

# Sprite Baking Studio

(ver. 2.2.0)

## 용어 및 기호

「SBS」는 Sprite Baking Studio 의 약어 입니다.

「객체」는 오직 GameObject 만을 의미합니다.

「피봇」은 스프라이트의 'Custom Pivot' 으로 변환되는 3차원 상의 위치를 의미합니다.



「뷰」는 촬영할 장면, 즉 모델의 방향을 의미합니다.

「굽다」는 모델을 촬영해서 PNG 파일로 만드는 것 까지를 포함합니다.

「속성」은 SpriteBakingStudio 혹은 StudioModel 컴포넌트의 입력 박스, 체크 박스, 콤보 박스 같은 항목 입니다.

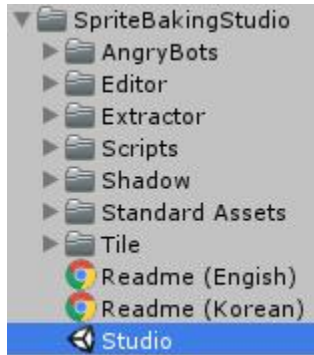
「※」, 「▼」, 「■」는 각각 속성, 열거형, 버튼에 대한 기호입니다.

' '(따옴표 쌍) 으로 묶인 단어들은 파일, 오브젝트, 속성, 열거형, 버튼 등의 이름입니다.

[ ](대괄호 쌍) 으로 묶인 단어들은 윈도우입니다.

## 빠르게 맛보기

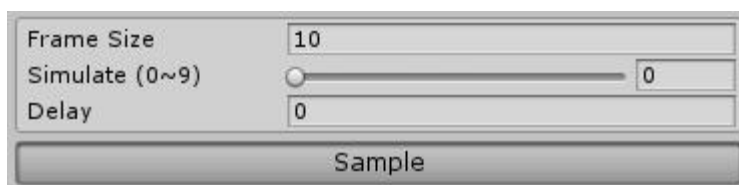
1. [Project] 윈도우에서 SpriteBakingStudio 폴더 안에 있는 Studio 씬을 로드하세요.



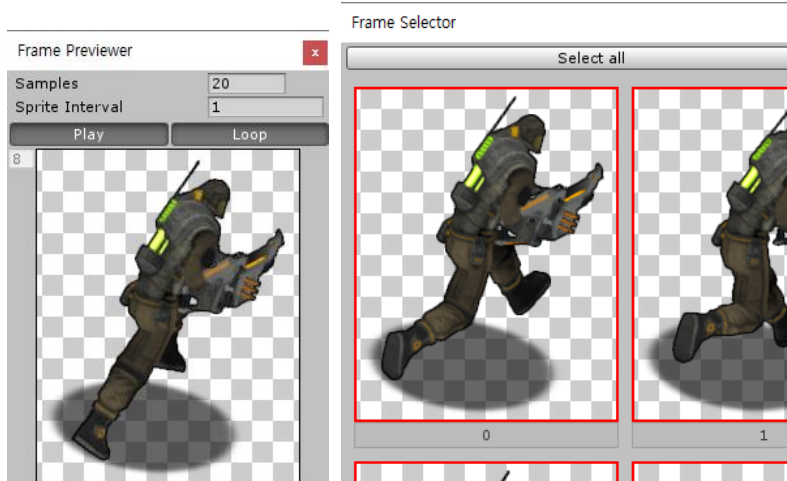
2. [Hierarchy] 윈도우에서 SpriteBakingStudio 객체를 선택하세요. 그러면 [Inspector] 윈도우에서 'Sprite Baking Studio' 컴포넌트가 표시됩니다.



3. 촬영을 위한 기본적인 준비가 되어 있으니, 중간에 있는 'Sample' 버튼을 클릭하세요.



4. 샘플링이 끝나면 두 개의 팝업창이 뜹니다. [Frame Selector] 윈도우에서 최종적으로 굿고자하는 프레임을 선택할 수 있고, [Frame Previewer] 윈도우에서는 선택된 프레임들이 연결된 동작을 확인할 수 있습니다.



기본적으로 모든 프레임이 선택되어 있으니 창을 닫습니다.

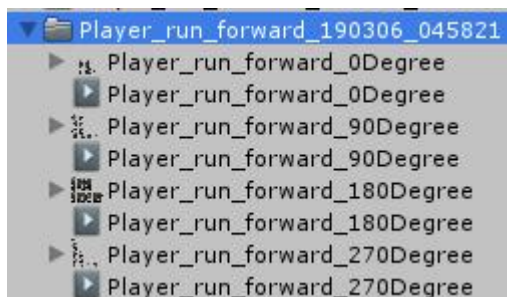
5. 제일 아래 쪽에 있는 'Choose Directory' 버튼을 클릭해 결과 파일들이 저장될 경로를 지정하세요.



6. 바로 아래 나타나는 'Bake selected frames' 버튼을 클릭하세요.

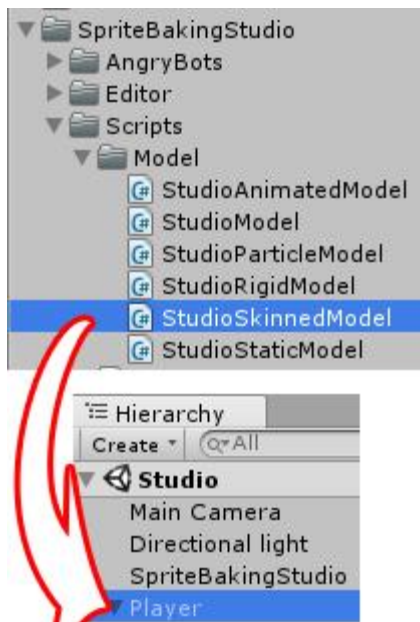


7. 베이킹이 끝난 후, 지정했던 경로에 결과 파일들(png and .anim)이 있는지 확인하세요.



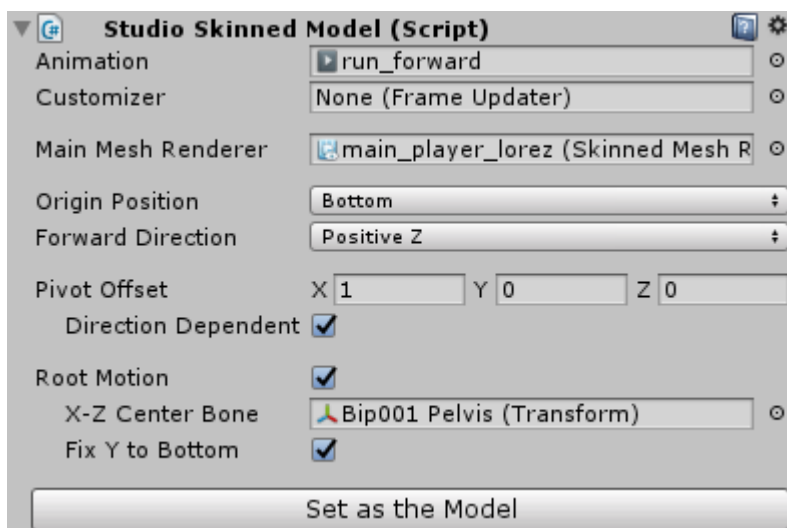
## 기능 설명 (모델 스크립트)

원하는 객체를 촬영하려면 적절한 모델 스크립트를 GameObject 에 컴포넌트로 추가해야 합니다.



### ◎ StudioSkinnedModel

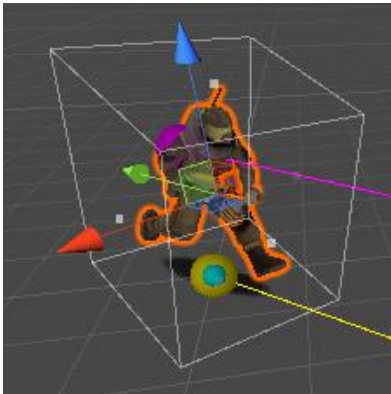
적어도 하나의 SkinnedMeshRenderer 를 지니고 있는 계층적인 뼈대 구조 객체를 위한 모델 스크립트.



※ Animation – 객체의 동작을 담고 있는 AnimationClip.

※ Customizer <없어도 무방> – 이 속성에 삽입된 FrameUpdater 인스턴스의 UpdateFrame 메서드는 매프레임 마다 호출됩니다.

※ Main Mesh Renderer <자동 선택> – 다양한 크기와 위치를 계산하기 위해, 객체 계층에서 가장 큰 바운즈를 가지고 있는 SkinnedMeshRenderer 를 지정해야 합니다.



※ Origin Type – 객체의 수직상의 중심 위치와 바닥 위치를 계산하기 위해, 열거형 항목 중 하나를 선택해서 transform.position 이 객체의 몸통 중 어느 부분에 위치하는 지를 명시해야 합니다. ('transform' 은 Component 클래스의 내장 변수입니다.)

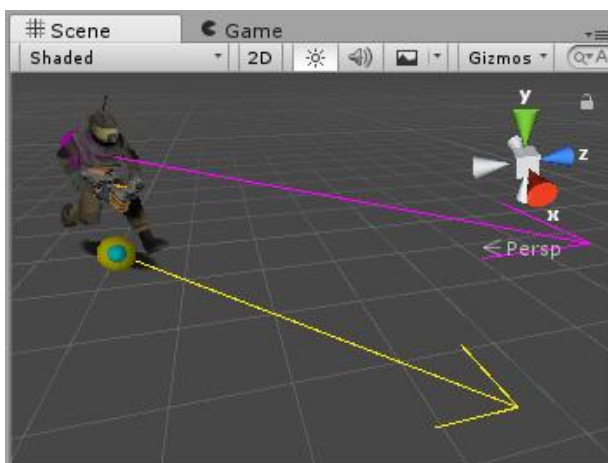
▼ Bottom <기본> – transform.position 이 객체의 바닥 위치인 경우로, 중앙 위치는 바닥 위치에서 주 메쉬 렌더러의 바운즈 y 길이의 절반만큼 떨어져 있는 곳이 됩니다.

▼ Center – transform.position 이 이미 객체의 수직상 중앙 위치인 경우입니다.

제대로 설정하였다면, [Scene] 윈도우에서 보았을 때 보라색 구가 객체의 중심 부근에 위치합니다. 참고로, 노란색 구는 transform.position 을, 자그마한 하늘색 구는 피봇을 의미합니다.

※ Forward Direction – 객체의 방향을 계산하기 위해, 열거형 항목 중 하나를 선택해서 객체의 앞쪽이 어느 방향인지 지정해야 합니다. Positive Z 가 기본입니다.

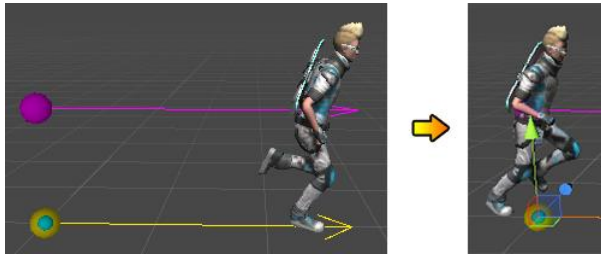
제대로 설정했다면, 화살표가 객체의 정면 방향을 향하게 됩니다.



※ Pivot Offset – 피봇은 객체의 바닥 위치에 이 벡터값을 더한 위치가 됩니다.

※ Direction Dependent – 객체의 정면 방향에 벡터값을 곱한 후 바닥 위치에 더합니다. 이 속성은 'Pivot Offset' 값의 크기가 0 보다 클 때만 나타납니다.

※ Root Motion – 이 속성을 키면, 애니메이션이 객체를 (0, 0, 0) 에서 벗어나거나, 몸통 뼈대 객체가 최상위 객체를 벗어나게 하는 것을 방지할 수 있습니다.



※ X-Z Center Bone – 척추, 엉덩이, 골반과 같이 수평상 객체의 중심이 되는 뼈대 객체를 지정해야 객체를 제자리에 고정시켜 놓을 수 있습니다.

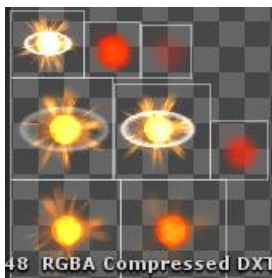


※ Fix Y to Bottom – 'Root Motion' 만 켜면 기본적으로 X-Z 평면 상에서만 고정시켜 줍니다. 이 속성까지 켜면 Y 축 상으로도 고정시켜 줍니다.

■ Set as the Model – 이 객체를 SpriteBakingStudio 컴포넌트의 'Model' 로 지정합니다.

## ◎ StudioParticleModel

적어도 하나의 ParticleSystem 를 지니고 있는 계층적인 파티클 객체를 위한 모델 스크립트. (레거시 파티클은 지원되지 않습니다.)

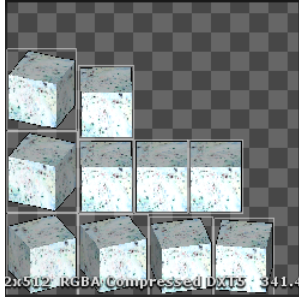


※ Main Particle System <자동 선택> – 계층에서 가장 오래 지속되는 ParticleSystem 을 지정해야 합니다.

※ Pivot Offset – 기본 피봇이 중앙이라는 것을 제외하고, StudioSkinnedModel 과 같습니다.

### ◎ StudioRigidModel

움직이는 강체 객체를 위한 모델 스크립트.



※ Mesh Renderer <자동 선택> – 다양한 크기와 위치를 계산하기 위해, 객체의 MeshRenderer 를 지정해야 합니다.

'Animation', 'Customizer' 은 StudioSkinnedModel 과 같습니다.

'Pivot Offset' 은 StudioParticleModel 과 같습니다.

### ◎ StudioStaticModel

움직이지 않는 객체를 위한 스트립트로, 한 프레임만 촬영됩니다.

'Mesh Renderer', 'Pivot Offset' 은 StudioRigidModel과 같습니다.

### ◎ StaticModelGroup

움직이지 않는 객체 그룹을 위한 스크립트로, 한 프레임씩 연속으로 촬영합니다.

※ Objects Root Directory – 객체들이 담겨있는 루트 폴더 경로

■ Refresh Sub Models – 클릭 시, 지정한 경로 아래의 모든 객체를 찾아서 StudioStaticModel 을 추가합니다.

※ 각 모델에 대한 체크 박스 – 선택한 모델만 굽습니다.

'Pivot Offset' 은 StudioRigidModel과 같습니다.

## 기능 설명 (SpriteBakingStudio)

객체를 촬영하려면 [Hierarchy] 윈도우에 Main Camera 가 반드시 존재해야 합니다. Main Camera 의 'Projection' 을 'Orthographic' 으로 설정하고 그 상태에서 'Size' 만 조정하기를 추천합니다.

※ Model – 모델 스크립트 중 하나를 부착한 객체를 지정해야 합니다.

※ Animation <참조> – 선택된 모델의 'Animation' 입니다. 이 속성은 움직이는 모델일 때만 나타납니다.

■ Refresh Sub Models – StaticModelGroup 컴포넌트에 있는 버튼과 같으며, StaticModelGroup 일 때만 나타납니다.

※ Camera Size <참조> – Main Camera 의 'Size'. Main Camera 의 'Projection' 이 'Orthogonal' 일 때만 보입니다.

■ Adjust Camera – [Game] 윈도우에 모델이 잘 보이도록 카메라 사이즈를 자동으로 조정해줍니다. 상황에 따라 보이지 않을 수도 있습니다.

※ Edge Detection <참조> – Main Camera 에 부착된 'Edge Detection' 컴포넌트 활성화 상태에 대한 참조.

※ Antialiasing <참조> - Main Camera 에 부착된 'Antialiasing' 컴포넌트 활성화 상태에 대한 참조.



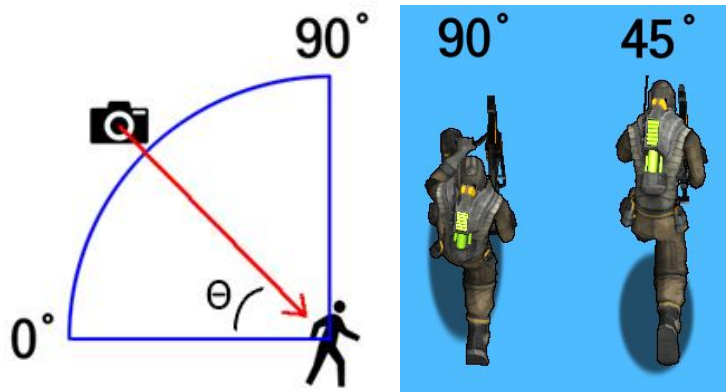
※ Main Light <자동 선택> – 촬영을 위한 Directional Light.

※ Follow Camera – 'Main Light' 가 Main Camera 와 항상 같은 위치에 있습니다.

※ Position – 'Main Light' 의 위치. 여기서 움직이면 항상 모델을 쳐다보게 됩니다.

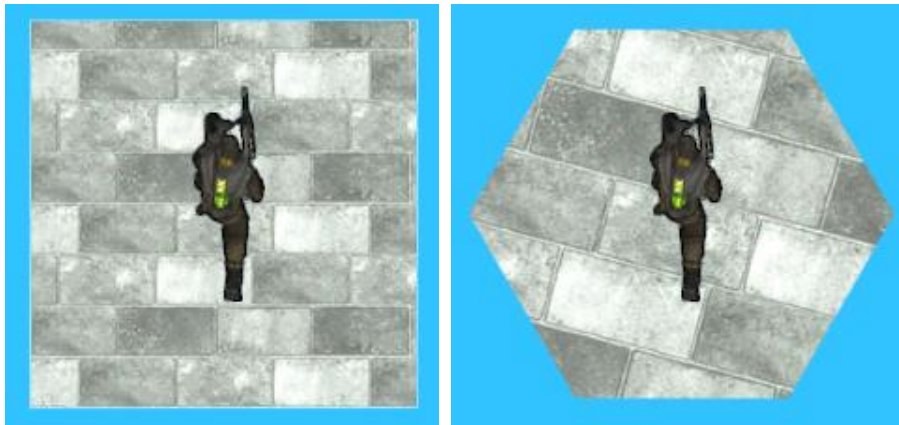


※ Slope Angle – 아래의 그림을 보세요.



※ Show Tile – 모델 아래쪽에 타일을 보여주지만, 촬영 시에는 사라집니다.

※ Tile Type – 두 종류의 타일이 있습니다.



※ Aspect Ratio – 이 비율 값에 따라 자동으로 경사각이 계산됩니다.



※ View Size – 아래의 초기 각도에서부터 시작해서 이전 각도에  $(360 / N)$  씩 반복적으로 더해서 이 값(N) 만큼의 뷰를 생성합니다.

※ Initial Degree – 초기 각도입니다. 범위는  $0 \sim (360 / N)$  입니다.

※ 각 뷰에 대한 체크 박스 – 선택한 뷰에 대해서만 최종적으로 굽습니다.

■ Apply – 모델을 해당 방향으로 회전합니다. 실제 구울때는 영향을 주지 않습니다.

■ Select All & Unselect All – 모든 뷰를 선택하거나 해지합니다.

※ Shadow Type

▼ None <기본> – 그림자 없음.

▼ Simple – 타원형 그림자.



※ Scale – 그림자 객체 크기 조절.

■ Unify – 스케일 비율을 같게 만듭니다.

※ Dynamic Size – 촬영 시 동적으로 그림자 객체의 크기를 조정합니다.

※ Opacity – 그림자 재질의 투명도입니다.

※ Shadow Only – 모델을 제외하고 그림자만 찍습니다. 'Variation' 의 'Exclude Shadow' 와 호환되지 않습니다.

▼ Real - 그림자 카메라가 모델을 촬영한 장면을 모델 아래 놓인 Texture 에 그림니다

※ Method – 현실적인 그림자를 만드는 두 가지 방법이 있습니다.

▼ Dynamic – Render Texture 에 실시간으로 그림니다. 파티클 모델은 이 방식을 사용할 수 없습니다.



※ Camera <자동 생성> – 그림자 전용 카메라.

※ Field <자동 생성> – 그림자 재질을 입힌 평면 객체.

'Opacity' 와 'Shadow Only' 는 'Simple' 과 비슷합니다.

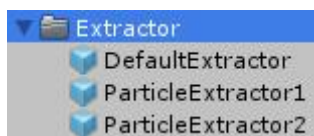
▼ Static – 'Dynamic' 방식과 비슷하지만, 매번 그림자 이미지를 생성합니다. 이 방식은 다소 느리지만, 파티클 시스템 모델처럼 'Dynamic' 방식이 사용 가능하지 않을 때 대안이 됩니다.



'Camera', 'Field', 'Opacity', 'Shadow Only' 속성은 'Dynamic' 방식과 비슷합니다.

■ Update & Hide – 'Update' 버튼을 클릭하면 모델을 촬영해서 그림자 이미지를 만듭니다. 'Hide' 버튼을 클릭하면 숨깁니다. 실제 촬영시에는 영향을 주지 않습니다.

※ Extractor – 배경으로부터 모델의 색상을 분리합니다. 우선 세 개의 Extractor 를 만들어 두었습니다.



기본적으로 DefaultExtractor 가 사용되며, 일반적인 불투명 모델에서는 이것을 바꿀 필요가 거의 없습니다.

파티클 모델은 반투명해서 정확히 원하는 색상을 추출하기가 까다롭습니다. ParticleExtractor1 은 모델의 계층 구조 상의 객체들의 셰이더들을 하나의 통일된 셰이더로 바꿔서 추출하는 방식이고, ParticleExtractor2 는 비슷하게 두 개의 통일된 셰이더를 사용하는 방식입니다.

ExtractorBase 를 상속받고 Extract 메서드를 구현해서 커스텀 Extractor 를 만들 수도 있습니다.

※ Variation – 결과 이미지의 색상과 투명도에 변화를 줄 수 있습니다.



※ Tint Color – 이미지의 원래 색상과 이 색상을 섞습니다.

▼ Tint & Sprite Blend Factor – 틴트 색상과 이미지 색상의 블렌딩 방식.

※ Exclude Shadow – 그림자는 틴트 색상 적용 대상에서 제외합니다. 이 속성은 그림자의 'Shadow Only' 와 호환되지 않습니다.



▼ Body & Shadow Blend Factor – 본체의 색상과 그림자 색상의 블렌딩 방식.

(참고: <https://www.khronos.org/registry/OpenGL-Refpages/es2.0/xhtml/glBlendFunc.xml>)

※ Preview – 이 속성이 켜져 있을 땐, [Preview] 윈도우를 보여줍니다.



아무 속성을 수정할 때마다 내부적으로 촬영하기 때문에 전반적으로 느려집니다. 그림자 타입이 'Static' 일 때는 특히 더 그렇습니다.

※ Background – [Preview] 윈도우의 배경 타입. 체크와 단일 색상이 있습니다.

■ Update Preview – [Preview] 윈도우를 수동으로 갱신합니다. 그림자를 갱신하는데 좋습니다.

※ Texture Resolution – 결과 스프라이트의 해상도.

※ Frame Size – 각 뷰 당 촬영할 프레임 수.

※ Simulate – 각 프레임에서의 모델의 모습을 볼 수 있습니다.

※ Delay – 정적 그림자 타입을 선택 후 샘플링하거나 구울 때, 촬영 속도와 그림자 이미지 만드는 속도 차이 때문에 스프라이트의 모델 부분과 그림자 부분이 맞지 않는 경우가 있습니다. (Mac에서 그랬습니다.) 그 경우에 이 값을 적절히 설정하면 문제를 해결할 수 있습니다.

■ Sample – 원하는 프레임만 선택하기 위해, 지정한 프레임 수만큼 모델 애니메이션을 촬영해 볼 수 있습니다.

스태틱 모델인 경우는 한 프레임만 촬영하기 때문에 특정 필드들이 보이지 않습니다.

※ Trim – 이 속성이 켜져있을 땐, 결과 스프라이트의 바깥 부분을 잘라냅니다.

※ Sprite Margin – 잘라낼 때 반드시 유지해야할 여분의 공간.

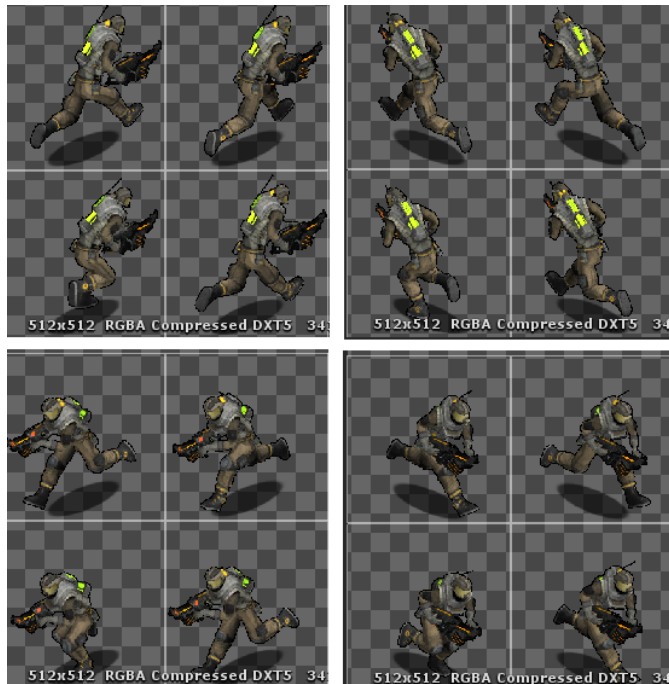
※ Unified Size – 스프라이트의 크기가 같게 잘라냅니다.



※ Pivot-Symmetric – 피봇을 중심으로 좌우 길이가 같게 잘라냅니다.



※ for All Views – 선택한 모든 뷰에서 촬영한 모든 스프라이트들의 크기가 같게 잘라냅니다.



※ Output Type – 두 개의 결과물 타입이 있습니다.

▼ Sprite Sheet – 각각 텍스처 좌표를 가지고 있는 스프라이트가 합쳐져 있는 PNG 파일을 만듭니다.

※ Packing Algorithm – 두 개의 패킹 알고리즘이 있습니다.

▼ Optimized – 유니티 자체의 알고리즘.

※ Max Size – 스프라이트 시트의 최대 크기.

▼ In Order – 좌측 상단부터 차례대로 스프라이트를 배치합니다.

※ Min Size – 스프라이트 시트의 최소 크기.

※ Padding – 각 스프라이트 사이의 간격.

※ Animation Clip – 스프라이트 시트를 만들 때 애니메이션 클립도 함께 만듭니다.

※ All In One Atlas – 모든 스프라이트를 하나의 스프라이트 시트에 넣습니다. 이 속성은 정적 모델 일 때만 나타납니다.

▼ Separate – 스프라이트 당 하나의 PNG 파일을 만듭니다.

※ Auto File Naming – 출력 파일들의 이름 앞 부분을 자동으로 작성합니다.

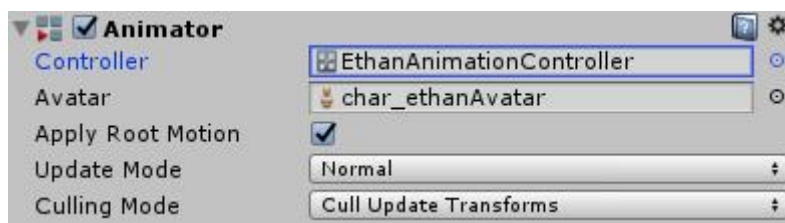
※ File Name – 출력 파일들의 앞 부분 이름.

※ Output Directory – 출력될 파일들이 저장될 경로. Asset 폴더 아래에 위치해야 합니다.

■ Bake all frames 혹은 Bake selected frames – 각 뷰 당 'Frame Size' 값만큼의 프레임을 촬영하거나 샘플링 후 선택한 프레임만 촬영합니다.

## 메카닉 모델 설정

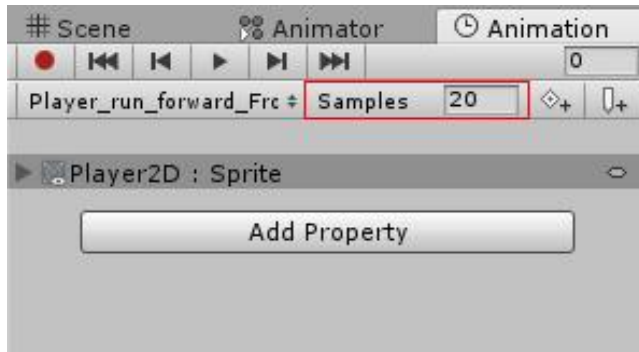
메카닉 모델인 경우에는 해당 객체에 Animator 컴포넌트가 있을 것이고, Controller 도 지정되어 있을 것입니다. 없다면 만들어서 넣어야 합니다.



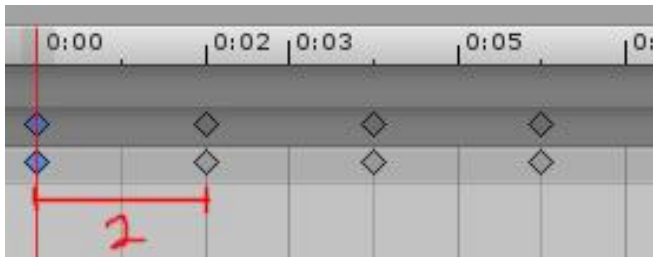
## 미리보기 윈도우



※ Samples – 생성된 AnimationClip 의 Samples 가 됩니다.

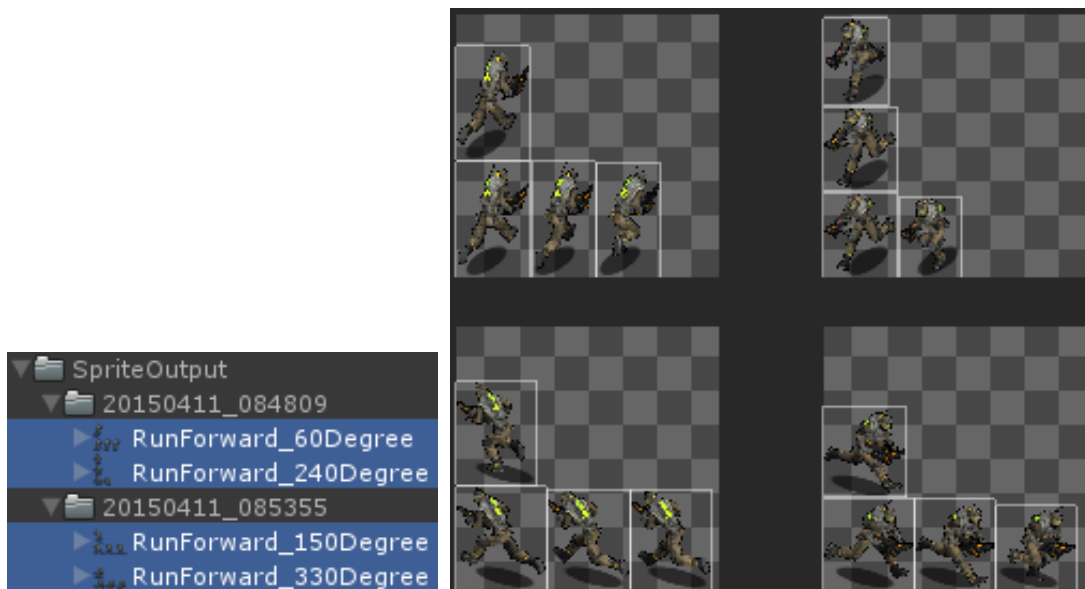


※ Sprite Interval – 생성된 AnimationClip 의 각 프레임들 사이의 시간 간격입니다.



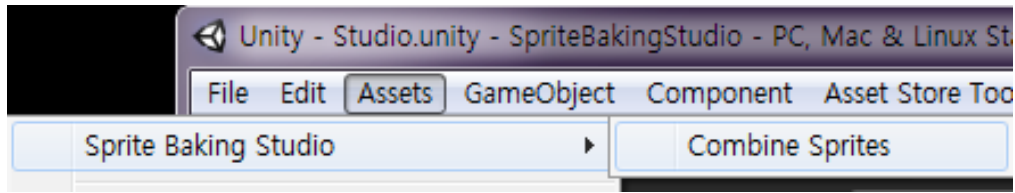
## 스프라이트 시트 합치기

1. 두 개 이상의 스프라이트 시트를 선택합니다.



2. 상단 메뉴에서 Assets > Sprite Baking Studio > Combine Sprites 혹은 [Project] 윈도우에서 오른쪽 마우스클릭 > Combine Sprites를 실행합니다.





그러면 선택된 스프라이트 시트들이 합쳐져서 첫 번째 스프라이트 시트의 위치에 저장됩니다.



## 경고 (발견된 문제들)

- \* 패키지 안에 Standard Assets이 포함되어 있습니다. 이미 [Project] 윈도우에 Standard Assets이 존재한다면 임포트가 잘 되지 않을 수도 있습니다.
- \* 만약에 [Inspector] 윈도우에서 폰트가 깨진다면 SpriteBakingStudioEditor 혹은 StudioOOOModelEditor 의 인코딩 방식을 UTF-8로 바꿔주세요.
- \* 빌드 세팅에서 플랫폼이 WebPlayer 로 되어있으면 StandAlone 으로 바꿔주세요.
- \* 카메라들(Main Camera 와 그림자 카메라) 의 Projection 이 Perspective 인 경우 기능이 제한될 수 있습니다.

## 연락처

Name: YoungWook Cho

Email: beggu84@naver.com