

# TNT Animator Studio v1.00

TNT Animator Studio can be used in free and commercial products. If you have enjoyed its use, I would appreciate it if you would donate towards its continued development. Your donations help me stay motivated in order to fix bugs and add new features. Plus anyone who makes a donation will be provided with the **FULL SOURCE CODE** for the TNT Animator Framework.

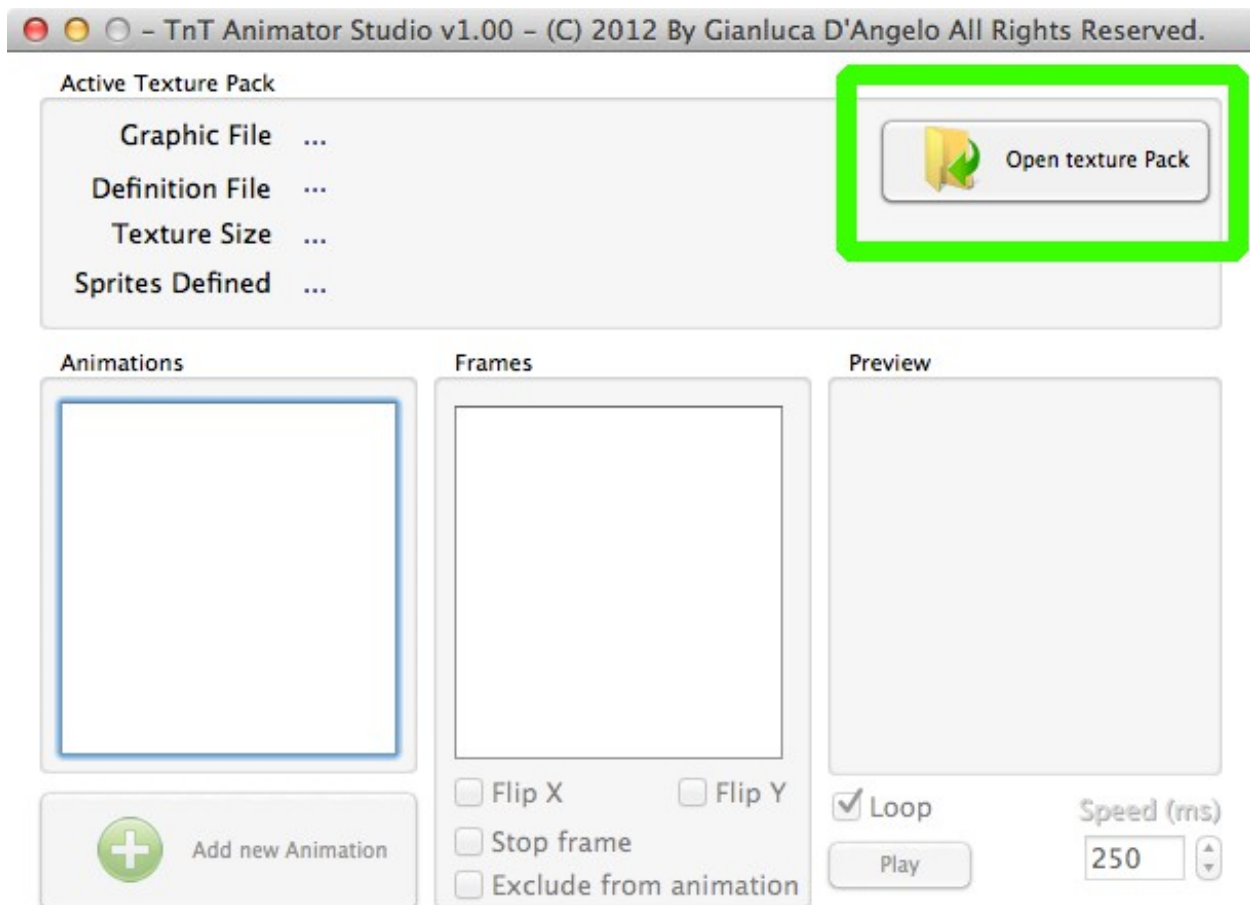
TNT Animator Studio is composed of two components, the **Visual Editor** and the **Framework**. The Visual Editor provides a means to create animations by adding, deleting, and customizing sprite frames. You can manage the speed of the frames, loop the animation, and customize its orientation on screen. When you have finished customizing your animation, it can be exported into a pack file that can be loaded into your Gideros Studio game.

The visual editor is available for use on Mac and Windows computers. (Mac screenshots used for the examples in this document)

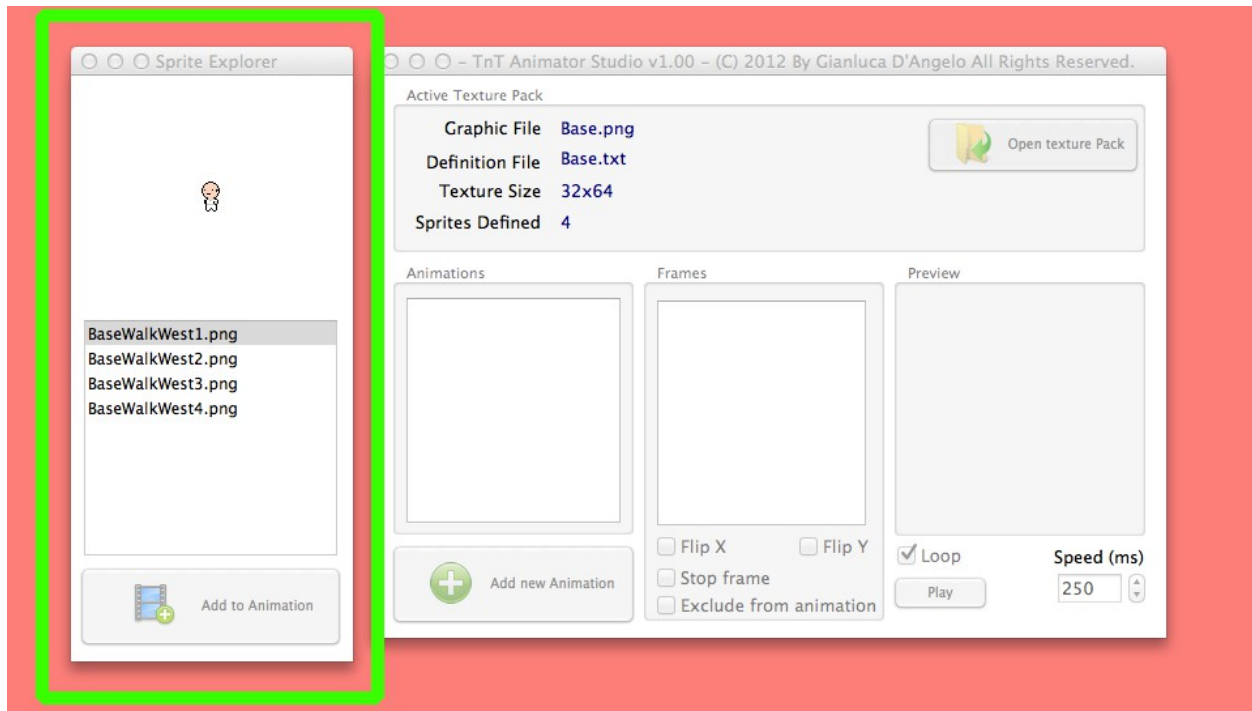
You might ask, "Why use this framework over Gideros' movieClip animation class?" The reason being, TNT Animator's framework uses a "time based" system to manage your animations whereas Gideros uses a frame based system. A time based system allows for an animation to run at the same speed no matter what system it is played on. Also, you get the added bonuses of controlling the orientation of the frames and including a stop frame and realtime animation speed change. TNT Animator is based on Gideros' movieClip so that it retains its efficiency and speed. This way you get all of the benefits with none of the draw backs!

## Part 1 - Create a new animation project

To begin, select the **Open Texture Pack** button in order to import a Gideros Studio compatible texture pack. A texture pack contains all of the frames you will use to create your animation.

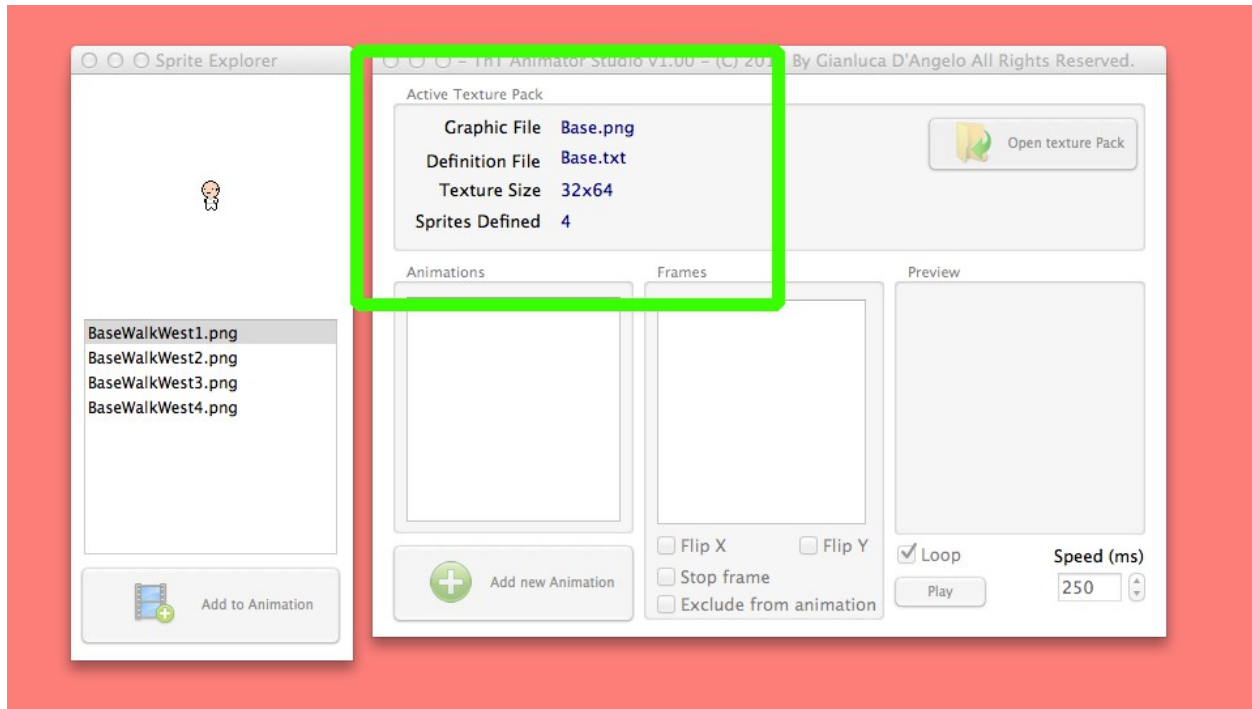


After you have selected your texture pack a new window will appear titled **Sprite Explorer**. This window will contain all of the frames from your texture pack.



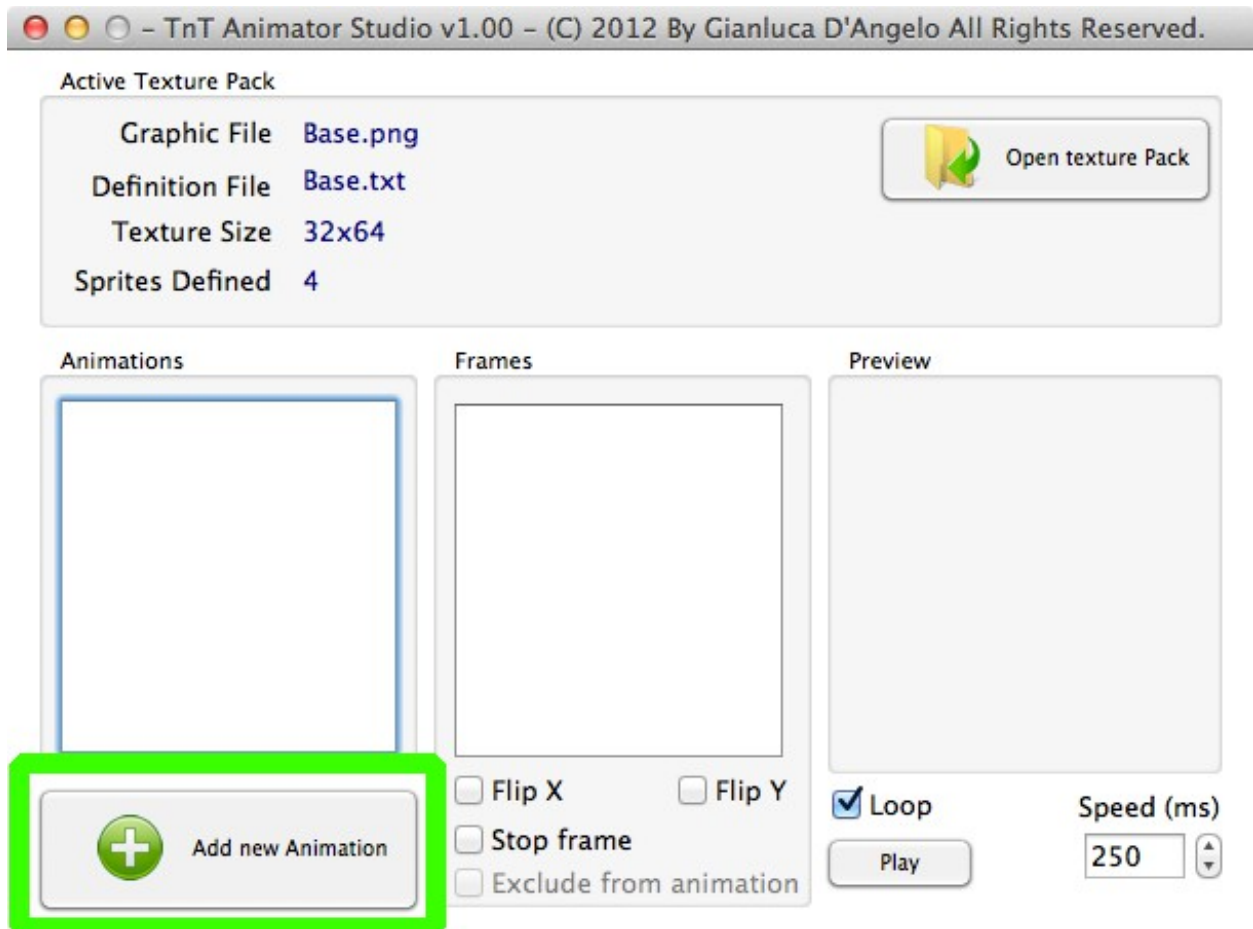
In the main window, you will see useful information about your texture pack in the **Active Texture Pack** box.

- Graphic File Name
- Texture Pack Definition File
- Texture Size (width and height in pixels)
- Number of Sprites Defined in the Texture Pack

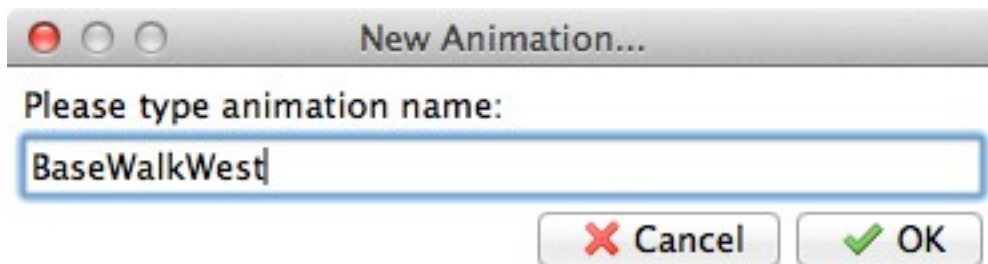


## Part 2 - How to Create an Animation

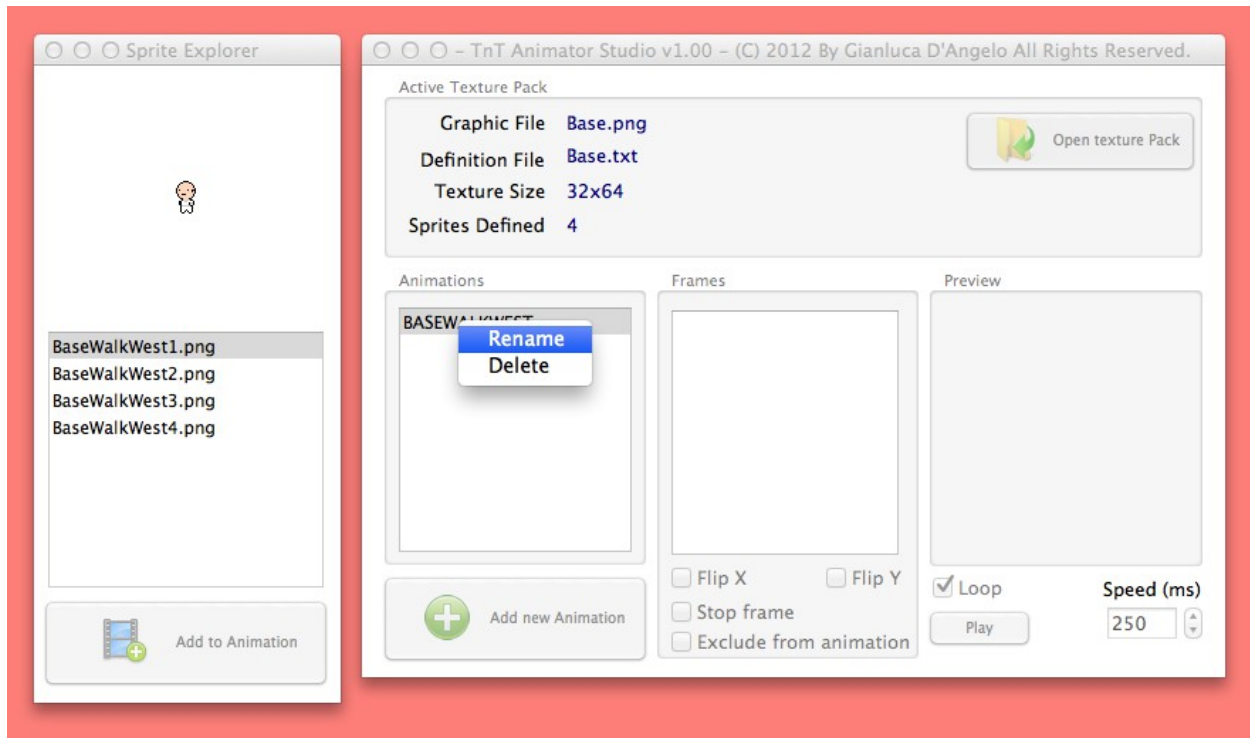
To create a new animation select the **Add New Animation** button.



A dialog box will appear asking you to name your animation. This is the same name you will use to load that particular animation in your code. (So you might want to give it a useful and descriptive name.)

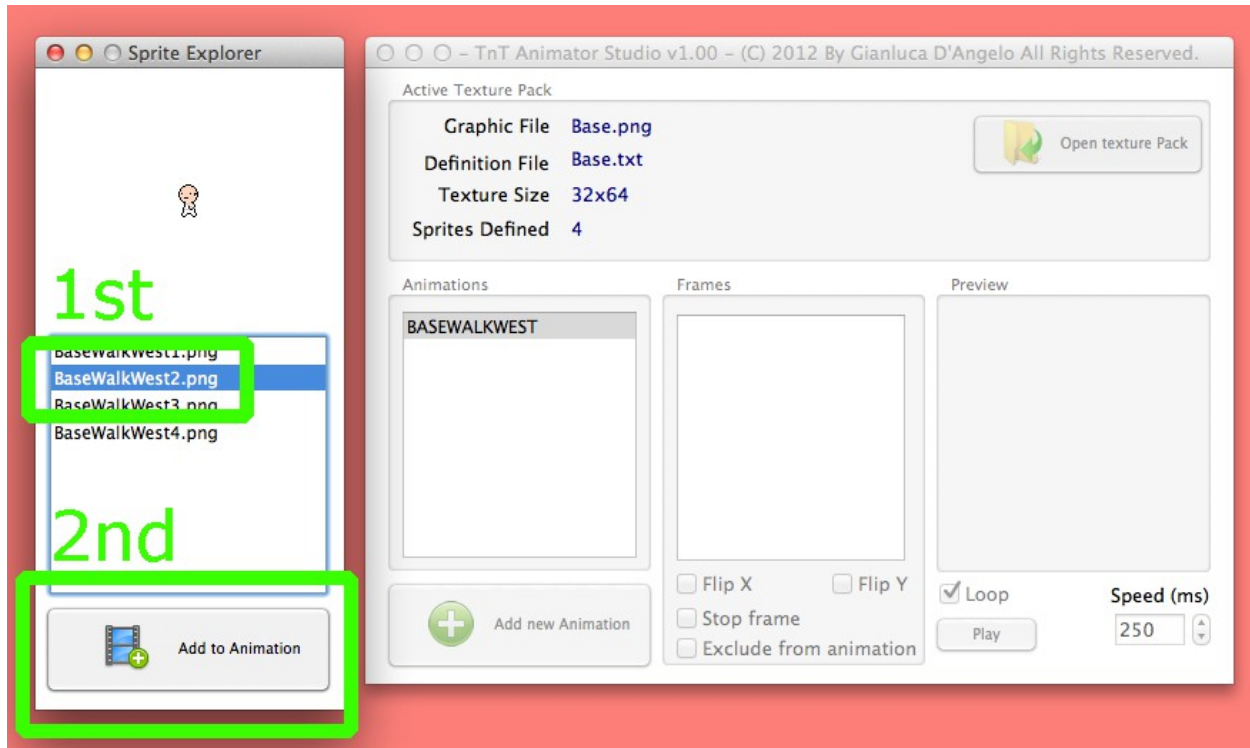


After you have clicked the OK button the new animation will appear in **Animations Frame**. At any point you can right click on the animations name to delete or rename it.



## Part 3 - How to set the frames of the animation

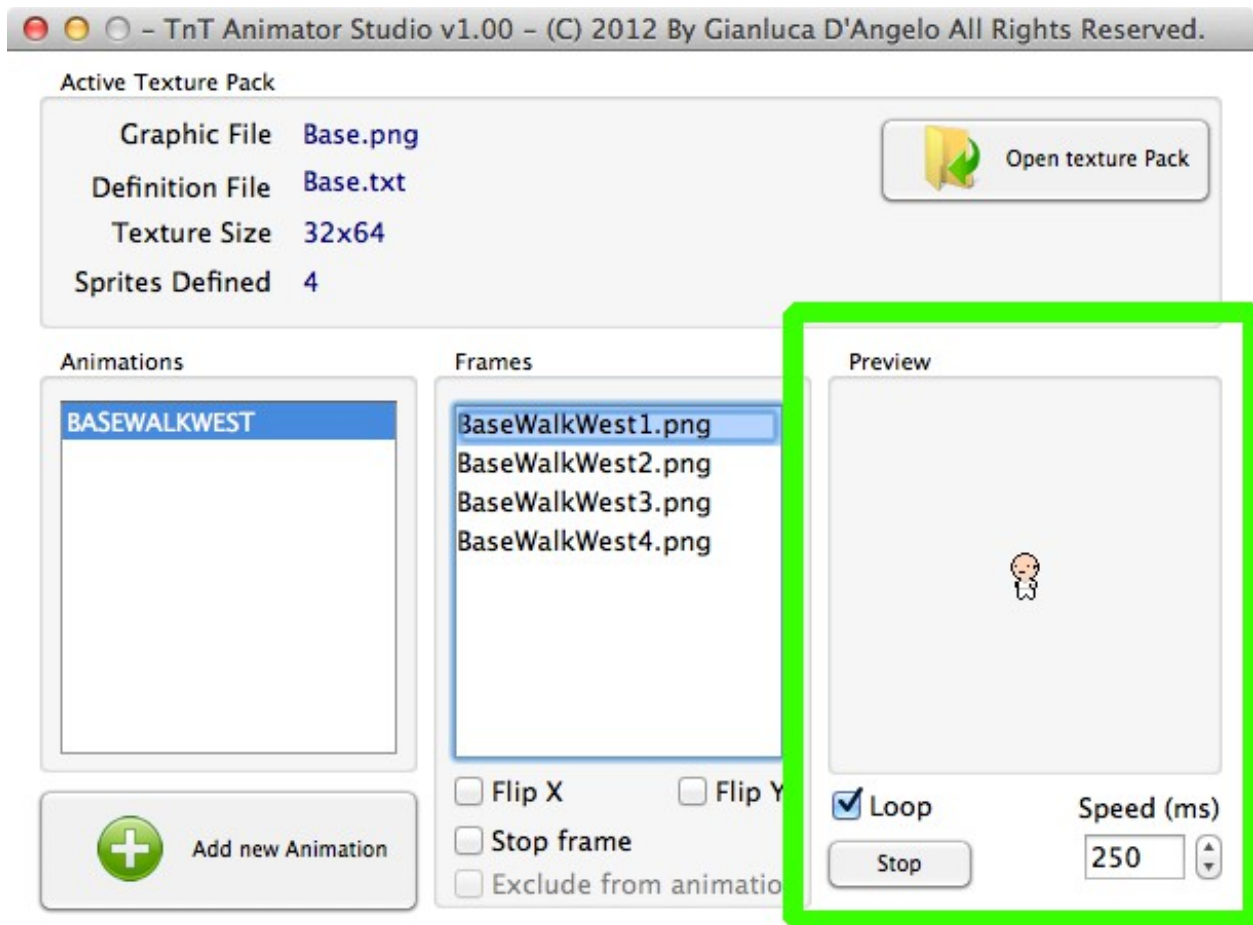
To start adding frames to your animation, you must select a frame from the **Sprite Explorer Window** and then click the **Add to Animation Button**.



Then you can select your frame in the **Frames Frame** (sorry!) and edit its properties.

- Flip X - Flip the frame's picture horizontally
- Flip Y - Flip the frame's picture vertically
- Stop Frame - The stop frame will be the final frame shown when the animation ends (by default the stop frame is the first frame)
- Exclude Frame - Is only available when Stop Frame is selected. This frame will be excluded from the animation loop.
- Drag the frames with your mouse to rearrange them
- Right click a frame to delete it

The finished animation will now be shown in the **Preview Frame**. You can click the **Play/Stop Button** to preview the animation. Select **Loop** to make the animation loop continuously, and select the speed in milliseconds that the animation plays at.





## Part 4 - Save your animation pack

In the file menu you will find many choices.

- **New TNT Animation** - Create a new animation pack
- **Open Animation Project** - Open a previously made animation project to edit
- **Save Animation Project** - Save the animation project
- **Save As Animation Project** - Save the animation project with a new name
- **Publish TNT Animation** - Export your animation pack with a filetype “.TAN” This is the animation pack that your game will use.
- **Exit** - Exit the program



New TnT Animation



Open Animation Project



Save Animation Project



Save As Animation Project



Publish TnT Animation



Exit

## Part 5 - Include the Framework in your Project

First you must add the TNT Animator Framework to your existing project.

### \*\*\* IF YOU HAVE THE SOURCE CODE VERSION

Right click on your project and select **Add Existing File** then select **tntanimator.lua** file.

### \*\*\* IF YOU HAVE THE PRECOMPILED VERSION

Right click on your project and select **Add Existing File** then select **tntanimator64.lua** and **tntanimator32.lua**. Now you must right click both of these files and select **Exclude From Execution**. Finally in your **main.lua** file you must include the code.

```
local function is32bit()  
    return string.dump(is32bit):byte(9) == 4  
end
```

```
if is32bit() then  
    require("libs/tntanimator32")  
else  
    require("libs/tntanimator64")  
end
```

## Part 6 - Use the Framework in your Project (Reference)

The TNT Animator Framework provides two classes. (For examples of usage please see the two provided example projects provided with the framework.)

- **CTNTAnimatorLoader (core class)** - Load animation file definitions
- **CTNTAnimator (derived from Gideros' Sprite class)** - Manage loaded animations

### CTNTAnimatorLoader (core class)

#### **CTNTAnimatorLoader:init()**

Init and allocate memory for animation loader. No parameters for CTNTAnimatorLoader constructor are needed.

Example:

```
local animLoader = CTNTAnimatorLoader.new()
```

#### **CTNTAnimatorLoader:loadAnimations(fileName, texture, [midHandle])**

load TNT Animation file (.tan)

fileName:

TNT Animation file (.tan)

texture:

texture pack that contain all sprites

midHandle:

if true all animations' anchor points are set to the middle

If false all animations' anchor points are set to (0, 0) or the top left corner

Example:

```
animLoader:loadAnimations("crook.tan", crookTexturePack, true)
```

#### **CTNTAnimatorLoader:Free()**

Free all animatorLoader allocated memory and release nil.

Example:

```
animLoader = animLoader:free()
```

## CTNTAnimator (derived from Gideros' Sprite class)

### **CTNTAnimator:init(animatorLoader)**

*Init animator class*

#### **animationLoader:**

*AnimatorLoader class reference.*

*Example:*

```
self.anim = CTNTAnimator.new(animLoader)
```

### **CTNTAnimator:free()**

*free animator class memory allocated.*

*Example:*

```
self.anim = self.anim:free()
```

### **CTNTAnimator:getLoop()**

*return true if current animation is looped else false*

*Example:*

```
local animationIsLooped = self.anim:getLoop()
```

### **CTNTAnimator:setLoop(setLoop)**

*set/unset loop for current animation*

#### **setLoop:**

*true/false set/unset loop to current animation.*

*Example:*

```
self.anim:setLoop(true)
```

### **CTNTAnimator:getSpeed()**

*return (in milliseconds) the current speed of current animations*

*Example:*

```
local speed = self.anim:getSpeed()
```

### **CTNTAnimator:setSpeed(newSpeed)**

*set new animation speed for current animation.*

**newSpeed:**

*speed of animation in milliseconds.*

*Example:*

```
self.anim:setSpeed(100)
```

### **CTNTAnimator:getFrameCount()**

*return frames count of current animation*

*Example:*

```
local totalFrames = self.anim:getFrameCount()
```

### **CTNTAnimator:setAnimation(animationName)**

*set current animation.*

**animationName:**

*the name of the animation. Remember to write all in UPPERCASE*

*Example:*

```
self.anim:setAnimation("CROCK_DOWN")
```

### **CTNTAnimator:animationRunning()**

return true if current animation is running else return false.

Example:

```
local runningAnim = self.anim:animationRunning()
```

### **CTNTAnimator:getAnimation()**

return the name (string) of current animation.

Example:

```
local currentAnimName = self.anim:getAnimation()
```

### **CTNTAnimator:setAnimAnchorPoint(xHandle, yHandle, [animation])**

set graphics handle for all animation frames or only user specified animation.  
Work exactly the same as standard Gideros setAnchorPoint()

**xHandle, yHandle:**

define x and y handle (see setAnchorPoint in Gideros Documentation)

**animation: (optional parameter)**

define animation to set handle

Example:

```
self.anim:setAnimAnchorPoint(1,1) – set bottom right handle for all sprites  
and all animations defined in “self.anim”
```

```
self.anim:setAnimAnchorPoint(1,1, “CROCK_DOWN”) – set bottom right  
handle for only sprites in “CROCK_DOWN” animation.
```

### **CTNTAnimator:addChildAnimation(parentGroup)**

work like Gideros addChild but for animated sprites; so add an animated sprite as child to parentGroup

#### **parentGroup:**

parent of animated sprite

Example:

```
self.anim:addChildAnimation(self)
```

### **CTNTAnimator:playAnimation([playFromFrame])**

start to play current animation. If animation is already running playAnimation code is ignored.

playAnimation is automatically updated (only when needed) by TNT Animator Framework; so it's very fast and optimized. (is not called at every frame update)

#### **playFromFrame: (optional parameter)**

if defined, animation start from this frame else start from first frame.

Example:

```
self.anim:playAnimation()  
self.anim:playAnimation(3) – start animation from frame 3
```

### **CTNTAnimator:stopAnimation([stopOnFrame])**

stop current animation (if it's running else the code is ignored). It's stop on predefined (in the editor) stop frame else it stop on last frame.

#### **stopOnFrame: (optional parameter)**

if defined animation stop here.

Example:

```
self.anim:stopAnimation()  
self.anim:stopAnimation(3) – stop animation on frame 3
```

## Part 7 - TNT Animator Framework Events

The TNT Animator Framework has the capability to raise EVENTS so you can control animations and react to events.

At the moment there are only two events defined and they are:

- **ANIM\_START** - Raised when an animation starts to play
- **ANIM\_STOP** - Raised when an animation stops or ends.
- **ANIM\_CHANGE** - Raised when an animation is set or changed.

Additional information is defined in the event data **animationName** which is the name of the animation when the event was raised.

For an example on events, try example 2.

(This example works only on real device because it use the accelerometer.)



# Acknowledgement

Special thanks to the Gideros forum, Gideros Team (Atilim, Gorkem and Deniz) for making Gideros the best Mobile SDK out there!.

Please visit: [www.giderosmobile.com](http://www.giderosmobile.com)

Thanks to all my beta tester (in alphabetical order):

Atilim  
Fierceblaze  
Gleen Benson  
Magnusviri  
Ruxpin (special thanks for docs!)  
Scouser  
Teranth

Please visit my website:  
[www.tntengine.com](http://www.tntengine.com) or [www.tntparticlesengine.com](http://www.tntparticlesengine.com)

follow me on twitter: <https://twitter.com/#!/GregBUG>  
on facebook: <http://www.facebook.com/gianluca.greg.dangelo>