GeometryCore

FmeshShapeGenerator

메쉬를 만들기위한 기본 클래스이다. 상속을 받으면 무조건 generate() 를 구현해야한다.

내부에 가지고 있는 변수들은 public 으로 되어있다. 좀 충격인데 성능을 위해서 그런거같다.

TArray<FVector3d> Vertices;

TArray<FVector2f> UVs;

TArray<int> UVParentVertex;

TArray<FVector3f> Normals;

TArray<int> NormalParentVertex;

TArray<FIndex3i> Triangles;

TArray<FIndex3i> TriangleUVs;

TArray<FIndex3i> TriangleNormals;

TArray<int> TrianglePolygonIDs;

FBXMesh 에서 보자

지금 찾아야하는 내용. Vertext 에 값을 넣는 방법 + 그걸 구체적으로 스테틱 메쉬에 어떻게 넣는지. 스태틱 메쉬에 넣었으면 이제 그걸 렌더링을 어떤 걸 실행시켜서 가져오는지. Vertex 의 내용은 무엇인지. 해당 버텍스의 내용을 수정할 수 있는지.

여러가지 배치 조건을 찾아봐야한다.

벽의 점은

브릿지 패턴은 그냥 다른 객체의 인터페이스를 컨트롤 하는 구체클래스로 컨트롤 하는것이다.

FDynamicMeshBuilder

해당 클래스는 사용성에 집중한 것으로 FPrimitiveDrawInterface 라는 다이나믹 요소에서 사용하기 위한것이다. 성능을 요하는 부분에서는 사용을 주의해야한다.

FPrimitiveDrawInterface

해당 클래스는 동적요소에 대한 쿼리를 위한 인터페이스 클래스이다. 내부를 보면 동적으로 무언가를 만들기 위한 모든게 있는데, 성능이슈가 발생할 수 있다. 한번 주의해서 코드를 보자

FMeshBuildVertexData

해당부분은 버텍스를 개별로 취급하지 않고, color 배열,position 배열,tangentX,Y,Z 배열,UV 배열의 배열이 있따.

탄젠트란 무엇인가? 탄젠트란 삼각함수에서 높이/밑변의 길이이다. 이를 기울기라고도 부르며 sinA/cosA 와도 같기 때문에 무조건 0~1 사이의 값이나오며, A 가 0일 때 높이는 0 이기 때문에 0 이고 반대로 90일땐 cosA 가 0 이기 때문에 무한대가 나온다. 1/0 인 무한대이기때문이다. 물론 극한의 개념이 존재할때이다. 하튼 극한도 사용하는데 뭐가 문제일까. 그래서 특수각에서 탄젠트의 무한대가 존재하는거다.   
하여튼 탄젠트는 기울기 혹은 원에서 표면의 접선이다. 표면의 접선의 기울기가 탄젠트라는 것이고, 이를 다르게 표현하면 탄젠트라는건 표면의 기울기가 된다. 어떤 한 공간에서 탄젠트가 있다면 기울기인것이다. 그럼 왜 노말로 표현을 하지 않고 탄젠트라고 표현을 했을까? 사실 노말만 가져도 해당 표면을 ‘추측’ 할 수는 있다. 하지만 정확히 기울기를 구하기는 쉽지 않으며 탄젠트 Z 나 노말벡터나 같은 의미가 되지만 탄젠트 표면이라고 한다면 정확한 한 표면을 표현할 수 있기때문이기도 하면서 그냥 같은의미인데 노말쓰기 싫었나보지. 근데 사실 점이기 때문에 표면이라기 보다는 방향이 맞겠네. 해당 점의 표면 기울기가 이렇다 이거니깐. 점은 원이 아니기 때문에 뭐 이렇게 표현한거같다.

FMeshBuildVertexView

해당 클래스는 버텍스 데이터의 상세 기입 정보이다. View 라는것에서 알수있듯이 해당 뷰로 특정 데이터를 봤을 때 어떻게 처리할것이고, 해당 데이터를 어떻게 표현할지이다. 이말은 데이터는 그냥 정보덩어리고 그걸 A 로 해석할 것인가 B 로해석할 것인가인데, 사실 그런거 없고 그냥 자료형이나 아니면 크기를 표현하는거임. 별건없음.