

Animation Baking Studio

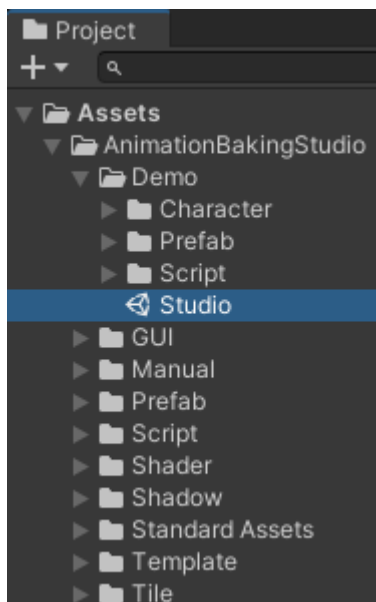
(ver. 3.3.0)

Animation Baking Studio is an asset that takes pictures of animating 3D objects as it displayed in the Game window creating sprite sheets (PNG), animation clips, etc.

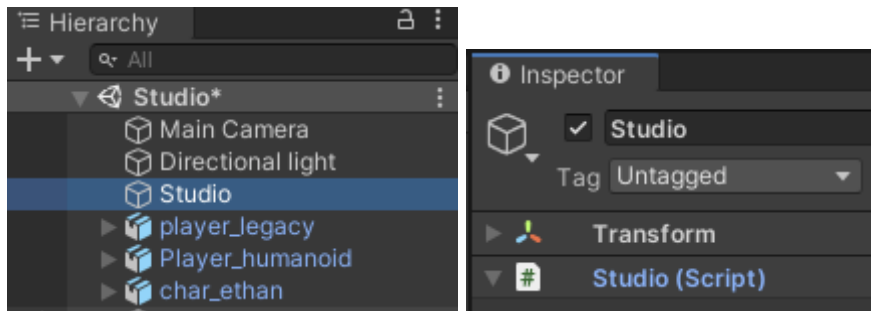
Quick Guide

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

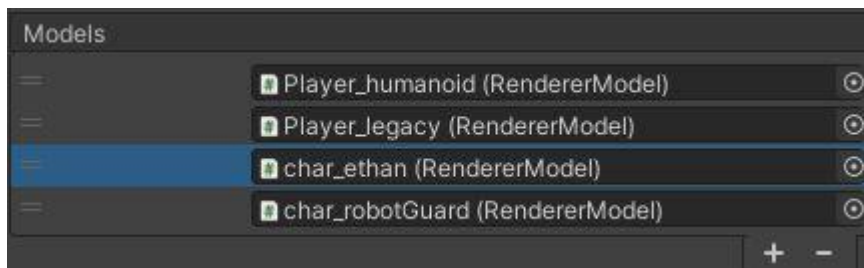
1. Open the enclosed Studio scene in the Project window.



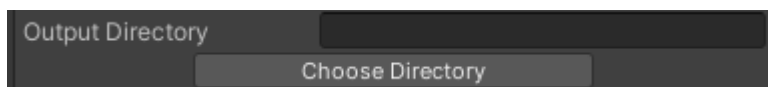
2. Select the Studio object in the Hierarchy window and see the **Studio** component in the Inspector window.



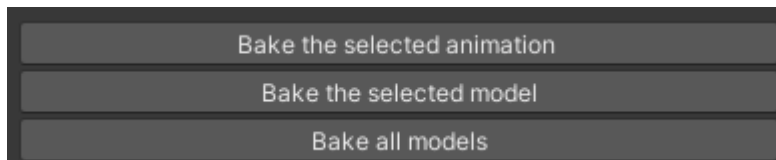
3. Select one model in the model list.



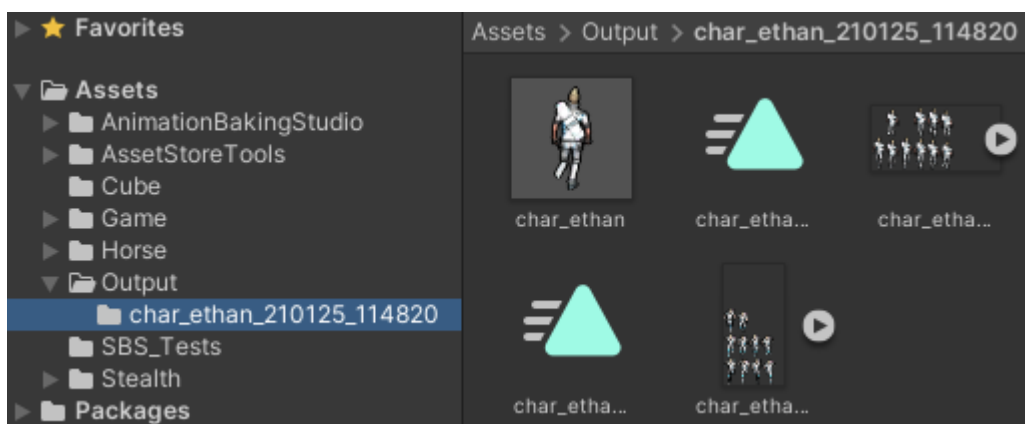
4. Click **Choose Directory** button at the bottom to select a folder where you want to save output files.



5. Clicking one of the baking buttons starts baking.



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



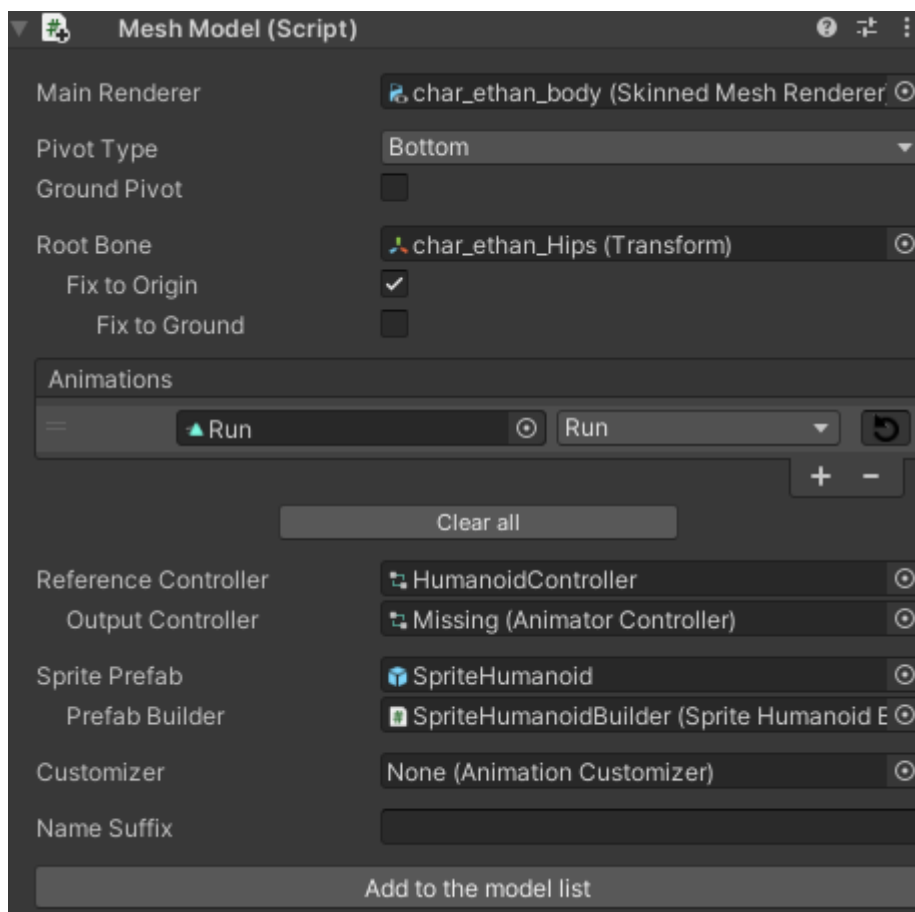
Model

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

Mesh Model

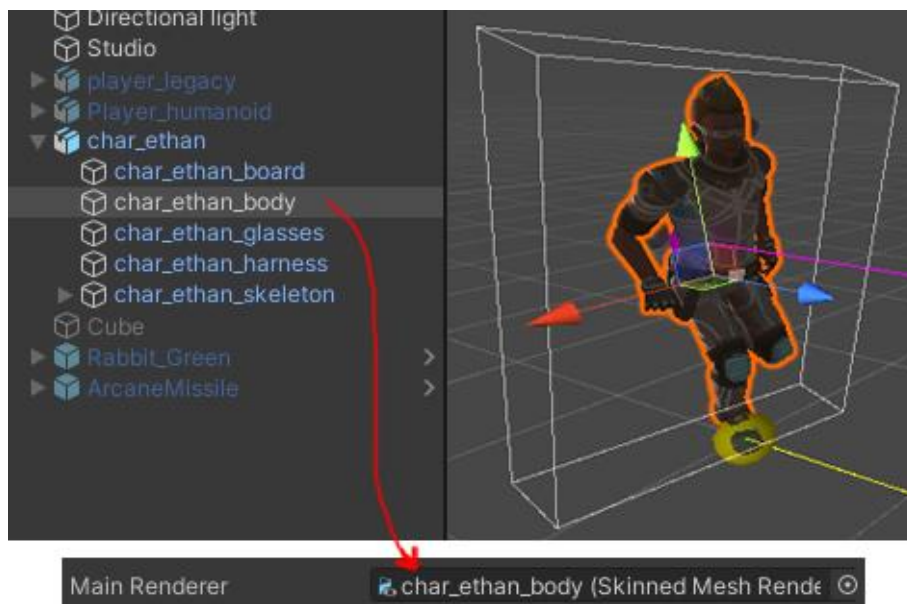
To take pictures of a humanoid object or a mechanim object, Avatar and Animator Controller should be specified in its Animator component.

Adding a **MeshModel** script to a typical object which has at least one Mesh Renderer or a hierarchical object which has at least one Skinned Mesh Renderer makes the object a Mesh Model.



Main Renderer is required to calculate the model's size and to determine whether it is a hierarchical object. If it is a usual object, the main renderer is automatically selected when a **MeshModel** script is attached. If there is any larger renderer in the object's hierarchy than the

auto-selected one, it is recommended to replace it.



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



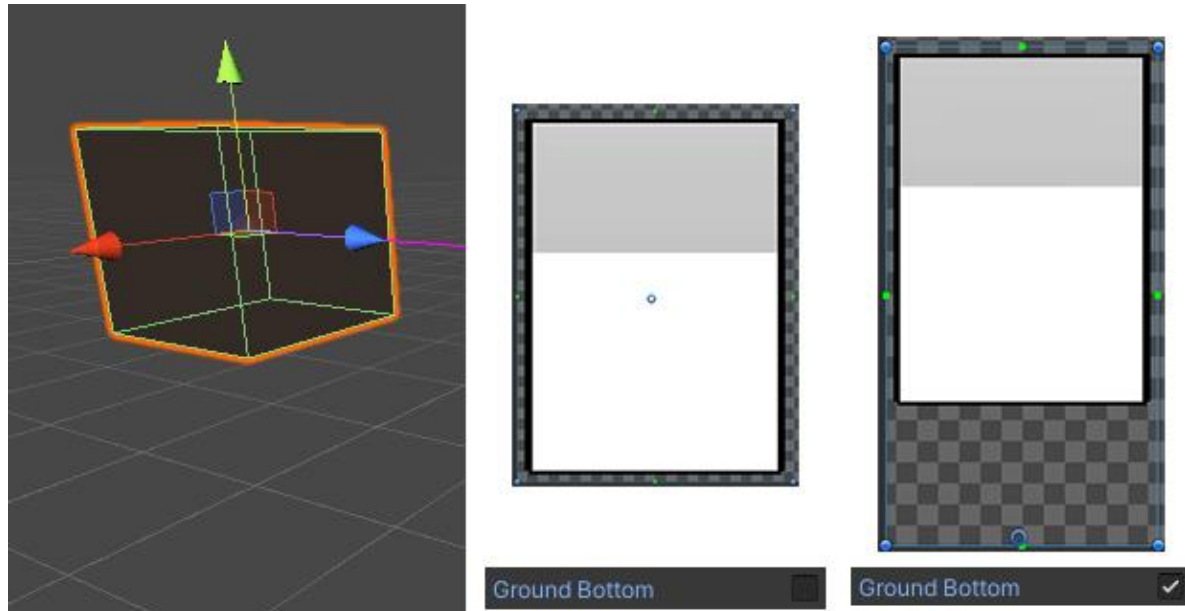
However, if the world position is at the model's vertical center, you should change it to the Center.

As the world position of non-hierarchical models is at the vertical center, the pivot type is hidden and is internally set to the center.

If the model is rotated around the y-axis, create an empty object, add the current model object as its child, and then add a **MeshModel** script to it.

When the model which is floating from the ground is taken with the trimming feature turned on,

some part of output images corresponding to the area under the model's bottom is cut. 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 an animation runs, if child objects deviate from the root object, turn on Root Motion in Animator component. If the root motion doesn't work properly, 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, you can also stick child objects to the ground when a jumping or a flying animation runs.

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

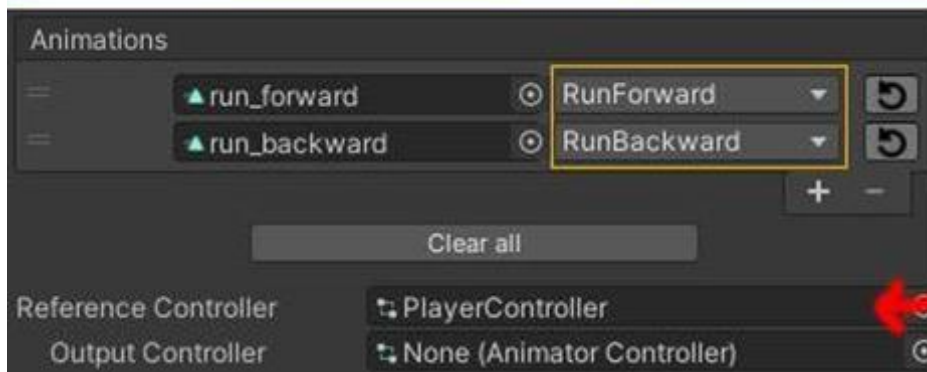
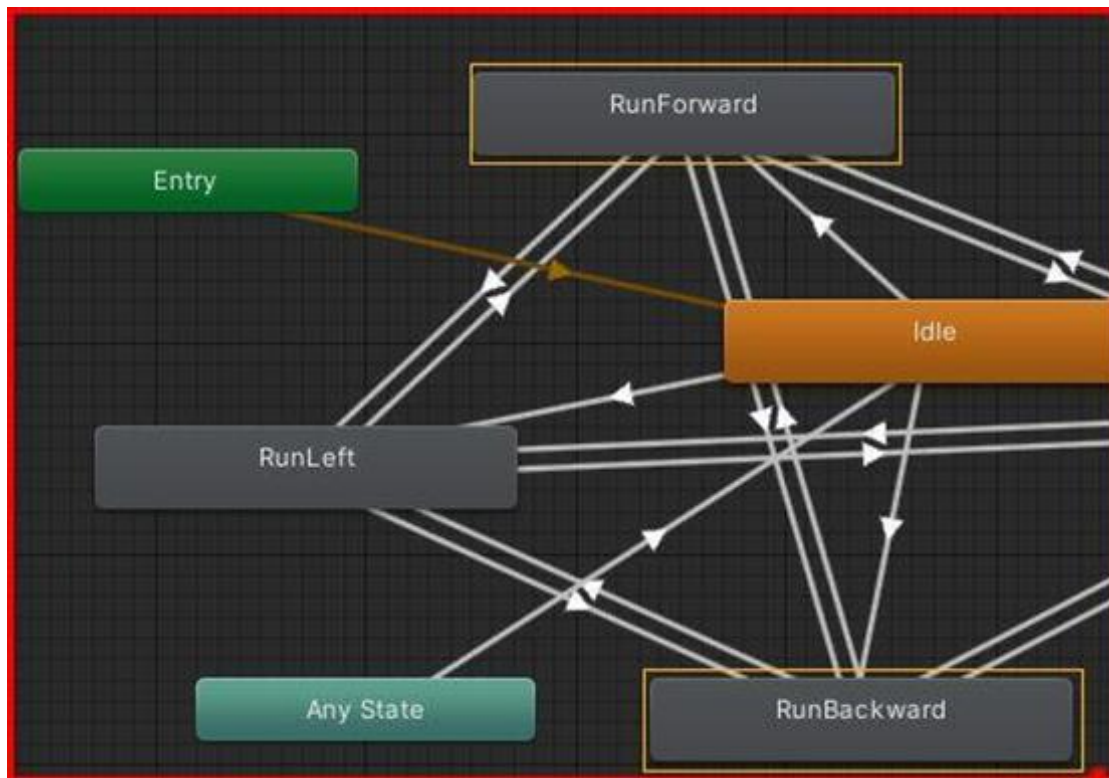
To take pictures of an animated model, animations should be registered in **Animation**. You can add a field by clicking '+' button and then put an animation clip into the field, or you can drag and drop an animation clip into the list to add a field.



During baking, if  of each field is turned on, a looping animation clip is generated.

Reference Controller is required to be used when an output controller is created.

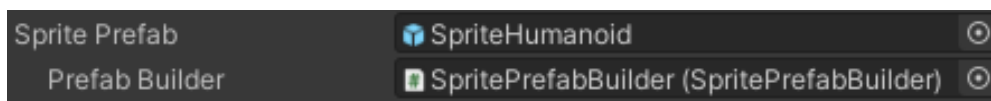
If there are some animation states in the specified reference controller, a combo box appears in each field, where you can choose one of the states, which is corresponding to the animation clip in the field.



During baking, output animations states and transitions are added to the controller specified as **Output Controller**, but if the field is 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.

When you turn on **Make Prefab** in the Studio, which will be described later, the prefab object to be referenced must be assigned to **Sprite Prefab** for each model. Additionally, you need to specify **Prefab Builder** to connect the sprite and 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 modify or transform this model object during capturing, if you add a script that inherits **AnimatorCustomizer** and implements **UpdateFrame** function to an empty object, and then put this object in the customizer.

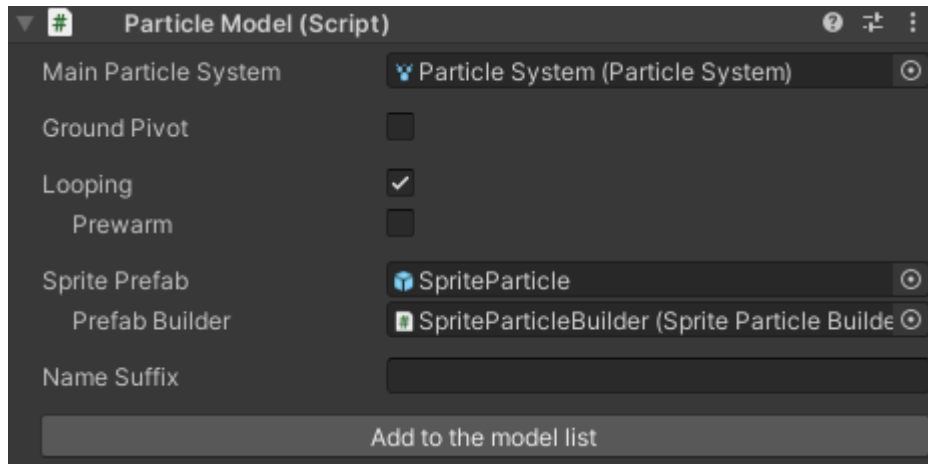
To add a specific word to the model's name, which is used for naming output files, enter it in **Name Suffix**.

Clicking '**Add to the model list**' button registers 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 model and to simulate the whole particle systems. Usually, it is auto filled when a **ParticleModel** script is attached to. If there is any major particle system in the object's hierarchy than the auto-selected one, it is recommended to replace it.

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.

It is recommended to turn on or off **Looping** and **Prewarm** according to the same name fields of the main particle system component.



As mentioned in [## Mesh Model](#) part, **Sprite Prefab** and **Prefab Builder** must be specified to generate prefabs.

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

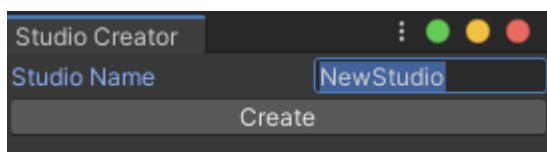
Clicking '**Add to the model list**' button registers model in the model list of the studio.

*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

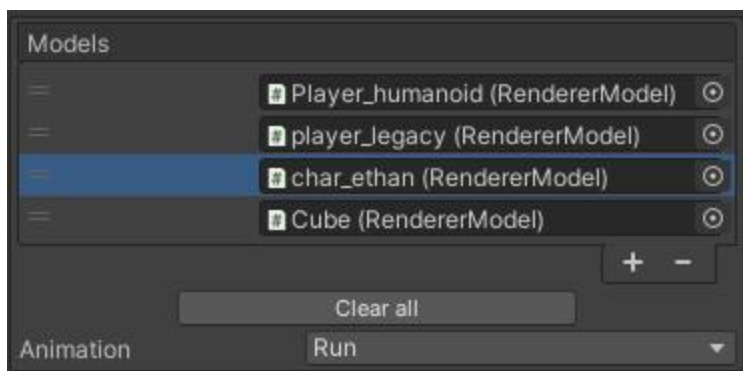
To take pictures of models, a studio is required. Create an empty object and add a **Studio** script on it.

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 the studio and then clicking **Create** button generates a new scene and open it.*

Model List & Animation

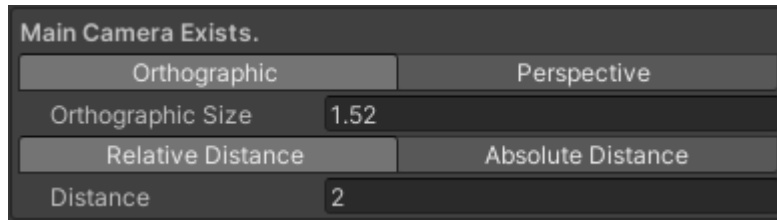


At first, target models should be registered in the model list. You can add a field by clicking '+' button and put a model into it or drag-and-drop it. Also, you can also register a model by clicking 'Add to the model list' button in each (Render/Particle) Model component, as explained in **# Model**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

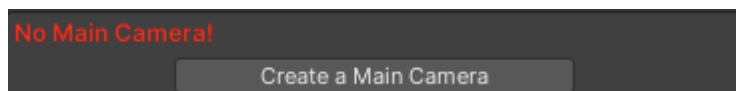
When a model in the model list is selected, some fields of the component could be changed according to it.

When you choose a mesh model in which has animations in its animation list, **Animation** combo box appears where you can choose one of them.

Main Camera



A main camera – a camera with the MainCamera tag - is required for filming. 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 modify 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.

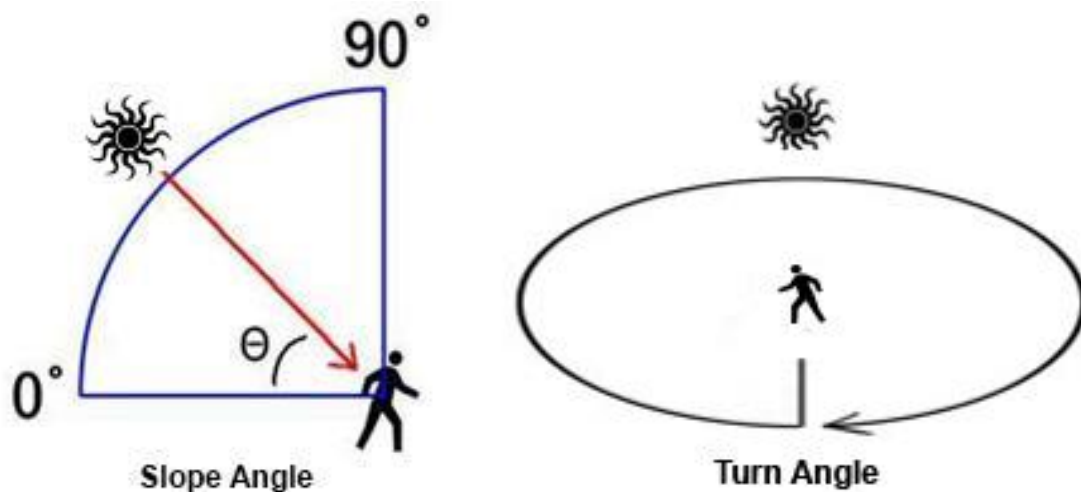
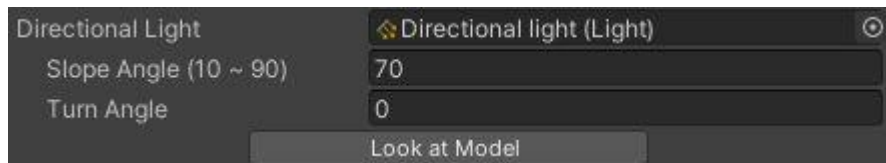
Directional Light



Directional Light is not required, but it can be specified for convenience.

To minimize the shadow cast on models when the main camera rotates, it is recommended to

turn on **Follow Camera Rotation** to rotate the directional light in the same direction. Or you can directly rotate the directional light by adjusting **Slope Angle** and **Turn Angle** that appear when **Follow Camera Rotation** is turned off.



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

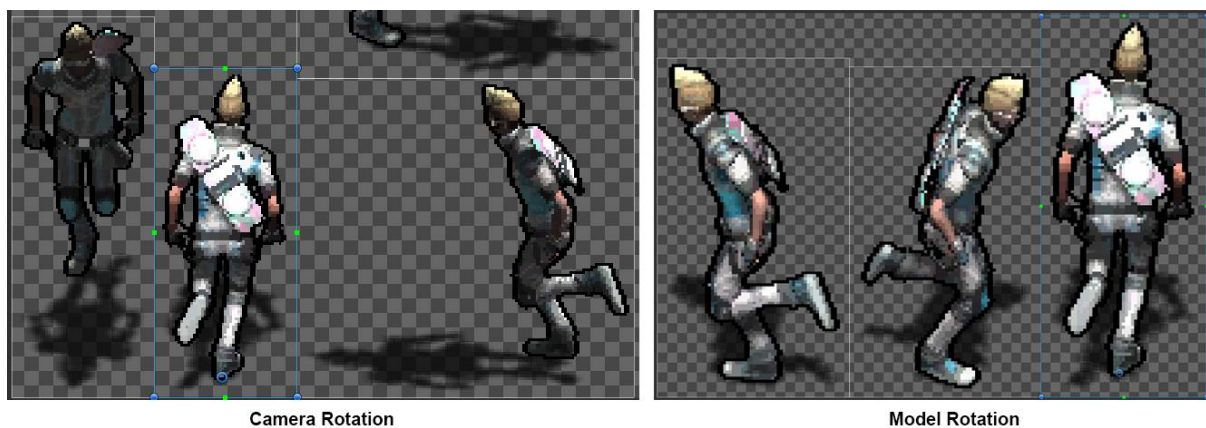
If you want the directional light to move to the same position when the main camera is moving, turn on **Follow Camera Position**. But as it is a 'directional' light, the position of the light is not very important.

View

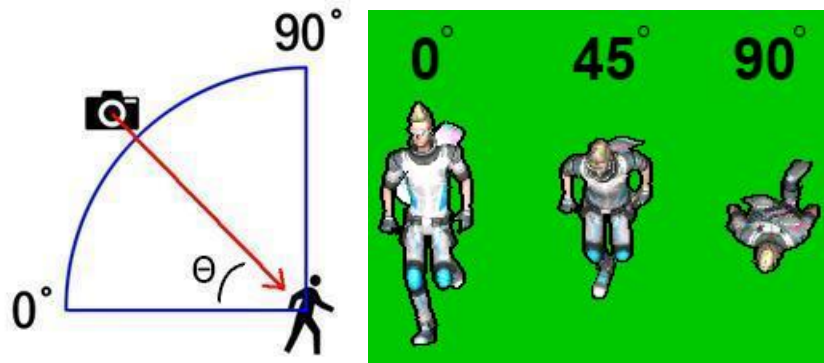
Camera Rotation		Model Rotation	
View Slope Angle (0 ~ 90)	30		
Show Tile	<input checked="" type="checkbox"/>		
Tile Type	Square		
Aspect Ratio	X 2	Y 1	
View Size	4		
Base Angle (0 ~ 90)	0		
0°	<input checked="" type="checkbox"/>		Apply
90°	<input type="checkbox"/>		Apply
180°	<input checked="" type="checkbox"/>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

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 the tilted shadow type is selected, or the model has an animation that moves in absolute coordinates.



You can rotate the main camera around x-axis by adjusting the 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.



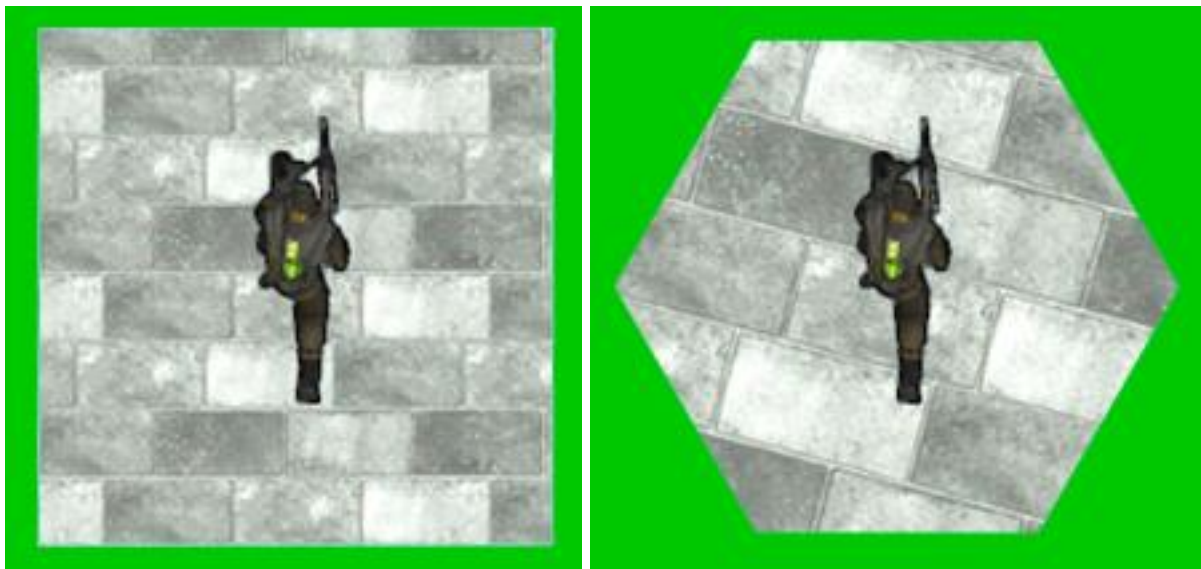
If you plan to make tile-based 2.5D game, characters would be drawn according to the tile's ratio.



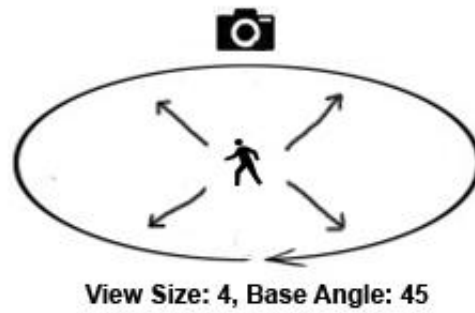
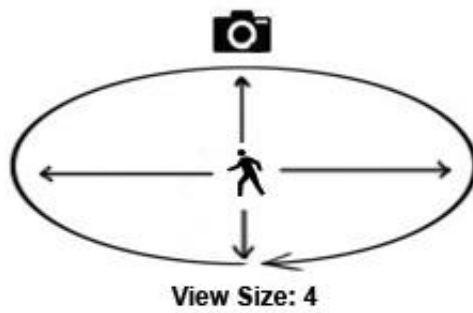
When **Show Reference Tile** is turned on, a tile appears, which disappears during filming.

When you modify **Aspect Ratio**, the slope angle is auto calculated with the main camera's rotating. Conversely, modifying the slope angle affects the aspect ratio.

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

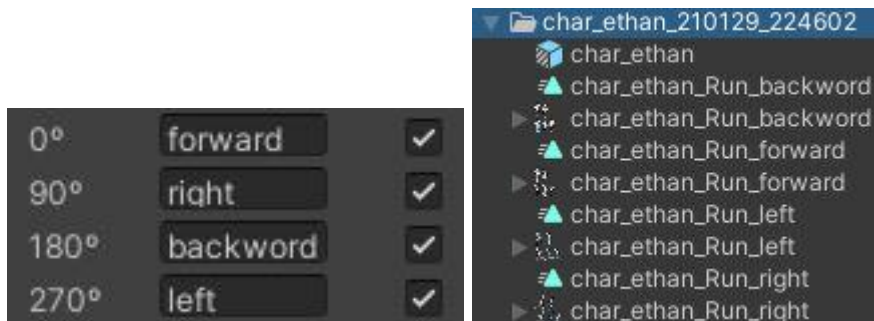


As many views as $\lceil 360 \text{ degree} / \text{View Size} \rceil$ are created and each view has a $\lceil \text{Base Angle} + (360 \text{ degree} / \text{View Size}) + i \rceil$ turn angle.



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

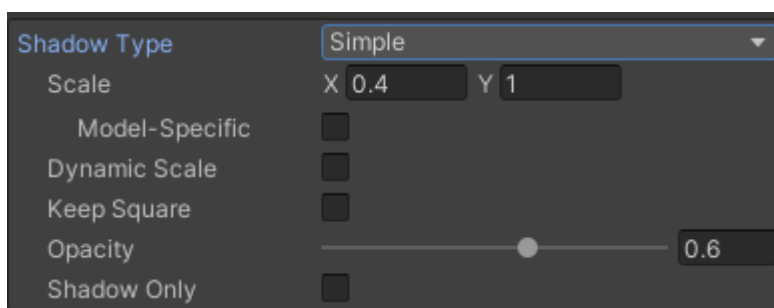
During baking, turned on views are continuously filmed and the turn angle is attached as a string to the end of generated files' name. If you name each view, it is used instead of the angle string.



Shadow

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

Simple Shadow



It is oval-shaped shadow.



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

The same scale 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 model's size while the model animates, turn on **Dynamic Scale**.

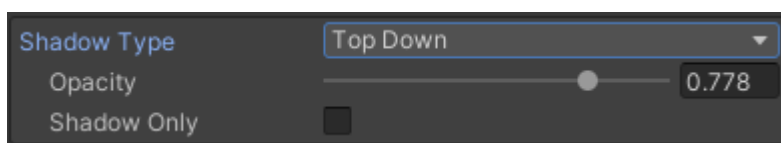
If you want to keep the aspect ratio uniformly, turn on **Keep Square**.

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.

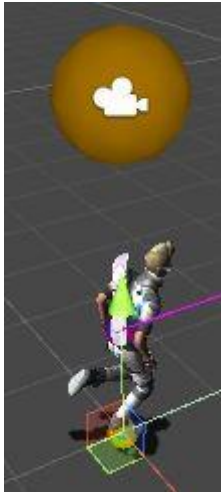
If you turn on **Shadow Only**, you can hide the model and only take the shadow during capturing.

Particle models are very large in size, so you will need to adjust the scale very small.

Top-Down Shadow



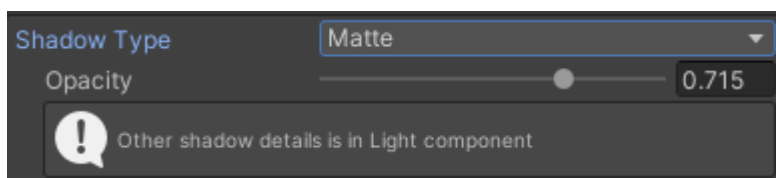
It is a shadow using the shadow camera that looks down the model vertically.



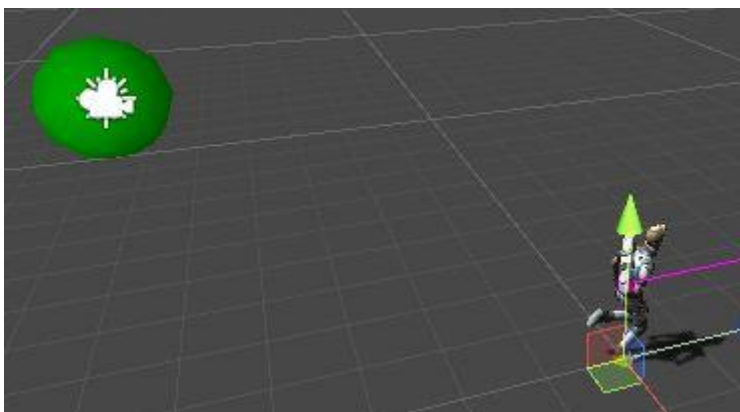
You can adjust **Opacity** and take the shadow without the model during filming by turning on **Shadow Only**.

If a particle model is selected, you will not see the top-down shadow in the Game window. It's only taken during sampling and baking, and the internal shadow logic is different from that for Mesh Model, so there will be a slight delay.

Matte Shadow

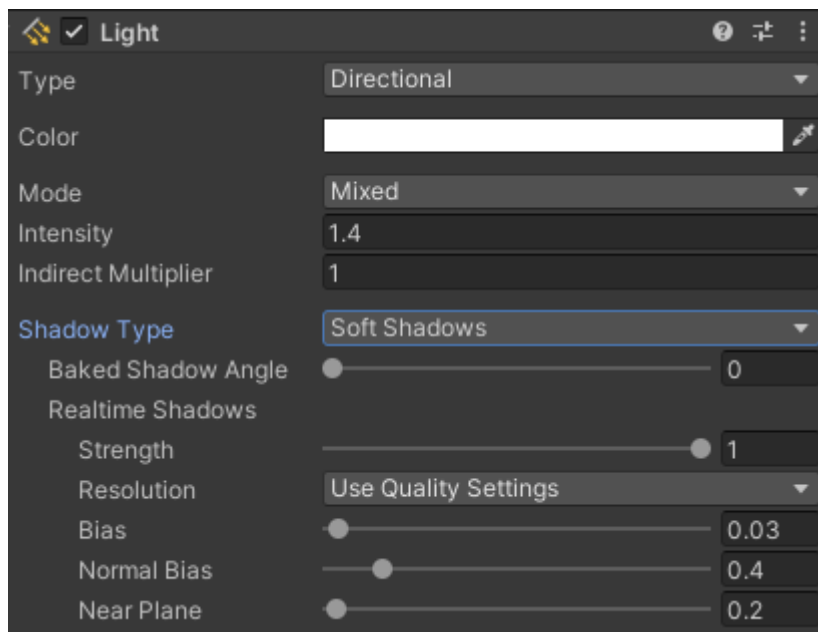


It is a shadow using a special transparent floor, which is the term used in 3ds Max or Maya.



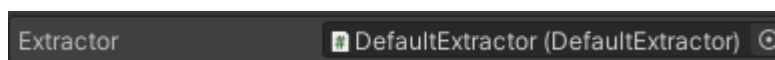
If you turn on **Follow Camera Rotation** of the light properties and select this shadow type, you will not be able to see the shadow in the Game window because the model completely blocks the sight.

You can adjust **Opacity**, and you can change other settings in the Light component of the directional light object.



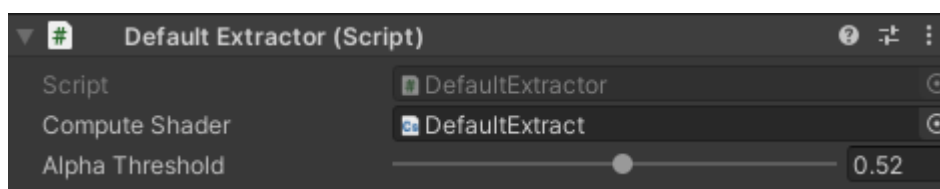
This shadow type doesn't work well for a particle model.

Color 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/DefaultExtractor', and like any other extractors in '~/Demo/Prefab', you can directly inherit **Extractor** abstract class and implement **Extract** function to create a new extractor.

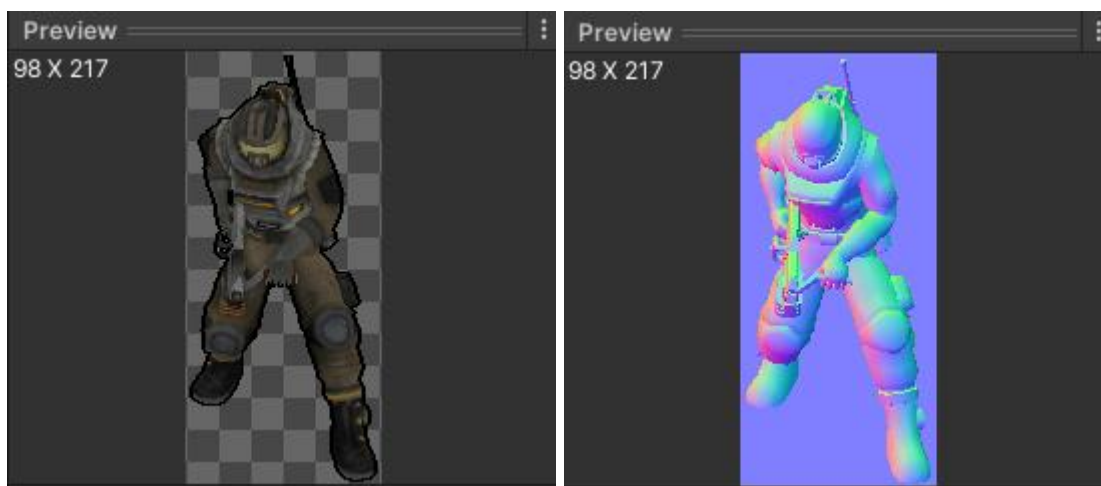
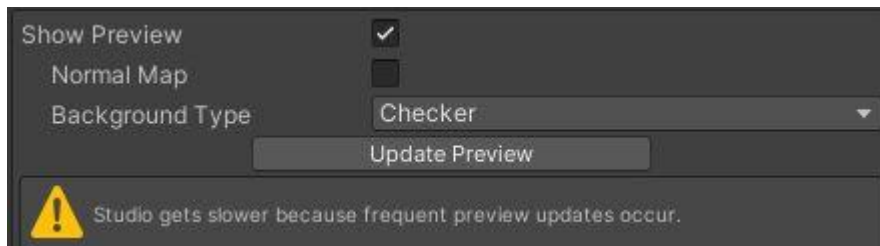


If **Alpha Threshold** is turned on, pixels with alpha under the specified value is considered background.

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/OpaqueExtractor'.

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.

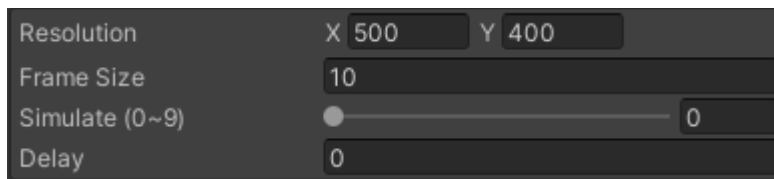
If **Make Normal Map** of **# Extra Output** is turned on, **Normal Map** appears, and when you turn it on, you can see the normal map result image.

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

When you leave the preview turned on, the Studio does a capturing work whenever any of the studio's properties change so slows down.

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

Resolution and Frame



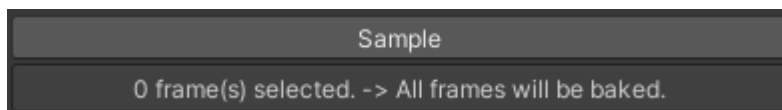
Resolution is the size of the result image that is saved as an image file or a sprite with that size during baking, if the trimming is not turned on.

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

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

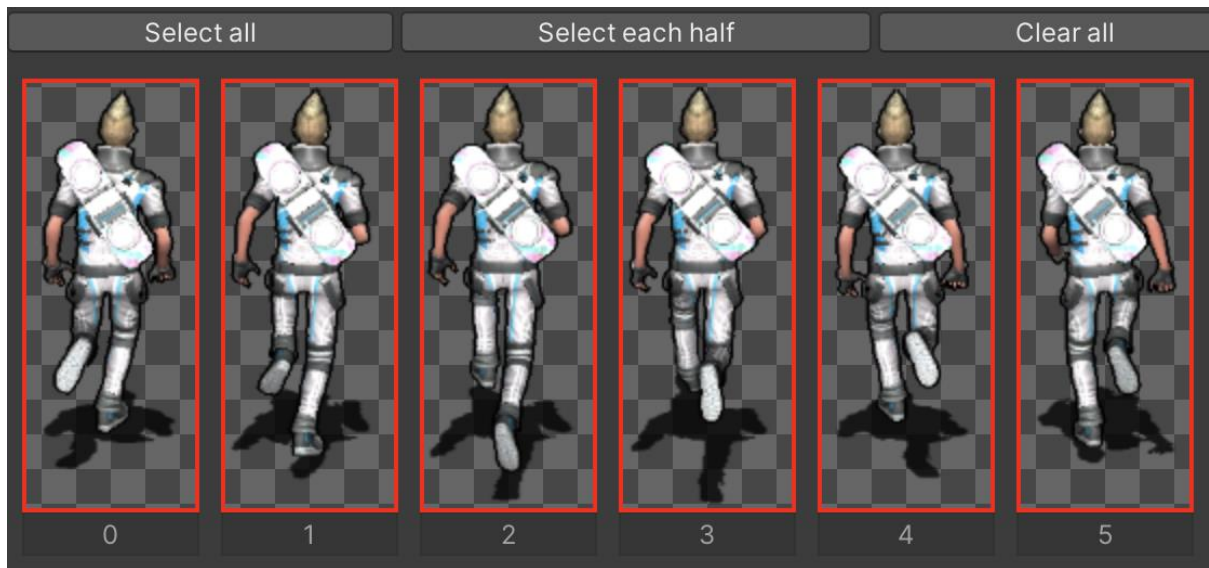
When you use the top-down shadow, some mismatch between the model and the shadow could occurs. In that case, you can raise **Delay** a little to synchronize them.

Sampling and Selecting Frames

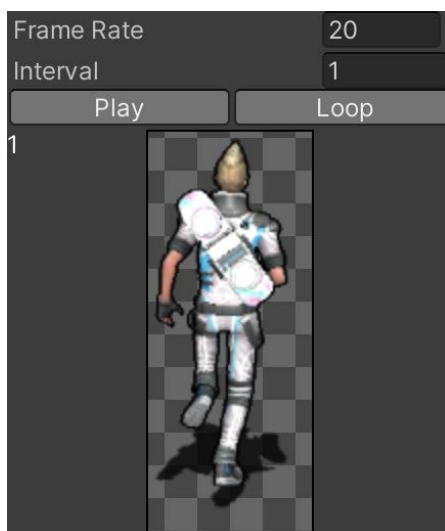


You can sample the selected animation by clicking **Sample** button. When sampling is complete, both the Frame Selector window and the Animation Preview window appear.

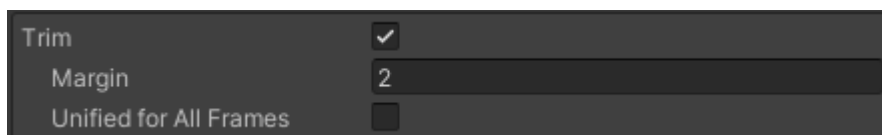
In the Frame Selector window, you can choose proper frames to take during baking. The images are cut fitting the model, but it does not affect whether the trimming occurs or not during baking.



You can preview 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.



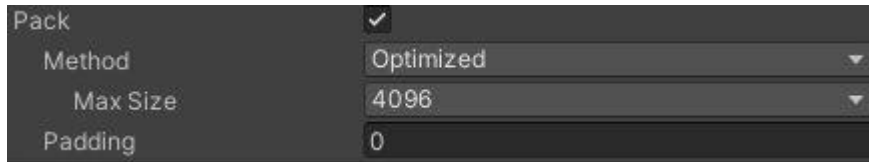
Trimming



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

If **Unified for All Frames** is turned on, all images are cut to the equal size for all frames.

Packing (Sprite Sheet)



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

There are two packing methods:

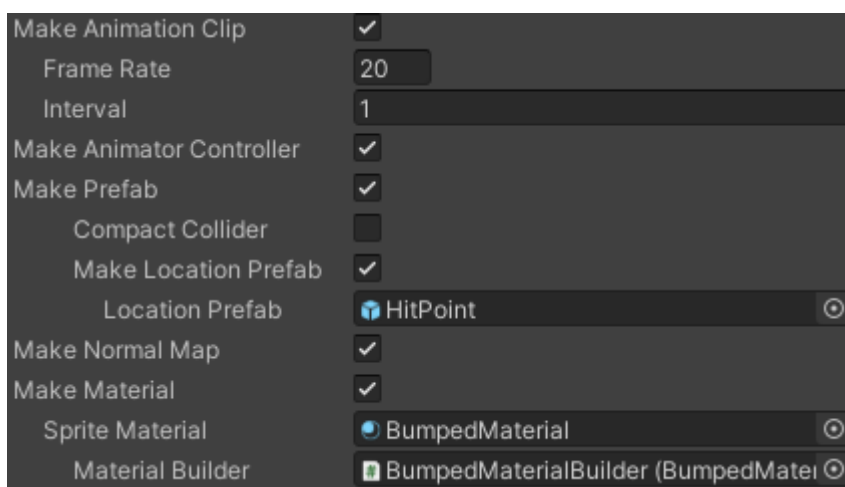
Optimized method is owned in Unity Engine and places sprites properly to create the smallest sprite sheet. The smallest sprite sheets are created, the size of which is not bigger than **Max Size**.

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.

Extra Outputs

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



If you turn on **Make Animation Clip**, you can create animation clips associated with sprites,

materials, etc. that are generated during baking.

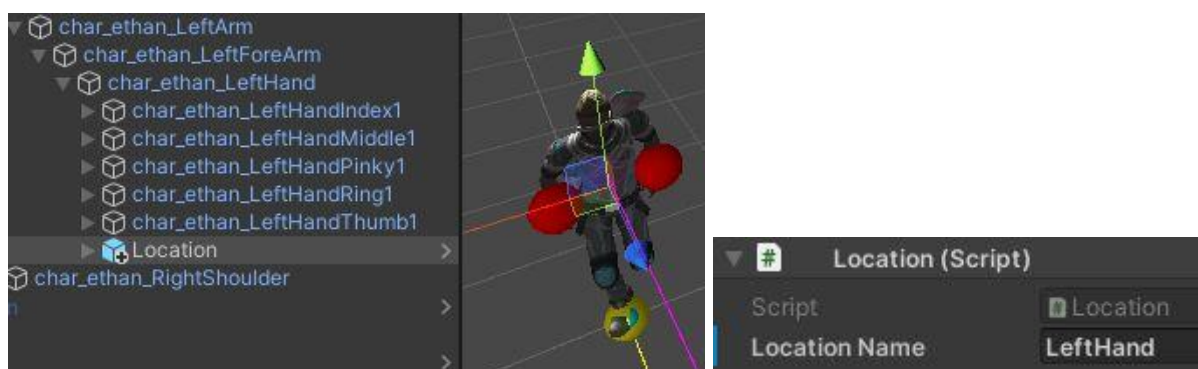
If you turn on **Make Animator Controller**, an animator controller is created, in which animation clips are registered to each animation states.

The above two fields are only exposed when packing is on.

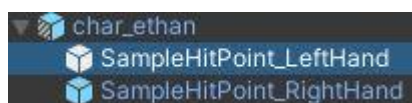
If you turn on **Make Prefab**, the object that is specified as **Sprite Prefab** of each model is instantiated per model during baking.

When there is a Box Collider 2D component inside the specified the Sprite Prefab object, if you turn on **Compact Collider**, the collider size and offset are calculated to perfectly fit to the model's size even if the trimming margin is larger than 0.

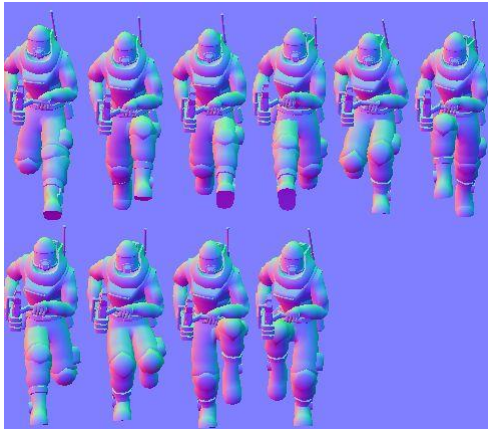
If you may want to track the location of bone objects within the resulting prefab object, at first, you can add '~/Prefab/Location' objects to proper bone objects and enter **Location Name** in it.



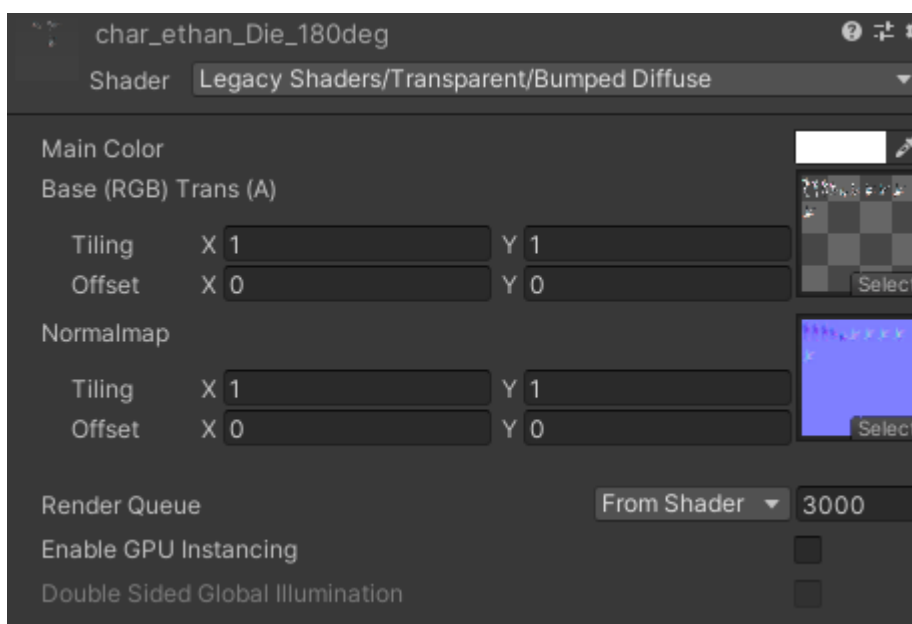
If you turn on **Make Location Prefab** and specify **Location Prefab**, the location objects are instantiated inside the resulting prefab object.



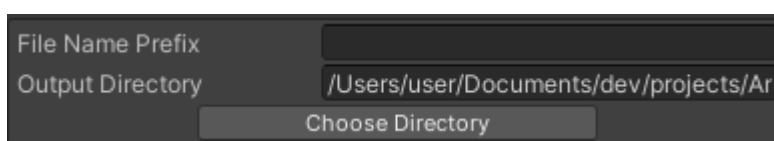
If you turn on **Make Normal Map**, normal map files corresponding to each model image file are created. In order to make normal maps, **Model Rotation** of **# View** must be selected.



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 **Make Material** and assign the material to be created to **Sprite Material**. In general, the model image and normal map image are automatically linked to the material, but if you need to link them in person, specify a prefab object that includes a script implementing **MaterialBuilder** class to **Material Builder**.



Output Files Path



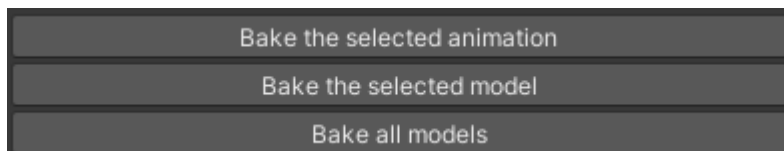
File Name Prefix allows you to put a common name before the name of result folders and files.

Output Directory is the root folder of the result folders and files, and you can choose the folder by clicking **Choose Directory** button.

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).

Baking



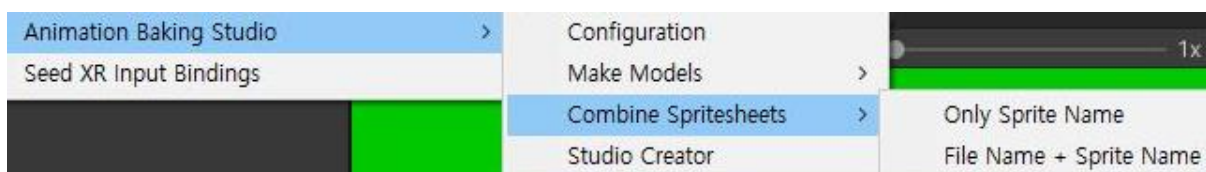
You can bake the selected animation, the selected model, or all models in the model list at once.

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

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

Combining Sprite Sheets

After selecting more than one sprite sheet in the Project window, you can combine the selected sprite sheets by clicking one of the two menu items under 'Animation Baking Studio > Combine Spritesheets' in the Assets menu or the context menu.



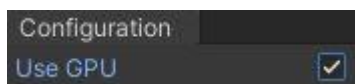
When running by clicking the Only Sprite Name, only the source sprite name is used.

When running by clicking the 'File Name + Sprite Name', the sprite sheet name is attached ahead of the sprite name.

GPU Usage

In order 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