Animator Sprite Swap

Animator Sprite Swap system allows users to reuse their existing editor animation assets for objects that share the animation system structure but differ in sprite data, by overriding the sprites set by the animator during runtime with sprites from user-specified sources.

Why use the **Animator Sprite Swap** system?

- You will be able to share the same Animator Controller and Animation Clip assets for a variety of different animated objects, without having to remake the animation assets for each character
- If you are making multi-directional animations, you will no longer have to set up 4 - 8 different Animation Clips per animation for each of the character facing directions
- You will be able to change between different spritesheets during runtime with ease

How it works:

1. User selects an Animator Controller asset which he wants to reskin
2. The tool goes through all of the Animation States found in the user selected Animator Controller asset to collect every Animation State
3. The tool saves all the sprite-related keyframe data for every Animation Clip which is referenced by the Animator States
4. The tool goes through all the sprite references to find their source texture assets (original spritesheets)
5. User selects the replacement textures/spritesheets for the original texture assets
6. The tool finds each sprite‘s subasset index in its original texture asset, which is then used to find a replacement sprite from the user selected replacement texture (e. g. if an Animation Clip references the fifth sprite in a texture/spritesheet asset in one of the animation keyframes, it will find the fifth sprite from the user selected replacement texture/ spritesheet)
7. All Animation States with keyframe data and replacement sprites are saved in a scriptable object asset which can be assigned to a Reskin script component to override object‘s Animator data during runtime
8. During runtime, Reskin script component automatically overrides the sprites set by the Animator component with ones from the generated Reskin asset

Simple Animator Reskin tutorial

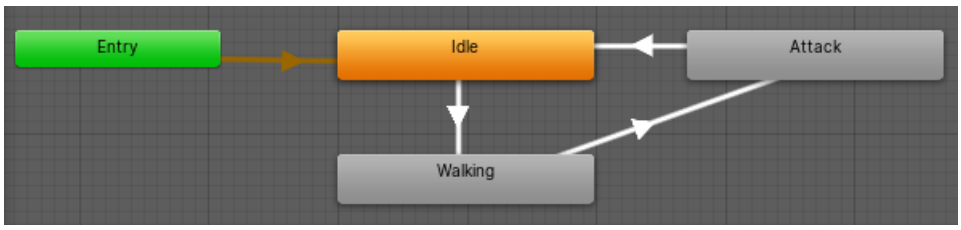
Simple **Animator Reskin** assets are useful when you have simple sprite animations with no alternate animations for your animated object.

Video tutorial is available here: https://www.youtube.com/watch?v=3SD9iFN4B8U

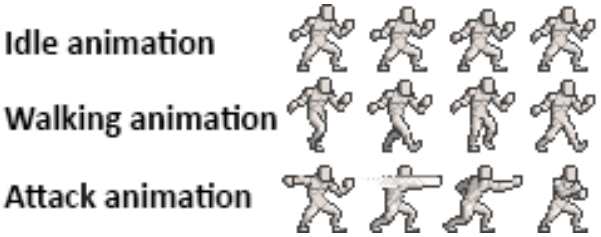
Preparation

To begin using the Animator Sprite Swap system you will need to have an existing Animator Controller asset with some Animation States.

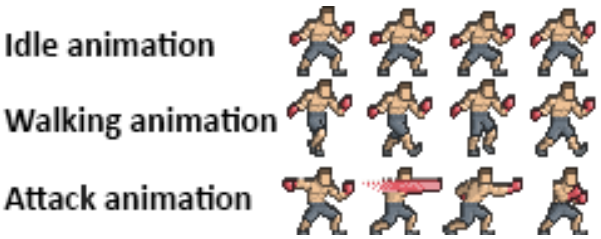
For this example we will be using a simple Animator Controller setup with 3 Animation States:



Sprites used for these animations can be found in the same spritesheet:



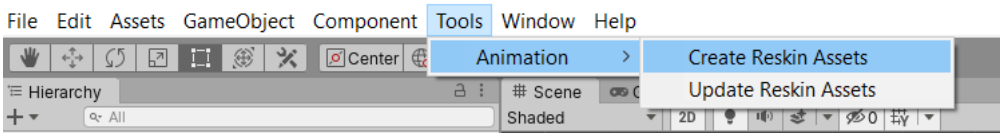
In this tutorial we will be replacing this spritesheet with the one below:



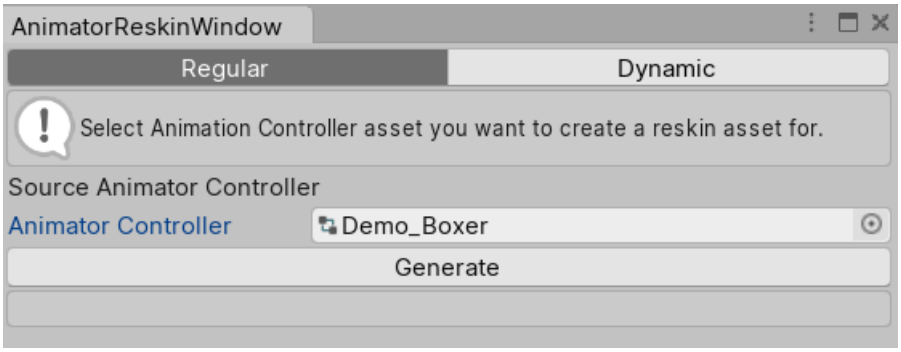
Before that, make sure both spritesheets are sliced similarly using the Sprite Editor. Sprite count and sprite order needs to be the identical, otherwise the Reskin asset might not be generated as expected.

Creating an Animator Reskin asset

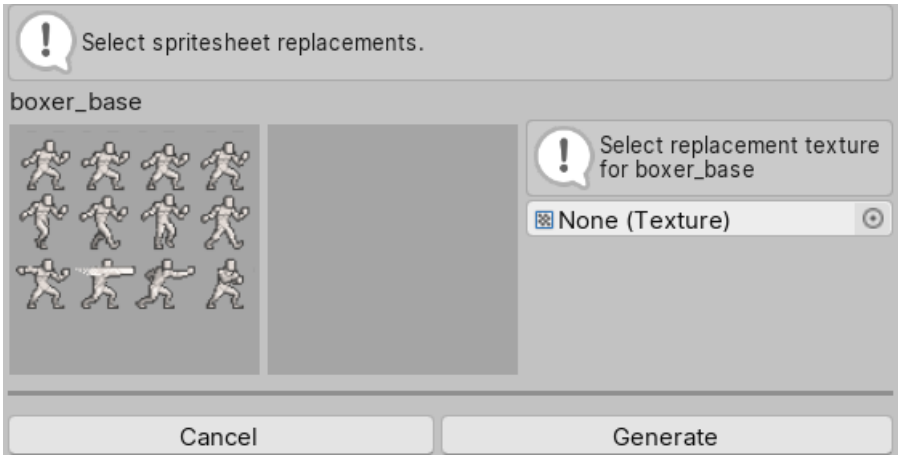
1. Open **Tools->Animation->Create Reskin Assets**



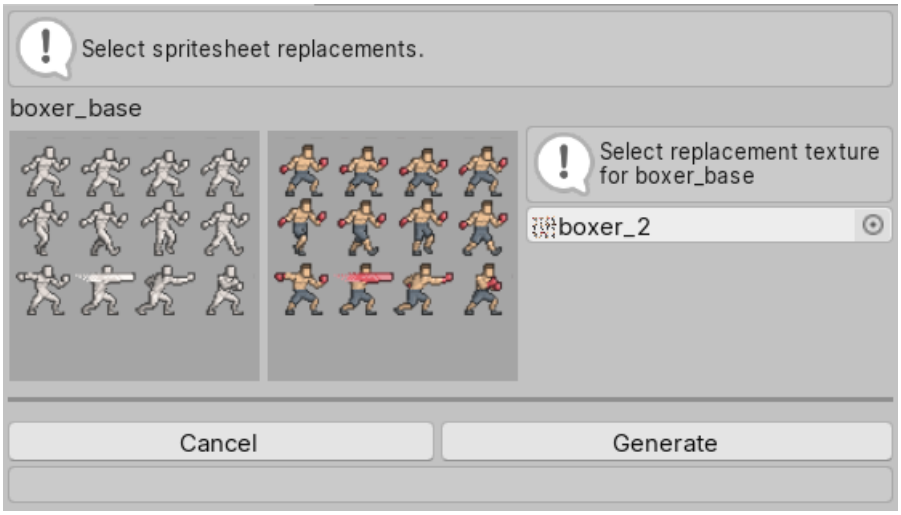
2. Select your character's **Animation Controller** asset you want to create a **Reskin asset** for



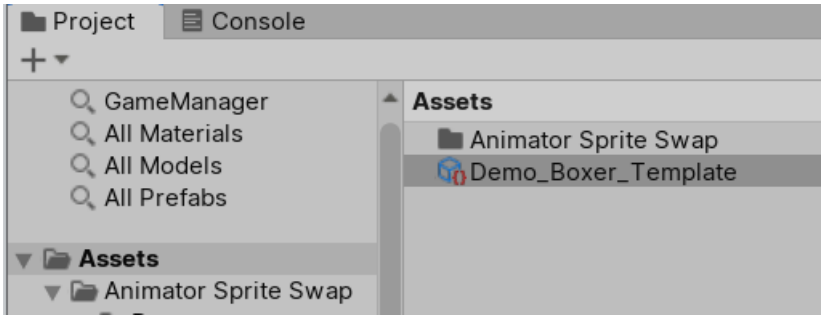
3. Click on **Generate**
   You will be shown a list of spritesheets that are used for the Animations found in the Animation controller asset



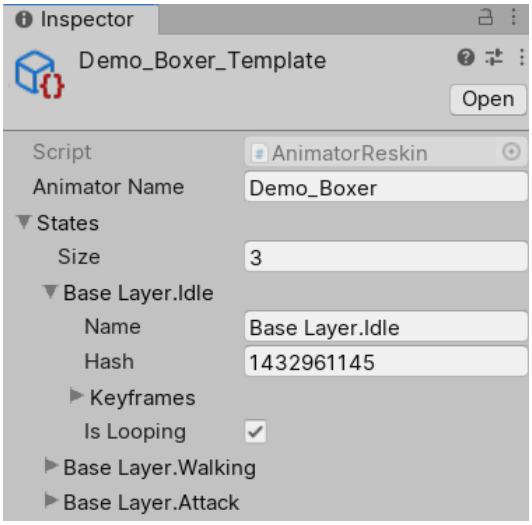
4. Select the sliced spritesheet texture you want to replace the original with



5. Click on **Generate**
An Animator Reskin asset should be generated in /Assets/ directory by the name of your Animator Controller.

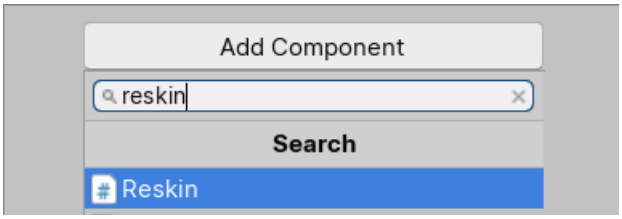


Inside it, you will find all of the states in the Animation Controller that were found in the Animator Controller asset. Each state in the Reskin asset contains keyframe data and sprite. Each Animation State will be labled by its respective directory in the Animator Controller (e.g. “Idle” state in the “Base Layer” layer will be named “Base Layer.Idle”)

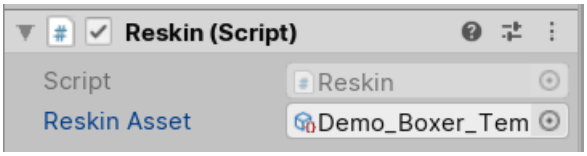


Setting up your character and animations

1. Add a **Reskin** script component to a gameobject/prefab you want to reskin



2. Assign the **Animator Reskin** asset you previously created to the **Reskin** component's **Reskin Asset** property



Once you are finished with these steps, upon entering Play mode your character should be using the sprites from the Animator Reskin assigned in Reskin component for it's animations:

Dynamic Animator Reskin tutorial

Animator Reskin Dynamic assets are useful when you have multiple alternate sprite animations for the same animated object. One example could be for top-down multi-directional animations where animations need to have variations for all directions (top, down, left, right), another example could be simply having alternative animations for different character states (e.g. normal, injured, very injured).

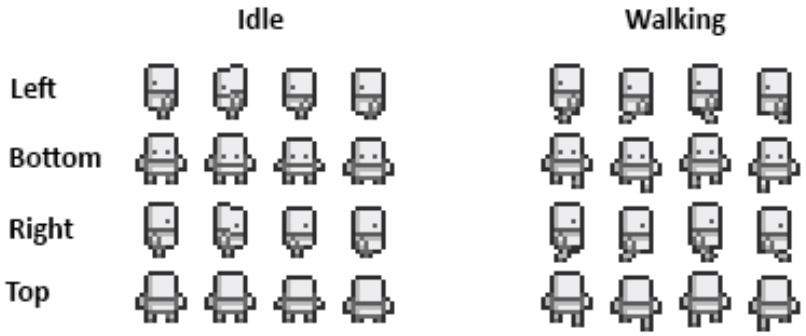
Video tutorial is available here: https://www.youtube.com/watch?v=i8KAllDt9zk

Preparation

To begin using the Animator Sprite Swap system you will need to have an existing Animator Controller asset with some Animation States.

For this example we will be setting up a character with 4 facing directions (Top, Bottom, Left, Right) and we will be using a simple Animator Controller setup with 2 Animation States:

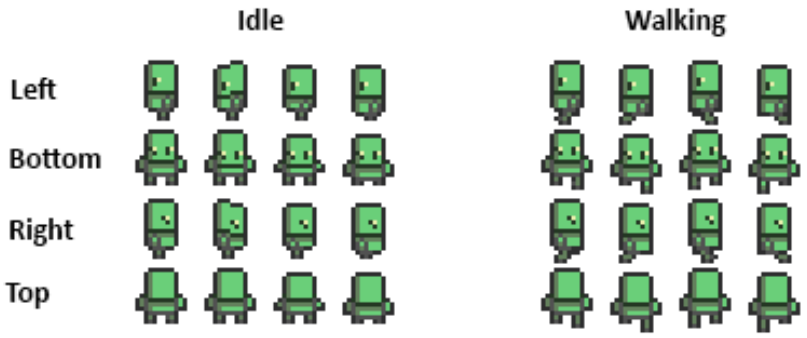
Sprites used for these animations can be found in two separate spritesheets. First spritesheet contains the “Idle“ animation sprites, second spritesheet contains the “Walking“ animation sprites:



“Idle“ animation clip uses the first 4 sprites in from the “4Dir_base_idle.png“spritesheet

“Walking“ animation clip uses the first 4 sprites in from the “4Dir_base_walking.png“ spritesheet

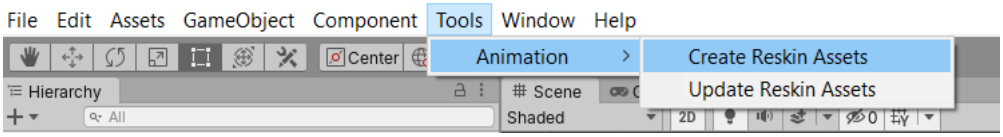
In this tutorial, we will be setting up a character which will be using sprites from the spritesheets below:



Before that, make sure both spritesheets are sliced similarly using the Sprite Editor. **Sprite count and sprite order needs to be the identical**, otherwise the Reskin Dynamic asset might not work as expected.

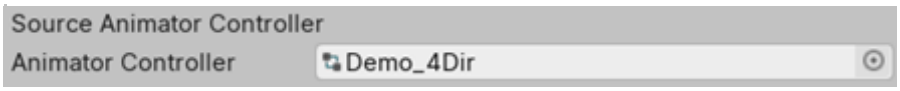
Creating an Animator Reskin asset

1. Open **Tools->Animation->Create Reskin Assets**



1. Select **Dynamic** at the top of the Reskin window

2. Select your character's **Animation Controller** asset you want to create a **Reskin Dynamic asset** for



3. Set the **Alternate Sprite Source** to **Same Texture** (alternate sprites for each animation are stored in the same spritesheet texture)

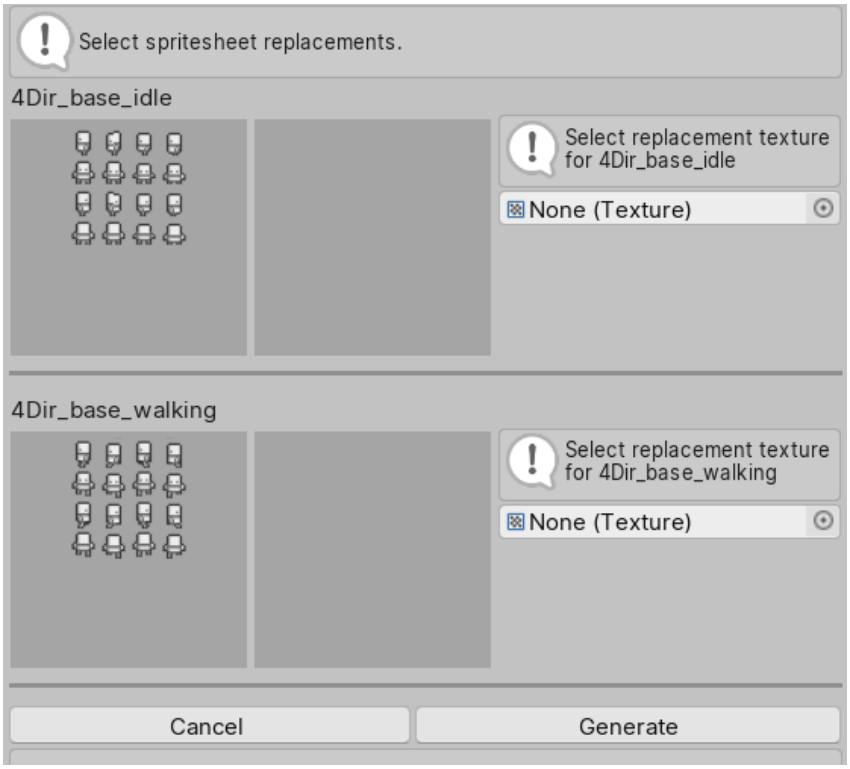
Same Texture is used when alternate sprites for each animation are stored in the same file. For example, Idle animation and all its alternative sprites are the same spritesheet and each set of alternate animation sprites are ordered one after another (Idle_Left uses sprites 1-4, Idle_Bottom uses sprites 5-8, Idle_Right uses sprites 9-12, Idle_Top uses sprites 13-16 and are stored in Character_Idle.png spritesheet). Multiple animations can be stored in the same spritesheet, they just have to follow a strict order.

Separate Textures is used when alternate sprites for each animation are stored in separate files following the same sprite order. For example, Idle_Left (sprites 1-4) and Walking_Left (sprites 5-8) are stored in spritesheet Character_Left.png, Idle_Bottom (sprites 1-4) and Walking_Bottom (sprites 5-8) are stored in spritesheet Character_Bottom.png, etc.

4. Set the **Sprite Alternate Count** to **4** (4 alternate animations for each character facing direction)

This will automatically pick up the remaining sprites from the spritesheet to be used in alternate animations. In this example, by setting the **Alternate Sprite Source** to **Same Texture** and **Sprite Alternate Count** to **4**, since our default "Idle" animation (Left direction) used sprites 1-4 from the "Idle" spritesheet, the tool will automatically pick up the other alternate animations from the texture as well - second alternate animation (Bottom) will use sprites 5-8, third alternate animation (Right) will use sprites 9- 12, fourth alternate animation (Top) will use sprites 13-16. The same will be done with all other animations.

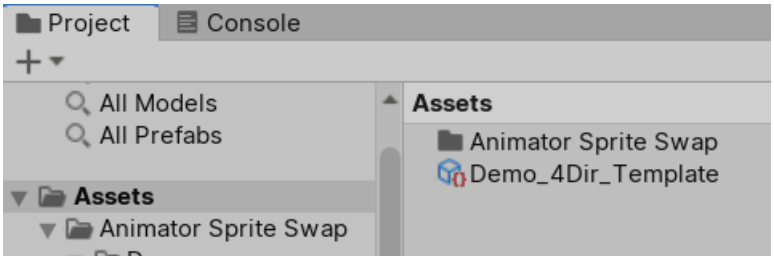
5. Click on **Generate**
You will be shown a list of spritesheets that are used for the Animations found in the Animation controller asset



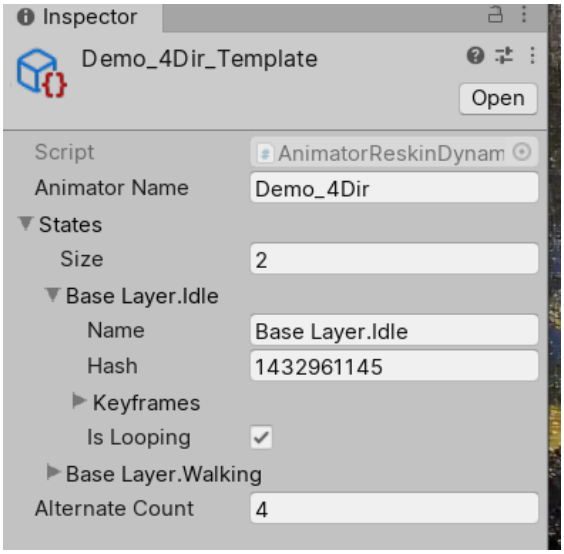
6. Select the sliced spritesheet textures you want to replace the originals with

7. Click on **Generate**

An Animator Reskin Dynamic asset should be generated in /Assets/ directory by the name of your Animator Controller.

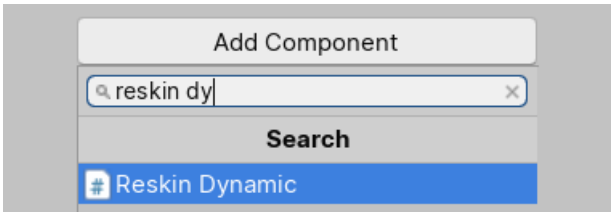


Inside it, you will find all of the states in the Animation Controller that were found in the Animator Controller asset. Each state in the Reskin asset contains keyframe data and sprite. Each Animation State will be labled by its respective directory in the Animator Controller (e.g. “Idle” state in the “Base Layer” layer will be named “Base Layer.Idle”

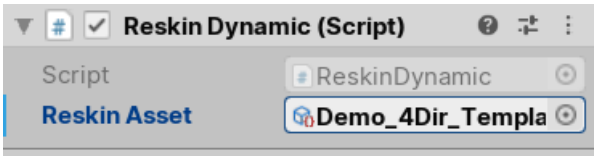


Setting up your character and animations

1. Add a **Reskin Dynamic** script component to an gameobject/prefab you want to reskin



2. Assign the **Animator Reskin Dynamic** asset that was generated to the **Reskin Dynamic** component's **Reskin Asset** property



Once you are finished with these steps, upon entering Play mode your character should be using the sprites from the Animator Reskin Dynamic asset assigned in Reskin Dynamic component for it's animations:

Using alternate animations

1. Get a reference to the **ReskinDynamic** script

```
reskin = GetComponent<ReskinDynamic>();
```

2. Use **SetIndex(index)** function to use an alternate animation

```
reskin.SetIndex(1);
```

In this example, index 0 is used for Left facing animations, 1 is used for Bottom facing animations, 2 is used for Right facing animations, 3 is used for Top facing animations. Calling **reskin.SetIndex(1)** set out character facing direction to Bottom:

By applying your own game logic you can call **reskin.SetIndex** whenever you need to play a certain alternate animation. Example script below changes the animation index based on user WASD input (plays Left (index 0) animations when key “A” is pressed, plays Bottom (index 1) animations when key “S” is pressed, plays Right (index 2) animations when key “A” is pressed, plays Top (index 3) animations when key “W” is pressed).

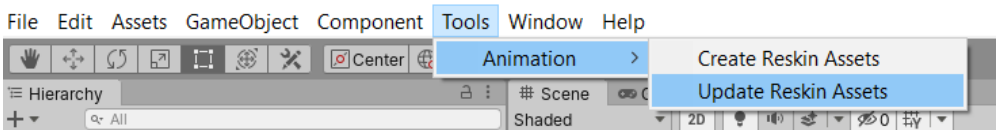
```
public class Demo_4Dir : MonoBehaviour {

    private ReskinDynamic reskin;

    //look direction index (0 - left, 1 - bottom, 2 - right, 3 - top)
    private int directionIndex;

    void Start()
    {
        reskin = GetComponent<ReskinDynamic>();
    }

    void Update()
    {
        //find the correct direction index for animations
        if (Input.GetKey(KeyCode.A))
            directionIndex = 0;     //walking left
        else if (Input.GetKey(KeyCode.S))
            directionIndex = 1;     //walking bottom
        else if (Input.GetKey(KeyCode.D))
            directionIndex = 2;     //walking right
        else if (Input.GetKey(KeyCode.W))
            directionIndex = 3;     //walking top

        //set the animation index
        if(reskin != null)
            reskin.SetIndex(directionIndex);
    }
}
```

Regenetaring Reskin Assets (version 1.1)

If you modify your reskinned Animator Controller or Animation Clip assets referenced in the Animator Controller, your existing Reskin assets will need to be regenerated.

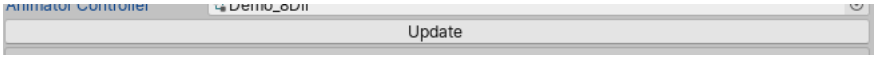
Updating an Animator Reskin asset

1. Open **Tools->Animation->Update Reskin Assets**



2. Select your character's **Reskin Asset** and the **Animation Controller** it is based on

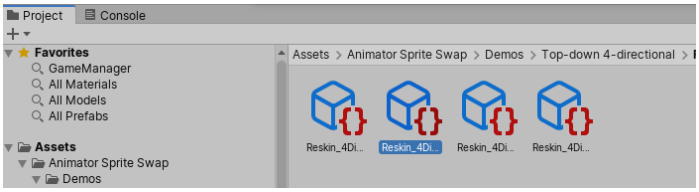
3. Click on **Update**



The **Reskin Asset** will be updated using the data from the selected **Animator Controller** asset. Newly added animation sprites will be automatically replaced with sprites from the replacement spritesheets already referenced in the asset or if a replacement cannot be found, from a user selected spritesheet instead.

Alternatively, Animator Reskin asset can be updated by selecting it in the Project Window and clicking Update in the Inspector:

1. Select your character's **Reskin Asset** in the Project Window



2. Click on **Update** in the Inspector

Release notes

Animator Sprite Swap System Version 1.1 (2020-02-06)

Improvements

- Added the ability to regenerate/update Reskin Assets through the Update Window or through Reskin Asset selection

Other changes

- Added AnimatorSpriteSwapSystem namespace for all scripts

Contact information

If you encounter any issues, please contact me at **rinenproductions@gmail.com**