

## Lighting(Lightmap을 중심으로)

### 개요

게임에서 라이팅은 Real-time 라이팅과 미리 렌더링된 소스를 이용하는 방식이 있다.

미리 렌더링된 소스를 이용하는 건 Vertex Light와 Light map을 이용할 수 있을 것이다.

그밖에 다른 여러 기술이 있을 수 있으나 현재 라이팅을 테스트 하면서 내가 아는 부분은 여기 까지다.

그 중 아래 라이팅을 테스트 한 부분은 Light map의 소스를 이용한 방식으로 정리하였다.

### 목적

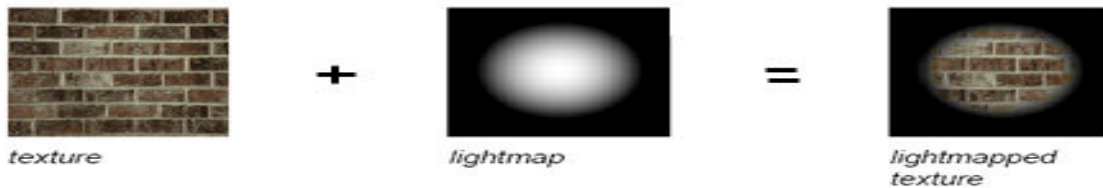
빠른 작업환경과 게임에서 작은 리소스를 이용하여 리얼리티한 라이팅 효과구현이 목적이다.

먼저 빠른 작업환경 구현을 위해서 맥스에서 기본렌더링을 사용하지 않고 Vray란 서드파티 렌더링을 사용하였다. (맥스에서 기본GI의 렌더링이 Vray보다 속도가 느리다. 그리고 제가 Vray를 더 잘 아는 관계로..) 두번째로 적은 리소스를 위해서 Light map의 소스를 작게 만들고자 노력하였다.

### Lightmap이란?

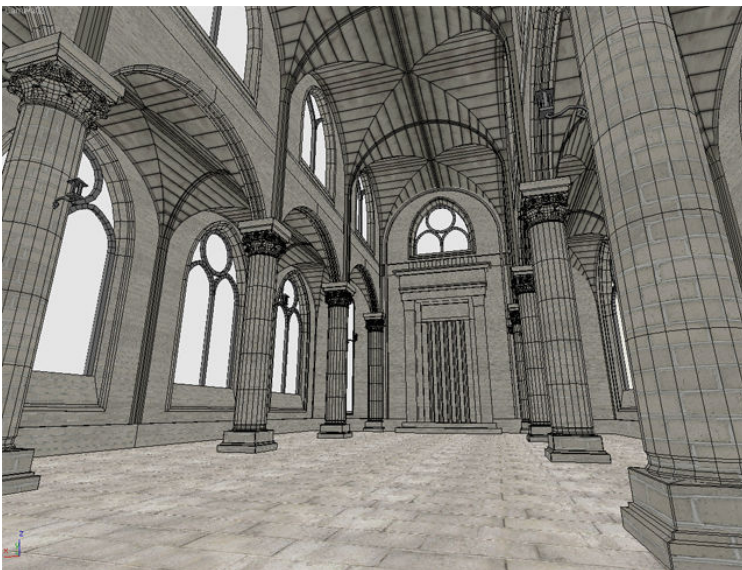
Lightmap이란 말 그대로 라이트의 맵을 만들어 라이트 대신에 적용하는 방식이다.

기본 맵핑된(대부분 타일링되어 있을 것이다.) 오브젝트에 또다른 맵소스(라이트맵)을 겹쳐서 셰이딩 표현이 된 것 처럼 표현하는것이다. (이해 안되면 아래 그림 참조)



### 1. 모델링, 맵핑

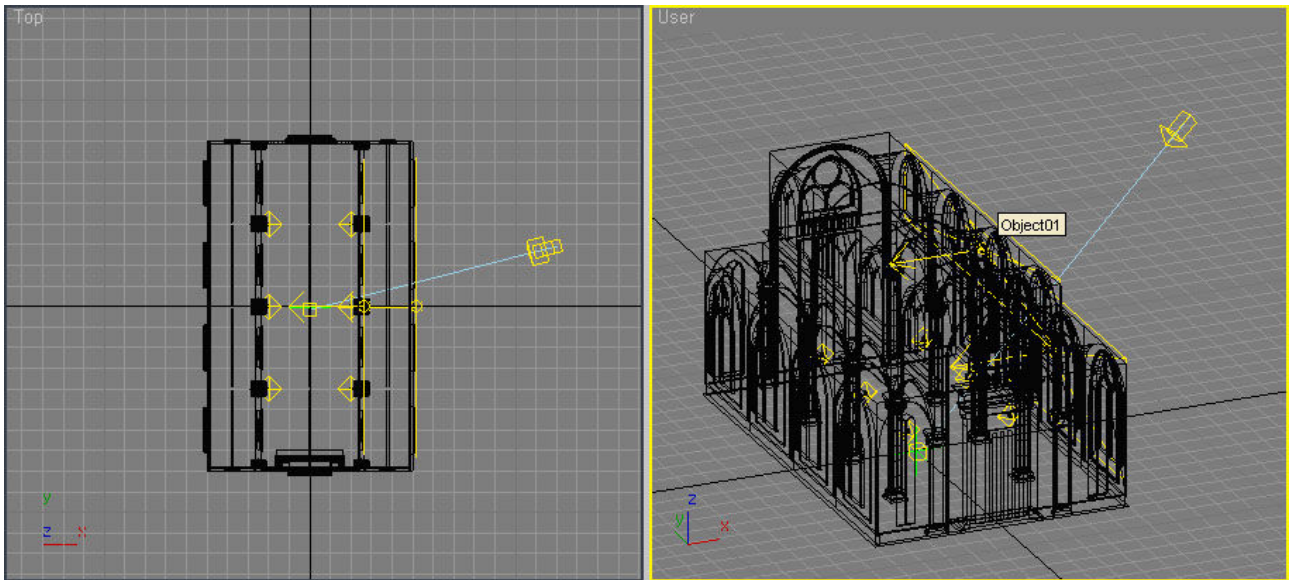
아래와 같이 만들었다. (모델링은 지난번 예칭테스트를 위해 윤석씨와 함께 만들어진 모델링을 이용하였다.)



## 2. Lighting

맥스의 기본렌더링을 사용하지 않고 Vray를 사용하였다.

라이트는 주광원을 2개를 (Vray에서 제공하는 GI용 라이트)를 배치하였고 그림자를 위한 Direct라이트 하나 그리고 부분조명을 위한 Omini 6개를 사용하였다.



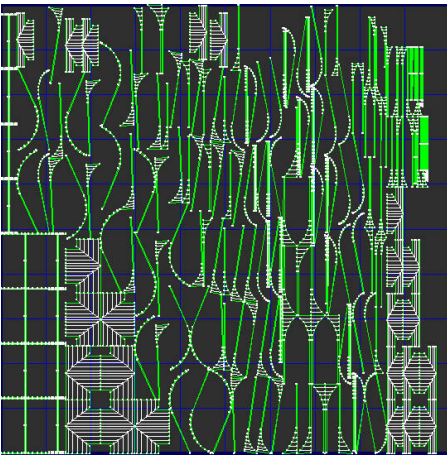
## 3. Rendering



#### 4. Lightmap을 위한 Unwrap

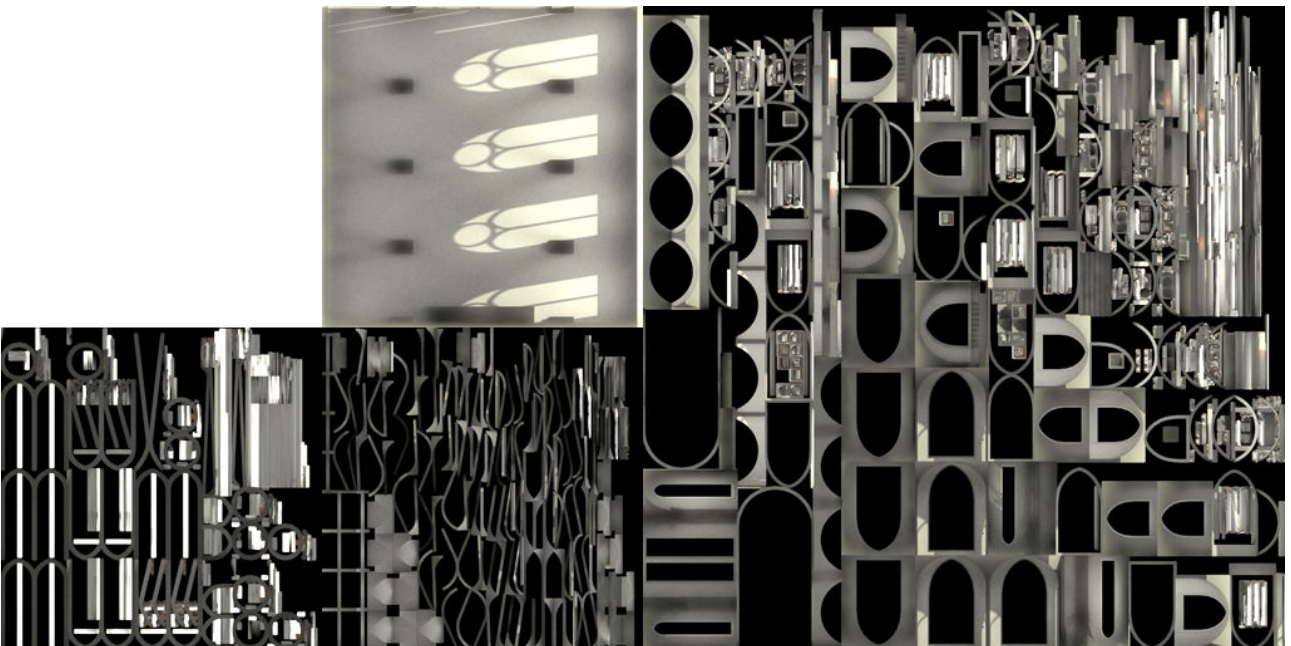
이제 Vray를 이용한 렌더링된 이미지를 라이트맵으로 만드는 작업이 필요하다. 이는 맥스에서 제공하는 Render To Texture의 기능으로 Bake 시키면 되지만, 그전에 먼저 라이트맵을 위한 Unwrap작업이 선행되어야 한다.

Render To Texture에는 Bake를 위한 Unwrap을 자동으로 만들어 주는데 이는 아래의 그림과 같다. 그림을 보면 Automatic Unwrap의 옵션을 아무리 잘 만져도 맵이미지의 낭비가 많으니, 다시 수작업으로 빈공간을 없애준다. (사실 이 부분이 귀찮아서 자동으로 생성된 Unwrap를 이용했지만 실무에선 꼭 수정할 부분이다.)



#### 5. Render To Texture를 통한 Lightmap 소스 제작

원래 맥스의 기본렌더링은 Render To Texture를 통해서 라이트맵을 만들어 주지만, Vray를 이용한 Render To Texture작업은 라이트맵을 만들수 없고 텍스처까지 들어간 이미지로 만드는 버그(버그인지 아직 미지원인지 모르지만)가 있어서 텍스처를 썬 상태로 렌더링을 걸어 라이트맵과 같은 이미지를 만들수 있었다. (512\*512 1장, 256\*256 3장)



## 6. 최종

라이트맵은 오픈지엘에서 안되고 다이렉트기반에서 확인이 가능하기 때문에 맥스의 뷰포터를 다이렉트기반으로 바꾼후, 라이트맵을 오브젝트에 적용하여 확인한다.

아래의 이미지와 같이 맥스에서 렌더링된 이미지와 비슷한 퀄리티로 맥스의 뷰포터에서 확인가능하고 렌더웨어엔진에서도 아래와 같은 퀄리티로 확인 가능하다. (렌더웨어엔진에서 확인은 못해 봤음 )



### 문제점 및 보완사항

1. 적은 리소스를 위해서 공간의 분할이 필요하며, 라이트맵을 위한 맵좌표도 수작업이 많이 필요하다.
2. Render To Texture에서 라이트맵생성단계에서 부분적으로 이미지가 깨지는 경우가 있다.. 모가 문제지 아직 찾지 못했지만 계속 테스트 할 부분이다.
3. Vray자체가 Render To Texture와 잘 연동되지 않는다.