소프트웨어 렌더러 개발 일지.

2025년 04월 21일 작성 시작

너무 오래전에 짜기도 했고, 개발 기록이 남아있지 않아서 어떻게 이걸 만들었는지부터 분석해야 할 것 같음.

추가로, 내가 이걸 만들다가 막힌 부분이 렌더링 파이프라인의 지오메트리(W\*V\*P)부분이라 이부분부터 재작업 들어가야 함.  
DIrectXMath를 오래 사용해 왔기 때문에 일단 이 부분을 자체 수학이 아닌 DirectXMath 함수를 사용해보고 그걸 자체 함수로 바꾸는 식으로 진행하려고 함.

참고로 이 문서는 별도의 퇴고 없이 그냥 진행한다.

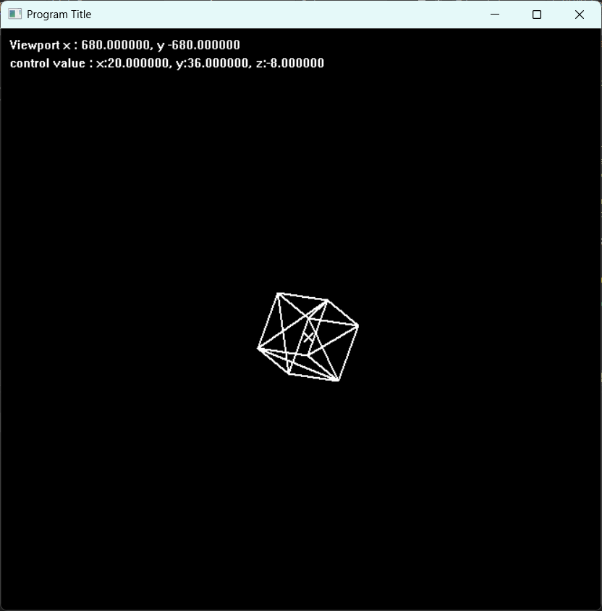
일단 확인 해야 하는거

1. 왜 GDI Renderer의 생성자에서 Cam와 Coner를 초기화 하지? 이거 이상함.
2. 왜 Coner 그리는 부분이 Coner 밖에 있지?

그외

실패한 이유. 투영나누기가 없다.  
Wt \* Vt \* Pt는 잘 되는거 같음

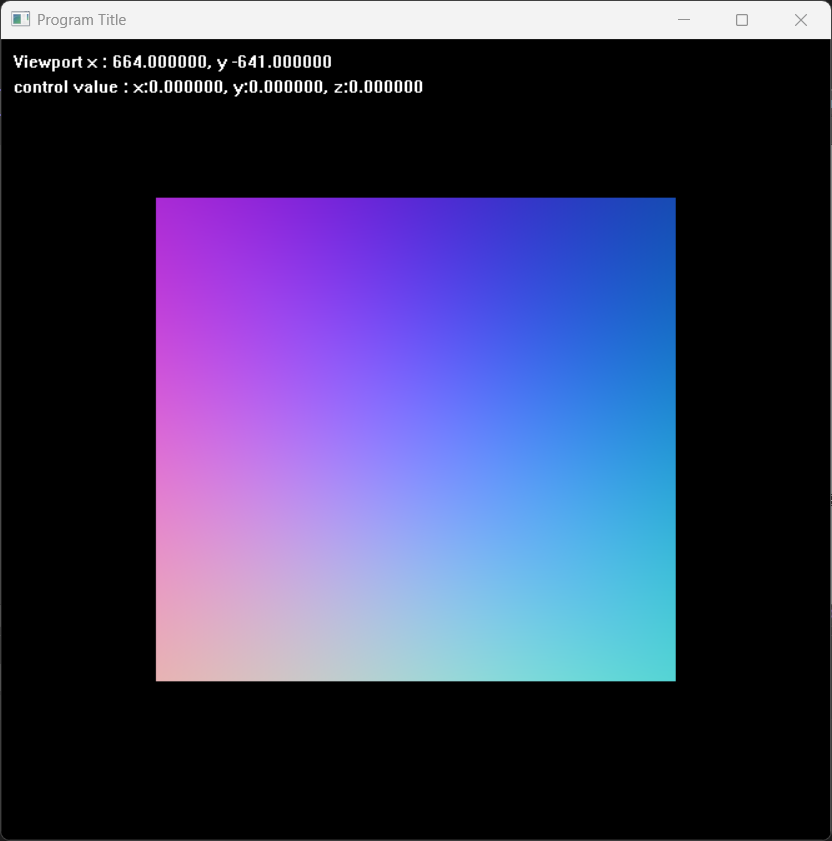
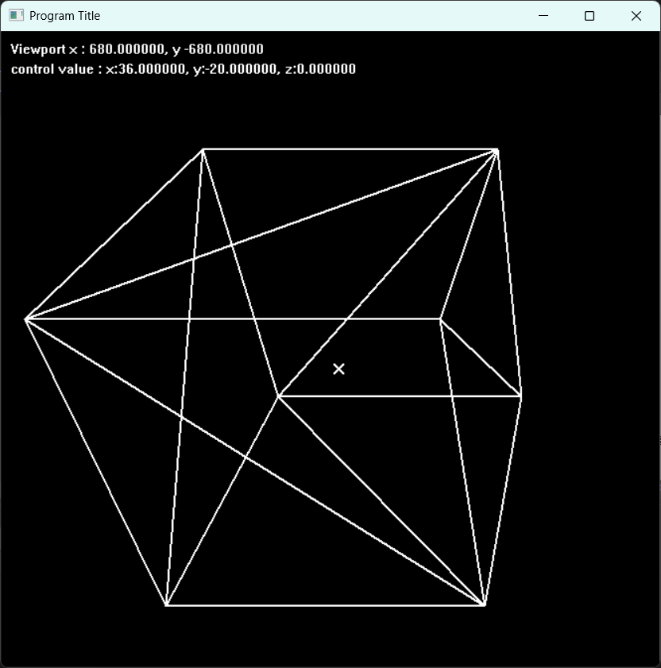
투영나누기가 없고, 뷰포트연산이 잘 안되는게 문제인거 같다.  
근데 투영나누기를 넣으니까 오브젝트가 망가지기 시작해서, 이거 원인을 찾아봐야 할거 같음.



2025년 05월 19일 한달만의 재개

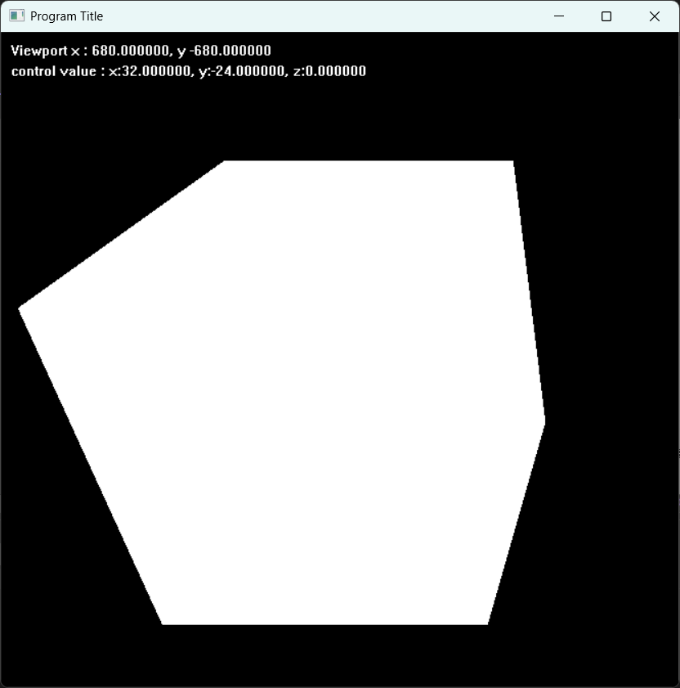
알고보니 투영 나누기 할 때 w/w를 해줘야 하는데 이걸 안해준게 문제 였음.

이걸 해주고 viewport 연산까지 적용해주니까 제대로 나온다.



2025년 05월 20일 레스터 라이징 시작

바운딩박스 -> AABB로 엣지 펑션. -> 그려지는것 판별해서 WC인 삼각형만 dot에 점찍기 진행 중…

기존 DrawLine으로는 제대로 동작이 안되었기에 점찍기 하도록 새로운 구조 변경이 있었음

2025년 05월 21일 노말 시작

레스터라이저는 어제 사소한 코드 문제(휴먼에러)가 있었는데 고쳐지면서 제대로 나옴. 근데 그냥 하얀 실루엣이 통체로 나온지라 모델 분간이 안됐던지라 노말을 추가함.

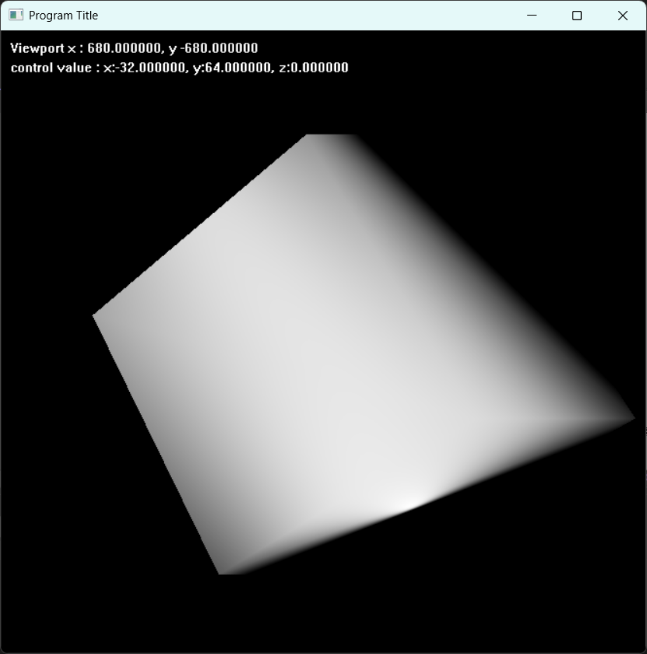
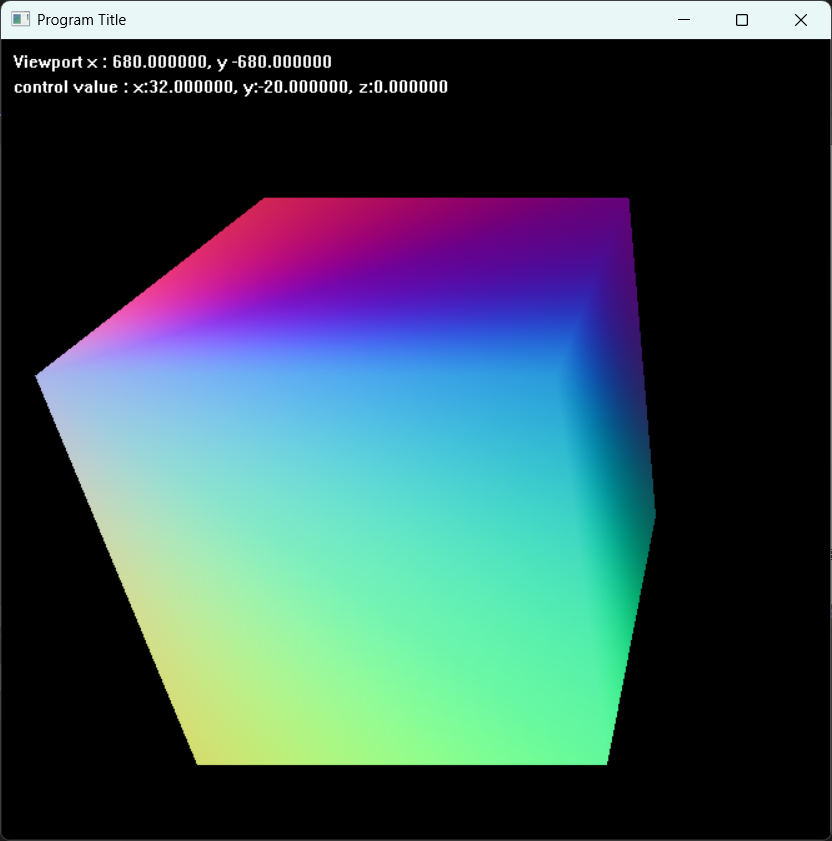
모델+view 연산의 역행의 전치 행렬을 구해서 노말 메트릭스를 만들어 놓음

버텍스 마다 노말메트릭스를 적용해서 정규화 해서 노말 버텍스 연산

RGB 입히는 디버그도 빠르게 해보고 바로

Point L = Point::Normalize(Point(0.0f, 0.0f, 1.0f));

로 간단한 라이트 하나 넣어서 라이트 노말도 해봄.



2025년 05월 22일 텍스처 이전 구조정리

구조를 한번 점검할 필요가 있어서 거기에 힘썼다.

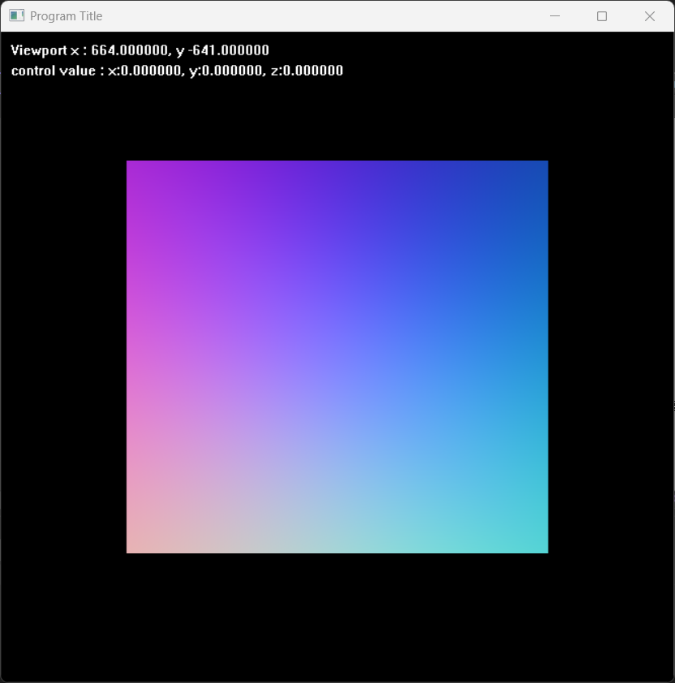
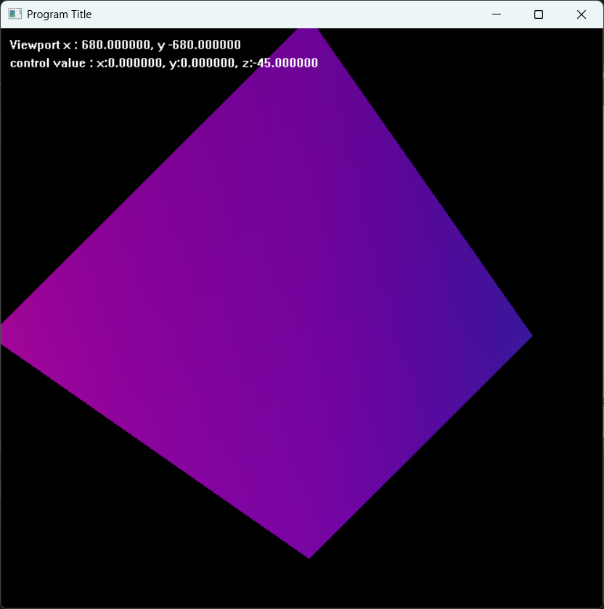
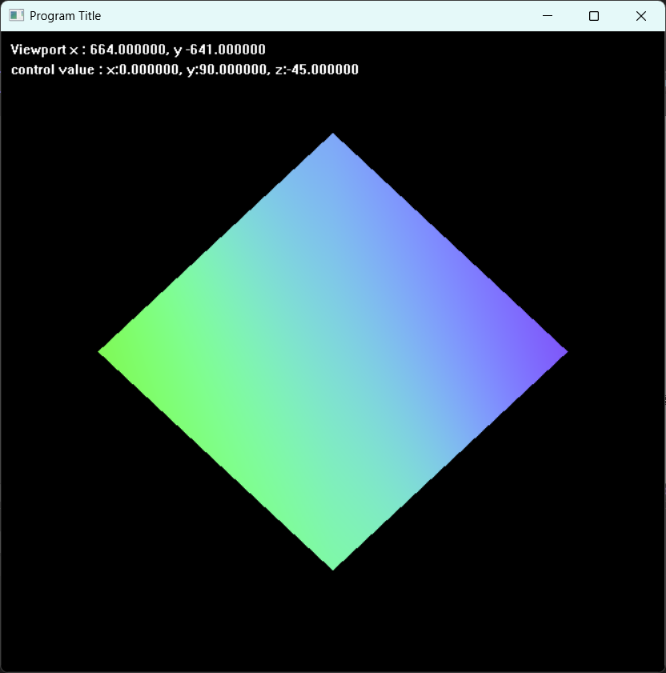
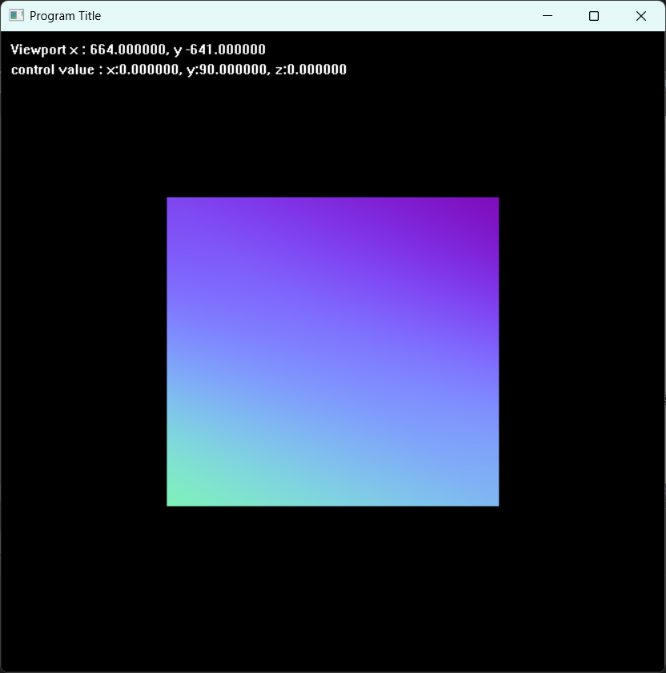
추가로, 그동안 Cube가 회전이 이상하게 걸리는 현상이 있었던걸 뒤늦게 알아차려서 그부분을 수정했다.

result = Transform,  
result = rotation \* result

result = scale \* result

이렇게 넣었는데 이건 T -> R\*T -> S\*R\*T 가 되어서 의도한 TRS가 아니라 SRT 순으로 연산되고 있었다.

\* TRS, SRT, 컬럼(열)벡터, 로우(행)벡터 문제 잘 기억하기

그외 조작키 상하를 y, 좌우를 x에 뒀었는데 이부분이 조작이 햇갈렸지만 그냥 쓰고 있었는데 같이 수정. 게임에서 많이 쓰던 익숙한 방식으로 물체를 좌우로 조작하려면 y축, 물체를 상하로 움직이려면 x 축을 조작해야 한다.

여전히 회전에 문제 발생.

큐브가 0,0,0에 있고, 카메라가 0,0,5에 있는데 정마름모가 아닌 찌그러진 형태가 나왔다.

문제원인을 찾아야 한다.

지금은 축 문제로 추측 중이고, 이 때문에 새로운 오브젝트를 추가해야 할 것 같다.

일단 기존 큐브 자체가 오브젝트를 겸 하고 있었기 때문에 큐브와 오브젝트를 분리하는 구조 수정을 한번 해야 할 듯. 그리고 디버그를 위한 그리드와 좌표 확인을 위한 삼발이? 를 추가하기 위해 라인 레스터라이저를 추가해야 할 것 같다.

할일 정리

1. 부모 클래스 ‘오브젝트’ 추가 밑 큐브가 상속을 받도록 변경
2. 라인레스터라이징 추가
3. 그리드와 좌표축 디버그 오브젝트 추가
4. 회전문제 원인 탐색 및 문제 해결

2025.06.02

일주일만에 돌아온 소프트웨어 렌더러. 지난주는 과제와 자격증 시험 이슈 때문에 패스 했고, 다시 제개.

먼저 구조정리를 해야 할 것 같아서 그거부터 할 예정.

지금 Render 함수가 엔진의 Update() 처럼 동작하고 있는데 그거 먼저 수정할것임.

렌더링 오브젝트로 바꾸고나서 레스터 라이징이 안됨  
그냥 점찍기는 잘 되는거 확인했고 레스터라이징 부분인건 확실히 확인함

근데 여기서 어디냐? 모르겠다. 일단 프레임 버퍼에 안담기는거 보면 그 전 문제임

보니까 색칠하는 부분 프레임버퍼 index가 계속 0으로 잡히는데 y값 설정해 주는 부분이 한참 윗부분. Y가 minY값 받아오기 때문에 그거 설정해 주는 그부분에 문제가 있는 것 같음. 정작 그 부분에서 뭐가 문제인지 모르겠다.

이전 코드를 남겨뒀어야 보면서 비교 하는데…

근데 이게 이전 코드도 내가 다른건 건드린건 없이 렌더오브젝트로 바꾸면서 이름 바뀐 부분만 바꿔줬었는데, 그렇기 때문에 레스터라이징 과정에 저런 문제가 생겼다기 보단, 내가 렌더오브젝트의 설정해줘야 하는 부분을 빼먹었거나 아니면 렌더 오브젝트에서 바뀐 이름 고쳐주는 과정에서 실수 한 것 같다.

하나하나 찾아가는 수밖에…

찾았다. 어이없는 실수 였음.

최종행렬을 각 버텍스에 적용한 finalVertex라는 벡터가 따로 있었는데 얘에서 받아와야 할걸 그냥 이전의 vertecies를 가져와서 레스터라이징 하고 있었음. 그러니 될 턱이 있나…

아무튼 해결.

이제 그리드랑 기즈모 추가하고 라인레스터라이징 할 차례

2025.06.10

다시 일주일 만에 복귀. 사실 그간 짬짬히 했는데 일지 쓰는걸 깜빡

일주일간 여러모로 바빠서 짬짬이 밖에 못했다보니 일지 내용이 별로 없었음.

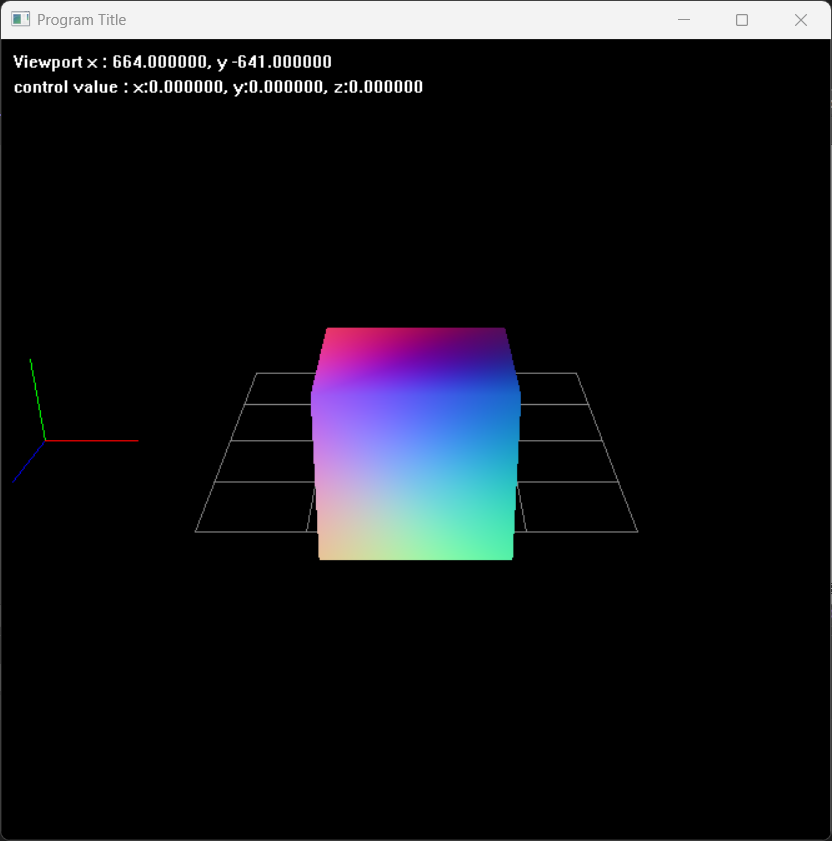
그동안 한걸 요약하자면, RenderObject라는 class를 만들어서 cube를 RenderObject를 상속받는 구조로 바꿨음. 그리고 똑같이 상속받는 Grid와 Axis\_Widget을 만들었다.

겸사겸사 라인 레스터라이징도 했음.

지금 레스터라이징 후 라인레스터라이징을 진행하기 때문에 무조건 라인이 최상위에 그려지는 문제가 있는데, 이부분을 어떻게 하면 블렌더에서 보는 것처럼 자연스럽게 만들 수 있을지 고민중.

추가로 RenderObject에 enum calss type을 추가했고, 지금은 NONE, Face, Line만 있음.

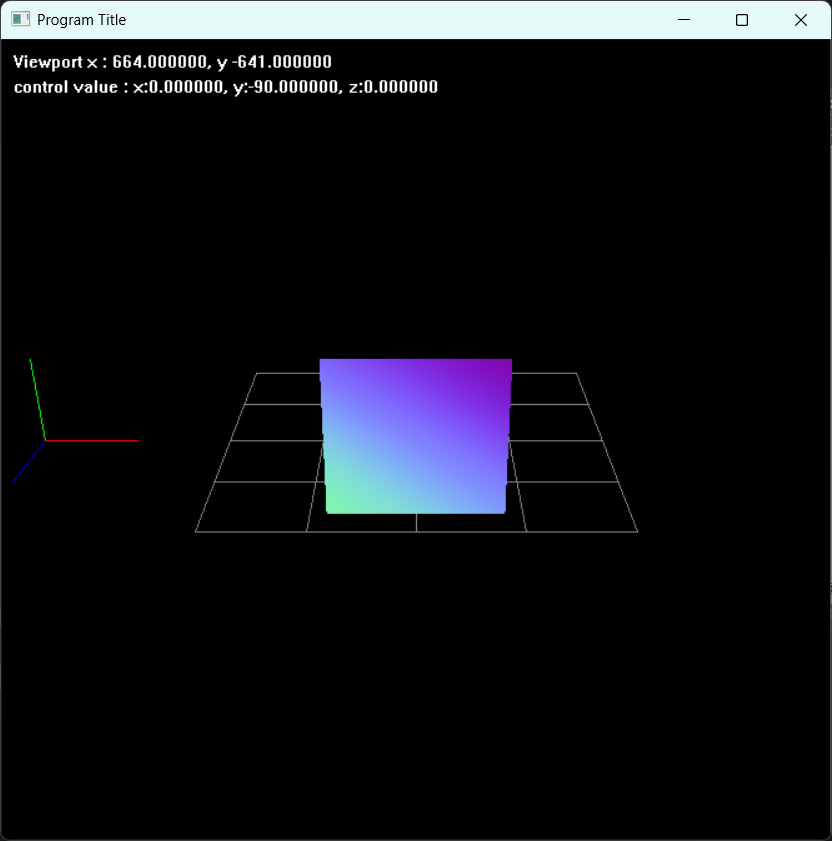
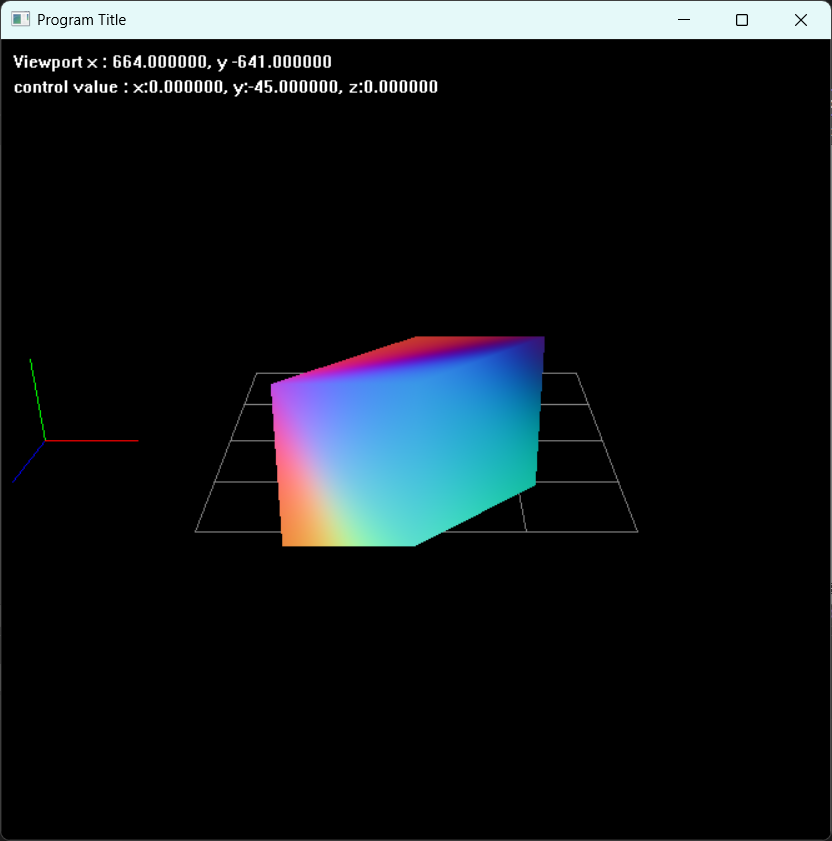
오브젝트가 많아진 만큼 관리 해줘야 할 필요도 있었기 때문에 추가했고, 오브젝트들을 하나의 풀에 등록해서 순회하며 type에 따라 레스터라이징과 line 레스터 라이징을 해주도록.



그리고 지금 추가로 진행중인건 카메라를 RenderObject로 묶는건데 어케 해야 할까 고민중.

다른 오브젝트들 처럼 pos,rot로 움직이고 싶은데 Lookat은 그방식으로 하기엔 여간 귀찮은게 아니다보니까 고민중임. Transform을 적용하면 카메라 행렬이 달라져야 하는건지…

gpt에 따르면 그냥 카메라 월드 행렬의 inverse를 먹이면 된다는데 그건 안되더라고 역시 그렇게 간편하게 할 수는 없나봄.



그리고 카메라 각도와 그리드 넣고 보니까 확실히 이상한게 눈에 보인다. 뭐가 문제인지 한번 찾아봐야 할 듯

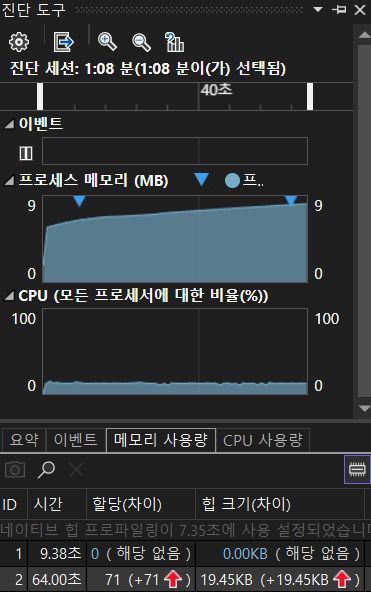
2025.06.11

x축 회전은 멀쩡하다. y축과 z축 회전이 이상한 것 같음. 특정 좌표만 문제가 생길 수 있는가? 이게 좀 애매하다.

생각 해보니 내가 오랫동안 미뤄 왔을 뿐이지 예전 지오메트리 파이프라인 가지고 고민 할 때도 있었던 현상임. 그땐 점찍고 선으로 냅다 이어놓은게 다고 원근도 없어서 햇갈린다고 미뤄놨었다.

지오메트리 부분이 이상한게 맞는 것 같다. 카메라의 타겟을 0,0,0에 두고 pos만 바꿔가며 관찰 했는데 (카메라 기준에서) 오브젝트가 정상적으로 이동, 회전하는 것처럼 보임.

TRS 함수를 전체적으로 점검해야 할 듯.



추가. 메모리 릭 발견

정확하게는 GDI가 일정 시점부터 더 이상 화면을 갱신하지 못하는 문제가 있어서 찾아보다가 발견함.

매번 멈추는 시점을 비교해 보니까 메모리가 천천히 늘어나다가 10mb에 도달하는 시점에서 화면 갱신이 멈추고 더 이상 메모리가 늘어나지 않음.

이 부분 이슈로 달아두고 일단 먼저 회전 문제 먼저 하자.

하나씩.

DirectX 기준 정리  
오일러는 X\*Y\*Z 지오메트리 파이프라인은 S\*R\*T 렌더링파이프라인은W\*V\*P 좌표계는Yup에 Zforward 행벡터(Row Vector)고, 인덱싱은 시계방향(CW)

https://math-development-geometry.tistory.com/51  
이 글에 따르면 난 그동안 이동 행렬을 열벡터 기준으로 만들고 있었다.