제3장 레스터화 같이 공부하겠습니다

3장 레스터화 공부하기 전에 2장 그 정점 처리했던 것을 잠깐 복습해 보겠습니다

자 우리 2장에서 정점처리를 배웠는데요 정점 처리 잠깐 복습하겠습니다

1장에서 게임을 제작하는 단계를 모델링 애니메이션 렌더링 이렇게 간단히 분류했고

그 중에서 우리 책은 렌더링을 포커스 합니다. 렌더링은 렌더링 파이프 라인을 거치는데

첫번째가 정점 처리 두번째가 레스터화 세번째가 프레그먼트 처리 네번째가 출력 병합입니다.

정점 처리는 3D 맥스로 모델링한 오브젝트를 오브젝트에서 정점 삼각형의 정점들로 처리하는

건데 대표적인게 변환입니다. 2장에선 변환을 주로 배웠습니다.

그리고 레스터화 단계에서는 이 삼각형 안에 들어가는 픽셀들을 결정하는 이런 프레그먼트들을

찾아내는게 레스터화 단계이다 3장에서 배울 것입니다.

프레그먼트 처리는 이 프레그먼트의 색깔을 결정합니다. 색깔은 텍스처 매핑이나 조명 이런 것을

거치게 되면 색깔이 결정이 되는 거고요 이게 프래그먼트 처리이고

출력 병합은 삼각형들이 겹쳐 있거나 이럴 때 앞뒤를 따지거나 혹은 반투명이면 색깔을

섞거나 해서 최종적인 컬러 버퍼의 색깔을 결정하는게 출력 병합입니다

다시 한번 렌더링 파이프라인은 정점 처리 레스터화 프레그먼트 처리 출력병합 내단계를 거친다

그 중에서 정점 처리하고 프래그먼트 처리는 프로그래밍 할 수 있는 즉 쉐이더 프로그래밍을 할

수 있는 단계이다

자 2장에서 배울 정점 처리는 변환 조명 애니메이션 등이 있는데 2장에서는 변환을 주로 다룰

것 입니다.

어떤 변환이 있느냐 세 개의 변환이 있습니다. 이것도 외워야 되는 거죠 어떤 변화에 있느냐 첫

번째 월드변환

월드 변환이 뭐냐 3D max로 모델링한 오브젝트들의 오브젝트 좌표계에서 오브젝트들을

클라이언트 프로그래머가 게임월드에 좌표계를 하나 딱 설정하고요 그곳으로 오브젝트들을

가져오는게 월드 변환하는 거죠

그리고 나서 카메라의 움직임에 따라서 이 LCD 화면에 보이는 게 달라지니까 월드 좌표계는

좌표계가 고정입니다. 클라이언트 프로그래머가 딱 하나 만들어 놓은 건데 카메라의 움직임에

따라서 화면에 보이는게 달라져야 하니까 카메라 공간으로 바꿔야 되겠다 카메라 공간으로

바꾸는 변환을 뷰 변환이라고 한다

그리고 나서 3차원 공간을

2차원 LCD 모니터에 투영하기 위한 이 클립 공간 클립 공간으로 바꾸는게 투형 변환이다 그래서 월드 변환 뷰변환 투영 변환 세 가지를 배웠고요

각각을 살펴보면 월드 변환은 클라이언트 프로그래머가 게임월드에 딱 설정한 월드 좌표계로

가져오는게 어떤게 있냐면 SRT가 있습니다.

축소 확대 회전 이동 srt가 있다

이동 변환 같은 경우에는 3차원 벡터일 때는

이렇게 벡터의 덧셈으로 보통 표현을 하고 하는게 보통인데

스케일링 축소 확대나 회전은 3X3 행렬로 표현할 수 있으니까

이동 변환도 행렬로 표현을 하면 나중에 복합 변환을 행렬로 표현할 수

있으니까 이동 변환을 행렬로 표현하기 위해서 이제부터 3차원 카테시안

좌표를 4차원 동차 좌표로 확장을 시켰다 3차원

직교좌표계를 4차원 동차 좌표계로 확대를 시키면 이동 변환이 벡터의 덧셈이 아니라

이렇게 4바이 4 변환 행렬로 여기 변환되는

요소를 여기다 써 주기만 하면 이제는 4바이 4 행렬로 변환을 나타낼 수 있다 오픈 젤은 벡터를 오른쪽에 이렇게 열벡터로

4:37

나타내고 왼쪽에 변화를 쓴다고 했어요 자 이런 식으로 이동

4:42

변환을 행렬로 나타내고 축소 확대 회전도 다 4바이사 행렬로 확장을 하면

4:48

복합 변환을 행렬로 나타낼 수 있다 복합병을 행렬로 나타낼 수 있고요

4:55

동차 좌표는 나중에 3차원 카테션 좌표계로 변환을 시킬 때이 w에 해당하는

5:02

값으로 나눠 주게 되면 카테션 좌표계 즉 직교좌표계로

5:07

변환이 된다 직결좌표계로 변환이 된다 그런 얘기를

5:17

삼각형의 노말은 비균등 축소 확대를 하게 되면

5:23

비균등 축소 확대 여기에서 예로 나온 것은 x 방향으로 1/2 y 방향으로 1만큼 축소

5:31

확대 했더니 비균등 이잖아요 균등하지 않으면 삼각형을 변환하고 나서

5:37

법선 벡터가 수직이 안 되더라 그래서 놈을

5:43

즉 법선 벡터의 변화는 이 변환 행렬 대신에

5:48

행렬의 역행렬에 전치행렬을 적용을 시켜야 된다 그래서

5:54

간단한 게임 엔진 같은 경우에는이 비균등 축소 확대를 지원하지 않는다 이렇게

6:00

역행렬에 전치행렬을 만들어서 적용을 시켜야 되니까 시간이 많이 걸려서

6:06

지원하지 않는 경우도 있다 자 그리고요 뷰변화는 카메라 공간으로

6:14

바꾸는 건데요 카메라 공간을 나타내는 카메라의 외부 파라미터 3개가 있다 그랬어요 카메라의 위치 I 바라보는

6:22

지점 at 업백터이 세 가지에 의해서 카메라

6:27

카메라 공간의 기저인 uvn을 만들어낼 수 있다 uvn 그 중에서

6:33

uvn 중에서 뭘 먼저 만드느냐 i에서 at을 바라보는 반대 방향인

6:40

n을 먼저 만든다 i에서

6:45

엣을 바라보고 있는데 그것의 반대 방향 이게 나중에 z좌표가 될 거예요 i-at 크기를 1로 만들고요 그리고

6:53

업백하고 n을 외적 해서 유 벡터를 만들고요 마지막으로 n 벡터와 유 벡터를 외적 해서

7:00

V 벡터를 만든다 이게 카메라 공간에 기저가 되겠고요

7:06

카메라 변환을 하는 방법은 두 단계를 거친다 카메라의 위치 i를 월드 좌표계의

7:14

원점으로 이동 먼저 이동하고 그리고 나서 카메라 공간의 기전이 uvn을

7:21

월드 좌표계 x y z로 맞추는 회전

7:27

회전 변환 이동하고 회전하는게 뷰변 아니다 이때 이동 변화는

7:34

이동 변화는 카메라의 위치에 반대 방향으로 이동하면 되는 거고요

7:40

회전 변화는 카메라 공간의 기저인 uvn을 차례로 써주기만 하면 이게

7:47

회전 변환 행렬이 되더라 이것을 합하면 이게

7:52

뷰변한 행렬이다 비변을 행렬이다 오픈 지엘은 벡터를 오른쪽에 열벡터로 사용하기

7:59

때문에 먼저 적용하는게 오른쪽에 쓰는 거죠 이동하고

8:04

회전해야 된다 그것의 복합 변환이 뷰변한 뷰변한 행렬이 되더라 이거죠

8:13

정점별 조명 대신에 5장에서는 fragment별 조명을 할거다

8:19

자 그리고 투영 변환을 하기 위해서이 카메라 공간을 카메라의 내부 파라미터 4개로

8:28

시야 절두체를 구성을 합니다 시야 절도죄 영어로는 뷰플러스 텀 시야 절도체는

8:34

머리가 잘린 피라미드 자 머리가 잘린 피라미드를 구성하는

8:39

카메라의 내부 파라미터 4개는 뭐냐 fovy 수직 시야가 수직 시야가 그

8:48

다음에 화면에 종횡비 화면에 종횡비 요즘은 와이드 화면이 유행하죠 h분의

8:55

W 종횡비 두 번째 그 다음에 전방 평면의 거리 n 후방평면의 거리 F 그거에 의해서

9:03

시야 절도죄 머리가 잘린 피라미드가 결정이 된다 이것은 오른손

9:08

좌표계이구요 uvn을 uvn을

9:14

x y z로 나타낸 것이에요 여기가 카메라의 위치 아이고요 카메라 위치 아이가

9:22

s를 바라보는 반대 방향이 n이었잖아요 그래서 오른손 좌표계에

9:29

z의 반대 방향으로 이렇게 씨앗 절두체가 보이게 된다

9:34

자 그리고 이 씨앗 절도죄 바깥에 있는

9:40

물체들 여기 빨간색으로 표시된 물체들이나 걸치는 물체들은 나중에

9:45

잘라내야 된다 이것을 컬링 혹은 클리핑이라고 합니다 자 클리핑을 하는데

9:51

이 시야 절두체에서 클리핑을 하는 것은 쉽지 않아요 그래서

9:57

쉽게 클리핑을 하기 위해서이 시야 절두체를 직육면체로

10:04

2 곱하기 2 곱하기 1 2 곱하기 2 곱하기 1

10:10

직육면체로 변환을 시키는게 투영 변환이다

10:15

그러면 바라봤을 때

10:20

투영선으로 바라봤을 때 겹치는 것을 쉽게 XY 좌표에서

10:26

XY 좌표가 같을 때 Z 값을 가지고 어떤게 카메라의 앞에 있는지 쉽게

10:32

결정을 할 수 있다 그리고 클리핑도 쉽다 그래서이 씨앗절두체 대가리가 잘린

10:40

피라미드를 직육면체로 바꾸는 변환을 투영 변환이라고 한다

10:45

투영 변환이라고 한다 자 그리고 투영 변환을 하게 되면 이렇게

10:53

카메라에서 씨앗 적절체를 바라봤을 때는 카메라의 중심에서 투영선을 사선으로 그어

10:59

줬는데 투영 변화를 하게 되면 이렇게 직육면체로 바꾸게 되면

11:05

투영선이 사선이 아니라 평행하게 돼서 투영선의 벡터가 다 일치하게 되고요

11:12

투영 변화를 한 이후에는 멀리 있는 물체는 작아집니다 가까이 있는 물체는 커지는

11:17

그래서 원근 투영 효과를 가지게 된다

11:24

자 투영 변환 행렬은 지난 시간에 조형변환 행렬을 유도했고요

11:32

이 투형변환 행렬에이 결과는 사실은이

11:38

Z 값이 Z 값이 1이 아닌 w값이 1이 아닌 값이 나오게 되는데이 투형 변환

11:45

행렬을 좀 쉽게 구하기 위해서 그렇게 만든 거고요 또 하나이 렌더링 파이프라인 단계에서

11:52

정점처리 다음에 레스터와인데요 레스터와 단계에서는 레스터화 단계에서는

11:58

왼손 좌표계를 사용해야 됩니다 그래서 투형변환 최종적으로는

12:03

왼손 좌표계로 바꿔야 돼요 이렇게 왼손 좌표계로 바꿔야 된다 왼손 좌표계 이게 왼손 좌표계죠

12:10

왼손 좌표계 그래서이 투형 변환 행렬을 구한 다음에

12:16

z의 값을 - zfo를 반대로 하기 위해서 세 번째 행을

12:22

-를 붙인다 세 번째 행을 -를 붙여서 최종적인

12:27

투영 변환 행렬을 구성을 하면 이것은 왼손 좌표계로 바꿔 주게 되더라이

12:34

얘기입니다 왼손 좌표계로 바꿔주게 되더라 그래서 최종적으로

12:43

정점 처리해서 월드 변환 뷰변환 투형 변환을 거치게 되면 왼손 좌표계의

12:48

직육면체 클릭 공간으로 바뀌게 됩니다 이거가 이제

12:54

지난 시간에 했던 거고요 다이렉트 3d는 2 곱하기 2 곱하기 1 클릭

12:59

공간이 직육면체인데 오픈 gl은 2 곱하기 2 곱하기 2 정육면체였습니다 그래서

13:05

투영 행렬이 조금 달랐어요 지난 시간에 이렇게 배웠고요 유도하는

13:11

것은 지난 시간에 했습니다 자 그러면 오늘은이어서

13:19

레스터와 레스터와 렌더링 파이프 라인의 두 번째 단계인 레스터와 정점 처리

13:26

이후에 rest와 정점 처리는 삼각형 정점에 대표적인 변화를 한

13:32

거예요 월드변환 뷰변환 투영 변환 다

13:37

했다고 치고 이제 레스터화 단계에서는이 삼각형 안에 들어가 있는 들어가는이

13:43

fragment들을 찾아내는 건데이 플래그먼트들은 나중에 픽셀의

13:49

픽셀 값을 결정하는 데이터의 집합이라고 보시면 됩니다 [음악]

13:55

이 픽셀마다 프래그먼트가 생성이 돼야 되는데 이것은 하드와이어드

14:01

단계이구요 렌더링 파이프라인의 두 번째 단계인데 이 레스터화 단계는 또 5단계를

14:08

거칩니다 다섯 단계 이것도 외우셔야 돼요 다섯 단계 첫 번째 클리핑 그 다음에

14:14

원근 나루 샘 뒷면 제거 뷰포트 변환 스캔 변환

14:20

특히이 스캔 변환이 좁은 의미에서 레스터와 단계이고요이 앞에 4개는

14:28

넓은 의미의 레스터화 단계에 포함된다라고 보시면 되겠습니다 정점 처리 이후에

14:35

삼각형 안에 들어가는이 프레그먼트들을 결정하는게 레스터화 단계인데 다섯 단계를 거친다 이것을

14:42

외우셔야 돼요 클리핑 원근 나눗셈 뒷면 제거 뷰포트 변환 스캔 변환 5단계를

14:50

거친다 자 그러면 차례차례 차례차례 보게 보겠습니다

14:56

먼저 클리핑 자 클리핑은 뭐냐 우리가 투영 변환에서이

15:03

머리가 잘린 피라미드를 직육면체로 바꿨어요 자 그런데이 카메라 공간에서

15:09

카메라의 내부 파라미터 4개에 의해서 결정되는이 시야 절두체 바깥에 있는

15:16

삼각형들은 다 잘라야 돼요 이걸 컬링 혹은 클리핑이라고 하는데이 T1 같은

15:21

경우에는 완전히 제거해야 되고요 T2 같은 경우에는이

15:27

씨앗을 수치 안에 있으니까 레스터와 단계로 들어가야 되는 거고요

15:32

세 번째 삼각형은 걸쳤어요 그래서 이걸 잘라내야 돼요 자 이거를이 씨앗

15:39

절도체에서 하는 것은 어려워요 계산이 어렵다 그래서 클리핑을 한 이후에

15:46

투영 변화를 한 이후에이 투영 변환 투영 변환이죠

15:52

투영 변화를 한 이후에 [음악] 클립 공간에서 자르게 되면

15:59

클릭 공간에서 클리핑을 하게 되면 훨씬 쉽다 따라서

16:04

투형 변환 한 이후에 클릭 공간에서 클리핑을 수행한다 이게

16:10

레스터와의 첫 번째 단계 당연한 거죠 클립 공간 안에 있는 것들만 화면상에 보일 거예요 직육면체 안에 있는

16:16

것들만 보인다이 피라미드 대가리가 잘린 피라미드에서 클리핑하는게 아니라

16:22

투영 변환 한 이후에 클릭 공간 직육면체에서 자른다 그게

16:28

클리핑 자 레스터와의 5단계 중에서 첫 번째 클리핑

16:33

투영 변환 한 이후에 직육면체 바깥에 있는 것들은 잘라낸다 클리핑이고요 당연한 거고요 자 두

16:41

번째 두 번째 단계는 원근 나눗셈 자 원근 나눗셈 자 우리가

16:48

투영 변환 행렬을 구해냈는데요이 투형 변환 행렬은

16:53

행렬을 쉽게 만들기 위해서 사실은이 씨앗 절두체 안에 들어있는

17:00

X Y z에다가이 구한 투형 변환 행렬을 적용시키면요

17:07

W 값이 1이 아니에요 왜냐하면 투영 변화행렬을 좀 쉽게

17:13

만들기 위해서 이 변환되는 값이 w가 1이 아니도록

17:20

일부러 만든 거예요 그러면 이것을 직교좌표계 3차원 카테션 좌표계로

17:27

만들기 위해서는 투영 변환된 이후에이 w값으로 즉

17:32

w값이 4행 1년 해보면 -z가 나오도록 만든 건데

17:38

w값 즉 - z로 나눠줘야 돼요 -z로 나눠줘야 된다

17:46

그래야지만 w값이 1로 바뀌고요 세 가지 좌표가 3차원 직교좌표계가 되는

17:52

것이다 자 투영 변환 한 이후에 w값이 1이

17:58

아니니까 w값으로 나눠준다 당연한 거잖아요 동차 좌표계에서

18:04

변환한 이후에 W 값이 1이 아니면 w로 나눠주는게 3차 카테션 좌표계로

18:10

변환하는 거잖아요 그리고 투영 변환에서는 W 값이 1이

18:15

아니도록 투형 변환 행렬을 쉽게 만들기 위해서 그렇게 만들었다고 했잖아요 그 W

18:20

값은 - z예요 적용을 시켜보면요네 번째 행위

18:27

00 -1 0이니까 x y z 1에다가 적용을 시키면

18:32

W 값이 - z가 나옵니다 따라서 w값 - z로 나눠줘야 된다

18:39

나눠줘야 된다 나눗셈이 해야 되는데 왜 원근 나눗셈이라고 부르느냐 그 다음을

18:46

보면 알 수 있어요 왜 원근남은 샘이라고 하느냐 자

18:52

투영 변환을 할 때 투영 변환한 이후에 w값이 - Z

18:58

값이 의미하는게 뭐냐면요 카메라 공간에서 [음악] XY

19:04

평면으로부터의 거리의 양수 값입니다

19:09

무슨 말이냐면 -z라는게 뭐냐면 여기요 그림을 한번 보세요요 길이

19:16

요 그림은 오른손 좌표계에서 아직 투영 변환하지 않을 때 오른손 좌표계에서

19:25

이쪽이 z축 위쪽이 y축

19:31

x축은 화면 안쪽으로 들어가는 거예요 엄지손가락이 x축

19:36

검지 손가락이 y축 세 번째 손가락이 z축 오른손 좌표계에서 z의 반대

19:42

방향으로 피라미드 대가리가 잘린 피라미드를 바라본 거예요 자

19:48

그렇다면이 씨앗 절두체 안에 들어있는 P1 P2 q1 q2의

19:55

요 요 값 Z 성분 예를 들어서 p2는

20:01

Z 성분이 -2거든요 -2라는 의미는

20:07

Z 값이 -2니까 그 Z 값의 부호를 반대로 해주면

20:12

원점에서의 거리가 되는 것입니다 원점에서의 거리 XY 평면에서의

20:18

거리가 되는 것이죠 원점이 아니라 참 XY 평면으로부터의 거리가 된다

20:27

이 피라미드 대가리가 잘린 피라미드 씨앗 절두차 안에 있는 물체

20:32

Z 성분의 -를 붙이면 XY 평면에서의 거리 거리가 된다

20:39

즉 거리로 나누게 되면 멀리 있는 물체는 작아지고 가까이 있는 물체는 커지는 원근 값이

20:46

생긴다이 얘기에요 그래서 원근 나눗셈이라고 하는 겁니다 자 다시 보세요

20:53

투영 변환 행렬을 우리가 2장에서 구했습니다 요것은

20:58

외울 순 없어도 나중에 시험 문제를 낼 때 제가 여러분들 한

21:03

페이지에 정리해서 시험 보라고 할 거거든요 그럼 요걸 딱 정리하셔 가지고요 예를 들어서 이런 투형 변환

21:09

행렬이 있는데 수직 시야각이 2분의 파이 수직

21:15

시작하게 2분의 파이다 그러면 이거는 파이널 라디아닐테니까

21:22

파이 라디안은 각도로 나타내면 180도죠 2분의

21:27

파이는 90도에요 자 그리고 종횡비는 1

21:32

화면의 종행비가 1이라는 얘기는 직사각형이 아니라 정사각형이라는 얘기의 화면이 그리고 가까운 평면의

21:39

거리가 1 먼 평면의 거리가 2 지금이 시야 절두체를 결정하는

21:46

카메라의 내부 파라미터 4개가 지금 수직 시야각이 90도 가까운 평면의

21:54

거리가 1 먼 평면의 거리가 2 화면에 종횡비가 1

21:59

이랬을 때이 투영 변환 행렬이 가장 간단하게 나타내집니다 자 보세요

22:07

코탄젠트 2분의 fovy 자 수직 시작하기 2분의 파이

22:13

90도라고 했어요 90도의 절반은 45도죠 탄젠트 45도는

22:21

탄젠트 45도 이렇게 직각삼각형이 있을 때 얘가 45도면

22:27

45도면 1 1 루트 2잖아요 탄젠트 45도는 1이에요

22:34

따라서 코탄젠트 45도도 1이죠 에스펙트는 1

22:39

따라서요 부분이 1이에요 코탄젠트 45도 1 그 다음에 f-n

22:46

분의 - F 2-1 -2 즉

22:52

-2죠 2 - 1분의 -2 곱하기 1

22:57

-1이죠 여기는 무조건 상수 마이너스 1 이게

23:04

4가지 카메라의 내부 파라미터일 때 가장 간단한 투영 변환 행렬이다

23:10

자 그러면 이 투영 변환 행렬을 이용해서 직육면체로 변환시켜 보는 거예요 그게

23:17

투영 변환해져 투형변환 행렬은 레스터화 단계를 위해서

23:22

왼손 좌표를 바뀐다 그랬어요 왼손 좌표계 여기가 z축 y축

23:28

x축은 하면 안쪽으로 들어가요 자 왼손 좌표계로 바뀐다 자 그러면

23:35

p2에 대해서 투영 변화를 해볼게요 P2 0

23:41

-2를 투영 변환 해보면

23:46

0 1 2가 나옵니다 w값이 1이 아니에요 w값이

23:54

- z가 나온다고 했죠 W 값이요 Z 성분의 부호가 반대인 거

24:00

- Z 즉 2가 나오는 거예요 그 w로 나눠준다 즉

24:06

XY 평면에서의 거리로 나눠준다 그래서 키가 작아진다 이거예요

24:13

p prime p2의 프라임은 키가 작아진다 멀리 있는 물체는 키가 작아진다 w를

24:20

1로 바꿨더니 w로 나눠줬더니 키가 작아졌다 멀리 있는 물체는

24:26

멀리 있으면 멀리 있을수록 나누는 분모가 커지니까 더 작아지겠죠

24:34

그래서 원근 나노쌤이라고 하는거다 투영 변환 한 이후에

24:40

직육면체 상태에서 직육면체 바깥에 있는 것을 잘라내는 것은 클리핑 그리고

24:46

투영 변화를 했더니 w값이 1이 아니니까 W 값으로 나눠주는데 그 w가

24:53

성분이 -z다 - z의 의미는 XY 평면에서의 거리니까

24:59

거리로 나눠주니까 거리가 크면 클수록 분모가 커지므로 거리가 크면 클수록

25:06

멀리 떨어져 카메라에서 멀리 떨어져 있으록 작게 된다 가까이 있을수록

25:12

커지게 된다 원근감을 나타내는 것이니까 원근 나눗셈이라고 한다

25:19

자 q2에 대해서도 적용을 해보면 Q2 q2는 여기 있잖아 Q2

25:24

0 -2 0 -1을 적용했더니

25:30

0 2가 되더라 w가 1이 아니더라

25:36

w로 나눠주게 돼서 0이 되더라 0 1이 되더라 자 이런

25:43

피라미드를 투형변환하면 직육면체가 된다 이렇게 요런 직육면체가 된다요 직육면체

25:50

바깥에 있는 물체들을 다 클리핑하고 w로 나눠줬더니 가까이 있는 물체는

25:57

커지고 멀리 있는 물체는 작아지는 원근 나눗셈이 된다 자 그리고요

26:04

이렇게 원근 나눗셈 한 이후에 나온 3차원

26:11

좌표를 정규화된 좌표 ndc라고 합니다

26:16

원근 나눗셈을 한 좌표를 노멀라이즈드 디바이스 코디네이트

26:22

정규화된 좌표 ndc라고 합니다 ndc라고 한다

26:29

자 레스터화 단계의 5가지 중에서 첫 번째이 직육면체를 만든 후에 클릭

26:35

공간으로 투영 변환한 이후에 클리핑 직육면체 바깥에 있는 것을 잘라내는 거 클리핑 두 번째

26:44

투영 변환 한 이후에 w값이 1이 아니니까 w값으로 나눠준다 나눗셈인데

26:51

W 값이 - Z 즉 XY 평면에서의 거리니까

26:57

거리로 나눠준다 멀리 있는 물체는 작아진다 가까이

27:02

있는 물체는 커진다 원근 나눗셈을 하는거다 이렇게

27:09

자 이렇게 두 가지가 됐고요 자 세 번째 뒷면 제거

27:15

레스터와 단계 사는 거 뒷면 제거를 또 해야 됩니다 자 뒷면 제거는 무엇이냐

27:23

레스터화 단계는 투영 변환 한 이후에요 클릭 공간으로 가져온 다음에 하는

27:29

건데 일단은 개념상 이렇게

27:34

폴리곤 메쉬가 이렇게 9가 있다고 가정을 해보고요 카메라가 이렇게 있을 때

27:42

카메라에서 바라보는 카메라 벡터가 C1 C2 c3가 있으면요

27:49

아직 변환하기 전이라고 가정했을 때

27:55

각각의이 폴리곤 메쉬에서요 어떤 삼각형은 나중에 화면에 보일

28:01

거고요 어떤 삼각형은 화면에 안 보여요 어떤 거냐면 예를 들어서

28:07

T2 삼각형 같은 경우에는요 화면에 카메라에서 바라볼 때 보이잖아요

28:14

삼각형이 그러니까 얘는 나중에 화면에 보일 거고요 그래서 T2 삼각형 같은 경우에는요

28:21

이 삼각형의 법선 벡터가 카메라의 시선 벡터와

28:27

둔각을 이뤄요 바라보는 방향이 서로 대충 반대 방향이에요 둔각을 이루면 나중에 보인다

28:36

즉 얘는 앞면이다 앞면이다 앞면 카메라를 향한다 카메라를 향하는

28:44

삼각형은 나중에 화면에 보일거다 앞면을 어떻게 판정하느냐

28:49

이 삼각형의 법선 벡터와 카메라의 시선 벡터가 왜 둔각을 이룬다 즉

28:56

둔각을 이룬다는 얘기는 내적을 하면 음수란 얘기에요 내적의 유클리드 기와

29:02

같은 경우에는 벡터 n과 c를 외적 하면 벡터n의 크기 곱하기

29:09

벡터 c의 크기 곱하기 코사인 세타잖아요 자 코사인 그래프는 이렇게 생겼어요

29:17

그래서 둔각일 경우에 90도보다 크면 음수란 말이에요

29:24

즉 내적을 했을 때 음수이면 둔각이다 둔각이란 얘기는 앞면이란 얘기에요

29:30

그리고 만약에 삼각형의 법선 벡터 T3 같은 경우에

29:35

법선 벡터와 시선 벡터가 90도를 이루면 선만 보여요 선

29:42

삼각형이 면으로 안 보이고 선으로 보인다 내적을 하면

29:48

0이 된다 왜 코사인 90도는 0이니까

29:53

마지막 T1 같은 경우에 삼각형의 법선 벡터와 카메라의 시선

30:00

벡터가 바라보는 방향이 거의 일치해요 즉 예각이에요 예각

30:06

그러면 내적을 하면 양수 내적을 하면 양수 그러면

30:12

뒷면이고요 얘는 어차피 요쪽 부분만 보이기 때문에 뒷면은 안 보인다

30:19

어차피 안 보이는 삼각형 뒷면들은 렌더링 파이프라인에서 뒤에 할 일이

30:25

많거든요 따라서 얘네들은 다 걸러야 돼 뒷면을 제거해야 돼요 영어로는

30:31

백페이스 컬링 렌더링 파이프라인에서 레스터와 단계에서 다 없애야 된다

30:38

계산을 줄이기 위해서 하는 방법은 이렇게 내적을

30:43

통해서

31:06

이렇게 판정할 수도 있는데요 이것은요 이런 식으로 하는 것은 비용이 많이 들어요

31:13

그래서 투영 변환 한 이후에 뒷면 제거하는 것이 훨씬 효과적입니다

31:18

왜냐하면 지금 카메라 공간에서 카메라 시선

31:24

벡터는요이 카메라의 위치에서 방사선 모양으로 투영선을 그어야 되니까 카메라 벡터 시메라 벡터가 다

31:32

달라요 근데 투형 변화를 하고 나서는이 씨앗 절도체가 왼손 좌표계

31:39

클립 공간으로 바뀌면요 왼손 좌표계 leftend 시스템

31:44

왼손 좌표계로 바뀐다고 했어요 직육면체 2 곱하기 2 곱하기 1로

31:50

바뀌어야 이것은 왼손 좌표계에요 왼손으로 표현해 보세요

31:56

그러면 투영선이 다 평행해진다 카메라

32:03

시선 벡터가 다 일치한다 어떻게 되느냐 XY 성분은 없고

32:08

Z 방향으로만 성분이 있다 즉 카메라 시선 벡터가 001이 돼서

32:15

이 투영 변화한 이후에 클릭 공간에서 뒷면을 제거하는게

32:22

훨씬 쉽다 자 그리고 뒷면을 제거할 때

32:28

내적을 쓰지 않고도 뒷면을 제거하는 방법이 있습니다 어떻게 하느냐

32:34

이 투영 변환한 이후에 투영 변환한 이후에 클릭 공간에서

32:40

삼각형을 바라봤을 때 예를 들어서 T1 같은 경우에

32:47

여기 카메라에서 바라보는 방향이요 방향이면 카메라에서 바라보는 방향이요

32:52

방향일 때 만약에 이게

33:00

시계방향으로 정렬이 되어 있다면 v1 V2

33:05

v3 시계방향으로 정렬이 되어 있다면 뒷면입니다 왜냐하면 우리가 오른손

33:13

좌표계에서는 반시계 방향으로 정렬을 한다고 했어요 반시계 방향으로 정렬을 한다고 했는데

33:19

반식의 방향으로 정렬이 되어 있다면 이제 정확하게

33:25

앞면인 거예요 예를 들어서 T2 같은 경우에 v1

33:30

V2 v3가 이렇게 바라봤을 때

33:37

바라봤을 때 이렇게 반시계 방향이니까 이것은 앞면인 거예요

33:43

근데 T1 같은 경우에는 시계방향으로 정점이 정렬이 되더라

33:50

그러면 뒷면이라 이겁니다 뒷면 자 그리고

33:55

시계방향으로 정렬되어 있는지 어떻게 하느냐 v1에서 v2를 가는 벡터

34:02

v1에서 v3로 가는 벡터의 행렬식을 구해보면

34:07

행렬식이 음수이면 행렬식이 음수이면

34:13

시계방향 뒷면입니다 반대로 행렬식이

34:18

행렬식이이 T2 같은 경우에 v1에서 v2로 가는 벡터 v1에서 v3로 가는 벡터의 행렬식이

34:26

양수이면 양수이면 반시계 방향이고 앞면이다 이렇게 생각하는 겁니다 자

34:34

오른손 좌표계에서는 모델의 바깥쪽으로 법선 벡터가

34:40

튀어나오게 하기 위해서는 정점에 정렬 순서가 반시계 방향이라고 했어요 1장에서 반시계

34:48

방향이고요 그래서 이걸 직교 투영해서 투영 변환한 이후에 2차원에서

34:53

바라봤을 때 반시계 방향으로 보이면 앞면인데 시계방향으로 보이면

34:59

뒷면이더라 시계방향인지 어떻게 아느냐 반시계방향인지 어떻게 하느냐

35:06

v1 v2에 있는 벡터와 v1 v3를 있는 벡터의 행렬식을 나타냈을 때

35:12

행렬식이 양수면 반시계 방향 음수면 시계방향이더라 그래서요

35:19

뒷면은 제거한다 이 얘기입니다 이렇게

35:25

레스터와이 5단계 중에서 클립 공간으로 투영 변환 이후에

35:32

직육면체 바깥에 있는 것은 클리핑 그리고 투영 변환한 이후에 w로 나눠주는

35:38

원근 나눗셈 세 번째 뒷면은 제거한다 뒷면 제거

35:45

자 그리고요 뒷면 제거할 때 뒷면 제거할 때

35:51

우리가 오픈 주엘이나 다이렉트 3D 같은 경우에는요 오픈 지역 오른손

35:57

좌표계는 반시계 방향으로 정렬되어 있고 다이렉트 3d는

36:03

왼손 좌표계 시계방향으로 정렬되어 있어야지 앞면이라고 생각했는데요 그래서

36:09

뒷면 제거할 때 그렇게만 고정이 되어 있으면이

36:15

좌표계를 오른손 좌표계에서 왼손 좌표계 왼손 좌표계에서 오른손 좌표계로 바꾸게 되면이 정렬의 순서를

36:22

또 바꿔 줘야 되는데 그렇게 할 필요가 없다라고 얘기했어요 왜냐하면이 프로그래밍을 할 때 오픈 젤이나

36:29

마찬가지로 다이렉트 3이나 둘 다 선별을 해서 뒷면을 제거할 수 있습니다

36:36

어떻게 GL cal face라는 함수가 있는데

36:41

glase는 glase는

36:49

기본이 뒷면을 제거하는 건데요 뒷면을 제거하는데

36:55

앞면을 제거할 수도 있고요 뒷면을 제거할 수도 있고요 앞면 뒷면을 다 제거할 수도 있습니다

37:01

파라미터로 게다가 앞면을 앞면을 설정할 때

37:07

오른손 좌표계는이 ccw 즉 반시계 방향 정렬일 때

37:14

앞면인데 이것을 또 파라미터에서요 CW 즉 시계방향을

37:20

앞면으로 설정할 수도 있습니다 그러니까 프로그램에서 얼마든지 앞면 뒷면에이

37:27

시계방향이냐 반시계 방향을 바꿀 수가 있다 그리고

37:32

앞면을 제거할 거냐 뒷면을 제거할 거냐도 프로그램에서 설정할 수 있다

37:37

오픈젤뿐만 아니라 다이렉트3도 마찬가지여서 유연하게 프로그래밍 할 수 있다

37:44

따라서 좌표계를 변환할 때 시계방향에서 반시계 방향 꼭 굳이

37:49

변환할 필요는 없습니다 자 그런데

37:56

좌표계의 변환을 다시 한번 생각해 보면 우리가 1장에서는 오른손잡이에서

38:03

왼손 좌표계로 변환할 때 그냥 변환하면 좌우가 바뀌니까 Z 부호를 밖으로

38:08

바꿔졌었는데요 그거뿐만 아니라 그것뿐만 아니라 이 3D MAX 좌표계 오픈 듀얼

38:16

좌표계 다이렉트 시리즈 좌표계 각각에서 변환할 때 어떻게 돼야 되는지를 한번 살펴보겠습니다

38:22

자 우리 김영 교수 김영 교수님 수업시간에 3D 모델링 3D MAX

38:28

좌표계는요 오른손 좌표계 오른손으로 표현하죠 그런데 z축이 하늘 방향이에요

38:36

xy가 바닥 바닥을 나타내고 있습니다 z축이 하늘 방향입니다 오른손

38:43

좌표계이고요 오픈 듀얼 좌표계도 오른손 좌표계인데

38:49

오른손 좌표계인데 하늘 방향이 y축이에요 하늘 방에 y축이다

38:56

다이렉트 3d는 왼손 좌표계 왼손 좌표계고 마찬가지로 하늘 방향이 y축이다

39:04

자 그러면 김현 교수님 수업시간에 3D 맥스로

39:10

모델링한 오른손 좌표계의 모델을 다이렉트 3d에서 사용하려고 하면

39:16

어떠한 변환을 거쳐야 되는지 한번 살펴보겠습니다

39:21

오른손 좌표계의 다이렉트 3d에서 왼손 좌표계의

39:27

잘못 말했어요 오른손 좌표계의 3D MAX 모델에서

39:33

다이렉트 3D 왼손 좌표계로 바꾸려면 어떻게 하느냐

39:38

첫 번째 첫 번째 우선은

39:44

이 3D max에서 같은 오른손 좌표계인 오픈 젤로 변환을 해보겠습니다

39:51

3D max는 z축이 하늘 방향인데 이것은 그대로

39:56

오른손 좌표계로 바꾸면 y축이 하늘 방향 즉 z가 누여 있는

40:02

방향으로만 바꾼다고 하면 이걸 그대로 다이렉트 3D

40:09

max의 값을 오픈 gl로 그냥 그대로 내보내면요 그냥 그걸 그대로 갖고

40:15

오면 모델이 바닥에 누워 있어요 모델이 바깥에 바닥에 누워있다 그래서

40:21

모델을 세우려면요 어떻게 해야 하냐면

40:27

모델을 세우려면 즉 3D MAX z축이 하늘 방향에서 오픈 젤 y축이

40:35

한 알 방향으로 바꿀 때이 오브젝트들이 그대로 꼿꼿이서 있으 있게 하려면서 있게 하려면

40:45

yz 축을 플립 해야 됩니다 y z축을 플립 해야 된다

40:51

yz축을 대칭 이동한다 x 축을 중심으로

40:58

yz 평면을 90도 회전해야 된다 이어 있는 것을

41:03

x축을 중심으로 yz 평면을

41:09

90도 회전해야 된다 그래야지만 꼿꼿이 서게 된다

41:15

자 다시 한번 다시 한번 보세요 3D 맥스에서 오픈 gl로 변환할 때 그냥

41:22

그대로 변환하면 모델이 누워요 그래서

41:28

꼿꼿이 세우려면 어떻게 해야 되느냐 오픈 gl로 변환할 때는 이렇게

41:35

yz 플립을 해야 된다 즉 3D MAX 좌표계 z축이 하늘

41:41

방향인 오른손 좌표계에서 y축이 하늘 방향인 오픈 젤로 바꾸면서 모델이 꼿꼿이서 있게 하려면

41:50

yz 플립을 해야 된다 즉

41:55

삼각형의 정점 제로제로 20 미 오픈

42:01

지혈 좌표로 바뀌었을 때는 0 20 0이 됐고요 그 다음에 3D max에서 0

42:08

180이라는 정점은 오픈 젤로 바꿨을 때는

42:14

0 0 -18이 돼야 된다 이렇게 바꾸려면 어떻게 되냐면요 3D

42:19

max의 xyz에서 yz를 순서를 바꾸고요 세 번째

42:26

-를 붙여준다 이렇게 하면 이렇게 하면 3D max에서 오픈 GL

42:33

똑같은 오른손 좌표계인데 y축이 하늘 방향이고요 오브젝트가

42:38

눕지 않게 되더라이 얘기입니다 yz 플립 yz축 플립

42:45

y하고 z의 값을 부호로 바꾸고 -를 붙인다 요게 yz 플립입니다 그러면 오픈

42:52

지열 좌표계로 내보내지게 된 거고요 자 그리고 나서

43:00

그리고 나서 다이렉트 3d로 바꿔보겠습니다 이 얘기는 일단 오픈

43:07

gl로 바뀐 다음에 다이렉트 3d로 바꾸는 것으로 축약을 해보면 오른손

43:12

좌표계에서 왼손 좌표계로 변환하는 건데 1장에서 이미 했어요

43:18

오른손 좌표계에서 왼손 좌표계로 변환을 하게 되면 그냥

43:24

그대로 바꾸게 되면 지금 이 아까 봤던 건데요 이게 오른손

43:29

좌표계에요 카메라는 z의 z축의 반대 방향으로 바라보고 있어요

43:35

카메라 그러면 왼쪽이 왼쪽이 3연체 오른쪽이 주전자예요

43:41

이것을 그대로 왼손 좌표계로 바꾸게 되면 즉 다이렉트

43:47

3D 왼손 좌표계로 바꾸게 되면 z축이요 방향으로 바꾸게 되고요 카메라는

43:53

z의 반대 방향으로 바라보고 있는데 그렇게 되면 이게 다이렉트 3D 왼손

43:59

좌표계에요 그렇게 되면 카메라가

44:06

z에서 z의 음의 방향으로 바라보게 되면

44:11

왼쪽이 주전자가 된다 왼쪽이 주전자가 된다 화면이 좌우가

44:16

바뀌게 된다 영상이 1장에서 얘기했어요 그래서

44:21

이걸 해결하려면 어떻게 했었냐 오브젝트에 z의 부호를

44:29

zf를 바꾼다 z의 부호를 바꾸게 되면이

44:35

오브젝트들은 z의 음의 방향이 아니라 z의 양야 방향으로 옮겨지게 되고 카메라도

44:42

z의 양의 방향으로 바라보게 되면 이때는 왼쪽이 사면체가 보이게 되어서

44:50

화면이 좌우 대칭이 되는 문제를 해결할 수 있다이 얘기입니다

44:56

다 정리를 하면 자 3D max에서 다이렉트 3d로

45:02

변환하는 과정을 살펴보면 첫 번째 3D MAX 좌표계에요 오른손 좌표계

45:10

z가 하늘 방향 그때의 인덱스가 있는

45:17

트라이앵글리스트 정점 버퍼와 인덱스 버퍼 인덱스 트라이앵글 리스트 방법으로 가장

45:25

많이이 모델 정보들을 저장한다고 했어요 정점 버프에는 정점들을 한

45:33

번씩만 저장을 한다 그리고 인덱스를 부여한다 0 1 2 3 정점에 저장

45:38

순서는 상관없구요 삼각형마다 인덱스의 순서를 반시계 방향으로

45:45

0 3 1 뭐

45:51

0 0 1 2 이런 식으로 저장한다 이때 0부터 시작하지 않고 1부터 시작하면 120 이런 식으로 해도

45:58

되는 거예요 자 이런 식으로 반시계 방향으로 ccw로

46:04

저장을 했다 자 이것을 [음악] 똑같은

46:10

오른손 좌표계인데 y축이 하늘 방향인 오픈 젤로 바뀌었다 오픈 줄 오른손

46:16

좌표계 오른손 좌표계로 바꾸려면 yz축 플립 즉

46:23

yz 값을 뒤바꾸고 yz 값을 뒤 바꾸고 마지막에

46:30

-를 붙여준다 -를 붙여준다 그러면 모델이 눕지 않고 꼿꼿이서 있다

46:38

여전히 ccw 인덱스에 저장 순서고요

46:43

자 마지막으로 오른손 좌표계 오픈 젤에서

46:48

그냥 왼손 좌표계 z축 이쪽 방향이잖아요 왼손 좌표계로

46:54

왼손 좌표계로 바꾸면 왼손 좌표계로 바꾸면 카메라가 반대

47:01

방향 즉 z의 반대 방향으로 방해 보면 왼쪽이 주전자가 되는 화면 대칭이 생기니까 이걸 해결하기

47:09

위해서 z의 부호를 반대 방향으로

47:14

z의 부호를 반대로 해줬다 z의 부호를 반대로 해주면

47:20

카메라도 이제 Z 방향 양의 방향으로 바라보면 왼쪽이 사면체가 돼서

47:25

화면이 좌우대칭이 해결이 되더라 그러나 여전히

47:32

인덱스에 저장 순서는 반시계 방향 ccw이다 왼손 좌표계로 바뀌었는데도

47:37

ccw인 거죠 그러면 이것을 그대로

47:44

다시 한번 정리해 보면 3D max에서 오픈 지혜를 거쳐서

47:49

다이렉트 3d로 변환할 때 3D max에서 똑같은 오른손 좌표계 오픈 젤로 바꿀

47:57

때는 yz축 플립 yz 값을 바꾸고 -

48:02

붙이고 그리고 다이렉트 3D 왼손 좌표계로 바꾸려면

48:10

화면에 좌우대칭이 안 되게 하려면 z의 반대방 z의 부호를 반대로

48:17

그러면 결국은 3D max에서 다이렉트 3d로 바꿀

48:22

때는 yz 값만 플립을 한 yz 값만 수압을 하면

48:29

되겠다이 얘기입니다 yz 값만 수압을 하면 된다 자 그런데 아까도 얘기했듯이

48:36

여전히 인덱스에 저장 순서는 ccw예요 그래서

48:41

이것을 그대로 렌더링하면 뒷면 제거해서 프로그램을 할 때

48:47

조작을 잘 하지 않으면 뒷면이 제거되는게 아니라

48:52

앞면이 제거되는 불상사가 생길 수 있어요 왜냐하면 다이렉트 3D 왼손 좌표계에서는

48:59

시계방향 클락 yg로 저장된게 앞면으로 생각하는데 ccw로 저장을 해 놨잖아요

49:06

그래서 [음악] 앞면이 제거되는 불상사가 생길 수 있는데

49:12

다이렉트 3d나 오픈gl은 컬링 할 때 앞면을 뭐 cw로 제거할 거냐

49:20

ccw를 제거할 거냐 이렇게 프로그래밍 할 수 있으니까 굳이

49:27

삼각형 인덱스의 정렬 순서를 ccw에서 cw로 변환하지는 않아도 된다

49:35

변하지는 않아도 된다 자 레스토랑 5단계 중에서

49:42

투영 변화는 한 이후에 클립 공간으로 바꾼 다음에 클리핑 원근 나눗셈

49:47

뒷면 제거이 3단계를 했구요 이제 여기까지 하고 그 다음 강의 시간에네

49:55

번째 뷰포트 변환 5번째 스캔 변환 그리고

50:01

피킹 이런 것들을 배워보도록 하겠습니다 그러면 수고하셨습니다