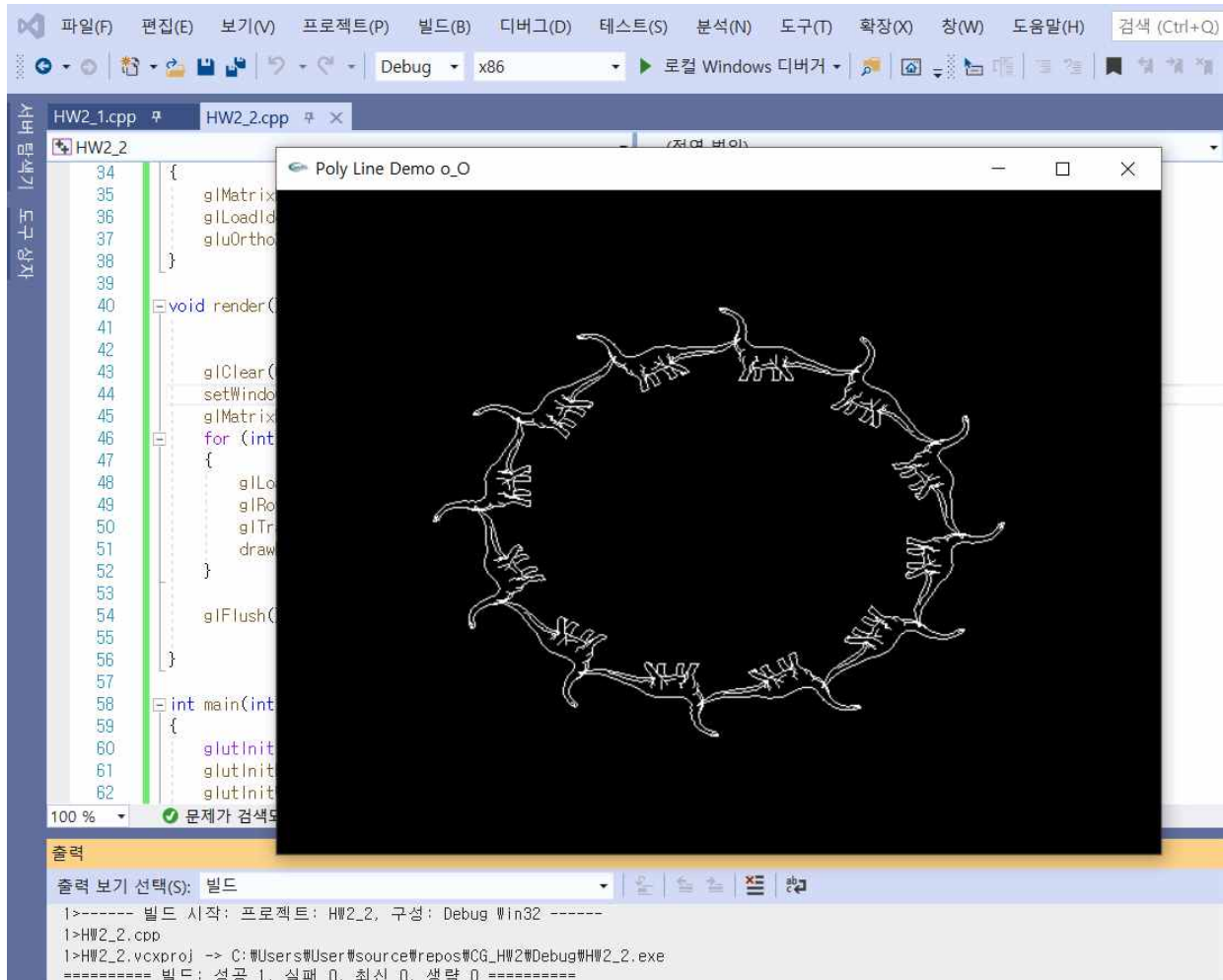
▶변환 - rotation, reflection

(3) 토의사항

spoke를 만들어서 6번 회전하여 위쪽 반을 완성한 후 reflection하라는 조건이 헛갈려서 꽤 헤맸다. 6번을 회전하면 불완전하더라도 아래쪽도 만들어지기에 x축을 기준으로 위쪽만 일단 완벽하게 만든 후 x축 대칭을 하면 되는 것인지, spoke를 reflection하여 완전한 하나의 살을 만들어 회전시키라는 것인지, 아니면 다른 무엇인지 한참 고민했었다. spoke의 위쪽 반을 만들어서 회전을 시켜놓고 있다가 그 상태에서 바로 접어보니 긴 고민이 무색하게 비워진 부분이 바로 채워지는 것을 알 수 있었다.

2.

### (1) 실행화면



### (2) 코드설명

12마리의 공룡이 지구를 걷는 모양이 되기 위해서는 공룡의 발을 중심에서 떨어뜨린 후(이동), 12번 회전시킨다.

복합변환은 순서가 중요하므로 dinosaur그림을 그리는 코드 바로 위에 먼저 이루어지는 이동변환을 작성하고, 이동변환 코드 위에 마지막 회전 변환 코드를 작성한다. 항등행렬로 초기화도 잊지 않는다.

▶가시공간 : (-2000, 2000.0, -2000, 2000.0)

▶변환 - 이동변환 후, 회전변환하는 복합변환

▶CTM -  $CTM = I \rightarrow CTM = I * R = R \rightarrow CTM = R * T \rightarrow CTM = R * (T * d)$

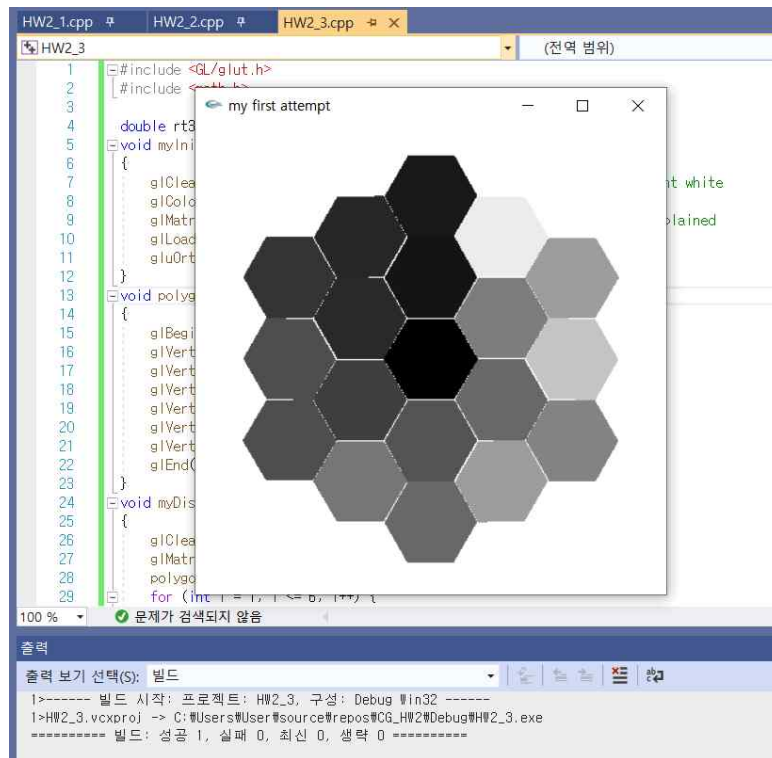
(공룡 = d, I - 항등행렬, R - Rotatef, T - Translatef)

### (3) 토의사항

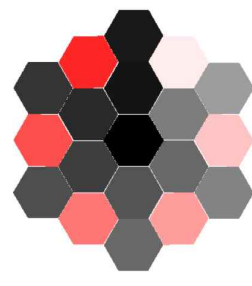
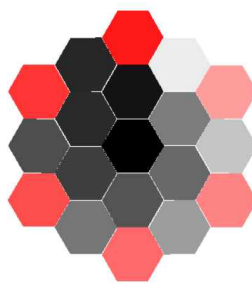
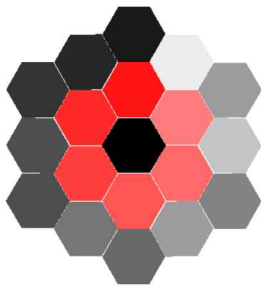
항등행렬로 초기화해주는 코드의 위치에 따라 결과가 상당히 달라졌다. 공룡의 몸통이 돌아간다는지, 공룡이 잘못 그려진다는지, 결과값이 다른 이유 역시 이후에 추가로 알아봐야 할 것이다.

3.

### (1) 실행화면



### (2) 코드설명



CTM을 총 3번 실행했다. 위 그림 3가지 모두 방식은 동일하다. 항등행렬로 초기화한 후에 위치를 이동시키고 60°씩 6번 회전한다. 정육각형 폴리곤이기에 초기 이동 위치를 구하는 것은 어렵지 않다.

- ▶가시공간 : (-100.0, 100.0, -100.0, 100.0)
- ▶변환 : 이동변환 후, 회전변환하는 복합변환
- ▶CTM : ( p는 폴리곤)

$$CTM1 = R(60^\circ) * (T(30, 10 * \sqrt{3}, 0) * p) \mid CTM2 = R(60^\circ) * (T(60, 20 * \sqrt{3}, 0) * p) \mid CTM3 = R(60^\circ) * (T(30, 30 * \sqrt{3}, 0) * p)$$

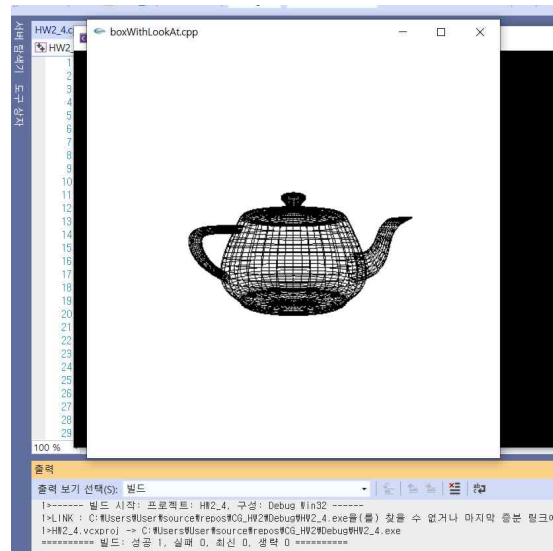
### (3) 토의사항

복합변환 한번에 처리할 수 있는 방법은 생각해내지 못했다. 복합변환을 따로 여러번 하지 않는 방법도 생각해봐야 할 것이다.

4.

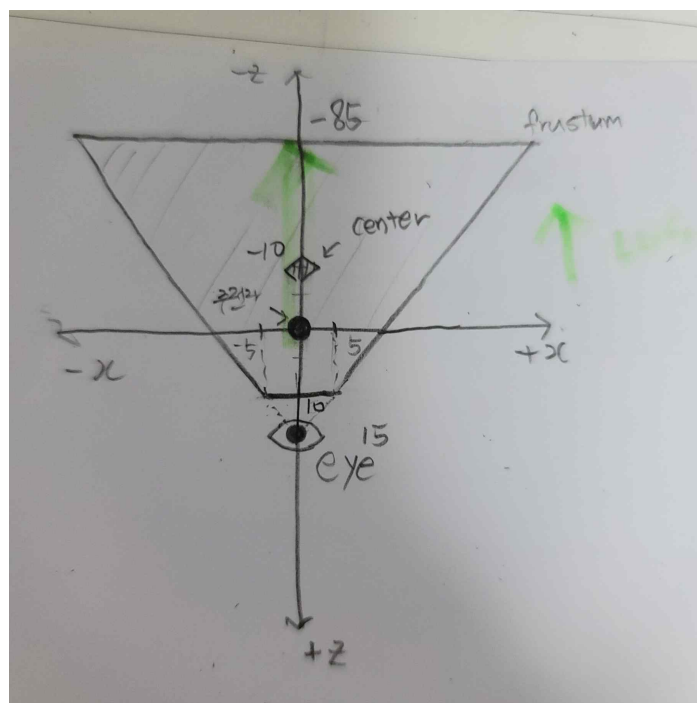
(1) 0.0, 0.0, -10.0

① 출력 결과



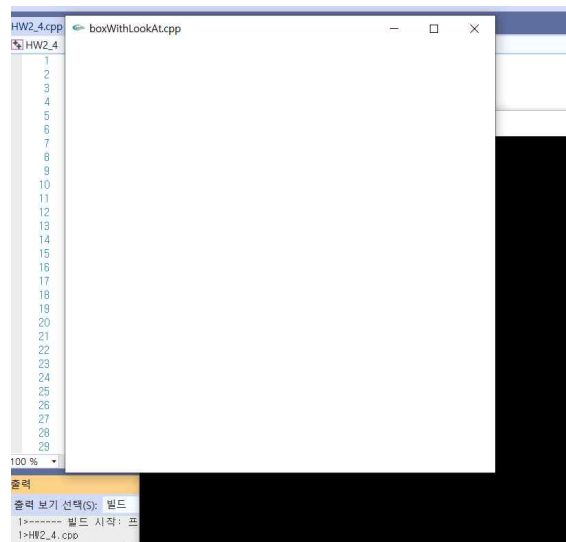
② 설명

카메라의 위치(eye)는 (0,0,15)이다. 카메라가 바라보는 점 center는 (0,0,-10)이며 LOS는 -z축방향이므로 (0,0,0)에 존재하는 주전자가 frustum내에 정방향으로 존재한다.



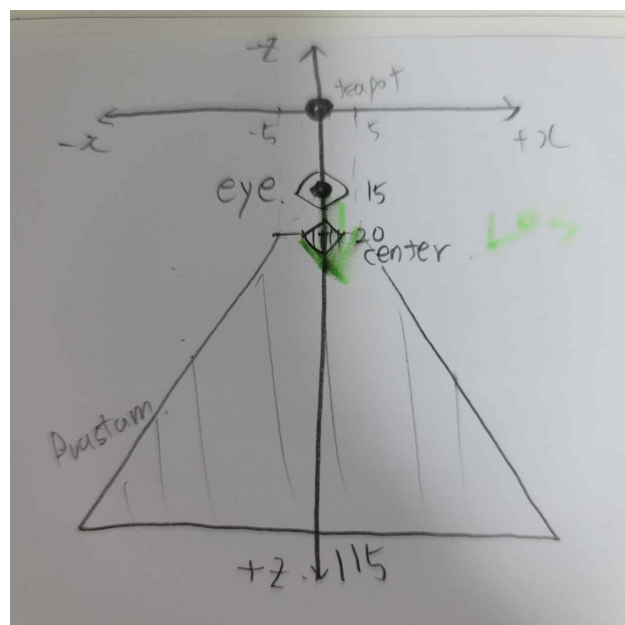
(2) 0.0, 0.0, 20.0

① 출력 결과



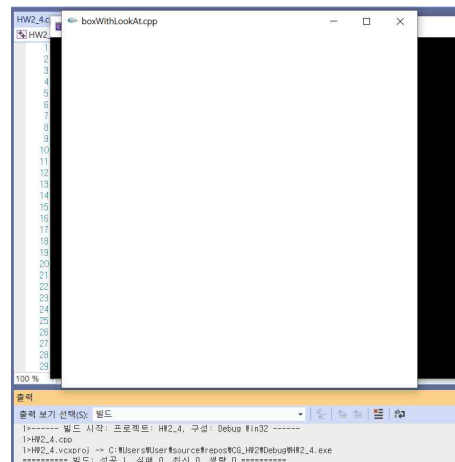
② 설명

카메라의 위치(eye)는 (0,0,15)이다. 카메라가 바라보는 점 center는 (0,0,20)이며 LOS는 z축방향이므로 (0,0,0)에 존재하는 주전자를 등지게 된다. frustum에 존재하지 않아 보이지 않는다.



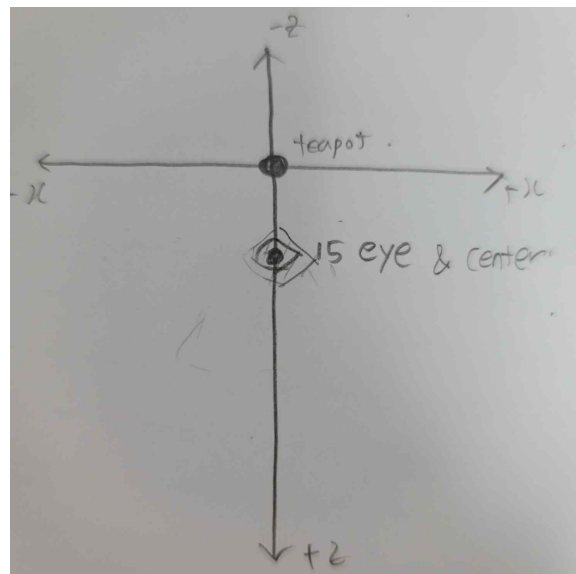
(3) 0.0, 0.0, 15.0

① 출력 결과



② 설명

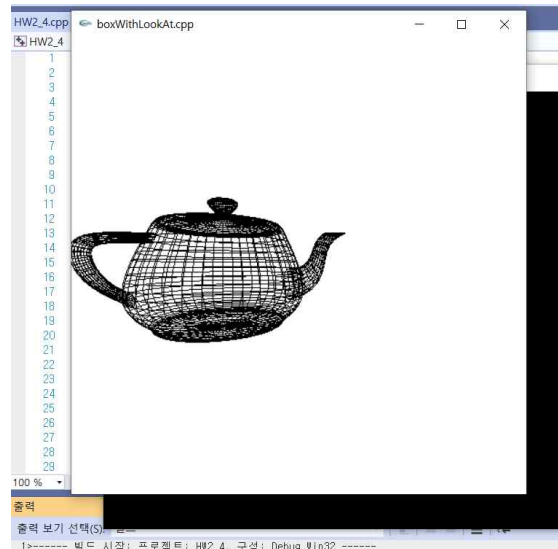
카메라의 위치(eye)는 (0,0,15)이다. 카메라가 바라보는 점 center는 (0,0,15)이며 카메라와 바라보는 점이 일치하게 되어 LOS 벡터는 (0,0,0) 방향을 가지지 않는다. frustum에 존재하지 않아 보이지 않는다.



(4) 5.0, 0.0, 0.0

① 출력 결과

② 설명



카메라의 위치(eye)는 (0,0,15)이다. 카메라가 바라보는 점 center는 (5,0,0)이며 LOS는 (5,0,-15)이므로 약간 x축 양의 방향으로 치우친 +z축방향으로 frustum이 존재하게 되며 해당 범위내에서는 주전자가 왼쪽즈음에 보인다.

