Sprite Baking Studio

(ver. 2.5.2)

Terms and Symbols

「SBS」 is the abbreviation of Sprite Baking Studio.

「object」 means only Unity GameObject.

「pivot」 indicates a position in 3D which is going to be transformed into 'Custom Pivot' of sprites.

「view」 means the scene to be filmed, the direction of the model.

「bake」 includes screenshotting a model and creating PNG files.

「property」 means items such as input box, checkbox, and combo box in SpriteBakingStudio and StudioModel components.

「※」, 「▼」, 「■」 are symbols for a property, an enumeration item, and a button, respectively.

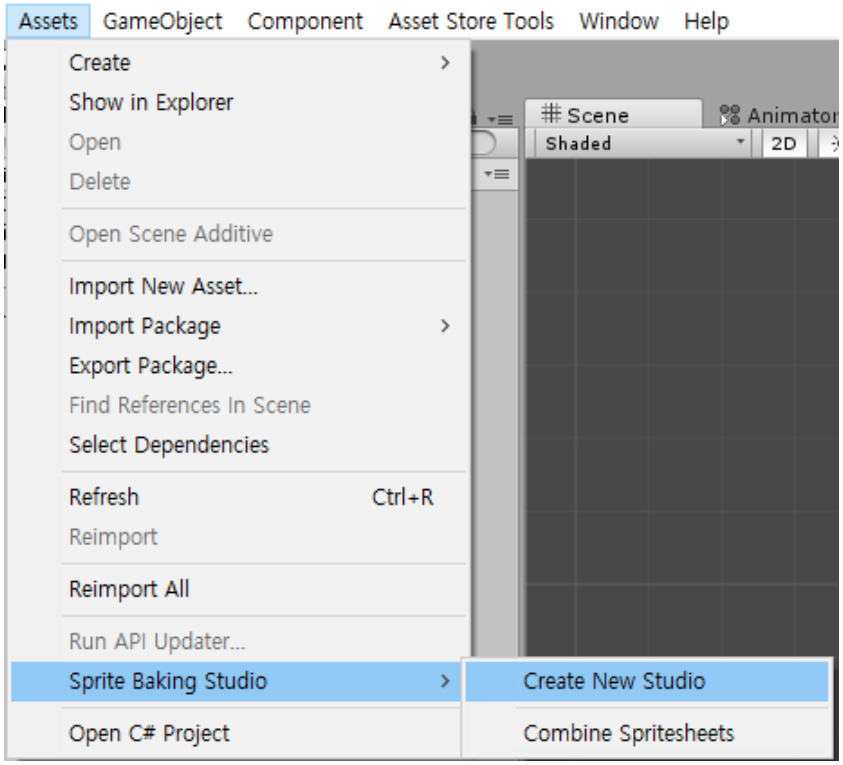
Terms wrapped by quotation marks indicate the name of a file, an object, a property, an enumeration item, a button, etcetera.

Words enclosed by square brackets are the name of windows.

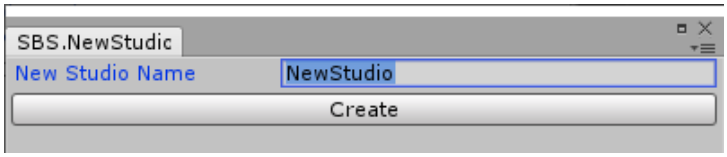
After using this asset, if you like it, please rate it. https://assetstore.unity.com/packages/slug/31247

Quick Tutorial

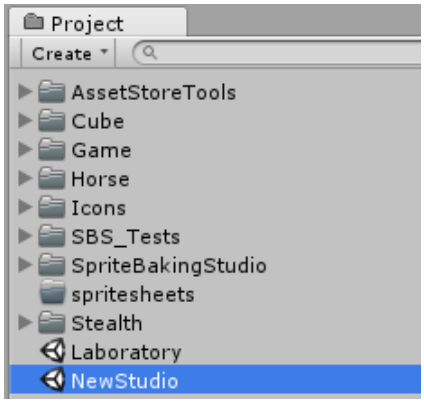
1. Click 'Assets/Sprite Baking Studio/Create New Studio'



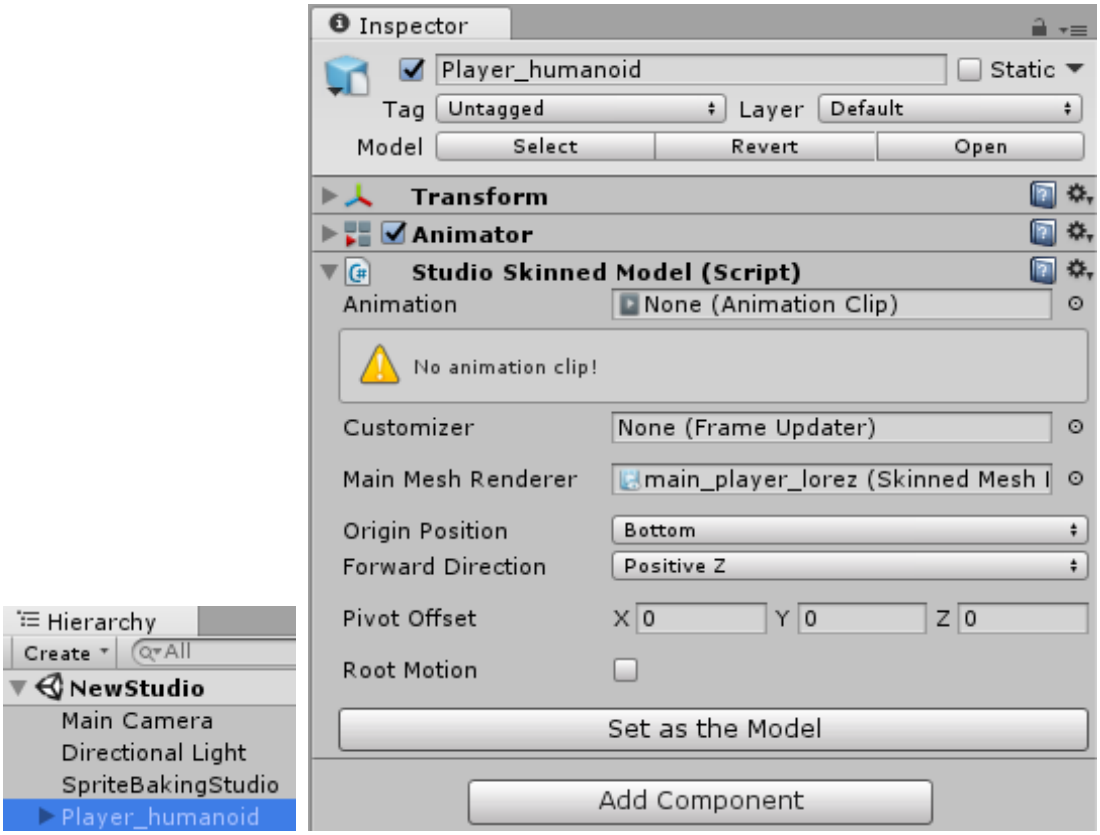
2. Write down a new Studio(Scene)'s name and click 'Create' button.



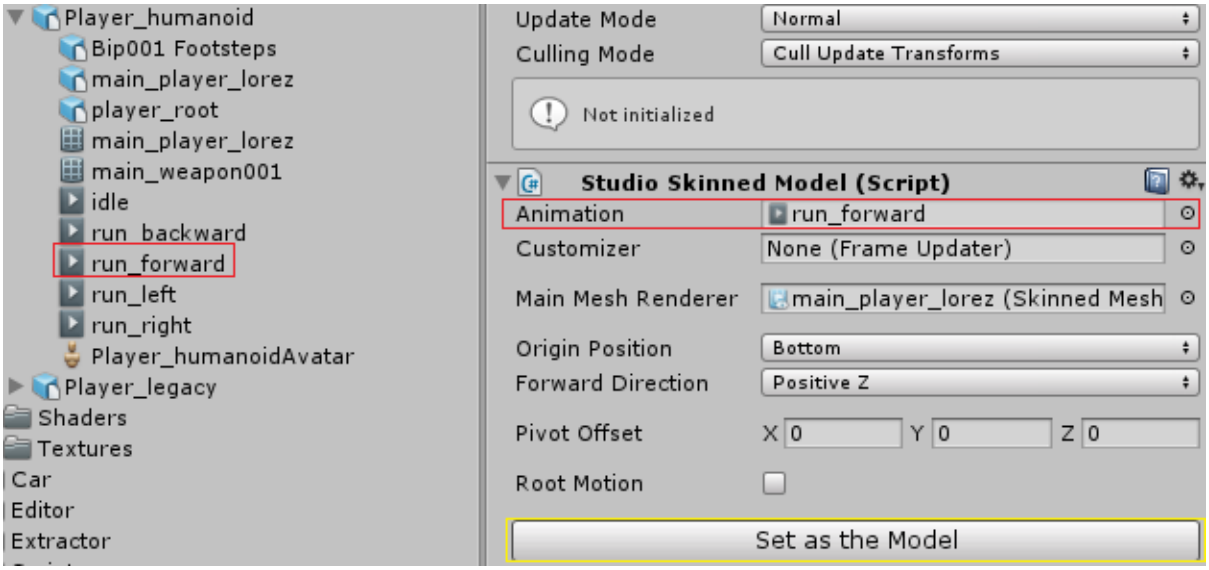
3. Open the new scene just created in [Project] window.



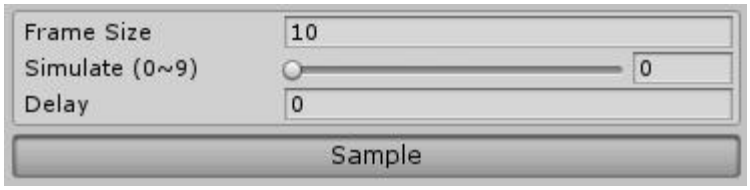
4. Instantiate a model you want to take pictures and add StudioSkinnedModel script(component) to the model object.



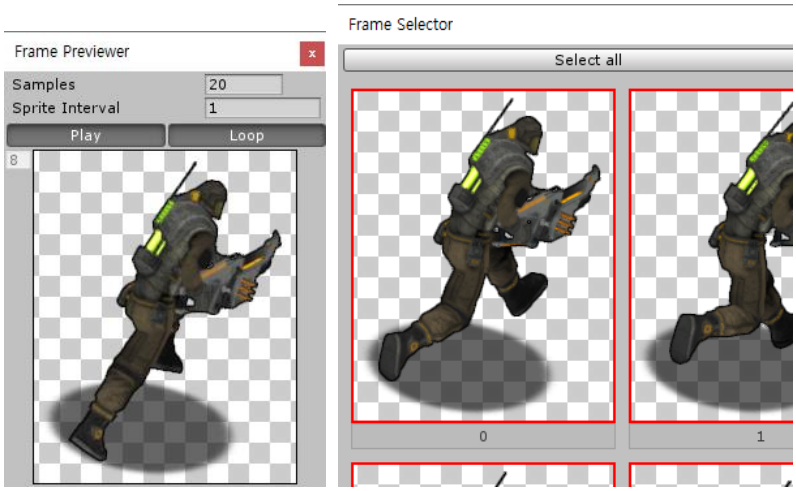
5. Set up an animation you want to capture and click 'Set as the Model' button. (You can do these actions in SpriteBakingStudio as well.)



6. Set frame counts and click 'Sample' button in SpriteBakingStudio.



7. When a sampling operation ends, two windows are popped up. You can select frames in [Frame Selector] window for final baking operations and preview the continuous action of the selected frames in [Frame Previewer] window.



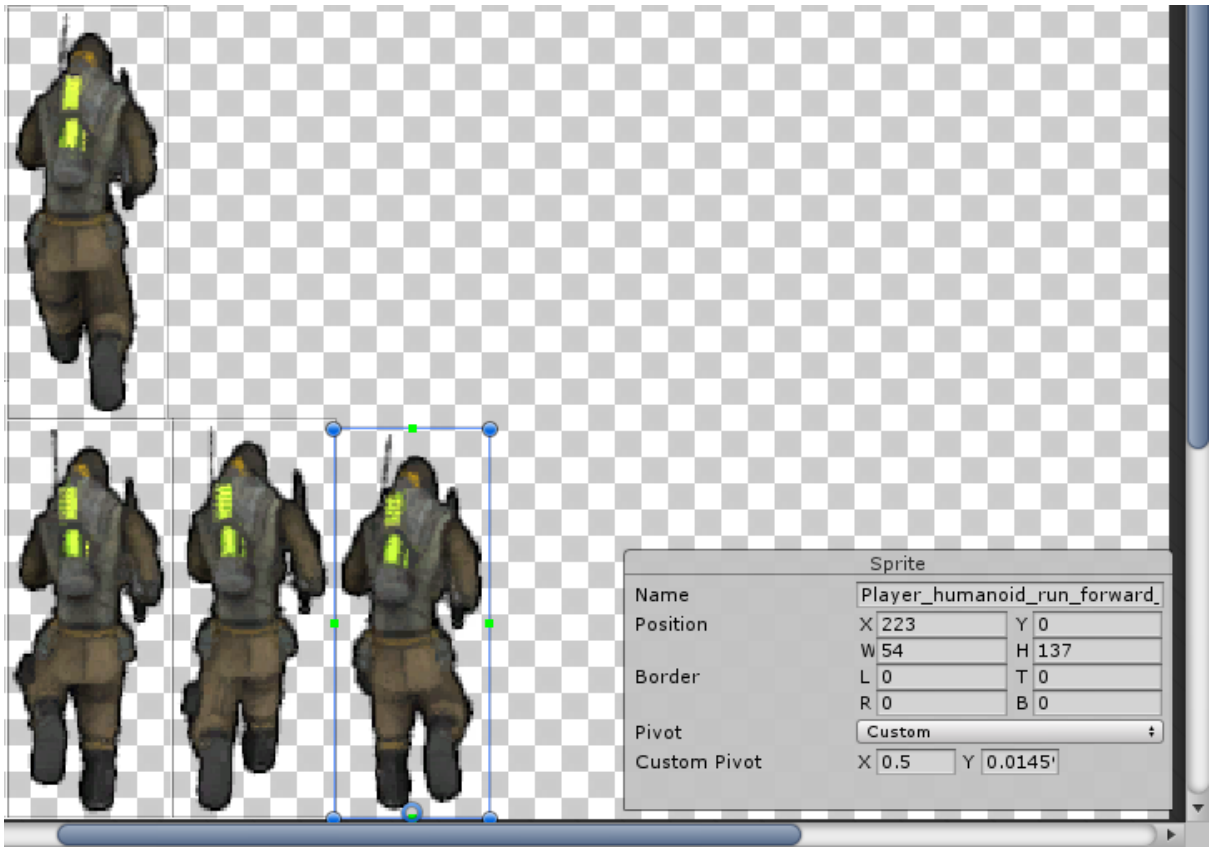
Close a window with all frames selected.

8. Click on 'Choose Directory' button at the bottom to set a path where resulting files are going to be saved.

9. Click on 'Bake selected frames' button shown up just below.

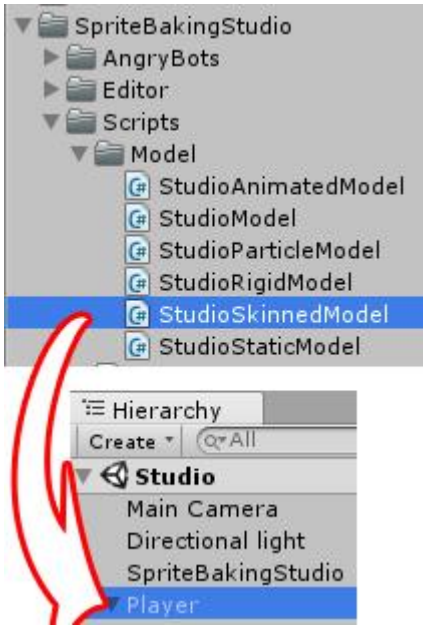
10. When a baking operation ends, check output files (.png and .anim) in the directory.

11. You can see custom pivot of the sprites in [Sprite Editor] window.



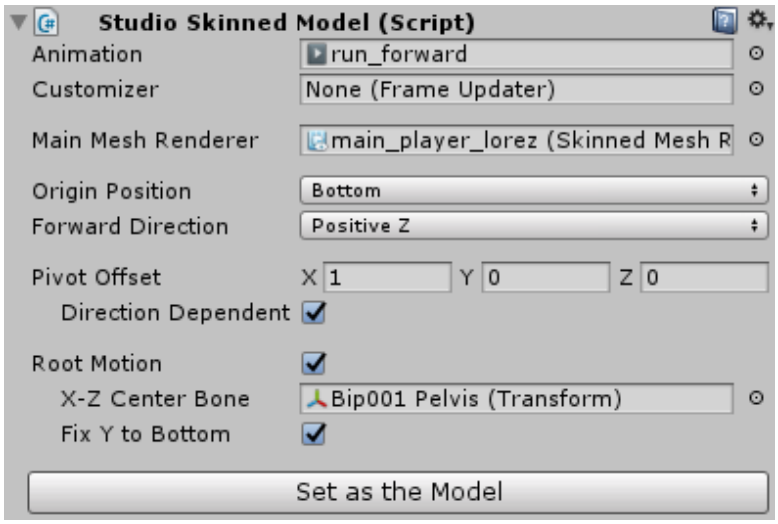
Functional Description (Model Scripts)

You must attach an appropriate model script to a GameObject as a component to take pictures of it.



◎ StudioSkinnedModel

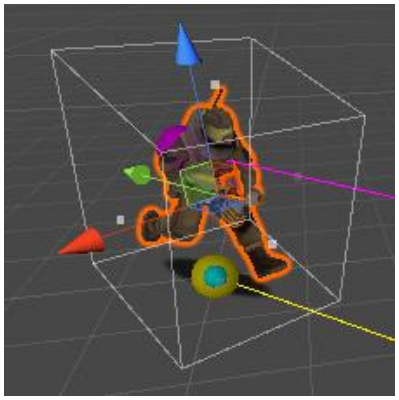
Model script for hierarchical bone structure objects in which at least one SkinnedMeshRenderer exists.



※ Animation – An AnimationClip containing a motion of the object.

※ Customizer <optional> – The UpdateFrame method of the FrameUpdater instance specified to this property is called on every frame.

※ Main Mesh Renderer <auto-selected> – To compute various sizes and positions, you must specify a SkinnedMeshRenderer that has the biggest bounds in the object hierarchy.



※ Origin Type – To calculate the vertical center and bottom position of the object, you must specify in which part of the body transform.position is located by selecting one of the enumerated items. ('transform' is the embedded variable of the Component class.)

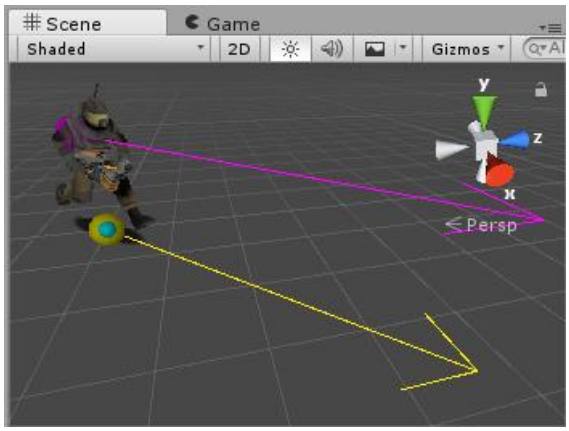
▼ Bottom <default> – transform.position is on the bottom of the object, and the vertical center becomes a position which is far from the bottom position by half the y-length of the bounds of the main mesh renderer.

▼ Center – transform.position is already in the vertical center position of the object.

If set correctly, a purple sphere is located near the center of the object in the [Scene] window. For your information, a yellow sphere means transform.position, and a small sky-blue sphere means a pivot.

※ Forward Direction – To calculate the orientation of the object, you must select one of the enumerated items to specify which direction is forward. The default is 'Positive Z'.

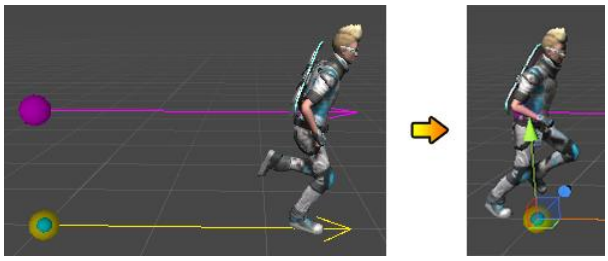
If set correctly, arrows orient the forward direction of the object.



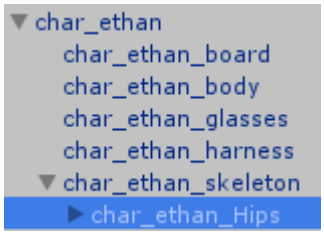
※ Pivot Offset – The pivot becomes the object's bottom position plus this vector value.

※ Direction Dependent – The vector value is multiplied by the forward direction of the object and then added to the bottom position. It only appears if the vector value is larger than 0.

※ Root Motion – Turning on this property prevents the object from moving out of (0, 0, 0) by the animation or prevents a body bone object from moving out of the root object.



※ X-Z Center Bone – You must specify a bone object that is a horizontal center object such as the spine, hips, or pelvis to hold the object in place.

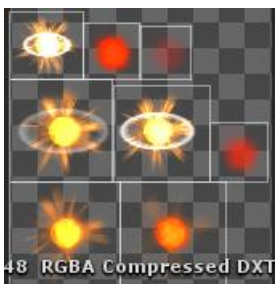


※ Fix Y to Bottom – Turning on 'Root Motion' basically constraints the object in the X-Z plane. When this property is turned on, the object is also held in the Y axis.

■ Set as the Model – This object is specified as 'Model' of SpriteBakingStudio component.

◎ StudioParticleModel

Model script for hierarchical objects in which at least one Particle System exists.



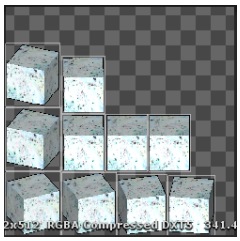
※ Main Particle System <auto-selected> – You must specify a Particle System that lasts longest in the hierarchy.

※ Pivot Offset – It is the same as that of StudioSkinnedModel except that the default is central.

'Prewarm' should be turned off in order to capture Particle System.

◎ StudioRigidModel

Model script for moving rigid objects.



※ Mesh Renderer <auto-selected> – To compute various sizes and positions, you must specify a MeshRenderer of the object.

'Animation' and 'Customizer' are the same as those of StudioSkinnedModel.

'Pivot Offset' is the same as that of StudioParticleModel.

◎ StudioStaticModel

Model script for motionless objects. Only one frame is taken.

'Mesh Renderer' and 'Pivot Offset' are the same as those of StudioRigidModel.

◎ StaticModelGroup

Model script for a group of motionless objects, taking one frame in a row.

※ Objects Root Directory – Path to a root folder containing objects.

■ Refresh Sub Models – Clicking on this button browses to all objects below the specified path and add a StudioStaticModel.

※ Checkboxes for each model – The final baking occurs for only selected sub models.

'Pivot Offset' are the same as those of StudioRigidModel.

Functional Description (SpriteBakingStudio)

To take pictures of an object, Main Camera must exist in [Hierarchy] window. It is recommended that setting 'Projection' of Main Camera to 'Orthographic' and then adjusting 'Size' in that state.

※ Model – You must specify an object with one of the model script attached.

※ Animation <Reference> – 'Animation' of the selected model. This property appears only for animated models.

■ Refresh Sub Models – It's the same as that of StaticModelGroup component and appears only when the model is StaticModelGroup.

※ Camera Size <Reference> – 'Size' of Main Camera. This property is visible only when the 'Projection' value of Main Camera is 'Orthogonal'.

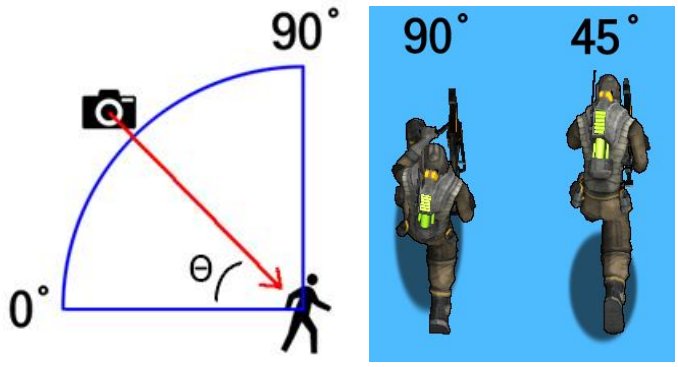
■ Adjust Camera – It adjusts the camera size to make the model look better in the [Game] window. Depending on the situation, it may not be visible.

※ Main Light <auto-selected> – A Directional Light for filming.

※ Follow Camera – 'Main Light' is always the same position as Main Camera.

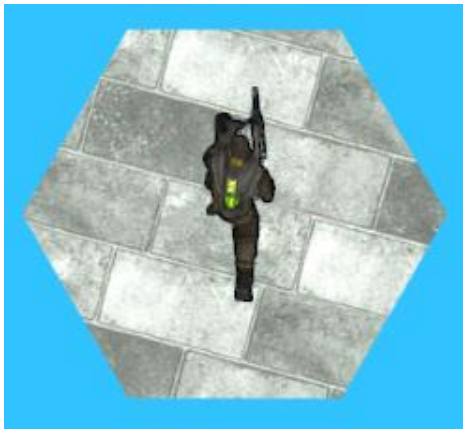
※ Position – The position of 'Main Light'. The light always looks at the model. This field is only shown when `Follow Camera` is checked.

※ Slope Angle – Look at the images below.



※ Show Tile – A tile is shown up under the model, which disappears during filming.

※ Tile Type – There are two types of tile.



※ Aspect Ratio – Depending on this ratio value, the slope angle is automatically calculated.

※ View Size – Views are created as many as this value(N) by repeatedly adding (360 / N) degrees to the previous one starting from the initial degree below.

※ Initial Degree – The range is 0 ~ (360 / N).

※ View name text field – The view name which is used for making files. If it is empty, (the initial degree + the degree of each view) is used for output files' name.

※ Checkboxes for each view – The final baking occurs for only selected views.

■ Apply – It rotates the model toward in that direction. It doesn't affect actual baking.

■ Select All & Unselect All – They selecting all views or unselecting all.

※ Shadow Type

▼ None <default> – No shadow.

▼ Simple – Oval-shaped shadow.

※ Scale – Scaling the shadow object.

■ Unify – It makes the scale ratio the same.

※ Dynamic Size – The shadow object is auto-sized during filming.

※ Opacity – Transparency of the shadow material.

※ Shadow Only – Only the shadow is taken without the model. It's not compatible with 'Exclude Shadow' of 'Variation'.

▼ Real - The shadow camera take a picture of the model on the texture under the model.

※ Method - There are two methods to implement a real shadow.

▼ Dynamic – Draws on the Render Texture simultaneously. This method doesn't support particle system models.

※ Camera <auto-created> – A shadow-only camera.

※ Field <auto-created> – A plane object wearing the shadow material.

'Opacity' and 'Shadow Only' are like 'Simple' type.

▼ Static – It's like 'Dynamic' but update the shadow PNG per every screenshot. It's slow

but could be an alternative when the model is not compatible with 'Dynamic' type such as particle system models.

'Camera', 'Field', 'Opacity' and 'Shadow Only' are like 'Dynamic' type.

■ Update & Hide – Clicking on 'Update' button shoots the model and makes a shadow image. Clicking on 'Hide' button hides the shadow object. It doesn't affect actual baking.

※ Extractor – It extracts pixel colors of the model from the background. I've created four Extractors.

By default, DefaultExtractor is used, and there is little need to change it for a typical opaque model.

Particle system models are translucent and difficult to extract precise colors. Especially, in case of an addictive shader, extracting colors is more difficult because particle colors are added to background colors. Each of them has a different method to extract particle colors. If you are curious about it, please refer to each code.

You can inherit ExtractorBase and implement Extract methods to create a custom extractor.

※ Variation – You can change the colors and transparency of the resulting image.

※ Tint Color – This color is mixed with all the pixel colors of the image.

▼ Tint & Sprite Blend Factor – Blending factors for tint color and image color.

※ Exclude Shadow – The shadow is are not blended with the tint color. It's not compatible with 'Shadow Only' of 'Shadow'.

▼ Body & Shadow Blend Factor – Blending factors for body color and shadow color.

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

※ Preview – When this property turned on, [Preview] window is seen.

Each time you modify any property, it is shot internally, which slows down overall. With the static shadow type, it worse.

※ Background – [Preview] window background type. There are a checker and a single color.

■ Update Preview – It updates [Preview] window manually, which is useful for shadow.

※ Resolution – The resolution of resulting sprites before trimming.

※ Frame Size – The number of frames to be taken per view.

※ Simulate – You can see the model at each frame.

※ Delay – There are times the model part and the shadow part of the resulting sprite don't match because of the speed gap between the screenshot speed and the speed in which the static shadow PNG file is updated. In that case, setting this value properly could fix the issue.

■ Sample – To select desired frames, you can take pictures of the model animation as many frames as the 'Frame Size' value.

For static models, only one frame is taken, so some properties can be invisible.

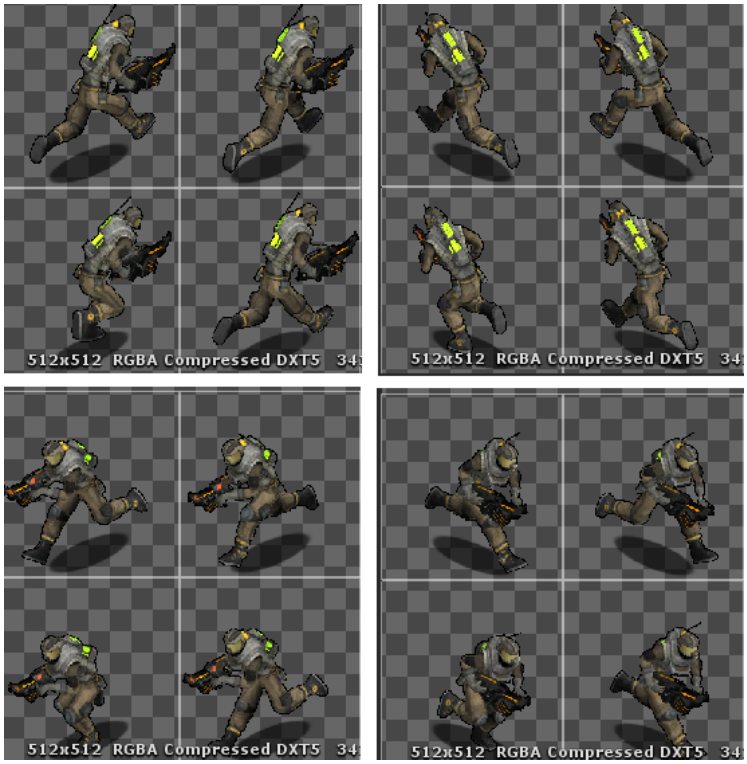
※ Use Trim – When this property is turned on, outer area of resulting sprites is cut.

※ Sprite Margin – The length of the area that must be preserved when trimming.

※ Unified Size – The size of sprites becomes identical.

※ Pivot-Symmetric – Sprites are trimmed symmetrically around the pivot.

※ for All Views – The size of all the sprites for all the selected views becomes identical.



※ Output Type – There are two types of output type.

▼ Sprite Sheet – PNG files are created that combines sprites each of which has a texture coordinate.

※ Packing Algorithm – There are two packing algorithms.

▼ Optimized – Unity built-in algorithm.

※ Max Size – The max size of sprite sheets.

▼ In Order – It arranges sprites in order from top-left.

※ Min Size – The min size of sprite sheets.

※ Padding – The interval length between each sprite.

※ Loop Animation Clip – Looping Animation Clip will be created.

※ All In One Atlas – All sprites are placed on a single sprite sheet. This property appears only for static models.

▼ Separately – One PNG file is created per sprite.

※ Auto File Naming – The front part of the name of output files is automatically written.

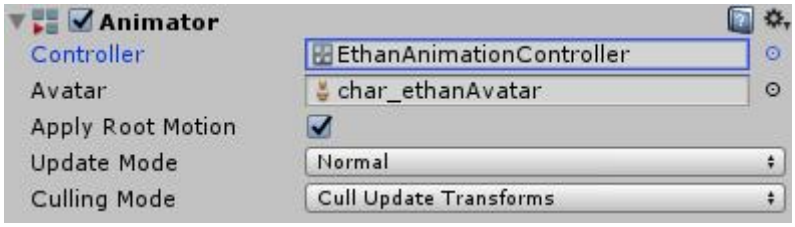
※ File Name – The front part of file names.

※ Output Directory – A path where output files being saved. It must be under the Asset folder.

■ Bake all frames or Bake selected frames – It bakes the model animation for the selected views as many as the 'Frame Size' value or the number of the selected frames.

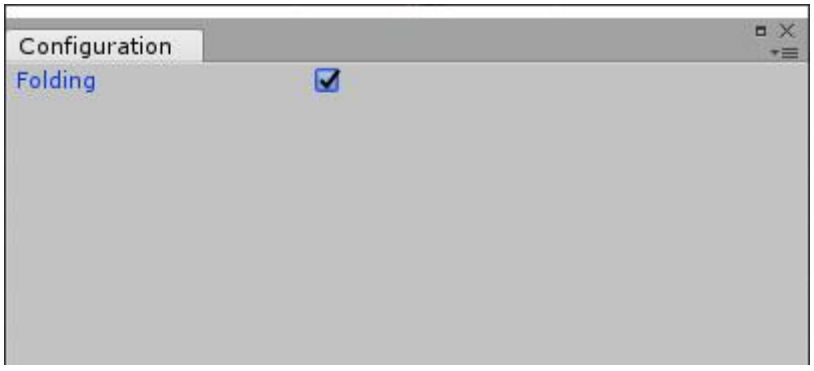
How to set up a mechanim model.

In the case of a mechanim model, the object must have an Animator component which has a specified Controller. If it does not, you must make an AnimationController and put it int the field.

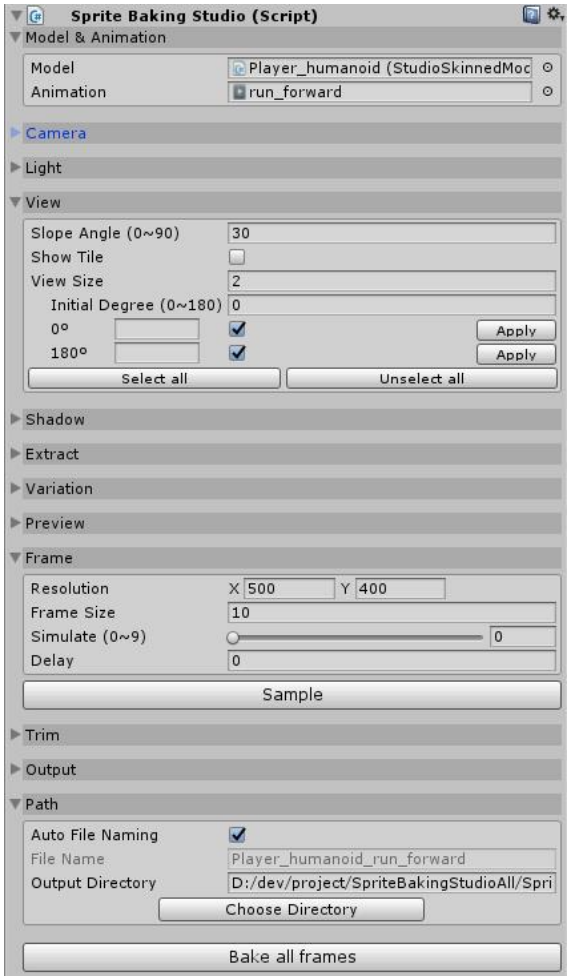


Configuration Window

If you click on Assets > Sprite Baking Studio > 'Configuration', the following window is shown.

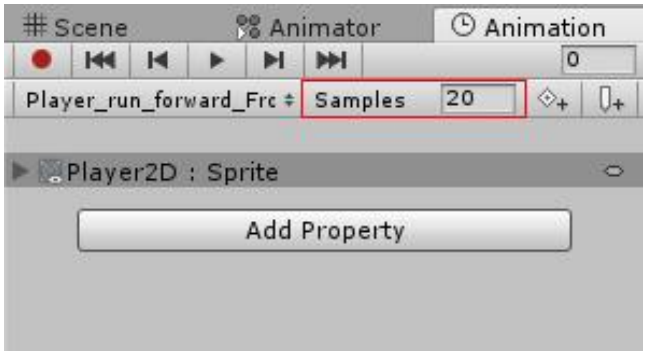


※ Folding – Field groups become foldable if checked.

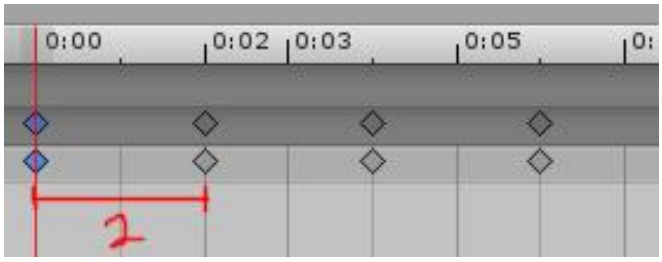


Preview Window

※ Samples – It is going to be 'Samples' of the resulting AnimationClip.

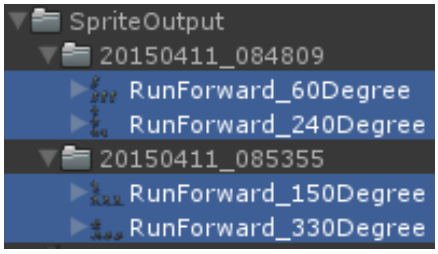


※ Sprite Interval – It is going to be interval time between frames of the resulting AnimationClip.



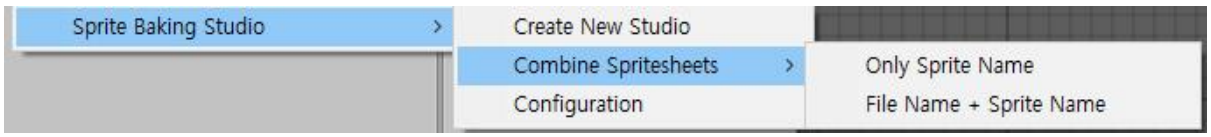
Combining sprite sheets

1. Select more than two sprite sheets.



2. Execute one of the sub menus under 'Assets > Sprite Baking Studio > 'Combine Spritesheets' in the main Menu. You can do this on the context menu as well.

If only one sprite sheet is selected or files of other type are selected, the two sub menu is disabled.



When each sprite name of a combined sprite sheet is determined,

※ Only Sprite Name: A sprite name of a selected sprite sheet is only used.

※ File Name + Sprite Name: A sprite name is appended to a selected sprite sheet name.

Then, selected sprite sheets are combined and saved in the directory where the first selected file is located.

Troubleshooting

* This asset includes Standard Assets. If your project already has 'Standard Assets' folder, some part of this asset could not be imported well.

* If some fonts are broken in [Inspector] window, change the encoding method of SpriteBakingStudioEditor.cs or StudioOOOModelEditor.cs to 'UTF-8' not 'UTF-8(BOM)'.

* If 'Build Setting' > 'Platform' is WebPlayer, change it to StandAlone.

* If 'Projection' of Main Camera or the shadow camera is 'Perspective', some functions couldn't work well.

Contacts

Name: Yeonuk Jo

Email: beggu84@naver.com