**석진호 졸업 작품 진행 일지**

**2018-01-18**

**공부한 내용**

나는 3D 게임에 필요한 기초 필수 수학 중 행렬 대수를 공부하였다.

**요약**

**3차원 컴퓨터 그래픽에서 행렬** : 비례, 회전, 이동을 간결하게 서술

점 벡터의 한 기준계를 다른 기준계로 변환 할때도 사용.

**정의**

m x n 행렬 M 은 m개의 행, n개의 열 실수들의 정사각 배열

행렬의 차원 : m x n (개수)

**원소(성분) :** 행렬을 구성하는 수들

**표기법 :** Mij ( i 는 행, j 는 열)

**행렬의 덧셈과 곱셈이 만족하는 법칙**

1. A + B = B + A 덧셈에 대한 교환법칙
2. (A + B) + C = A + (B + C) 덧셈에 대한 결합법칙
3. r ( A + B ) = rA + rB 스칼라 분배법칙 (r 은 스칼라)
4. (r+s)A = rA + sA (r ,s 는 스칼라)

**행렬 곱셈**

A = m x n , B = n x p 일때, AB 정의 . AB = m x p 행렬. 이를 C 라고 할 때,

Cij = A i , \* • B \*, j

AB ≠ B

**결합 법칙**

A ( B + C ) = AB + BC , ( A + B ) C = AC + BC

* (AB)C = A(BC)
* 행렬들의 곱하는 순서를 적절히 선택가능

**전치 행렬**

: 행렬의 행들과 열들을 바꾼 것

예 ) m x n 의 전치 행렬은 n x m 행렬

**표기법** : Mt

**전치 행렬의 유용한 성질**

1. ( A + B )t = At + Bt
2. ( c A )t = c At
3. (AB)t = BtAt
4. (At)t = A
5. ( A-1)t = (A-1)t

**단위 행렬**

주대각 성분들만 1이고 나머지는 모두 0인 정방행렬.

* 단위 행렬은 곱셈의 항등원 역할. A = m x n , B = n x p , I = n x m
* AI = A 이고 IB = B
* 어떤 행렬에 단위 행렬을 곱해도 그 행렬은 변하지 않는다.
* 정방행렬 M , I 와의 곱셈은 교환 법칙 만족 MI = IM = M.

**행렬식**

**표기법 :** A의 행렬식 det A

* 3차원 입체의 부피와 관련.
* 선형 변환 하에서 그 부피가 변하는 방식에 대한 정보 제공.
* 1차 연립방정식을 푸는데도 쓰임.

**3D게임에서 행렬식이 중요한 이유**

: 행렬의 역을 구할 때 행렬식이 쓰이기 때문

* 정방행렬 A 는 만일 det A ≠ 0 이면, 그리고 오직 그럴때에만 가역행렬(역행렬이 존재하는 행렬) 이다.

**소행렬**

m x n 행렬 A , 소행렬 는 A의 i번째 행과 j번째 열은 삭제해서 나온 ( n – 1 ) x ( n – 1) 행렬

**행렬식의 정의**

행렬의 행렬식은 재귀적으로 정의

예 ) 4x4 행렬의 행렬식은 3x3행렬의 행렬식들로 정의되고, 3x3행렬의 행렬식은 2x2 행렬의 행렬식들로, 2x2 행렬의 행렬은 1x1 행렬의 행렬식들로 정의된다.

1x1 행렬 A = [A11] 의 행렬식은 det[A] = A11 이다.

* 3차원 그래픽에서는 주로 4x4 행렬을 다룬다.

**딸림 행렬**

A 가 n x n 행렬이라고 할 때, 곱 Cij = ( -1 )i+jdet 여인수라고 부른다.

A의 각 성분의 Cij 를 계산해서 해당 ij 번째 위치에 배치한 행렬 CA 를 행렬 A의 여인수 행렬 이라고 부른다.

CA 의 전치행렬을 A의 딸림 행렬이라고 부른다.

**표기법** : A\* = CAT

* 이 딸림 행렬을 이용하면 행렬의 역을 계산하는 명시적 공식을 구할 수 있다.

**역행렬**

행렬 대수는 곱셈의 역원을 정의.

곱셈의 역을 행렬의 역 or “역행렬”

**주요 정보**

1. 역행렬은 정방행렬에만 있다.
2. n x n 행렬 M의 역은 n x n 행렬, M-1 로 표기.
3. 모든 정방행렬에 역행렬이 있는 것은 아니다. 역행렬이 있는 행렬. 가역행렬

없는 것은 특이행렬.

1. 역행렬이 존재하는 경우, 그 역행렬은 고유 하다.
2. 행렬에 그 역행렬을 곱하면 단위행렬이 나온다. MM-1 = M-1M = I
3. 행렬의 역은 행렬 방정식을 만족하는 행렬들을 구할 때 유용하다.  
   Ex) p’ = pM , p’M-1 = pMM-1 , p’M-1 = p

* 작은행렬 (4x4 행렬 및 그 이하) 딸림 행렬을 이용한 계산법이 효율적.
* 그런데 3차원 그래픽에서 주로 다루는 행렬들은 그 역행렬 공식을 미리 알 수 이쓴 특별한 형태( 3장에 나온다.) 이므로 일반적인 행렬의 역행렬을 구하는 공식을 사용해서 CPU 주기를 낭비할 필요가 없다.
* (AB)-1 = B-1A-1

**DirectXMath의 행렬**

3차원 그래픽에서 점과 벡터를 변환할 때에는 1x4 행 벡터와 4x4 행렬을 사용

**행렬 형식들**

DirectXMath 4x4 행렬을 표현할 때 XMMATRIX 타입 사용.

* XMMATRIX 는 SIMD 활용을 위해 XMVECTOR 인스턴스 네개 사용.
* XMMatrixSet 함수
* 클래스의 멤버변수로 XMFLOAT 4x4형식 사용.

XMFLOAT4x4의 자료를 XMMATRIX에 적재할 때

* XMLoadFloat4x4(const XMFLOAT4x4\* pSource);

XMMATRIX의 자료를 XMFLOAT4x4에 저장할 때

* XMStoreFloat4x4(XMFLOAT4x4\* pDestination, FXMMATRIX M);

**행렬 함수**

DirectXMath의 유용한 행렬 함수들

**XMMatrixIdentity();** : 단위행렬 I를 돌려준다.

**bool XMMatrixIsIdentity(FXMMATRIX M);** : M이 단위행렬인지의 여부를 돌려준다.

**XMMATRIX XMMatrixMultiply(FXMMATRIX A, CXMMAATRIX B) :** 행렬 곱 AB를 돌려준다.

**XMMATRIX XMMatrixTranspose(FXMMATRIX M) :** MT 를 돌려준다.

**XMVECTOR XMMatrixDeterminant(FXMMATRIX M) :** (det M, det M, det M, det M) 을 돌려준다.

**XMMATRIX XMMatrixInverse(XMVECTOR\* pDeterminant, FXMMATRIX M);**

**:** M-1 을 돌려준다.

**규칙 ( XMMATRIX 매개변수를 선언할 때)**

XMMATRIX 하나가 XMVECTOR 매개변수 네 개에 해당.

FXMMATRIX 첫번째 매개변수. 나머지는 CXMMATRIX.

**팀과의 회의**

FBX SDK 를 이용한 애니메이션 부분을 공부 하기로 결정.