

Unity. Lighting 전체 정리

1. Global Illumination

3D 공간 상에서 빛을 표현하는 그 자체에 대한 방법, 알고리즘을 나타내는 개념.

- Direct Illumination : 광원에서 처음 반사된 빛을 보여주는 것.
- Indirect Illumination : 광원에서 반사되고 반사되고, 반사반사된 빛을 보여주는 것.

Direct / Indirect 는 이를 이루는 굉장히 기초적인 개념에 대한 설명.

2. Lighting in Unity

실제로 Unity 안에서는 **빛을 표현하는 방법**이 다음과 같이 나누어짐.

- Light Component in GameObject : Scene 상에 배치되며, 광원으로서의 역할을 수행. Component 내부 Property 를 바꿈에 따라, 광원의 Type 및 기타 설정을 바꿀 수 있음.
- Realtime : 매 프레임마다 광원 계산하여 빛을 반영하는 방법. 매 프레임마다 물체의 밝기 및 그림자를 조정함.
- Bake : 유니티가 Lighting Value를 고려하여 Rendering 한 결과를, 맵 위에 입힐 수 있는 큰 Texture로서 만든 다음에, 이를 오브젝트 위에 입히는 것으로서 빛을 반영하는 방법. 처음 한번만 렌더링하며, 렌더링한 결과는 더 이상의 연산을 필요로하지 않으므로, 높은 퀄리티를 얻을 수 있다.

이를 간단하게 정리하면 다음과 같다.

Light Source
(Light Component
In GameObject)

— Bake
— Realtime.

(물론 나쁜 게 항상 있지만,
unity에서 다 해서 뭐가
전혀 별 없다.)

— Bake 라면 : ① Light를 Bake.
② Object를 Static으로.

→ Light Mapping Setting 이
중요한 것임.

간략하게 보면,
천장에 붙어!!

정말 간단해 보인다.

그런데, 이 각각의 요소 안에 내부적으로 들어있는 굉장히 많은 Property 들과.

이를 활용하여 내가 원하는 Lighting 의 방향을 구현하는 것, 이끌어내는 것이 문제다.

이를 각각 살펴보자.

3. Light 의 세부요소

1) Light Source - Component

Type

- Directional
- Spot
- Point
- Area : TV 화면 처럼 빛이 나오게 하는 역할을 한다.

Color

빛의 색을 결정

Mode

- Realtime : 매 프레임마다 광원 계산 -> 매 프레임마다 밝기 및 그림자 조정
- Baked : 조명을 그대로 구워서, 그 상태 그대로 보여줌 : 그림자가 고정되어 따로 움직이지 않는 경우가 많음.
- Mixed : 둘 다 섞었다.

Intensity

빛의 세기

Indirect Multiplier

간접광의 세기 // 간접 간접광에 영향을 미침.

Shadow type

그림자의 성격을 결정함.

- Strength : 그림자의 세기
- Resolution : 그림자의 해상도 (퀄리티)
- Bias : 그림자를 밀어냄 (부자연스러운 그림자나 버그 있으면 사용)
- Normal Bias : 그림자를 쪼그려트림
- Near Plane : Point 등과 관련 - 가까이 있는 것부터 사라지게 만드는 Clipping 느낌.

Cookie

빛을 방해할 녀석을 설정. 구름을 등의 Texture 를 입혀준다고 생각하면 됨.

- Cookie size : 방해할 Texture 의 Size 를 조정

Draw Halo

후광 설정. 주변에 후광을 비칠지 말지를 설정 (Point 등에서 의미가 있음.)

Flare

카메라로 태양을 바라볼 때, 생기게 되는 Flare 현상을 넣을 수 있음.

Camera 로 태양을 보면, Lens Flare 처럼 빛을 발산하는 효과를 넣음.

(Texture 넣어서 모양 결정할 수 있음.)

Render Mode

Important 선택하면 조명 품질이 좋아짐.

Not Important 는 중요하지 않은 광원 대충 처리.

Culling Mask

특정 Layer 를 비출지 안 비출지. (조명의 영향을 받을 Layer 를 고르는 것.)

2) LightMap Setting

Bake(Lightmap) 에 관한 설정을 담당.

[Unity Manual - Progressive Lightmapper](#) 를 참고하면, 더 자세하고 많은 항목에 대해 탐색할 수 있다.

Lightmapper

Bake(Lightmap) 해주는 기술을 결정 : Enlighten / Progressive이 존재.
Progressive를 추천함. (실시간으로 Rendering 되는 걸 보여주고, 더 신기술.)

Progressive Updates (Prioritize View에서 이름 바뀜.)

현재 보고 있는 부분을 우선적으로 Rendering 하는 것을 결정.

+ Multiple Importance Sampling

최근에 추가되었다. 이 설정을 활성화하면 환경 샘플링에 중요도 기반 멀티 샘플링을 사용할 수 있다.

- 장단점 : This generally leads to faster convergence when generating lightmaps, but can lead to noisier results in certain low frequency environments.

Samples

직접광과 반사광에서 작용하는 퀄리티를 결정.

정확히는, 각 텍셀(텍스처를 구성하는 단위)에서 방출된 샘플(경로)의 수를 의미한다.
(우리가 보통 생각할 때, 광원에서 나온 빛이, 반사하여 물체에 맞닿고, 반사하여, 우리 눈에 들어 온다고 생각하는데. Ray Tracing 에서는 그 반대. 물체에서 나온 것을 반사시켜 광원에 어떻게 들어 가나를 체크한다.)

- Direct
- Indirect
- Environment

Bounces

반사광이 튕기는 개수, 노이즈에 관여.

Resolution

단위당 Texture 개수. Bake 된 Texture의 실질적인 해상도를 결정

Lightmap Size

Lightmap의 크기.

Ambient Occulsion

인접한 두 오브젝트 사이에 있는 Shadow 다.

Lightmap Setting 에 있는 이 수치를 낮춤으로서 해당 내용 또한 조정할 수 있다.

Compress Lightmap

Light Map 의 Texture 에 대한 압축을 지원한다. 이 경우, Light Map 의 품질은 조금 떨어질 수 있지만, 용량이 확 줄어들게 된다.

Indirect Intensity

반사광의 세기이다.

Albedo boost

반사광 / 간접광에 대해 세기와, 튀어나온 간접광의 개수.

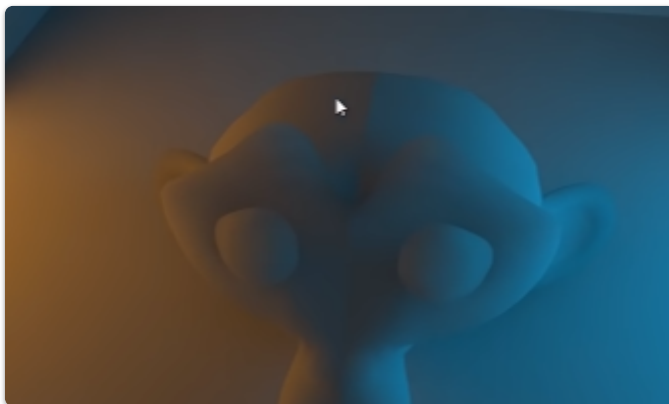
3) 그 외

Enable Generate Light Map UV

3D Object 의 Import 창에 떠 있는 성질로, 우리가 해당 오브젝트가 LightMapUV 를 가지고 있음을 체크해줄 수 있다. 이를 체크해줄 경우, Unity 가 자동으로 해당 Object 의 Light Map UV 를 연산하고 설정할 수 있다.

Stitching

물체의 UV 등으로 인해 Light 의 연산에서 다음과 같이 일자로 선이 가 있는 것을 없애줄 수 있는 방법.



문제 예시 사진 :

- Object 의 Mesh Renderer 에 가면, Lightmap Setting 이 있고, Stitch Seams 를 체크하면 해당 문제가 없어진다.

색이 뒤섞이거나, 잘못 표시되어 있는 문제 해결

해당 현상을 Bleeding 이라고 한다. 이를 해결하기 위해서는 LightMap padding이나 Lightmap Resolution 을 조정하면 된다.

씬이 어두울 경우

너가 지금 가지고있는 씬이 너무 어둡다면, Indirect Intensity를 늘리거나, Albedo Boost 를 높임으로서 조정할 수 있다.

Glow

이는 Post Processing 의 기술과 관련되어 있다. 우리의 Light 에 Glow를 더할 수 있는 방법 또한 소개하는데, Glowing Halo System 을 사용하고자 한다.

Post Processing 을 더하고, Bloom 을 선택하여, Camera가 빛에 대해 가질 Effect를 조정하자.

+ Area Light (23.07.09)

Area Light 는 추가를 하더라도 생각하거나 의도한 대로 제대로 써지지 않는 경우가 굉장히 많았으므로, 추가적인 학습 및 정리를 하기로 함.

[\(3\) Area Light in Unity - Unity Lighting Tutorial 07 - YouTube](#) : 해당 영상을 참고.

- Realtime / Bake Lighting 에 대한 설명을 우선적으로 진행.
- Area Light > Width 와 Height 를 변경
- 반영될 물체에 대해 Static 을 체크.
- 딱히 도움이 될만한 내용이 해당 영상에 나오진 않은 듯함.

[광원 타입 - Unity 매뉴얼 \(unity3d.com\)](#) : Range 를 신경써 봐야 겠음.

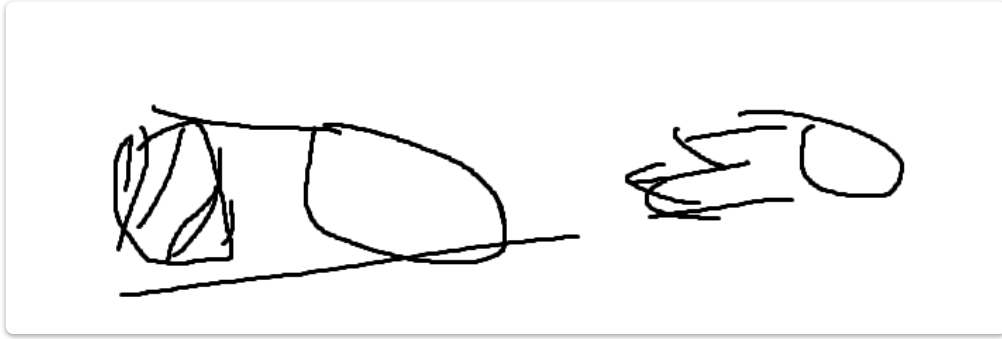
알아낸 결론

- Area Light 는 자신이 가진 Z 축을 방향으로 발사된다.
- Range 방향을 고려하여야 한다.
- Area Light 는 거리가 멀 수록 그 세기가 꽤나 엄청 줄어든다.
- Transparent Object 인 경우 있든 없든, 그 차이는 별로 없다.

Q. 추가로 궁금한 거 : Mesh Renderer 에 있는 Cast Shadow 가 의미하는 바는 무엇?

[Mesh Renderer 컴포넌트 - Unity 매뉴얼 \(unity3d.com\)](#) : 다음 내용을 정리하면 다음과 같다.

- Cast Shadow : 이 친구가 그림자를 만드냐 안 만드냐. 즉, 밑의 그림처럼 막히게 되는 물체임



을 의미.

- Receive Shadow : 그림자를 받냐 안 받냐. 자신의 표면에 그림자를 만들지, 만들지 않을지를 의미.

4. 사례 예시

LMHPOLY's Lighting And Post-Processing Low Poly Scene Tutorial

[출처](#)

Setting

1. Window > Lighting > Scene Section
2. Enable Auto Generate (to generate lighting data)
=> 이걸 하고 나면, Scene 은 Skybox의 ambient color 를 사용하며, black shadow 가 없어진다. (어둡던 게 엄청 밝아짐)
3. Project Setting > Player > Other Setting > Color Space - Linear
Color 가 좀 더 사실적으로 만들어진다.
4. Quality Setting 에서 Quality 를 ultra 로 설정한다.
Shadow 를 Hard and Soft Shadows 와 Shadow Resolution 을 Very High Resolution 으로 설정한다.

Light

5. Directional Light 의 값을 조정하여 시간을 표현할 수 있다.
6. Directional Light 의 Normal Bias 를 0 로 한다. (Light Bleeding 을 없애기 위해)
7. Lighting > Environment > Environment Lighting > Source 를 Color 로.
이후 Ambient Color 를 조정하여 환경을 설정할 수 있다.

Baking

8. Auto Generate 를 끈다. (우리는 수동으로 Lighting 만들어 줄 것이므로!)
9. Baked Global Illumination 은 켜져 있어야 한다!

10. 바로 아래의 Lighting Mode 를 Baked Indirect로 바꾼다. (오직 bake Indirect lighting data 만 원하므로)
11. Light Mapper 를 Progressive GPU 로 설정한다.
12. Lighting 하고 싶은 물체를 Static 으로 설정하고, Lighting Setting 으로 돌아가, Clear Lightmap + Generate Lightmap 을 선택한다.
13. Environment > Other Setting > Fog Enable and Mode Linear.
이를 통해 카메라 포그 또한 설정할 수 있다!

5. 추가 정리 / 추가 개념

Light Probe

Light Probes provide a way to capture and use information about light that is passing through the empty space in your scene.

Light Probe 는 빛에 대한 정보 - 내 씬에서 통과하고 있는 빛 - 를 캡처하고 사용하는 방법을 제공한다.

라이트 프로브는 라이트맵과 마찬가지로 “베이킹된” 조명 정보를 씬에 저장한다.

라이트맵은 씬의 표면에 닿는 광원에 대한 조명 정보를 저장하지만 라이트 프로브는 씬의 **빈 공간**을 지나가는 광원에 대한 정보를 저장한다는 점에서 차이가 있다.

Light Probes are positions in the scene where the light is measured (probed) during the bake.

Light Probe는 씬 안에 있는 위치들인데, bake 되는 동안 해당 위치에 어떤 빛이 지나가는지가 측정된 위치들이다. 런타임 시점에 동적 게임 오브젝트에 닿는 간접광은 해당 오브젝트와 가장 가까운 라이트 프로브의 값을 사용하여 근사화된다.

(해당 위치를 지나가는 베이킹 된 빛 정보가 Light Probe 에 저장. 이는 Runtime 중 물체의 간접광을 표시할 때 사용.)

용도

- 씬에서 움직이는 오브젝트에 고품질 조명(간접 반사광 포함)을 제공하는 것
- 배경에 Unity의 **LOD 시스템** 이 사용될 때 Static 배경에 대한 조명 정보를 제공하는 것. (LOD 시스템에 따라, 해당 배경에 관한 정보가 연산되지 않을 경우, Light Probe 를 바탕으로 한 간접광 정보가 반영된 표면이 LOD 시스템에 따라 전달되나 봄.)

Light Probe 의 사용

Light Probe Group

씬에 라이트 프로브를 배치하려면 Light Probe Group 컴포넌트가 연결된 게임 오브젝트를 사용해야 한다.

메뉴에서 Component > Rendering > Light Probe Group 으로 이동하여 Light Probe Group 컴포넌트를 추가할 수 있다.

Light Probe Group 을 활용한 Light Probe 배치 / Light Probe Group 내 컴포넌트 등과 관련해서는 [링크](#) 에 내용이 나와있으므로 참고할 것.

Lightmap 관련 Error 발생

! [Pasted image 20230304221835.png](#)

이렇게 LightMapping 이 깨지는 문제가 발생함에 따라 해결 방향 모색.

[라이트매핑 깨짐 문제](#) 링크를 살펴보면, 보통 모델링이나 머테리얼에 문제가 있는 것이라고 함. 다음 방법을 시도해보려 함.

1. 모델링의 Generate Lightmap UVs 옵션을 활성화 후 Lightmap UVs Setting에서 Pack Margin 을 조절
2. Material의 Double Sided Global illumination을 활성화

1번 방법의 시도에서 잘 먹혔음. Generate Lightmap UVs를 활성화 하였고, Pack Margin의 경우, Margin Method에 따라 달라졌는데, 현재 버전은 Calculate 가 기본인 것을 확인함.

! [Pasted image 20230304222942.png](#)

잘 적용된 것을 확인할 수 있었음.

Q. Generate Lightmap UVs 의 Margin Method란 무엇인가?

A. Margin Method 란, UV island 사이의 빛 번짐(bleeding of light)을 줄이거나 없애기 위해서, UV island 주변의 패딩 또는 마진을 조정하기 위해 Unity에서 사용하는 기법이다.
(Light bleeding은 UV island의 빛이 인접한 UV island로 흘러넘쳐 최종 조명 결과에 이음새나 아티팩트를 유발할 때 발생한다)

Margin Method는 두 가지 방법을 지니는데,

1. "Calculate" method는 Unity에서 라이트맵 UV를 자동으로 생성하는 방법이다. Unity는 메시의 모양과 크기, 사용자가 지정한 패딩 또는 마진 세팅을 기반으로 UV 좌표를 자동으로 계산한다.
2. "Manual (수동)" method는 라이트맵 UV를 생성하는 더 정밀하고 커스터마이징이 가능한 방법이다. Unity에 내장된 UV 에디터나 Maya 또는 Blender와 같은 외부 프로그램을 사용하여 메시의 각 버텍스에 대한 UV 좌표를 수동으로 생성하고 조정할 수 있으며, 복잡한 물체에 대해 더 높은 퀄리티를 보장할 수 있다.

이후 LightMapping 의 더 자세한 기술적 사항을 파악하고 싶다면, [Manual - Lightmapping](#) 참고할 것.

그런데, 기술적 사항들에 대해 집중적으로 나와있어서, 그리 권장하지는 않음. 차라리 Youtube tutorial 로 보는 게 좀 더 실증적일 것 같고. 이 기술적인 내용은 좀 더 완속하고, 진짜 기술적으로 뭔가 바꾸어 보고 싶을 때 하면 좋을 듯함.

자료

HIYU's Lighting World in Unity 1 minute

출처

1. Material > Shader > Skybox/Procedural > Drag it to Sky
=> Material 을 조정하여 하늘 환경을 바꿀 수 있음.

2. Post Processing > Post - process Volume Component
Is Global
New Profile
Color Grading > Mode > ACES

Camera > Add Post Process Layer Component
Layer 지정 > Volume Component 에서 지정했던 Layer.

3. Light in Unity Modes
Realtime
Mixed
Bake

Light 의 Mode 를 Bake 로 설정한 후? > 작업을 완료하기.

Bake 할 것들을 전부 선택하고 Static 으로 하기.

Window Menu > Lighting Setting

Set the lightmapper to GPU > Bounce to 4 > Lightmap Resolution 을 좀 낮추기.

Reflection Probe를 맵의 중앙에 두기

Generate Lighting!

=> 예쁘게 구워진 모습을 확인할 수 있음!

Brackeys's LIGHTING in Unity

출처 : 2018년 자료이므로, 꽤나 다름이 있을 수 있음.

Global Illumination의 개념

3d 그래픽스에서 씬을 렌더링할 때 사용하는 조명 알고리즘

Light 와 Light 가 반사된 것을 바탕으로 우리는 물체를 바라본다. 이런 실제의 동작과 Computer 내부의 동작이 동일하게 작동한다.

단, CG 에서, Lighting 을 다음과 같은 두 가지로 구분한다.

- Direct Illumination : 광원에서 처음 반사된 빛을 보여주는 것.
- Indirect Illumination : 광원에서 반사되고 반사되고, 반사반사된 빛을 보여주는 것.
(이것은 너무 자원을 많이 잡아먹는다.)

Global Illumination 에서 처음부터 끝까지 Indirect Illumination을 계산하기에는, 우리 컴퓨터가 좋지 못하다.

Enlighten : Solution for delivering real-time global illumination.(real-time global illumination의 해법) 이것은 Realtime 에서 Light 의 연산을 잘 수행할 수 있도록 도우면서도, 퍼포먼스에 무리가 가지 않도록 한다. 이는 다른 말로 **Baking**이라고도 부른다.

Baking 이란? : 유니티가 Lighting Value를 고려하여 Rendering 한 결과를, 맵 위에 입힐 수 있는 큰 Texture로서 만든 다음에, 이를 오브젝트 위에 입히는 것.

Enlighten 및 Baking을 이용하기 위해선, New Progressive LightMapper 를 이용한다.

이는 Path Tracing Based light mapper 로, 해당 단어는 실제처럼 scene 을 simulate 하는 lighting 을 진행하는 것을 의미한다.

다시 rendering 할 필요 없이 Light Properties 를 적용할 수 있다.

In Unity

Setting

Light Source (광원) 을 우선적으로 만들어 볼텐데.

1. 평소에 사용하는 Light 를 사용할 수도 있고.
2. Emisive (발광하는) Object 를 만들 수도 있다.

해당 튜토리얼에선 발광 오브젝트를 만드는 방법을 보여준다.

발광 오브젝트를 만드는 방법은 다음과 같다.

1. Material 을 만든다.
2. Material 의 Emission 을 체크한다.

이 경우, 실제 Scene 에서는 아직 적용되지 않는 것을 볼 수 있는데, 이는 Object 의 Emission 의 경우, 리얼타임에서는 적용되지 않기 때문이다. LightMap 을 사용해야 해당 Object 에서 발광되는 느낌이 드는 것을 확인할 수 있다.

단, Bake 를 위해서는 발광하는 물체와, 나머지 발광에 영향을 받는 물체가 Static 이어야 한다. 그래야 발광을 잘 적용할 수 있을 것이기 때문이다.

| Scene 을 보는 설정을 Light 로 설정하면, Light 가 보이는 모습을 확인할 수 있네.

Enable Generate Light Map UV

3D Object 를 Import 할 때 마다, 우리는 그 오브젝트가 LightMapUV 를 가지고 있음을 체크해주어야 한다. 이를 체크해줄 경우, Unity 가 자동으로 해당 Object 의 Light Map UV 를 연산하고 설정할 수 있다.

Lighting Setting

Window > Lighting > Lighting Setting

이 동영상에서 한 것.

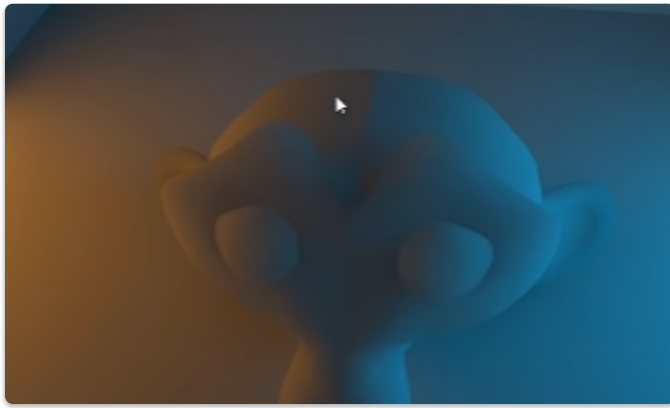
1. Environment 의 Skybox Material 삭제.
=> 이 경우, 좀 어두워지고 푸르러진 모습을 확인할 수 있는데, 이는 Unity 의 default ambient color가 dark blue 이기 때문이다. Ambient Color 를 없앤다.
2. Realtime Global Illumination의 체크를 해제한다.
=> Unity 가 사용하던 Real time Mapping 을 해제하였기 때문에, 이제 완전히 Bake 된 Map 만을 사용할 것이다.
3. Lightmapping Setting 의 경우, Enlighten 에서 Progressive 로 전환한다.
4. 가끔씩, Auto Generating 을 해제하고, Clear Bake Data 한 후, 다시 AutoGenerate를 체크해 줄 필요가 있다.
이후, Dark 해졌는데, 이를 해결하기 위해서 발광 오브젝트의 Emission Mode 설정을 Realtime에서 Bake 로 바꾼다.

Testing

이제, 굉장히 사실적인 모습을 볼 수 있고, 하지만 연산에 과정이 오래 걸리기 때문에, 노이즈가 생기는 모습을 확인할 수 있다.

Stitching

물체의 UV 등으로 인해 Light 의 연산에서 다음과 같이 일자로 선이 가 있는 것을 없애줄 수 있는 방법.



문제 예시 사진 :

- Object 의 Mesh Renderer 에 가면, Lightmap Setting 이 있고, Stitch Seams 를 체크하면 해당 문제가 없어진다.

LightMap Setting

사실적이어 보이는 것은 지금도 충분하다. 이에 따라 Light Map Size 를 줄이고자 한다.
LightMap Size를 512로 줄이고, Compress LightMap 을 체크한다.

Bleeding 이 발생한 것을 볼 수 있는데, 이를 해결하기 위해서는 LightMap padding이나
Lightmap Resolution 을 조정함으로써 해결할 수 있다.

Direct Sample 과 Indirect Sample 을 조정함으로써, 반사되는 광선의 수를 조정할 수 있다.

Final Render 에 가까워 지고 있다!

Direct Sample 이나 Bounce 는 Light Map 의 Size에는 영향을 주지 않고, Render 시간에만 영향을
주기 때문에, Final Render 하기 전에 한번 올리자.

Ambient Occlusion & Filtering

Ambient Occlusion 에 대해 얘기해보자.

인접한 두 오브젝트 사이에 있는 Shadow 다.

Lightmap Setting 에 있는 이 수치를 낮춤으로서 해당 내용 또한 조정할 수 있다.

Filtering : 아마 이까지 했다면, 물체에 발생하는 부자연스러움은 Light 때문이 아니라, Unity 에서
이루어지는 Filtering Stage 에서 발생하는 것일 거다. (조금의 노이즈가 껴보이는 이유)

이는 Post - Processing 과정을 통해 조정할 수 있다.

Light map 에 관한 Filtering 을 다행히 우리는 조정할 수 있는데, Lightmap setting 에 있는 수치를
조정함으로써 조정할 수 있다!

Glow

너가 지금 가지고있는 씬이 너무 어둡다면, Indirect Intensity를 늘리거나, Albedo Boost 를 높임
으로서 조정할 수 있다.

우리의 Light 에 Glow를 더할 수 있는 방법 또한 소개하는데, Glowing Halo System 을 사용하고자 한다.

Post Processing 을 더하고, Bloom 을 선택하여, Camera가 빛에 대해 가질 Effect를 조정하자. 이를 통해 잘 조정할 수 있다.

LMHPOLY's Lighting And Post-Processing Low Poly Scene Tutorial

[출처](#)

Lighting

Setting

1. Window > Lighting > Scene Section
2. Enable Auto Generate (to generate lighting data)
=> 이걸 하고 나면, Scene 은 Skybox의 ambient color 를 사용하며, black shadow 가 없어진다. (어둡던 게 엄청 밝아짐)
3. Project Setting > Player > Other Setting > Color Space - Linear
Color 가 좀 더 사실적으로 만들어진다.
4. Quality Setting 에서 Quality 를 ultra 로 설정한다.
Shadow 를 Hard and Soft Shadows 와 Shadow Resolution 을 Very High Resolution 으로 설정한다.

Light

5. Directional Light 의 값을 조정하여 시간을 표현할 수 있다.
6. Directional Light 의 Normal Bias 를 0 로 한다. (Light Bleeding 을 없애기 위해)
7. Lighting > Environment > Environment Lighting > Source 를 Color 로.
이후 Ambient Color 를 조정하여 환경을 설정할 수 있다.

Baking

8. Auto Generate 를 끈다. (우리는 수동으로 Lighting 만들어 줄 것이므로!)
9. Baked Global Illumination 은 켜져 있어야 한다!
10. 바로 아래의 Lighting Mode 를 Baked Indirect로 바꾼다. (오직 bake Indirect lighting data 만 원하므로)
11. Light Mapper 를 Progressive GPU 로 설정한다.

12. Lighting 하고 싶은 물체를 Static 으로 설정하고, Lighting Setting 으로 돌아가, Clear Lightmap + Generate Lightmap 을 선택한다.
13. Environment > Other Setting > Fog Enable and Mode Linear.
이를 통해 카메라 포그 또한 설정할 수 있다!

Post Processing

대강 Post Processing Volume 이랑, Camera 에 붙여서, Profile 및 기타 설정하고 적용하는 과정.

케이디 라이트 입문 강좌

출처 : Component 의 각 부분에 대해서 설명해주는 이론 위주의 강좌이긴 하지만, 앞에서 실증적인 케이스를 많이 봐서 그런지. 한번 이해하는데 큰 도움이 되는 느낌이다!

Light Component

Type

- Directional
- Spot
- Point
- Area : TV 화면 처럼 빛이 나오게 하는 역할을 한다.

Color

빛의 색을 결정

Mode

- Realtime : 매 프레임마다 광원 계산 -> 매 프레임마다 밝기 및 그림자 조정
- Baked : 조명을 그대로 구워서, 그 상태 그대로 보여줌 : 그림자가 고정되어 따로 움직이지 않는 경우가 많음.
- Mixed : 둘 다 섞었다.

Intensity

빛의 세기

Indirect Multiplier

간접광의 세기 // 간접간접광에 영향을 미침.

Shadow type

그림자의 성격을 결정함.

- Strength : 그림자의 세기
- Resolution : 그림자의 해상도 (퀄리티)
- Bias : 그림자를 밀어냄 (부자연스러운 그림자나 버그 있으면 사용)
- Nomal Bias : 그림자를 쪼그려트림
- Near Plane : Point 등과 관련 - 가까이 있는 것부터 사라지게 만드는 Clipping 느낌.

Cookie

빛을 방해할 녀석을 설정. 구름을 등의 Texture 를 입혀준다고 생각하면 됨.

- Cookie size : 방해할 Texture 의 Size 를 조정

Draw Halo

후광 설정. 주변에 후광을 비칠지 말지를 설정 (Point 등에서 의미가 있음.)

Flare

카메라로 태양을 바라볼 때, 생기게 되는 Flare 현상을 넣을 수 있음.
Camera 로 태양을 보면, Lens Flare 처럼 빛을 발산하는 효과를 넣음.
(Texture 넣어서 모양 결정할 수 있음.)

Render Mode

Important 선택하면 조명 품질이 좋아짐.
Not Important 는 중요하지 않은 광원 대충 처리.

Culling Mask

특정 Layer 를 비출지 안 비출지. (조명의 영향을 받을 Layer 를 고르는 것.)

Baked 자세히

1. Light Bake로 설정.
이 경우, 기존 영향을 미치던 Light 가 확 사라지는데, 이 이유는 Unity 가 해당 Object 가 배경 인지, 배경에 영향을 미칠 소품일지를 파악할 수 없기 때문에.
(움직이는 Object 의 경우, Bake 하면 의미가 없으니까.)
2. 고정된 녀석임을 알려주기 위해 Static 으로 설정.
(Lightmap Static 으로 설정할 수도 있음.)

Bake 의 단점 : 움직이는 객체 있으면 영향을 받지 않음.

=> 이를 임시방편적으로 해결하기 위해 작용하는 것이 Probe.

Probe가 촘촘하면 촘촘할 수록 자연스러운 빛 표현이 가능하다고 한다.

Light Probe Group을 바탕으로, Probe를 굉장히 많이 설정하고 있음.

(주변광 정보를 취득해서 받는 식으로?) => 그림자 자체는 없고, 물체의 Object 만 영향을 끼치도록 함.

LightMap Setting

Bake(Lightmap) 에 관한 설정을 담당.

Lightmapper

Bake(Lightmap) 해주는 기술을 결정 : Enlighten / Progressive이 존재.

Progressive를 추천함. (실시간으로 Rendering 되는 걸 보여주고, 더 신기술.)

Prioritize View

현재 보고 있는 부분을 우선적으로 Rendering 하는 것을 결정.

Samples

직접광과 반사광에서 작용하는 퀄리티를 결정.

Bounces

반사광이 튕기는 개수, 노이즈에 관여.

Resolution

단위당 Texture 개수. Bake 된 Texture의 실질적인 해상도를 결정

Lightmap Size

512 : Size 크기

Ambient Occlusion

인접한 두 오브젝트 사이에 있는 Shadow 다.

Lightmap Setting 에 있는 이 수치를 낮춤으로서 해당 내용 또한 조정할 수 있다.

Compress Lightmap

Indirect Intensity

Albedo boost

반사광 / 간접광에 대해 세기와, 튀어나온 간접광의 개수.

Script

Light Component 에 대한 값을 변경하여 주는 Script를 짤 수 있다! 정도.