

CHAPTER 8

라이트의 원리: 라이트 모빌리티와 라이트맵

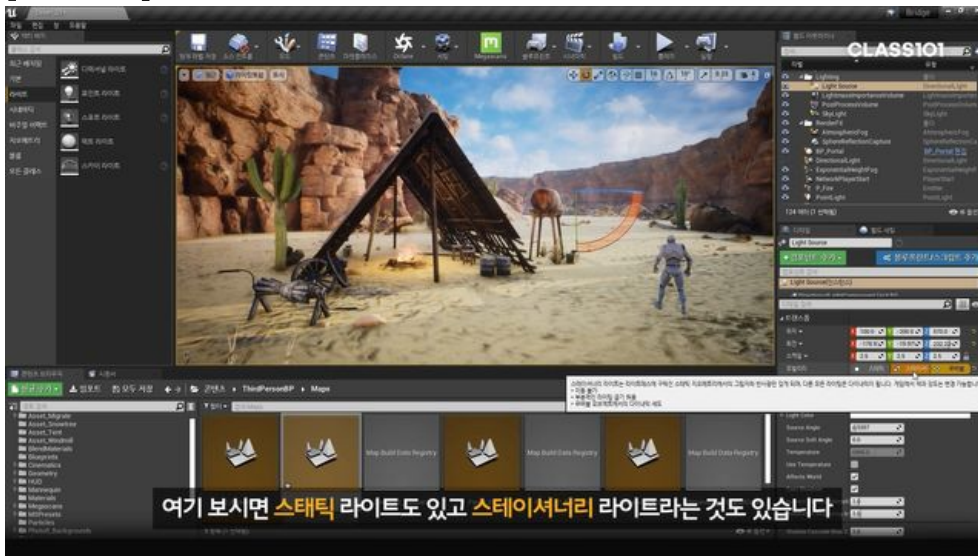
읽어보면 좋은 문서

[수업 목표]

여러분 반갑습니다. 러셀입니다.

이번 시간엔 그 동안 항상 무버블로만 설정해왔던 라이트의 모빌리티에 관해 이야기하는 시간을 가져보겠습니다.

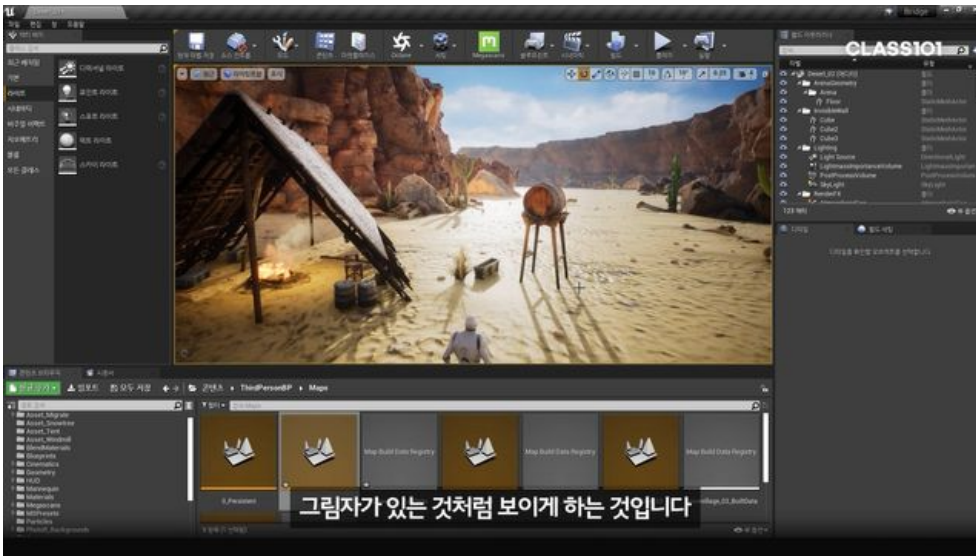
[수업 개요]



언리얼 라이트의 종류엔 크게 세 가지가 있습니다.

스태틱, 스테이셔널, 그리고 무버블 입니다.

각각 연산되는 방식이 다르고 활용하는 방법도 다르기에 어느 정도는 꼭 알아두시면 좋습니다.

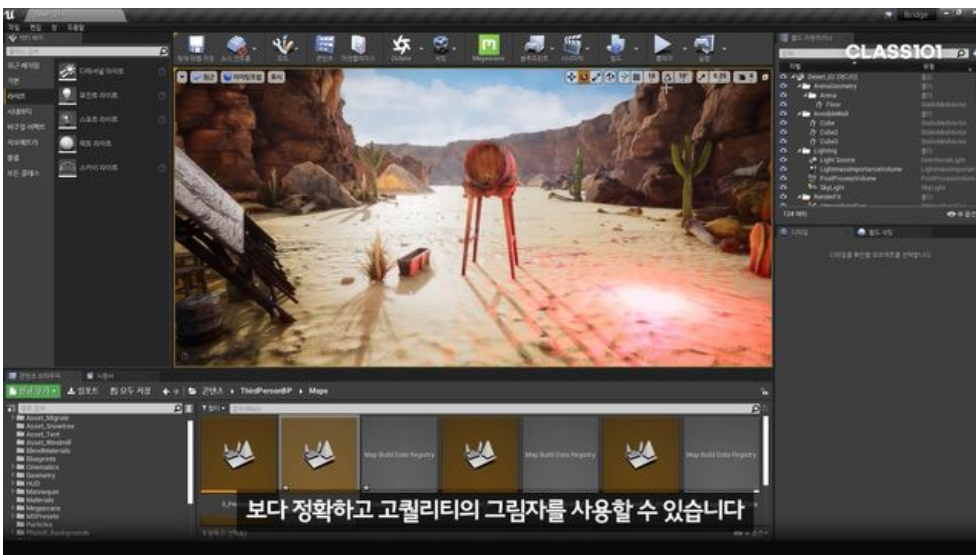


2:50 스태틱 라이트

게임에서 가장 사양을 많이 소비하는 부분이 바로 그림자 계산입니다. 광원의 위치를 바탕으로 영향을 받는 스태틱 메시를 계산하고, 그림자의 영향을 받을 다른 스태틱 메시까지 모두 계산해서 산출되기 때문입니다.

스태틱 라이트는 라이트 빌드라는 과정을 통해 게임 플레이 이전, 이런 그림자들을 텍스처와 혼합하는 과정을 거칩니다.

영향을 받을 그림자를 미리 계산해두고, 그것을 다른 스태틱 메시의 텍스처에 구워버리는 것이죠. 이것을 보고 라이트맵이라고 부릅니다.

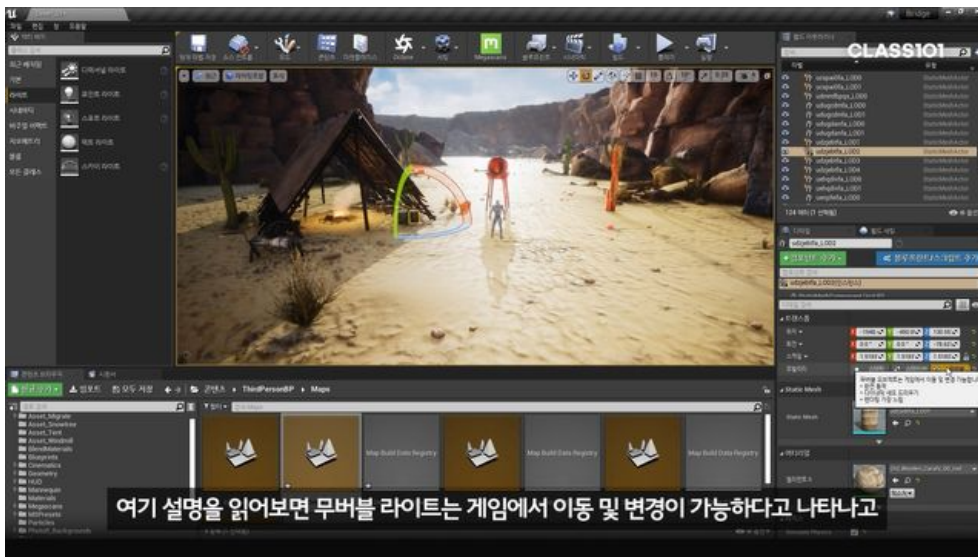


스태틱 라이트의 장점을 요약하면,

- 보다 퀄리티 좋은 그림자를 사용할 수 있다.
- 실제 게임 플레이 시 성능을 전혀 소비하지 않는다.

하지만 그만큼 단점도 있는데요,

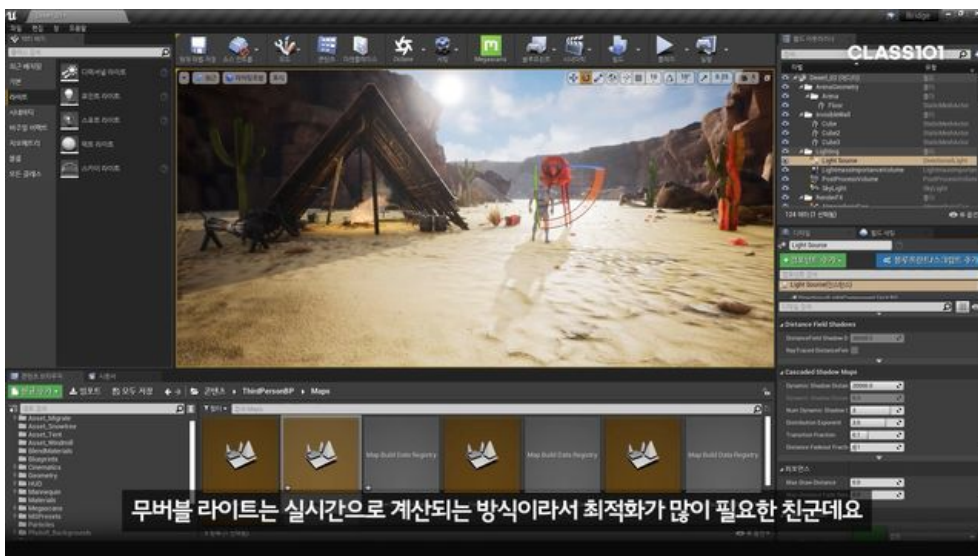
- 게임 플레이 도중 설정 값을 전혀 바꿀 수 없다. (고정됨)
- 변경 사항이 생긴다면 라이트 빌드를 해주어야 한다.
- Actor Hidden In Game을 체크한 오브젝트의 Cast Shadow를 꺼주어야 한다.



무버블 라이트는 게임 플레이 도중 실시간으로 계산되는 라이트입니다.

때문에 라이트 빌드가 필요 없는 대신 스태틱 라이트보다 많은 성능을 사용합니다.

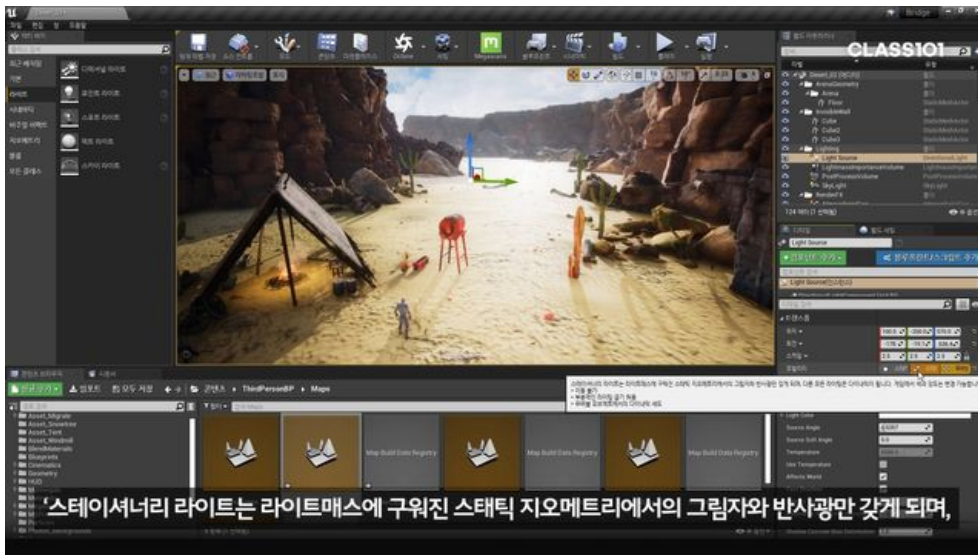
지금까지 무버블 라이트를 사용해왔던 것은 라이트 빌드가 필요 없기 때문에 빠른 결과를 확인할 수 있어 사용했습니다.



실시간으로 계산되는 그림자이기에 최적화의 방법도 여러 가지가 있습니다.

그림자에 대해서 Cascaded Shadow Map, CSM을 사용하는데요,

여기에서 그림자가 얼마나 많이 계산될지, 그리고 얼마나 디테일한 그림자를 만들 것인지 등을 제어할 수 있습니다.



11:59 스테이셔널 라이트

스테이셔널 라이트는 무버블과 스테틱의 중간 단계의 라이트입니다.

두 라이트 모빌리티의 특성을 모두 활용합니다.

스테이셔널 라이트는 게임 플레이 도중 일부만 바뀔 수 있습니다. 색, 강도는 바뀔 수 있지만 회전, 위치는 바뀔 수 없습니다.

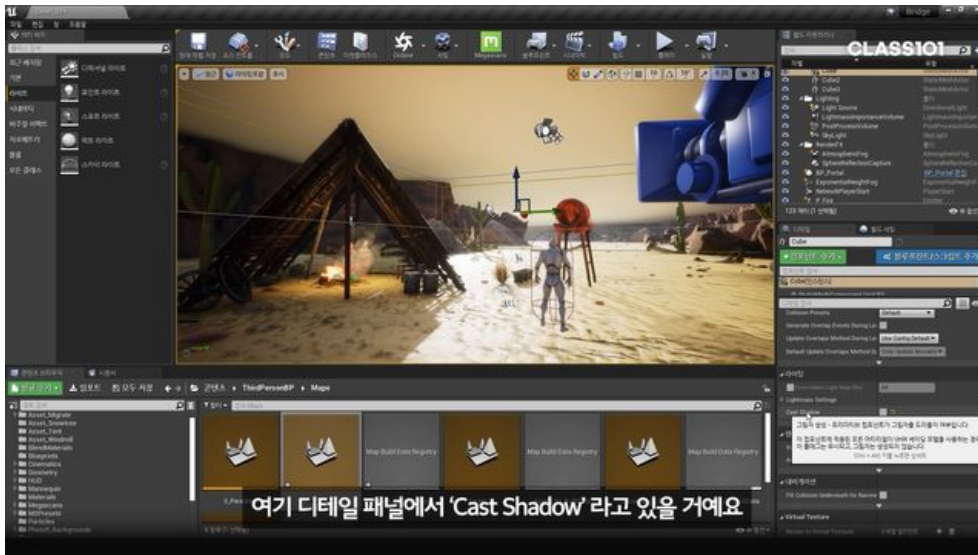
스테이셔널 라이트 또한 빌드 과정을 거쳐야 합니다. 스테이셔널리는 주로 실내와 실외 환경이 공존하는 레벨에서 사용하기 좋습니다.



스테이셔너리 라이트의 그림자는 텍스처의 각 채널에 저장되기 때문에 4개 이상 겹칠 경우 X 표시가 나타나며 무버블로 계산됩니다.

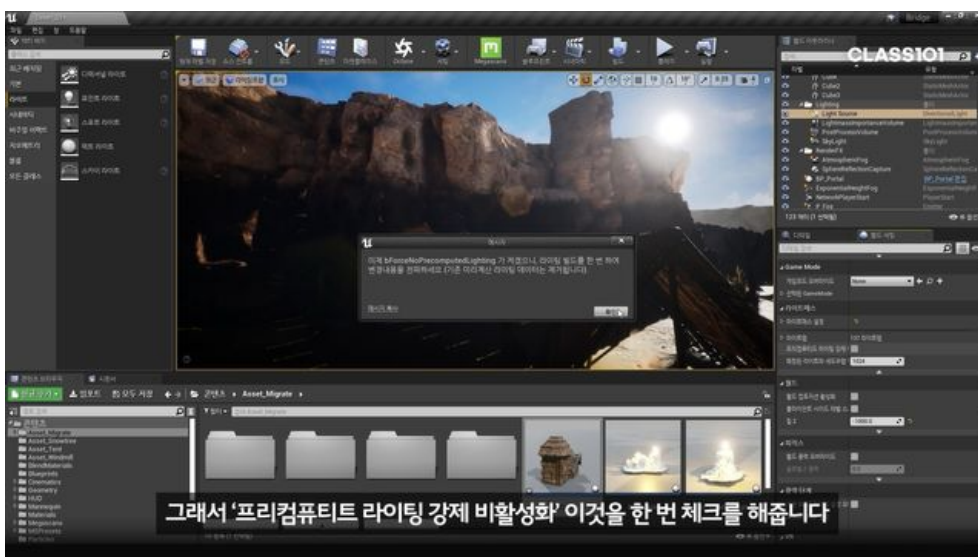
그렇기에 스테이셔너리 라이트의 Attenuation Radius를 조절하거나 위치를 적당히 겹치지 않게 설정하여 세팅하는 것이 좋습니다.

특히, 디렉셔널 라이트는 환경 전체에 영향을 주기 때문에 여러 개 사용하지 않는 것이 좋습니다.



스태틱 및 스테이셔너리 라이트를 빌드할 때 Actor Hidden in Game으로 체크하여 플레이 도중에만 보이지 않게 세팅한 것들도 모두 켜진 상태로 빌드가 되기 때문에 이것들의 디테일 패널에서 Cast Shadow를 모두 꺼주어 그림자가 생성되지 않도록 세팅합니다.

이것은 클래스 블루프린트 내에서 작업했던 것도 동일하게 해주어야 합니다. (예를 들면 콜리전 큐브 같은)



잘못 빌드된 라이트맵은 월드 세팅 패널에서 지워줄 수 있습니다.

프리컴퓨티드 라이팅 강제 비활성화를 체크한 뒤 빌드를 해주시면 빌드가 1초만에 완료되며 라이트맵이 모두 지워집니다.

라이트 관련 내용은 더 복잡하고 중요한 개념이 많아, 수업 노트 상단에 달아둔 링크를 통해 더 공부해보시는 것을 권장드리고 싶습니다!

[다음 수업 예고]

다음 시간에는 언리얼 엔진에서의 다양한 최적화 방법에 대해 알아보겠습니다. 감사합니다!