
Synfig Documentation

Release 1.5.1

Synfig Community

Dec 28, 2022

CONTENTS:

1	Installation	1
2	Quick Start	3
3	Main concepts	55
4	Tutorials	85
5	Importing your artwork	111
6	Tools	119
7	Layers	151
8	Parameters	247
9	Converters	279
10	Configuration	281
11	Canvas	283
12	Interface	289
13	Export	325
14	Plugins	327
15	Glossary	333
16	Release Notes	335
17	About this Manual	387
18	How to Help	391
19	Misc	393
20	Interpolations	395
21	Indices and tables	397

INSTALLATION

1.1 Windows

1. Download the .exe file.
2. Run the file. A wizard will then guide you through the rest of the installation.

1.2 Linux

1. Download the AppImage file.
2. Right-click on the file and select Properties on the context menu.
3. Navigate to Permissions tab and enable checkbox that says “Allow executing file as program”.
4. Close the dialog.
5. Double-click on the AppImage file to run the application.

For more information see [this forum thread](#).

1.3 MacOS / OSX

1. Download the .dmg file.
2. Open downloaded dmg file.
3. Drag the Synfig Studio icon file into your Applications directory. Wait for the copy process to be completed.
4. Run Synfig Studio from your Applications directory.

If you get an error message like “can’t be opened because it comes from a non identified developer”, then do the following:

1. Locate the Synfig Studio app in Finder (don’t use Launchpad to do this - Launchpad doesn’t allow you to access the shortcut menu)
2. Control-click the app icon, choose “Open” from the shortcut menu.
3. Click Open.

The app is saved as an exception to your security settings, and you can open it in the future by double-clicking it just as you can any registered app.

Note: You can also grant an exception for a blocked app by clicking the “Open Anyway” button in the General pane of Security & Privacy preferences. This button is available for about an hour after you try to open the app. To open this pane, choose Apple menu > System Preferences, click Security & Privacy, then click General.

For more details please refer to [this page](#).

QUICK START

2.1 Introduction

As you probably know, animation is the rapid display of a sequence of images in order to create an illusion of movement. Traditionally 2D animation is created by drawing each displayed image individually. Those images are called “frames” and thus such method is called “frame-by-frame animation”. To create a good illusion of movement you need to draw many frames, that’s why this method requires a lot of time and resources.

A traditional frame-by-frame animation, taken from : <http://flipily.com>

Synfig Studio is an [open-source](#) 2D vector animation software. It is designed to produce film-quality animation with less people and resources.

Synfig Studio is built to eliminate the need to draw each frame individually. There are two techniques for that:

- Morphing animation
- Cutout animation

2.1.1 Morphing

Morphing animation of a rose, by Rore

Morphing is a technique that takes two images and creates a smooth transition between them. In the process of morphing, one shape is deformed into another and this transformation is usually defined by control points.

In Synfig Studio images are constructed from vector shapes and the morphing is done automatically. This allows us to create animations by drawing only the key positions at relatively wide time intervals. You need only to draw a few frames as needed to create a basic sense of motion for the scene, and Synfig Studio will create the in-between frames.

2.1.2 Cutout animation

Cutout animation in Synfig's tutorials

Cutout animation is created by splitting objects into parts and applying some simple transformations to them (like translation, rotation or scale) at different moments of time.

Synfig Studio uses those values to interpolate the motion for in-between frames. Cutout animation can be produced from bitmap images or vector graphics.

2.1.3 Synthesis and other functionalities

In both cases the role of Synfig Studio is to fill the gaps between the drawn frames (also called “keyframes”) and produce smooth and fluid animations. This process is called “tweening”.

Although Synfig Studio is not directly intended to draw animation frame-by-frame, it can be used to bring your hand-drawn frame-by-frame animation to the film-quality level by converting bitmap data of each frame into vector format. This process is called “tracing” and usually done by hand by constructing vector shapes on top of bitmap images. In the process of construction you can apply a lot of fascinating effects built into Synfig Studio to achieve a professional look for your animations.

Whether you do frame-by frame animation or not, Synfig Studio gives you flexible control over the repeated data, such as colors, outline characteristics, textures, images and many more, even animation trajectories and their sets (actions). Reusing repeated data is achieved via linking. This is a power of Synfig Studio, which is especially important for big animation projects.

Among the plain linking pieces of artwork data you can also define relations between them using a set of functions. That allows to create automatic animation based on the defined laws and bring whole animation process to the next level.

Parabolic shot in Synfig's tutorials

All those features of Synfig Studio are covered in detail in the chapters of this manual.

2.2 Getting Started

2.2.1 User interface

The screenshot below displays Synfig Studio's window layout:

Synfig Studio main interface components are:

- *Toolbox* — is the main Synfig Studio window. It contains tools and more to create and edit your artwork.
- *Canvas* — displays your artwork and animation.
- *Panels* — contain tools and information about certain elements of your project. Some panels will allow you to modify those elements.

Note: Synfig Studio can be reset to its default window arrangement (as shown in the screenshot). In the **File** menu select “Window -> Workspace -> Default”.

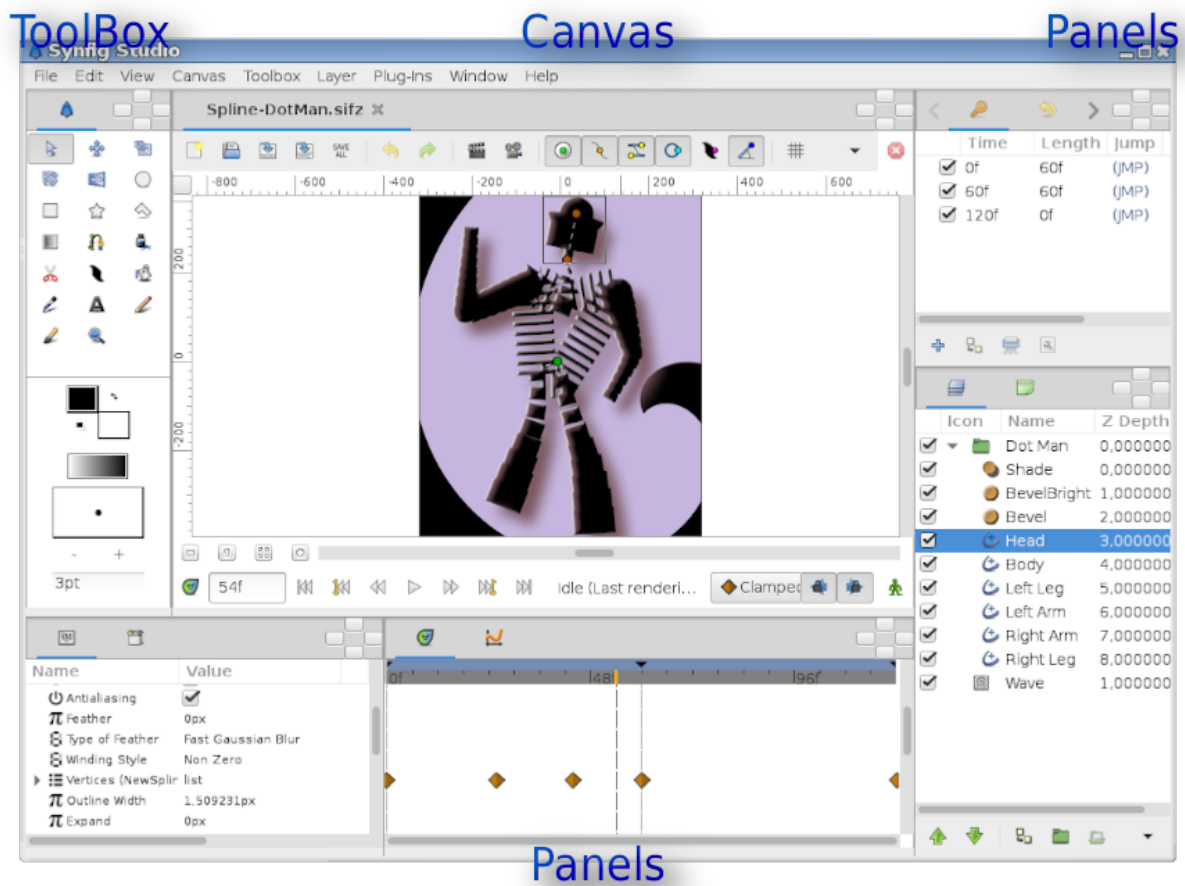


Fig. 1: Default interface layout of Synfig Studio

The center of the window is the *Canvas*. A new Canvas Window appears each time Synfig Studio starts. The window represents the Root Canvas, not that it means much to you at the moment, but that's OK — we're just trying to show you around. In the upper left corner of the Canvas Window, you'll see a button with a *caret*. If you click on this caret button, the *canvas window menu* will pop up. If you right-click in the canvas area and there is no *Layers* under the mouse position, this menu will also appear. So now you know where the most important canvas menu is.

The other part of the window (on the bottom/top, to the right/left) are customizable *dock panels*. Each dock contains a set of *Panels*, arranged horizontally or vertically. Some panels share the same space inside the dock and you can switch between them by clicking on their *panel tab*. You can rearrange the contents of dock panels as you wish by dragging the panel tab to where you want it. You can even create a new dock by dragging a tab out.

If you accidentally close a *panel* (by dragging it out, and closing the new dock that gets created), no worries. Simply go to the *Main Menu*, select *Window Menu* there and then click on the name of the panel you need.

The most important panels are:

- *Layers Panel* — shows you the hierarchy of the layer of your working canvas. It also allows you to manipulate these layers.
- *Parameters Panel* — shows you the parameters of the layer currently selected. When multiple layers are selected, only the parameters that the selected layers have in common are displayed.
- *Tool Options Panel* — shows you any options specific to the currently selected tool.
- *Navigator panel* — shows a thumbnail image of what the currently selected canvas looks like. You can also zoom in and move the focus around with this panel.
- *History Panel* — shows you the history stack for the current composition. You can also edit the actions in history.

There are also many other *panel* in Synfig Studio. If you have no idea what a panel does, simply hold your mouse over its icon and a tooltip will pop up describing its function.

2.2.2 Under the hood

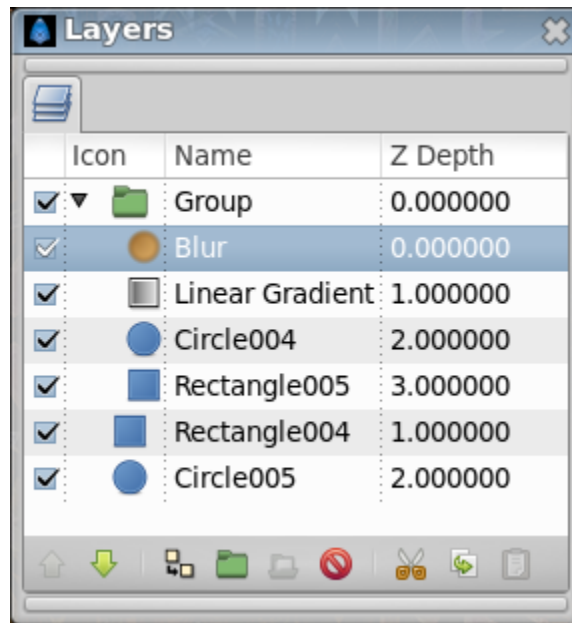


Fig. 2: Layers Panel

Synfig Studio, like most every other competent graphics program, breaks down individual elements of a *Canvas* into *Layers*. However, it differs from other programs in two major ways:

1. An individual layer in Synfig usually represents a single “Primitive”. I.e. a single region, an outline of a region, an imported image, etc... This allows you to have a great deal of flexibility and control. It is not uncommon for a composition to have hundreds of layers (organized into a hierarchy for the artist’s sanity of course).
2. A layer can not only add information on top of the image below it, it can also distort and/or modify it in some other way. In this sense, Synfig’s Layers act much like filters do in Adobe Photoshop or GIMP. For example, we have a *Blur Layer*, *Radial Blur Layer*, *Spherize Layer*, *Bevel Layer*, etc...

Each layer has a set of parameters which determine how it behaves. When you click on a layer (either in the Canvas Window, or in the illustrated *Layers Panel*), you will see its parameters in the *Parameters Panel*.

2.2.3 First steps

Let’s create something fun so that we can play with it!

First, go over to the toolbox and click on the *Circle Tool* (if you don’t know which one it is, just mouse over them until you find the one with the tooltip that says “Circle Tool”).

When you click on the Circle Tool, you should notice that the *Tool Options Panel* changed. But we’ll get to that later.

With the Circle Tool selected, you can now create circles in the *The Canvas window*. This works as you might expect — click on the canvas, drag to change length of the radius, and release the mouse button when you are done. Go ahead and create two circles (or more, if you fancy). If you accidentally release the mouse button before dragging, you end up creating a circle with 0 radius and it is effectively invisible! No need to worry, you can easily fix this. In the Parameters Panel, you can change the parameters of the selected object. If you just made a 0 radius circle, it should be the current selected object. You can change its radius to some value other than 0, say 10, and manipulate it to your liking with the handles later.

Note: Some users might experience the following problem: when you click and drag on the canvas using the Circle Tool, either nothing seems to happen or you end up making insanely huge circles. To fix this go to **File|Input Devices** and **disable all the devices** you can find there. If you have an extended input device that you want to use, such as a pressure-sensitive pen, then enable it in this screen. After this change Synfig will work as expected.

Now go back to the toolbox and click on the *Transform Tool* (the button with the arrow on it). After you do this, click on one of your circles. You will see a “bounding_box” (which is kind of useless at this point in time, but we digress), a green dot at the center, and a cyan dot on the radius. Those dots are called “handle”. If you want to modify the circle, grab a handle and drag it around. Easy!

You can select a *Layers* by clicking on it. If you want to select more than one layer, hold down **ctrl** key while you are clicking — this works in both the Canvas Window and the *Layers Panel*. Try it!

You can also select multiple handles. You can do this in several ways. First, you can hold down **ctrl** and individually click the handles that you want selected, but this can be tedious. However, there is a much faster method — just create a selection box by clicking the mouse and dragging it over the handles that you want.

Go ahead, select two circles and select all of their handles. With several handles selected, moving one handle will move all of them.

Note: Synfig Studio has an autorecovery feature. If it crashes, even if the current file has not been saved, you will not lose more than 5 minutes of work. At restart it will automatically prompt to recover the unsaved changes. Unfortunately history isn’t recovered yet.

The *Rotate Tool* and *Scale Tool* tools work much like the Transform Tool, except in the case where you have multiple handles selected. It is much easier just to try, than read about it. Select a few circles, select all of their handles, and try using the rotate and scale tools.

Note that tools manipulating with handles have options associated with them. If a particular tool isn't doing what you want, take a look at the *Tool Options Panel* to see available options.

2.2.4 Linking

Now let's try *Linking*. Suppose we always want these two circles to be the same size. Select two circles, and then select both of their radius handles (the cyan dots).

To select multiple handles, either drag a rectangle around them, or select the first one, then hold the `ctrl` key while selecting the rest. Once you have the two radius handles selected, right click on either of them and a menu will pop up. Select *Link*. Boom. The parameters are linked together. You can prove it to yourself by selecting just one of the circles and changing its radius — the other one will change as well. Neat stuff, eh? This is how *Outline Layer* are attached to their *Region Layer* — but we're getting ahead of to the chapter.

Linking is a fundamental concept in Synfig. You can create links not only for handles, but also between parameters as well by selecting multiple layers, right clicking on the parameter in the Parameters panel, and selecting *Link*.

2.2.5 Color selection

Let's say you want one of the circles to be a different color. If you look in the toolbox below the tools, you'll see the outline/fill color selector, the outline width selector, and some other stuff like the default blend method and gradient. The outline/fill color widget works exactly as you might expect — you can click on the fill color, and a modest color chooser will appear. Now you can change the color pretty easily.

But sometimes you just want to click on a color and go. This is where the palette editor tab comes in.

Click on the Palette Editor panel tab and have a look — it's the one with the palette-ish looking icon. Clicking on colors with the left mouse button will immediately change the default outline color and clicking with the middle mouse button will change fill color.

That's all great, but we still haven't changed the color of the circle. There are three ways to do this. The first is to click on the *Fill Tool* from the toolbox, and then click on the circle in the Canvas Window. Boom. Circle changes color. This works with more than just circles. Also, you can select the circle layer you want to modify, go to the Parameters panel, right-click on the Color parameter and select *Apply Fill Color* or *Apply Outline Color* at your preference. Or simply double-click on the Color parameter - a color selector dialog will show up, and you can just tweak away.

Try playing around with the circles for a bit. Muck around with the parameters, and see what happens. To get you started, try out to set the *Feather Parameter* to 5.

2.2.6 Digging deeper

Of course, so far you just found out how to use the basic features of Synfig Studio but not how you animate a drawing. This is covered in the next section .

You can download an archive which contains the project used for wiki illustration *Interface.sfg* You can download the project used for the illustration of the illustration *Spline-DotMan.sifz*

2.3 Animation Basics

2.3.1 Introduction

Creating an animation in Synfig Studio is really easy. It basically means to change a drawing — you just need to create the first stage and last stage of a change, and Synfig takes care of the steps in between.

Let's look at a simple example. Consider a moving light like the one at the front of the Knight Rider car. Drop the realism, you get a circle moving from left to right and back. In other words, you need to create three 'steps' or 'stages':

1. The circle is on the left.
2. The circle is on the right.
3. The circle is back on the left.

2.3.2 Setting up the workspace

Let's do it. Start Synfig Studio. A new file is created at the start automatically. Click the *canvas window menu* (between the horizontal and vertical rules, in the top left hand corner of the canvas), then select "Canvas Properties" or select "Canvas Properties" from the menu. The *Canvas Properties Dialog* will appear.

Give a name and description for your canvas, then click **Apply** (don't click **OK** yet — we're not quite done with the Properties dialog). Go to the **Time** tab and make sure to edit **End Time**. Change "5s" to "2s" — that will make our animation 2 seconds long.

Note: Good to know: Synfig and time representation Synfig Studio can display the time in various formats (Timestamp). You can configure the Timestamp in Preferences. If the endtime is not displayed in seconds then go to "Edit Preferences System" and change the Timestamp into: (HHh MMm SSs) FFf. More about the Timestamp and its settings can be found here ([Preferences_Dialog#Time_Stamp]).

Now click "OK", select the *Rectangle Tool* and create a simple black rectangle that will serve as our background. It's not necessary to make it cover the whole canvas.

Now we need a circle. Change the fill color to red, select the *Circle Tool* and create a circle. It doesn't matter if it's not perfect: You can edit it. Select the *Transform Tool* and click the circle. It will go into an editing mode which is easy to detect by the small green dot in the middle and the white rectangle around it. You can move the circle by grabbing it on its green dot (the *Origin*) in the middle.

These are the first steps to draw an object and to move it, but not an animation yet, you may say. Indeed. Let's have a look how this works.

2.3.3 Adding movement

In the beginning, you entered a value of 2 seconds in the Properties dialog. Because the length of your animation is non-zero, your canvas window (the one where you draw) has a grey time slider at the bottom, the *Timebar*. You can click on it, and a small orange indicator will appear indicating your position in time. Try clicking in several places on the time slider and notice that the entry field on the left of the time slider is changing its values to something like "12f", "1s 15f", etc. You can set your position on the time slider by changing values in that field. For example, if you enter "1s" and press **enter**, the orange indicator will move in the middle of the time slider, and entering "2s" will move it to the end of the time slider.

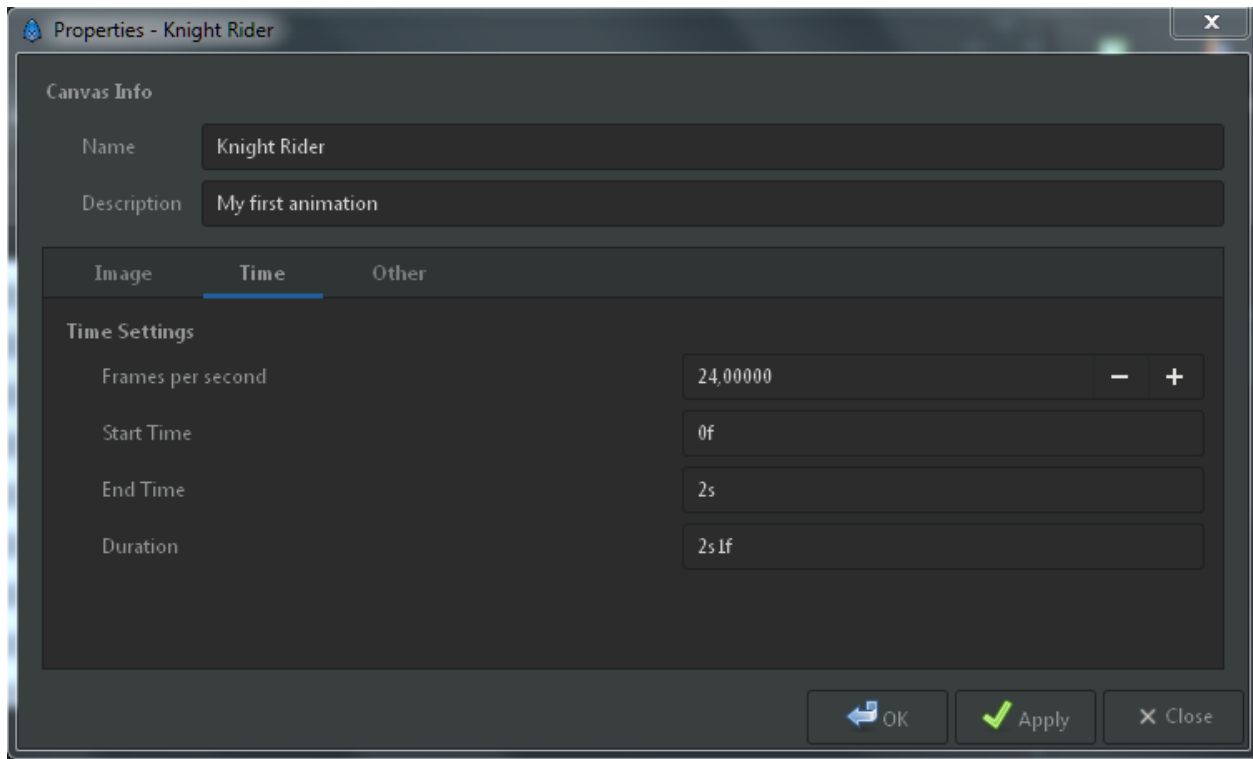
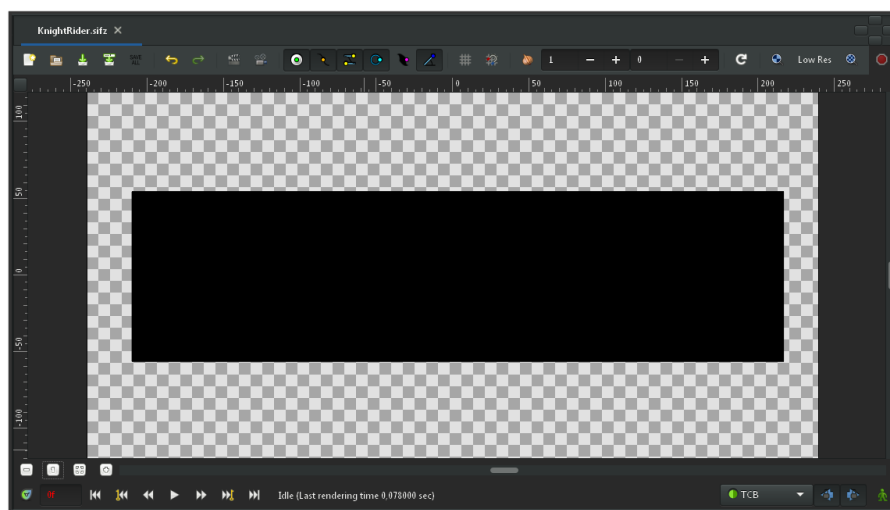
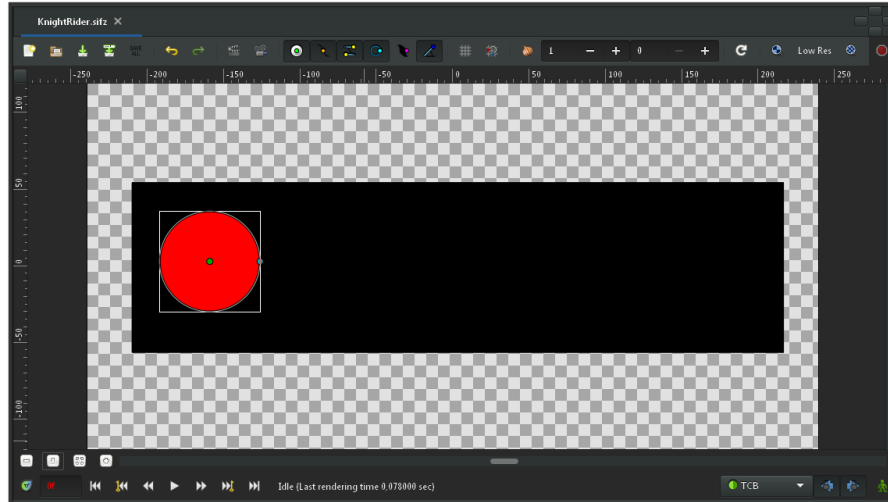


Fig. 3: Canvas Properties Dialog

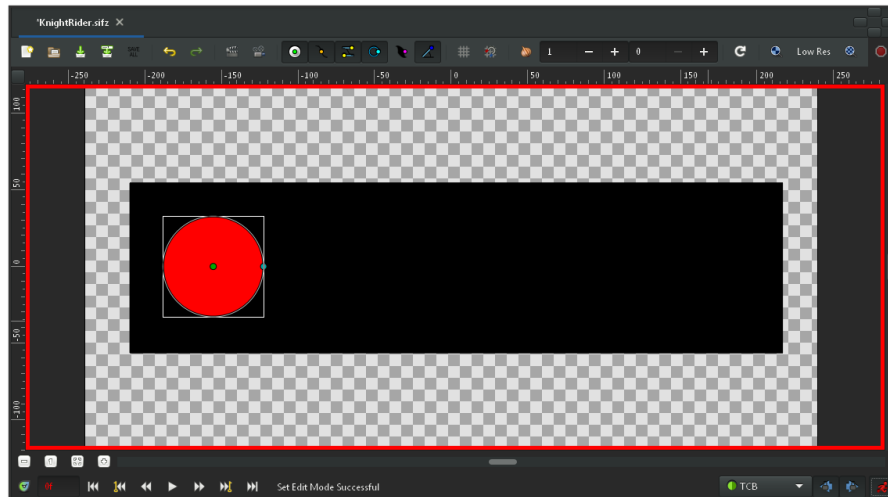




Note: At 2s the orange indicator won't be visible. That's because "2s" is at the far right boundary of the time slider, putting the indicator out of view.

You may notice that nothing changes on the canvas at this point. Return to "0s" and switch to *Animate Editing Mode* by clicking the green man button to the right of the gray time slider. The canvas will display a red outline; it reminds you that changes to your objects now affect your animation at the time shown in the time slider.

In animate editing mode, every change to your objects creates a *Waypoint* that associates the changes with the current time. As you will see, Synfig can create smooth intermediate changes between waypoints, and you can even choose the way in which the intermediate changes take place. You will probably find it helpful to associate some or all of your waypoints with *Keyframe*.



Previously, three "steps" or "stages" were mentioned. These are represented by *Keyframe*. (Just in case you're familiar with video encoding: No, it's not the same!) A keyframe is an image in time where something important happens with your objects.

Note: Default keyframe at 0f

When creating a new project, a default *keyframe* is already set at 0f. If for any reason you do not have this default

keyframe, go to the *Keyframe Panel* — click on the little tab with the small key icon in the bottom window — to edit keyframes. Now press the small button with the “plus” sign and you should get a new entry in the list displaying 0f, 0f, (JMP).

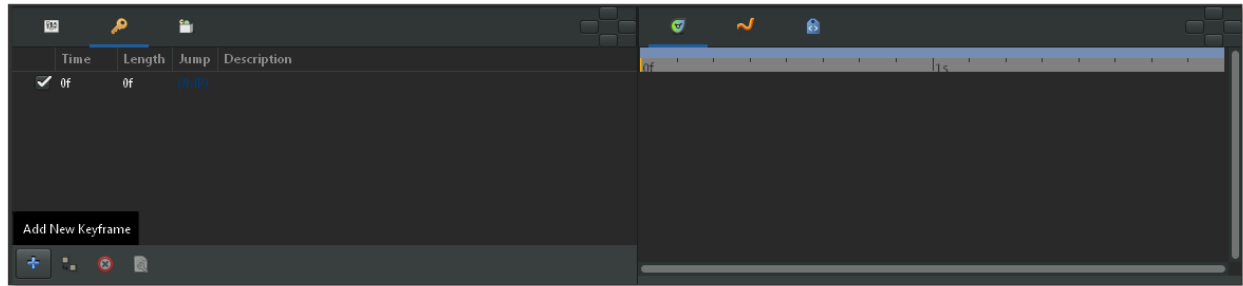
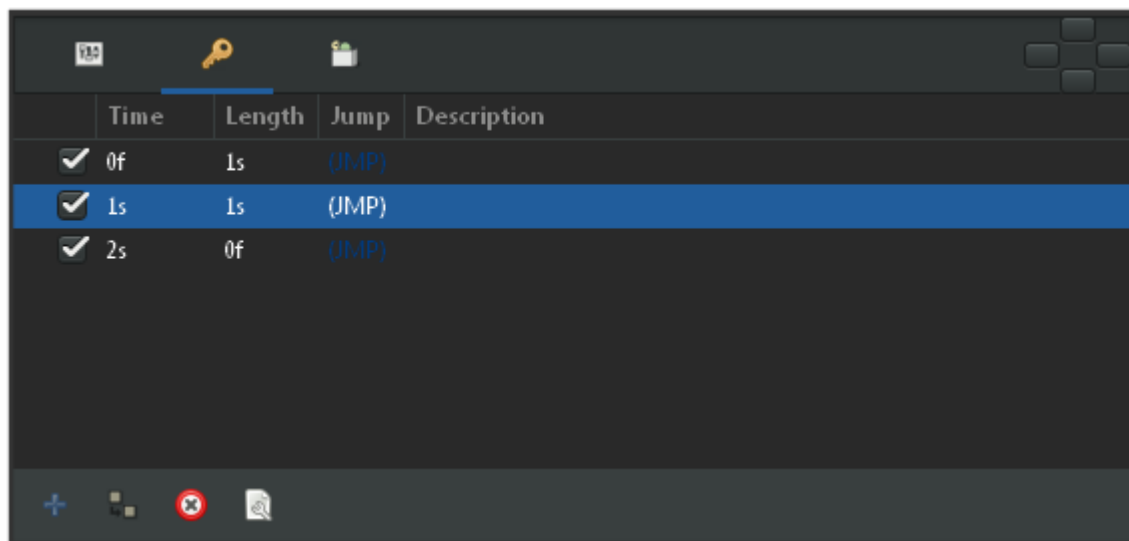


Fig. 4: Keyframes Panel and Time Track Panel

Now, go to the “1s” mark in the time slider. The small orange indicator should move there. Then add another keyframe by clicking the small plus sign. Repeat the process with the time slider indicator set to “2s” (it’s at the end of your animation). You should now have three keyframes in the list.



2.3.4 The s’s and f’s: Understanding the Timeline

By now, you may have figured out what those mysterious “1s 10f”-type marks represent. They indicate a specific point on the timeline, expressing a location in terms of seconds (s) and frames (f).

By default each second is divided into 24 frames, much like a meter on a measuring tape is divided into 100 centimeters. The frame markings begin at zero (0) and go up to 24, whereupon a new second is entered and the framecount returns to zero.

For example, when five whole seconds and three frames have passed, using this timeline notation would be “5s 3f”.

2.3.5 The Keyframes Panel

The *Keyframe Panel* is rather easy to understand. It displays **Time** which is basically the start time, **Length** which is self-explanatory, **Jump** which we'll cover next, and **Description** which is, again, self-explanatory.

You might be wondering about the entries called (JMP). In fact, these are links just like web links: click them, and the indicator in your time slider will jump to the correct time.

You can use this to edit your image for a given moment in time. For instance, you can now jump to the first second "1s", and move the red circle to the right. There! You made your first movement, your first animation with Synfig!

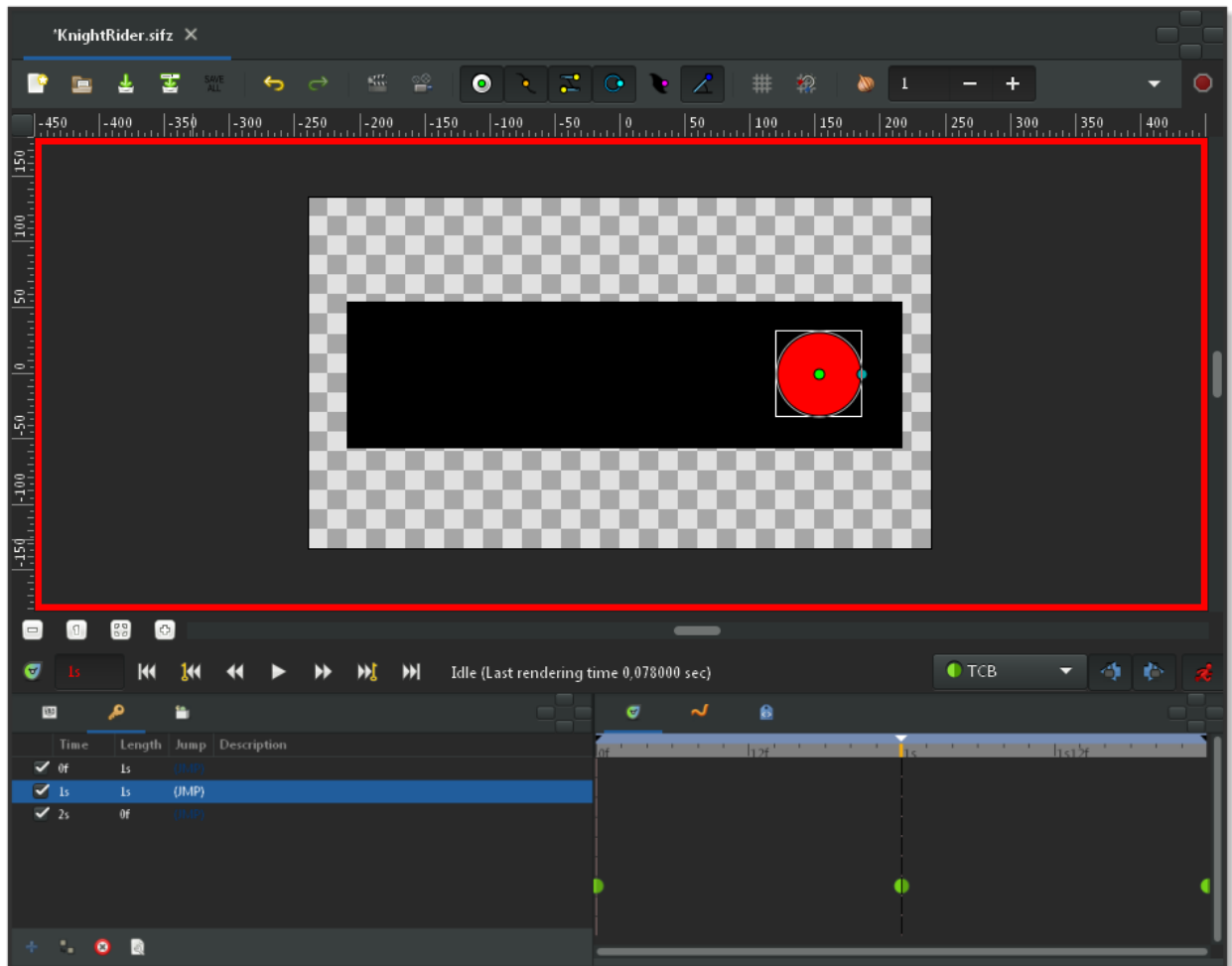


Fig. 5: Canvas Window, Keyframes Panel and Time Track Panel

Wondering where the animation is? Just click to an arbitrary position on the time slider: You will note that the red circle is in a new position, one that you didn't specify! So what happened? Synfig figured out what you would like to do, namely move the circle, and drew all the images between these states. Each image will later make a frame in your animation and the circle will appear to be moving.

Note that you don't need to go to the last keyframe at "2s" and move your circle back to the left. Keyframes make Synfig remember the image states at particular times. That's why when we modified the circle's position at "1s", it stayed on the left at "2s" (as well as at "0s"). If you switch back to the Parameters Panel, and look at the *Timetrack Panel* you will see that three orange diamonds (or green dots depending on the default interpolation) appeared on the right of the *Origin* parameter. Those are called *Waypoint*, and they represent times at which object's parameters, like location or color, are instructed to take on specified new values.

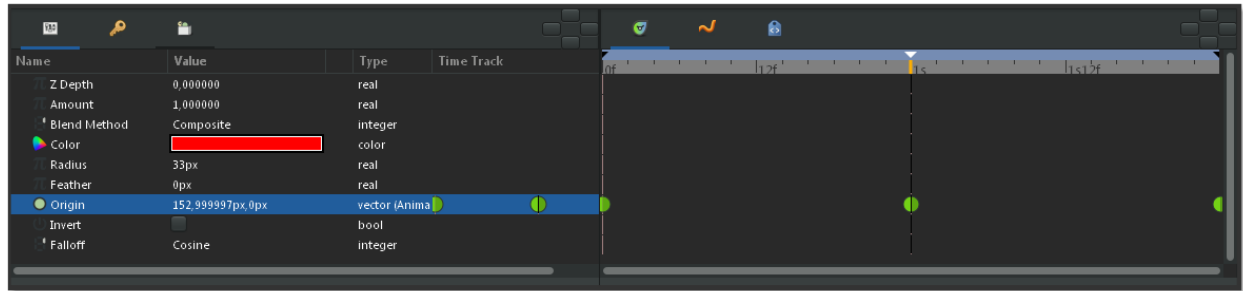


Fig. 6: Parameters Panel and Time Track Panel with green Waypoints

2.3.6 Rendering your animation

Before you can see your animation, you need to render your work. There are two ways to do so: using the Synfig Studio (what you have been using so far) or the command-line program called “synfig”.

Let’s try the first way. Leave the *Animate Editing Mode* by clicking on the red man icon in the timeline editing widget, and save your file; for instance under the name “BasicKnightRider.sifz”. Then go to menu in the Canvas Window (*canvas window menu* button in the upper left corner) and select “FileRender” or click on the render icon. Change the filename to “BasicKnightRider.gif” in the same location where you saved “BasicKnightRider.sifz” and choose gif target format instead of Auto, then click Render. Depending on your processor speed it should take a few moments, but finally the image window status bar (located on the bottom of the window) should say “File rendered successfully”.

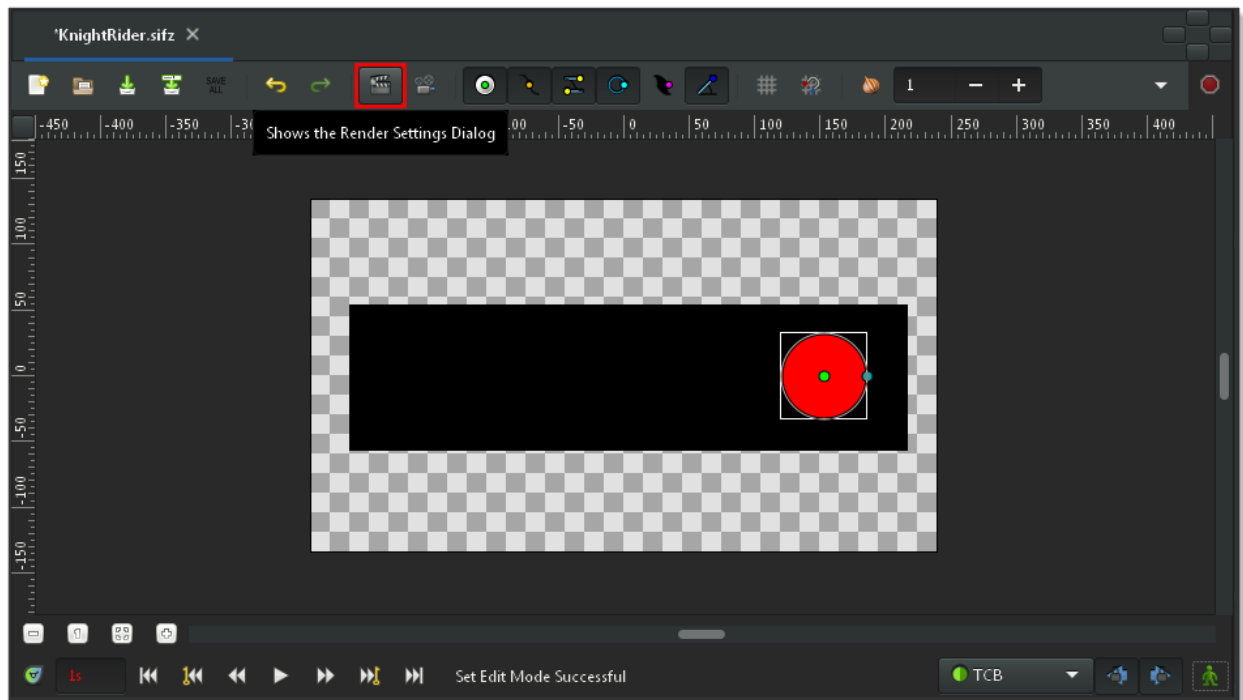


Fig. 7: The Render Icon

Note: The “magick++” target (if it is available) produces much better gif files than the “gif” target because it can optimize the palette for the image.

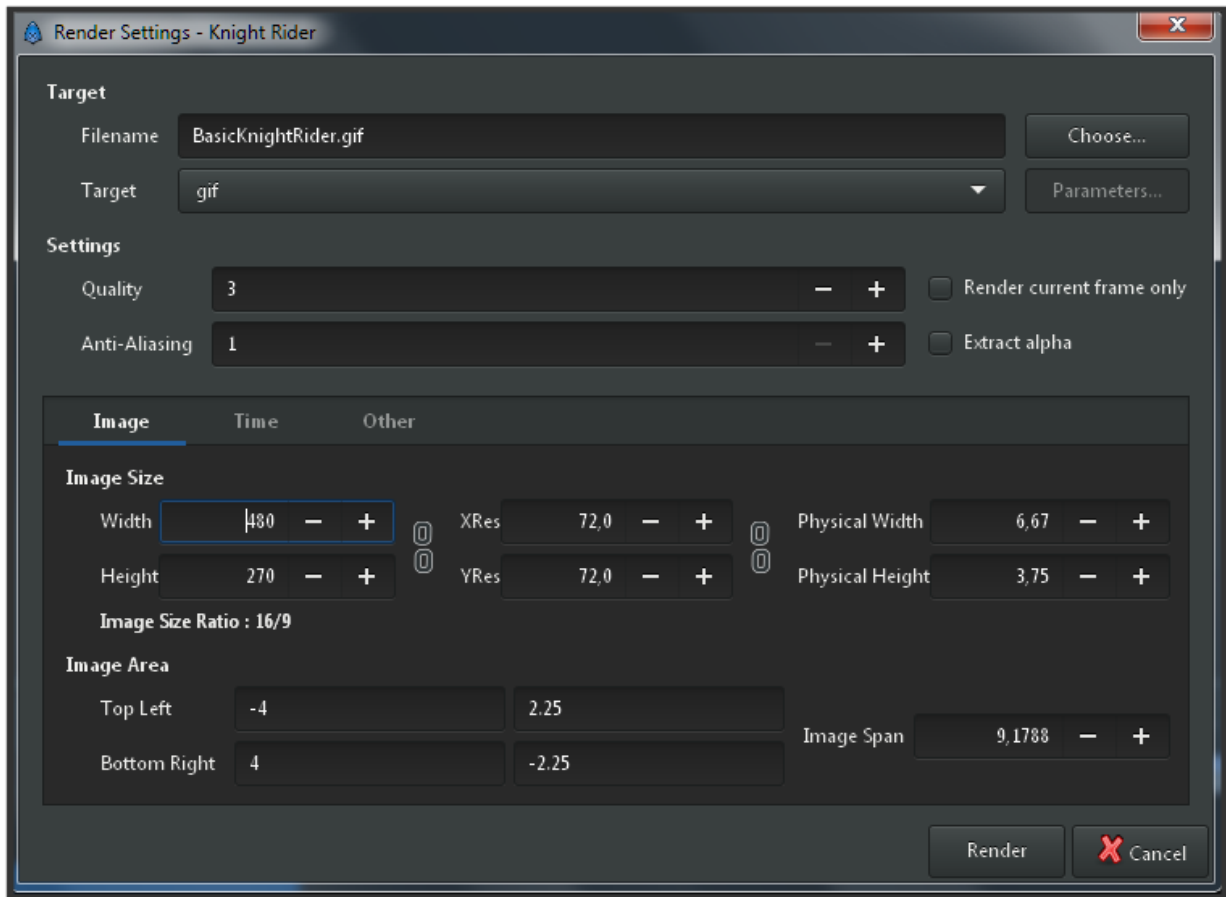


Fig. 8: Render Settings

Open `BasicKnightRider.gif` in Firefox or another application that is able to show animated gif's. However, Firefox will replay the GIF all the time which makes your short animation a rather long one. If you're now seeing a red circle moving from the left to the right and back, congratulations! You just made your first animation!

Note: You can also preview your animation. Press the “caret” menu button in the upper left corner of the Canvas Window and choose “FilePreview”.

If you would rather use the command line instead of the menu to render your animation, then open a terminal (on Windows, go to “StartRun”, type `cmd` and press `enter`), change to the directory you saved the file in, and type something like this:

```
synfig -t gif BasicKnightRider.sifz
```

A few messages appear that don't matter right now. Depending on your processor speed it should take a few moments, but finally a line like this will appear:

```
BasicKnightRider.sifz ==> BasicKnightRider.gif: DONE
```

Then you are done and can view your animated gif using Firefox or another program as mentioned above and it should look like this.

Fig. 9: Rendered Animation

2.3.7 Conclusion

Of course, the position of an object is not the only thing you can change with Synfig Studio. Other possibilities include its size, its outline, its color, etc. Synfig comes with several example files that should let you dig deeper into the possibilities.

Now, let's continue with the next manual chapter: *Working with layers*

You can download the project used : `BasicKnightRider.sifz`

2.4 Working with Layers

2.4.1 Introduction

In the [previous tutorial](#), you made your first simple animation by changing the attributes of *Primitive Layer* objects, such as: position, color, and size. These simple types, however, are seldom sufficient to create advanced characters and objects. To do so, Synfig uses *Layer*. They are similar to layers used in other drawing applications in that they are used to separate different elements of an image.

However, Synfig's layers have following important specific:

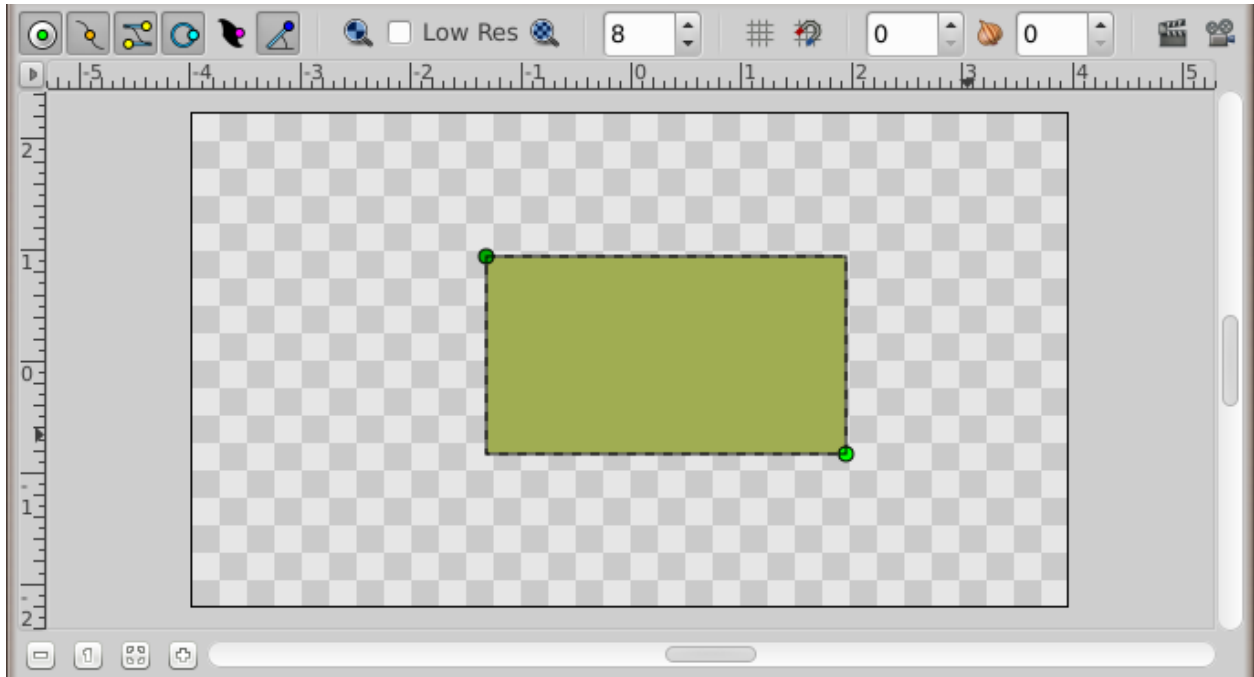
1. Every object or *Primitive Layer* gets its own layer.
2. You can organize layers into hierarchical groups.
3. You can use upper layers to change the behavior (or look) of underlying layers. Those are called *Filter Layer* or effect layers.

As you will see, layers are an extremely important aspect of Synfig, much more so than most graphics programs. Understanding the concept of layers is an important part in understanding how Synfig works.

2.4.2 Combining layers

So let's look at a simple example of how we can combine two layers to create a gradient effect on a rectangle.

Create a new file with any duration. There's no need to bother with a timeline at this point. Next, create a simple rectangle with the *Rectangle Tool*.



Pick the *Gradient Tool* from the (Composite ([Blend Method Parameter#Composite]) Blend Method), press the left mouse button on the canvas, drag to change the gradient direction and release the button when you are done. You should note that another layer was added in the *Layers Panel* called **Gradient**. This is nothing special.

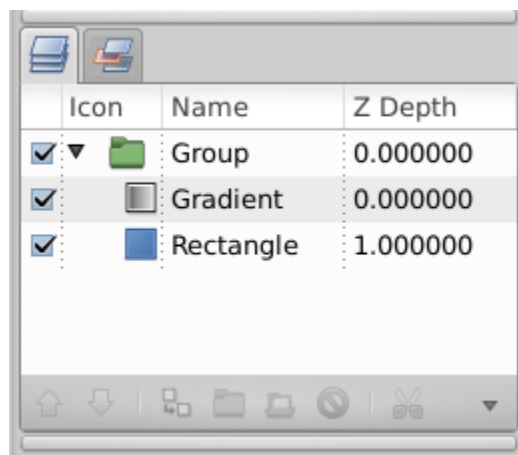
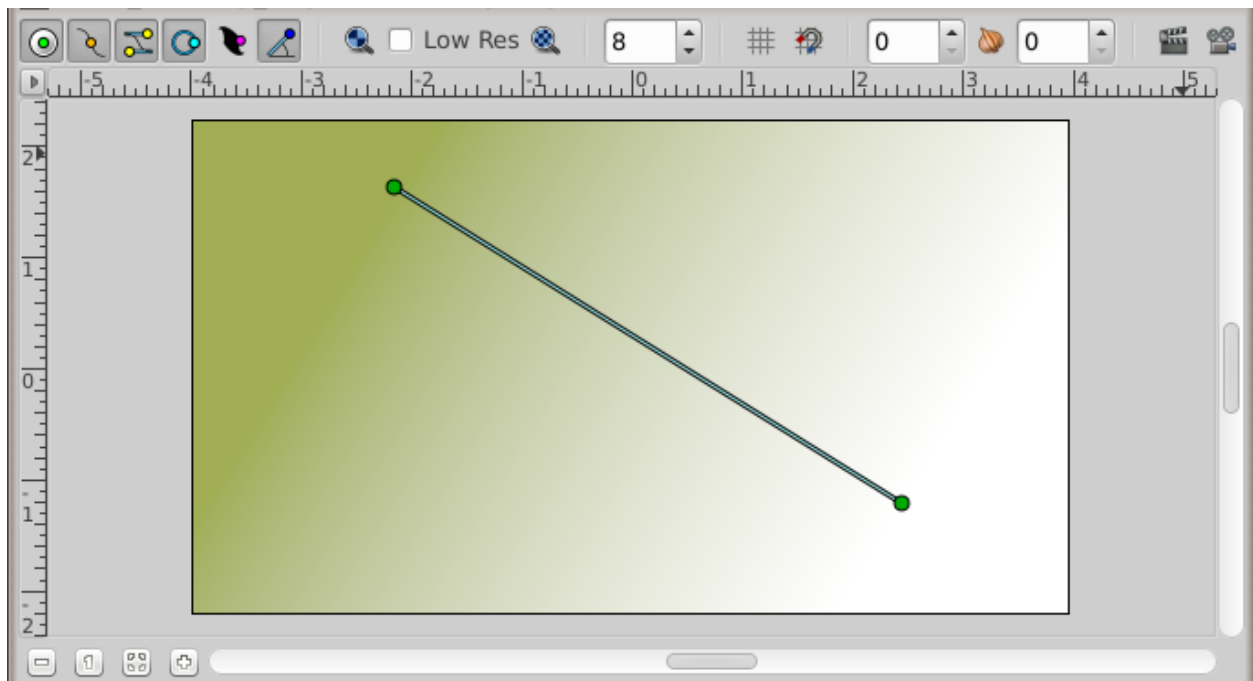
Note: If you see no gradient but just a plain color, that means that you probably just clicked on the canvas without dragging your mouse. To fix that pick the *Transform Tool*, click into the canvas to activate the gradient's handles. You need to grab the one you see and move it a bit until a gradient appears.

Note: If you see a spiral gradient, go to the *Tool Options Panel* and change Layer Type to linear by clicking on the far left icon.

You now have a gradient, but it is not what you wanted: it spreads across the whole canvas. The goal was to have a gradient in the rectangle. So, let's fix this now.

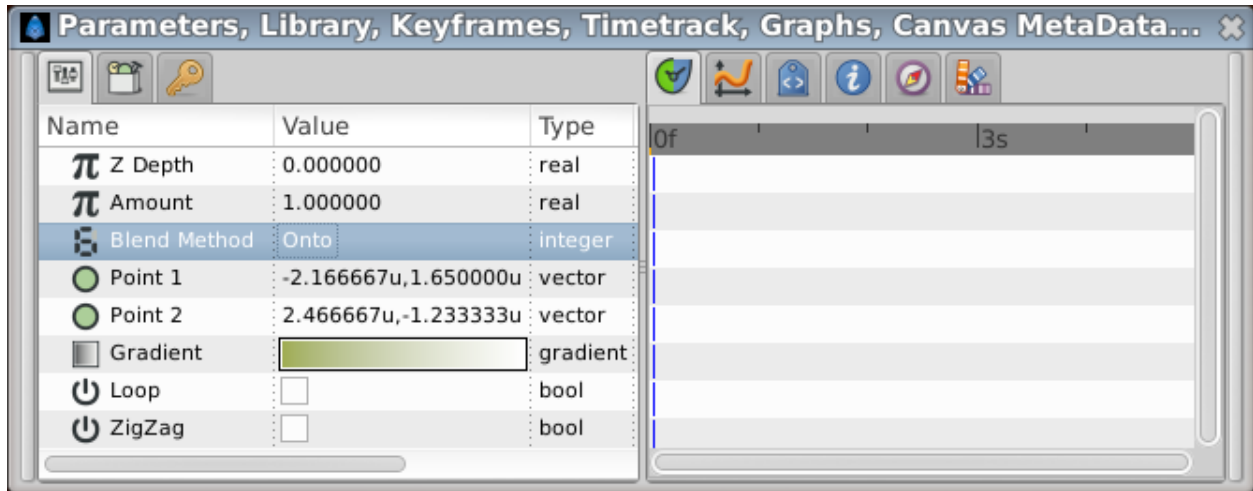
In the *Layers Panel*, select both the gradient and the rectangle layer. Then, right-click and select **Group Layer** from the menu. The view of your Layers Panel should change now, showing a small box called *Group Layer* with a  in front. By clicking on the  you can expand the group layer to see its contents, your previous two layers: the gradient and the rectangle.

You can treat this layer like any other layer — move it around, duplicate it, copy and paste it. If you want to change the name of it to something more descriptive, just select the layer in the layer tab and click on its label. Then you just edit it in place. You can do this for ANY layer, and are strongly encouraged to do so.



2.4.3 Using locality

However, there is still a problem: the gradient still covers the whole canvas although we wanted it to be restricted to the rectangle. To do so, activate the gradient layer in the Layers Panel. Now go to the *Parameters Panel* (by default it resides in the bottom window), and search for the attribute called *Blend Method Parameter*. Double-click the entry and select *Onto Blend Method Parameter:Onto* from the drop-down menu.



The gradient should now be restricted to the rectangle. Congratulations! You just made your first effect by interacting layers with Synfig.

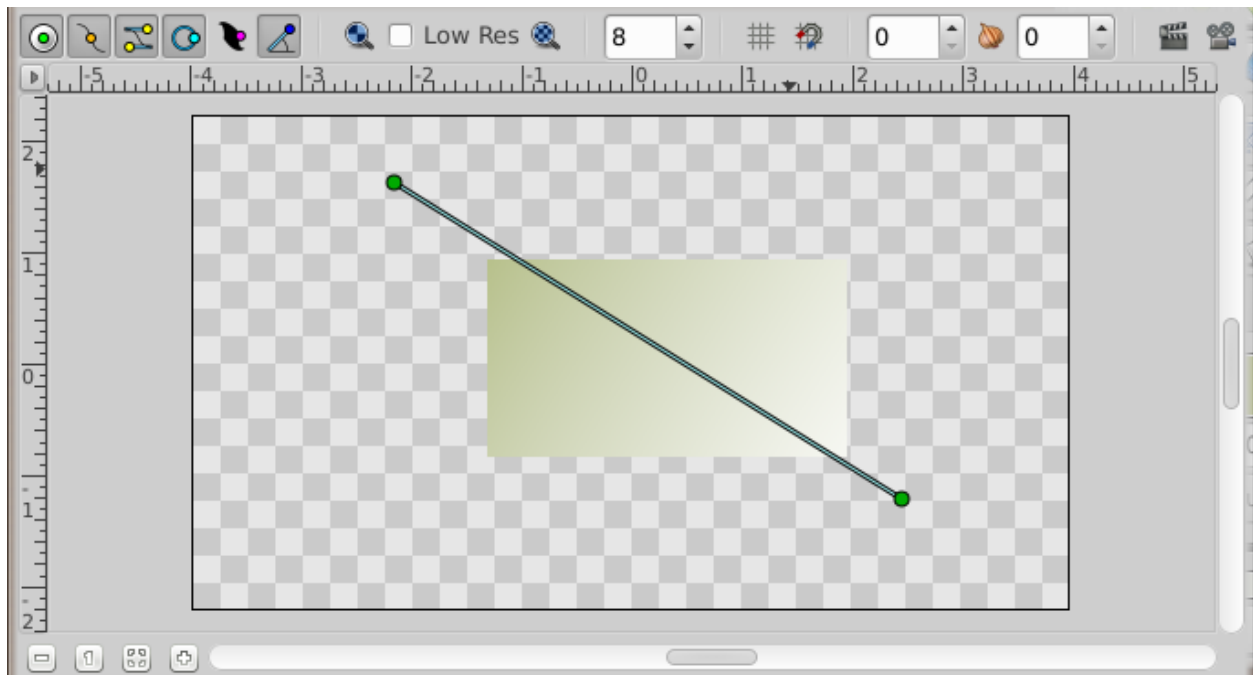


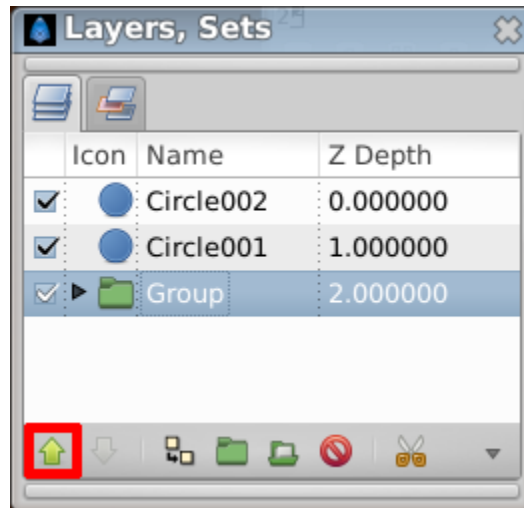
Fig. 10: The gradient is restricted to the rectangle. You will find more information about the various Blend Method in the *Blend Method Parameter* documentation.

If only for the additional organization, grouping layers dramatically improves the ease of use of Synfig Studio. But lots of programs can do this. The concept of *scope* as just demonstrated sets Synfig apart from other programs with layer hierarchies. The key point is that a layer, inside its own group, will **only** modify the data contained by all the layers

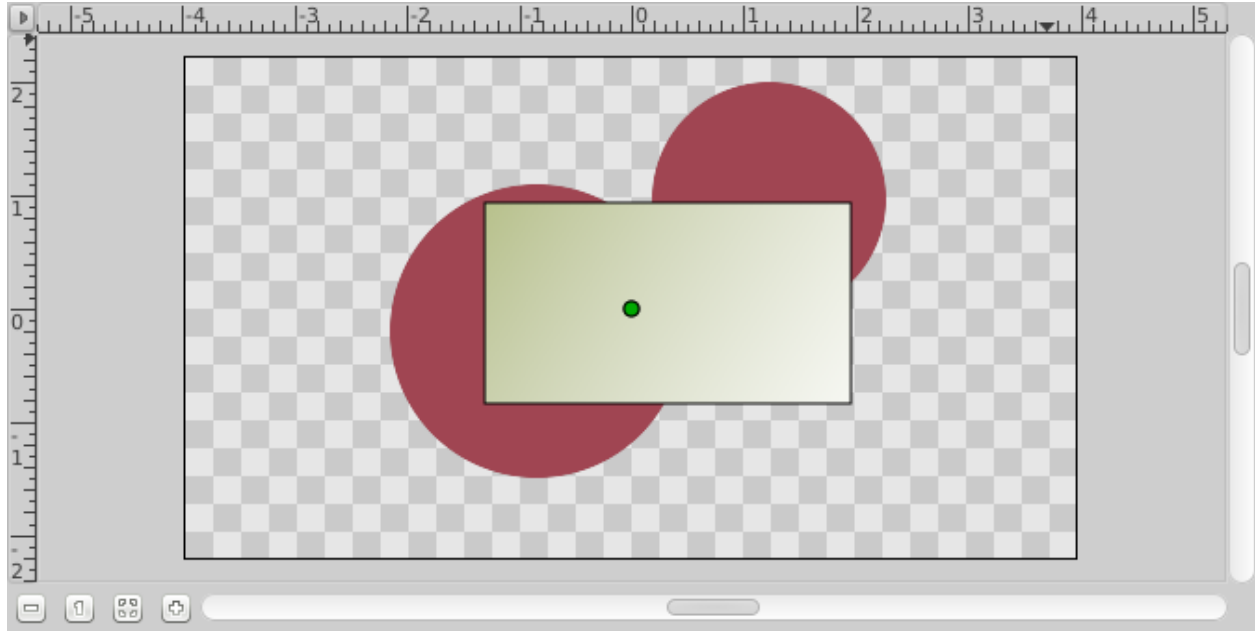
and groups **below** it. In other words, if you were to throw a *Blur Layer* on top of the layers inside the *Group Layer* we created, it would just blur them — anything under the group layer would not be blurred! Let's try it.

2.4.4 Using layers to modify other layers

Make sure you have the group layer selected and create two red *Circle Tool*. They will appear on top of the group layer. Select the group layer and use the *Raise Layer* button in the Layers Panel to place it on top of the circles.

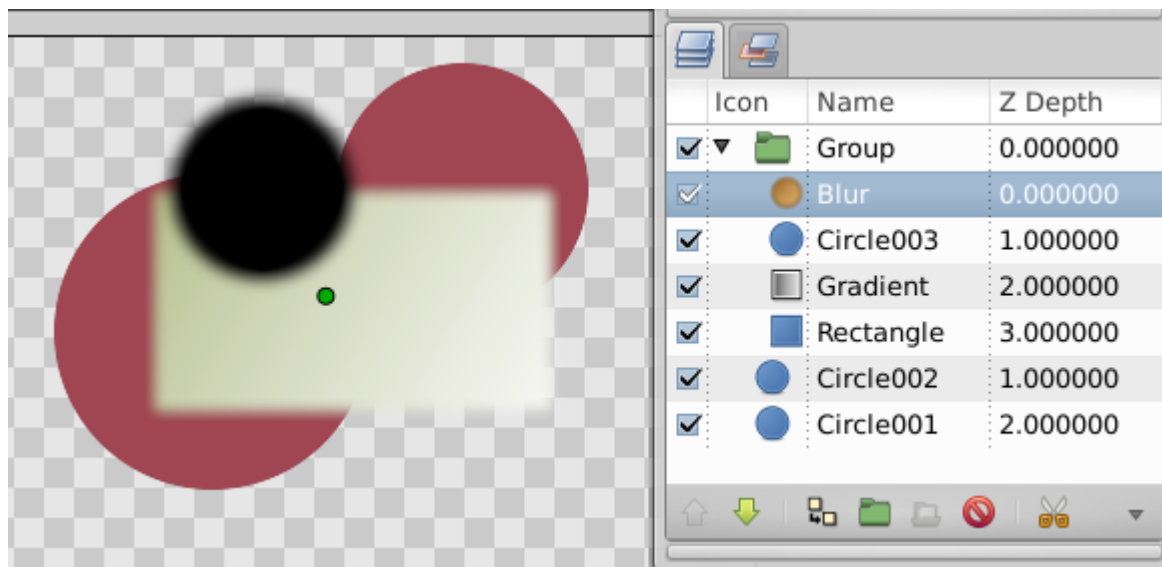
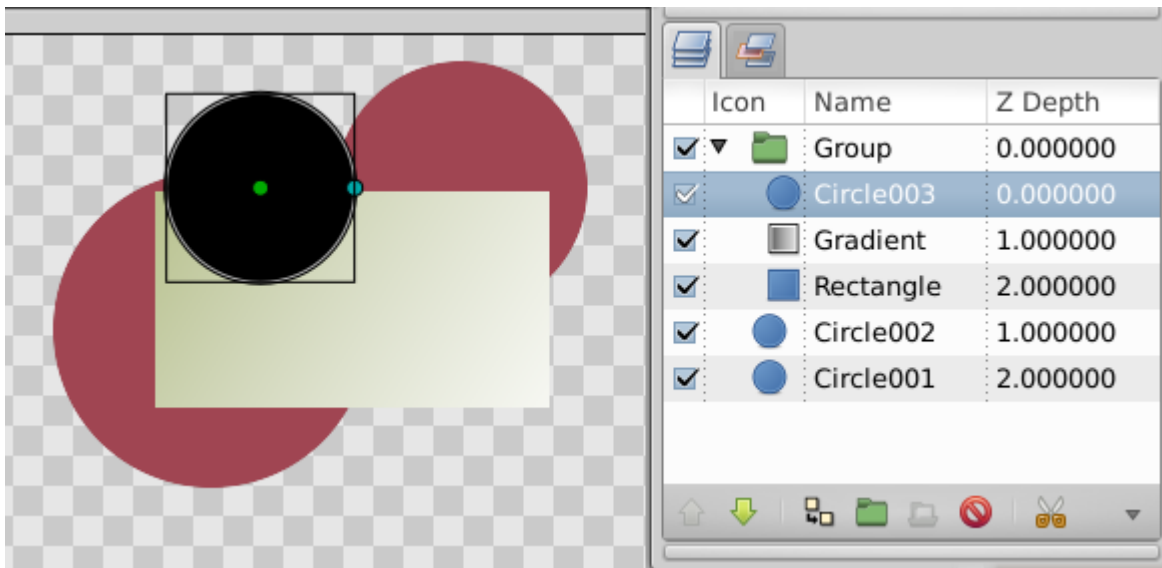


Now our group layer (with rectangle and gradient) is in front of those two circles.



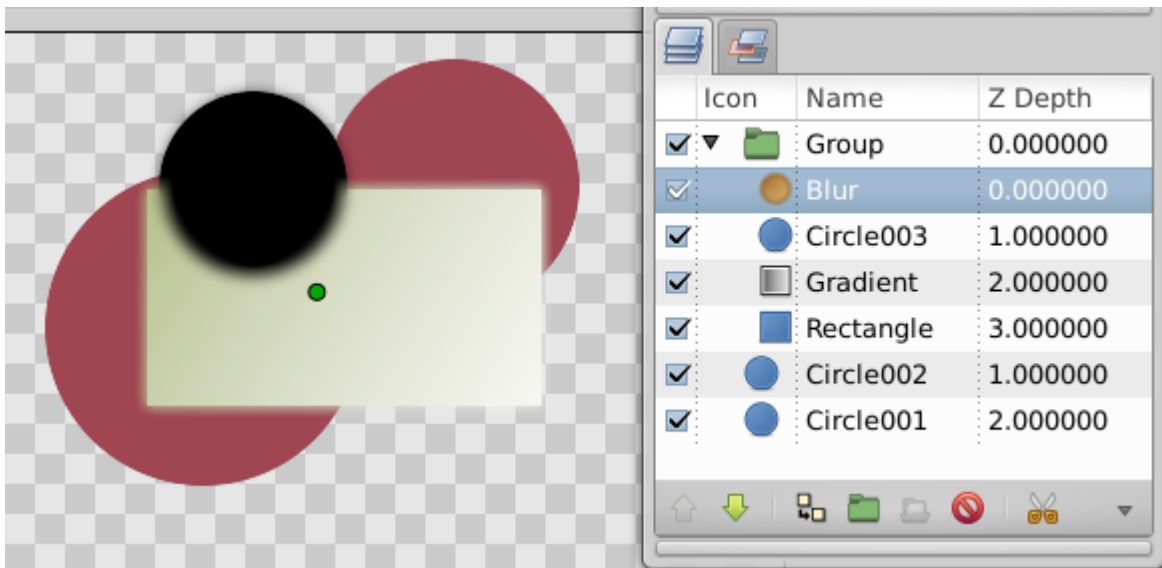
Expand the group layer to show its contents, and select the top layer inside of it (should be the gradient layer). This is where we want to insert the new layer. Create another circle filled with a black color. The black circle layer will be created over the gradient layer inside the group layer.

Now, right click on the black circle layer in the layers panel and a popup menu will appear. The first item in that popup is *New Layer*. Inside of the *New Layer* menu, you'll see several categories of layers you could create, but what we want is a blur, so go to the blurs category and select the *Blur* layer (so that would be *New Layer | Blurs | Blur*).



It blurred! Has the blend method for newly created blur is **Straight** (if the default blend method in the Toolbox is set to **By Layer Default**) it blurred all around the outside edge of the contents of the group layer. You can change the default blend method *New Layer Defaults:Blend Method* for new layers from the *New Layer Defaults* section of the Toolbox.

Note: In the Synfig version before “0.62.02”, the blend method for newly created blur layers was defaulted to “Composite”, that cause the outside edge of the contents of the group layer is still sharp. What we want is a blend method of “Straight”. Just select the blur layer, and change the Blend Method to “Straight” in the Parameters Panel.



Ok, now we have all of the contents of the group layer blurred, but everything under it is sharp! This is because the effect of the Blur Layer over the underlying layers is limited to the scope of the group layer because the blur layer is inside it.

You can download the `File:Doc Adding Layers.sif` used for this example.

2.4.5 Digging further...

If you care to look into Synfig’s main menu under **Layer|New Layer** you will note quite a lot of different possibilities for making layers. Several of them sound rather unusual, like **Transform|Rotate** for example. You can use this to add new attributes to your objects. And just like other, basic attributes in the *previous animation tutorial* *Previous Tutorial*, you can change them to be different on certain *Keyframe*. Synfig will take care of interpolating the steps in between.

For example, you could create a shape and add a *Rotate Layer* over it. Combine this with the lesson learned in the *Previous Tutorial* and you will obtain a rotating effect. This technique is used for the creation of *Cut out animation*.

Let’s continue digging further on shapes creation *Creating Shape*.

2.5 Creating Shapes

2.5.1 Introduction

Basic *Primitive Layer* such as *Circle Layer* or *Rectangle Layer*, that we have learn to work with *previous tutorial*, are all great but they are pretty much geometrically inflexible. What about creating more complex shapes? To do this, we use the *Spline Tool*.

A video on this subject is available [here](#).

2.5.2 Spline Tool

In Synfig, the construct for describing shapes is called a Spline. This is roughly analogous to a “path” in other programs, except that it is strictly a [cubic hermite spline](#).



Before we start with the Spline tutorial, let's look at some additional notes on how Synfig works. When you click on the *Spline Tool*, you will see that the handles from your currently selected object (if there was one) will disappear, but the layer(s) will still remain selected in the *Layers Panel*. This is normal. Anything you create with the Spline Tool will be inserted above the currently selected layer. Keep in mind that if you want to insert a shape somewhere, you

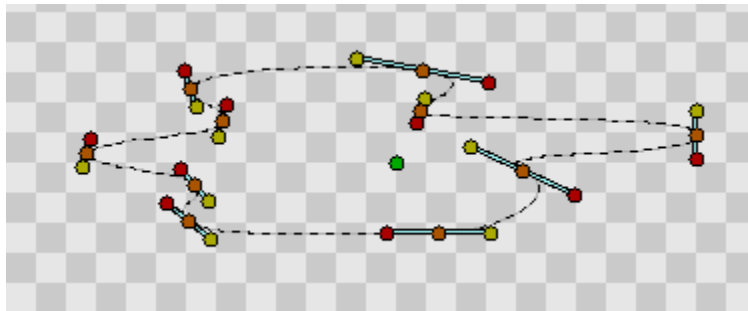
should select where you want to insert it before you go into the Spline Tool — changing the selection afterward will automatically swap you back to the Transform Tool.

Now, go ahead and click on the **Reset Colors** button in the lower left corner of the FG/BG color widget in the *Toolbox*. This will reset us back to the default black and white. Also, set the line width ([Spline Tool#Options]) to something nice and thick — 10pt should do the trick.

After you switched to Spline Tool, take a look at the *Tool Options Panel*. Make sure that only **Create Region**, **Create Outline** and **Link Origins** are checked.

Clicking with your mouse in the canvas will place vertices. While you are placing a vertex, you can drag out its tangent by dragging the mouse. Do this over and over, and you construct a Spline.

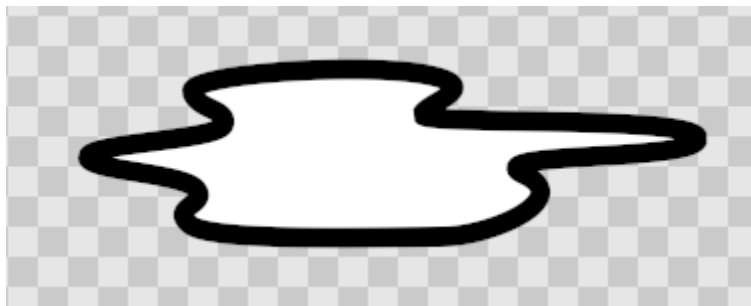
Keep in mind, however, that during this construction, there is nothing stopping you from just moving it if you don't like where you placed a vertex or a tangent. Honest! If you want to remove a vertex, right click on it and select **Delete Vertex**. Want to split the tangents? Right click on the tangent and hit **Split Tangents**. Want to loop the Spline? Right click on the first vertex and select **Loop Spline**.



When you are finished placing vertices, you must exit construction mode in order to actually create the Spline layer(s); there are 2 ways to do this:

1. Switch to another tool.
2. Press the **Create** button at the bottom of the Tool Options Panel (it's the icon that looks like a gear).

For now, just go ahead and switch to the *Transform Tool*, because we are done with the Spline Tool.



2.5.3 Editing Splines

Ok, we now have a nice pretty white region with a thick black outline. Since we checked `Create Region` and `Create Outline` in previous steps, you'll notice that there are two layers that we have created — the *Outline Layer* and the *Region Layer* in the Layers Panel. Despite the fact that they are two separate layers, their vertices parameter has already been *Linking* — so you can select either one and move its handles around and the other one will also change.

If you want to manipulate the vertices *after* you have created the layers, it is very easy to do so. Just click on one of the layers and have at it. If you want to remove a vertex, right click on it and hit `Remove Item (smart)`. Want to insert a point somewhere? Right click on the segment where you want to insert something and hit `Insert item (smart)`.

This may appear to be leading to a mess of layers. And yes, if you aren't using the software properly, that is exactly what you will get. But there is a way to make this more sane. As mentioned in the *previous tutorial*, you can *Group* layers into hierarchy.

One quick thing to mention before I finish up. You can change the width of an outline at each vertex. You do this by selecting the outline layer (NOTE: you must select the Outline Layer, the Region Layer has no width data) and tweaking with the width handles. By default, these are masked. To show them, press `Alt+5` or click `Toggle width handles` button at the top of the canvas window (the fifth one from the left). Repeat to hide them again. You can also see other things to mask via the *Canvas Menu Caret*: `View|Show/Hide Handles`.

2.5.4 Using tablet to draw shapes

If you have a *graphic tablet* you can use Draw Tool to create Splines.

Synfig Studio supports pressure sensitivity, but you need to configure it first. Go to `File|Input Devices...` from toolbox menu. In the Input dialog find your tablet's stylus device and set its Mode to Screen. Click Save and then Close.

Now grab your stylus, create a new file and click on the Draw Tool button in the toolbox. Set the default line width value to be big enough — say, 15pt — otherwise you will not notice any pressure sensitivity effect. Choose brown as the default fill color ([`New_Layer_Defaults#Brush_Colors`]).

Note: Steps above should be done with the stylus of your tablet, not the mouse. Synfig Studio remembers settings for each input device independently. That's why if you set those options with your mouse device they will not have any effect when you switch to stylus.

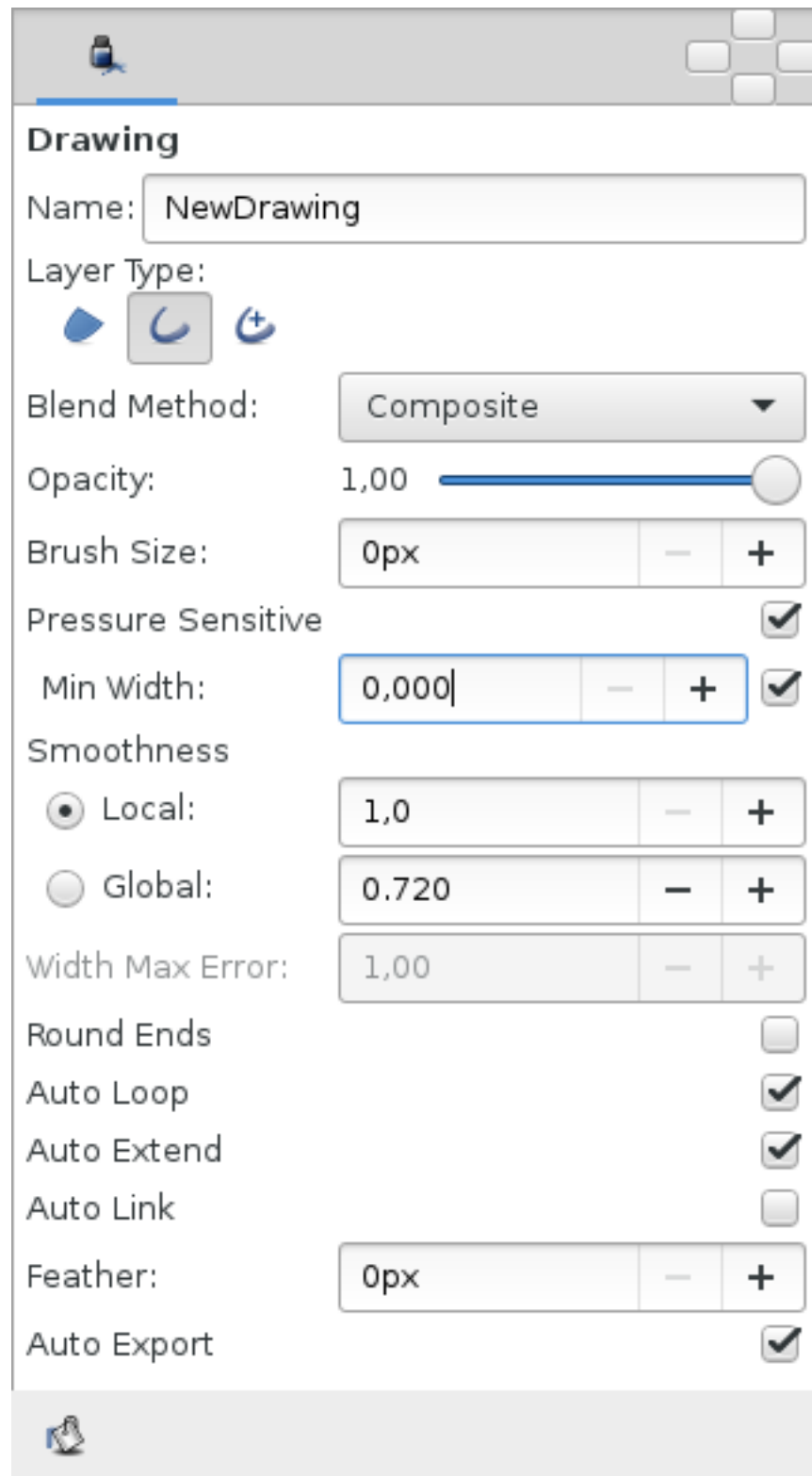
In the Tool Options Panel, make sure that you have the same options as shown on the screenshot below.

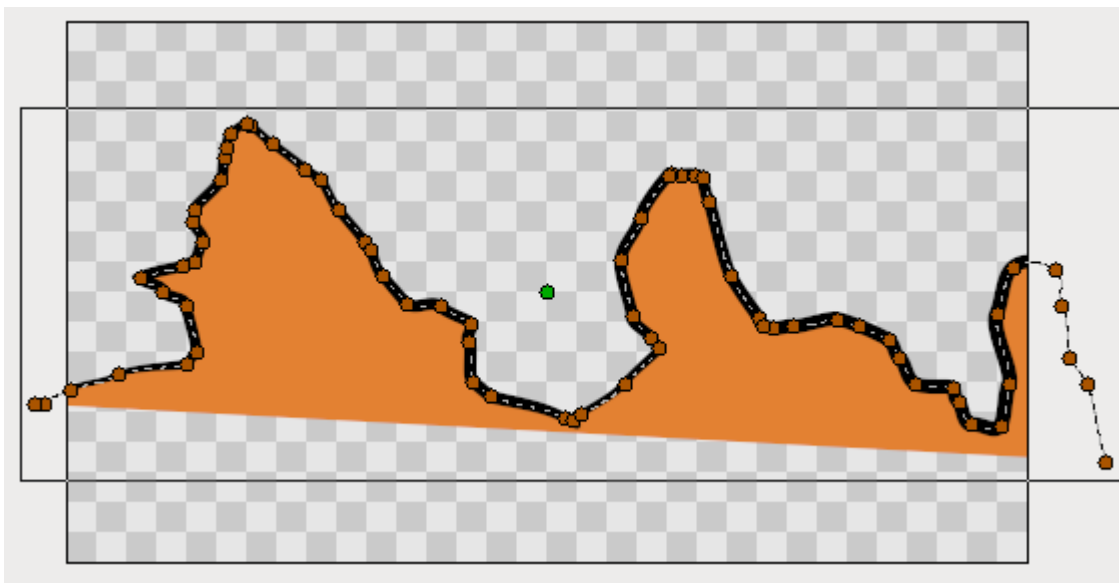
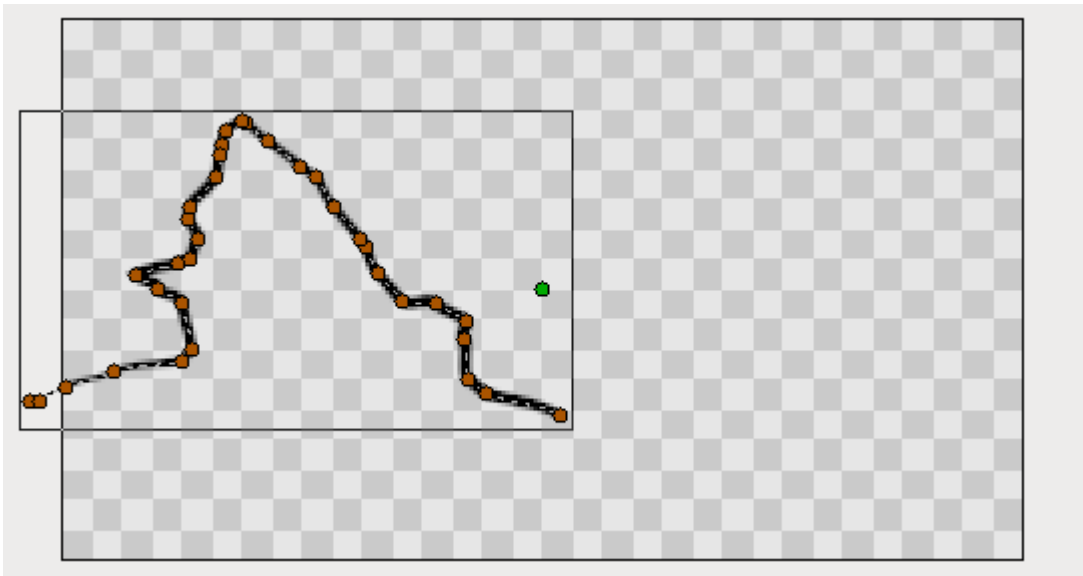
Now let's draw some thing like a curvy mountain background. Start drawing a line from the left border to the middle of the canvas. Try to vary your stylus pressure while you are drawing. Stop near the center of the canvas. This is your first line. Notice the new outline layer created in the Layers Panel.

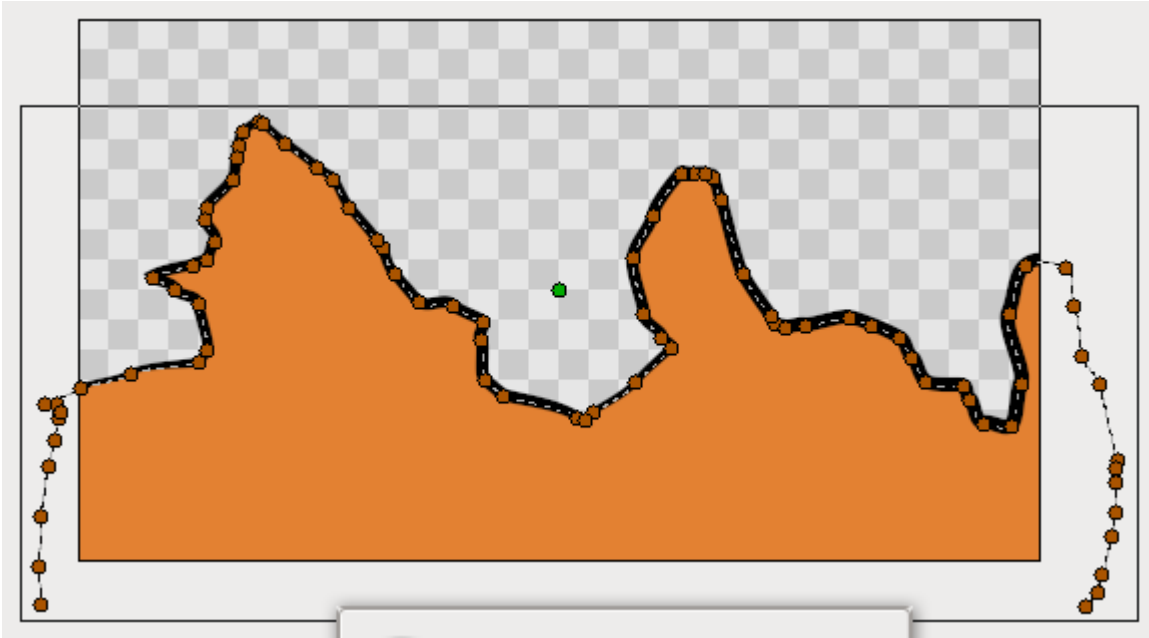
Point your stylus at the last handle of your new Spline and continue drawing to the right border of the canvas. When you finish, look at the Layers Panel again. There's still only one outline layer. Synfig Studio is smart enough to figure out that you don't need a new outline layer and properly extends the last one. You can extend the Spline from both ends, but if you start drawing from any other place of the canvas a new outline layer will be created. Though your first line will remain selected and nothing stops you from extending it later.

Back to our artwork. In the Tool Panel hit the button with the bucket icon to fill the outline we just created. A region layer will appear at the top of the layer we are working with. Select the outline layer and press the `Raise Layer` button in the layers panel to put the outline layer on top of the region.

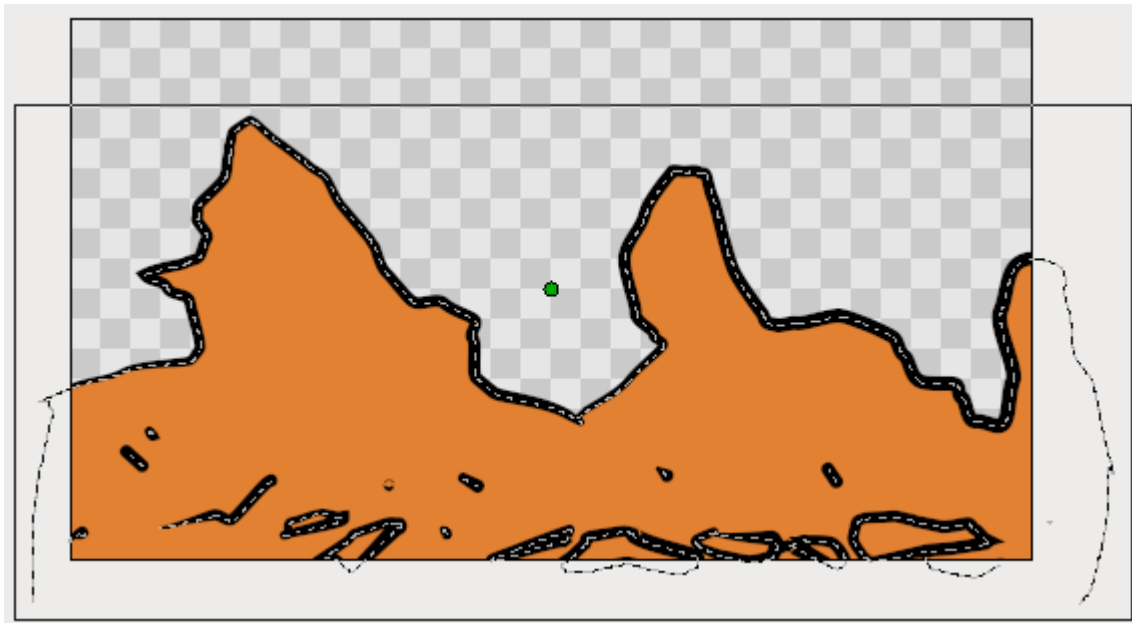
Extend a line from both sides down to the corners of the canvas to make the fill appear at the bottom. Great.







Go ahead and add a few more lines on top of the filled area to give it a mountain-like look. If brown handles are in your way, you can hide them by clicking the **Toggle vertex handles** button at the top of the canvas window (the second one from the left).



Draw tool is great for drawing complex shapes, but you end up with a bunch of handles, which are hard to manipulate with the Transform Tool in the way we described above. There are two solutions here.

First, you can increase the **Smooth** value in the Tool Options Dialog while using Draw Tool. That will reduce the count of vertices produced at drawing time, but will make your shape less detailed.

Second, you can use Smooth Move Tool to deform an existing shape. Go for it and click the Smooth Move Tool button in the toolbox. The trick about this tool is that it affects *selected handles only*. Press and hold your left mouse button in an empty place of the canvas. Drag to create a selection box. Release the mouse button when you are done. Or just

hit **Ctrl|a** to select all handles. Now you can deform the selected segments of Spline. You can change the size of the influence area by tweaking **Radius** in the Tool Options Panel.

What about outline width? There is a **Width Tool** for that purpose. It is designed for increasing or decreasing the width of a line much like you would with a pencil on paper. Click the **Width Tool** button in the toolbox, move your stylus over the line you want to change, press and move the cursor back and forth along the line, like you are scratching something. The width of the outline will be increased at the places where you moved the cursor. If you want to decrease the width, just hold “**Ctrl**” while scratching. Easy!

If you don’t want width handles to be displayed, during usage of the **Width Tool**, just turn them off by pressing the **Toggle width handles** button at the top of the canvas window.

2.5.5 Other ways to create Splines

Is that all? Not yet. You can use **Circle**, **Rectangle**, **Star** and **Polygon** tools to create Splines too. Just check the **Create Outline** and **Create Region** options in the Tool Options Panel when using those tools.

Creating geometric primitives as Splines gives you better control over their look and shape. For example, if you want a deformed star, then you can use the **Star Tool** to create it as outline and region Splines and then use the **Transform Tool** to deform it.

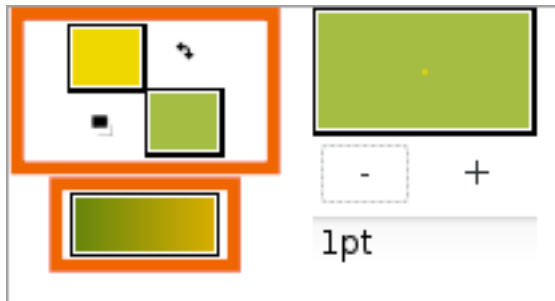
2.5.6 Links

A video on this subject is available [here](#)

2.6 Animating Shapes

2.6.1 Basic settings

In the [previous tutorial](#), you have learned to create shapes with the *Spline Tool*, in this tutorial we will learn how to create a simple animation of a growing flower using *Splines*.



Start Synfig Studio — a new animation will be created. If you already have Synfig Studio started, select **File|New** in the toolbox.

First, we need to create a gradient for a background. Click on the outline and fill colors in the *Toolbox* to select the colors our gradient will have. You can also directly edit the gradient by clicking the gradient line in the toolbox.

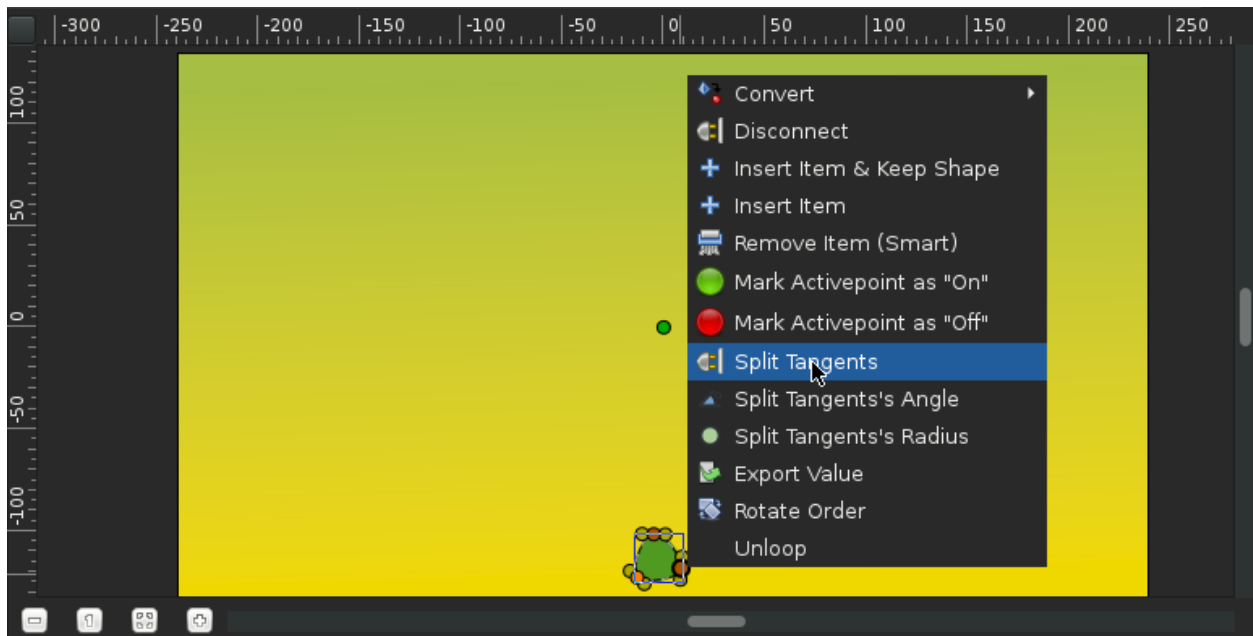
Select the *Gradient Tool* and drag your cursor vertically across the canvas to fill it with the gradient.

Next, select the *Spline Tool* and in the Tool Options Panel, make sure that only **Create Region Spline** is checked. In the toolbox, set the fill color to green. Draw a kind of triangle with the Spline tool. To close the shape after drawing the 3 vertices, right click on the first vertex and choose **Loop Spline**.

Now that the trace of the form is closed, you can generate the proper form by selecting another tool or by pressing the button with the gear icon at the bottom of the *Tool Options Panel*.



This will be the base of the stem. You can tweak the tangent handles (red dots) a bit to make a rounder triangle. With the *Transform Tool*, right-click on each vertex and select *Split Tangents*, so the tangent handles of each vertex can be moved separately.



Note: Origin parameter

Be certain that the shape's 'Origin' parameter is set to 0,0. This will save you from headache in further steps.

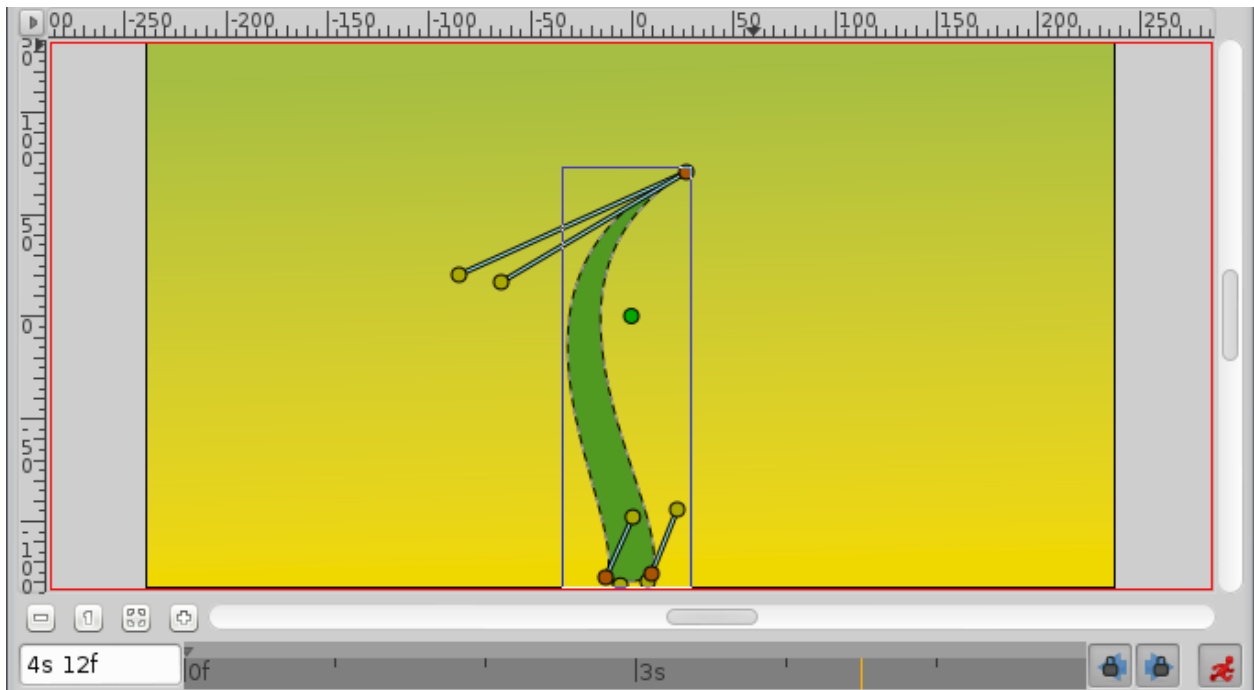
We're done with the basic settings.

2.6.2 Animate the stem

In the Canvas Menu, select the *Canvas Menu Caret* icon in the upper left hand corner, where the rulers intersect, and then select **Edit | Properties**. Go to the Time tab, set the End time to 6s and click OK button.

Click at the beginning of the timetrack (0f), then, in the *Keyframes Panel* (the one with a key icon) click the button with a "+" icon (add a new keyframe). *Keyframes* allow us to *settle down* the scene; i.e. on a keyframe, every element of the scene will have all its properties remembered. Click again on the timetrack, at 4s 12f (ie 4.5s at 24 fps). Press the green man at the bottom right of the canvas (or whatever icon you have there, depending on your icon theme) to switch to the *Animate Editing Mode* (the man is now red).

With the *Transform Tool*, select the green sprout, and move the upper vertex up to make a stem. You can play with the vertex handles to bend the shape a bit if you want.



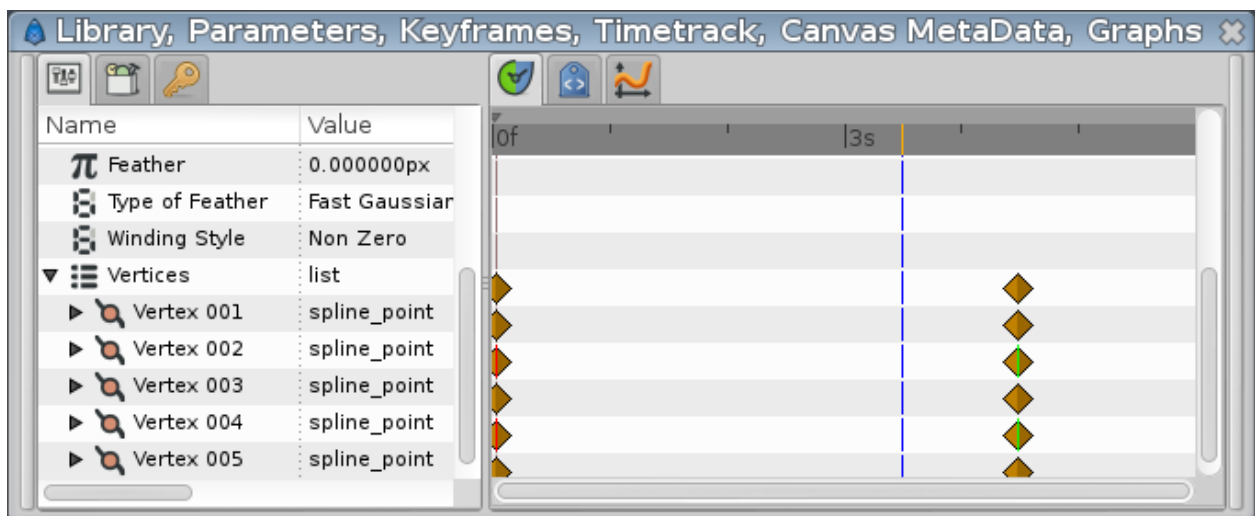
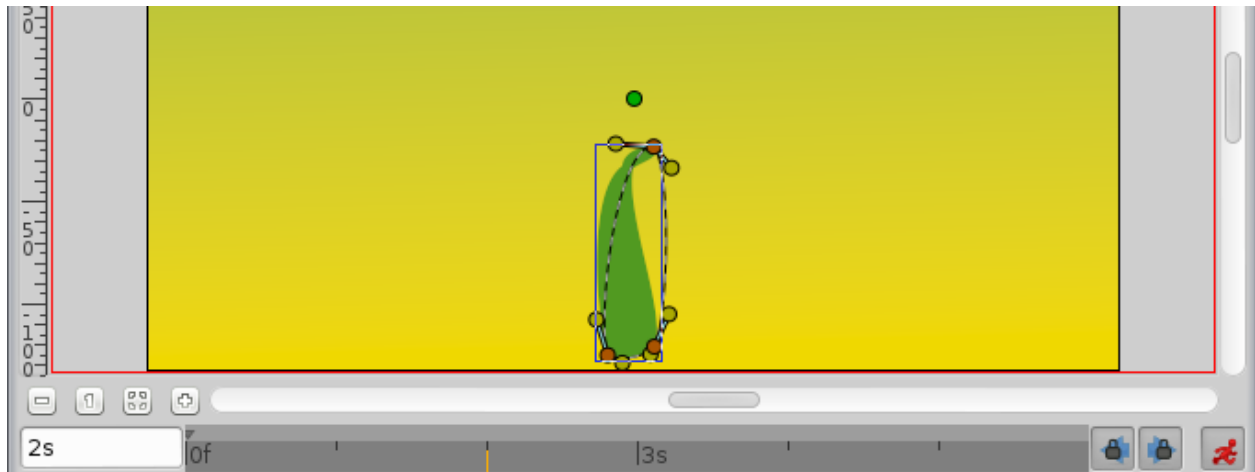
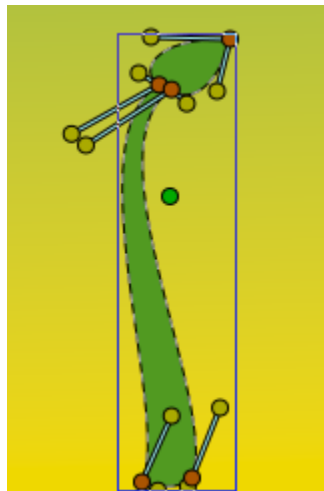
While you are still at 4s 12f, right-click on the stem border, close to the top, and choose **Insert Item (smart)**. Do the same on the other side of the stem. Right click on those new points and choose **Split Tangents**. Then try to make a shape that looks like the one on the image, to create the flower bud.

Now if you click on 2s (for example), you'll see that the shape of the bud is slightly visible, even if the sprout is rather small, and even if the bud handles are invisible.

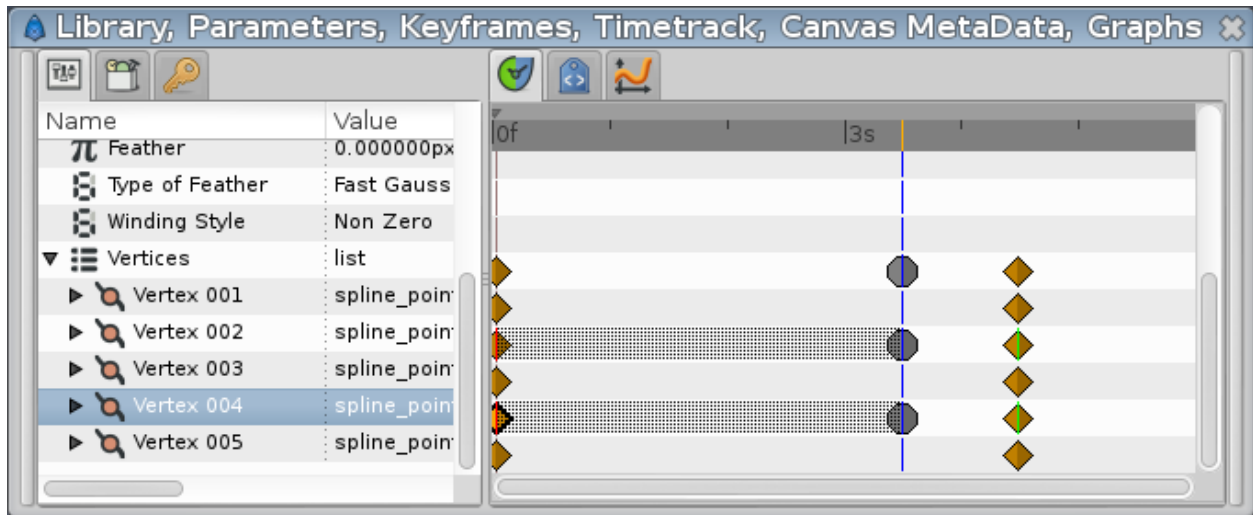
Let's say we want the bud to appear only at 3s 12f, and be full size at 4s 12f.

Click on 3s 12f on the timetrack. Now take a look at the *Parameters* and *Timetrack* panels at the bottom. You'll see that each parameter in the *Parameters Panel* matches a row in the *Timetrack Panel*. The last parameter is the vertices list. Click on the small arrow on the left to unfold the list. You should see something like this:

Each brown diamond (or *Waypoints*) stands for a recorded value (here the vertices positions were recorded at 0f with the keyframe, and at 4s when we moved some vertices or vertices handles). The two vertices we added to make the bud are marked with green and red vertical line on their 0s and 4s waypoints. Right-click on them in the parameters list, and select **Mark Activepoint as Off**.



The panel should now look like this, the grayed part being the part where the bud vertices have no effect on the stem.



For example if you click on 2s or even 3s now, the bud shape is not visible. It starts to appear only a little after 3s 12f.

However, the shape of the stem may not look very nice during its growth between 0 and 4s. Make sure you're still in Animate Edit Mode, and tweak the shape at various moments in time, to get something you like.

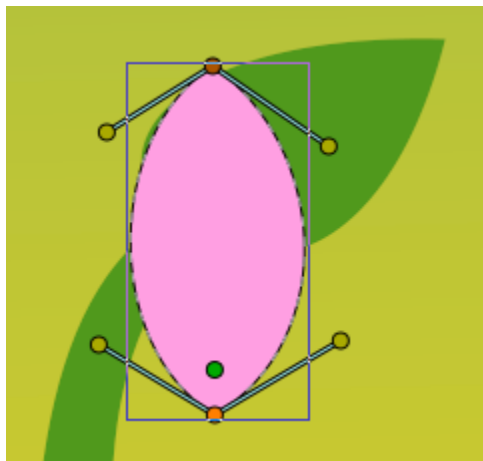
The animation of the stem is now finished, but it still lacks the petals. You can watch a preview of your animation: Go to File|Preview, validate, wait for the preview to be generated, and watch.

Note: Previews are often pixelated and blurry, but the final render will be clean-cut. Higher quality previews are obtainable by using higher values for 'Zoom' and 'Frames per second' in the preview dialog window.

2.6.3 Adding the petals

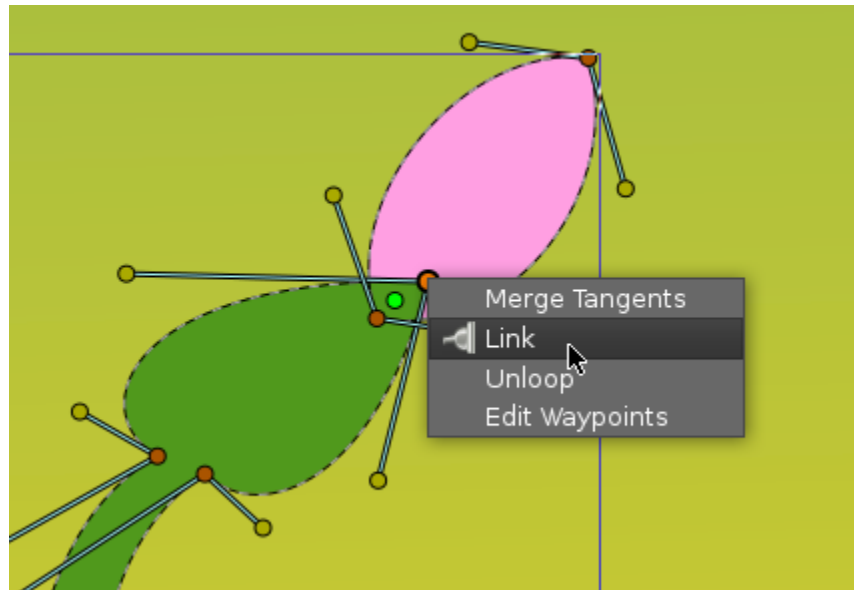
Now leave the **Animate Editing Mode** by clicking on the red circle at the right bottom of the canvas.

Change the fill color to pink, and create a petal with the Spline Tool. You'll notice that the green **handle** that allows easy movement of a shape is at the center (*Origin Parameter* : 0,0) of the canvas. Select all the vertices of the petal with **Ctrl|a** and move them close to the green handle (with the Transform Tool), as shown.



Then drag the green handle very close to the top of the bud. Hit **Ctrl|a** again to select all vertices of the petal and tweak it a bit with Rotate Tool. Also, in the *Layers Panel* select the petal layer and put it **under** the stem layer. Click on the petal to select it, then ctrl-click on the stem. Both objects should be selected.

Now click on the vertex at the top of the stem and ctrl-click on the green handle of the petal (both should appear in a lighter color, as they are selected). Then right-click on the stem top vertex, and select **Link**. The petal will move a bit as the green handle is snapped on the stem vertex.



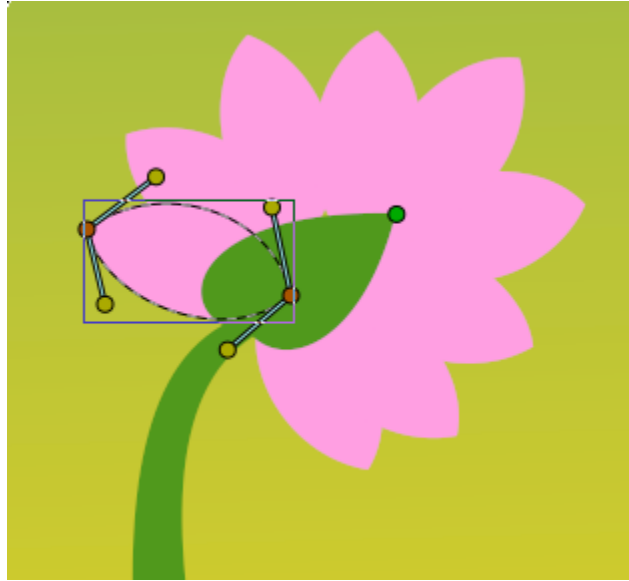
Now that there's a link between the petal and the top of the stem, when the top of the stem moves, the petal will follow the move. (And if the green handle of the petal moves, the top of the stem will move, but we don't want to do that here.)

Note: I'm going mad, can't link the stem and the petal!

The problem is trivial. The 'origin' parameter of 'Stem' and 'Petal' layers is different. They should be same, because you're linking origin of layer with some vertex that has its own layer's origin. Set both origins to the same values (0,0), then link.

On the Layers Panel, select the newly created Petal layer and duplicate it (with the third button, on the bottom of the Layers Panel). On the canvas, press **Ctrl|A** to select all the vertices of the duplicated petal, and move them a little, so the petals are no longer overlaid. (Don't move the green handle, just the orange ones). Repeat the process several time, to get something looking like this image.

Note that the duplicated petals are also linked to the stem. If you go back to the first keyframe, you'll see that the petals are visible. We don't want that. We want the petals to appear and bloom almost at the end of the growth.

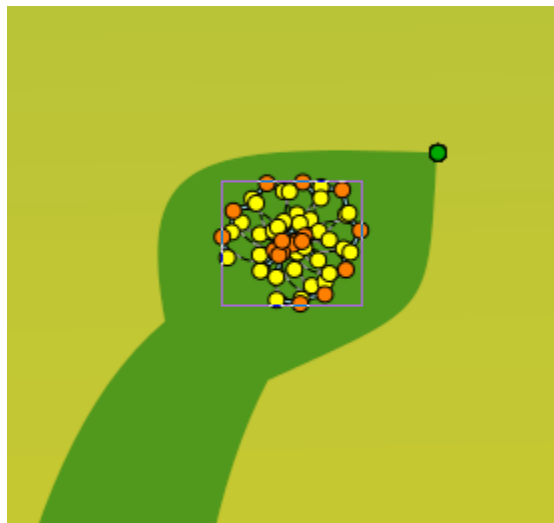


2.6.4 Hiding the petals

Let's say we want the petals to appear a little after 4 seconds in the animation, and be full size at 5 seconds, instead of being visible and full size all the time.

Switch to **Animate Editing Mode** again by clicking on the green circle at the bottom right of the canvas. But if we will go to 4s and modify them, then they also change at 5s. Because the shape/position of the petals is not fixated at this moment of time by any waypoints or keyframes. That means that we need a keyframe at 5s. On the *timetrack*, click to place the cursor at 5 seconds. On the *Keyframes Panel*, click on "+" to add a new keyframe.

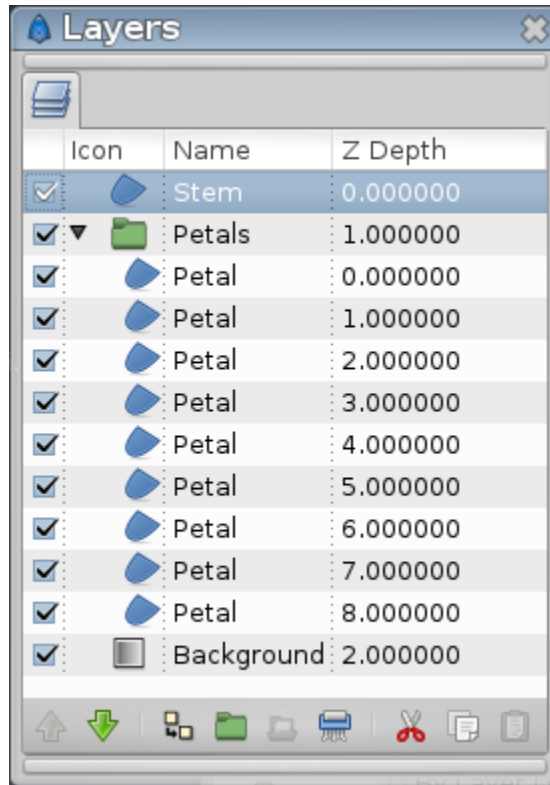
Now click on 4s, and on the *Layers Panel*, select all the petals layers (with ctrl+click), then press **Ctrl | A** to select all the petals vertices. Scale them down with the *Scale Tool*, and move them, so they are hidden by the stem, as shown.



From 4s to 5s, the petals will now appear and bloom. But notice that we have a keyframe at 0s which also remembers petals shape. That makes the problem — the petals are still visible from the first keyframe to the 4s keyframe. We could either make the petals tiny and hidden tweaking their size on every frame from 0s to 4s, or we could make them invisible on this interval.

Let's choose the second solution. To make things easier, we are going to group the petal layers into an *Group layer*.

With all the petal layers selected, right-click on them on the Layers Panel and select **Group**. You can rename the layers to make things more understandable.



Select the “Petals” Group Layer and jump to the first keyframe. In the Parameters Panel, set the **Amount** value to **0**. The petals are now invisible on that keyframe. Note that two waypoints were added in front of the **Amount** parameter, one at 0s and the other at 5s. Drag the 5s waypoint to 4s, so that the opacity of the petals will be 1 at 4s.

There is still one problem left: from 0s to 4s, the opacity of the petals slowly increases, making the petals visible when they shouldn’t. To solve this, we will change the **Amount** interpolation method. Right click on the **Amount** waypoint at 0s, and select **Edit**. A new dialog will appear, in which you can choose the **In** and **Out** interpolation. Set the **Out** Interpolation to **Constant**.

Note: Tip

You can also change waypoint **Out** interpolation by right-clicking on it and selecting **Out | Constant**.

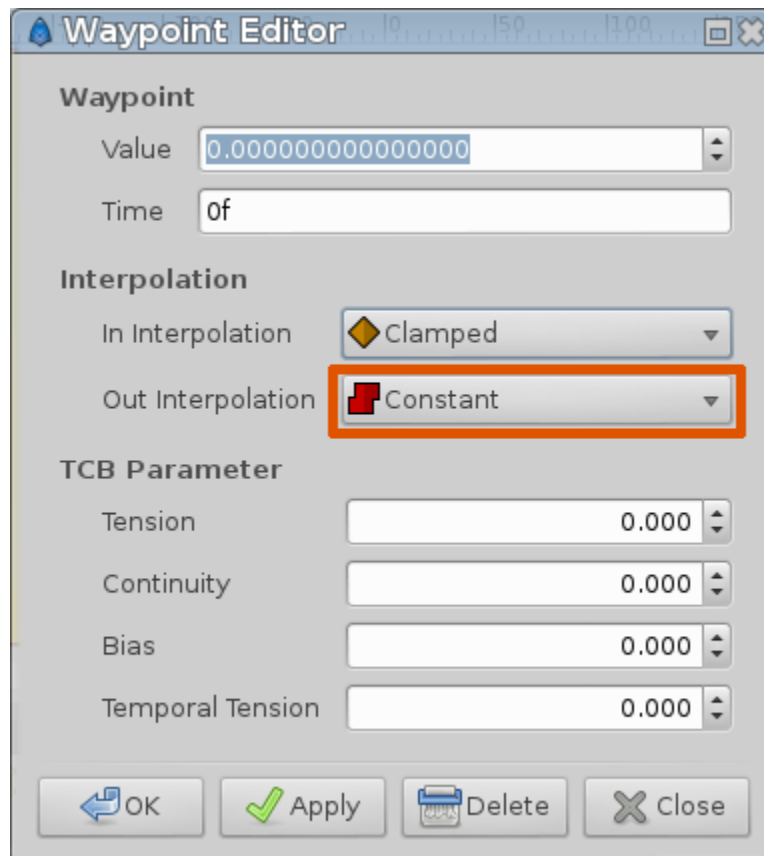
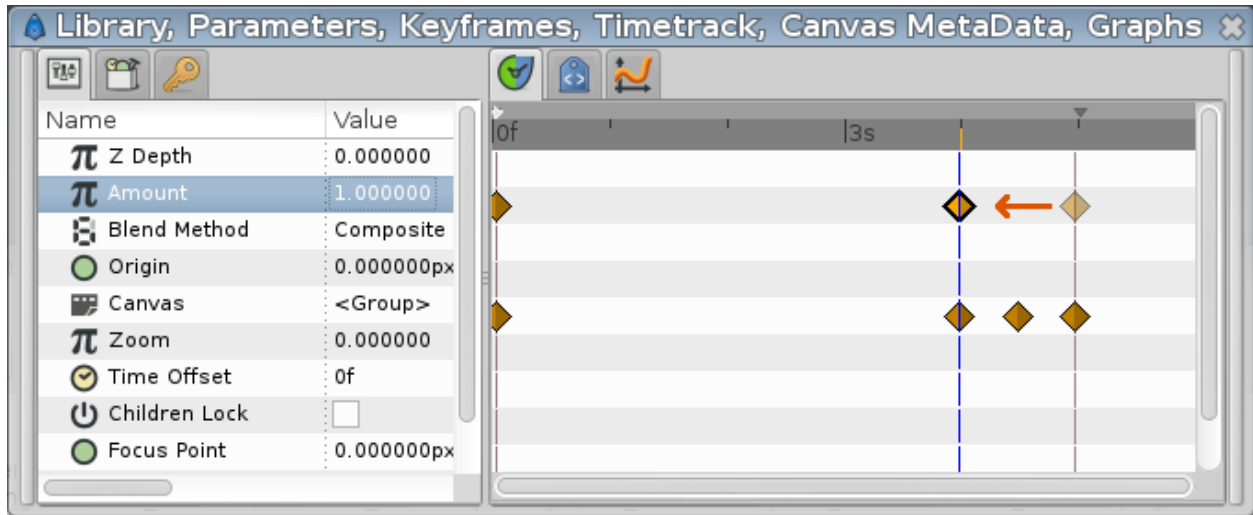
This means that*after* that waypoint, the **Amount** value will remain constant, until another waypoint is encountered. So from 0s to 4s the **Amount** value will be equal to 0, and at 4s it will suddenly changed to 1, and make the petals visible, as expected. Alternatively, we could have achieved the same effect by setting the **In** Interpolation of the waypoint at 4s to **Constant**.

Notice how (half of) the waypoint changes from a green circle (meaning smooth animation of the amount parameter) to a red step (meaning that the amount parameter is suddenly stepped).

Now you’re done. The stem grows for 4.5 seconds and then stays still the last 1.5 seconds. The petals are hidden until 4 seconds, and then grow quickly between 4 and 5 seconds, and stay still the last 1 second too.

Click on **File|Render** to render your animation. Select any format you want, and ensure that **Use current frame** option is unchecked (otherwise, one frame only will be rendered).

You can download the project used for the screenshots.



2.6.5 What's next

Synfig Studio have a *Skeleton Layer*, in the *next step* you will learn how to setup a basic skeleton.

2.7 Basic Bone Tutorial

2.7.1 Introduction

In this tutorial we are going to rig an arm with fixed joints using the *Skeleton Layer* and three bones. The arm consists of three parts: the **upperArm**, the **lowerArm** and the **hand**. (pic.1) Each part is grouped in its own layer and the three groups are again grouped in a layer that is called **arm** (pic.2).

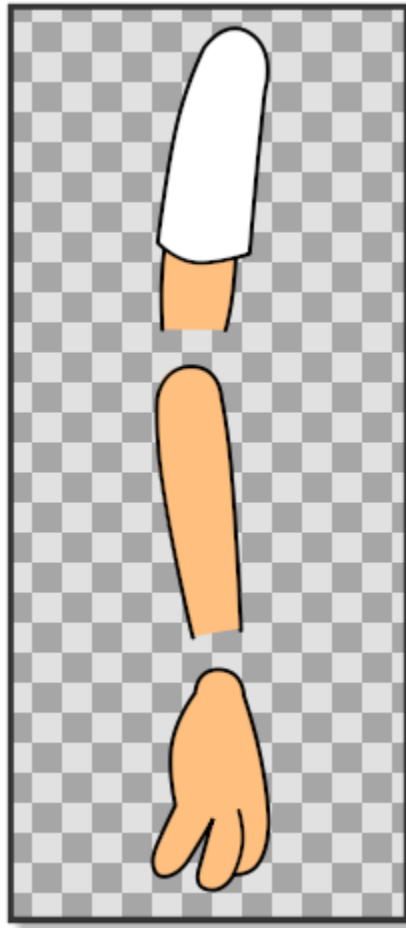


Fig. 11: PICTURE 1: Three parts of the arm

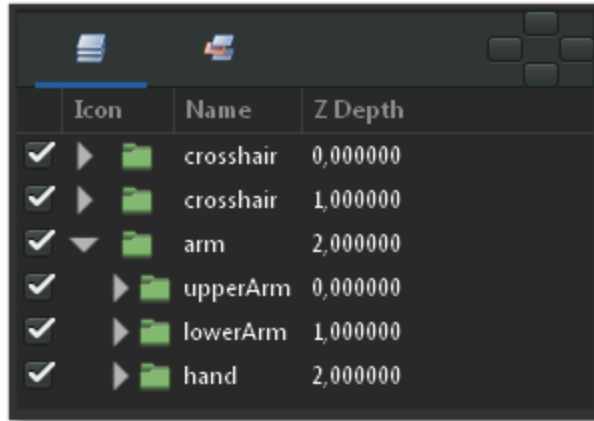


Fig. 12: PICTURE 2: The layers

2.7.2 Prepare the joints

Before we start rigging the arm we have to prepare it to make sure that the joints properly rotate.

In *picture 1* we can see that the lower side of the upper arm is straight and does not have an outline, while the upper part of the lower arm has an outline that is curved. The bottom part of the lower arm does not have an outline and is also straight. The joint of the hand is also curved with an outline.

To make sure that the joint parts of the arm rotate correctly, we made them with the help of a couple of crosshairs *crosshair* (picture 3). In *picture 4* and *picture 5* we can see how the horizontal crosshair line matches the straight line of the bottom of the upper arm and at the same time the outline of the circle of the crosshair matches the top curved outline of the lower arm.

The same principle is applied to the hand and the bottom part of the lower arm. The crosshairs also mark the centre of the rotation point that will later be used when placing the bones. Before placing the bones we can align the *Group Transformation Widget* (press control and move the left corner handle of the *Group Transformation Widget* to move it.) of the **lowerArm** group and the **hand** group with the crosshairs of the elbow and the wrist to check if the joints rotate correctly.

Don't forget to bring the arm into its original position by pressing undo. (Don't worry if the hand does not rotate with the elbow when you check the elbow joint by using the *Group Transformation Widget*.)

2.7.3 Adding the bones

Now it's time to add a *Skeleton Layer* that will provide the bones we need to rig the arm.

Right-click on any layer in the *Layers Panel* and choose New Layer -> Other -> Skeleton. A new *Skeleton Layer* (pic. 7) will appear in the *Layers Panel* alongside with a small bone in the canvas (pic.6). Put the *Skeleton Layer* above the **arm** group if it is not already there. The green handle of the bone is the origin of the bone and is used to move the bone in place and is also its centre of rotation. The blue handle is used to rotate the bone. The orange handle determines the length of the bone is used to stretch the bone and its contents. The first bone is the parent bone and should be moved toward the shoulder and stretched so that it almost reaches the outlines of the first crosshair of the elbow. Right-click on any of the handles of the parent bone and select Create Child Bone. Move

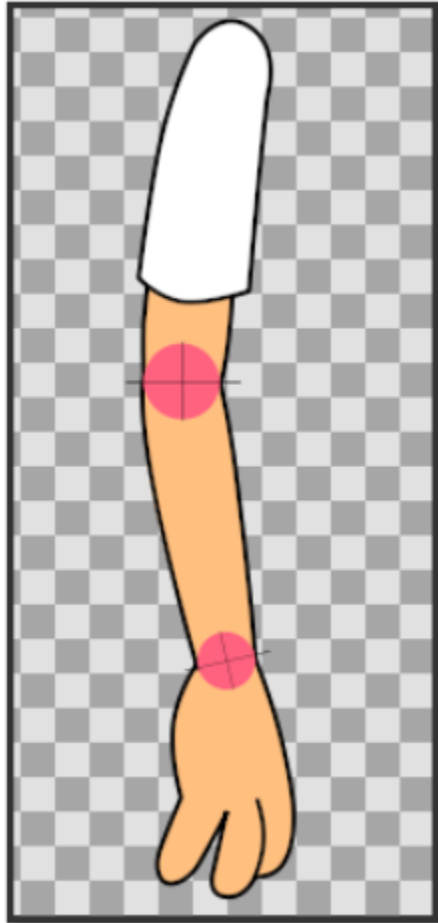


Fig. 13: PICTURE 3: Joints

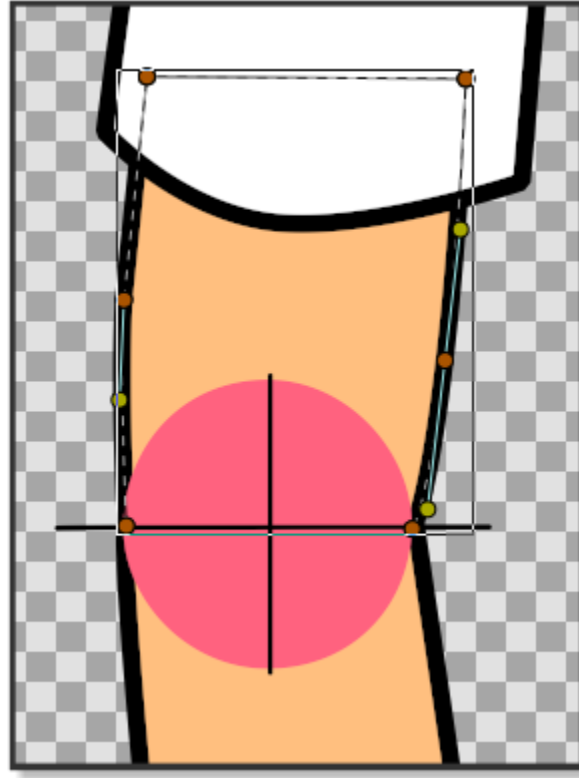


Fig. 14: PICTURE 4: Joints align

the green origin point of the child bone so that it matches the crosshair of the elbow and stretch it until it almost reaches the crosshair of the wrist. Right click on any of the handles of the child bone and create another child bone. Move the green origin point of the second child bone so that it matches the crosshair of the wrist (pic.8). Stretch the bone until it reaches the end of the fingers.

2.7.4 Attaching the bones

Now that the *Skeleton Layer* is in place we have to attach the bones to the parts of the arm.

In the *Layers Panel* click on the **upperArm** group, right-click and Select All Child Layers. Then press `ctrl|A` in the canvas window. With everything in the **upperArm** group highlighted, `ctrl`-click on any bone in the canvas window and right-click on any of the handles of the parent bone and click **Link to Bone**. The **upperArm** group is now linked to the parent bone. Go the *Layers Panel* again and select the **lowerArm** group. Right-click and Select All Child Layers. Press `ctrl|A` in the canvas window and `ctrl`-click on one of the bones. Then right-click on any of the handles of the first child bone and select **Link to bone**. The **lowerArm** group is now linked to the second child bone. In the *Layers Panel* select all the child layers of the **hand** group and press `ctrl|A` in the canvas window. `ctrl`-click on any bone and then right-click on any of the handles of the second child bone and select **Link to bone** (pic.8).

The hand is now linked to the second child bone and the entire skeleton is now linked to the arm.

It is worth to note for new users that in case the bone is linked to an image, selecting all its handles and linking them to the bone could lead to an initially unpredictable behavior. If we need for example to apply only a rotation around the pivot point defined by the bone, it is necessary to select only the green handle that defines its position. Then use the

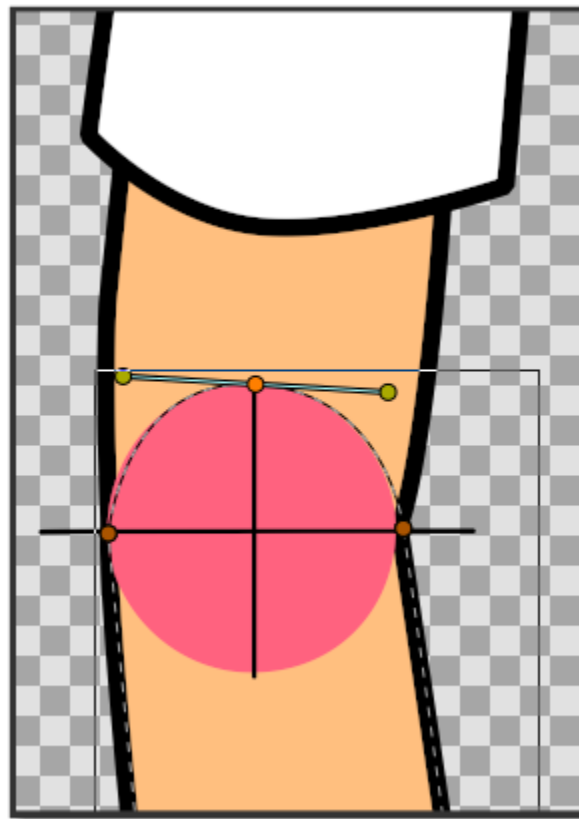


Fig. 15: PICTURE 5: Joints align

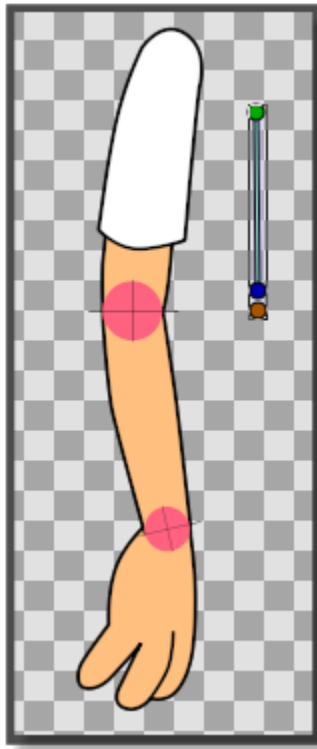


Fig. 16: PICTURE 6: Adding bone

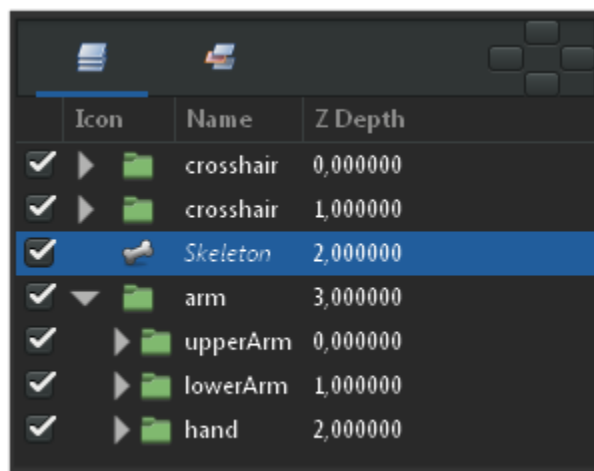


Fig. 17: PICTURE 7: Adding skeleton layer

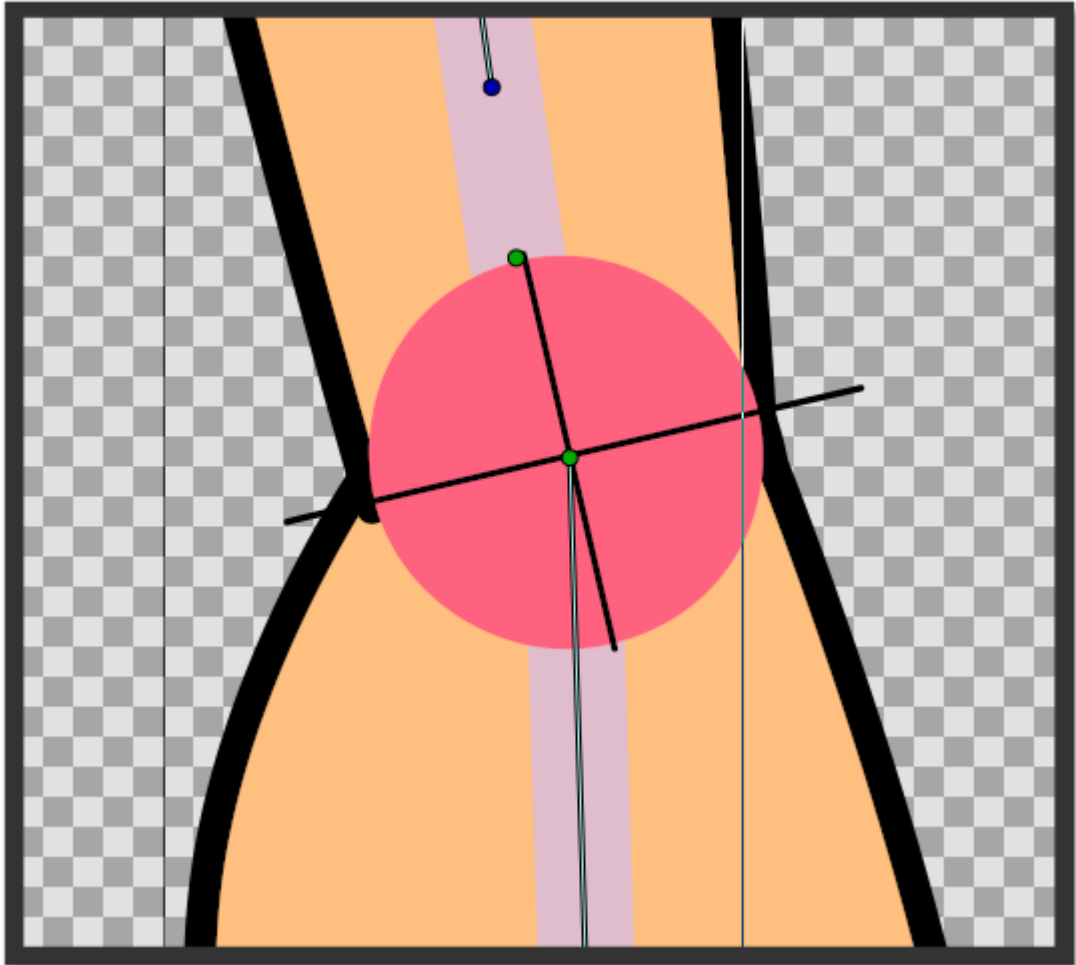


Fig. 18: PICTURE 8: Green origin point hand

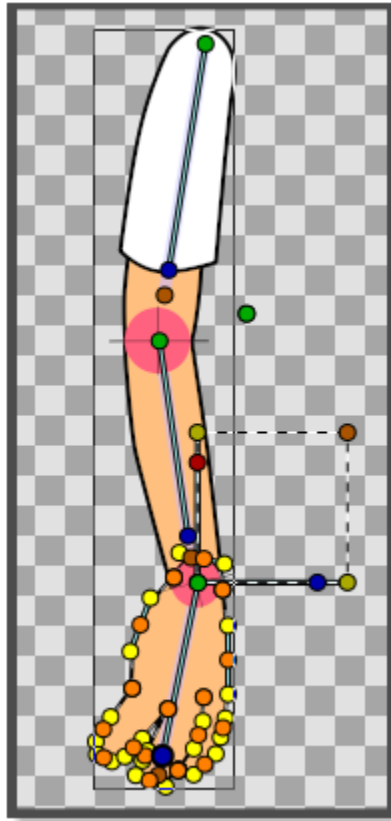


Fig. 19: PICTURE 8: Attaching hand to second child bone

blue handle of the bone to apply the rotation.

2.7.5 Tip for making a crosshair

Making a crosshair in Synfig is easy. Draw a circle. Draw the outlines of a square and align them with the circle. Draw an X using the corners of the square as a reference. Delete the square. Group the circle and the X. Rotate 45 degrees using the *Group Transformation Widget*.

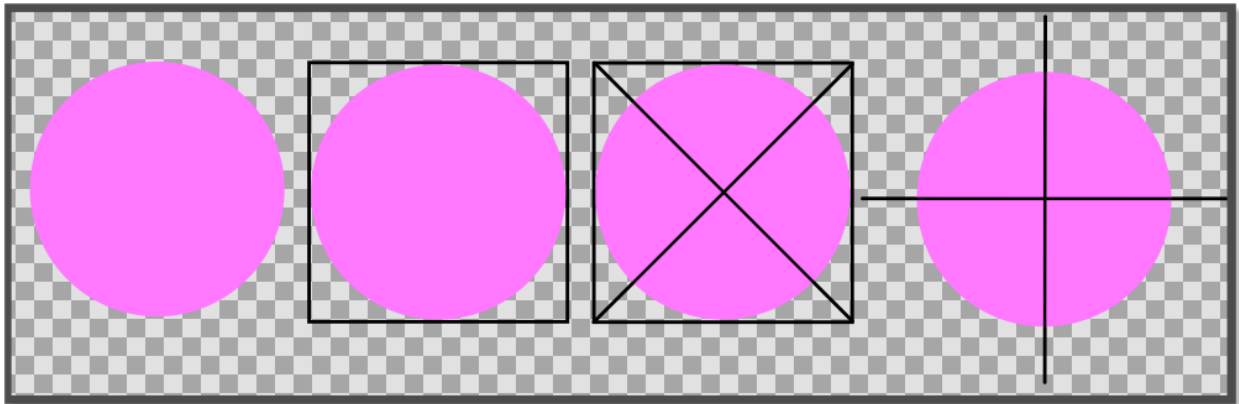


Fig. 20: PICTURE 9: Drawing a crosshair

2.7.6 Project file

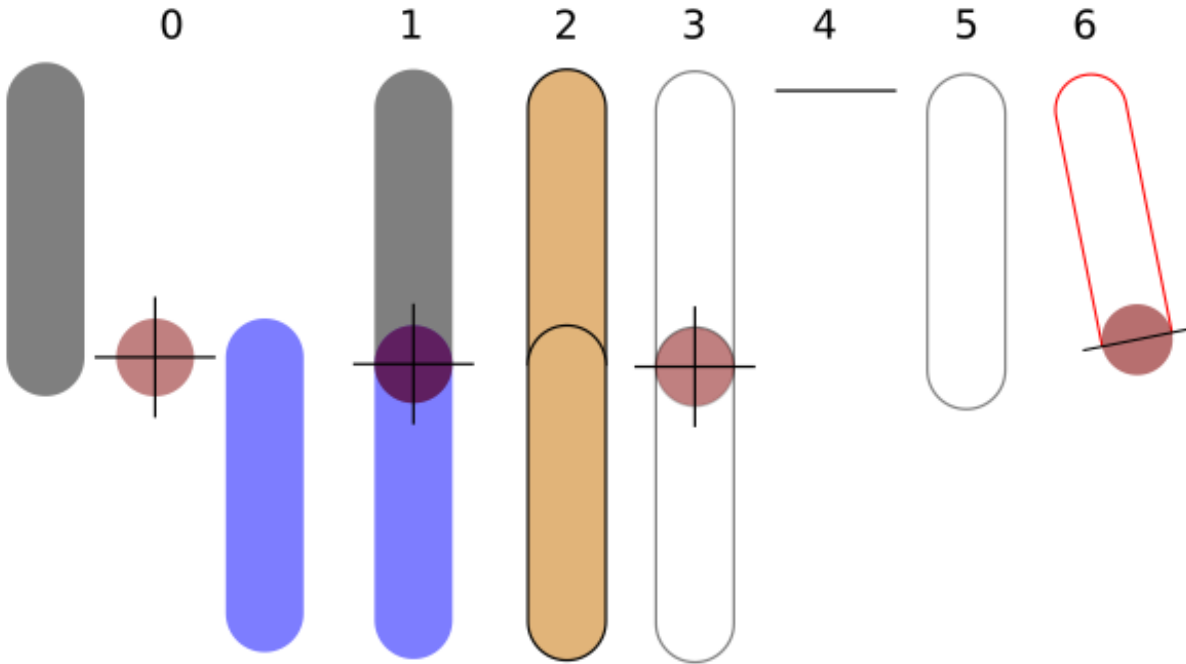
The sif file containing the parts of this tutorial can be found [here](#)

Explanation n°2 additional

2.7.7 To transform 2 sticks into 1 leg

The top stick is a “thigh”, the bottom stick is a calf.

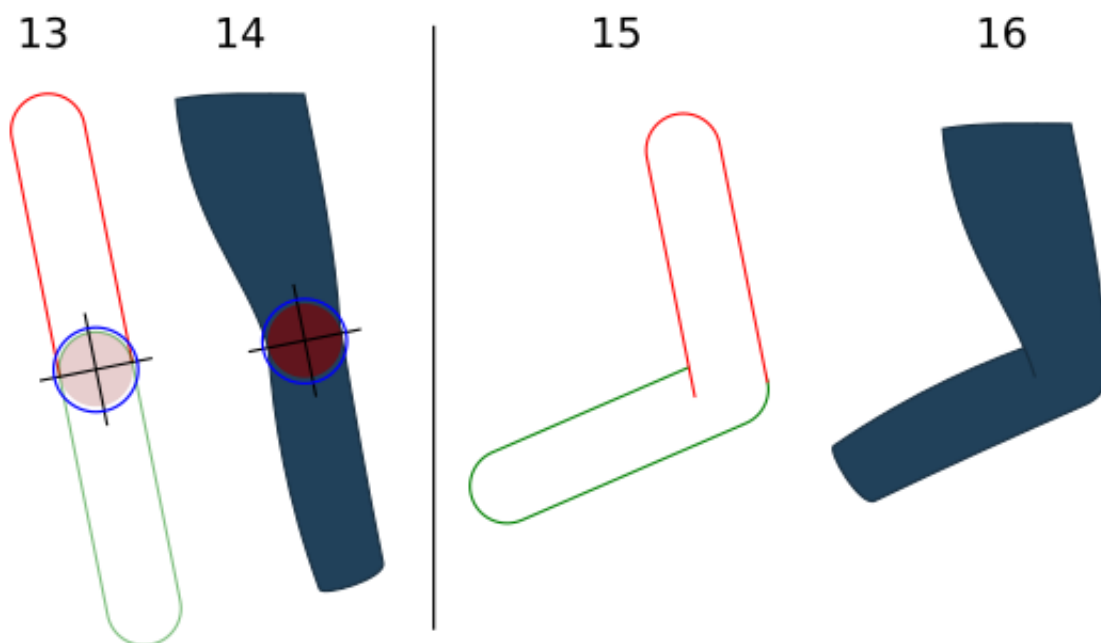
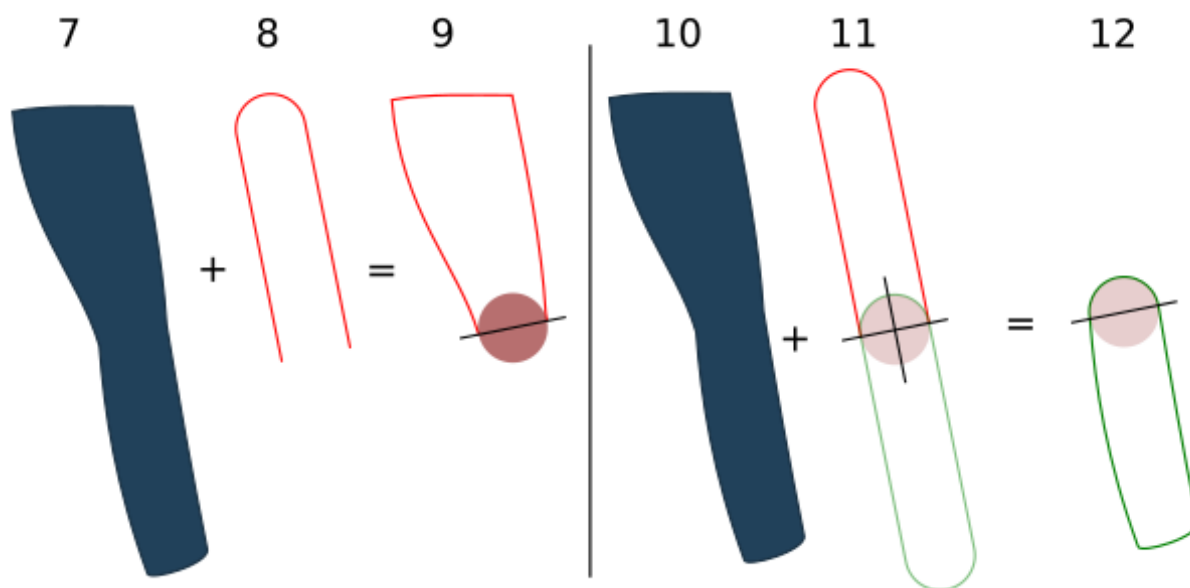
- 0: 2 identical sticks + a crosshair.
- 1: The 2 sticks touch edge to edge to form a crosshair.
- 2: The 2 sticks without crosshair of flesh color.
- 3: The 2 sticks with crosshair without color.
- 4: This mark must be used.
- 5: To cut the upper limb.
- 6: Like here.
- 7: Then the limb must be merged (full).



- 8: With the cut part.
- 9: Result.
- 10: Same for bottom limb.
- 13: The only part not to be merge is the one where the circle (blue) it is the crosshair that should not be touched.
- 15: Then the rotation will be correct.
- 16: To be able to do any rotation.
- 17: Limb of the top.
- 18: Limb of the bottom.
- 21: Limbs in movement (detached).
- 22: Limbs in movement (attached).
- 23: The leg finished with the crosshair.

The crosshair must stay with our 2 sticks, for the duration of the limb's creation.

Now you are ready for the **last tutorial** in this section. Hang on!



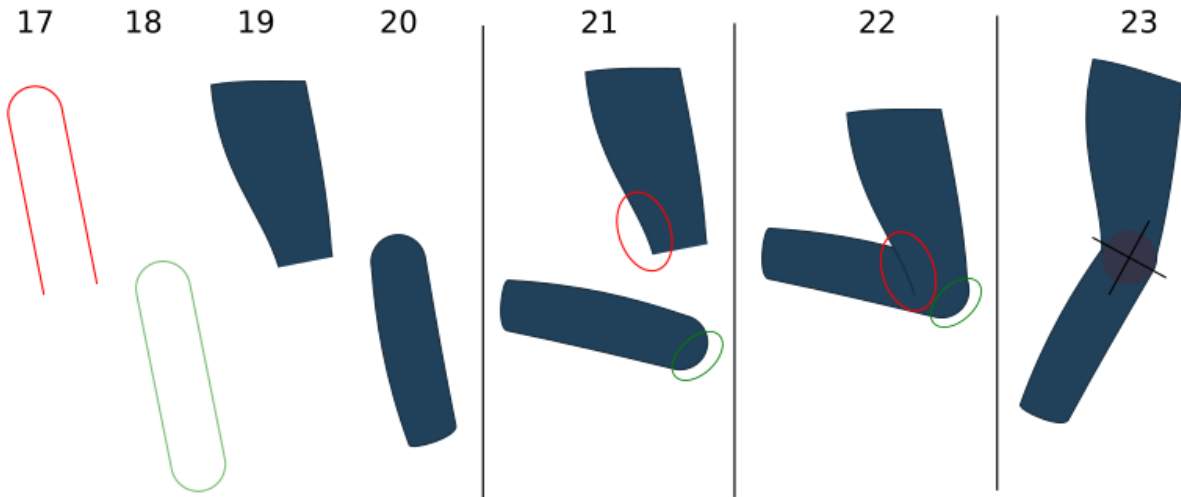


Fig. 21: Animated Leg.

2.7.8 Project File

The zip file containing the parts of this tutorial can be found here: [Crosshair-leg.zip](#)

2.8 Basic Masking

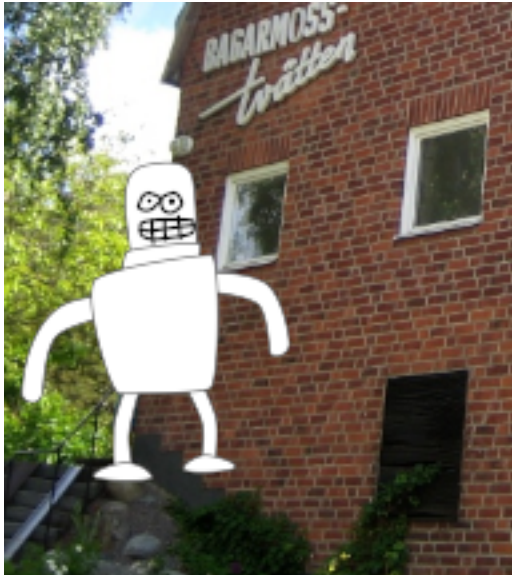
This tutorial gives you an overview of how masking can be done in Synfig.

2.8.1 About masking

Sometimes you want your characters to go behind objects, a building in the background for instance. If you create your background in Synfig you can probably just place the character behind the object in your layer stack but if your background is an image you need to do this by masking.

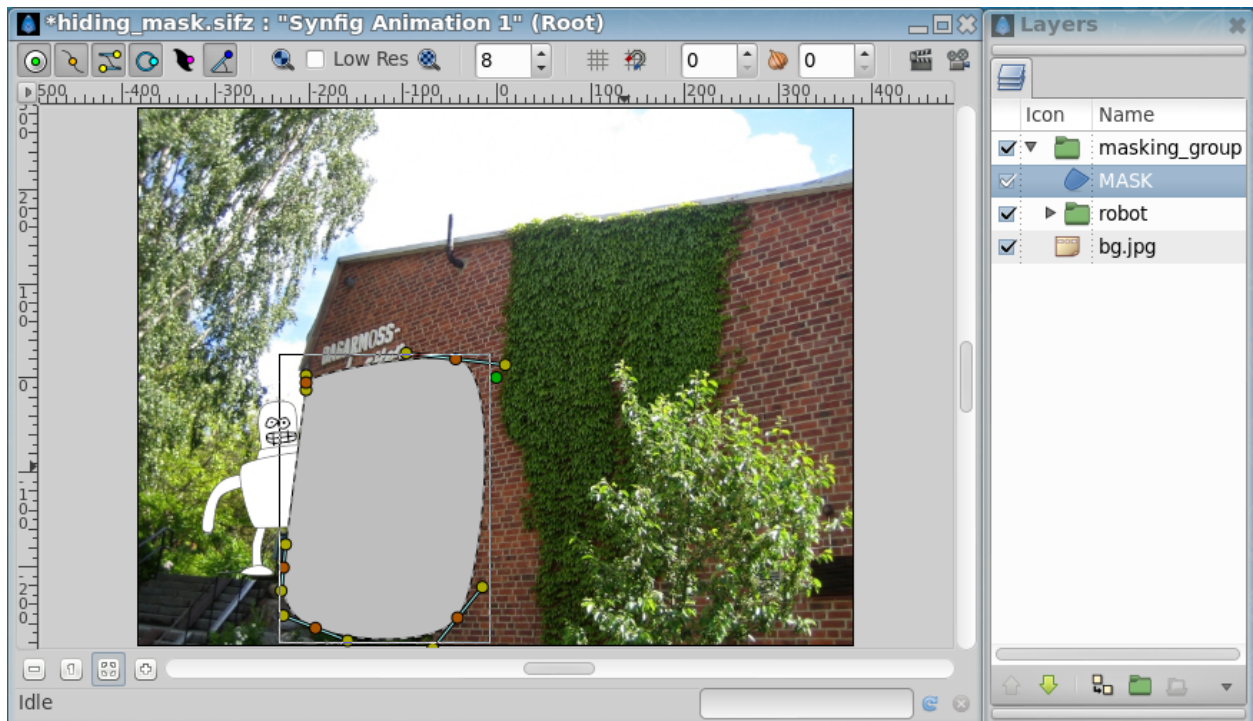
This tutorial will show you two basic ways of masking; hiding and revealing using different *Blend Method Parameter*

2.8.2 Hiding



In this masking mode everything covered by the mask shape is hidden.

- Create a mask shape and place it above the elements you want to mask, either within the same group layer or above it and then grouping the mask and elements together.



- Set the blend method of the mask to “Alpha Over” ([Blend_Method_Parameter#Alpha_over]).
- Everything below the mask shape will now be hidden.



2.8.3 Revealing

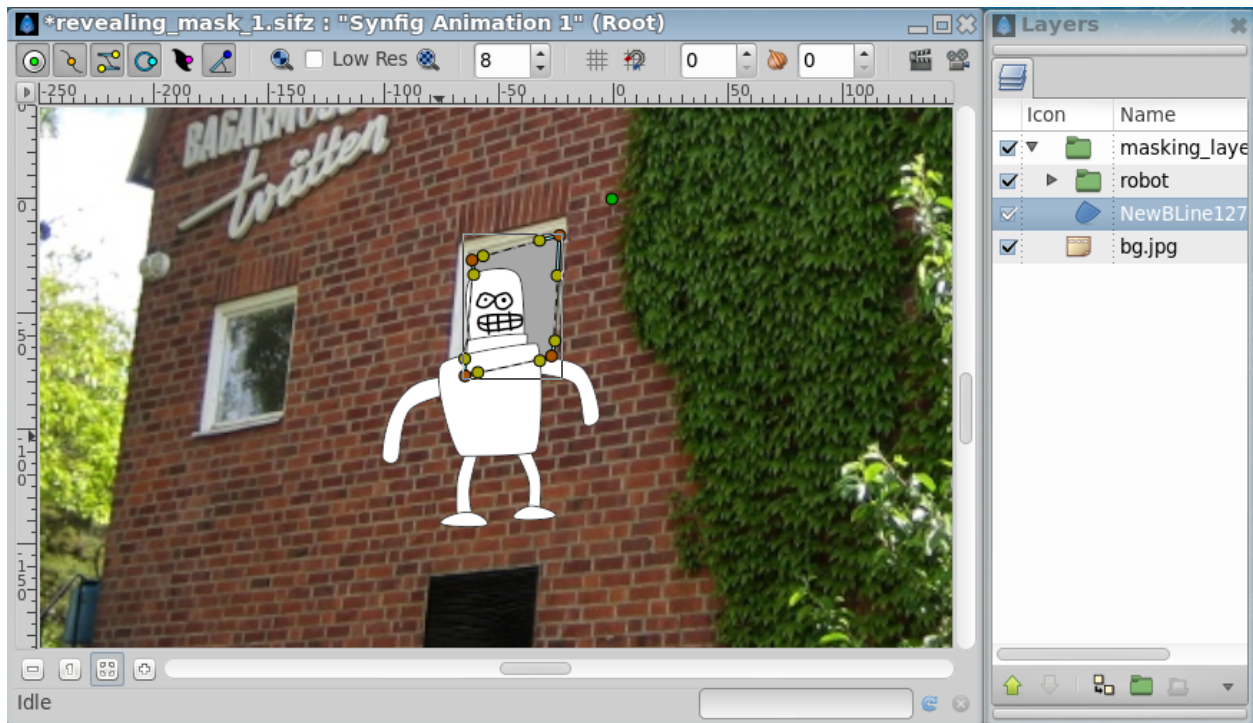
A revealing mask can be achieved by two different approaches using different blend modes.

2.8.4 Revealing mask method 1.

Note: Warning, *actually* broken in some cases

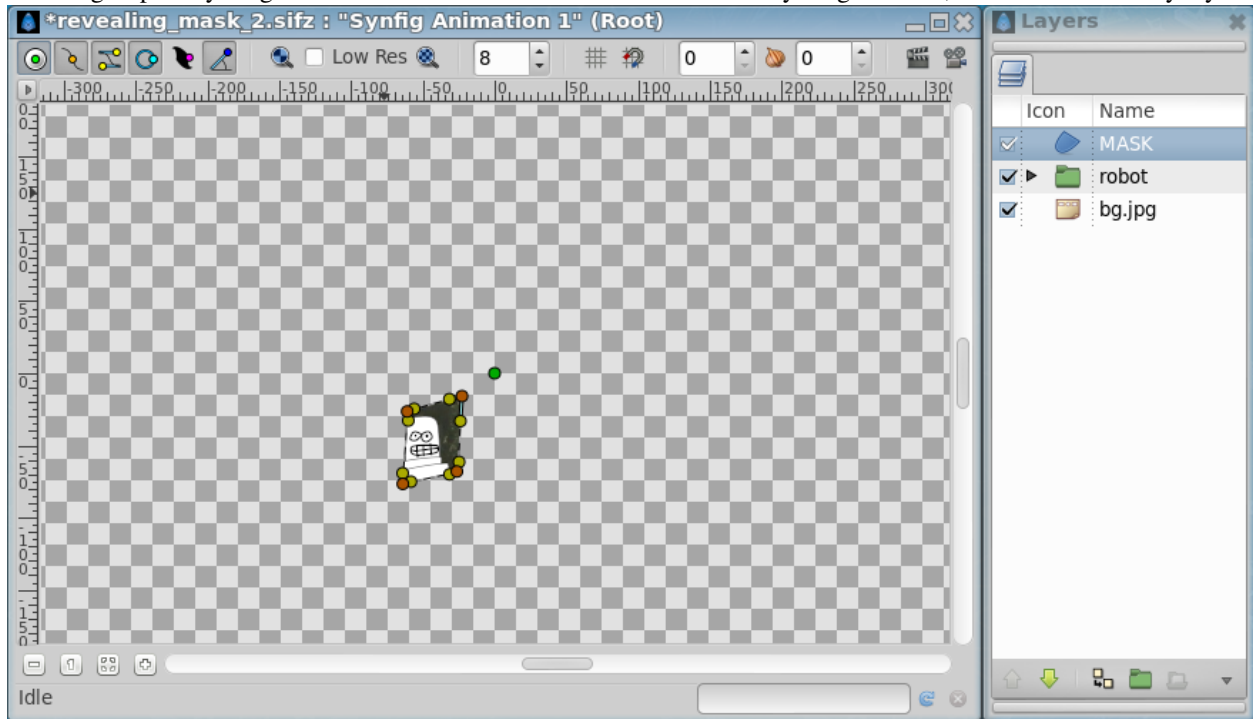
[Bug report #868](#) - Straight Onto blend method broken (both render engine) : When using groups result is the same has “Straight” on the workarea and when rendering and cobra (0991d44751f) to render empty on vector artwork.

- Create a masking shape and place it below the elements you want to mask.
- Set the blend method of the elements you want to mask (robot layer in this case) to “Straight Onto” *Blend Method Parameter: Straight Onto*. You can only use this method on one layer at a time so if you have several objects you need to group those into one group layer.
- Everything above the mask will be visible, everything outside will be hidden.



2.8.5 Revealing mask method 2.

This mask method is similar to the hiding mask and a bit more flexible than revealing method 1 in that you don't have to group everything that should be masked. This mask reveals everything below it, no matter how many layers.



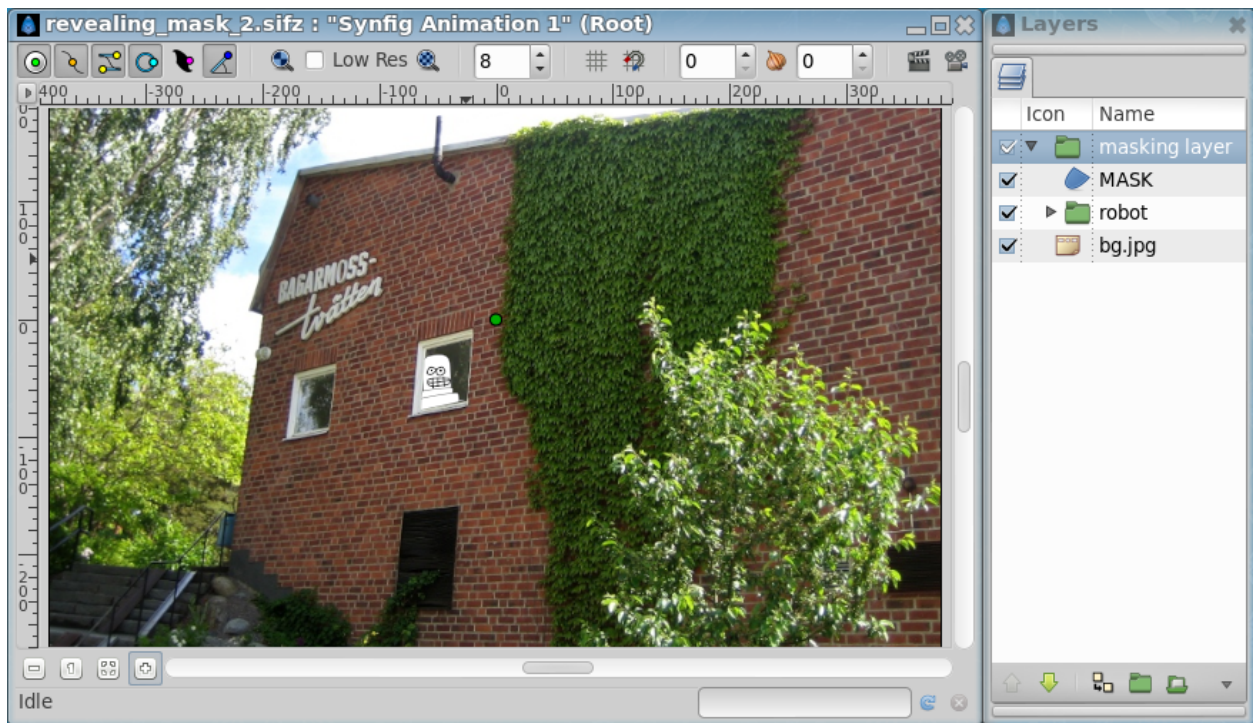
- Create a mask shape above the objects you want to mask.
- Tick the “Invert” option of the mask.
- Set blend method of mask to “Alpha Over” *Blend Method Parameter: Alpha Over*. Everything below the mask shape will be visible and the rest is masked off.
- You can group the mask and the objects that should be masked. Anything outside of this group layer is not masked so leave the background outside in this example.

2.8.6 Revealing mask method 3.

Using one ‘mask’ and one ‘full mask’. To be describe (using the linked project)

2.8.7 Tutorial files

Download tutorial files



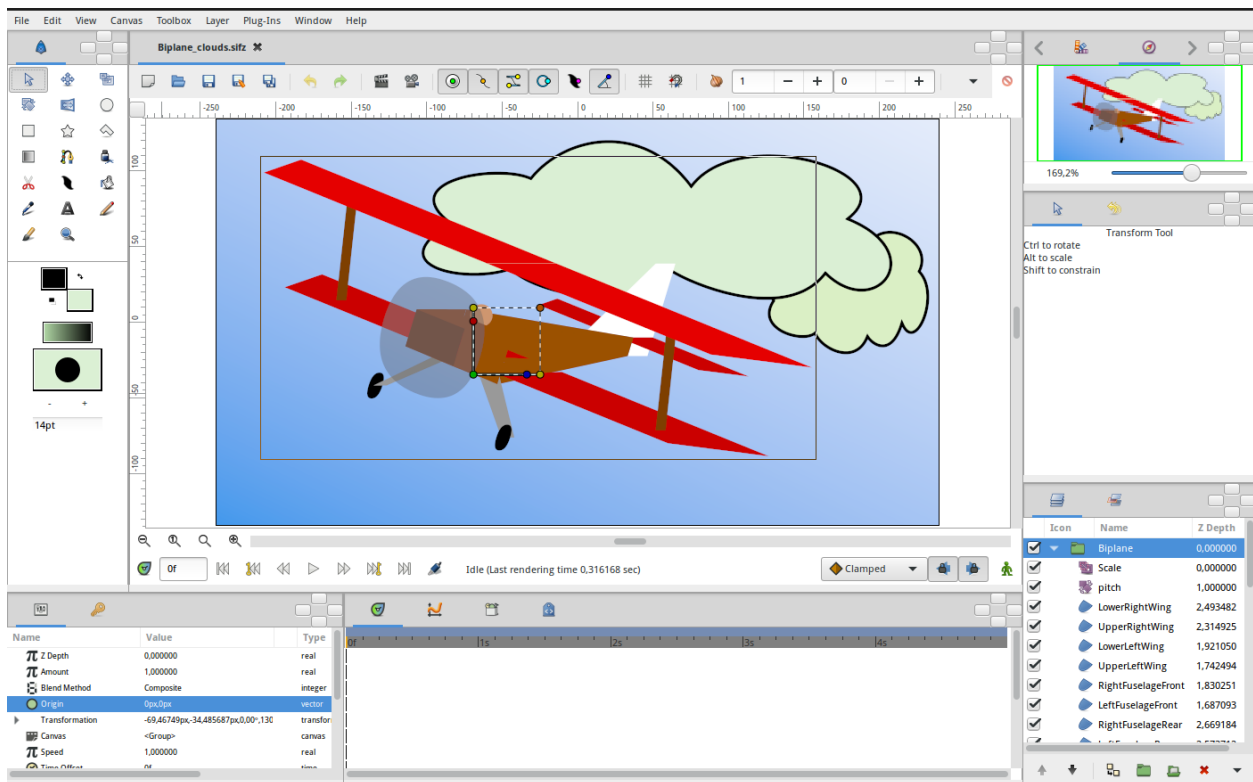
2.8.8 Next Steps

This is the end of the introductory tutorials. From here you can take a look at the [Interface](#) page, or continue reading or doing the rest of . The gives you a list of the available articles to read for a more complete understanding of Synfig.

MAIN CONCEPTS

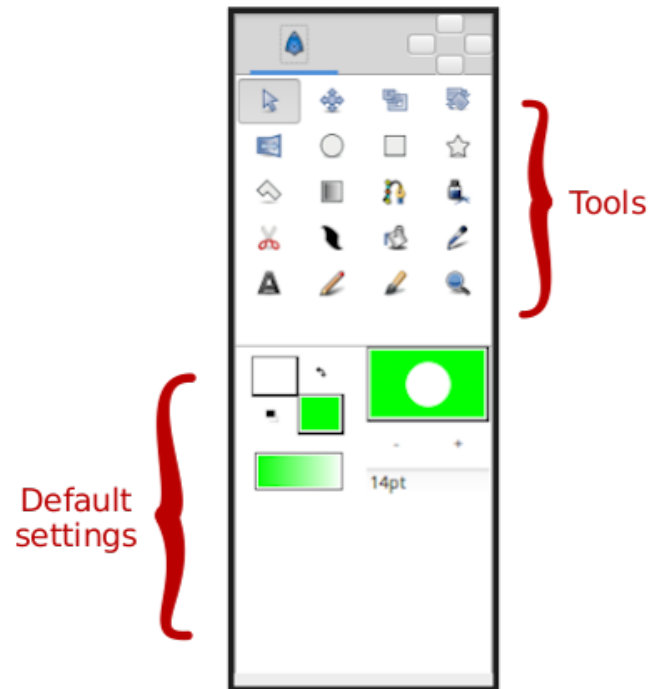
3.1 Interface

Synfig Studio start in one window, and dock a number of individual *Panels* into that window. The default layout of user interface is divided in three parts : the toolbox, the canvas and the other panels.



This page tries to give you a short introduction to the different elements you will find after opening synfig.

3.1.1 The Toolbox window

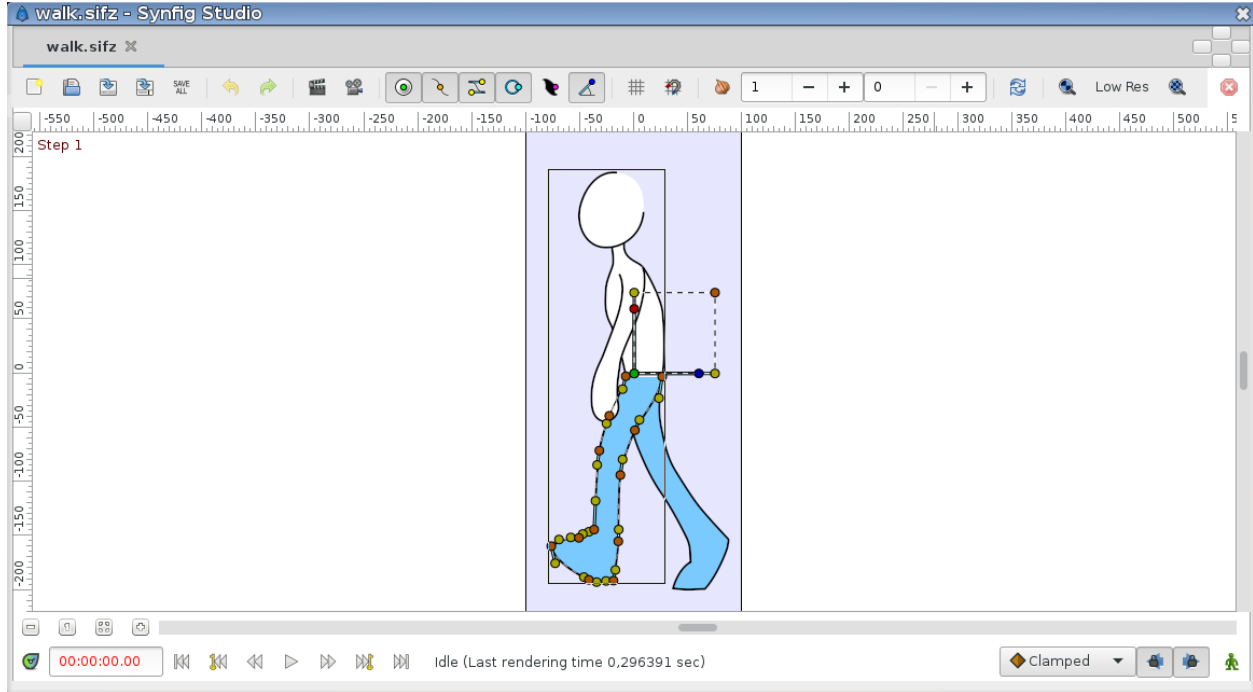


The Toolbox Panel is a major piece of Synfig : even if you have several open projects, there will only be one Toolbox.

The toolbox is separated into two areas:

- The upper palette contains the **tools** which allow you to create and manipulate objects. In synfig we will talk of *Layer*, as each object lies in its own layer. You can find a detailed description of each tool [here](#).
- The lowest palette contains the **default settings** for new layers: fore- and background colors *Brush Colors*, *Current Gradient* and Current Gradient *Current Gradient*. You can find a detailed description of each default settings at *New Layer Defaults*.

3.1.2 The Canvas window



Once you have opened a file or created a new project you'll see the Canvas Window. This is where you'll create your animation!

In addition of set of menus at the top of the screen, or the top of the drawing window, Synfig has a *Canvas Menu Caret*: the little black triangle in the upper left corner of the canvas. Beneath are all the menus you would expect and that allow you to access most of Synfig's features.

The area with the grey check-board pattern is your **working area** in which you can create elements/layers and manipulate them.

The *Timebar* that you can see in the picture here only appears when you have defined a non-zero duration in the *Canvas Properties Dialog* of your project. To the left you can see the number of the current frame and on the right side buttons to Lock Keyframes the keyframes and to switch the *Animate Editing Mode*. The *Animations Basics* Tutorial gives a good introduction on how to work with this buttons.

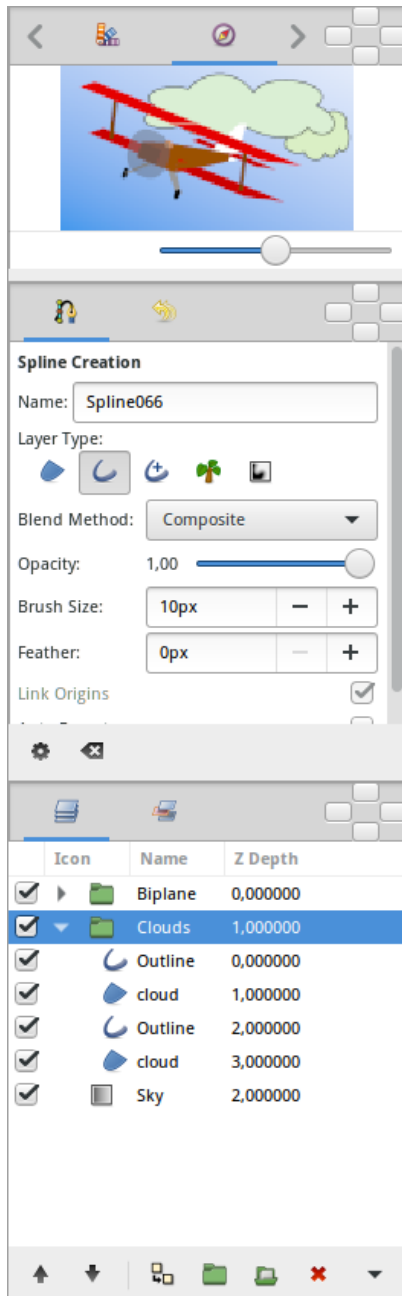
In the picture above there are three elements on the canvas, the black curve - called *Spline* - is selected. The little orange and brown dots along the spline control the curve and are called *Handles*. Also you can see the *Group Transformation Widget* of the selected *group*.

Show/Hide Menu Bar

With the menu entry `| Show Menu Bar |` you can hide the menu bar. To show again the menu bar use the same entry from the *Canvas Menu Caret*.

3.1.3 The other panels

The Navigator, Tool Options and Layers Panel



To the right side, three areas, each of which can show different panels: in this picture the *Navigator Panel*, the *Tool Options Panel* and the *Layers Panel* are active.

You can access detailed descriptions of the other panels [here](#).

The Parameter and Time Track Panel



Finally, at the bottom you can see the *Parameters Panel*, where you can find detailed *Parameters* and settings for the active element like colour, width, opacity, location and so on. To the right is the *Timetrack Panel* that allows you to create and modify *waypoints*.

3.2 Animation Mode

3.2.1 Summary

Each *Canvas* is either in Animate Editing Mode or it isn't.

3.2.2 When Animate Editing Mode is ON

When on Animate Editing Mode, each time you edit a parameter (whether directly in the *Parameters Panel* or indirectly by moving a Handle), a *Waypoint* is created to remember the change, and the position on the *Timetrack Panel* at which the change happened : you are creating an animation.

Depending on the value of the *Lock Keyframes* setting, waypoints may also be created in the next and previous *Keyframes* as well.

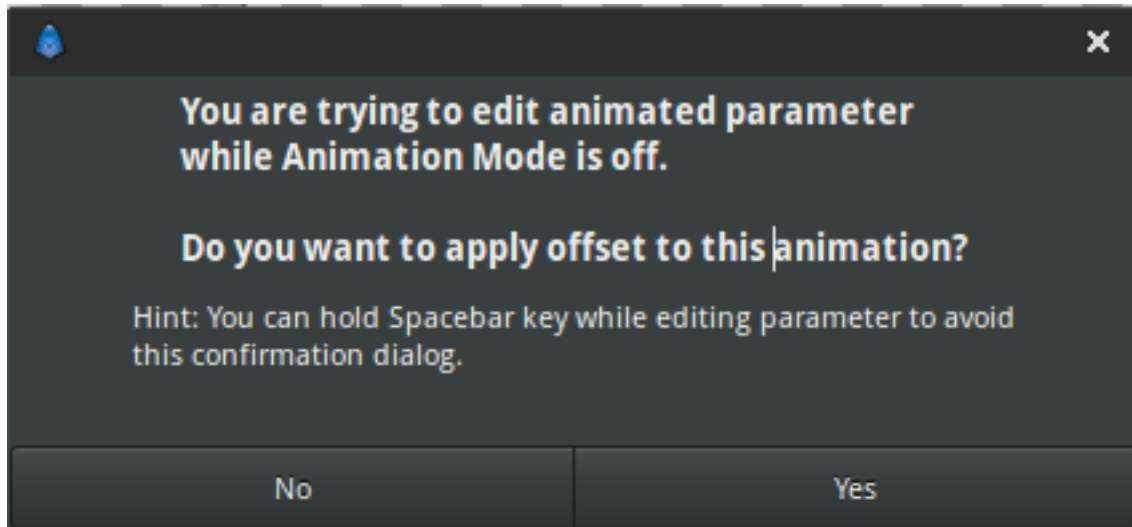
3.2.3 When Animate Editing Mode is OFF

When not in Animate Editing Mode, changes to a parameter will be applied throughout the entire timeline of the animation.

If there are already waypoints on the timetrack (*ValueNodes*) for the parameter in question, this causes a warning and your edit will be not allowed. A warning message asking you : “Do you want to apply offset to this animation?” will offers the opportunity to apply an offset for this parameter for the entire timeline of the animation.

3.2.4 Toggle Animate Editing Mode

Animate Editing Mode can be toggled on and off by clicking the round button to the right of the timetrack at the bottom of the canvas window. It will only be visible if your canvas has a non-zero end time, and will only be active if you're not currently using a tool which disables the timetrack, such as the *Spline Tool* or the *Draw Tool*.



Name	Icon	Behavior
Animate Editing Mode Off		Changes will be applied throughout the entire timeline of the animation.
Animate Editing Mode On		Changes will create a <i>Waypoint</i> to remember the value.

When toggling the Animate Editing Mode ON, the *Canvas Window* will be highlighted with red border line.

3.3 Waypoints

3.3.1 Introduction

Waypoints are the graphical symbols that show up on the *Timetrack Panel*. Each time the value of a parameter is edited in *Animate Editing Mode*, a Waypoint will be created. For example, when you adjust a *Tangent Handle* of a vertex in a *Spline* <[https://en.wikipedia.org/wiki/Spline_\(mathematics\)](https://en.wikipedia.org/wiki/Spline_(mathematics))>, a Waypoint will be created to note the change. However, you didn't really adjust the tangent, you adjusted the two parameters which define the tangent: its angle (Theta), and its length (Radius). Opening up the tangent's nodes in the *Parameters Panel* will show that Waypoints have been created against each of these 2 parameters.

As a result of adjusting a single *Tangent Handle*, Waypoint have been created as follows:

- the length of the tangent changed (Radius)
- the angle of the tangent changed (Theta)
- the tangent changed (Tangent 1)
- the vertex changed (Vertex 001)
- the vertices changes (Vertices)

Only the first 2 of these are 'leaf waypoints' - the rest are parents (and grandparents, etc) of these 2 waypoints.

3.3.2 Selecting Waypoints

Holding Ctrl allows you to select more than one waypoint for a given valuenode at a time.

*When nothing is selected, clicking on a point in either normal mode or additive mode will select the time point closest to the click. Subtractive click will do nothing.

*When things are already selected, clicking on a selected point does nothing (in both normal and add mode). Add mode clicking on an unselected point adds it to the set. Normal clicking on an unselected point will select only that one time point. Subtractive clicking on any point will remove it from the the set if it is included. Normal click out of a waypoint unselect all.

3.3.3 Editing Waypoints

Move it

Waypoints can be dragged left or right with the left mouse button to change the time at which they act upon their parameter(s). If you drag a parent's Waypoint, all its child Waypoints will move with it.

Move some

You can select multiple Waypoints using Ctrl|Click, and then they can be dragged left or right has moving single waypoint *Waypoint: Move It*.

More actions

Right-clicking on Waypoint shows a context menu, containing:

- **Jump to:** Set the current time to be the time of the waypoint.
- **Duplicate:** [STRIKEOUT:Only appears for single-leaf Waypoints.] Creates a copy of the current Waypoint at the currently selected time.
- **Remove:** Only appears for single-leaf waypoints. Deletes the Waypoint.
- **Remove ``\ \*\*`n`\*\*\ `` Waypoints:** Only appears for multi-leaf waypoints. Deletes the **n** Waypoints.
- **Edit:** [STRIKEOUT:Only appears for single-leaf Waypoints.]

TODO: writeme (brings up a dialog allowing various parameters to be set).

- **Both.:** Allows you to change the In and Out interpolation ([Waypoint#Interpolation]) setting for the Waypoint and all its children.
- **In.:** Allows you to change the In interpolation setting for the Waypoint and all its children.
- **Out.:** Allows you to change the Out interpolation setting for the Waypoint and all its children.

If the Waypoint is *Linking* other menu entries related to the link will appear up to those one.

TODO: writeme (describe added menu entries in case of linked waypoint).

3.3.4 Interpolation

Default Interpolation

The default type of interpolation *Default Interpolation* for new waypoints is defined by a dropdown list at bottom of the Canvas Window.

It is possible to fix interpolation by parameter. Right-click any and choose its default interpolation from a submenu. The defined interpolation is indicated in the *Parameters Panel* with corresponding icon near the parameter value.

Interpolation In & Out

Each Waypoint has an In and an Out interpolation setting, which determines the manner in which its parameter changes - whether it changes linearly over time, or follows a curve.

The In interpolation defines how the parameter changes in the moments before the Waypoint is reached, and Out defines how it changes in the moments after the Waypoint. Available interpolation types are as follows:

- *TCB*: If you imagine a graph of the parameter's value against time, using the TCB interpolation type will fit a smooth curve between adjacent waypoints, much like the **Spline_Tool** fits smooth curves between adjacent spline vertices.
- *Clamped*: Acts like TCB interpolation, but never “overshoots” (or “undershoots”) the values you set in the *Parameters Panel*.
- *Constant*: Stops the animation.
- *EaseIn/Out*: The graph is horizontal as it leaves the Waypoint.
- *Linear*: The graph of parameter value against time is a straight line.
- *Undefined*: This is only shown for multi-leaf Waypoints, where there is more than one different interpolation type amongst its leaf Waypoints.

Graphical Representation

The color and shape of the Waypoint as displayed in the *Timetrack* indicates its interpolation type:

Waypoints

					
Clamped	TCB Smooth	Constant	Ease In/Out	CLinear	Undefined

Each Waypoint is split into two halves. The left half indicates its In interpolation and the right half indicates its Out interpolation, so many different combinations are possible:

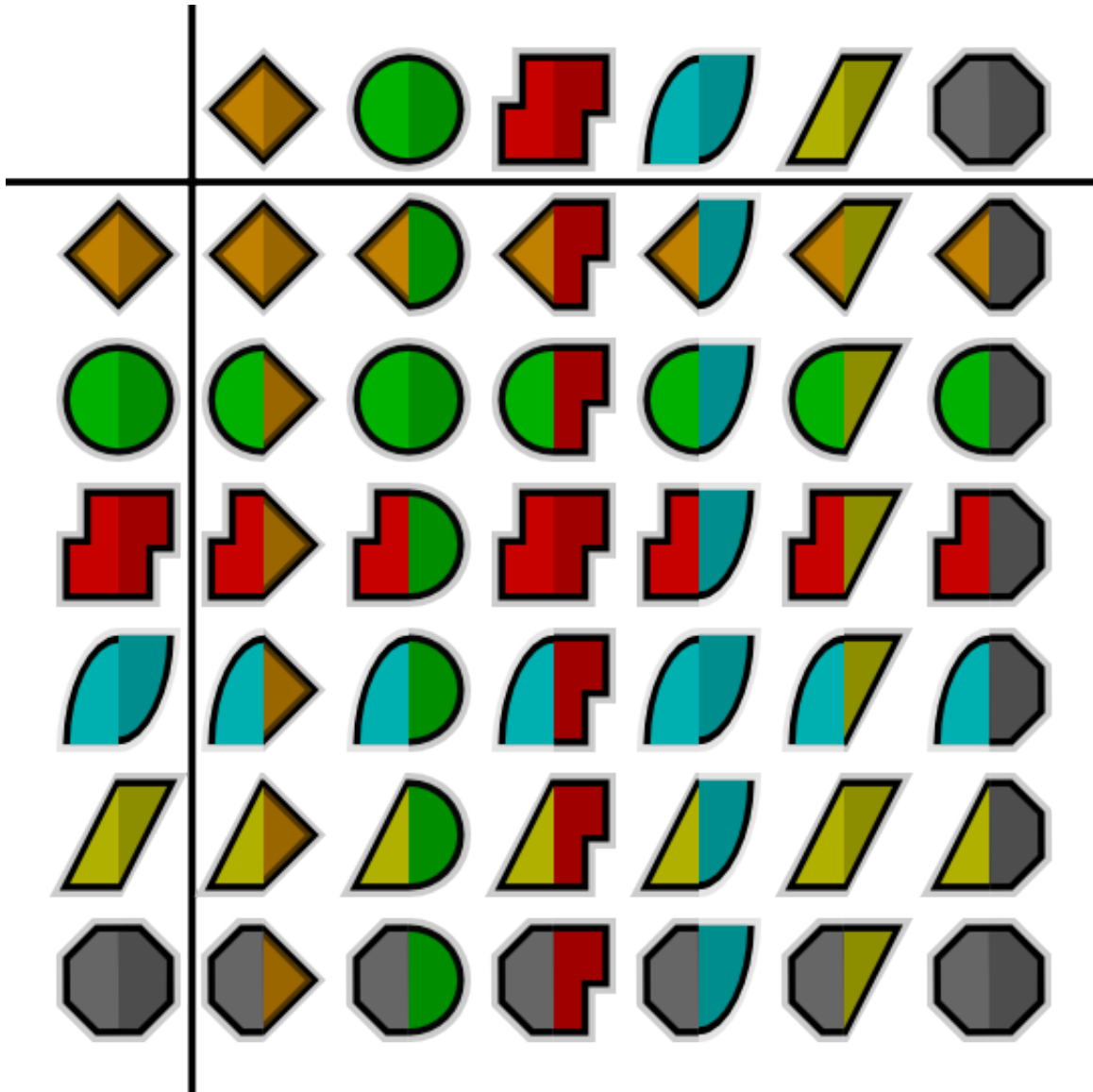
In the chart above the In interpolation is shown on the left, and the Out interpolation is shown across the top.

Note: the .sif file used to generate these screenshots is available.

Waypoint-examples.sif

Here is an *Interpolation.sif* showing 25 different blobs, and how they move with different combinations of Waypoints. It renders to a 634K .avi file, and is available in lower resolution on [YouTube](#). Notice how:

- the blobs with any red (*Constant*) on them don't move at all



- the yellow (*Linear*) sides of blobs ‘bounce’ off the walls
- the cyan (*Ease*) sides slow gracefully to a halt at the walls
- the green (*TCB*) sides bounce at the top (the animation is a single down-and-up animation, looped, so there’s no ‘context’ at the top for the TCB to fit the curve to) but act smoothly at the bottom

The ‘undefined’ (grey) symbol is used when the row in the *Timetrack Panel* represents multiple Waypoints. For example, the ‘vertices’ row represents all the vertices making up a Spline. Each of those vertices can have multiple Waypoints, each with different interpolations. If all the interpolations are the same, that interpolation’s symbol will be used. Otherwise, the grey ‘undefined’ symbol is used.

Look at the Waypoints below. They are from the Timetrack for the Vertices of an *Outline Layer*.

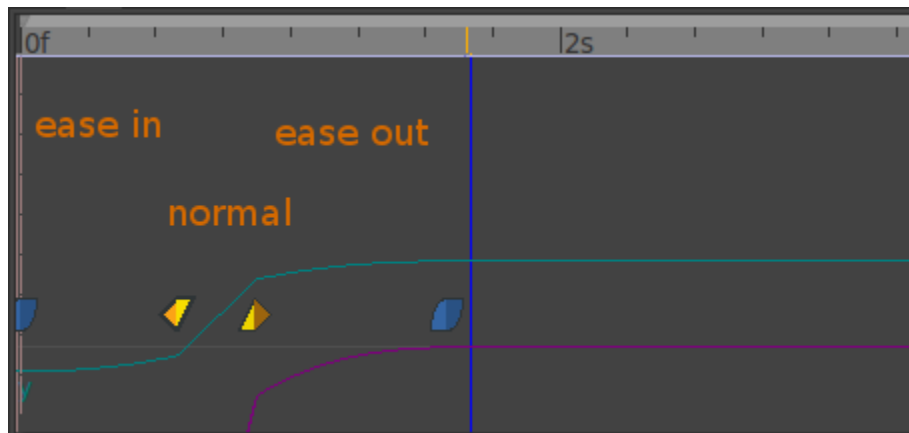


You’ll see the left side of each of the Waypoints is colored. This means the In interpolation for each vertex is the same. However, the right side is grey, indicating that the Out interpolation for each vertex differs.

TODO: rewrite the above so it doesn’t hurt the brain so much.

3.3.5 Example combining Waypoints

Ease in - Normal - Ease out



3.4 Keyframes

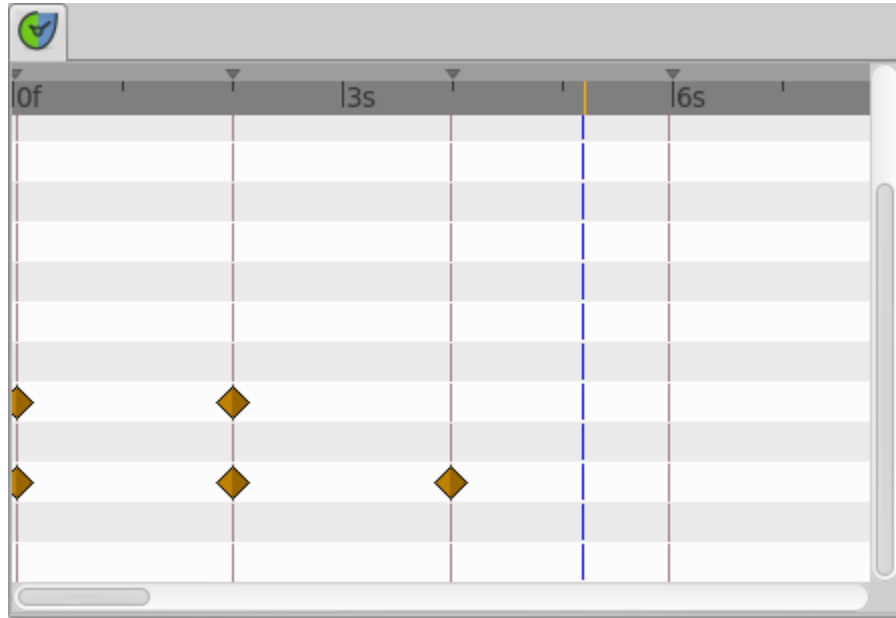
3.4.1 What is a keyframe?

A keyframe is a basically a “mark” in the timeline. This mark allows the user to make Synfig remember the state of the animation at that point (frame). It means that the keyframe is like a label that tell Synfig that this frame should be taken into account when creating waypoints. It also indicates that the marked frame is a special frame where the information of *every parameter of every layer is stored in order to be reused later*.

Each keyframe is associated with a particular frame and a frame can only have one keyframe.

3.4.2 What does a keyframe looks like?

A keyframe looks like a light brown vertical dashed line in the *Timetrack Panel* placed at the corresponding frame. You can distinguish it from the *Time Cursor* by its color (the time cursor is blue).



The symbols shown in the image are *Waypoints*.

The keyframe representation in the *Timebar* change according their states : Normal, Selected or Deactivated



Three keyframes with three different states : **Normal**, **Selected** and **Deactivated**

Keyframes also appear as entries in a list in the *Keyframes Panel*

Documentation writers note: You can download the project to generate the screenshot: `Keyframe-looklike.sifz`

3.4.3 Keyframes and waypoints

A keyframe doesn't necessarily imply a waypoint, and a waypoint doesn't necessarily imply a keyframe.

A keyframe could live all the time without any waypoint but it stores the information of the values of the parameters on that specific frame. If there is a waypoint there then the waypoint information (only the parameter value) is stored too. If there is no waypoint in the keyframe then its "stored" value is the result of the surrounding waypoints, its parameter values and the interpolation values the waypoints have. This means that a keyframe remembers the values of the parameters at that frame but does not keep them static at that frame. To maintain a parameter's value static in a certain frame you must use a waypoint.

The creation of a waypoint can cause the creation of new waypoints on the neighboring keyframes depending on the current value of the *Lock Keyframes* state. So, maybe, the creation of a waypoint (modifying a parameter or pasting

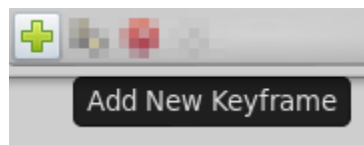
	Time	Length	Jump	Description
☒	0f	2s	(JMP)	
☒	2s	2s	(JMP)	
☒	4s	2s	(JMP)	
☒	6s	0f	(JMP)	

or moving a waypoint or even duplicating a keyframe) can lead to the creation of a waypoint in the keyframes that are immediately before and after the inserted waypoint's frame. The waypoints created in the neighboring keyframes are created according to the *New Layer Defaults: Default Interpolation* in the *Toolbox*.

See the *examples* to understand how this works.

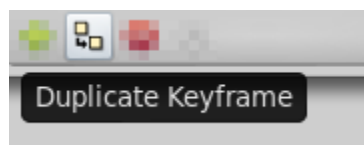
3.4.4 Adding, duplicating and removing keyframes

Add a keyframe



Place the time cursor at a frame where there isn't currently any keyframe. Then press the **Add new Keyframe** button. If you place the time cursor at a frame where there is currently an existing keyframe or if animation Start Time equals animation End Time (animation Duration is 0m 0s 1f) then the **Add new Keyframe** button is disabled. Once you press the button then a new entry is added to the list of keyframes and a vertical dashed line is added in the time line. No waypoint is created.

Duplicate a keyframe



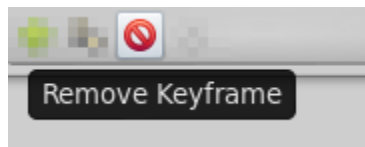
Select a keyframe in the keyframe list of the *Keyframes Panel* and place the cursor at a frame where there isn't currently any keyframe. Then press the **Duplicate Keyframe** button. This would have two separated effects:

1. If there is a waypoint at the original keyframe then the waypoint is duplicated. Its duplication includes the parameter value and its interpolation types.
2. If there is no waypoint in the original keyframe for any particular parameter then two things could happen:
 - There is no waypoint for that parameter at ANY frame in the time line: Then NO waypoint is created.
 - If there is a waypoint in the time line for that parameter, but not in the keyframe that is going to be duplicated, then in the duplicated keyframe is created a new waypoint with a value for the parameter of the result of the current value at the original keyframe and a TCB Smooth interpolation type for both In and Out.

Of course, duplicate a keyframe will produce a new keyframe at the place pointed by the time cursor and will add a new one to the keyframe list in the proper place. In the keyframe list, the new added keyframe have the same description than the original, plus a (Duplicate) at the end.

[video explanation](#)

Remove a keyframe



Just select a keyframe from the keyframe list and press the Remove keyframe button. It will remove the keyframe and all the waypoints for all parameters for all layers that are currently there.

NOTE: If you move a keyframe by modifying its `time` in the keyframe list dialog and immediately press the Remove Keyframe button then the waypoints are not deleted. It seems to be a bug but also can be considered a feature if you really want to keep the waypoints and not the keyframe.

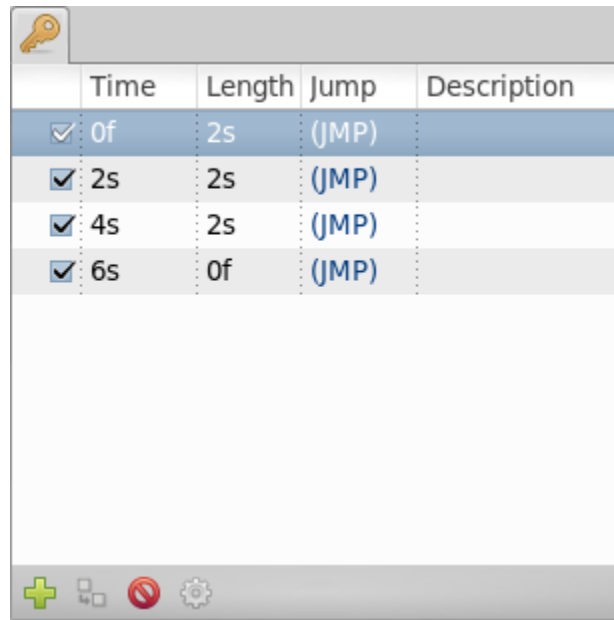
3.4.5 Editing keyframes: time, length & description

You can see in the Keyframe list dialog that there are four headers and before that, an empty column. This empty column maintain checkboxes related to keyframe activation : enabled or disabled.

- “Empty” [CheckBox]
- Time
- Length
- Jump
- Description

Activation

By changing the state of the checkbox you can activate or disable the keyframe. A visual information about the keyframe state is displayed in the *Timebar*.



	Time	Length	Jump	Description
☒	0f	2s	(JMP)	
☒	2s	2s	(JMP)	
☒	4s	2s	(JMP)	
☒	6s	0f	(JMP)	

Time

You can modify the time (frame) where the keyframe is placed just making a click in the corresponding Time cell. It will allow modify the time forward or backward the amount that you want. You can also manually place a keyframe at the desired time using the *Timebar*.

Modifying the Time of a keyframe has the following effects:

1. The existing *Waypoints* in the keyframe will move to the new position.
2. If any parameter have a a waypoint in the time line, then the moved keyframe will have a new waypoint set to *New Layer Defaults: Default Interpolation* on those paramter(s).
3. According to the default interpolation method and the *Lock Keyframes* status and to the parameters that have any waypoint in the time line, new waypoints will be created on the neighbouring keyframes of the destiny time (frame). The original neighbouring keyframes will be untouched if don't coincide with the destiny neighbouring keyframes.
4. If a keyframe is displaced and doesn't "jump" over other existing keyframe then the waypoints that are surrounding the original position of the moved keyframe are compressed and / or expanded in the timeline depending on the displacement of the keyframe. See the examples. This is a recent discovered behaviour

You cannot set the time of other keyframe. If you try to set the time of a certain keyframe to be the same time of another existing keyframe then the program gives you this message:

```
keyframe_set: Cannot change keyframe time because another keyframe already exists
with that time.
```

See *the example* to see how changing the time of a keyframe works.

Length

Length parameter sets the time the keyframe is exposed in the timeline until next keyframe. You can also manually change the length parameter using the *Timebar* and holding alt key on releasing the mouse button.

Changing the parameter shifts all following keyframes and *Waypoints* forward or backwards.

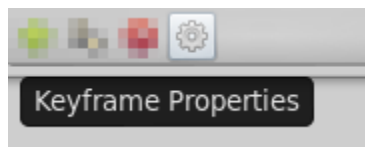
Jump

The Jump column is only a short cut to place the *Time Cursor* at the keyframe where you make a click in the (JMP) label.

Description

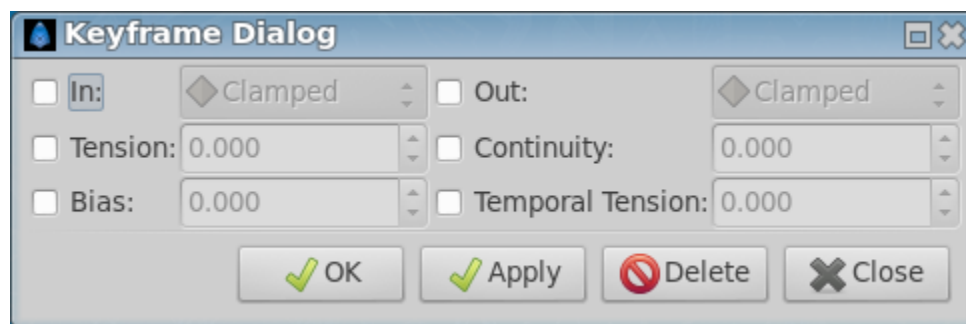
This cell allow the user insert a short description of the meaning of the keyframe. Just make click on it and change the text.

Editing Keyframe Properties



Hitting the keyframe Properties button, the *Keyframe Properties* dialog will appear. This dialog allows change the interpolation method for all the waypoints on the keyframe at the same time. Even if, for a certain parameter, there is no waypoint on the keyframe but the parameter have other waypoints in the time line, then when you apply the *Keyframe Properties* you will add a waypoint at that keyframe were there aren't currently any waypoint. The added waypoints have the interpolation methods stated by the dialog. It means that the *Keyframe Properties* dialog will modify the interpolation methods for all the parameters that have any waypoint in the time line.

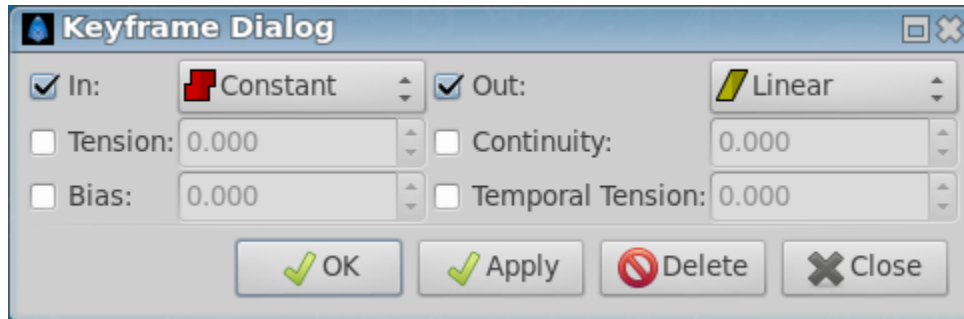
The dialog have the following parameters:



- In: Checking this value you can change the interpolation method of the left part of the waypoints of the current selected keyframe of all the layers of the canvas to the selected *Waypoints Interpolations* in the drop down menu.
- Out: Same but for the right part of the waypoint.
- Tension: See *TCB*
- Bias: See *TCB*
- Continuity: See *TCB*

- Temporal Tension: See [TCB](#)

You can check only one of both In or Out check boxes to only affect the change to the left or right part of the waypoints. The non checked part would not be modified. Same comment applies for the Manual interpolation method parameters (Tension, Bias, Continuity and Temporal Tension)



This dialog would not affect what's the interpolation method for a new waypoint created by the user, automatically created by the [Keyframe duplication](#) or by the [Lock Keyframes](#) state. The interpolation methods for new waypoints created in those cases will be both the same (In and Out or Left and Right) and depend only on the [New Layer Defaults: Default Interpolation](#) method of the [Toolbox](#) window.

See the [examples](#) to understand better how it works.

3.4.6 Edit a keyframe from the timebar



Three keyframes with three different states : **Normal**, **Selected** and **Deactivated**

You can adjust the [Keyframe time](#) of a keyframe by a normal drag and drop. To adjust its [Keyframe Length](#), hold **Alt** key when releasing the mouse button.

During a keyframe drag and drop operation a tooltip indicator will be displayed with some time informations.

Right click over a keyframe or in upper part of the Timebar open the Keyframe Menu giving access to the keyframes actions.

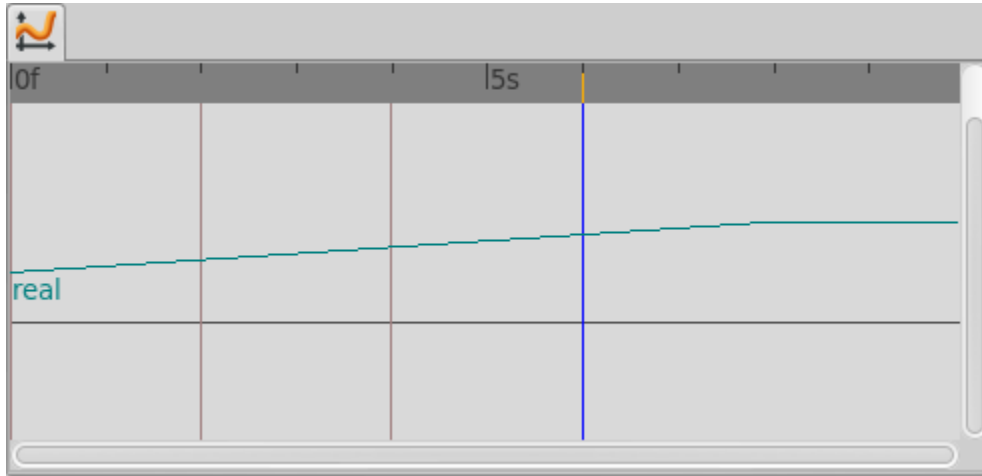
3.4.7 Examples

Duplicate a keyframe with no waypoint on it

For example, imagine that you have following set of keyframes and waypoints and the corresponding parameter of the radius of a circle:

Before duplicate keyframe at 2s to 6s

Frame	Keyframe	Waypoint	Radius	Interpolation
0s	yes	yes	20.0	TCB Smooth
2s	yes	no	25.0	n/a
4s	yes	no	30.0	n/a
8s	no	yes	40.0	TCB Smooth



Note: Notice that although the interpolation between 0s and 8s is TCB Smooth the real result is linear due that they are the only two waypoints of the animation for that parameter.

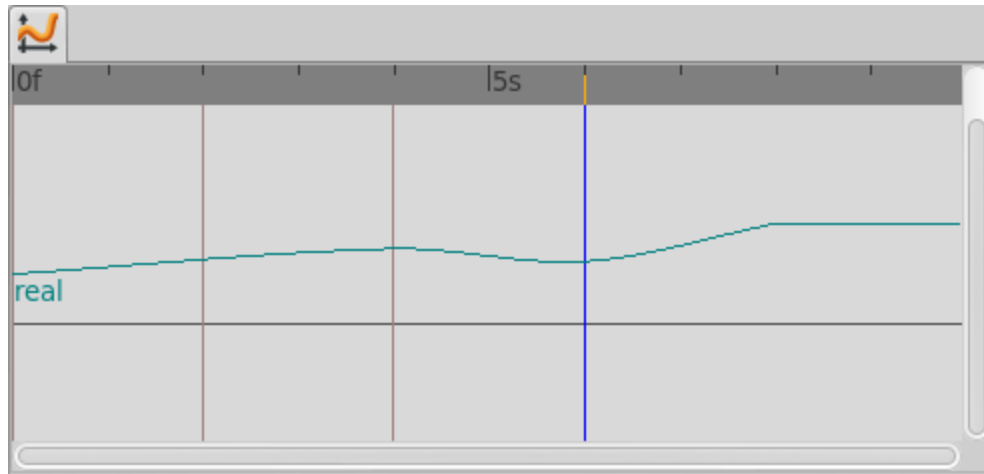
If you select the keyframe at 2s, place the time cursor at 6s (where there isn't a keyframe), set the *New Layer Defaults: Default Interpolation* to *TCB*, and have the *Lock Keyframes* to *All keyframes* locked and press the *Duplicate keyframe* button, then the result is the following:

After duplicate keyframe at 2s to 6s

Frame	Keyframe	Waypoint	Radius	Interpolation
0s	yes	yes	20.0	TCB Smooth
2s	yes	no	25,78125	n/a
4s	yes	yes	30.0	TCB Smooth
6s	yes	yes	25.0	TCB Smooth
8s	no	yes	40.0	TCB Smooth

You can see that:

1. At 0s none has changed. Not affected by the insertion of the keyframe. It is two keyframes away from 6s and also have a waypoint.
2. At 2s there was a keyframe and stills there. But previous to the creation of the keyframe at 6s the current interpolated value of the radius was 25.0. After the creation of the keyframe at 6s the radius is the result of the interpolation between 0s and 4s frames waypoints with its radius values and its interpolation methods. That is 25.78125. This keyframe is more than one keyframe away from the new 6s keyframe so no waypoint is created.
3. At 4s there was a keyframe and still being there. But in this case the 4s keyframe is a neighbor of the new 6s keyframe. As well as the lock keyframe state was set to *All keyframes* locked then the keyframe at 4s has been locked adding a waypoint on it. The radius value hasn't changed (still being 30.0) because it was locked adding a waypoint with its current value). The Interpolation mode of the waypoint was set to *TCB Smooth* as stated by its default value.

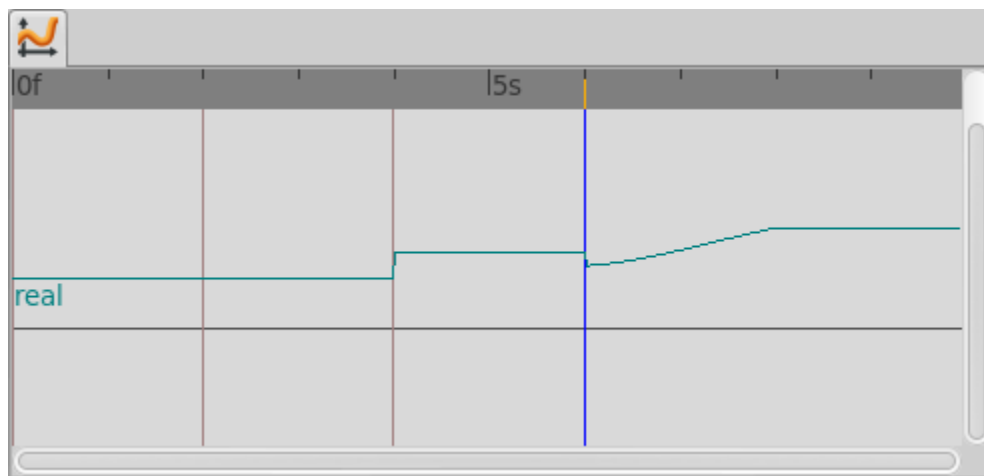


4. At 6s there is a new keyframe with a new waypoint with the old value of the interpolated value of the keyframe at 2s. That is a radius of 25.0.
5. At 8s nothing has changed. There wasn't any keyframe and there was a waypoint so nothing is expected to change.

Return to the previous state before you duplicate the keyframe with the *History Panel*, and imagine now that you do the same operations but you choose the default interpolation set to *Constant*. Then the result is the following:

After duplicate keyframe at 2s to 6s (constant interpolation)

Frame	Keyframe	Waypoint	Radius	Interpolation
0s	yes	yes	20.0	TCB Smooth
2s	yes	no	20.0	n/a
4s	yes	yes	30.0	Constant
6s	yes	yes	25.0	TCB Smooth
8s	no	yes	40.0	TCB Smooth



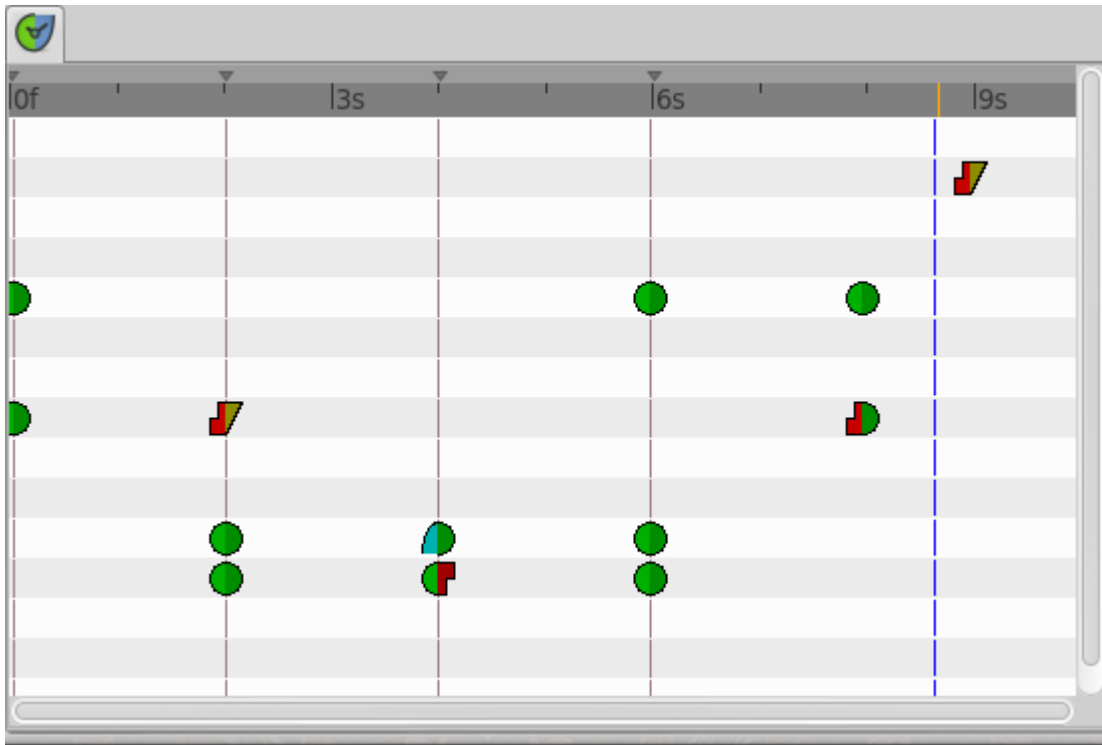
Now you can see that the keyframe at 2s doesn't hold the value of the parameter by itself. It only remember the value if a waypoint is created on it, by the result of the insertion of a neighbour waypoint, or if a keyframe is duplicated and the lock keyframe status affects that keyframe. In this example the value at 2s has changed drastically due to the different interpolation method for the created waypoint on 4s. If in this situation you duplicate again the keyframe at 2s to other

frame (ej. 10s) then it would copy a keyframe with a waypoint on it with a radius's value of 20.0, what is the current value of the parameter in that keyframe before duplicate it.

Documentation writers note: You can download the project to generate the screenshot: `Keyframe-example1.sifz`

Editing Keyframe Properties

Consider this situation for a certain layer:

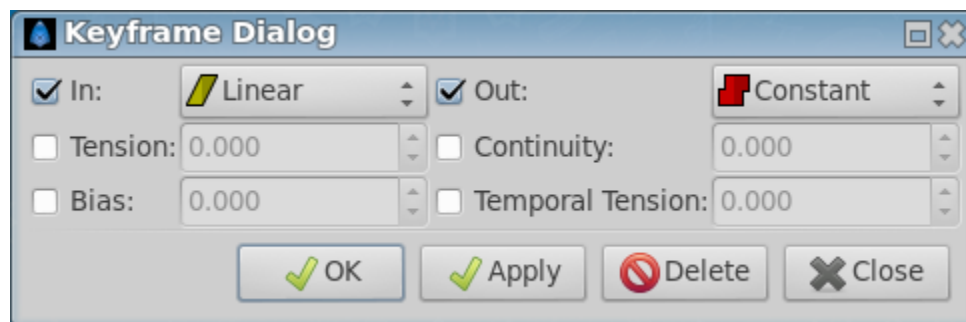


In the sample the animation duration is 10 seconds so the image shows all the existing waypoints and keyframes. The time cursor isn't over any keyframe.

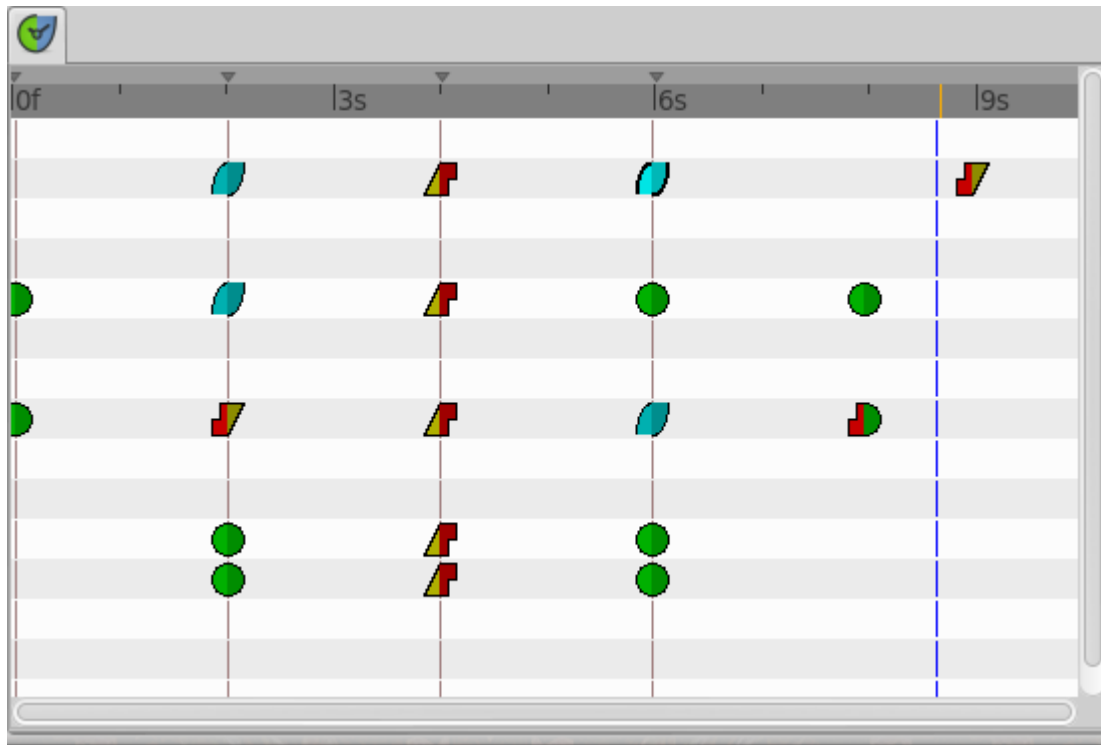
Now consider that you have the following default values:

- *New Layer Defaults:* *Default Interpolation* method set to *Ease in/out*
- *Lock Keyframes* status set to *All Keyframes Locked*

Now select the keyframe at frame 4s in the keyframe list. Press the *Keyframe Properties* button and set the following interpolation method:



and press **Apply** button. The result will be this:



You can see the following effects:

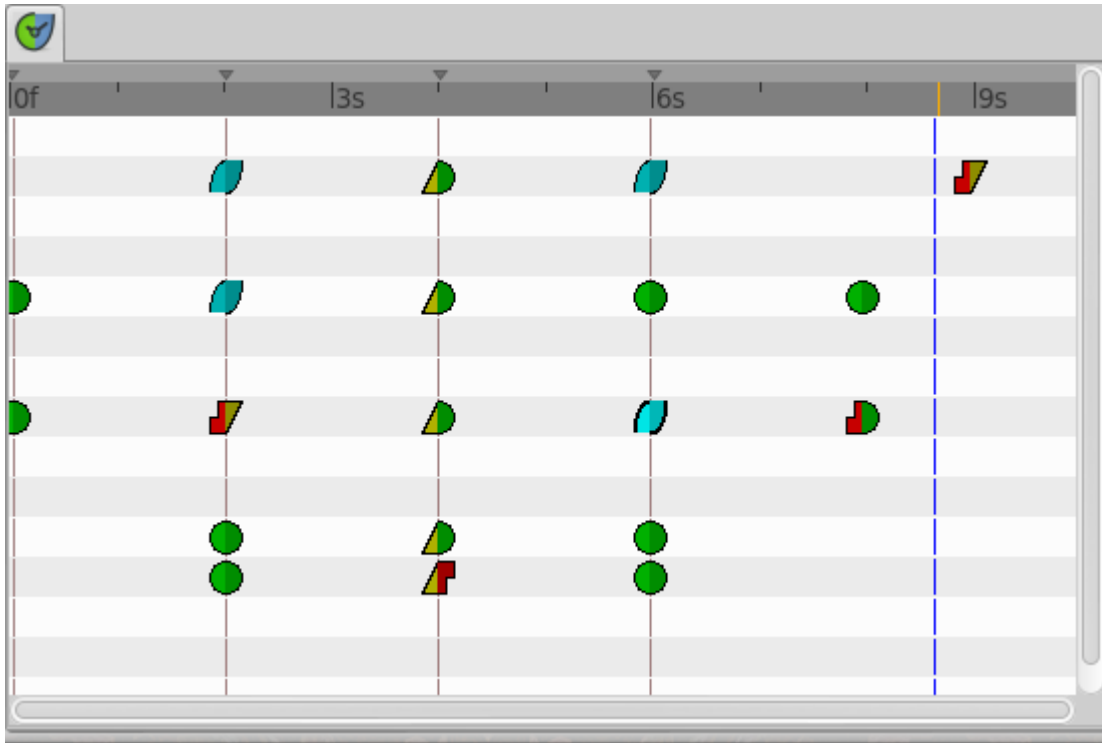
1. The existing waypoints at 4s keyframe have changed its interpolation methods according to the **Keyframe Properties** dialog.
2. There are new added waypoints at 4s keyframe. The waypoints are added to the parameters that have almost one waypoint in the time line (for example the one that have only a waypoint at 9s). The added waypoints at 4s keyframe have the interpolation settings that was stated by the **Keyframe Properties** dialog.
3. New waypoints have been created for the neighbouring keyframes to 4s (2s and 6s) for all the parameters that have any waypoint in the time line. The waypoints are created in the neighbouring keyframes according to the *Lock Keyframes* status. Also the created waypoints interpolation method responds to the *New Layer Defaults: Default Interpolation* method you have set.

If in the **Keyframe Properties** dialog you were checked off the **Out** or the **In** check boxes then it would have happened two things:

1. The existing waypoints at 4s would only change its interpolation method on the side the check box was checked on. The other side will be untouched.
2. The new added waypoints will have the interpolation method set to **TCB Smooth** method where the check box is off and the interpolation method set by the **keyframe properties** dialog where the check box is on.

In this sample it was only checked on the **In** check box.

Documentation writers note: You can download the project to generate the screenshot: `Keyframe-example2.sifz`



Change Keyframe Time

Without waypoints between keyframes

Consider again this situation for a certain layer:

Now consider that you have the following default values:

- *New Layer Defaults: Default Interpolation* method set to *Ease in/out*
- *Lock Keyframes* status set to *All Keyframes Locked*

Now select the keyframe at frame 4s in the keyframe list. Make a click in the Time cell and modify the time to be 3s. The result will be this:

With waypoints between keyframes

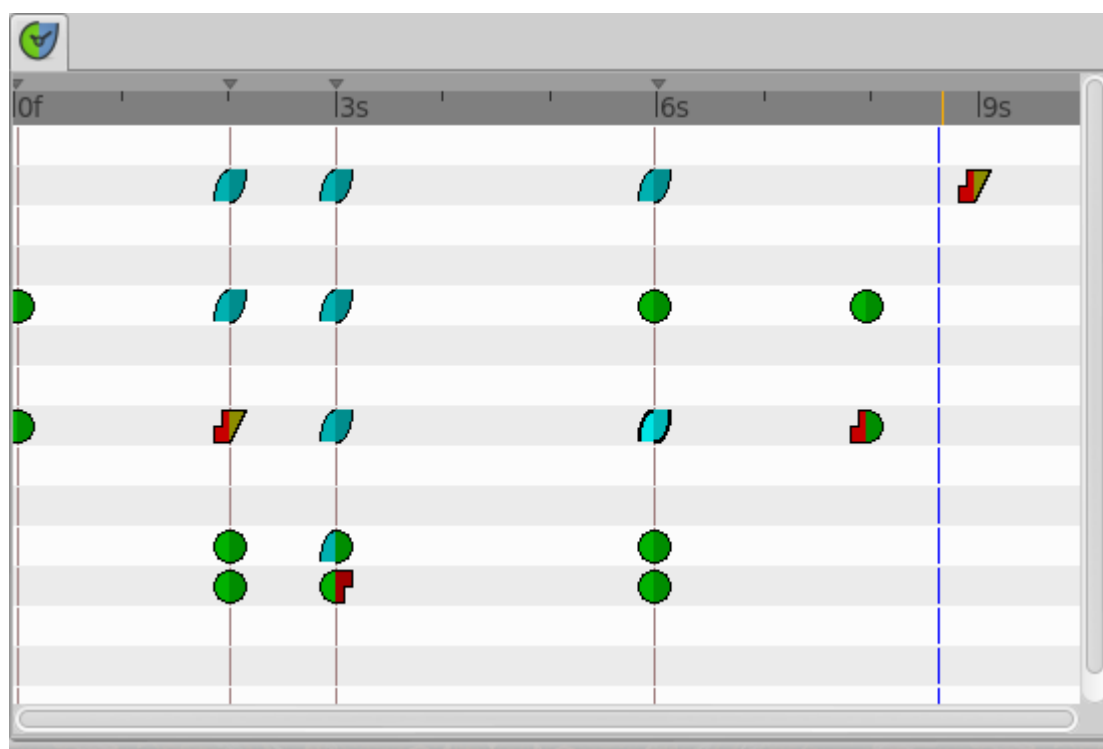
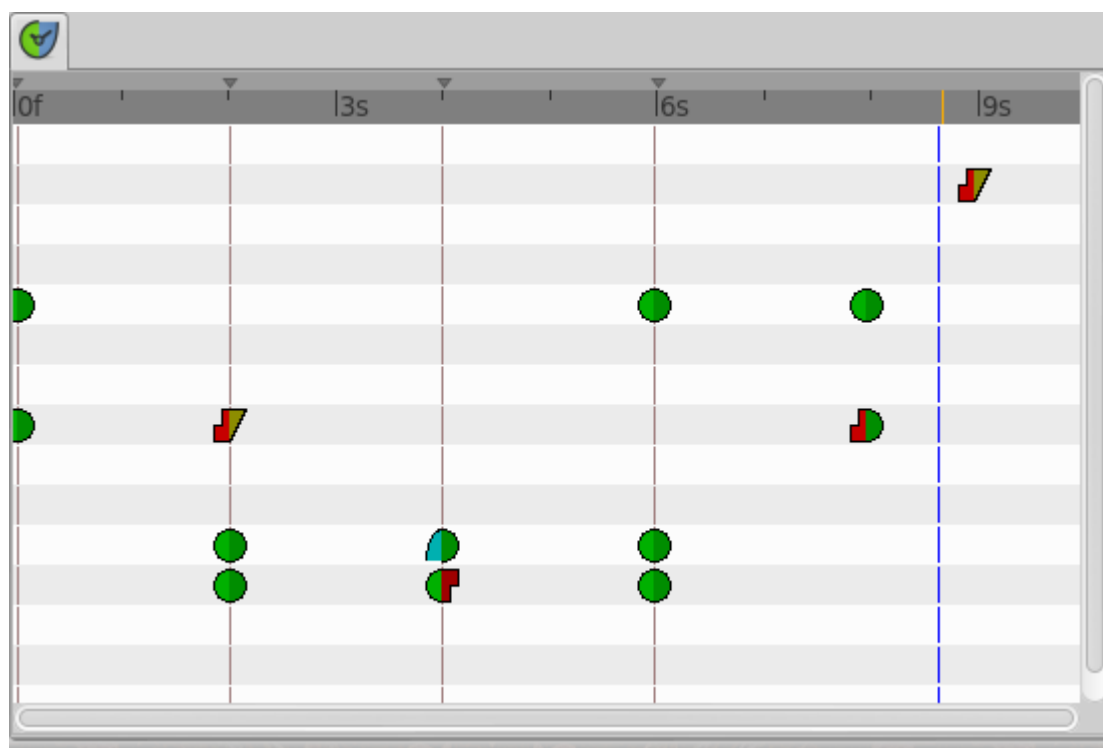
Consider now this situation for a certain layer:

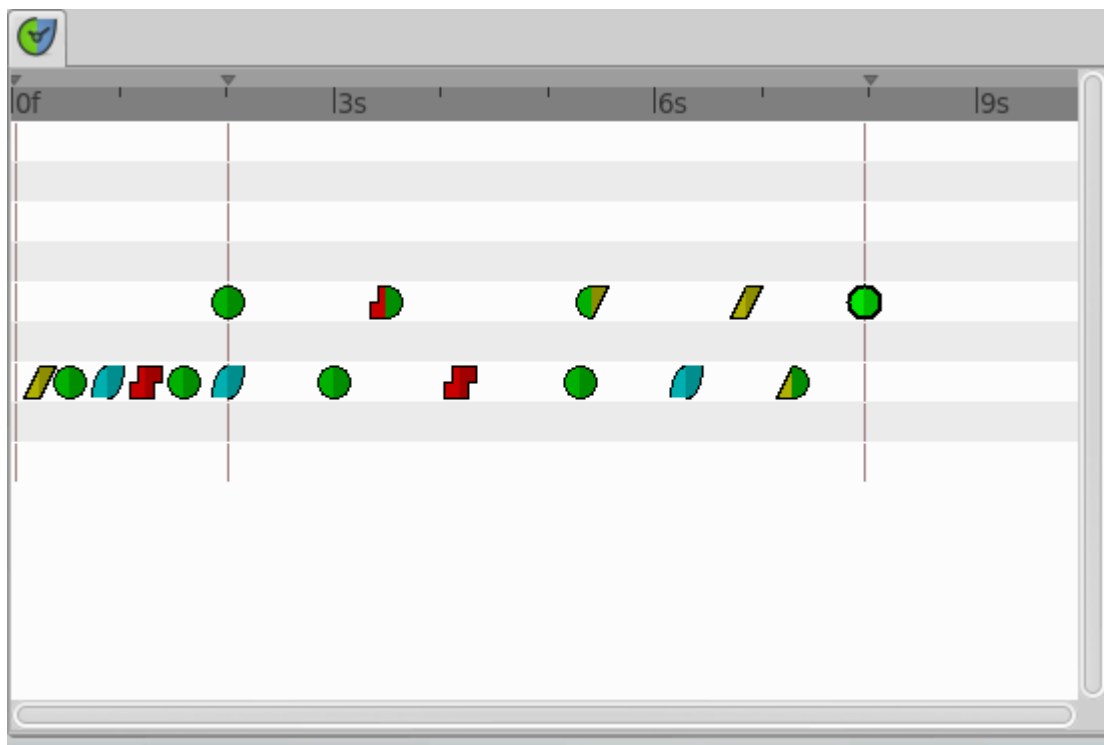
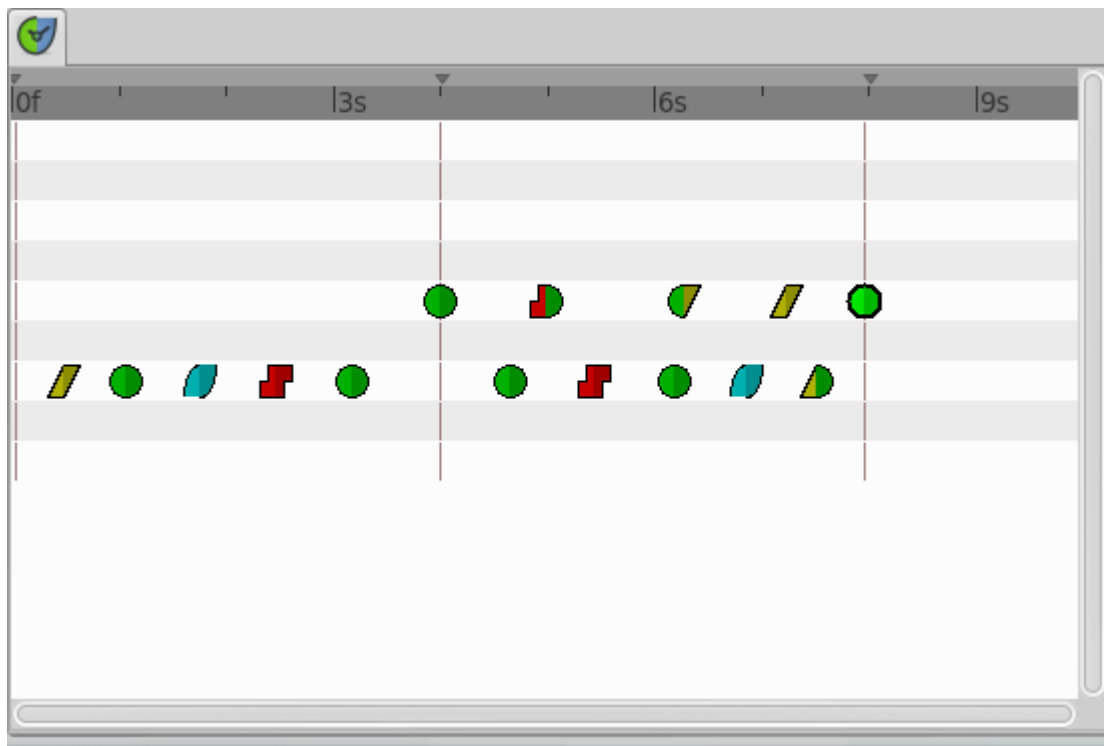
Now consider that you have the following default values:

- *New Layer Defaults: Default Interpolation* method set to *Ease in/out*
- *Lock Keyframes* status set to *All Keyframes Locked*

Now select the keyframe at 4s in the keyframe list. Make a click in the Time cell and modify the time to be 2s. The result is this:

You can see how the waypoints at right and left of the moved keyframe have been compressed and expanded in the time line. Also notice that any waypoint has been formed in the moved keyframe at the parameter at the bottom of the list but yes in the static keyframes.





Note: It seems to be a bug (?) - to be verified.

Trying to understand this behaviour I see that also the keyframes keep the waypoints between two adjacent keyframes although you move them, keeping the distribution of the waypoints in the portion of time line between keyframes. This behaviour doesn't happen if the moved keyframe "jumps" over other keyframe when moved. Please add here as much information you can discover about keyframes behaviour. It seems that there are some bugs and any information is welcome

Documentation writers note: You can download the project to generate the screenshot: `Keyframe-example3.sifz`

3.4.8 Advanced uses of keyframes

Reusing keyframes

If you want to learn more about advanced uses of keyframes see this tutorial about reusing animations. Keyframes can be like stored "poses" that can be reused several time in the animation. Very useful for lip sync.

Reuse Animations

Usage of Onionskin

To properly use the onion skin feature (`Alt+0` or `Menu Caret -> View -> Toggle Onion Skin`) you should consider the frame where the keyframes are set. Onion skin will show you the before and after keyframes images with a 50% opaque copy of the current view. Also the current view is 50% opaque.

See *Onion Skin* for more detail.

Quickly creating/importing Keyframes

Using a [Synfig plug-in](#) it is possible to quickly import keyframes from a list of timings, as explained in the *Audio Synchronisation* tutorial.

3.5 Linking

3.5.1 Introduction

It's possible to link two *handles* together so that when one moves, the other moves with it, like it is explained in *Animating Shapes: Adding the petals*.

Automatic links

This is done automatically when you create an *Outline Layer* and *Region Layer* at the same time in the *Spline Tool*; a single spline is created, and is linked to the vertices parameter of both the outline and the region. This allows you to move the outline around and have the region it surrounds automatically move with it.

Fig. 1: With "Link Origins" checked : Move the outline around and have the region it surrounds automatically move with it.

Manual links

To link some handles together, select them by selecting their layer(s) and dragging a box around them, or by holding the **Control** key and toggling individual handles on, then right-click on one of the selected handles and select “Link” from the context menu.

Link offset by Origin

Everything links perfectly as long as neither origin is moved before the link !

However, as soon as you move either origin, it offsets the linking location by however far the origin was moved. So, it appears that a shape moved the origin of the other shape after she placed on the origin. The way to fix it easiest is to not move the origin before you link.

Link offset by origin, is not absolutely intuitive, but allows for some powerful and flexible editing when things get complicated. Play with it until you understand it ;-)

Handles type

All of the selected *handles* must be of the same basic type. You can't link a real value to a vector value. You can usually link a vertex to a tangent if you like, because they are both vector values. I'm not sure it makes much sense to do so, however.

3.5.2 Which Handle Moves and Which Stays Still?

So you've selected a bunch of handles, right clicked to bring up the menu and clicked 'link'. Which of the handles' positions is used as the new shared position?

It works like this: a tiered decision process:

Tier 0

Tier 0 is applied only for handles which are simple values (ie. none of them correspond to a *ValueNode*). There could be two cases:

- If each handle is from a different layer, then the 'first' one is the one from the topmost layer. Before that the one from the lowest layer was used.
- If the handles are from the same layer, it seems to be the handle listed last in the parameter dialog that has its value used.

This case won't happen if any of the handles are animated, converted, or already linked to anything else, including being part of a spline.

Tier 1

If any of the handles are an exported value:

- If all the handles which are an exported value are the same exported value, then that value is used.
- Otherwise it's an error; linking isn't allowed to change an exported value.

Tier 2

So none of the handles are an exported value.

If any of the handles are referenced more than the others, then one of those is used.

What does 'referenced' mean? It's a count of how many times the value is used. If 2 handles are linked together, that value will be referenced twice.

Notice that if you draw an outline and region at once using the spline tool, the points in the created layers will each have a reference count of one, since the points themselves aren't linked to each other. Rather, it is the blines (the lists of blinepoints) that are linked. On the other hand, if you draw an outline and a region separately and link their vertices individually, then each vertex will have a reference count of 2.

Tier 3

If it's still not decided which one to use, and some of the handles are animated and others are constant values, then one of the animated ones will be used.

Tier 4

After that, if two or more handles are animated, the one with the most waypoints gets priority.

Tier 5

If it's still not decided, then the one that was least recently modified will get priority.

So if all other things are equal, you can decide which handle gets moved by nudging the one you want to move a little just before linking.

Tier 6

If even the modification date of the ValueNodes is the same, there's nothing to base the decision on and so the 'first' handle's value is used.

3.6 Exported Parameters

Exporting and converting a *Value Node* are among the most interesting features of Synfig. A combination of them allows the user to organize complex animations modifying just a few *parameters*. You can right-click any parameter in the *Parameters Panel* and Export it. This adds it to the *Library Panel*, and allows it to be used as the value of other parameters. You cannot export two parameters with the same name.

3.6.1 Unexport

You can delete an exported parameter. To do this right click on the exported parameter (or the *Value Node*) and select “Unexport”. It will delete the current selected exported Value Node but it will not affect to any parameter that was connected to the exported *Value Node*.

Exported canvases cannot be deleted for the time being.

3.6.2 Scope

Exported *Value Node* are visible in the *Library Panel* only for the current edited canvas. So if you export the canvas parameter and then double click in the *Canvas Dialog* you will open the canvas in edit mode in another window. You can export a parameter while editing that canvas, and then the exported parameter will be visible only in the canvas scope. It will allow have same exported name for several *Value Node* in different canvases.

3.6.3 Rename

You can also rename an exported parameter. Just make right click on the exported parameter (or the exported Value Node). It will not affect to the connected parameters that will remain connected to that renamed *Value Node*.

3.6.4 Convert

You can convert any exported *Value Node* using the *Convert* right click menu. But there is a difference: converting a connected parameter will disconnect from its exported *Value Node*. But if you convert a Value Node all the connected parameter will follow the converted Value Node.

3.6.5 Reuse

Sometimes you would like to do the same combination of exported values and convert types. And it is so tedious to repeat each time. There is a way to avoid this and reuse the exported and converted values.

3.7 Exported Canvases

3.8 Static Parameters

Some *Parameter* can be defined has statics : they will **not accept changes** over the time.

Once Static Parameters are toggled to no static (ie *ValueNode* , like any other parameters in Synfig) they can be animated, so that they change over time. This can be confusing, **make sure** you aren't in *Animate Editing Mode* toggling from Static to no static parameter, unless you know what you're doing!

- Static parameter representation in parameter panel: 
- Change static to no static (right click over param panel)
- ex: *Time Loop Layer* parameter static by design

3.9 ValueNode

A ValueNode is a value that can change over time.

A ValueNode represents a description of a value and how it changes (or doesn't!) over time.

ValueNode are the things we see as the value for every parameter of every *Layers*. The *Waypoint* visible in the *Timetrack Panel* are also part of the ValueNode - waypoints are how Animated ValueNode work out what value to be at each point in time.

Each ValueNode (and hence each *Parameter* in Synfig has one of 13 Types.

3.9.1 Kinds of ValueNodes

There are three kinds of ValueNode in Synfig. In the following examples the ValueNode's type is a *real number* in each case:

Constant ValueNodes

These have a single value for all time, and no waypoints. An example of such a ValueNode would be:

```
"3.4, for ever"
```

Animated ValueNodes

These have **Waypoints** that specify their value at particular points in time. For times which don't have a value specified, the value is calculated by interpolating between the waypoints. An example would be:

```
"3.4 at the beginning of the animation,  
move smoothly up to 7.6 at time = 10 seconds,  
then jump instantly to 2.0  
and stay there until the end of time"
```

Converted ValueNodes

These are ValueNode which have been *Convert* into sub-parameters, each of which is itself a ValueNode. Right-clicking on a parameter and selecting a type from the 'convert' sub-menu allows you to convert a ValueNode. Converted ValueNode don't have waypoints themselves, but their sub-parameters may do. An example would be:

```
"start at 3.4 and linearly increase by 1.2 per second"
```

3.9.2 Actions on ValueNodes

Convert

Convert a ValueNode into a Linkable type. It disconnects from the previous value node then creates a new Linkable ValueNode and connects the parameter to that new value node.

Disconnect

Creates a Constant value node based on the value of the parameter at the frame where the action is executed. It disconnects from the previous value node and connects the parameter to that new constant value node.

ValueNodes are not discarded completely until all parameters that were connected to get disconnected. So when you disconnect from a value node it doesn't mean that the ValueNode is completely lost. Maybe other parameters still connected to the value node too, so it is not deleted.

Export

Export, takes a ValueNode (animated, constant or linkable) and labels it with a unique name. The exported ValueNode is now known at any place of the document (by any layer) so any other parameter can Connect to it. All exported ValueNodes are stored in the *Library Panel*.

Connect

When an Exported ValueNode is selected in the *Library Panel* it is available to be **Connect** to any parameter, so the parameter is plugged to that exported value node. When you Connect a parameter to an exported value node, the parameter is disconnected from the previous value node. Parameters and Exported value nodes can be Connected only if they are type compatible (i.e. angle with angle and not angle with vector).

Link

When two or more parameters are selected at the same time (usually from two layers) it is possible to *Link* the ValueNode. In this case all except one of the selected previous parameters are disconnected from their value nodes and then connected to the value node selected from the selected. The decision on which value node is kept and used as link value node is taken inside the code and cannot be user decided (but it should be).

TUTORIALS

4.1 Basic Bone Morphing

4.1.1 Example1 (easy understanding)

In this tutorial, we will learn how to move a character with the “Morphing by bones” we arrive at this result:

Fig. 1: A puppet animation.

Preparation

First, we need to create a character in 1 piece, then create a skeleton in a well defined order, to avoid that everything does twist at the same time during the animation.

Once your “puppet” in the Canvas It must be added a skeleton. To add a skeleton layer, New layer → Other → Skeleton

☒		Scale	0,000000
☒		Skeleton	1,000000
☒		puppet	2,000000
☒		background	3,000000

Fig. 2: Place the layer skeleton next to the puppet.

Here is a shema which shows the order in which you must create each bone:

First 1 then use the bone of 1 to create the 2 etc...

To create the 5 bone, use the 2 bone, the arrows represent what “bone” must be used to create the new “bone”. To create the 7 bone, use the bone 1, and 8 use the bone 1.

After you create the 1st bone (the head) click: Switch handles widths

This will show you the widths of handles, it remains only to completely cover your puppet with each “bone” use the “Purple handle” to change the diameter of the width.

The rest you already know the “orange/brown” allows the lengthening of the segment (bone). the “blue” handle this is for orientation (corner handle) from the origin with a determined length, “green” This is the original/axis of rotation (handle position) to move the whole bone.

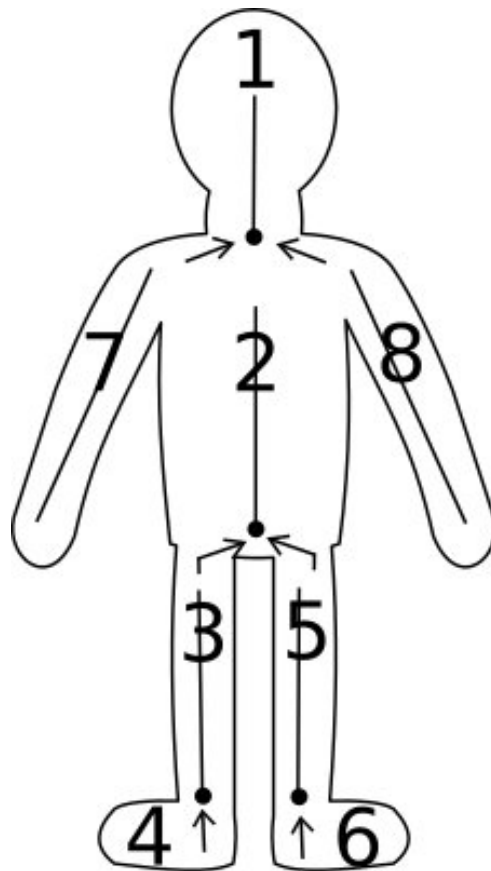


Fig. 3: Schema numbered to place the bones.



Fig. 4: Button: Switch handles widths.

Once your first “bone” covers the entire head and neck, right-click on the skeleton and Create a child bone place him then on your puppet the shema will help you have a correct order (for the animation then).

Once finished and that your puppet is covered with bones, you have this:

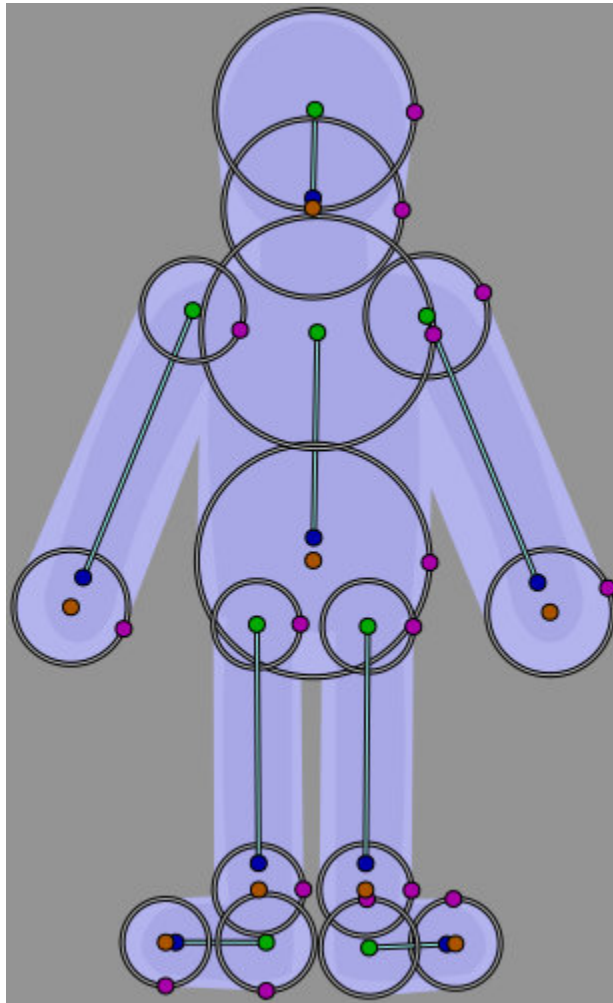


Fig. 5: Puppet covered with bones.

Linking bones to members

Then select your puppet, in the Layers panel and then make a `Ctrl|A` in the canvas to select all vertices.

Then selected the layer `skeleton` We have create earlier, making `Ctrl|Clic`, in the layers panel. In the canvas now right-click on one of the handles of the skeleton and make `Link to the skeleton`.

You just have to create your animation.

Project file

The zip file containing parts of this tutorial may be found here: [Skeleton-puppet](#)

See also

Here's a video using the same technique on a character dispute: [Development: Bones \(part7\)](#)

4.1.2 Example 2 (more detailed understanding)

Now that you know:

- The button “Toggle the width handles.”
- The order of “creation of bones”.
- The Handles.
- Linked, a bone to a limb.

We will do it on a more complex man, the method being simpler because we designate “each limb” in “linking the bone to the skeleton” gradually as (1 to 1), and we verify that the link is made.

Fig. 6: Here is our man once animated.

Let's start by seeing which limbs should be grouped, as in this image:

Then import your limbs, group layers, like this:

Once properly classified, this will go very fast, let's do it again as we saw on the “black man”, but the only difference so, is that we will link to the skeleton each limb and check that it works before moving to the next.

As we already know the method, this will be very simple. Add a skeleton and place it at the top of our group, after positioning it on the head and the hat.

Then click on the group “head” in the “Layers panel” right click and “Select All Layers Kids” we must also select our “hat” here we will make rather a `Ctrl` | click once the layers of “hat and head” to select, let's move on.

`Ctrl` | `A` (to select the vertices) selecting in the “Layers panel” the skeleton. In the “canvas” now right click on one of the skeleton handles and do “Link to the Skeleton”. (You see that the color of the handles changes) this means that the link is created.

To check our link, select only the skeleton, and move the “blue handle”, the head and hat should come with your skeleton.

Well now you have to cancel, make a `Ctrl` | `Z` or “edition > cancel”, (Don't press too many times otherwise your connection to the skeleton will have to be started again).

Now, you just do the same thing to, for all other members 1 by 1.

Let's repeat it one last time for the “neck”.

First we need to extend “bone” so in the canvas we make a right click and “Create a Child Bone” positionner-le pour prendre le cou.

Click on the group “neck” in the “Layers panel” right click and “Select All Layers Kids” `Ctrl` | `A` (to select the vertices) selecting in the “Layers panel” the skeleton. In the “canvas” now make a right-click on one of the handles of the skeleton in the “neck” and made “Link to the Skeleton”. (you can see that the color of the handles changes) this means that the link is created.



Fig. 7: Limbs are separated.

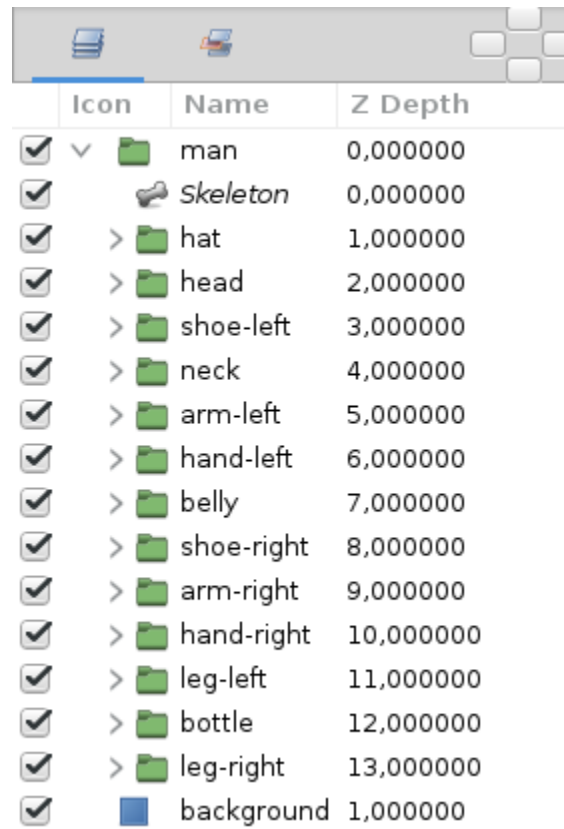


Fig. 8: Arrange layers by group.

We check the link “nake” and also “head”, (if a part of the vertices twists oddly, replace the bone and repeat the binding, if it is really impossible to correct, must then be “Link to the bone”). After checking that it works, we cancel, with the shortcut, or yellow arrow (cancel), green arrow (redo).

Then you just have to do everything else helping you with the scheme of the “black man”.

Once the “bone” add and “Bind to the skeleton or Bind to the bone” you will have something like this:



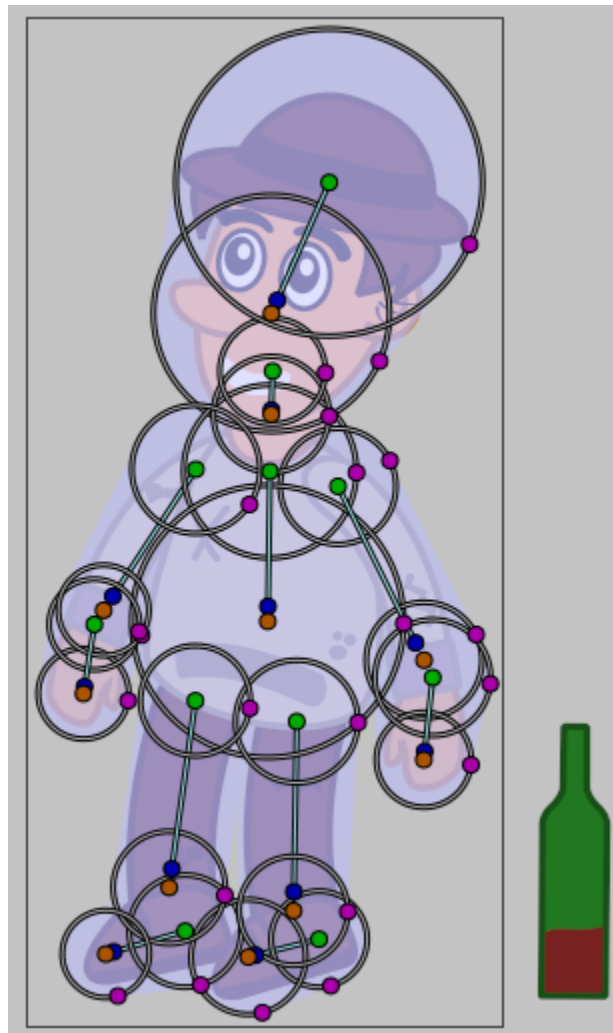
All you have to do is animate your man.

“Linked to the skeleton” OR “linked to the bone”

On the left image, you see a link made on: “Linked to the skeleton”, the limb will twist according to where it will be on your man.

On the right image, you see a link made on: “linked to the bone”, the limb will remain straight as a piece of wood.

A cartoon character, will appear more alive with a few “Links to the skeleton”. So use the one that best fits the end use.



Important

Significant accuracy on the binding of a limb.

Example for a leg that is composed of 3 parts:

Create a bone for the foot, link, check. For the calf, link, check. For the thigh, link, check. (They are all to be done separately).

Project File

The zip file containing the parts of this tutorial can be found here: `Man-vagabon-archive.zipz`

4.1.3 Example 3 - Skeleton for Walking

Here is a schema of “skeleton” that was made to “walk”, this skeleton was built especially for this function.

There are 3 skeletons, that of the left leg, right leg, trunk.

A “hook” is a primary bone that prevents the skeleton from sliding unintentionally during animation.

Here the 3 hooks are located at the position 1,5,9 there are dotted.

To start creating our skeleton it is necessary, create the bone “1”, it will be the hook, then create a child bone, move it to “2”, then continue until 4.

As for the 2nd skeleton, place the “hook” bone “5”, place your bone in the same direction that the limb.

Here the shoe has the same angle that “bone”.

We pass to the skeleton 3, we create the bone “9” which will be the “hook”, then once the bone 10 is placed, we create the child bone 11 until the 13th, we create the child bone 14 from the 10th until the 16th.

And we end up creating the child bone 17 from the 10th.

To summarize the bone 10, is the bone that must be selected to create the child bones 11,14,17.

The advantage of this skeleton is that the feet will not slip “unintentionally” during the animation, this is why it is better to use this type of skeleton if you want to move the lower body.

Project File

The zip file containing the parts of this tutorial can be found here: `Skeleton-to-walk-archive.zip`

4.1.4 Example 4 - Skeleton general (default)

Here is the schema of a general “skeleton” that at the moment is the best if you want to do something other than walk, so there is only 1 skeleton.

The only difference with the “schema of the black man” and that he can move the head and that the center of gravity is in the bone “1” green handle.

The main disadvantage is at the level of the feet, having no hook in this place, the feet will certainly move during the animation.

If you want to do a specific action, then it will be easier to create a skeleton adapted to the desired action.

There is only 1 skeleton, including 1 “hook” it starts to bone “1”, the green position handle, which is surrounded by pink is the handle that is to use to move the whole character.

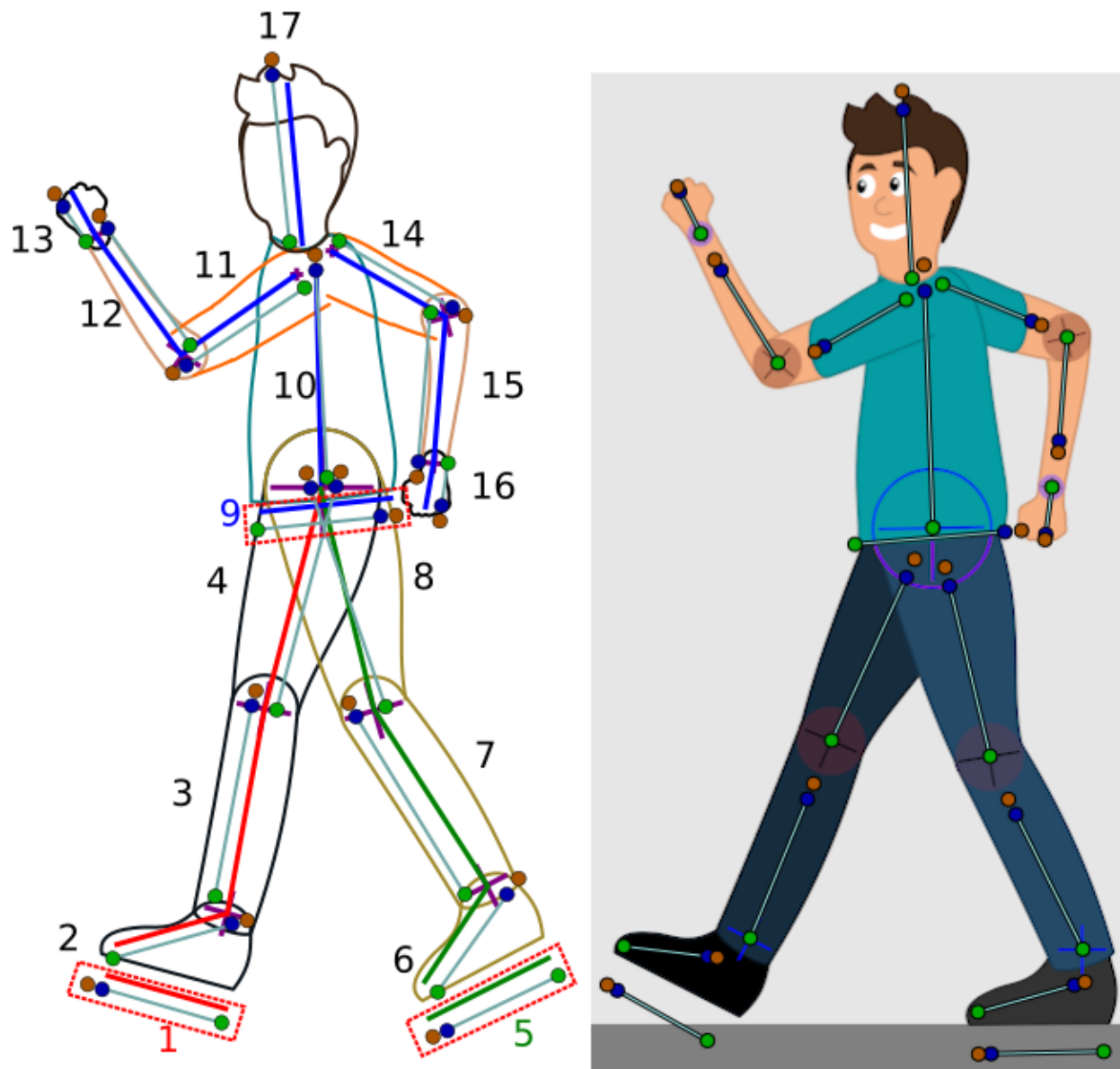


Fig. 9: Schema for walking.

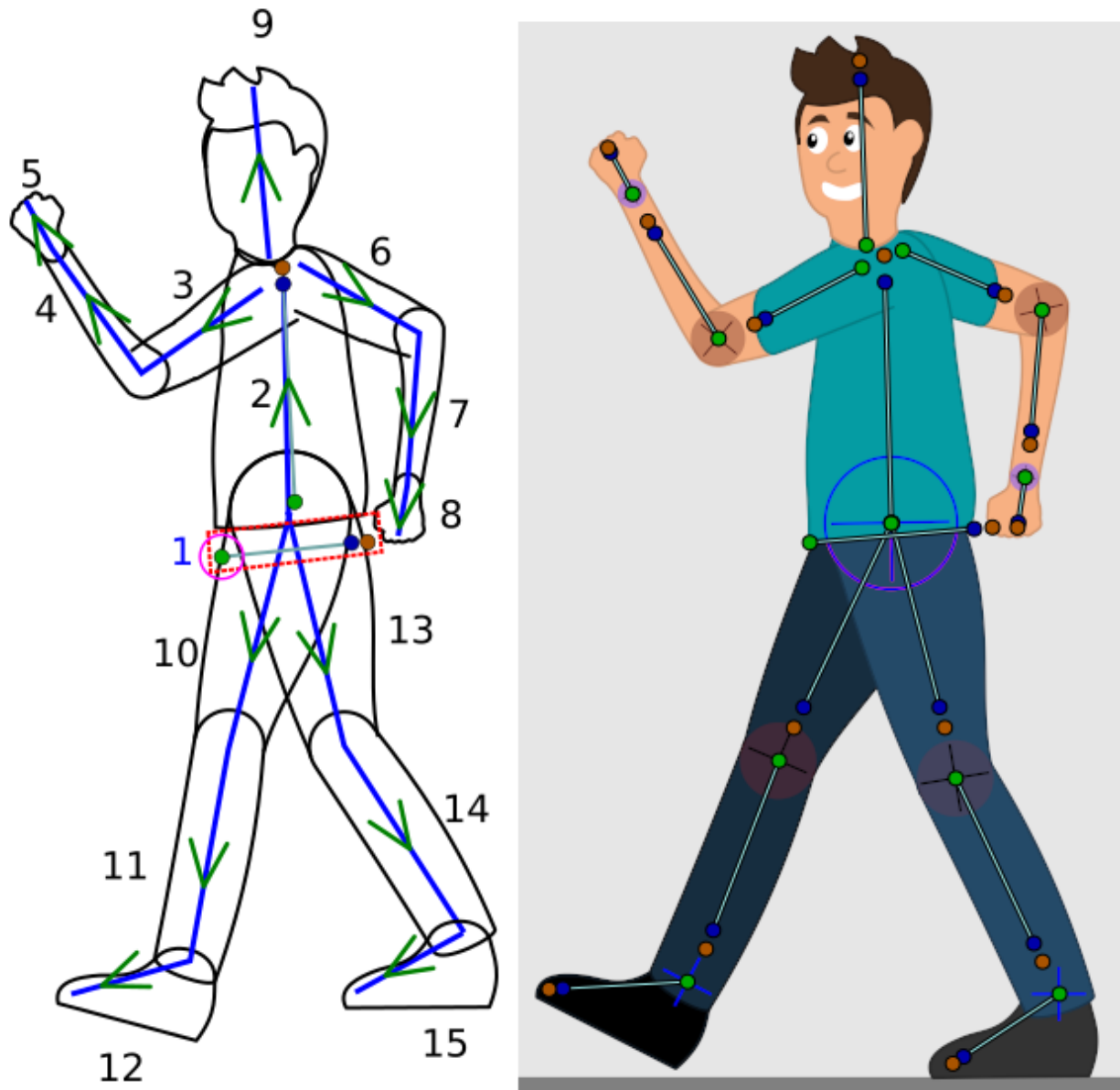


Fig. 10: Schema of a basic skeleton.

Then we create only “bones children”: we must use the schema to create the bones in the right direction.

Project File

The zip file containing the parts of this tutorial can be found here: `Skeleton-general-archive.zip`

4.2 Reuse Animations

4.2.1 Introduction

One of the goals of all animators (especially the lazy ones like me) is to have the opportunity to reuse pieces of animation. It allows you to save a lot of time if you can insert some portions of animations already recorded into any other position in time.

This is especially useful for making characters speak because you have to move your character’s mouth to repeated positions depending on the phonemes it describes while speaking.

This can be done easily just using a combination of *Keyframe* and exported canvases.

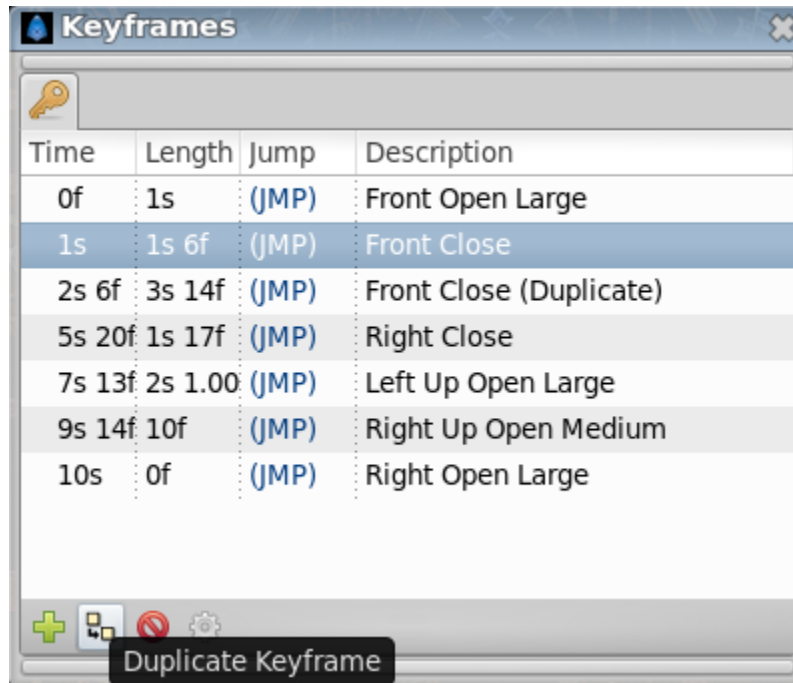
4.2.2 Keyframes

Our goal is to record some sort of animation and reuse it later. This can be done using keyframes. If you create some keyframes at the beginning of your animation you can reuse these “poses” at a later point in time just by duplicating the keyframes at another time position. To do that just do following:

1. Create a Keyframe at a frame (all of our keyframes should be created close to each other to use a small portion of time. We only want to record a “pose” not a transition).
2. Modify your objects in the way you want (for example make an eye close just by moving the points of the eyelid).
3. Give a name to the keyframe just by clicking on its corresponding Description cell.
4. Repeat the above steps as many times you need to make a new “pose”. Let’s say you have created a keyframe at frames number 2 (eye open) and 4 (eye closed).
5. Once done then go to another frame with the *Time Cursor*, select the keyframe you want to introduce and press the “Duplicate keyframe” button. You’ll obtain a copy of the selected keyframe at the current *Time Cursor* position.

There is a problem with this technique. You are making copies of the entire animation poses that you have stored in the first keyframes of the time (frames 2 and 4 of the sample) and therefore you have made copies of all the other objects existing in the scene (following the example, the eyeball).

If you already have an eyeball animation recorded and you overlap an eyeblink (open and closed) set of keyframes in the middle, then the eyeball animation would be broken by the insertion of the copies of the eyelid movement keyframes.



4.2.3 Exporting the Canvas Parameter

Every time you group the layers, you obtain a *Group Layer* that prevents the composition of the contained layers over layers outside of it.

One of the parameters of the Group Layer is the *Canvas*. The canvas is like a workspace that represents all the grouped layers.

To avoid the problem described in the previous section (the keyframes affecting all the objects in the scene) you can do following:

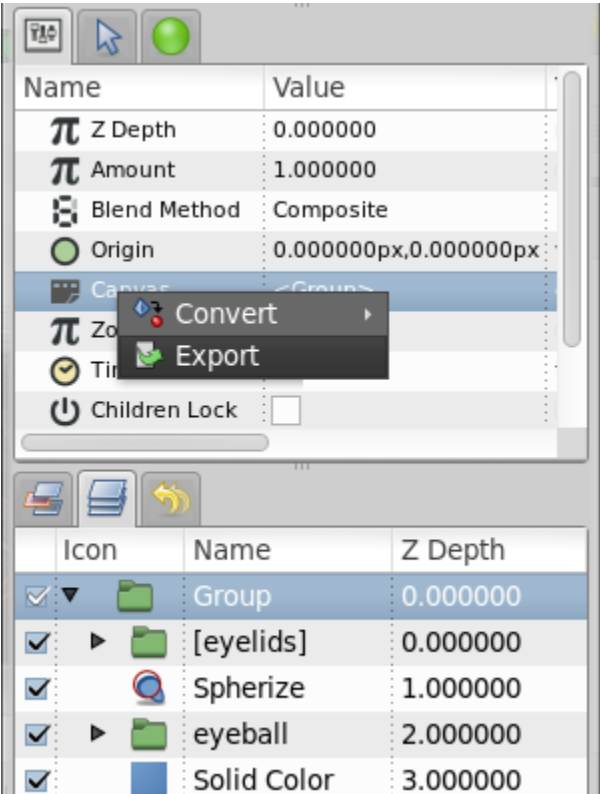
1. Before creating the keyframe poses of the eyelids, group all the layers that form the eyelids.
2. Select the Group layer and select the Canvas parameter in the Parameter Dialog.
3. Right click the Canvas parameter, export it, and give it a name (in the sample this will be “eyelids”).

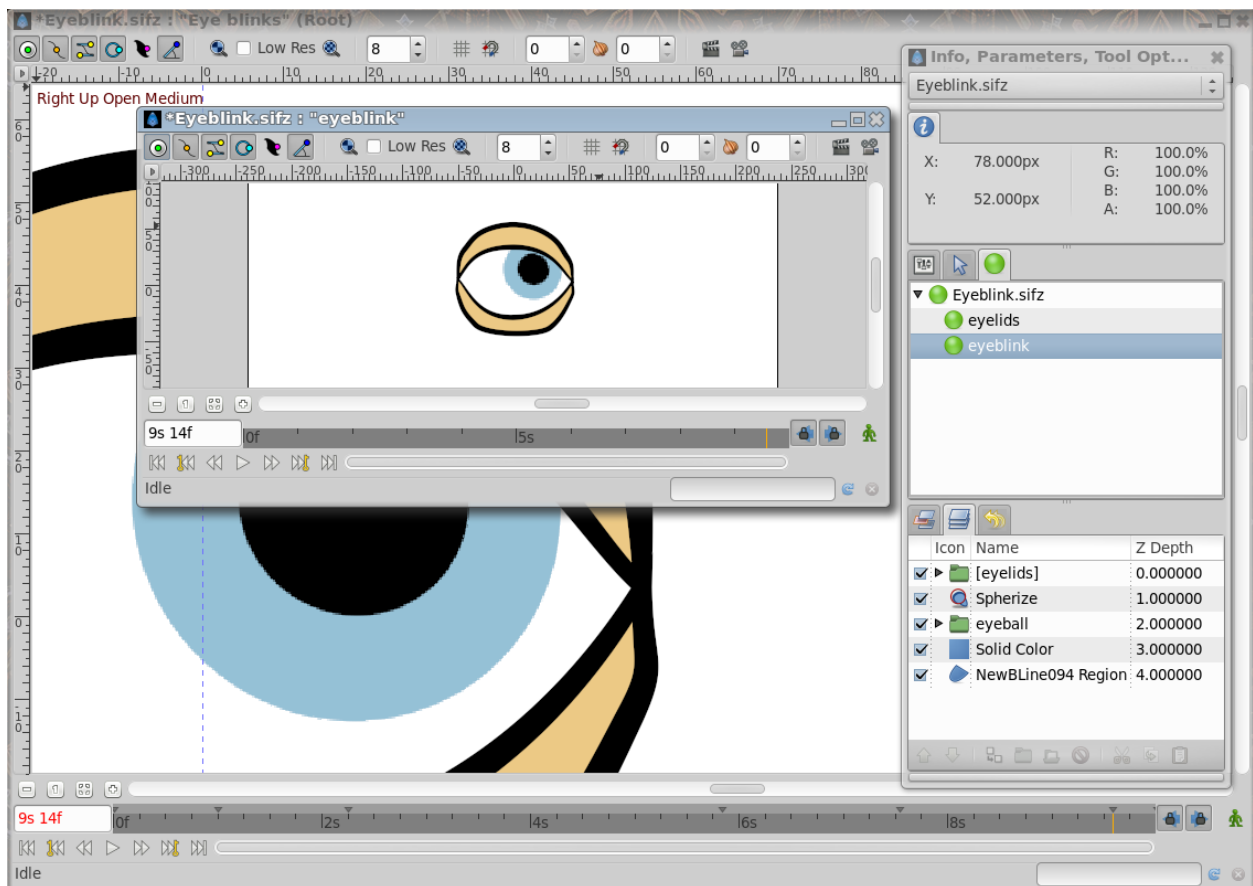
Go to the *Canvas Browser Panel* and select the just exported canvas.

Double click it and a new workarea window will open with just the layers that were grouped at the step 1 - the “eyelids” canvas in the sample.

Create all the keyframes you need to store your “poses”. Once done go to the proper frame and insert a copy of the pose keyframe. It will produce a keyframe in the “eyelids” canvas, but will not produce any keyframe on the other layers (for example the eyeball). This allows to independently animate of a portion of the model separated from the rest. Be sure that the exported canvas is as long as the animation.

Now once you have stored the eye blinks (open and closed) at the desired position you can go to the main window (just close the “eyelids” canvas work area). You’ll see that all the modifications have been transmitted to the main animation but they haven’t created any keyframes in the main work area. Even the layers that are inside the “eyelids” Group layer don’t have any keyframes (you can see an indication that there are keyframes in the exported canvas - dashed vertical lines - but no keyframe is displayed in the keyframe dialog). Anyway, you can see the waypoints created by the keyframes and tweak them, but not the keyframes themselves. To modify the keyframes you should edit the exported canvas again in its own work area. If you modify the grouped layers from the main work area, waypoints will be





created according to the main work area keyframes, not the exported canvas work area, so you will get different effects depending upon which work area you use to modify the grouped layers.

Now, once you have created your animation of the eyelids you can go to the eyeball and modify it to your taste, inserting keyframes or waypoints with no worries about interfering with the eyelid animation. Also you can animate the eyeball before and make the animation of the eyelids later. They won't interfere with each other.

It would be a great improvement if you could connect the time cursors of the main work area and the exported canvas work area to show both windows at the same current time. This would give feedback on where to insert the 'pose' keyframes in your local animation.

4.2.4 A sample

Here you can find a sample animation of a blinking eye (the closed and open positions are copies of the keyframes "Open" and "Closed", while the eyeball moves independently in its own animation.

I have stored the poses "Open" and "Closed" at frames 0 and 2. The animation is defined to start at frame 6.

Sample file

All comments are welcome.

4.3 Snowflake with the Duplicate Layer

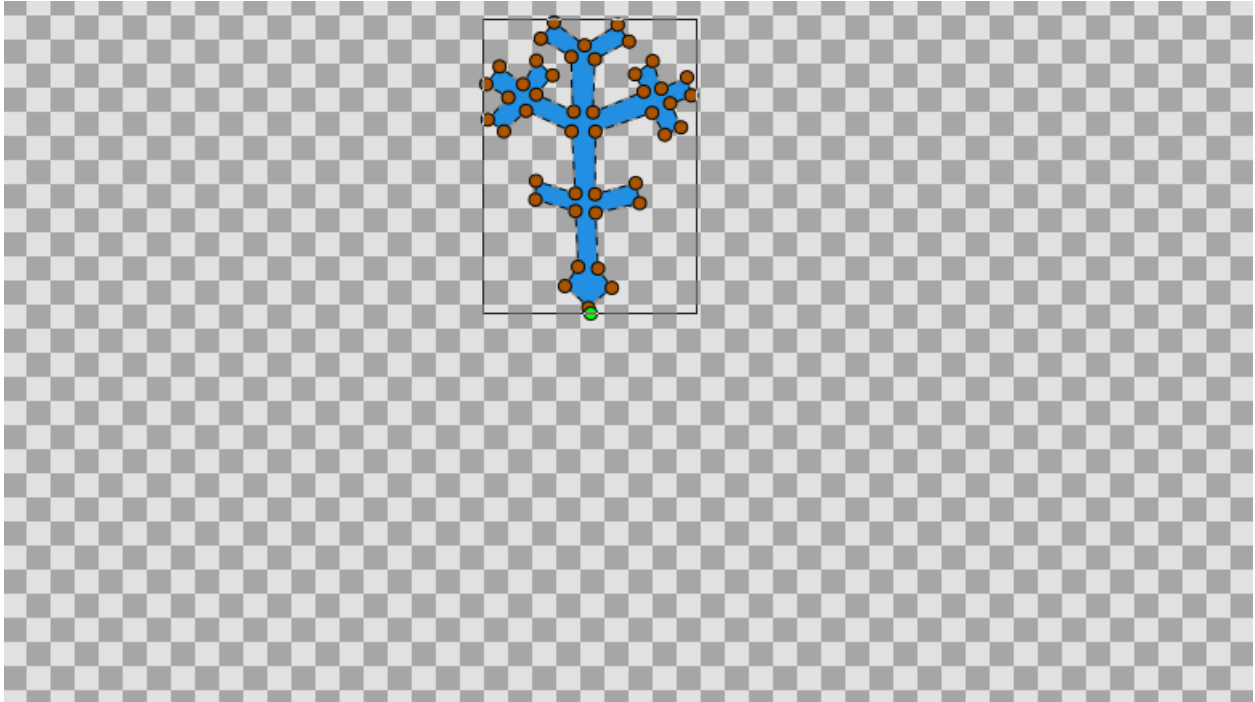
This tutorial explain a simple example of the use of the `|Duplicate_Layer|`, to create a snowflake.

4.3.1 First branch of the snowflake

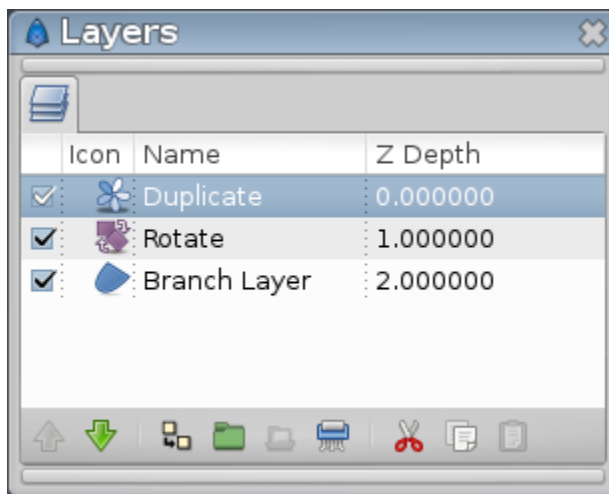
The 1st part is very simple : We'll create a single snowflake branch with the *Spline Tool* (or the *Polygon Tool*).

Select the `Spline` tool, and in the tools options, make sure that only `Create Region` is checked. (If you use the `Polygon` tool, check either `Create Polygon Layer` or `Create Region`, as you wish).

Draw the branch in the upper middle part of your image. Once it's drawn, select the newly created layer (I will refer to it as the "Branch layer") and make sure that the bottom of the branch is more or less at the same place as the green dot showing the center of the image. You can move your branch by selecting all the points `Ctrl | A` and using the *Transform Tool*



4.3.2 Adding the Rotate and Duplicate Layers



To make a whole 6-branches snowflake, we need to make several copies of the branch, and rotate them.

First, we need a *Rotate Layer*. Right click on the image and select |New Layer|Transform|Rotate. You should now have a Rotate Layer on top of the Branch layer.

Then, we'll use the Duplicate Layer. Right click on the image and select |New Layer|Other|Duplicate.

So far, no change is visible on our image, it's normal :).

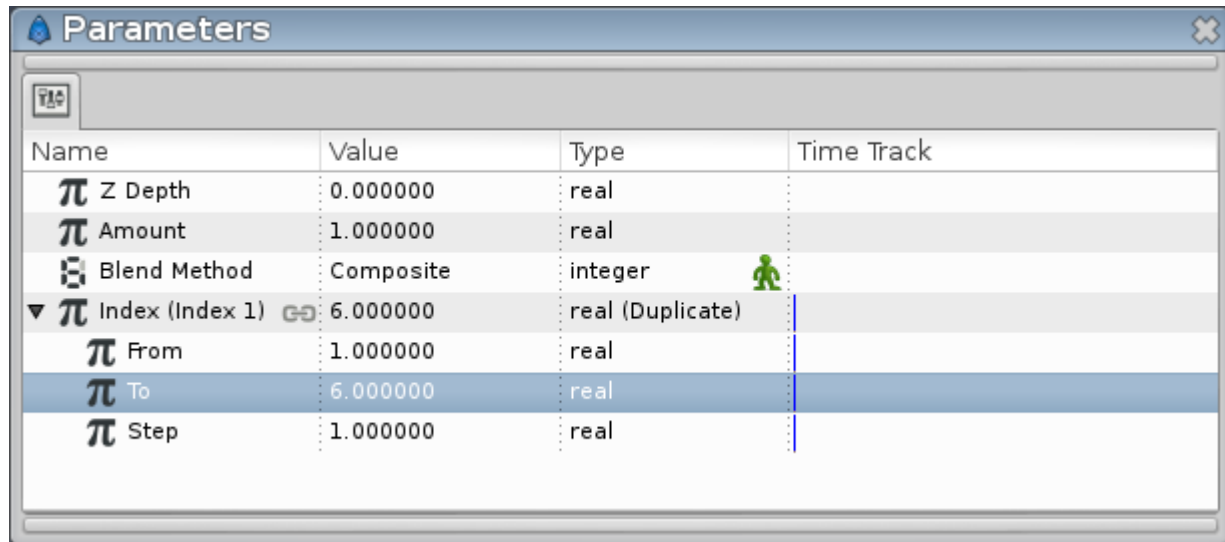
In the *Layers Panel*, make sure that the layers are in the following order : Duplicate on top, Rotate in the middle, and the "Branch" layer on the bottom.

4.3.3 Connecting everything together

Select the Duplicate layer and look at the *Parameters Panel*. There's an Index value that is automatically exported. Its name is Index 1. (It looks like Index (Index 1) in the panel).

This value is made of 3 subparameters : From, To and Step. Those parameters tells how many times an object is duplicated.

We need 6 branches so set the To parameter to 6 (keep From and Step to 1).



Now move to the *Library Panel*, and select our Index 1 value. (Unfold the ValueBase Node if necessary).

In order to have 6 rotated branches, we need to link the duplication and the rotation together.

Select the Rotate layer, and go back to the Parameters panel.

The Rotate layer has just 2 parameters: Origin and Amount. The important one here is Amount, it tells the angle of the rotation.

Right click on Amount and select <Convert> → Scale.

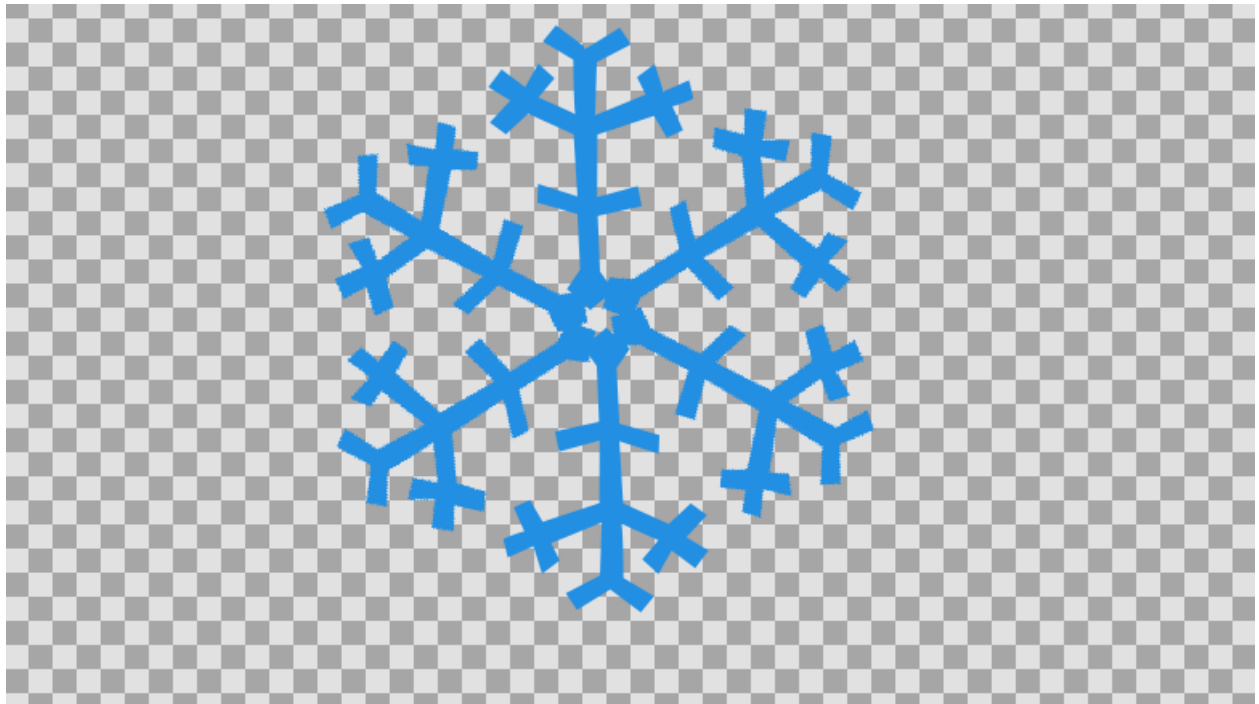
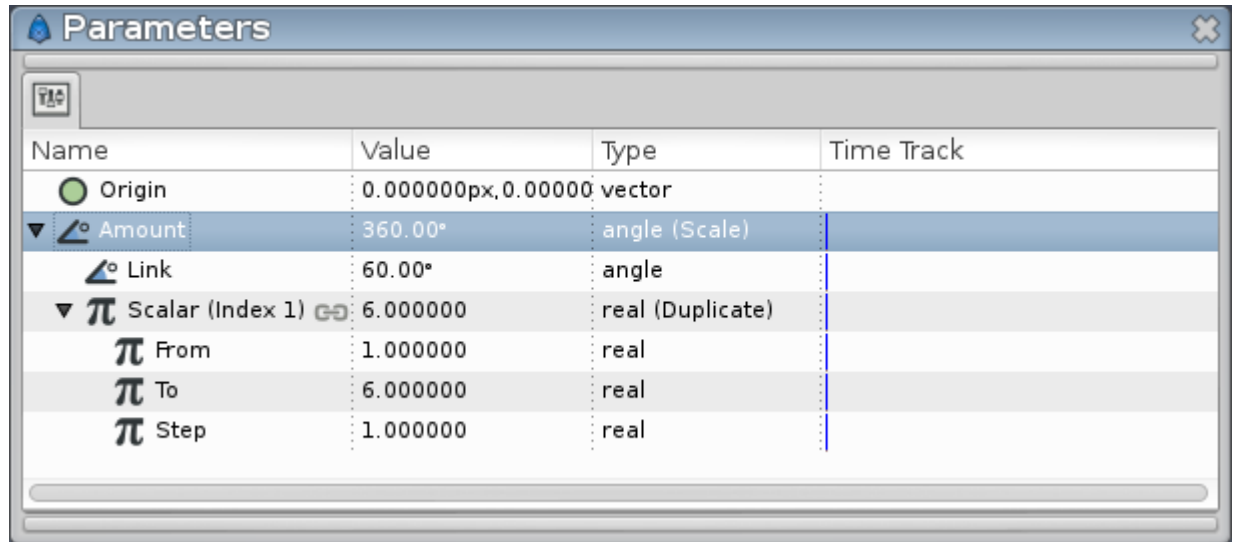
Now Amount is made of 2 subparameters, Link (an angle) and Scalar (a real), the whole rotation being equal to Link * Scalar.

Set the Link subparameter to 60 (that's the angle between the snowflake branches).

Right click on the Scalar parameter and select Connect. It will connect the previously selected Index 1 value (in the Library panel) to the Scalar parameter.

How it looks after the connection:

You should now have a complete snowflake, yay! (maybe you will need to adjust Rotate Layer's *Origin Parameter*)



4.3.4 What happened?

We linked the “Scalar” value of the rotation to the duplicate layer:

Thanks to the Duplicate, the Scalar parameter was set to values between 1 and 6 (From and To), by steps of 1 - instead of being set to a single value.

So we had 5 rotations in addition to the original branche : One of 60° (Scalar * Link = $1 * 60^\circ$), another of 120° ($2 * 60^\circ$), another of 180° , etc. And all of these rotations were displayed together.

You can think of the Duplicate Layer as something allowing you to have a list of values, that you can link to a (real) parameter in the same way you could link a single value to a parameter.

You can download the project

4.4 Subtracting Shapes

4.4.1 Introduction

In this tutorial we are going to draw a shape and subtract another shape from it. We assume that you know how to draw a circle using the *Circle Tool* and a rectangle using the *Rectangle Tool*. If not then check *Circle Tool* and *Rectangle Tool*.

4.4.2 Drawing the shapes

Click the *Circle Tool* on the *Toolbox* and draw a circle on the canvas with an outline of 3 pixels. Make sure that the create region layer and the create outline layer in the *Tool Options Panel* of the *Circle Tool* are selected.

Click the *Rectangle Tool* on the *Toolbox* and draw a rectangle on the canvas with an outline of 3 pixels. Again make sure that the create region layer and the create outline layer in the *Tool Options Panel* of the *Rectangle Tool* are selected.

For the sake of this example name the circle outline layer circleOutline and the circle region layer circle in the *Layers Panel*. Again in the *Layers Panel* rename the rectangle region layer baseShape and name the rectangle outline layer baseShapeoutline. If it is not already the case then select the circleOutline layer and the circle layer and move them to the top using the green arrows in the *Layers Panel*. Select all layers in the *Layers Panel*, right-click and choose group. Name the group baseShape.

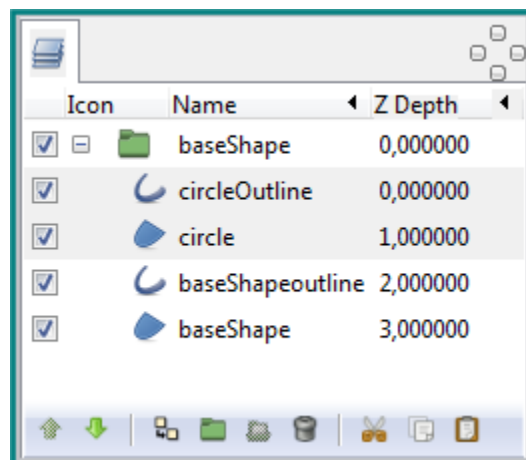


Fig. 11: Layers Panel should look like this

4.4.3 Subtracting the shape

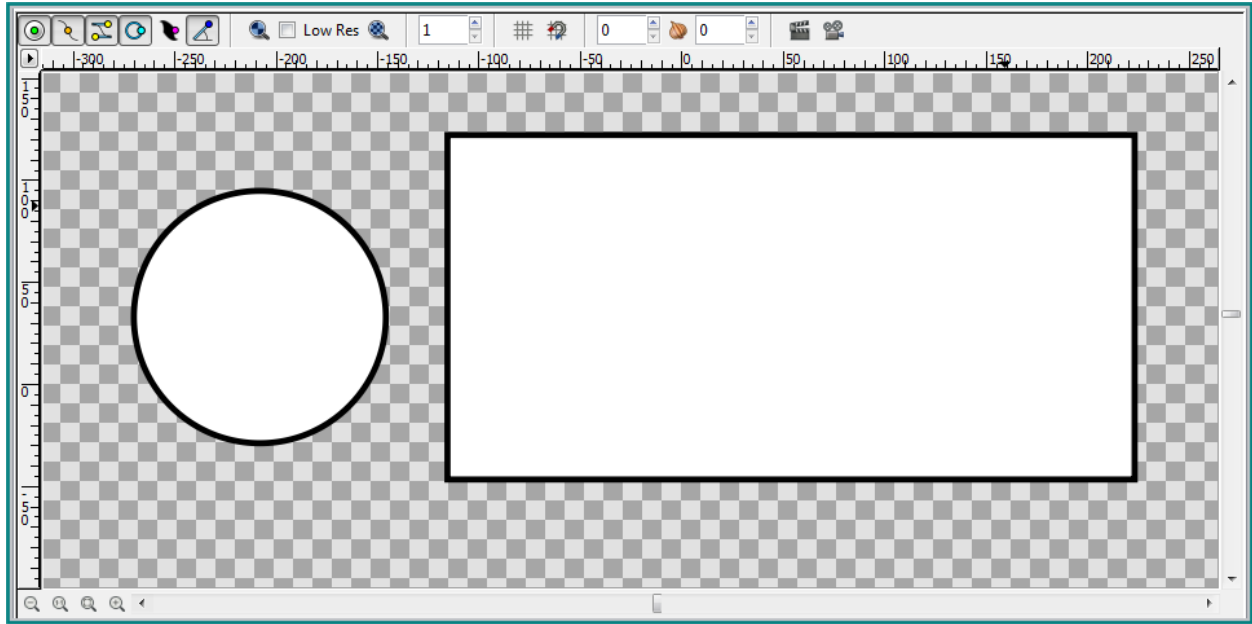


Fig. 12: Canvas should look like this

Click on the circle layer in the *Layers Panel* or on the circle shape in the canvas and select the green handle in the middle of the circle and drag it over the baseShape. Release the mouse.

Select the circleOutline layer in the *Layers Panel* go to the *Parameters Panel* and change the *Blend Method* of the circleOutline layer to *Blend Method: Onto* Select the circle layer in the *Layers Panel*, go to the *Parameters Panel* and change the *Blend Method* of the circle layer to *Blend Method: Alpha Over* .

The circle layer was subtracted from the baseShape but the *Outline Width* of the cut out part varies from the *Outline Width* of the baseShape. To adjust this we have to change the *Outline Width* of the circleOutline layer. Select the circleOutline layer in the *Layers Panel*, go to the *Parameters Panel* and change the *Outline Width* to 6 pixels. After we adjusted the *Outline Width* of the circleOutline layer the circle layer was subtracted correctly from the baseShape.

4.4.4 Tip

If we want to add a circle to the baseShape then all we have to do is change the *Blend Method* of the circleOutline layer from *Blend Method: Onto* to *Blend Method: Alpha Over* and change the *Blend Method* of the circle layer from *Blend Method: Alpha Over* to *Converter: Composite*.

Below we can see an image of a piece of cheese that is almost entirely constructed by subtracting and adding circles to a basic underlying shape.

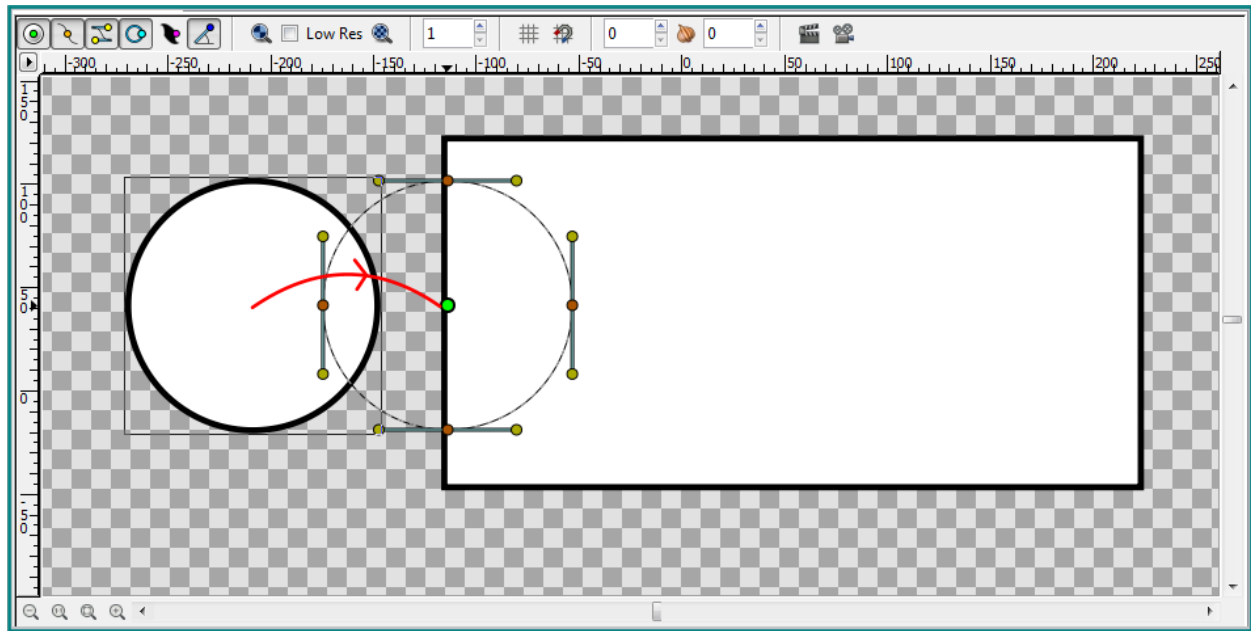


Fig. 13: Drag the circle

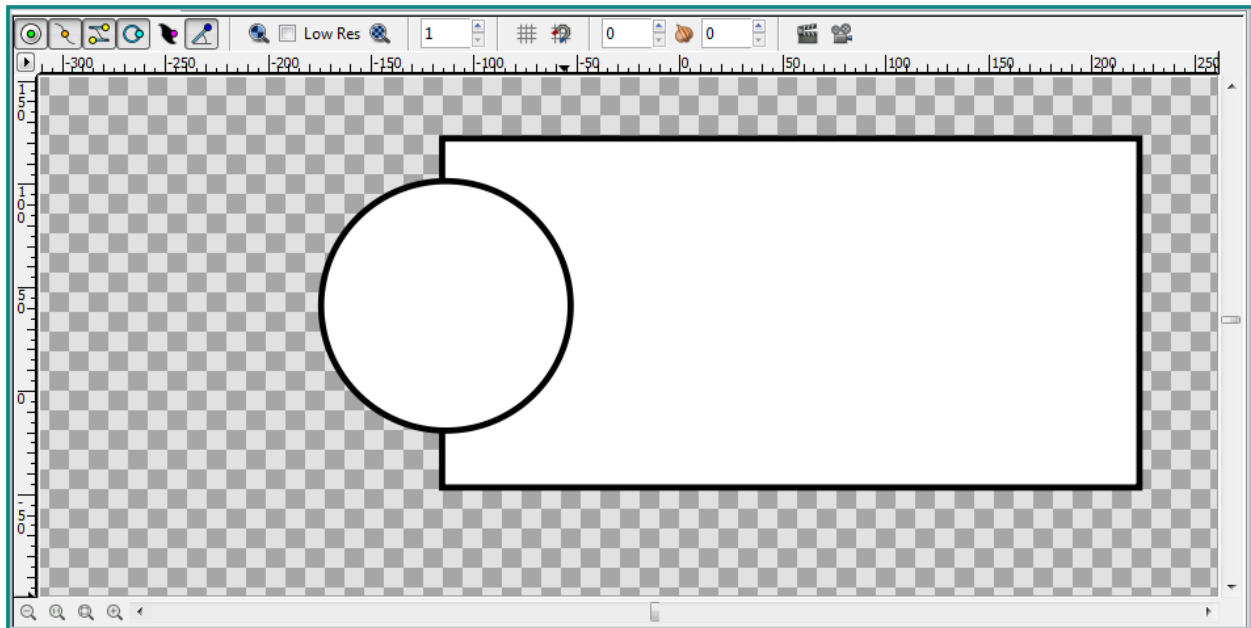


Fig. 14: Canvas should look like this

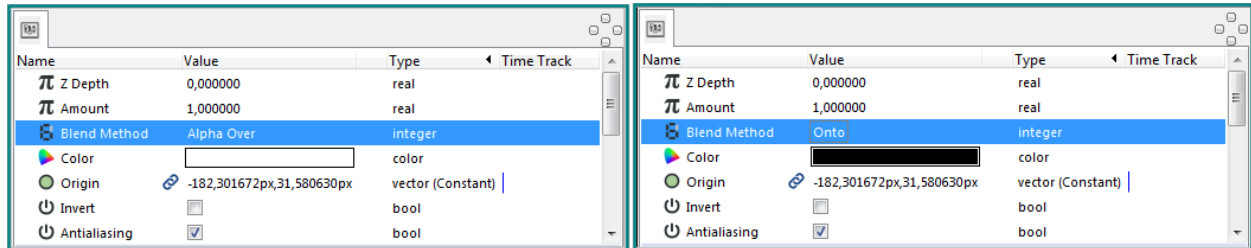


Fig. 15: Parameters Panel with blend method of Circle layer and CircleOutline layer

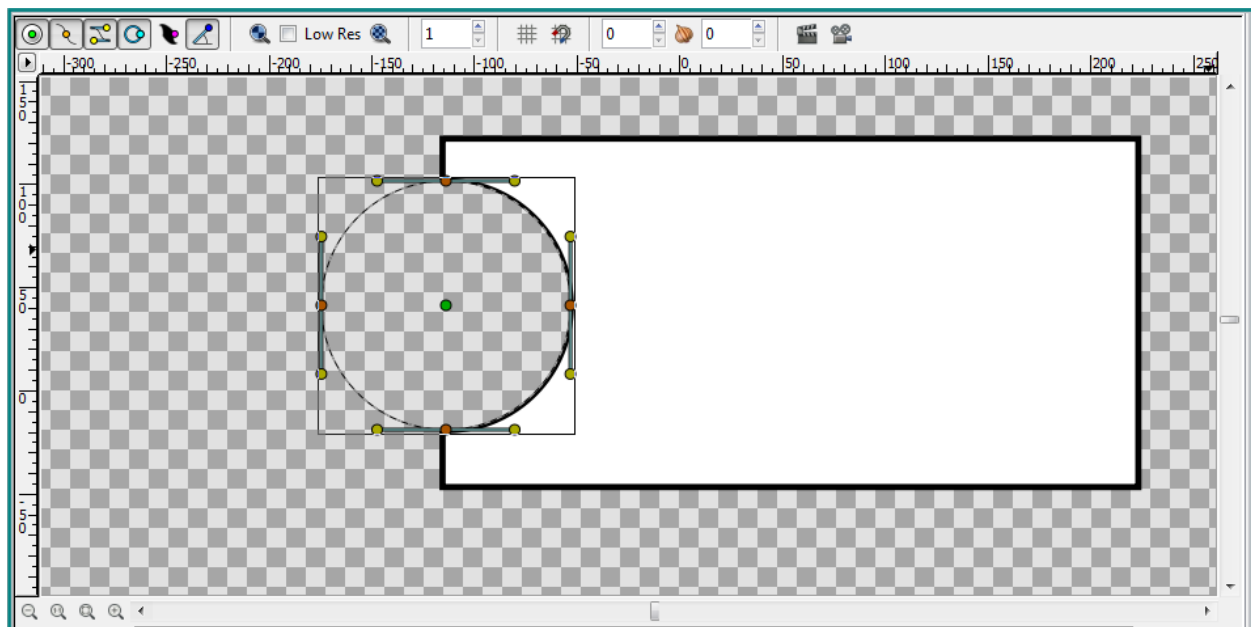


Fig. 16: Outline Width is not the same

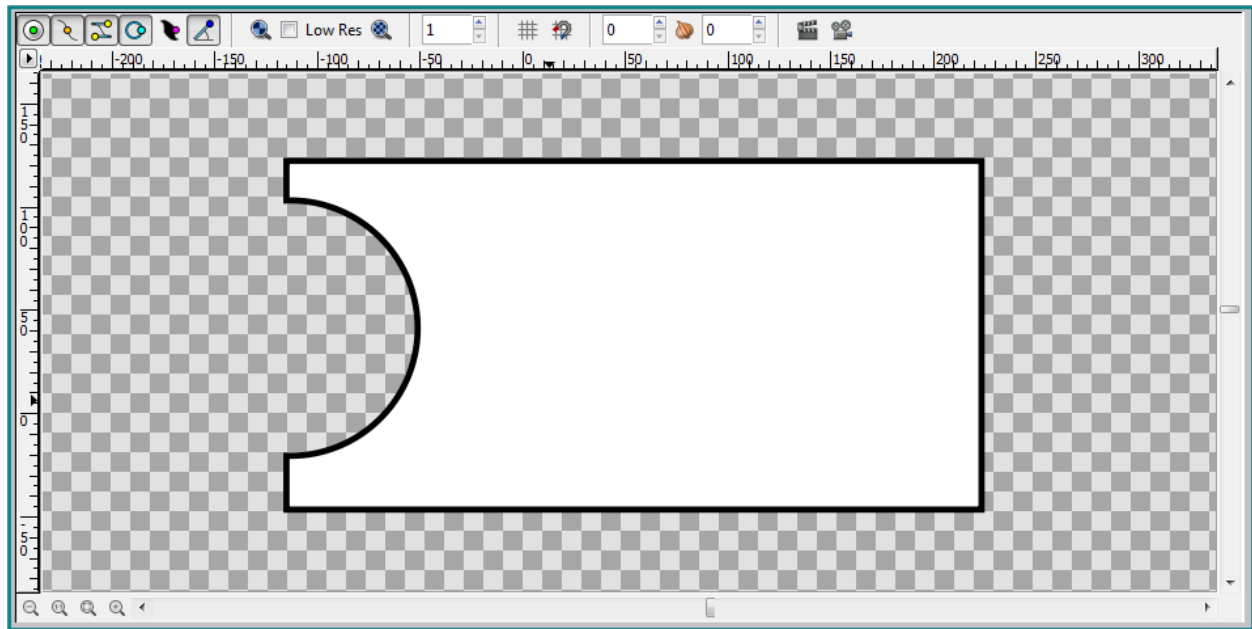


Fig. 17: Final image

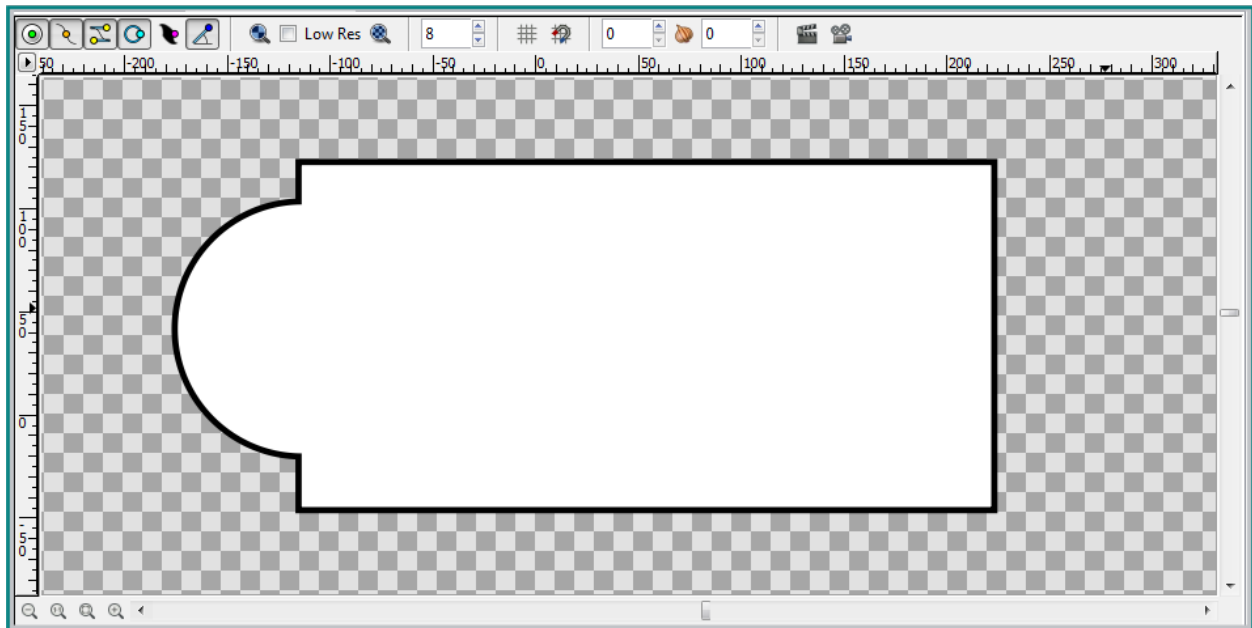


Fig. 18: Circle added to the baseShape

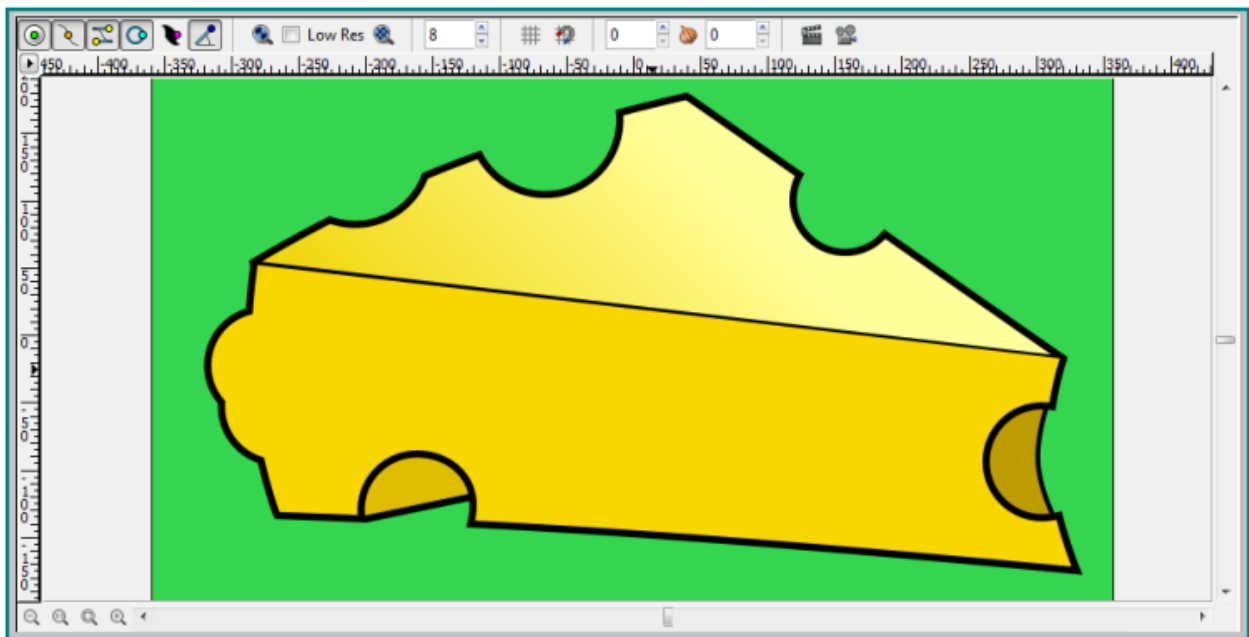


Fig. 19: All holes in the cheese image are subtracted circles

IMPORTING YOUR ARTWORK

5.1 Importing images

5.2 Importing image sequences

5.2.1 General informations

The list importer allows you to import a text file containing a list of still images and have it be treated as a single animated layer.

To use it, simply create a text file with the filename of each still frame listed per-line. Save this text file with the extension `lst` (That is LST in lowercase). Import it into Synfig like you would any other image.

Warning: Important for `lst` file creation

The list importer is quite sensitive and may fail to import if the `.lst` file contains extra space characters or non unix line terminator.

See [this forum thread](#) for more info.

The default framerate is 15 frames per second. If you would like to use a different frame-rate, add a line to the file like this:

`FPS 24`

Now the frame-rate will be 24 frames per second.

The list importer allows also import [Papagayo](#) output lipsync voice files.

As well as Papagayo only export the phonemes it is assumed by default that the image file is a jpeg image with “`jpg`” extension. This means that the standard phonemes image files must be called like this:

AI.jpg
E.jpg
etc.jpg
FV.jpg
L.jpg
MBP.jpg
O.jpg
rest.jpg

U.jpg
WQ.jpg

If you want to change the type of image file just insert a valid extension before the phonemes. It should be done manually as well as the FPS thing.

This is how a valid “1st” file from papagayo looks after inserting the FPS and the change to other image file type:

MohoSwitch1
FPS 24
png
4 O
7 L
10 AI
13 rest
22 rest
23 MBP
26 E
29 etc
31 AI
33 MBP
35 O
37 etc
39 E
41 etc
43 E
45 etc
47 E
49 rest

It says: “*Hola, me llamo Genete*”

In this case it has been changed to png.

Allowed image types are:

- jpg (default)
- png,
- ppm
- tiff
- gif
- SVG (since 1.5.2)

5.2.2 Useful scripts

- `Convert-movie-to-png-seq.sh`: Nautilus (Gnome file manager) script which creates a png-sequence from a given movie file and creates a list-file which you can then import into Synfig. By Rylleman, [Synfig forum thread](#) for more info,
- [Listail - List File Generator](#) : Listail create a list of images to 'lst' file, consisting of a frame rate followed by a list of images to display. [listail](#) by Anderson Prado (AndeOn).

Vectorization of Bitmap Vectorization of Bitmaps:

5.3 Vectorization of Bitmaps

Note: Available since Synfig Studio 1.4.0.

Synfig Studio can convert raster (bitmap) images into vector artwork constructed from outlines. The vectorization algorithm is ported from [OpenToonz](#) animation software by Ankit Kumar Dwivedi.

Vectorization of Bitmap Video Tutorial:

5.3.1 Video Tutorial

Vectorization of Bitmap What does Vectorization mean?:

5.3.2 What does Vectorization mean?

There are two common ways to represent a two-dimensional image:

- Raster (or Bitmap) Graphics
- Vector Graphics

In Bitmap Graphics image is stored as a two-dimensional grid of pixels. This is great for storing images of any complexity (i.e. photos), but transformation (i.e. scaling) of bitmap image usually results in quality loss - it becomes blurry and pixelated.

In Vector Graphics image is constructed from elements which described by mathematical formulas. As result, it is possible to transform such images without quality loss.

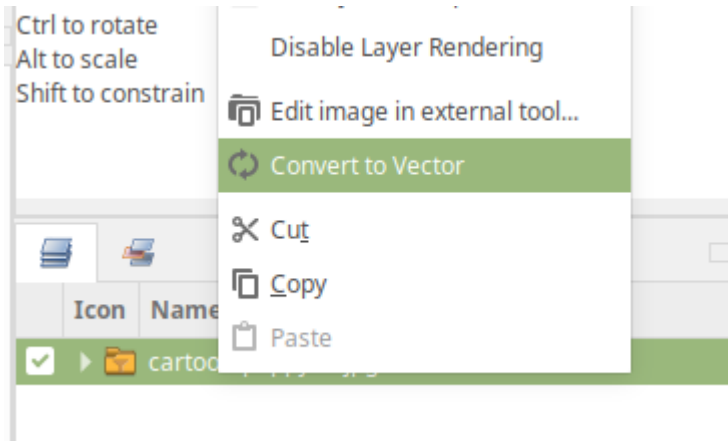
For more information about Raster and Vector graphics you can refer to [this page](#).

It is very common situation when user have an input artwork as raster image (i.e. scanned sketch) and wants to convert it into vector graphics. The process of conversion from raster to vector is called vectorization.

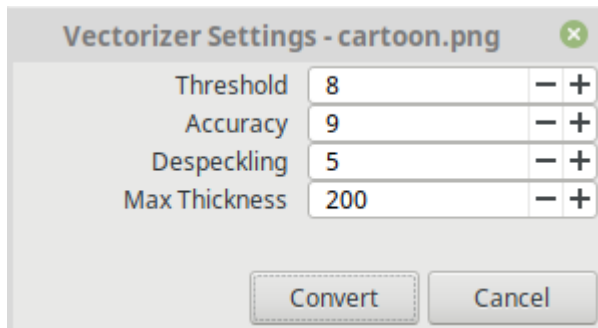
Vectorization of Bitmap Usage:

5.3.3 Usage

Right-click any Image Layer in Layers Panel and select “Convert to Vector” command.



You will see a dialog box with the following settings:



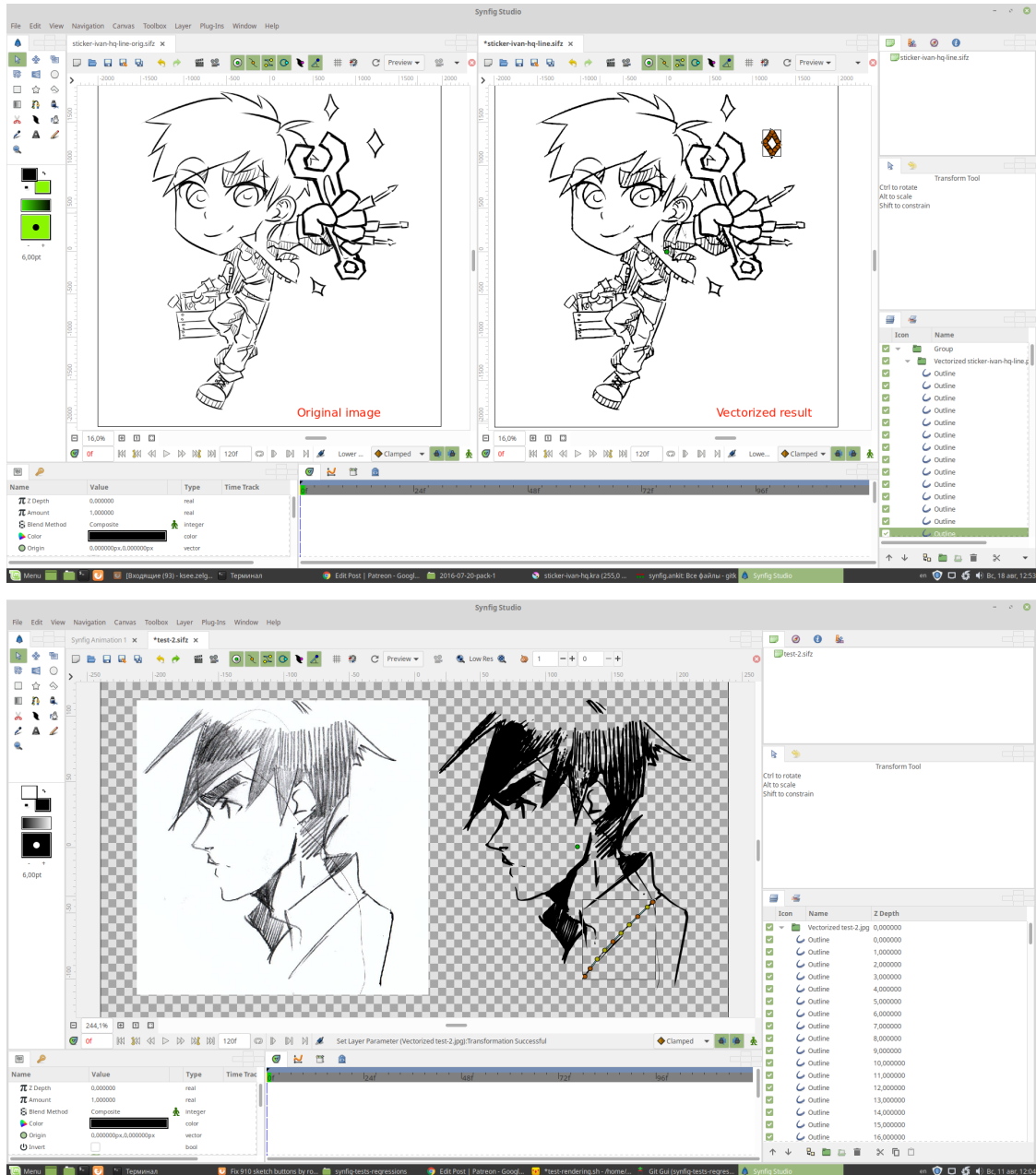
- **Threshold** - sets the value of the darkest pixels to be taken into account to detect lines to be converted to vector.
- **Accuracy** - sets how much the vector stroke will follow the shape of the original drawing lines. High values create more precise strokes but makes them more complex.
- **Despeckling** - ignores during the conversion small areas generated by the image noise; the higher the value, the larger the areas ignored.
- **Max thickness** - sets the maximum vector stroke thickness; if this value is low very thick lines will be converted in two centerline strokes defining the line outline; if this value is high, they will be converted in a single centerline stroke.

Click “Convert” button and the image will be converted into a vector outlines (you will see a new Group Layer created).

Note: Synfig currently supports only centerline vectorization

Vectorization of Bitmap Examples:

5.3.4 Examples



5.4 Importing SVG

5.5 Importing videos

5.6 Swf2svg

Convert **SWF** in **SVG** and can edit in inkscape with [|svg2sif|](#)

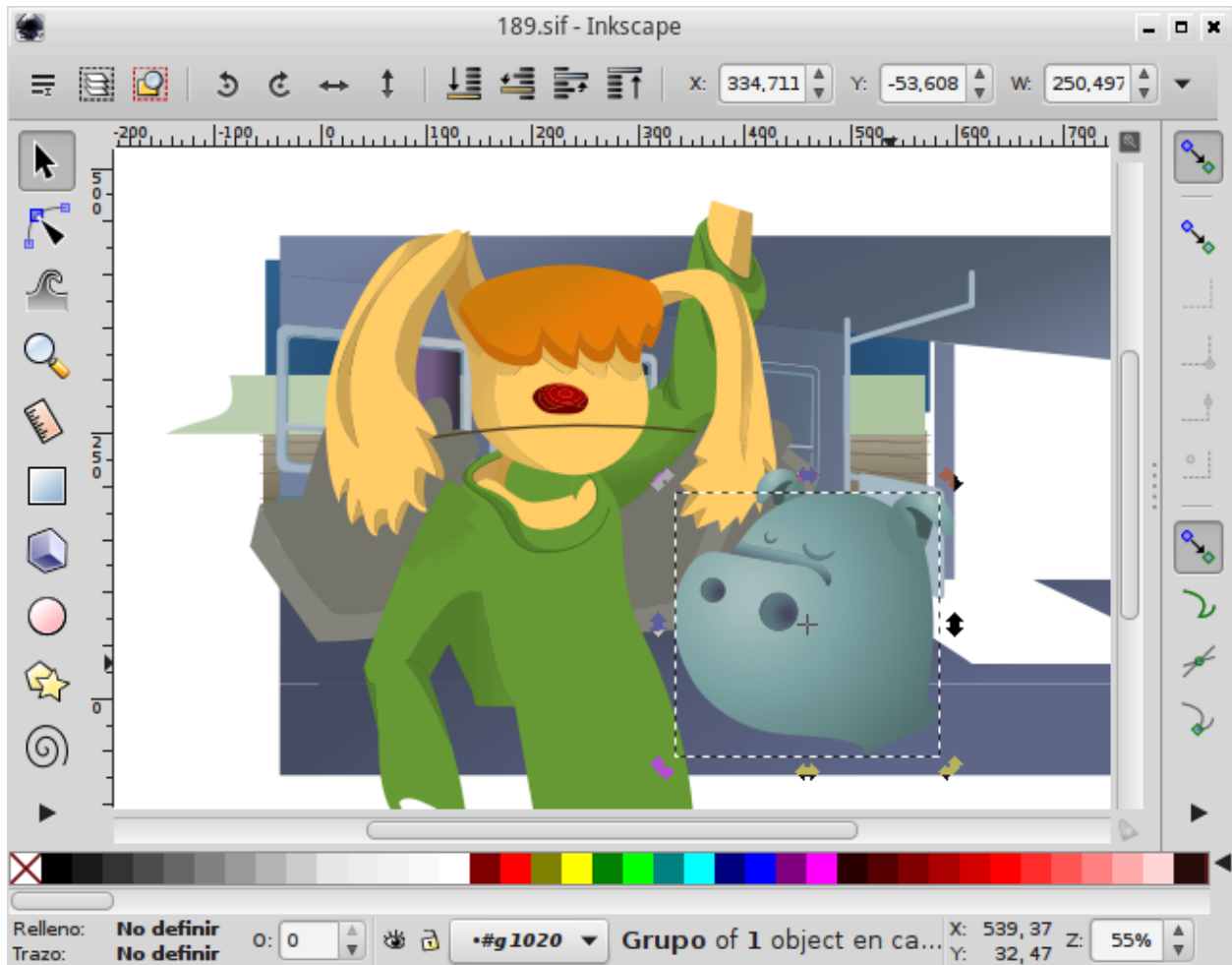
5.6.1 Example

Animation reproduced in gnash:



Mattered in Inkscape afterwards to export it with **SWFTTools**

Yes it degrade them export in first black export in PDF, matter the PDF and afterwards go back to export in SIF.



5.6.2 License

Author: Michael Fuller

License MIT

6.1 Brush Tool

Note: This feature is still in development.



ALT-?

6.1.1 Introduction

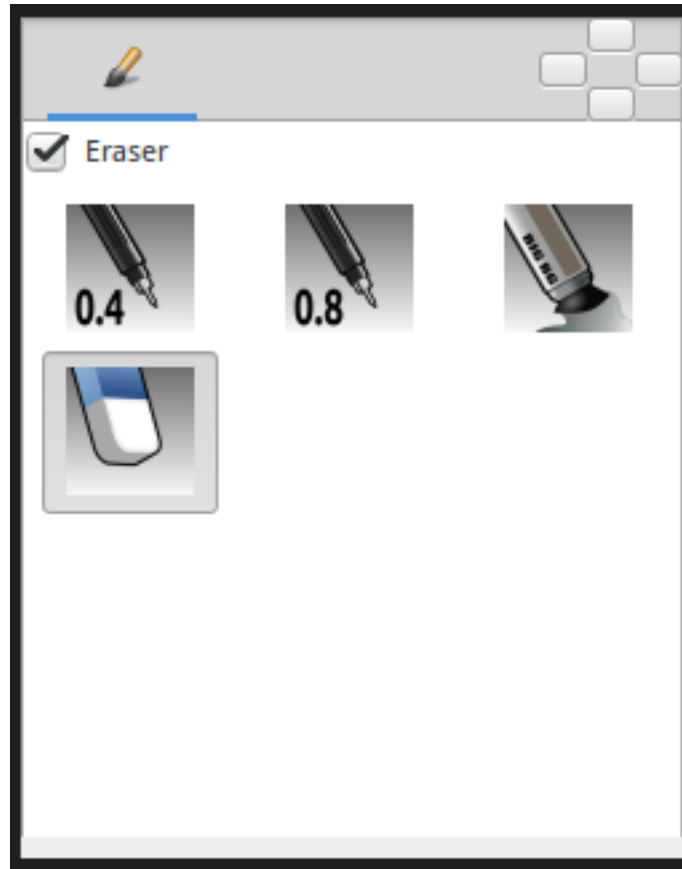
The Brush Tool allows you to draw freehand over the top of a *Image Layer* using the MyPaint brushes. Combined with *Switch Group Layer* you can create frame by frame animations.

- It is possible to paint on the *Image Layer*
- Basic pressure sensitivity is supported.
- Image size is automatically expanded when you paint outside of image boundaries.
- An *Image Layer* is automatically created if the current layer isn't an *Image Layer*.

6.1.2 Options

- Eraser check box. When checked, the brush acts as a rubber.
- Display the set of configured brushes (MyPaint brushes format). By default, Synfig is installed with a selected set of brushes from “Concept Design (C_D)” pack by Ramon Miranda. From the *Preferences Dialog System*, you can configure the Brush Presets Path to use other brushes package (for example the ones listed by MyPaint)

The Brush Tool uses the color of the *New Layer Defaults: Brush Color* and the *New Layer Defaults: Brush Size* set in the *Toolbox* to paint.



6.1.3 Other sources of information

Here you can find a video about brush usage and frame by frame animation

Development: [Frame-by-frame animation \(Part 4\)](#)

6.2 Circle Tool

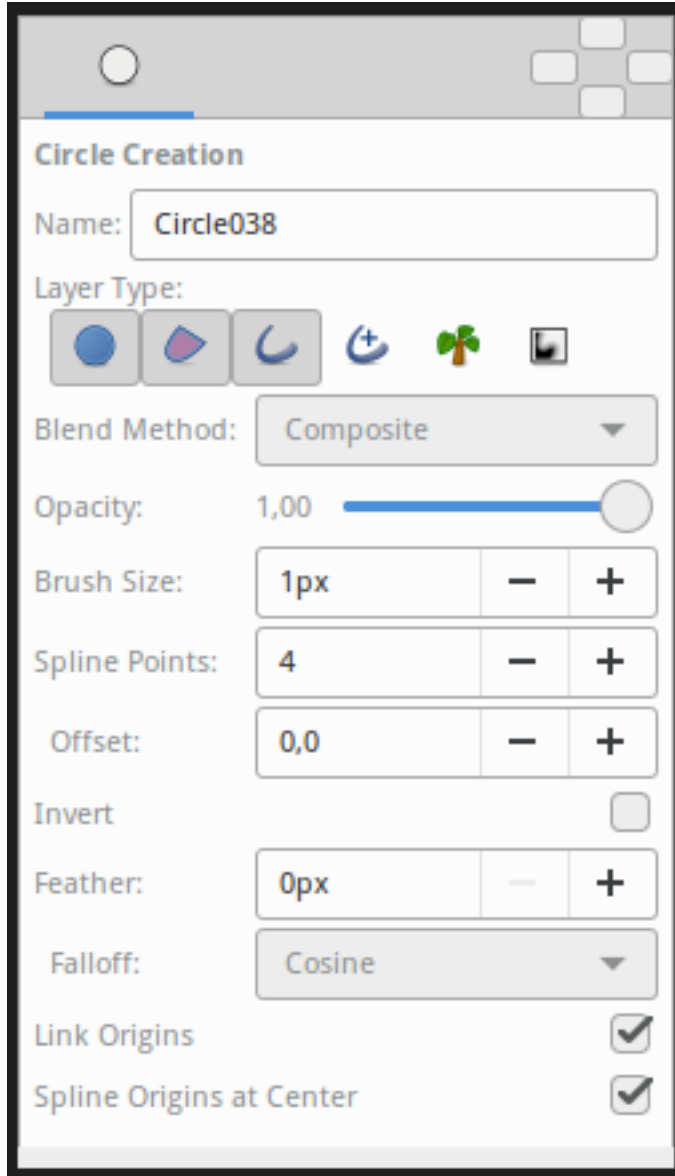


6.2.1 Introduction

The **Circle Tool** is used for creating new *Circle Layer*. Click where you want the *Circle Layer: Origin* of the circle to be, and drag to set the *Circle Layer: Radius*.

The **Circle tool** can also be used to create *Outline Layer*, *Region Layer*, *Curve Gradient Layer*, and *Plant Layer*, each with approximately circular *Spline*.

6.2.2 Options



The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created
- **Layer Type** : Whether create *Star Layer* and/or *Outline Layer*, *Advanced Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines)
- *Blend Method Parameter* : The blending method used to composite on the layers below
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque)
- **Brush Size** : Sets the *Outline Layer: Outline Width* or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*)
- **Spline Points** : Specify the number of Spline Points that will be used in the new Splines (for other than *Circle Layer*)
- **Offset** : Offset of the tangent of the Spline Points (for other than *Circle Layer*)

- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert*.
- *Feather Parameter* : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*)
- **Falloff** : Determines the falloff function for the feather (only for *Circle Layer*)
- **Link Origins** : Whether Link the Offset of Splines
- **Spline Origins at Center** : Set the origin of the layer at the center of the circle, else the center is set at the *Canvas* center

6.2.3 Usage

Click on the canvas where you want to have the center of the circle and drag to obtain the desired radius value, and release the mouse button when you are done.

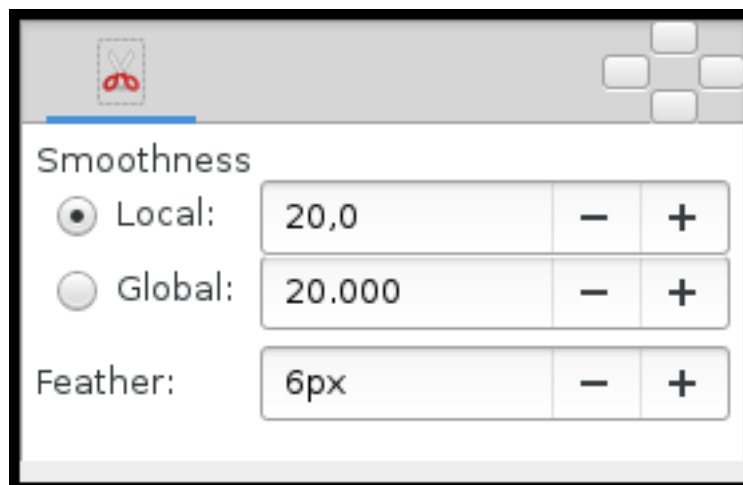
6.3 Cutout Tool



ALT-?

The Cutout Tool is designed to select a piece of the composition by adding a mask.

6.3.1 Options



- **Smoothness** : Adjust the Spline defining the Mask. Could be Local or Global.
 - **Local** : Adjust the Spline between the Splines points.
 - **Global** : Adjust the Spline Globally.
- *Feather Parameter* : Width of the area that is going to be dissolved at the edge of the mask.

6.3.2 Video howto

- [Cutout Tool](#)

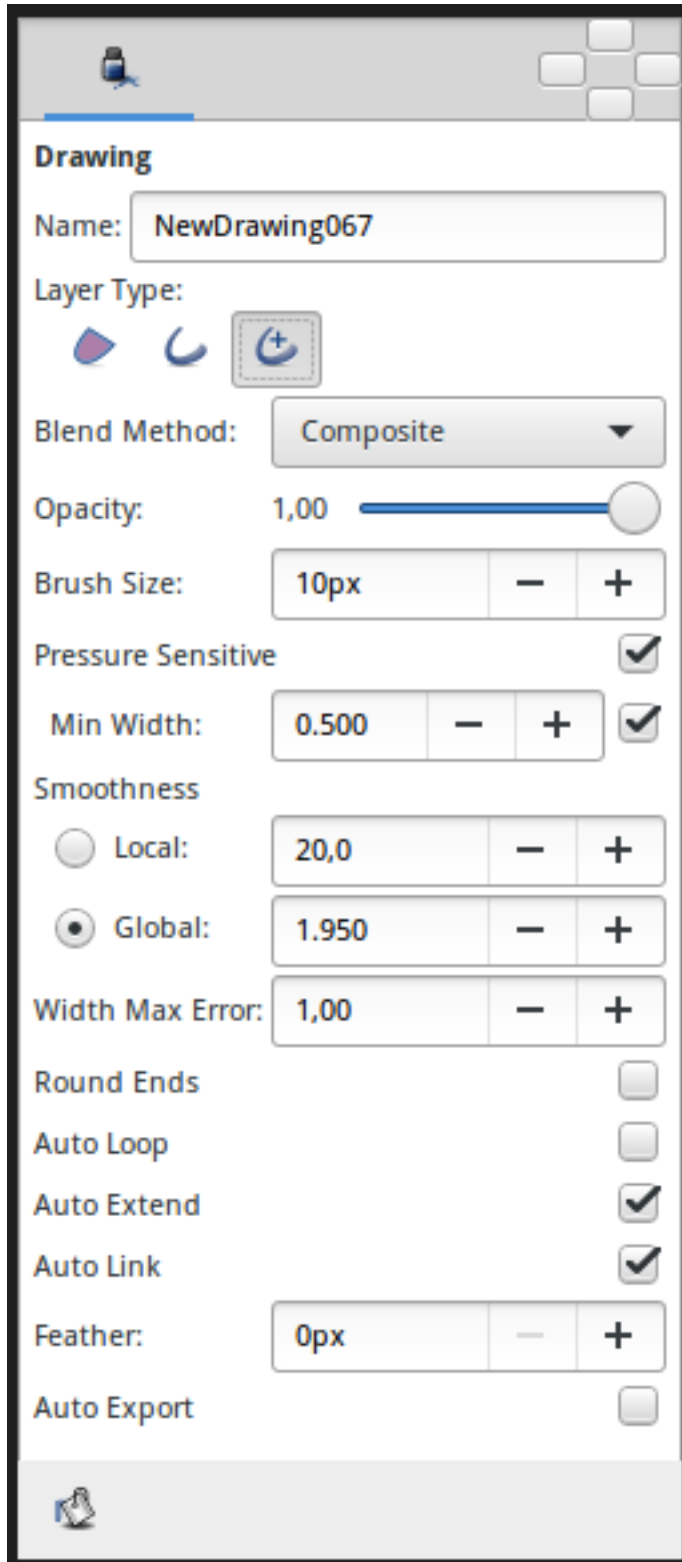
6.4 Draw Tool



The Draw Tool is the one that most artists will be familiar with. This is best used with a drawing tablet, although it can be used with a mouse.

The Draw Tool can generate a *Region Layer*, an *Outline Layer* and an *Advanced Outline Layer* with pressure output from the tablet mapped to the line's width.

6.4.1 Options



The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created
- **Layer Type** : Whether create *Outline Layer* and/or *Advanced Outline Layer*, *Region Layer* layers (Splines)
- **Blend Method** : The blending method used to composite on the layers below
- **Opacity** : Sets the Opacity for new layers. Defaults to 1.00 (Completely opaque)
- **Brush Size** : Sets the *Outline Layer: Outline Width* or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*)
- **Pressure Sensitive**
 - **Min Pressure** : The value of this option clamps the lower end of the Pressure sensitivity to a specific value, essentially determining how thin the thinnest possible segment is relative to the *New Layer Defaults: Brush Size* parameter in the *Toolbox*.
- **Smoothness** : Determines how much tablet jitter is removed, and as a result, how many vertices the resulting splines are composed of. Could be Local or Global
 - **Local** : Adjust the Spline between the Splines points.
 - **Global** : Adjust the Spline globally.
- **Width Max Error** : for advanced outline layer.
- **Round End** : The ends of the spline will be rounded, only for *Outline Layer*, *Advanced Outline Layer* layers.
- **Auto Loop** : When the stroke of the Draw Tool ends near the beginning of the stroke, automatically connect the last *Handle* to the first.
- **Auto Extend** : When the stroke begins and/or end at or near an endpoint *Handle* of another *Outline Layer*, the resulting *Outline Layer* is simply appended to the existing *Outline Layer*
- **Auto Link** : When the stroke begins and/or ends at or near a vertex *Handle* in another *Outline Layer*, and the outline isn't being extended by the **Auto Extend** option above, the first and/or last vertex of the resulting *Outline Layer* is *Linking* to the *Handle* on the existing *Outline Layer*.
- **Feather** : Set the *Feather Parameter* on the layer
- **Auto Export** : Automatically *Export* the Spline list points into the Values Base Nodes of the *Library Panel*

6.4.2 Buttons

Fill Last Stroke :  Hit this button after having created a shape with the Draw Tool to create a *Region Layer*, filled with the *New Layer Defaults: Brush Colors Fill Color*, of the last trace you have drawn.

6.5 Eyedrop Tool



ALT-E

The Eyedrop Tool is very simple - it allows you to select colors from the *WorkArea*.

6.5.1 Usage

Select the Eyedrop Tool, then click in the work area to set the default foreground color in the *Toolbox* to be the color you're currently pointing at.

For setting the background color, just switch colors (↻) before and after using the Eyedrop Tool.

6.5.2 Options

- None

6.6 Fill Tool



ALT-F

The Fill Tool allows you to change the color of some of the geometry type layers very quickly. It uses the default foreground color to be applied to the color of the layer you click on.

6.6.1 Usage

To use this tool simply click on the Fill Tool button and after that click onto a non transparent part of a visible layer.

The Fill Tool acts the same that if you right click the color parameter of any of the Layers that have it and select Apply Default Color from the context menu.

If the layer accepts the Fill Tool message it would change its color to the current foreground color. If the layer doesn't accept the message from the Fill Tool it complains and shows a message error.

Those are the layers that currently are affected by the Fill Tool:

- *Region Layer*
- *Outline Layer*
- *Circle Layer*
- *Rectangle Layer*
- *Polygon Layer*
- *Star Layer*
- *Checkerboard Layer*

The rest of visual layers doesn't accept the Fill Tool usage over them because they use a gradient to fill the visual area (like the Gradient Layers, the *Plant Layer* and others) or simply it is not possible determine if you're clicking over the layer or over a transparent area although it has a single color definition for the visual area (*Text Layer* for example).

6.6.2 Options

- None

6.7 Gradient Tool

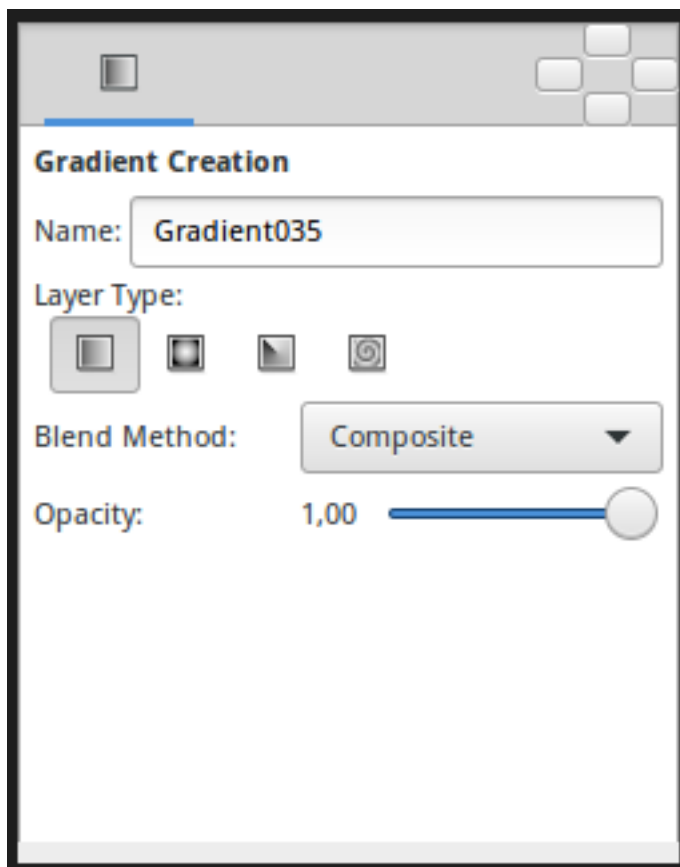


ALT-G

6.7.1 Introduction

The Gradient Tool is used to create smooth transitions between two or more colors in an object.

6.7.2 Options



When you select the Gradient Tool, the *Tool Options Panel* will show the options for the Gradient Tool.

These allow you to:

- **Name** : Set a name for the layer you are about to create. The name of the layer can always be changed later via the *Layers Panel* or the *Parameters Panel* if necessary, but the type of gradient needs to be chosen now.

- **Layer Type** : Choose the type of gradient to create (Linear, Radial, Conical, Spiral).
- *Blend Method Parameter* : The blending method used to composite on the layers below
- **Opacity** : Sets the *Opacity* for new layers. Defaults to **1.00** (Completely opaque)

For help with editing gradient colors see the section on *Gradient Editor Dialog*.

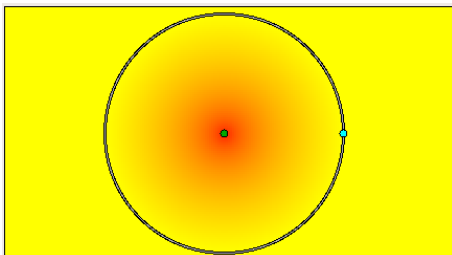
6.7.3 The 4 types of gradients

[Linear_Gradient_Layer]



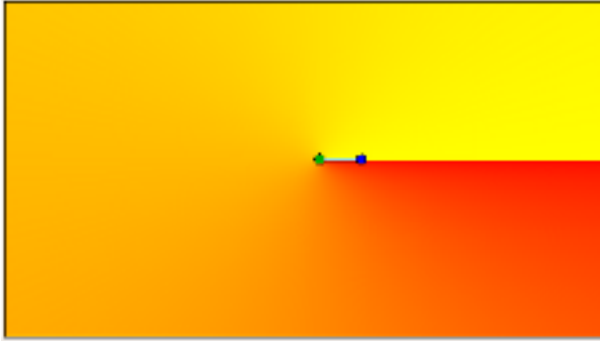
This produces a simple transition in a straight line. Set your foreground and background colors. Click where you want the gradient to begin, and drag to where you want the transition to end. The gradient will be created perpendicular to the line you drag out. At any time, you can edit the gradient by moving either endpoint in any direction. You must use the *Transform Tool* to be able to edit the endpoints.

Radial Gradient Layer



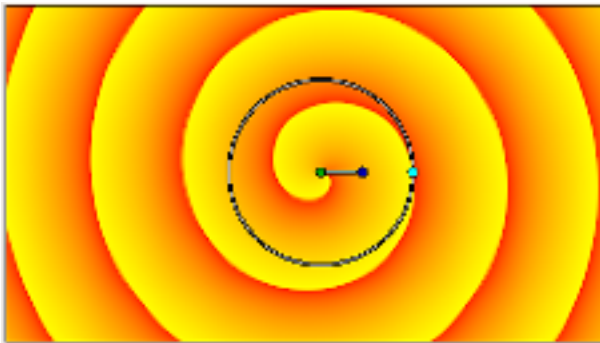
This produces circular colors with the transition being at the center of those circles. Click where you want the center of the circles to be, and drag to set the radius of the transition. Use the *Transform Tool* to edit the position (center endpoint), or the radius (surface endpoint), of the radial gradient.

Conical Gradient Layer



This has the appearance of looking down on a tip of a cone. The gradient is along the circular arc of the center and goes in all directions. Click to set the center, and drag to indicate the direction in which the foreground and background colors should go. To edit afterwards, use the *Transform Tool*. The center endpoint can be adjusted to move the center of the gradient. The other endpoint adjusts the direction of the gradient.

Spiral Gradient Layer



This produces a spiral gradient. Click to set the center of the spiral and drag to set the 'tightness'. To edit afterwards, use the *Transform Tool*.

Documentation writers note: You can download the project to generate the screenshot: `Gradient_options.zip`

6.8 Mirror Tool

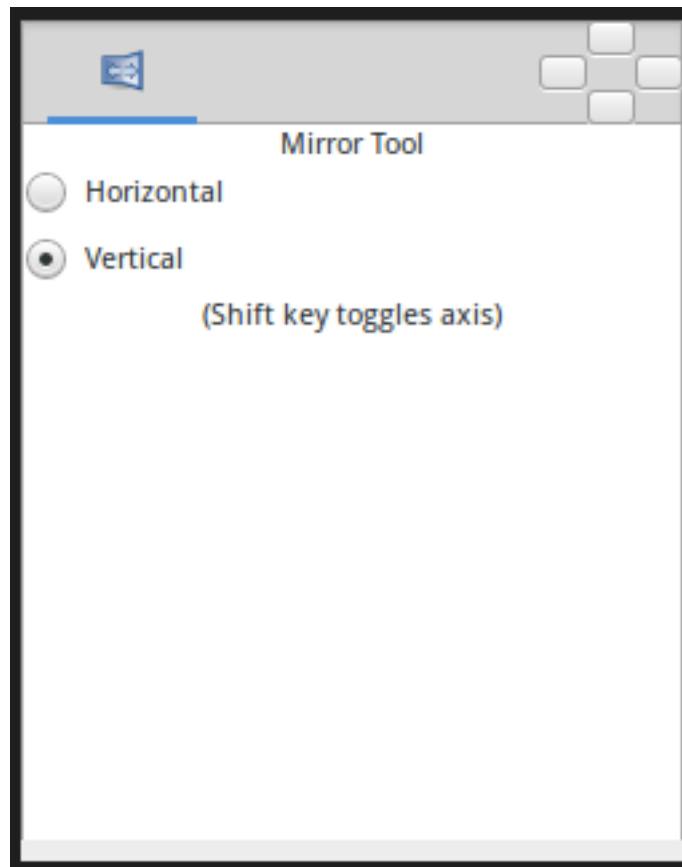


ALT-M

6.8.1 Introduction

The Mirror Tool will mirror all the selected *Handle* about a horizontal or vertical axis passing through the mouse cursor.

6.8.2 Options



- **Horizontal** : Horizontal axis mirror.
- **Vertical** : Vertical axis mirror.

6.8.3 Usage

For example, to mirror a spline horizontally:

- select the Mirror Tool, either by clicking in the *Toolbox* or hitting **Alt+m**
- check **Horizontal** in the *Tool Options Panel*
- select the Spline Layer
- select all handles in the spline by hitting **Ctrl+a**
- drag one of the selected handles a little

As soon as you start dragging, the handles will all flip about a vertical axis passing through the duck you're dragging. You will be able to drag the handles around, but the motion will be constrained to only move left and right.

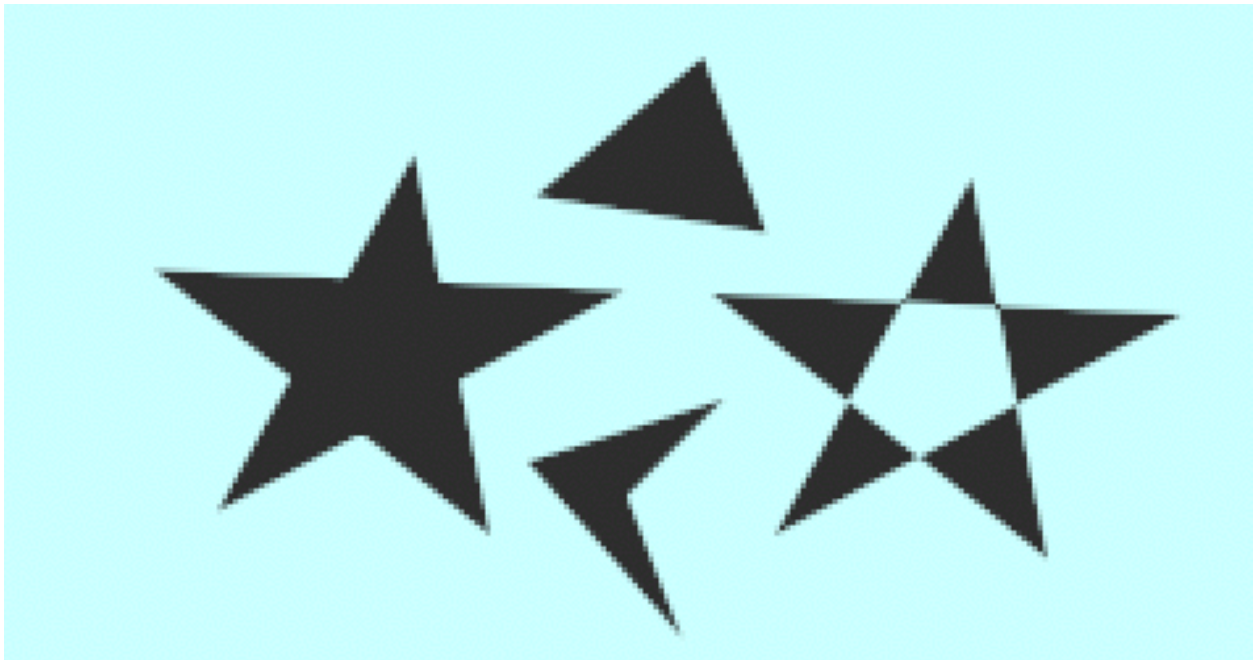
6.9 Polygon Tool



6.9.1 Introduction

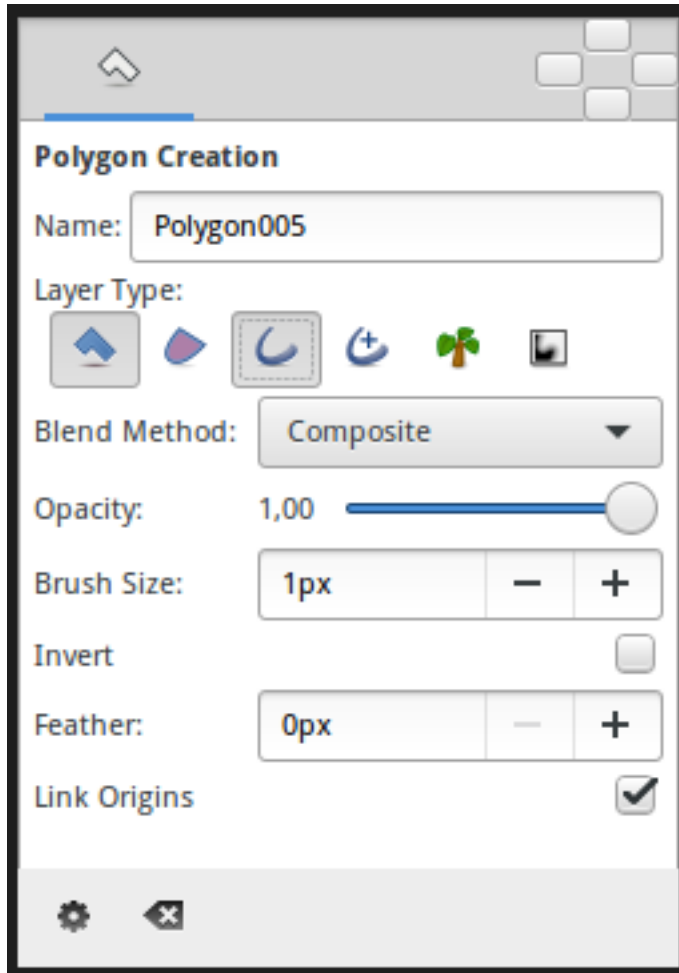
This tool draws regions with straight edges. It is deprecated in favor of the *Spline Tool* which can draw straight edged polygons and also curved shapes.

Here are some example polygons (in case you forgot what they look like):



Note that the edges can cross each other. When they do, the *Winding Style Parameter* comes into play.

6.9.2 Options



The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created
- **Layer Type** : Whether create *Polygon Layer* and/or *Outline Layer*, *Advanced Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines)
- *Blend Method Parameter* : The blending method used to composite on the layers below
- **Opacity** : Sets the *Opacity* for new layers. Defaults to **1.00** (Completely opaque)
- **Brush Size** : Sets the *Outline Layer: Outline Width* or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*)
- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert Parameter*.
- *Feather Parameter* : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*)
- **Link Origins** : Whether Link the Offset of Splines

Usage

Just click over the different corners of the polygon.

To finish the creation of the spline you have different options :

- Click again over the first vertex.
- Using the gear Make Polygon button at the bottom of the *Tool Options Panel*.
- Select any other tool from the *Toolbox Panel*.

The Clear Current Polygon button at the bottom of the Tool Options Panel could be used to clear the current polygon.

6.10 Rectangle Tool



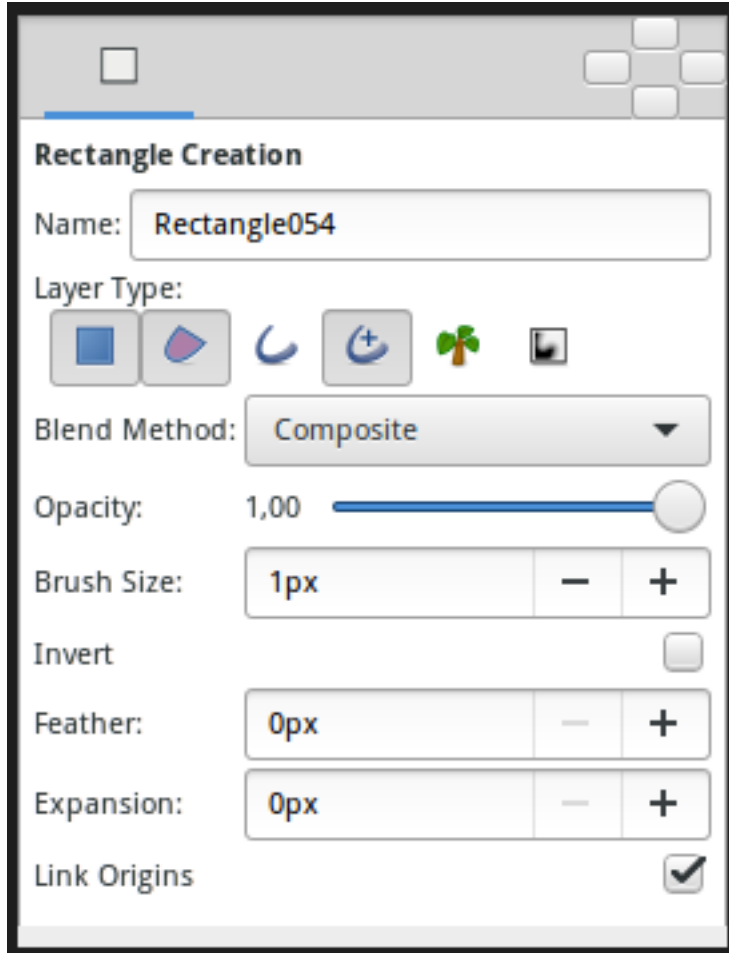
ALT-R

6.10.1 Introduction

The Rectangle Tool is a who creates *Rectangle Layer* and easy creation of splined rectangles. Click where you want any corner of the rectangle to be, and drag to the opposite corner.

The Rectangle Tool can also be used to create *Outline Layer*, *Advanced Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* .

6.10.2 Options



The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer name; any number in this field will automatically increment with each layer created.
- **Layer Type** : Create *Rectangle Layer* and/or *Region Layer*, *Outline Layer*, *Advanced Outline Layer*, *Plant Layer* and *Curve Gradient Layer* (Splines).
- **Blend Method Parameter** : How to combine the new layer with the background.
- **Opacity** : Sets the *Opacity* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size** : Sets the *Outline Layer: Outline Width* or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Expansion** : Expand the rectangle from his corners (only for *Rectangle Layer*. Is related to his *Expansion Amount Parameter*).
- **Feather Parameter** : Width of the area to be dissolved at the edge (not for *Rectangle Layer*, *Plant Layer* and *Curve Gradient Layer*).
- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert Parameter*.
- **Link Origins** : Whether Link the Offset of Splines.

6.10.3 Usage

Click on the *Canvas* where you want to have corner of the rectangle to be, drag to the opposite corner, and release the mouse button when you are done.

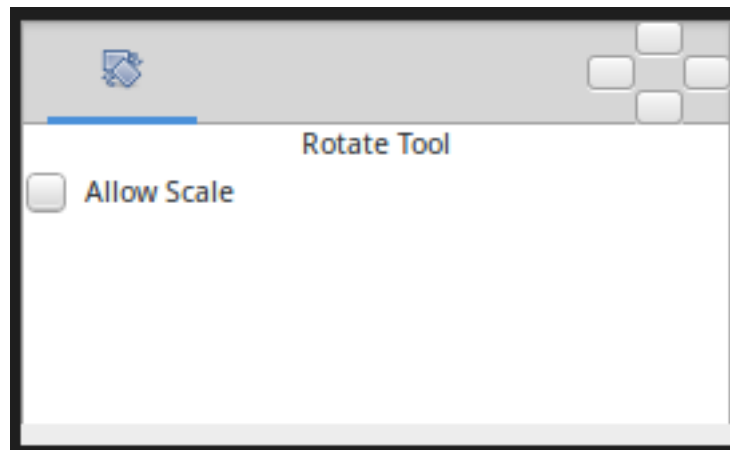
6.11 Rotate Tool



6.11.1 Introduction

The Rotate Tool rotate the selected *Handle*. It works much like to *Transform Tool* except for when multiple *Handle* are selected.

6.11.2 Options



- **Allow Scale** : Allow the selected Handles to be scaled and rotated at the same time.

6.11.3 Usage

Handles: Choose your handles, by either:

- Pressing Ctrl+A to select all Handles.
- Dragging a rectangle around a group of Handles.
- Holding the Control key while clicking Handles to toggle their selected status.

Then drag one of the selected Handles. All the selected Handles will move, rotating about the center of the selected handles group.

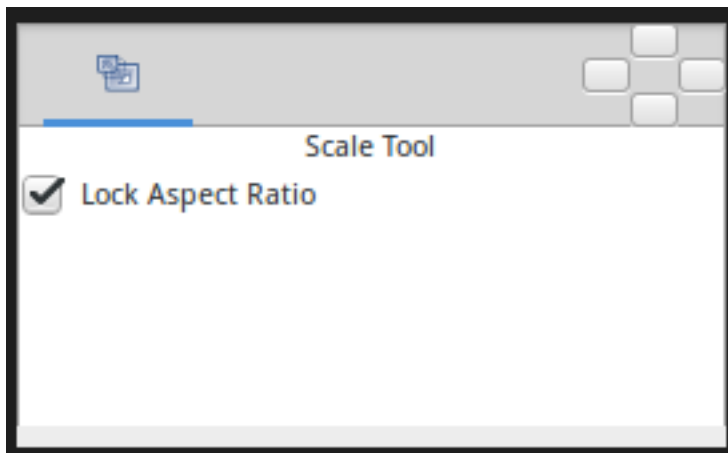
6.12 Scale Tool



6.12.1 Introduction

The Scale Tool permit to scale the selected handles. It works much like to *Transform Tool* except for when multiple *Handle* are selected.

6.12.2 Options



The *Tool Options Panel* allows you to specify:

- **Lock Aspect Ratio** : Prevent from changing the ratio x:y of the bounding box of the selected Handles.

6.12.3 Usage

Handles: Choose your handles, by either:

- Pressing Ctrl+A to select all Handles.
- Dragging a rectangle around a group of Handles.
- Holding the Control key while clicking Handles to toggle their selected status.

Then drag one of the selected Handles. All the selected Handles will be scaled relative to the center of the selected group.

When scaling a selection don't drag a Handle that its position is close to the X or Y center of the bounding box. Select a Handle that is close to one of the corners of the bounding box. If you drag from a Handle that have a X or Y coordinate that is the same than the center of the bounding box it would produce an infinite scale factor and a very difficult to control scaled shape. (The Handles that aren't over the bounding box X or Y center are scaled by a huge factor). This is a behavior that should be corrected (for example giving to the user corner pulling anchors to allow make a correct scaling).

6.13 Skeleton Tool

Note: Available since Synfig Studio 1.5.0.

6.13.1 Introduction

Synfig supports animating objects using bones and the Skeleton tool makes it easy to build any bone structure.

6.13.2 Video Tutorial

6.13.3 Usage

Select “Skeleton Tool” in the Toolbox Panel or use the keyboard shortcut <Alt><o> to enable it. To leave the tool, you can just select another tool from the toolbox.

There are two types of layers in Synfig that are built with bones: Skeleton Layer and Skeleton Deformation Layer. We can build any of these layers.

You can create a bone by either

- **Click-release at a position:** This will create a bone with default length and angle but with the width value as set in the tool options menu. (The tip and origin width of the bone will be same) or,
- **Click-drag-release between two positions:** This will create a bone between the points you dragged between with the width value as set in the tool options menu. (The tip and origin width of the bone will be same)

If you try to use this tool when the selected layer is not any of the bone layers, a new layer is created with the bone added. The layer that is to be created can be set in the tool options menu. You can always create an empty bone layer using the “Create Layer” button in the tool options menu.

The parent child relationships of bones are represented by a dashed line from parent’s tip to the child’s origin.

When you click on a bone, the bone is set as the “**Active Bone**”. Active bone is highlighted with a yellow outline. Any bone that’s created is added as a child to the active bone. When there is no active bone set, then the bone created after will have no parent.

The handles of the bone will have same meaning as they had before. A point to be noted is that, you can even click on a Handle to set a bone as “Active Bone”. When you use a handle to modify a bone, note that the bone is set as the “Active Bone”.

The parent-child relationships of bones can be changed by using the “**Make parent to active bone**” option in the context menu of any bone handle.

When you are modifying a skeleton deformation layer using the Skeleton tool, it is to be noted that

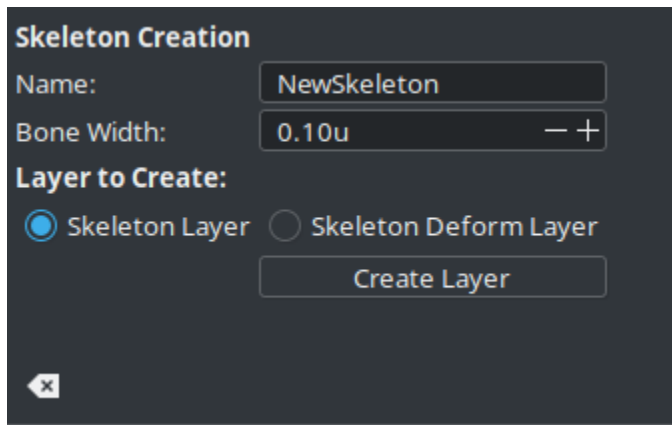
- The layer gets disabled.

- Width handles are turned on.
- Any changes you make to the rest position effects the deformation too.

6.13.4 Options

You can find the following options in the tool options menu:

- **Name:** The name of the layer that will be created.
- **Bone Width:** The width of the bone that will be created next.
- **Layer to Create:** The layer type that will be created.



6.14 Sketch Tool



ALT-K

The Sketch Tool allows you to draw freehand over the top of a *Canvas*.

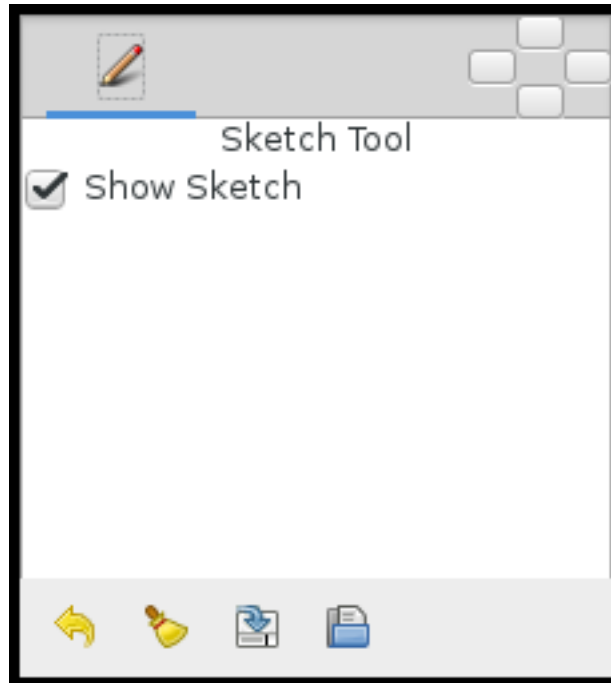
Each workarea has a single sketch overlay associated with it.

Sketches are not saved as part of the **.sifz** file when you save your work, but can be saved separately as **.sketch** files for later reloading.

6.14.1 Options

The Sketch Tool has the following in its *Tool Options Panel* :

- **Show Sketch:** a checkbox for toggling the visibility of the sketch overlay **On** and **Off**.
- **Undo Last Stroke:** cancels the most recent edit to the sketch overlay. Can be repeatedly clicked to cancel more edits.
- **Clear Sketch:** clears the current workarea's sketch overlay.
- **Save Sketch As...:** writes the sketch overlay from the current workarea to a **.sketch** file.



- **Open a Sketch:** loads a previously saved .sketch file, overwriting any previously drawn or opened sketch in the current workarea.

Sketch uses the color of the default foreground color in the *Toolbox*.

6.15 Spline Tool



6.15.1 Introduction

Spline are the most used objects in any normal animation done with Synfig, and the Spline Tool allows you to create them.

6.15.2 Usage

Select the **Spline Tool** in the or use keyboard the shortcut **Alt+b** to enable it. To leave that mode just select another from the *Toolbox*.

It is easy to use this tool

- Each single click creates a new **Vertex**. Its **Tangent** values are set to (0,0) and the **Vertex** becomes a sharp corner.
- Click and drag creates a new **Vertex** and allows the modification of the **Tangent** value at that **Vertex**. Extending the **Tangent** smooths the corners.
- Each new **Vertex** is attached to the previous by a **Bézier Spline** defined by the vertices' position and tangents.
- The **Spline** is created sequentially, one **Vertex** follows the previous one.
- The end point of the previous **Spline** arc guides the next arc created until you close it. The next created **Spline** will be independent of the previously created one.
- You can tweak the newly created Vertices by clicking on their *Handle* and dragging them.
- You can also right click on a **Vertex** handle or a **Tangent** handle and it will provide a context menu:
 - For **Tangent** handles: **Split Tangent's Angle** and **Split Tangent's Radius** will split the **Tangent** so that you can independently modify the arch of an end point of a **Bézier** curve. On the opposite, **Merge Tangent's Angle** and **Merge Tangent's Radius** will merge the independent **Tangents** so that you can align the arches of the end points of two connecting **Bézier** curves.

- For Vertex handles: Loop Spline or Unloop Spline (depending if already looped or not) allows you to close or open the Spline. Delete Vertex allows you to delete the Vertex.
- You can also right click in the middle of a Bézier curve. Insert Vertex allows the insertion of a Vertex (and sets its Tangents intelligently according to the neighboring Vertices) where you clicked.

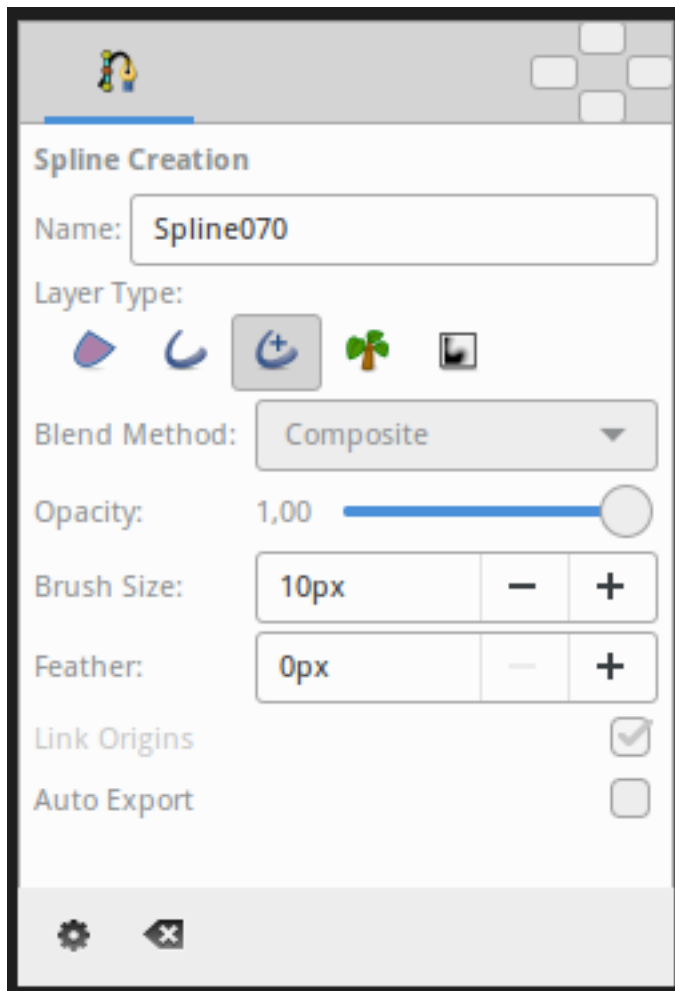
To finish the Spline

1. Select another tool in the .
2. Or click on the gear button  of the *Tool Options Panel*.

To clear out the working Spline, press the Esc button.

Category:Toolbox

6.15.3 Options



This tool has the following options:

- **Name:** Sets the name of the Spline that you will create, in this case “Spline070”. If the ending part of the string contains an integer number, this input text box will parse it and increase that number for the next created Spline. Otherwise, it will add a three digit number while creating the next one.

- **Layer Type:** Sets whether to create *Region Layer*, *Outline Layer*, *Advanced Outline Layer*, *Plant Layer* and/or *Curve Gradient Layer* Layers.
- **Blend Method Parameter:** Sets the blending method used to composite on the layers below.
- **Opacity:** Sets the *Opcaity* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size:** Sets the size of the line for *Outline Layer* and/or *Advanced Outline Layer*.
- **Feather:** Sets the *Feather Parameter* of the *Region Layer* or Outline Layer created.
- **Link Origin:** If checked, links the *Origin Parameter* for the *Plant Layer*, *Region Layer* or Outline if two of them (or all them) are checked.
- **Auto Export:** If checked, exports automatically the Vertices Parameter (that's a *Spline* type parameter)

6.15.4 Other sources of information

TODO: Add sources

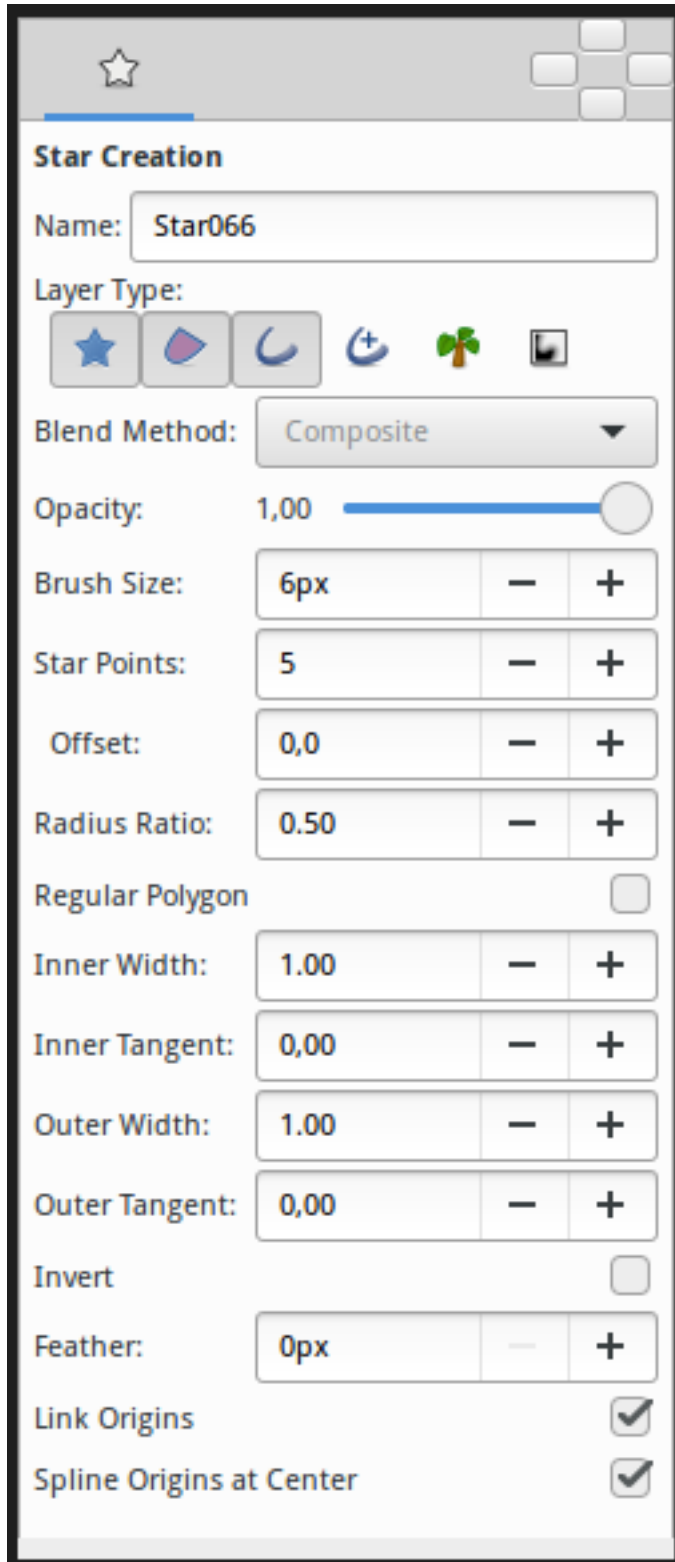
6.16 Star Tool



The Star Tool allows the easy creation of *Star Layer*.

Select the Star Tool icon from the *Toolbox*, then click in the *Canvas* where the center of the star should be created and drag to specify the outer radius.

6.16.1 Options



The *Tool Options Panel* allows you to specify:

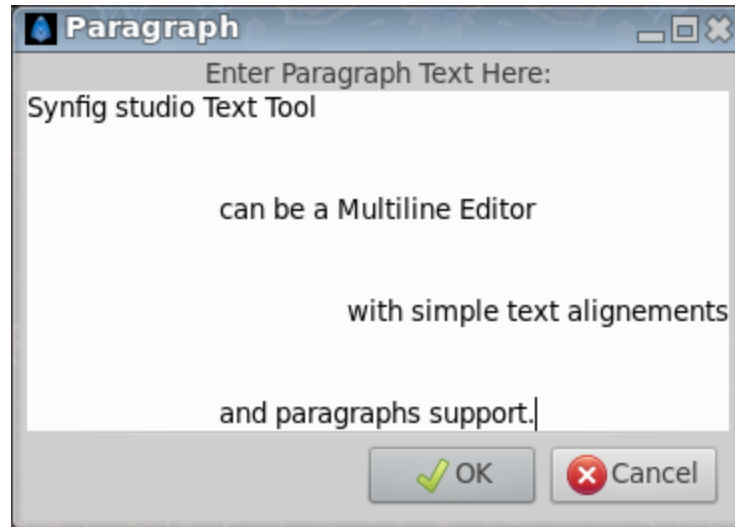
- **Name** : The name used for the new layer; any number in this field will automatically increment with each layer created
- **Layer Type** : Whether create *Star Layer* and/or *Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines)
- **Blend Method Parameter**: How to combine the new layer with the background.
- **Opacity** : Sets the *Opacity* for new layers. Defaults to **1.00** (Completely opaque)
- **Brush Size** : Sets the *Outline Layer: Outline Width* or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*)
- **Star Points** : The number of peaks of the star.
- **Offset** : The rotation offset of the created star.
- **Radius Ratio** : The ratio between the peaks and the troughs of the star
- **Regular Polygon** : Whether create a regular polygon instead of a star.
- For Splines only, the tangent lengths for outer and inner vertices. Outer is limited to [-3,10] and inner is limited to [-3,3]. (*If you find useful open those limitations please write it in the talk page*)
 - **Inner Width**
 - **Inner Tangent**
 - **Outer Width**
 - **Outer Tangent**
- **Invert** : Whether invert the create(d) layer(s)
- **Feather Parameter** : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*)
- **Link Origins** : Whether Link the *Origin Parameter* of Splines
- **Spline Origins at Center** : Set the *Origin Parameter* of the layer at the center of the star, else the center is set at the *Canvas* center

6.17 Text Tool



Select the Text Tool icon from the *Toolbox*, then click in the *Canvas* where the *Text Layer* should be created.

A dialog box will appear, allowing you to enter the text you want in the created layer. Enter the text, click OK button, and a new Text Layer will be created. In the dialog box, if Multiline Editor is checked, Enter will add a new line, you can hit Ctrl | Tab to give focus to OK button to validate.



6.17.1 Options

The *Tool Options Panel* allows you to specify:

- the **Name** used for the new layer; any number in this field will automatically increment with each layer created.
- **Multiline Editor** - whether to use a single or multi-line editor for entering the text. If a single line is used then the Enter key will submit the value. In the multi-line editor, you'll need to click the OK button with the mouse to confirm the text or hit Ctrl|Tab to give focus to OK button.
- **Size X Y** - the horizontal and vertical size of the text.
- **Orientation X Y** - the orientation of the text. The default (0.5, 0.5) means that the text will be centered around the point you clicked on. (0, 0) means that the top left corner of a box containing the text will be placed where you clicked. (1, 1) means that the bottom right corner of the text's bounding box will be placed where you clicked.
- **Family** - the font family to be used.

6.18 Transform tool



ALT-A

6.18.1 Introductions

The Transform Tool lets you select an object and manipulate its handle (move, rotate or scale).

6.18.2 Usage

Clicking an object in the *Canvas Window* with the **Transform** tool selects it. You can select multiple objects by holding down the **Control** key and clicking on the additional objects.

*If the layer is a “no edge layer” (gradient for example) you cannot click on the layers below, even if you have clicked in the transparent area of the layer. This allow to select only current visible layers. Consider also that some layers have not yet defined a way to click on them to be selected (*Text Layer* for example).*

If, with the **Transform Tool**, you can not select an object because it is obscured by another object you can select its layer in the *Layers Panel*.

Once a layer is selected, its *Handle* is displayed. The **Transform** tool works on selected handles only.

Click on a handle to select it. A selected handle becomes highlighted. You can select more handles by **CTRL** Click on each one. You can also drag a Selection box around the handles to select them.

Ctrl+A will select all handles in the selected layers. (**Ctrl+D** will deselect all handles)

Dragging a Selection box while pressing **CTRL** will toggle handle selection. Currently selected handles within the Selection box will become deselected and currently non selected handles will become selected.

Clicking on selected handles with the **Transform** tool will allow you to drag them around in the canvas. Make sure you click on them, clicking outside will deselect the handles.

Pressing **shift** while dragging allows you to restrict the movement to the X or Y coordinates.

Pressing **CTRL** while clicking handles will allow you to rotate them around a calculated center for the currently selected handles.

Pressing **ALT** while dragging handles will scale them around their calculated common centre. Pressing **SHIFT** while scaling will preserve the ratio of the handles.

If you want to constrain a tangent handle to X or Y movement you have to press **SHIFT** after you start to drag it.

Otherwise, pressing **SHIFT** before dragging a tangent will split it into separate left and right tangents.

6.18.3 Options

- None

Note: The “Transform Tool” was called the “Normal Tool” before Synfig version 0.62.01.

6.19 Width Tool



6.19.1 Introductions

The Width Tool is designed for increasing or decreasing the width of a line much like you would with a pencil on paper.

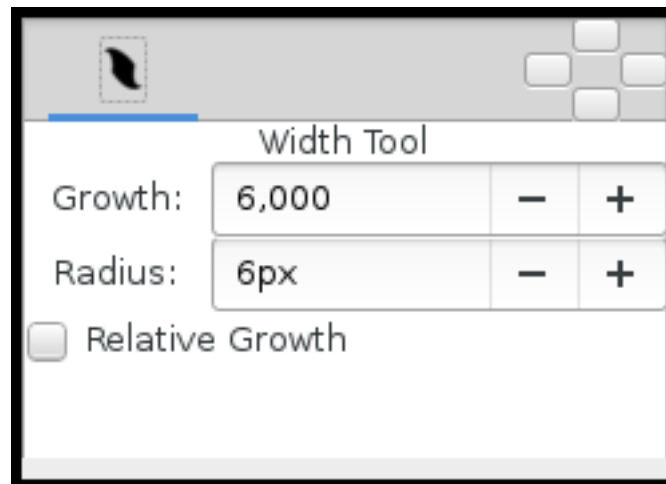
6.19.2 Usage

To change the width on a particular segment of a line, just select the *Outline Layer* you want, move your mouse over that line, press the left mouse button and move cursor back and forth along the line, like you're scratching something. The width of outline will be increased at the places where you moved the cursor. If you want to decrease the width, just hold **Ctrl** while scratching.

The Width Tool was primarily designed for the "tablet" workflow, so you will get the best experience by using it with the tablet stylus. It is similar to making line thicker with a pencil or thinner with an eraser (when holding the **Ctrl**).

Note: The Width Tool is intended to fine-tune line width. If you want to heavily increase/decrease the line width it's better to use the Transform Tool to directly manipulate the width *Handle*.

6.19.3 Options



- **Growth** - Defines how much a single mouse move will affect the line width. It must be non-zero for this tool to have an effect on the spline.
- **Radius** - Defines the size of area around current cursor position in which a vertex will be affected. It allows you to achieve a noticeable effect without having to follow the line precisely with the mouse. The Width Tool works fine on splines with lots of segments (such as those created with the *Draw Tool*), you will see it works on more than just the first vertex. It is a full circular area, other points may get in the way though.
- **Relative Growth** - Doesn't really work very well unless the Radius parameter is set to some huge value (like a million).

6.19.4 Nota

If you don't want the `Width Handles` to be displayed, while using the `Width Tool`, just turn them off by pressing `Alt+5`. You can turn them on again with the same keybinding.

6.20 Zoom Tool



6.20.1 Description

The `Zoom Tool` allows you to move closer to, or further away from, objects that are being edited in the working area. Note that the `Zoom Tool` does not affect the output render as described in the [Render Options](#).

6.20.2 Usage

Click on the `Zoom Tool` button to activate. Now try the following:

- Click anywhere in the working area: it will zoom to 125% of the current level. The zoom will be centered such that the point clicked on remains in the same position.
- `CTRL` + click anywhere in the working area: it will zoom to 80% of the current level. (Notice that 80% of 125 is 100, so zooming in one step and out one step will leave you back at the original zoom level).
- Click at any place and drag to create a rectangle: it will zoom to fit the rectangle in the current working window.
- Click at any place and drag to create a rectangle, holding `CTRL` as you release the mouse button: it will zoom out to fit the current working window in the rectangle.

Alternatively you can always use the existing [Mouse Shortcuts](#) for that task (`CTRL` + mouse wheel) or the zoom buttons at the left bottom corners of the working area.



They do following:

- Zoom in to 125% of the current zoom level
- Zoom to 100% (press again and restore last zoom)
- Zoom to fit the window (press again and restore last zoom)
- Zoom out to 80% of the current zoom level

6.20.3 Options

- None

7.1 Advanced Outline Layer



7.1.1 About Advanced Outline Layers

To create an Advanced Outline Layer use the Spline Tool and check **Create Advanced Outline** in the *Tool Options Panel*. Once you finish the definition of the geometry of your outline and after pressing the **Make Spline** button (or selecting another tool or state) you create the Advanced Outline Layer with the *New Layer Defaults: Brush Color Outline Color*.

Depending on the options you choose in the *Tool Options Panel* other like *Star Tool* or *Circle Tool* can also create Advanced Outline Layers.

Using the *Canvas Layer Menu* **Make Advanced Outline**, **Make Outline** or **Make Region** commands you can also create those layers.

Differences with Outline Layer: Outline width

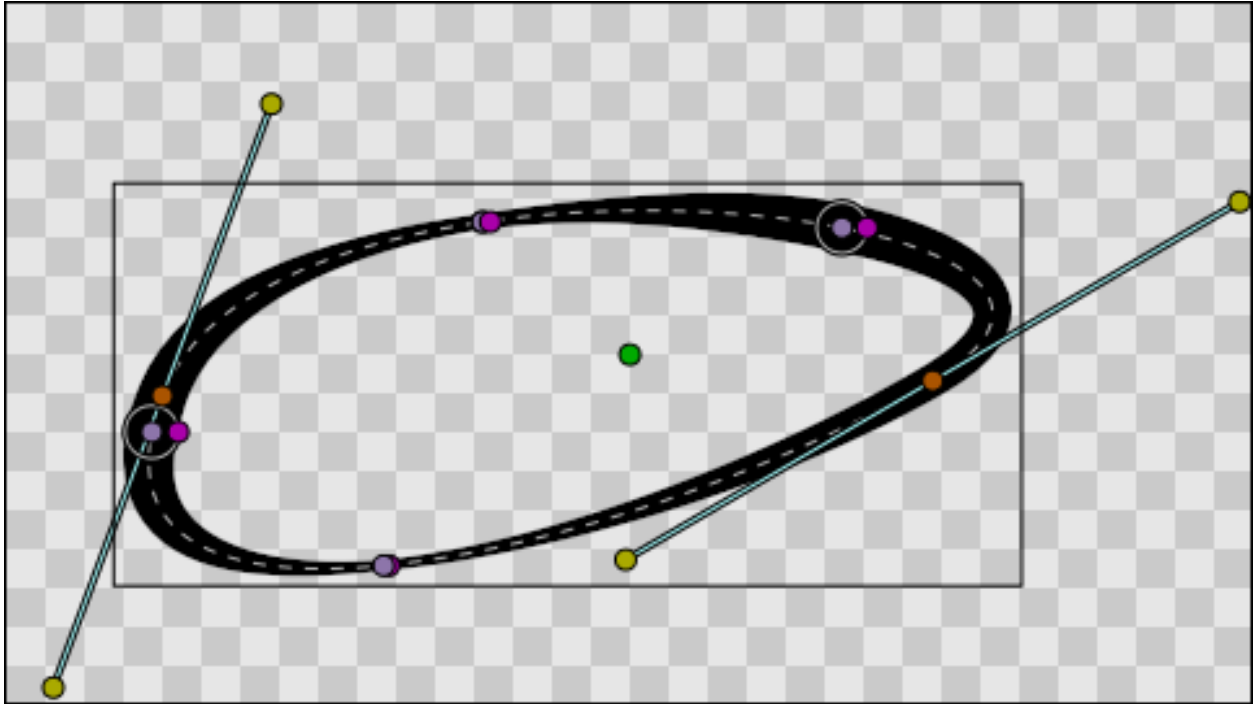
The **width Handle** defines the outline width at a particular place.

In the regular *Outline Layer*, each width Handle is connected to the vertex that defines the outline shape at that place. That means that the user needs to insert a new vertex on the Spline [<https://en.wikipedia.org/wiki/Spline_\(mathematics\)>](https://en.wikipedia.org/wiki/Spline_(mathematics)) every time he wants to change the outline width in some place. This behaviour can cause problems when the shape of the outline is very simple but the width varies a lot.

Practice shows that most of the time the user wants to specify the width independently from the placement of Spline [<https://en.wikipedia.org/wiki/Spline_\(mathematics\)>](https://en.wikipedia.org/wiki/Spline_(mathematics)) vertices. Such functionality is available in the Advanced Outline layer, which Handles its width in a different way: the width Handles are freely moved around the Spline and define its width at any point.

7.1.2 Features

Advanced Outline introduces some features to the regular outline, as shown in this [Advanced outline demo video](#). They are listed with examples below:



No need to place a vertex to change the width.

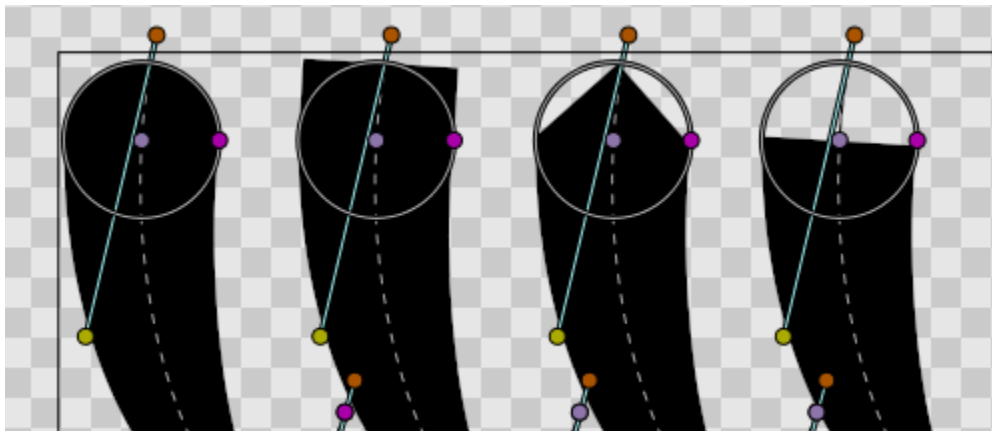
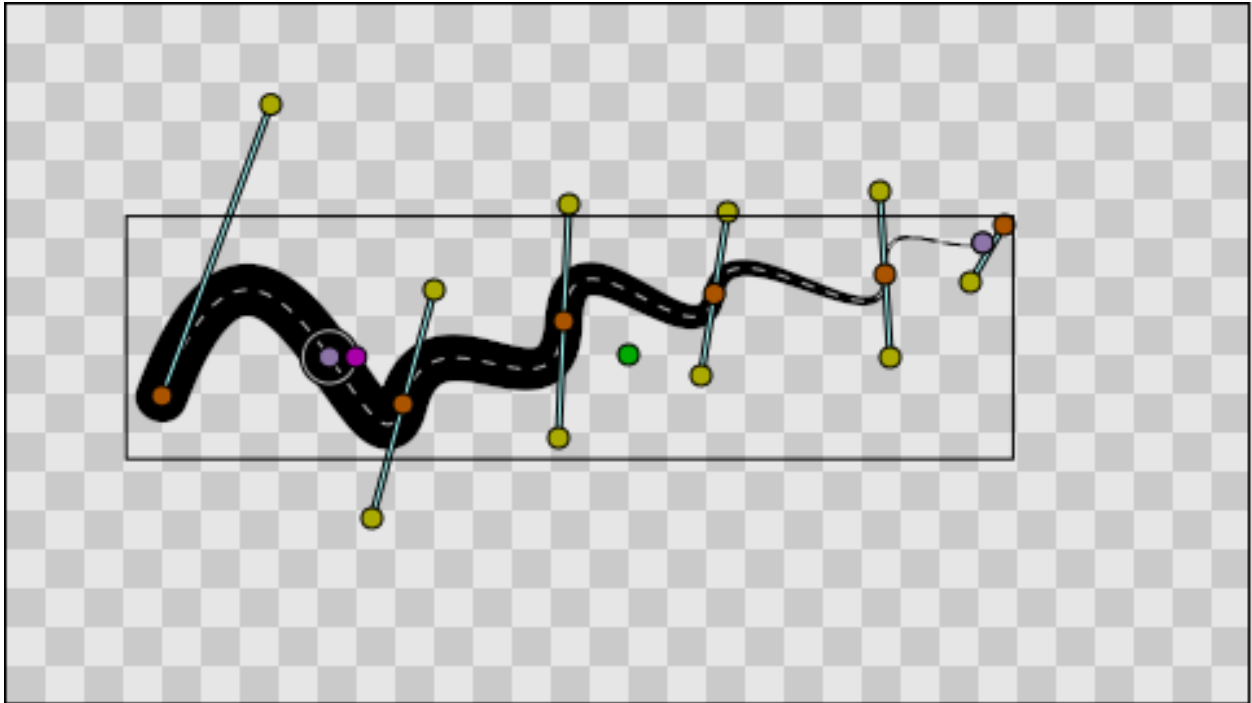
The width is controlled just by two points, while the Spline is constructed of many vertices.

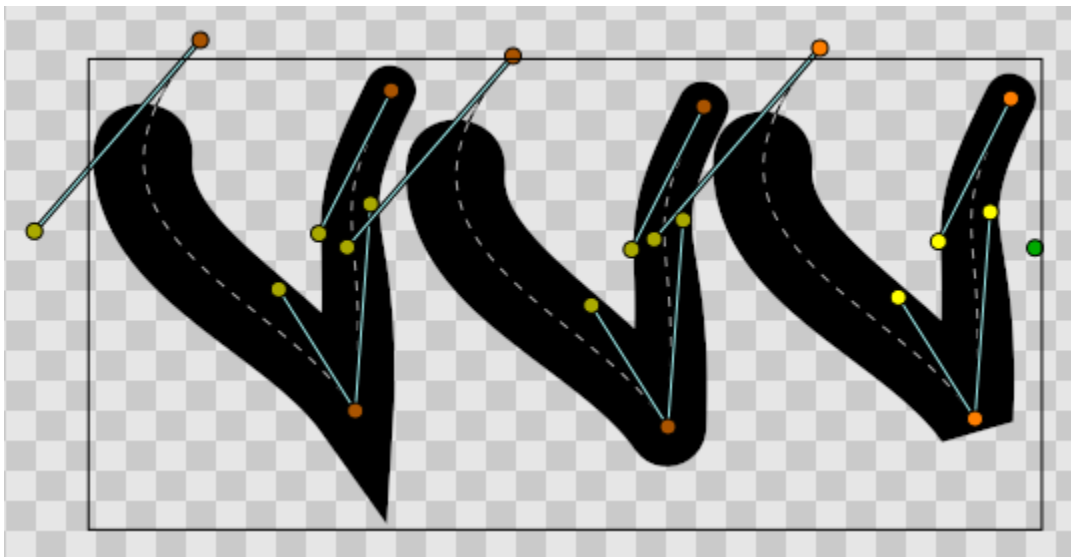
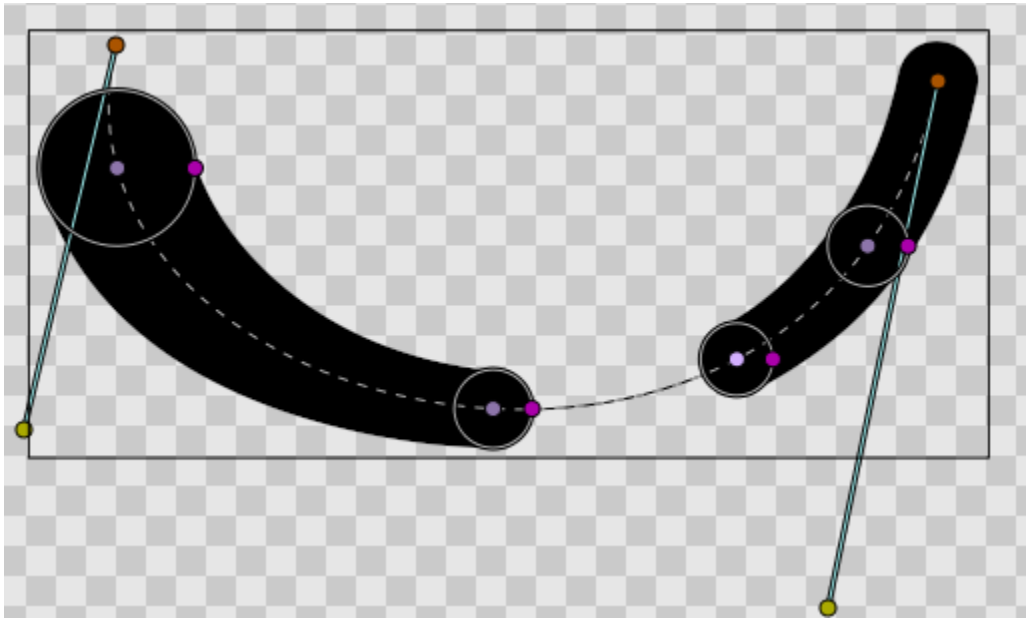
Different types of tips: Rounded, Squared, Peak, Flat.

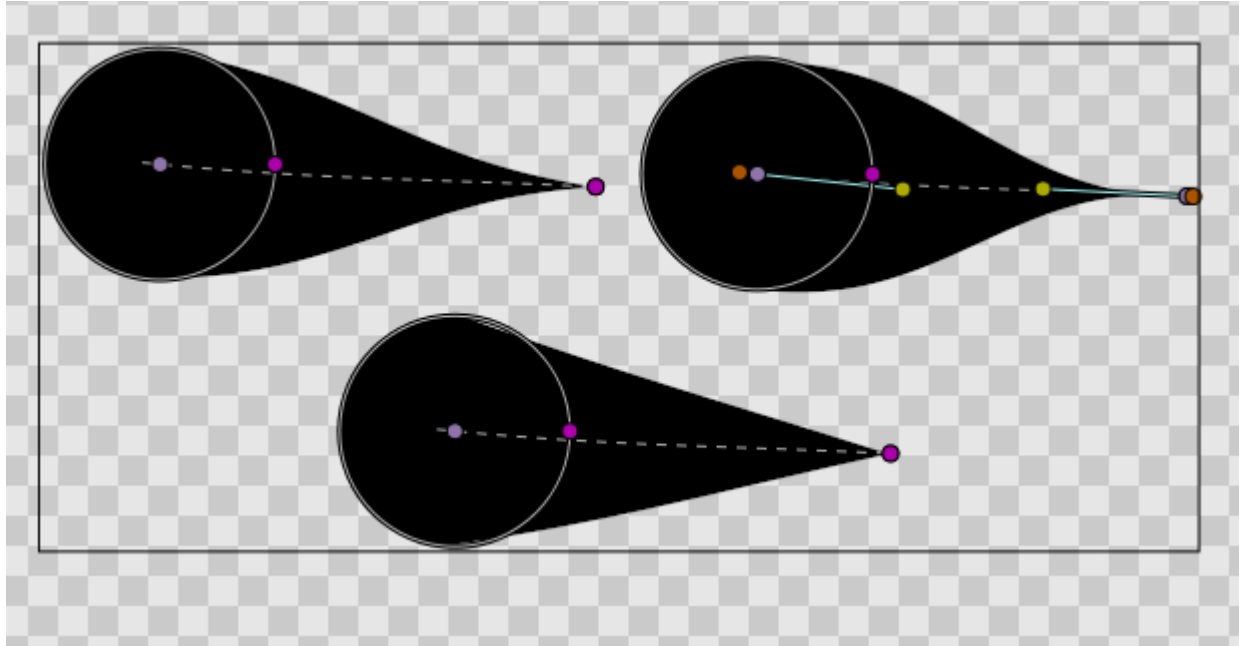
Segmented outline.

Three global cusps types: Sharp, Rounded, Bevel.

Control of smoothness from linear (0.0) to smooth (1.0).







7.1.3 Parameters of Advanced Outline Layers

The parameters of the Advanced Outline Layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Origin</i>	0.000000u,0.000000u	vector
 <i>Invert Parameter</i>	☐	bool
 <i>Antialiasing Parameter</i>	☐	bool
 <i>Feather Parameter</i>	0.000000pt	real
 <i>Feather Parameter: Type of Feather</i>	Fast Gaussian Blur	integer
 <i>Winding Style Parameter</i>	Non Zero	integer
 <i>Vertices</i>	List	list (Spline)
 <i>Outline Layer: Outline Width</i>	2.000000pt	real
 <i>Outline Layer: Expand</i>	0.000000pt	real
 <i>Tip type at start/end</i>	Rounded Stop	integer
 <i>Tip type at start/end</i>	Rounded Stop	integer
 <i>Cusps Type</i>	Sharp	integer
 <i>Smoothness</i>	0.500000	real
 <i>Homogeneous</i>	☐	bool
 <i>Width Point List</i>	List	list(WPList)
 <i>Fast</i>	☐	bool
 <i>Dashed Outline</i>	☐	bool
 <i>Dash Item List</i>	List	list(WPList)
 <i>Dash Items Offset</i>	0.000000u	real

7.1.4 Specific parameters for Advanced Outline Layer

This section lists only the parameters specific to the Advanced Outline Layer. For documentation about the other parameters, refer to [Outline Layer](#).

- Tip Type at Start
- Tip Type at End
- Cusps Type
- Smoothness
- Width Point List

Tip Type at Start / End

As with width points, the end and start of the unlooped Advanced Outlines has a type of tip defined. The user can choose between the same types of tips as for a width point. When the first/last width point has its before/after interpolation type set to Interpolate the start/end of the outline is rendered using the Tip type at start/end parameter.

Those parameters don't have any effect if the Spline is looped or the first/last width point has its before/after interpolation type set to anything but Interpolate. In that case, the segment between the start/end width point and the start/end of the Spline is not rendered.

The types of tips are the same as the width point tip types except that it doesn't offer the Interpolate type because it would not make sense:

Parameters:

Tip Type	Example
Rounded	
Squared	
Peak	
Flat	

Cusp Type

There are three types of cusps in the Advanced Outline:

Parameters:

Cusp Type	Example		
Sharp			
Rounded			
Bevel			

The type of cusp is controlled for the entire layer so currently it is not possible to control the type of corner individually. Maybe in future versions it will be possible.

Smoothness

The Smoothness controls how the width is calculated between widthpoints. The width at a position p is a function of the surrounding width points. When smoothness is **zero** interpolation is lineal, when smoothness is **1.0** interpolation is given by a 5th degree smooth Spline.

Homogeneous

Enabling Homogeneous changes the way the position of the *widthpoint* change according modification of the outline.

- When false, the “Position” parameter for a width point is equally distributed **among the vertices**. For example, in a spline with five vertices, “Position” values of 0, 0.25, 0.5, 0.75, and 1 correspond to the first, second, third, fourth, and fifth vertex, respectively, regardless of how close or far apart those vertices are. Between vertices, the “Position” parameter is based on spline length.
- When true, the “Position” parameter increases smoothly from the start to the end of the spline **based on the length of the entire spline**. This means a “Position” of 0.5 will always correspond to the halfway point of the spline, not to a particular vertex.

Width Point List

Each Advanced Outline has a list of parameters that represent the information for each width item. They are called Width Points and consist of four sub-parameters:

- **Position** (Real number): represents the position of the width point along the Spline. Although it is allowed to be any real number, its meaning is only from **0.0** to **1.0**. 0.0 corresponds to the start of the Spline (first Spline point on the Spline list) and 1.0 to the last Spline point. For looped Splines 0.0 and 1.0 are equal. The position is represented by the light purple Handle that always lies on the Spline. The point on the spline corresponding to a particular “Position” value is affected by the “Homogeneous” parameter, explained above.
- **Width** (Real number): It is the width multiplier of the global Width parameter of the Advanced Outline Layer on the position given by the Position parameter. The final width is calculated multiplying the global Advanced Outline’s Width (W) by the Width of the widthpoint (w) and adding the Expand parameter (E). Calculated width = $W * w + E$
- **Tip Side Before/After**: Those two sub-parameters controls how the width is interpolated before and after the current widthpoint. The sub-parameter can have four values:
 - **Interpolate**: Between the previous/following width point, the width is calculated by interpolation based on smoothness value.
 - **Rounded**: There is a rounded tip that points to the width point before or after. If the previous/following width point is **Interpolate** on its posterior/previous side it considers that the width of the widthpoint in question is zero just before/after it. If the previous/posterior width point is other than **Interpolate** then the segment between those two width points is empty. See examples to understand it fully.
 - **Squared**: Same as Rounded but using square tip.
 - **Peak**: Same as Rounded but using peak tip.
 - **Flat**: Same as Rounded but using flat tip.

The Width Point list has one internal non-animatable parameter called **loop**. You can reach it by right clicking the Width Point List parameter. If the Width Point list is unlooped, then any width point that has a Position outside the range of [0,1] is clamped to (brought within) that range. For example: a Position = 1.35 is clamped to 1.0 then the Width Point List is unlooped. Otherwise, if the Width Point List is looped and a width point has a Position of 1.3, its modulus based on the range [0,1] is used, so it is turned to a position of 0.3.

Fast

... TODO WriteME.about...

Dashed Outline

... TODO WriteME.about...

7.1.5 Working with the Advanced Outline

Creation of the Advanced Outline

You can create Advanced Outlines by various ways:

- With the *Draw Tool* (check *Create Advanced Outline* in the tool options).
- With the *Spline Tool* (check *Create Advanced Outline* in the tool options).
- Using a *Layer* (check *Create Advanced Outline* in the tool options).
- With the menu “<Insert> → <Layer> → Geometry → Advanced Outline”.

Change the width of the Width Points

Initially the width Handles are hidden. You can make them visible by pressing (Alt+5) or clicking on the width toggle button. It is possible to change the width using the *Width Tool* using the same procedure as for regular outlines. If you want more control over the width you can modify the width Handles with the *Transform Tool* (Alt+A). More fine tuning is possible by expanding the width point sub-parameter and entering a specific value for the Width sub-parameter. In that case negative values are allowed to produce nice effects.

Change the position of the Width Points

When you make the width Handles of the width points visible/invisible, the position Handles also become visible/invisible. This way, the user has a single way to hide/show the position and width Handles.

Note: This may change in the future

The position Handles of the width points can be modified using the *Transform Tool* (Alt+A) and clicking and dragging the position Handle. You will notice that Handles are tied to the Spline so once clicked and dragged they can be placed at any way on the Spline. Notice that if you have a width point position Handle at position 0.2 and you click and drag at position 0.9 it may happen that you obtain a value of -0.1 because you dragged it in one step and the Width Point List may be looped. If you want to avoid those problems do the movement in small steps to indicate the correct path to follow when calculating the new position.

Adding or removing width points

To add a new width point you have to right click on the width point position Handle (purple Handle) to get the context menu. Then select *Add Item (smart)* here to create more width points entries. The width points are created this way:

In the general case, the new widthpoint is created between the width point you click on and the “previous” width point. Depending on the loop status of the Spline, the “previous” width point can be the start of the Spline (unlooped) or the last widthpoint (looped). The worst case is when there is only one width point on the Width Point List. If you add one new item it will lie over the existing one.

The newly added width point will have the interpolated width at the position where it is created.

Specific actions for Width Points

Some needed actions have been added to width points as well as functionality for existing actions for items of the List type parameters. You can reach the list of available actions by right clicking on the width point position or directly on the width point item of the Width Point List. Here are some descriptions of the available actions:

- **Convert:** Width Points are composite Value Nodes so its natural format is Composite (you can access its components). But you can convert it to other formats. See [Convert](#) for details.
- **Disconnect:** This will disconnect the width point item from the Composite type Value Node. That means that there won't be any Handle to show and that any of its values can be modified. Maybe it is useful for some types of workflows. To restore its Composite status, choose [Convert->Composite](#).
- **Insert/ Remove Item Smart:** This will add or remove a width point. If Remove is used in animation mode, it will also set the current width point as OFF instead of effectively removing it from the Width Point List. Opposite for the [Insert Item Smart](#).
- **Mark Active Point as ON/OFF:** You can set width points off to make it not count for width control while the width point is off. When the width point is half on to off or viceversa, the used width point should be the interpolation between the on and off status. But it doesn't work properly at the moment. On / Off values are not interpolated but fully on or fully off. This has to be corrected.
- **Export:** See [Export](#).
- **Loop:** When set as looped, as mentioned before, it allows the width point positions loop around the range [0,1].
- **Rotate Order:** Doesn't do anything.
- **Set Side Before/After to:** Interpolate, Rounded, Squared, Peak, Flat. Those are shortcuts to do the same as going directly to the sub-parameter and choosing the appropriate Tip Type. Maybe it should be renamed to "Set Tip Before/After".
- **Set Width to default/zero:** those actions have been added for two reasons. It is very common to want to set the width of a width point to zero. Despite the possibility of use the User Preference of Restrict Radius to First Quadrant, this way it is acceded quickly from the same usual menu. Also when a width point position Handle and a width point width Handle are on top of each other (width = 0) it is hard to modify the width or the position without trouble. To solve this the position Handle has preference over the width Handle in case of coincidence. So access the width Handle it is needed to have a quick action to set it to its default value (1.0).

7.2 Bevel Layer



7.2.1 About Bevel Layer

Bevel layer is used to create a highlight and shadow over a shape. It pretends to give some volume to the shape by adding a dark area on the opposite place of the light direction and a light area on the light direction side.

7.2.2 Parameters of Bevel Layer

The parameters of the Bevel Layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Blur Type</i>	Fast Gaussian Blur	integer
 <i>Hi-Color</i>		color
 <i>Lo-Color</i>		color
 <i>Light Angle</i>	135.00°	angle
 <i>Depth Of Bevel</i>	12.000000pt	real
 <i>Softness</i>	6.000000pt	real
 <i>Luma</i>		bool
 <i>Solid</i>		bool

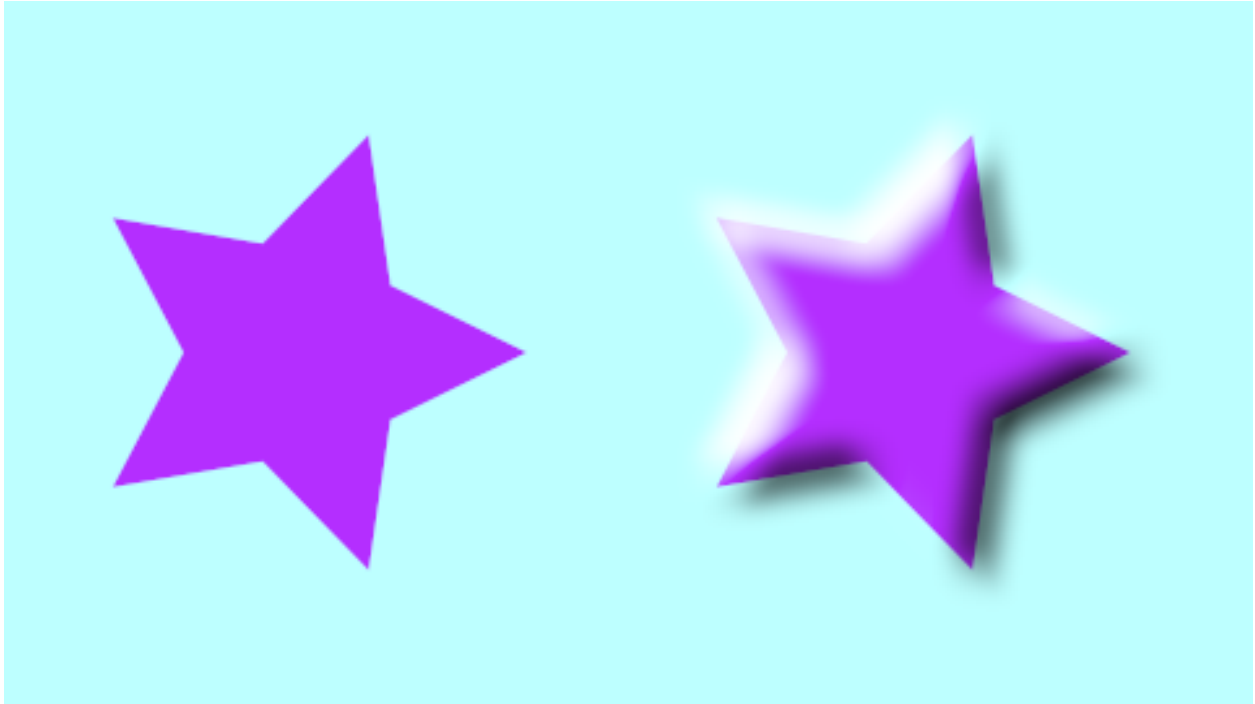
7.2.3 Specific Parameters for Bevel Layers

Only particular parameters of the Bevel Layer are described here. The other parameters are common for other layers and are explained separately.

The Bevel Layer produces two effects over the context of the layer that are based on the Alpha channel of each pixel of the context. Imagine that the Bevel Layer traces light rays from the light source. When the ray line finds a transition from transparent to solid it creates a light region (Hi-color). When the ray finds a transition from solid to transparent it creates a shadow region (Lo-color). The Light Angle parameter determines the source of the light. The Depth of Bevel determines how wide is the region of light or shadow created and the Softness parameter produces a feathered light or shadow region instead of a sharp one. The feather type is determined by the Type parameter and can be one of the known *Blur Layer: Types*.

Light Angle

Modifying the light angle changes the regions that are high lighted and darkened. It is an angle parameter and it is defined from the X positive axis counter clock wise.



Depth of Bevel

The Depth of Bevel parameter determines the width or expansion of the bevel on both sides (dark and light side). The bigger the parameter is the softer the produced bevel is.

Softness

Softness parameter controls how feathered is the bevel. Its type of feather is controlled by the Type parameter.

Use Luma

Please write me.

Solid

Please write me.

7.2.4 Samples

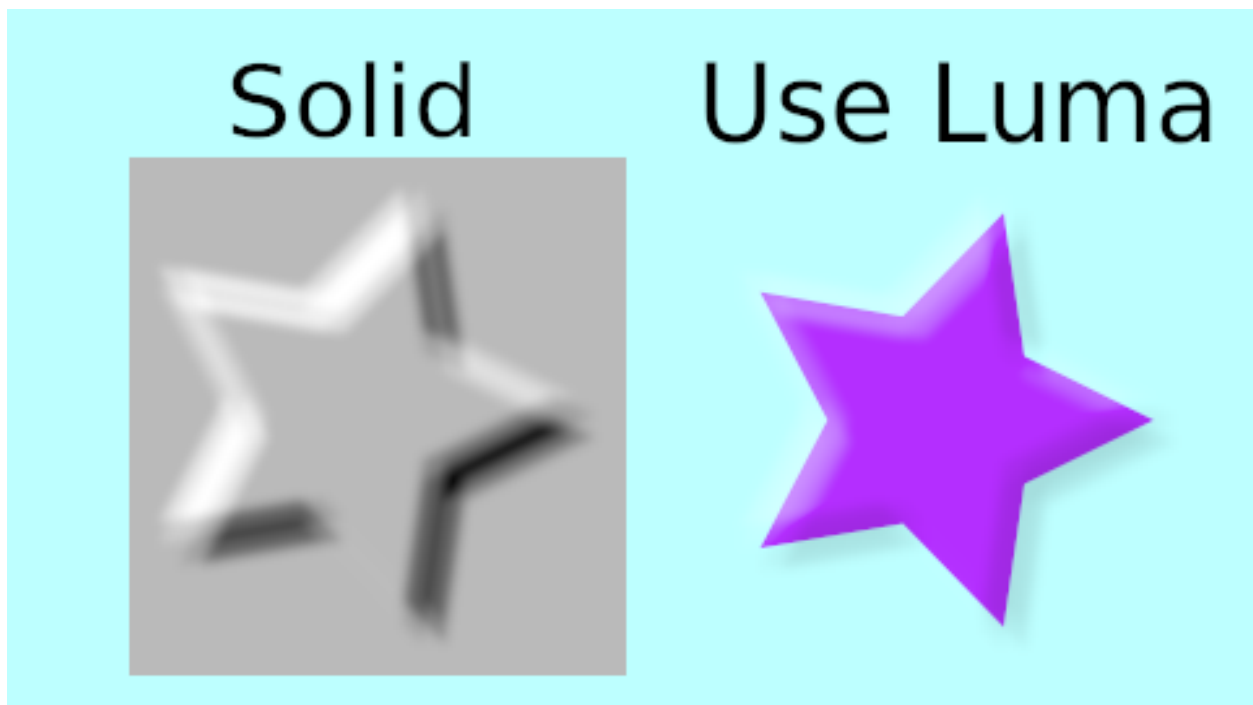
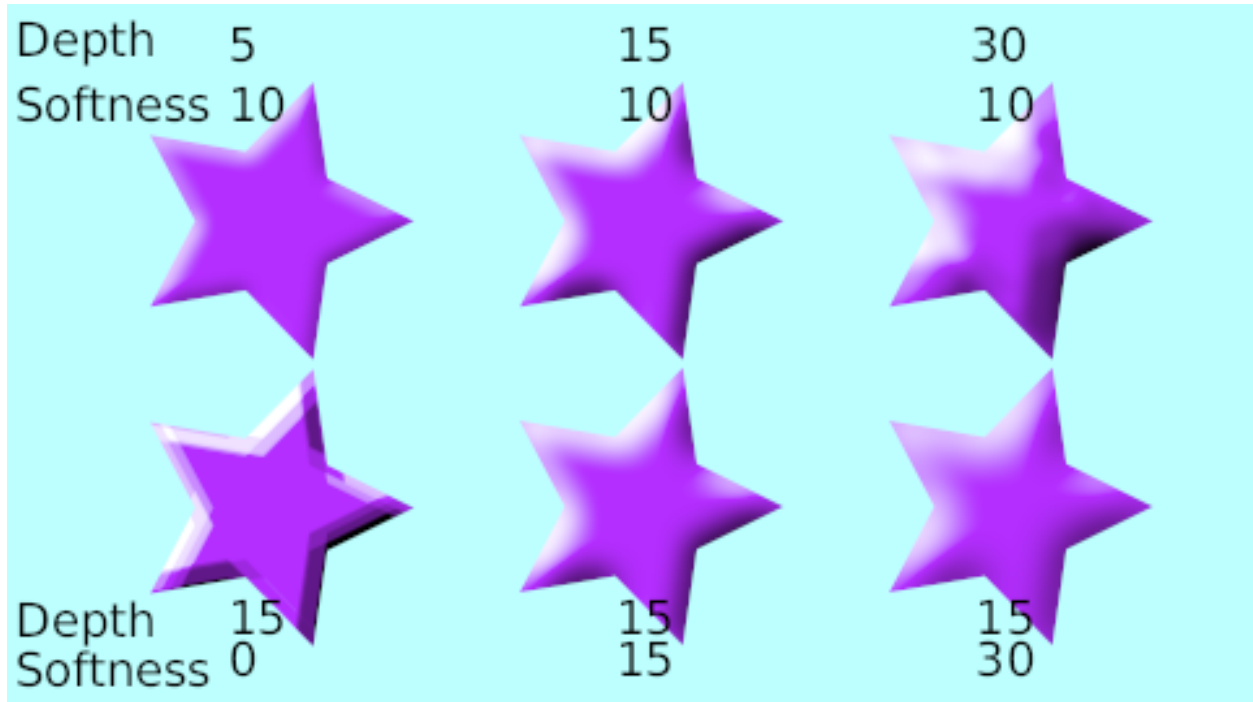
Here you can find some visual examples of how the parameters affect to the bevel. Unless specified, all parameters are the default ones:

The Blend Method has been set to *Onto* in the following image

`Bevel-sample1.sifz`

Here you can see a sample usage of *Solid* and *Use Luma* parameters:

`Bevel-sample2.sifz`



One usage of Solid parameter combined with Hard Light Blend Method in the resulting bevel over the gradient.



Bevel-solid-hardlight.sifz

7.3 Blur Layer



7.3.1 About Blur Layer

Blur is a graphical effect that aims to imitate an unfocused image. In a photography context blur can occur when the [focal point](#) of the lens is not at the right position of the target of the photograph. Also blur can have other origins. Blurs can happen as the result of a poor exposure during its recording (too long exposure time for a large movement of the object or a high movement of the recorder). Also blurring occurs when you use a very wide [aperture](#) (f4). In that case the objects that are out of focus are very blurred compared to the situation where the aperture is fully closed (f22).

In a graphical context, blurs are effects that imitate those situations (with more or less success).

7.3.2 Parameters of the Blur Layer

The parameters of the Blur Layer are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Size</i>	6.000000pt,6.000000pt	vector
 <i>Type</i>	Fast Gaussian Blur	integer

Size

The Size parameter controls the size of the blur. Depending on that value the blur can have different effects. Note that rendering time quickly increases as you increase the size parameter, so be careful.

This parameter is a compound value of two real numbers, x and y. Each of them indicates the size of the blur along that direction.

Type

There are 5 types of blur:

- *Fast Gaussian Blur*
- *Box Blur* ([More info](#))
- *Cross Hatch Blur*
- *Disc Blur*
- *Gaussian Blur* ([More info](#))

7.3.3 Samples

Star animation

If you wonder how the animations were made ([sample file](#)):

- a *Star Layer* is positioned at the center of the Frame, with an *Outline Layer* of width 0.05u
- the animations have 120 frames, and the *Size* parameter of the blur goes from 0u,0u to 1u,1u
- the *Render Dialog* settings are set to Quality 9 and Anti-Aliasing 1

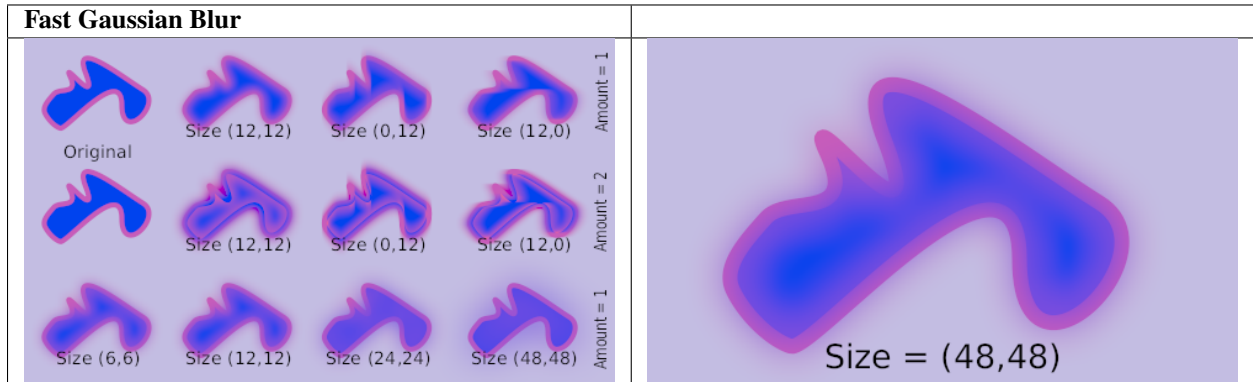
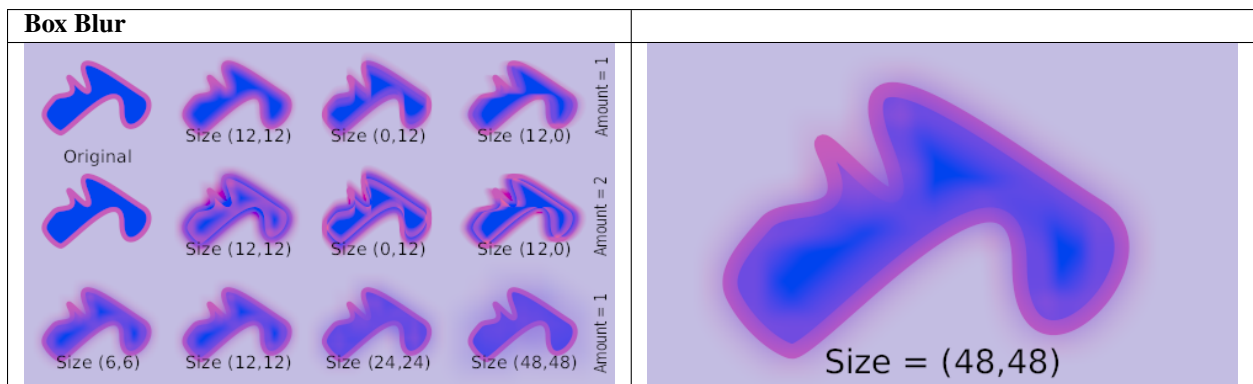
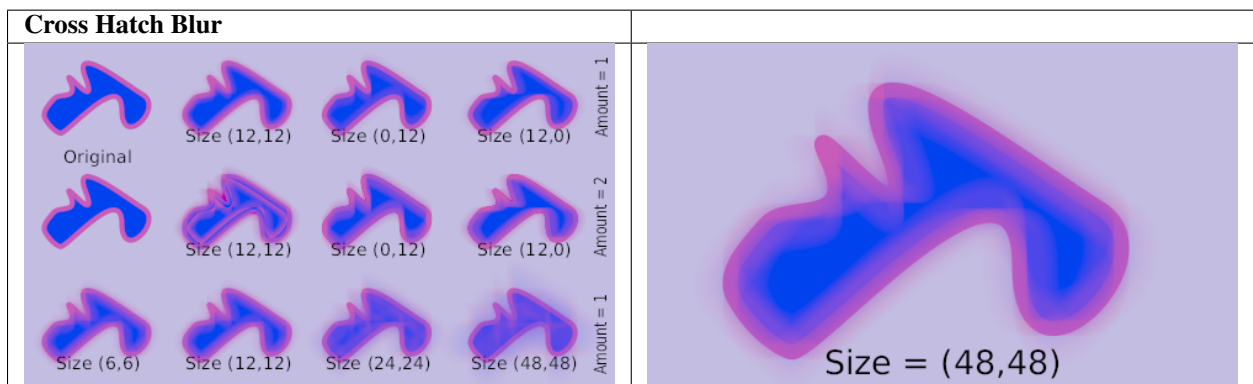
Fast Gaussian :

Box :

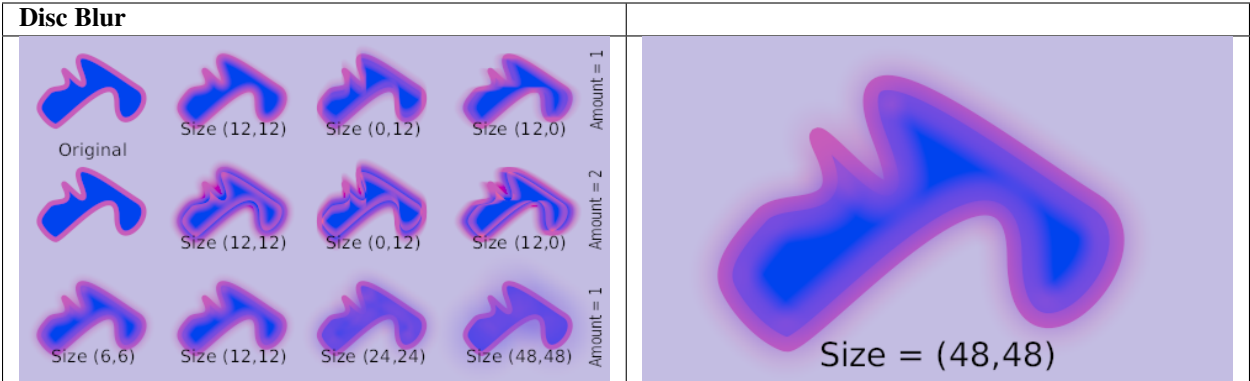
Cross Hatch :

Disc :

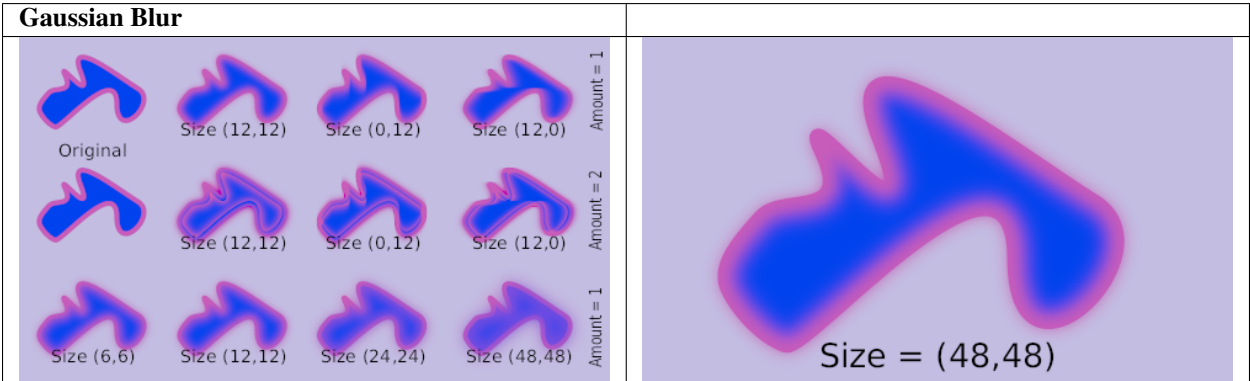
Gaussian :

Fast Gaussian**Fast Gaussian Blur****Box****Cross Hatch**

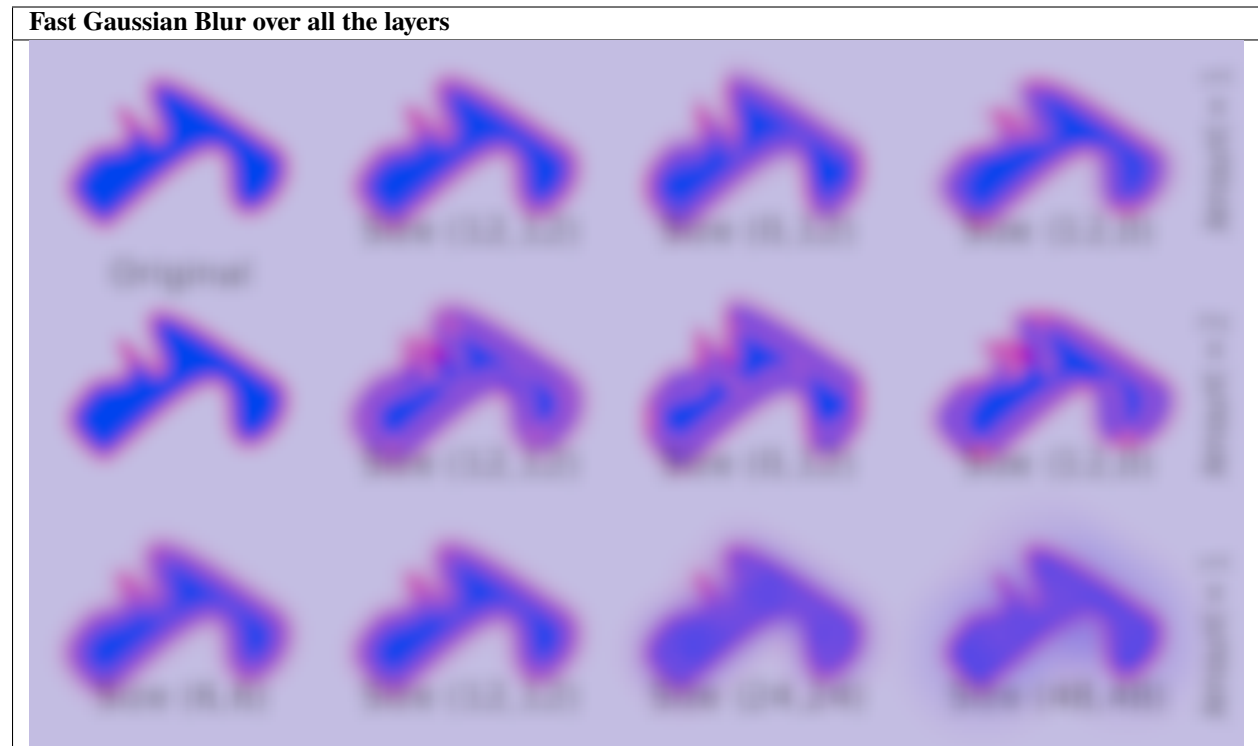
Disc



Gaussian



Fast Gaussian Blur over all the layers



7.3.4 See also

- *Radial Blur Layer* with a radial control of the size of the blur.
- *Motion Blur Layer* a type of blur that aims to simulate a motion blur (the one that happens when the object's movement is too fast for the exposure time). It is controlled by a single parameter called "Aperture".

7.4 Checkerboard Layer



7.4.1 About Checkerboard Layer

Checkerboard layer is used to create... [write me]

7.4.2 Parameters of Checkerboard Layer

The parameters of the Checkerboard Layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 Color Parameter		color
 <i>Origin Parameter</i>	7.500000pt,7.500000pt	vector
 <i>Size Parameter</i>	15.000000pt,15.00000	vector

7.5 Circle Layer



7.5.1 About Circle Layers

Circle layers are used to represent circles =). A circle is the flat region defined by the inner area of a circle. A circle is a flat line on which all the points are at the same distance from the center or Origin. The distance to the center is called Radius.

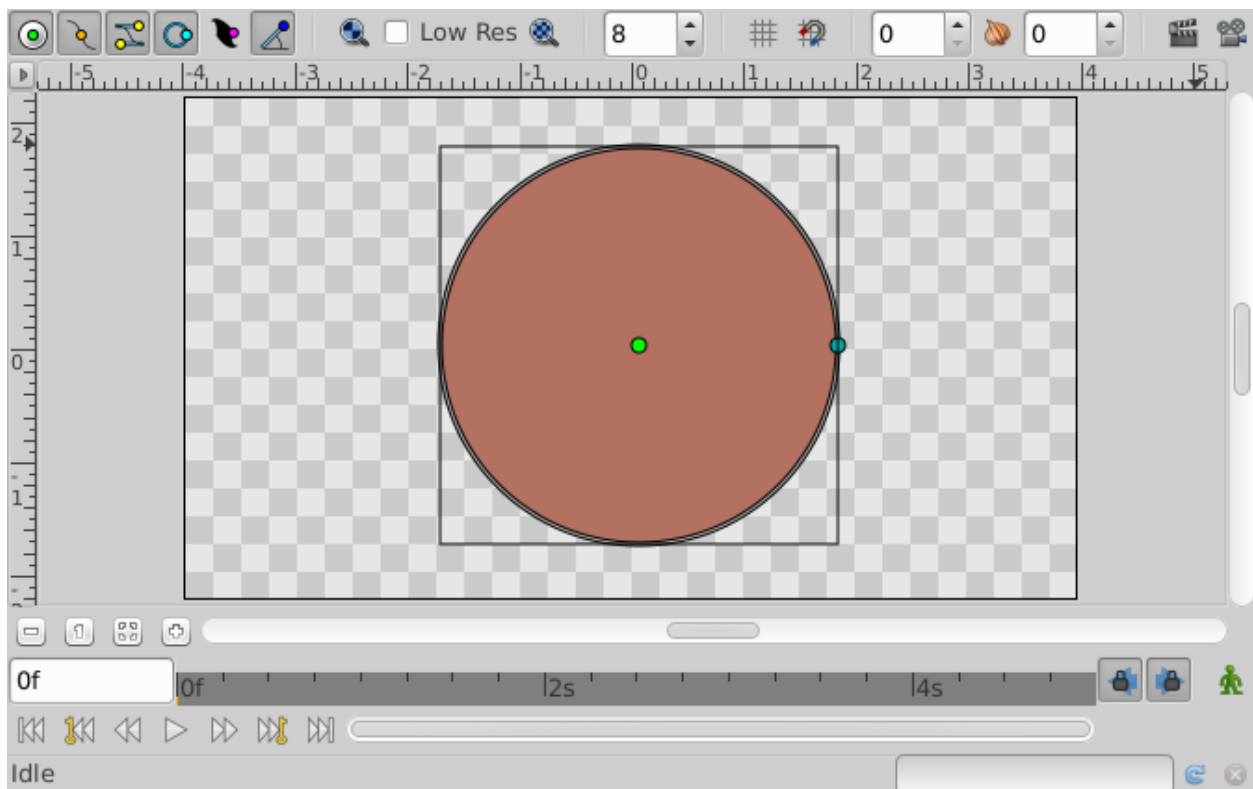
To create a Circle Layer you would use the *Circle Tool*. Make a first left mouse button click where the Origin should go and hold-drag from the center to define the Radius when you release the left mouse button. The circle will be created with the current foreground Color.

7.5.2 Parameters of the Circle Layers

The parameters of the Circle layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Blur</i>	Fast Gaussian Blur	integer
 <i>Color</i>		color
π <i>Radius</i>	59.999998pt	real
π <i>Feather</i>	0.000000pt	real
 <i>Origin</i>	0.000000pt,0.000000pt	vector
 <i>Invert</i>	☐	bool
 <i>Fallof</i>	Linear	integer

A simple Circle Layer in action:



7.5.3 Specific parameters for Circle Layers

Some of the parameters of Circle Layers are common parameter that are shared for several types of Layers. Click on the links to see their definitions. Only the particular parameters for the Circle Layers are described here.

Radius

As mentioned before is is the radius of the geometric circle or circumference. It is modified dragging the cyan handle at the edge of the circumference. Although you rotate it it always becomes horizontal.

Origin

It is a parameter type Vector that defines the center of the circle/circumference. Just edit its x,y coordinates or drag the green handle to change this value.

7.6 Conical Gradient Layer



7.6.1 About Conical Gradient Layer

The *Conical Gradient Layer* gives the sensation of looking down at the tip of a cone, which appears to be illuminated with the background color from a direction determined by the direction of the drawn line.

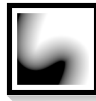
Conical (asymmetric) is similar to Conical (symmetric) except that the “cone” appears to have a ridge where the line is drawn.

7.6.2 Parameters of Conical Gradient Layer

The parameters of the Conical Gradient Layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Gradient</i>		gradient
 Center	0.000000pt,0.000000pt	vector
 Light Angle	0.00°	angle
 Symmetric	☐	bool

7.7 Curve Gradient Layer



7.7.1 About Curve Gradient Layer

As you already guessed, a Curve Gradient is a gradient that follows a curve. One side of the curve will be filled with the color at the beginning of the current gradient (see the [Gradient](#)), and the other side will be filled by the color at the end of the gradient. In other words, the gradient will fully cover your canvas (as the other gradients do), and you'll have to play with the [Blend Method](#) if you want to bind it to a specific shape.

There are two ways to display a gradient: the default one (fast), and an improved one that can remove any render artifacts you have with the first one. To switch from one method to the other, check/uncheck the Fast [Parameter](#) from the [Parameters Panel](#).

7.7.2 Parameters of Curve Gradient Layer

The parameters of the Curve Gradient Layer are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Origin Parameter</i>	0.000000pt,0.000000pt	vector
π <i>Width</i>	5.500000pt	real
 <i>Vertices</i>	List	spline
 <i>Gradient</i>		gradient
 <i>Loop</i>	☐	bool
 <i>ZigZag</i>	☐	bool
 <i>Perpendicular</i>	☐	bool
 <i>Fast</i>	☐	bool

7.7.3 Creating a Curve Gradient

There are several ways to create a gradient that follows a curve.

- With the *Spline Tool*, check **Gradient** in the *Tool Options Panel*. It will create a gradient that follows the curve you draw.
- In the *Canvas: Menu* (from the *Canvas: Menu Caret* or with a right click on a layer in the *Layers Panel*), select *Gradients > Curve Gradient*. It will create a gradient with a default shape. On the *Parameters Panel*, right click on **Vertices**, and select *Convert > Spline*, so you can edit the shape of the gradient.
- To make a Curve Gradient follow an **existing** shape, place the *Curve Gradient Layer* above the layer containing the shape in the *Layers Panel*. Select those two layers, then on the *Parameters Panel*, right-click on **Vertices** (greyed) and select **Link**.
- Alternately, to make a Curve Gradient follow an existing Spline shape (*Outline Layer*, *Region Layer*, or other Curve Gradient), select the shape you want it to follow, right-click its **Vertices** Parameter in the *Parameters Panel* and chose **Export** from the menu. Type a name in the dialog that pops up. Then select the *Curve Gradient Layer* where you want to use this exported shape. Open the *Library Panel*, find the shape you just exported and select it. Then right-click on the **Vertices** parameter of the exported Layer where you want to use the shape and select **Connect** from the menu.

7.7.4 Spreading the gradient

Your Curve Gradient is now created, but it may not look at all like a gradient. Instead it may look like each side of the curve was filled with a solid color. That's because the width of the gradient is too small. Make sure that the *Curve Gradient Layer* is selected, and in the *Parameters Panel*, increase the **Width** value. Your gradient will appear.

An easier way to increase the gradient size is to use the **Width Handle**. In the *Canvas: Menu Caret* select *View > Show/Hide Handles > Show Width Handles*. The **Width Handles** appear in purple on the Canvas. Grab one of them to adjust the width. Each **Width Handle** can be adjusted independently. The width of the gradient will follow the width of the Spline at each point, so it's possible to have the gradient very wide in some places and less wide in others.

7.7.5 Fast Option

Within the *Parameters Panel* of the *Curve Gradient Layer* is a **Fast** option. I'm unsure exactly what speed up it produces, but it does cause a visual artifact when the gradient is constrained with transparency as show in the *Gradient* page, and curve vertices are set to **width = 0**.

The **Fast** option can be turned **Off** in the curve gradient layer's *Parameters Panel*.

7.8 Curve Warp Layer



7.8.1 About Curve Warp Layer

Curve Warp Layers maps the context where it is applied from a straight line to a curved bezier shape. Each point on the source context can be defined by two perpendicular coordinates values based on the position and the length of the source straight line. Those coordinates are like the X and Y coordinates of a scaled and rotated 2D coordinate system.

Then each pair of coordinates is mapped onto the curve considering that the distance from the start point of the curve (the X coordinate) and the perpendicular distance to the straight line (the Y coordinate) are now the length of the spline from the first splinepoint and the perpendicular distance to the curve on that point.

7.8.2 Parameters of Curve Warp Layer

The parameters of the Curve Warp Layer are:

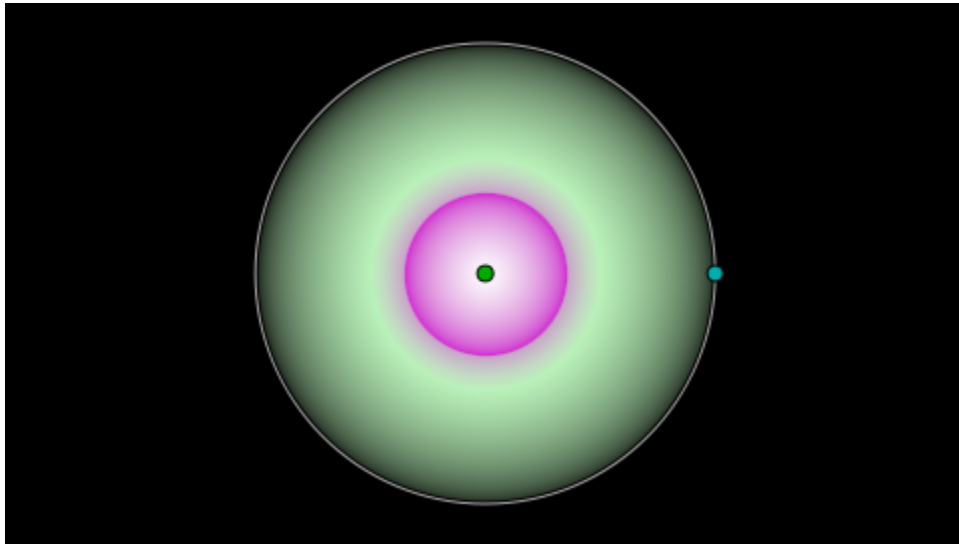
Name	Value	Type
 Origin	0.000000pt,0.000000pt	vector
 Width	5.500000pt	real
 Start Point	-150.0000pt,-30.00000pt	vector
 End Point	150.00000pt,-18.00000pt	vector
 Vertices	List	Spline
 Fast		bool

7.8.3 Using a Curve Warp

Curve Warp layer is inside the Distortion sub menu of the New Layer menu. There is no tool to create this type of layer.

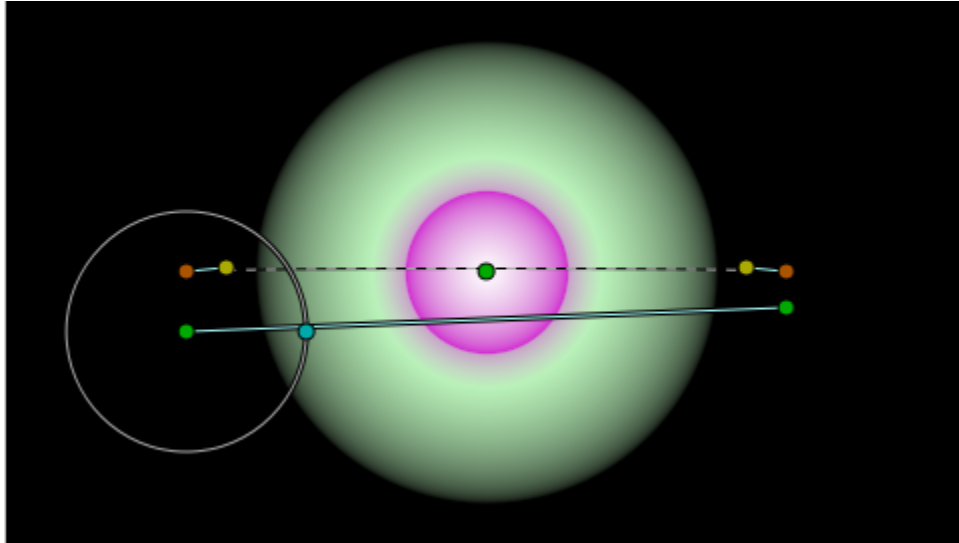
Let's show how to use a Curve Warp layer and what's the effect it produces by a direct example.

I first create one radial gradient at the center of the screen. I modified the default gradient to give me that result.

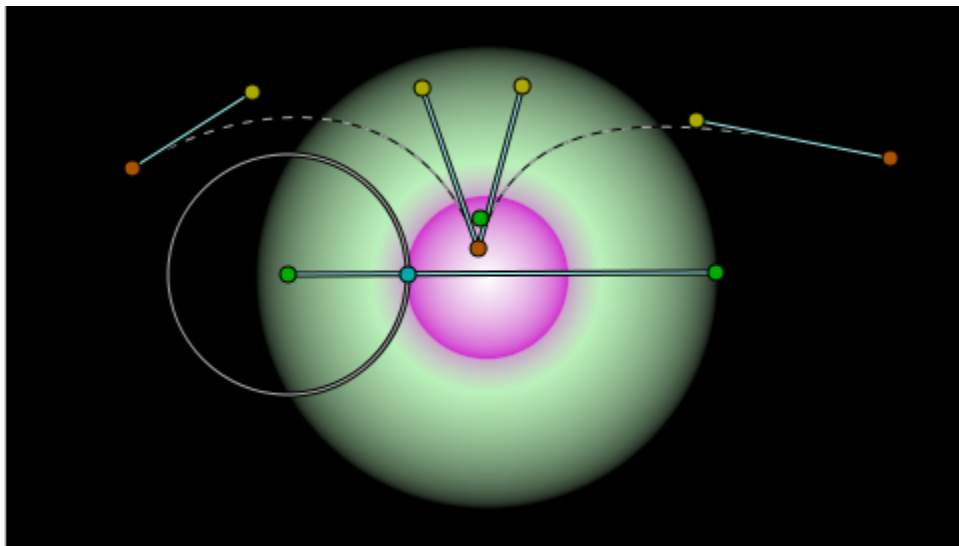


Then we create a Warp Layer over that radial gradient layer.

When you first create the Warp Layer you obtain a almost straight horizontal line defined by two points (below) and a slightly moved spline curve near it (above). I've disabled the curve warp layer on the screen shot to allow to see clearly



the handles of the curve warp layer. The straight line defines the source coordinate system where for the context. The Spline defines the curve that is going to be used to map the context along the curve. Let's edit the curve warp components by placing the straight line along the green area of the gradient and inserting another splinepoint on the curve and tweaking the position of the tangents and vertexes.



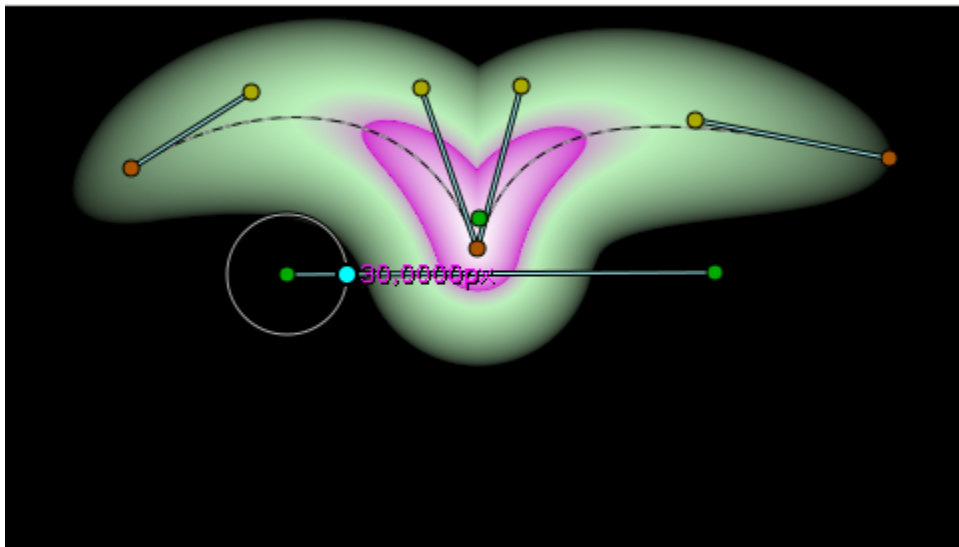
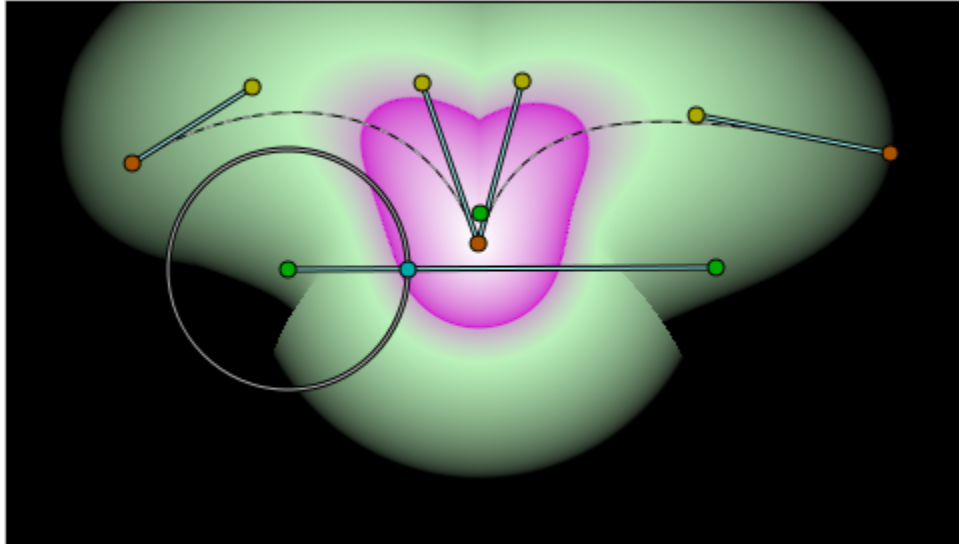
Let's enable the curve warp layer to see the effect.

As you can now understand, the context is read based on the straight line position and mapped along the spline from the first to the last vertex. Due to the curvature of the spline, there are points from the source that are mapped on the same place, and so some kind of render artifact is produced. It can be view like a feature if some kind of special effect is wanted to be achieved.

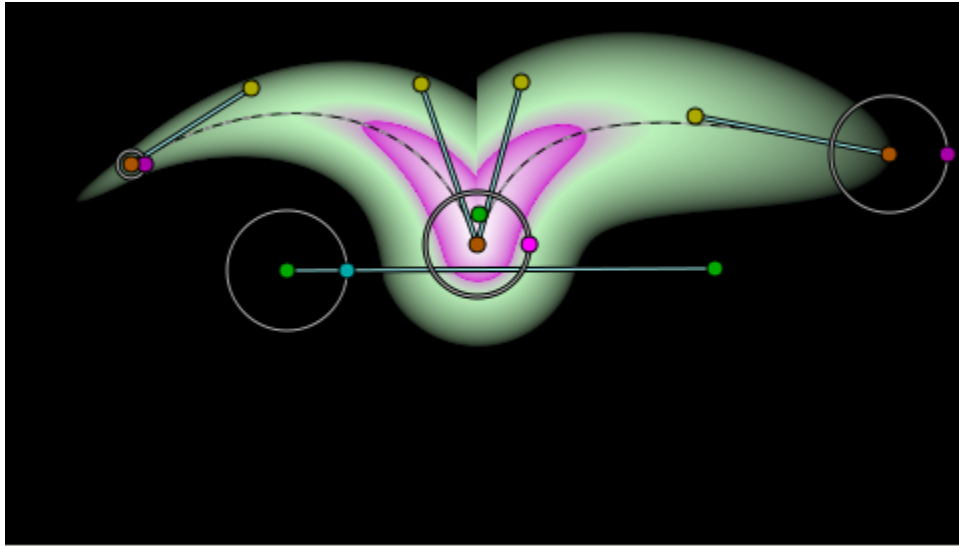
There is one radial handle over the first point of the straight line. It has two missions:

- First, it defines the perpendicular width of the mapped result.
- Second, it is placed on the start point of the straight line.

If we decrease the width the effect is that the whole warp is stretched in the perpendicular direction of the line and the curve.

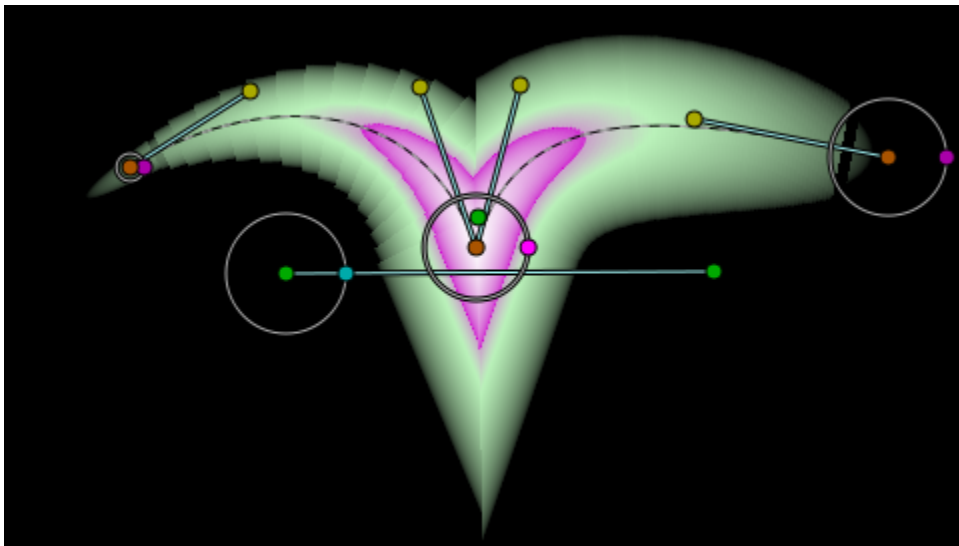


It defines the global width but it is possible to use the local width of the splinepoints to modify the width along the spline. Enable the visibility of the width handles and modify them.



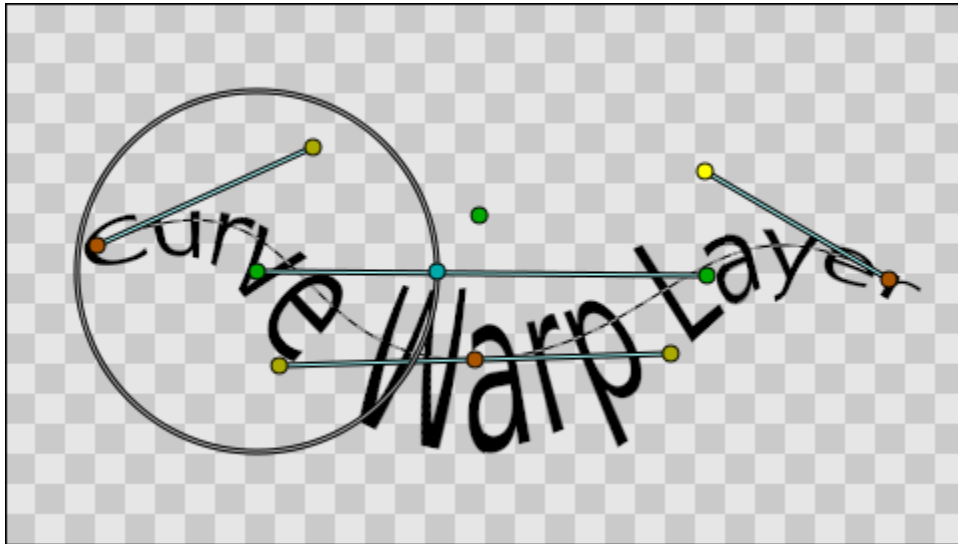
7.8.4 Fast Option

The above screen shots were taken using the Fast option disabled. When enabled, calculations are made in a rough mode what produces more artifacts but it renders faster.



7.8.5 Other examples

Curve warp layer are ideal for curved texts...



Tails, tentacles...

Tail.sifz

7.9 Duplicate Layer



7.9.1 About Duplicate Layer

The Duplicate Layer makes multiple copies of the layers under it in real time.

The Duplicate Layer works like a loop over the content below it and provides a changing variable to that content. This variable (the *Export* Index) can now be used (*Connect*) within that content.

7.9.2 Parameters of Duplicate Layer

The parameters of the Duplicate Layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
π <i>Index</i> 	3.000000	Duplicate

The Index Parameter

The **Index** is automatically *Export*. This is the only **ValueNode** that will change from one copy to the next. This exported value can then be selected in the *Library Panel* and *Connect* to the parameter(s) in the layer under the duplicate dialog which should change in the copies.

The **Index** parameter has 3 sub-parameters, **From**, **To**, and **Step**. The value of the exported **Index** parameter varies from the value of **From** to the value of **To** in steps of size **Step**.

From can be higher or lower than **To**. It doesn't matter whether **Step** is positive or negative. The steps will be in the direction from **From** to **To**.

The duplicated layers are placed in the layer stack in order, so that those corresponding to the **From** value will appear lower down (i.e. least visible with normal composite blend mode) than those corresponding to the **To** value (most visible).

7.9.3 How to use

- *Snowflake with the Duplicate Layer Tutorial* : *Duplicate Layer* + *Rotate Layer*

7.9.4 Known Problems

- The Duplicate Layer doesn't do anything about bounding boxes. Doing so could help to speed up rendering when the duplicated layers are outside the visible area. It's not clear how useful or possible this would be. To calculate its bounding box, the duplicate layer would need to loop through all values of **Index** to get the underlying bounding boxes and union them together. Maybe it's worth doing anyway.
- Editing a Spline below a Duplicate Layer becomes very difficult while a recent edit is still being rendered, because the Spline Handles move around as the render runs (if the duplications are at different positions or scale). I tried using the same mutex around the Duplicate **ValueNode**'s `operator()` method as is used in the Duplicate Layer's code, but it lead to a *deadlock*.

7.10 Group Layer



7.10.1 About Group Layer

The **Group Layer** is a special layer that can hold other layers. It is generated via the *Group* command accessed via the context menu in the *Layers Panel* or through the in the *Canvas: Menu Caret*.

As well as grouping a set of layers, it can also apply transformations to the contained layers like translation, scaling, and even modify their time offset.

A **Group Layer** can also be created through the *New Layer Menu*, using *New Layer* -> *Other* -> *Group Layer*.

7.10.2 Parameters of Group Layer

The parameters of the Group Layer are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Origin Parameter</i>	0.000000u,0.000000u	vector
<i>Transformation</i>	0.000000, 0.000000, 0.00°, 60, 60	transformation
 <i>Canvas</i>		canvas
 <i>Speed</i>	1.000000	real
 <i>Zoom Parameter</i>	0.000000	real
 <i>Time Offset</i>	Of	time
 <i>Lock Selection</i>	☐	bool (Static)
 <i>Focus Point</i>	0.000000u,0.000000u	vector
 <i>Outline Grow</i>	0.000000	real
 <i>Z Range</i>	☐	bool
 <i>Z Range Position</i>	0.000000	real
 <i>Z Range Depth</i>	1.000000	real
 <i>Z Range Blur</i>	0.000000	real

Transformation Parameter

The **Transformation** parameter is a composite parameter, it holds an Offset, a rotation Angle, a Skew Angle and a Scale value for the group. The **Transformation** parameter is already *Convert* when the *Group Layer* is created and so directly exposes the Offset Parameter, Rotation Parameter, Skew Parameter and Scale Parameter subparameters.

See also *Group Transformation Widget*

Canvas Parameter



This is “Group” by default if the *Group Layer* was created by grouping other layers, or *No Image Selected* if it was created from the *New Layer*. This parameter lets you select another canvas.

The Canvas parameter presents a drop-down menu of the exported canvases, plus an extra entry called “Other...”. Selecting “Other...” presents the user with a text entry box asking for the name of the canvas to use. The name typed should have the following format (where [] indicates an optional part, () is for grouping, and * means “0 or more times”):

```
[[filename]#[[:id(:id)*
```

In its simplest form, this is just an *id*, ie. the exported name of one of the child canvases of the current canvas.

Other possibilities are:

- if a '#' is present, the part before the '#' is interpreted as the filename of an external .sif file to use.
- if the '#' is the first character of the string (ie. the filename is blank) then the '#' is ignored, and the current canvas is used instead
- if a ':' appears before the first *id*, it means to start at the root canvas of the current canvas
- each subsequent :id steps down into the specified child

Examples:

- **/usr/share/doc/synfig/examples/business_card.sifz#:IndividualCard** – gives the absolute path to a .sifz file, and says to use the canvas that was exported from its root canvas as “IndividualCard”
- **../examples/business_card.sifz#:IndividualCard** – the same, but with a relative path to the .sifz file
- **#:sy:head:eyes:left** – look in the current composition, and starting from the root, navigate down through the canvas tree. Find a child canvas of the root canvas called 'sy', look in 'sy' for a child canvas called 'head', and so on.
- **:sy:head:eyes:left** – exactly as above. an empty filename is the same as not using the '#' at all
- **eyes:left** – without a ':' before the first *id*, this starts at the current canvas (presumably the Group in question is in the “head” subcanvas of the “sy” subcanvas of the root)

Outline Grow Parameter

This parameter allows to control thickness of all outline layers inside. Assigning positive value to this parameter makes all child outlines rendered thicker, while negative value makes them look thinner. This feature is very helpful for tuning outlines in complex artwork and it also allows to achieve some nice effects like constant outline width at any zoom level.

Note: The Outline Grow parameter can not be applied to exported and imported (external) Groups layers.

Z Range parameters

Z Depth Range Enabled

When checked, only layers inside range are visible and the visible layers are signaled with bolded font in the *Layers Panel*

Z Depth Range Position

Starting Z Depth position where layers are visible

Outline Grow = 0.0



Outline Grow = 1.0



Artwork by Eleonora Pala & Nikolay Mamashev

Z Depth Range Depth

Depth where layers are visible in Z Depth range

Z Depth Range Transition

Z Depth area where layers inside are partially visible.

Example of Z Range use : <http://www.youtube.com/watch?v=UPpmOz0wUY4>

Speed

Alters the velocity group's child animated layers (not the `Group Layer` itself)

Set it to zero, it would stop the animation. A value of one makes the animation run at normal speed and a value of two makes it run double speed. Minus one reverses the animation.

7.10.3 See Also

Switch Group Layer

7.11 Linear Gradient Layer



7.11.1 About Linear Gradient Layer

FIXME!!!

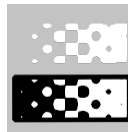
A short description of the layer can be found here: *Linear Gradient Layer*

7.11.2 Parameters of the Linear Gradient Layer

Parameters of the Linear Gradient Layer are:

Name	Value	Type
π Z Depth	0.000000	real
π Amount	1.000000	real
 Blend Method	Composite	integer
 Point 1	1.000000,1.000000	vector
 Point 2	-1.000000,-1.000000	vector
 <i>Gradient Editor Dialog</i>		gradient
 Loop	☐	bool
 ZigZag	☐	bool

7.12 Luma Key Layer



7.12.1 About Luma Key Layer

FIXME!!!

The Luma Key layer does the following: For every pixel of image it assigns the alpha value, based on its Luminosity. For example, black pixels become 100% transparent. White pixels are absolutely opaque. Gray pixels are semi-transparent.

7.12.2 Parameters of the Luma Key Layers

Parameters of the Luma Key Layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer

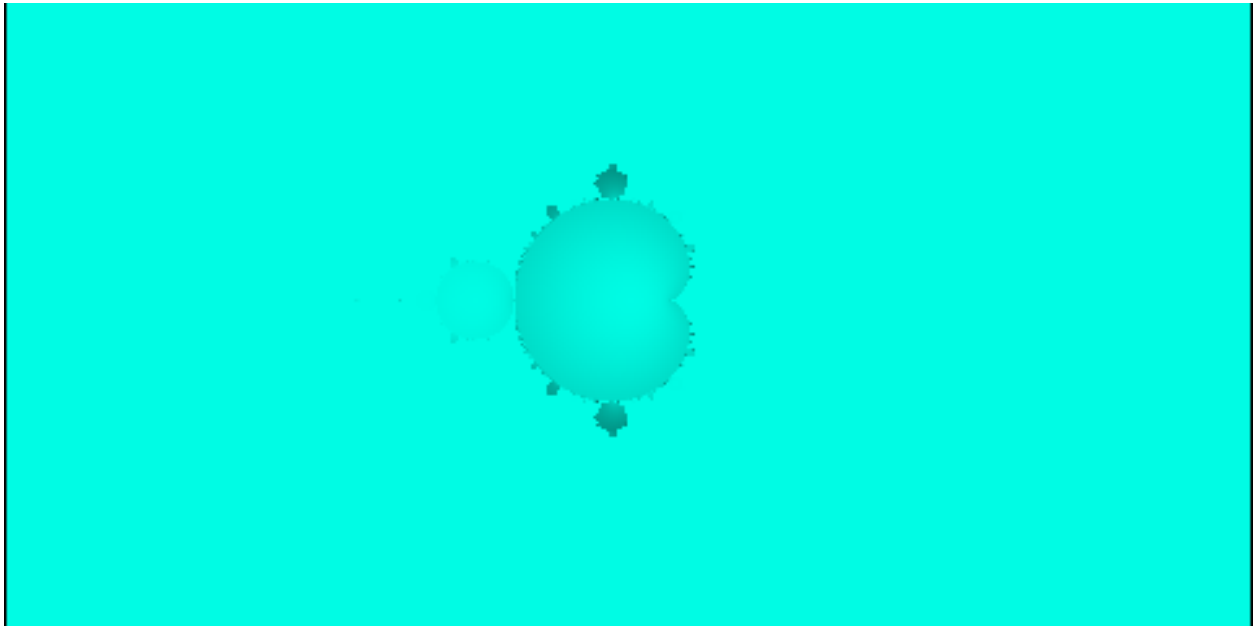
7.13 Mandelbrot Set Layer



7.13.1 About Mandelbrot Set Layer

The [Mandelbrot set](#) is a compact set, since it is closed and contained in the closed disk of radius 2 around the origin.

7.13.2 Sample



7.13.3 Parameters of the Mandelbrot Set Layers

Parameter of the Mandelbrot Set Layers are:

Name	Value	Type
 Iterations_Parameter	256	integer
π Bailout_ValueBase_Parameter	2.000000	real
 Break_Set_Parameter	☐	bool
 Distort_Inside_Parameter	☐	bool
 Shade_Inside_Parameter	☒	bool
 Solid_Inside_Parameter	☐	bool
 Invert_Inside_Parameter	☒	bool
 Gradient_Inside_Parameter		gradient
π Offset_Inside_Parameter	0.000000	real
 Loop_Inside_Parameter	☒	bool
 Distort_Outside_Parameter	☐	bool
 Shade_Outside_Parameter	☒	bool
 Solid_Outside_Parameter	☐	bool
 Invert_Outside_Parameter	☒	bool
 Gradient_Outside_Parameter		gradient
 Smooth_Outside_Parameter	☐	bool
π Offset_Outside_Parameter	0.000000	real
π Scale_Outside_Parameter	0.000000	real

7.14 Metaballs Layer



FIXME!!!

7.14.1 About Metaballs Layer

Metaballs is effect of the interaction of the circular surfaces – balls. Every from balls has any radius from center on the vertex of the curve and the weight.

7.14.2 Parameters of the Metaballs Layers

The parameters of the metaballs layer are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Gradient</i>		gradient
 Balls Parameter	List	list (Dynamic List)
 Radii Parameter	List	list (Dynamic List)
 Weights parameter	List	list (Dynamic List)
π Gradient Left Parameter	0.000000	real
π Gradient Right Parameter	1.000000	real
 Positive Only Parameter		bool

7.15 Motion Blur Layer



7.15.1 About Motion Blur Layer

Blur is a graphical effect that aims to imitate an unfocused image. In a photography context blur can occur when the [focal point](#) of the lens is not at the right position of the target of the photograph. Also blur can have other origins. Blurs can happen as the result of a poor exposure during its recording (too long exposure time for a large movement of the object or a high movement of the recorder). Also blurring occurs when you use a very wide [aperture](#) (f4). In that case the objects that are out of focus are very blurred compared to the situation where the aperture is fully closed (f22).

In a graphical context, blurs are effects that imitate those situations (with more or less success). Motion Blur Layer allows to simulate the blur of picture, that arises by movement.

7.15.2 Parameters of the Motion Blur Layer

The parameters of the Motion Blur Layer are:

Name	Value	Type
 <i>Aperture</i>	0f	time
π <i>Subsamples Factor</i>	1.000000	real
 <i>Subsampling Type</i>	Hyperbolic	integer
π <i>Subsample Start Amount</i>	0.000000	real
π <i>Subsample End Amount</i>	1.000000	real

Aperture

The **Aperture** parameter indicates the number of frames used to compute the effect of motion. The concept is analogous to the shutter speed or exposure time in photography.

Subsamples Factor

The **Subsamples Factor** affects the number of Subsamples to be computed. Using a high value, the smudge is finer, but the rendering takes longer as it involves much more computations.

Subsampling Type

The **Subsampling Type** parameter indicates the type of Subsampling used between:

- Constant
- Linear
- Hyperbolic

Subsample Start Amount

The **Subsample Start Amount** parameter indicates the value of Subsample at the start. For example, if **Subsample Start Amount** is set to 0, it won't have any blur on start. Otherwise the motion of objects from lower layers begin as a smudge.

Subsample End Amount

The **Subsample End Amount** parameter indicates the value of Subsample at the end. For example, if **Subsample End Amount** is set to 0, the last state of the objects from lower layers will appear without any blur.

7.15.3 Samples

	From Subsample Start 0 to Subsample End 1	From Subsample Start 1 to Subsample End 0
0f-0f		
0f-24f		
24f-24f		
24f-0f		

Notes:

- Rows in table – **Aperture** value from beginning (frame 0) to end (frame 120).
- Columns in table – **Subsample Start Amount** value and **Subsample End Amount** value

Sample file

7.15.4 See also

- A short description of the blur layers can be found here: [Layer](#)
- [Radial Blur Layer](#) with a radial control of the size of the blur.
- [Blur Layer](#) mostly used types of blur.

7.16 Noise Distort Layer



FIXME!!!

7.16.1 About Noise Distort Layer

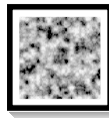
FIXME!!!

7.16.2 Parameters of the Noise Distort Layers

The parameters of the noise distort layer are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Straight	integer
 Displacement Parameter	0.250000u,0.250000u	vector
 Size Parameter	1.000000u,1.000000u	vector
 RandomNoise Seed Parameter	1299635156	integer
 Interpolation Parameter	Cosine	integer
 Detail Parameter	4	integer
π Animation Speed Parameter	0.000000	real
 Turbulent Parameter		bool

7.17 Noise Gradient Layer



7.17.1 About Noise Gradient Layer

Noise Gradient Layer is a type of gradient generated by procedural texture. This method consists of a creation of a lattice of random interpolated gradient.

You can adjust various parameters to create different textures. You can easily founds some clouds...

7.17.2 Parameters of Noise Gradient Layer

The parameters of the Noise Gradient Layer are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Gradient</i>		gradient
 RandomNoise Seed Parameter	1299736373	integer
 Size Parameter	60.000000u,60.000000u	integer
 Interpolation Parameter	Cosine	integer
 Detail Parameter	4	integer
 Animation Speed Parameter	0.000000	real
 Turbulent Parameter		bool
 Do Alpha Parameter		bool
 Super Sampling Parameter		bool

- Size How much separated are two consecutive distortions

7.18 Outline Layer



7.18.1 About Outline Layers

Outline Layers are used to hold strokes or edges of filled shapes. They give a cartoon look to an animation when added as the edge of the filled shapes. Also standalone Outlines (strokes) are used to define fold of 3D drawings and used together with shadows (created using gradients or feathering) are the basis of an artist's work.

To create an Outline Layer use the *Spline Tool* and check **Create Outline** in the *Tool Options Panel*. Once you finish the definition of the geometry of your outline and after pressing the **Make Spline** button (or selecting another tool or state) you create the Outline Layer with the *New Layer Defaults: Brush Colors Outline Color*.

Depending on the options you choose in the *Tool Options Panel* other like *Star Tool* or *Cicle Tool* can also create Outline Layers.

Using the *Canvas: Layer Menu* **Make Advanced Outline**, **Make Outline** or **Make Region** commands you can also create those layers.

7.18.2 Parameters of Outline Layers

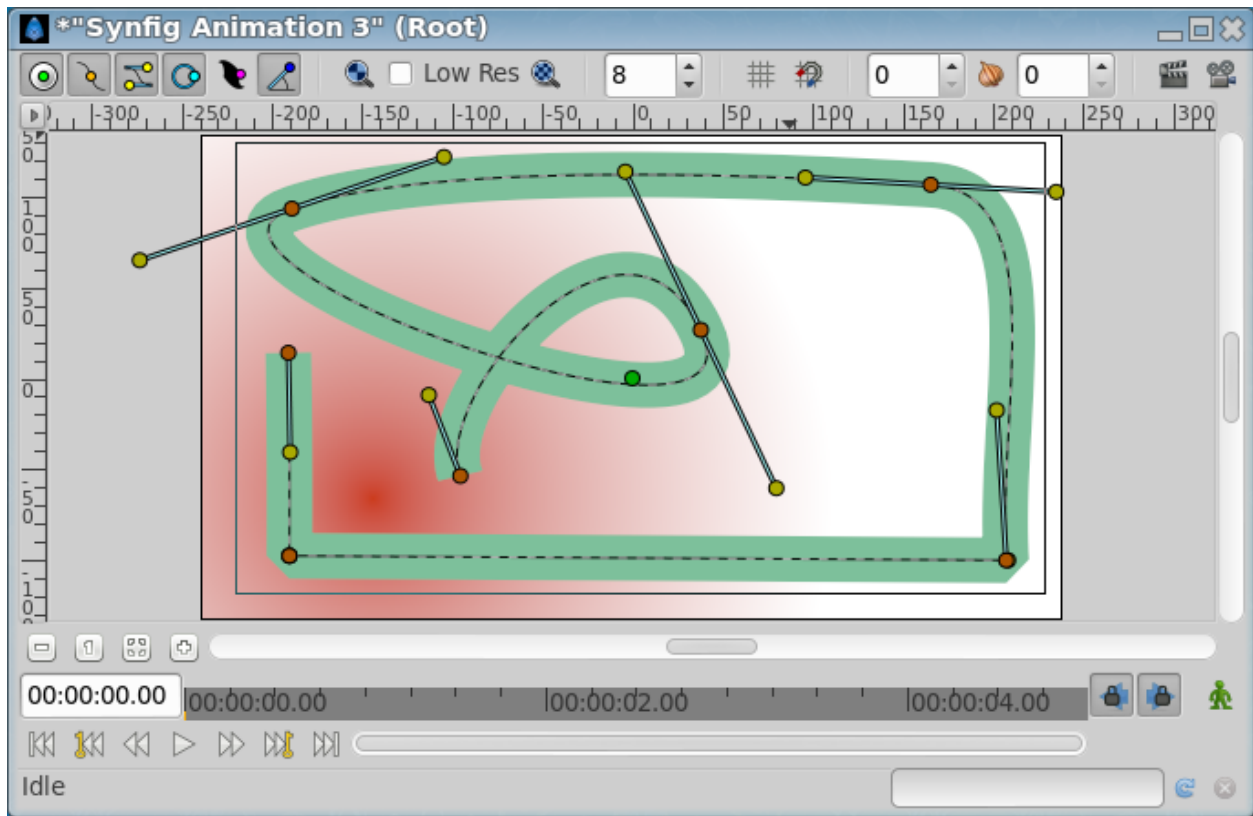
The parameters of the Outline layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Origin</i>	0.000000u,0.000000u	vector
 <i>Invert</i>	☐	bool
 <i>Antialiasing</i>	☐	bool
π <i>Feather Parameter</i>	0.000000pt	real
 <i>Type of Feather</i>	Fast Gaussian Blur	integer
 <i>Winding Style Parameter</i>	Non Zero	integer
 <i>Vertices</i>	List	list (Spline)
π <i>Outline Width</i>	2.000000pt	real
π <i>Expand</i>	0.000000pt	real
 <i>Sharp Cusps</i>	☐	bool
 <i>Rounded Begin</i>	☐	bool
 <i>Rounded End</i>	☐	bool
π <i>Loopyness</i>	1.000000	real
 <i>Homogeneous</i>	☐	bool

7.18.3 Specific parameters for Outline Layers

The first 11 parameters of the Outline are common parameters that are shared for several types of Layers. Click on the links to see their definitions. Only the particular parameters for the Outline Layer are described here.

This is a sample screenshot of an Outline Layer in action:



Outline width

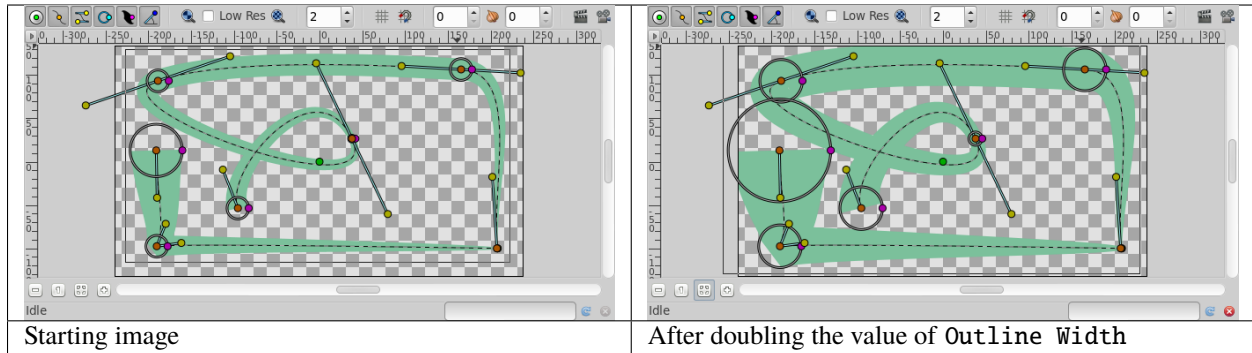
The Outline width parameter is a float value in pixels that represents the basic value of the width for all the points of the curve at the same time. It works together with the width parameter of the vertices' child values. So setting the Outline Width to 1.0px then the resulting value for the rendered width in that vertex comes from this formula: $W = OLW * VW + 2E$.

Where:

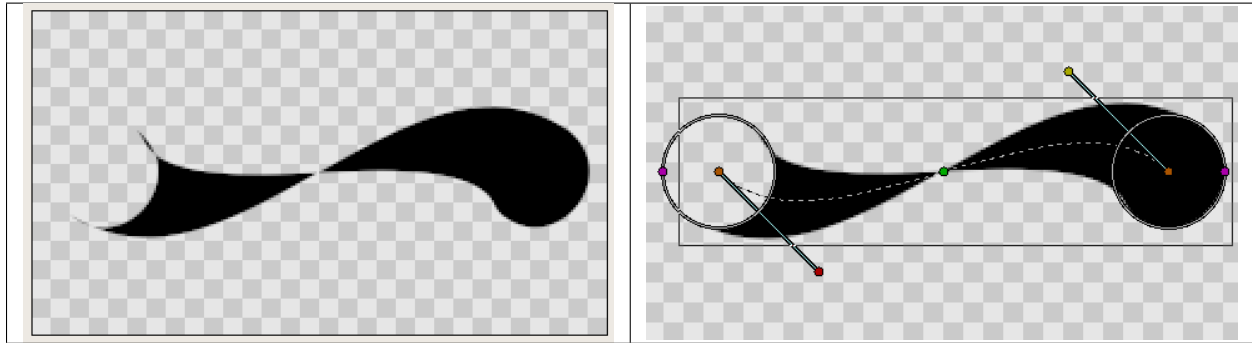
- W = resulting width of the outline at that vertex.
- OLW = overall parameter for the width of all the vertices (that is the Outline Width parameter).
- VW = specific vertex width value.
- E = value of the Expand parameter.

The width *handle* would show the radius value for that vertex ($W/2$).

The width that ends up being used for each vertex is $((VertexWidth * OutLineWidth) + Expand * 2)$:



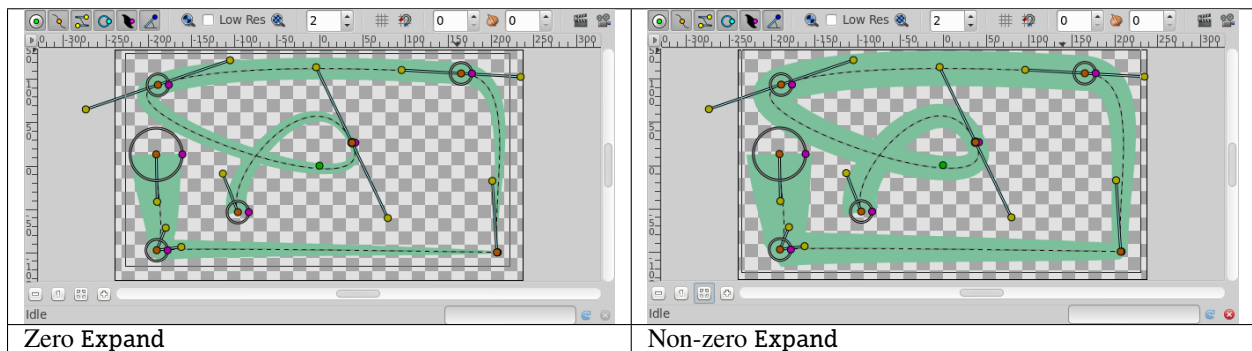
It's possible to specify negative values for the width values, which kind of turns the vertex inside out. Here's an example where the left vertex has a negative width and the right has a positive width. Note how the edges of the spline between the vertices cross each other in the middle:



Expand

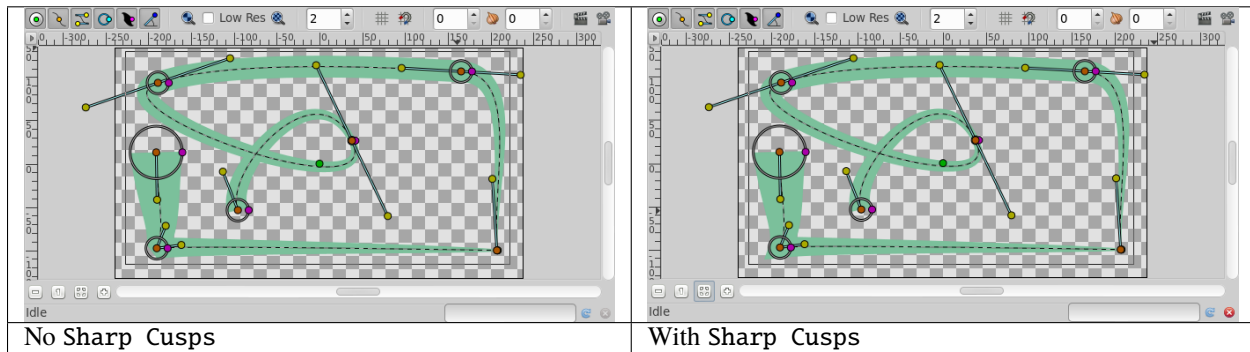
The Expand parameter is similar to the Outline Width parameter, but rather than being multiplied by each vertex's width, it is added to the radius of each vertex. In other words $(2 * \text{Expand})$ is added to the diameter of each vertex.

The width that ends up being used for each vertex is $((\text{VertexWidth} * \text{OutLineWidth}) + \text{Expand} * 2)$:



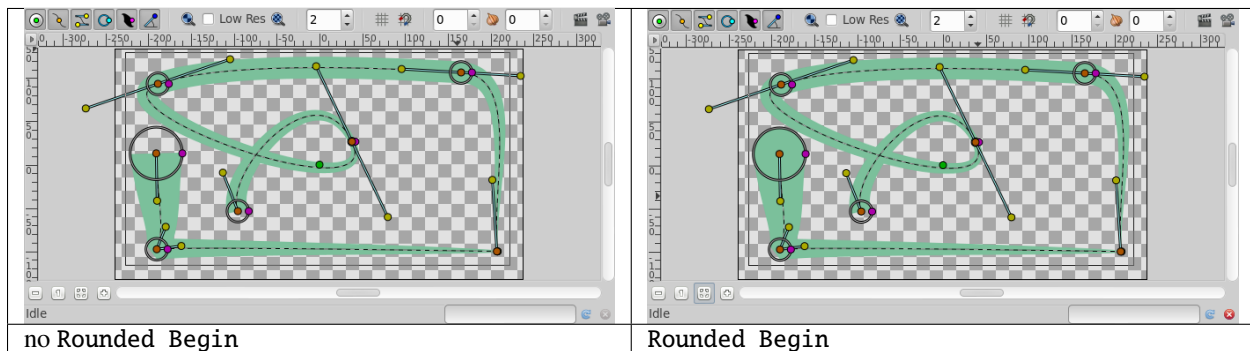
Sharp cusps

Enabling Sharp Cusps makes the corners pointy when the tangents are split:



Rounded Begin

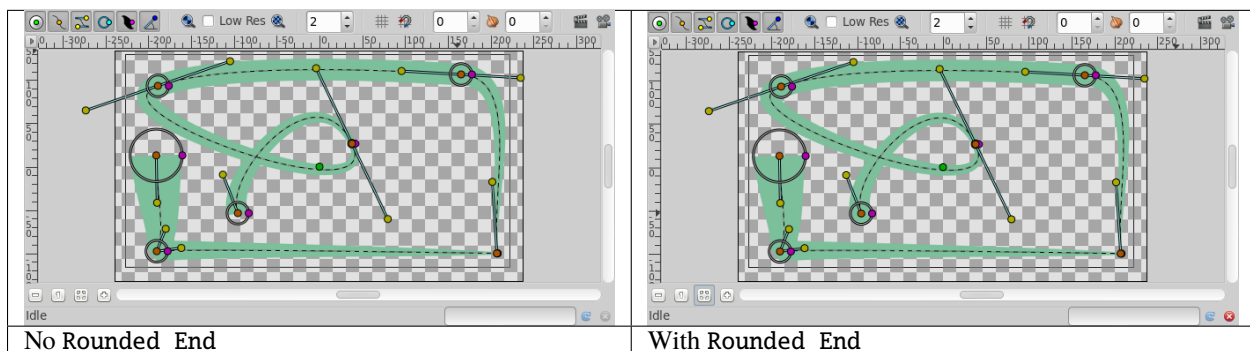
Enabling Rounded Begin makes the start of the outline rounded:



Note: With cairo render activated, if the quality is superior to 6, enabling Rounded Begin enable also RRounded End.

Rounded End

Enabling Rounded End makes the end of the outline rounded:



Note: With cairo render activated, if the quality is superior to 6, enabling Rounded End enable also Rounded Begin.

Loopyness

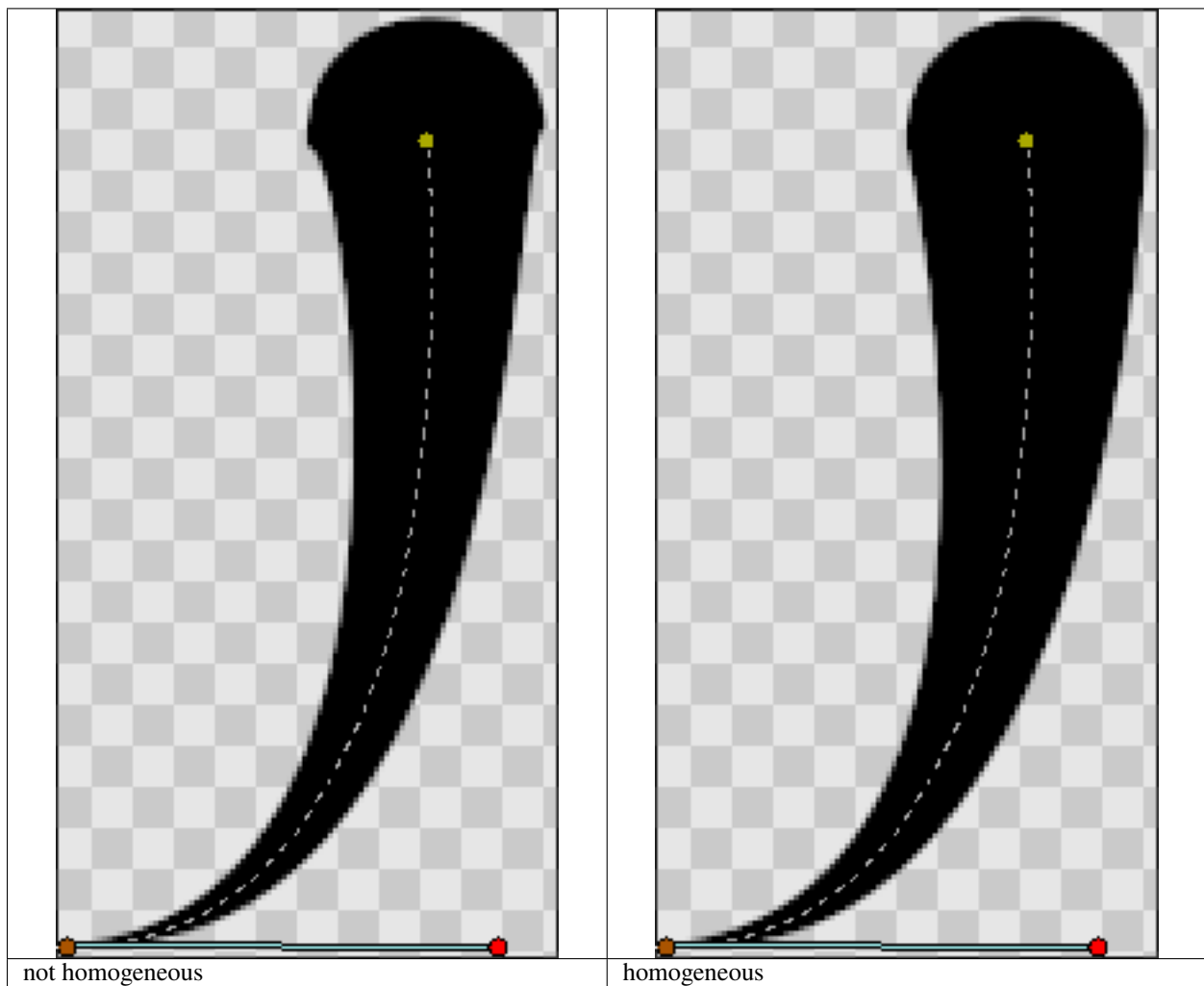
This parameter currently does absolutely nothing at all!

Homogeneous

Enabling 'Homogeneous' changes the way that the width of the outline changes from one spline point to the next. Each point in the spline has its own width, and the spline has its own 'Outline Width' and 'Expand' parameters which are used to give the final width at each spline point. The 'Homogeneous' parameter controls how the width is interpolated between two neighboring spline points:

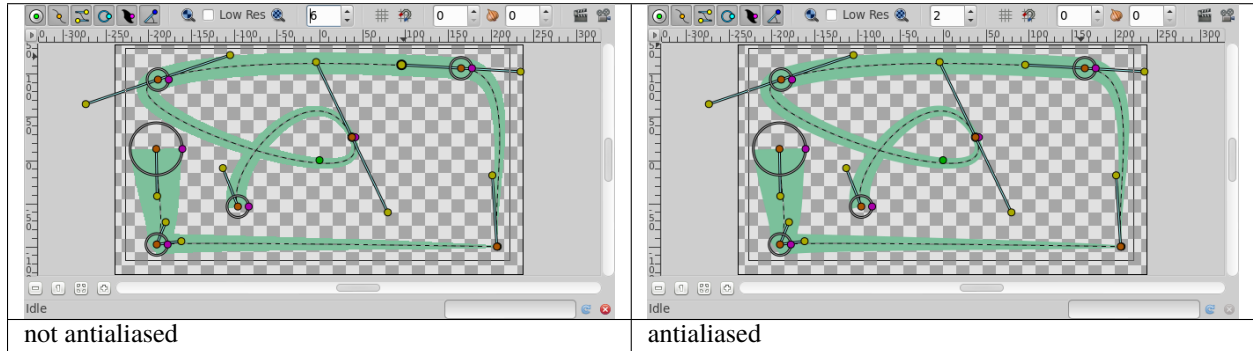
- When 'Homogeneous' isn't checked, the width is interpolated linearly with the spline's 't' parameter.
- When 'Homogeneous' is checked, the width is interpolated linearly with the spline's length.

Turning off homogeneous often makes the fat end of a spline look lumpy. Maybe that's a bug:



Misc

The [Antialiasing Parameter](#) isn't specific to the Outline layer, but it is particularly useful for the Outline layer. It makes the edges appear smoother:



Note : File used

7.19 Plant Layer



7.19.1 About Plant Layers

7.19.2 Parameters of Plant Layers

The parameters of the Plant layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Origin</i>	0.000000u,0.000000u	vector
 Vertices	List	list (Spline)
 Origin	0.000000u,0.000000u	vector
 Gradient		gradient
 Split Angle	10.00	angle
 Gravity	0.000000u,-0.100000u	vector
π Tangential Velocity	0.300000	real
π Perpendicular Velocity	0.000000	real
π Stem Size	0.015000u	real
 Size As Alpha	☐	bool
 Reverse	☐	bool
π Step	0.010000	real
 Seed	1299911698	integer
 Splits	5	integer
 Sprouts	10	integer
π Random Factor	0.200000	real
π Drag	0.100000	real
 Use Width	☐	bool

7.20 Polygon Layer



7.20.1 About Polygon Layer

Polygon Layers are made by the *Polygon Tool*. Their use is deprecated, with *Region Layer* being the preferred layer type.

Note that you can make regular polygons with even numbers of sides using the *Star Layer: 2N-sided Polygons*.

7.20.2 Parameters of the Polygon Layers

The parameters of the Polygon layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Origin</i>	0.000000u,0.000000u	vector
 <i>Invert</i>		bool
 Antialiasing		bool
 <i>Feather Parameter</i>	0.000000u	real
 <i>Type of Feather</i>	Fast Gaussian Blur	integer
 <i>Winding Style Parameter</i>	Non Zero	integer
 Vector List Parameter	List	list (Dynamic)

7.20.3 Examples

Here are some example polygons (in case you forgot what they look like):

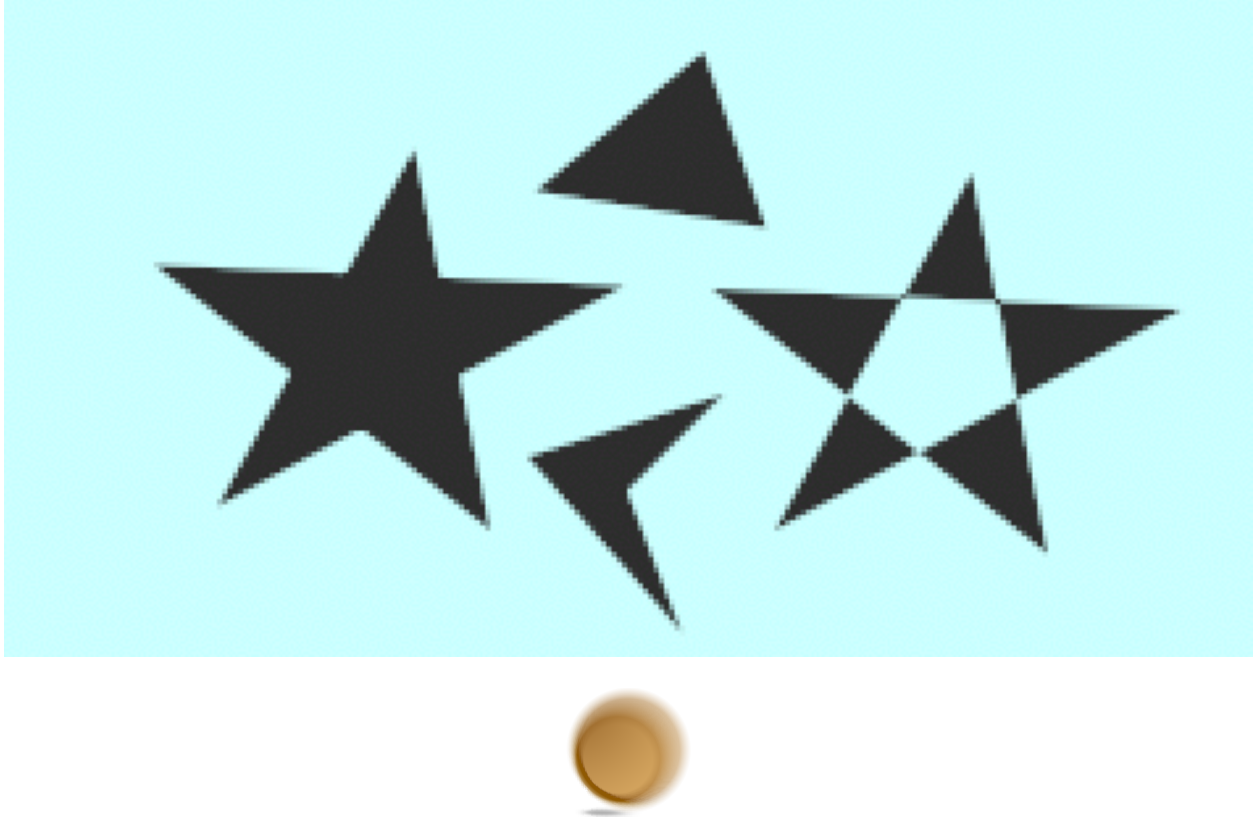
Polygons.sifz

Note that the edges can cross each other. When they do, the *Winding Style Parameter* comes into play.

7.21 Radial Blur Layer

7.21.1 About Radial Blur Layers

Radial control of the size of the blur.



7.21.2 Parameters of Radial Blur Layers

The parameters of the Radial Blur Layer are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Origin</i>	0.000000u,0.000000u	vector
π <i>Size</i>	0.200000	real
 <i>Fade Out Parameter</i>		bool

7.21.3 Samples

If you wonder how the animations were made (sample file):

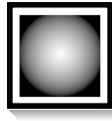
- a *Star Layer* is positioned at center of the Frame, with an *Outline Layer* of width 0.05u
- the animations have 120 frames, and the Size parameter of the blur goes from 0u,0u to 1u,1u
- the *Render Dialog* settings are set to Quality 9 and Anti-Aliasing 1

First sample is without *Fade Out*:

7.21.4 See also

- A short description of the blur layers can be found here: [Layer](#)
- [Blur Layer](#) mostly used types of blur.
- [Motion Blur Layer](#) a type of blur that aims to simulate a motion blur (the one that happens when the object's movement is too fast for the exposure time). It is controlled by a single parameter called **Aperture**.

7.22 Radial Gradient Layer



7.22.1 About Radial Gradient Layers

FIXME!!!

7.22.2 Parameters of Radial Gradient Layers

The parameters of the radial gradient layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Gradient Editor Dialog</i>		gradient
 Center	0.000000u,0.000000u	vector
π Radius	1.000000u	real
 Loop	☐	bool
 ZigZag	☐	bool

7.23 Rectangle Layer



7.23.1 About Rectangle Layers

The Rectangle Layer lets you draw rectangles by specifying two opposite corners.

7.23.2 Parameters of Rectangle Layers

The parameters of the rectangle layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 Point 1	0.000000u,0.000000u	vector
 Point 2	1.000000u,1.000000u	vector
π Expand Parameter	0.000000u	real
 Invert		bool

7.24 Region Layer



7.24.1 About Region Layers

The Region Layer is much like the *Outline Layer*, except that the shapes it describes are filled in, rather than just being an outline.

To create a Region Layer use the *Spline Tool* and check **Create Region** in the *Tool Options Panel*. Once you finish the definition of the geometry of your outline and after pressing the **Make Spline** button (or selecting another tool or state) you create the Region Layer with the *New Layer Default: Brush Color Fill Color*.

Depending on the options you choose in the *Tool Options Panel* other like *Star Tool* or *Cicle Tool* can also create Region Layers.

Using the *Canvas Layer: Menu* **Make Advanced Outline**, **Make Outline** or **Make Region** commands you can also create those layers.

7.24.2 Parameters of Region Layers

The parameters of the region layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Origin</i>	0.000000u,0.000000u	vector
 <i>Inver Parameter</i>	☐	bool
 <i>Antialiasing</i>	☐	bool
 <i>Feather</i>	1.000000u	real
 <i>Type of Feather</i>	Fast Gaussian Blur	integer
 <i>Winding Style</i>	Non Zero	integer
 <i>Vertices</i>	List	list(Spline)

The parameters are exactly the same as in the *Outline Layer*, except that region layers don't have the following 7 parameters, which are all specific to outlines:

- *Outline Width*
- *Expand*
- *Sharp Cusps*
- *Rounded Begin*
- *Rounded End*
- *Loopyness*
- *Homogeneous*

7.25 Rotate Layer



7.25.1 About Rotate Layers

A Rotate Layer change the position of any Primitive Layer below it in the *Layers Panel*. It apply an Angle from his Origin to the resulting calculated points to render the shape, it doesn't modify the values of the underlying layers. Rotate Layer is part of Layers Transform category.

7.25.2 Parameters of Rotate Layers

The parameters of the rotate layers are:

Name	Value	Type
 <i>Origin</i>	0.000000u,0.000000u	vector
 Angle	0.00°	angle

7.26 Scale Layer



7.26.1 About Scale Layers

A Scale Layer scales the size of any Primitive Layer (*Region Layer*, Layer Gradients, etc.) below it. Scale Layer is part of Layers Transform category.

7.26.2 Parameters of Scale Layers

The parameters of the Scale Layers are:

Name	Value	Type
π <i>Amount</i>	1.000000	real
 <i>Origin</i>	0.000000u,0.000000u	vector

The Amount parameter may take the inverse of the exponential function (the natural log), as a raw value. For example:

- the natural log of 2.00 is 0.693147181, which up-scales to 200.0%
- the natural log of 1.00 is 0.0, which scales by 0.0%
- the natural log of 0.50 is -0.693147181, which down-scales to 50.0%
- the natural log of 0.10 is -2.302585092994045, which down-scales to 10.0%
- the natural log of 0.01 is -4.605170185988091, which down-scales to 1.0%

The *Origin* is the center of the scale.

In versions prior to 0.63.05 Scale Layer was called “Zoom layer”.

7.27 Shade Layer



7.27.1 About Shade Layers

The Shade Layer creates a drop shadow underneath objects.

7.27.2 Parameters of Shade Layers

The parameters of the Shade Layer are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Origin</i>	0.200000u,-0.200000u	vector
 <i>Size</i>	0.100000u,0.100000u	vector
 <i>Type</i>	Fast Gaussian Blur	integer
 <i>Invert</i>		bool

7.28 Simple Circle Layer



7.28.1 About Simple Circle Layers

FIXMM!!!

7.28.2 Parameters of Simple Circle Layers

The parameters of the simple circle layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 <i>Center</i>	0.000000u,0.000000u	vector
π <i>Radius</i>	0.500000u	real

7.29 Skeleton Layer



7.29.1 Parameters of Skeleton Layers

The parameters of the skeleton layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	0.500000	real
 <i>Name</i>	skeleton	string
 <i>Bones</i>	list (Static List)	list

7.29.2 Summary

In the skeleton layer, each vertex is influenced by a weighted average combination of bones *maybe from different skeletons?*

There are two ways to make a vector be influenced by bone(s). First is the Link to Bone action that would give a 100% of influence of one bone to one vector. Second is the Link to Skeleton that is a semiautomatic way to give weighted influence to a vector from all the bones that overlaps its influence area over the vector position. The more covered the vector is by the influence area, the higher is the influence weight. In any case you can manually specify the bone and the influence that you want by editing the values in the vector (sub)parameters. We need to modify the bone(s) with handles to perform the indirect movement to the vectors that are influenced by it(them).

7.29.3 Skeleton construction

Add a Skeleton layer :

- If your *Canvas* is empty you can do that from the main or *Canvas: Caret* menus **Layer -> New Layer -> Other -> Skeleton**.
- If your canvas is not empty, right clicking inside the *Layers Panel*.

Once you have a Skeleton layer, the first bone appears. You have created the first bone of your Skeleton, this is your **first parent bone**.

Each bones have the following *Handle* to control them : **Origin**, **Angle**, **Local Length Scale**, **Bone Width** and **Tip Width**. Take a look inside the *Parameters Panel* of the Skeleton layer, those handles are relative to some of those values (with also some advanced parameters).

From the *Workarea*, with the Bones *Handle* you can easily move and adjust part of your Skeleton.

Now, you can had child bones :

- Right click on any *Handle* of the parent bone you want and select **Create Child Bone**.
- Repeat the operation, choosing accordingly the parent bone of your future bone (the parent of a hand bone is the arm, the parent bone of a finger is the hand bone...), to create your Skeleton.

When created, a Skeleton layer, is *Canvas: Layer Menu* : the layer is visible in the *Workarea* but disabled in the final Render.

7.29.4 Parameters of a Bone

The parameters of a bone are:

Name	Value	Type
 Name	Bone 1	string
Parent	No Parent	bone_valuenode
 <i>Origin</i>	0.00000, 0.00000	vector
 Angle	0.00°	angle
 Local Length Scale	1.00000	real
 Bone Width	0.10000	real
 Recursive Length Scale	1.00000	real
 Tip Width	0.10000	real
 <i>Z Depth Parameter</i>	0.00000	real
 Length Setup	1.00000	real

Parent

Angle

Local Length Scale

Bone Width

Has **Tip Width**, **Bone Width** is used to set the influence of the bone before the use of **Link to Skeleton** action. **Bone Width** parameter can be adjusted by the width *Handle*. When adjusting the **Bone Width**, the bone's shape change accordingly.

Recursive Length Scale

Tip Width

Has **Bone Width**, **Tip Width** is used to set the influence of the bone before the use of **Link to Skeleton** action. **Tip Width** parameter can be adjusted by the width *Handle*. When adjusting the **Tip Width**, the bone's shape change accordingly.

Z-Depth

This parameter can be used to change the 'depth' (order) of a Bone in a Skeleton Layer.

Length Setup

7.29.5 Skeleton / Bones actions

Create Child Bone

When you add a **Skeleton Layer**, only one bone is created. In order to create a usable skeleton, you will need to add more bones.

Right clicking on any *Handle* of bone display a contextual menu. From this menu select **Create Child Bone** and a new child bone will be created. This child bone will move accordingly to his parent.

Link to Bone

Link to Bone link artwork, for example handles of a shape created with the *Spline Tool*, to a bone.

Select a single or multiple handles from your artwork, right click on a bone and from the contextual menu, select **Link to Bone**. Now when you move the Bone, the selected handles also move.

See it in action in [bones dev part 5](#)

Link to Skeleton

Before you use **Link to Skeleton** you should adjust the **Bone width** and **Tip width** of each bones of you skeleton to set the bone shape.

Select one or more handles from your artwork, select the **Skeleton layer** right click on it and from the contextual menu, select **Link to Skeleton**. All the handles covered by the bone shape will be automatically linked.

See it in action : [Bones dev part 7](#)

Disconnect

If you want to disconnect / unlink a bone from the part of the artwork it is linked to.

- From the *Layers Panel*, select the **Skeleton Layer**
- From the *Parameters Panel*, open the **Bones** list
- Select the corresponding **Item** (the according bone's *Handle* will be highlighted whitth red squares)
- Right click to open the context menu and select **Disconnect**.

Change parent

You can also change the parent of a bone in the skeleton, even during animation.

- From the *Layers Panel*, select the **Skeleton Layer**
- From the *Parameters Panel*, open the **Bones** list
- Select the corresponding **Item** (the according bone's *Handle* will be highlighted whitth red squares)
- Click on the **Value of Parent** to change the parent or set the parent to none.

7.29.6 See Also

- To deform raster images : *Skeleton Deformation Layer*.
- Doc:Basic_Bone_Tutorial
- A SIF-file of a rigged character using bones can be found here [Riggedchar.zip](#)

7.30 Skeleton Deformation Layer



7.30.1 About Skeleton Deformation Layers

Skeleton Deformation Layer is for raster deformation (pixel level) and can be used to image layers or whatever is in the context of the skeleton deformation layer (the rendered result of what's below it).

Skeleton Deformation acts like a normal “raster” layer (i.e. Blur, Warp), it is a raster deformation.

The Skeleton Distortion layer allows to set the distortion area and mesh resolution.

The *Layers Panel*'s Show/Hide checkbox of the Skeleton Deformation Layer activate/deactivate the deformation and the visibility of the distortion area.

7.30.2 Parameters of Skeleton Deformation Layers

The parameters of the Skeleton Deformation Layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Blur Layer</i>	Fast Gaussian Blur	integer
 Bones		list
 Point 1		vector
 Point 2		vector
 Horizontal subdivisions	32	integer
 Vertical subdivisions	32	integer

7.30.3 Usage of the Skeleton Deformation Layer

Set the distortion area

The Skeleton Deformation Layer's bound rectangle define the distortion area. You can modify it by dragging the handles of the first and second corners (Point 1 up left and Point 2 down right).

Build the skeleton

To setup the bones of the Skeleton Deformation Layer, first, deactivate the deformation unchecking the Show checkbox of from the *Layers Panel*. Now you can move the first bone using its *Origin* handle, change its orientation with the *Angle* handle and its length using the *Local Length Scale* handle. Right click in any handle of bones and choose *Create Child Bone* to create another bone and adjust it with its handles. Any child bones will follow recursively his parents position.

Once setup the armature, control the influence. Enable *Width Handle* (Alt+5) and set the influence area for each bone.

Reset the Pose

Once you have terminated to setup the bones of your skeleton, from the *Layers Panel*, right click on the Skeleton Deformation Layer and from the contextual menu choose **Reset Pose** to create the initial pose (without deformation) of the skeleton.

Now you can animate your skeleton !

Video Howto

- [Bone-driven Image Distortion \(part 4\)](#)

Why using skeleton deformation layer parts of the image are disappearing?

If parts of the image are disappearing when you enable the Skeleton Deformation Layer (and have checked that you don't have Cairo rendering ([Setup_Dialog#Render_Tab]) enabled), is't surely because the image is cropped by bone influence area. You need to turn off bone distortion layer (from the *Layers Panel*, select it, enable width handles (Alt+5) and set the influence area for each bone.

7.30.4 Specific parameters for Skeleton Deformation Layers

Some of the parameters of Skeleton Deformation Layers are common parameter that are shared for several types of Layers. Click on the links to see their definitions. Only the particular parameters for the Skeleton Deformation Layers are described here.

Bones

The **Bones** parameter is a list of Bone elements.

The parameters of a **Bone** are:

Name	Value	Type
 Name	Bone 1	string
Parent	No Parent	bone_valuenode
 <i>Origin</i>	0.00000, 0.00000	vector
 Angle	0.00°	angle
 Local Length Scale	1.00000	real
 Bone Width	0.10000	real
 Recursive Length Scale	1.00000	real
 Tip Width	0.10000	real
 Z Depth	0.00000	real
 Length Setup	1.00000	real

Name

Parent

Origin

Coordinate of the Origin of the Bone.

Angle

Angle of the Bone.

Local Length Scale

Bone Width

Recursive Length Scale

Tip Width

Z Depth

This parameter can be used to change the 'depth' (order) of a Bone in a Skeleton Layer.

Length Setup

Point 1

The parameter Point 1 is a Vector point defining the upper left corner of the distortion area.

Point 2

The parameter Point 2 is a Vector point defining the bottom right corner of the distortion area.

Horizontal Subdivisions

Horizontal definition of the meshes. This integer value affect the distortion quality.

Vertical Subdivisions

Vertical definition of the meshes. This integer value affect the distortion quality.

7.30.5 See also

- *Cutout Tool* to select a piece of the composition by adding a mask.
- Vector art specific *Skeleton Layer*.

7.31 Solid Color Layer



7.31.1 About Solid Color Layers

A Solid Color Layer create a plain color ground. Usefull for example for scene background.

7.31.2 Parameters of Solid Color Layers

The parameters of the solid color layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color

7.32 Sound Layer



7.32.1 About Sound Layers

Note: Actually the “Sound Layer” isn’t *rendered*!

You can use [ffmpeg/avconv](#) for manual merging. Read about [simple](#) or [advanced](#) way

The Sound Layer ...

7.32.2 Parameters of Sound Layers

The parameters of the Sound Layer are:

Name	Value	Type
π Z Depth	0.000000u	real
“ File Name	sound file path	string
⌚ Delay	0.000000u	time
π Volume	1.000000	real

7.32.3 How to use Sound Layer

Check this simple tutorial : [Sound Layer \(part 1\)](#)

7.33 Spherize Layer



7.33.1 About Spherize Layers

FIXMM!!!

7.33.2 Parameters of Spherize Layers

The parameters of the spherize layers are:

Name	Value	Type
● Position	0.000000u,0.000000u	vector
π Radius	1.000000u	real
π Amount	1.000000u	real
⏻ Clip		bool
⚙ Distort Type	Spherize	integer

7.34 Spiral Gradient Layer



7.34.1 About Spiral Gradient Layers

The Spiral Gradient Layer draw a gradient following spirals, clockwise or anticlockwise, at determined center, radius and start angle.

7.34.2 Parameters of Spiral Gradient Layers

The *Parameter* of the spiral gradient layers are:

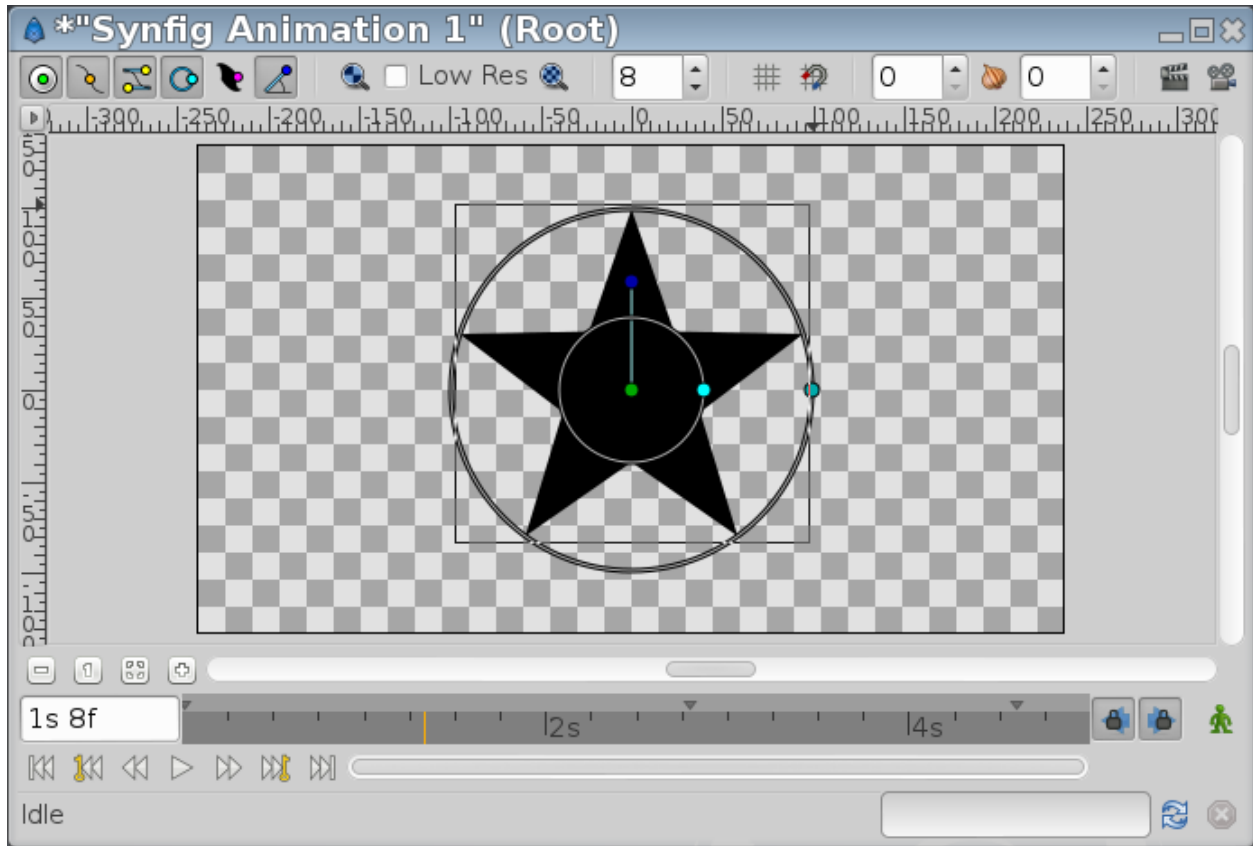
Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Gradient</i>		gradient
 Center	0.500000u,0.000000u	vector
 Radius	0.500000u	real
 Angle	0.00	angle
 Clockwise	☐	bool

7.35 Star Layer



7.35.1 About Star Layers

A Star Layer is a geometric object that is made by the filling the region resulting from connecting the points of two concentric circles with straight lines. The number of points on which the circles are divided defines the number of peaks of the star. The points over the circles are spread regularly over both circles but one of them has the points rotated $N/360/2$ degrees from the other circle (being N the number of peaks or points).



7.35.2 Parameters of Star Layers

The parameters of the star layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Color</i>		color
 Origin	0.500000u,0.000000u	vector
 <i>Inver Parameter</i>	☐	bool
 Antialiasing	☐	bool
 Feather	0.500000u	real
 Type of Feather	Fast Gaussian Blur	integer
 Winding Style	Non Zero	integer
 Outer Radius	1.500000u	real
 Inner Radius	0.500000u	real
 Angle	0.00	angle
 Points	5	integer
 Regular Polygon	☐	bool

The parameters of the star layers are the same as the majority of the shape-like objects but it has specific parameters for its own properties.

Outer Radius

Defines the radius of the circle where the peaks of the star lay.

Inner Radius

Defines the radius of the circle where the valleys of the star lay.

Angle

Is the rotation angle for the first peak of the star. Its default value is **90** degrees.

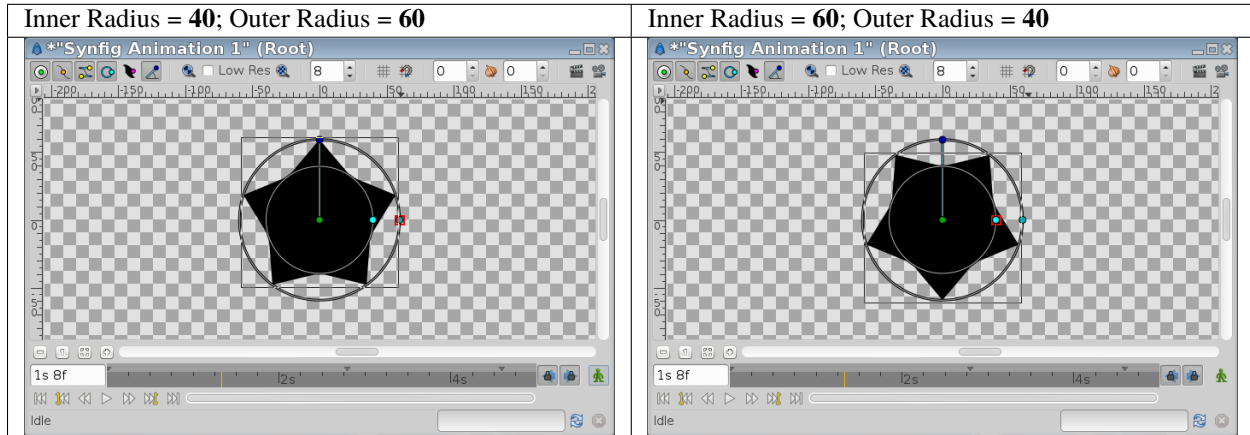
Points

Defines the number of divisions done in the circles and therefore the number of points and peaks in the star.

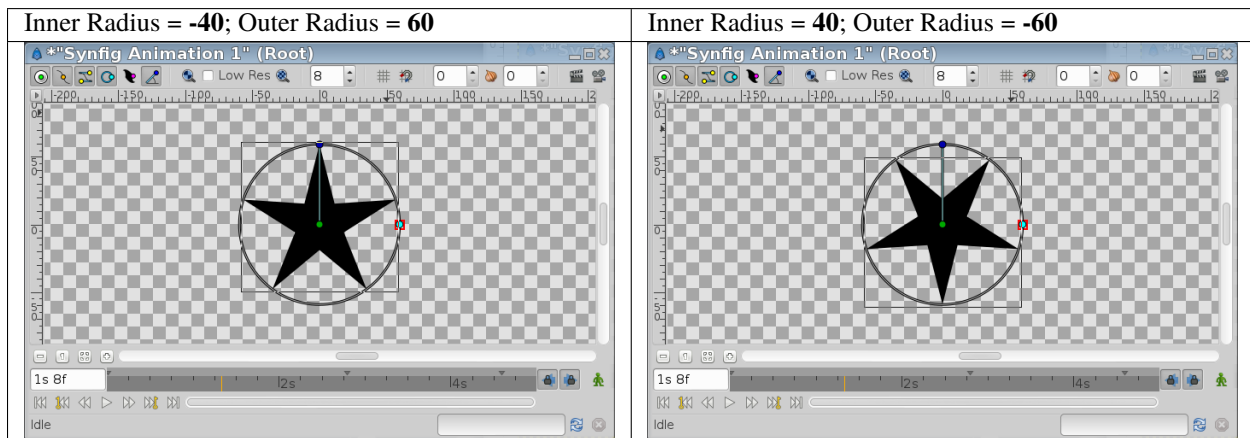
7.35.3 Playing Around

Crazy Radii

The Outer Radius shouldn't be greater than Inner Radius. It only changes the star's orientation:

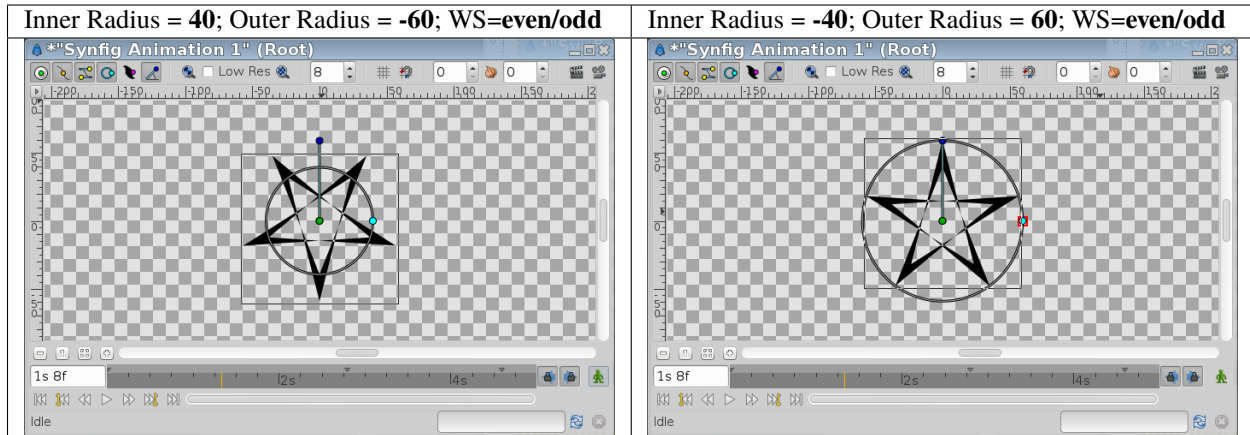


You can also play with negative values:



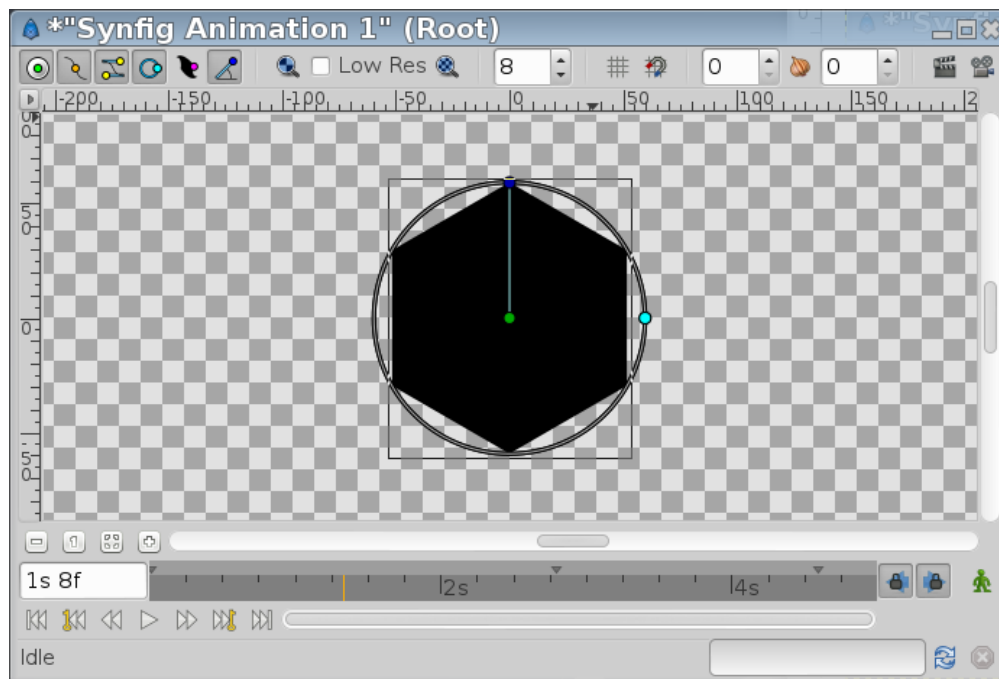
Winding Style Hacks

You can even play with the `|Winding_Style_Parameter|` and negative values to obtain some effects:



Regular 2N-sided Polygons

Also you can *Linking* both radii to create a $2*N$ sided regular *Polygon Layer*, where N is the number of points (3 points for this case (six sides)):



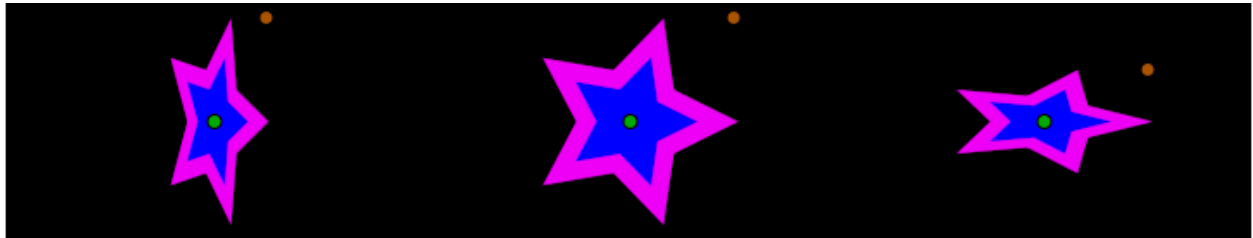
7.36 Stretch Layer



7.36.1 About Stretch Layers

Stretch Layer distorts objects of the lower layers in both axis.

7.36.2 Sample



- Center image is default size of Stretch Layer with *Amount* – 1u,1u
- *Amount* for left image – 1u,0.5u
- *Amount* for right image – 0.5u,1u

7.36.3 Parameters of Stretch Layers

The parameters of the Stretch Layers are:

Name	Value	Type
 Amount	1.000000u,1.000000u	vector
 Center	0.000000u,0.000000u	vector

7.37 Stroboscope Layer



7.37.1 About Stroboscope Layers

The stroboscope layer shows the layers under it at regular intervals and then freezes them for the remainder of the interval duration. It is similar to decreasing the frame rate of an animation while preserving its length. It has a single parameter called “Frequency” (in times per second), which determines how often samples are taken. A frequency of exactly the frames per second of your project will not affect your animation, while smaller frequencies will appear as if you had lowered the frame rate to that amount.

7.37.2 Parameters of Stroboscope Layers

The parameters of the stroboscope layers are:

Name	Value	Type
 Z Depth	0.000000u	real
 Frequency	2.000000u	real

7.38 Super Sample Layer



7.38.1 About Super Sample Layers

The Super Sample layer is used to anti-alias all the layers under it.

It isn’t often needed, since most layers produce smooth output already.

7.38.2 Parameters of Supersample Layer

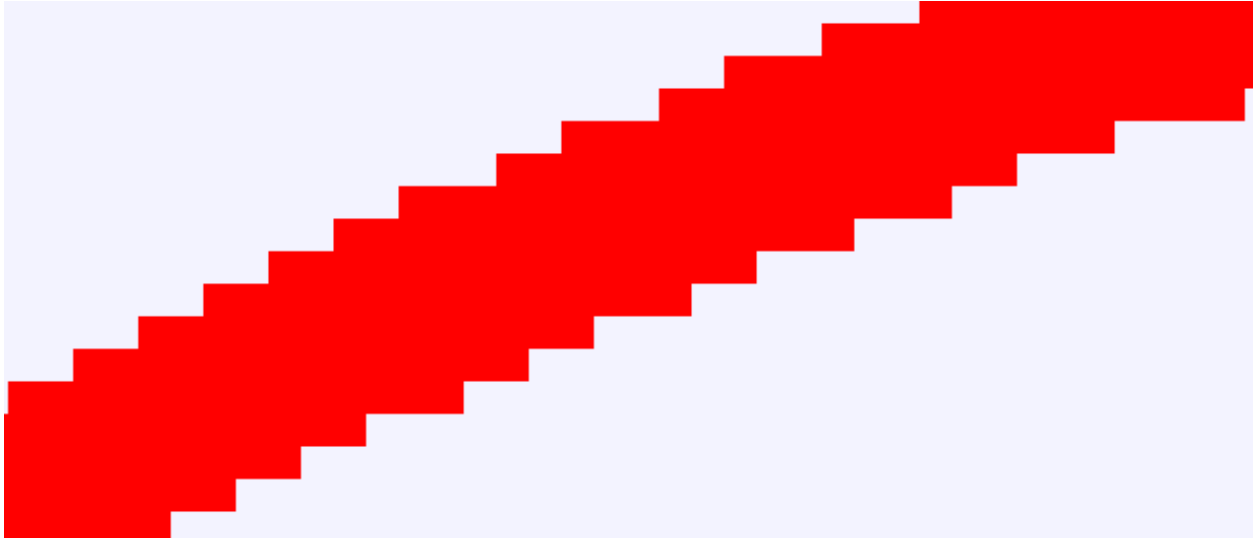
The parameters of the supersample layer are:

Name	Value	Type
 Width	2	integer
 Height	2	integer
 Use Parametric	☐	bool
 Be Alpha Safe	☐	bool

You specify width and a height parameters. Internally, the scene is rendered times wider and times taller than the output image, and then each by block is averaged down to a single pixel.

For example, the output image is 320x240 pixels, the width and height parameters of the Super Sample layer are both set to the default value of 2. Synfig will render a 640x480 image internally, and then average the 4 pixels in each 2x2 block down into a single pixel that it then uses in the output image.

This shows a zoomed image of an outline with anti-alias turned off, and without a supersample layer:

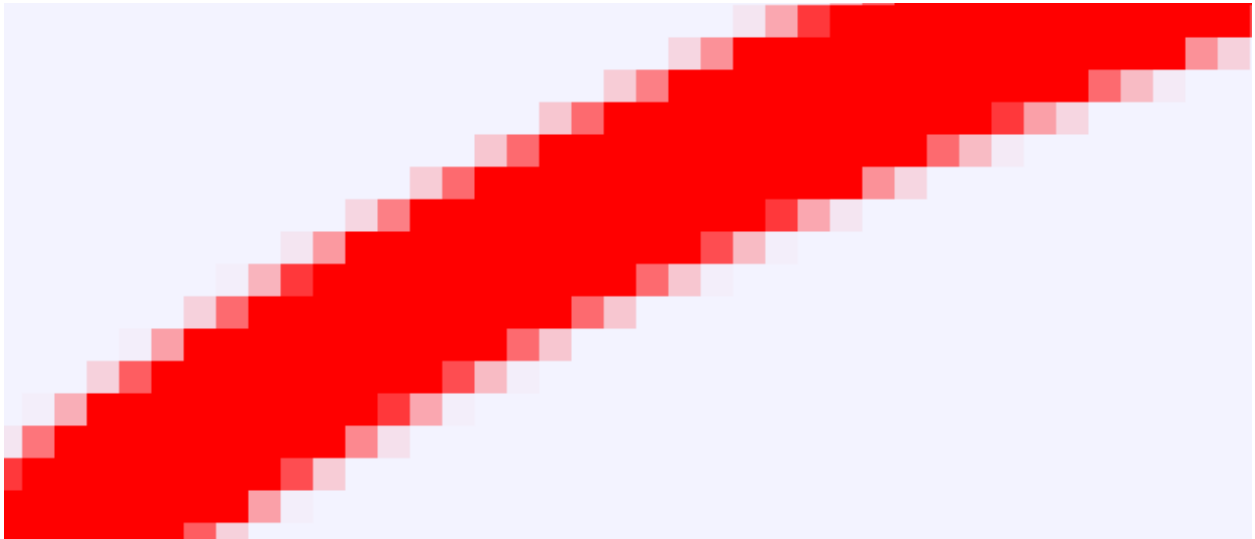


The 'Use Parametric' checkbox makes it use the parametric renderer rather than the default accelerated renderer.

The 'Be Alpha Safe' checkbox makes the averaging process aware of the alpha channel. When 'Alpha Safe', the average is $\text{sum}(\text{color} * \text{alpha}) / \text{sum}(\text{alpha})$, with an alpha value of $\text{sum}(\text{alpha}) / \text{sum}(1)$. When not 'Alpha Safe', the average is simply $\text{sum}(\text{color}) / \text{sum}(1)$.

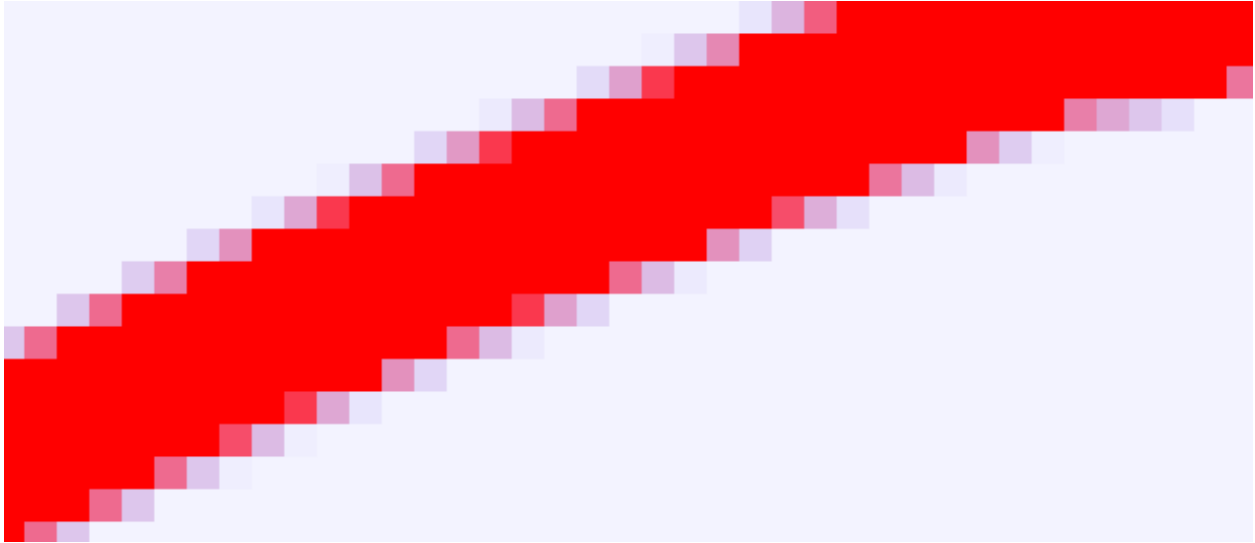
For instance, consider averaging two pixels, one a completely transparent blue (R:0 G:0 B:1 A:0) and the other an opaque red (R:1 G:0 B:0 A:1). When being alpha safe, the average is (R:1 G:0 B:0 A:0.5). When not being alpha safe, the average is (R:0.5 G:0 B:0.5 A:0.5).

This is the same outline as before, but with a Super Sample layer over the top of it. This one is with 'Be Alpha Safe' enabled:



And this one is without 'Be Alpha Safe'. The background is a strong blue, but with a very low Alpha. The edges are much bluer than they would be if alpha was taken into account:

It's worth pointing out that the Super Sample layer is disabled when the 'quality' setting is 10 or higher. During editing, the quality is 10.

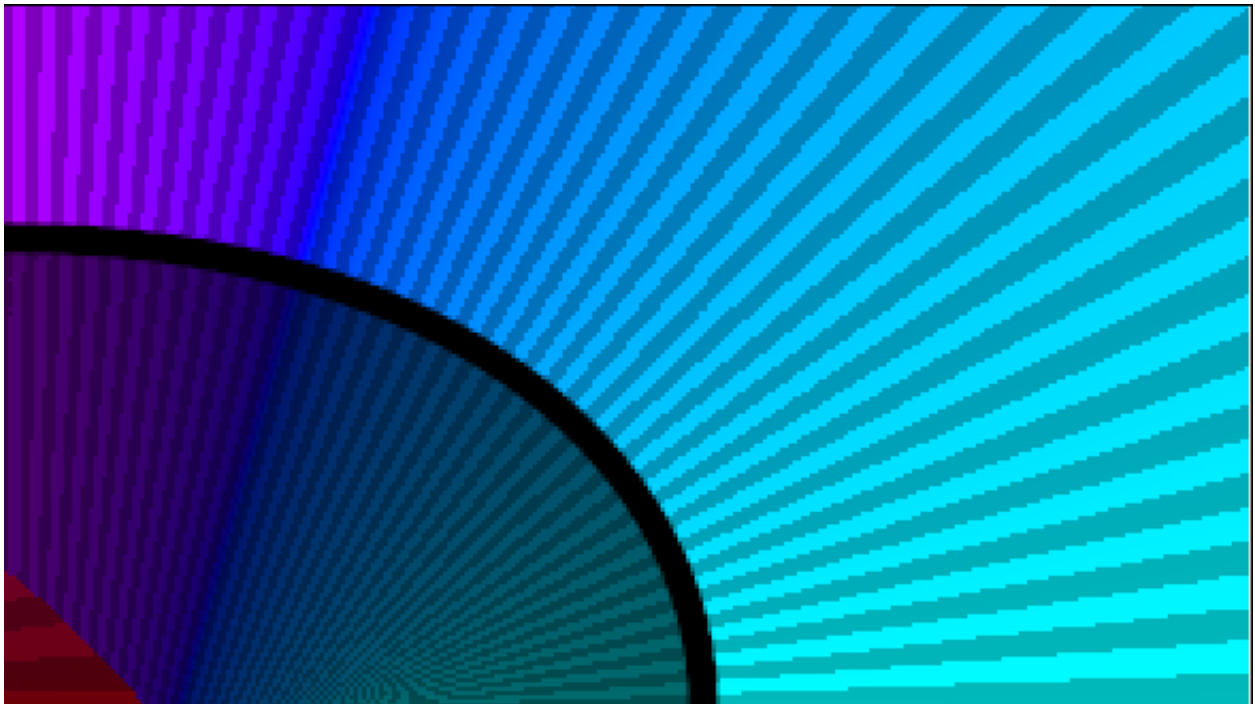


If you want to see an example of what Super Sample does, create an outline and turn off it's anti-aliasing parameter. It will look jagged around the edges. Add a Super Sample layer over the top of it, save the file and render it to an image. Rendering with 'synfig -Q 10 file.sif' will leave it jagged, and 'synfig -Q 9 file.sif' will make it smoother.

7.38.3 Examples

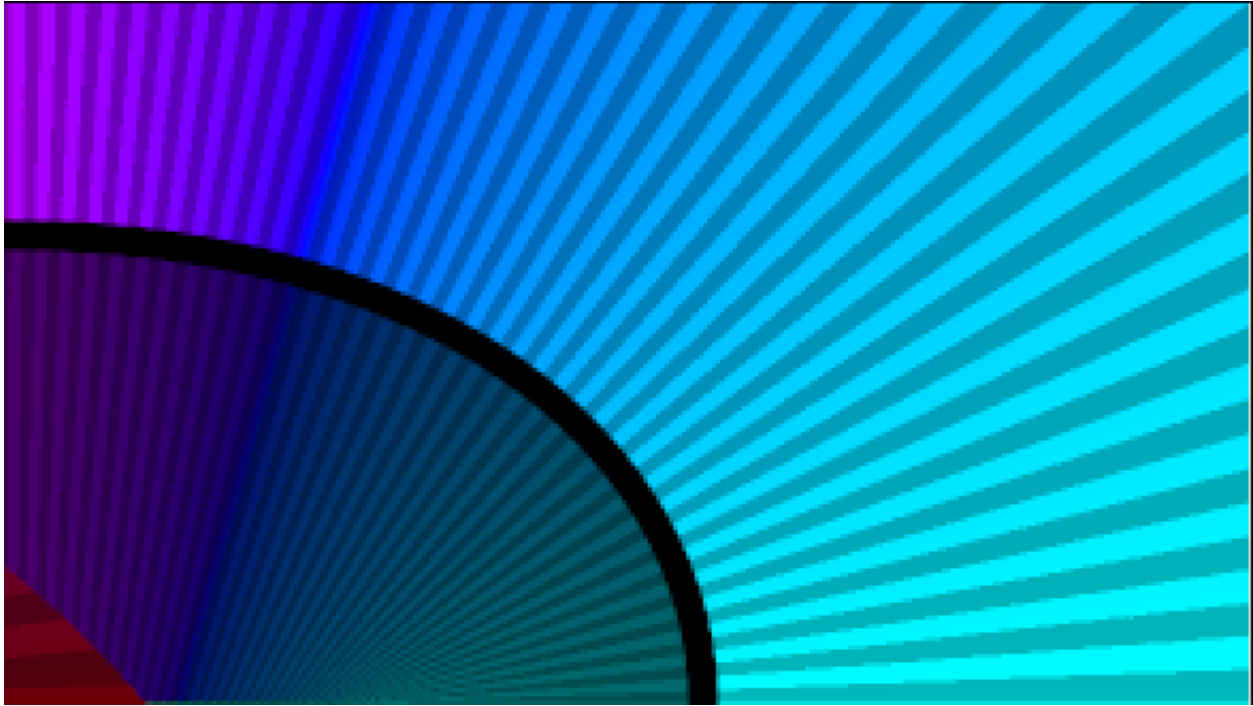
width=1 height=1

This is the same as not using Super Sample at all.



width=2 height=2

Notice how there's only one intermediate color used to antialias in the very bottom right corner.



width=3 height=3

Now two intermediate colors are used, and the result is smoother.

width=4 height=4

This looks good, but a 4x4 Super Sample layer increases the render time by a factor of 16.

width=1 height=4

Vertical lines are jagged, horizontal lines are smooth.

width=4 height=1

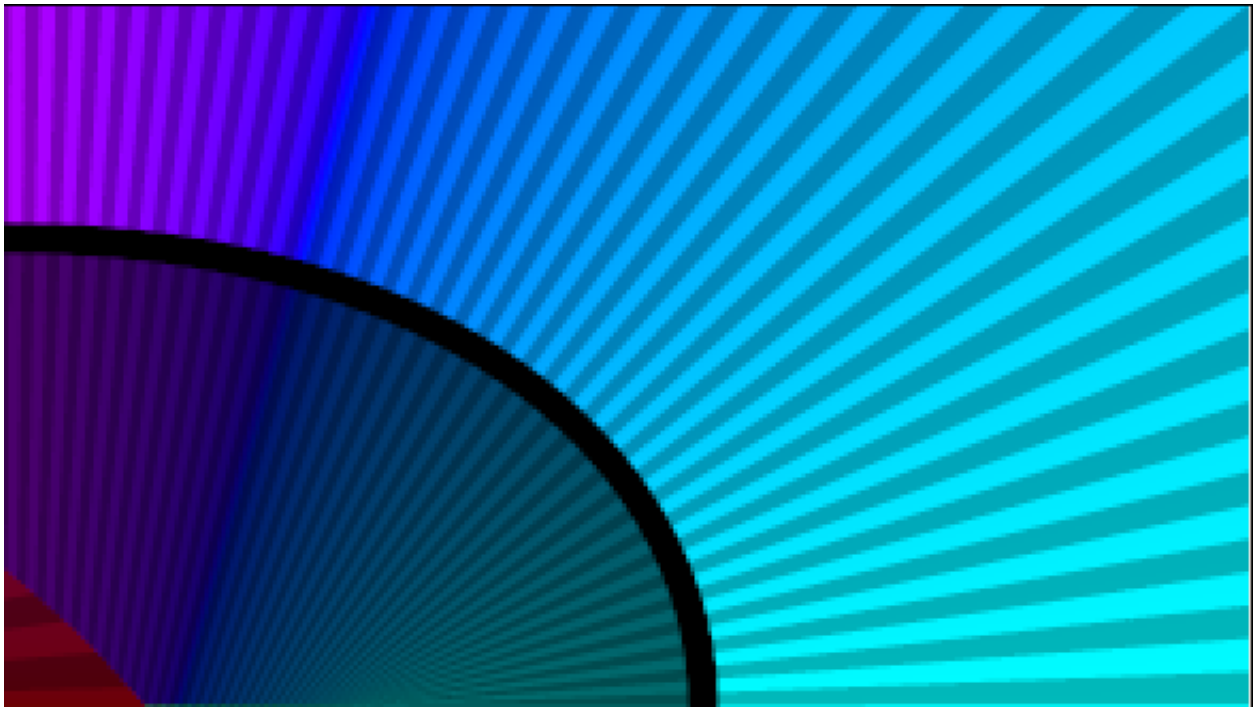
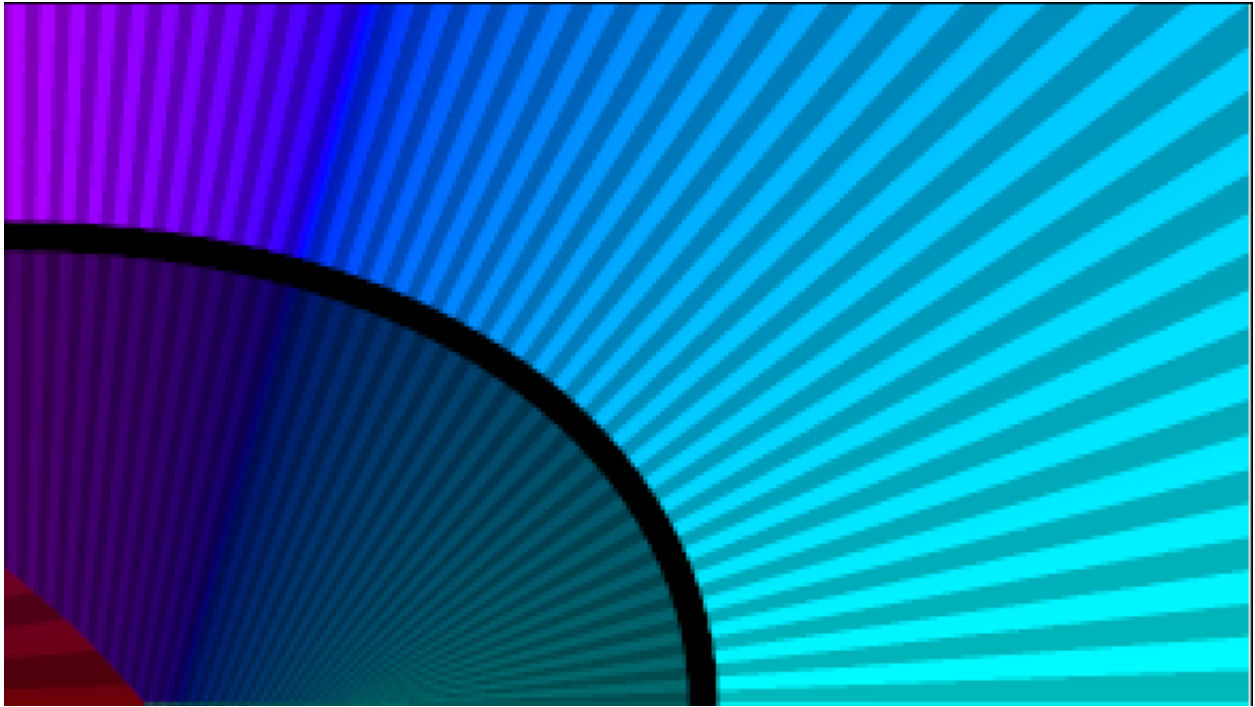
Vertical lines are smooth, horizontal lines are jagged .

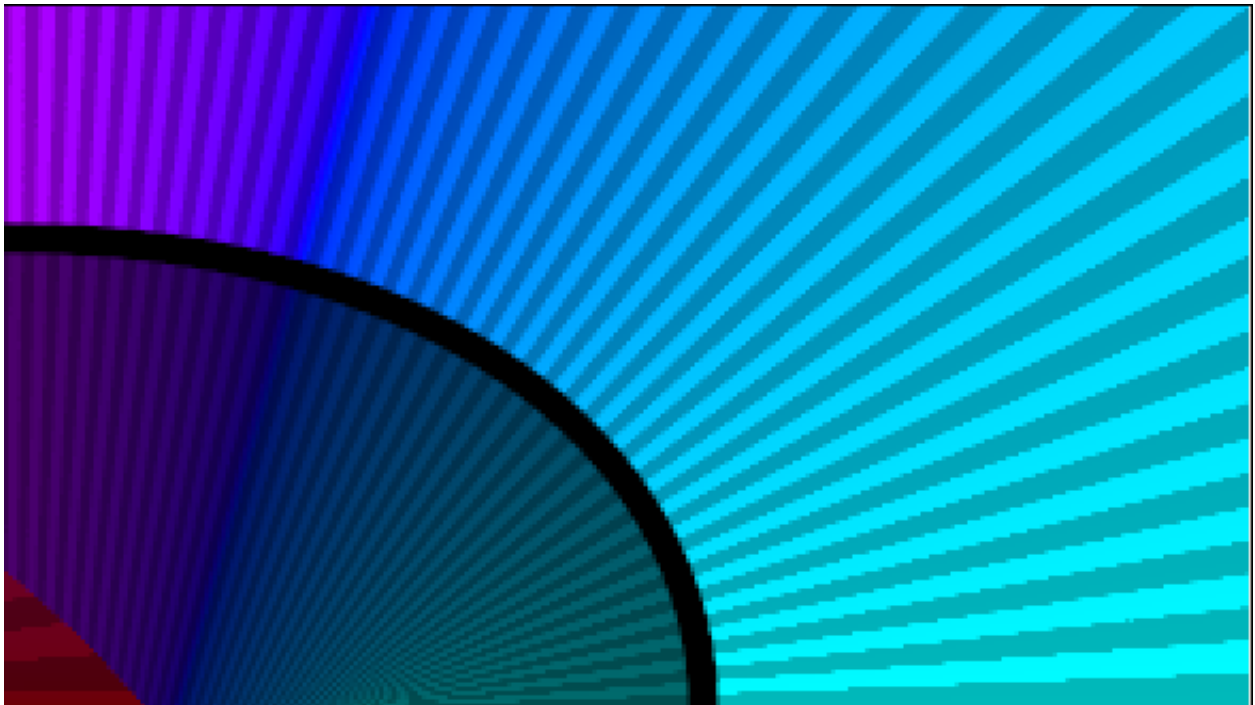
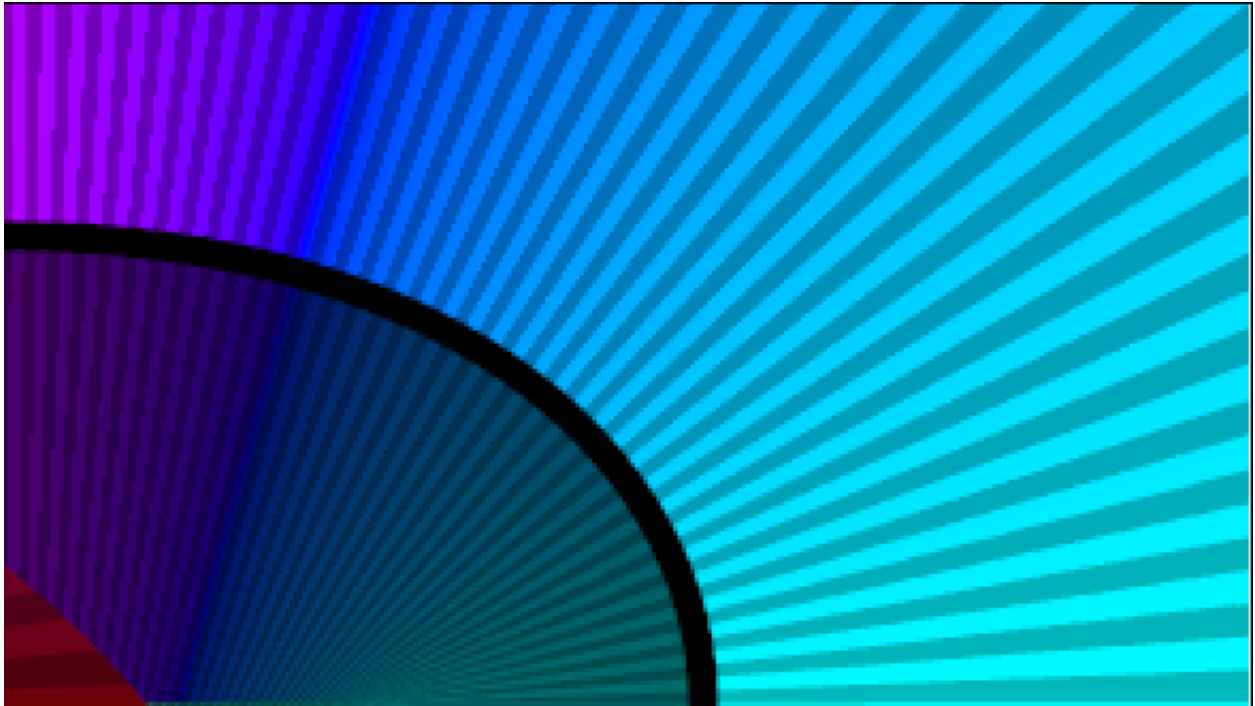
7.39 Switch Group Layer

7.39.1 About Switch Group Layer

The Switch Group Layer is a special layer that can group many layers, only displaying one layer at a time.

A Switch Group Layer can be created through the New Layer Menu ([Category:Layer_Menu#New_Layer_Menu]), using New Layer -> Other -> Switch Group Layer| or through the .





7.39.2 Parameters of Switch Group Layer

The parameters of the Switch Group Layer are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer (static)
 <i>Origin</i>	0.000000u,0.000000u	vector
Transformation Parameter	0.000000, 0.000000, 0.00°, 60, 60	transformation
 Canvas Parameter		canvas
 Speed	1.000000	real
 <i>Time Offset</i>	Of	time
 <i>Lock Selection</i>	☐	bool (Static)
 Outline Grow	0.000000	real
 Active Layer Name	<empty>	string

The displayed layer is chosen by the Active Layer Name parameter. If there are several layers with the same name in the Switch Group Layer, then only the topmost one is displayed.

7.39.3 Video Usage

Simple frame by frame animation

7.40 Text Layer



7.40.1 About Text Layers

[todo]

A Text Layer can be created with the *Text Tool* or by drag and drop.

7.40.2 Parameters of Text Layers

The parameters of the text layers are:

Name	Value	Type
 <i>Z Depth Parameter</i>	0.000000	real
 <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 Text	Text Layer	string
 <i>Color Dialog</i>		color
 Font Family	Sans Serif	string
 Style	Nomral	integer
 Weight	Normal	integer
 Horizontal Spacing	1.000000	real
 Vertical Spacing	1.000000	real
 Size	0.500000u,0.500000u	vector
 Orientation	0.500000u,0.500000u	vector
 Origin	0.000000u,0.000000u	vector
 Kerning	☐	bool
 Sharpen Edges	☐	bool
 <i>Inver</i>	☐	bool

7.40.3 How to use Text Layer

Quick trick that allows to use custom fonts in Synfig.

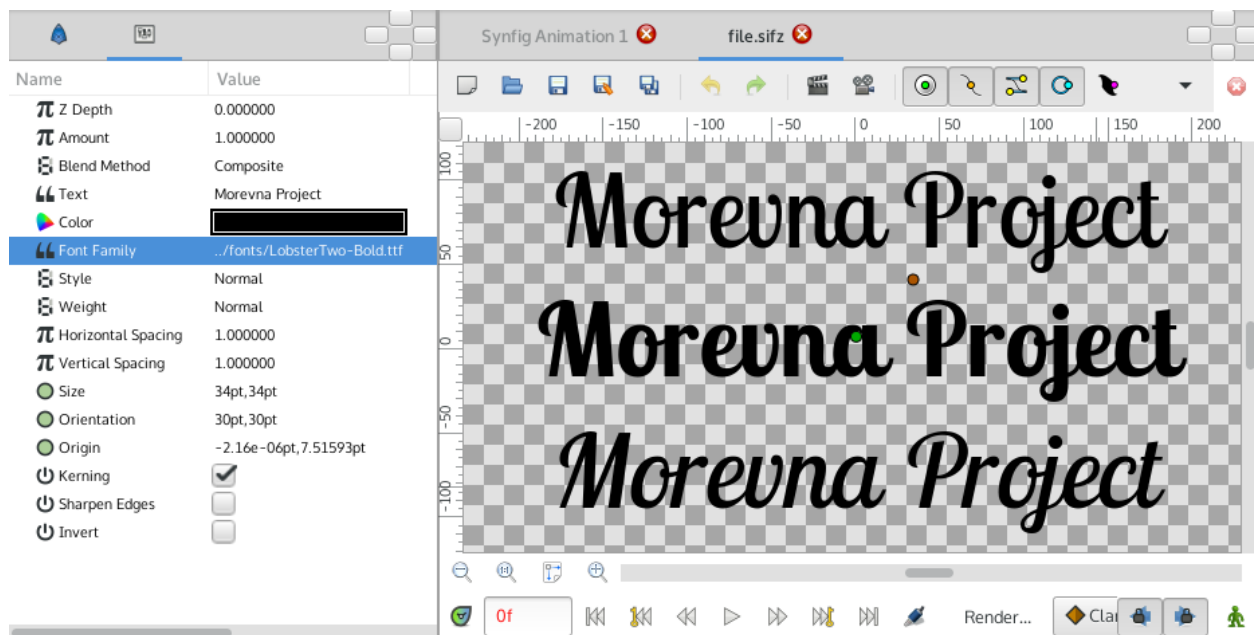
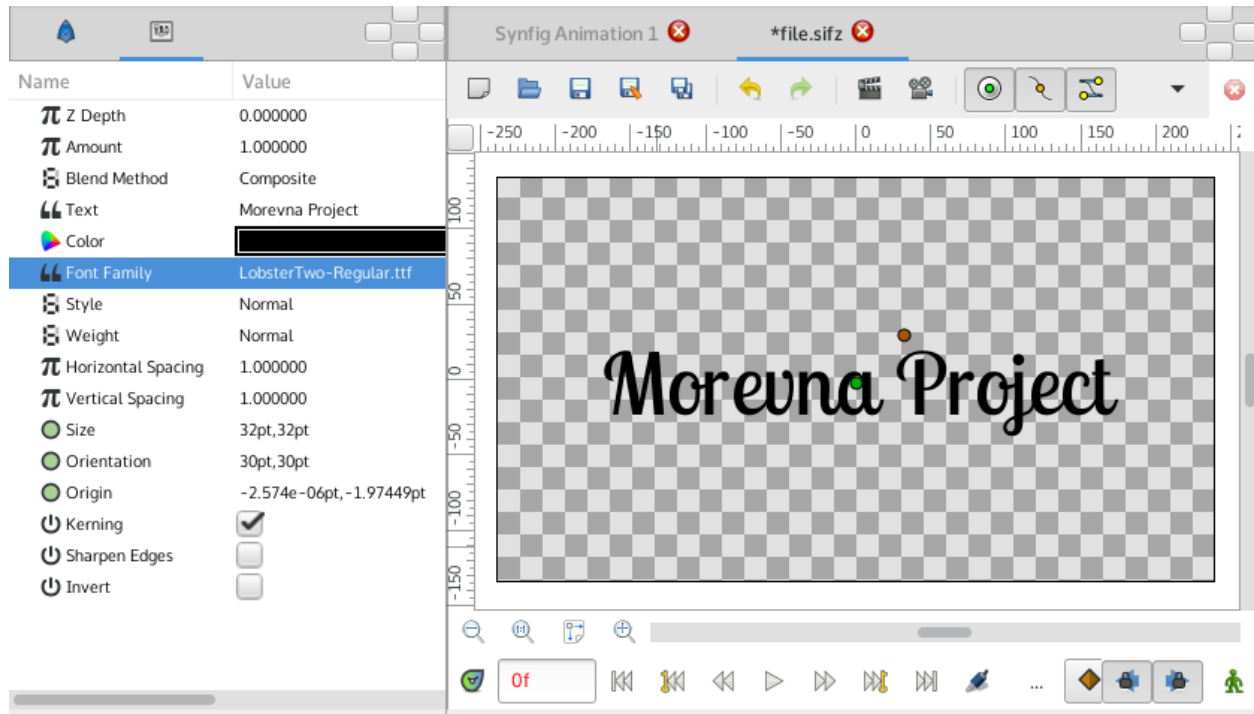
Here's what you need:

- Put your font (ttf) file in the same dir as your Synfig (sif or sifz) file.
- Create Text Layer and in the Font Family parameter put full filename of font file (i.e. "LobsterTwo-Regular.ttf", without quotes).

That's it!

After examining font files, you might notice that different variants of font style are usually residing in separate files (i.e. "LobsterTwo-Italic.ttf", "LobsterTwo-Bold.ttf", "LobsterTwo-BoldItalic.ttf", etc). So you can use different font variants in the same way just by referencing the appropriate font file.

If your font file is placed in different location, then you can specify absolute or relative path to it (i.e. "../fonts/LobsterTwo-Bold.ttf").



7.41 Time Loop Layer



7.41.1 About Time Loop Layers

The `Time Loop` layer is used to repeat an animation over and over. It loops a section of the layers below and within the same canvas over and over.

See also the *Convert: Time Loop* conversion, which can be used to loop the value of a single parameter, rather than an entire layer or group of layers.

7.41.2 Parameters of Time Loop Layers

These parameters, to **prevent undesired modification**, by default are Static Parameters.

The parameters of the time loop layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
 <i>Link Time</i>	0f	time
 <i>Local Time</i>	0f	time
 <i>Duration</i>	1s	time
 <i>Only for positive duration</i>	☐	bool
 <i>Symmetrical</i>	☐	bool

Link Time (time)

Start time of cycled material/child layers.

Local Time (time)

Start time of loop.

Duration (time)

Number of seconds or frames that are looped in the child layer.

Only For Positive Duration (bool)

- If checked and Duration is zero or negative, then the time loop layer is effectively disabled, and acts as if it wasn't there.
- If not checked and the Duration is zero, the Time Loop layer freezes the animation of the children layers at the value of Link Time.

Symmetrical (bool)

If not checked, and the current time is less than Local Time, then Duration is taken off the resulting time. This is to provide compatibility with [version 0.1](#) of the time loop layer.

7.41.3 How to use the Time Loop Layer ?

The Time Loop layer repeatedly loops through the Duration seconds of its child layers, from Link Time to Link Time + Duration.

Local Time is used to line up the offset of the time looping.

When the Time Loop layer is asked to set its time to Local Time, it sets the time in its child layers to be Link Time, ie. the start of the loop.

Breaking loop

1. Be Sure you are not in *Animate Editing Mode*
2. In the layer's parameters of the time loop, you need to remove the 'green guy' () , indicating that the parameter is Static, from *Duration*. Just right-click on him and set Allow animation. See Static Parameters for more informations.
3. On the *Timebar* navigate to a place where you want to break the animation. And now enter the *Animate Editing Mode*.
4. In the time loop parameters, set 'Duration' to EOT (End-of-Time).
5. Now you have three *Waypoint*. One before the current, we left it as it is. One is current, we *Convert* it to the constant (right-click on it -> Both -> Constant). And one after current, this one have to be removed (right-click on it -> Remove).
6. Now test it. Pay attention, that loop will always break at the very first frame of you animation, so you need to break it in the right place.

Detailed explanation

‘Green guy’ () - Static Parameters - is a guard that says “You shall not pass!” to the animation. So when you attempt to change such parameter in animation mode it just changes the usual way. It’s done to prevent unnecessary parameters to be animated, e.g. **[Blend_Method_Parameter]**.

Setting **Duration** to EOT. If you set it to 0f, then the whole animation will stop, because you try to play it from 0f (local time) to 0f (Duration). And EOT means it’ll continue to play your animation to the end of the time...

Removing *Waypoint* after current one. If you skip this step, then at the next waypoint **Duration** parameter will be restored to the value before current waypoint. It’s a standard Synfig’s behavior, but there’s a magical button somewhere that can change it.

Converting current waypoint to the constant. Without it, **Duration** will be smoothly interpolated from value you set to the EOT and the constant type will just set immediately at the waypoint. Interpolation can be a bit confusing sometimes, so you better read this about *Waypoint*.

7.41.4 Visualized Example

For example, suppose:

- Link Time is 5s
- Duration is 3s
- Local Time’ is 4s

And suppose that the Time Loop layer is applied over an existing animation. The **Link Time** and **Duration** specify that the section from 5s to 8s in the children layers will be looped. The **Local Time** specifies that this loop will be at the beginning at 4s. (And so also therefore at 1s, 7s, 10s, etc).

This is how the mapping actually works:

real time	child time (symmetrical = true)	child time (symmetrical = false)
0	7	4
1	5	2
2	6	3
3	7	4
4	5	5 (local time = 4; link time = 5)
5	6	6
6	7	7
7	5	5 (duration = 3, so loop repeats after 3 seconds)
8	6	6
9	7	7
10	5	5

Specifying a huge number for the **Duration** parameter effectively turns the Time Loop layer into a Time Shift layer. The **Link Time** and **Local Time** parameters controls which time in the children lines up with which time in the Time Loop layer, giving the amount of the timeshift, with both positive and negative differences working as expected.

7.41.5 Contrived Example

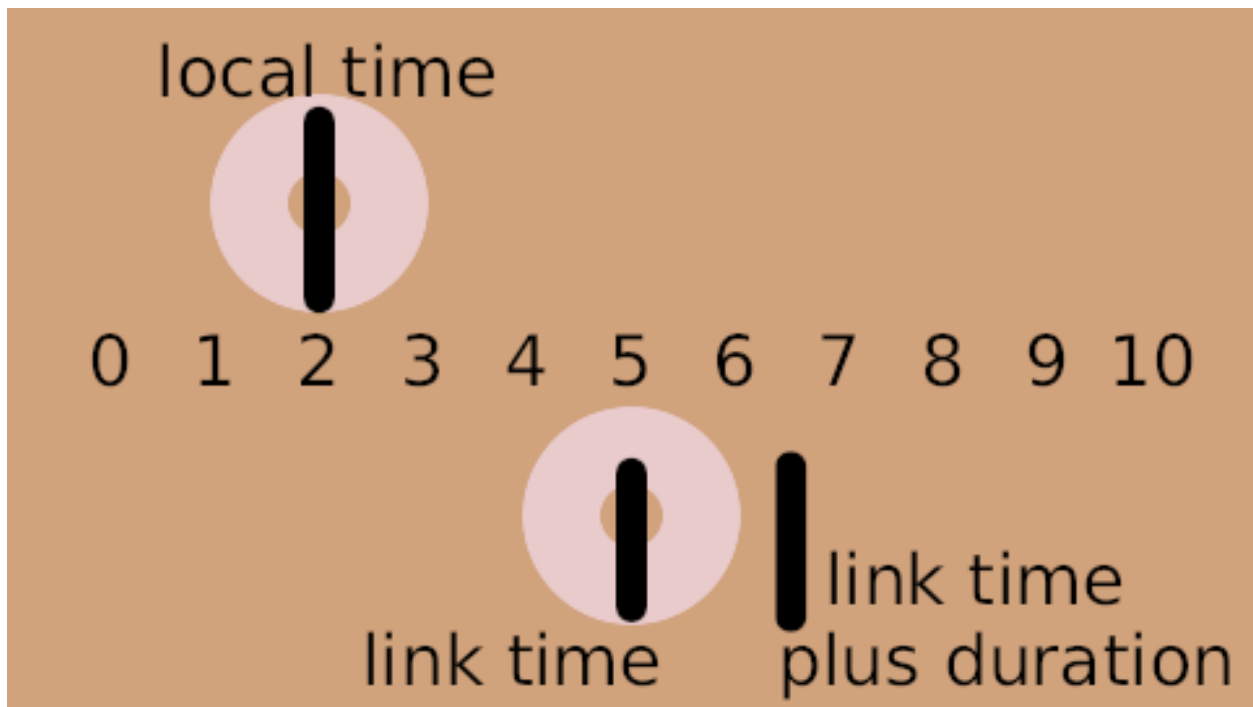
Download and examine this example file: `Time-loop-demo-0.2.sifz`

It's a 10 second animation, and shows 2 circles. The top one moves linearly from the left to the right. Its position is marked by static text digits 0 through 10.

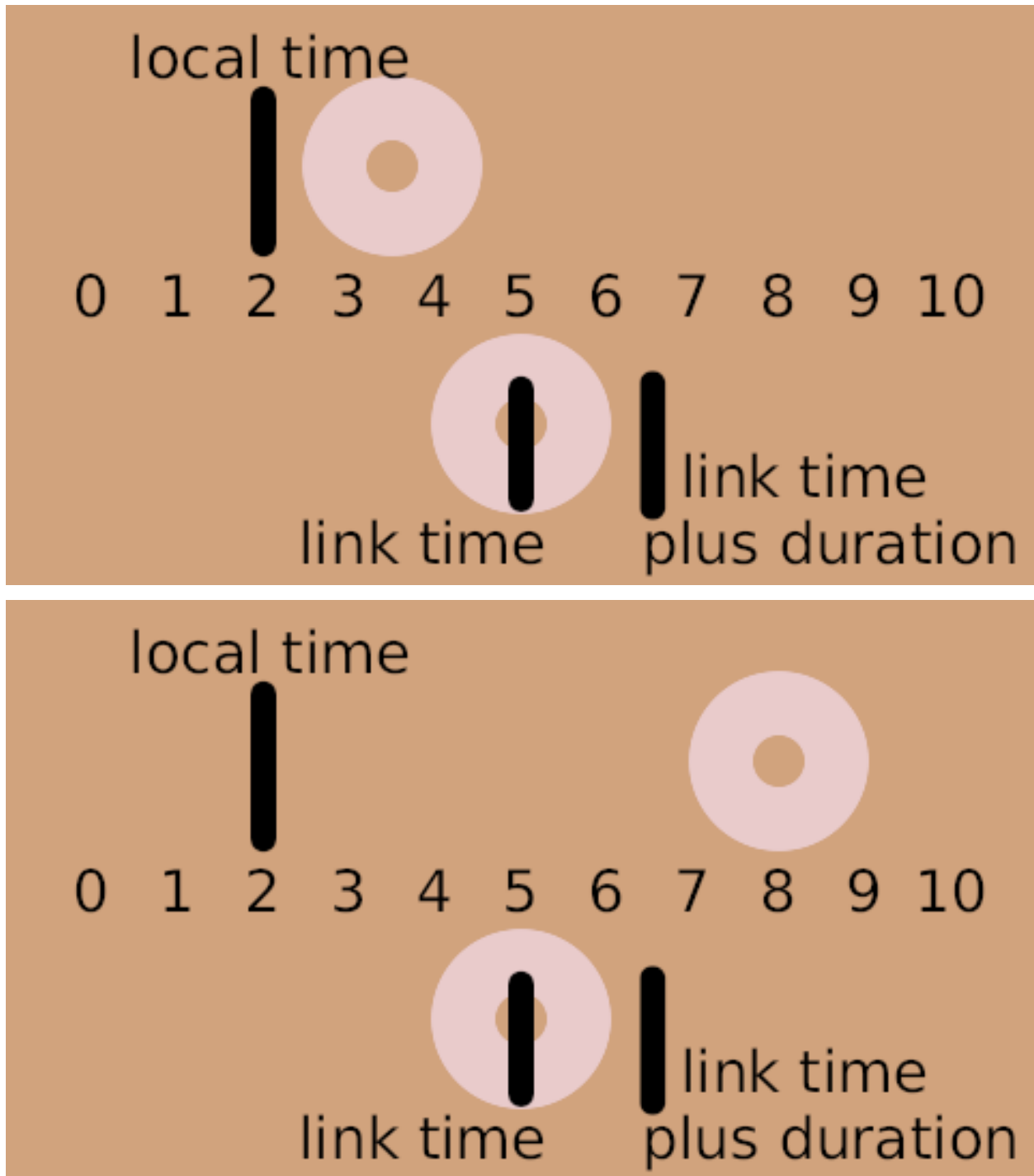
The other circle is an identical copy of the first one, with the same waypoints, but it's inside an Group layer. The parameters are:

- Link Time: 5s
- Duration: 1.5s
- Local Time: 2s
- Symmetrical: true

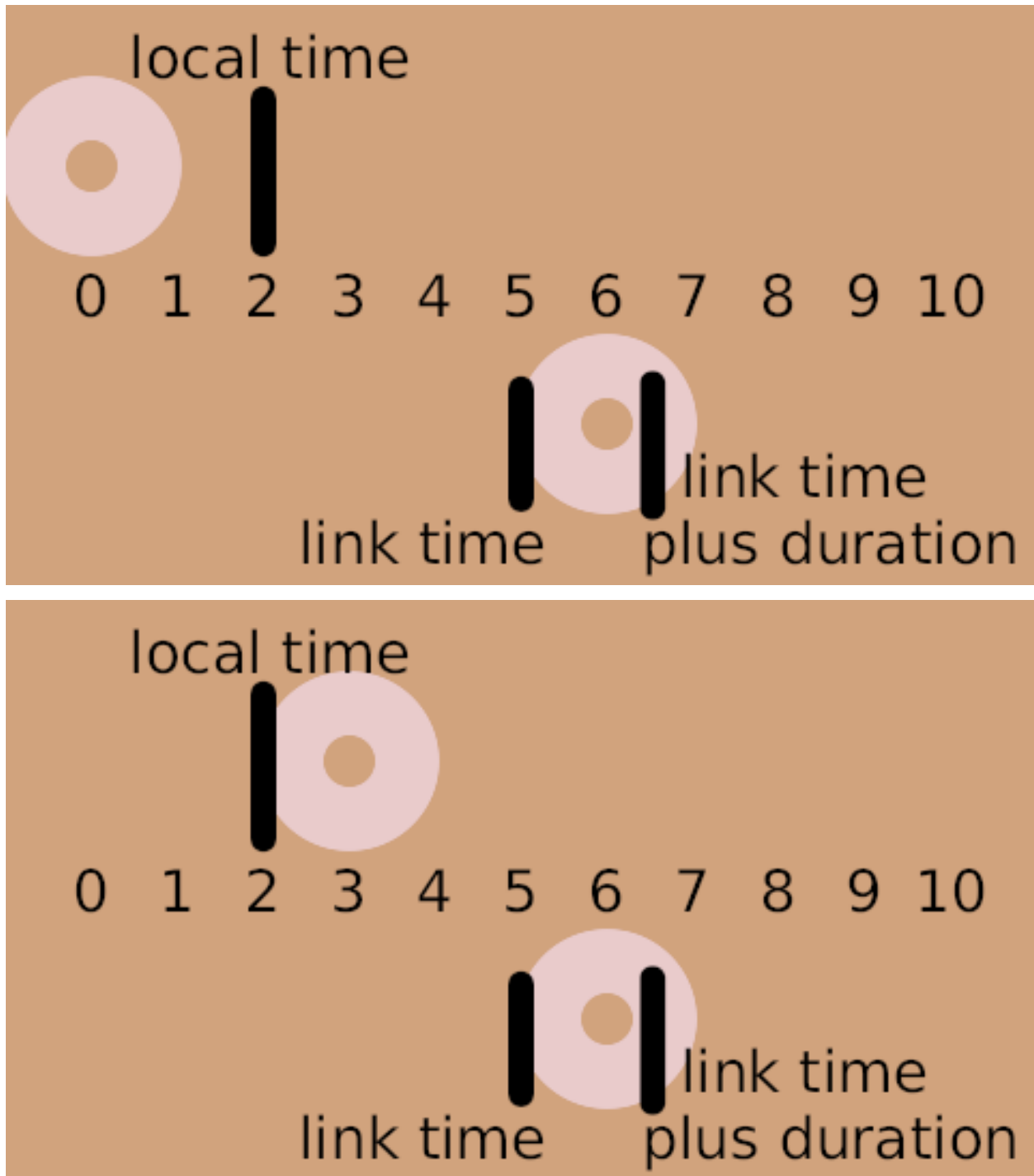
So as time=2s, the top circle is at position 2 (local time) and the bottom circle is at position 5 (link time):



The loop is 1.5s long, so the bottom circle is also at position 5 every 1.5 seconds before and after this point in time, for example at $t=3.5s$ and at $t=8s$:



The following two images show the positions at $t=0$ s and $t=3$ s. The loop starts at $t=2$ s, so it's also at the start at $t=0.5$ s. So at $t=0$ s it's half a second before finishing the previous loop. And at $t=3$ s the same is true, but 2 loops later on:



There's a rendered copy of this example on [YouTube](#), and it's also available for download: [Time-loop-demo-0.2.avi](#).

7.42 Translate Layer



7.42.1 About Translate Layers

Translate Layer change the position of any Primitive Layer below it. It apply an offset to the resulting calculated points to render the shape, it doesn't modify the values of the underlying layers. **Translate Layer** is part of Transform Layers ([Layers#Transform]) category.

7.42.2 Parameters of Translate Layers

The parameters of the translate layers are:

Name	Value	Type
 <i>Origin</i>	0.00000u,0.00000u	vector

7.43 Twirl Layer



7.43.1 About Twirl Layers

Twirl is the layer which deforms lower layers so that it seem as the vortex. The intensity of the vortex is set over the rotation angle - higher, the more intense.

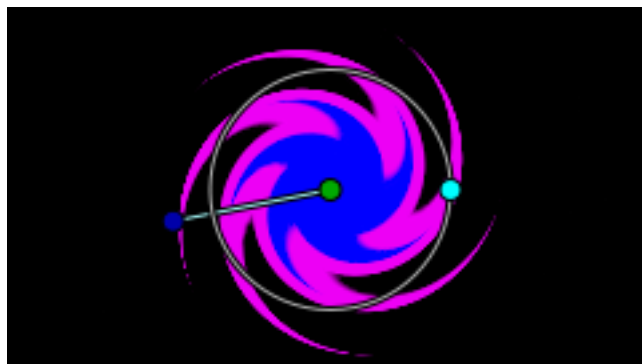
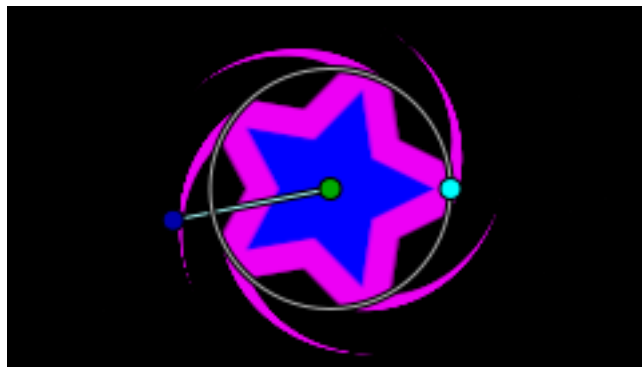
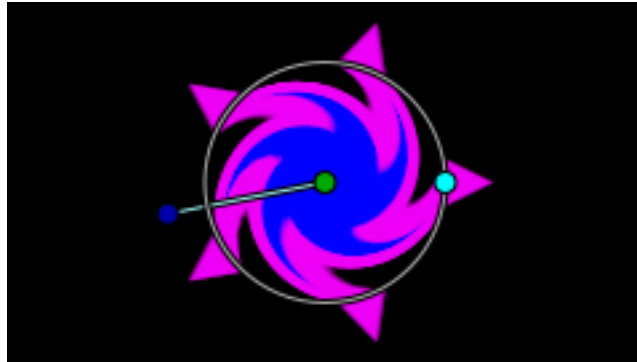
Size of vortex is determinated by the radius of a circle, and distortion can be activate separately for both the sides – inner and outer.

7.43.2 Sample

Distort Inside Parameter On

Distort Outside Parameter On

Both Distort Parameters On



7.43.3 Parameters of Twirl Layers

The parameters of the twirl layers are:

Name	Value	Type
 <i>Center</i>	0.00000u,0.000000u	vector
π Radius	1.00000u	real
 Rotations	0.00	angle
 Distort Inside		bool
 Distort Outside		bool

7.44 Warp Layer

Somes of the screenshots needs to be updated with 0.64.0



7.44.1 About Warp Layers

The warp layer is a simple but powerful distortion layer. In a few words it takes a rectangular portion of the resulting render of the layers that are behind it and maps the four corners of the rectangle to four arbitrary points in the plane. It is a 2D -> 2D transformation.

To keep the object in place when apply the perspective, you need to set the corners of the perspective destination properly around the object. Notice that the source rectangle must be centered on the object, to achieve a good effect.

If source == destination then object is not warped.

7.44.2 Parameters of Warp Layers

The parameters of the warp layers are:

Name	Value	Type
 Source TL	-2.00000u,2.000000u	vector
 Source BR	2.00000u,-2.000000u	vector
 Dest TL	-1.80000u,2.100000u	vector
 Dest TR	2.20000u,-2.000000u	vector
 Dest BR	-2.20000u,-2.000000u	vector
 Dest BL	0.00000u,0.000000u	vector
 Clip		bool
π Horizon	1.00000	real

Source TL

The Top Left corner (vertex) of the source that is going to be mapped.

Source BR

The Bottom Right corner (vertex) of the source that is going to be mapped. Combined with Source TL ([Warp_Layer#Source TL]) it defined the Source rectangle.

Destination TL

The Top Left corner (vertex) of the destination where the source is going to be mapped.

Destination TR

The Top Right corner (vertex) of the destination where the source is going to be mapped.

Destination BL

The Bottom Left corner (vertex) of the destination where the source is going to be mapped.

Destination BR

The Bottom Right corner (vertex) of the destination where the source is going to be mapped.

Clip

When checked (boolean) it only maps the pixels which lie inside the Source rectangle.

Horizon

For infinite layers (gradients, checkerboard, etc.) it define (Real) where to stop rendering the points of the vanishing point.

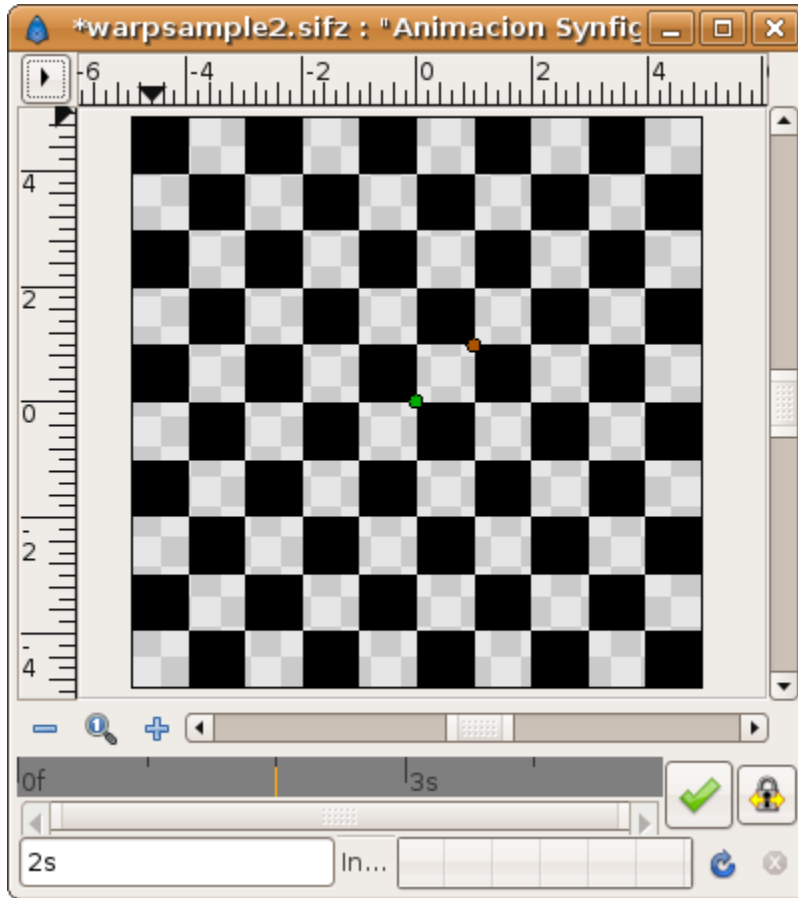
- From 0.0 to 1.0 it renders all the points that are backwards on the perspective (in opposite direction to the vanishing point).
- From 1.0 to +infinite it renders the points that go in the same direction of the vanishing point.

High values of Horizon make Synfig spend a lot of time rendering and the result is slightly visibly better.

7.44.3 Some Samples

Here are a few sample images of the result of applying the warp layer over a *Checkerboard Layer*.

Before distortion:



The handles of the warp layer. The dotted lines represent the Source rectangle. Notice how it corresponds to a 4 by 4 section of the checkerboard:

The resulting distortion (horizon = 15.0 and Clip = off). The destination rectangle can be seen to contain the same 4 by 4 section of the checkerboard that the source rectangle contains:

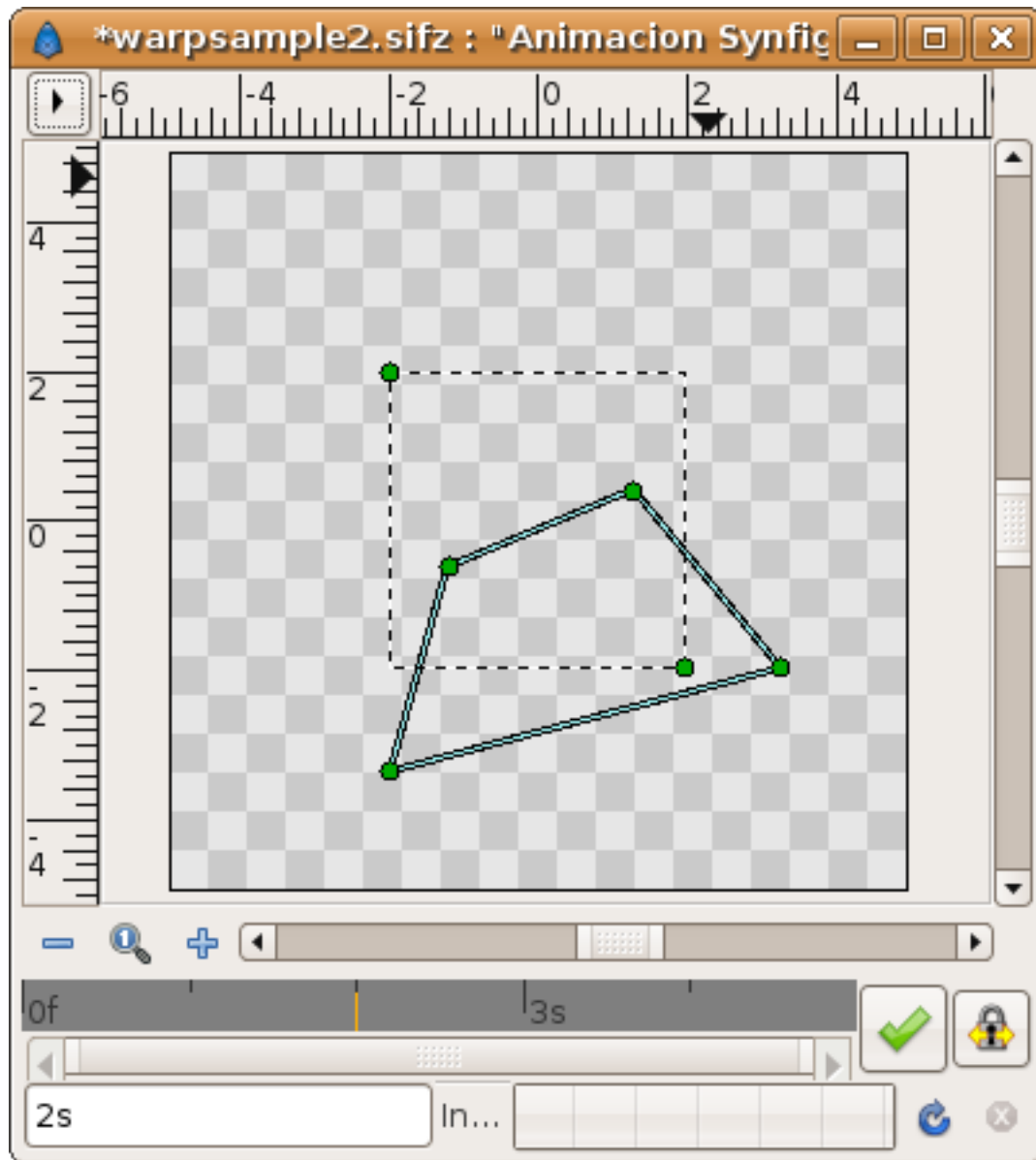
You can see that the rendered horizon is the result of connecting the two vanishing points of the perspective distortion. In this case there are two vanishing points given by the intersection of the lines that connects the following points:

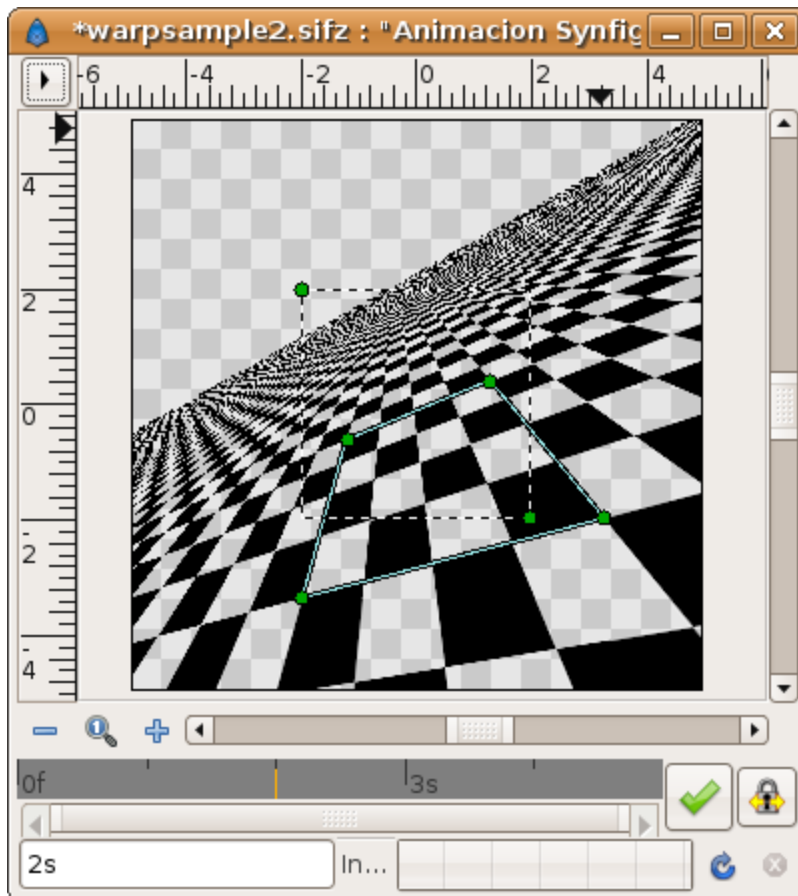
- Intersection of the line that passes by Destination TL point and Destination TR point with the line that passes by Destination BL point and Destination BR point. It gives vanishing point V1 (not shown in the diagram because it is outside the visible area, far off to the left).
- Intersection of the line that passes by Destination TL point and Destination BL point with the line that passes by Destination TR point and Destination BR point. It gives vanishing point V2

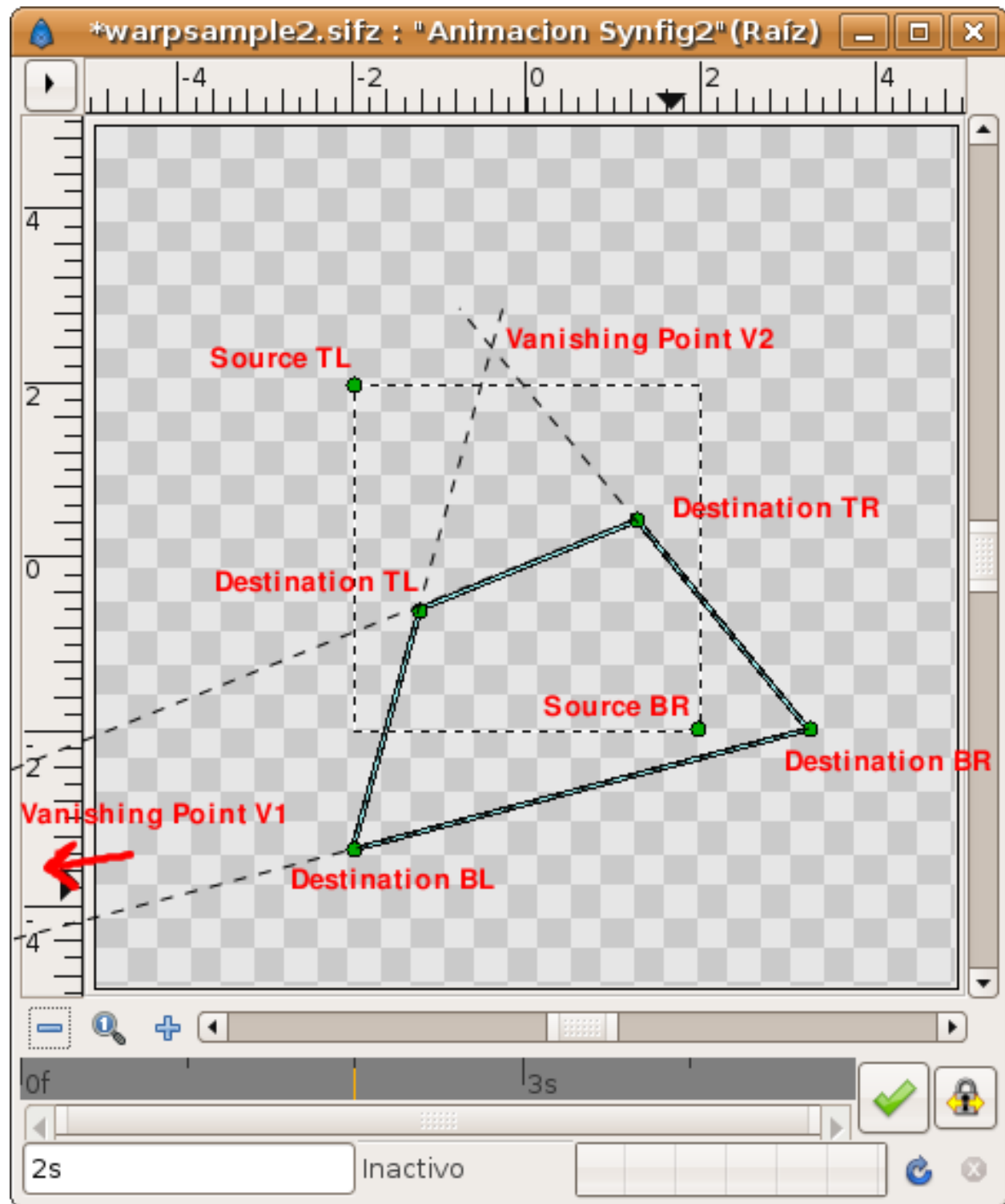
Connecting the Vanishing points V1 and V2 gives the horizon line.

See the diagram:

To understand how the Horizon parameter works see this animated image. It moves the Horizon parameter from 0.0 to 30 in four seconds. Notice that from horizon = 0.0 to 1.0 it renders from minus infinite to the current observer position (it is made very fast in the animation, and takes about in 4/30 seconds and most of the time the rendered section is



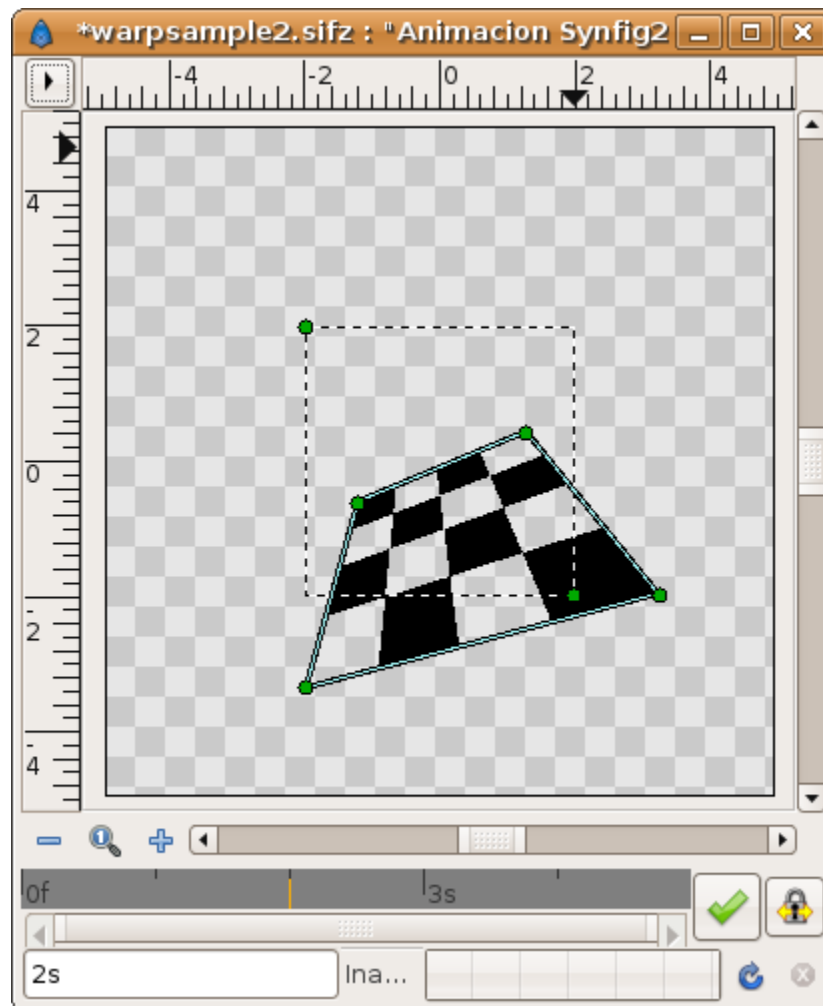




outside the visible area of the canvas - you can see it if you place a zoom out layer on top of them). This means that the portion of plane that is behind the observer point of view is rendered deformed and then very enlarged). Notice too that from horizon = 1.0 to 30.0 it renders the rest of the visible plane. As the horizon parameter gets larger, the final visible horizon gets further away.

7.44.4 The Clip parameter

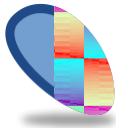
This image shows what you will get when check it on. Only the pixels from inside the Source rectangle are mapped. In this case, the 4 by 4 section of the checkerboard.



7.44.5 Turning warp on/off

Simply set the destination handles to the same position as the source handles.

7.45 XOR Pattern Layer



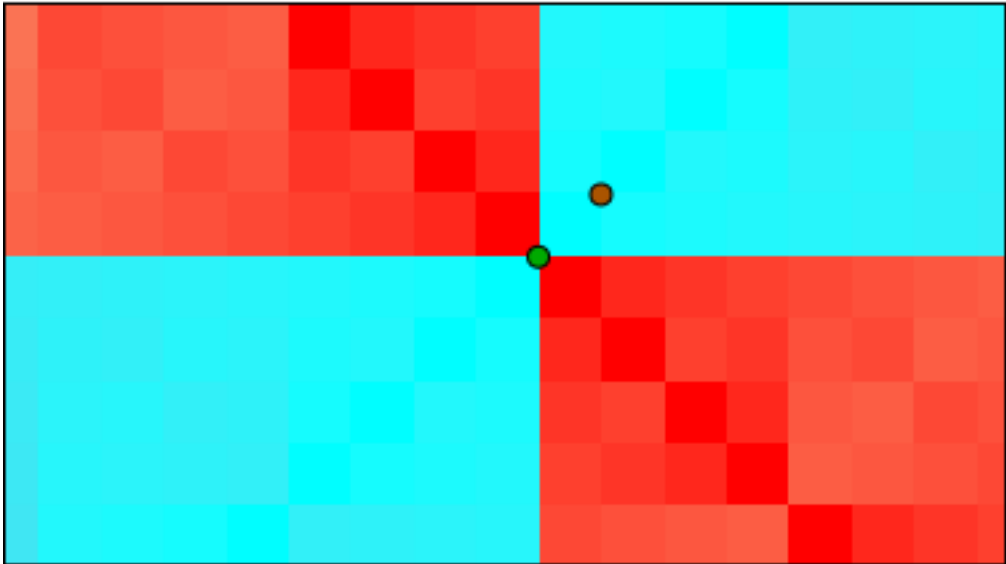
7.45.1 About XOR Pattern Layers

[todo]

7.45.2 Parameters of XOR Pattern Layers

The parameters of the XOR pattern layers are:

Name	Value	Type
π <i>Z Depth Parameter</i>	0.000000	real
π <i>Opacity</i>	1.000000	real
 <i>Blend Method</i>	Composite	integer
 <i>Origin Parameter</i>	0.125000u,0.125000u	vector
 <i>Size</i>	0.250000u,0.250000u	vector

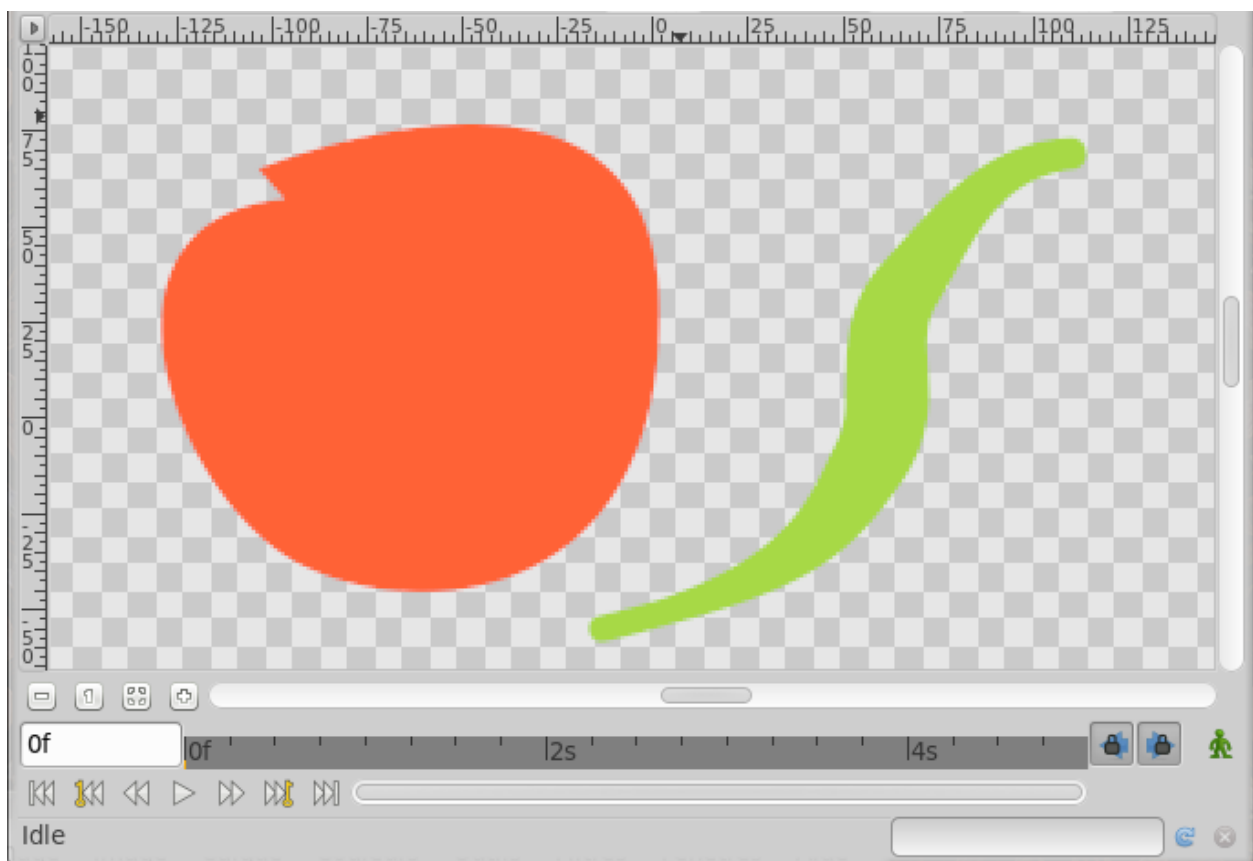


PARAMETERS

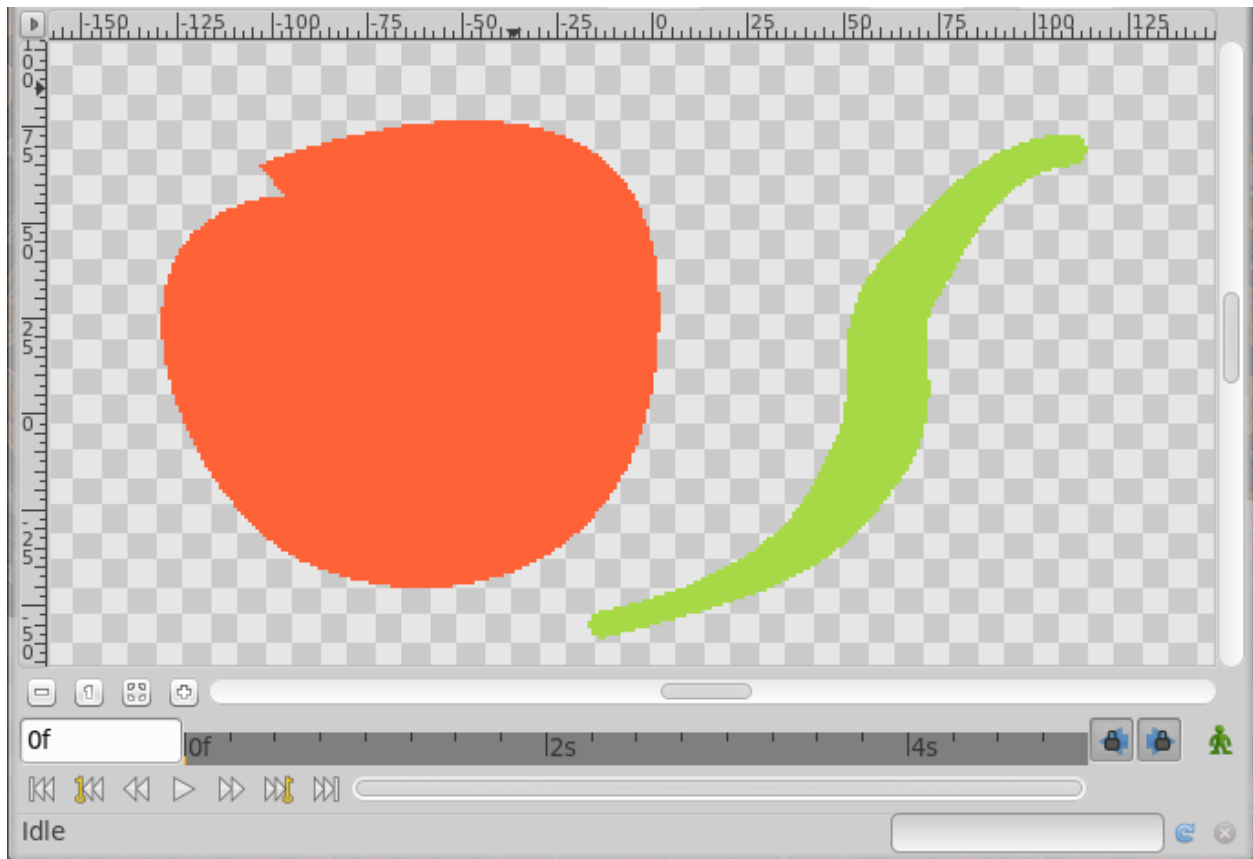
8.1 Antialiasing

The Antialiasing Parameter is a boolean parameter that activate the [antialiased](#) render of the shape (region, outline, etc.) what makes the edge of the shape to not be crisped against the background. When unchecked the antialiasing effect disappear and the shape is rendered more crisped.

8.1.1 Antialiasing on



8.1.2 Antialiasing off



8.2 Blend Method Parameter

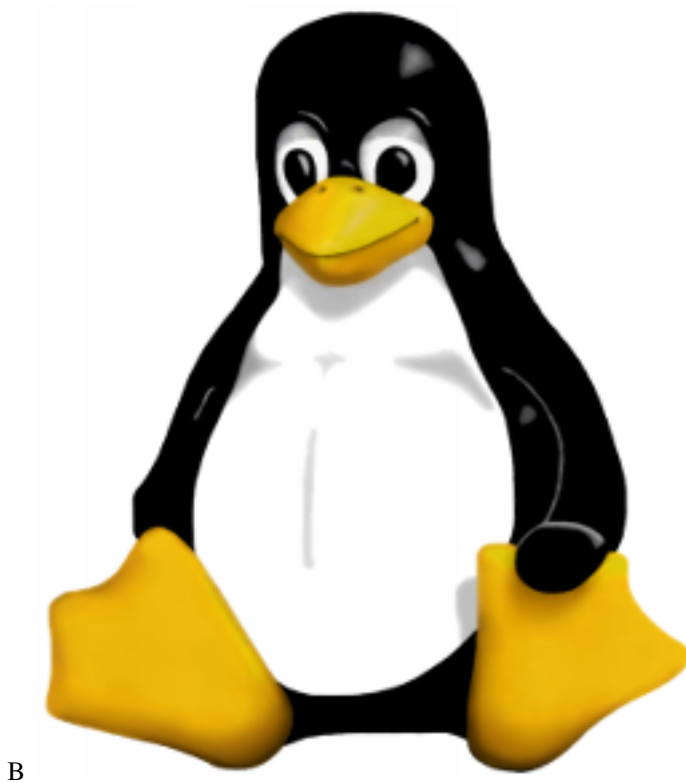
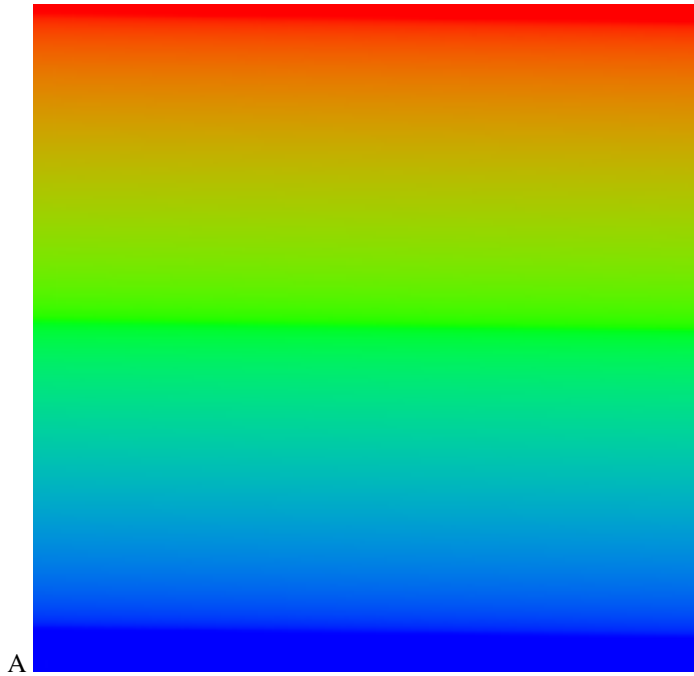
8.2.1 General informations

You can select the Blend Method of a layer during his creation in *Tool Options Panel* and change it from *Parameters Panel*. By default, Blend Method parameter is Static Parameters. To animate it over the time, right click on it and choose Enable animation.

Selecting a layer Blend Method changes the appearance of the layer or composition, based on the layer or layers beneath it. If there is only one layer, the blend method has no effect. There must therefore be at least two layers in the composition to be able to use blend method.

In the following descriptions, 'A' refers to the color on the layer with the blend method setting, and 'B' refers to the color on the layers beneath it. Note that in almost all layers, the alpha channel of the colors will have a scaling effect on the blending. The *Opacity* will also have a scaling effect. In most descriptions these 2 details have been glossed over.

In the examples that follow a gradient (A) is blended on the image of Tux penguin (B). The transparent areas are indicated with the checkerboard pattern.

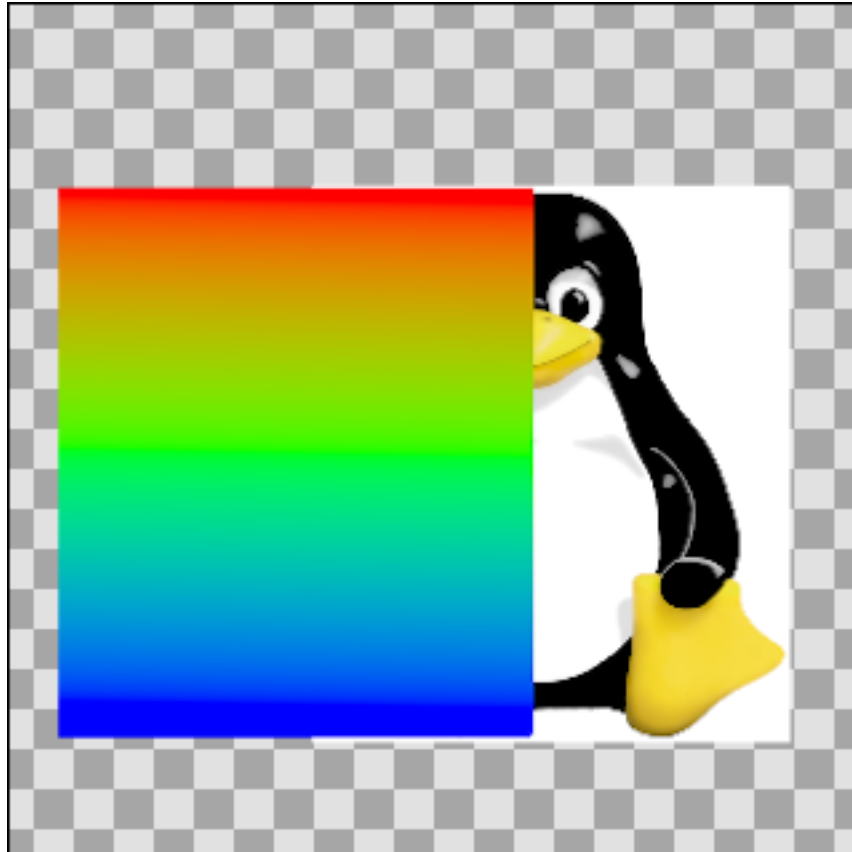


8.2.2 List of available blend methods

Composite

This blend method is the default option, it simply displays the content of the layer : Color A is composited onto B (Taking A's alpha into account).

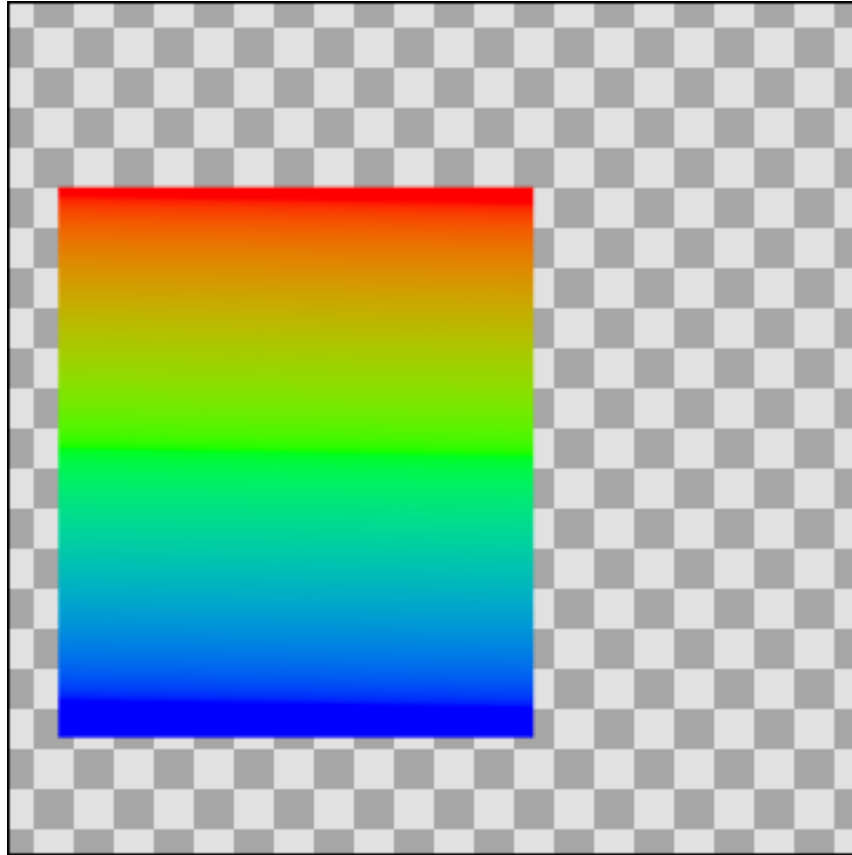
This blend mode is similar the layer blend mode **Normal Mode** often found in 2D programs.



Straight

This blend method looks similar to the previous one, except that the objects under a *straight*-object will be invisible. So if there is a more or less transparent object on the *Straight mode* layer, the objects on the layers underneath won't show through it.

More precisely, the resulting color is $(A-B)*\text{amount} + B$. So if amount is 1 the result is A and if amount is 0 the result is B. In particular, if amount is 1 and A is a very transparent color, the resulting color will also be A; despite the fact that A is very transparent, none of B's color is used.



Onto

If a layer is set to the *Onto* blend method, only the parts of the layer that are over a not transparent area will be visible.

Precisely: this is the same as the Composite blend method except that the transparency of the resulting color is set to be the same as the transparency level of layer B.

Straight Onto

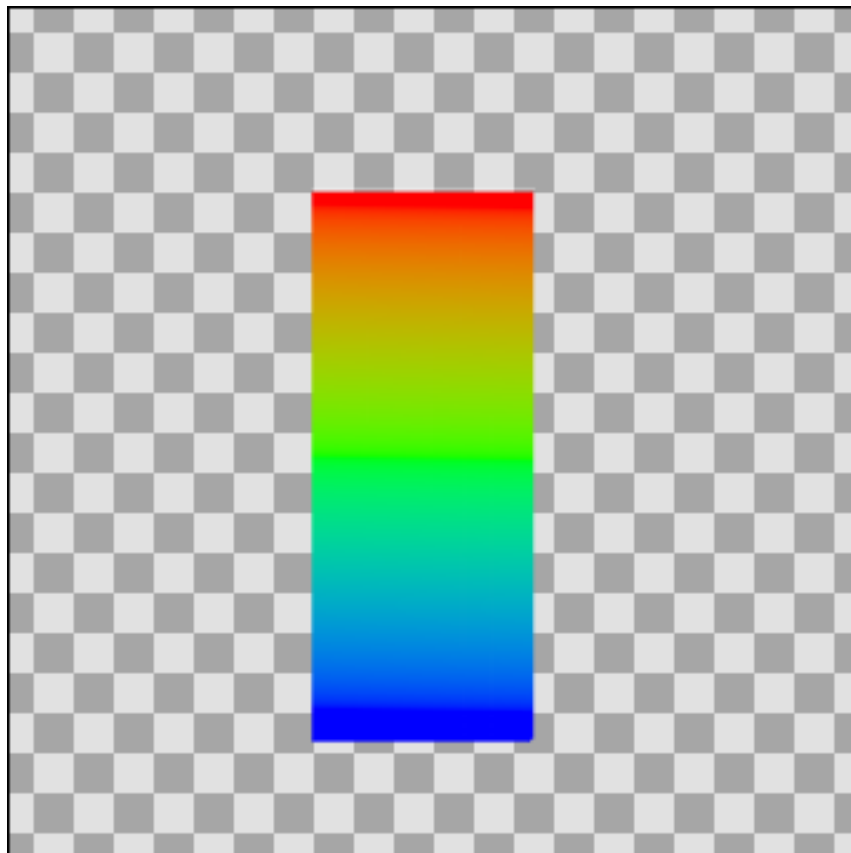
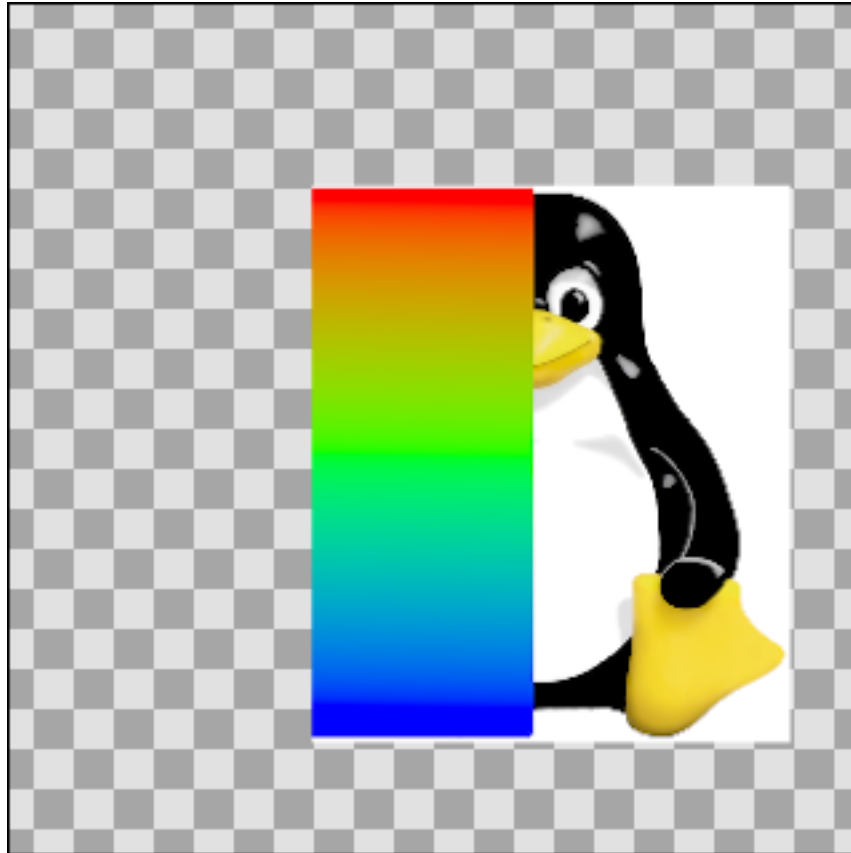
Note: Warning, [actually](#) broken in some cases. [Bug report #868](#) - Straight Onto blend method broken (both render engine) : When using groups result is the same as “Straight” on the workarea and when rendering and cobra (0991d44751f) to render empty on vector artwork.

This method is a combination of the two methods above. E.g. if an half-transparent object is set to *Straight Onto*, it will only be visible over a non-transparent area, and the non-transparent part under that object won’t be visible.

Precisely: the resulting color is “(X-B)*amount + B” where X is A but with its transparency set to A’s transparency times B’s transparency.

So if amount is 1 the result is A, but with its transparency multiplied by that of B, and if amount is 0 the result is B. In particular, if amount is 1 and A is a very transparent color, the resulting color will be a possibly more transparent version of A; despite the fact that A is very transparent, none of B’s color is used in the result.

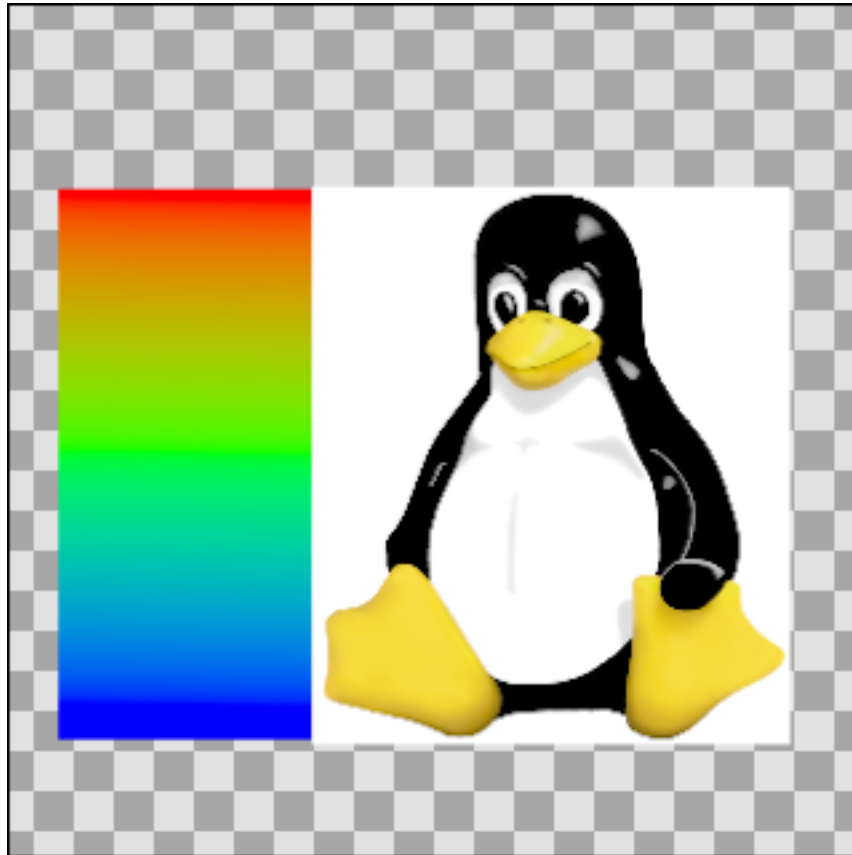
(Yuck. Are these ‘precisely’ comments useful?)[Yes!]



Behind

This blend method makes the layer visible over transparent areas, and invisible over non-transparent areas, giving the impression that the layer is behind the other layers. It is often used for the “Shade” effect layer, to make a drop-shadow effect.

Precisely: this is the same as the composite blend method, but with A and B swapped. B is composited onto A instead of A being composited onto B.



Screen

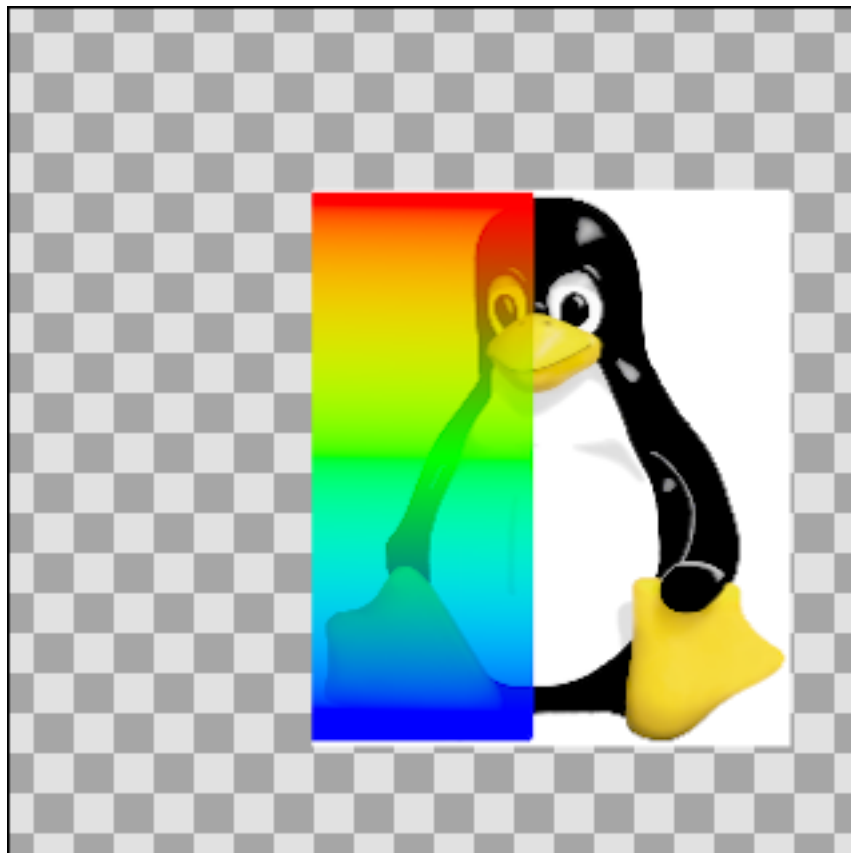
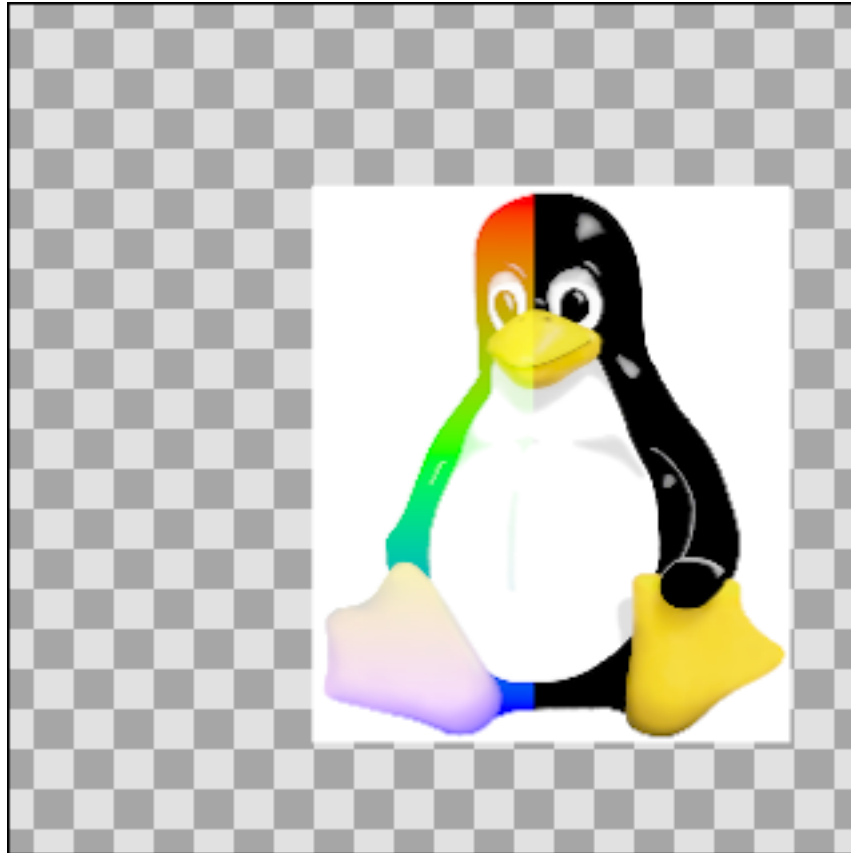
This blend method is similar to the **Screen Mode** often found in 2D programs. It combines the colors of the *screen mode layer* and the ones behind it, and gives a lighter result in general.

Overlay

This is similar to **PhotoShop** - layer blend mode **Overlay Mode**

Precisely: define 3 new colours: $RM = A * B$; $RS = 1 - (1 - A) * (1 - B)$; $RET = A * RS + (1 - A) * RM$ then blend RET onto B as in the Onto method above(!)

Any idea what that is aiming to do? Or what the layer does in this **PhotoShop** program? This appears to emulate the effect of a cross-fade between the two layers if they are set to equal amounts - i.e like ‘add’, but maintaining the overall brightness of the image

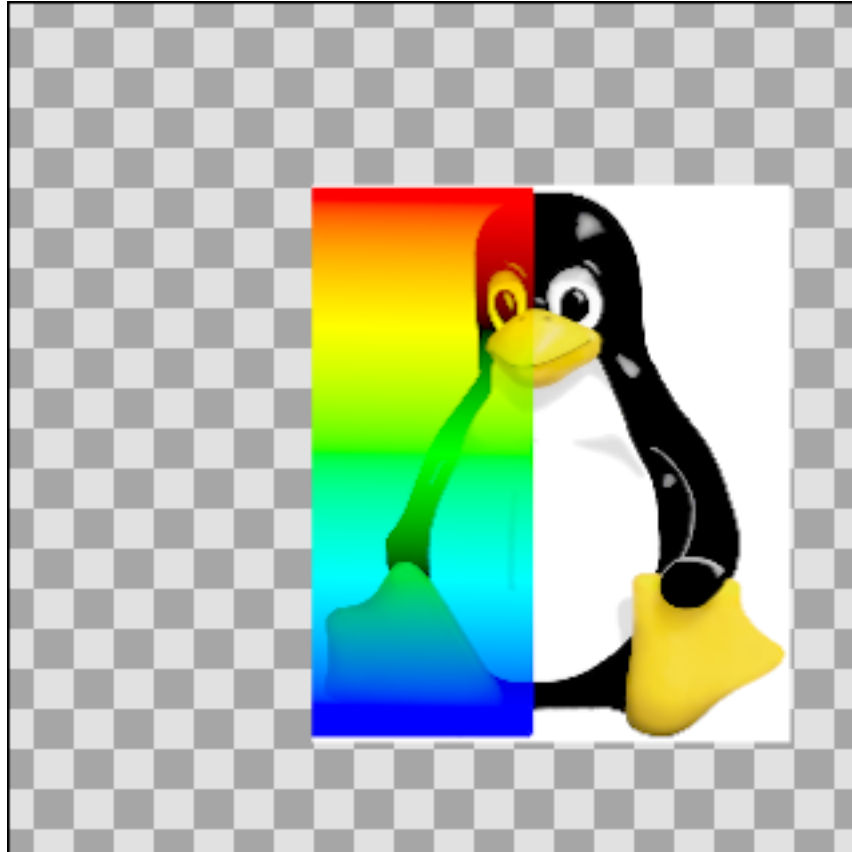


Hard Light

This is similar to **PhotoShop** - layer blend mode **Hard Light Mode**

For each of red, green and blue, if the component is in the top half of its range then calculate $X=1-(1-(2A-1))*(1-B)$, otherwise calculate $X=2AB$, then blend X onto B as in the **Onto** method above.

Is this aiming to make bright colours brighter and dark colours darker?



Multiply

This is similar to **PhotoShop** - layer blend mode **Multiply Mode**

Precisely: the resulting colour is $((A*B)-B)*amount+B$. The calculation is performed independently on red, green, and blue components. When amount is 0, this simplifies to B. When amount is 1 it simplifies to $A*B$.

Divide

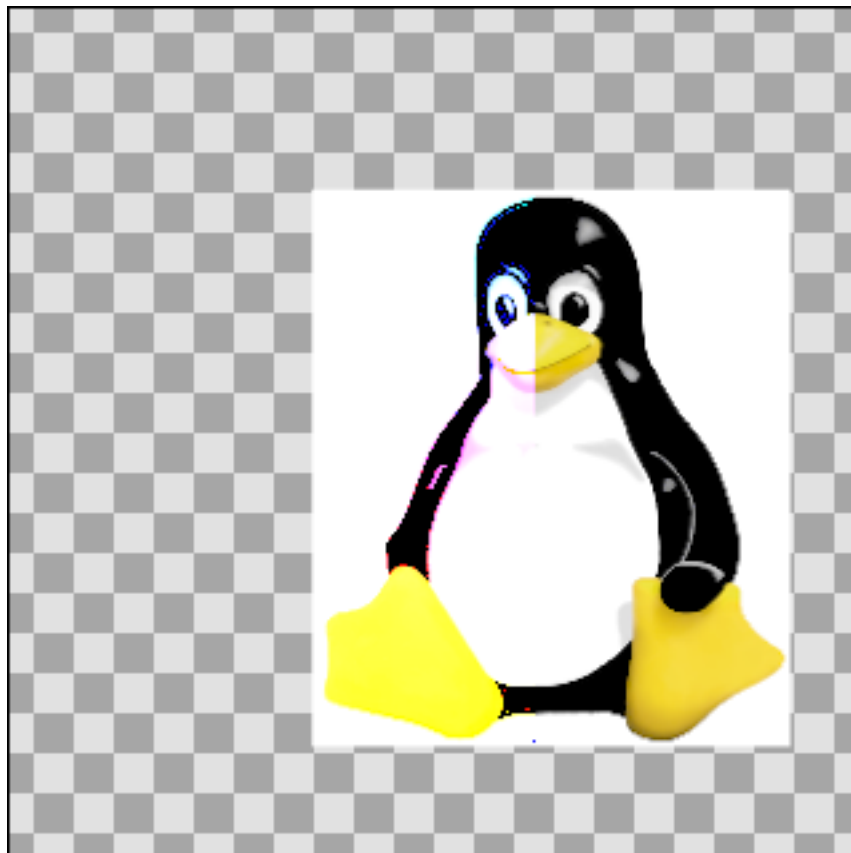
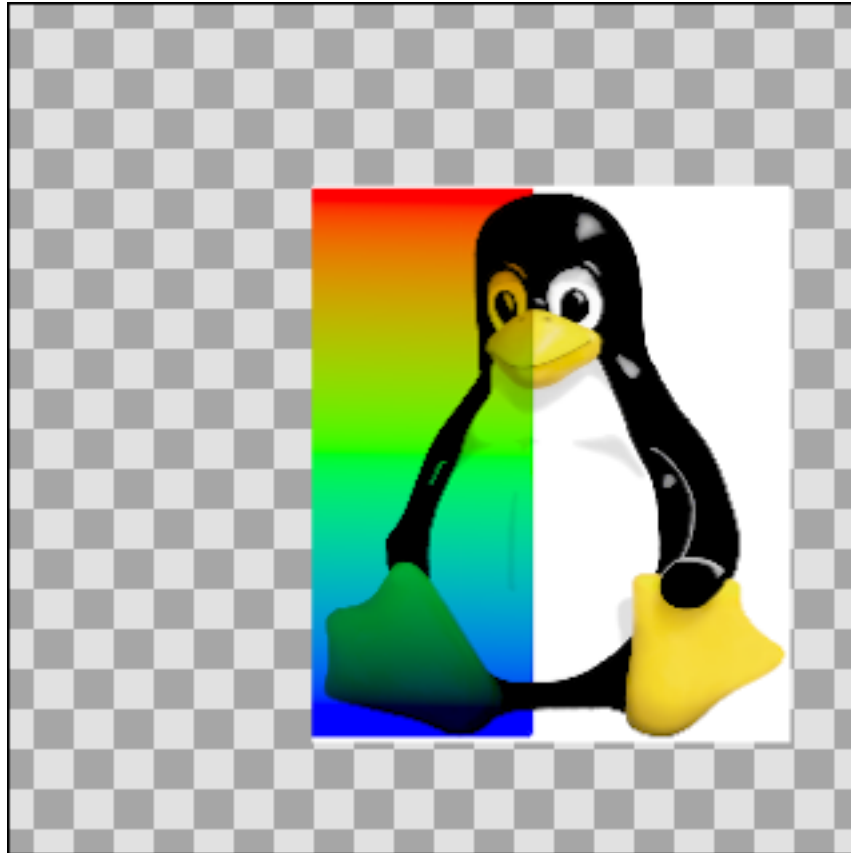
Describe me

Precisely: the resulting color is $((B/A)-B)*amount+B$.

When amount is 0, this becomes simply B.

When amount is 1, this becomes B/A .

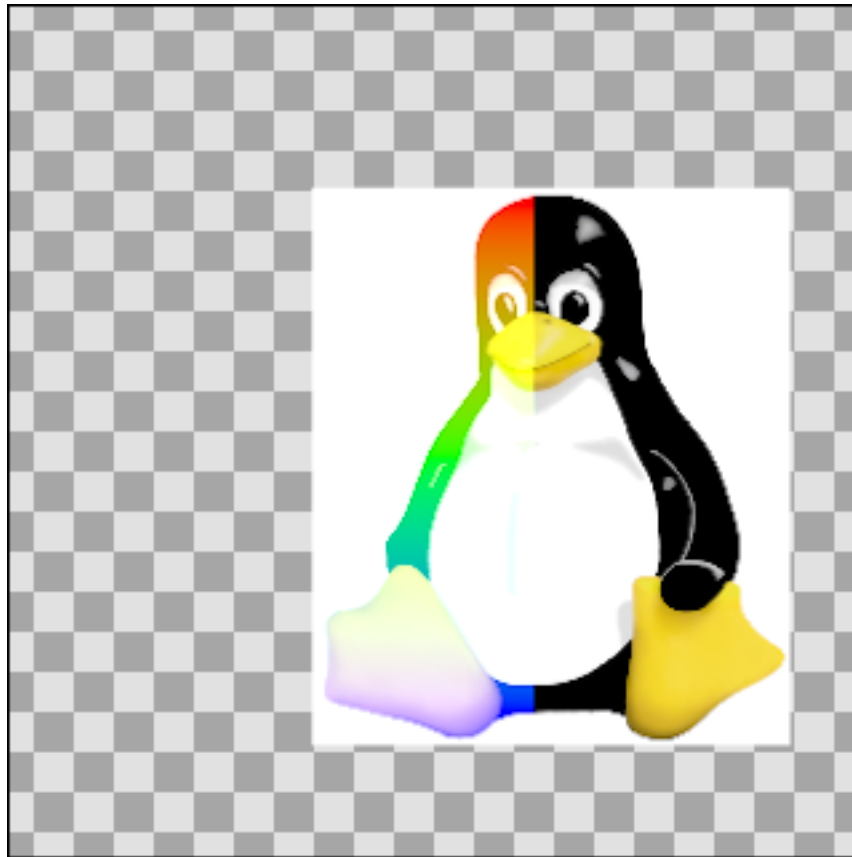
A very small quantity is added to A before dividing by it to avoid a divide-by-zero condition. This causes the divide blend method to bias toward positive values, but the effect is really negligible.



Add

Describe me

Precisely: the resulting color is $(B + A * A.alpha * amount)$. The calculation is performed independently on red, green, and blue components. The resulting color's alpha is $B.alpha$.



Subtract

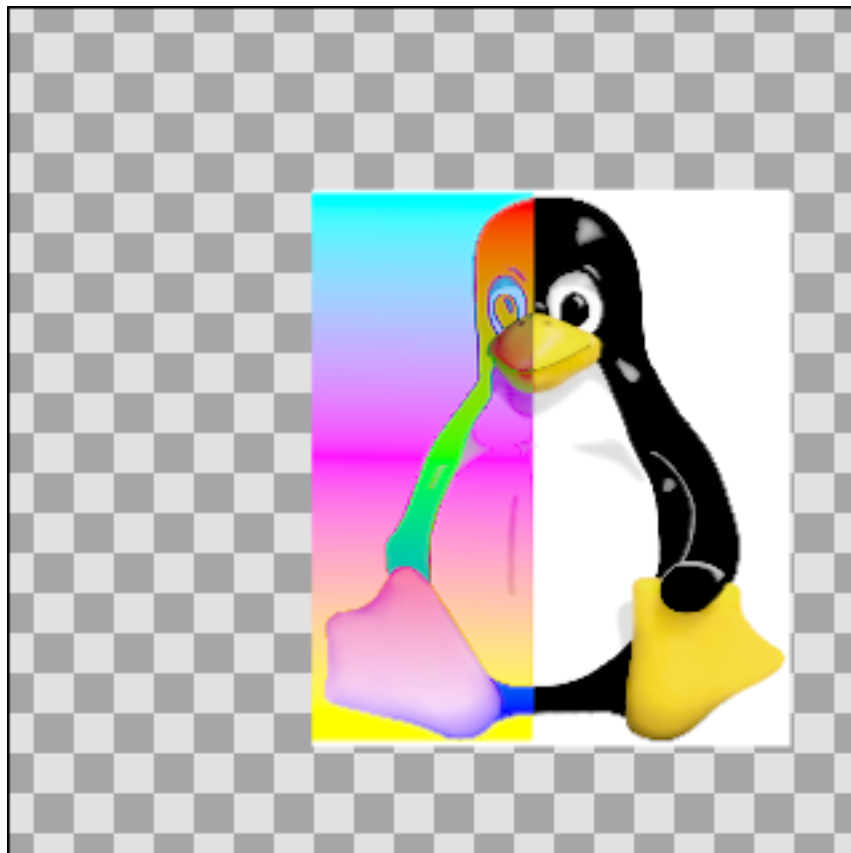
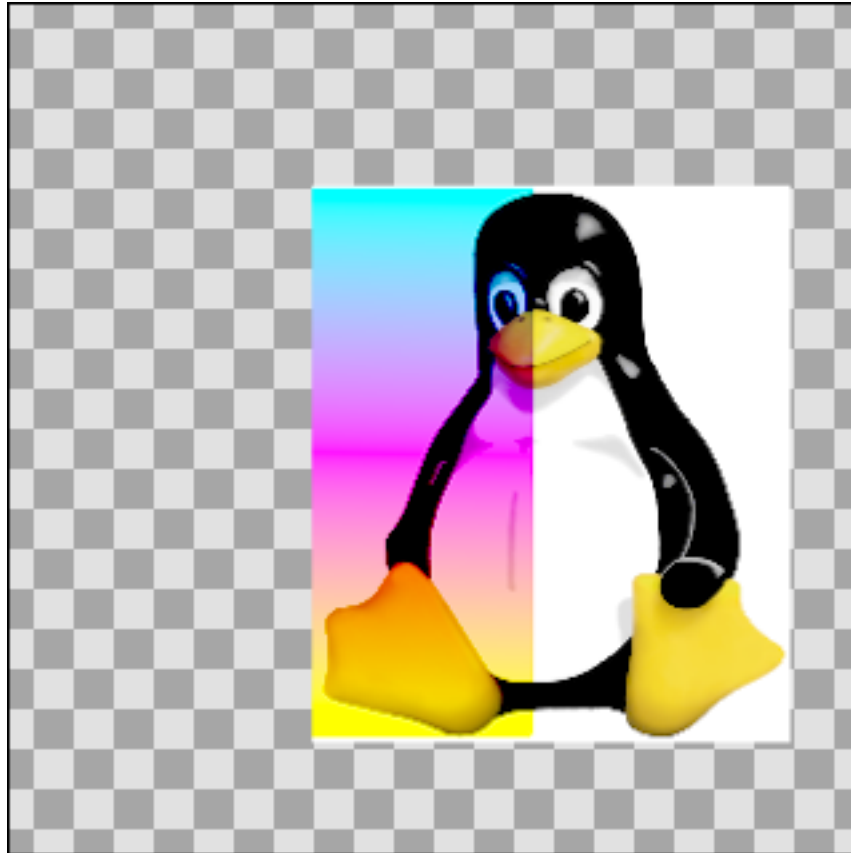
Describe me

Precisely: the resulting colour is $(B - A)$. The calculation is performed independently on red, green, and blue components.

Difference

Describe me

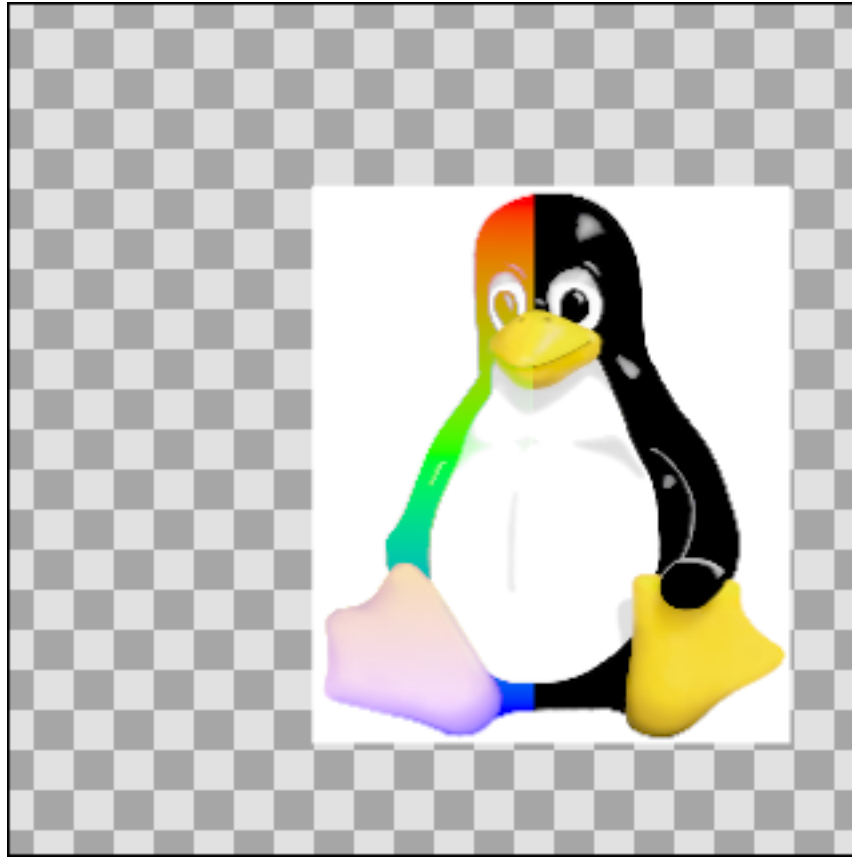
Precisely: the resulting colour is the absolute value of $(B - A)$. The calculation is performed independently on red, green, and blue components.



Brighten

Describe me

Precisely: for each of the red, green, and blue components, compare A's value with B's value and use the higher of the pair.



Darken

Describe me

Precisely: for each of the red, green, and blue components, compare A's value with B's value and use the lower of the pair.

Color

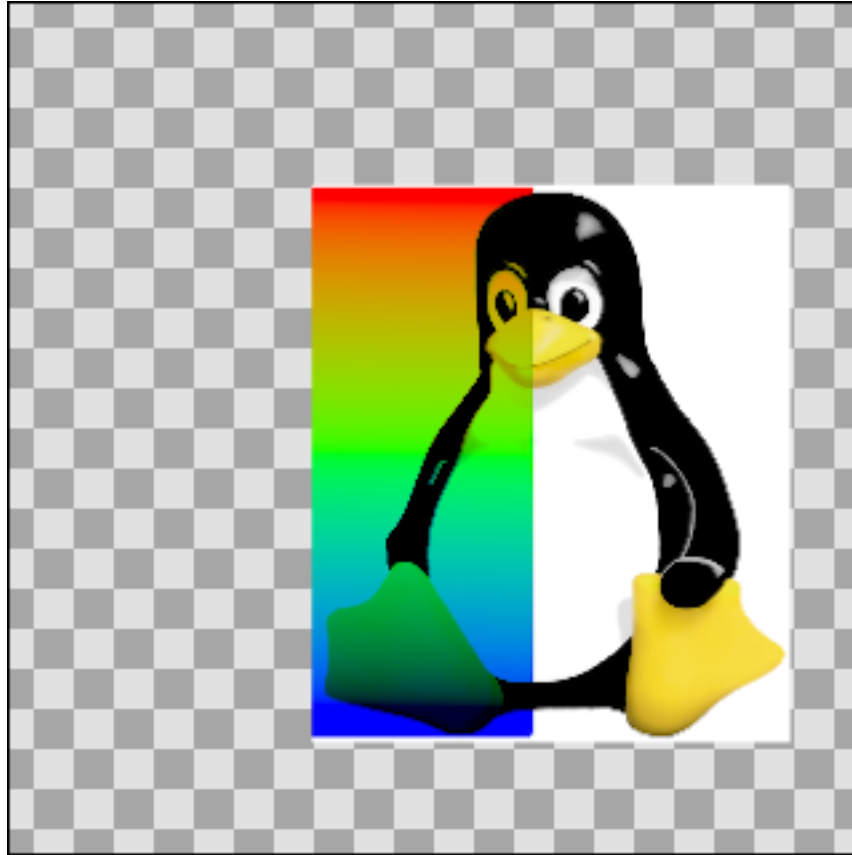
Describe me

Precisely: the resulting colour is obtained by adjusting B to have the same U and V values as A, while keeping Y the same.

straight.png + synfigtux.png = color.png

As this example looks just like the saturation one, perhaps a yellow gradient instead of a white one would be more illustrative

yellowgradient.png + synfigtux.png = coloryg.png



Hue

Describe me

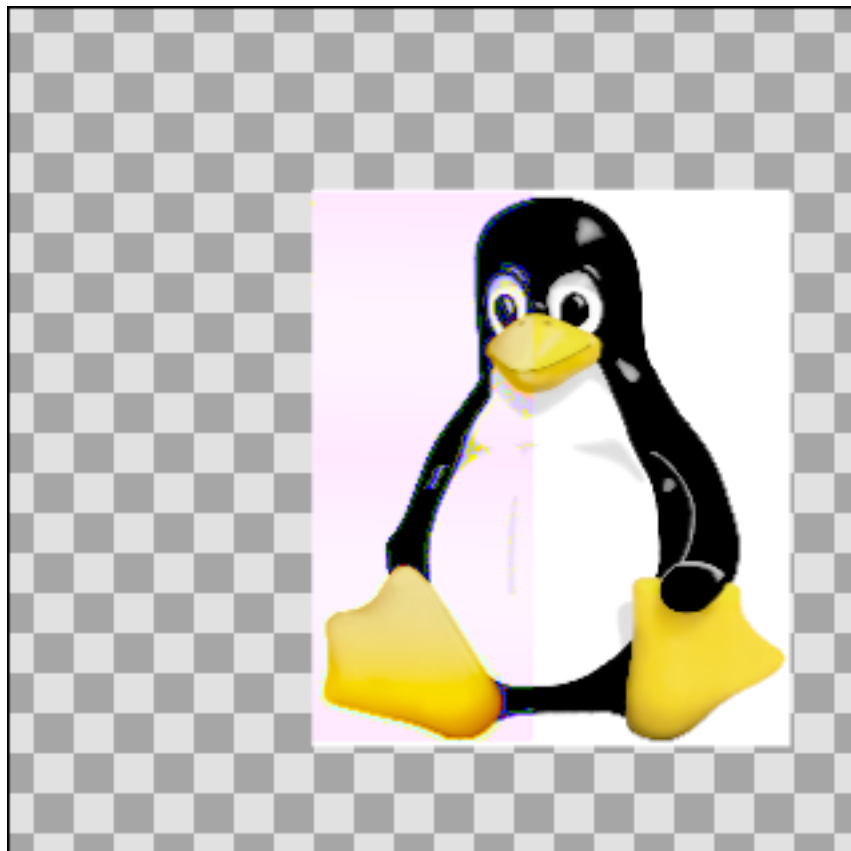
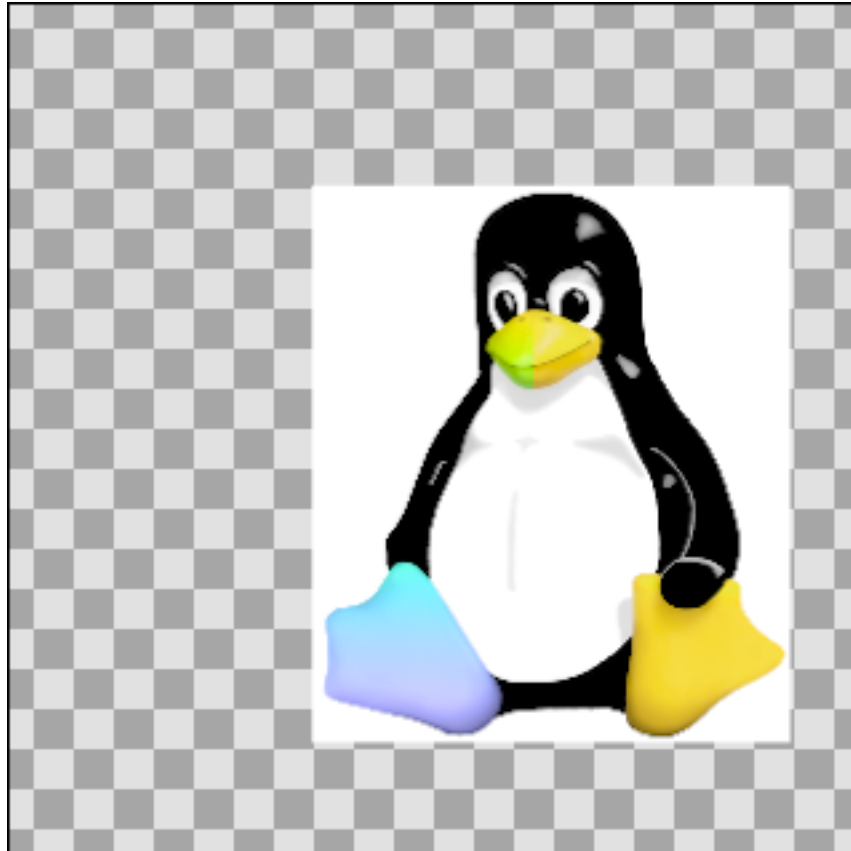
Precisely: the resulting colour is obtained by adjusting B to have the same hue as A.

Saturation

Describe me

Precisely: the resulting colour is obtained by adjusting B to have the same saturation as A. Saturation is the magnitude of the [UV vector](#).

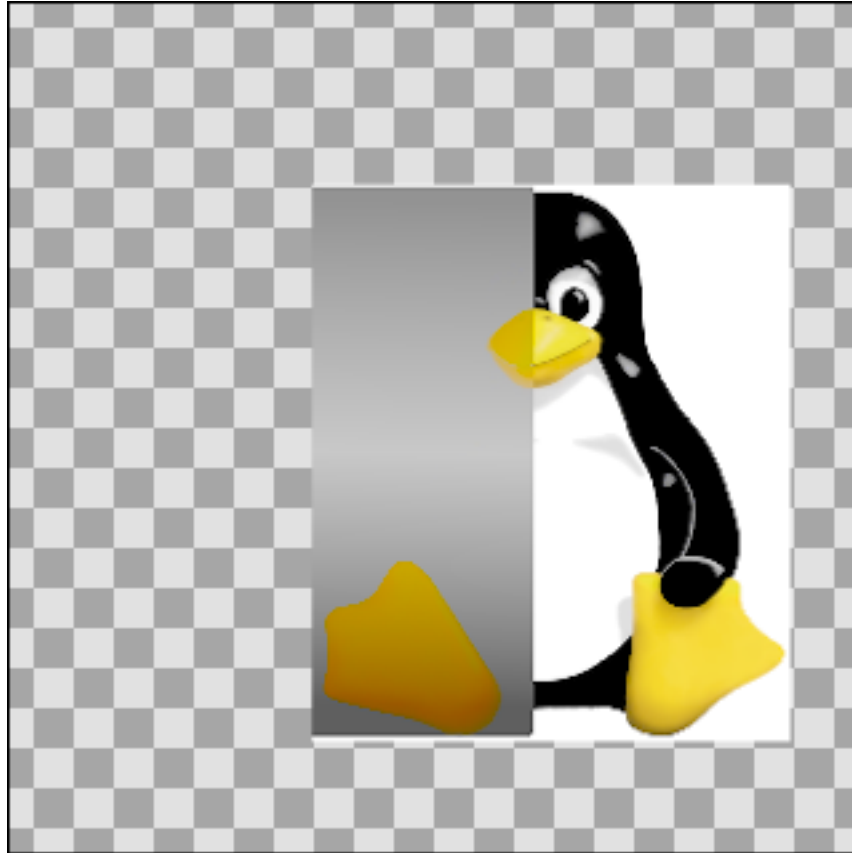
As this example is so similar to the 'Color' blend example, here it is with a yellow gradient -
yellowgradient.png + synfigtux.png = saturationyg.png



Luminance

Describe me

Precisely: the resulting colour is obtained by adjusting B to have the same Y (luma) value as A, while keeping U and V the same.



Alpha over

Layer A act like a mask on B.

Have a look to Doc:Basic_Masking and [|Subtracting_Shapes|](#) tutorials for usage example.

Precisely : multiply alphas and then straight blends using the amount.

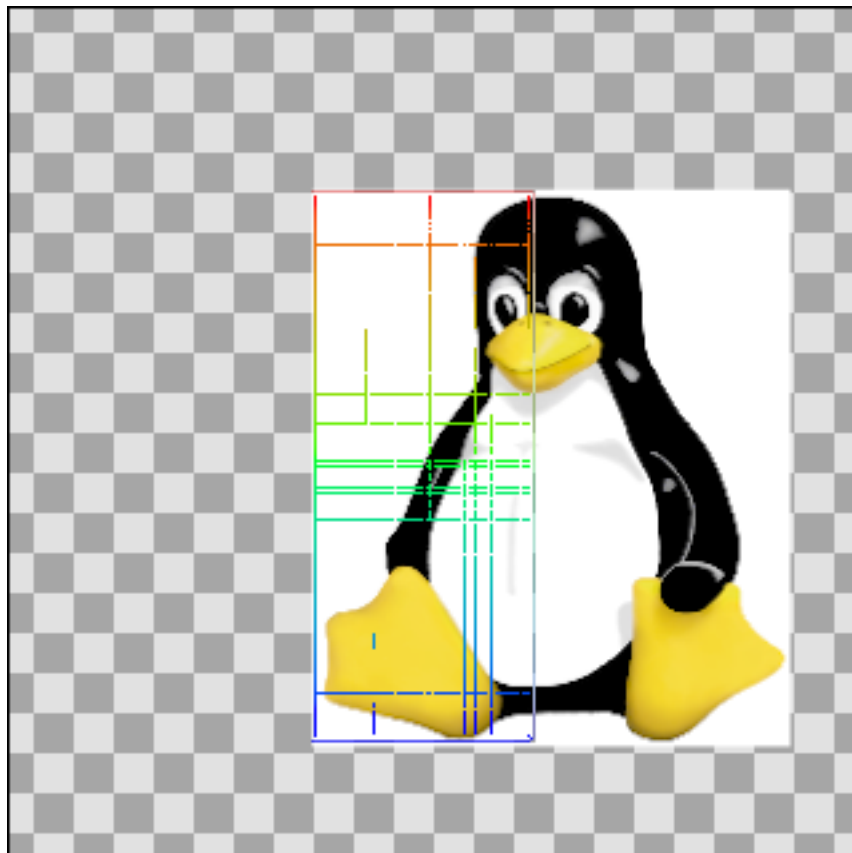
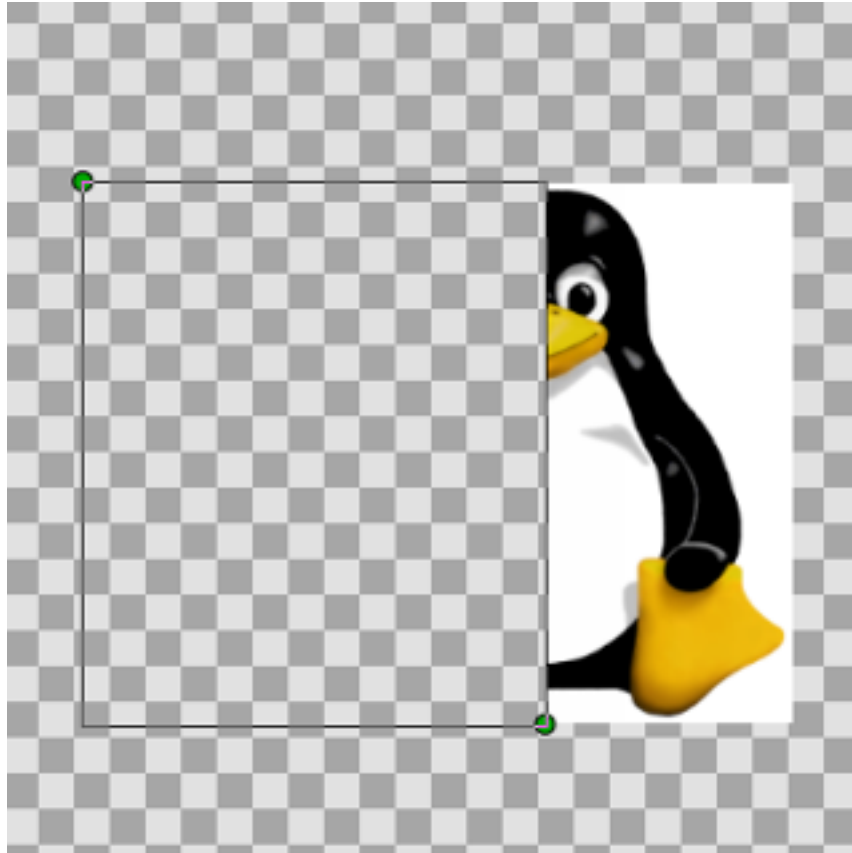
Alpha Brighten

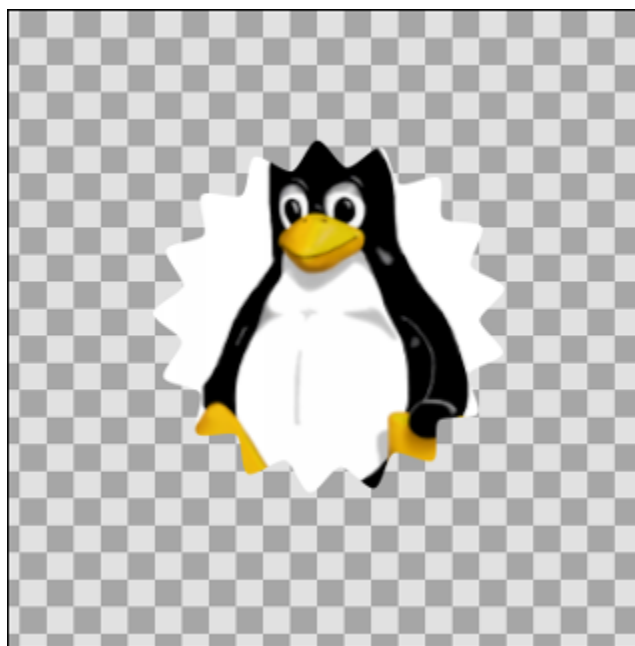
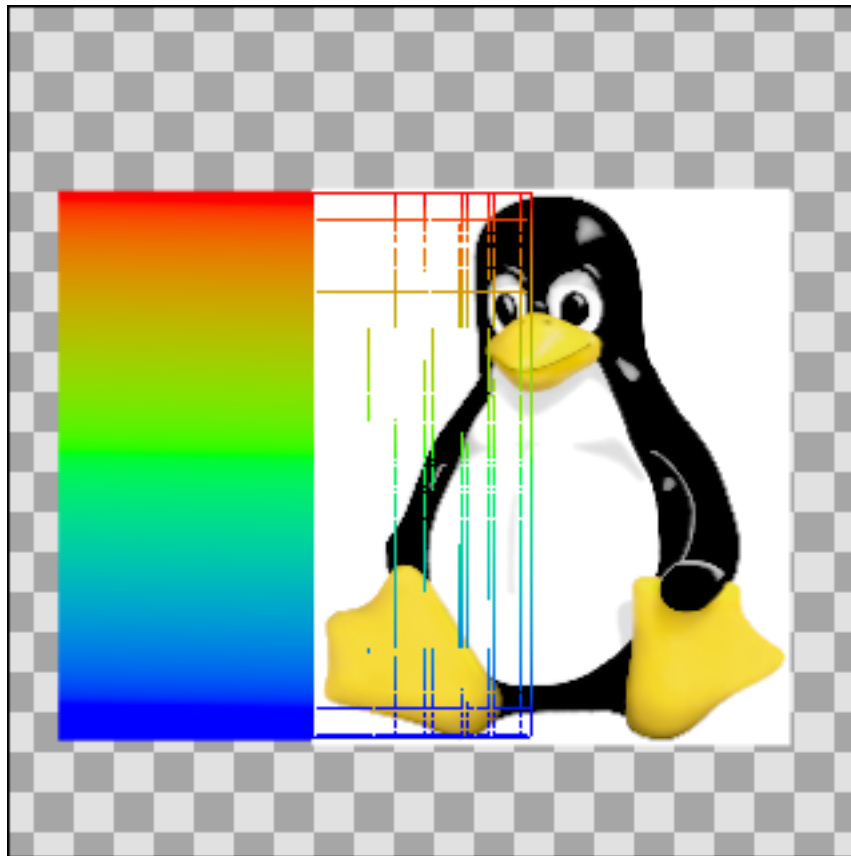
Alpha Darken

8.2.3 Use cases of the Blend Method Parameter

Hide behind a shape

Import Image a picture, add a *Star Layer* up of the image, set the Stars Layer's *Blend Method* to *Blend Method: Alpha Over* and finally check *invert*.





8.3 Feather

8.3.1 About the feather concept

Feather is a float value in points (or whatever unit you have defined in File > Setup > Misc > Unit System) that represents the width of the area that is going to be dissolved at the edge. Feather is the light horny waterproof structure forming the external covering of birds. See [English Feather definition](#).

The Feather parameter is an standard parameter for the following types of layers:

- *Outline Layer*
- *Region Layer*
- *Circle Layer*
- *Polygon Layer*
- *Star Layer*

Strangely *Rectangle Layer* doesn't have the Feather parameter.

When you apply a Feather to any of those layers, then the edge of the shape becomes dissolved and spread out and shrunk the amount that the Feather value indicates.

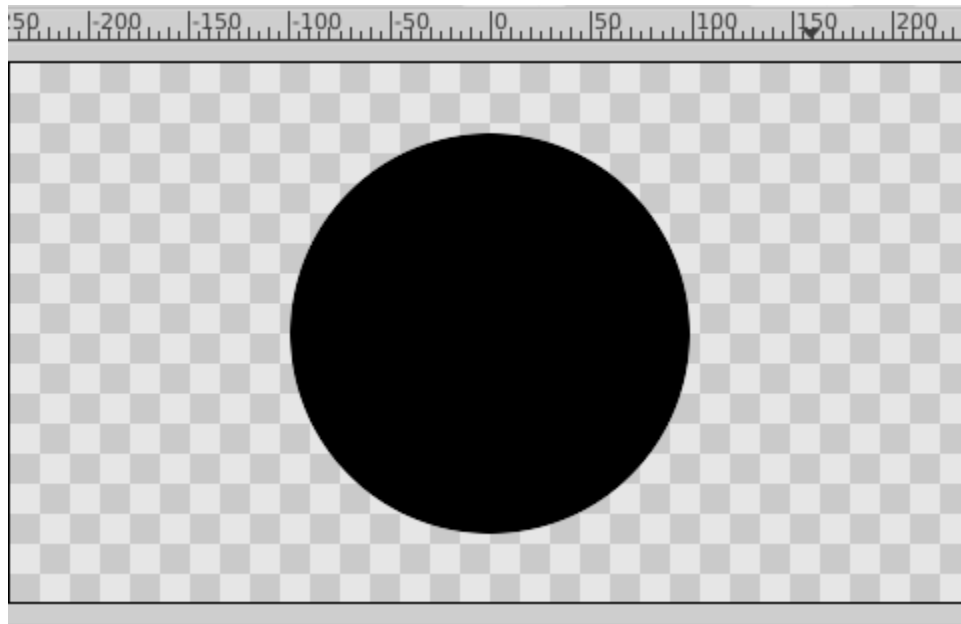


Fig. 1: Circle with Feather set to 0 points

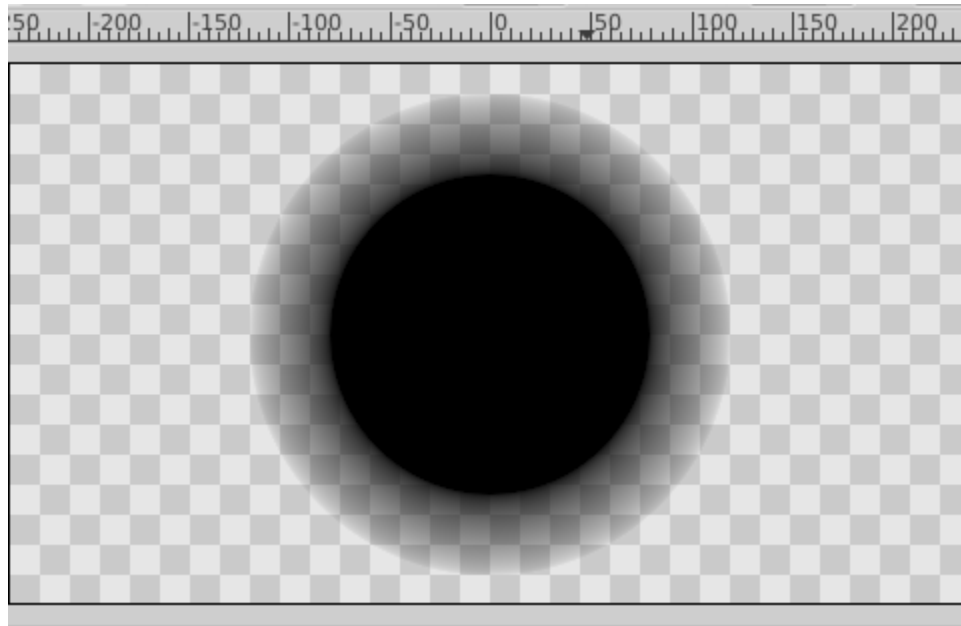


Fig. 2: Circle with Feather set to 20 points

8.3.2 Feather's Complementary Parameters

There is another parameter that works together with the Feather parameter, and which specifies the type of feathering to be used. On Circle layers the parameter is called *Fall Off* and for the other featherable layers it is called *Type of Feather*.

8.3.3 Type of Feather

This complementary parameter can be set in Outline Layer, Advanced Outline Layer, Region Layer, Polygon Layer and Star Layer. All the example images below have the same amount of Feather (set to 20 points).

8.3.4 Fall Off

That's the type of feather for the Circle Layer. All the example images below have the same amount of Feather (set to 20 points).

8.4 Focus Point

The “Focus Point” parameter is present only in the *Group Layer* layer. It has a Vector value. It defines which point in the grouped layers will stay still as the Group layer's *Zoom Parameter* changes.

The Focus Point is relative to the Group layer's *Origin* and defaults to (0,0) so its handle will be hidden behind the green Origin *Handle* initially. Its handle can be seen by turning the Origin handles off temporarily (by hitting Alt1) or by editing its value to something small by non-zero in the *Parameters Panel*.

It's best to edit the Focus Point parameter while the Group's Zoom is set to 0. Otherwise you'll be seeing a scaled version of the contents.

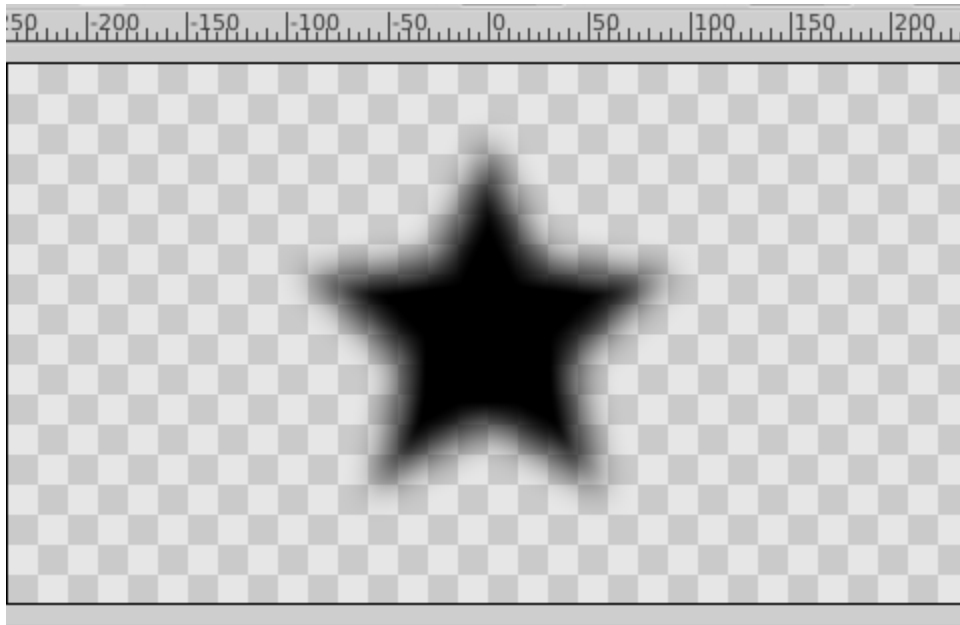


Fig. 3: Fast Gaussian Blur

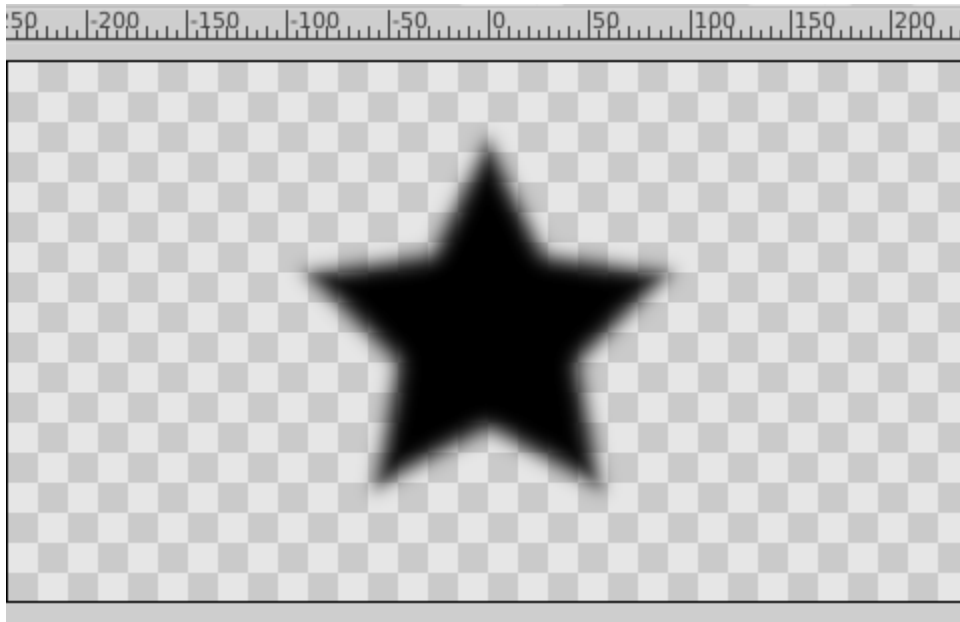


Fig. 4: Gaussian Blur

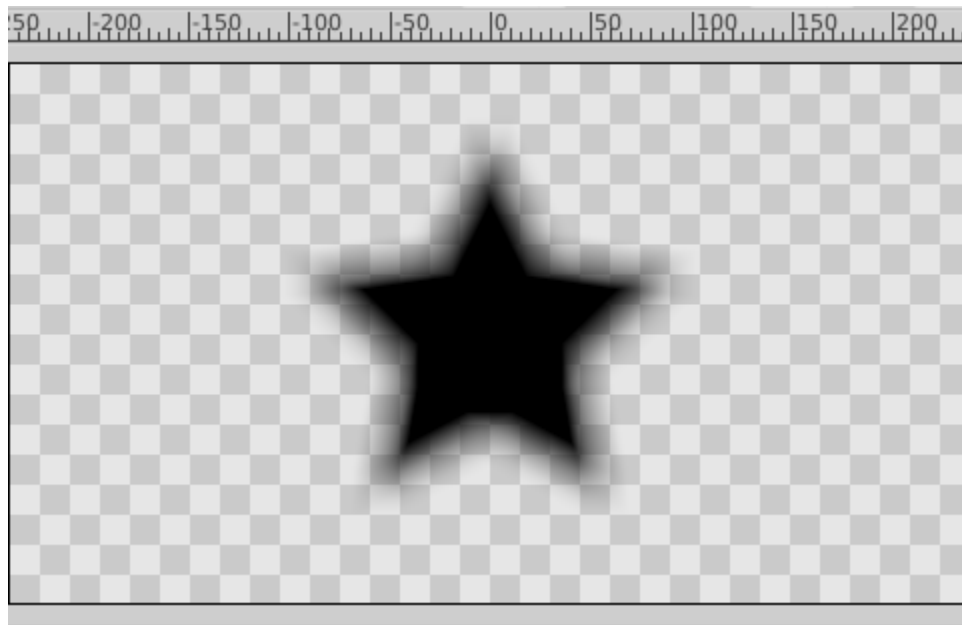


Fig. 5: Box Blur

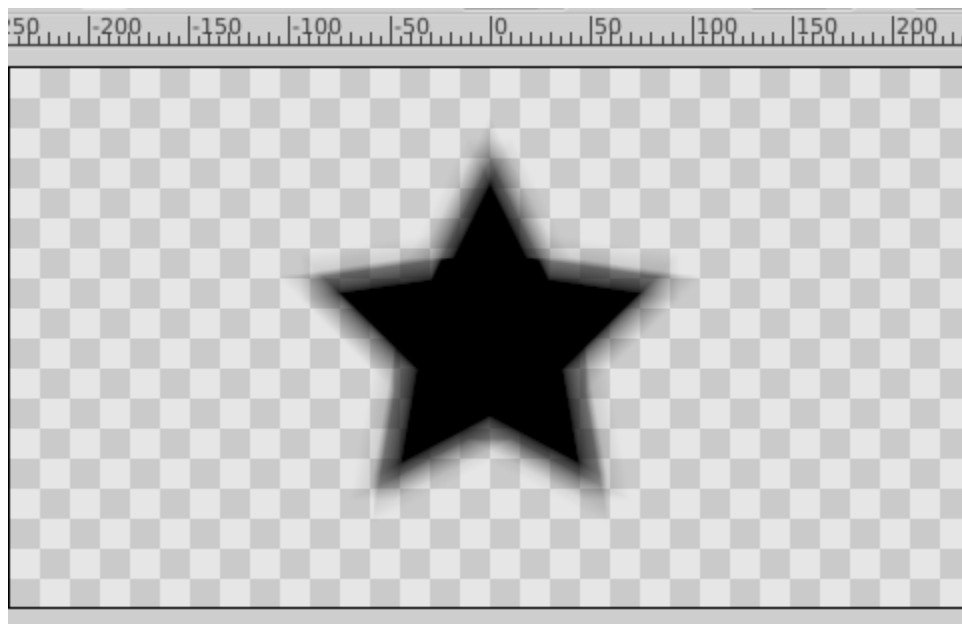


Fig. 6: Cross-Hatch Blur

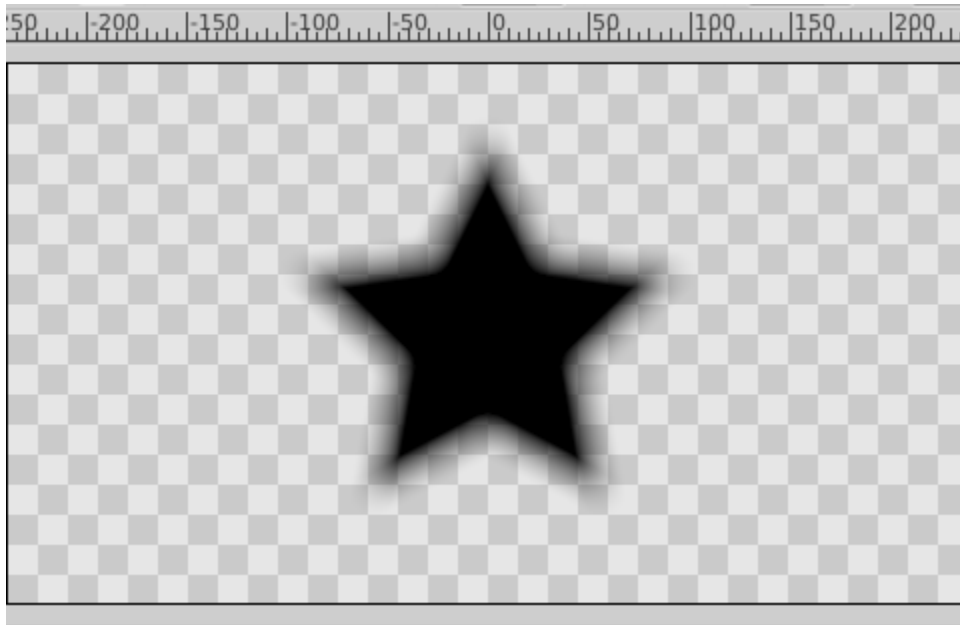


Fig. 7: Disc Blur

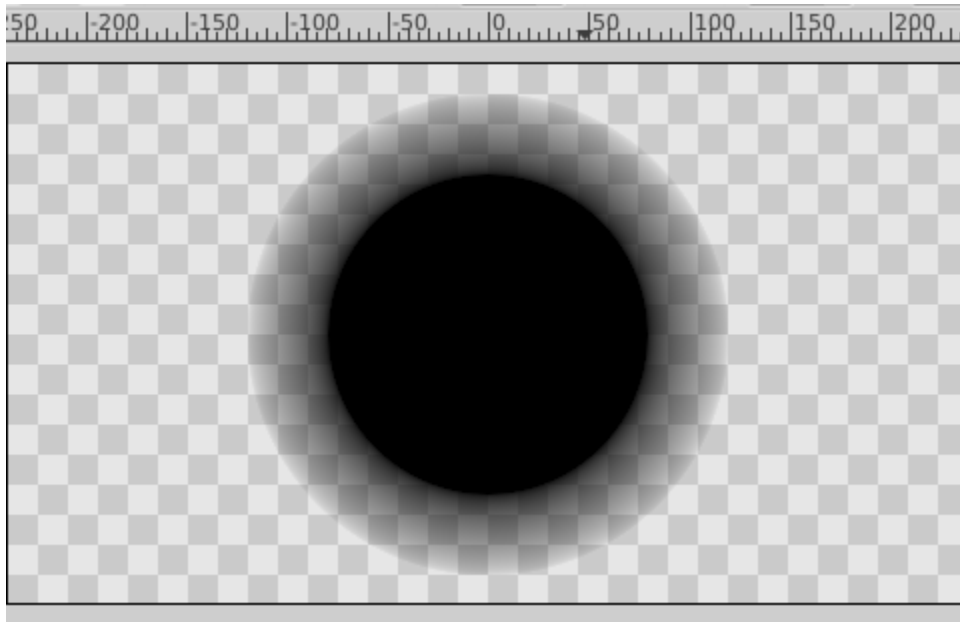


Fig. 8: Square Root

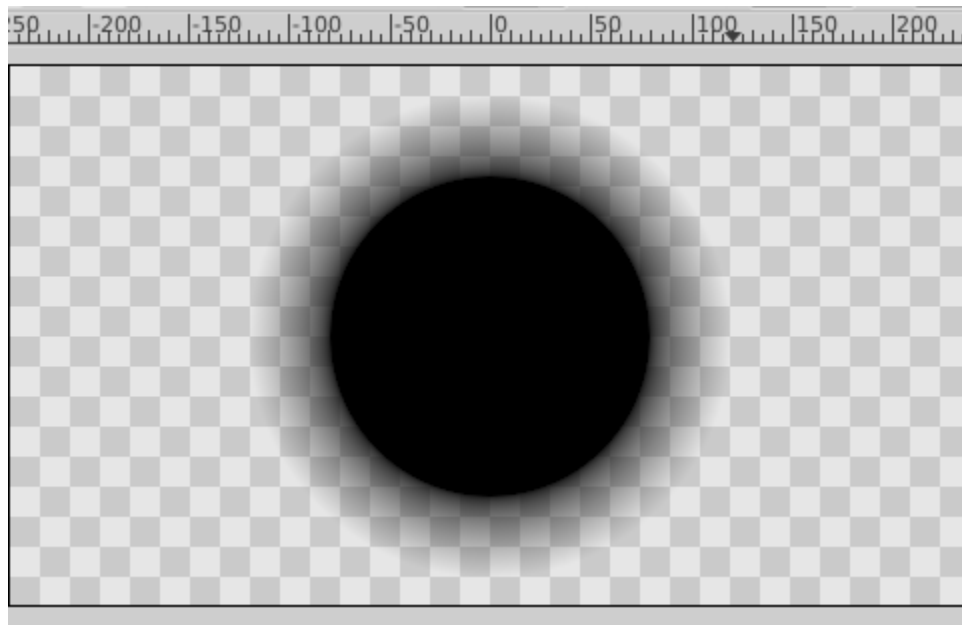


Fig. 9: Square

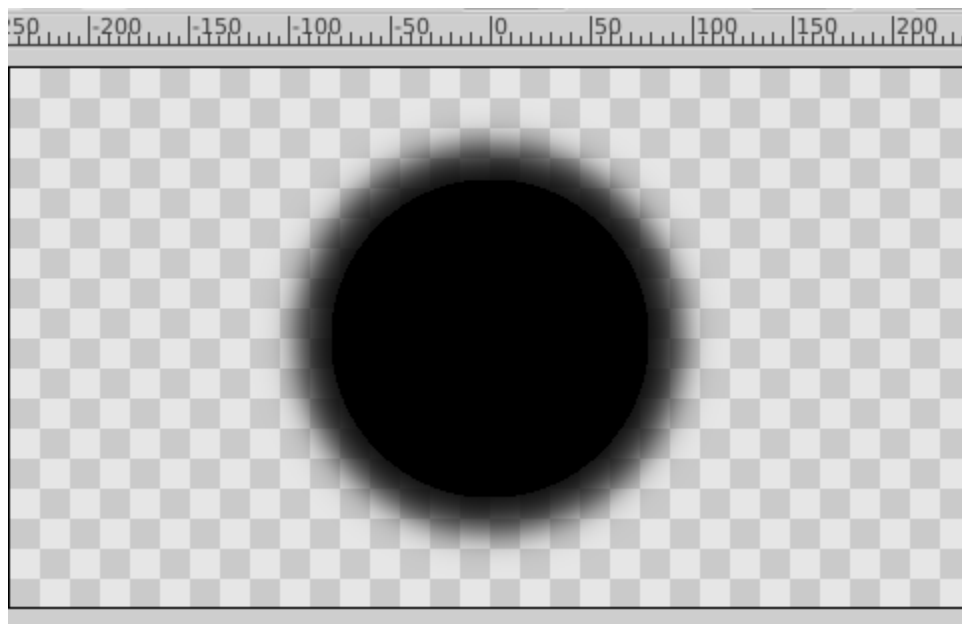


Fig. 10: Sigmond

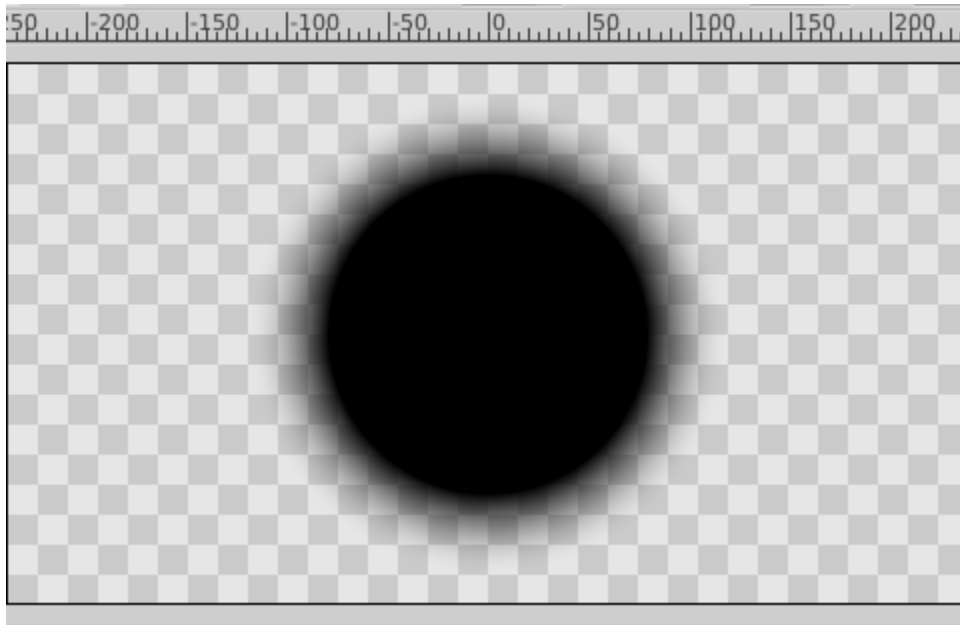


Fig. 11: Cosine

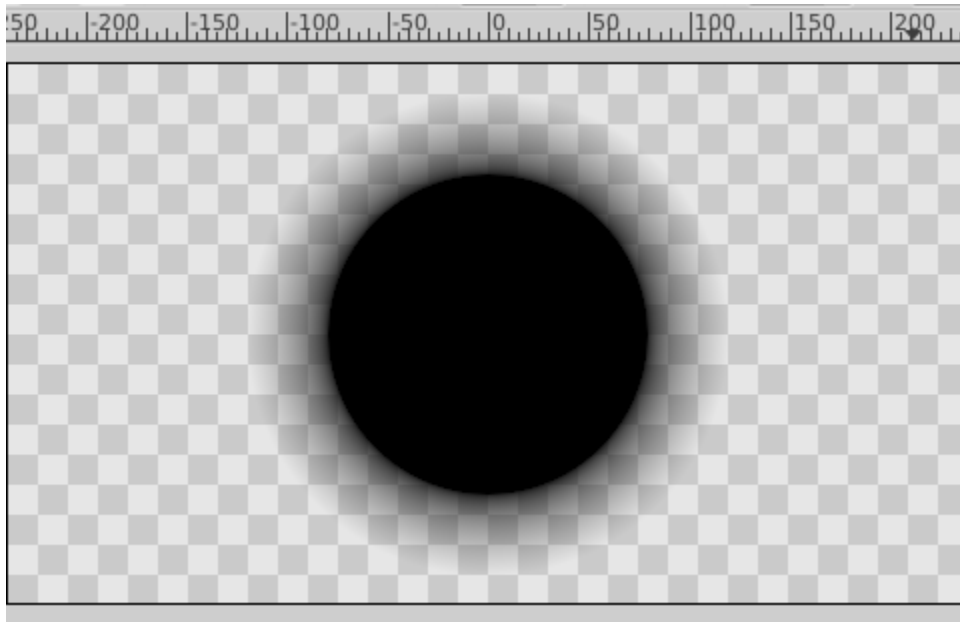


Fig. 12: Linear

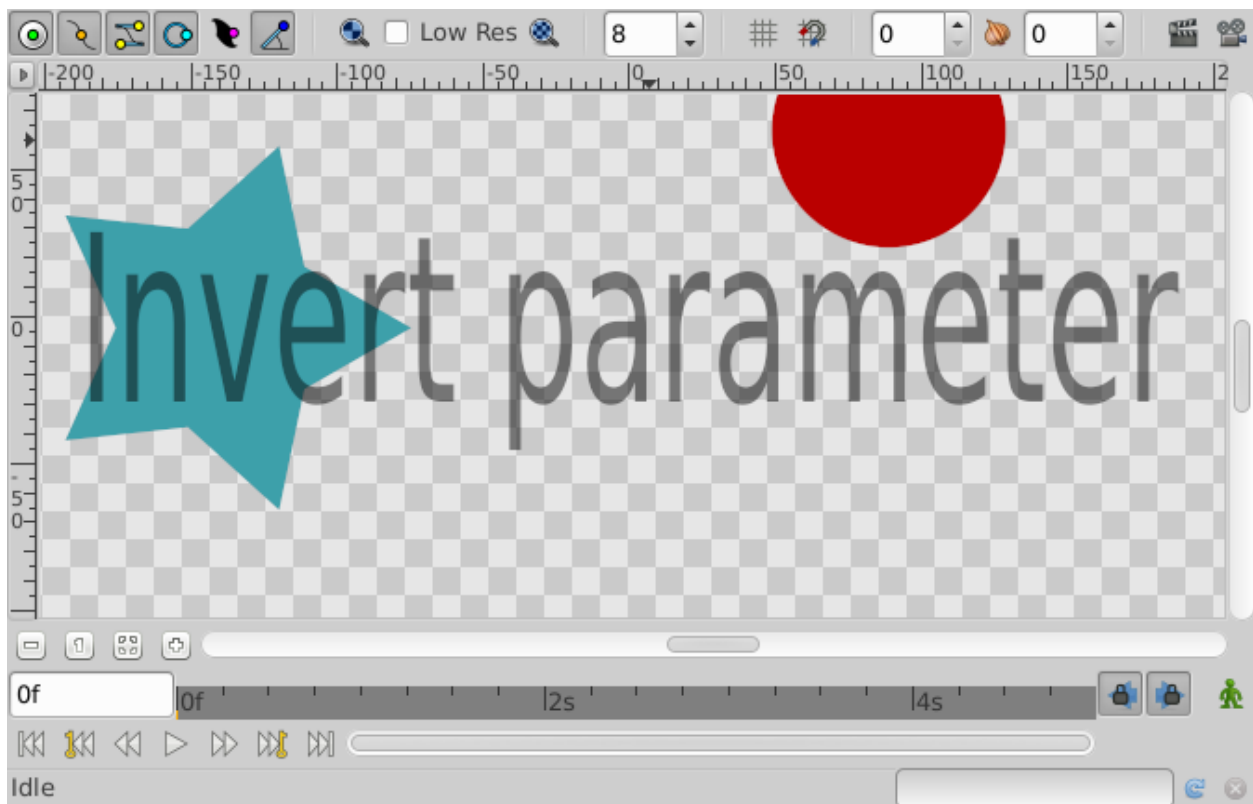
8.5 Invert

Invert is a boolean property that applies to the following Layers:

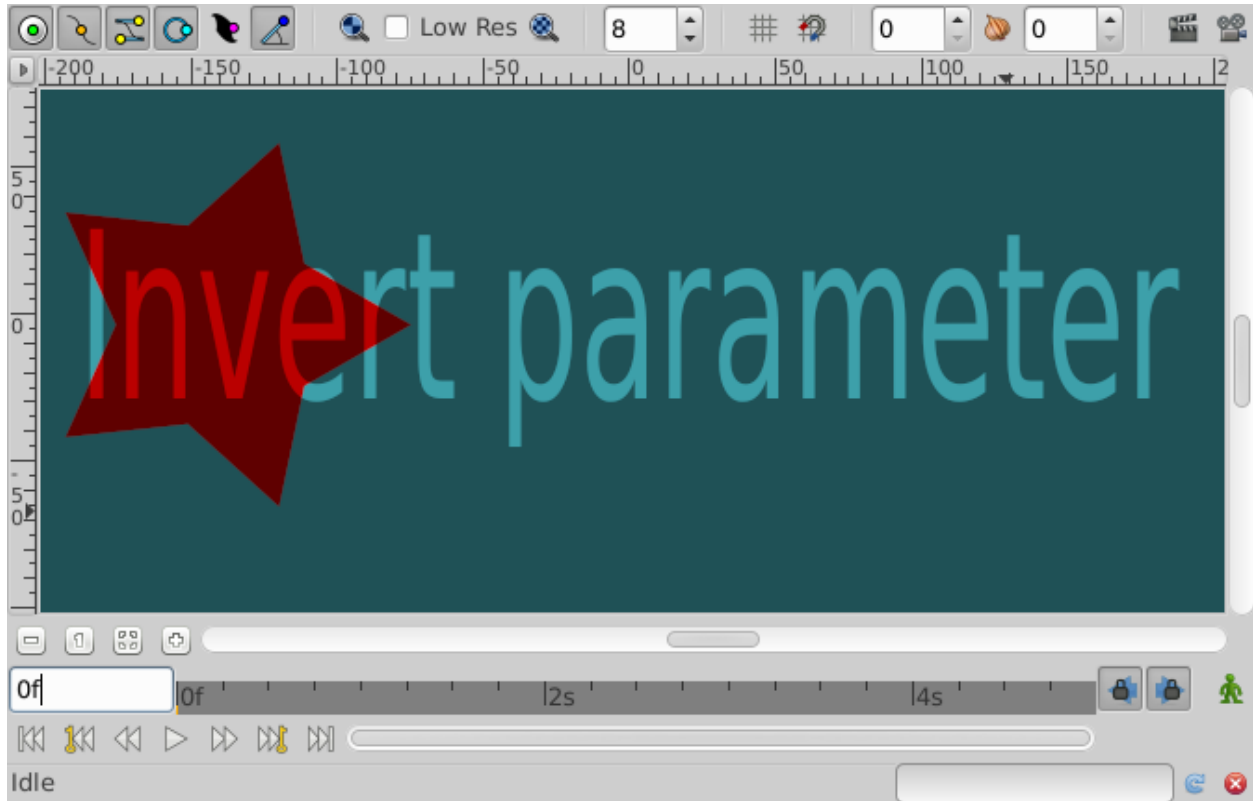
- *Region Layer*
- *Outline Layer*
- *Cicle Layer*
- *Polygon Layer*
- *Rectangle Layer*
- *Star Layer*
- *Text Layer*

This parameter inverts the filling of the shape to the outside part of its contour when checked on. Its default value is checked off (shape looks as usual).

A sample of Invert Parameter set to off:



A sample of Invert Parameter set to on:



8.6 Lock Selection

The **Lock Selection** parameter is a boolean value (default value **Off**), present in all *Group Layer*. These are the layers that are created when you *Group* a set of layers.

If the **Lock Selection** parameter is turned 'On, then:

- If you click on a grouped layer in the *WorkArea* window, the *Group Layer* will be selected, rather than the layer you clicked on. For example, you draw an *Outline Layer*, and then group it. By default the **Lock Selection** is **Off**, and clicking on your outline in the *WorkArea* will select the outline layer. With **Lock Selection On**, the group layer will be selected instead.
- Right-clicking on a group layer offers a context menu entry to **Select All Child Layers**, which recursively selects all the layers under the current layer. If the **Lock Selection** is engaged for any group layer in the hierarchy, it stops the layers inside that group layer from being selected by this operation.
- Hitting **CTRL|A** to select all the *handle* in the currently selected layers will select the *Group Transformation Widget*'s handles of selection-locked group layers, whereas it won't when their **Lock Selection** are **Off**.

The effect of these three behaviors is for a group layer to act as if it was a primitive layer, hiding the details of its contents from selection or manipulation.

8.7 Opacity

This parameter controls the layer visibility (it is more like an alpha value).

Example :

- **0** means the layer is invisible,
- **1** means the layer is visible.

How to show or hide a layer, or fade the effect of a blur?

In the *Parameters Panel*, look for an option labeled *Opacity* - this controls how much of the blended result of the layer is composited with the blend of the layers beneath it.

In other words, for a typical layer, this will ‘fade it out’. For a *Blur Layer* set to “*Blend Method: Straight*”, this will fade *between* the blurred version and the unblurred version of the canvas. If you want it to become less blurry, adjust the *Blur Layer: Size*.

If you want to affect multiple layers at the same time, you have several options:

- You can *Group Layer* your layers and use the group’s *Opacity Parameter*,
- *Export* one of the layers *Opacity Parameter* and *Linking* others layer’s *Opacity Parameter*,
- *Linking* all layer’s *Opacity Parameter*. Select the layers you want to affect in the *Layers Panel*, right click to *Opacity Parameter* in the *Parameters Panel* and choose *Link*.

8.8 Origin Parameter

The *Origin Parameter* is on the *Group Layer*, *Rotate Layer*, *Translate Layer* and *Scale Layer* layers. It is used as center of layer’s related operation : to translate the entire canvas referenced by the *Group Layer*, has center for *Rotate* and *Scale* layers ...

8.9 Time Offset Parameter

The “Time Offset Parameter” is a *Parameter* of the *Group Layer*. It brings the animation of the entire contents of the referenced layer forward by the given value. Using negative values it is possible to delay the grouped layer, too.

Note that the “Time Offset Parameter” applies to the contents of the *Group Layer*, not to the layer itself. So if the parameters of the *Group Layer* itself are animated, such as its *Origin*, the “Time Offset Parameter” won’t cause them to change.

The “Time Offset” can be animated, just like any other parameter, so it can be used in various non-obvious ways:

- **Repeat**: Set the offset to 0s at 0s and to -10s at 10s, and the first 10 seconds of the encapsulated layer will play twice (at 0s through 10s, and 10 through 20s). At 20s it will continue to play as normal, but delayed by 10s.
- **Fast Forward**: Set the offset to 0s at 0s and to 10s at 5s, and the first 10 seconds of the encapsulated layer will play at double speed (at 0s through 5s). After 5s it will continue to play at normal speed.
- **Reverse**: Set the offset to 10s at 0s, and to -10s at 10s, and the first 10 seconds of the encapsulated layer will play backwards. After 10s, it will play forward, from the beginning.

8.9.1 Examples

See `Time-offset-demo.sifz` for an example of each of the above.

See [this example](#) for another use of the Time Offset Parameter, exploring the *Time Loop Layer*.

8.9.2 Control the animation speed of a Group

Convert the “Time Offset Parameter” to linear, and it will offers two sub parameters: “Rate” & “Offset”

- Stop the animation : Setting the “Rate” to -1s.
- Slow down it : Setting it to something between -1s and 0s.
- Play reversed : Set the “Rate” less than -1s.
- Play accelerated : Set the “Rate” up than 0s.

8.10 Winding Style Parameter

The Winding Style parameter is available in these layers:

- Outline Layer
- Region Layer
- Polygon Layer
- Star Layer (where it can be used to produce some interesting effects)

It determines the way in which Synfig decides whether a point is ‘inside’ or ‘outside’ which coloring the layer in.

The Winding Style parameter has two possible values:

- Non Zero
- Even/Odd

The easiest way to see the distinction is to draw a region which contains a loop inside itself:

The top two images have their region winding styles set to “Non Zero” and the bottom two have their region winding styles set to “Even/Odd”. “Even/Odd” makes the region see-through when it crosses itself.

The left two images have their outline winding styles set to “Non Zero” and the right two have their outline winding styles set to “Even/Odd”. “Even/Odd” makes the outline see-through when it crosses itself.

<http://dooglus.rincevent.net/synfig/regions.html> has notes I made while investigating how region filling works, and describes the Winding Style parameter.

8.11 Z-Depth

This parameter can be used to change the ‘depth’ (order) of a layer in the layer stack, or of a Bone in a *Skeleton Layer*.

By default, each canvas gives its layers zdepths which depend on their order in the canvas. The first layer has a depth of 0, the 2nd has a depth of 1, and so on.

Zdepth may be thought of as an indication of the distance to an observer: layers with a lower zdepth are ‘nearer’ to an observer than layers with a higher zdepth.

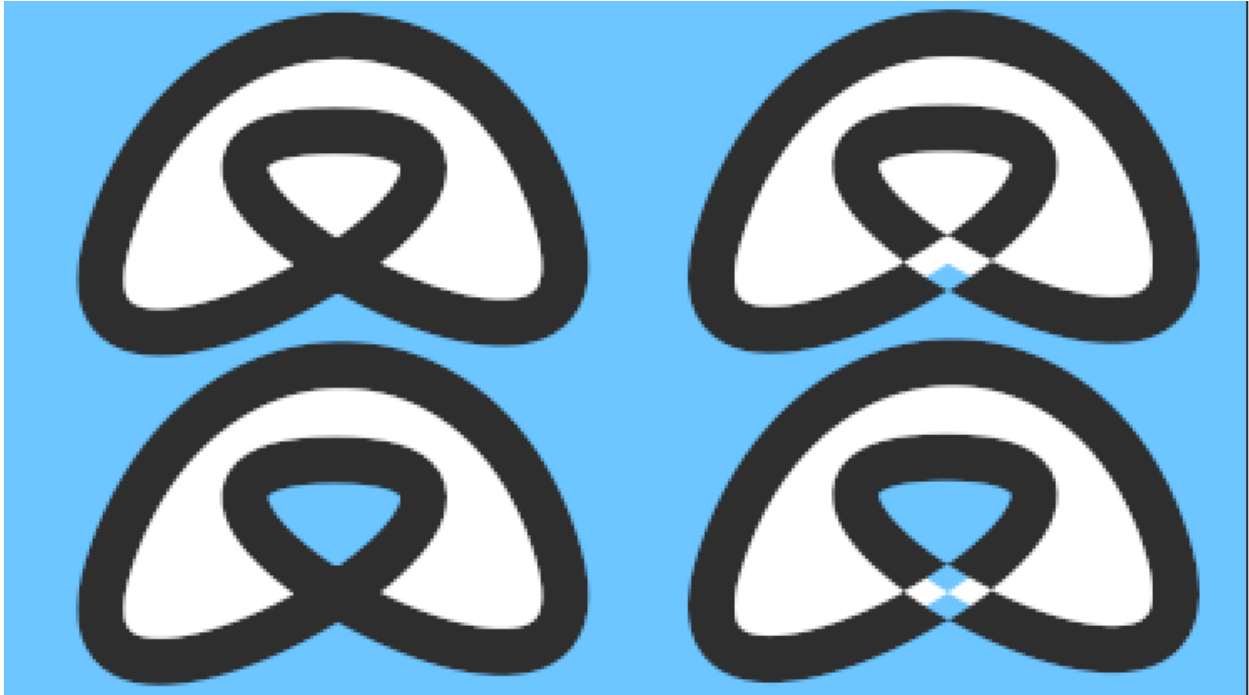


Fig. 13: winding_style.sif

The Z Depth parameter on each layer can be used to adjust this default depth. The value of the Z Depth parameter is added to the layer's 'natural' depth, given by its order in its canvas.

For example, suppose we have 2 layers, first a circle, and then a rectangle. The circle will have a 'natural' depth of 0, and the rectangle's will be 1, so the circle will be drawn on top of the rectangle.

If we use the parameters dialog to set the rectangle's zdepth to -2, however, -2 will be added to its natural depth of 1, giving a new depth of -1, and so it will be drawn above the circle.

The parameter can be animated, so that layers change order throughout the animation.

Here's an example which shows the Z Depth parameter being animated to bring one circle in front of another at a certain point in time:

source sif file

Warning: This feature is broken in some cases: See [Bug report #868](#) - a layer's Z-Depth parameter animation is not affected by a time offset manipulation of the group is included to.

8.11.1 Too Much Detail

If you want to see a more complex Z depth order animation and an explanation to how it was done please watch this animation and this PDF file.

Animation: <http://www.youtube.com/watch?v=YTpSfUthuVE>

Project : *complex Z depth order animation by genete* Zdepth-balls.sifz

Explanatory PDF file: <http://www.darthfurby.com/genete/synfig/Balls.pdf>

Same file but ODT format: <http://www.darthfurby.com/genete/synfig/Ballsv2.odt>

8.12 Zoom Parameter

The Zoom Parameter in the *Group Layer* allows you to zoom in or out on the contents of the grouped layers.

The Default Value is 0, which means no zoom at all. Positive values will allow you to zoom in, while negative values will allow you to zoom out.

The zoom is exponential - each increase of 1 in the parameter zooms in by a factor of 2.72. A zoom of 2 will zoom in by a factor of $2.72 * 2.72$ or around 7.4 times. The same factor can be applied to zooming out.

CONVERTERS

9.1 Composite

CONFIGURATION

11.1 Animate Editing Mode

11.1.1 Summary

Each *Canvas* is either in Animate Editing Mode or it isn't.

11.1.2 When Animate Editing Mode is ON

When on Animate Editing Mode, each time you edit a parameter (whether directly in the *Parameters Panel* or indirectly by moving a Handle), a Waypoint is created to remember the change, and the position on the *Timetrack Panel* at which the change happened : you are creating an animation.

Depending on the value of the *Editing Lock Keyframes* setting, waypoints may also be created in the next and previous *Keyframe* as well.

11.1.3 When Animate Editing Mode is OFF

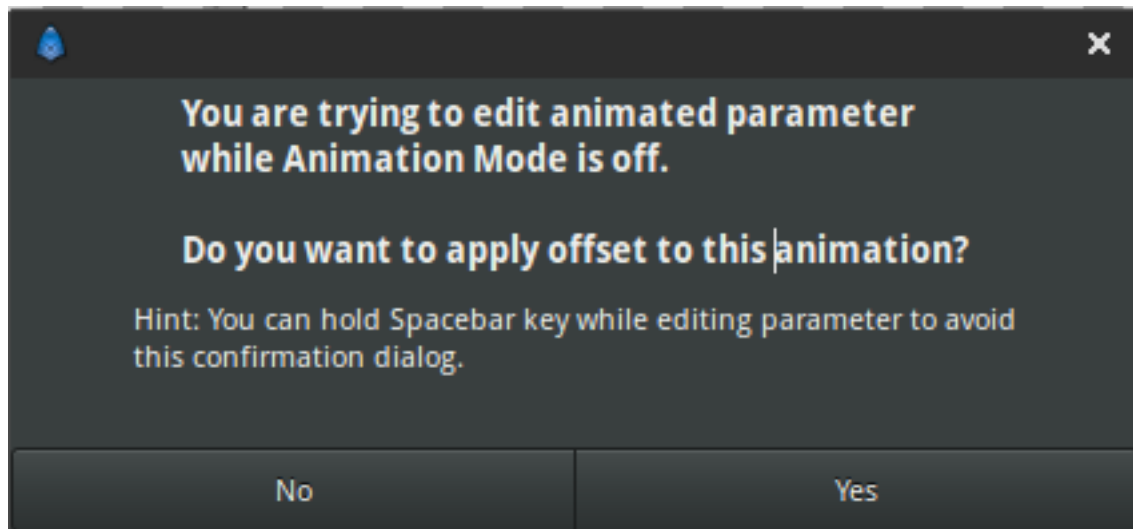
When not in Animate Editing Mode, changes to a parameter will be applied throughout the entire timeline of the animation.

If there are already waypoints on the timetrack (*ValueNodes*) for the parameter in question, this causes a warning and your edit will be not allowed. A warning message asking you : “Do you want to apply offset to this animation?” will offers the opportunity to apply an offset for this parameter for the entire timeline of the animation.

11.1.4 Toggle Animate Editing Mode

Animate Editing Mode can be toggled on and off by clicking the round button to the right of the timetrack at the bottom of the canvas window. It will only be visible if your canvas has a non-zero end time, and will only be active if you're not currently using a tool which disables the timetrack, such as the *Spline Tool* or the *Draw Tool*.

Name	Icon	Behavior
Animate Editing Mode Off		Changes will be applied throughout the entire timeline of the animation.
Animate Editing Mode On		Changes will create a Waypoint to remember the value.

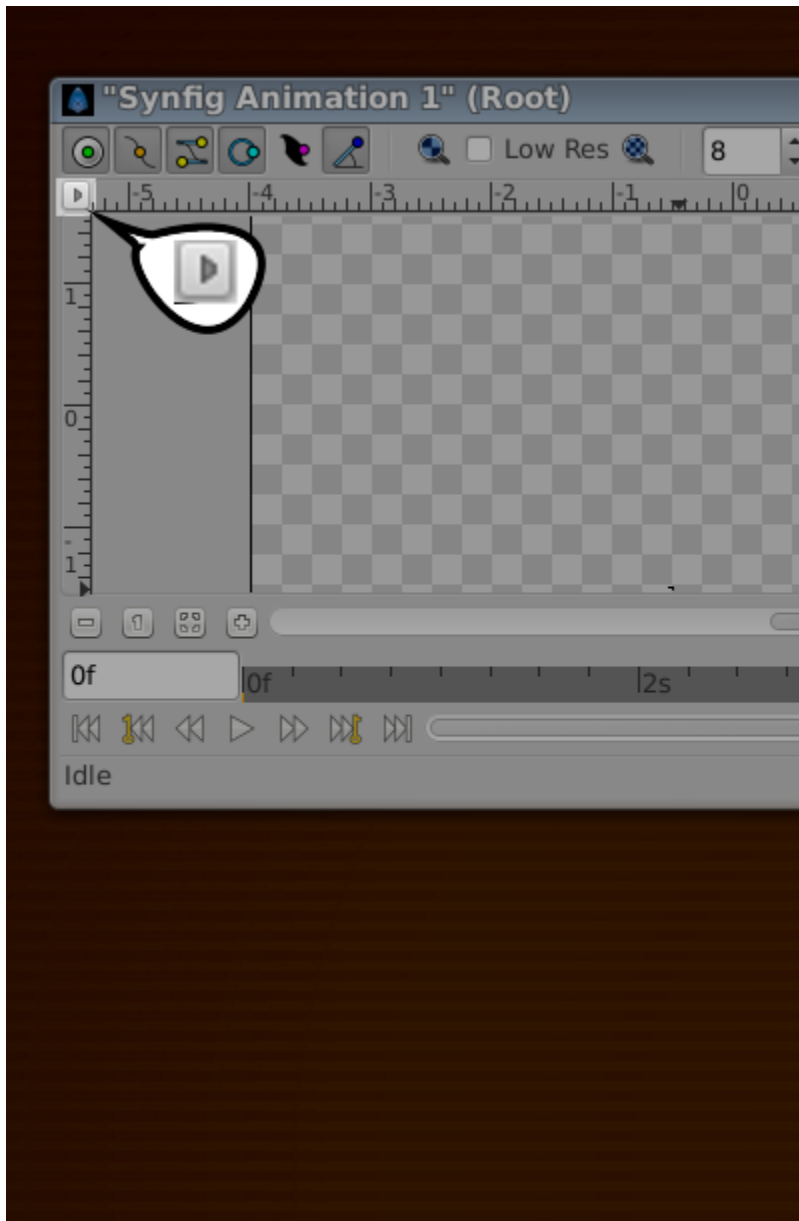


When toggling the Animate Editing Mode ON, the *Canvas Window* will be highlighted with red border line.

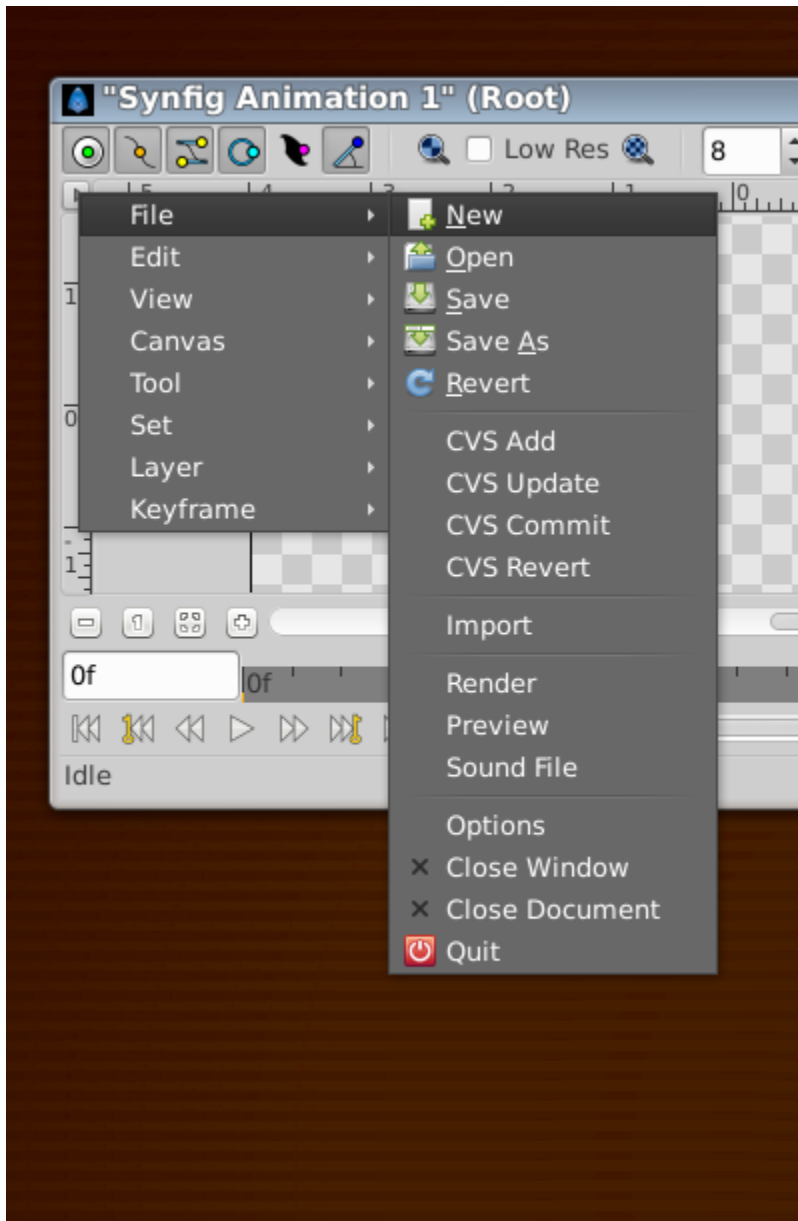
Fig. 1: Animate Editing Mode Toggle (On/Off)

11.2 Canvas Menu Caret

Whereas most graphics apps have a set of menus at the top of the screen, the top of the *MDI* window, or the top of the drawing window, Synfig has a Caret. A sideways one. It is located in the upper left hand corner of the *Canvas Window*, and looks like this:



Beneath this button are all the *Main Menu* you would expect, from which you can access most of Synfig Studio's features:



In v.1.0.2 it may appear as a blank button in the top left.

11.3 Editing Lock Keyframe

11.4 Grid

11.5 Guides

11.6 Layer Menu

11.7 Main Menu

11.8 New

11.9 Open

11.10 Preview

11.11 Canvas Properties Dialog

11.12 Render

11.13 Save

11.14 Save All

11.15 Save As

11.16 Timebar

11.17 Workarea

11.18 Zoom In

11.19 Zoom Normal

11.20 Zoom out

INTERFACE

12.1 Timebar

The Timebar is the area above the *Timetrack Panel*, the *Graphs Panel* and optionally at the bottom of the Canvas Window.

In the Timebar you could find informations about :

- **Project duration** : as a frame ruler.
- **Current Time** : The horizontal orange/blue line. You can click and drag within it to scrub your timeline.
- **Keyframe** : Their time, activation state and the selected one (not in the *Graphs Panel*'s Timebar).



To change the scale of the Timeline put your mousepointer over the grey frame ruler and use your mouse wheel to zoom in or out.

The Timebar at the bottom of the Canvas Window is optionally displayed by toggling the  button at bottom left of the Canvas window.

Note: Good to know : Timeline visibility

The timeline at the bottom of the Canvas Window will not appear when you have defined zero duration in the *Canvas: Properties Dialog*.

12.1.1 Keyframes in the Timebar

You can adjust the Time *Keyframe: Time* of a keyframe by a normal drag and drop. To adjust it's Length *Keyframe: Length*, hold **Alt** key when releasing the mouse button.

During a keyframe drag and drop operation a tooltip indicator will be displayed with some time informations.

Right click over a keyframe or in upper part of the Timebar open the Keyframe Menu giving access to the keyframes actions.

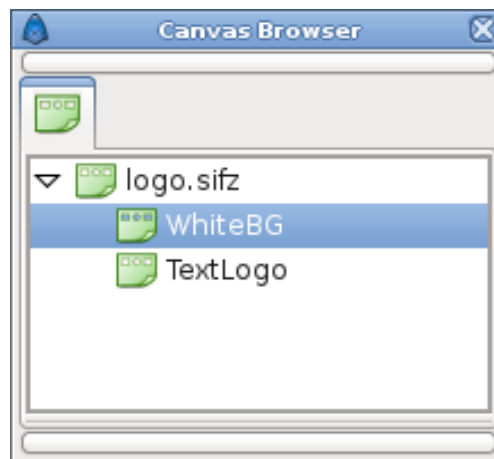


Fig. 1: Three keyframes with three different states : **Normal**, **Selected** and **Deactivated**

12.2 Canvas Browser Panel



The Canvas Browser dialog shows the tree of canvases that are present in the currently selected document:



Export a canvas parameter of a *Group* Layer will cause the canvas to be added to the Canvas Browser dialog. Double-clicking on a sub-canvas in the dialog will open a new Work Area window allowing you to work on the sub-canvas separately.

12.3 Library Panel

The Library Panel shows all exported *ValueNode* and all canvases in the current animation.

If you find yourself using a particular Color over and over again, it's a good idea to export it into the Library Panel so you can easily pick the same Color in the future. Simply right-click on the Color in the *Parameters Panel* and choose “*Export*” from the menu. You will be prompted for a name to associate with the value. In order to see the newly exported value in the Library Panel, you will have to open up the ValueBase Nodes tree if it isn't already opened. The same technique works for any other parameter as well, even for complete animated Spline paths.

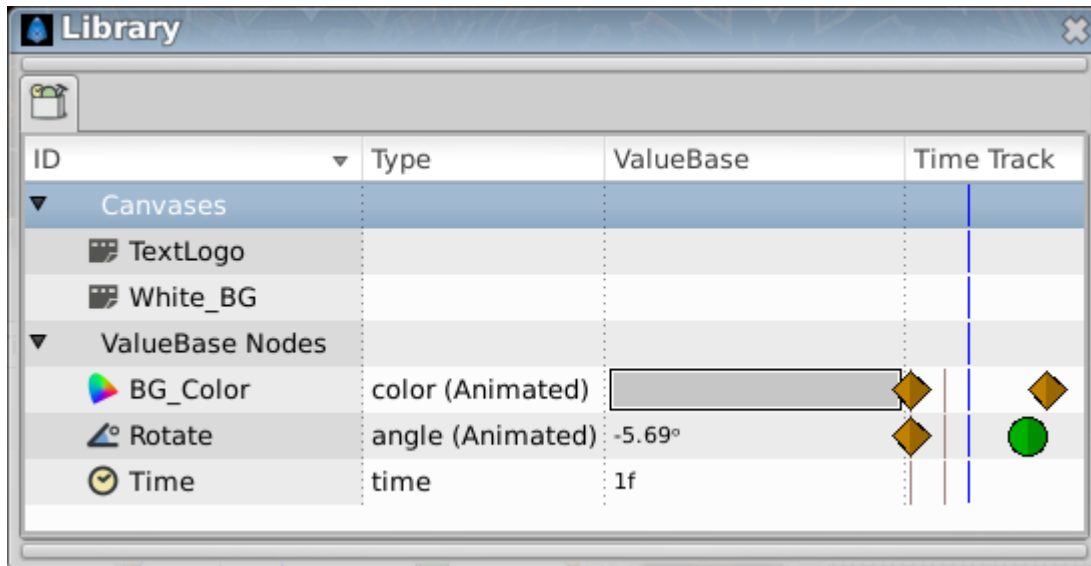
To use a value that you have exported into the Library Panel in a different layer, simply select the value in the Library Panel then right-click the parameter you want to apply it to in the *Parameters Panel*. Select “*Connect*” to use the value.



The values stored in the **Library Panel** store their animations with them. So if you have a Circle whose Color changes from red to yellow over time, and export that Circle's Color parameter and use it to Color a Rectangle, the Rectangle will also change from red to yellow over time, in exactly the same way. The Color of the two shapes will now be linked, so editing either one of them will affect the other.

When a value is selected in the **Library Panel**, its handle will be shown in the *Work Area Window*. These handle will remain as you switch from layer to layer, which can be confusing.

It is possible, using drag and drop, to organise the values in the **Library Panel** into a hierarchical structure, but currently the structure you create isn't saved into the .sifz file, so any organisation you do will be lost when you next open the document.

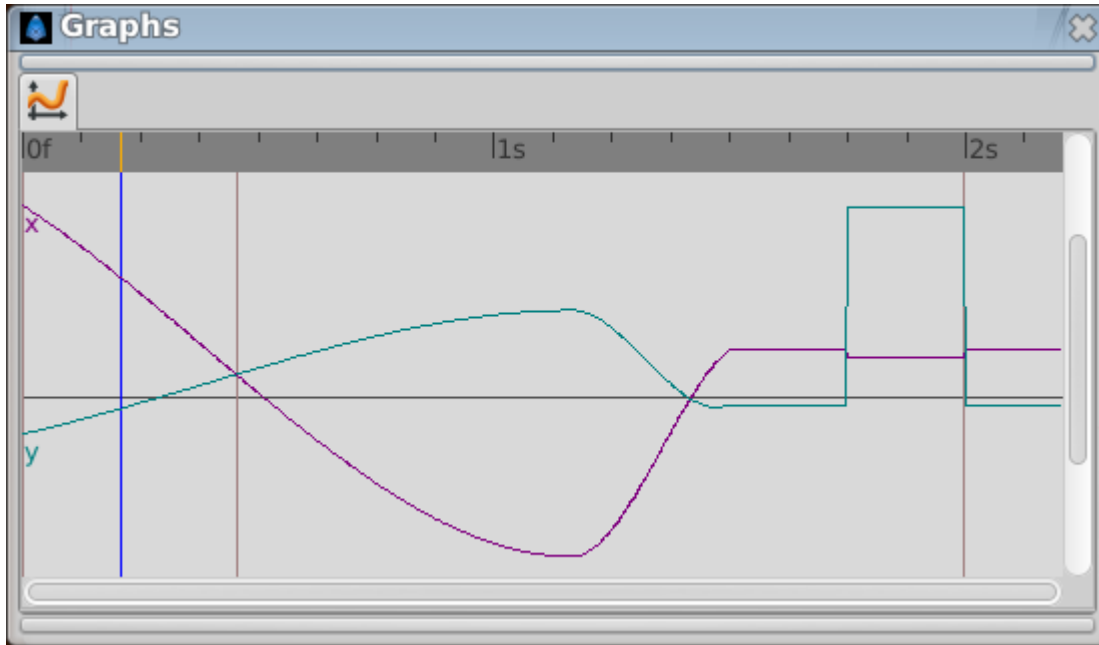


12.4 Graphs Panel



Displays the parameter currently selected in the *Parameters Panel* as a function of time.

ctrl + mouse wheel : Change zoom factor



12.5 History Panel



The history panel keeps track of all the actions that are made in Synfig Studio while editing a file. This history list is empty for a new clean file and also when the file already exists and is opened, so it is not saved with the file (is it worth to save it in the sifz file?).

When a single user operation produces a lot of actions (typically when manipulating lots of *Handle* at the same time) all those common actions are grouped into a expandable one. See the small triangle in the image sample. By clicking on the small triangle you can expand the grouped actions and access them individually.

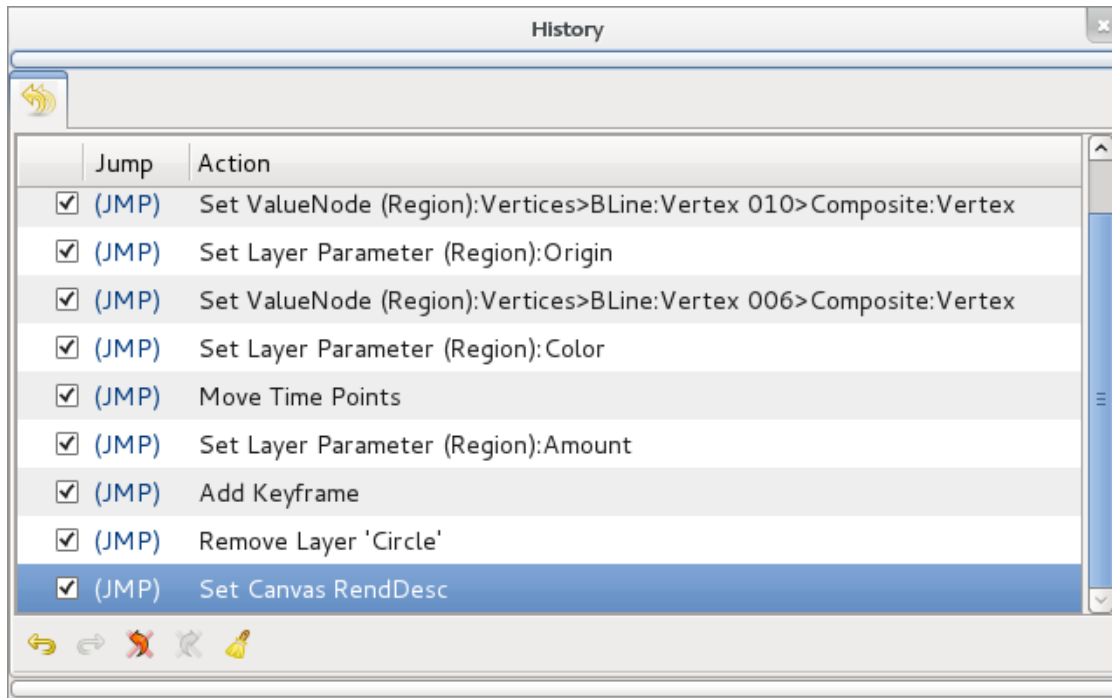
The check box column is used to disable specific actions without having to go back through the entire actionlist. So if you want to redo or undo a specific action (or group of actions) just click on the check box.

The (JMP) column is used to redo or undo all actions from the current situation to the action where you clicked on the (JMP) word.

The Action column gives a description of the action itself. It attempts to give a concise but meaningful description of the action.

The Undo and Redo buttons undo or redo a single action each time they are pressed.

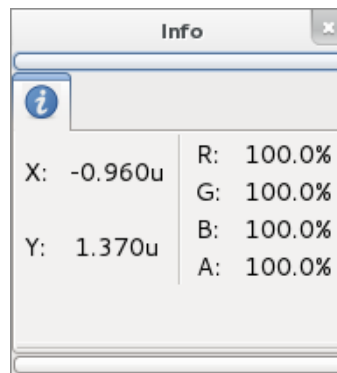
Finally the Clear Undo and Clear Redo buttons clean the history list of Undo and Redo actions. The Clear Undo and Clear Redo actions cannot be undone.



12.6 Info Panel



The Info panel

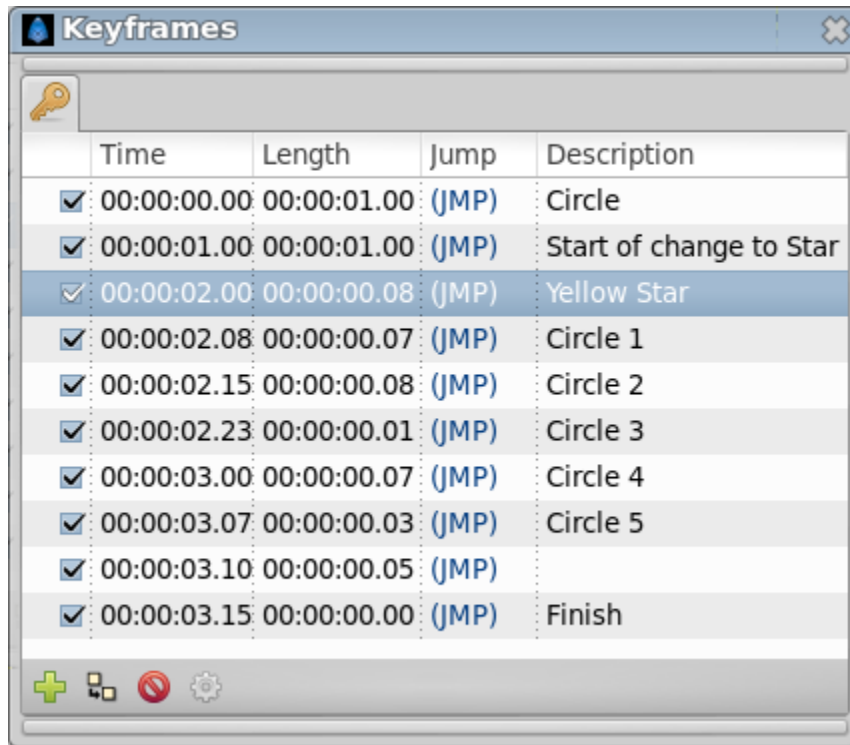


The info panel shows information, such as the current cursor/selection location on the canvas. Also it displays the current RGBA values of the cursor position on the canvas.

12.7 Keyframes Panel



The Keyframes panel is where you make new *Keyframe* for the animation.



These will show up in the *Timetrack* dialog. To jump to a specific keyframe, double click on the (JMP) column.

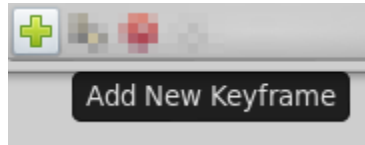
The Add keyframe *Keyframe: Add a keyframe* button adds a new keyframe to the current frame - a frame where there is no keyframe set already. Remove keyframe *Keyframe: Remove a keyframe* removes a keyframe. The Duplicate keyframe *Keyframe: Duplicate a keyframe* button copies the keyframe to the current frame. The Keyframe Properties *Keyframe: Editing Keyframe Properties* button display the Keyframe Dialog *Keyframe: Editing Keyframe Properties* of the selected keyframe.

Don't forget to set the *Description* of your keyframes, this description will be displayed in the upper left corner of the canvas window.

Right click in keyframe list open the keyframe contextual menu for the selected keyframe.

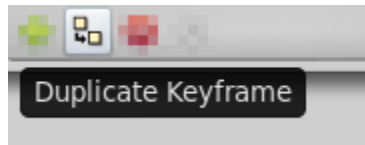
12.7.1 Adding, duplicating and removing keyframes

Add a keyframe



Place the time cursor at a frame where there isn't currently any keyframe. Then press the **Add new Keyframe** button. If you place the time cursor at a frame where there is currently an existing keyframe or if animation Start Time equals animation End Time (animation Duration is 0m 0s 1f) then the **Add new Keyframe** button is disabled. Once you press the button then a new entry is added to the list of keyframes and a vertical dashed line is added in the time line. No waypoint is created.

Duplicate a keyframe



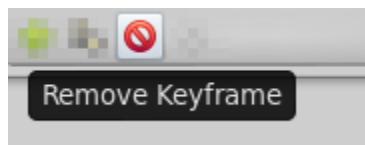
Select a keyframe in the keyframe list of the *Keyframes Panel* and place the cursor at a frame where there isn't currently any keyframe. Then press the **Duplicate Keyframe** button. This would have two separated effects:

1. If there is a waypoint at the original keyframe then the waypoint is duplicated. Its duplication includes the parameter value and its interpolation types.
2. If there is no waypoint in the original keyframe for any particular parameter then two things could happen:
 - There is no waypoint for that parameter at ANY frame in the time line: Then NO waypoint is created.
 - If there is a waypoint in the time line for that parameter, but not in the keyframe that is going to be duplicated, then in the duplicated keyframe is created a new waypoint with a value for the parameter of the result of the current value at the original keyframe and a TCB Smooth interpolation type for both In and Out.

Of course, duplicate a keyframe will produce a new keyframe at the place pointed by the time cursor and will add a new one to the keyframe list in the proper place. In the keyframe list, the new added keyframe have the same description than the original, plus a (Duplicate) at the end.

[video explanation](#)

Remove a keyframe



Just select a keyframe from the keyframe list and press the **Remove keyframe** button. It will remove the keyframe and all the waypoints for all parameters for all layers that are currently there.

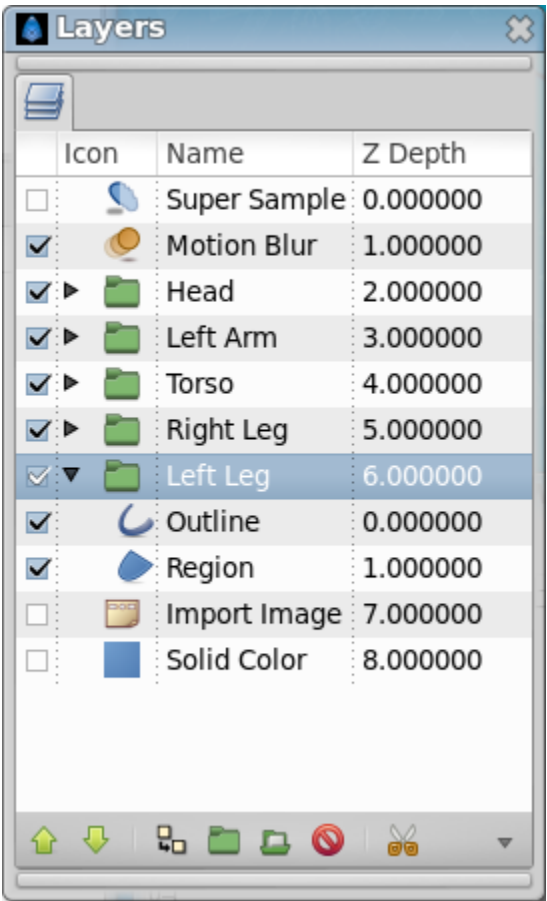
Note: If you move a keyframe by modifying its time ([#Time]) in the keyframe list dialog and immediately press the Remove Keyframe button then the waypoints are not deleted. It seems to be a bug but also can be considered a feature if you really want to keep the waypoints and not the keyframe.*

12.8 Layers Panel



12.8.1 Summary

The Layers Dialog allows you to view and manipulate the layer hierarchy in your scene. This is where you can determine which objects get drawn first, and create *Group Layer* for the purpose of applying effects.



12.8.2 Elements of the Layers Dialog

The Layers Dialog contains a detailed list of the *Layer* in your scene. Each row represents a layer, and the columns are:

Show/Hide checkbox

Turns the layer on or off in the composition. This parameter is not keyable.

Icon

The icon representing the layer type. For *Region Layer*, *Outline Layer*, and , the icons will be those of the *Spline Tool*, the *Width Tool*, and the *Gradient Tool*, respectively. For both *Group Layer* and *Switch Group Layer*, an expand icon (generally an arrow) precede the layer icon. This expander can be controlled with + and - respectively to open or close the group.

Name

The name of the layer you're currently working on.

Z Depth

The Z Depth column indicates the Z position of the layer, which can also affect layer compositing. It does not always appear, for example, if the time of the canvas is "0", this column will disappear.

Layer Commands

Below the layer list, are buttons (and a drop down menu containing any buttons that don't fit on the palette). These expose the most useful layers' commands, more commands can be found in the *Canvas: Layer Menu* from the *Canvas: Menu Caret*, or in the context menu of the Layer Panel.

Moving layers

To move multiple layers first select them by holding `ctrl` or `shift`. Then at the last of the layers you want to move click it but do not release the mouse button, instead release the keyboard controller (`ctrl` or `shift`). Now you can drag the layers to where you want them, release mouse button to drop the layers in place.

Gradients

12.9 Navigator Panel



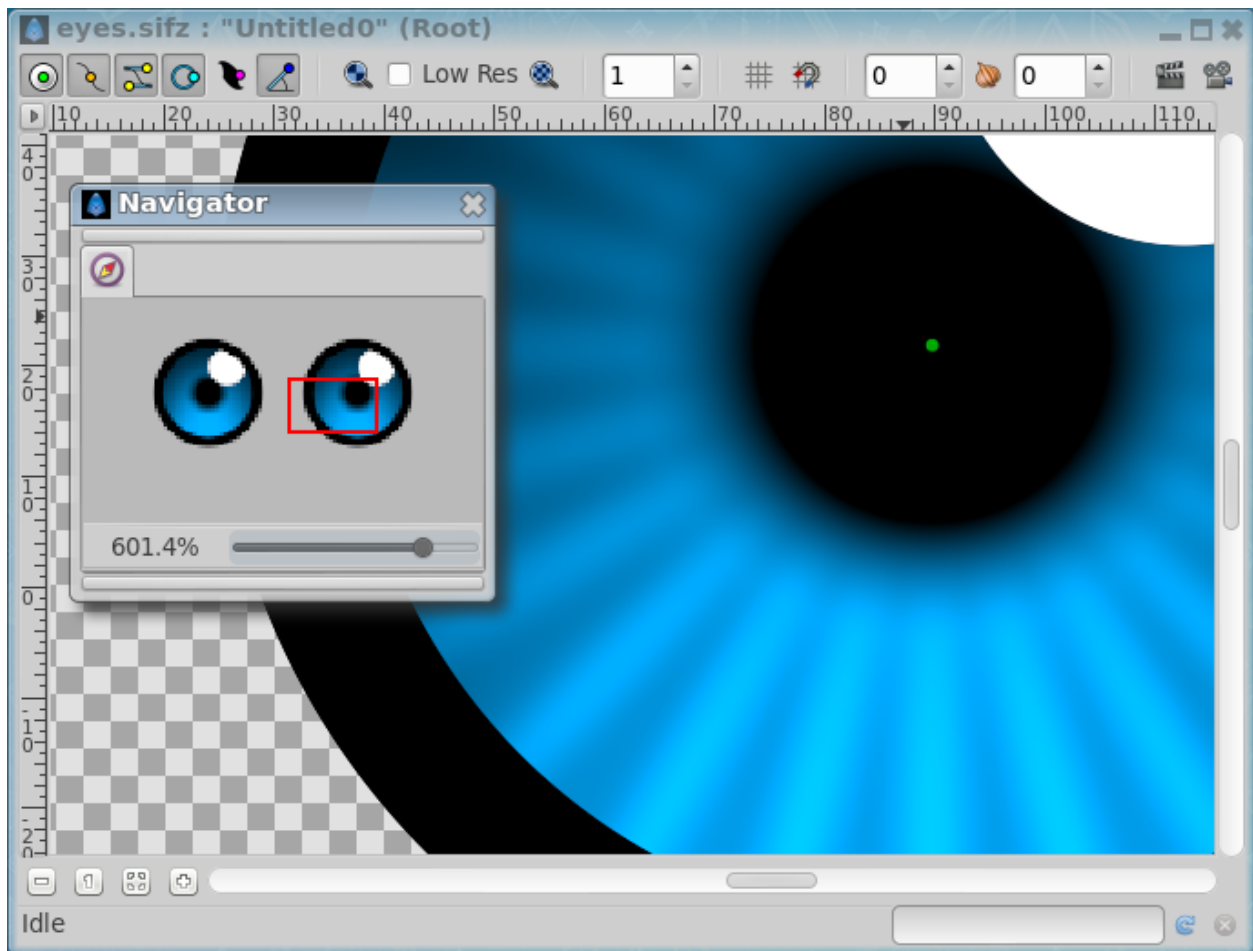
The Navigator dialog allows the user to position the zoomed working view quickly to the desired area.

The navigator dialog shows you a thumbnail view of the currently selected canvas with a red rectangle indicating which part of the canvas is currently displayed in the editing window.

You can zoom in and out by dragging the zoom slider at the bottom. Also the zoomed area can be panned by dragging the red rectangle around.

Alternatively you can zoom and pan the *Workarea* using the following mouse shortcuts in the *Canvas* window:

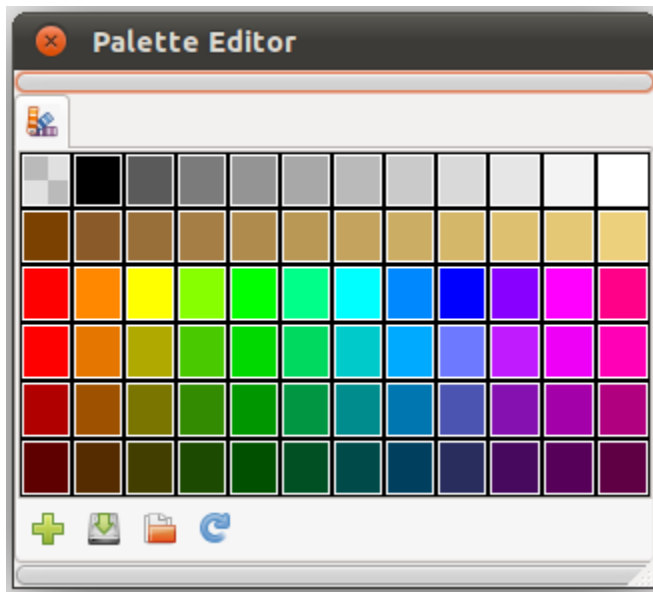
1. Mouse wheel pans up and down.
2. SHIFT + mouse wheel pans left and right.
3. CTRL + mouse wheel zooms in and out.



Zoom in limit is 1600% and out limit is 6.2%.

It is possible to zoom in with CTRL + mouse wheel beyond 1600% but can cause Synfig Studio to hang or crash in some situations.

12.10 Palette Editor Panel



Here you can add, edit or delete your preferred colors from your document. A right click on a color gives you 2 options: color (to edit the color) and delete (to delete it).



The buttons at the bottom of the panel allow you to:

- **Add** the current outline color to the palette
- **Save** the current palette in Synfig Palette format (.spal)
- **Load** a palette in Synfig Palette (.spal) or GIMP Palette (.gpl) formats
- **Reset** the current palette colors, discarding any change you've done.

12.11 Parameters Panel



12.11.1 Introduction

The Parameters Panel is in many ways the heart of the Synfig interface. This is where all the *Parameter* of the *Layer* you create are edited, and in some less obvious ways.

The layout of the Parameters Panel is quite simple - it is a table.

Name	Value	Type	Time Track
Z Depth	0.000000	real	
Amount	1.000000	real	
Blend Method	Composite	integer	
Color		color	
Origin	-86.074205px,19.417741px	vector (Animated)	
Invert	☐	bool	
Antialiasing	☒	bool	
Feather	0.000000px	real	
Type of Feather	Fast Gaussian Blur	integer	
Winding Style	Non Zero	integer	
Outer Radius	65.920000px	real	
Inner Radius	35.931692px	real	
Angle	47.40°	angle (Animated)	
Points	5	integer	
Regular Polygon	☐	bool	

Fig. 2: Stars Layer Parameters

The first column, named Name, is simply an expandable tree listing of the parameters of the selected layer. Most layers do not have many nested parameters, with the exception of the Vertex List on most *Geometry Layer*.

The second column, Value, is where the data for each of the parameters are listed. Several different types of data can be shown here.

The third column, Type, show the data *type*.

The last column, Time Track, display a simplified *Timetrack Panel*

12.11.2 Parameters and Layers

An interesting feature of the Parameters Dialog is that - when you select two or more layers in the *Layers Panel*, the Parameters Panel will only show the parameters that are **shared** between the layers. When the dialog is in this state, context-clicking on the parameters will allow parameters to be *Linking* between the two layers. (*write more*)

12.11.3 Default Parameters

There are three parameters that are shared between nearly every layer:

- *Z Depth Parameter*
- *Opacity*
- *Blend Method*

12.11.4 Interpolation

It is possible to fix interpolation by *Parameter*. Right-click any parameter and choose its default interpolation from a submenu. The defined interpolation is indicated in the Parameters Panel with corresponding icon near the parameter value.

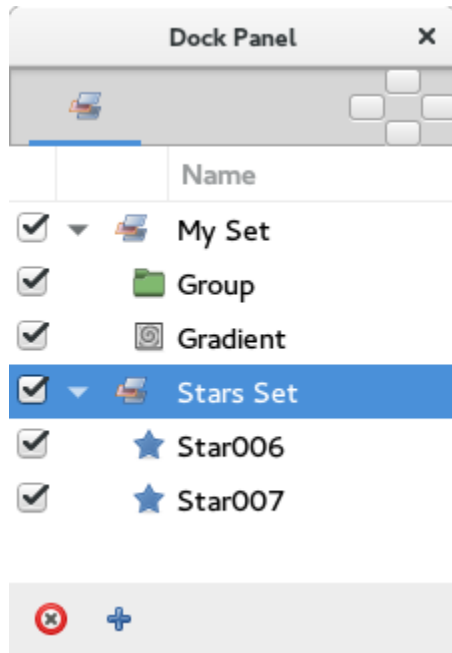
12.12 Sets Panel



images outdated!

12.12.1 Introduction

Suppose you want to treat several shapes in the same way, but they are scattered in your project and placed on different layers. Rather than develop each and select them individually each time you want to change them, you can use sets.



To add some layers to a set select them in the *Layers Panel*, right click, and choose **Add to Set**. It will prompt you for a Set name. Type a name for a new, or existing set and click OK.

Now you can select that set and animate all those layers together (for instance, change all their *Opacity*, or move their *Origin*).

12.12.2 Digging deeper

Action over Sets

Current interaction with sets and its layers in Sets Panel can be described in following terms:

- Double click over a Set icon selects all its layers in the Layers Panel. **It allows a multiple selection of scattered layers with just one double click.** Notice that double click does not highlight the layers in the Sets Panel.
- Double click over a icon of a layer from a set, selects that layer on the Layers Panel and unselects all others.
- Double click holding `ctrl` key over a Set icon or layer icon from a set, adds the selection to the current one at the Layers Panel.
- Double click over a set or layer name allows to rename it.
- Clicking the checkbox on the left of layer icon will deactivate the layer in the Layers Panel.
- Clicking the checkbox near the set icon will deactivate all layers inside of this set.
- A layer in a set can be moved to any other set (empty or not).
- If the set is empty it can't be activated and won't be saved.
- Single left or right click over a Set makes it selected. You can use it to enable the **Remove Set** button.
- Single left click over layer in a set selects it. You can select multiple layers if you click holding `ctrl` or `shift` key, but the effect on subsequent commands only affect the layer that you right-click on.

Action over Layers

Single right click over a layer in a set pops up the context menu with the same elements as you have for each layer in the *Layers Panel*:

- Command:Select All Children : This menu entry appears only if the layer from the set is a Group layer. The behavior is the same as in the Layers Panel - all layers inside of the Group layer are selected.
- Command:Add Layers to Set : Displays a dialog to enter a Set name and moves the select layer to the new set. Only works for the layer below the cursor when multiple selection is done.
- Command:Remove Layers from a Set : Removes the selected layer from the set. Only one layer at a time can be removed. Even if multiple layers are selected only the layer below the cursor when right clicking is removed.
- Command:Duplicate layer : Works the same as duplicate layer from the Layers Panel and also add the duplicated layer to the current Set. It works partially because you cannot duplicate multiple layers as in the Layers Panel. It only acts over the layer you did the right click. The selection is ignored.
- Command:Group: It works the same as the Command:Group layers from the Layers Panel but only affects to the layer where the right click is done.
- Command:Remove Layer : It works the same as the Command:Remove Layers from the Layers Panel but only affects to the layer where the right click is done.
- Command:Set Layer Description : Displays a dialog for renaming the layer.
- Command:Raise Layer / Command:Lower Layer : Does the same than in the Layers Panel. It doesn't modify the position of the layer in the Set but the position in the Layers Panel hierarchy is modified as expected.

Notice, that a layer can only belong to one Set at a time. Adding a layer to a Set when it's already in previous set will remove it from the previous set.

Sets can be nested

You can make nested sets one of the following ways:

- by adding a layer to a set with a name containing dot as separator: "parent.child"
- by dragging a set inside other set.

The contents of the nested sets is selected when you double-click the parent set icon. Depending on what set level you click you can select more or less layers.

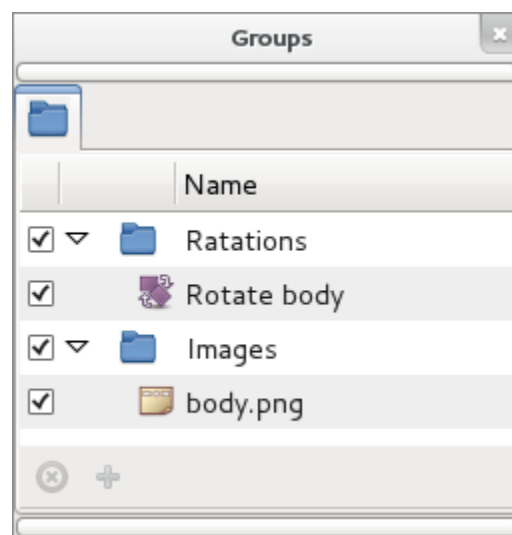
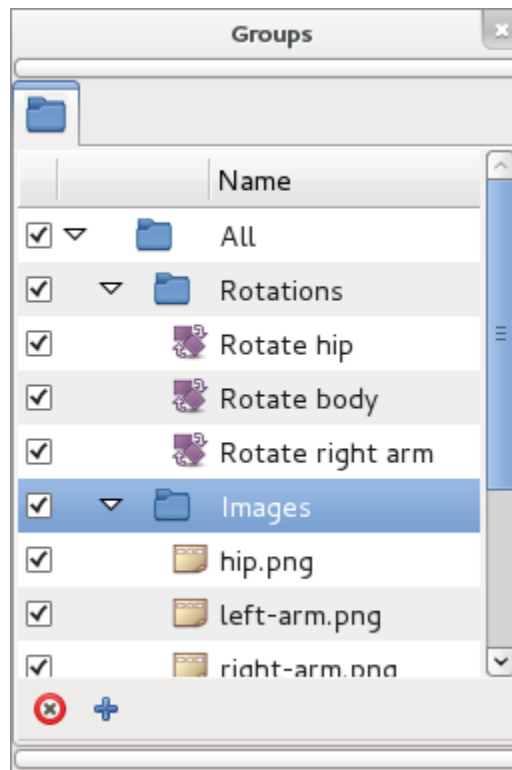
For example making a double click on the 'All' Set will select Rotation and Image layers in once double click.

Regarding to the buttons:

- Add New Set button adds empty deactivated set.
- Remove Set button removes selected set. If the set contains other sets they are not recursively deleted, only the first child level of layers.

Also Sets scope is the current edited Canvas. That means that only the layers in a set that belong to the current Canvas being edited are show in the Sets Panel. It also means that Sets are File/Canvas dependent. It means that a layer can only belong to a set defined in its exported canvas scope. For example, if there are two canvases where B is the root one and A is a exported one hold by a Group Layer inside B, then a layer that belongs to exported canvas A can belong only to Sets defined at canvas A scope. Although you can select the layer from the outer root canvas (B) you cannot add it to a set defined at B scope. You can also have sets with the same name in different file/canvas scope. If a layer is dragged out from a pasted canvas and it belongs to a set, the set name is dragged out also.

Here are shown some examples of what can be done using sets for a complex file:



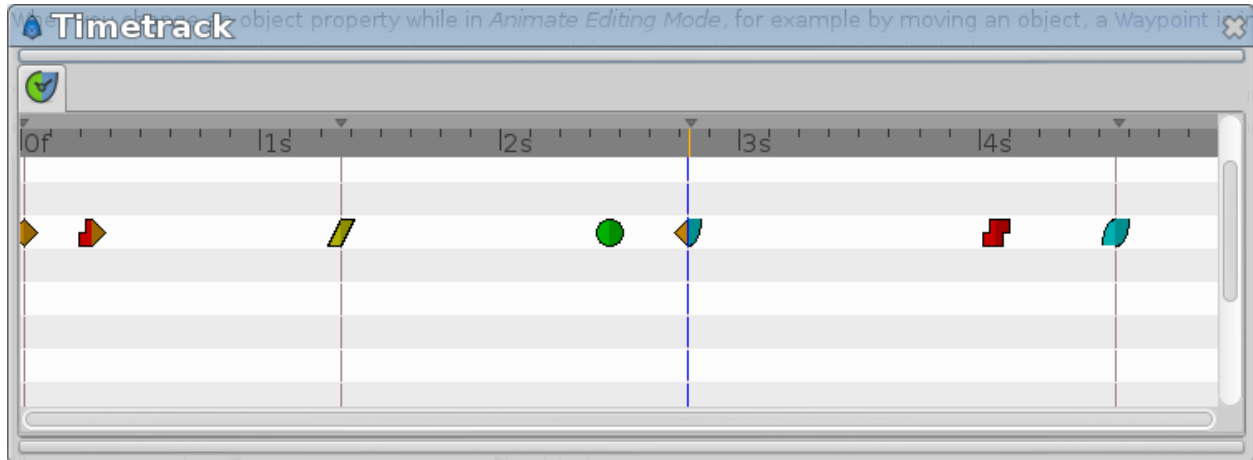
The figure consists of two side-by-side screenshots of a 3D software interface, specifically the 'Sets' panel. The left screenshot shows the 'Groups' panel with a list of objects and their Z Depth values. The right screenshot shows the 'Layers' panel with a list of objects and their Z Depth values. Both panels have a search bar at the top and a list of objects with checkboxes and Z Depth values. The 'Groups' panel has a 'Name' column and a 'Z Depth' column. The 'Layers' panel has a 'Name' column and a 'Z Depth' column. The 'Groups' panel shows a hierarchy of objects, with 'Rotations' selected. The 'Layers' panel shows a list of objects, with 'right arm' selected. Both panels show a list of objects and their Z Depth values.

Panel	Object Name	Z Depth
Groups (Left)	All	
	Rotations	
	Rotate body	
	Rotate right leg	
	Rotate left leg	
	Rotate body	
	Rotate right arm	
	Images	
	right-arm.png	
	left-leg.png	
	body.png	
	right-leg.png	
Layers (Right)	Inline Canvas	0.000000
	right arm	0.000000
	Rotate body	0.000000
	Rotate right arm	1.000000
	right-arm.png	2.000000
	right-leg	1.000000
	Rotate right leg	0.000000
	right-leg.png	1.000000
	left-leg	2.000000
	Rotate left leg	0.000000
	left-leg.png	1.000000
	body	3.000000
	Rotate body	0.000000
	body.png	1.000000

12.13 Timetrack Panel



The Timetrack Panel shows a symbol ([Waypoints#Graphical_Representation]) (usually a green dot) for every *Waypoint* in the timeline. *Keyframe* are represented by vertical lines through the timeline. The *Time Cursor* shows you with another vertical line at which frame you currently are at.



When you change an object parameter while in *Animate Editing Mode*, for example by moving an object, a *Waypoint* is inserted. The rows of the Timetrack Panel are aligned with the *Parameters Panel*. If you are animating the individual vertices, you have to expand the vertices list to see their individual waypoints.

You can drag those *Waypoint* through time, and you can change their interpolation method ([Waypoints#Interpolation]) by right-clicking (eg. a sudden step change instead of a smooth transition). You can also change the default interpolation method from the Toolbox Panel in the *New Layer Defaults* section. The very bottom control sets the default method, and is initially set to *Clamped*.

Holding CTRL allows you to select more than one waypoint for a given valuenode at a time.

- When nothing is selected, clicking on a point in either normal mode or additive mode will select the time point closest to the click. Subtractive click will do nothing
- When things are already selected, clicking on a selected point does nothing (in both normal and add mode). Add mode clicking on an unselected point adds it to the set. Normal clicking on an unselected point will select only that one time point. Subtractive clicking on any point will remove it from the the set if it is included.

Normal click out of a waypoint unselect all.

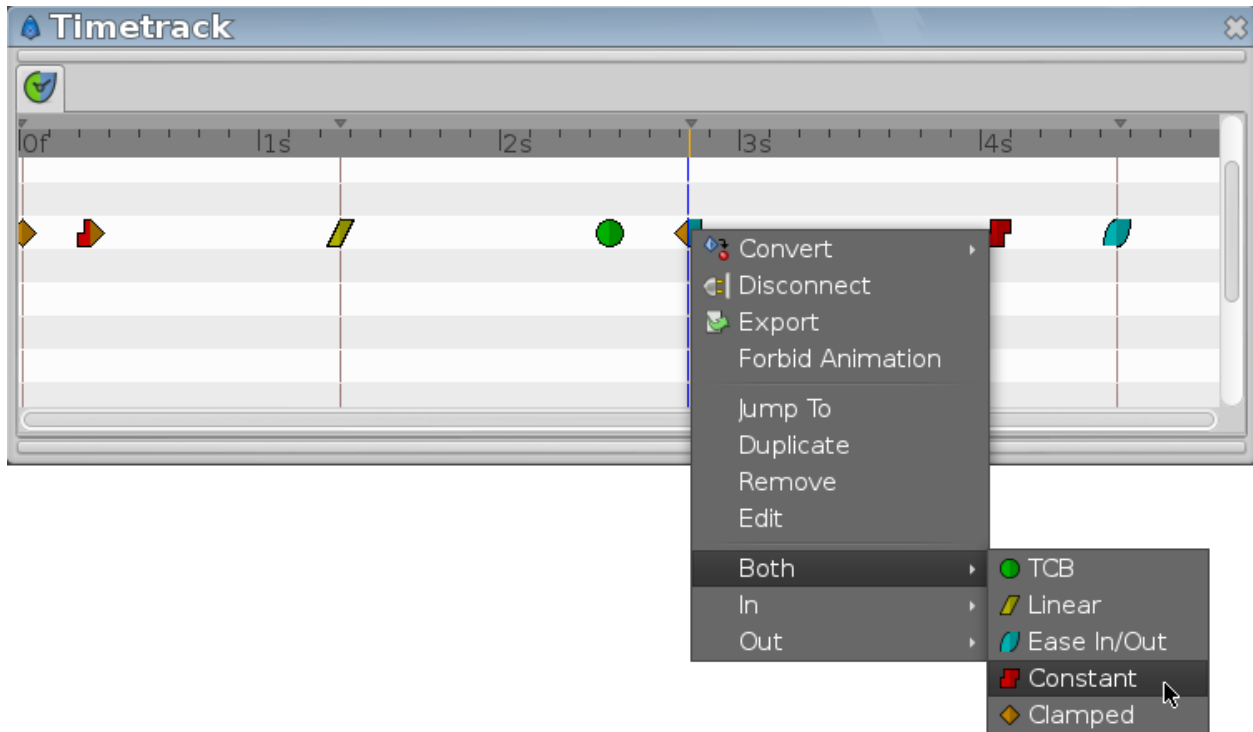
Holding SHIFT when beginning to drag waypoints will copy them rather than moving them.

Holding ALT while clicking waypoints will delete them.

Further reading:

- *Keyframe*
- *Waypoint*
- *Timebar*

Interested in future things ? [Waypoints ... what could be the future of...?](#) in the synfig forum.



12.14 Tool Options Panel

12.14.1 Introduction

The Tool Options Panel displays the options you have available to apply to the *Toolbox* you're using. For the tools that can create multiple layers at the same time the panel contextually display the available options for the selected type of layers.

The icon for the Tool Options Panel always changes to what tool you're currently using.

The values of the Tool Options Panel parameters are stored between tool usages and between SynfigStudio sessions, so you can reuse them as many times as you like.

This is the list of available options for each tool:



Transform Tool

- None



Smooth Move Tool

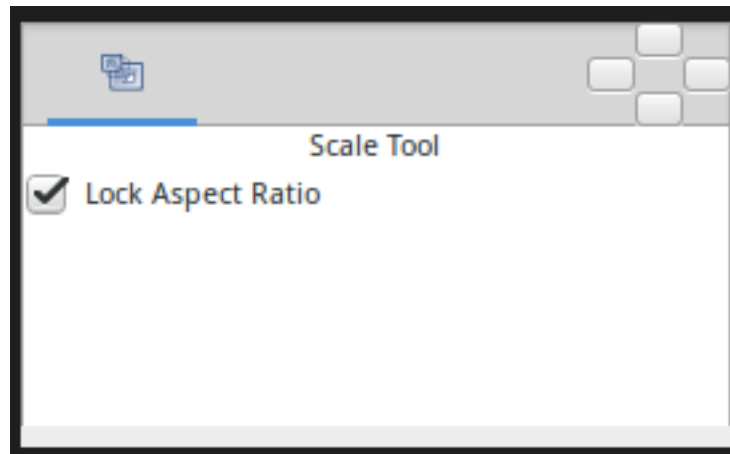
- Radius (real number)



Scale Tool

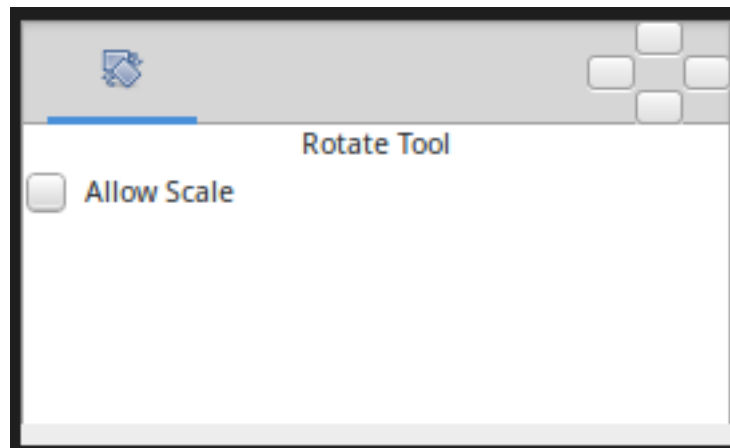
The *Tool Options Panel* allows you to specify:

- **Lock Aspect Ratio** : Prevent from changing the ratio x:y of the bounding box of the selected Handles.



Rotate Tool

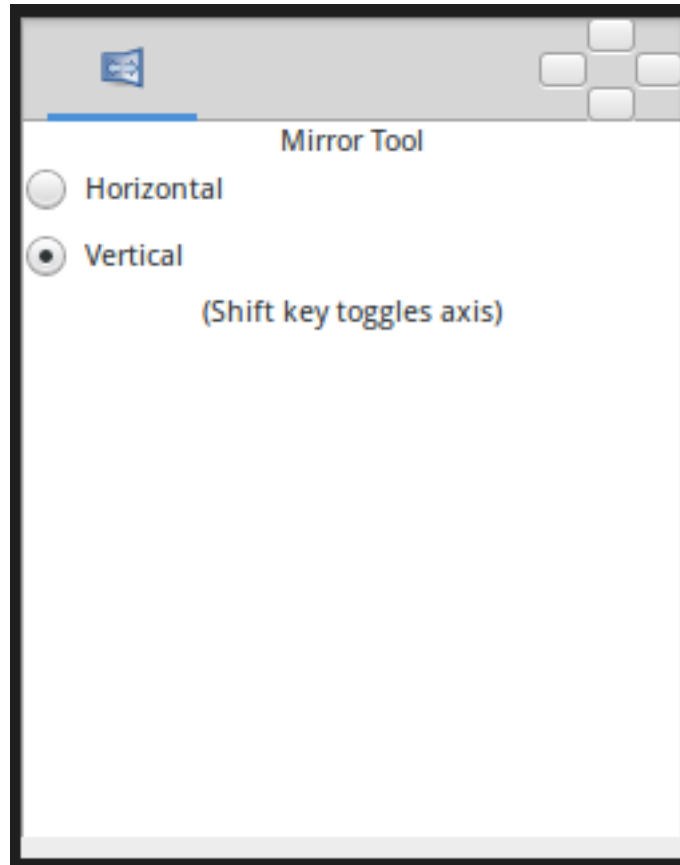
- **Allow Scale** : Allow the selected Handles to be scaled and rotated at the same time.

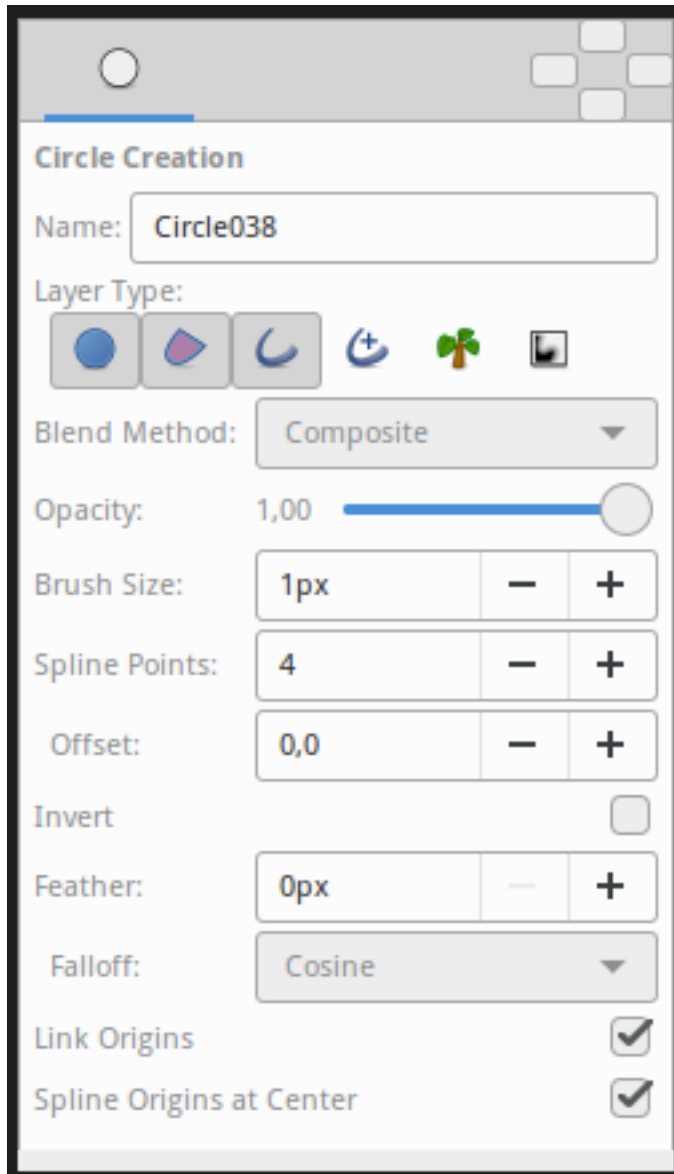




Mirror Tool

- **Horizontal** : Horizontal axis mirror.
- **Vertical** : Vertical axis mirror.



 Circle Tool

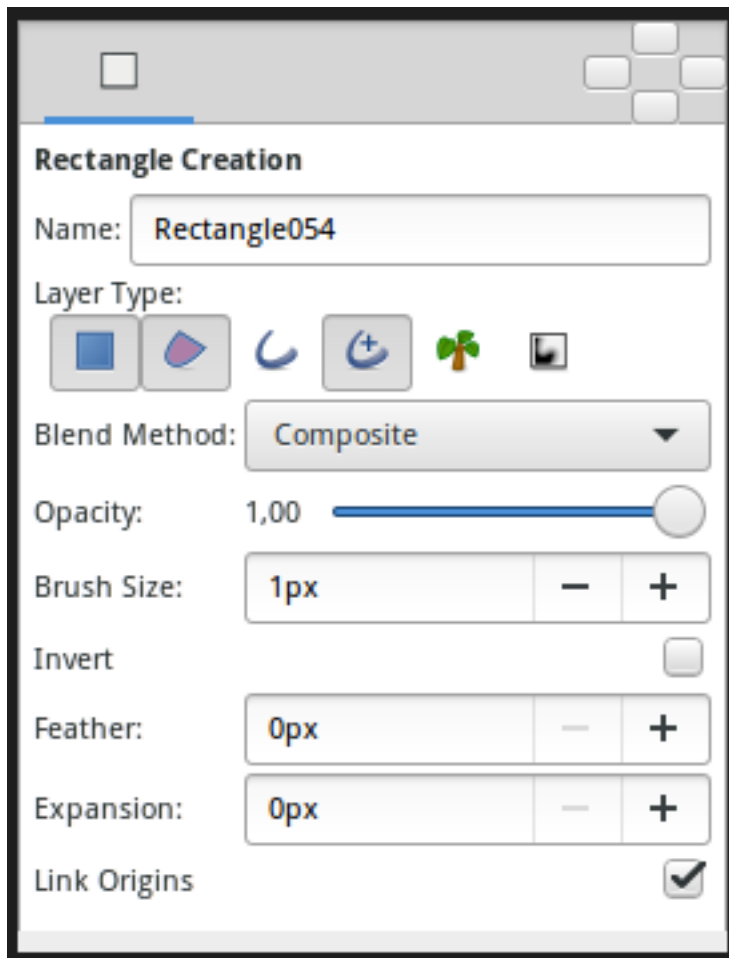
The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created.
- **Layer Type** : Whether create *Star Layer* and/or *Outline Layer*, *Advanced Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines).
- **[Blend_Method_Parameter]** : The blending method used to composite on the layers below.
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size** : Sets the Outline width ([Outline_Layer#Outline_width]) or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Spline Points** : Specify the number of Spline Points that will be used in the new Splines (for other than *Circle Layer*).

- **Offset** : Offset of the tangent of the Spline Points (for other than *Circle Layer*).
- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert*.
- **[Feather_Parameter]** : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*).
- **Falloff** : Determines the falloff function for the feather (only for *Circle Layer*).
- **Link Origins** : Whether Link the Offset of Splines.
- **Spline Origins at Center** : Set the origin of the layer at the center of the circle, else the center is set at the *Canvas* center.



Rectangle Tool

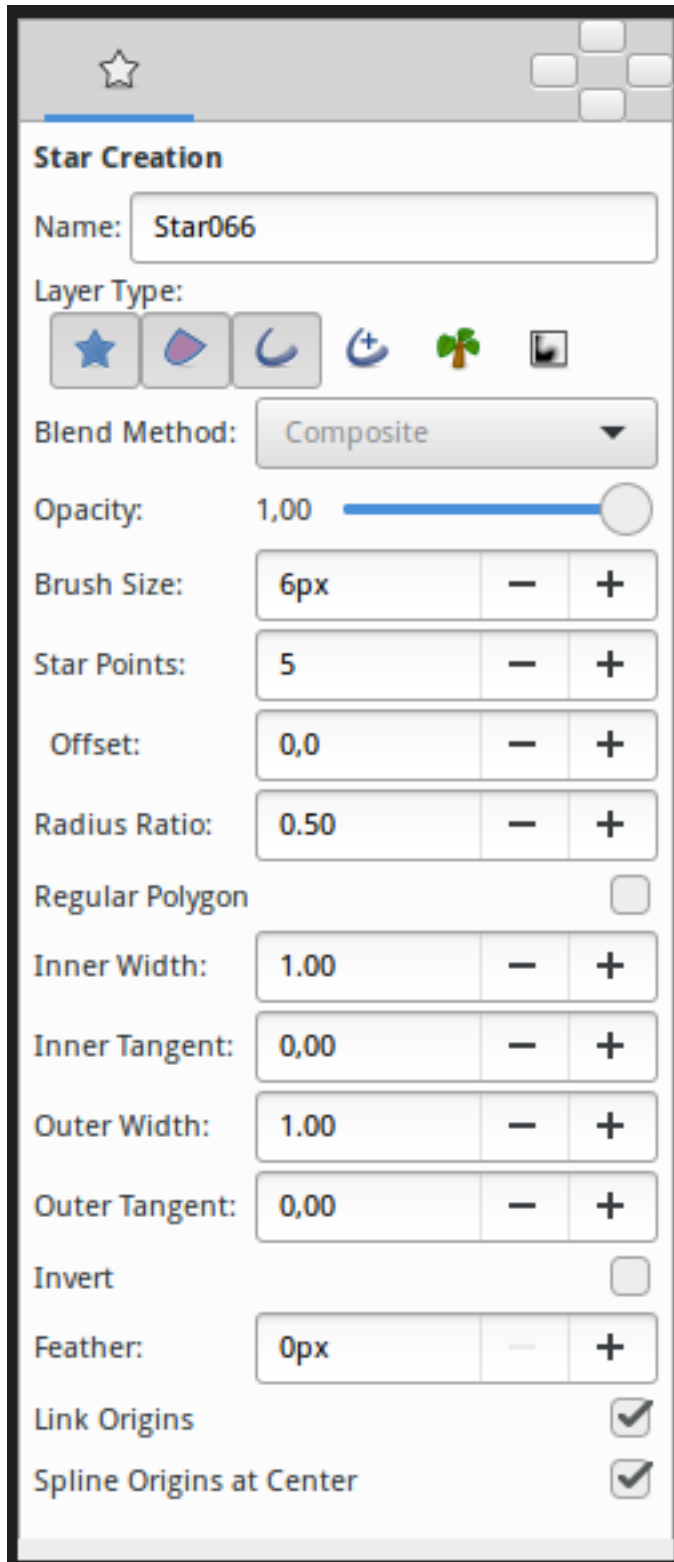


The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer name; any number in this field will automatically increment with each layer created.
- **Layer Type** : Create *Rectangle Layer* and/or *Region Layer*, *Outline Layer*, *Advanced Outline Layer*, *Plant Layer* and *Curve Gradient Layer* (Splines).
- **[Blend_Method_Parameter]** : How to combine the new layer with the background.

- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size** : Sets the Outline width ([Outline_Layer#Outline_width]) or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Expansion** : Expand the rectangle from his corners (only for *Rectangle Layer*. Is related to his Expansion Amount Parameter).
- **[Feather_Parameter]** : Width of the area to be dissolved at the edge (not for *Rectangle Layer*, *Plant Layer* and *Curve Gradient Layer*).
- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert*.
- **Link Origins** : Whether Link the Offset of Splines.

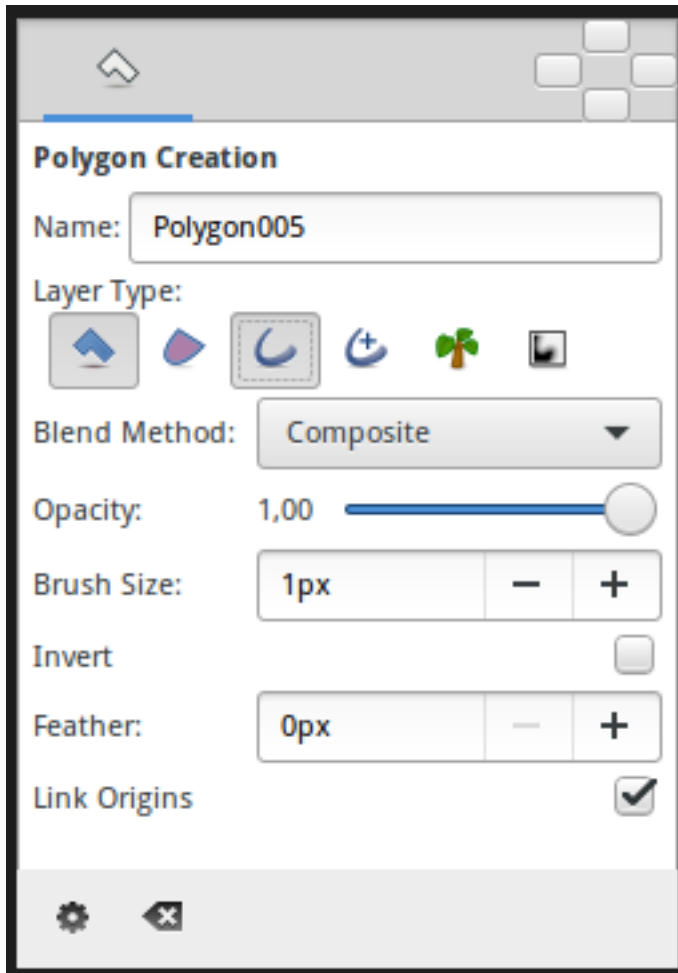
Star Tool



The *Tool Options Panel* allows you to specify:

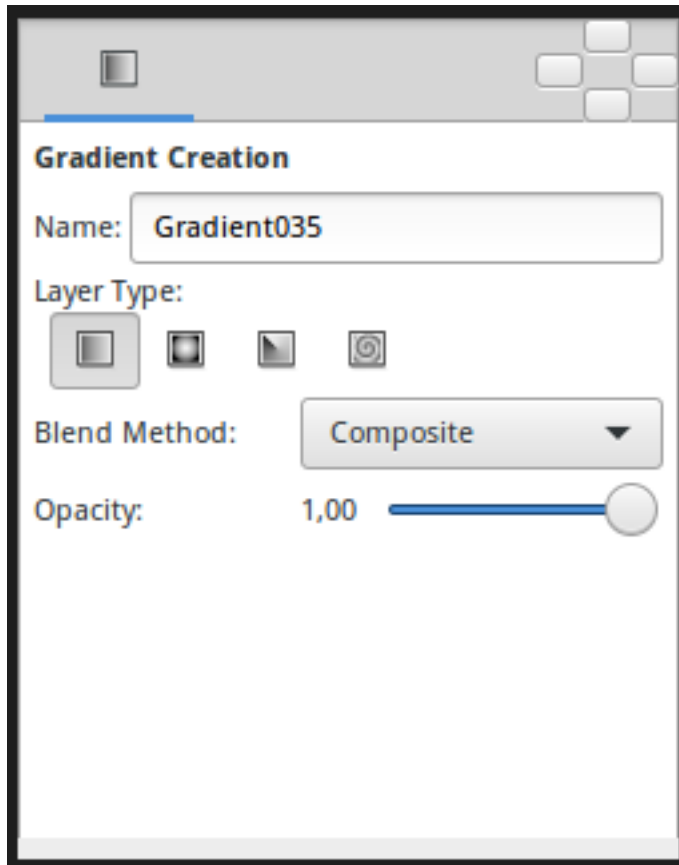
- **Name** : The name used for the new layer; any number in this field will automatically increment with each layer created.
- **Layer Type** : Whether create *Star Layer* and/or *Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines).
- **[Blend_Method_Parameter]** : How to combine the new layer with the background.
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size** : Sets the Outline width ([Outline_Layer#Outline_width]) or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Star Points** : The number of peaks of the star.
- **Offset** : The rotation offset of the created star.
- **Radius Ratio** : The ratio between the peaks and the troughs of the star.
- **Regular Polygon** : Whether create a regular polygon instead of a star.
- For Splines only, the tangent lengths for outer and inner vertices. Outer is limited to [-3,10] and inner is limited to [-3,3]. (*If you find useful open those limitations please write it in the talk page*).
 - **Inner Width**
 - **Inner Tangent**
 - **Outer Width**
 - **Outer Tangent**
- **Invert** : Whether invert the create(d) layer(s).
- **[Feather_Parameter]** : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*).
- **Link Origins** : Whether Link the *Origin* of Splines.
- **Spline Origins at Center** : Set the *Origin* of the layer at the center of the star, else the center is set at the *Canvas* center.

Polygon Tool



The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created.
- **Layer Type** : Whether create *Polygon Layer* and/or *Outline Layer*, *Advanced Outline Layer*, *Region Layer*, *Plant Layer* and *Curve Gradient Layer* layers (Splines).
- **[Blend_Method_Parameter]** : The blending method used to composite on the layers below.
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size** : Sets the Outline width ([Outline_Layer#Outline_width]) or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Invert** : Whether invert the create(d) layer(s). Is related to his *Invert*.
- **[Feather_Parameter]** : Width of the area to be dissolved at the edge (not for *Plant Layer* and *Curve Gradient Layer*).
- **Link Origins** : Whether Link the Offset of Splines.

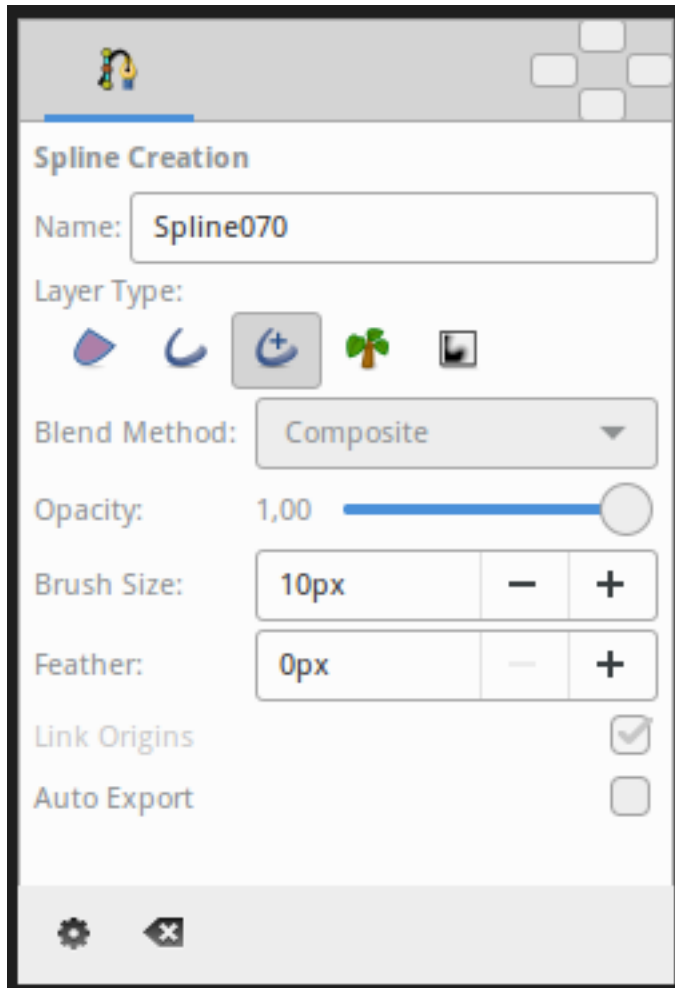
**Gradient Tool**

When you select the Gradient Tool, the *Tool Options Panel* will show the options for the Gradient Tool.

These allow you to:

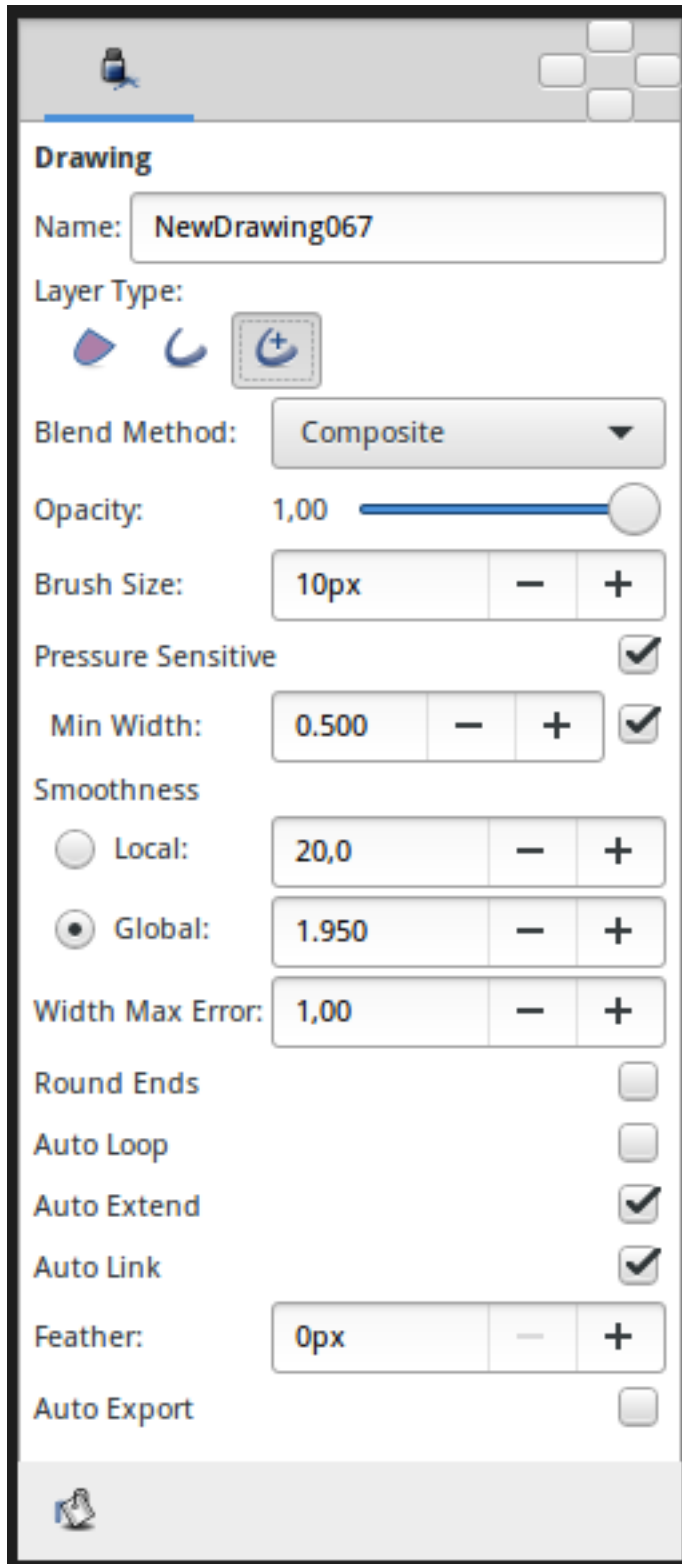
- **Name** : Set a name for the layer you are about to create. The name of the layer can always be changed later via the *Layers Panel* or the *Parameters Panel* if necessary, but the type of gradient needs to be chosen now.
- **Layer Type** : Choose the type of gradient to create (Linear, Radial, Conical, Spiral).
- **[Blend_Method_Parameter]** : The blending method used to composite on the layers below.
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).

For help with editing gradient colors see the section on *Gradient*.



This tool has the following options:

- **Name:** Sets the name of the Spline that you will create, in this case “Spline070”. If the ending part of the string contains an integer number, this input text box will parse it and increase that number for the next created Spline. Otherwise, it will add a three digit number while creating the next one.
- **Layer Type:** Sets whether to create *Region Layer*, *Outline Layer*, *Advanced Outline Layer*, *Plant Layer* and/or *Curve Gradient Layer* Layers.
- **[Blend_Method_Parameter]:** Sets the blending method used to composite on the layers below.
- **Opacity:** Sets the *Amount Parameter* for new layers. Defaults to **1.00** (Completely opaque).
- **Brush Size:** Sets the size of the line for *Outline Layer* and/or *Advanced Outline Layer*.
- **Feather:** Sets the *Feather Parameter* of the *Region Layer* or *Outline Layer* created.
- **Link Origin:** If checked, links the *Origin* for the *Plant Layer*, *Region Layer* or *Outline* if two of them (or all them) are checked.
- **Auto Export:** If checked, exports automatically the Vertices Parameter (that’s a Spline type parameter).

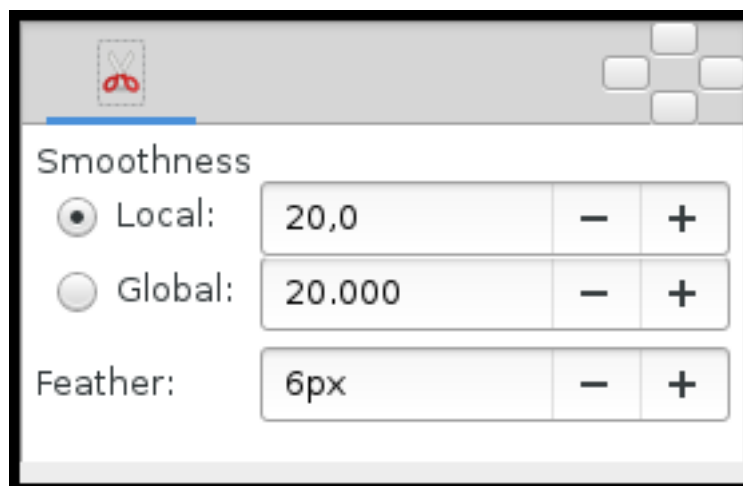


The *Tool Options Panel* allows you to specify:

- **Name** : Used for the new layer; any number in this field will automatically increment with each layer created.
- **Layer Type** : Whether create *Outline Layer* and/or *Advanced Outline Layer*, *Region Layer* layers (Splines).
- **[Blend_Method_Parameter]** : The blending method used to composite on the layers below.
- **Opacity** : Sets the *Amount Parameter* for new layers. Defaults to 1.00 (Completely opaque).
- **Brush Size** : Sets the Outline width ([Outline_Layer#Outline_width]) or the size of the gradient (only for *Outline Layer*, *Advanced Outline Layer* and *Curve Gradient Layer*).
- **Pressure Sensitive**
 - **Min Pressure** : The value of this option clamps the lower end of the Pressure sensitivity to a specific value, essentially determining how thin the thinnest possible segment is relative to the Brush Size ([New Layer Defaults#Brush Size]) parameter in the *Toolbox*.
- **Smoothness** : Determines how much tablet jitter is removed, and as a result, how many vertices the resulting splines are composed of. Could be Local or Global
 - **Local** : Adjust the Spline between the Splines points.
 - **Global** : Adjust the Spline globally.
- **Width Max Error** : for advanced outline layer.
- **Round End** : The ends of the spline will be rounded, only for *Outline Layer*, *Advanced Outline Layer* layers.
- **Auto Loop** : When the stroke of the Draw Tool ends near the beginning of the stroke, automatically connect the last *Handle* to the first.
- **Auto Extend** : When the stroke begins and/or end at or near an endpoint *Handle* of another *Outline Layer*, the resulting *Outline Layer* is simply appended to the existing *Outline Layer*.
- **Auto Link** : When the stroke begins and/or ends at or near a vertex *Handle* in another *Outline Layer*, and the outline isn't being extended by the *Auto Extend* option above, the first and/or last vertex of the resulting *Outline Layer* is **[Linking]** to the *Handle* on the existing *Outline Layer*.
- **Feather** : Set the *Feather Parameter* on the layer.
- **Auto Export** : Automatically *Export* the Spline list points into the Values Base Nodes of the *Library Panel*.



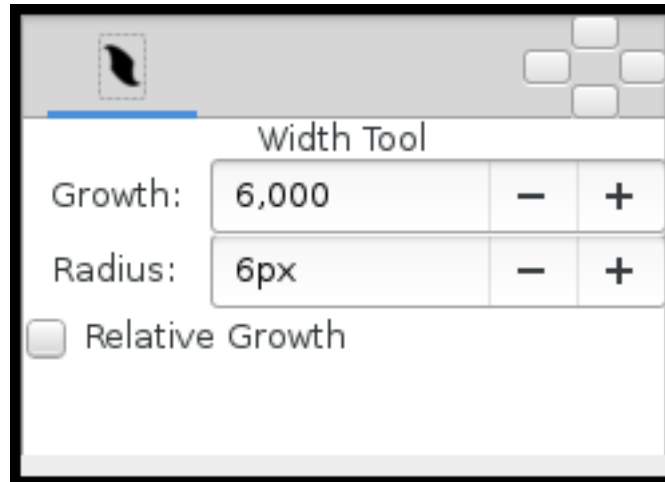
Cutout Tool



- **Smoothness** : Adjust the Spline defining the Mask. Could be Local or Global
 - **Local** : Adjust the Spline between the Splines points.
 - **Global** : Adjust the Spline Globally.
- **[Feather_Parameter]** : Width of the area that is going to be dissolved at the edge of the mask.



Width Tool



- **Growth** - Defines how much a single mouse move will affect the line width. It must be non-zero for this tool to have an effect on the spline.
- **Radius** - Defines the size of area around current cursor position in which a vertex will be affected. It allows you to achieve a noticeable effect without having to follow the line precisely with the mouse. The Width Tool works fine on splines with lots of segments (such as those created with the *Draw Tool*), you will see it works on more than just the first vertex. It is a full circular area, other points may get in the way though.
- **Relative Growth** - Doesn't really work very well unless the Radius parameter is set to some huge value (like a million).



Fill Tool

- None



Eyedrop Tool

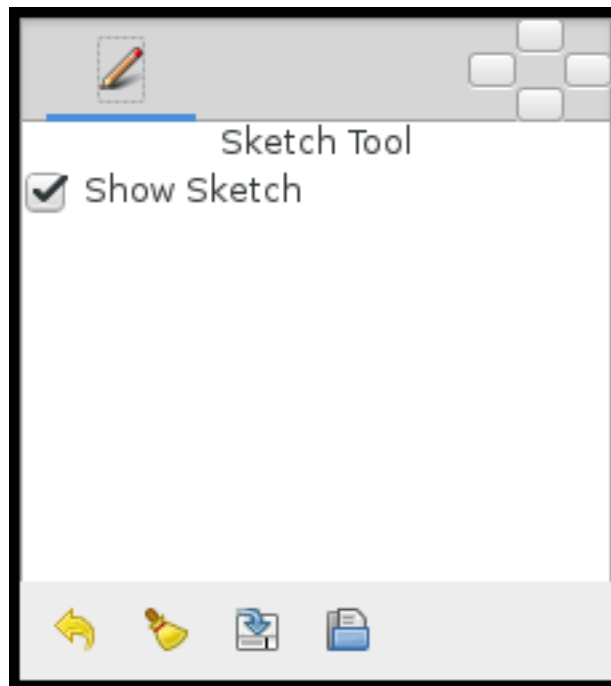
- None

Text Tool

The *Tool Options Panel* allows you to specify:

- the **Name** used for the new layer; any number in this field will automatically increment with each layer created.
- **Multiline Editor** - whether to use a single or multi-line editor for entering the text. If a single line is used then the **Enter** key will submit the value. In the multi-line editor, you'll need to click the **OK** button with the mouse to confirm the text or hit **Ctrl|Tab** to give focus to **OK** button.
- **Size X Y** - the horizontal and vertical size of the text.
- **Orientation X Y** - the orientation of the text. The default **(0.5, 0.5)** means that the text will be centered around the point you clicked on. **(0, 0)** means that the top left corner of a box containing the text will be placed where you clicked. **(1, 1)** means that the bottom right corner of the text's bounding box will be placed where you clicked.
- **Family** - the font family to be used.

Sketch Tool

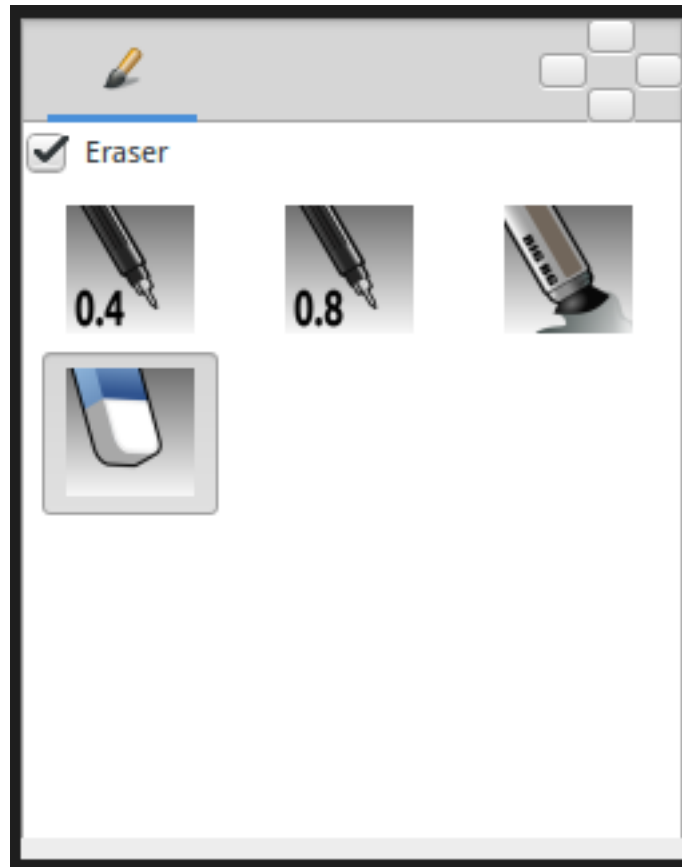


The Sketch Tool has the following in its *Tool Options Panel* :

- **Show Sketch**: a checkbox for toggling the visibility of the sketch overlay **On** and **Off**.
- **Undo Last Stroke**: cancels the most recent edit to the sketch overlay. Can be repeatedly clicked to cancel more edits.
- **Clear Sketch**: clears the current workarea's sketch overlay.
- **Save Sketch As...**: writes the sketch overlay from the current workarea to a .sketch file.
- **Open a Sketch**: loads a previously saved .sketch file, overwriting any previously drawn or opened sketch in the current workarea.

Sketch uses the color of the default foreground color in the *Toolbox*.

Brush_Tool



- Eraser check box. When checked, the brush acts as a rubber.
- Display the set of configured brushes (MyPaint brushes format). By default, Synfig is installed with a selected set of brushes from “Concept Design (C_D)” pack by [Ramon Miranda](#).

From the *Preferences Dialog System*, you can configure the Brush Presets Path to use other brushes package (for example the ones listed by [MyPaint](#)).

The Brush Tool uses the color of the Outline Color ([New_Layer_Defaults#Brush_Colors]) and the Brush Size ([New_Layer_Defaults#Brush_Size]) set in the *Toolbox* to paint.



Zoom Tool

- None

12.15 Toolbox



The **Toolbox Panel** is one of the main SynfigStudio *Panels* - you can have several Canvas Window open, but only one **Toolbox Panel**.

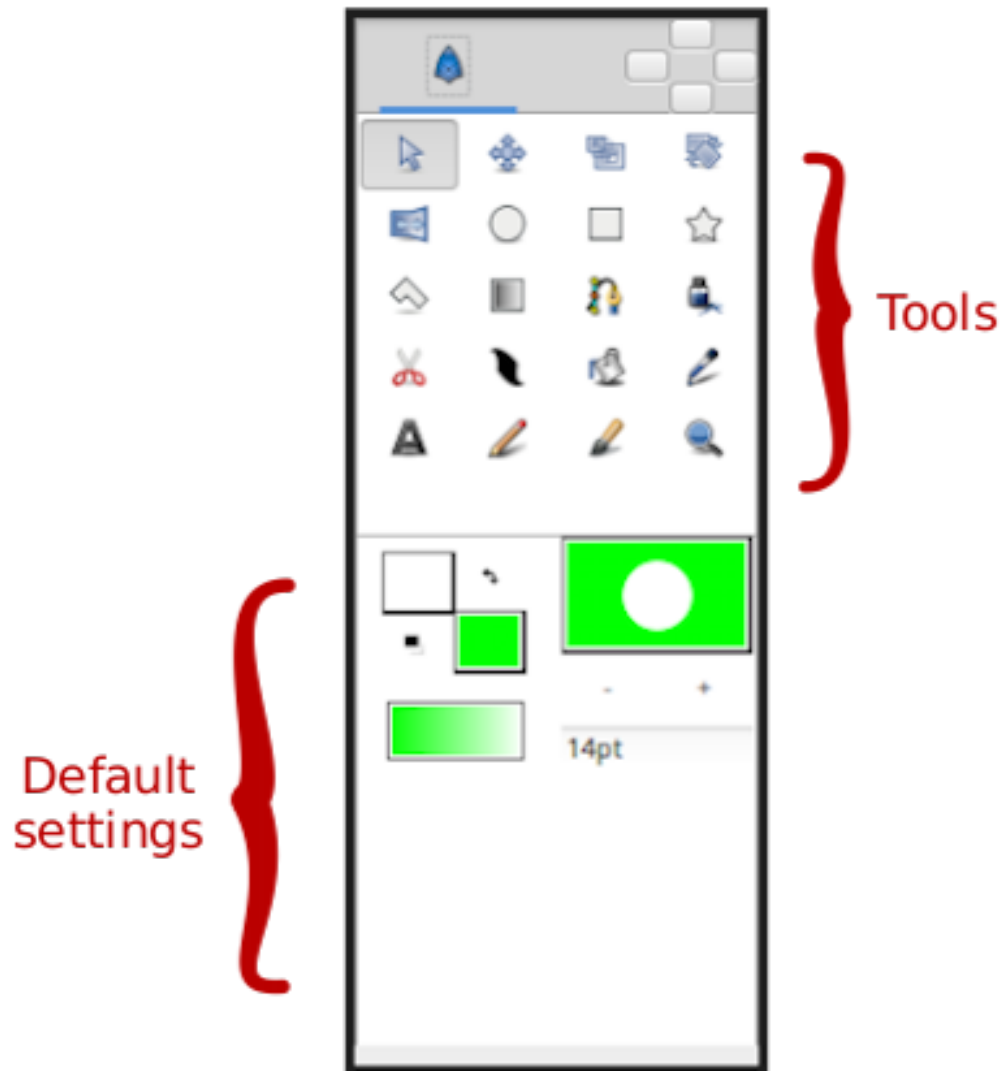


Fig. 3: The Toolbox

12.15.1 Features of the Toolbox:

- Tools
- *New Layer Defaults*

13.1 Export Animation

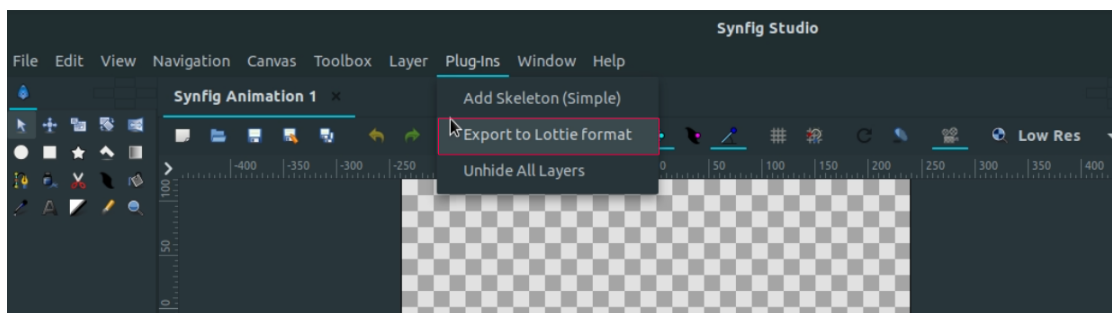
13.2 Export for Web (Lottie)

Note: Available since Synfig Studio 1.4.0.

Synfig Studio can export animations to [Lottie JSON format](#), which allows to embed them into web pages, play them natively in web browser and on mobile platforms. This format is also [used](#) for creating Telegram’s animated stickers!

Currently it is capable to export most main graphic elements, with animation and some converters applied to their parameters (see detailed list of supported layers and parameters on [this page](#)).

To export your animation file, choose “Plugins” - “Export to Lottie” from menu. This will export the currently opened file to Lottie format.



After the export, you’ll have 3 files in the same directory as your animation: **FILE.html**, **FILE.json** and **FILE.log**.

- **FILE.json** is the main file where the Lottie format animation resides. Use it if you want to embed your animation to some webpage. Here is an [example implementation at codepen.io](#). FILE.json can be directly played using [lottiefiles.com](#).
- **FILE.html** can be used to watch exported animation in a browser like Firefox or Google Chrome.
- **FILE.log** stores all messages/errors appeared during the export. If a layer is not supported, it will be notified in this file.

Limitations:

- Exporting Bones (Skeleton) animation is not implemented yet.
- Layers like region, outline, and polygon are exported frame-by-frame because there is no exact alternative provided by Lottie and hence the animations might lag on larger displays.

- Animation of addition/deletion of spline points is not supported as Lottie format does not provides this feature.

13.3 Exporting images

PLUGINS

14.1 Summary

The Plugins feature allows to run custom python scripts directly from Synfig Studio menu. Each script takes .sif file as first argument and should modify its contents in some way. After script execution finished, the file is automatically reopened back in Synfig Studio.

All plugins are located in the “Plug-Ins” submenu of the canvas.

14.2 Rationale

People often write some scripts to make useful things on Synfig (sif) files. The most of these scripts are written in python. But for ordinary users running custom scripts from terminal is tricky. With plugins feature user can install scripts as easy as they copy files and transparently run them in the same way as they use standard Synfig Studio commands. Also, running scripts from menu is much faster than from terminal and it greatly improves the workflow for advanced users. Having this feature allows to easily add simple functions to Synfig Studio by writing scripts in python.

14.3 How to install plugins

Plugin is a directory, containing the python script (*.py file), plugin.xml and maybe some other files if they are required by python script.

To install the plugin user should copy its directory into the following location:

- **Windows Vista/7/8:** C:\Users\USERNAME\Synfig\plugins
- **Windows XP:** C:\Documents and Settings\USERNAME\Synfig\plugins
- **Linux:** ~/.config/synfig/plugins
- **OSX:** /Users/USERNAME/Library/Synfig/plugins

The system-wide location for the plugins is USER_DIRECTORY/SYNFIG_CONFIGURATION_DIR/plugins

14.3.1 Where to find plugins

- [Yoyobuae:FreeForm Deform](#)
- [Morevna:Plugins feature in Synfig Studio](#)
- [Berteh:Import Labels and Timings](#)
- [Synfig forum: scripts/plugin thread](#)

14.4 Plugin structure

A plugin keeps all its files inside a directory as described above.

Synfig parses the file called `plugin.xml` found in each plugin directory, which defines plugin metadata and how to invoke the script.

This section describes the xml elements available and what they do.

14.4.1 <plugin>

Root element. Can contain the following:

- `<name>` The name of the plugin, must have at least the default version
- `<exec>` Script to run when the user clicks on the Plugin menu. Can be omitted if there are importers or exporters
- `<exporter>` Defines an exporter, a plugin can have multiple of these
- `<importer>` Defines an importer, a plugin can have multiple of these

14.4.2 <name>

Name of the plugin, can be specified multiple times to provide translations

Example:

```
<name>This is the default name</name>
<name xml:lang="it">This is name it will show if you set Italian as your language</name>
```

14.4.3 <exec>

Defines a script to run, the text contents must be a path (relative to the plugin directory) of a script to run.

If present inside `<plugin>`, the plugin will be shown in the Plugins menu, and the script will be invoked when you click on the corresponding menu item.

It has a few attributes, all optional.

type

python selects the interpreter (currently only Python is supported)

stdout

ignore What to do with the script standard output:

- **ignore** output is discarded

- **log** the output is shown in the Synfig log
- **message** an error message is shown to the user

stderr

message Same as above, but for standard error

Example:

```
<exec>myscript.py</exec>
```

Changing stream behaviour:

```
<exec stdout="log" stderr="ignore">myscript.py</exec>
```

14.4.4 <exporter>

Defines a new exporter, used to convert synfig files into other formats.

A plugin can define multiple exporters.

Exporters will be shown in the Export dialog.

The exporter contains the following sub-elements:

- <exec> must have exactly one of these
- <extension> must have at least one of them
- <description> must have the default version

Example:

```
<exporter>
  <extension>svg</extension>
  <extension>svgz</extension>
  <description>Scalable Vector Graphics (*.svg, *.svgz)</description>
  <exec>svg-exporter.py</exec>
</exporter>
```

14.4.5 <importer>

Works the same as <exporter>, but provides script to convert from other formats into synfig.

A plugin can define multiple importers.

Importers will be shown in the Open file dialog.

Example:

```
<importer>
  <extension>svg</extension>
  <extension>svgz</extension>
  <description>Scalable Vector Graphics (*.svg, *.svgz)</description>
  <exec>svg-exporter.py</exec>
</importer>
```

14.4.6 <extension>

For <importer> and <exporter>, which extensions are supported.

Multiple <extension> elements may be present in an importer or exporter (at least one is required)

Example:

```
<extension>svg</extension>
```

14.4.7 <description>

For <importer> and <exporter>, the text to be shown in the file dialog.

Similarly to <name> this can be translated using xml:lang

Example:

```
<description>Scalable Vector Graphics (*.svg, *.svgz)</description>
```

14.5 Script Invocation

14.5.1 Plugins Menu

For scripts run when the user clicks on the plugin name in the Plugins menu, synfig will save a copy of the open canvas and pass the path to that file as the argument to the script.

The script can then modify that file and synfig will reload the canvas to reflect any changes.

14.5.2 Exporter

For an exporter, synfig will pass two arguments to the script: the first is the path to a synfig file containing the open canvas; the second is the file name.

14.5.3 Importer

For an importer, synfig will pass the file selected in the open dialog as first argument, and the path to a temporary synfig file as second argument.

Once the script is completed, synfig will load that second file, so the plugin script should populate it appropriately.

14.6 Tutorial

14.6.1 Details

Each plugin located in a separate subdirectory with unique name. The part of the name before first “-” symbol is used to set the group plugin belongs to (not implemented yet). The main information about plugin (plugins name and script to execute) is stored in the plugin.xml file. It’s self-explanatory :

plugin.xml :

```
<?xml version="1.0" encoding="UTF-8"?>
<plugin>
<name>Unhide All Layers</name>
<name xml:lang="es">Activa todas las capas</name>
<name xml:lang="eu">Erakutsi geruza guztiak</name>
<name xml:lang="eu_ES">Erakutsi geruza guztiak</name>
<name xml:lang="fr">Afficher Tous les Calques</name>
<name xml:lang="lt">Parodyti visus sluoksnius</name>
<name xml:lang="ru">    </name>
<exec>view-unhide-all-layers.py</exec>
</plugin>
```

view-unhide-all-layers.py :

```
#!/usr/bin/env python

#
# Copyright (c) 2012 by Konstantin Dmitriev <k....z...gmail.com>
#
# This program is free software; you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation; either version 2 of the License, or
# (at your option) any later version.

import os
import sys

def process(filename):

    # Read the input file
    inputfile_f = open(filename, 'r')
    inputfile_contents = inputfile_f.readlines()
    inputfile_f.close()

    # Now write results to the same file
    inputfile_f = open(filename, 'w')

    for line in inputfile_contents:
        if "<layer " in line:
            inputfile_f.write(line.replace(' active="false" ', ' active="true" '))
        else:
            inputfile_f.write(line)
    inputfile_f.close()

if len(sys.argv) < 2:
    sys.exit()
else:
    process(sys.argv[1])
```

All scripts are interpreted with python 3.

In Linux and Mac OSX case Synfig Studio looks for “python” or “python3” binary. For windows case Python binary is expected at INSTALL_PREFIX/python/python.exe. New environment variable SYNFIG_PYTHON_BINARY allows to set custom path to the python 3 binary.

GLOSSARY

15.1 Spline

Splines are vector shapes made up of *Bezier curves* joined together by *vertexes*.

Splines are the most commonly used objects in any normal animation done with Synfig, and can be made using the *Spline Tool* and many other Tools.

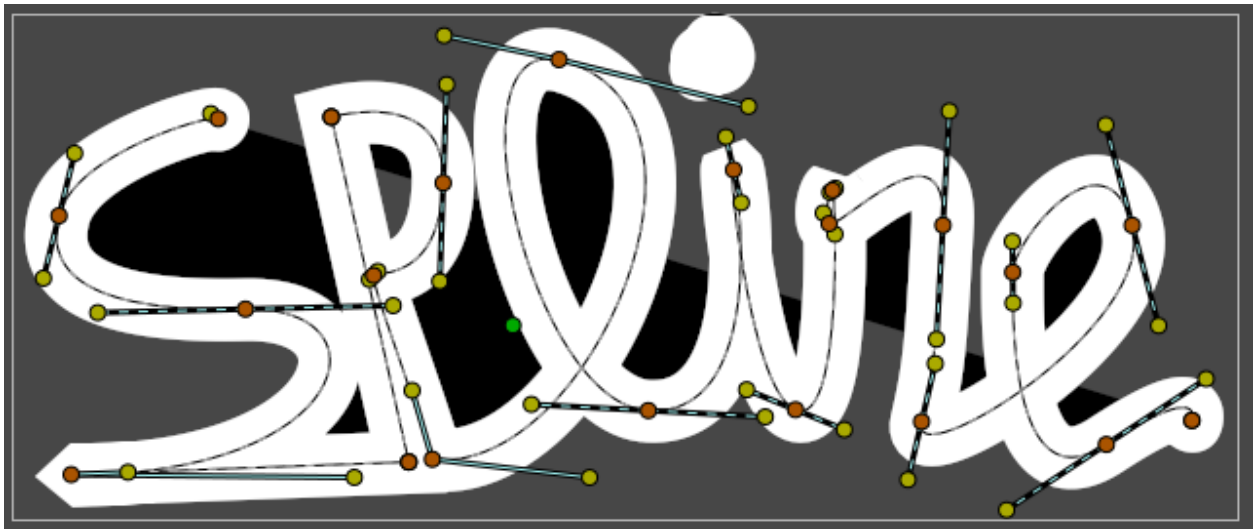


Fig. 1: A spline of “Spline”. Can see orange vertex and yellow tangent *Handle*

On the *WorkArea* you can manipulate the Spline shape by both handles (vertex or tangent) and directly from the spline segment (the curve).

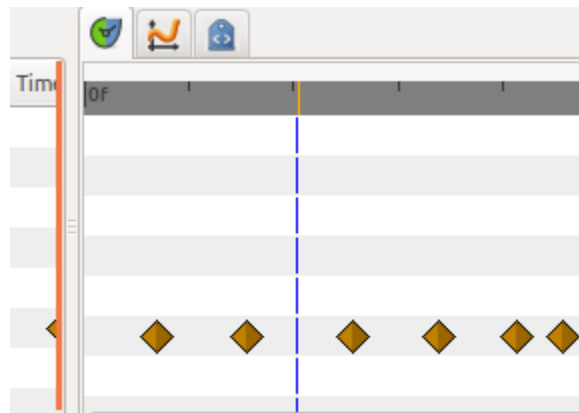
15.1.1 See Also

- [Spline Tool](#) page for more information.
- The [Draw Tool](#) has an option to Extend a spline.
- The [How Do I](#) page's [Artwork_Composition](#) ([Doc:How_Do_I#Artwork_Composition]) section has some entries about spline stuff.

Project for illustration : `Spline_of_spline.sifz`

15.2 Time Cursor

The time cursor show you, in the [Timetrack Panel](#) and [Graphs Panel](#) panels, at which frame you currently are at. It is represented by a vertical line and moves with your pointer as you drag along the [Timebar](#).



15.3 WorkArea

TODO : add a screenshot

The WorkArea is the part the Canvas Window where the art work is done. [Handle](#), Bounding box, Guides, [Grid](#) could be also shown up to your art work.

RELEASE NOTES

16.1 Development Versions

16.1.1 Synfig 1.3.14 Release Notes

Improvements

- Show interpolation type directly in waypoint's context menu (see [details](#)).

Bugfixes

- Fixed crash when removing a Spline vertex using “Remove item (Smart)” ([issue #1102](#)).
- Fixed incorrect placement of width points on outline when loading old files (see details [here](#) and [here](#)).
- Fixed crash when Shade Layer has no sublayers ([issue #1272](#)).
- Fixed popup menu disappearing right after button release for Widget Gradient/Spline ([issue #1274](#)).
- Fixed hang when opening a second .sif file from Explorer on Windows ([issue #291](#)).
- Fixed behavior “Local Time” parameter of Time Loop layer ([issue #479](#)).
- Fixed wrong percentage displayed when exporting a subset of frames ([issue #1304](#)).
- Fixed crash when undoing deletion of Group Layer ([issue #1070](#)).
- Fixed TimeTrack not updating when new waypoints added to bone ([issue #1342](#)).
- Fixed importing of 16-bit PNG files (issues [#1160](#) and [#1371](#)).
- Fixed some memory leaks ([PR #1292](#), [#1293](#), [#1319](#)).

16.1.2 Synfig 1.3.15 Release Notes

Critical fixes

- Fixed issue when copying skeletons from one file to another. Previously this was resulting in corrupted file, but now it isn't.

Advanced Outline fixes

- Fixed glitches when moving Widthpoints on unlooped Advanced Outline ([issue #1401](#)).
- Fix Incorrect position of Widthpoints on Homogenous Advanced Outline ([issue #1431](#)).
- Fixed Widthpoints that were wrongly computed on non-homogenous Advanced Outline ([issue #1523](#)).
- Fixed “Link to Spline”: now places vertex into correct position when Spline is looped ([issue #1467](#)).
- Fixed rendering of Advanced Outline width point tips ([issue #1480](#)).
- DashItem Offset and Length parameters now use consistent units when editing them ([issue #1265](#)).

Windows version fixes

- Fixed issues opening and exporting of files with multibyte (i.e. Arabic) symbols in filename.
- Fixed random crashes when editing splines ([issue #1442](#)).

UI/UX improvements

- Removed unused Change Preview Quality menu item.
- Activate widget_filename right after user chooses file via button/file dialog ([issue #1425](#)).
- Homogenous parameter of Advanced Outline is enabled and static by default.
- Set Animation Speed parameter of Noise Distort Layer as static by default.

Other bugfixes

- Fixed crash when exporting with “pngspritesheet” target ([issue #356](#)).
- Fixed crash when removing spline vertex too fast.
- Fixed crashes related with exceptions thrown in GTK event callback handlers.
- Fixed crash when undoing creation of a moved/changed vertex of spline.
- Fixed crash when unexporting value node ([issue #231](#)).
- Fixed time and delta values in tooltip while dragging in Graph panel ([issue #1405](#)).
- Fixed renaming layer set affecting other names ([issue #1146](#)).
- Fixed initial value for filename entry in Render Settings dialog.

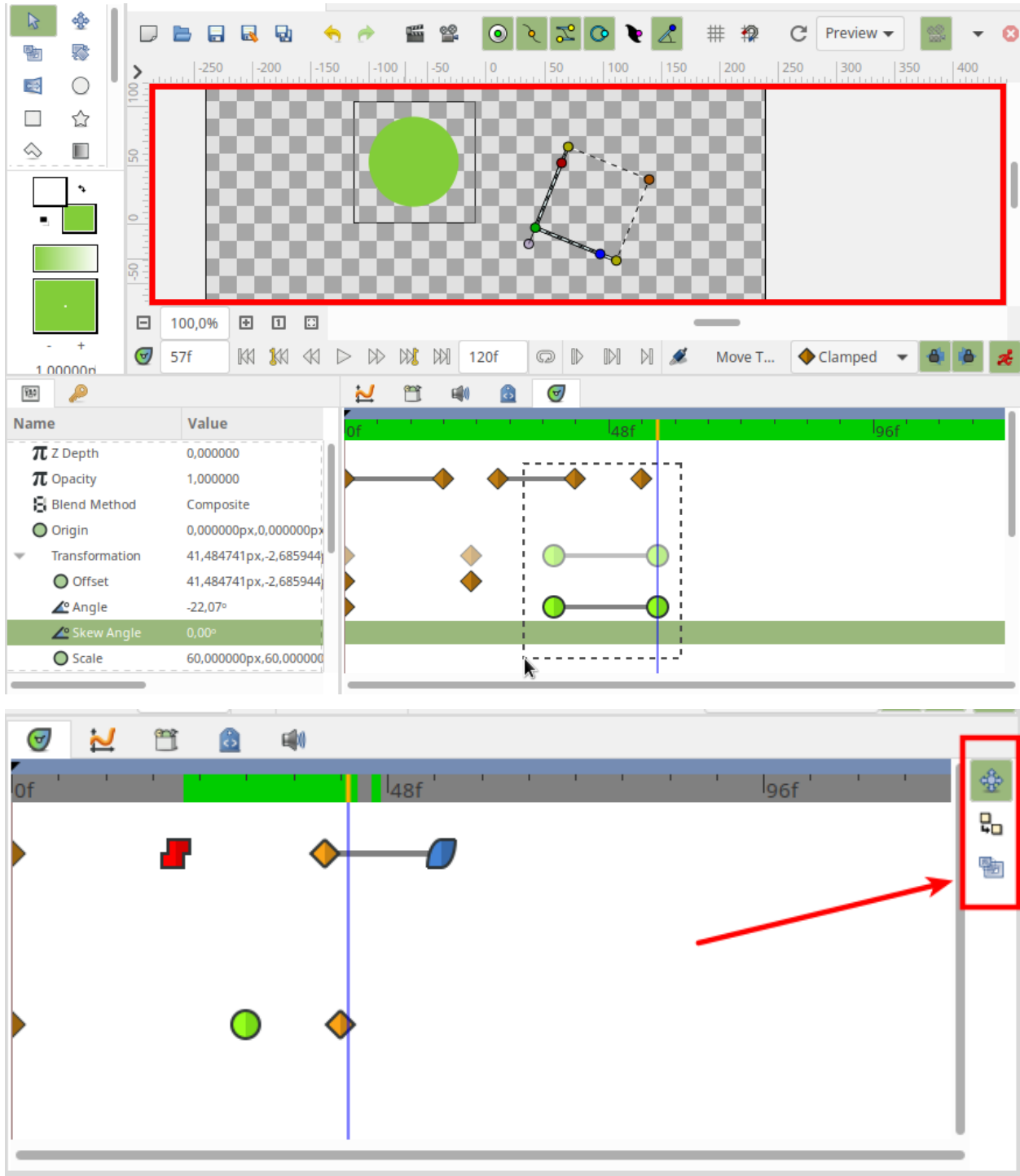
16.1.3 Synfig 1.3.16 Release Notes

- Fixed crash when right-clicking on any Waypoint in TimeTrack Panel.
- Fixed incorrect height of Keyframes widget in Sound Panel ([issue #1601](#)).
- Fixed incorrect visualisation of dashes on Advanced Outline ([issue #1589](#)).

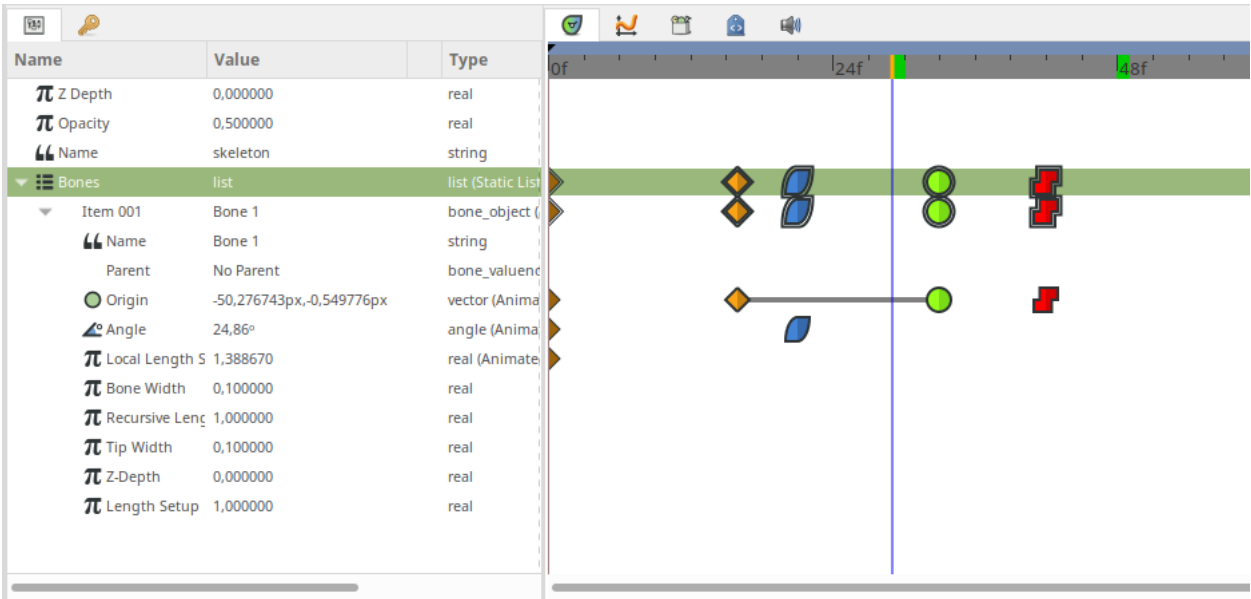
16.1.4 Synfig 1.5.0 Release Notes

Rewritten TimeTrack Panel

Now working with waypoints is so much easier - you can select multiple items using [bounding box selection](#), duplicate and scale them, navigate to next or previous waypoints using “n” and “b” keys (holding Shift skips 5 waypoints).

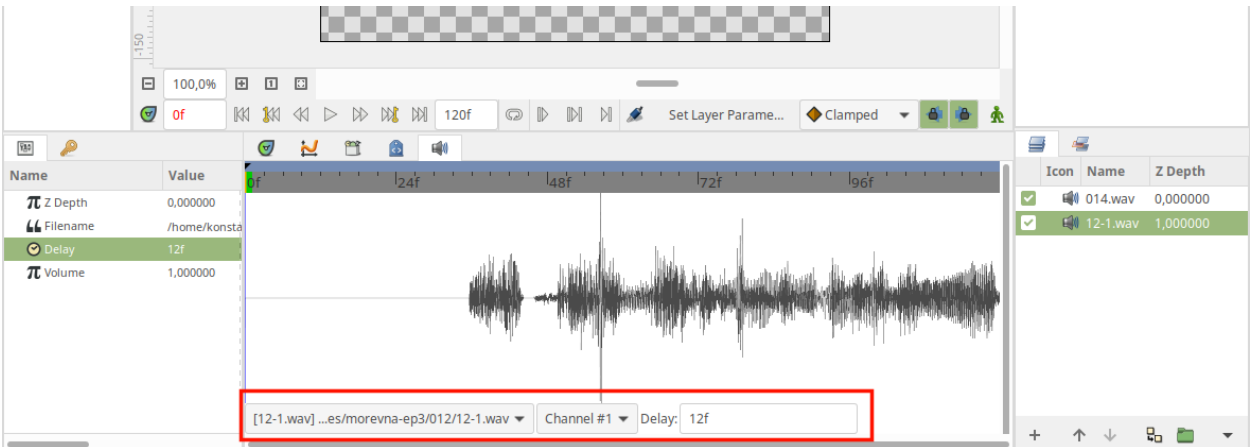


Also, the new Timetrack Panel now indicates segments when value of parameter remains constant.



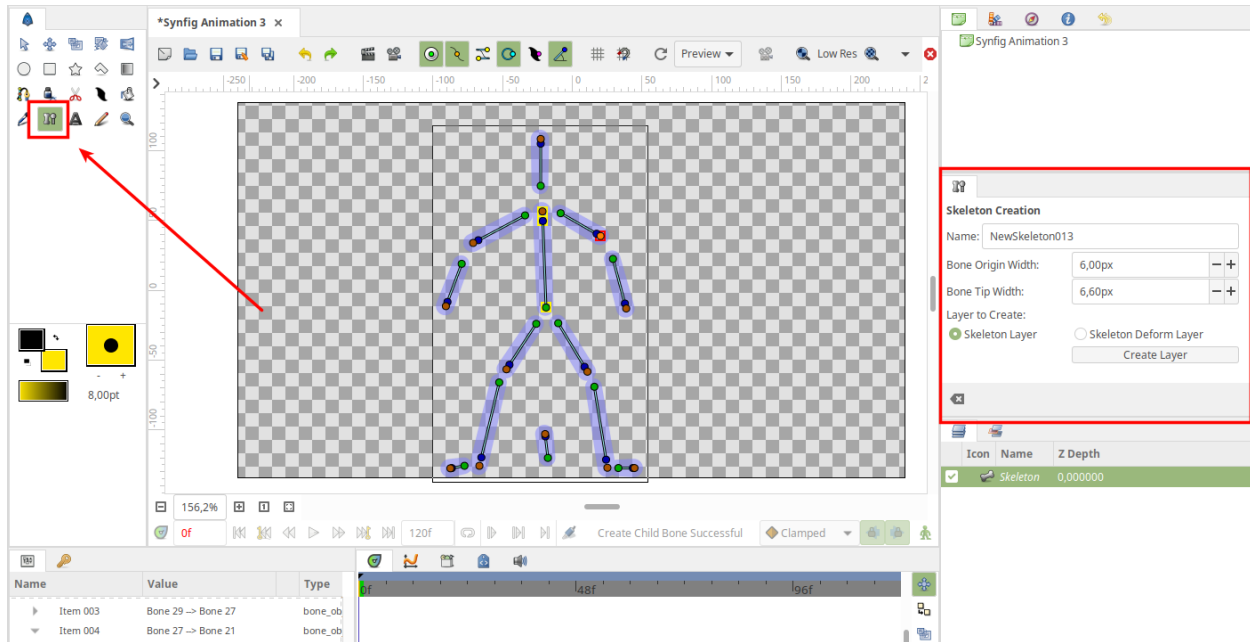
Improved Sound Panel

Sound Panel now has a list of available sound layers. This way it's much easier to choose which waveform to show in the panel. See [issue 1346](#) for details.



New Skeleton Tool

The new intuitive Skeleton Tool allows users to easily create bones. With this tool it is also possible to easily change the structure of skeletons by re-parenting bones.



See this documentation page to learn more about this feature - <https://synfig.readthedocs.io/en/latest/tools/skeleton.html>.

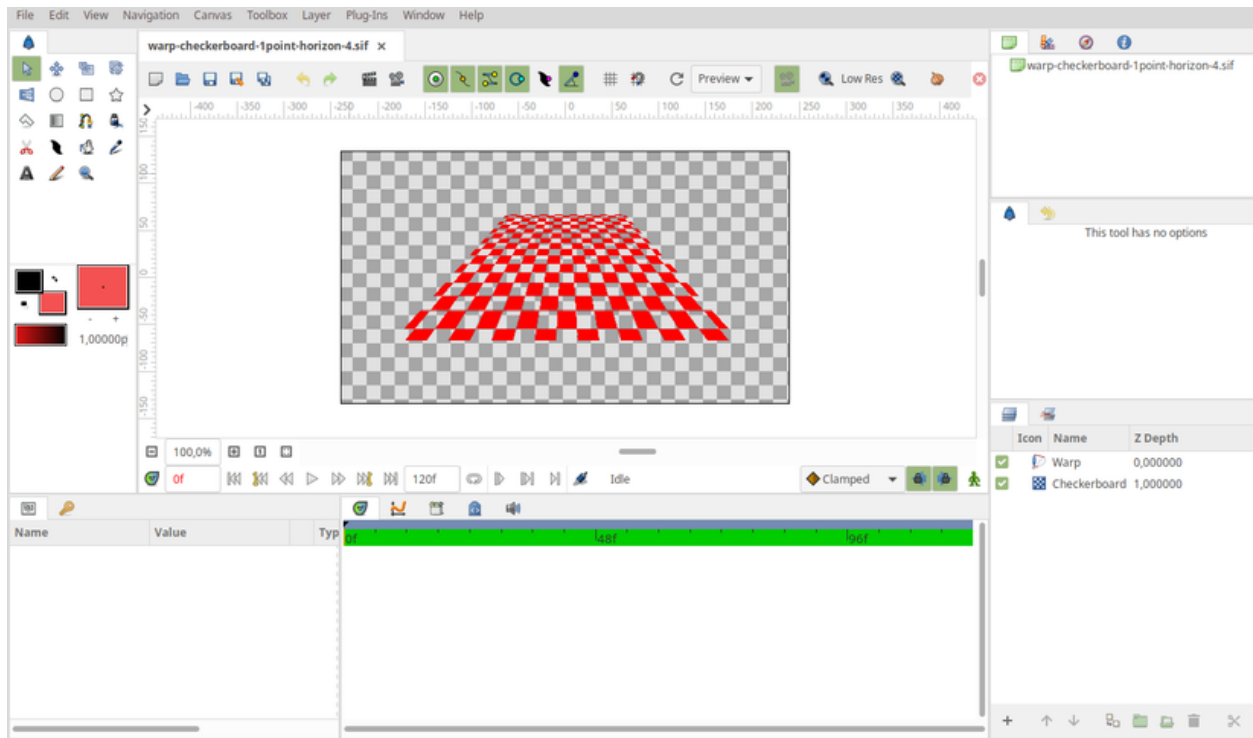
Optimized Perspective Layer

The Warp Layer was completely rewritten and ported to the Cobra render engine. We also renamed it to Perspective Layer, as it sounds more intuitive for users.

As result, the new Perspective Layer works much faster than the old Warp Layer (see [detailed report about speed tests](#)) and also fixes some old nasty bugs (see issues [#318](#) and [#418](#)).

Some other improvements in Perspective Layer:

- Perspective Layer is rendered with simplifications when the workarea uses Preview or Draft mode. This allows to achieve more than 3x speed increase.
- Perspective Layer now has a new parameter - “Interpolation” - which allows you to optimize quality for final Render.



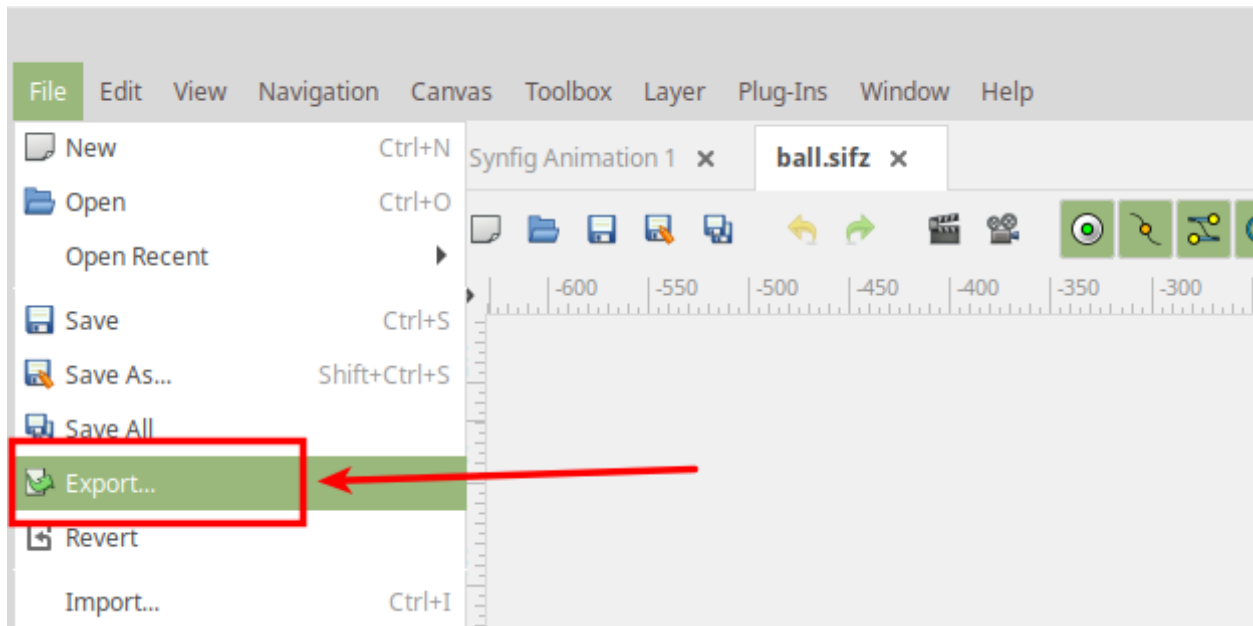
Reworked Plugins Infrastructure

Now anyone with knowledge of Python language can write plugins which allows SYnfig to export/import new formats and those formats will become available

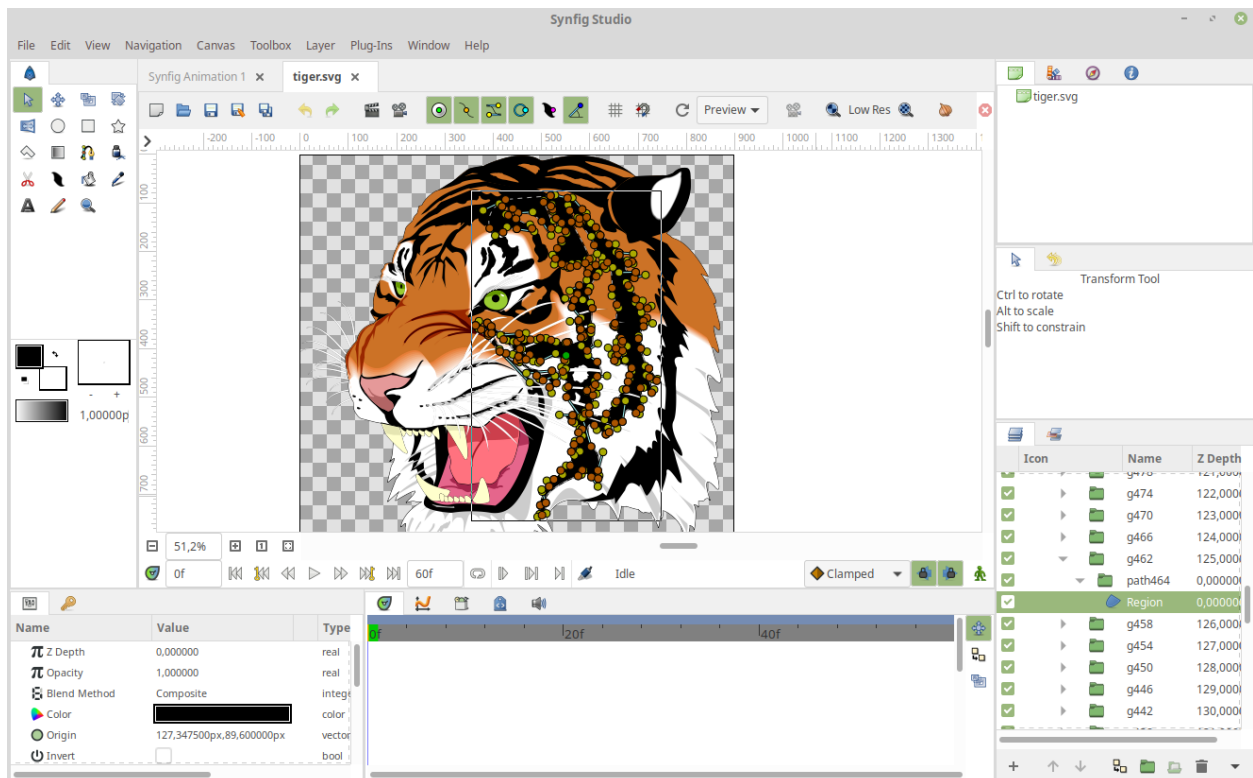
In this version the Plugins system was *improved* to allow writing import/export functions using Python language. Now anyone with Python knowledge can extend Synfig's capabilities to import or export new file formats!

Such import/export plugins are not displayed in the regular "Plugins" menu, their functionality is available through "File -> Import" or "File -> Export" menu commands instead.

With this change, the Lottie Exporter now called through "File -> Export" menu.



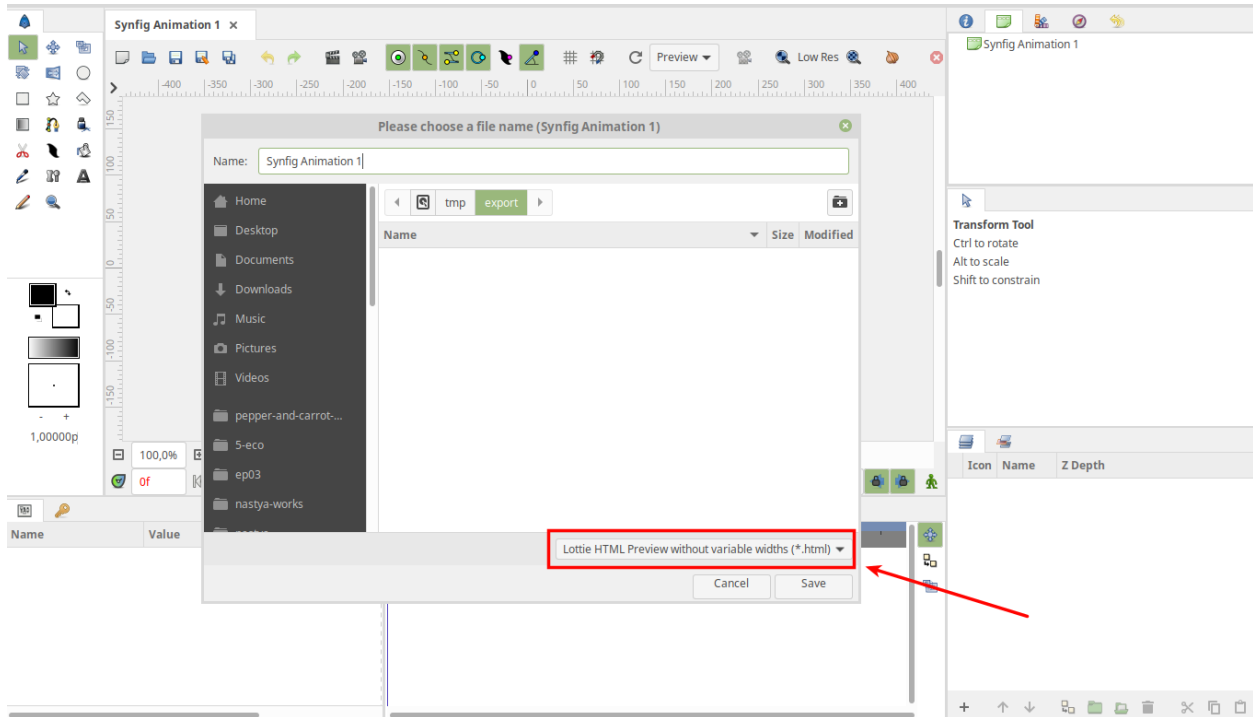
As another notable example, you might want to check [this awesome plugin](#), which adds capabilities for import/export of SVG, Telegram Animated Stickers (TGS) and Lottie (alternative implementation). The image below shows a sample “tiger.svg” file imported into Synfig using this plugin.



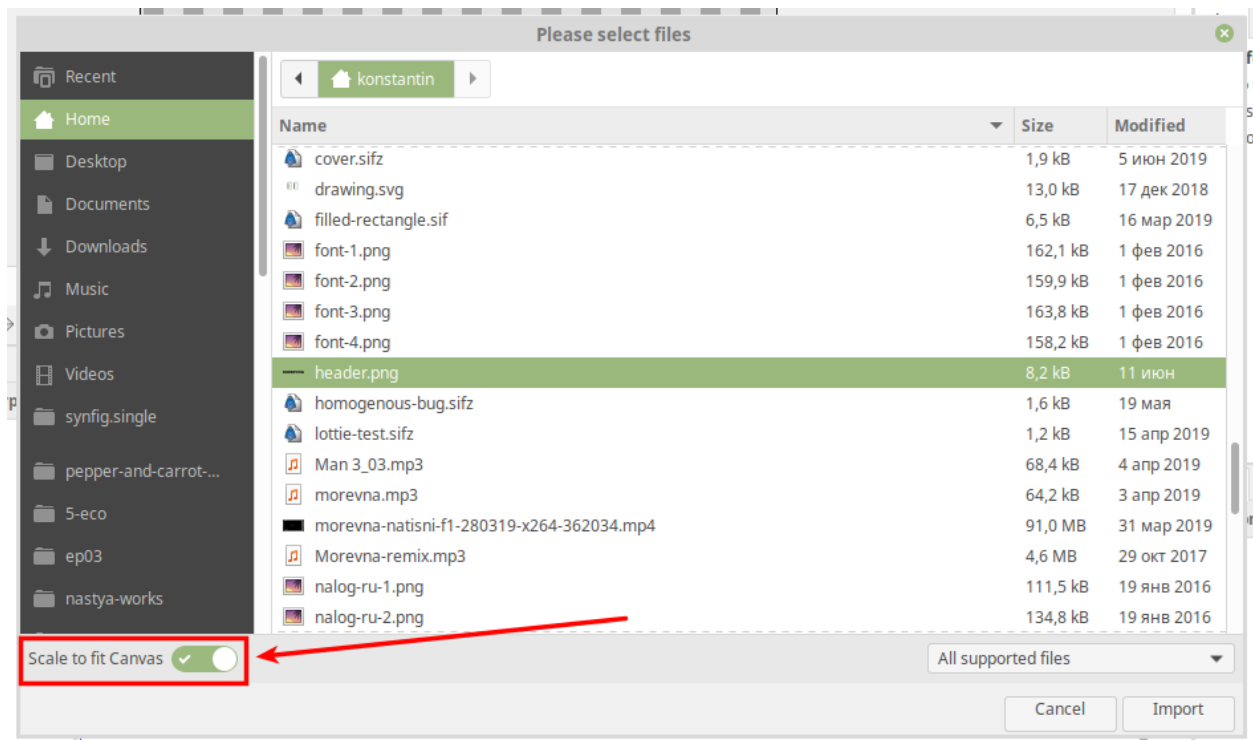
For more information about the improved plugins system check [this documentation page](#). There are also relevant discussions [here](#) and [here](#).

Improvements for Import and Export

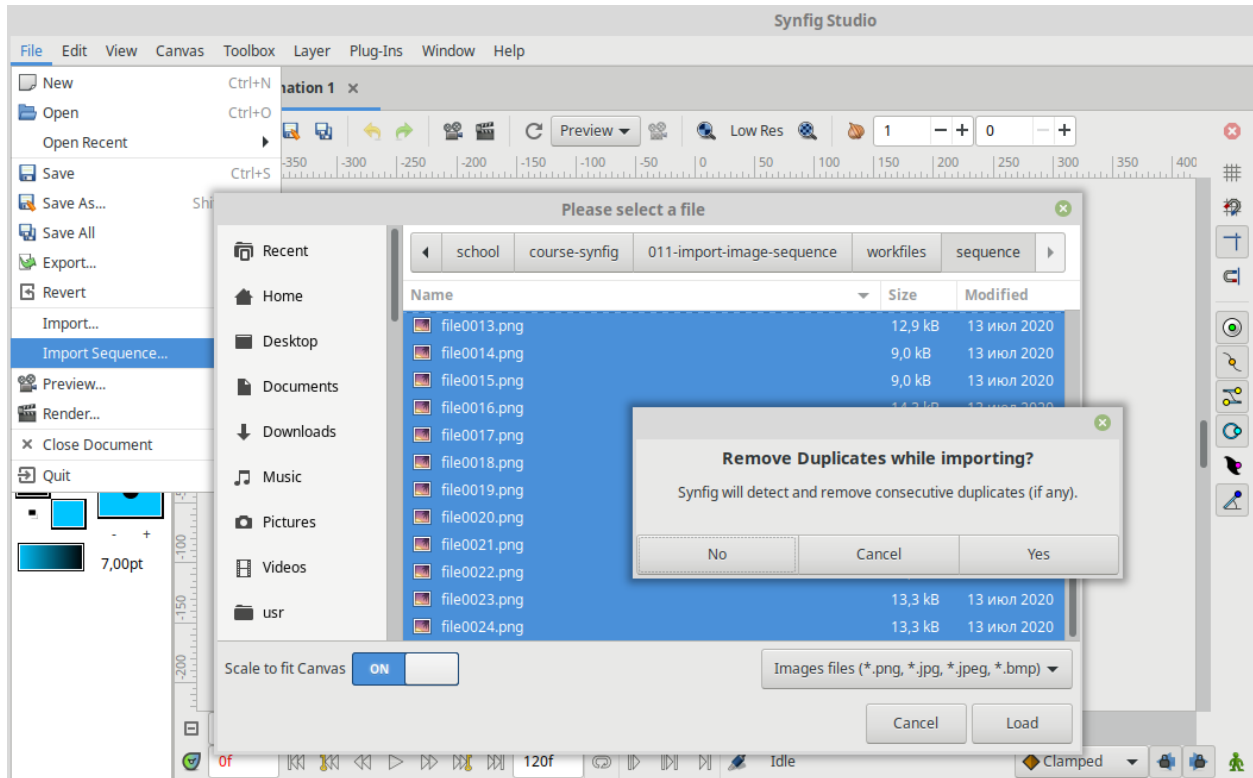
Lottie Exporter now has new mode, which exports outlines without variable widths (see PR` <<https://github.com/synfig/synfig/pull/1647>>`__#1647). This greatly reduces the size of exported files and we advise to use it if your animation does not have outlines with variable widths.



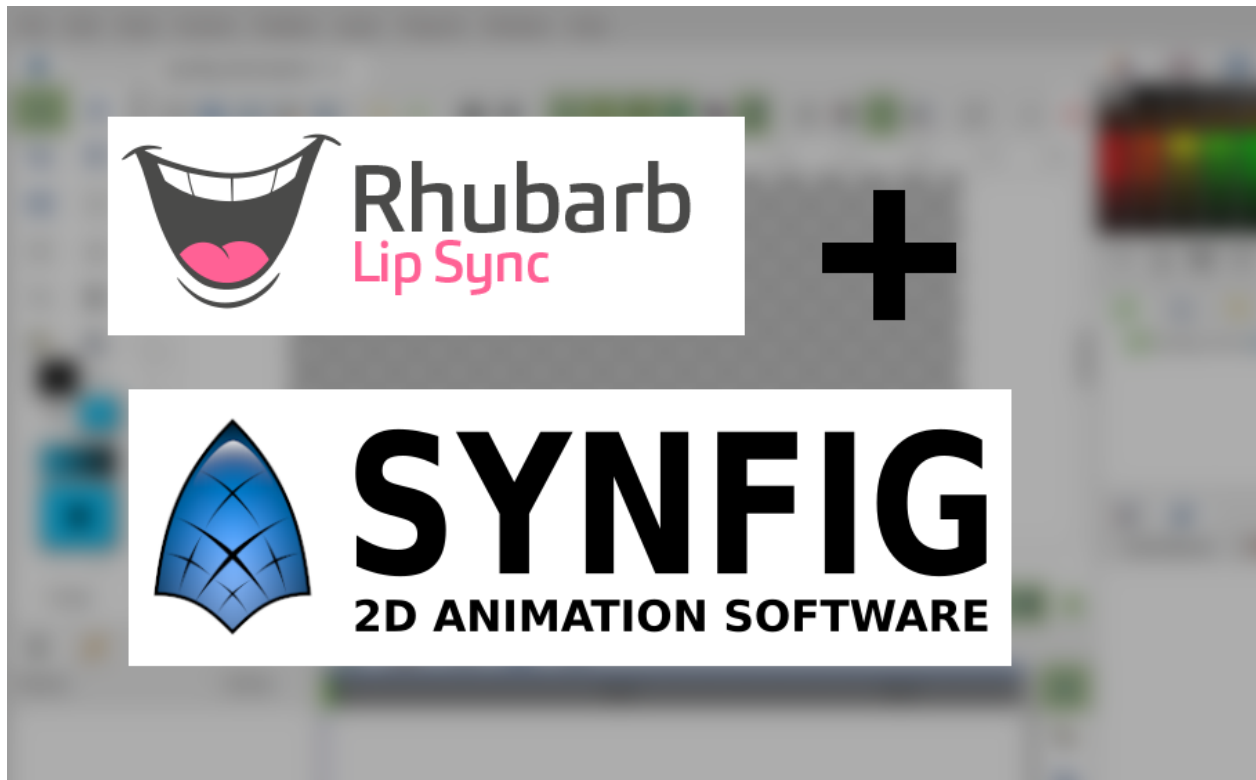
Import image dialog now has “Scale to fit canvas” option (PR #1507).



“Import Sequence” function now allows to detect duplicate frames in image sequences. In case if a duplicate frame is detected it doesn’t put extra waypoint, which results in cleaner timeline and better readability.



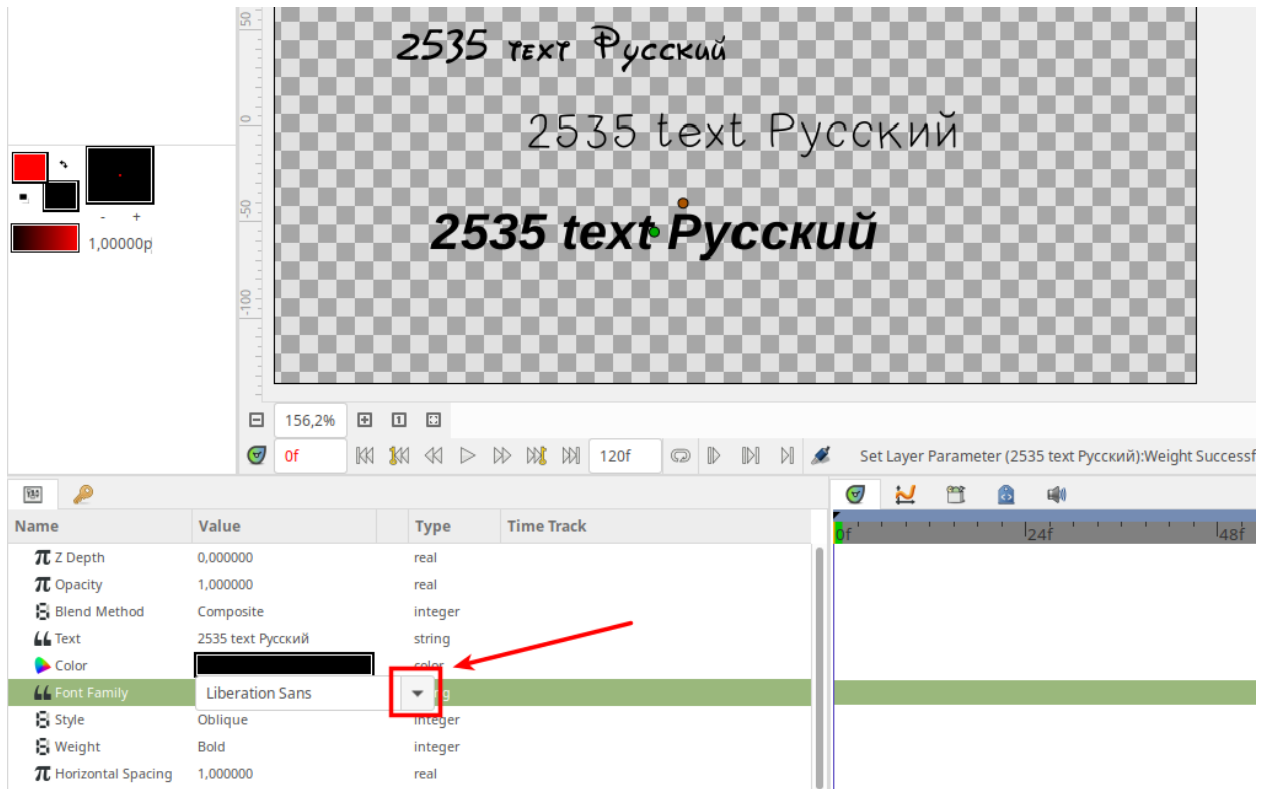
Synfig now capable to import TSV and XML file formats, generated by [Rhubarb Lip Sync](#) (see [issue #2169](#) for details). It works in the same way as Papagayo Lipsync import, shown in [this video](#).



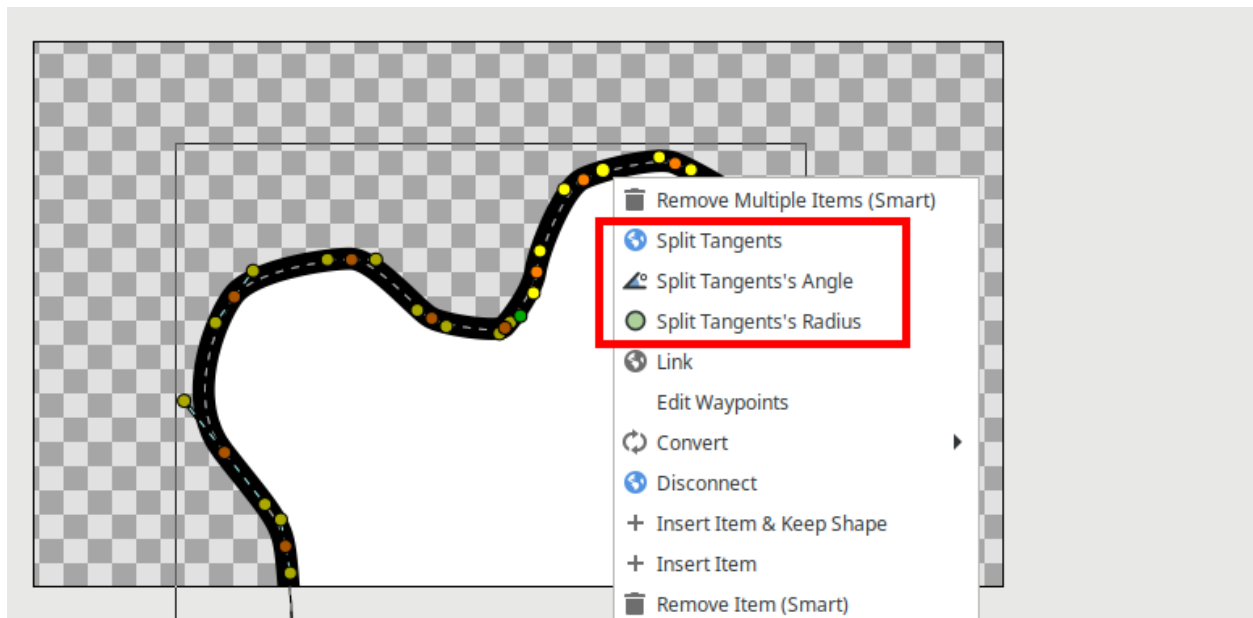
AnimatedFile Converter got support for more data types - String, Angle (degrees), Bool, Integer, Real, Time and Vector. This is useful for cases when you have animation data (like object coordinates along time) generated by another software and you wish to use that data in Synfig. See issue [#2211](#) for details.

Usability Improvements

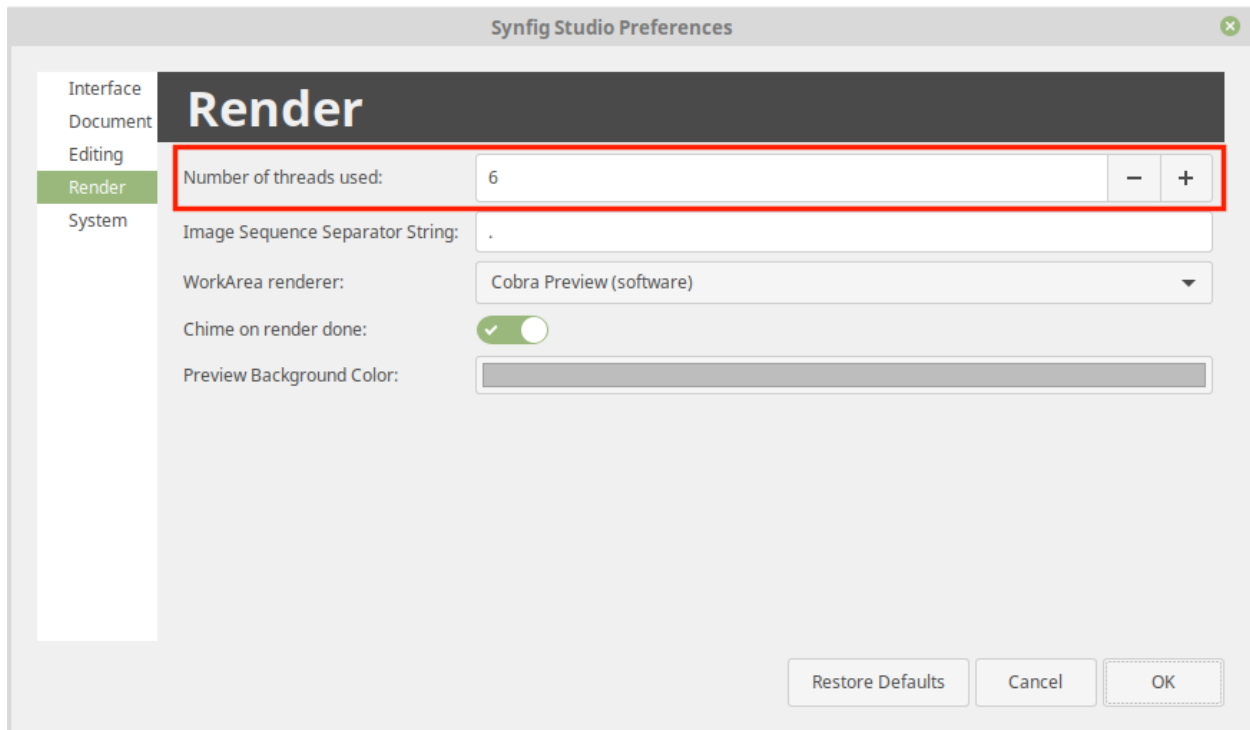
For Text Layer it is now possible to select font family from drop-down menu (PR [#1432](#)) - the list shows all fonts available in your system, so it is so easy to choose one now!



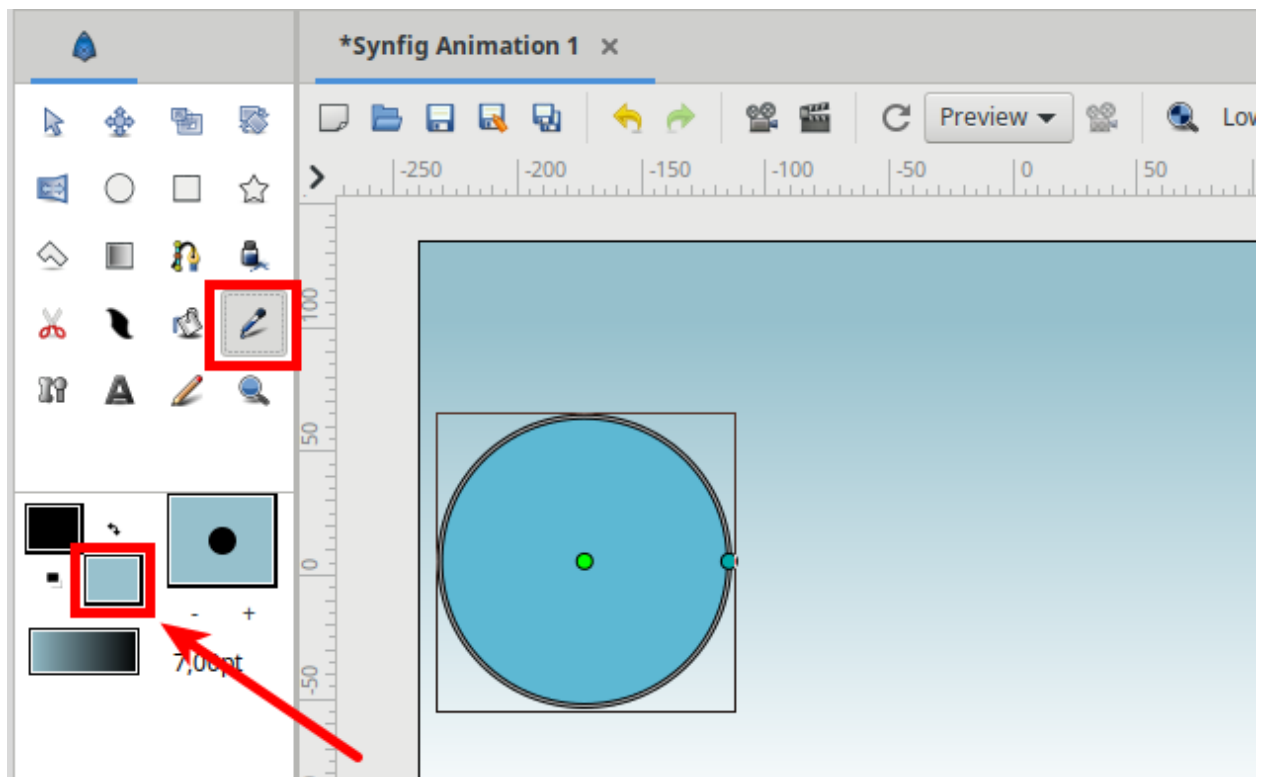
In this version it is possible to merge/split tangents for multiple vertices. See [issue #2010](#) for details.



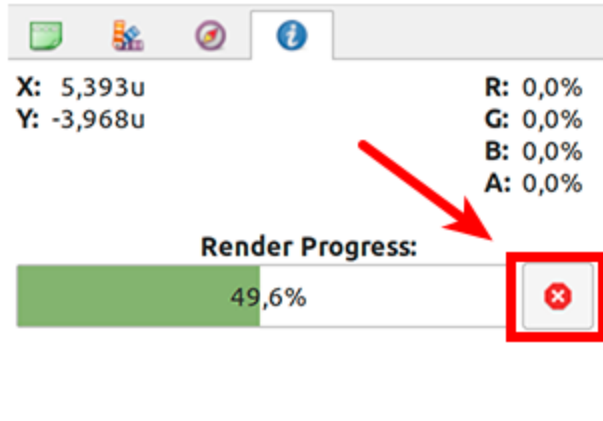
Now it is possible to set the number of rendering threads from Preferences dialog ([#1999](#)).



Eyedropper Tool now has the ability to select fill color using Ctrl+click.

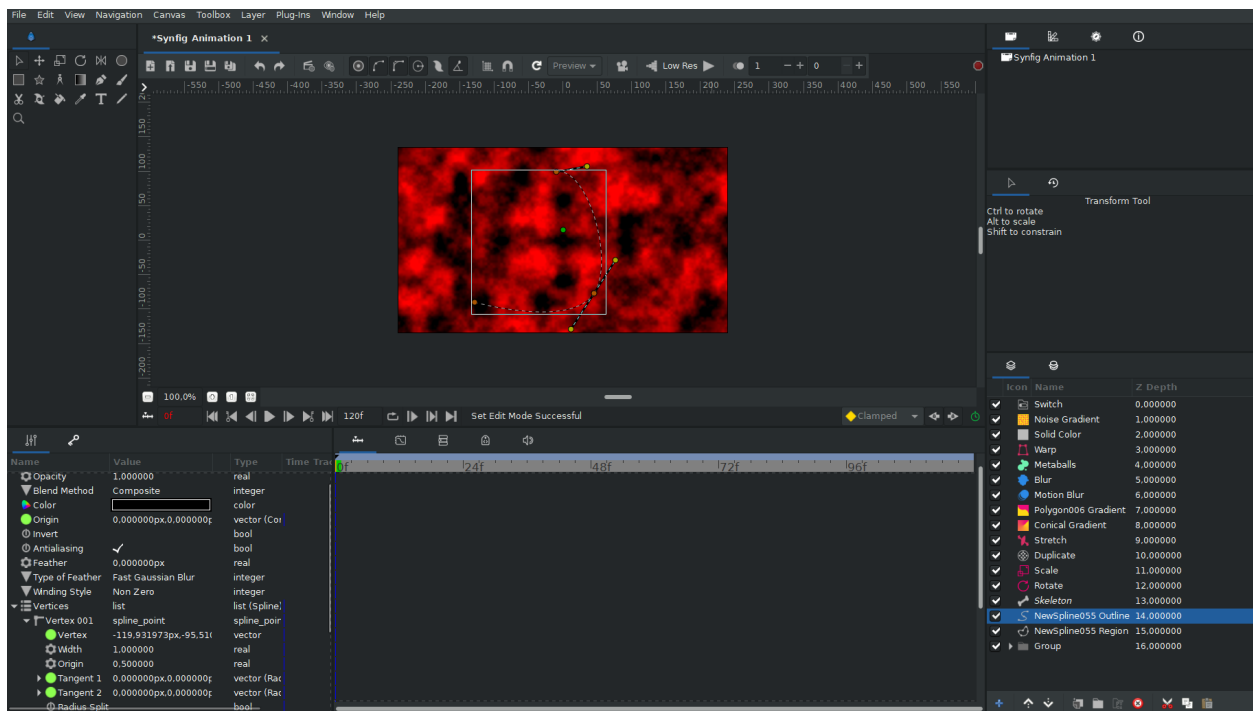


Added possibility to stop rendering process.



Icon themes

Now it is possible to change icon theme for Synfig Studio. Icon set (theme) can be placed in a folder inside “share/synfig/icons/” path (eg. share/synfig/icons/new-theme-folder) and then activated by setting the SYNFIG_ICON_THEME environment variable (eg. SYNFIG_ICON_THEME=new-theme-folder). See [PR #1483](#) for details.



Please notice that this is just a first step in ongoing work on implementing theming support for Synfig Studio. You can check [this discussion](#) for details.

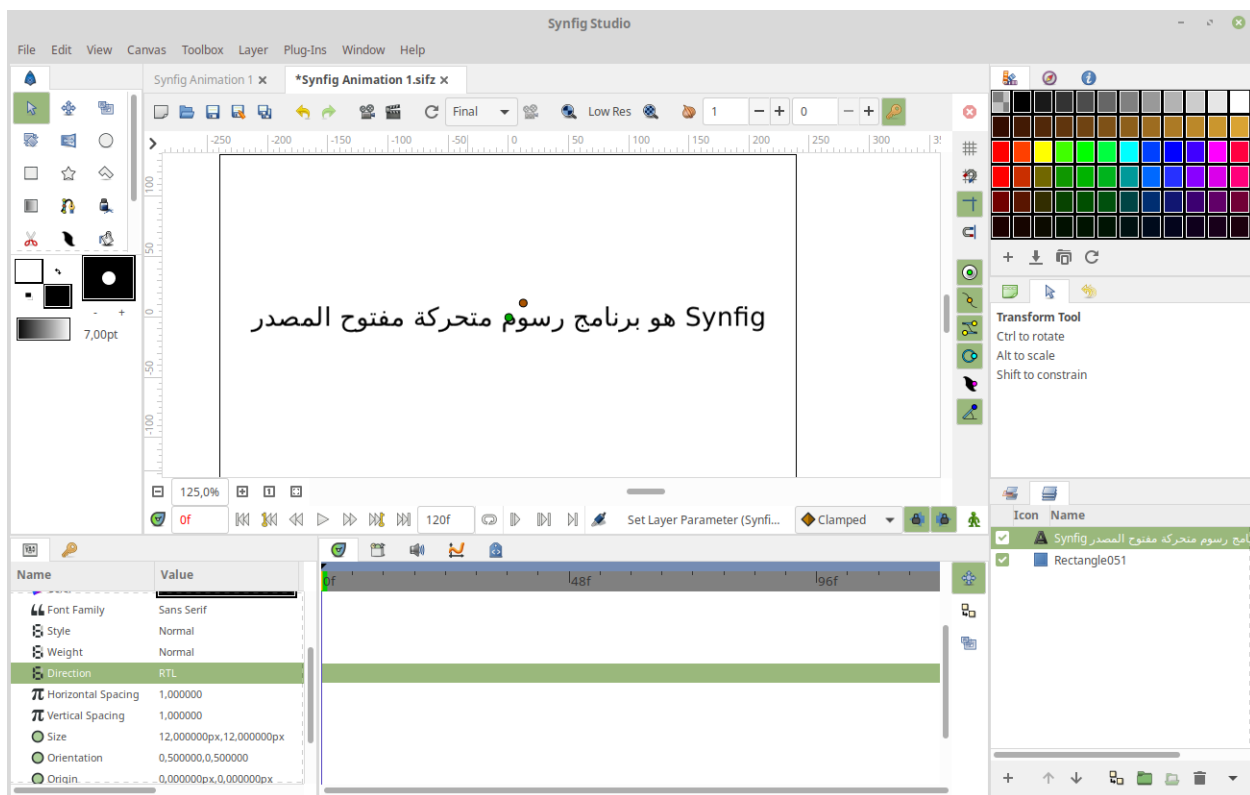
CMake build system

This version introduces the possibility to build Synfig using CMake. Although this build method is not yet suitable for creating packages, it greatly simplifies the build process for those who wish to compile Synfig for development or testing purposes. Please check [this documentation page](#) for instructions.

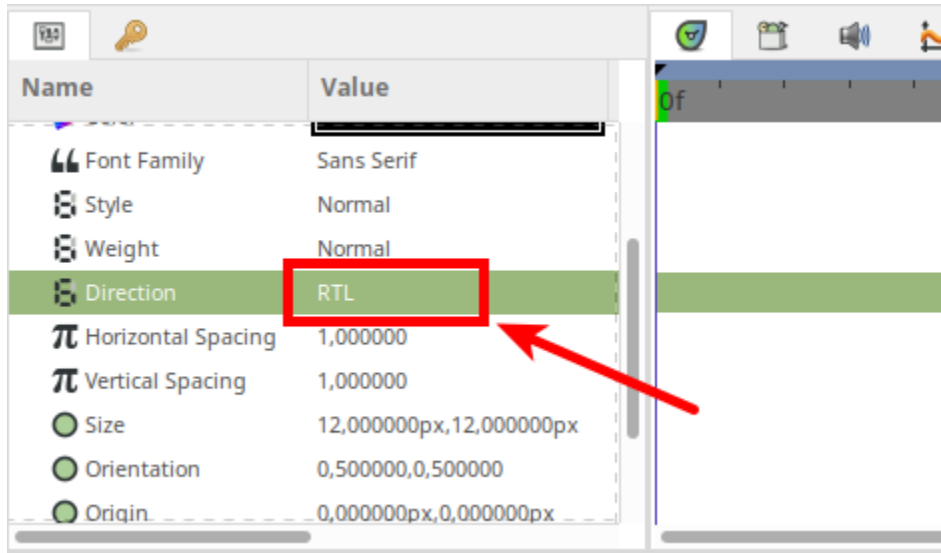
16.1.5 Synfig 1.5.1 Release Notes

This version features major improvements for Text Layer, fixes for Skeleton Tool, better onion-skin, as well as other important fixes and usability improvements. Please check the full list below.

Rewritten Text Layer

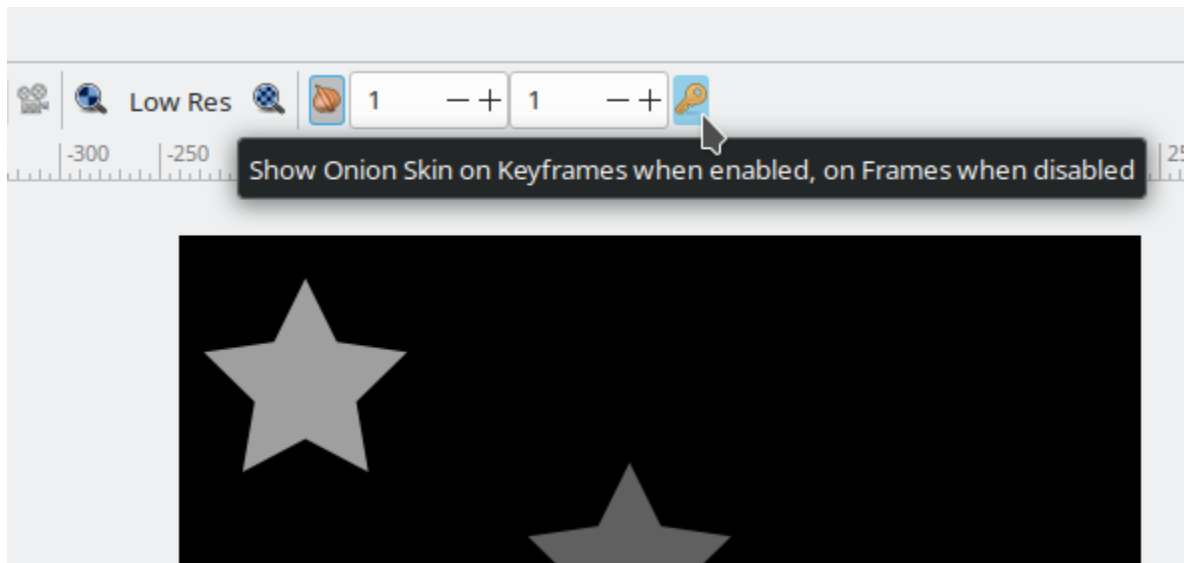


- Text Layer fully rewritten and optimized, which results in much faster rendering (#2362).
- Text Layer now have proper support for RTL languages (i.e. Arabic) (#2232). Please notice that if you use a string which mixes RTL and LTR text, then you might need to set a "Direction" parameter in Text Layer to indicate the proper order.



- Text Layers now visible when workarea displays its content in Draft mode (#631).
- Fixed issue with “jumping” letters in animated text (#389).
- Fixed issue with quality regression under Text Layer (#831).
- Fixed glitched rendering output when Text Layer is combined with Perspective Layer (#1566).
- Allow to select font from drop-down menu in Tool Options (#2355).

Improvements



- New option for Onion Skin allows to choose between Keyframes and Frames (#2166).
- The “Keep aspect ratio for width and height” option is now enabled by default in Render Settings (#2351).
- When the user imports the file the imported layer becomes selected (#2308).
- Now it is possible to create a new vertex on a Spline with double-click (#2312).

Fixes for Skeleton Tool

- Fix active bone selection for Skeleton Deformation Layer (#2340, #2365).
- Fix crash when moving origin point of child bone in Skeleton Deformation Layer using Bone Tool (#2049).
- Fix crash when undoing Child Bone until first one (#2341).
- Fix loading of default bone width preference value (#2343).

Fixes

- TimeTrack Panel: Fix missing context menu for waypoints at Canvas Parameter (#2357).
- TimeTrack Panel: Update time track area when keyframe is moved (#2289).
- Sound Panel: Fix loading of some audio files (#2301).
- Fixed incorrect framerate when importing image sequence (#2219).
- Fixed Link To Spline action (#2344).
- Fix workarea context menu action not affecting all layers (#2352).
- Fix crash when list of vertices is converted into Switch (#2354).
- Fix dialog responses for user pressing Esc key (#2309) (#2329).
- Don't let inner layer be duplicated twice when copying (#2317).
- Fix image size when importing onto canvas with non-standard size (#2252).
- Fix magick++ target missing in AppImage (issue #2320).

Other changes

- Circle Tool: Removed deprecated falloff option (#2366).
- Rectangle Layer: Bevel parameter now defined by relative value, not distance (#2350).
- Skeleton Deformation Layer: Better tooltip/description for “Bones” Parameter (#2332).
- Advanced Outline Layer: Improve hints for Dash Item Parameters (#2314).
- Added support for building with MLT++-7 (Autotools) (#2306).

16.2 Stable Versions

16.2.1 Synfig 1.4.0 Release Notes

Synfig Studio 1.4.0 is a major stable release, which is introducing many new features and improvements. Please read below for details.

Animation Tools

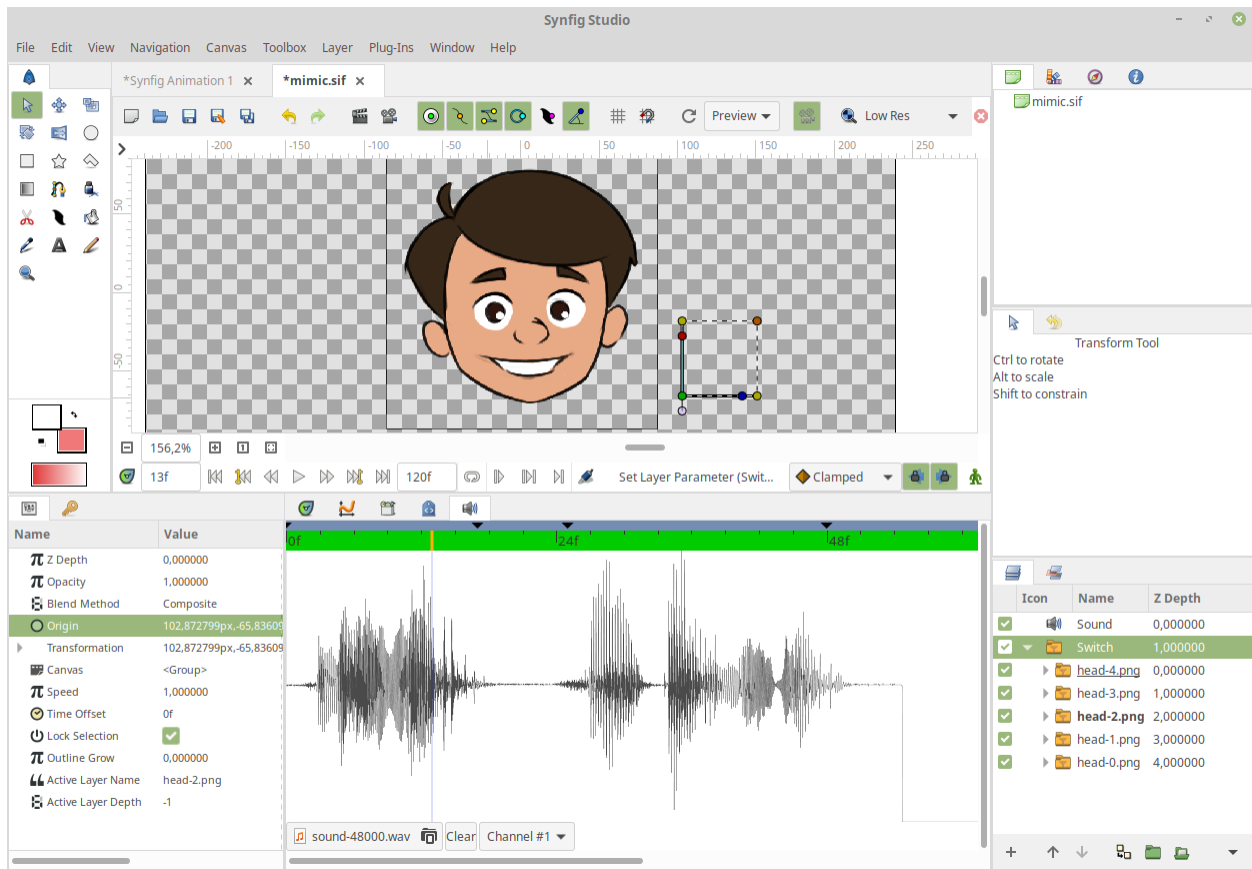
Editable curves in Graphs Panel

Now curves in Graphs Panel have control points, and you can drag to change curve shape. Multiple selection for waypoints is also available. You can also use arrow keys on keyboard to move points. Double-click on a curve creates a new waypoint, Ctrl+A/Ctrl+D shortcuts are selecting/deselecting all points. Much thanks to Rodolfo Ribeiro Gomes for implementing this!

Sound synchronization features

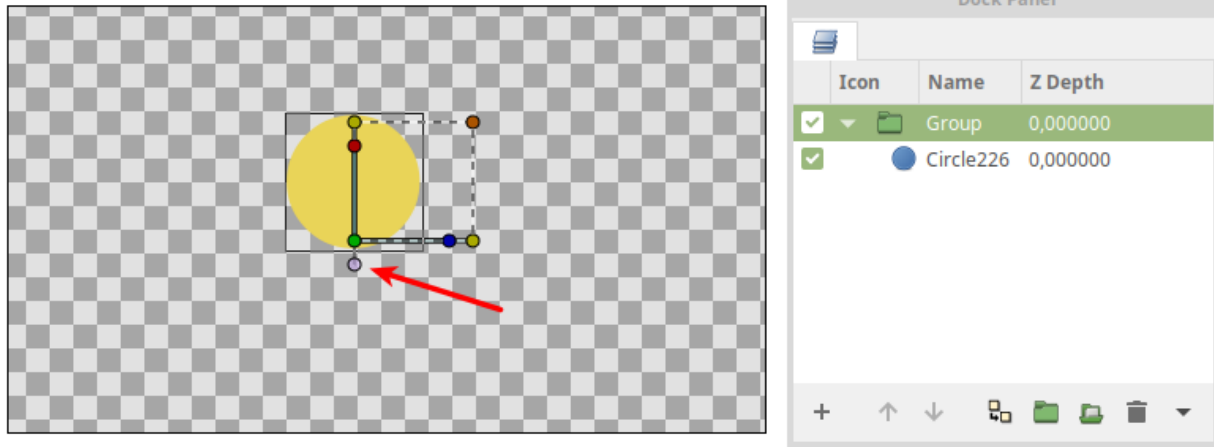
We did a major cleanup of code responsible for sound playback, fixed non-working volume parameter and eliminated synchronization issues.

In addition to that, Rodolfo Ribeiro Gomes implemented a special panel to display waveform. Now it's so easy to sync your animation with soundtrack!



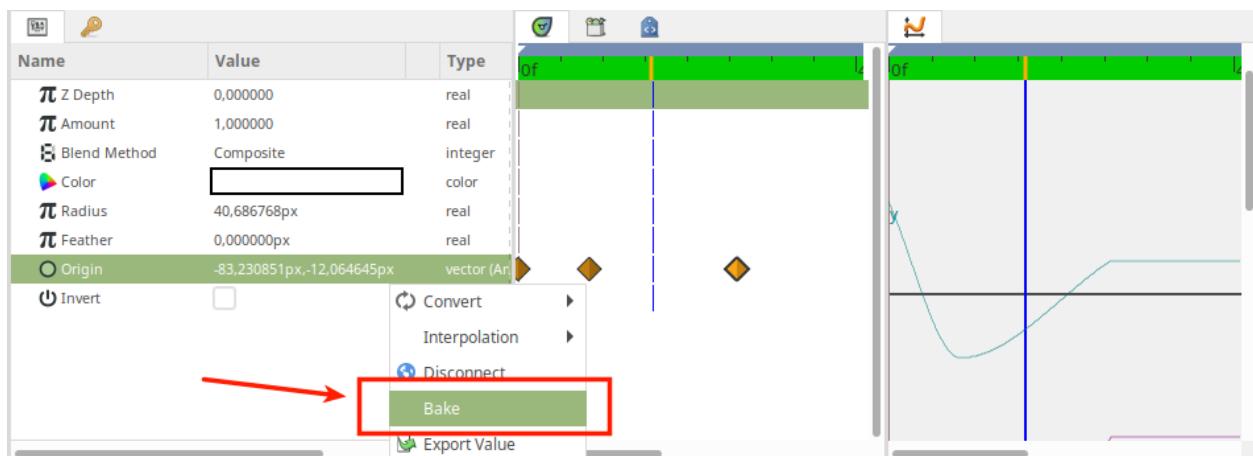
Control point to change Transformation Origin

Now transformation widget has a special control point to easily change origin of transformation.

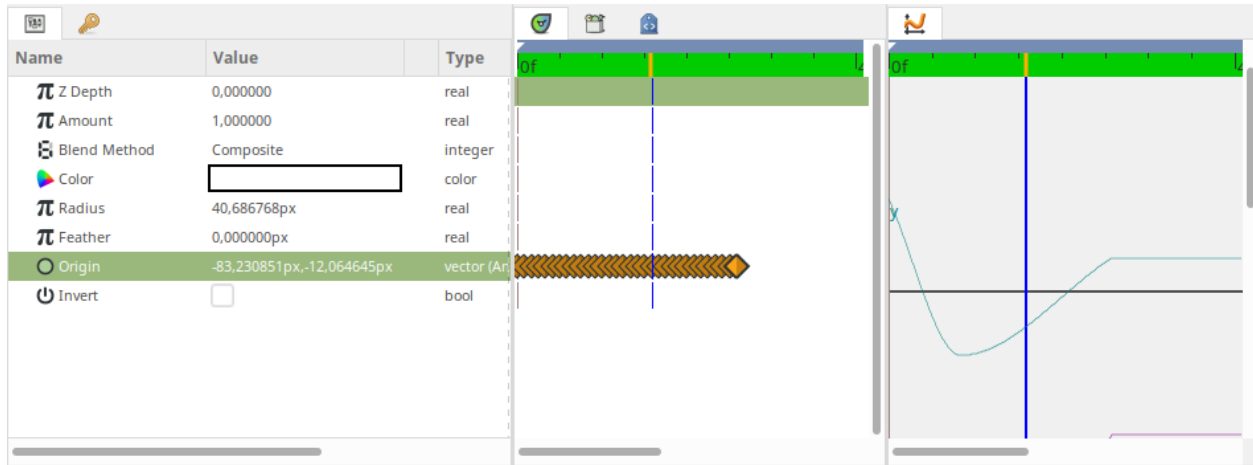


Animation Baking

Now you can “bake” animation of any parameter! This operation “burns” any interpolated curve as a sequence of waypoints.



Result:



Improved TimeTrack

TimeTrack Panel got several improvements. Now you can define playback range and make playback looped. Hovering mouse cursor over TimeTrack now shows frame preview.

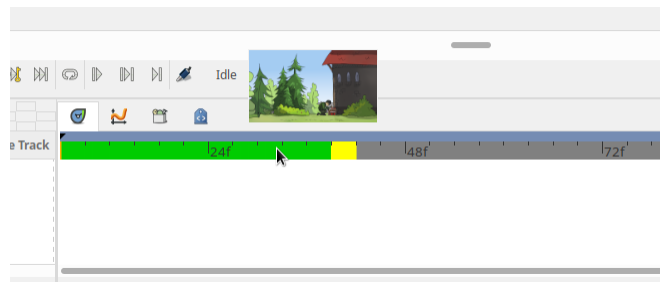
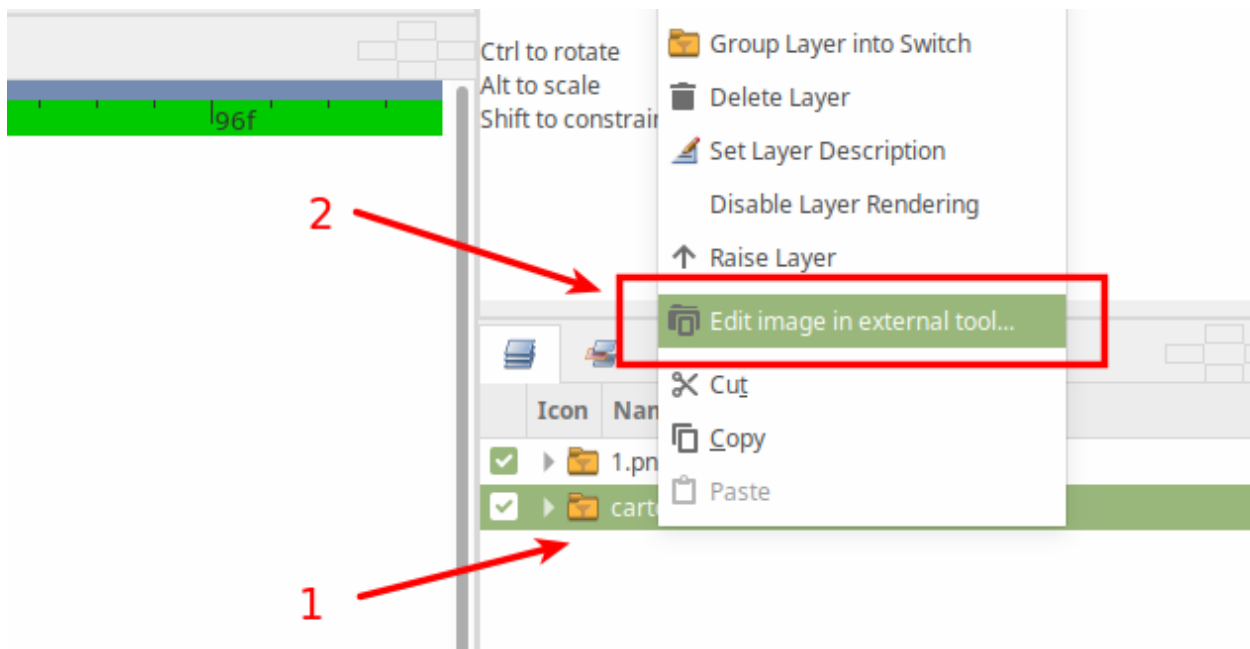
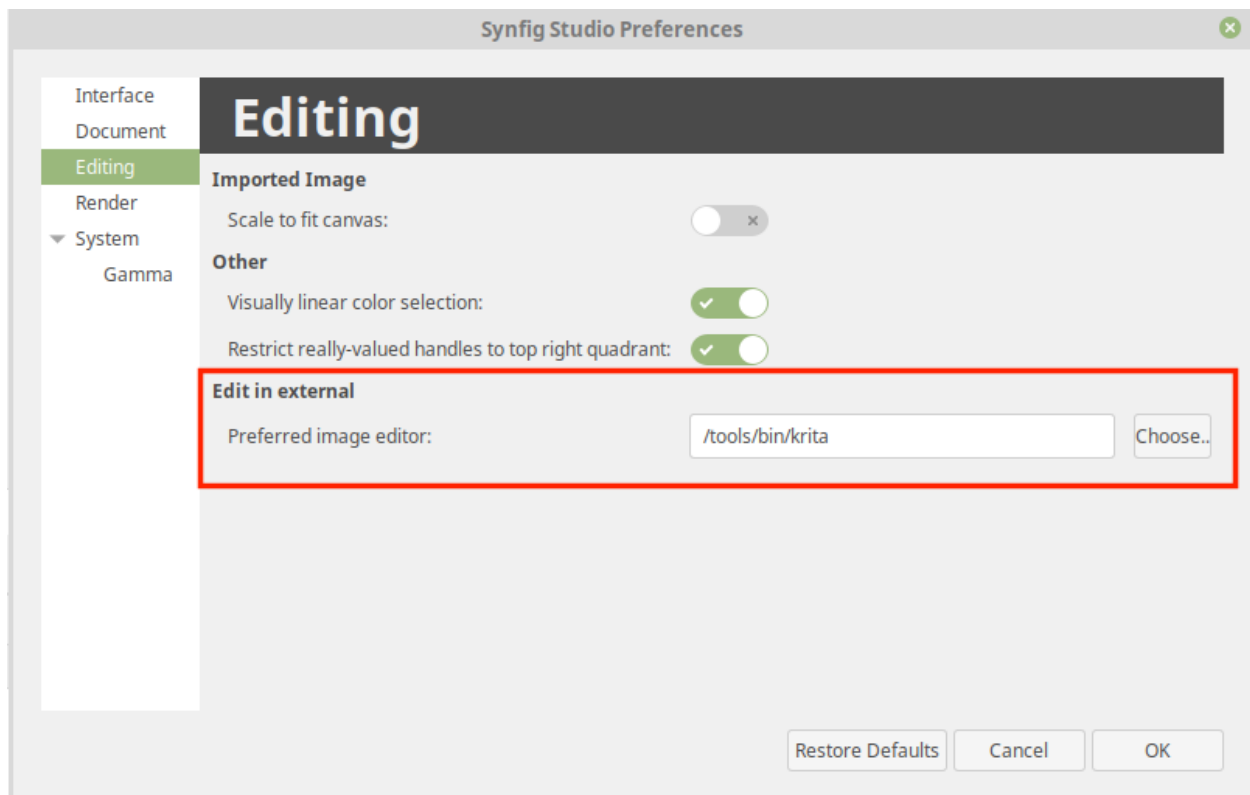


Fig. 1: Frame thumbnail shown when cursor hovers timeline

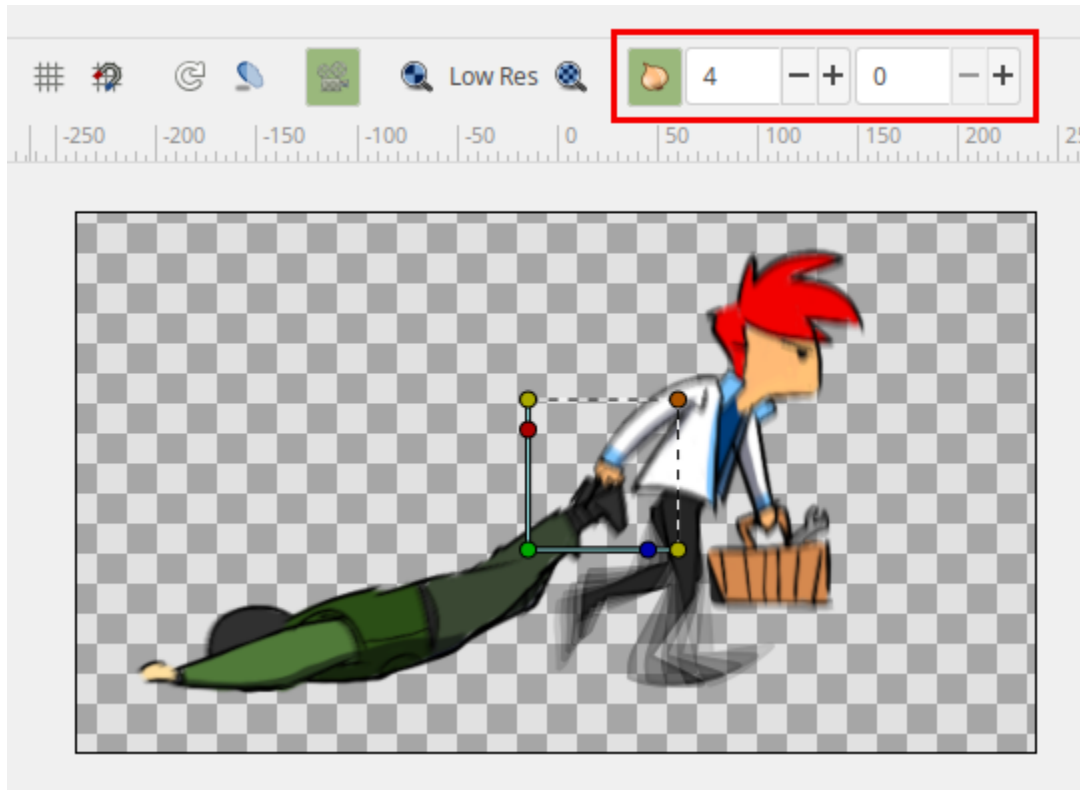
Open images in external tool

Ankit Kumar Dwivedi implemented a possibility to open images in external editor. Right now it is possible to configure preferred image editor via relevant option in Preferences. After that you can right-click any Image Layer and choose “Edit image in external tool...”.



Fixes and changes for onion-skin functionality

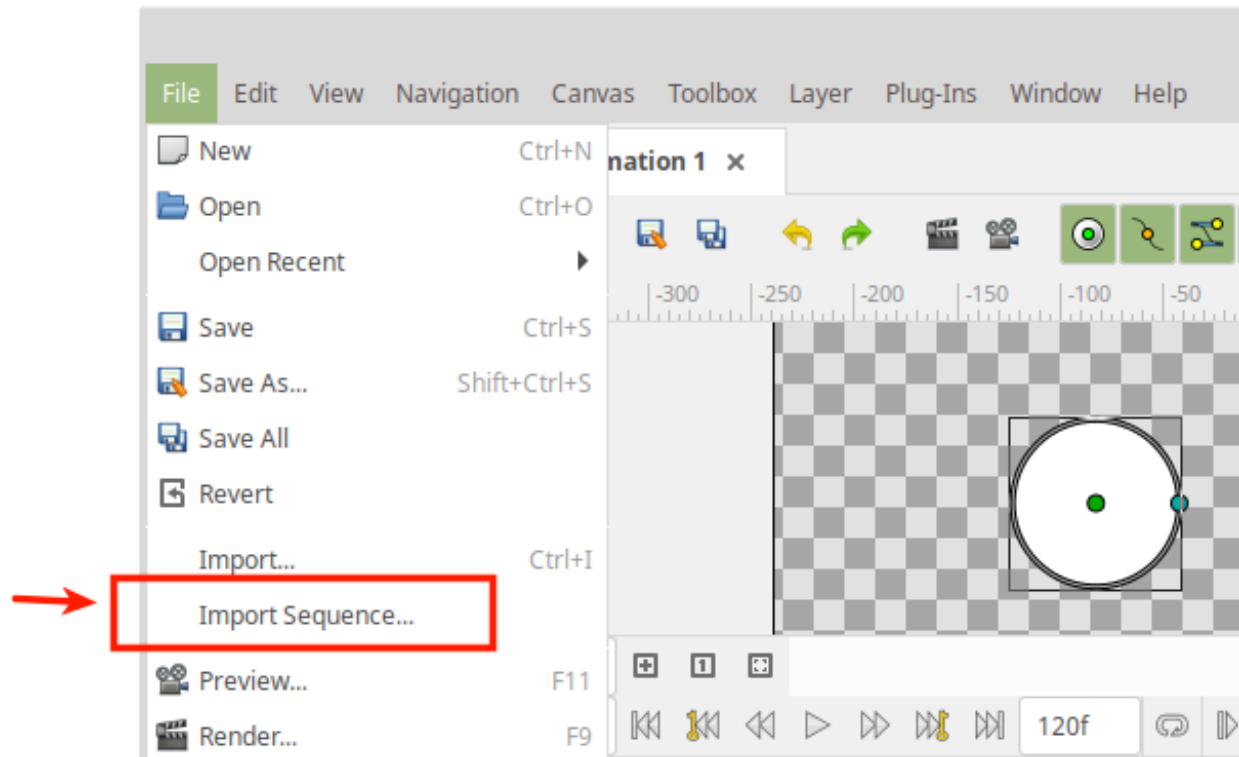
Onion-skin functionality was mainly broken since version 1.2.0 - it wasn't able to display more than one onion-skin layer. Now this ability is restored. At the same time, we have changed functionality of onion-skin: in all previous versions it was linked to keyframes only, now it is set relatively to ordinary frames. We plan to continue improving onion-skin features as part of [issue #600](#).



Importing

Simple import of image sequences

Now it is easy to import image sequences - just choose “File” -> “Import sequence...” from menu.



Vectorization of bitmap images

Synfig Studio now allows to convert raster (bitmap) image into vector image constructed from outlines. The vectorization algorithm is ported from OpenToonz animation software by Ankit Kumar Dwivedi, as part of Google Summer of Code 2019 program.

[Read more about this feature...](#)

Basic video import functionality

This version introduces basic functionality for importing video files. At the current state the system is still too slow to work with large AVI and MP4 videos, but it is already possible to play with small animated GIFs (yes, it can import GIF as well).

Rendering and Export

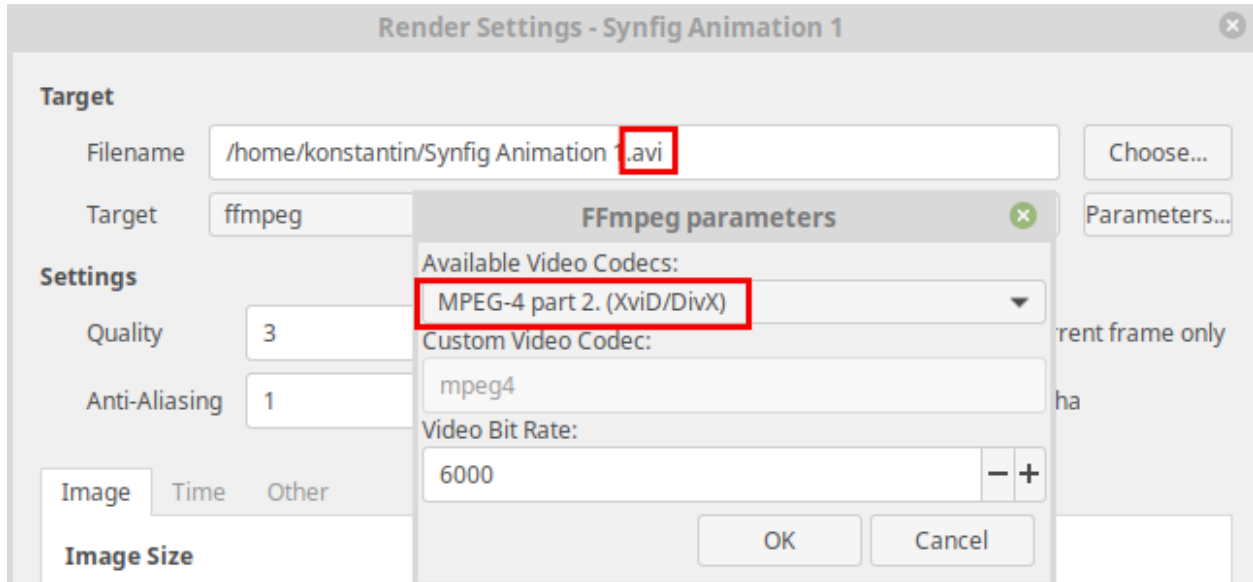
Export animation for web (Lottie)

Synfig Studio can export animations to [Lottie JSON format](#), which allows to embed them into web pages, play them natively in web browser and on mobile platforms. This format is also [used](#) for creating Telegram's animated stickers! This feature is implemented by Anish Gulati as part of Google Summer of Code 2019 program.

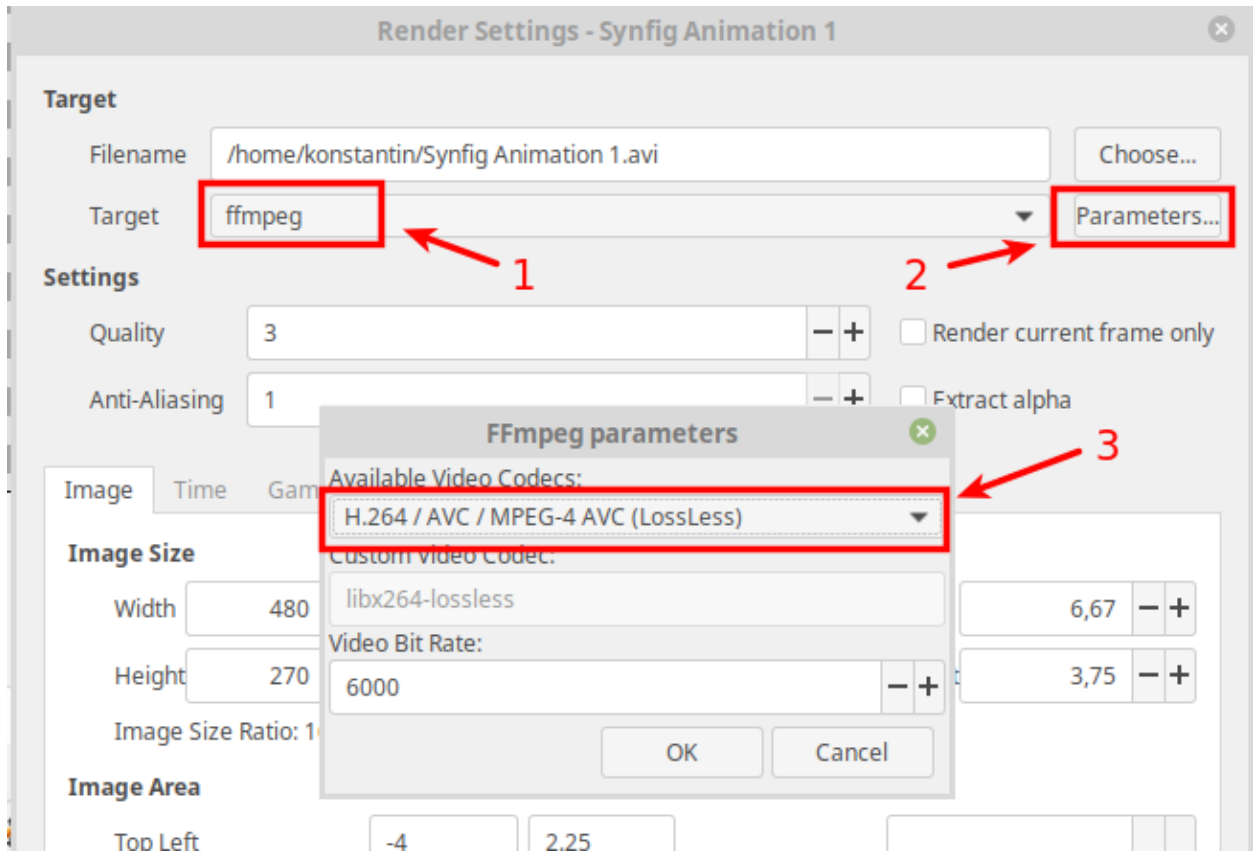
[Read more about this feature...](#)

Default render settings

Default rendering settings now configured to use AVI format with popular mpeg4 codec by default, which resolves issues with playback on most systems (issue #464).



Still, if you wish to achieve best quality, consider to choose “H.264 (Lossless)” codec option.



Other notable changes regarding rendering process:

- Changing rendering target now automatically changes extension of output file.
- Rendering status is now indicated with a progress bar at Info Panel (issues #383, #626).
- A sound notification is played when rendering is done (issue #648).

Render engine improvements

Render engine got some major optimizations, which greatly reduces time of handling scenes with many layers, comparing to version 1.2.2 (see [issue #314](#)).

In this version following layers got special optimizations: Outline Layer (completely rewritten), Region Layer, Checkerboard Layer, SuperSample Layer.

Also, render engine now has downsampling algorithm for bitmaps, which greatly improves quality resulting image.

Fig. 2: Rendering result in Synfig 1.2.2

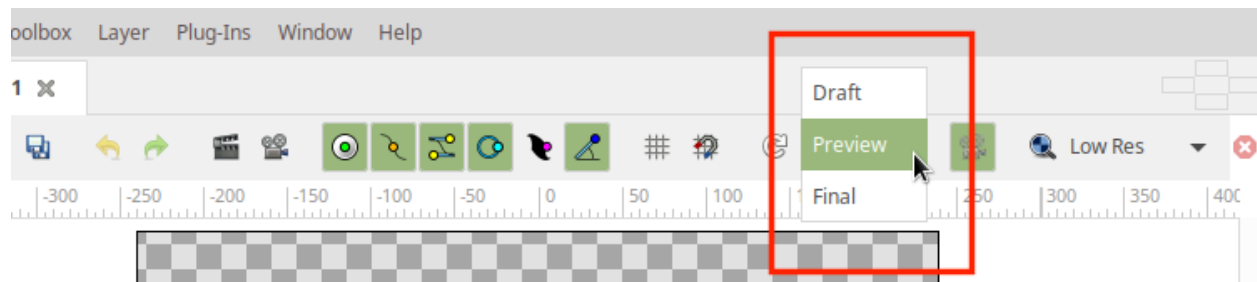
Fig. 3: Rendering result in Synfig 1.4.0

“Preview” rendering mode for workarea

Downsampling algorithm for images (mentioned above) greatly improves quality if your scene uses big images scaled to smaller size. Unfortunately, this operation demands additional CPU resources and you will notice general application slowdown when working with such scenes.

There is an option to enable Draft rendering mode (it disables downsampling), but this solution doesn't fits for all cases, as it is hides many layers, which could be important for you.

So, we added one more display mode for workarea - “Preview” - which displays all possible layers, but with some simplifications - i.e. no downsampling, approximations for blur algorithms and some other simplifications. “Preview” mode is giving a fair compromise between speed and display quality and is used by default in GUI. For more details please check [this discussion](#).



Workarea Cache and Background Rendering

Synfig now remembers already rendered frames, which is greatly improves experience of playback and timeline scrubbing.

Also, now it is possible to have background rendering of all frames in current animation. This allows to achieve very smooth playback and timeline scrub. You can enable background rendering using a relevant button on toolbar (see second video below).

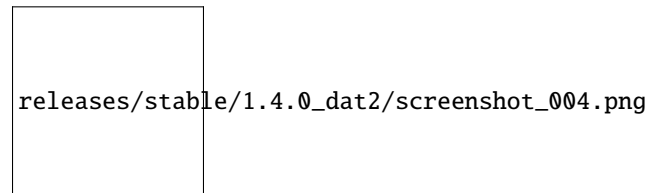
Gamma handling

Since the very beginning of its development, Synfig was handling all color calculations with Gamma value set to 2.2. As result, there were a lot of issues with color interpolation and blending, like [this one](#).

We decided, that it is good moment to get rid of this behavior and changed how Gamma handled in Synfig. The tricky part here was to keep compatibility with old files. Here's what we did:

- Gamma is now defined on per-document basis (in previous versions it was an application-wide setting). So, now every document has its own gamma value defined. You can see/change Gamma value by selecting "Canvas" -> "Properties" from menu and switching to "Gamma correction" tab.
- For all newly created documents Gamma is set to 1.0.
- Old documents (from previous versions) are opened with Gamma set to 2.2, so you won't notice changes (although you will still notice same artifacts from old version, like the one noted [here](#)). You can manually set Gamma to 1.0, to get rid of those issues, but this will change the look colors would look in your animation.

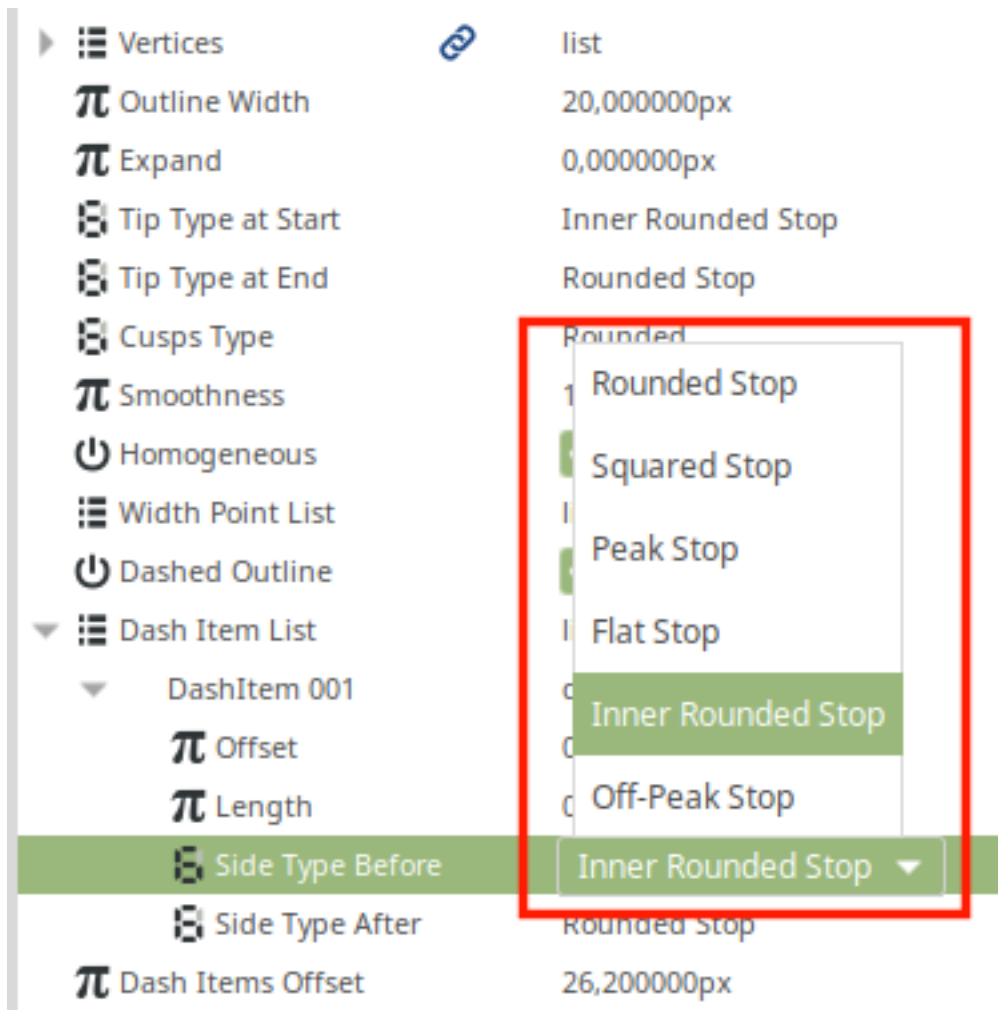
So, with this change users should not notice any change when working with their old documents. Still you can encounter an issue, when copying layers from old files to new (and vice versa) - the colors of copied layers appear different in destination document. This can be easily fixed manually as [described here](#).



Layers

Advanced Outline Layer

Advanced Outline Layer was completely rewritten and optimized in this version. Also, it is now capable to display two new types of tips - "Off-Peak Stop" and "Inner Rounded Stop", that can be used to define outline start/end for dashes.



With this addition you can easily do effects like shown here ([source file](#)):

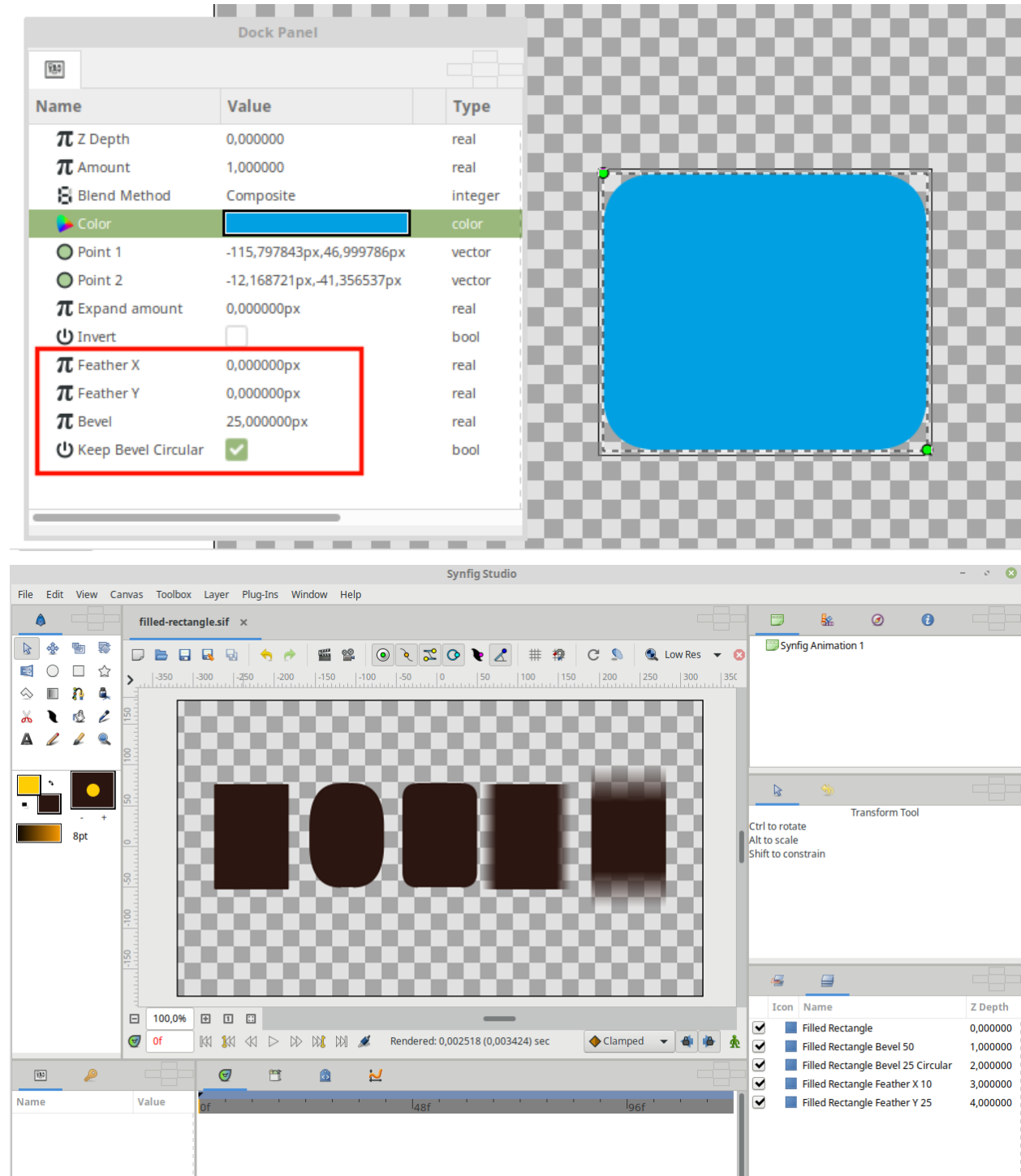
Switch Layer

Now it is possible to select visible layer by index in Switch Layer. This is done using a new parameter - “Active Layer Depth”. The parameter takes effect when “Active Layer name” parameter is empty.



Rectangle Layer

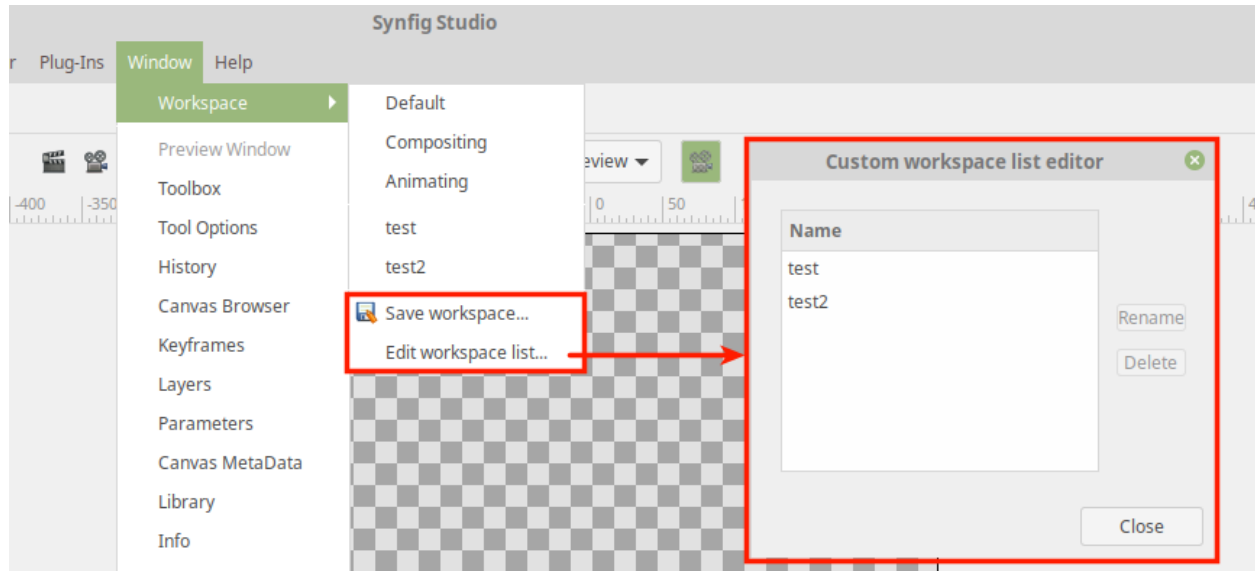
Rectangle Layer now capable to have two types of rounded corners and horizontal/vertical feather. Users with a deep knowledge of Synfig know, that those features were available in Filled Rectangle Layer (not the same as Rectangle Layer), which was a part of Example layers. Now all those features are merged into standard Rectangle Layer, and the Filled Rectangle Layer is removed. See GitHub's [issue #745](#) for details.



Settings & Customization

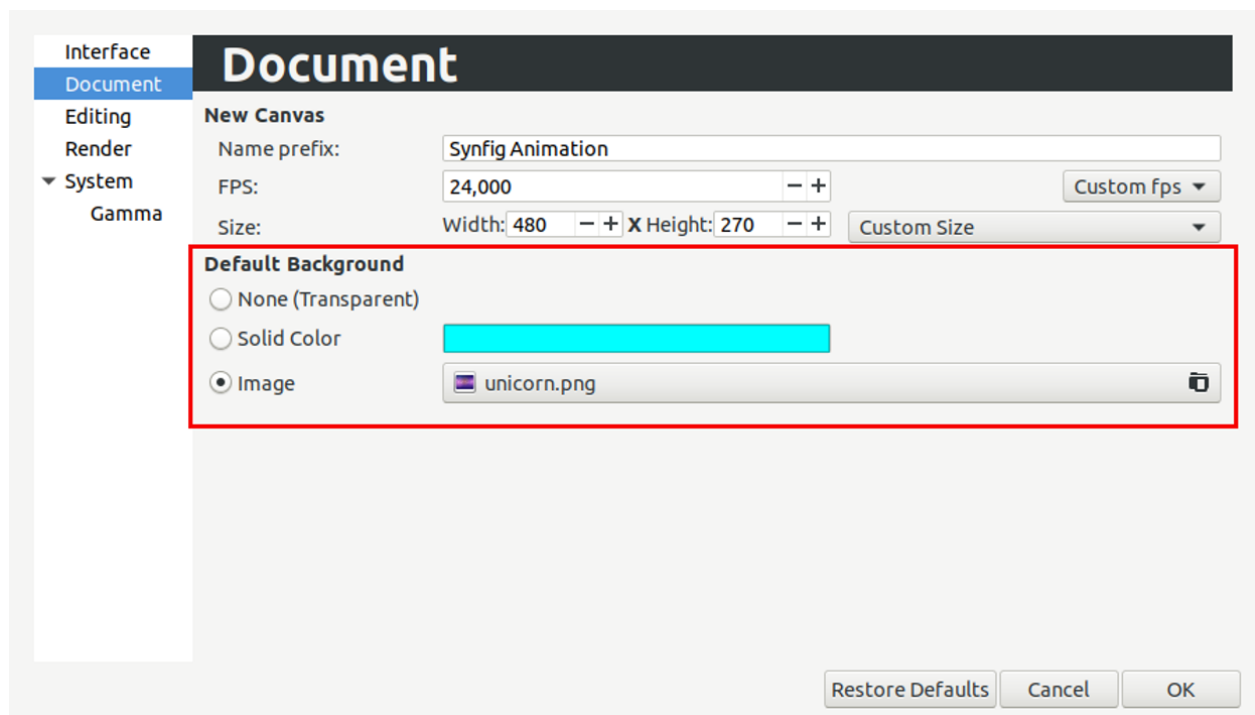
Saving workspace layouts

Now it is possible to save your workspace layout - just choose “Workspace” - “Save workspace” from menu. Also you can edit list of custom workspaces by calling “Workspace” - “Edit workspace list...”.



Default background for new files

Now it is possible to set default background for newly created documents ([issue #636](#)). The background can be a solid color or image. You can configure this via “Edit” -> “Preferences” -> “Document”.



Other notable improvements

- Improved process of creating Splines and Polygons: now it is possible to finish shape creation with double-click or by hitting Enter. Also you can click first vertex of Spline to loop and finish shape creation.
- Now it is possible to remove Layers by hitting Delete key.
- Tab placement crosses now hidden by default and revealed only when user starts dragging some panel. See [issue #238](#) for details.
- Now it is possible to change animation length directly from workarea (see [issue #684](#)).
- Double-clicking waypoint in Timetrack panel shows Waypoint Properties dialog.
- Now it is possible to configure color of transparent areas in Preview Window ([issue #636](#)).
- Workarea now has a widget to change length of timeline.
- Brush Tool is now disabled by default. User can get it back by enabling “Experimental features” option in Preference.
- Home/End keys now set time cursor to start/end of timeline.

Minor improvements

- DashItem Offset and Length parameters of Advanced Outline now use consistent units when editing them ([issue #1265](#)).
- Do not waste space for dropdowns in combo boxes ([issue #650](#)).
- Show interpolation type directly in waypoint’s context menu (see [details](#)).
- Disabled scientific notation when displaying numbers ([issue #635](#)).
- When new Text Layer is created, its name is set the same as text content ([issue #407](#)).
- Vertex handles now placed above tangent handles ([issue #645](#)).
- Color dialog now opens with HSV tab active ([issue #672](#)).
- Portable version of Synfig (zip) now writes settings to its own configuration directory ([issue #716](#)).
- “Amount” parameter renamed to “Opacity”.
- Playback is stopped when user clicks on timeline ([issue #415](#)).
- Removed unused Change Preview Quality menu item.
- Activate widget_filename right after user chooses file via button/file dialog ([issue #1425](#)).
- Homogenous parameter of Advanced Outline is enabled and static by default.
- Set Animation Speed parameter of Noise Distort Layer as static by default.

Bugfixes

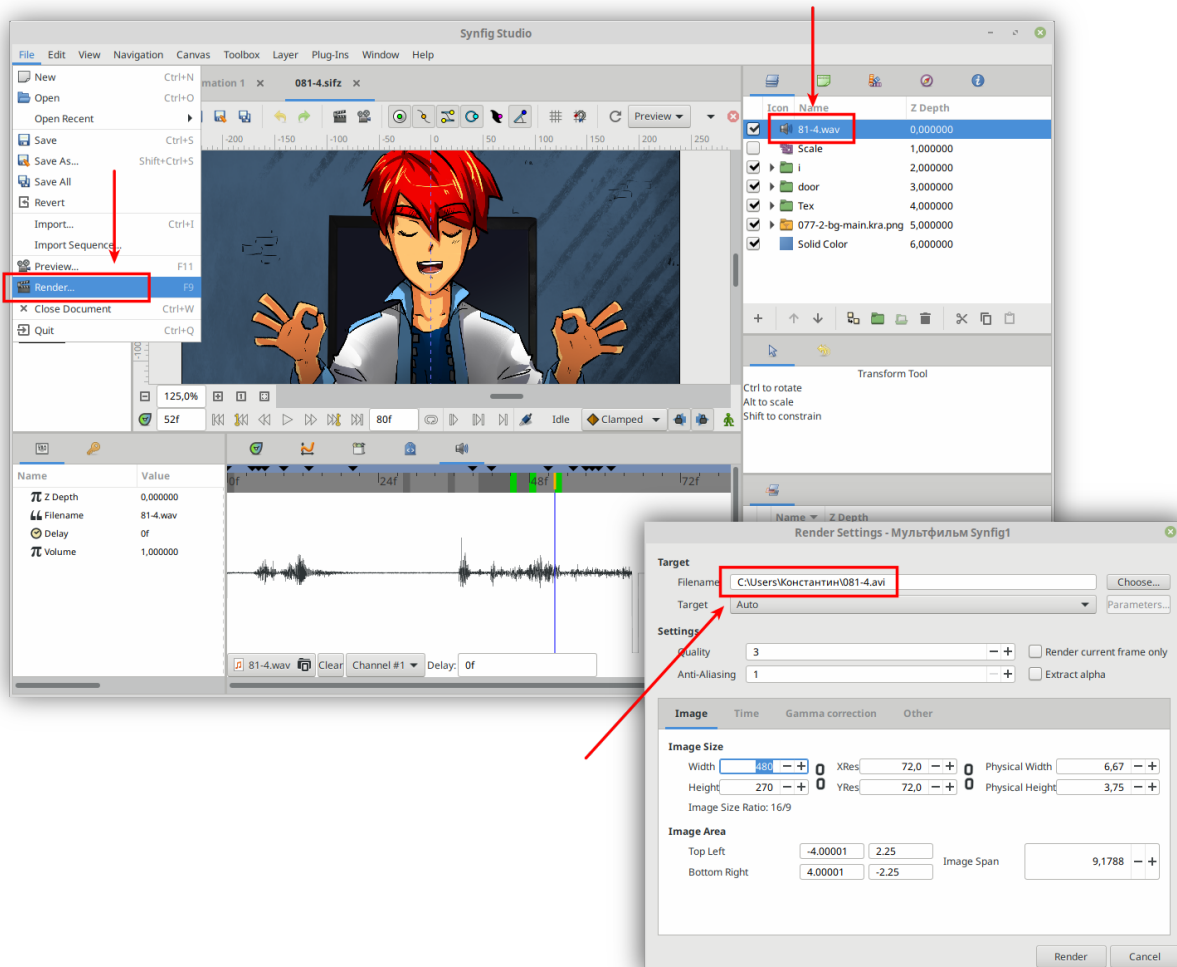
- Fixed issue when copying skeletons from one file to another. Previously this was resulting in corrupted file, but now it isn't.
- Fixed a very old and annoying [issue](#) with Ctrl+C/Ctrl+V shortcuts behavior, which was making impossible to use them for text entries in UI (layers were copy-pasted instead).
- Fixed behavior “Local Time” parameter of Time Loop layer ([issue #479](#)).
- Fixed issues opening and exporting of files with multibyte (i.e. Arabic) symbols in filename.
- Fixed hang when opening a second .sif file from Explorer on Windows ([issue #291](#)).
- Fixed “No disc in drive” error ([issues #489, #724](#)).
- Autosave now enabled by default ([issue #657](#)).
- Removed “Cancelled by user” dialog that happens when the user tries to move an animated object ([issue #693](#)).
- Fixed incorrect behavior of Draft rendering mode in some circumstances ([issue 582](#)).
- Timebar is not hidden anymore for cases when scene is only 1 frame short.
- Fixed issue with incorrect GUI font on some Windows systems ([issue #667](#)).
- Fixed issues with incorrect window placement on Windows ([issue #523](#)).
- Fixed issue which was preventing to change parameters at some circumstances ([issues #659, #526, #520](#)).
- Fixed crash when exported value is set from parameters panel in animation mode ([issue #588](#)).
- Fixed crash when user cancels parameter changing ([issue #671](#)).
- Fixed wrong behavior of “Lock Ratio” button in Canvas properties and Rendering window ([issue #771](#)).
- Fixed issues with Canvas not preserving Length and Resolution after exporting ([issues #715 and #874](#)).
- Fixed issue with Linear interpolation for Integer parameters ([issue #828](#)).
- Fixed bug with empty floating windows appearing after workspace switching ([issue #1143](#)).
- Fixed crash when removing a Spline vertex using “Remove item (Smart)” ([issue #1102](#)).
- Fixed popup menu disappearing right after button release for Widget Gradient/Spline ([issue #1274](#)).
- Fixed crash when exporting with “pngspritesheet” target ([issue #356](#)).
- Fixed crash when unexporting value node ([issue #231](#)).
- Fixed renaming layer set affecting other names ([issue #1146](#)).

16.2.2 Synfig 1.4.1 Release Notes

Synfig Studio 1.4.1 delivers some improvements and critical bugfixes. Please read below for details.

Features

- Synfig now can export videos with sound (#1623, #1830). In previous versions Synfig was ignoring all Sound Layers in animation file when doing export to video format. Now all sounds are mixed and added into video file.



Windows-specific fixes

- Fixed Synfig window not showing after closing it in minimized state (#1973).
- Fixed issue of rendering to file when using non-Latin filenames (#2011, #1940).
- Fixed incorrect import of lsf files when path contains non-Latin characters (#1924).
- Fixed non-working command “Edit image editing in external tool...” (#1912).
- Fixed problem with non-working plugins when loading them from path containing non-Latin characters (#1882).
- Fixed saving/loading palette to file with non-Latin name (#1889).

MacOS-specific fixes

- Fixed issue with Synfig not starting if SynfigStudio.app was renamed (#1834).

Bugfixes

- Fix Duplicate Layer ignoring all layers but the first one (#1829).
- Fixed crash when frame fails to render (#1861).
- Fix loading of Average Converter type (#1955).
- Fixed interpolation for parameters of Boolean type (#1826).
- Fixed crash when user attempts to drag-and-drop layer while renaming (issue #540).
- Do not show “Link to Spline” menu item when a vertex of spline itself is selected (#1988).
- Fixed crash when right-clicking on group in the Sets panel (#1920).
- Fixed crash in Spline Tangent converter (#1836).
- Fixed crash in Spline Vertex converter (#1802).
- Fixed crash on unsuccessful action with a clean undo history (#1643).
- Fixed issue with duplicating Duplicate Layer (#1971).
- Avoid possible random crashes related with non-unique node identifiers (#2029).

UI/UX improvements

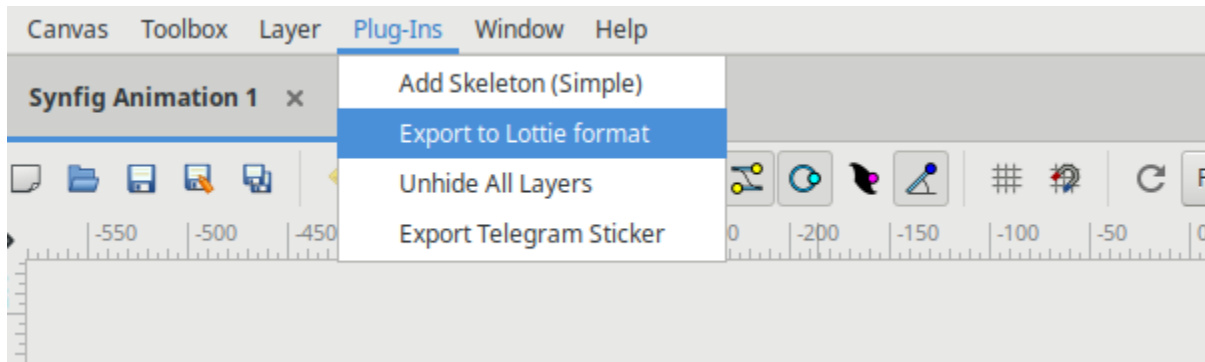
- Allow to use Ctrl+C/Ctrl+V/Ctrl+A shortcuts in Color and Gradient Dialogs (#1821).
- Fix layer duplication action for cases when group selected together with its children items (#2050).
- Waypoint Editor dialog now shows TCB options only if current point has TCB type (#1182).
- Change order of Waypoint Editor buttons to match other dialogs (#1880).
- Waypoint Editor now properly handles parameters of Enumerated type (issue #1965).
- Fix issue with closing some configuration dialogs: changes were applied anyway, as if “OK” button was pressed (#1573).
- The “View” -> “Toolbar” menu item now active and usable (#1386).

16.2.3 Synfig 1.4.2 Release Notes

Synfig Studio 1.4.2 delivers some improvements and critical bugfixes. Please read below for details.

Improved export for Web

The **Lottie** exporter plugin has gained new capabilities and includes all improvements made during Google Summer of Code 2020.



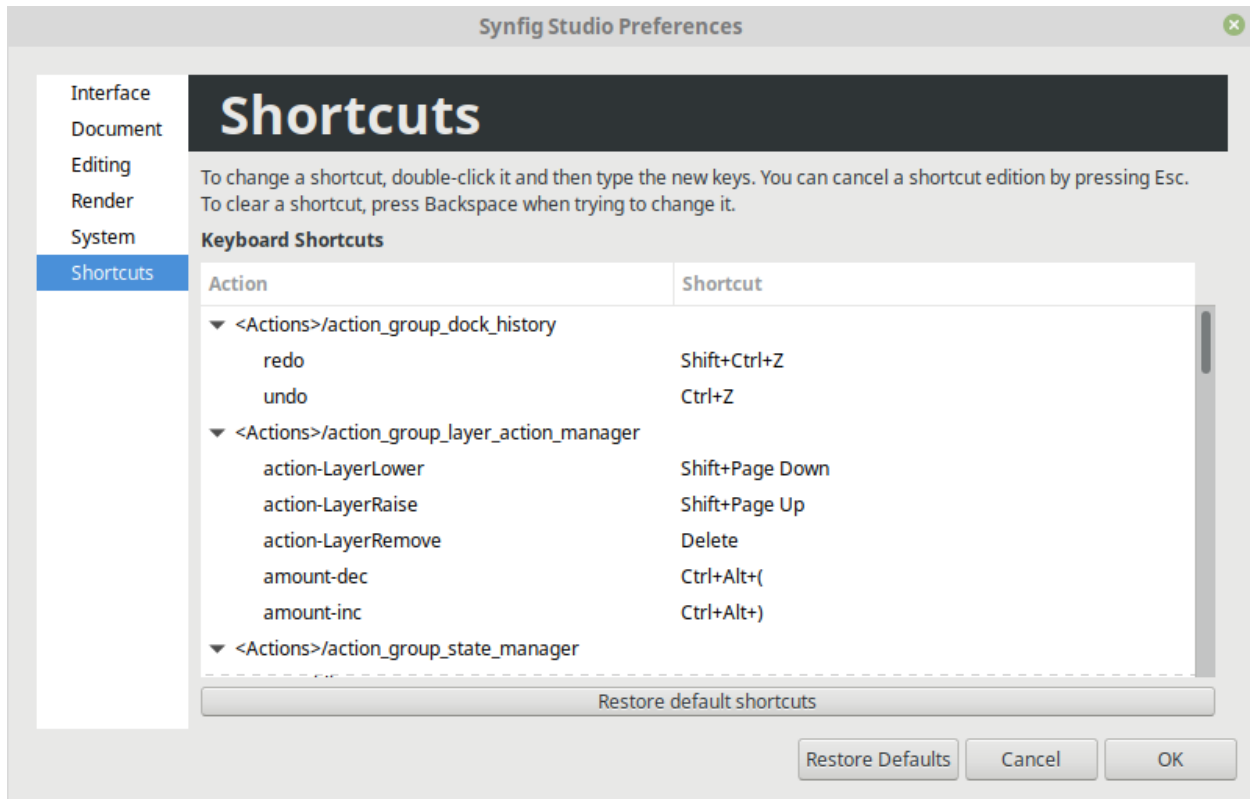
Here is what's new:

- The plugin now capable to export Advanced Outline layers (#2199) and Blur layers (#1505).
- Added support for converters: “Sine”, “Cos”, “Integer”, “Linear gradient”, “aTan2”, “Vector angle”, “Radial composite”, “Radial gradient”, “Vector X”, “Vector Y”, “Power”, “Dot product”, “Logarithm”, “Reciprocal”, “Range”, “Vector length”, “Not” (#1236 #1254 #1286 #1363 #1409 #1447 #1446 #1516 #1515 #1593 #1586 #1619 #1606 #1650 #1587).
- Added support for variable gamma correction, fixed issue with wrong colors when exporting files (#1174).
- Fix converter methods not working with radius of simple circle layer (#1101).
- Improved support for exported values (#2182).
- Use non-minified *bodymovin.js* script to avoid code obfuscation (#2202).

For more information about Lottie Exporter please check [this page](#).

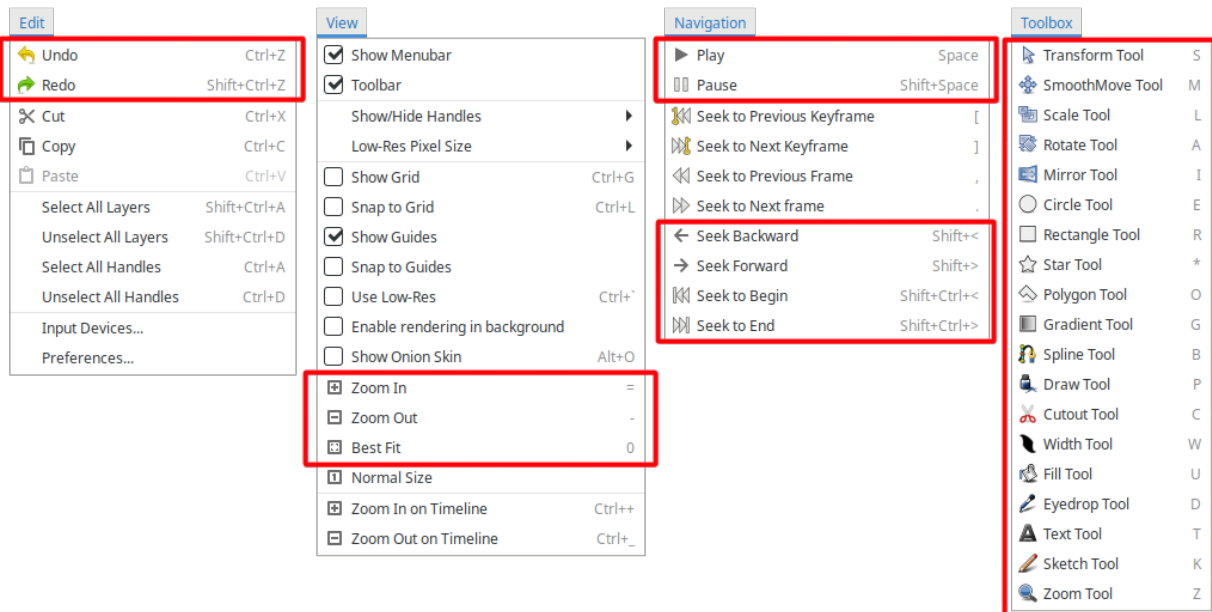
Better handling of shortcuts

This version got brand new Keyboard Shortcut Editor dialog (#2160 #2173). You can access it through “Edit” -> “Preferences” -> “Shortcuts”.



Also, some default shortcuts/hotkeys were changed for improving usability and compability with other apps like Inkscape (#1769 #1776 #1811 #1823 #1848 #1855 #1953 #2033 #2172).

For example, all tools shortcuts are remapped to single keys, no need to hold Alt key anymore!



Also:

- On MacOS “Undo” and “Redo” shortcuts now use (Command key) instead of Control.

- Added shortcut for toggling Animate mode (Ctrl Spacebar).

UI/UX improvements

- Editing parameters does not requires one extra mouse click anymore (#1337 #1351 #1325).
- TimeLoop valuenode now editable via workspace (#2066).
- LinkableValueNode now provides methods for inverse manipulation (#2057).
- Now it is possible to close tabs by clicking with middle mouse button (#2107 #2120 #2210).
- Add “Undock panel” context menu item for panel tabs (#1929 #2109).

Bugfixes

- Fixed incorrect background restoration on animated GIFs with transparent background (#2134).
- Fixed crash when user drags a Group layer to a child group (#2110 #2116).
- Fixed crash in Spline Width and return of improper type in Spline converters (#1890).
- Exported canvas can be reopened again after closing (#2092).
- Fixed action “Duplicate Layer” for Skeleton Layer and their bone links (#2059).
- Fixed action “Duplicate Layer” for Skeleton Deformation Layer and their bone links (#2071).
- Fixed error that prevented convert to dot-product valuenode (#1260).
- Fixed issue with saving *loop* flag in some dynamic lists (#2165).
- Fixed SVG importer ignoring some style properties set as attributes and fixed a lot of memory leaks (#2137).
- Fixed crash in rare cases caused by race condition when rebuilding LayerTreeStore (#2209).
- Fixed issue when Widget_Vector do not uses locale for decimal separator (#2102 #2157).
- Don’t show extra vertical scrollbar for Dock Sound (#1173).
- Fix missing `is_distance` hint for vertex coordinates (#2000).
- Fixed problem with missing *.glsf files in distribution (#1982).
- Fixed issue with AppImage taking too long to start at first run (#1340 #1849).

16.2.4 Synfig 1.4.3 Release Notes

Synfig Studio 1.4.3 introduces major performance optimizations for Windows users and important bugfixes for all platforms. Please read the details below.

Optimizations for Windows version

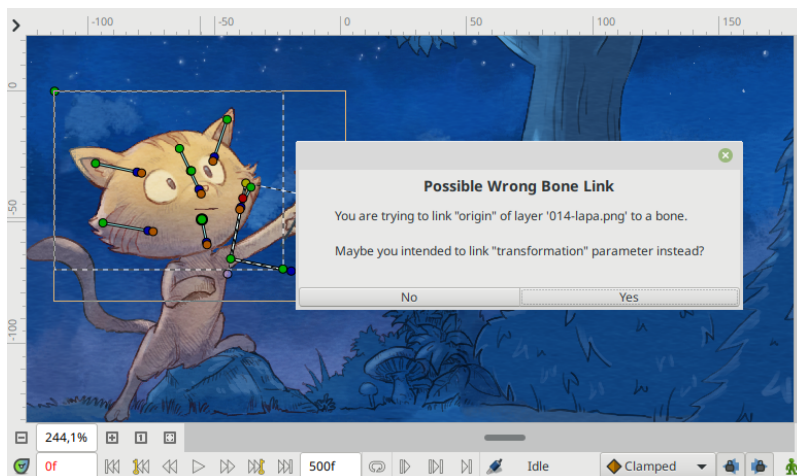


- We have removed many bottlenecks in Synfig's code, related with redundant calls of “getenv” function. This resulted in great performance improvement of Windows version (#2826). This fix also gives slight speed boost for MacOS, while Linux users will not see any significant difference.
- Windows version now shipped with embedded Python version 3.8.10, which results in smaller size of distributive and faster installation (#67).

Critical fixes

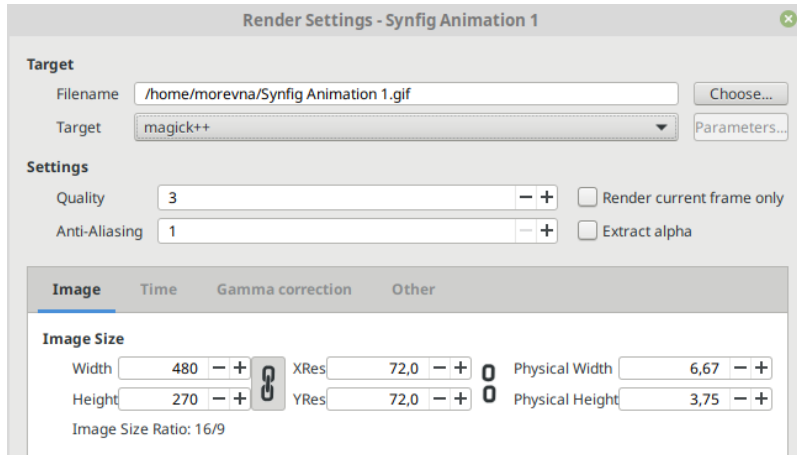
- Connecting bone parameter to exported value does not results in corrupted file anymore (#2433).
- Fixed issue with autosave not working when interval is set to 15 seconds (#2647).
- Fixed “Link To Spline” action not shown in menu when trying to connect vertex to spline (#2344).
- Fixed Appimage compatibility with recent Linux distributions (#33).
- Fixed issue with Text Tool not working on MacOS because of missing “fonts.conf” file (#2380).

Improvements



- Avoid mistakes when linking group to bone: the dialog asks if user wants to link transformation instead of origin (#2468).
- The “Link width and height” option is enabled by default in Render Settings (#2351).

Export



- Fixed issue with wrong image aspect when rendering to non-standard dimensions (#2463).
- Fixed exporting issue which was resulting in exported video trimmed to audio length (#2477).
- Fixed “magick++” render module not available in Linux AppImage (#2320).
- Fixed crash when rendering to gif target (#2389).
- Fixed compatibility with FFmpeg 3.x (#2628).

Import

- Fixed importing Papagayo (.pgo) file not showing the expected layers (#2495).
- Fixed issue with file corruption after importing an SVG file (#2482).
- Fixed incorrect framerate when importing image sequence (#2224).
- Fixed crash when trying to import a file into itself (#2239).
- Fixed wrong size of image which is imported to canvas with non-standard size (#2252).

Other fixes

- Apply action to all selected layers from choosing entry from context menu (#2352).
- Fixed Clamp Layer not rendering (#2550).
- Fixed crash when deleting last waypoint from exported parameter (#2587).
- Don’t allow to connect to exported value node if it has wrong type (#2694).
- Fixed crash when using Width Tool on Skeleton Layer (#2500).
- Fix Keyframes Panel doesn’t capture space input in entry text field (#1820).
- Parameters and Timeline panels show the same row selected (#2422).

16.2.5 Synfig 1.4.4 Release Notes

Synfig Studio version 1.4.4 delivers some important bugfixes for all platforms. Please read the details below.

Critical fixes

- Fixed missing last frame when exporting video (#2912).
- Fixed crash when using Draw Tool in Fedora/Flatpak version (#2445).
- Fixed issue with opening file via CLI when its path contain non-Latin characters (Windows) (#2819).
- Fixed issue when right-clicking a layer shows empty menu on just opened file (#2781).
- Fixed loading of palette (#2722).
- Fixed application freeze when importing palette (#2730).
- Fixed messed up translations for Russian language.

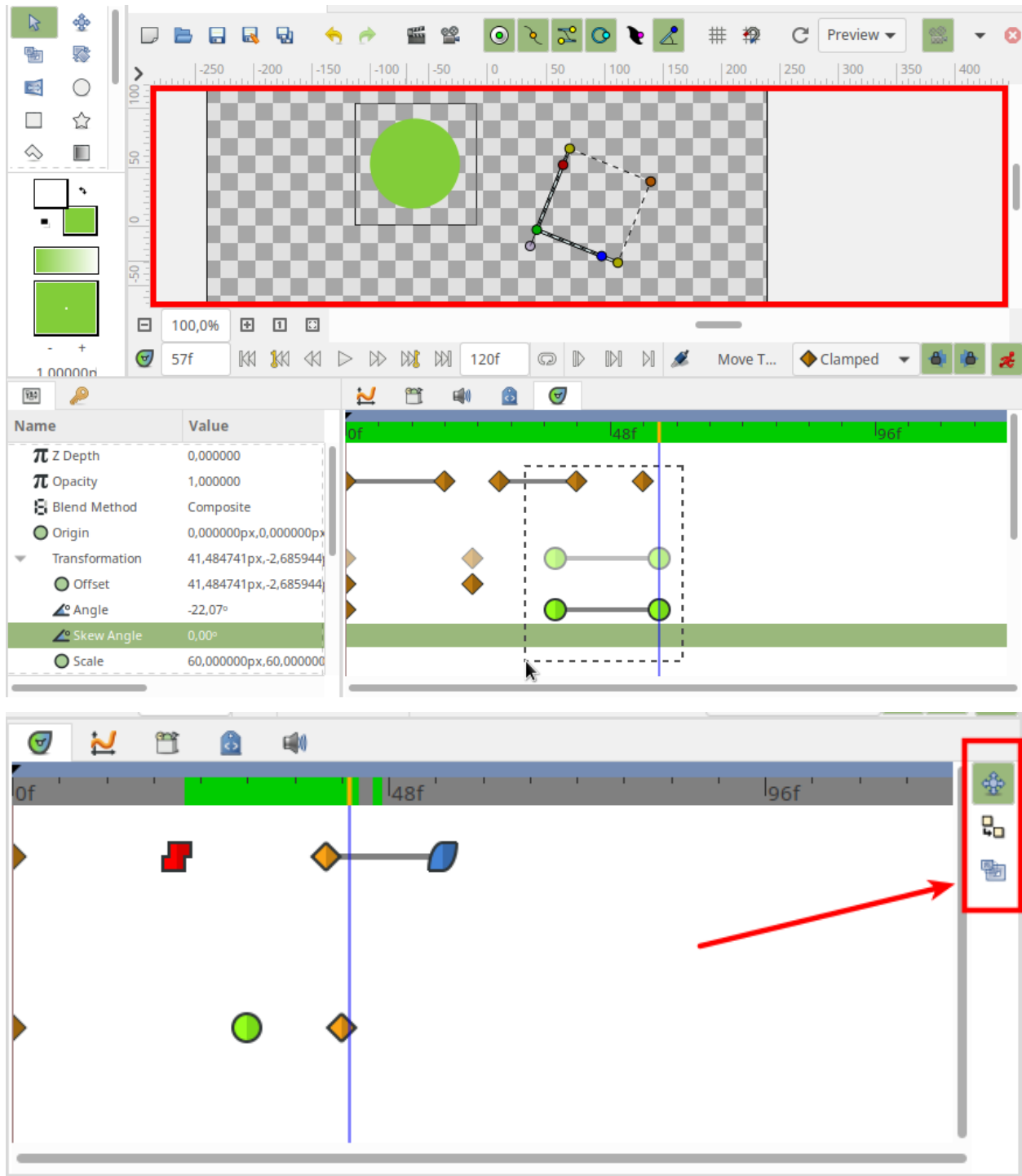
Fixes for build system (Autotools)

- Allow to compile release build with debug info (#2838).
- Add support for MLT++-7 (#2306).

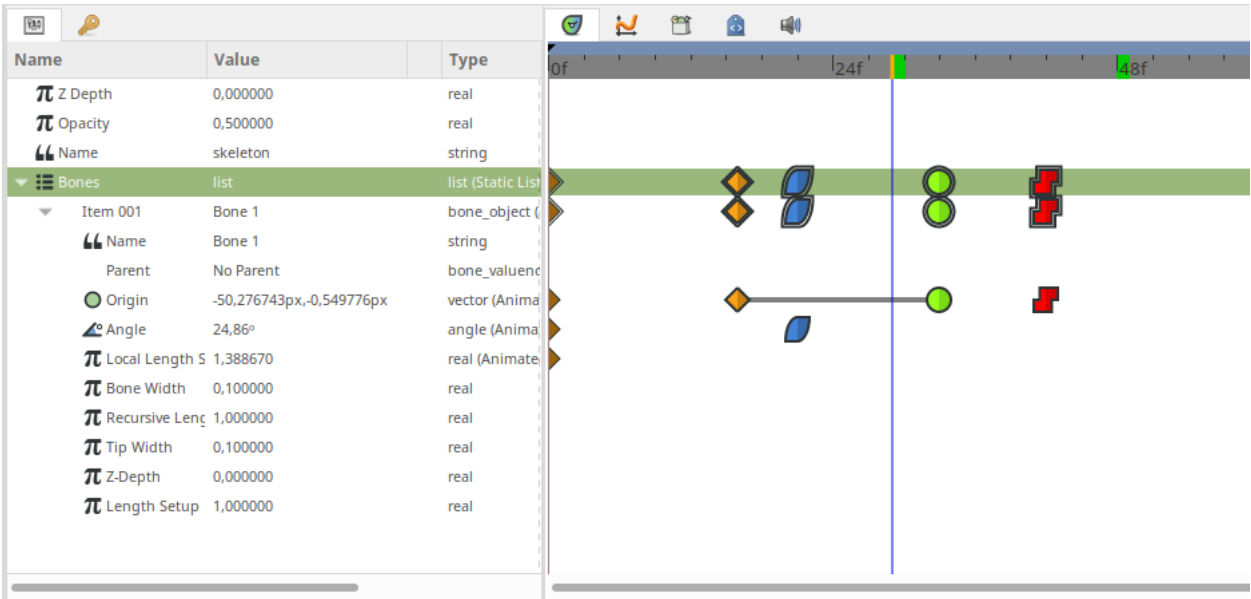
16.2.6 Synfig 1.5.0 Release Notes

Rewritten TimeTrack Panel

Now working with waypoints is so much easier - you can select multiple items using [bounding box selection](#), duplicate and scale them, navigate to next or previous waypoints using “n” and “b” keys (holding Shift skips 5 waypoints).

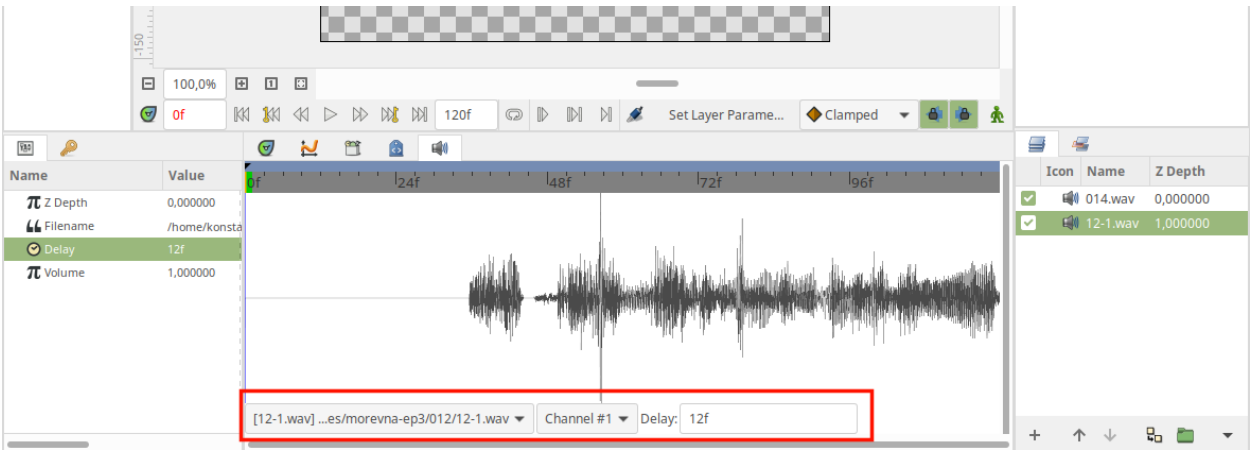


Also, the new Timetrack Panel now indicates segments when value of parameter remains constant.



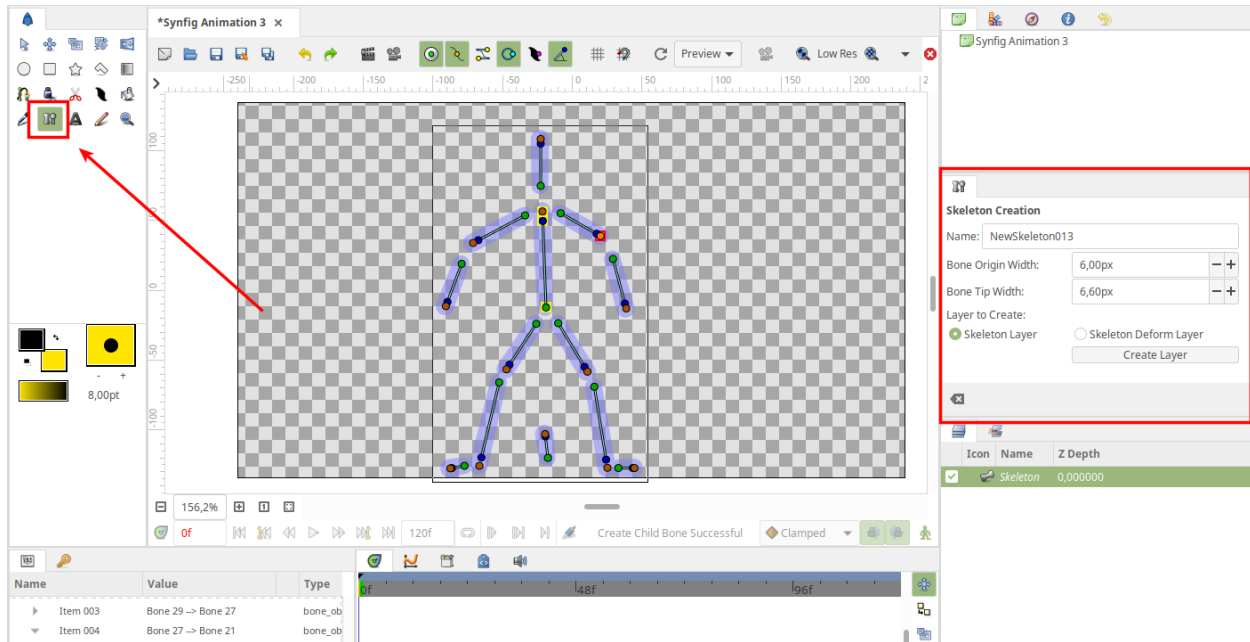
Improved Sound Panel

Sound Panel now has a list of available sound layers. This way it's much easier to choose which waveform to show in the panel. See [issue 1346](#) for details.



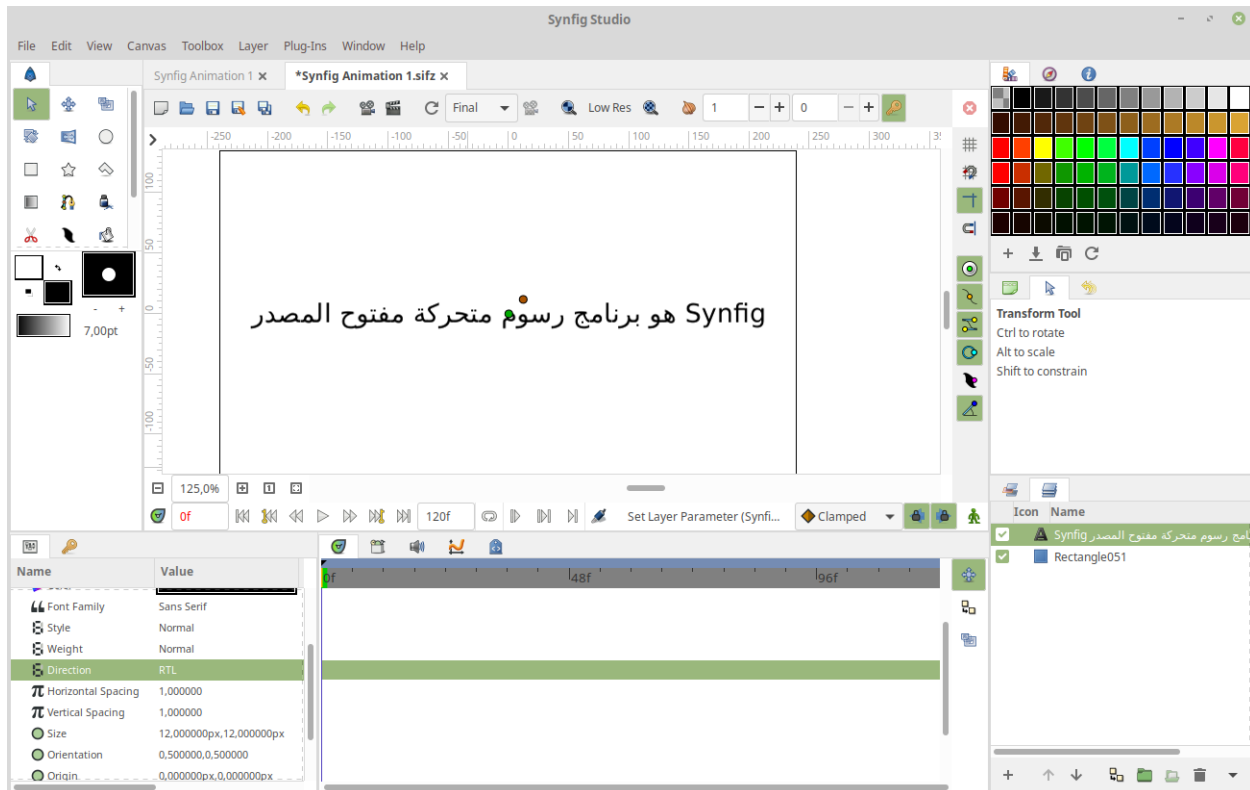
New Skeleton Tool

The new intuitive Skeleton Tool allows users to easily create bones. With this tool it is also possible to easily change the structure of skeletons by re-parenting bones.

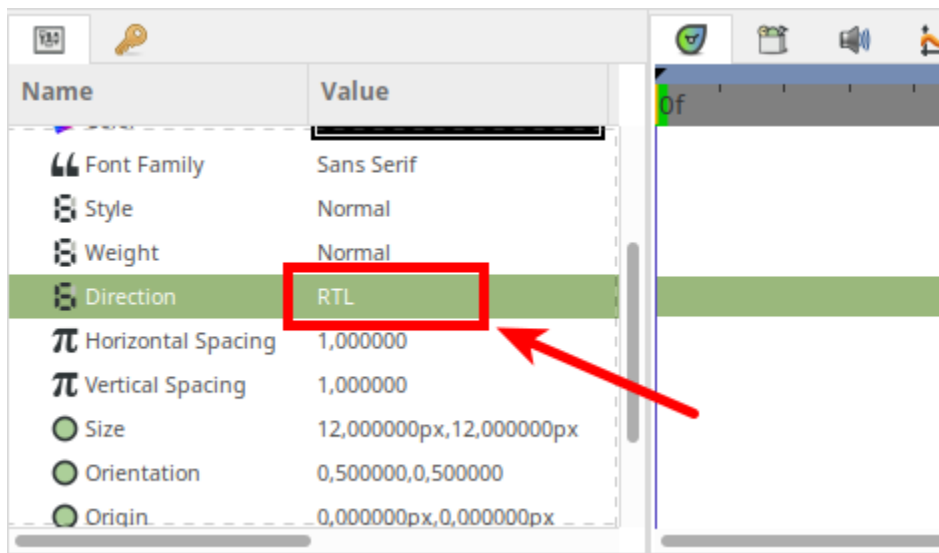


See this documentation page to learn more about this feature - <https://synfig.readthedocs.io/en/latest/tools/skeleton.html>.

Rewritten Text Layer



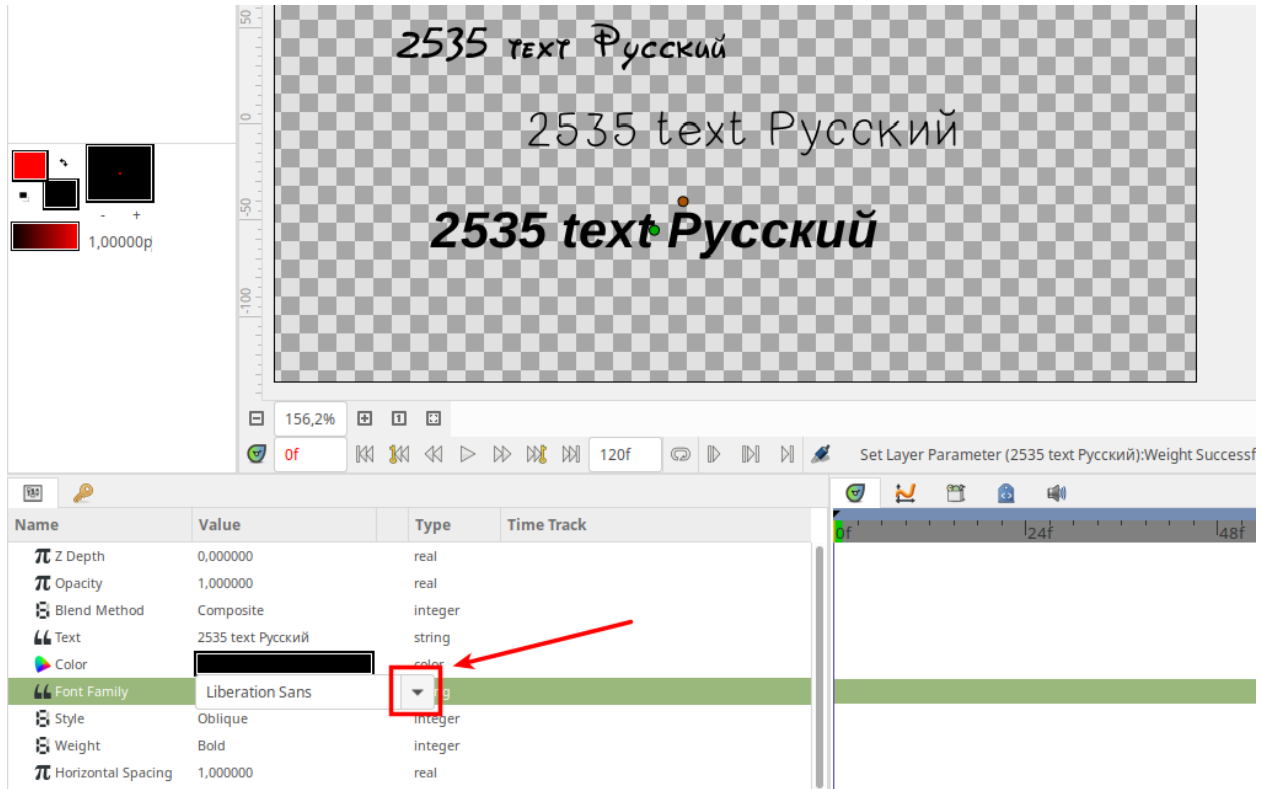
- Text Layer fully rewritten and optimized, which results in much faster rendering (#2362).
- Text Layer now have proper support for RTL languages (i.e. Arabic) (#2232). Please notice that if you use a string which mixes RTL and LTR text, then you might need to set a “Direction” parameter in Text Layer to indicate the proper order.



- Text Layers now visible when workarea displays its content in Draft mode (#631).
- Fixed issue with “jumping” letters in animated text (#389).

- Fixed issue with quality regression under Text Layer (#831).
- Fixed glitched rendering output when Text Layer is combined with Perspective Layer (#1566).
- For Text Layer it is now possible to select font family from drop-down menu in Parameters and Tool Options (PR #1432, #2355).
- the list shows all fonts available in your system, so it is so easy to

choose one now!



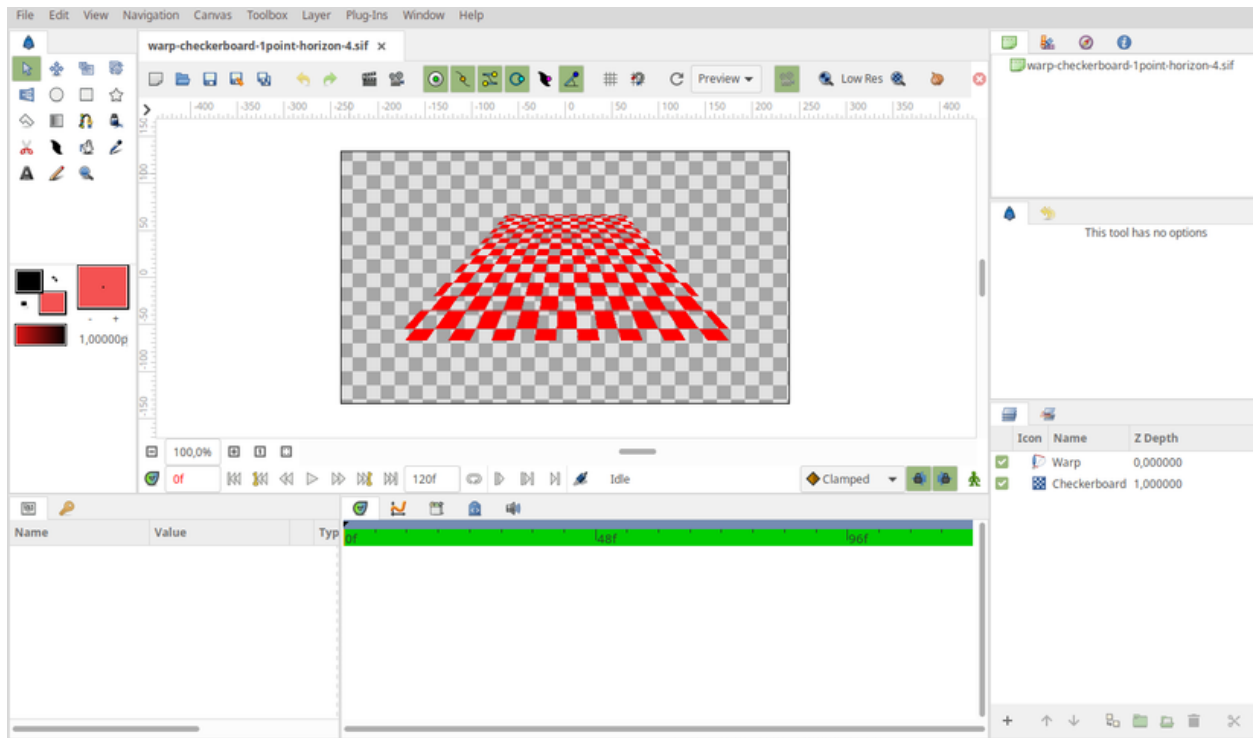
Optimized Perspective Layer

The Warp Layer was completely rewritten and ported to the Cobra render engine. We also renamed it to Perspective Layer, as it sounds more intuitive for users.

As result, the new Perspective Layer works much faster than the old Warp Layer (see [detailed report about speed tests](#)) and also fixes some old nasty bugs (see issues #318 and #418).

Some other improvements in Perspective Layer:

- Perspective Layer is rendered with simplifications when the workarea uses Preview or Draft mode. This allows to achieve more than 3x speed increase.
- Perspective Layer now has a new parameter - “Interpolation” - which allows you to optimize quality for final Render.



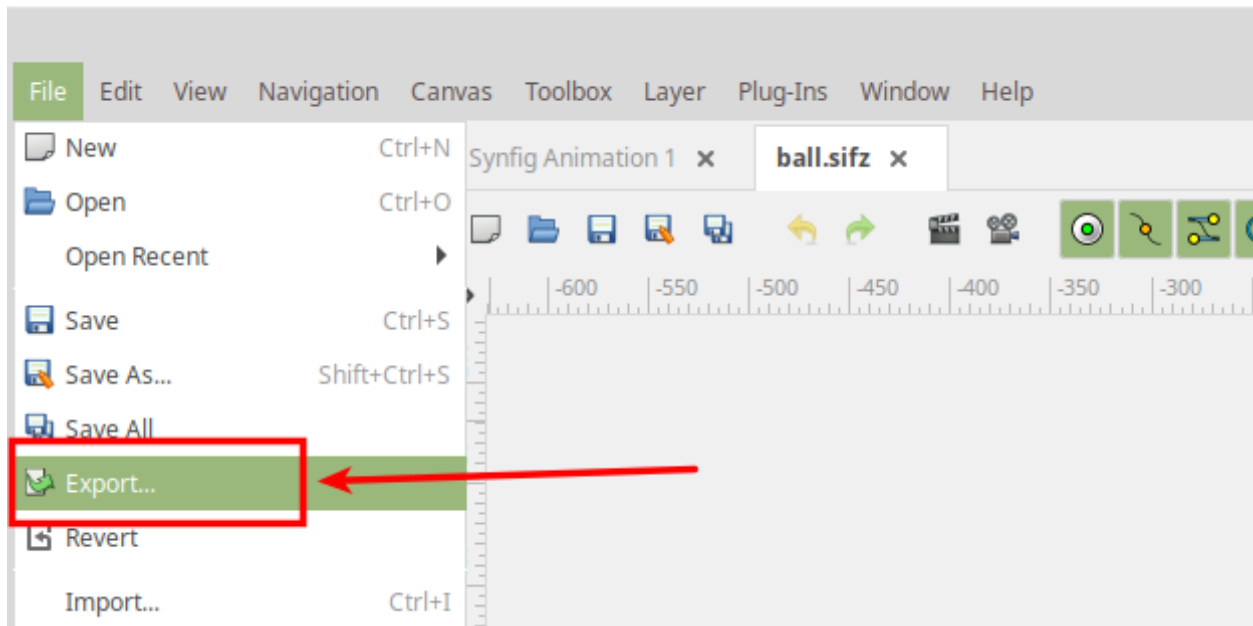
Reworked Plugins Infrastructure

Now anyone with knowledge of Python language can write plugins which allows Synfig to export/import new formats and those formats will become available

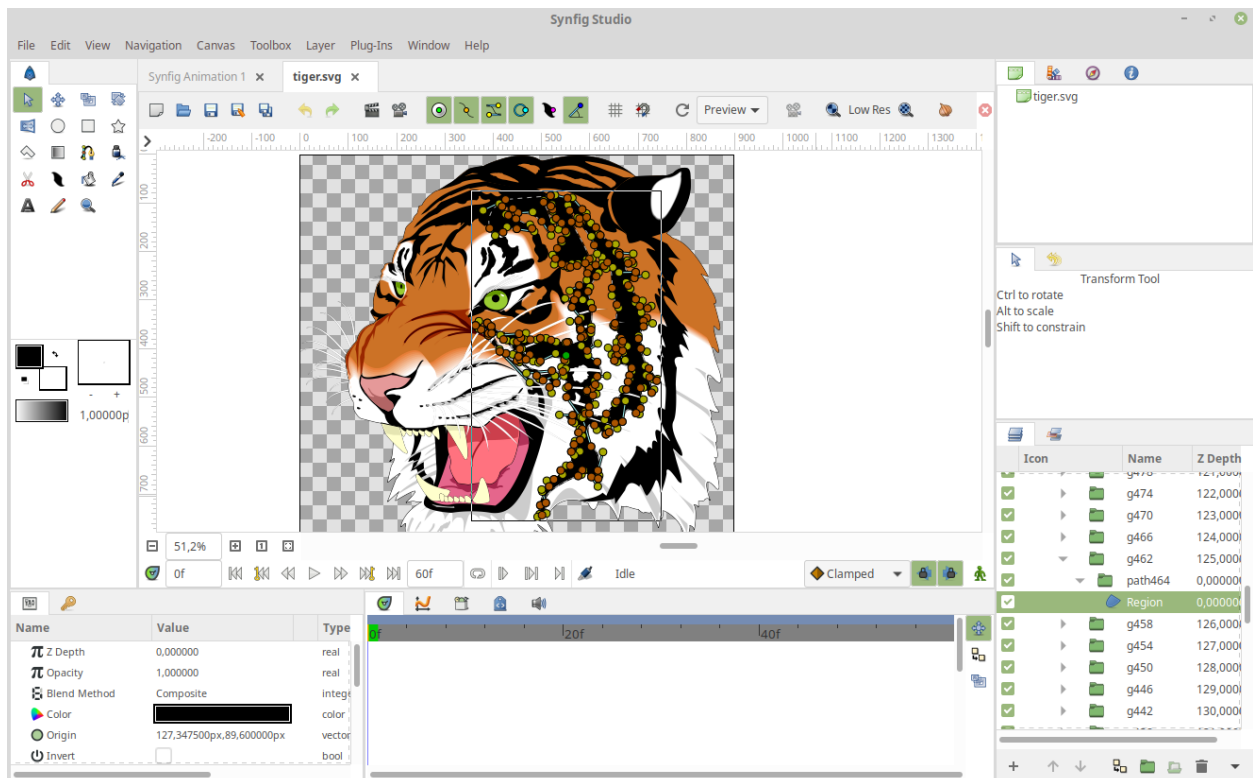
In this version the Plugins system was *improved* to allow writing import/export functions using Python language. Now anyone with Python knowledge can extend Synfig's capabilities to import or export new file formats!

Such import/export plugins are not displayed in the regular "Plugins" menu, their functionality is available through "File -> Import" or "File -> Export" menu commands instead.

With this change, the Lottie Exporter now called through "File -> Export" menu.



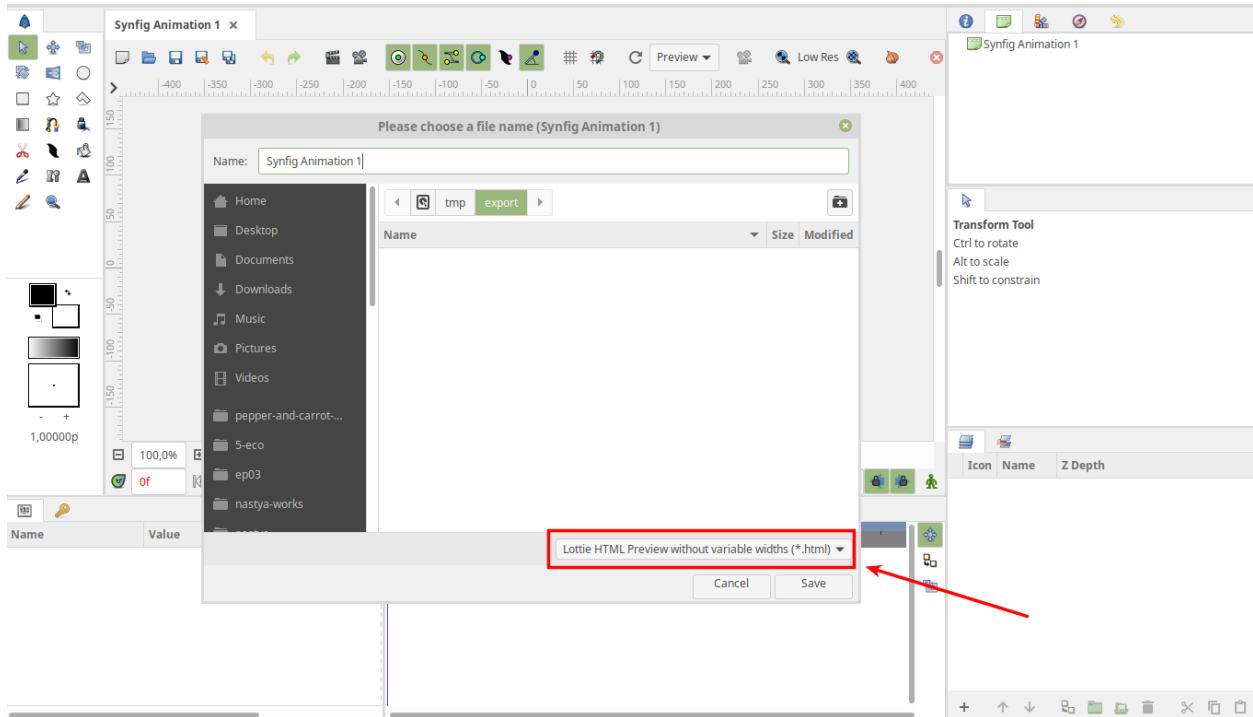
As another notable example, you might want to check [this awesome plugin](#), which adds capabilities for import/export of SVG, Telegram Animated Stickers (TGS) and Lottie (alternative implementation). The image below shows a sample “tiger.svg” file imported into Synfig using this plugin.



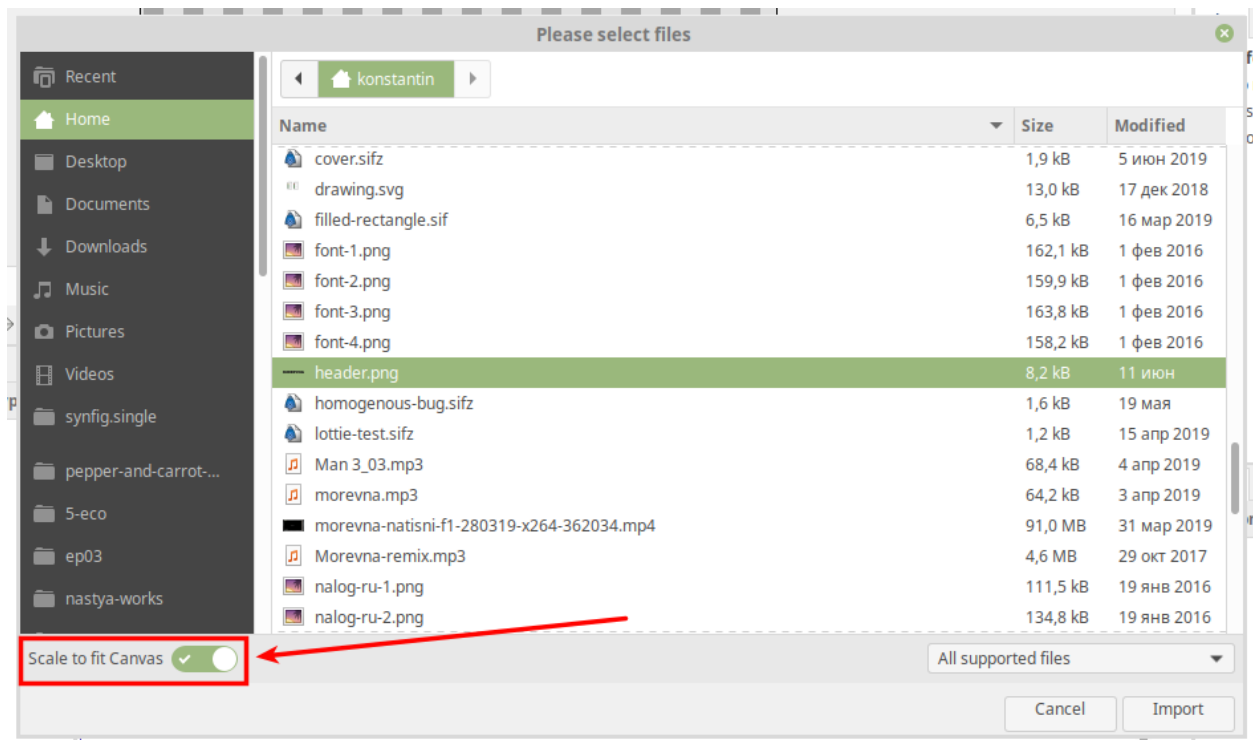
For more information about the improved plugins system check [this documentation page](#). There are also relevant discussions [here](#) and [here](#).

Improvements for Import and Export

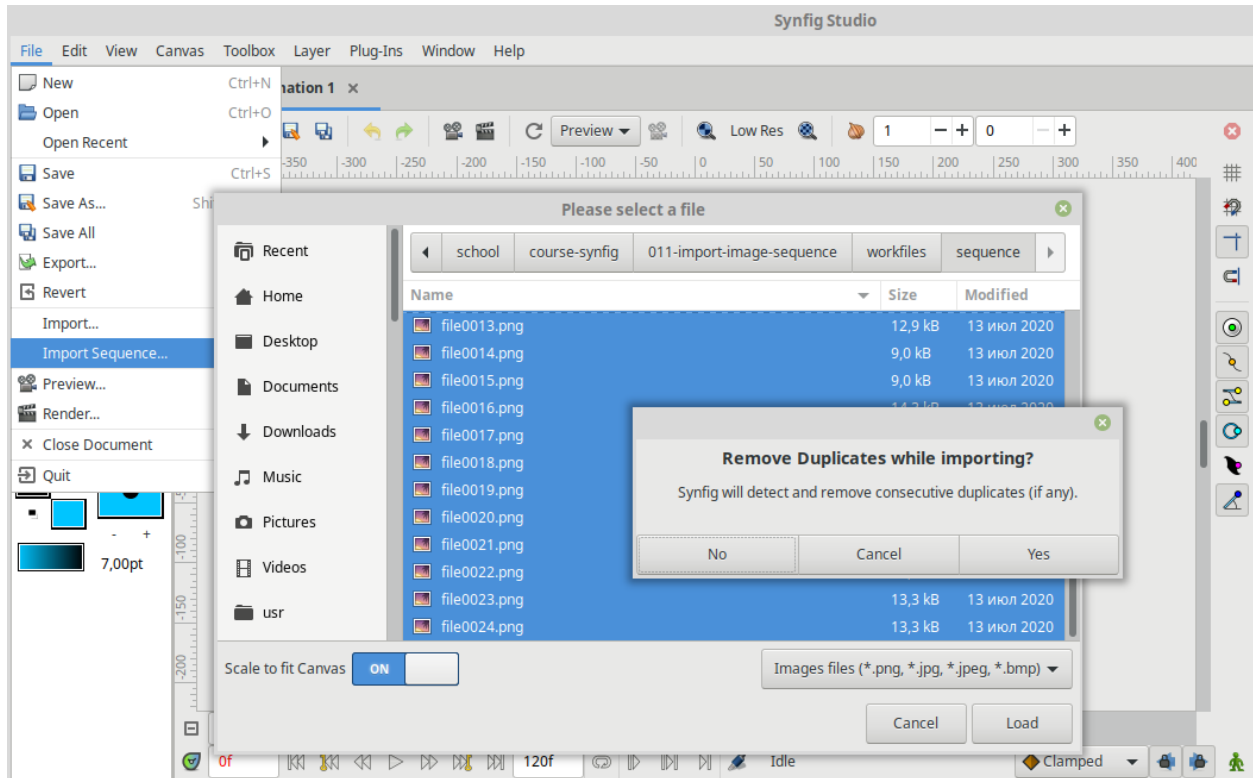
Lottie Exporter now has new mode, which exports outlines without variable widths (see PR` <<https://github.com/synfig/synfig/pull/1647>>`__#1647). This greatly reduces the size of exported files and we advise to use it if your animation does not have outlines with variable widths.



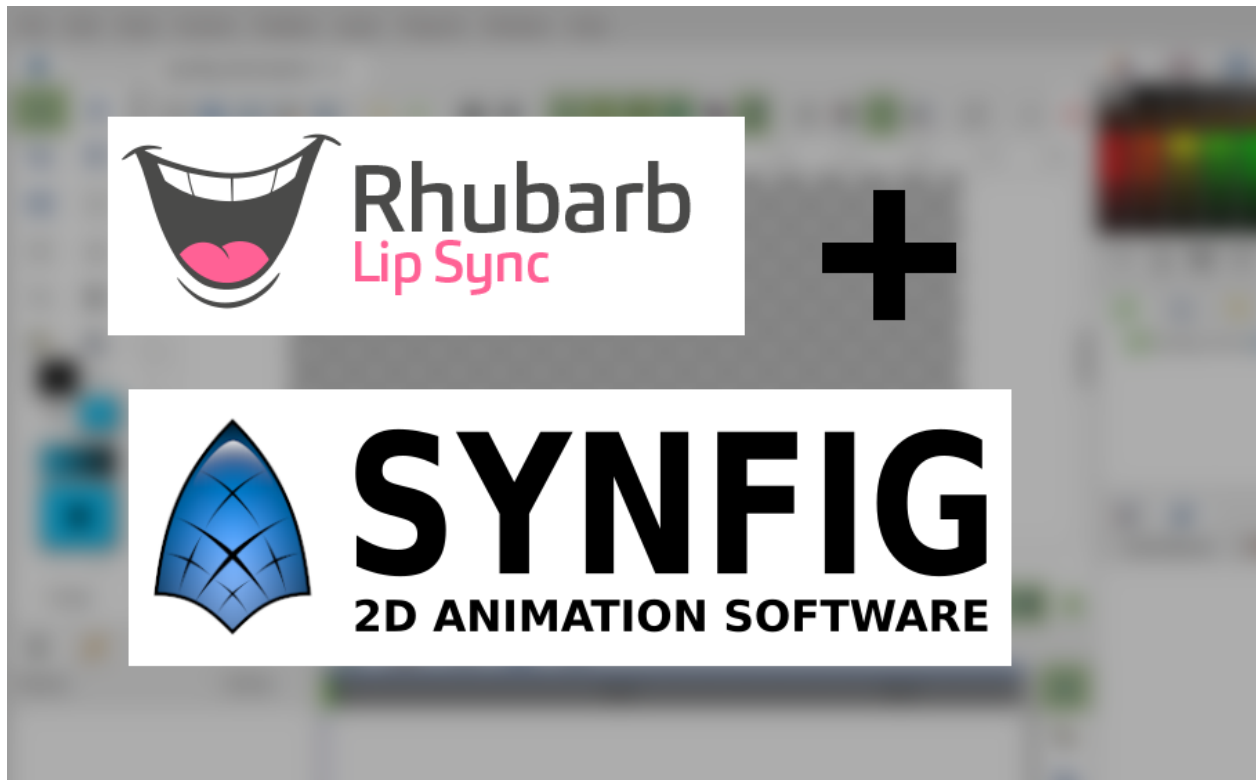
Import image dialog now has “Scale to fit canvas” option (PR #1507).



“Import Sequence” function now allows to detect duplicate frames in image sequences. In case if a duplicate frame is detected it doesn’t put extra waypoint, which results in cleaner timeline and better readability.



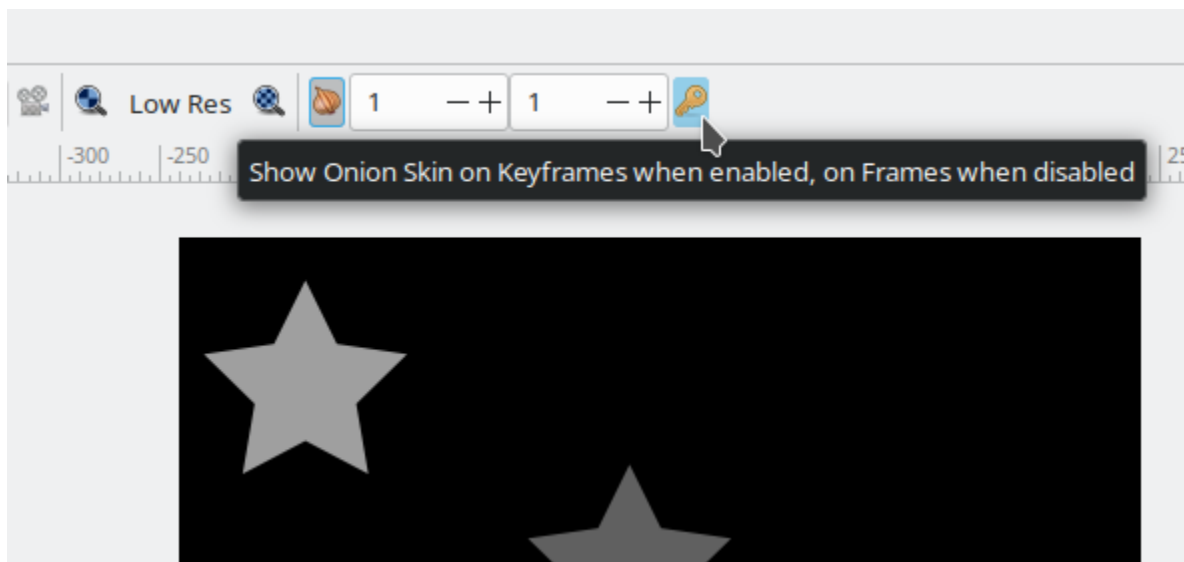
Synfig now capable to import TSV and XML file formats, generated by [Rhubarb Lip Sync](#) (see issue #2169 for details). It works in the same way as Papagayo Lipsync import, shown in [this video](#).



AnimatedFile Converter got support for more data types - String, Angle (degrees), Bool, Integer, Real, Time and Vector. This is useful for cases when you have animation data (like object coordinates along time) generated by another software and you wish to use that data in Synfig. See [issue #2211](#) for details.

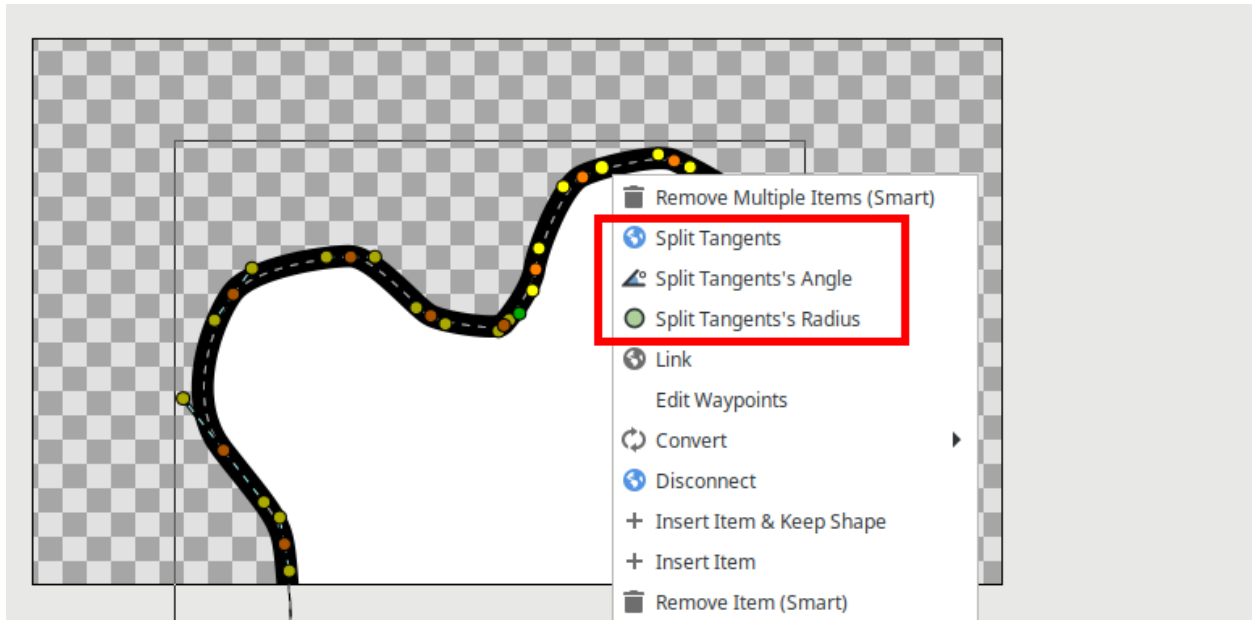
Usability Improvements

New option for Onion Skin allows to choose between Keyframes and Frames ([#2166](#)).

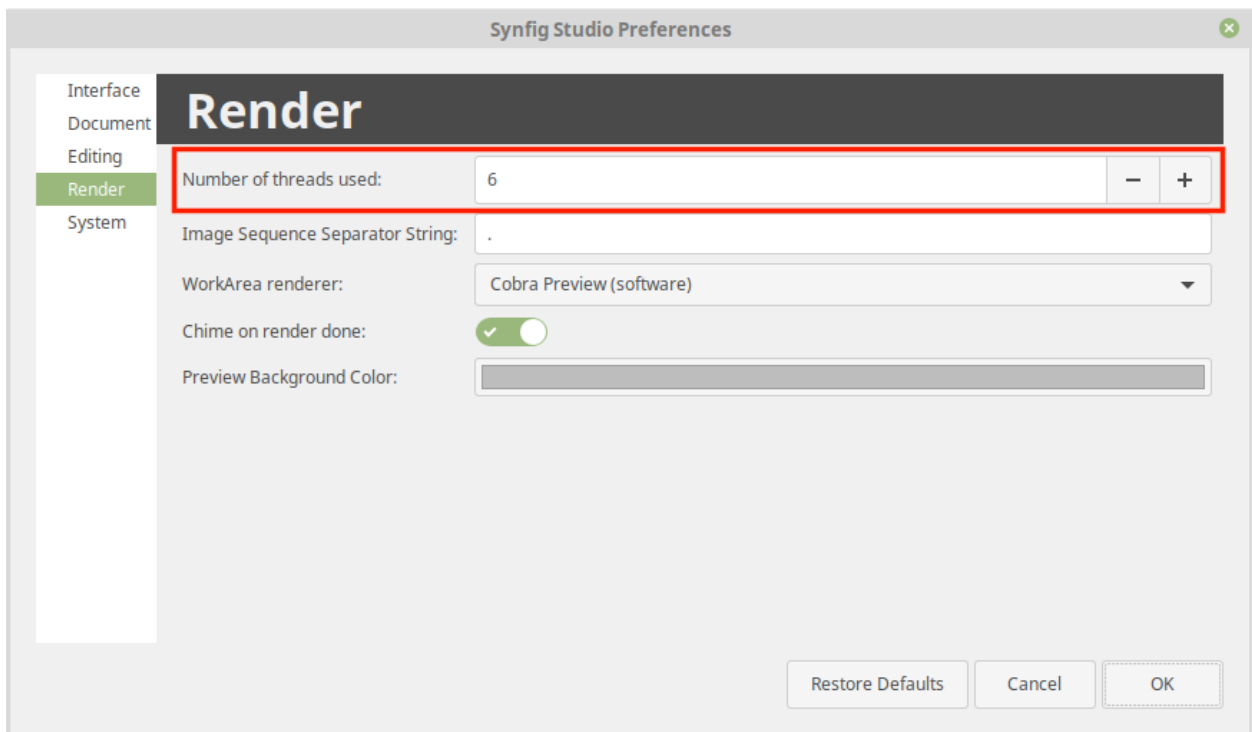


Now it is possible to create a new vertex on a Spline with double-click ([#2312](#)).

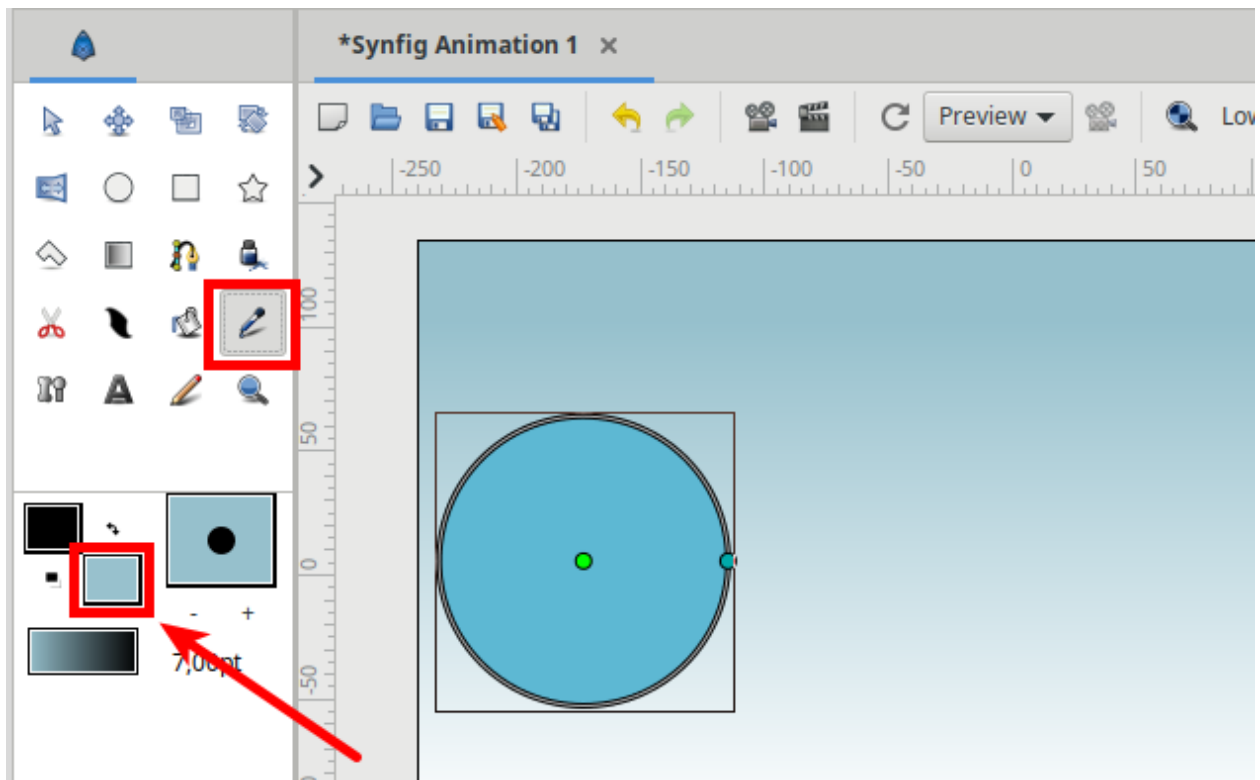
In this version it is possible to merge/split tangents for multiple vertices. See [issue #2010](#) for details.



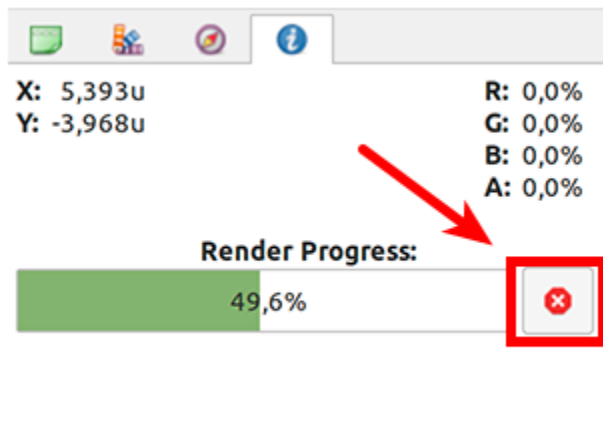
Now it is possible to set the number of rendering threads from Preferences dialog (#1999).



Eyedropper Tool now has the ability to select fill color using Ctrl+click.

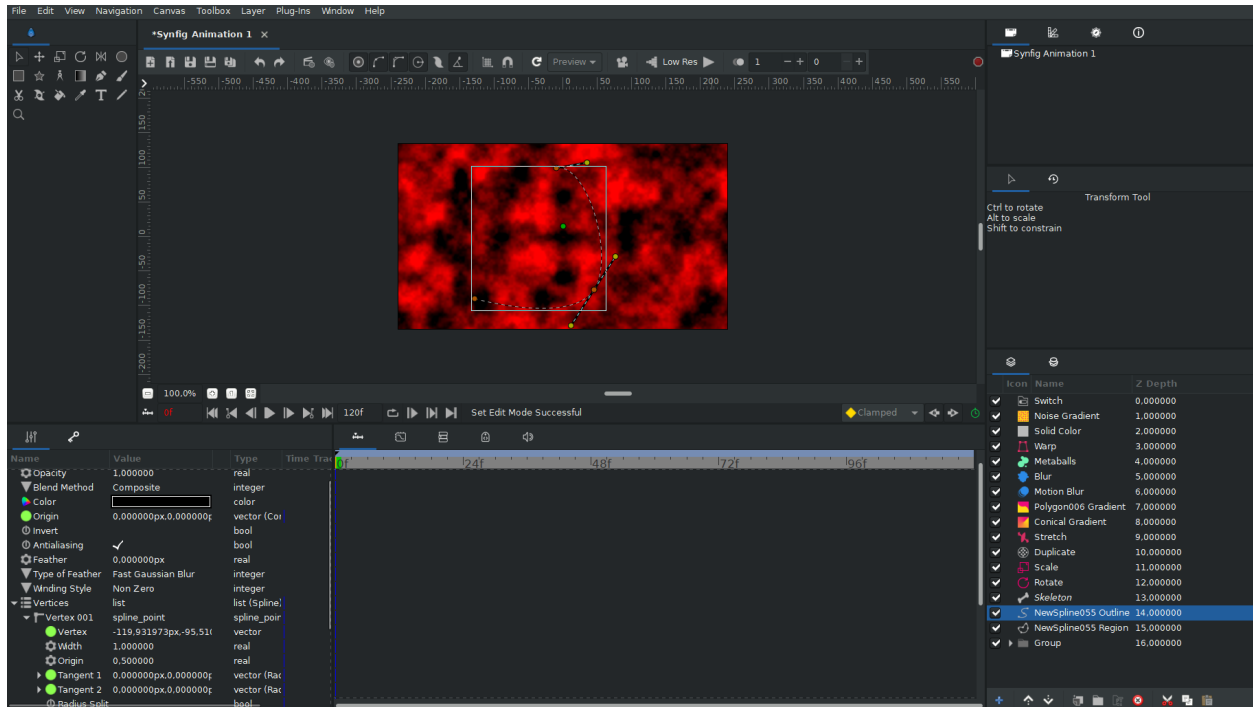


Added possibility to stop rendering process.



Icon themes

Now it is possible to change icon theme for Synfig Studio. Icon set (theme) can be placed in a folder inside “share/synfig/icons/” path (eg. share/synfig/icons/new-theme-folder) and then activated by setting the SYNFIG_ICON_THEME environment variable (eg. SYNFIG_ICON_THEME=new-theme-folder). See [PR #1483](#) for details.



Please notice that this is just a first step in ongoing work on implementing theming support for Synfig Studio. You can check [this discussion](#) for details.

CMake build system

This version introduces the possibility to build Synfig using CMake. Although this build method is not yet suitable for creating packages, it greatly simplifies the build process for those who wish to compile Synfig for development or testing purposes. Please check [this documentation page](#) for instructions.

ABOUT THIS MANUAL

The Synfig Manual is a community driven effort to which anyone can contribute. Whether you like to fix a tiny spelling mistake or rewrite an entire chapter, your help with the Synfig manual is most welcome!

17.1 License

Except where otherwise noted, the content of the Synfig Manual is available under a [Creative Commons Attribution-ShareAlike 4.0 International License](#) or any later version. Excluded from this license are used logos, trademarks, icons, source code and Python scripts.

When reproduced manual in full please make sure to include full list of contributors (below).

In case of partial re-use or when including full list of contributors is not possible, please attribute the “Synfig Documentation Team” and include a hyperlink (online) or URL (in print) to manual and list of contributors on this page. For example:

```
The Synfig Manual
by the Synfig Documentation Team
is licensed under a CC-BY-SA v4.0.
```

See [Best practices for attribution](#) for further explanation.

Please, that when contributing to the manual you do not hold exclusive copyright to your text. You are, of course, acknowledged and appreciated for your contribution. However, others can change and improve your text in order to keep the manual consistent and up to date.

17.2 Contributors

- Ankit Kumar Dwivedi (ankit-kumar-dwivedi)
- Mattia Basaglia (mbasaglia)
- Wiki:00000111
- Wiki:1158517325
- Wiki:AkhIL
- Wiki:Andrewkay
- Wiki:Auldsbel
- Wiki:AzfZtb
- Wiki:Bazza

- Wiki:BentFX
- Wiki:Berteh
- Wiki:BobSynfig
- Wiki:Bombe
- Wiki:Bottye
- Wiki:Carbonphobia
- Wiki:Ciaran
- Wiki:Ckjackson
- Wiki:Claus
- Wiki:Cyborg ar
- Wiki:DaveJ
- Wiki:Darco
- Wiki:Darkspace65
- Wiki:DeanBrettle
- Wiki:Deville
- Wiki:D.j.a.y - Jerome Blanchi (D.j.a.y)
- Wiki:Dooglus - Chris Moore (dooglus)
- Wiki:Dopski
- Wiki:Double.dose
- Wiki:Dragao
- Wiki:Dustinmoorenet
- Wiki:Eatmydata
- Wiki:EleFlameMax
- Wiki:Erubin
- Wiki:Evilkillerfiggin
- Wiki:Fabifabito
- Wiki:Felicidade
- Wiki:FirasH
- Wiki:Fortitude
- Wiki:Fprat
- Wiki:Freakabcd
- Wiki:Genete - Carlos López González (genete)
- Wiki:Gerco
- Wiki:Ghosthand - Chris Blount
- Wiki:Grafika lucian
- Wiki:Gremble

- Wiki:Grin
- Wiki:Hallel
- Wiki:Helloatfood
- Wiki:HoekstraF
- Wiki:HynShb
- Wiki:Jancsika
- Wiki:Jcome - Yu Chen (jcome)
- Wiki:Jyuzwalk
- Wiki:JniZfu
- Wiki:Kaleru
- Wiki:KingoftheBears
- Wiki:LdrGf8
- Wiki:Ii1Cuh
- Wiki:Marietto2008
- Wiki:Matumio
- Wiki:Maxy
- Wiki:Mfwitten
- Wiki:MihanEntalpo
- Wiki:Mikalaari
- Wiki:Mjtrac
- Wiki:MNeto
- Wiki:Mondojohnson
- Wiki:Mz-alph
- Wiki:Nikitakit
- Wiki:Nitrofurano
- Wiki:Nqatsi
- Wiki:Nqpz
- Wiki:Ohoservices
- Wiki:Orestes
- Wiki:PaulWise
- Wiki:Paulyvalent
- Wiki:Poupoumtralala
- Wiki:Pupazzo
- Wiki:Pxegeek - Chris Norman
- Wiki:Recycler
- Wiki:Rore

- Wiki:Rosenth
- Wiki:Rtsp
- Wiki:Rubikcube - Daniel Hornung
- Wiki:Rylleman - David Rylander (rylleman)
- Wiki:Sandra
- Wiki:Saorsa
- Wiki:SeparateSausages
- Wiki:SinSignificant
- Wiki:Sj6Tdm
- Wiki:Slulego
- Wiki:Smirkenburke
- Wiki:SnapSilverlight
- Wiki:Synme
- Wiki:Tsu Dho Nimh
- Wiki:Ugajin
- Wiki:VgvAdn
- Wiki:VovVhg
- Wiki:Want - Aleš Kapica
- Wiki:YyrR22
- Wiki:Yushi
- Wiki:Zelgadis - Konstantin Dmitriev (morevnaproject)
- Wiki:Zenoscope
- Wiki:ZwqV66

HOW TO HELP

Synfig is an open-source software, and “open-source” is all about community contributions! So even a regular (non-developer) user can help a lot! How? Read below.

1. Help other users

There are many people who are asking for help at [our forums](#). If you already know how to solve particular problem - don't hesitate to lend a hand to them!

2. Report issues

Found an issue? Report it to [our issue tracker](#)! Also, we recommend to read [this guide](#) on how to report bugs effectively.

3. Help to test development releases

Give a try to [development releases](#) and help to identify issues and weak places. The more people will test development release - the better stable one becomes!

4. Help to test open Pull Requests

You can help developers by giving your feedback about new improvements, which are proposed to be added into code (Pull Requests or PRs). This will help developers decide which features are ready to be included and allow to spot what should be improved. Check the list of open Pull Requests on [this page](#) and read `testing_prs` for details.

5. Help to translate user interface

Translation of Synfig's user interface takes place online at [Transifex platform](#). You can help improve translation to your language.

19.1 Audio Synchronisation

19.2 Colors Dialog

19.3 Gradient Editor Dialog

19.4 Group Transformation Widget

19.5 Handles

19.6 Import Image Layer

19.7 Mouse Shortcuts

19.8 New Layer Defaults

19.8.1 Brush Colors

19.8.2 Brush Size

19.8.3 Default Blend Method

19.8.4 Default Opacity

19.8.5 Current Gradient

19.8.6 Default Interpolation

19.9 Onion Skin

19.10 Preferences Dialog System

19.11 Timebar

INTERPOLATIONS

20.1 Clamped

20.2 Constant

20.3 EaseIN/Out

20.4 Linear

20.5 TCB

INDICES AND TABLES

- `genindex`
- `modindex`
- `search`