

Animation Baking Studio

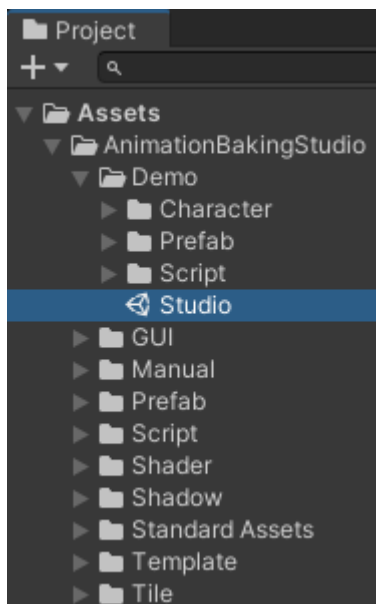
(ver. 4.0.0)

Animation Baking Studio is an asset that takes pictures of animating 3D objects creating sprite sheets (PNG), animation clips, etc.

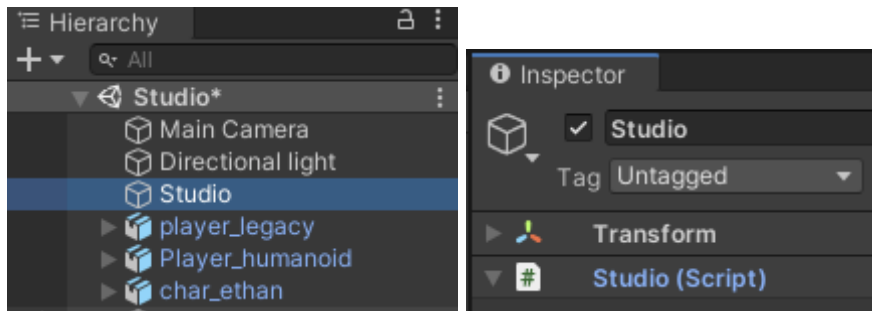
Quick Guide

You can take pictures if you have a studio, a main camera, and a model.

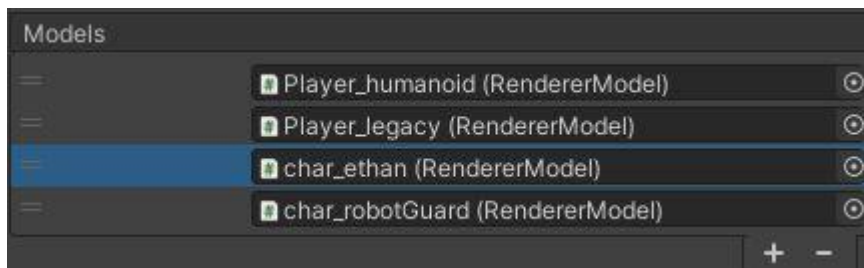
1. Open the enclosed Studio scene in the Project window.



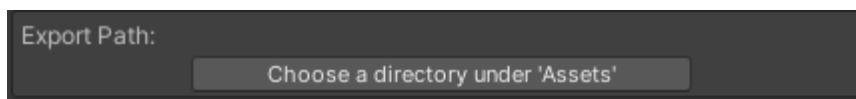
2. Select the Studio object in the Hierarchy window and see the **Studio** component in the Inspector window.



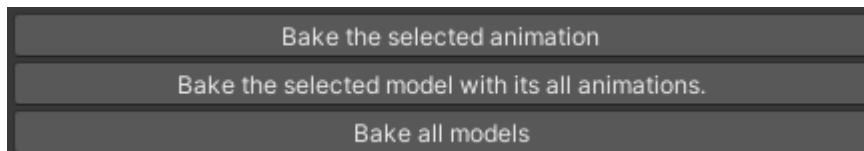
3. Select one model in the model list.



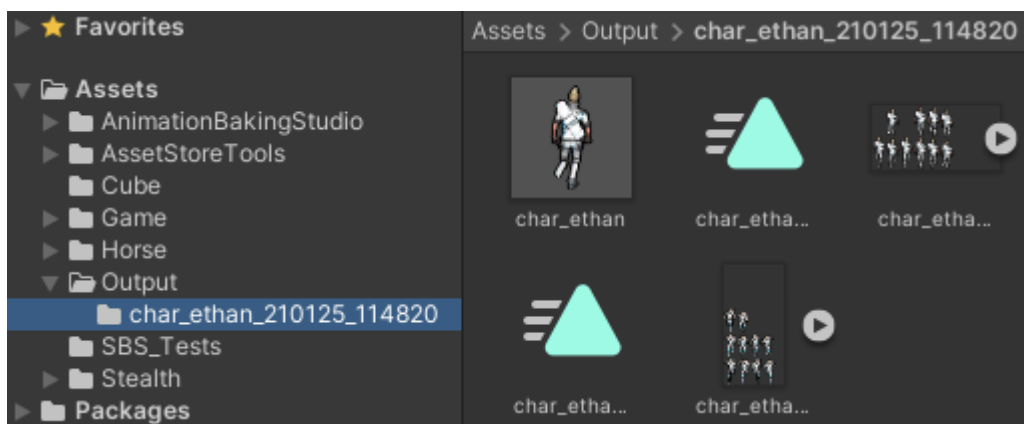
4. Click **Choose a directory under 'Assets'** button at the bottom to select a folder where you want to export output files.



5. Clicking one of the baking buttons starts baking.



6. Files such as sprite sheets, animation clips, etc. are generated in the selected folder.



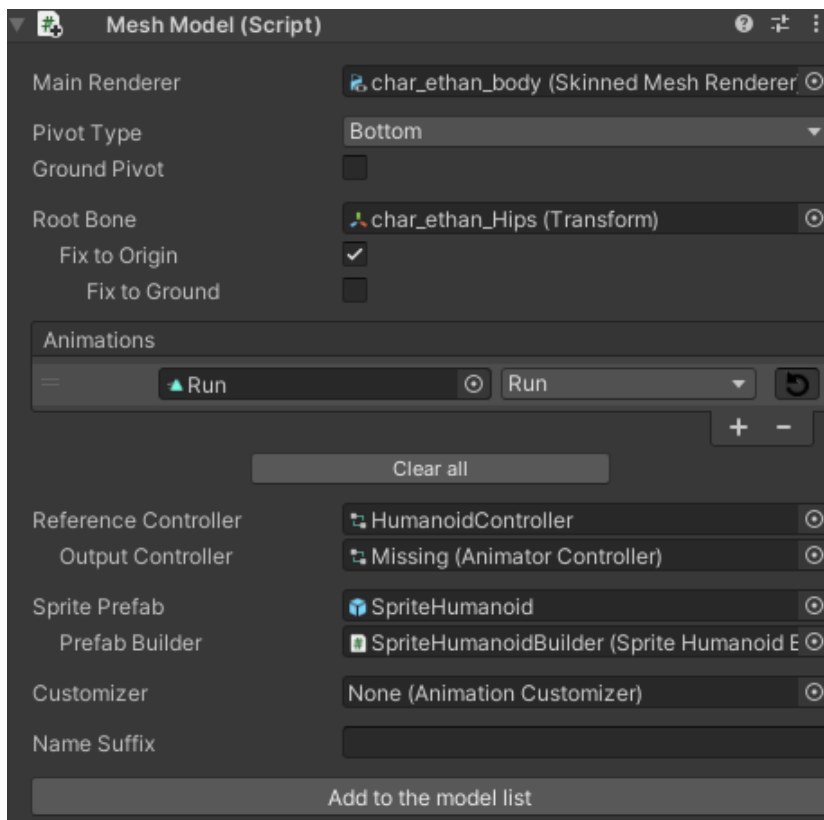
Model

Model is a target object to be taken pictures, and there are two types of models: Mesh Model and Particle Model.

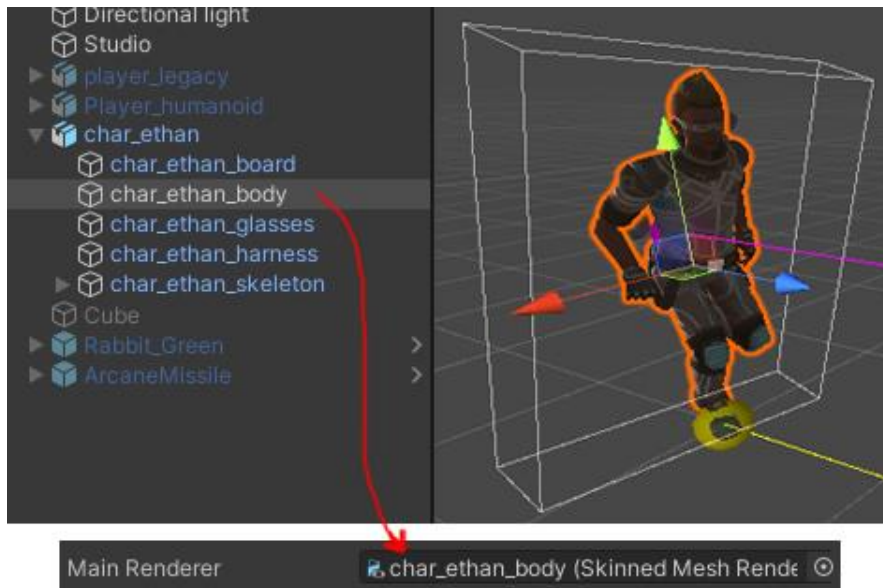
Mesh Model

To take pictures of an object that uses the Mechanim system, Avatar and Animator Controller should be specified in its Animator component.

Adding a **MeshModel** script to an object which has Mesh Renderer or Skinned Mesh Renderer makes it a mesh model.



Main Renderer is required to calculate the model's size. If there is another renderer within the object hierarchy that has a larger volume than that of a renderer automatically selected, it is recommended to designate it as the main renderer.



Usually, a hierarchical model's world position locates at around its bottom. That is, **Pivot Type** is **Bottom**.



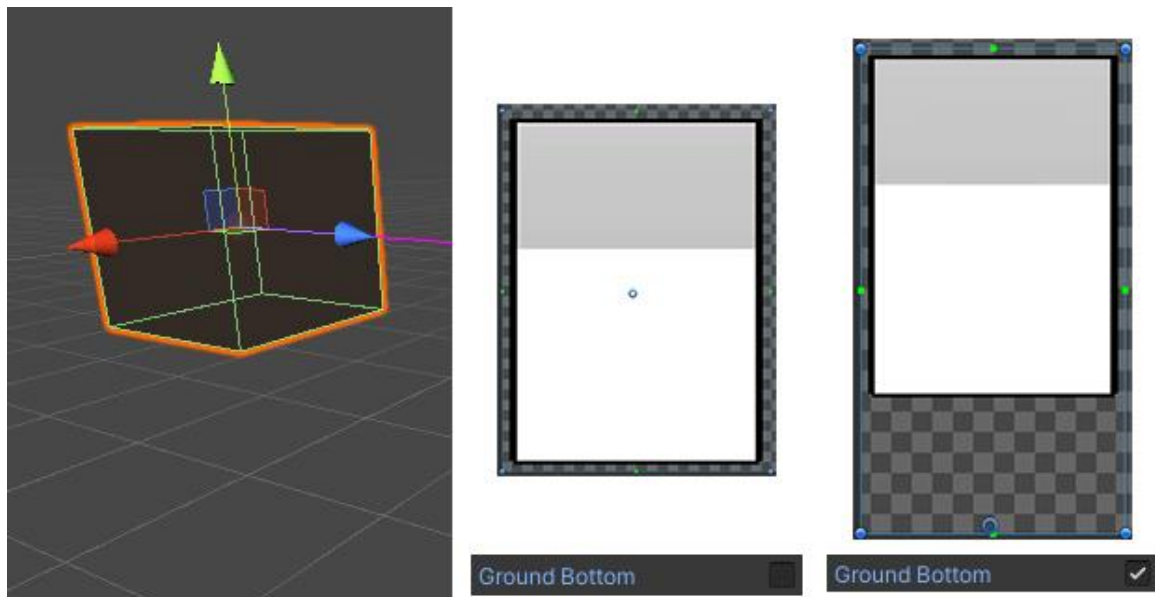
However, if the world position is at the model's vertical center, you should change it to **Center**.

As the world position of a non-hierarchical object is at the vertical center, **Pivot Type** is hidden and is internally set to **Center**.

If the object's front direction is not z-forward, you can create an empty object, place the rotated object under it, and add **MeshModel** script to the root object.

When a model which is floating from the ground is taken pictures with the trimming feature turned on, the output sprite will have all the area below the bottom of the model cut out.

Because turning on **Ground Pivot** makes the area from the ground to the model's bottom included in the model's area, you can prevent the part from cutting.



When taking pictures of an animating model, it is recommended to turn off Animator component's Apply Root Motion so that the animation does not dislodge the object out of place. Nevertheless, if the model is out of place when taking pictures, specify **Root Bone** and turn on **Fix to Origin**. A root bone object is likely to hip, pelvis, or spine.

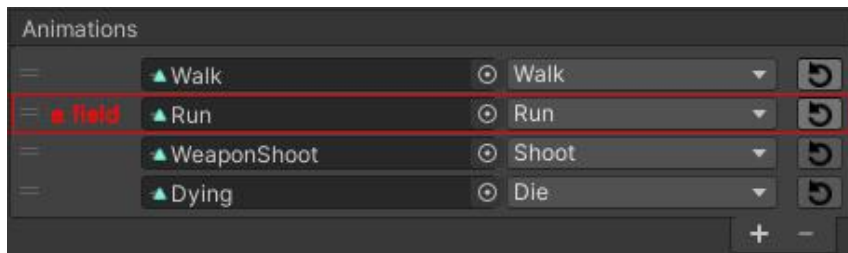


If both **Fix to Origin** and **Fix to Ground** are turned on, the model can be attached to the ground when the jump or fly animation is running.

Root Bone and its child items are not visible for non-hierarchical models.

To take pictures of animations, they should be registered in **Animations**. You can add a field by

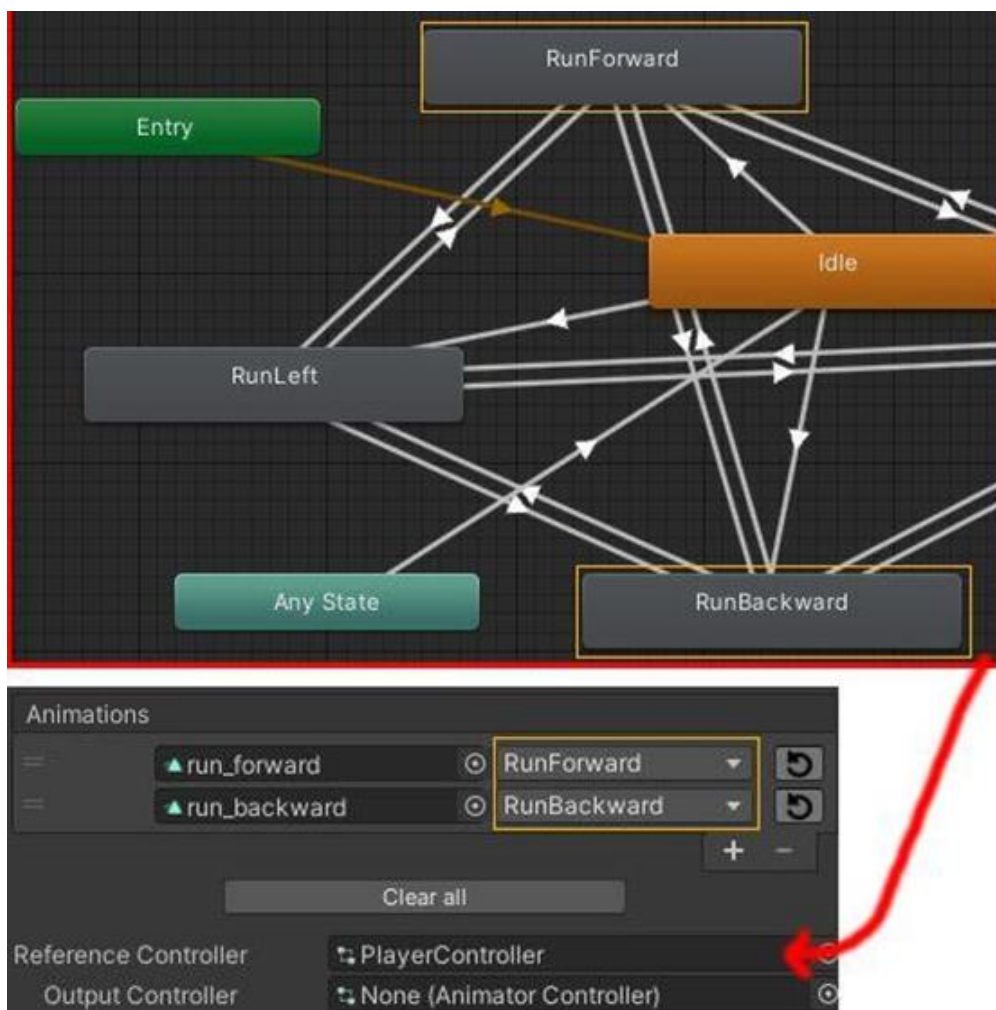
clicking '+' button and then put an animation clip into the field, or you can drag and drop an animation clip into the list to add a field.



If  of each field is turned on, a looping animation clip can be created.

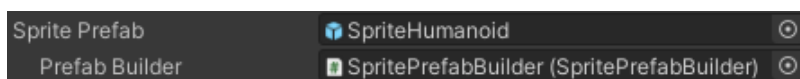
Reference Controller is needed as a reference when constructing the output controller.

If there are some animation states in the specified reference controller, a combo box appears in each field, where you can choose one of the states, which is corresponding to the animation clip in the field.



During baking, states and transitions are added to the controller you designate as **Output Controller**. If empty, a new controller is created and used. (Controllers with more than one layer or with any sub-machine can be used as the reference controller.)

Sprite Prefab is needed to create an output prefab object. Additionally, **Prefab Builder** should be assigned to connect output sprites and the output animator controller to the generated prefab object.



To create a prefab builder, you must create a new script to inherit **PrefabBuilder** class and implement required methods.

You can transform model objects during capturing, if you add a script that inherits **AnimatorCustomizer** and implements **UpdateFrame** function to a prefab object, and then put it in **Customizer**.

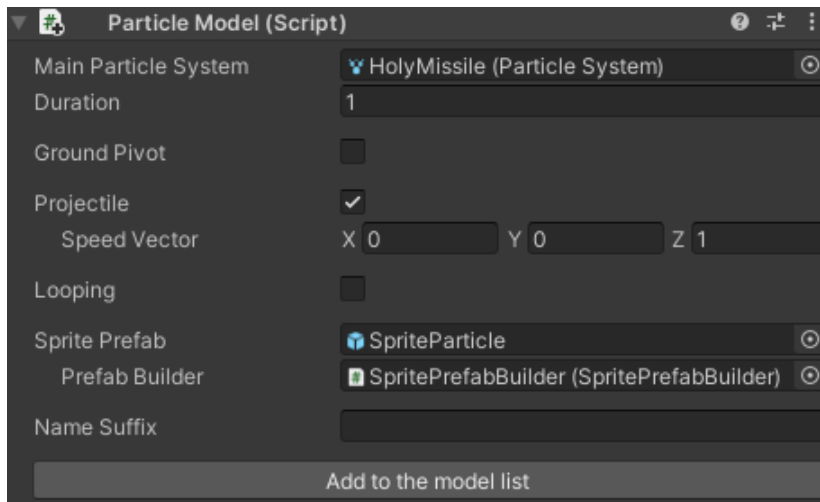
To add a specific word to the model's name, which is used for naming output files, enter it in **Name Suffix**.

Clicking '**Add to the model list**' button registers the model in the model list of the studio.

*If you select a folder in the Project window and run 'Animation Baking Studio > Instantiate Objects as > Mesh Model' in the Assets menu or in the context menu, you can instantiate all objects under the selected folder into the scene at once and add a **MeshModel** script into each created object.*

Particle Model

Adding a **ParticleModel** script to an object which has at least one Particle System makes it a particle model.



Main Particle System is required to calculate rough size of the object and to play the whole particle systems. If there is any major particle system in the object's hierarchy than the auto-selected one, it is recommended to replace it.

Duration is a single play time. Initially, it automatically uses the Duration value of the main particle system but can be set to a time too long to take pictures. So, it's better to set the proper time yourself by taking a few samplings. For more information about it, see [## Sampling](#) part.

To take pictures of a floating model with the trimming feature turned on, consider turning on **Ground Pivot**. For more information about it, see [## Mesh Model](#) part.

If you turn on **Projectile**, you can take a picture of moving as much as the Speed Vector at every update. This is a function only valid for play mode baking, see [### Play Mode Baking](#) part for details.

If **Looping** is turned on, a looping animation clip can be created.

Sprite Prefab and **Prefab Builder** must be specified to generate prefabs. For more information about it, see [## Mesh Model](#) part.

To add a specific word to the model's name, enter it in **Name Suffix**.

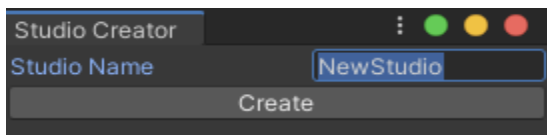
Clicking '**Add to the model list**' button registers model in the model list of the studio.

If you select a folder in the Project window and run 'Animation Baking Studio > Instantiate Objects as > Particle Model' in the Assets menu or in the context menu, you can instantiate all objects under the selected folder into the scene at once and add a [ParticleModel](#) script into each created object.

Studio

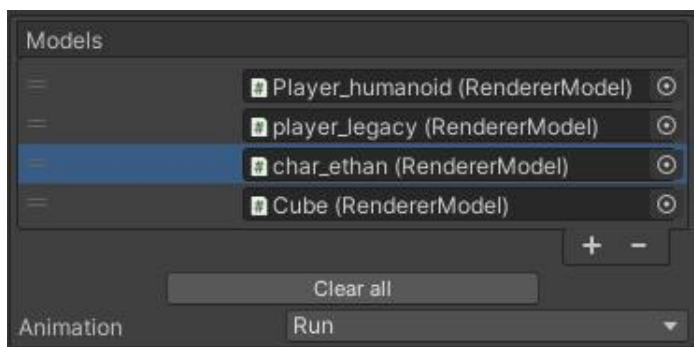
To take pictures of models, a studio is required. Create an empty object and add a **Studio** script on it.

You can also use 'Animation Baking Studio > Studio Creator' in the Assets menu or in the context menu in the Project window to create a scene where a studio object is set up.



*In the Studio Creator window, naming a studio and then clicking **Create** button generates a new scene and open it.*

Model list & Animation



At first, models should be registered in the model list. You can add a field by clicking **+** button and put a model into it or drag-and-drop it. Also, you can also register a model by clicking **'Add to the model list'** button in each (Mesh/Particle) Model component, as explained in [# Model](#) part.

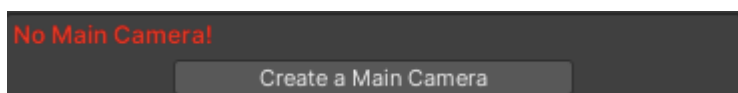
When a model in the model list is selected, some items of the studio could be changed according to it.

When a mesh model in which has animations is selected, **Animation** combo box appears where you can select one of the animations registered for the model.

Main Camera



A main camera – a camera with the MainCamera tag - is required for capturing. If there is a main camera, studio says **"Main Camera Exists."** Otherwise, a button appears to create a Main Camera object with a red statement that no main camera exists.



One of **Orthographic** projection and **Perspective** projection can be selected, and **Orthographic Size** and **Field Of View** appear, respectively.

The distance between the main camera and a model can be adjusted by choosing **Relative Distance** or **Absolute Distance**. The main camera is set far from a model by (the model's size X **Distance**) when the former is selected, and by just the distance for the latter.

In the case of the orthographic projection, you can't see any difference in the Game window, when you tweak the distance.

When a particle model is selected, the projection type toolbar is disabled and set to the orthographic projection internally. Also, the distance type toolbar is hidden and the distance between a model and the main camera is set far enough apart.

Directional Light

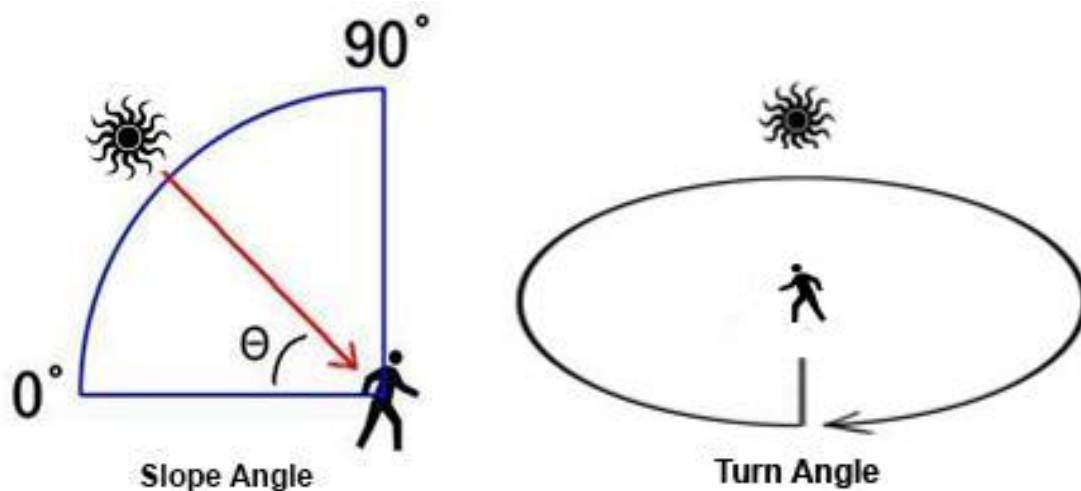


Directional Light is not essential for capturing.

To minimize the shadow cast on models when the main camera rotates, It's a good idea to turn on **Follow Camera Rotation** so that the directional lights are also looking in the same direction as the main camera. Or you can directly rotate the directional light by adjusting **Slope Angle** and

Turn Angle that appear when **Follow Camera Rotation** is turned off.

Directional Light	Directional light (Light)
Slope Angle (10 ~ 90)	70
Turn Angle	0
<button>Look at Model</button>	



Clicking **Look At Model** button rotates the directional light in the direction to the target model, and both angles are automatically modified.

If you want the directional light to move to the same position as the main camera when it is moving, turn on **Follow Camera Position**. But as it is a 'directional' light, the position of the light is not very important.

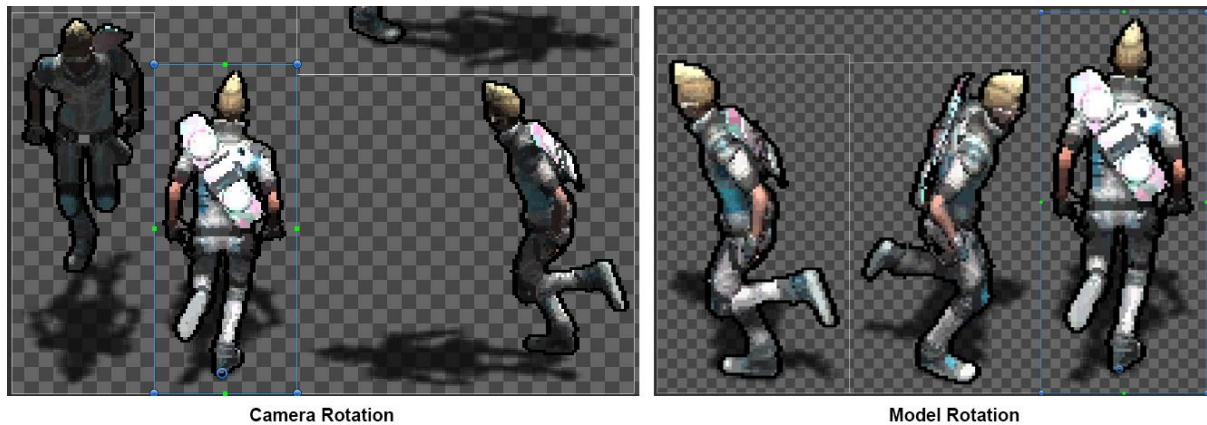
View

Camera Rotation		Model Rotation	
View Slope Angle (0 ~ 90)	30		
Show Reference Tile	<input checked="" type="checkbox"/>		
Tile Type	Square		
Aspect Ratio	X 2 Y 1		
View Size	4		
Base Angle (0 ~ 90)	0		
0°	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<button>Apply</button>
90°	<input type="checkbox"/>	<input type="checkbox"/>	<button>Apply</button>
180°	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<button>Apply</button>
270°	<input type="checkbox"/>	<input type="checkbox"/>	<button>Apply</button>
<button>Select all</button>		<button>Clear all</button>	

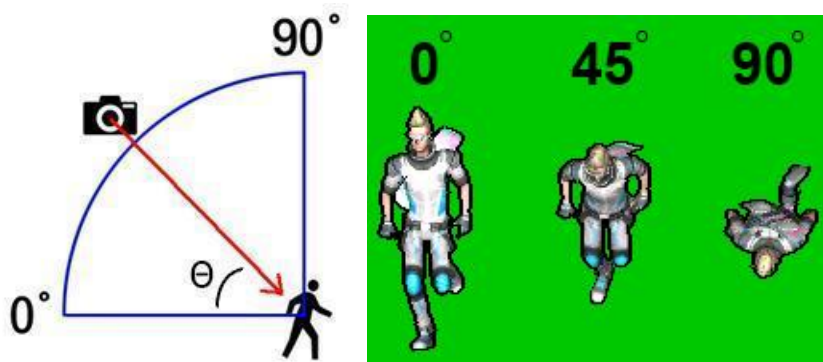
A view can be said to be a viewing direction, and there could be a lot of views according to the

main camera's direction and a model's direction.

In case of **Camera Rotation**, the main camera rotates around the selected model, and in case of **Model Rotation**, models rotate in place. Usually, the results of the two rotation methods are not different, but it could be different when **Matte** shadow is selected.



You can rotate the main camera around x-axis by adjusting **View Slope Angle**. When it is set to 0 degree, it is front view or side view, and when it is set to 90 degrees, it is top view.



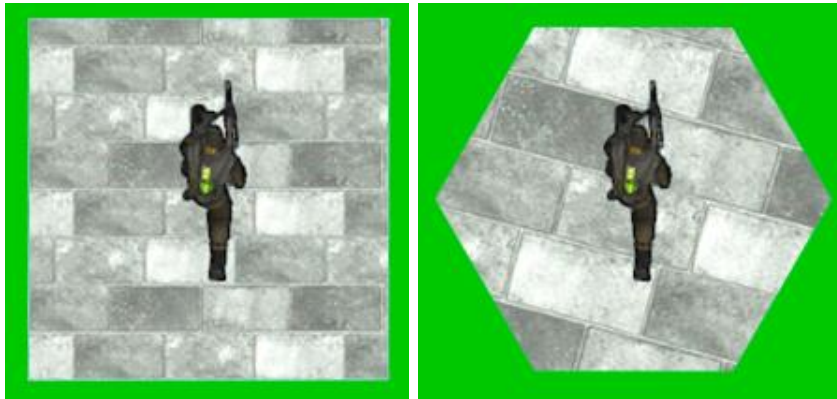
If you plan to make tile-based 2.5D game, characters would be drawn according to the tile's ratio.



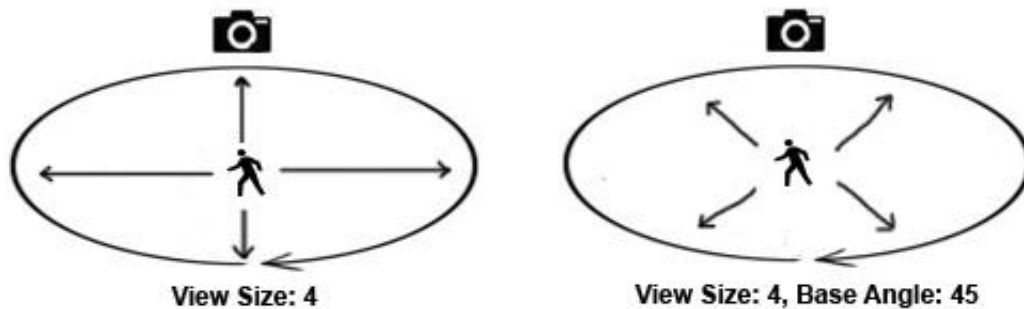
When **Show Reference Tile** is turned on, a tile appears, which disappears during capturing.

When you edit **Aspect Ratio**, the slope angle is auto calculated with the main camera's rotating. Conversely, editing the slope angle affects **Aspect Ratio**.

You can choose **Square** tile or **Hexagon** tile.

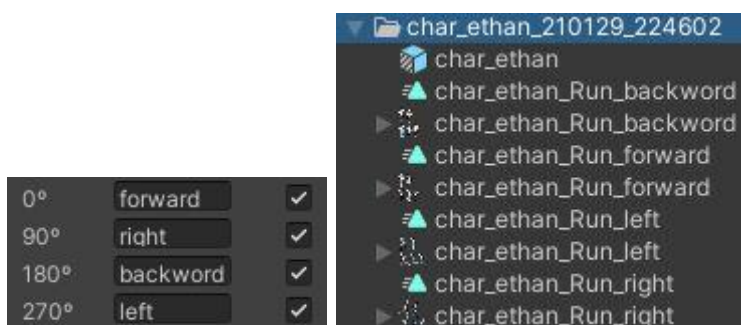


As many views as $\lceil 360 \text{ degree} / \text{View Size} \rceil$ are made and each view has a $\lceil \text{Base Angle} + (360 \text{ degree} / \text{View Size}) + \text{index} \rceil$ turn angle.



Click **Apply** button in each view rotates the main camera or models according to its direction.

Views turned on are continuously taken pictures, and the turn angle string is attached behind generated files' name. If you name each view, it is used instead of the turn angle string.

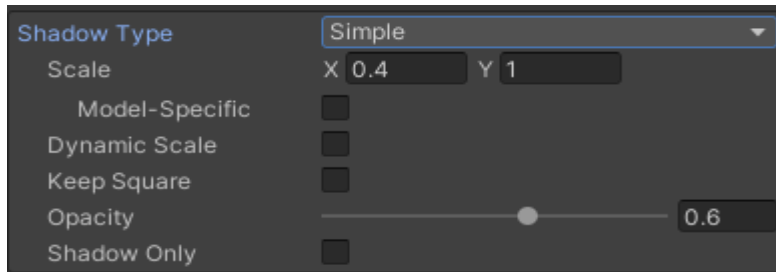


Shadow

There are three types of shadow: Simple Shadow, Top Shadow, and Matte Shadow.

(Shadows are not supported for particle models.)

Simple Shadow



It is oval-shaped simple shadow.



If you adjust **Scale**, the shadow field object is scaled horizontally or vertically.

The same scale value is applied to all models in the model list. If you want to apply a specific scale for each model, turn on **Model-Specific** and adjust the individual scale.

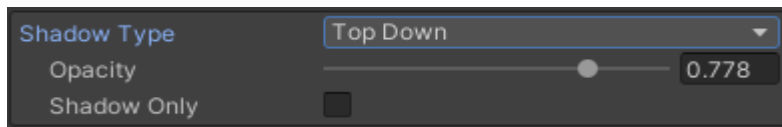
If you want to keep the shadow's size corresponding to the target model's size while it animates, turn on **Dynamic Scale**.

If you want to keep the aspect ratio uniformly, turn on **Keep Square**.

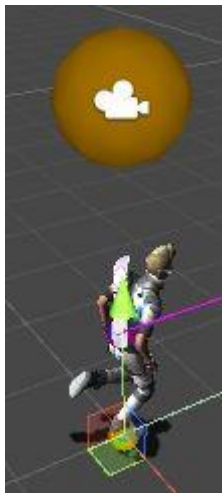
Opacity is the field that is associated with the main color of '~/Shadow/SimpleShadow' material with which you can adjust the transparency of the shadow. Also, you can create a different style shadow by modifying the texture registered on the material.

If you turn on **Shadow Only**, you can hide the model and only take the shadow.

Top-Down Shadow



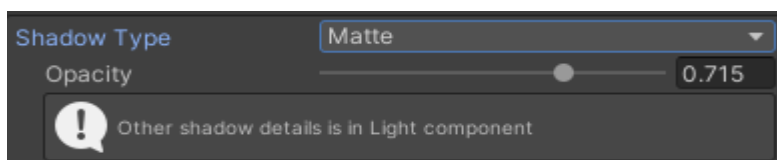
It is a shadow using the shadow camera that looks down the model vertically.



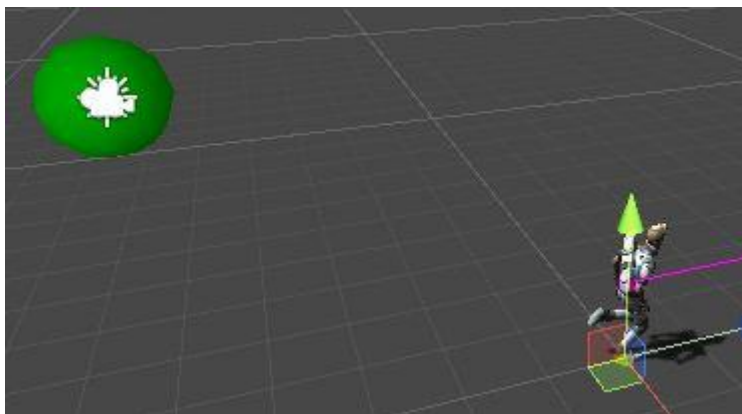
You can change the transparency by adjusting **Opacity**.

If you turn on **Shadow Only**, you can hide the model and only take the shadow.

Matte Shadow

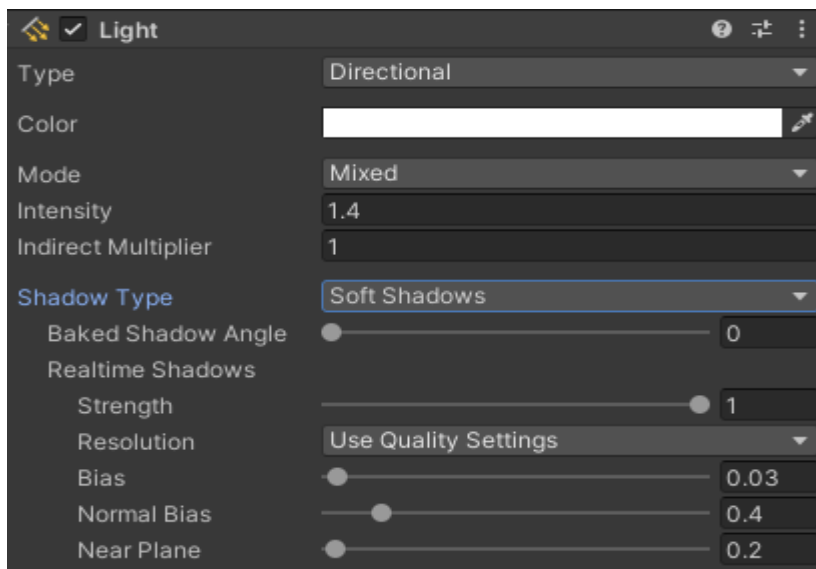


It is a shadow using a special transparent floor, which is the term used in 3ds Max or Maya.

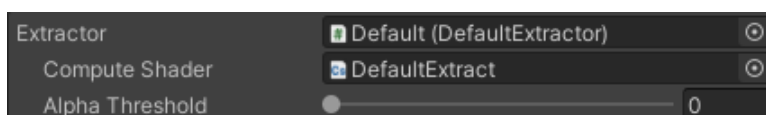


If you turn on **Follow Camera Rotation** of the light properties and select this shadow type, you will not be able to see any shadow in the Game window because the model completely blocks the sight.

You can easily change the transparency by adjusting **Opacity** and make detailed changes in the directional light object's Light component.



Pixel Extraction

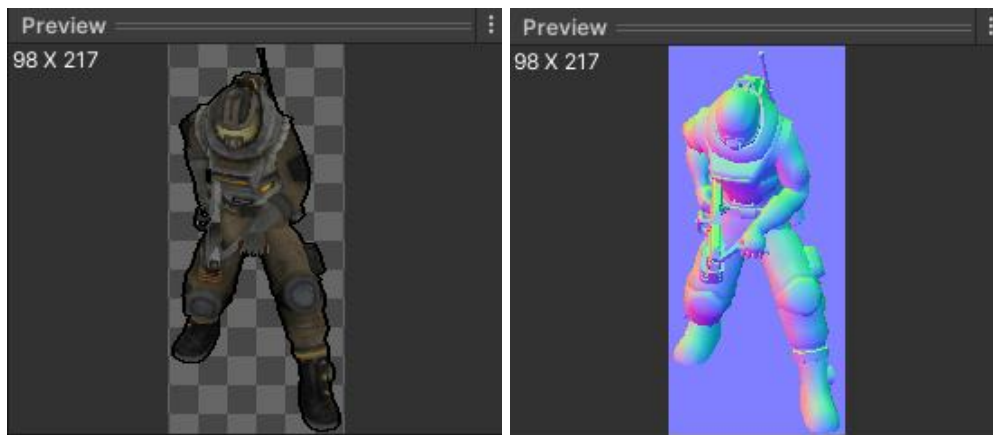
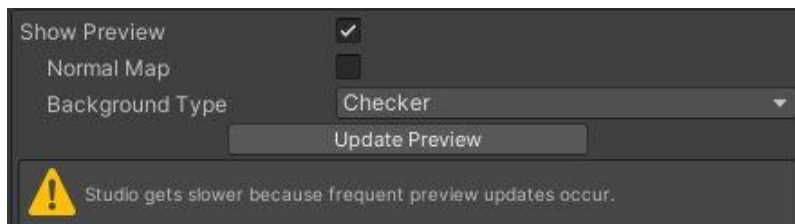


Extractor has a role that extracts the model's colors except background from the temporary image created by shooting the Game window.

By default, it is specified as '~/Prefab/Extractor/Default', and you can create new extractors by inheriting from **Extractor** abstract class and implementing **Extract** function.

If the target model is complete opaque and any anti-aliasing feature is not used, you can improve baking performance by replacing the Extractor with '~/Prefab/Extractor/Opaque'.

Preview Window



The Preview window appears at the bottom of the Studio component, when **Show Preview** is turned on, and it shows the result image in advance.

If [## Trimming](#) is turned on, the trimmed image size is reflected.

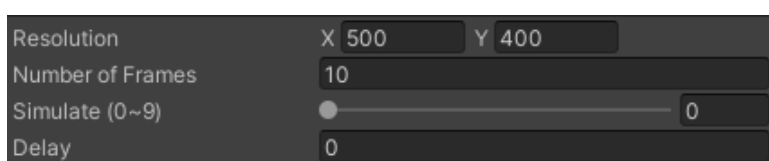
If **Make Normal Map** of [## Outputs](#) is turned on, **Normal Map** appears, and when you turn it on, you can see the normal map result image.

Although **Background Type** is defaulting **Checker**, and you can also select **Single Color** to fill the background with a single color.

If this feature is turned on, whenever any attribute in the studio changes, the model is taken pictures and a preview texture is created to slow it down.

If the preview does not update immediately, click **Update Preview** button.

Filming



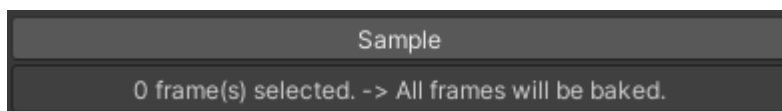
Resolution is a raw texture size before [## Trimming](#) is applied.

By default, an animation is captured per (the animation's length / **Frame Size**) seconds as much as **Frame Size** starting from zero.

You can see the motion of the animation at a specific frame by using **Simulate** slider.

When you use the top-down shadow, some mismatch between the model and the shadow could occurs. In that case, you can raise **Delay** a little to synchronize them.

Sampling

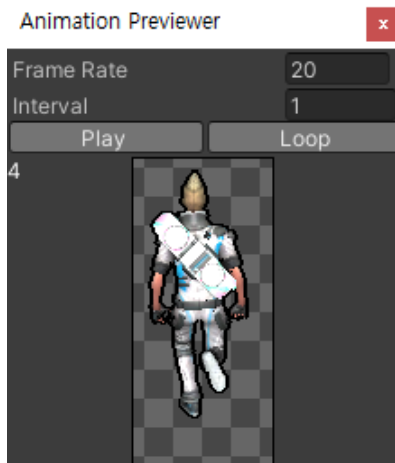


You can sample the selected animation by clicking **Sample** button. When sampling is complete, both the Frame Selector window and the Animation Preview window appear.

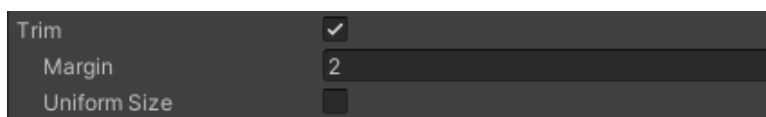
In the Frame Selector window, you can choose proper frames to take during baking. The images are cut fitting the model, but it does not affect whether the trimming occurs or not during baking.



You can see the animation of the selected frames in the Animation Previewer. Adjusting **Frame Rate** and **Interval** controls the animation speed and applies to output animation clips.



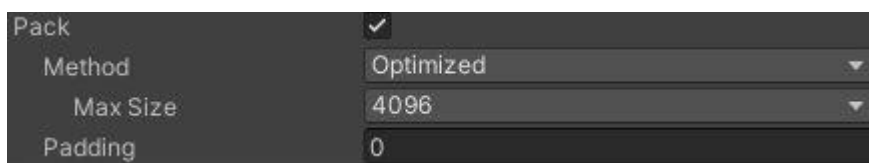
Trimming



If **Trim** is turned on, the output image is cut larger by **Margin** than the size that fits the model.

Turning on **Uniform Size** allows you to crop the image equally for all frames taken within a single view.

Packing



If **Pack** is not turned on, individual image files are created, and when turned on, sprite sheets are created.

There are two packing methods:

Optimized method is owned in Unity Engine and places sprites properly to create the smallest sprite sheet. The smallest sprite sheets are created, the size of which is not bigger than **Max Size**.

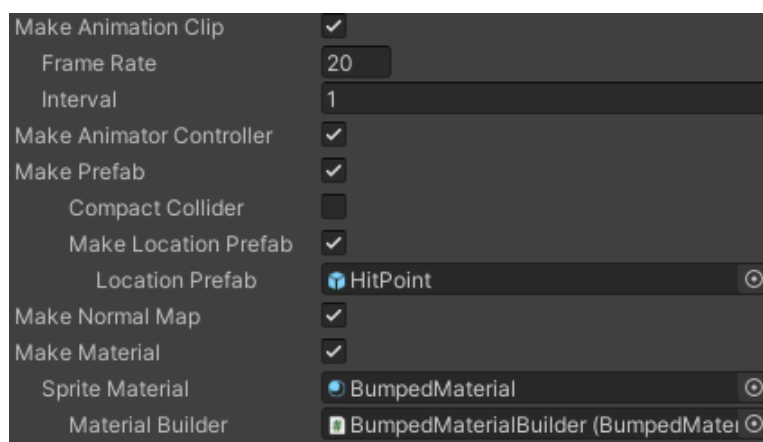
In Order method places sprites in order from the top-left to right-bottom. The largest sprite

sheets are created, which they totally embrace all sprites, and its size is not smaller than **Min Size**.

You can give as much space as **Padding** between sprites.

Outputs

In addition to image files, you can create animation clips, animator controllers, prefabs, normal maps, materials, etc., which are referenced together and ready to be used.



If **Make Animation Clip** is turned on, you can create animation clips associated with sprites, materials, etc. that are generated during baking.

This item is only exposed when [## Packing](#) is turned on.

If a target model has a reference controller and **Make Animator Controller** is turned on, an animator controller is created, in which animation clips are registered to each animation states. For more information, refer to [## Mesh Model](#) part.

This item is only exposed when [## Packing](#) is turned on.

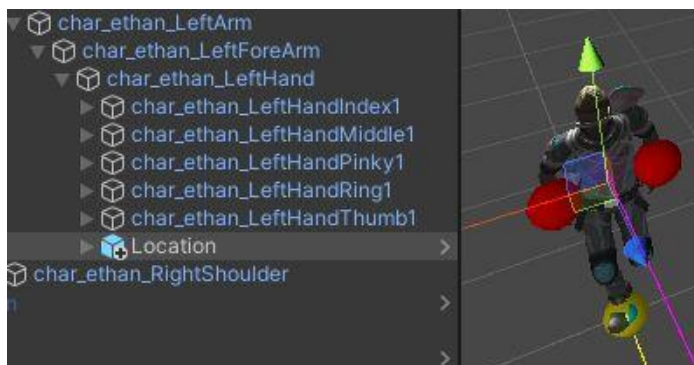
If **Make Prefab** is turned on, an object that is specified as the sprite prefab of each model is instantiated per model during baking. For more information, refer to [## Mesh Model](#) part.

When there is a BoxCollider2D component inside the specified sprite prefab, if **Compact Collider** is turned on, the collider size and offset are calculated to perfectly fit to the model's size even if

the trimming margin is larger than 0.



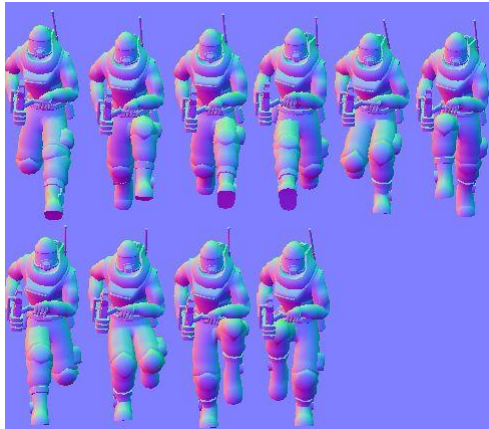
If you may want to track the location of specific bone objects within the output prefab object, at first, add a '~/Prefab/Location' object to the target object and rename it.



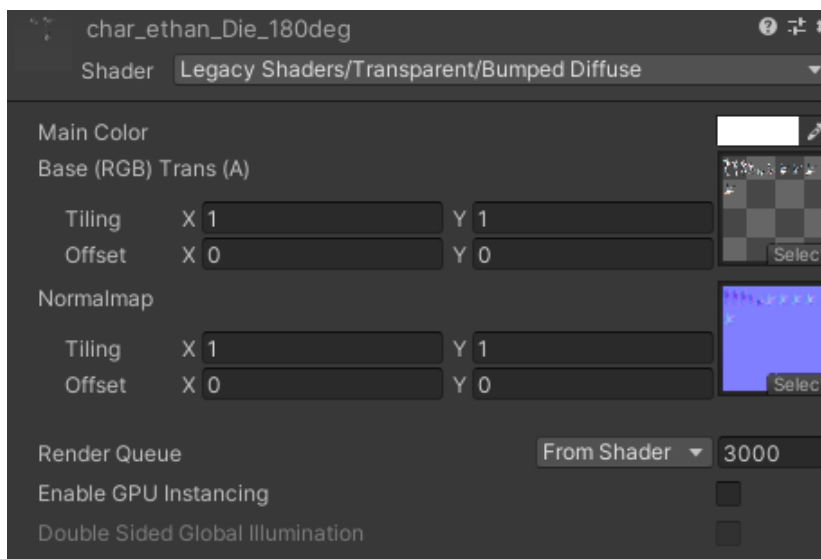
Then check **Make Location Prefab**, assign an output location object to **Location Prefab**, and you will see location objects in the output prefab object after baking. And that location object moves along with the output animation.



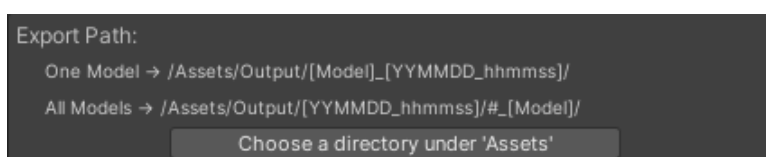
If **Make Normal Map** is turned on, normal map files corresponding to each model image file are created. To make valid normal maps, the model rotation type of [## View](#) should be selected.



Typically, Sprite Renderer uses Default-Sprite material, but there are cases when you want to designate a specific material. (e.g., when using Bumped Diffuse material with a normal map applied to it.) In this case, turn on **Make Material** and assign the material to be created to **Sprite Material. Material Builder** is also needed to connect output model images and normal map images to the output material. See 'Material/BumpedMaterial', 'Prefab/BumpedMaterialBuilder' and 'Script/BumpedMaterialBuilder' in Demo folder.



Directory



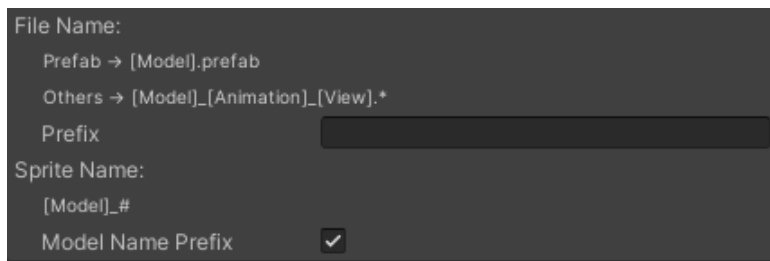
Export Path shows pathes where output files will be saved.

When you take pictures of the selected model or the selected animation, result files are created in the folder whose name is (the model's name + the date + the time)

When you take pictures of all models in the model list at once, folders and files are created for each model in the root folder whose name is (the date + the time).

You can choose the folder by clicking **Choose a directory under 'Assets'** button.

Naming

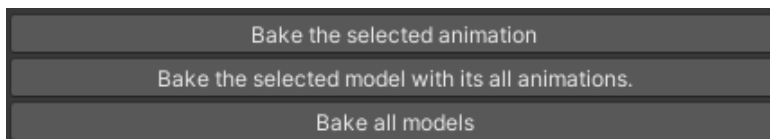


The screenshot shows a dark-themed panel for naming output files. It contains the following elements: 'File Name:' with two options, 'Prefab → [Model].prefab' and 'Others → [Model]_[Animation]_[View].*', a 'Prefix' text input field, 'Sprite Name:' with the value '[Model]_#', and a 'Model Name Prefix' checkbox which is checked.

File Name shows how output files are named. Entering **Prefix** allows you to prepend a common string to output file names.

Sprite Name shows how output sprites are named. If **Model Name Prefix** is turned on, the model's name will be prepended to sprites' name.

Baking



The screenshot shows three buttons stacked vertically: 'Bake the selected animation', 'Bake the selected model with its all animations.', and 'Bake all models'.

You can bake the selected animation, the selected model, or all models in the model list at once.


'Bake the selected animation' button is exposed only if the selected model is a Mesh Model which has animations.

If the model list contains both mesh models and particle models, **'Bake all models'** button are not visible.

Play Mode Baking (beta version)

Play mode baking function is added to support mirror animations, animation rigging, etc.

Unlike capturing a model forced to animate by simulating animation in editor mode, in play mode, you capture a model that animates by animation.

Press the play button  to enter play mode and do the same as you did in editor mode.

Features that work only in play mode

- * Projectile in [## Particle Model](#)
- * Saving selected frames as an image after [## Sampling](#)

Features that do not work in play mode

- * Root Bone and Customizer in [## Mesh Model](#)
- * [## Preview Window](#)
- * Simulation of [## Filming](#)

etc.

- * Adjust Apply Root Motion, etc. so that the model stays in place in play mode.
- * When baking, the following errors may occur.

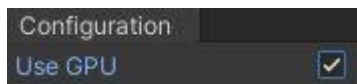
```
NullReferenceException: Object reference not set to an instance of an object
UnityEngine.UIElements.UIR.TextureBlitter.BeginBlit (UnityEngine.RenderTexture dst) (at <cd855c76ab374957a4384f21004cd44a>:0)
UnityEngine.UIElements.UIR.TextureBlitter.BlitOneNow (UnityEngine.RenderTexture dst, UnityEngine.Texture 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.UIElement element) (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)
```


Drawing the baking progress in the Studio Inspector window is an error in play mode, so please ignore it as it does not interfere with the creation of the output files.

GPU Usage

To extract colors and trim images faster, you can use GPU.

When you click 'Animation Baking Studio > Configuration' in the Assets menu, the Configuration window pops up, and here you can turn on the Use GPU.



The method using GPU is still experimental and has the following constraints.

- * If the trimming is on, the margin must be 0.
- * Normal maps are not generated properly.

Cautions

- * Delete SpriteBakingStudio folder and update it when you update it on SBS 2.
- * If there is an Object Baking Studio folder, delete it to avoid duplicated files.
- * If you use Post Processing like Bloom effect in HDRP/URP projects, adjust Alpha Threshold of Default Extractor larger than 0.
- * If you use HDRP, you must turn off Custom Frame Settings > Rendering > Post-process > Color Grading of Main Camera.
- * The package contains the Unity 5 version of Standard Assets. If Standard Assets already exists, the import might not work well.

After using this asset, if you like it, please rate it. <https://assetstore.unity.com/packages/slug/31247>

e-mail: beggu84@naver.com