Spritedow Animator

A plugin to do simple sprite animations avoiding the big and tedious Unitys's Mecanim system. Oriented to programmers, if you prefer visual scripting you maybe prefer using Mecanim instead of this.

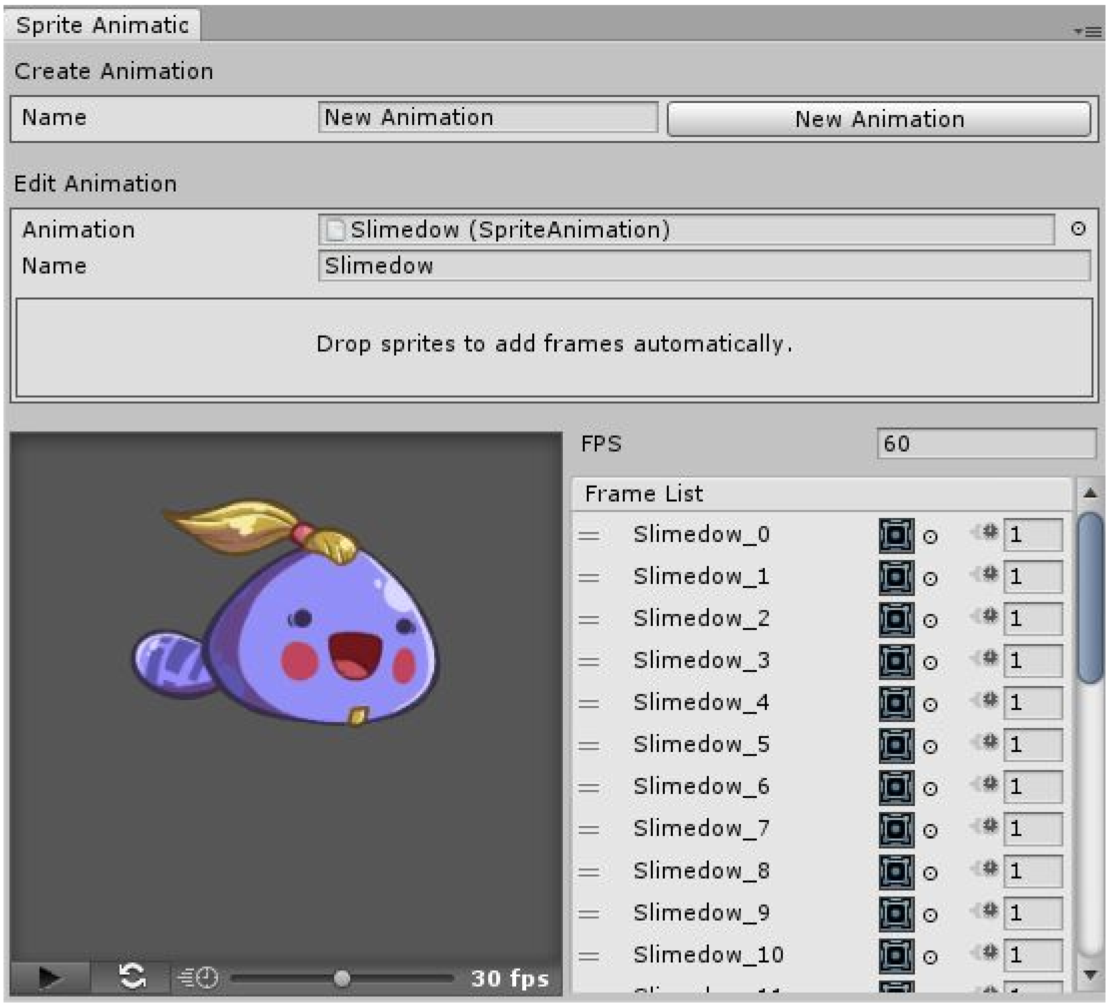
Thank you for your purchase! In this document you have all the info needed to use the plugin. If you have any questions don’t hesitate to send me a mail to help@elendow.com and I’ll try to help you out :)

You can also post your suggestions or questions in the Unity Forums thread.

Creating an animation

Use the animation editor to create new animation files. You can open it selecting Sprite Animation Editor on Elendow Tools tab.

- Give a name to the animation. This will be also the asset name. This name will be the one used to play the animations.
- Select the folder to save.
- Select the framerate of the animation.
- Add frames manually or dropping the sprite to the Drag&Drop box.
    - If you drop a Texture instead of a sprite to the Drag&Drop box, the plugin will take all the sprites on that Texture
    - You can change the duration of each frame, 1 by default, to any number greater than 0.
    - You can sort or delete the frames.
- Any change is automatically saved.
- You can preview the animation.
    - The speed and loop settings of the preview window are only for that window.



Sprite Animatic

Create Animation

Name: New Animation | New Animation

Edit Animation

Animation: Slimedow (SpriteAnimation)
Name: Slimedow

Drop sprites to add frames automatically.

FPS: 60

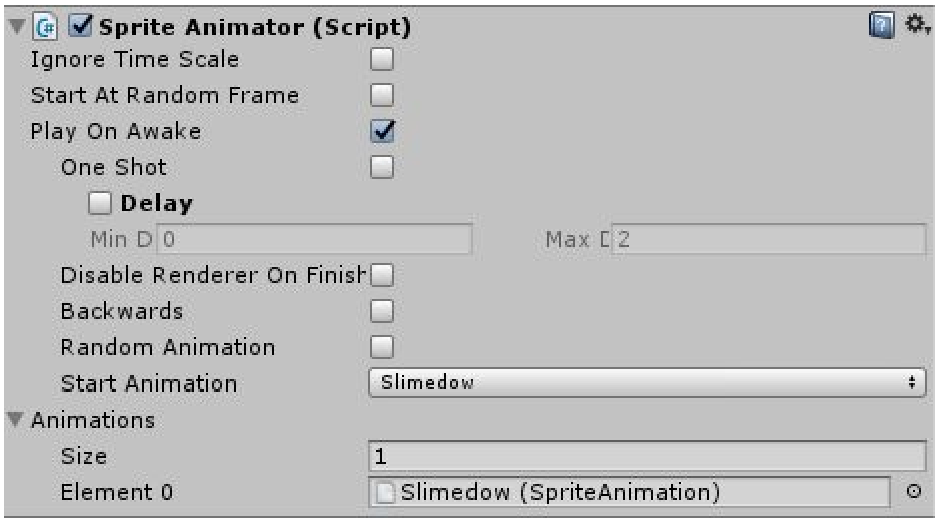
Frame List

Frame	Duration
Slimedow_0	1
Slimedow_1	1
Slimedow_2	1
Slimedow_3	1
Slimedow_4	1
Slimedow_5	1
Slimedow_6	1
Slimedow_7	1
Slimedow_8	1
Slimedow_9	1
Slimedow_10	1

30 fps

Inspector properties

- **Ignore TimeScale** will set the animation to ignore the game TimeScale.
- **Start At Random Frame** will set the animation to start at random frame when “Play” is called.
- **Play on Awake** will start playing when the object awakes.
  - **One Shot** if false the animation will loop infinite times.
    - **Delay** if true a delay between loops will be made
      - **Min** minimum delay time
      - **Max** maximum delay time
  - **Disable Renderer On Finish** will disable the renderer after every loop cycle if there’s a delay specified or will disable the renderer after the animation ends if it’s not looping.
  - **Backwards** if true the animation will play backwards.
  - **Random Animation** if true the start animation will be random, and the animation will randomly change after every loop cycle.
  - **Start Animation** is the animation that will plays when Play on Awake is true. This is disabled if Random Animation is selected.
- **Animations** is a list with all the animations.



Sprite Animator (Script)

Ignore Time Scale	☐
Start At Random Frame	☐
Play On Awake	✓
One Shot	☐
☐ **Delay**	
Min D 0	Max D 2
Disable Renderer On Finish	☐
Backwards	☐
Random Animation	☐
Start Animation	Slimedow
Animations	
Size	1
Element 0	Slimedow (SpriteAnimation)

Using the animations

Add the **SpriteAnimator** or **UIAnimator** component to the object you want to animate and fill the animations list with the animations you want. This component requires a SpriteRenderer or Image component to work. If the object doesn't have one, the animator will add it automatically. On your code, use **GetComponent<SpriteAnimator>** or **GetComponent<UIAnimator>** to get the reference and start using it.

Methods

- **Play (bool oneShot = false, bool backwards = false)** plays the first animation of the animation list.
- **Play (string animationName, bool oneShot = false, bool backwards = false)** plays the animation infinite times if oneShot = false, only one time if true, fordward if backwards = false and backwards if its true.
    - If the animation is the same that is playing, nothing will happend but the oneShot attribute will update.
    - If the animation is the same that was playing but its not playing now, the animation will Reset and Resume and the oneShot attribute will update.
    - If the animation is different, it will play the new animation from the start.
- **PlayRandom (bool playOneShot = false, bool backwards = false)** plays a random animation from the animation list.
- **Resume ()** resumes the current animation.
- **Stop ()** stops the current animation.
- **Reset ()** restarts the animation (playing or not) to its initial state. If the animation is not playing, the restart will be applied only when it start playing again.
- **SetActiveRenderer (bool active)** enable/disables the renderer.
- **FlipSpriteX (bool flip)** flips the sprite on the X axis.
    - Not working on UIAnimator yet.
- **FlipSpriteY (bool flip)** flips the sprite on the Y axis.
    - Not working on UIAnimator yet.
- **AddCustomEvent (int frame)** adds an event to a specific frame of the first animation of the animation list and returns it.
- **AddCustomEvent (string animation, int frame)** adds an event to a specific frame of an animation and returns it.
    - If the animation name is empty, it will get the first animation of the list.
    - The event subscriber must have this estructure MethodName(BaseAnimator caller){}
- **GetCustomEvent (int frame)** returns the event of the first animation of the animation list at the specific frame. Returns null if there's no event on that frame and animation.
- **GetCustomEvent (string animation, int frame)** returns the event of the animation at the specific frame. Returns null if there's no event on that frame and animation.
    - If the animation name is empty, it will get the first animation of the list.
- **SetRandomDelayBetweenLoops(float min, float max)** sets a random delay between loops. The animation will stay at the last frame, but you can use **DisableRenderOnFinish** to avoid this.
- **SetDelayBetweenLoops(float delay)** sets a fixed delay between loops. The animation will stay at the last frame, but you can use **DisableRenderOnFinish** to avoid this.
- **Initialize(bool playOnAwake, List<SpriteAnimation> animations, string startAnimation)** manually initialize the animator. Useful and **NECESSARY** if the animator was instanced on runtime.

Properties

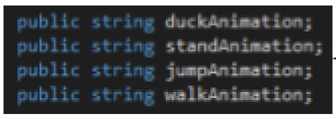
- **bool IsPlaying { get; }** returns true if the animation is playing and false if not.
- **string CurrentAnimation { get; }** returns a string with the current animation name.
- **int CurrentFrame { get; }** returns the current frame of the animation.
- **bool DisableRenderOnFinish { set; }** sets the disableRenderer attribute. This will disable the renderer when the animation ends.
- **bool RandomAnimation { set; }** if true the animator will get a random animation after every loop cycle
- **bool StartAtRandomFrame { set; }** if true the animator will start the animations at a random frame instead of the first one. Cool if you want to desynchronize animations.

Events

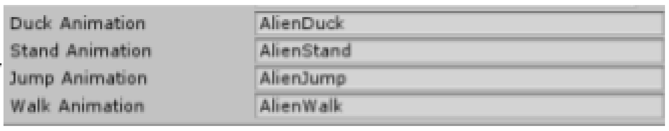
- You can suscribe to the animation events using the **AddListener(Listener)** method of the UnityEvent class.
- **onFinish** calls when the animation reach the last frame.
- **onPlay** calls when the animation starts playing.
- **onStop** calls when the animation is forced to stop.
- You can add an event to a specific frame of an animation using the method **AddCustomEvent**(string animation, int frame).
  - The event subscriber must have this structure **MethodName(BaseAnimator caller){}**
  - Ex: animation.AddCustomEvent("Walk", 3).AddListener(StepFrame). Now on the frame 3 of the animation "Walk" the method StepFrame will be called.

Editor Utils

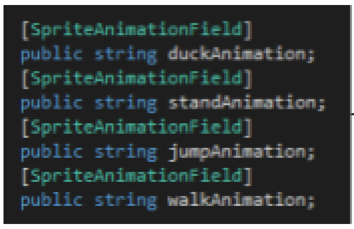
- **[SpriteAnimationField]** : Use this attribute to transform a string field in a animation list field. This is useful to avoid human errors when setting animations from the inspector.



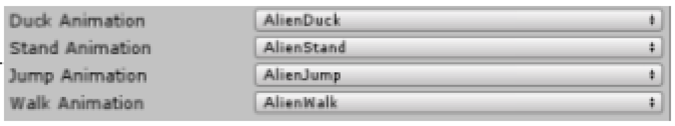
```
public string duckAnimation;
public string standAnimation;
public string jumpAnimation;
public string walkAnimation;
```



Duck Animation	AlienDuck
Stand Animation	AlienStand
Jump Animation	AlienJump
Walk Animation	AlienWalk



```
[SpriteAnimationField]
public string duckAnimation;
[SpriteAnimationField]
public string standAnimation;
[SpriteAnimationField]
public string jumpAnimation;
[SpriteAnimationField]
public string walkAnimation;
```



Duck Animation	AlienDuck
Stand Animation	AlienStand
Jump Animation	AlienJump
Walk Animation	AlienWalk