

Animation Baking Studio

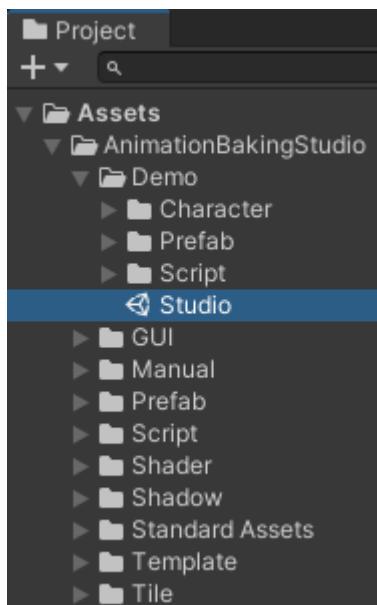
(ver. 4.1.2)

Animation Baking Studio는 움직이는 3D 객체를 촬영해서, 스프라이트 시트(PNG), 애니메이션 클립 등을 만들어 주는 에셋입니다.

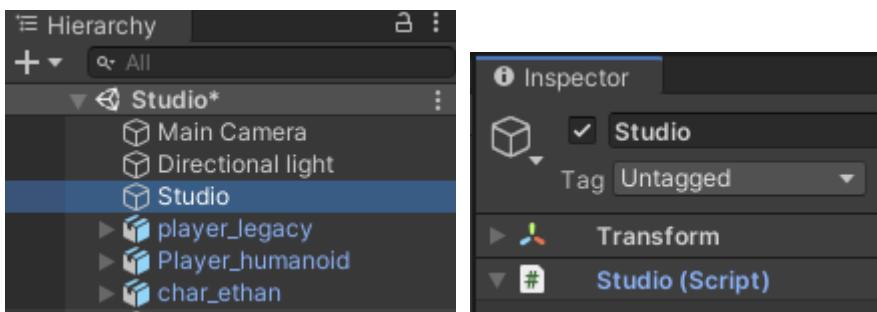
빠르게 맛보기

스튜디오, 주 카메라, 모델이 있으면 촬영할 수 있습니다.

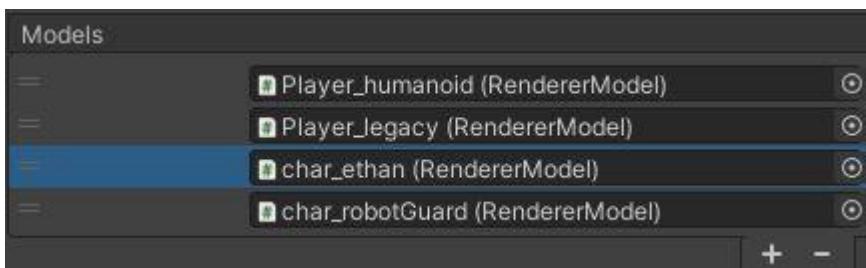
1. Project 창에서 동봉되어 있는 Studio 씬을 여세요.



2. Hierarchy 창에서 Studio 객체를 선택하고, Inspector 창에서 Studio 컴포넌트를 확인하세요.



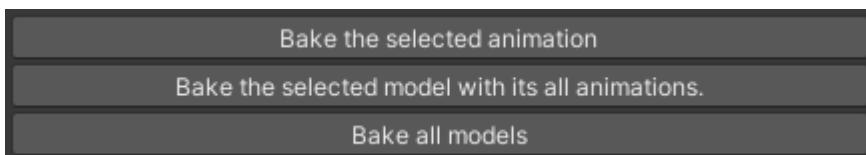
3. 모델 리스트에서 모델 하나를 선택하세요.



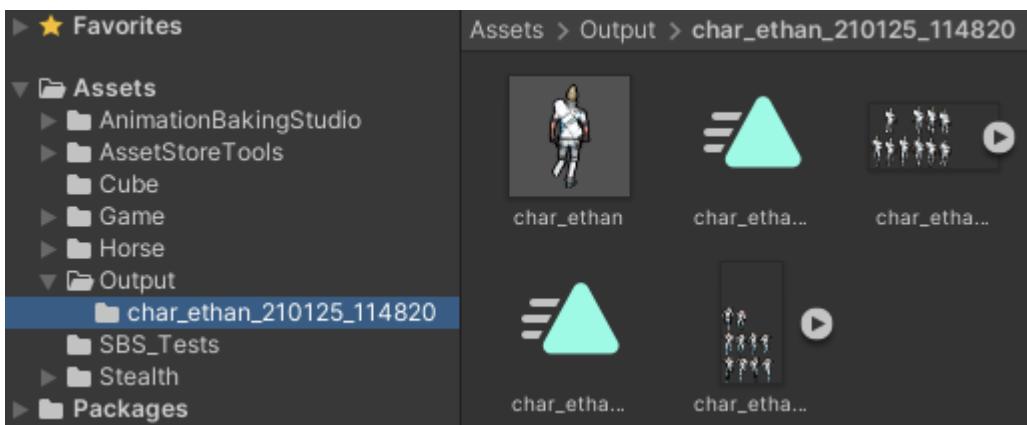
4. 맨 아래의 Choose a directory under Assets 버튼을 클릭하여 결과 파일들을 저장할 폴더를 선택하세요.



5. 세 가지 베이킹 버튼 중 하나를 클릭하면 베이킹이 시작됩니다.



6. 선택된 폴더에 스프라이트 시트와 애니메이션 클립 등의 파일이 생성됩니다.



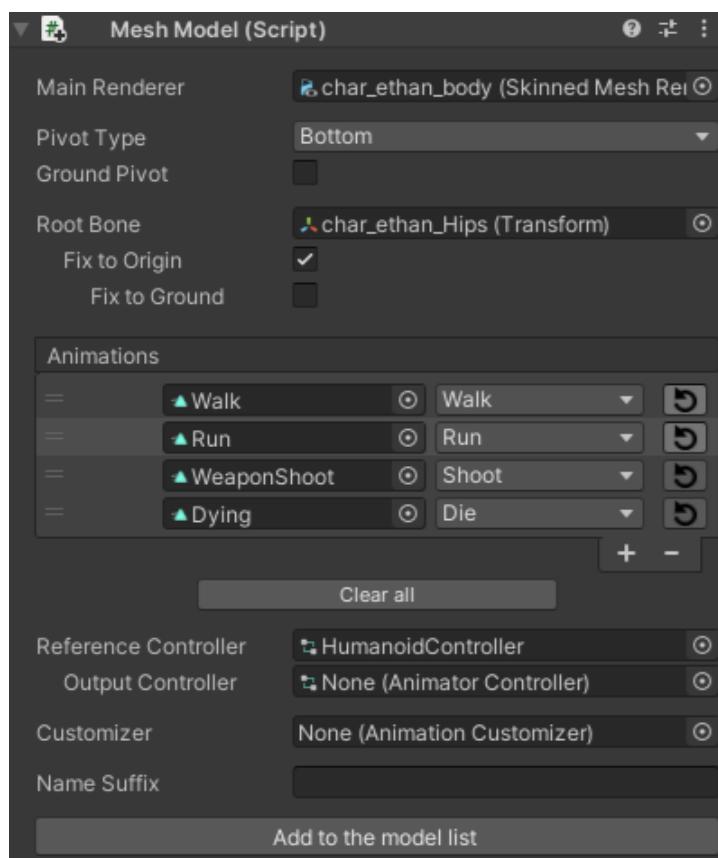
모델

모델은 촬영할 대상 객체로, 메시 모델과 파티클 모델, 두 가지 모델 타입이 있습니다:

메시 모델

메카님 시스템을 사용하는 객체를 촬영하기 위해서는 반드시 Animator 컴포넌트에 아바타와 컨트롤러가 지정되어 있어야 합니다.

Mesh Renderer 혹은 Skinned Mesh Renderer를 가지고 있는 객체에 MeshModel 스크립트를 추가하면 메시 모델이 됩니다.



Main Renderer는 모델의 크기를 계산하기 위해서 필요합니다. 만약에 객체의 계층 구조 안에 자동으로 선택된 렌더러의 부피 보다 더 큰 부피를 가지고 있는 다른 렌더러가 있다면, 그것을 주 렌더러로 지정하는 것이 좋습니다.



보통 계층적인 모델의 월드 위치는 객체의 바닥 부근에 위치합니다. 즉, Pivot Type은 Bottom이 됩니다.



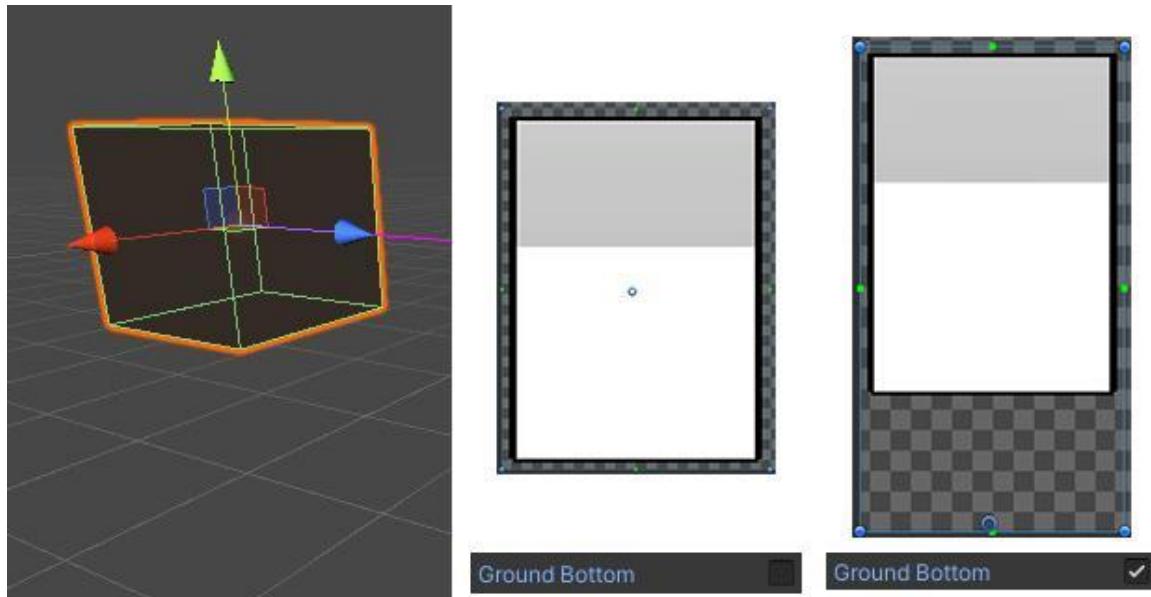
하지만, 혹시라도 월드 위치가 모델의 수직상의 중심에 있다면, Center로 바꿔야 합니다.

계층적이지 않은 객체의 월드 위치가 수직상의 중심에 있기 때문에 Pivot Type이 노출되지 않으며 내부적으로 Center로 설정됩니다.

만약 객체의 정면 방향이 z-forward가 아니라면, 빈 객체를 만들어서 회전된 객체를 그 아래에 놓고 MeshModel 스크립트는 루트 객체에 추가하면 됩니다.

잘라내기 기능을 켠 체로 바닥으로부터 떠 있는 모델을 촬영하면 결과 스프라이트에서 객체의 하단 아래 영역에 해당하는 부분은 전부 잘리게 됩니다. Ground Pivot을 켜면 모델의 하단부터 바닥까지의 영역이

모델의 영역에 포함돼서, 그 부분이 잘리는 것을 방지할 수 있습니다.



움직이는 모델을 촬영할 때는 애니메이션이 객체를 제자리에서 벗어나지 않게 하도록 Animator 컴포넌트의 Apply Root Motion을 끄는 것이 좋습니다. 그럼에도 불구하고, 촬영할 때 모델이 제자리를 벗어난다면, Root Bone을 지정하고 Fix to Origin을 켜세요. 루트 본 객체는 보통 엉덩이, 골반, 혹은 등골 뼈입니다.



만약, Fix to Origin 과 Fix to Ground를 모두 켜면, 점프 혹은 나는 애니메이션이 실행될 때 모델을 바닥에 붙여 놓을 수 있습니다.

계층적이지 않은 모델은 Root Motion 과 하위 항목들이 노출되지 않습니다.

애니메이션을 촬영하려면 애니메이션은 Animations에 등록되어야 합니다. '+' 버튼을 클릭해서 필드를

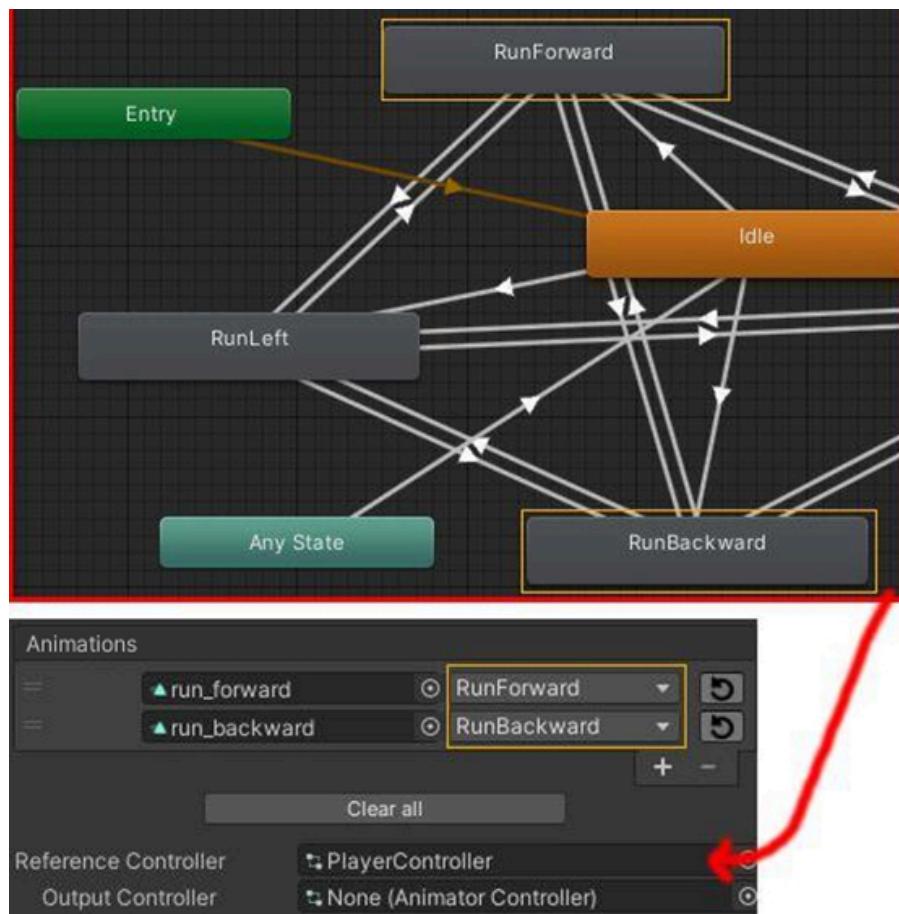
하나 추가한 후 애니메이션 클립을 필드에 넣을 수도 있고, 애니메이션 클립을 리스트에 드래그 앤 드롭해서 필드를 추가할 수도 있습니다.



각 필드의 이 켜져 있으면, 반복되는 애니메이션 클립이 생성되게 할 수 있습니다.

Reference Controller는 결과 컨트롤러를 구성할 때 참고하기 위해 필요합니다.

지정한 참조 컨트롤러 안에 애니메이션 상태가 있다면 애니메이션 리스트의 각 필드에 상태들 중 하나를 선택할 수 있는 콤보 박스가 나타나며, 필드의 애니메이션 클립에 대응되는 상태를 선택하면 됩니다.



베이킹 과정에서 상태와 트랜зision이 Output Controller로 지정한 컨트롤러에 추가됩니다. 만약 비어 있으면 새로운 컨트롤러가 생성되어 사용됩니다. (두 개 이상의 레이어를 가지고 있거나 서브 머신을

가지고 있는 컨트롤러는 참조 컨트롤러로 사용할 수 없습니다.)

AnimationCustomizer 클래스를 상속하고 UpdateFrame 함수를 구현한 스크립트를 프리팹 객체에 넣고, 그 객체를 Customizer에 넣으면, 뱠킹 시 애니메이션의 각 프레임마다 모델 객체를 원하는데로 움직일 수 있습니다.

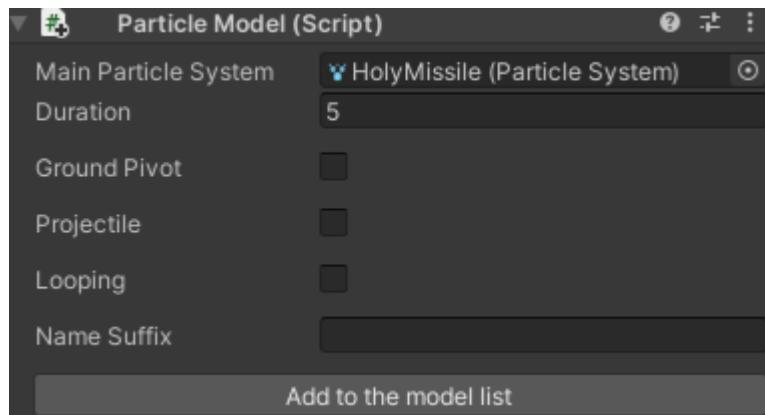
출력 파일명을 짓는데 쓰일 모델 이름에 특정 단어를 붙이고 싶다면 Name Suffix에 입력하세요.

Add to the model list 버튼을 클릭하면 Studio의 모델 리스트에 등록됩니다.

Project 창에서 폴더를 선택 후 Assets 메뉴 혹은 Project 창 컨텍스트 메뉴의 Animation Baking Studio > Instantiate Objects as > Mesh Model을 실행하면, 선택된 폴더 내의 객체들을 한꺼번에 씬에 생성하고 MeshModel 스크립트를 추가합니다.

파티클 모델

Particle System을 가지고 있는 객체에 ParticleModel 스크립트를 추가하면 파티클 모델이 됩니다.



Main Particle System은 객체의 대략적인 사이즈를 구하고 전체 파티클 시스템을 플레이하기 필요합니다. 계층 안에 자동으로 선택된 파티클 시스템 보다 더 주가 되는 파티클 시스템이 있다면, 교체하는 것이 좋습니다.

Duration은 단일 플레이 지속 시간입니다. 최초에 자동으로 주 파티클 시스템의 Duration 값을 사용하지만 촬영하기에 너무 긴 시간으로 설정될 수 있습니다. 그래서 몇 번의 샘플링을 통해 적절한 시간을 직접 설정하는 것이 좋습니다. 자세한 정보는 [## 샘플링](#) 파트를 확인하세요.

잘라내기를 캔 채로 공중에 떠 있는 모델을 촬영하기 위해서는 Ground Pivot을 켜는 것을 고려해보세요. 자세한 정보는 [## 메시 모델](#) 파트를 확인하세요.

Projectile을 켜면, 매 **Update**마다 **Speed Vector** 만큼 이동하는 모습을 촬영할 수 있습니다. 이것은 플레이 모드 베이킹 시에만 유효한 기능으로, 자세한 내용은 [### 플레이 모드 베이킹](#) 파트를 확인하세요.

Looping을 켜면, 반복되는 애니메이션 클립이 생성되게 할 수 있습니다.

프리팹을 생성하기 위해서는 **Sprite Prefab**과 **Prefab Builder**가 지정되어야 합니다. 자세한 정보는 [## 메시 모델](#) 파트를 확인하세요.

모델 이름에 특정 단어를 붙이고 싶다면 **Name Suffix**에 입력하세요.

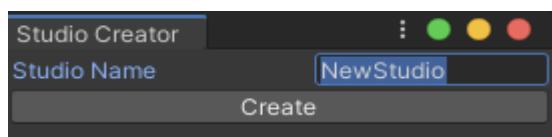
Add to the model list 버튼을 클릭하면 스튜디오의 모델 리스트에 등록됩니다.

Project 창에서 폴더를 선택 후 **Assets** 메뉴 혹은 *Project* 창 컨텍스트 메뉴의 *Animation Baking Studio* > *Instantiate Objects as* > *Particle Model*을 실행하면, 선택된 폴더 내의 객체들을 한꺼번에 씬에 생성하고 *ParticleModel* 스크립트를 추가합니다.

스튜디오

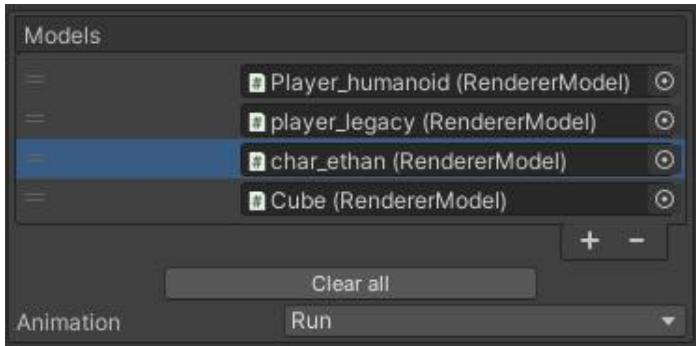
모델을 촬영하기 위해서는 스튜디오가 필요합니다. 빈 객체를 생성하고 **Studio** 스크립트를 추가하세요.

Assets 메뉴 혹은 *Project* 창의 컨텍스트 메뉴 - 마우스 오른쪽 버튼 클릭해서 나오는 메뉴 –에 있는 *Animation Baking Studio* > *Studio Creator*를 이용해서 스튜디오 객체가 설정된 씬을 생성할 수 있습니다.



Studio Creator 창에서 **Studio Name**을 입력하고 **Create** 버튼을 클릭하면, 새로운 씬이 만들어지면서 열립니다.

모델 리스트 & 애니메이션

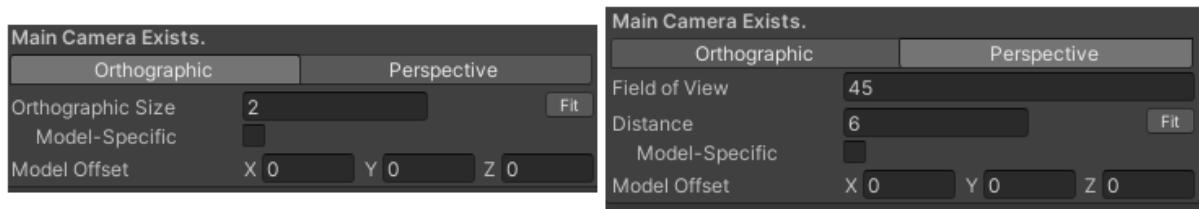


먼저 모델 리스트에 촬영하고자 하는 모델들을 등록해야 합니다. ‘+’ 버튼을 클릭해서 필드를 추가한 후 모델을 넣을 수 있고, 드래그 앤 드롭으로 등록할 수도 있습니다. 또한, [# 모델](#) 파트에서 설명했듯이, (Mesh/Particle) Model 컴포넌트에서 Add to the model list 버튼을 클릭해서 등록할 수도 있습니다.

리스트에 등록된 모델 중 하나가 선택되면, 그에 맞게 스튜디오의 일부 항목들이 바뀔 수 있습니다.

애니메이션을 가지고 있는 메시 모델이 선택되면, 해당 모델에 등록된 애니메이션 중 하나를 선택할 수 있는 Animation 콤보 박스가 나타납니다.

주 카메라



촬영을 하기 위해서는 주 카메라 - MainCamera 태그가 있는 카메라 - 가 반드시 필요합니다. 주 카메라가 있으면, 스튜디오에는 “**Main Camera Exists.**”라고 나옵니다. 그렇지 않으면, 주 카메라가 없다는 빨간 문구와 함께 주 카메라 객체를 생성할 수 있는 버튼이 나옵니다.



Orthographic 투영과 Perspective 투영 중 하나를 선택할 수 있으며, 각각 Orthographic Size / Field Of View와 Distance를 조정할 수 있습니다.

모델 리스트에서 메시 모델 중 하나가 선택된 상태에서 Fit 버튼을 누르면 모델에 맞게 Orthographic Size가 변경됩니다. 완벽하게 모델에 딱 맞지 않고, 움직이는 동안 모델의 바운더리가 바뀌기 때문에, 직접 세부 조절이 필요합니다.

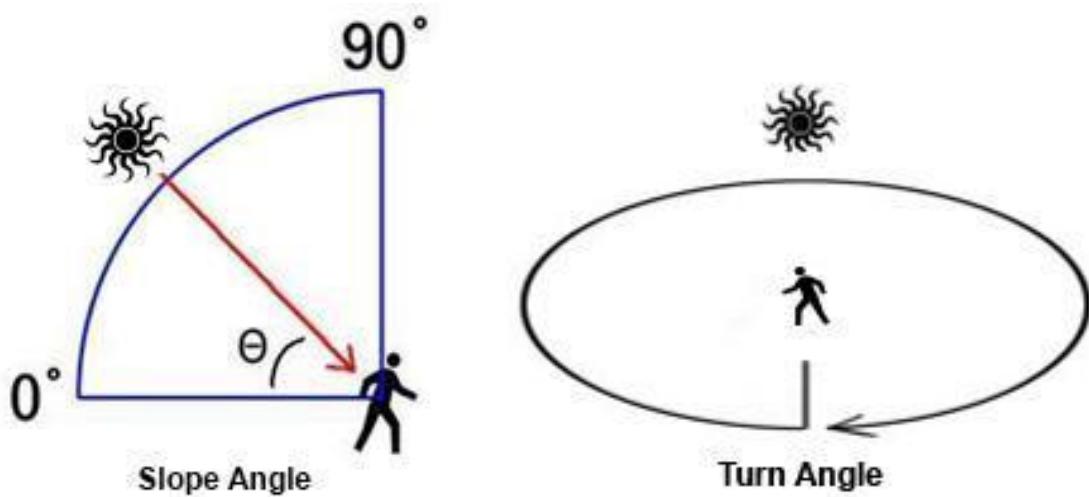
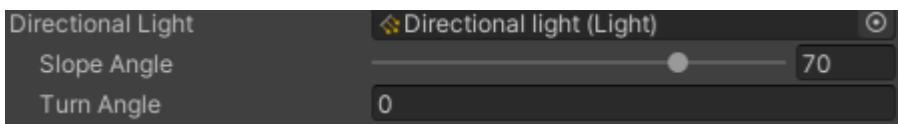
파티클 모델이 선택되면, 투영 타입 툴바는 비활성화 되고 내부적으로 직교 투영으로 설정됩니다.

방향 조명



Directional Light는 촬영하는데 있어서 필수는 아닙니다.

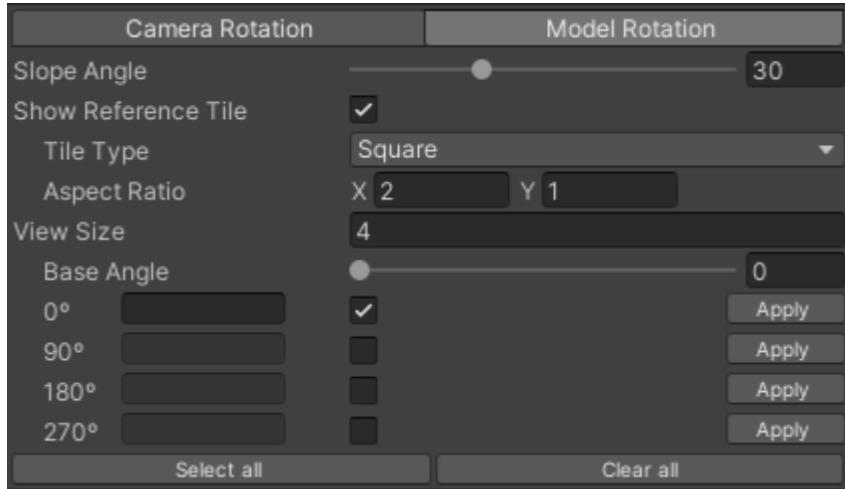
주 카메라가 회전할 때 모델에 드리우는 음영을 최소화하기 위해서는 Follow Camera Rotation을 켜서 방향광도 주 카메라와 같은 방향을 바라보게 하는 것이 좋습니다. 혹은 Follow Camera Rotation을 꺼울 때 나타나는 Slope Angle과 Turn Angle을 조정해서 직접 방향광을 회전시킬 수 있습니다.



Look At Model 버튼을 클릭하면 대상 모델을 바라보는 방향으로 방향광이 회전되고, 두 각도도 자동으로 수정됩니다.

주 카메라가 이동할 때 방향광도 주 카메라와 같은 위치로 이동시키고 싶으면 Follow Camera Position을 켜세요. 하지만, '방향'광이기 때문에 조명의 위치는 크게 중요하진 않습니다.

뷰

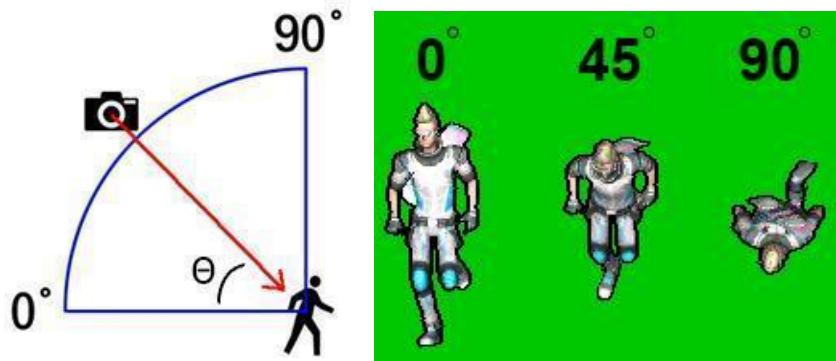


하나의 뷰는 하나의 바라보는 방향이라고 할 수 있고, 주 카메라와 모델의 방향에 따라 수많은 뷰가 있을 수 있습니다.

Camera Rotation의 경우에는 선택된 모델을 중심으로 주 카메라가 회전하고, Model Rotation의 경우에는 모델이 그 자리에서 회전합니다. 보통은 두 회전 방식의 결과가 차이가 없지만, 매트 그림자를 사용하는 경우는 달라집니다.



Slope Angle를 조정해서 x 축을 기준으로 주 카메라를 회전시킬 수 있습니다. 0도로 설정하면 정면 뷰 혹은 사이드 뷰가 되고 90도로 설정하면 탑 뷰가 됩니다.



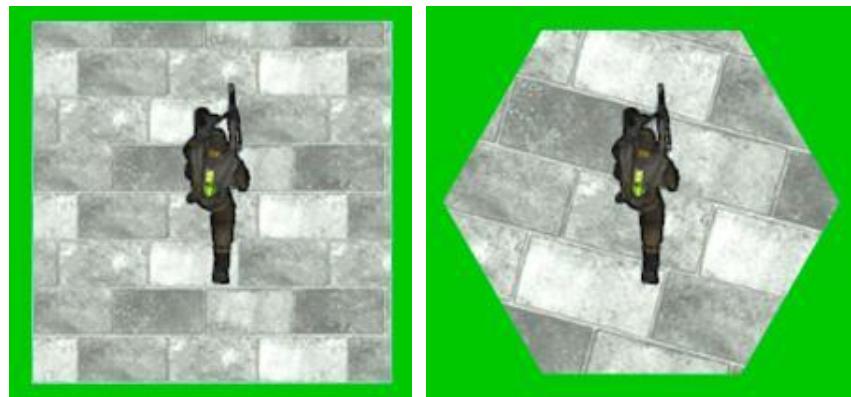
만약에 타일 기반의 2.5D 게임을 만든다고 하면, 타일의 가로/세로 비율에 맞게 캐릭터를 그려야 할 것입니다.



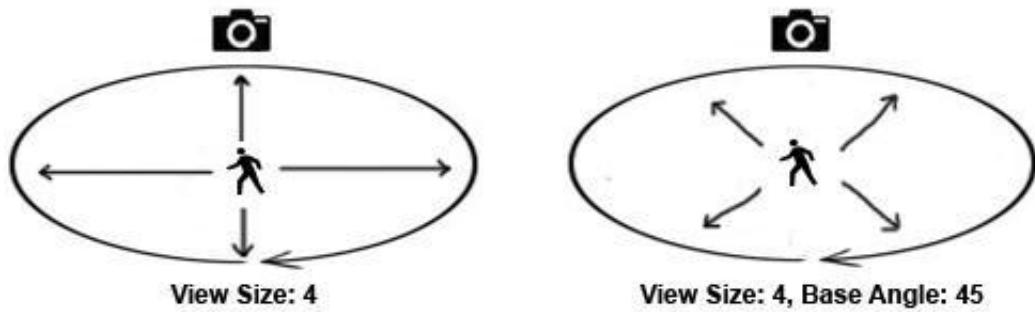
Show Reference Tile을 체크하면 참고용 타일이 나타나는데, 이 타일은 촬영 중에는 사라집니다.

Aspect Ratio를 조정하면 카메라가 회전되면서 경사각이 자동으로 계산됩니다. 반대로, 경사각을 조정해도 Aspect Ratio가 영향을 받습니다.

Square 타일과 Hexagon 타일 중 하나를 선택할 수 있습니다.

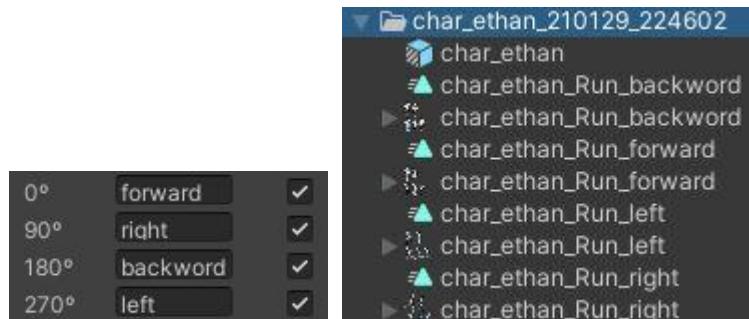


[360도 / View Size] 만큼의 뷰가 만들어지고, 각 뷰는 [Base Angle + (360도 / View Size) * index] 방향각을 가지고 있습니다.



각 뷰에 있는 Apply 버튼을 클릭하면 각 뷰의 방향에 맞게 주 카메라 혹은 모델이 회전합니다.

체크한 뷰들을 연속으로 촬영하게 되며, 생성된 파일의 이름 끝에 회전각이 문자열로 붙습니다. 만약 각 뷰 이름을 지으면, 회전각 문자열 대신에 그 이름을 이용합니다.

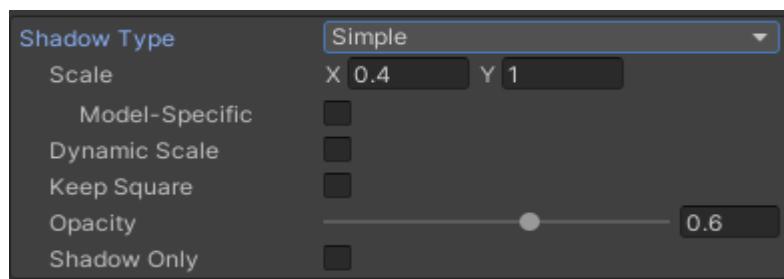


그림자

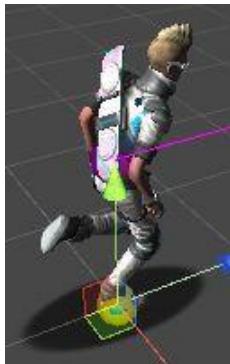
세 가지의 그림자 타입이 있습니다; 심플한 그림자, 탑 다운 그림자, 매트 그림자

(파티클 모델에는 그림자 기능이 지원되지 않습니다.)

간단한 그림자



타원형의 간단한 그림자입니다.



Scale을 조정하면 그림자 필드 객체가 가로 혹은 세로로 스케일 됩니다.

기본적으로, 모델 리스트에 등록된 모든 모델에 같은 스케일 값을 적용합니다. 모델 별로 스케일을 적용하고 싶다면, **Model-Specific**을 체크하고 개별적인 스케일을 조정하세요.

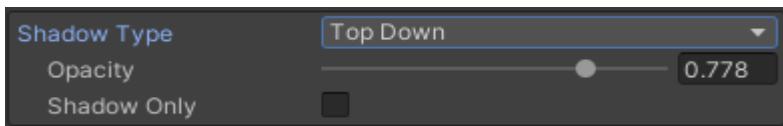
모델이 움직이는 동안, 모델의 크기에 맞게 그림자의 크기를 맞추고 싶으면, **Dynamic Scale**을 체크하세요.

가로 세로 같은 비율을 유지하고 싶다면 **Keep Square**를 체크하세요.

Opacity는 실질적으로 `~/Shadow/SimpleShadow` 재질의 주 색상과 연결된 속성으로, 수정하면 그림자의 투명도를 조정할 수 있습니다. 또한, 재질에 등록된 텍스처를 수정해서 다른 스타일의 그림자를 만들 수 있습니다.

Shadow Only를 체크하면 모델을 숨기고 그림자만 촬영할 수 있습니다.

탑 다운 그림자



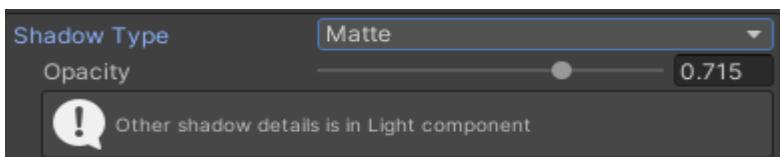
모델을 수직으로 내려다보는 별도의 카메라를 이용한 그림자입니다.



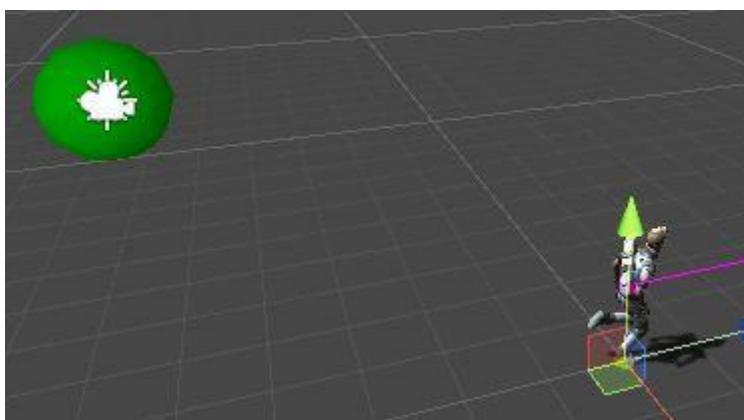
Opacity를 조정해서 투명도를 변경할 수 있습니다.

Shadow Only를 체크하면 모델을 숨기고 그림자만 촬영할 수 있습니다.

매트 그림자

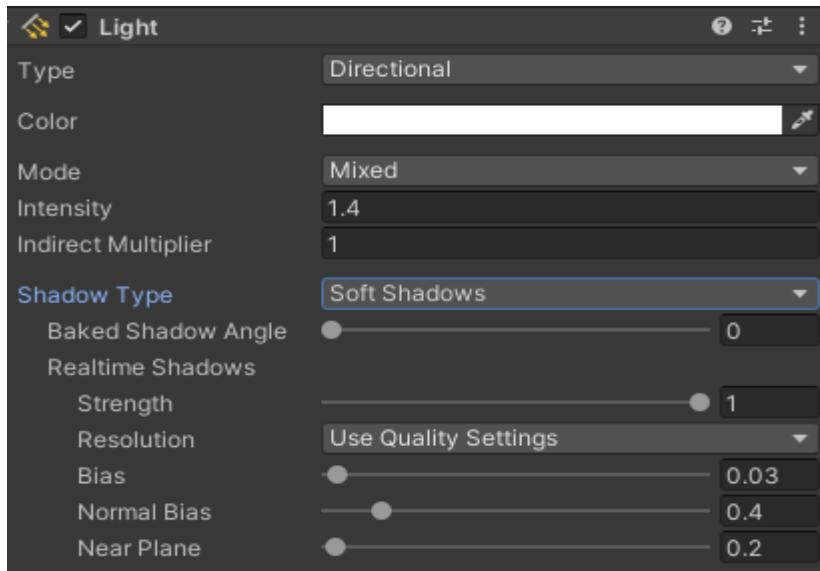


특수한 투명 바닥을 이용한 그림자로, 3ds Max 나 Maya에서 용어를 그대로 채용했습니다.

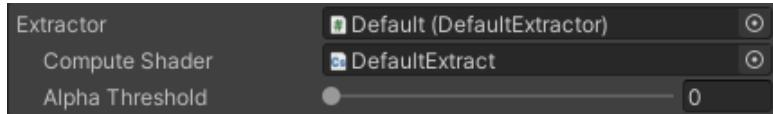


조명 속성 중 Follow Camera Rotation을 켜고 이 그림자 타입을 선택했다면, 모델이 시야를 완벽하게 가리기 때문에 Game 창에서는 어떠한 그림자도 볼 수 없을 것입니다.

Opacity를 조정해서 간단하게 투명도 변경할 수 있고, 방향광 객체의 Light 컴포넌트에서 상세하게 변경할 수 있습니다.



픽셀 추출

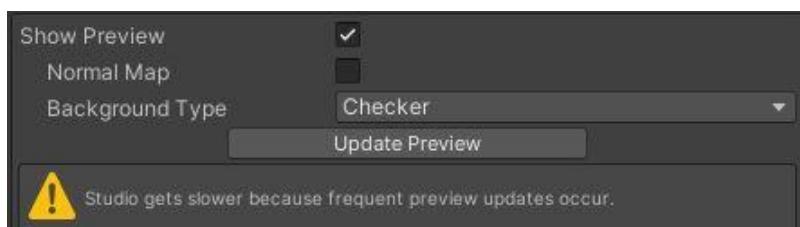


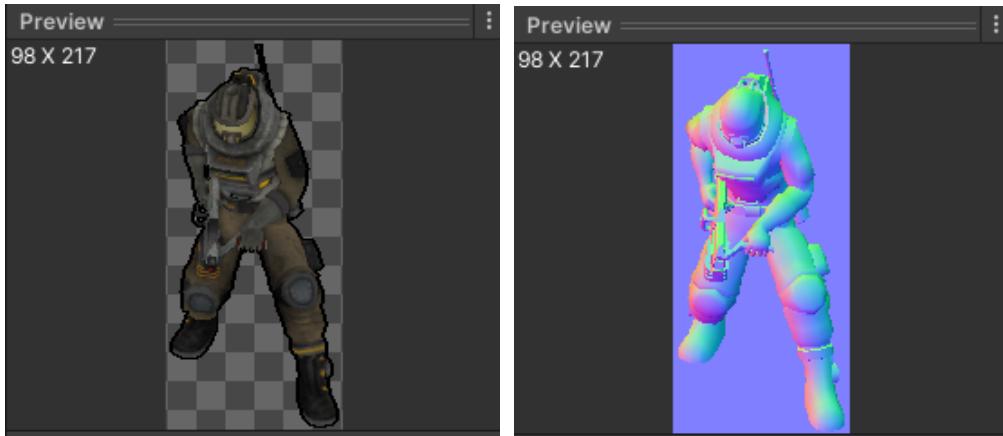
Extractor는 Game 창을 찍은 임시 이미지에서 배경을 제외하고 모델의 색상만 뽑아내는 역할을 합니다.

기본적으로 ~/Prefab/Extractor/Default로 지정되어 있으며, Extractor 추상 클래스를 상속하고 Extract 함수를 구현해서 새로운 추출기를 만들 수 있습니다.

만약, 대상 모델이 완전히 불투명이고 안티 에일리어싱 기능을 사용하지 않는다면, ~/Prefab/Extractor/Opaque로 교체해서 베이킹 속도를 향상시킬 수 있습니다.

프리뷰 창





프리뷰를 켜면 Studio 컴포넌트 맨 아래에 Preview 창이 나타나며, Show Preview를 켜면, 결과 이미지를 미리 보여줍니다.

만약, [## 잘라내기](#)가 켜 있으면, 잘려진 이미지 크기가 반영됩니다.

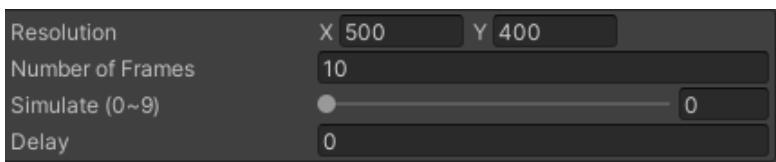
만약, [## 결과물](#)의 노말맵 생성이 켜 있으면, Normal Map 항목이 노출되고, 그것을 켜면 노말맵 이미지를 미리 볼 수 있습니다.

Background Type은 기본적으로 Checker인데, Single Color로 선택해서 단일 색상으로 채울 수도 있습니다.

이 기능이 켜 있으면, 스튜디오의 모든 속성이 바뀔 때마다 모델이 촬영되고 프리뷰 텍스처가 생성돼서 느려집니다.

프리뷰가 바로 업데이트 안되면 Update Preview 버튼을 클릭하세요.

촬영



Resolution은 [## 잘라내기](#)가 적용되기 전의 원본 텍스처 크기입니다.

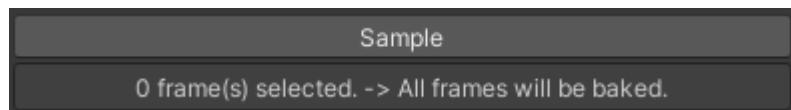
기본적으로, 각 애니메이션에 대해서, 0초부터 시작해서 (애니메이션 길이 / Frame Size) 마다 한 번씩 총 Frame Size 만큼 촬영됩니다.

Simulate 슬라이더를 사용해서 특정 프레임에서의 애니메이션 동작을 미리 볼 수 있습니다.

탭 다운 그림자를 사용할 때, 모델과 그림자가 잘 안 맞는 경우가 생길 수도 있습니다. 그럴 때는,

Delay를 조금 올려서 모델과 그림자 간의 싱크를 맞출 수 있습니다.

샘플링

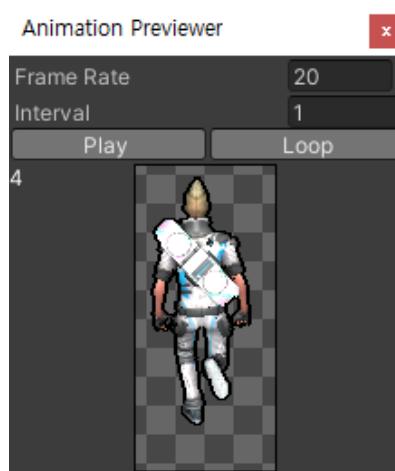


Sample 버튼을 클릭해서 애니메이션을 샘플링 해볼 수 있습니다. 샘플링이 완료되면 Frame Selector 창과, Animation Previewer 창이 등장합니다.

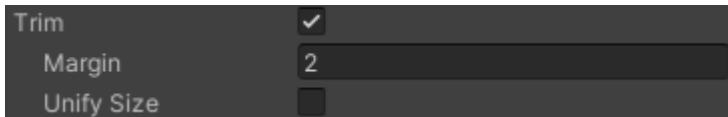
Frame Selector 창에서는 베이킹 과정에서 촬영할 프레임을 선택할 수 있습니다. 이미지들은 모델에 맞게 잘라져 있지만, 베이킹 과정에서 잘라내기를 수행 여부에는 영향을 주지 않습니다.



Animation Previewer에서 선택된 프레임들의 연결 동작을 확인할 수 있습니다. Frame Rate와 Interval을 수정해서 애니메이션 속도를 조정할 수 있고, 결과 애니메이션 클립에 적용됩니다.



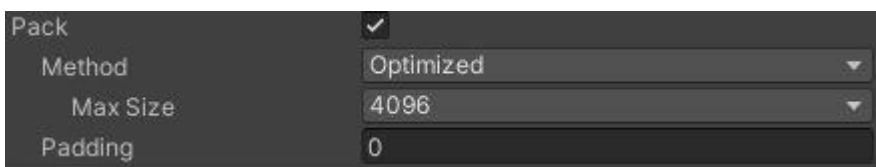
잘라내기



Trim을 켜면 결과 이미지가 모델에 딱 맞는 크기보다 Margin 만큼 더 크게 잘립니다.

Uniform Size를 켜면 단일 뷰 안에서 촬영되는 모든 프레임에 대한 이미지를 동일하게 잘라낼 수 있습니다.

패킹



Pack을 켜지 않으면 개별 이미지 파일이 생성되고, 켜면 스프라이트 시트가 생성됩니다.

두 가지 패킹 방식이 있습니다.

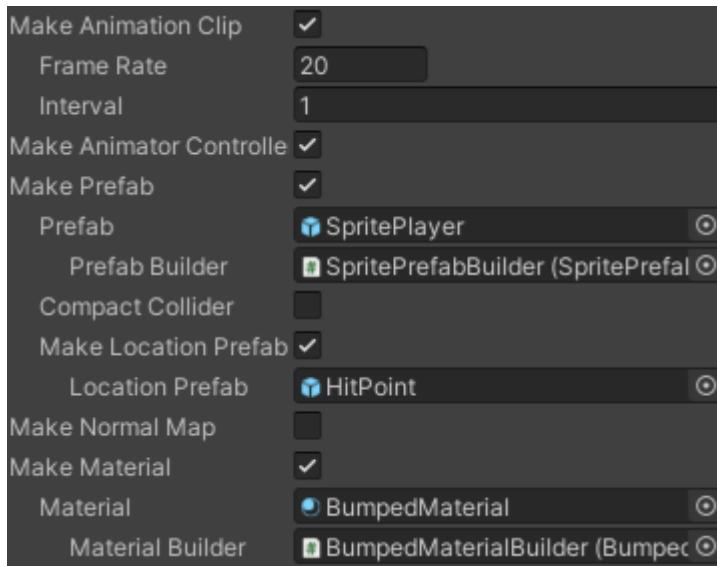
Optimized 방식은 유니티 엔진 자체의 기능으로, 가장 작은 스프라이트 시트를 만들 수 있도록 스프라이트를 배치합니다. Max Size 보다 크지 않은 가장 작은 사이즈의 스프라이트 시트가 생성됩니다.

In Order 방식은 스프라이트를 좌상단부터 순서대로 배치합니다. 모든 스프라이트를 담을 수 있되 Min Size 보다 작지 않은 스프라이트 시트가 생성됩니다.

스프라이트 사이에는 Padding 만큼의 간격을 줄 수 있습니다.

결과물

이미지 파일 외에도 애니메이션 클립, 애니메이터 컨트롤러, 프리팹, 노말 맵, 재질 등을 생성할 수 있으며, 이것들은 서로 참조되어 있어서 바로 사용할 수 있습니다.



Make Animation Clip을 켜면 베이킹 과정에서 생성되는 스프라이트, 재질 등과 연결된 애니메이션 클립을 생성할 수 있습니다.

이 항목은 [## 패킹](#)이 켜 있을 때만 노출됩니다.

선택된 모델이 참조 컨트롤러를 가지고 있고 **Make Animator Controller**를 체크하면, 애니메이터 컨트롤러를 생성하고 각 상태에는 애니메이션 클립이 등록되어 있습니다. 추가적인 내용은 [## 메시 모델](#) 파트를 참고해주세요.

이 항목은 [## 패킹](#)이 켜 있을 때만 노출됩니다.

Make Prefab을 켜면, 베이킹 과정에서 각 모델의 스프라이트 프리팹으로 지정한 객체를 생성합니다. 추가적인 내용은 [## 메시 모델](#) 파트를 참고해주세요.

Prefab은 결과 프리팹 객체를 만들기 위해서 필요합니다. 추가로, 생성되는 프리팹 객체에 결과 스프라이트와 애니메이터 컨트롤러를 연결하기 위해서 **Prefab Builder**도 지정되어야 합니다. 프리팹 빌더를 만들려면 새로운 스크립트를 생성해서 **PrefabBuilder** 클래스를 상속하고 필요한 메서드를 구현해야 합니다.

지정한 스프라이트 프리팹 안에 **BoxCollider2D** 컴포넌트가 있을 때, **Compact Collider**를 체크하면, 트림 마진이 0 보다 크더라도, 완벽하게 모델 크기에 맞도록 컬라이더 사이즈와 오프셋이 계산됩니다.



특정 빼대 객체의 위치를 결과 프리팹 객체 안에서도 추적하고 싶으면, 먼저, 원하는 대상 객체에 `~/Prefab/Location` 객체를 추가하고, 객체의 이름을 수정해 주세요.



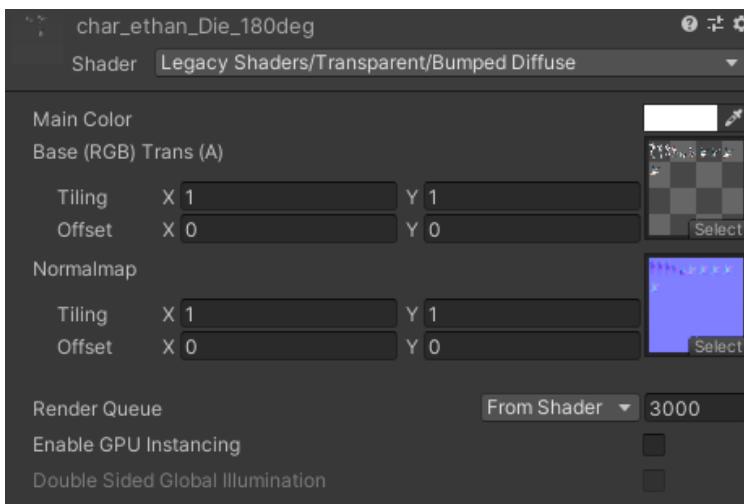
그 다음에 **Make Location Prefab**을 체크하고, 결과 로케이션 객체를 Location Prefab에 지정해두면, 베이킹 후에 결과 프리팹 객체안에 로케이션 객체가 있는 걸 볼 수 있을 것입니다. 그리고 그 로케이션 객체는 결과 애니메이션에 따라 함께 움직입니다.



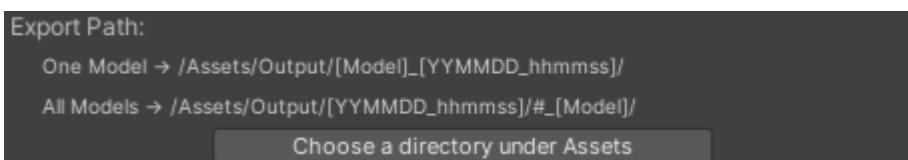
Make Normal Map을 체크하면, 모델 이미지 파일에 대응하는 노말 맵 파일이 생성됩니다. 제대로 된 노말 맵을 생성하기 위해서는 [## View](#)의 모델 회전 타입이 선택되어야 합니다.



대개는 **Sprite Renderer**에서 기본 스프라이트 재질을 사용하는데, 특정 재질을 지정하고 싶은 경우가 있습니다. (예를 들어, 노말 맵이 적용되는 **Bumped** 재질을 사용하는 경우) 이런 경우, **Make Material**을 켜고, 생성할 재질을 **Material**에 지정하세요. 결과 재질에 결과 모델 이미지와 노말 맵 이미지를 연결하기 위해서 **Material Builder** 또한 필요합니다. Demo 폴더 안의 **Material/BumpedMaterial**, **Prefab/BumpedMaterialBuilder**, **Script/BumpedMaterialBuilder**를 참고하세요.



디렉토리



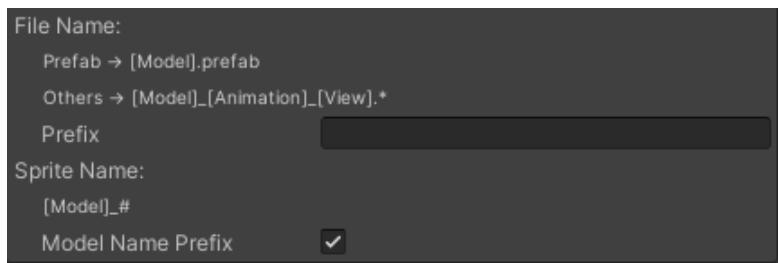
Export Path는 결과 파일들이 저장될 경로를 보여줍니다.

선택된 모델 혹은 선택된 애니메이션만 촬영하면, (모델명 + 날짜 + 시간)으로 된 폴더안에 파일들이 생성됩니다.

모델 리스트에 등록된 모든 모델들을 한꺼번에 촬영하면, (날짜 + 시간)으로 된 루트 폴더 안에 각 모델에 대한 폴더와 파일들이 생성됩니다.

Choose a directory under Assets 버튼을 클릭해서 폴더를 선택할 수 있습니다.

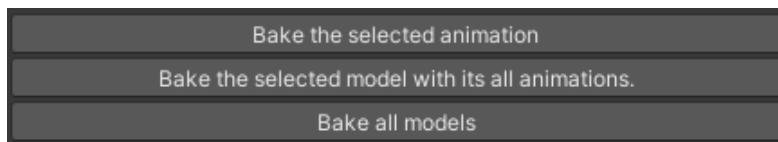
네이밍



File Name은 결과 파일 이름 방식을 보여줍니다. 아래의 Prefix를 입력하면 파일 이름 앞에 공통 문자열을 붙일 수 있습니다.

Sprite Name은 결과 스프라이트 이름 방식을 보여줍니다. Model Name Prefix를 켜면, 스프라이트 이름 앞에 모델 명이 붙습니다.

베이킹



선택된 애니메이션만, 선택된 모델만, 혹은 모델 리스트에 등록된 모든 모델을 동시에 베이킹할 수 있습니다.

선택된 모델이 메시 모델이고 애니메이션을 가지고 있을 경우에만 Bake the selected animation 버튼이 노출됩니다.

모델 리스트에 메시 모델과 파티클 모델이 섞여 들어가 있는 경우에는 Bake all models 버튼이 보이지 않습니다.

플레이 모드 베이킹 (베타 버전)

미러 애니메이션, 애니메이션 리깅 등을 지원하기 위한 플레이 모드 베이킹 기능이 추가되었습니다.

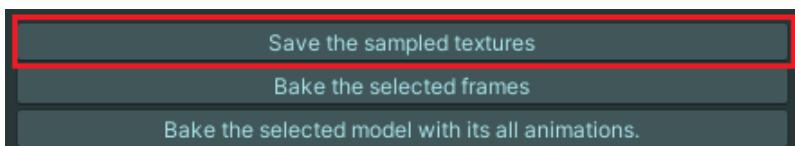
에디터 모드에서 애니메이션을 시뮬레이션하여 강제로 움직이게 한 모델을 촬영한 것과 달리 플레이 모드에서는 애니메이션에 의해 실제로 동작하는 모델을 촬영합니다. 그래서 몇몇 애니메이션은 에디터 모드와 플레이 모드에서 촬영한 모습이 다를 수 있습니다.

플레이 버튼  을 눌러 플레이 모드로 들어간 후 에디터 모드에서 한 것과 똑같이 하면 됩니다.

플레이 모드에서만 동작하는 기능

* [## 파티클 모델](#)의 Projectile

* [## 샘플링](#) 후 선택한 프레임을 그대로 이미지로 저장



플레이 모드에서 동작하지 않는 기능

* [## 메시 모델](#)의 Root Bone과 Customizer

* [## 프리뷰 창](#)

* [## 촬영](#)의 시뮬레이션

기타

* 플레이 모드에서 모델이 제 자리에 있도록, Apply Root Motion 등을 조정해 주세요.

* 베이킹 했을 때 다음과 같은 에러가 날 수 있습니다.

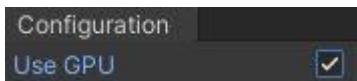
```
NullReferenceException: Object reference not set to an instance of an object
UnityEngine.UIElements.UIR.TextureBlitter.BeginBlit (UnityEngine.RenderTexture dst) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIR.TextureBlitter.BlitOneNow (UnityEngine.RenderTexture dst, UnityEngine.Texture src, UnityEngine.RectInt srcRect, UnityEngine.RectInt dstRect) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIRAtlasManager.UpdateAtlasTexture () (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIRAtlasManager.Commit () (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIR.RenderChain.Render (UnityEngine.Rect viewPort, UnityEngine.Matrix4x4 projection, UnityEngine.UIElements.UIRenderer render) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIRRepaintUpdater.DrawChain (UnityEngine.Rect viewPort, UnityEngine.Matrix4x4 projection) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIRRepaintUpdater.Update () (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.VisualTreeUpdater.UpdateVisualTreePhase (UnityEngine.UIElements.VisualTreeUpdatePhase phase) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.Panel.UpdateForRepaint () (at <cd855c76ab374957a4384f21004cd44a>:0)
```

Studio 인스펙터 창에 베이킹 프로그레스를 그리는 것이 플레이 모드에서 오류를 내는 것으로, 결과 파일들을 생성하는 데는 지장이 없으므로 무시해 주세요.

GPU 사용

더 빠르게 색상을 추출하고 잘라내기를 하기 위해서 GPU를 사용할 수 있습니다.

에셋 메뉴에서 Animation Baking Studio > Configuration를 클릭하면 Configuration 창이 등장하고, 여기서 Use GPU를 켜면 GPU가 사용됩니다.



GPU를 사용하는 방식은 아직 실험 버전이며 아래와 같은 제약 조건이 있습니다.

- * 잘라내기를 켠 경우, 여백이 0이어야 합니다.
- * 노말 맵은 제대로 생성되지 못합니다.

주의 사항

- * Object Baking Studio 폴더가 있다면, 파일이 중복될 수 있으니 삭제해 주세요.
- * HDRP/URP 프로젝트에서 Bloom 같은 Post Processing 효과를 사용한다면, Default Extractor의 Alpha Threshold를 0 이상으로 조정해보세요.
- * 매트 그림자는 아직 URP에서 지원되지 않습니다.

* URP에서 탑 다운 그림자를 사용할 때는, Game 창에 하나의 모델만 들어오도록 카메라 혹은 모델들의 위치를 조정해야 합니다.

* HDRP를 사용한다면 Main Camera의 Custom Frame Settings > Rendering > Post-process > Color Grading을 꺼야 합니다.

* 패키지 안에 Unity 5 버전의 Standard Assets이 포함되어 있습니다. 이미 Standard Assets이 존재한다면 임포트가 잘 되지 않을 수도 있습니다.

써 보시고 괜찮으면, 평점 부탁드립니다. <https://assetstore.unity.com/packages/slug/31247>

이메일: beggu84@naver.com