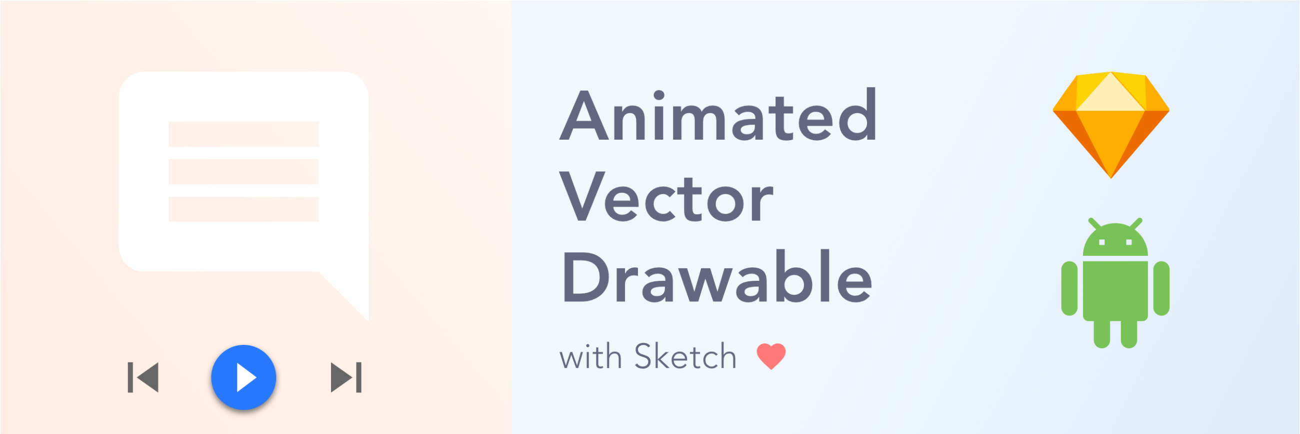


Sketch + Animated Vector Drawable = ❤

How to create beautiful AnimatedVectorDrawable with Sketch ?

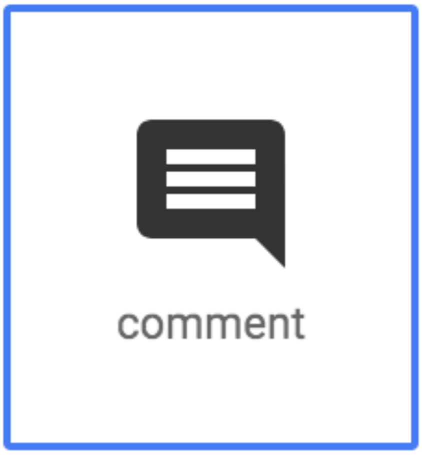
Philippe BOISNEY Follow
Aug 31, 2018 · 5 min read

UX/UI is one of the things I really love through my work. Actually, when you think about it, your users really don’t care about your code and how you’ve designed it : **they only judge what they can see**.

In this post, I will show you how to create an **Animated Vector Drawable** with two different tools, Sketch to prepare the assets and Shape Shifter to animate them.

1. Choosing an icon

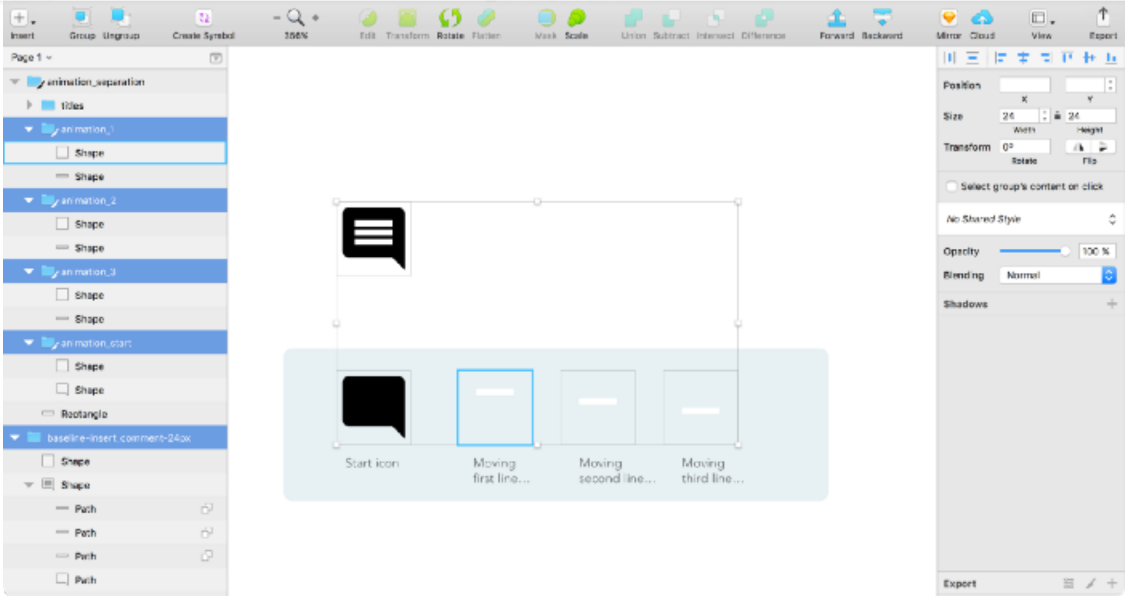
First of all, you will have to choose an icon you want to animate. For this post, I decided to download this icon from Material.io in SVG format :



2. Designing the animation

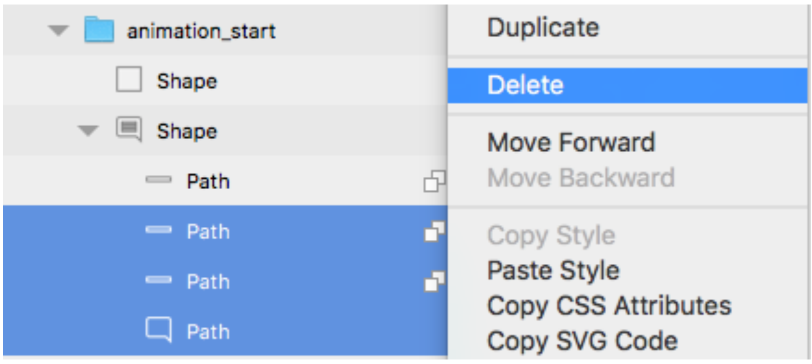
Next, you will have to **think about** the animation you want for your icon. Be creative ! In this case, I thought that it will be a good idea to **move the three lines**, *one after the other*, almost as if the user started typing some comment.

So, now we've got the idea, we need to **separate** each element ***we would like to animate*** on the final animation with Sketch, like this :



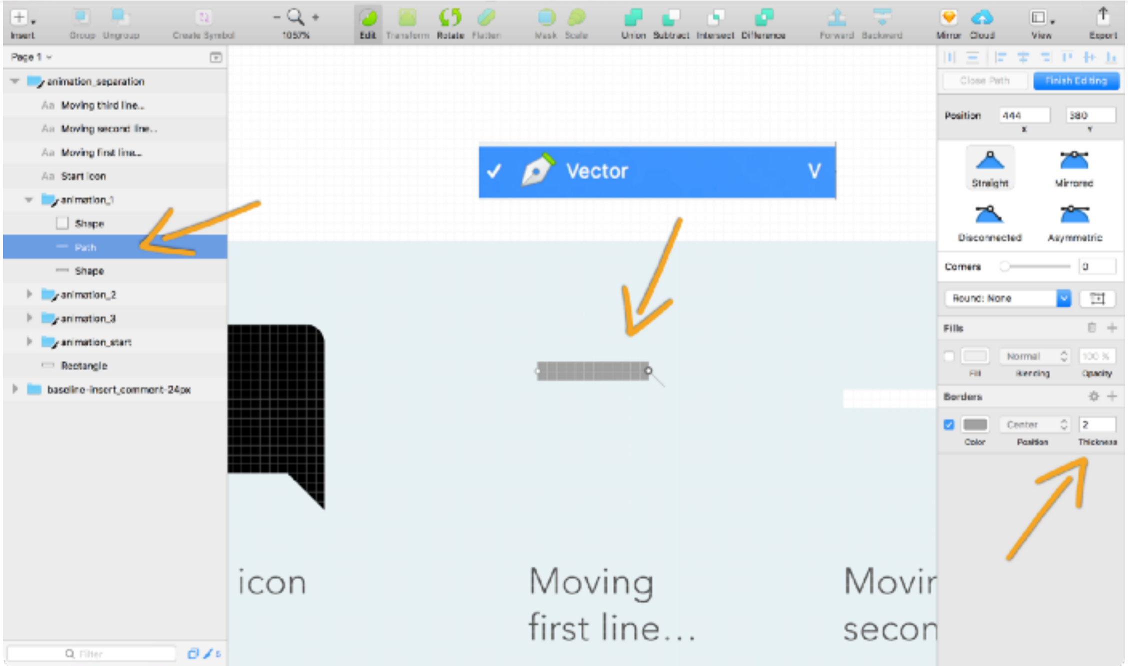
Separation of each item we want to animate

Remember, we only want **to move the three lines**. So, I **copied** and **pasted** the main icon **4 times** (and NEVER modified its size properties) and only kept the shape I wanted.



For each part of the final animation, I've deleted what I don't needed.

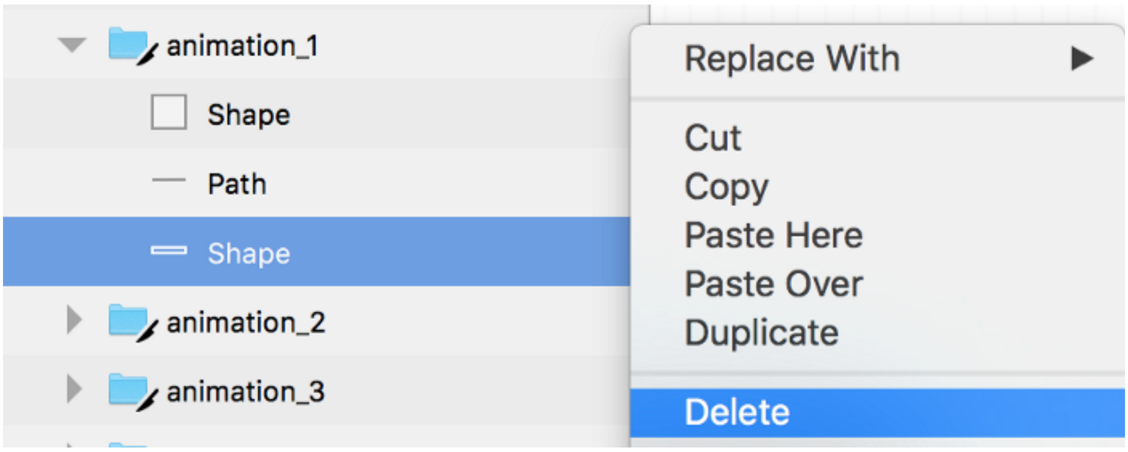
Next, **the tricky part** : you need to ***manually reproduce*** each line with the "**Vector Tool**" on Sketch. The main goal is to convert current **shapes** (hard to animate) into **paths** :



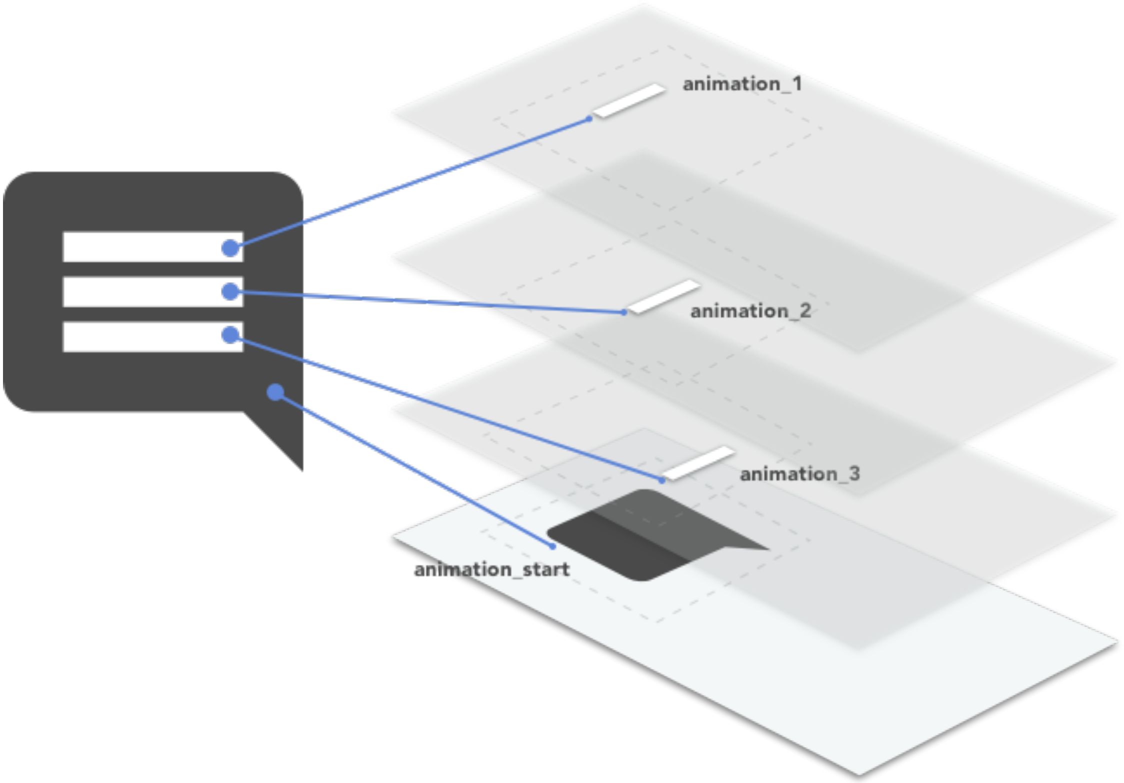
Reproducing each line with a Path

The real difficulty here is trying to **superimpose** the path you're drawing above the shape… Oh, and by default, path thickness is set to 1 : **set it to 2** to perfect fit the below line shape 👌

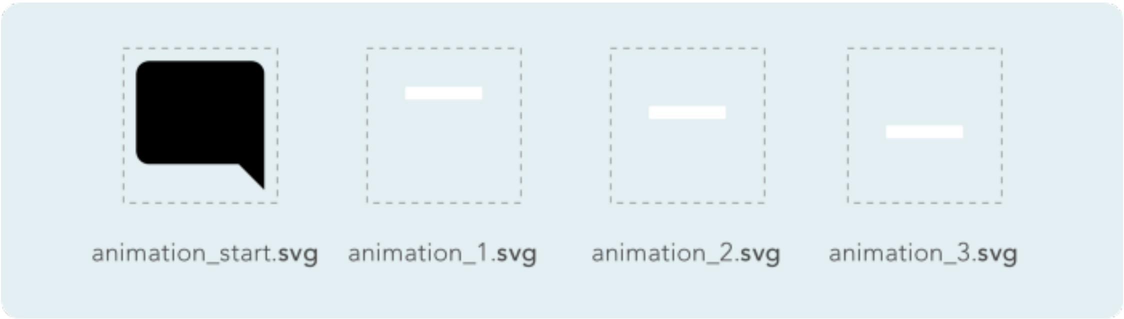
After that, **delete** for all animation groups the old shape to only keep the path.



I know, it's super tricky ! Actually, try to imagine you are a **cartoonist** who wants to animate its cartoon character using **superimposed transparent** papers… That what we try to achieve here.



Finally, **export** those 4 SVG images to your desktop, like this :

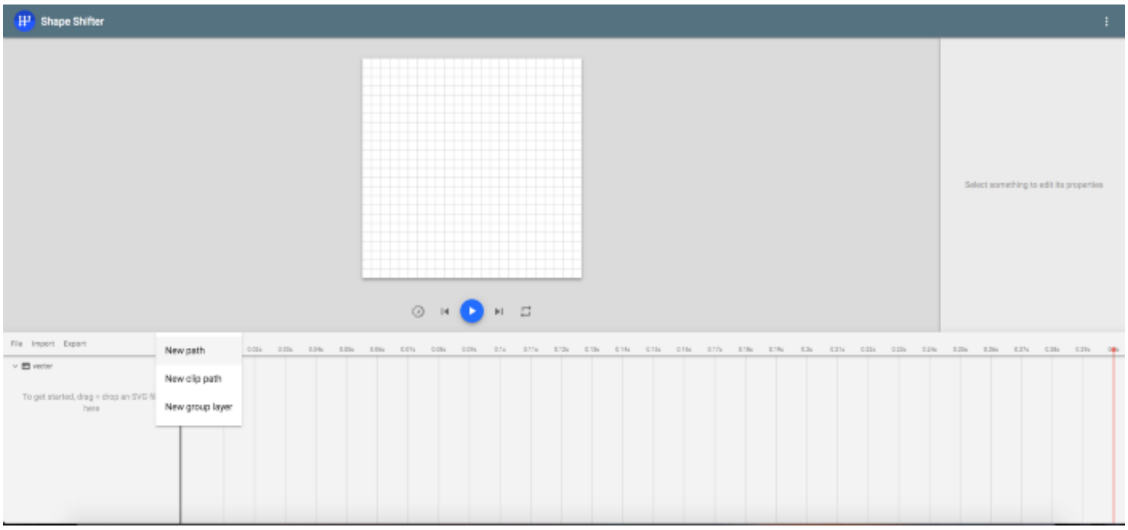


You will soon understand why we've prepared all this stuff… 😉

3. Preparing the icons

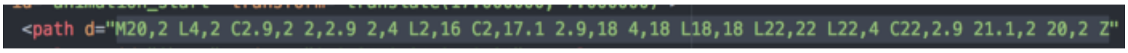
In order to animate our icon, we will use Shape Shifter created by the so talented Alex Lockwood. This tool saves us a lot of time !

First of all, go to Shape Shifter website and create a **new path** :


Shape Shifter
File
Import
Export
vector
To get started, drag + drop an SVG file here
New path
New clip path
New group layer
Select something to edit its properties

Creating a new path

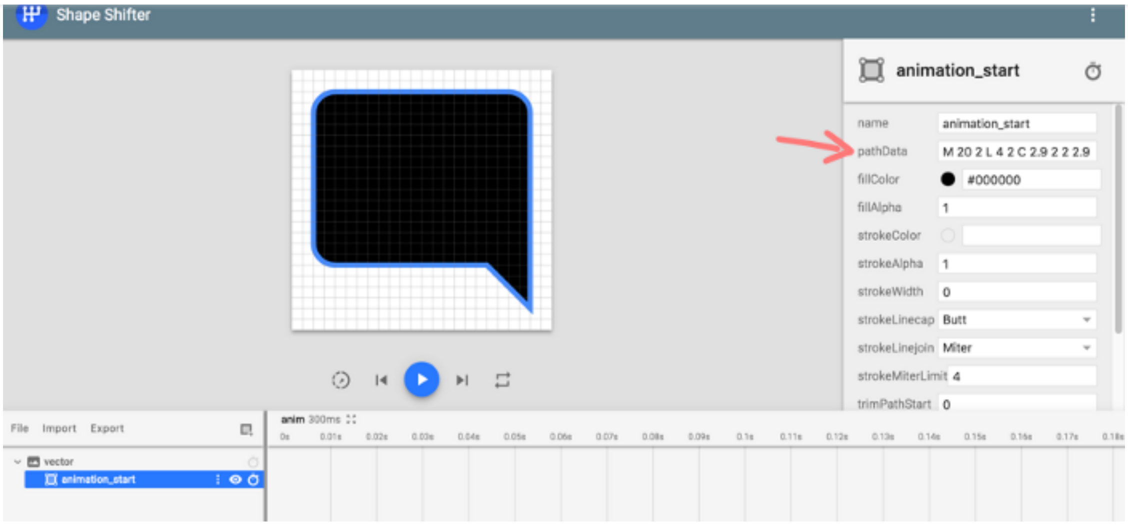
Then, open ***animation_start.svg*** with any code editor, and **copy** the **d=** value :



```
<path d="M20,2 L4,2 C2.9,2 2,2.9 2,4 L2,16 C2,17.1 2.9,18 4,18 L18,18 L22,22 L22,4 C22,2.9 21.1,2 20,2 Z"
```

Raw SVG code from **animation_start.svg**

Next, **past** it to the previously created path in the *"pathData"* field. Fill the *"name"* and *"fillColor"* fields too.



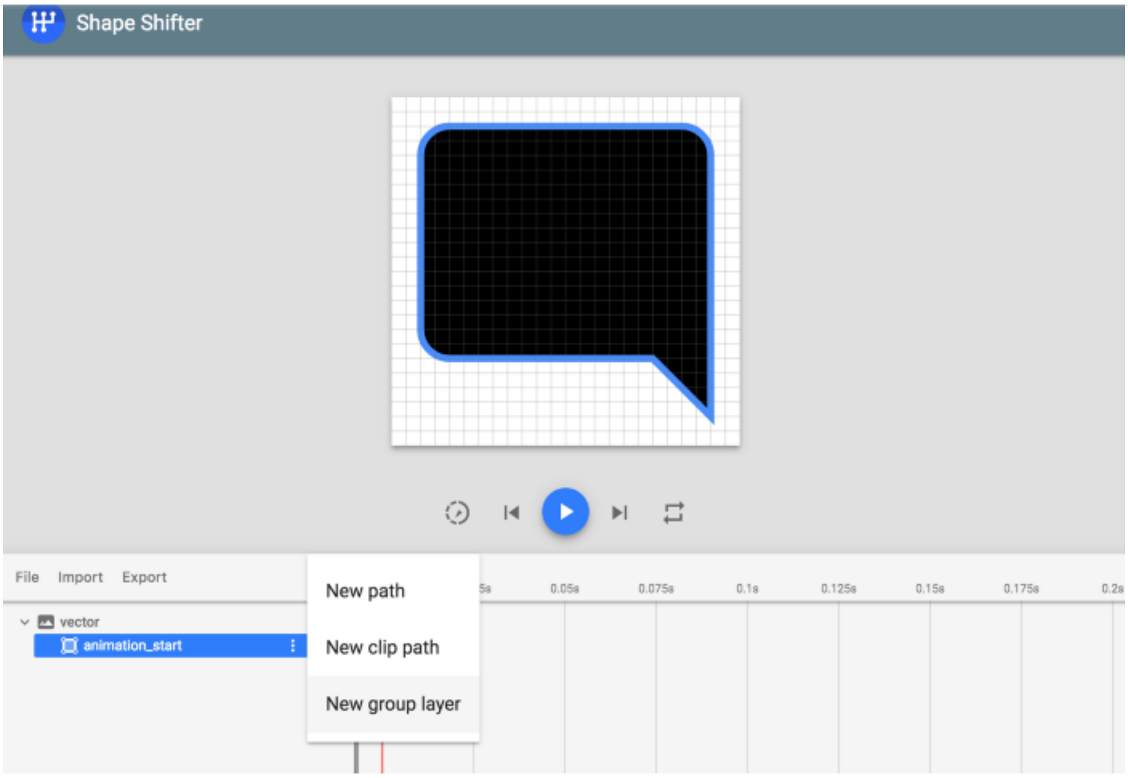
animation_start

name animation_start
pathData M 20 2 L 4 2 C 2.9 2 2 2.9
fillColor ● #000000
fillAlpha 1
strokeColor ○
strokeAlpha 1
strokeWidth 0
strokeLinecap Butt
strokeLinejoin Miter
strokeMiterLimit 4
trimPathStart 0

Copy SVG properties to path fields

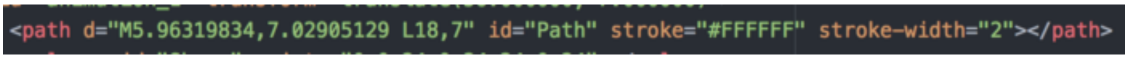
Great ! 🎉 Our first animation image (***animation_start.svg***) is correctly shown. Now, let's add the other images (***animation_1.svg***, ***animation_2.svg*** **&** ***animation_3.svg***) to create the full animation.

Add a new "**group layer**" :



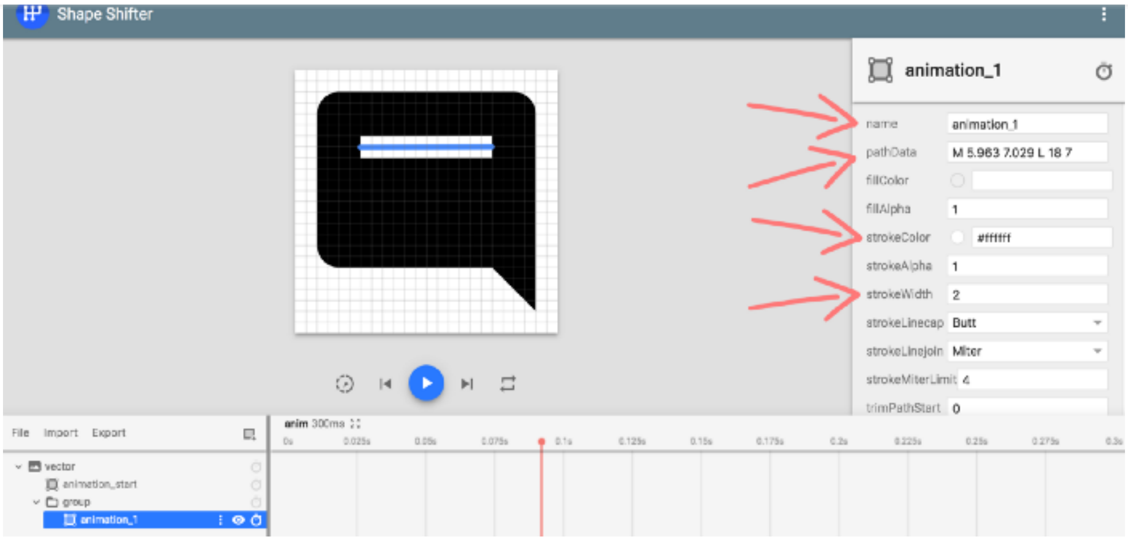
Add a “New Group Layer”

Finally, add inside it **three** new **paths** containing our previous SVG files (***animation_1.svg***, etc…) data.



```
<path d="M5.96319834,7.02905129 L18,7" id="Path" stroke="#FFFFFF" stroke-width="2"></path>
```

SVG data for **animation_1.svg** file

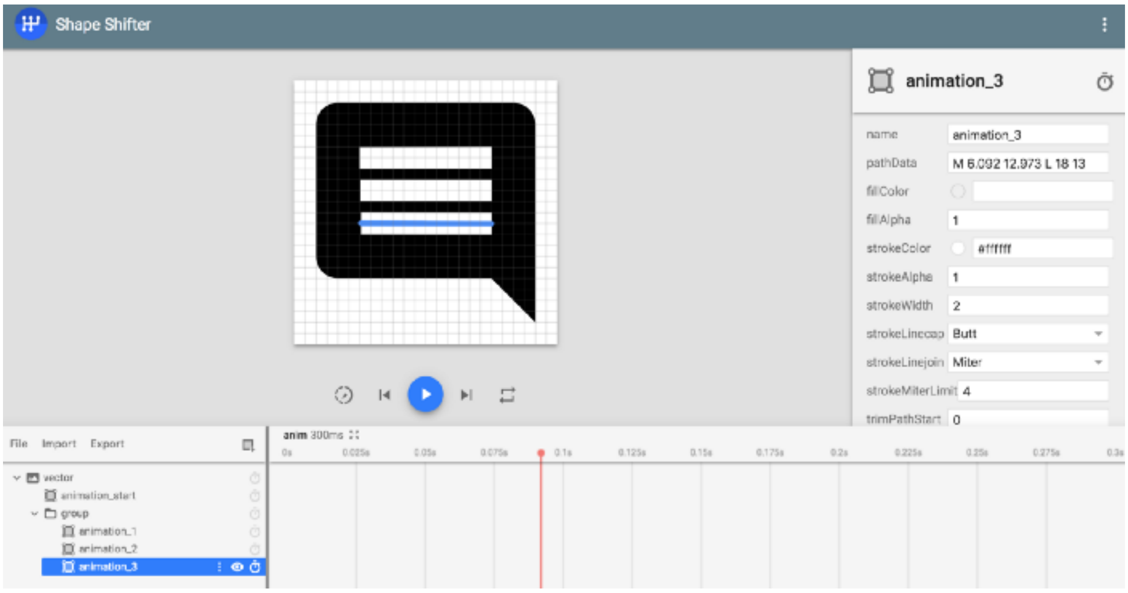


animation_1

name	animation_1	
pathData	M 5.963 7.029 L 18 7	
fillColor	○	
fillAlpha	1	
strokeColor	○	#ffffff
strokeAlpha	1	
strokeWidth	2	
strokeLinecap	Butt	
strokeLinejoin	Miter	
strokeMiterLimit	4	
trimPathStart	0	

New path from SVG data (**animation_1.svg**)

Repeat the operation for the **second** *(animation_2.svg)* & **third** *(animation_3.svg)* line. In the end, your screen must look like this :



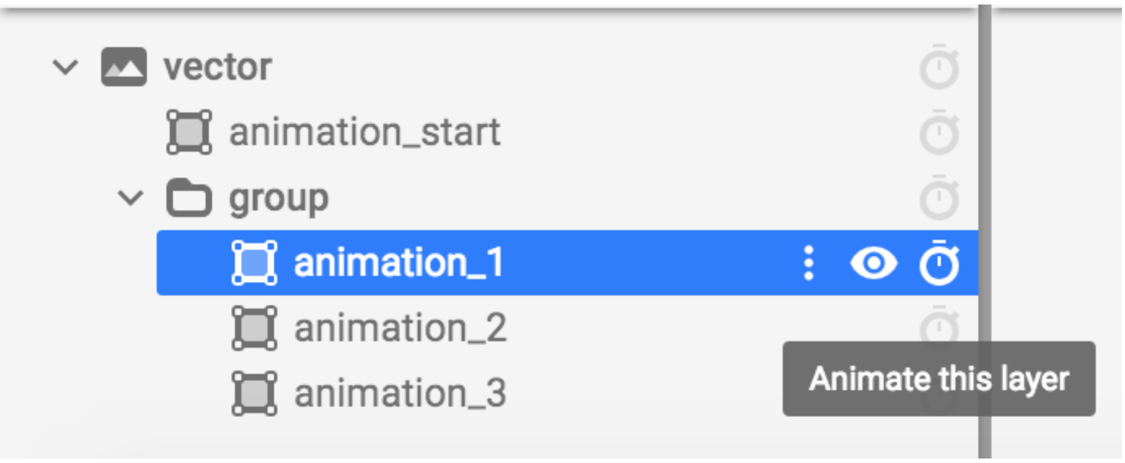
animation_3

name	animation_3	
pathData	M 6.092 12.973 L 18 13	
fillColor	○	
fillAlpha	1	
strokeColor	○	#ffffff
strokeAlpha	1	
strokeWidth	2	
strokeLinecap	Butt	
strokeLinejoin	Miter	
strokeMiterLimit	4	
trimPathStart	0	

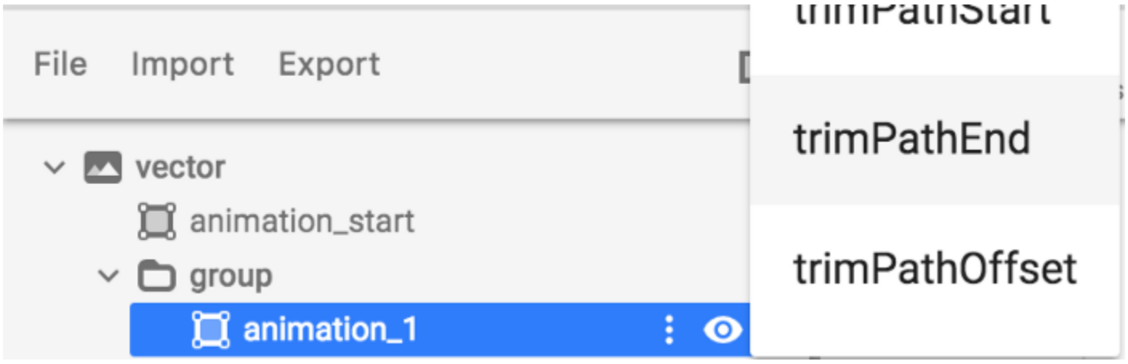
4. Animating the icons

Now, let's set up the fun part : the animations ! Remember, we want to **move the three lines**, *one after the other.*

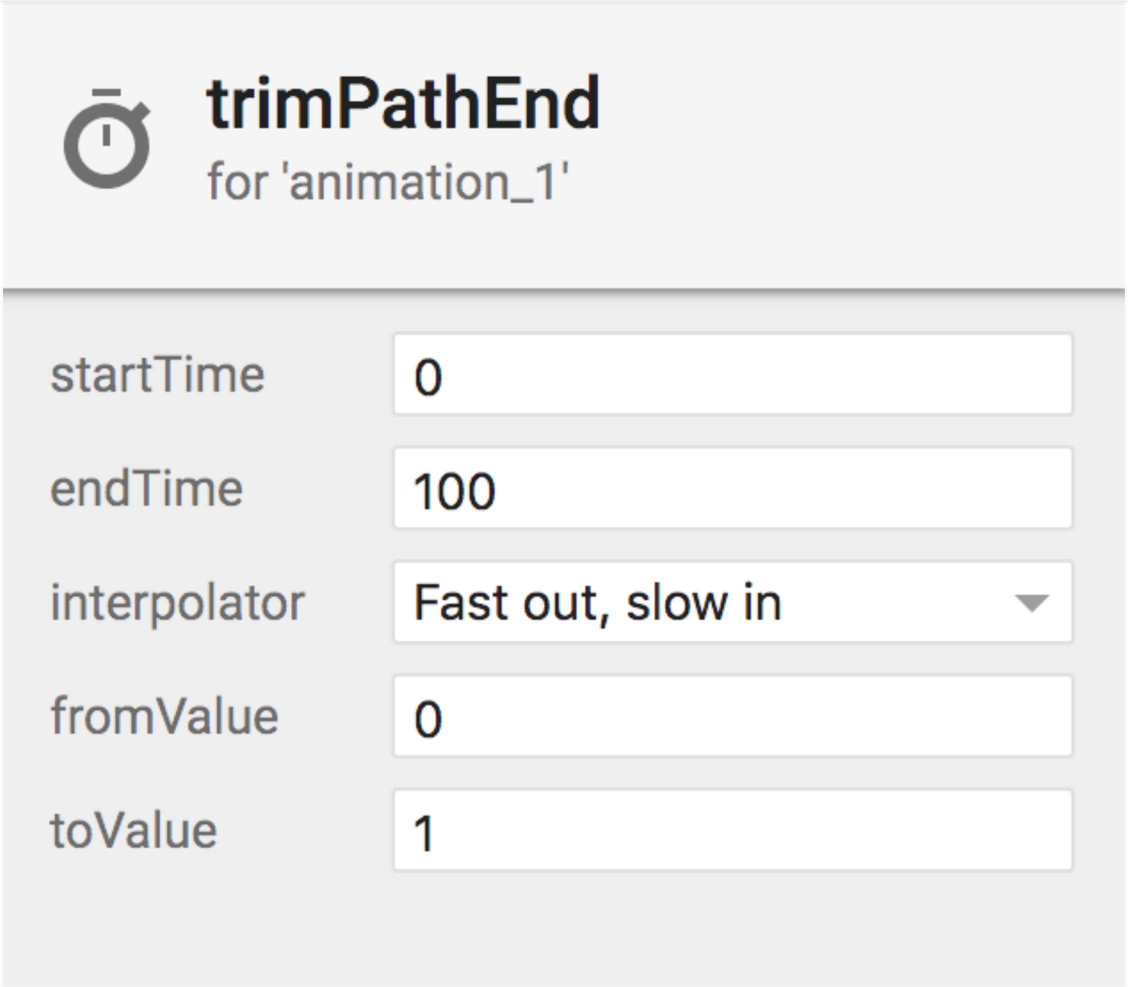
First, click on the clock icon beside the animation_1 :



Next, select “trimPathEnd” to animate the “progression” of the drawn path :



Configure the animation settings as below :



trimPathEnd
for 'animation_1'

startTime 0

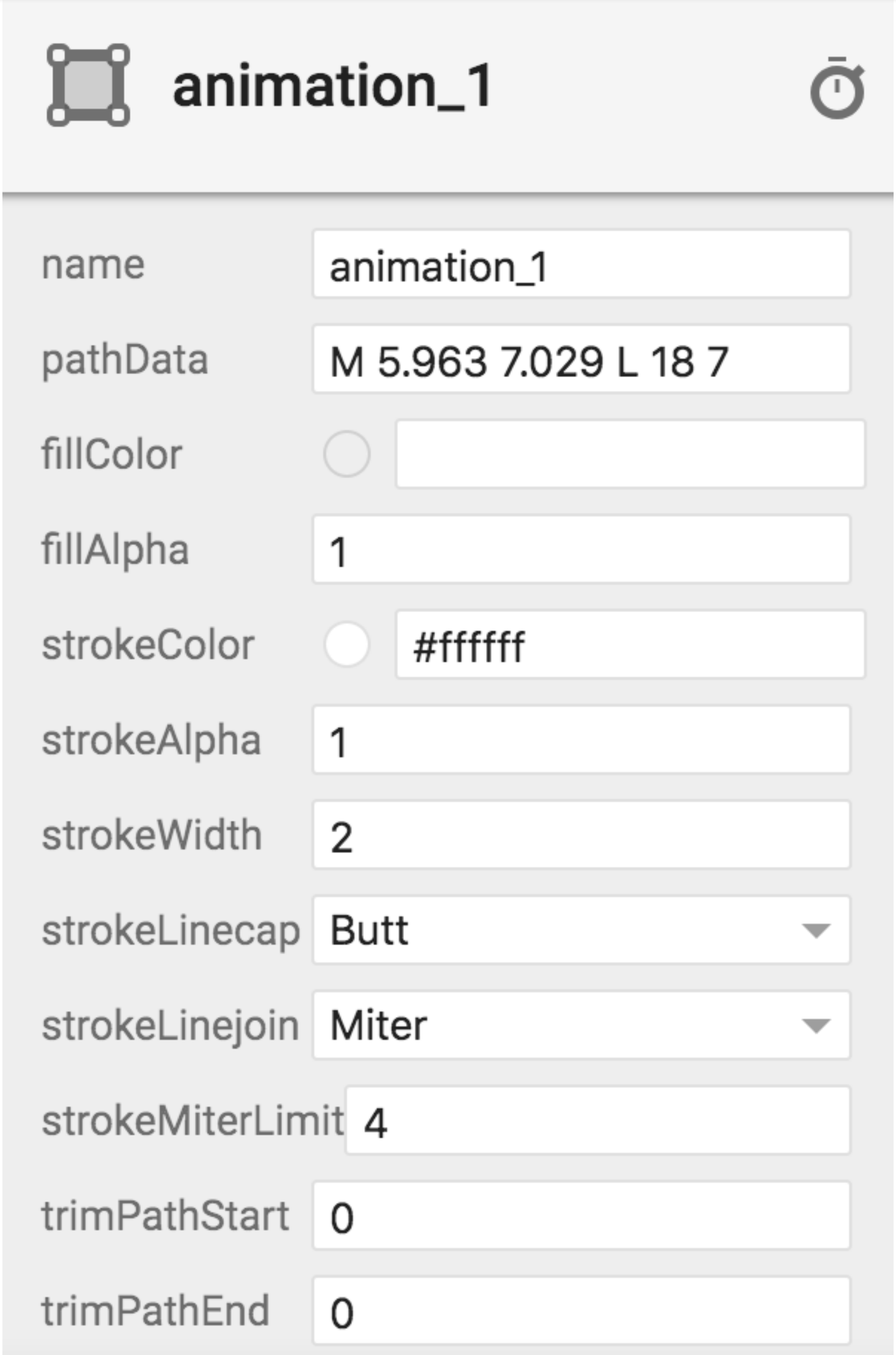
endTime 100

interpolator Fast out, slow in

fromValue 0

toValue 1

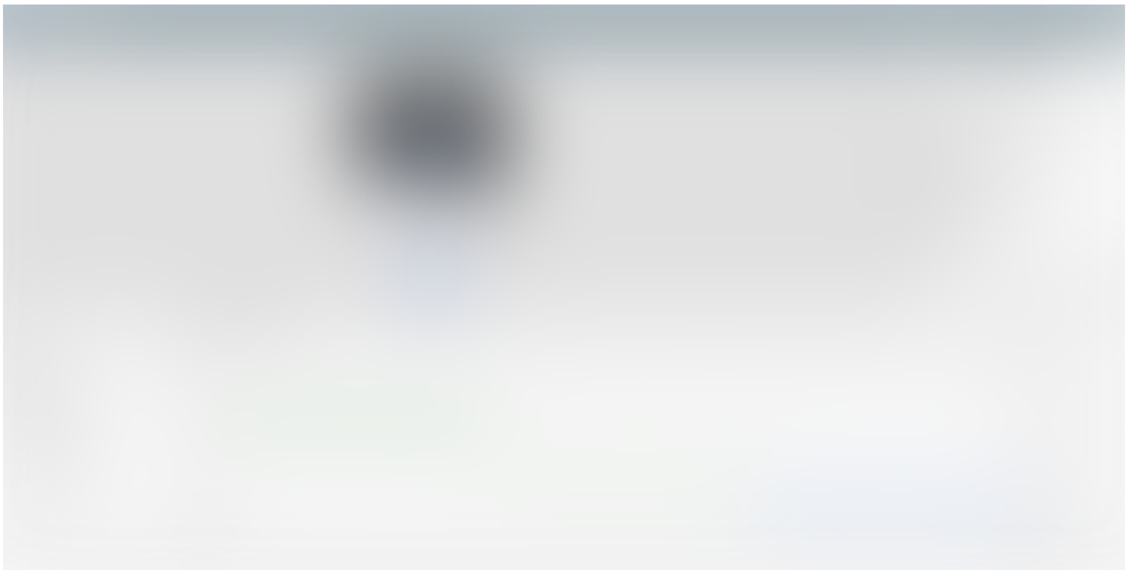
And finally, set the initial value of property “**trimPathEnd**” for **animation_1** to 0, forcing the path to be hidden when the animation starts :



animation_1

name	animation_1
pathData	M 5.963 7.029 L 18 7
fillColor	○
fillAlpha	1
strokeColor	○ #ffffff
strokeAlpha	1
strokeWidth	2
strokeLinecap	Butt
strokeLinejoin	Miter
strokeMiterLimit	4
trimPathStart	0
trimPathEnd	0

That's all ! Repeat this operation for *animation_2* & *animation_3*. Your screen should look like this :



Then, you just have to run the animation by clicking on the "Play" icon…

What's next ?

- An Introduction to Icon Animation Techniques by Alex Lockwood
- Creating AnimatedVectorDrawables with Shape Shifter by James Williams

Android Drawable Design SVG Android Animations