SpriteMan 3D Tools

Unity 5.6 added support for layered sprites using Sprite Renderer Sorting Groups. This is very useful for customizing characters by changing things like body parts, clothes and weapons.

Since this usually results in many more sprites and spritesheets needing to be made, SpriteMan comes with a set of special templates and productivity tools for quickly and easily processing hundreds of sprites.

These tools greatly speed up your workflow and simplify swapping the many sprites your games will use for multiple characters. Here's the main workflow these tools simplify.

> **PREPARE SPRITES** - First, prepare the sprite sheets that will be hooked up to your characters' SpriteManager components.
>
> **CREATE SPRITEMAN STATES** - Second, generate Sprite Manager State templates for customizable characters (body parts, clothing, weapons).
>
> **SWAP STATES AT RUNTIME** - Third, at runtime, you can now easily swap sets of hundreds of sprites with a few lines of code.

The templates and tools are described in the following sections.

Templates

- Sprite Sheet Map

  Contains the names and positions (top-left corner) of frames in a sprite sheet.

- Sprite Manager State

  The saved state of a SpriteManager you've configured in a scene.

- Sprite Sheet Batch

  Allows renaming frames in multiple sprite sheets at once.

- SpriteManager State Batch

  Allows creating multiple Sprite Manager States using a template SpriteManagerState and multiple sprites.

SpriteMan 3D Tools

- SpriteMan Tools Window

  Tools > SpriteMan 3D > SpriteMan Tools Window

- Save Sprite Sheet Names

  Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

- Replace Sprite Sheet Names

  Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

- Save SpriteMan State

  Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

- Convert Sprite Sheet To SpriteMan State

  Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

- Load SpriteMan State

  Loads a SpriteMan State to a SpriteManager in your scene.

Batching Templates (Not Available in Free Version)

- Sprite Sheet Batch

  Used to replace frame names in multiple sprite sheets, using a source SpriteSheetMap.

- SpriteMan State Batch

  Used to generate multiple SpriteMan States, using one source SpriteManState and multiple sprite sheets.

Common Use Cases

This last section describes some common uses of the SpriteMan tools and templates.

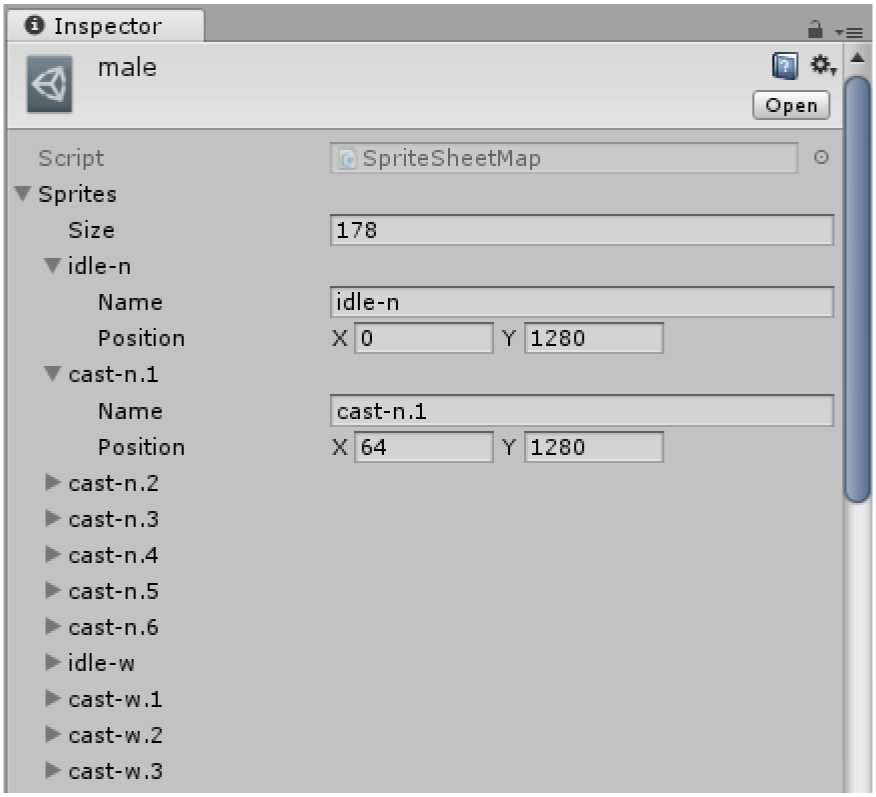
SpriteMan Templates

SpriteMan templates are used for saving the states of sprite sheets and SpriteManager components and for batch processing some commonly repetitive tasks. They will make more sense after reading the Tools and Batching sections.

Sprite Sheet Map

Contains the names and positions (top-left corner) of frames in a sprite sheet.

- Created using Save Sprite Sheet Names
- Used by Replace Sprite Sheet Names



Inspector

male

Open

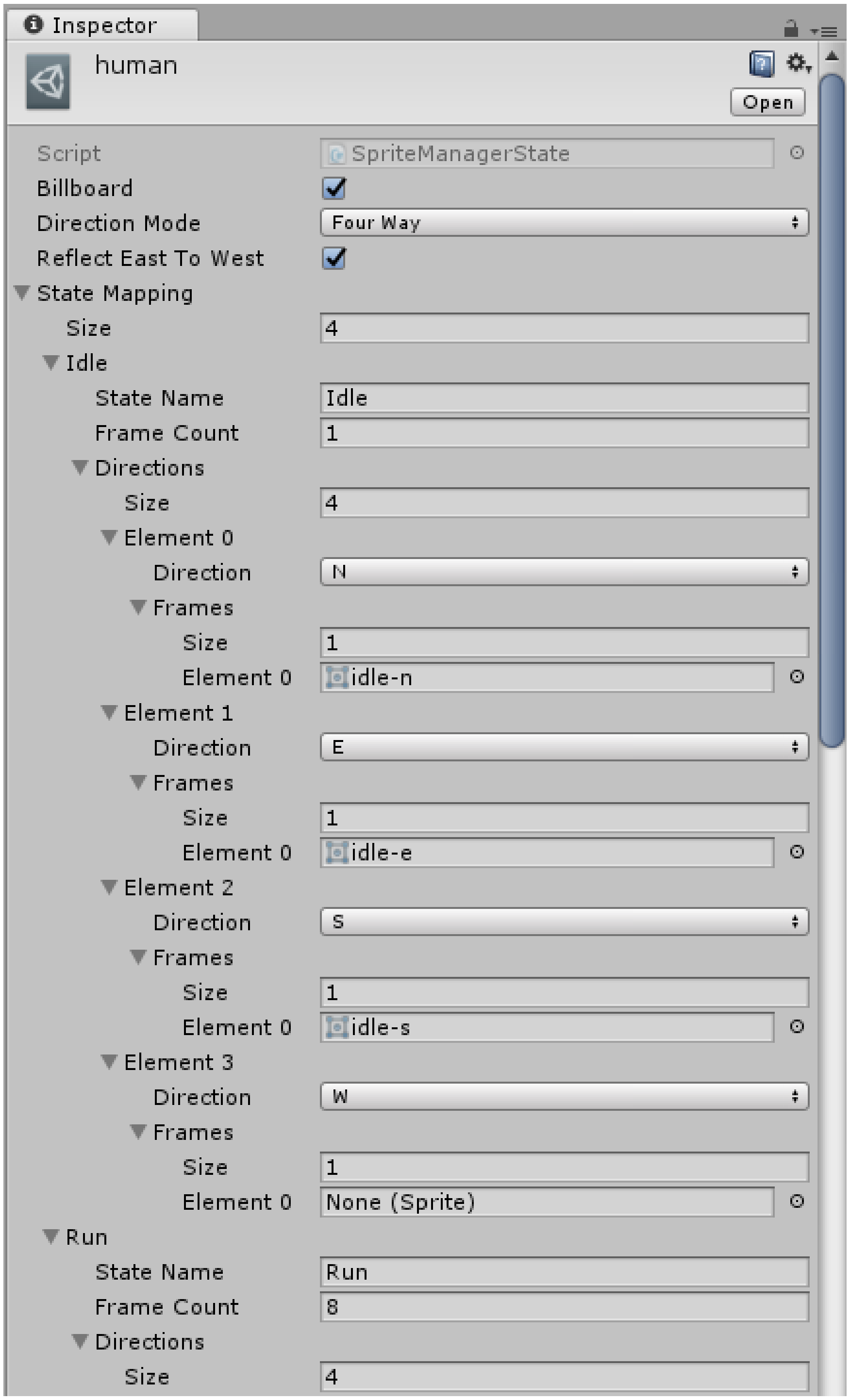
Script: SpriteSheetMap

Sprites
- Size: 178
- idle-n
  - Name: idle-n
  - Position: X 0 Y 1280
- cast-n.1
  - Name: cast-n.1
  - Position: X 64 Y 1280
- cast-n.2
- cast-n.3
- cast-n.4
- cast-n.5
- cast-n.6
- idle-w
- cast-w.1
- cast-w.2
- cast-w.3

SpriteManager State

Contains the states and sprites of SpriteManagers used in your scene. SpriteManager States are ScriptableObjects usable for quickly swapping SpriteManager content of characters in your scene.

- Created using Save SpriteMan State and Convert Sprite Sheet to SpriteMan State
- Used by Load SpriteMan State



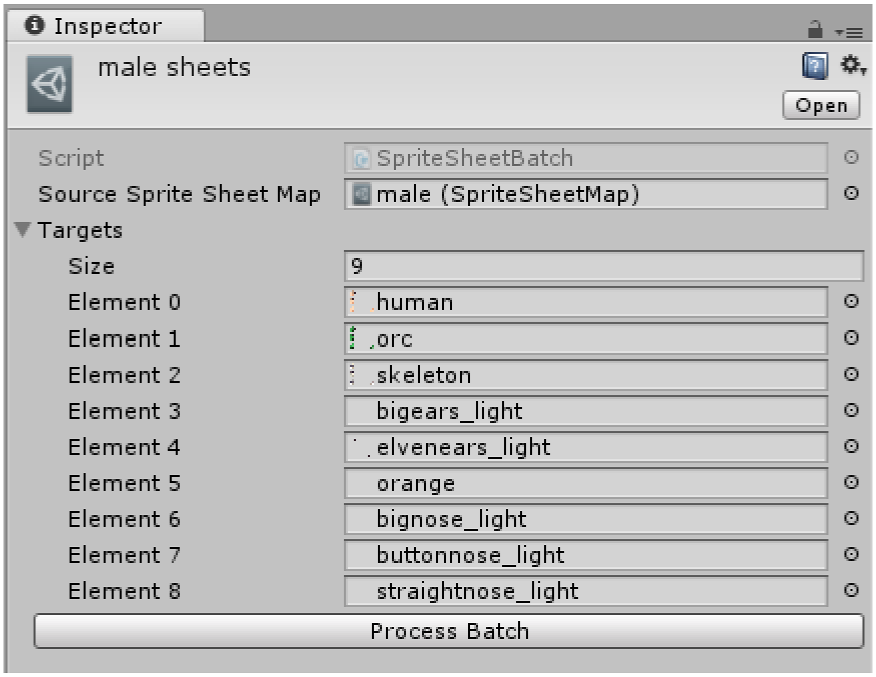
```
Inspector
human                                   Open

Script                  SpriteManagerState
Billboard               ☑
Direction Mode          Four Way
Reflect East To West    ☑
▼ State Mapping
    Size                4
  ▼ Idle
      State Name        Idle
      Frame Count       1
    ▼ Directions
        Size            4
      ▼ Element 0
          Direction     N
        ▼ Frames
            Size        1
            Element 0   idle-n
      ▼ Element 1
          Direction     E
        ▼ Frames
            Size        1
            Element 0   idle-e
      ▼ Element 2
          Direction     S
        ▼ Frames
            Size        1
            Element 0   idle-s
      ▼ Element 3
          Direction     W
        ▼ Frames
            Size        1
            Element 0   None (Sprite)
  ▼ Run
      State Name        Run
      Frame Count       8
    ▼ Directions
        Size            4
```

Sprite Sheet Batch (Not Available in Free Version)

Allows changing the frame names of multiple sprite sheets at once.

- Created using the Tools Menu. See Sprite Sheet Batching



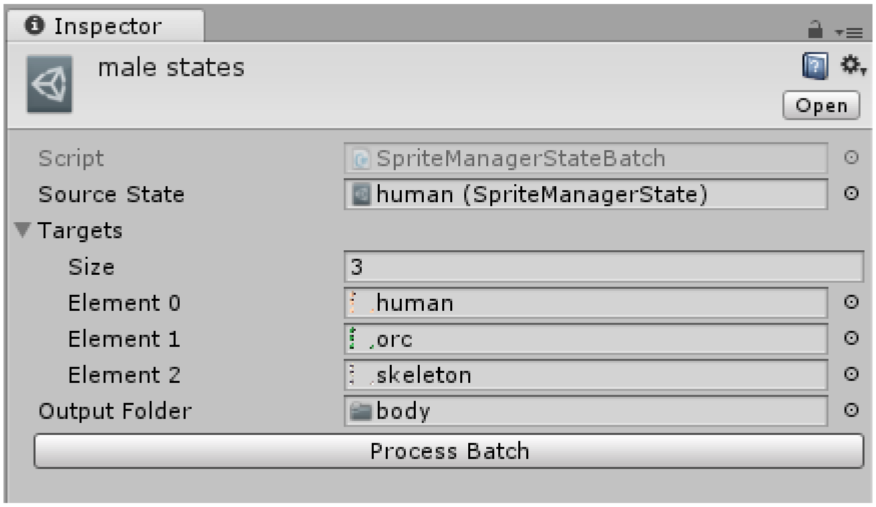
male sheets

Script	SpriteSheetBatch
Source Sprite Sheet Map	male (SpriteSheetMap)
Targets	
Size	9
Element 0	human
Element 1	orc
Element 2	skeleton
Element 3	bigears_light
Element 4	elvenears_light
Element 5	orange
Element 6	bignose_light
Element 7	buttonnose_light
Element 8	straightnose_light

SpriteManager State Batch (Not Available in Free Version)

Allows converting multiple sprite sheets to SpriteMan States.

- Created using the Tools Menu. See SpriteMan State Batching



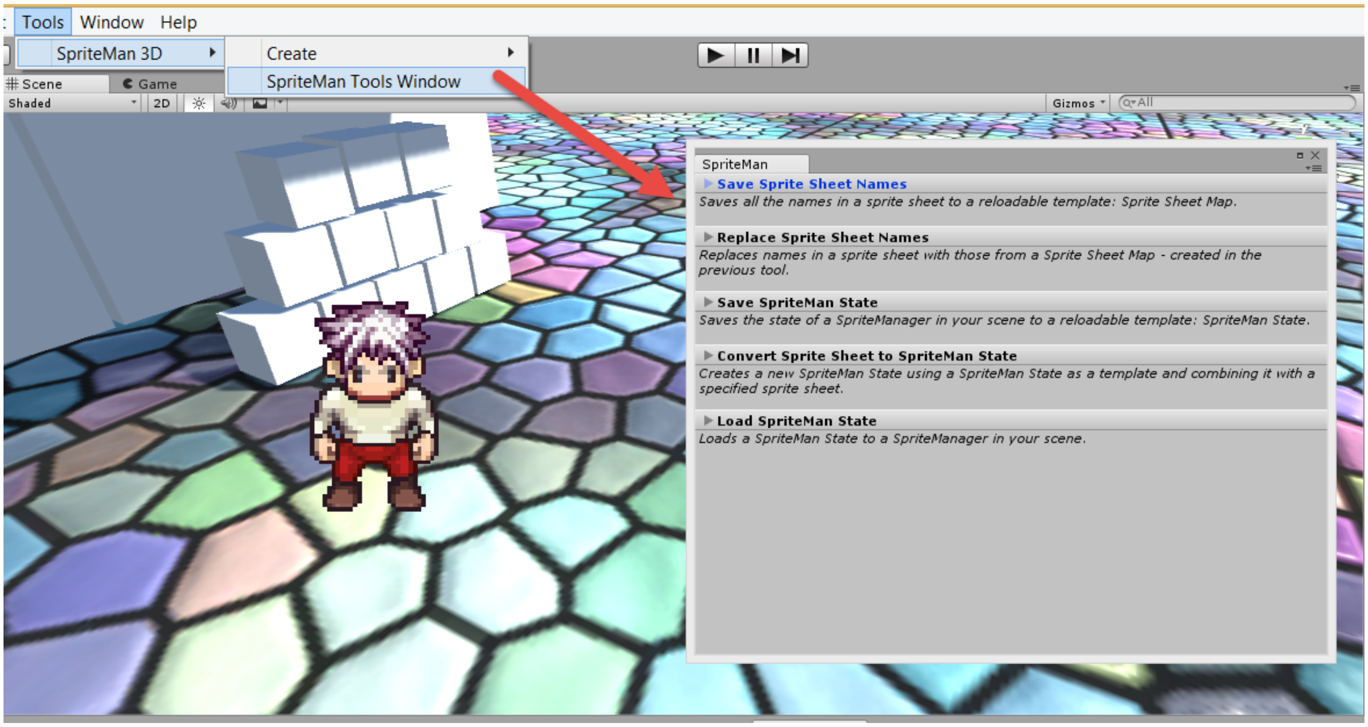
male states

Script	SpriteManagerStateBatch
Source State	human (SpriteManagerState)
Targets	
Size	3
Element 0	human
Element 1	orc
Element 2	skeleton
Output Folder	body

SpriteMan Tools

SpriteMan Tools Window

Open the **SpriteMan Tools Window** from the **Tools** menu: **Tools** > **SpriteMan 3D** > **SpriteMan Tools Window**



▸ **Save Sprite Sheet Names**
Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

▸ **Replace Sprite Sheet Names**
Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

▸ **Save SpriteMan State**
Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

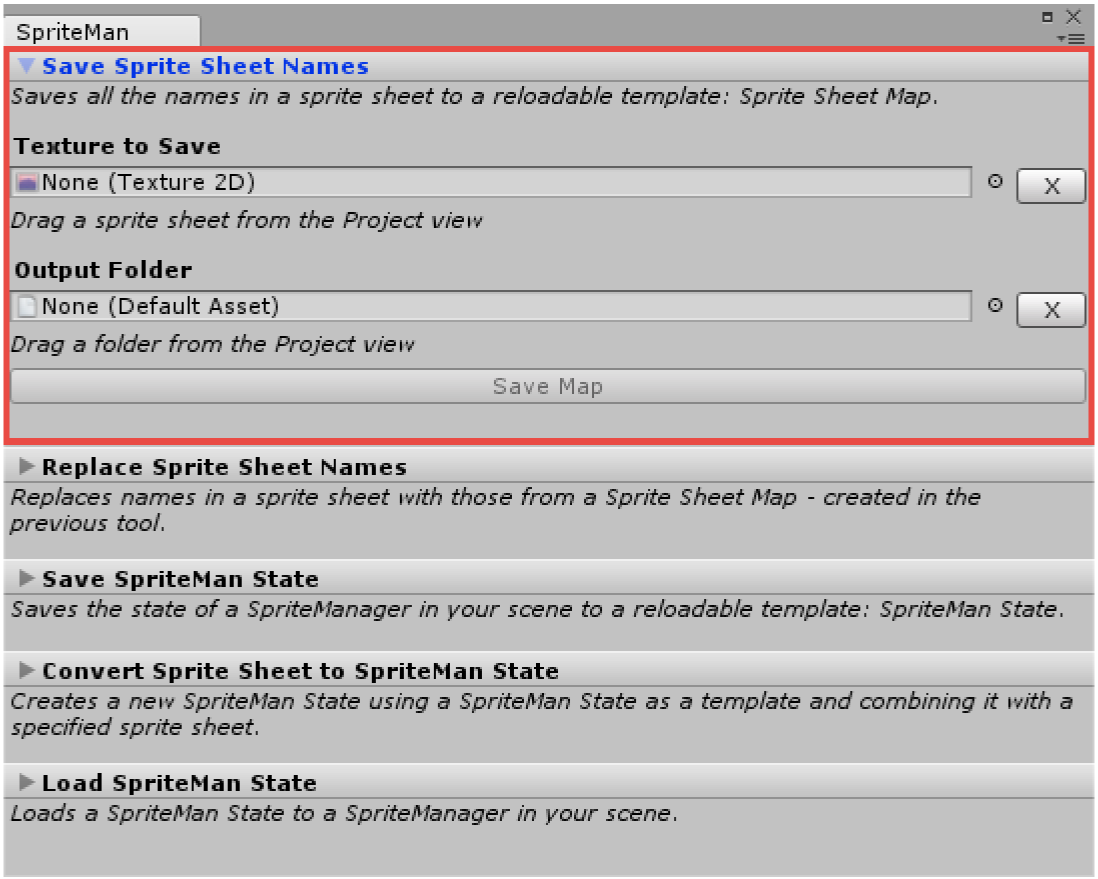
▸ **Convert Sprite Sheet to SpriteMan State**
Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

▸ **Load SpriteMan State**
Loads a SpriteMan State to a SpriteManager in your scene.

Save Sprite Sheet Names

This tool saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

This tool lets you save all the names of nicely named sprite sheets, so you can easily reload them later with a few clicks. This solves the problem of sprite sheet frame names getting reset when regenerating frames with new settings using Unity's Sprite Editor.



▾ **Save Sprite Sheet Names**
Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

Texture to Save
None (Texture 2D)
Drag a sprite sheet from the Project view

Output Folder
None (Default Asset)
Drag a folder from the Project view

▸ **Replace Sprite Sheet Names**
Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

▸ **Save SpriteMan State**
Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

▸ **Convert Sprite Sheet to SpriteMan State**
Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

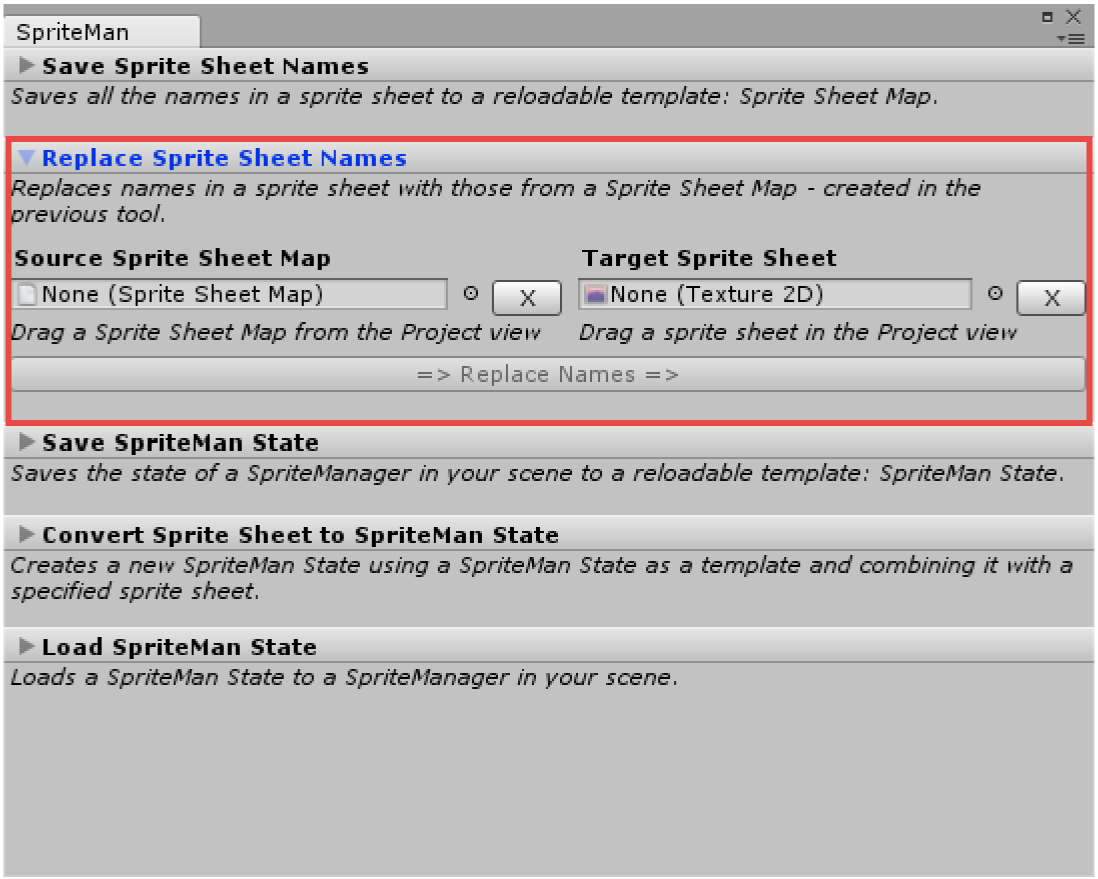
▸ **Load SpriteMan State**
Loads a SpriteMan State to a SpriteManager in your scene.

USAGE

- Set the **Texture to Save** with a sprite sheet from the Project view.
- Set the **Output Folder** with a folder from the Project view.
- Click **Save Map**.
- Inspect the newly created **Sprite Sheet Map**.

Replace Sprite Sheet Names

This tool replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.



▶ Save Sprite Sheet Names

Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

▼ Replace Sprite Sheet Names

Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

Source Sprite Sheet Map: None (Sprite Sheet Map)

Target Sprite Sheet: None (Texture 2D)

Drag a Sprite Sheet Map from the Project view

Drag a sprite sheet in the Project view

▶ Save SpriteMan State

Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

▶ Convert Sprite Sheet to SpriteMan State

Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

▶ Load SpriteMan State

Loads a SpriteMan State to a SpriteManager in your scene.

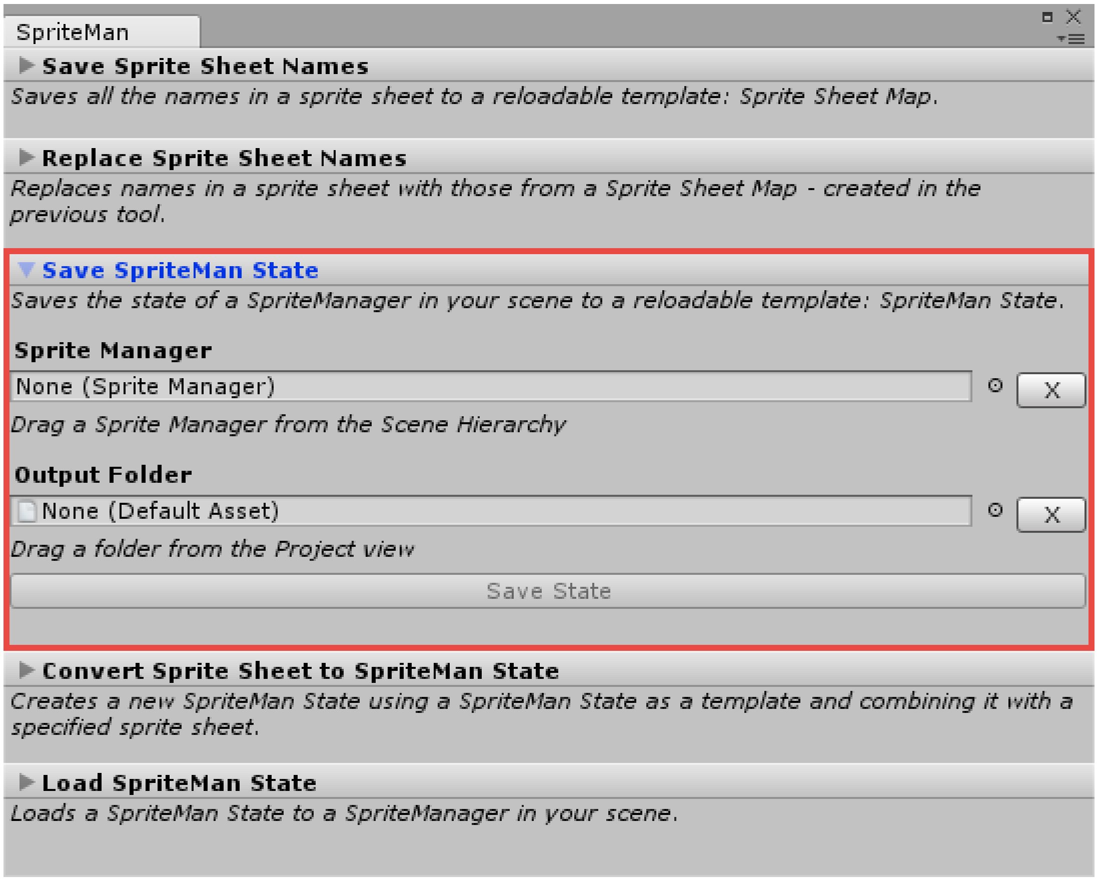
USAGE

- Set the **Source Sprite Sheet Map** from the Project view.
- Set the **Target Sprite Sheet** texture from the Project view.
- Click **Replace Names**.
- Inspect your target sprite sheet texture names.
- See the tool's generated output. Source frames that didn't have matching target frames will be listed. This happens when positions of frames don't match exactly. The output lets you easily see which frames have problems.

Having mismatched frames isn't always bad. In some cases, it's expected that certain frames of things like clothes, weapons or accessories will not have an existing frame. Those frames can safely be ignored.

Save SpriteMan State

This tool saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.



▶ **Save Sprite Sheet Names**

Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

▶ **Replace Sprite Sheet Names**

Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

▼ **Save SpriteMan State**

Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

Sprite Manager

None (Sprite Manager)

Drag a Sprite Manager from the Scene Hierarchy

Output Folder

None (Default Asset)

Drag a folder from the Project view

▶ **Convert Sprite Sheet to SpriteMan State**

Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

▶ **Load SpriteMan State**

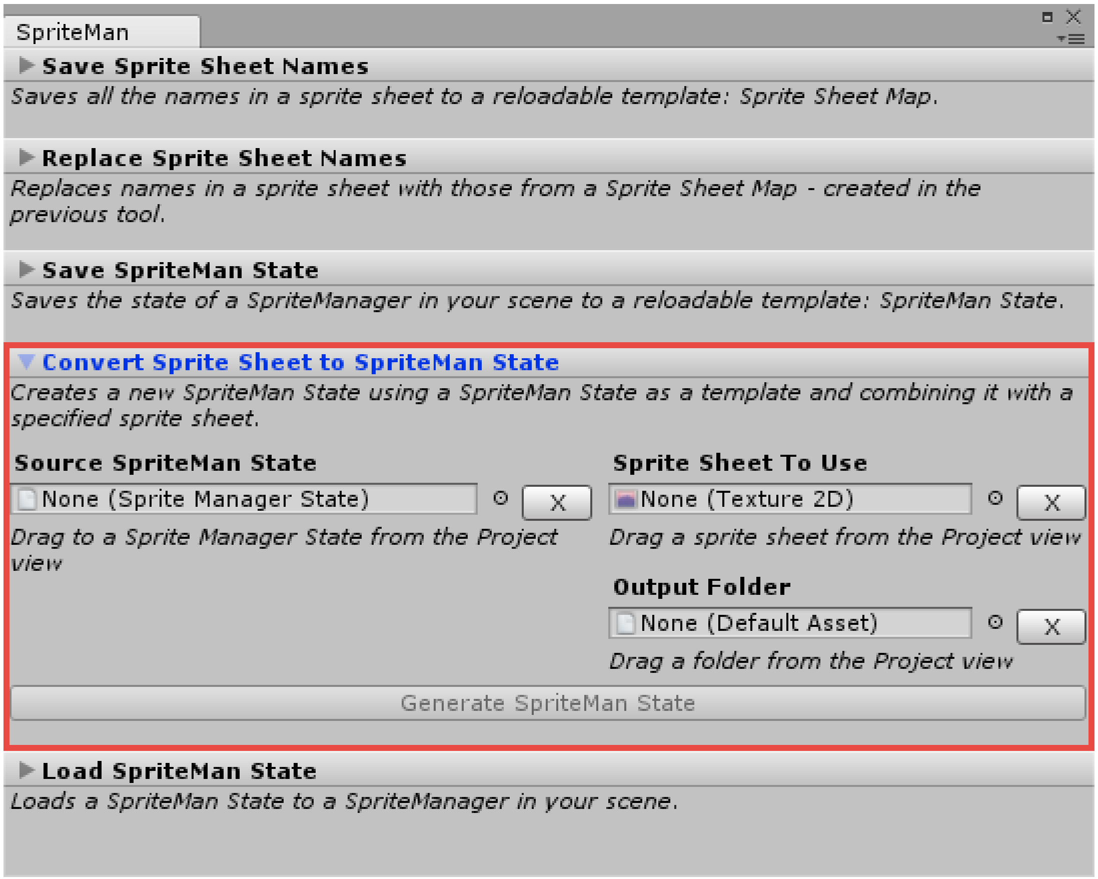
Loads a SpriteMan State to a SpriteManager in your scene.

USAGE

- Set the **Sprite Manager** with a Sprite Manager from your Scene hierarchy.
- Set the **Output Folder** with a folder from the Project view.
- Click **Save State**.
- Inspect the newly created **Sprite Manager State**.

Convert Sprite Sheet to SpriteMan State

This tool creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.



SpriteMan

▶ **Save Sprite Sheet Names**

Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

▶ **Replace Sprite Sheet Names**

Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

▶ **Save SpriteMan State**

Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

▼ **Convert Sprite Sheet to SpriteMan State**

Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

Source SpriteMan State

None (Sprite Manager State)

Drag to a Sprite Manager State from the Project view

Sprite Sheet To Use

None (Texture 2D)

Drag a sprite sheet from the Project view

Output Folder

None (Default Asset)

Drag a folder from the Project view

▶ **Load SpriteMan State**

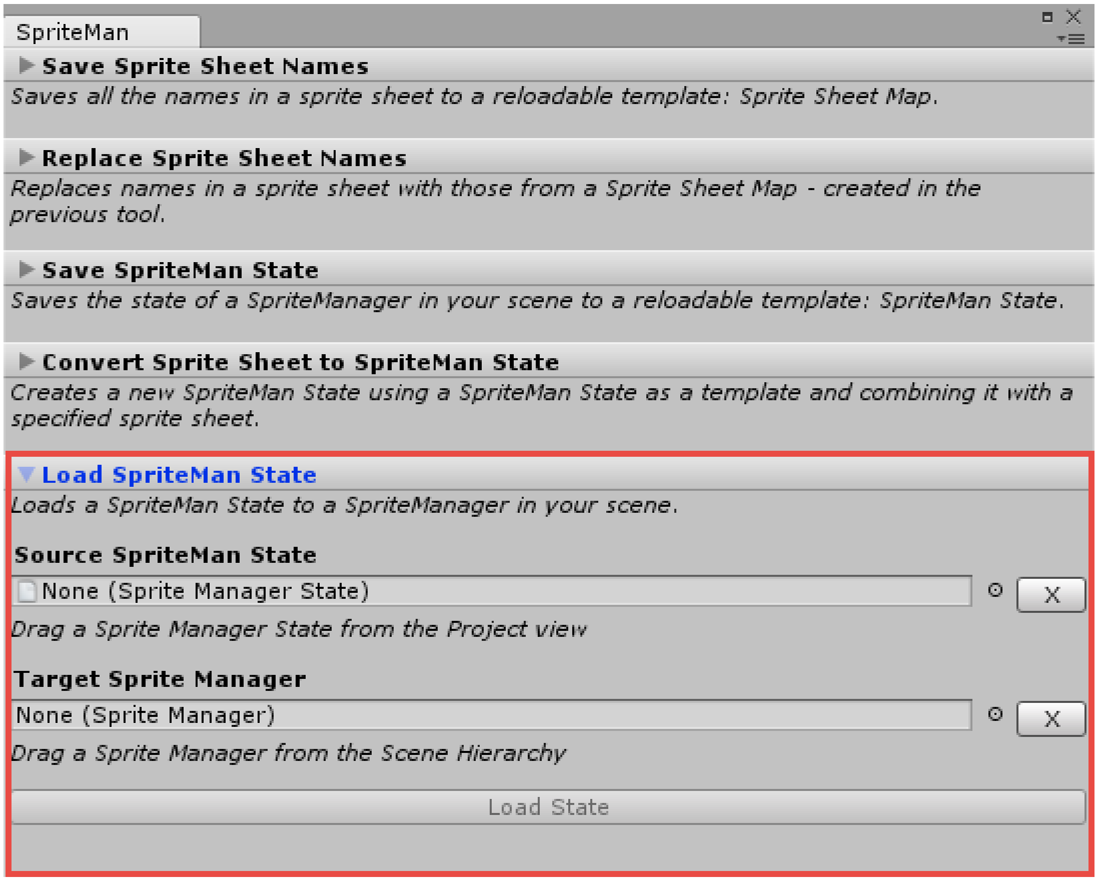
Loads a SpriteMan State to a SpriteManager in your scene.

USAGE

- Set the **Source SpriteMan State** with a SpriteManager State from the Project view.
- Set the **Sprite Sheet To Use** with a sprite sheet from the Project view.
- Set the **Output Folder** with a folder from the Project view.
- Click **Generate SpriteMan State**.
- Inspect the newly created **SpriteMan State**.

Load SpriteMan State

This tool loads a SpriteMan State to a SpriteManager in your scene.



Save Sprite Sheet Names

Saves all the names in a sprite sheet to a reloadable template: Sprite Sheet Map.

Replace Sprite Sheet Names

Replaces names in a sprite sheet with those from a Sprite Sheet Map - created in the previous tool.

Save SpriteMan State

Saves the state of a SpriteManager in your scene to a reloadable template: SpriteMan State.

Convert Sprite Sheet to SpriteMan State

Creates a new SpriteMan State using a SpriteMan State as a template and combining it with a specified sprite sheet.

Load SpriteMan State

Loads a SpriteMan State to a SpriteManager in your scene.

Source SpriteMan State

None (Sprite Manager State)

Drag a Sprite Manager State from the Project view

Target Sprite Manager

None (Sprite Manager)

Drag a Sprite Manager from the Scene Hierarchy

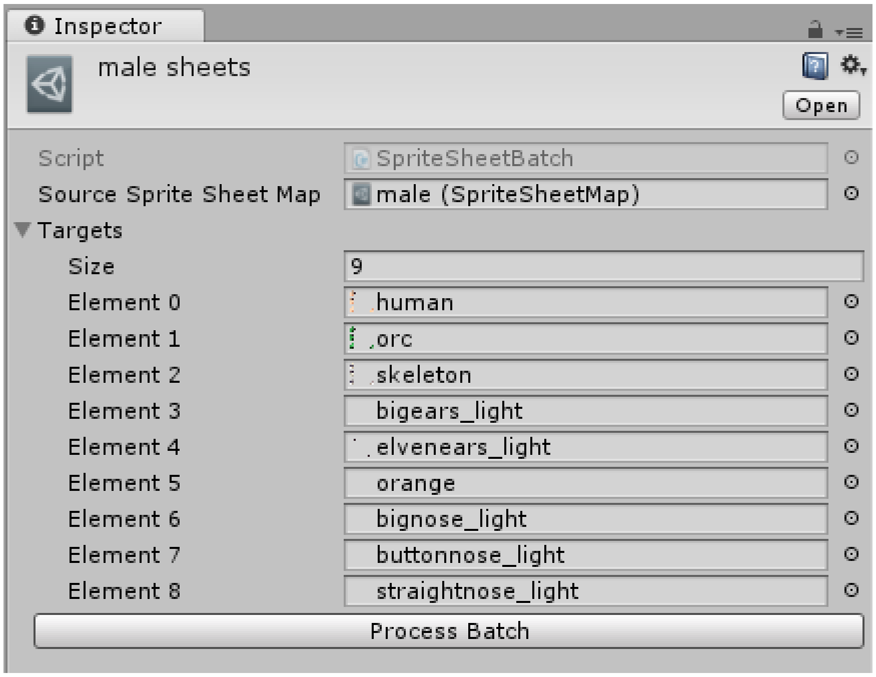
USAGE

- Set the **Source SpriteMan State** with a SpriteMan Staet from the Project view.
- Set the **Target Sprite Manager** with a Sprite Manager from your Scene hierarchy.
- Click **Save Map**.
- Inspect the newly created **Sprite Sheet Map**.

Template Batching (Not Available in Free Version)

Sprite Sheet Batching

Using a Sprite Sheet Batch, you can rename multiple sprite sheets at once.



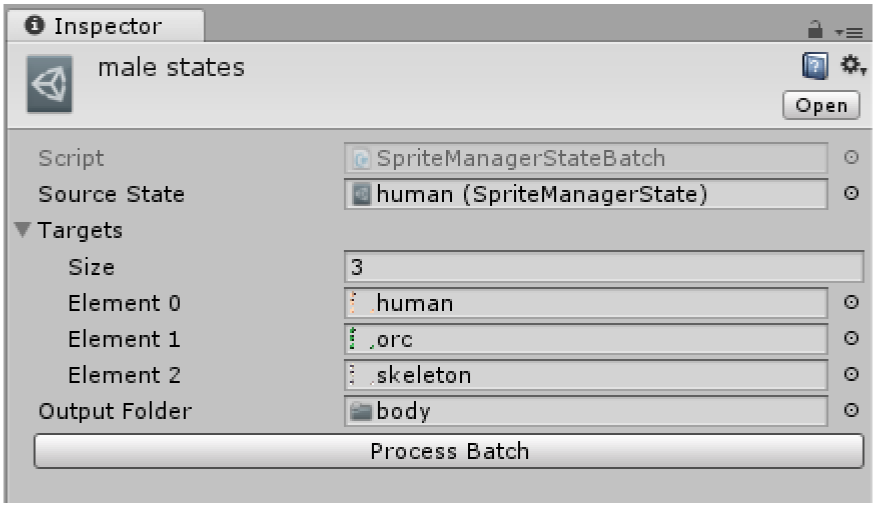
Script	SpriteSheetBatch
Source Sprite Sheet Map	male (SpriteSheetMap)
Targets	
Size	9
Element 0	human
Element 1	orc
Element 2	skeleton
Element 3	bigears_light
Element 4	elvenears_light
Element 5	orange
Element 6	bignose_light
Element 7	buttonnose_light
Element 8	straightnose_light

USAGE

- Create a new Sprite Sheet Batch from the Tools menu. (Tools > SpriteMan 3D > Create > Sprite Sheet Batch)
- Set the **Source Sprite Sheet Map** with a Sprite Sheet Map from your Project view.
- Set multiple **Targets** with sprite sheets from your Project view.
- Click **Process Batch** any time you wish to rename all the sprite sheets.
- Inspect the renamed sprite sheets.

SpriteMan State Batching

Using a SpriteMan State Batch, you can generate multiple SpriteMan states at once.



Script	SpriteManagerStateBatch
Source State	human (SpriteManagerState)
Targets	
Size	3
Element 0	human
Element 1	orc
Element 2	skeleton
Output Folder	body

USAGE

- Create a new SpriteMan State Batch from the Tools menu. (Tools > SpriteMan 3D > Create > SpriteMan State Batch)
- Set the **Source State** with a SpriteMan State from your Project view.
- Set multiple **Targets** with a sprite sheets from your Project view.
- Set the **Output Folder** with a folder from the Project view.
- Click **Process Batch** whenever you wish to regenerate your states.
- Inspect the newly created **SpriteMan States**.

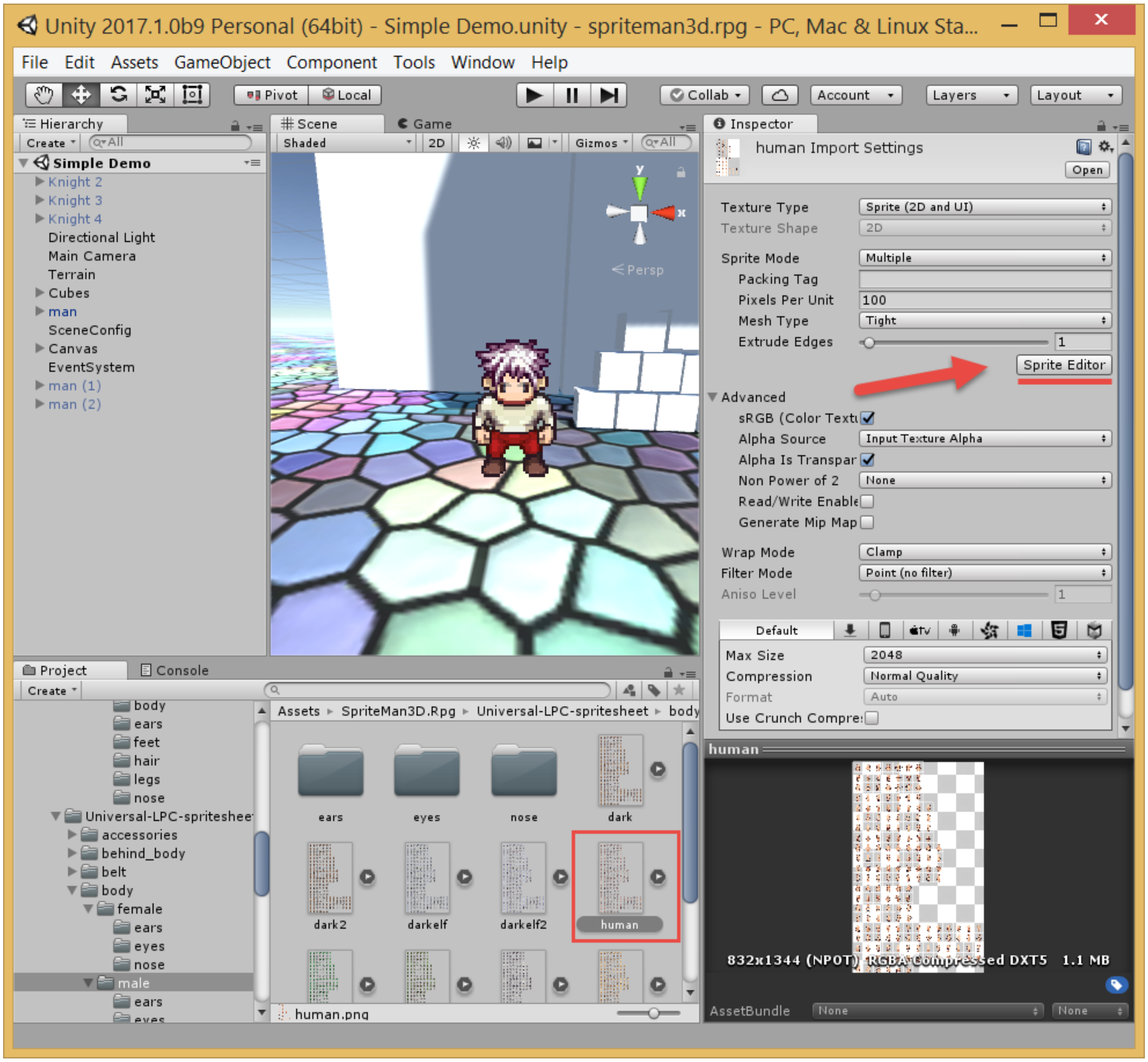
Common Use Cases

This section contains the following subsections:

- Scenario: Easily rename sprite sheet frames after Unity's Sprite Editor resets sprite sheet frame names.
- Scenario: You want to quickly name the frames of multiple spritesheets (ex: clothes, accessories, weapons).
- Scenario: Create multiple SpriteMan States for different sets of clothes.

Scenario: Easily rename sprite sheet frames after Unity's Sprite Editor resets sprite sheet frame names.

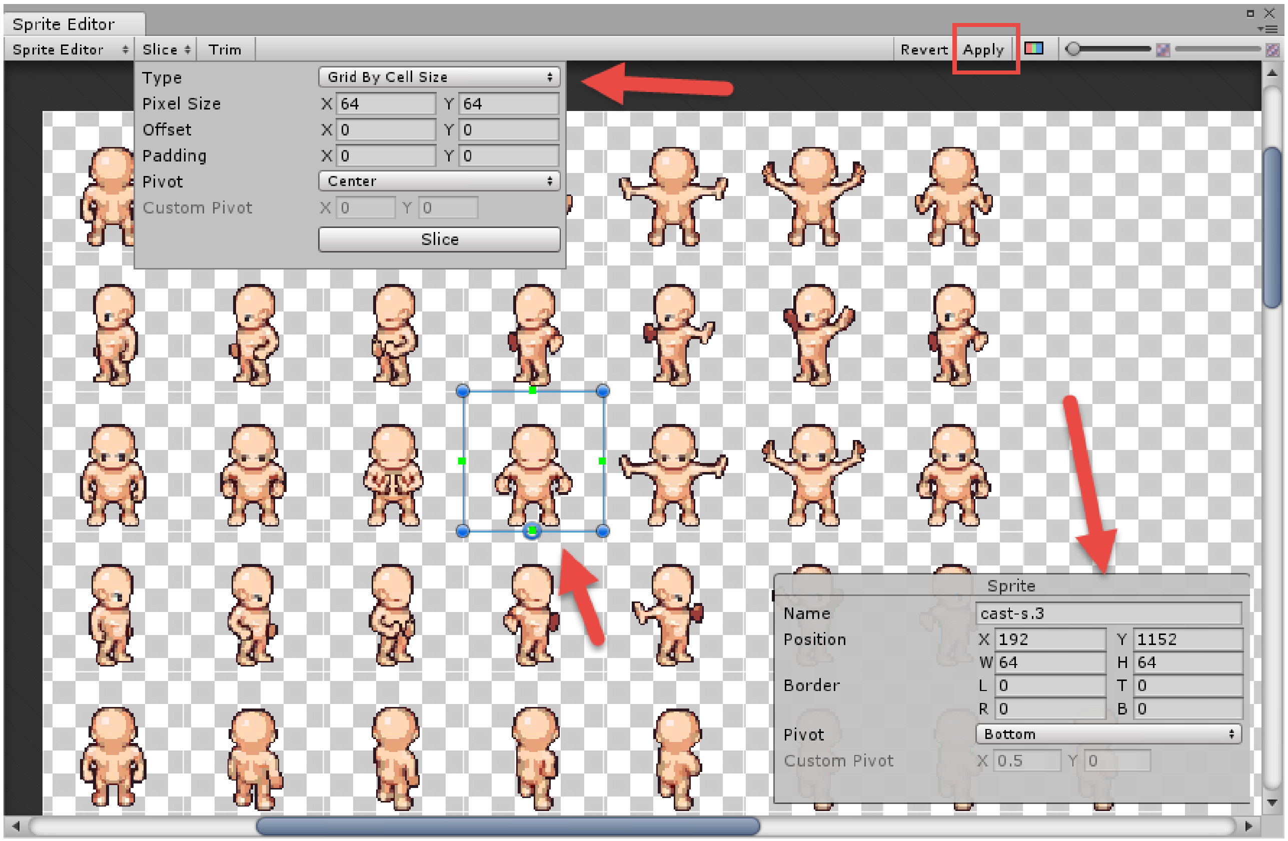
- Imagine you generate frames for a sprite sheet using Unity's Sprite Editor.



human Import Settings

Texture Type	Sprite (2D and UI)
Texture Shape	2D
Sprite Mode	Multiple
Packing Tag	
Pixels Per Unit	100
Mesh Type	Tight
Extrude Edges	1
Advanced	
sRGB (Color Textu	☑
Alpha Source	Input Texture Alpha
Alpha Is Transpar	☑
Non Power of 2	None
Read/Write Enable	□
Generate Mip Map	□
Wrap Mode	Clamp
Filter Mode	Point (no filter)
Aniso Level	1
Default	
Max Size	2048
Compression	Normal Quality
Format	Auto
Use Crunch Compre:	□

- You slice the sprite sheet using the following settings.



Type: Grid By Cell Size
Pixel Size: X 64 Y 64
Offset: X 0 Y 0
Padding: X 0 Y 0
Pivot: Center
Custom Pivot: X 0 Y 0

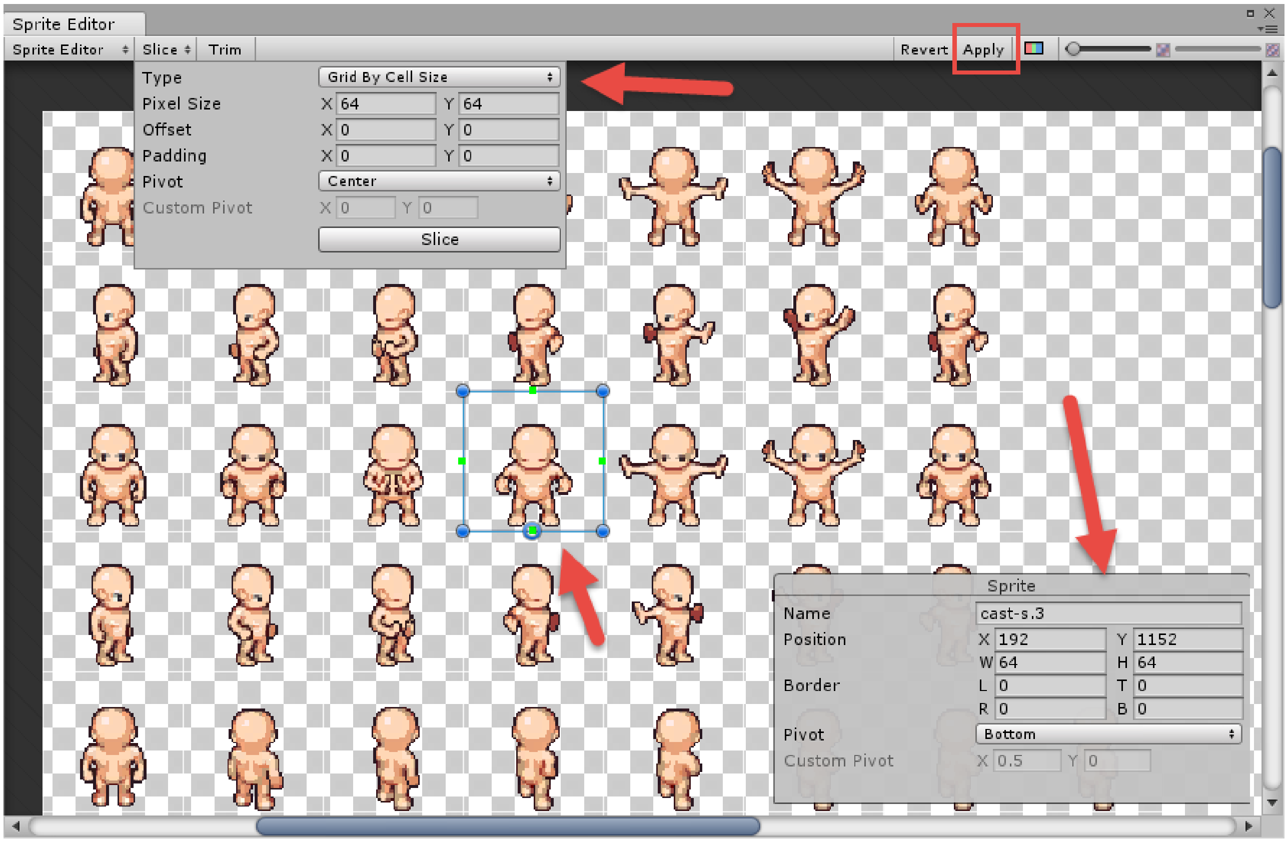
Name: cast-s.3
Position: X 192 Y 1152 W 64 H 64
Border: L 0 T 0 R 0 B 0
Pivot: Bottom
Custom Pivot: X 0.5 Y 0

- The frame names generated are in the form sheetName_0, sheetName_1.
- So, you rename your frames individually to be more readable (ex: idle-e, walk-e.1, walk-e.2).
- **Now use the Save Sprite Sheet Names tool to save all your sprite sheet names.**
- After using this sprite sheet a bit, you decide you'd like to change the frame pivot positions (say from Bottom to Center).
- You use Unity's Sprite Editor to make the change by regenerating the frames with new settings.
- This has reset all the frame names to defaults (sheetName_0, sheetName_1).
- **Use the Replace Sprite Sheet Names tool to reload your saved names.**

  (Without these tools, you have to manually rename your frames to their readable forms again - all 150 of them.)

Scenario: You want to quickly name the frames of multiple spritesheets (ex: clothes, accessories, weapons).

- Imagine you slice a sprite sheet using the following settings.



Type: Grid By Cell Size
Pixel Size: X 64 Y 64
Offset: X 0 Y 0
Padding: X 0 Y 0
Pivot: Center
Custom Pivot: X 0 Y 0

Name: cast-s.3
Position: X 192 Y 1152
W 64 H 64
Border: L 0 T 0
R 0 B 0
Pivot: Bottom
Custom Pivot: X 0.5 Y 0

- The frame names generated are in the form sheetName_0, sheetName_1.
- So, you rename your frames individually to be more readable (ex: idle-e, walk-e.1, walk-e.2).
- **Now use the Save Sprite Sheet Names tool to save all your sprite sheet names.**
- Say you have multiple sprite sheets containing various sets of clothing that will overlay the main body sprite.
- **Use the Replace Sprite Sheet Names tool to quickly replace the names of the additional sprite sheets.**
- **Alternatively, use a Sprite Sheet Batch to replace the names of multiple sprite sheets with one click.**

Scenario: Create multiple SpriteMan States for different sets of clothes.

There's a sample scene for this in the asset's Demo Scenes folder: Clothing Demo.

- This example has a rigged SpriteManager character with a base body.
- On top of that are displayed multiple layers with swappable clothes and weapons.
- To reproduce this with your own sprites, first, slice your sprite sheets.
- **Then rename the frames of a main body sprite sheet and use the Save Sprite Sheet Names tool to save your names.**

- **Next, use a Sprite Sheet Batch to rename your sprite sheet frames for all the other sprite sheets.**

- **Use a SpriteMan State Batch to generate all the swappable sets of clothes.**

- Overlay the new clothes using child SpriteManagers that render on top of the body SpriteManager.

  > ⚠ **Important**
  >
  > Make sure to use a SortingGroup for the layers to work correctly.

- **In the editor, use the Load SpriteMan State tool to test the new clothes on your character.**

- At runtime, you can load SpriteMan States from a Resources folder and swap them using the SpriteManager.LoadSpriteManState() method.

  (There's also a different working example in the demo scene.)

Swapping SpriteMan States

```
/// <summary>
/// This behaviour allows changing a character's clothes.
/// </summary>
public class StateSwap : MonoBehaviour
{
    private SpriteManagerState[] clothes;

    public SpriteManager clothingSpriteManager;

    void Start()
    {
        // load SpriteManagerStates at runtime from a Resources folder.
        clothes = new[]
        {
            Resources.Load<SpriteManagerState>("clothes/leather"),
            Resources.Load<SpriteManagerState>("clothes/mail"),
            Resources.Load<SpriteManagerState>("clothes/armor")
        };
    }

    /// <summary>
    /// This method can be called from a UI button.
    /// </summary>
    /// <param name="index"></param>
    public void ChangeClothes(int index)
    {
        if (clothingSpriteManager != null && index < clothes.Length)
        {
            // swap a SpriteManager's state using SpriteManager.LoadSpriteM
            clothingSpriteManager.CopyFrom(clothes[index]);
        }
    }
}
```