실습1

경제학과 2022100382 이찬

1.

Code:

12-19 line에서 XMVECTOR타입을 cout을 통해 처리하기 위해 input 오퍼레이터를 오버로딩하는 함수가 작성되어 있습니다. 이때, 매개변수로는 FXMVECTOR를 사용하였는데, 이는 매개변수로 사용할 경우 임의의 순차로 각각 F, G, H, C 등의 이니셜을 붙여 사용하는 컨벤션 때문입니다.

내부적으로, XMVECTOR를 XMFLOAT3 형식으로 변환시켜준 뒤 예쁜 문자열 형식으로 각 요소를 out stream에 뱉어줍니다.

22line에서 boolalpha 지정을 통해 출력 형식을 타입에 맞게 (정수는 정수로, 3자리 소수는 3자리까지 등)출력되도록 합니다.

26-30line에서는 DirectX 라이브러리의 유효성을 검증하기 위하여, XMVerifyCPUSupport로 검사하고, 내부적으로는 SSE를 지원하는지 검사합니다. 이후 XMVectorSet을 통해 4차원 float 벡터를 4개 선언합니다.

이후 벡터 덧셈, 뺄셈, 곱셈의 XMVECTOR에 대한 오버로딩을 통해 이를 계산하고 결과를 저장합니다.

XMVector3Length를 통하여 벡터의 크기, XMVector3Normalize를 통해 해당 방향으로의 단위벡터를 구합니다. 해당 메소드는 내부적으로 여러 벡터 구현체들에 대한 연산을 가지고 전처리를 통해 올바른 연산을 수행합니다.

XMVector3Dot, XMVector3Cross를 통해 각각 내적과 외적 연산을 수행합니다. 내적 연산의 경우, 모든 요소가 결과값인 벡터를 반환합니다. 이 또한 위와 같이 여러 구현체에 대한 연산을 지원합니다.

이후 XMVector3ComponentsFromNormal(XMVECTOR\* pParallel, XMVECTOR\* pPerpendicular, FXMVECTOR V, FXMVECTOR Normal)를 통하여 매개변수 벡터 V에 대하여 pParallel이 가리키는 벡터에는 Normal에 V를 투영한 벡터를, pPerpendicular이 가리키는 벡터에는 V – &pParallel의 값을 할당합니다. 이를 통해 투영된 벡터와 그 법선벡터를 얻을 수 있습니다.

이렇게 얻어진 projW와 perpW는 논리적으로 더한 값이 원래의 벡터(매개변수 상 V)와 같아야 합니다. 이를 XMVector3Equal, XMVector3NotEqual을 통해 비교해주고, 그 결과를 저장합니다.

또한 XMVector3AngleBetweenVectors를 통해 두 벡터의 각 요소 사이의 각도를 요소로 하는 벡터를 결과값으로 받고, 대표로 X요소의 값을 Degree로 바꾸어 저장합니다.

이후 결과를 예쁘게 출력하는 것으로 끝입니다.

2:

Code:

29line까지는 1번과 같습니다.

XMVectorZero를 통해 영벡터를, XMVectorSplatOne를 통해 모든 요소가 1.0f인 벡터를, XMVectorReplicate 통해 모든 요소의 값이 같은 벡터를, XMVectorSplatZ를 통해 매개변수로 받은 벡터의 Z요소값을 Replicate한 벡터를 선언 및 할당합니다.

XMVectorSplatZ와 유사하게, XMVectorSplatX, XMVectorSplatY, XMVectorSplatW도 존재합니다.

모두 해당하는 요소로 Replicate하는 기능을 가집니다.

이후 각 벡터를 오버로딩한 함수를 통해 출력합니다.

3:

Code:

마찬가지로, 29line까지는 같습니다.

5개의 벡터를 선언 및 할당합니다. 마지막인 w는 0, , *,* 를 각각 요소로 합니다.

이후 여러 연산의 결과를 출력합니다.

XMVectorAbs: 영벡터에서 매개변수로 받은 벡터를 빼서 반대 방향의 벡터를 만들고, 매개변수 벡터와 반대 벡터의 각 요소의 최댓값만을 취하는 벡터를 반환합니다.

XMVectorCos: 각 요소를 [-, ] 구간으로 매핑한 후 각 요소의 Cos 값을 요소로 하는 벡터를 반환합니다.

XMVectorLog: 내부적으로 XMVectorLog2를 사용하며, 해당 함수는 밑을 2로 가지는 로그로 각 요소를 계산한 결과를 요소로 하는 벡터를 반환합니다. Log10, LogE 등 밑을 다른 값으로 하는 함수도 존재합니다.

XMVectorExp: 로그 연산과 같이, 밑을 2로 가지는 함수를 내부적으로 사용하며, 10, E를 root로 가지는 함수가 존재합니다.

XMVectorPow: 앞 벡터의 요소를 a, 뒤 벡터의 요소를 b에 저장 후 pow(a, b)를 실행하여 그 결과를 요소로 하는 벡터를 반환합니다.

XMVectorSqrt: 각 요소를 제곱근한 값을 요소로 하는 벡터를 반환합니다.

XMVectorSwizzle: 각 요소를 재배열한 벡터를 반환합니다. 매개변수 E0 ~ E4는 4 미만의 자연수이어야 하며, 각 인덱스의 요소가 어디로 재배치될 것인가를 결정합니다. 내부적으로 순서대로 처리되므로, swizzle할 위치가 중복되는 경우, 나중에 입력되는 매개변수가 적용됩니다. (46line 참고)

XMVectorMultiply: 각 요소의 곱을 요소로 하는 벡터를 반환합니다.

XMVectorSaturate: 각 요소를 0~1사이의 값으로 재배치합니다. 0 미만 값은 0으로, 1 초과 값은 1로 만듭니다.

XMVectorMin: 두 벡터의 각 요소를 비교하고 더 작은 값을 요소로 취하는 벡터를 반환합니다.

XMVectorMax: 두 벡터의 각 요소를 비교하고 더 큰 값을 요소로 취하는 벡터를 반환합니다.

이후 예쁘게 각 결과를 출력합니다.

3:

Code:

11line에서, 이번에는 출력의 형식을 소수점 8자리까지로 설정합니다.

이후 13~18line은 같습니다.

20, 21line에서 벡터와 그 벡터를 단위벡터로 만든 벡터를 선언 및 할당합니다.

23line에서 해당 단위벡터의 크기를 구하고. 이를 할당합니다.

이후 그 크기를 출력하게 되는데, 0.99999994, 1이 아닌 값이 출력됩니다.