Super Animation Converter 2.0 Help Doc

Raymond Lu(Raymondlu1105@qq.com)

2013-02-23

Intro

Super Animation Converter is a simple SWF file(a file format supported by Adobe Flash Player software) converter. It can parse and extract transformation data from SWF file. With these transformation data and the right images, you can reproduce the animation in any game engine just as Adobe Flash Player can do.

The native animation solution in Cocos2d game engine is “frame by frame” animation, which means if there are 20 frames in your animation, you need 20 pieces of image(texture), and draw these images one by one on the screen to create animation effect. There are two flaws in this solution:

1. High resource consuming, one frame one image;
2. Hard to make smooth animation, since no interpolation between frames, which means it “jumps” directly from the current frame to the next frame.

So I try to provide a better animation solution which is based on Super Animation Converter for Cocos2d game engine. The idea is very simple:

1. Use Flash as your animation editor, create any animation you want in it, then export these animation as SWF file.
2. Parse the SWF file with Super Animation Converter, get the transformation data and images from the SWF file.
3. Reproduce the animation in Cocos2d game engine with the transformation data and images.

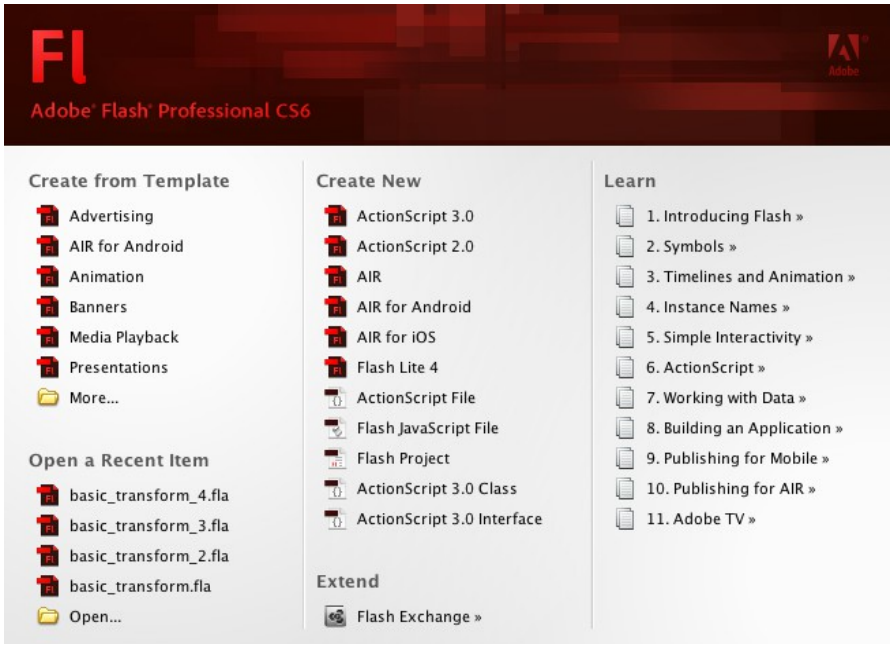
I will introduce rules & steps you need to follow when creating animation in Flash, and source code files need to be imported to the existed Cocos2d game engine in the next chapter.

Rules & Steps in Flash

I assume that you have the basic knowledge about Flash, you should know some key concept, like Stage, Layer, Key Frame, Symbol and Classic Tween, etc.

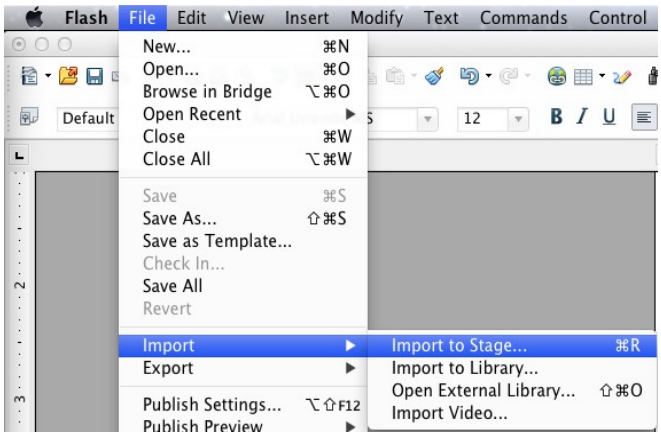
Before you start to create you animation, one key rule you should know: Super Animation Converter bases on **Graphic Symbol** in Flash, which means **Graphic Symbol** is the only symbol you can place on the stage. **Button** & **Movie Clip**, any V**ector Graphics** are NOT supported by Super Animation Converter.

Open the Flash, my version is CS6. Since we only use the basic function in Flash, you can use any old version you have. Create new ActionScript 3.0 file, save it as “basic_transform.fla”.

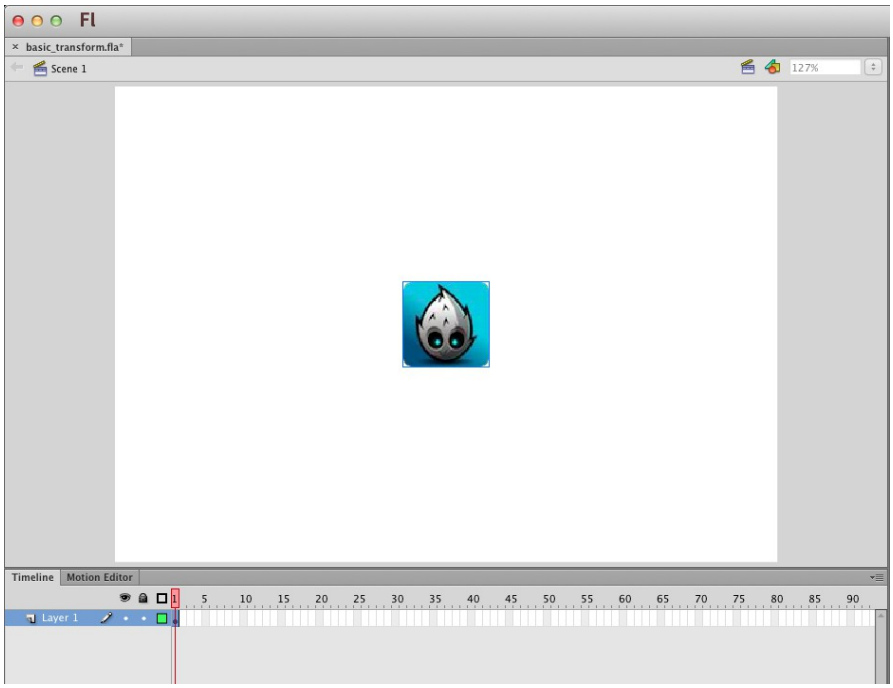


Let's import the little blue logo of Cocos2d-x into the editor, select “File/Import/Import to Stage”.

Please remember that we only support PNG image right now.



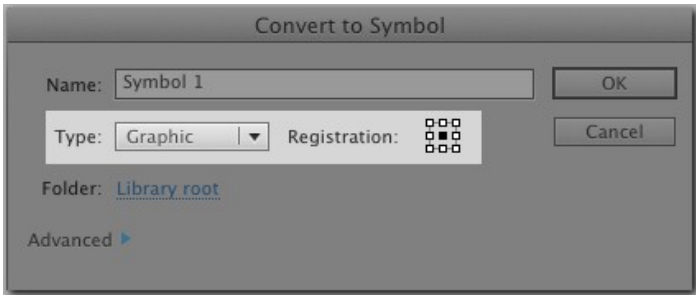
Now you can see the little blue logo in the stage.



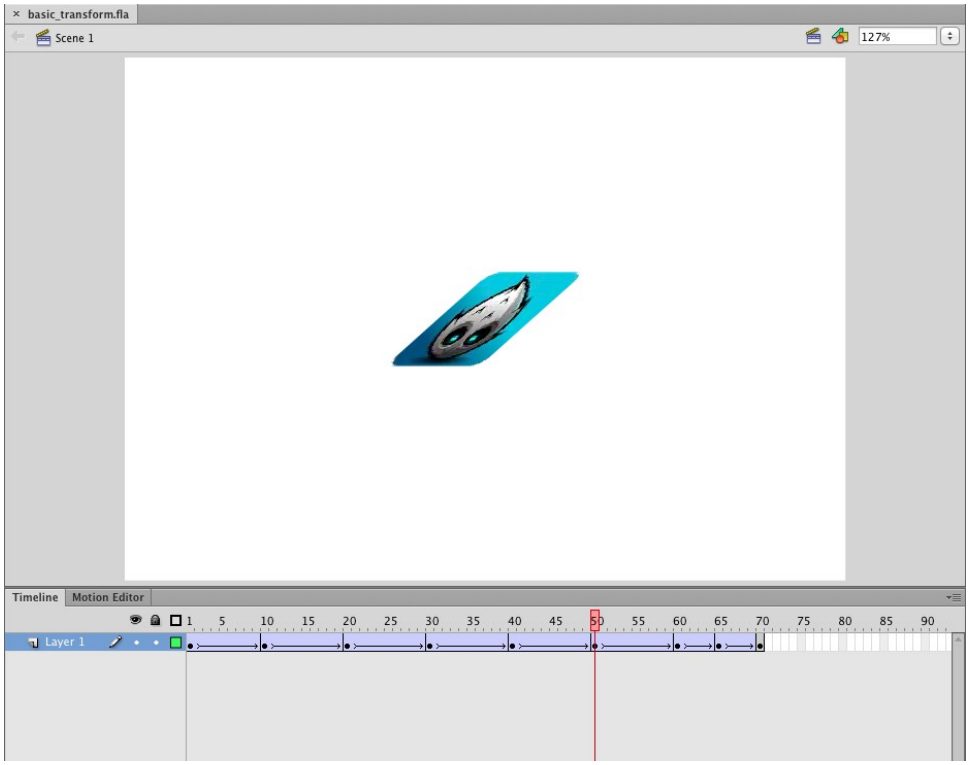
Remember, we only support **Graphic Symbol**. So you need to convert the little blue logo to graphic symbol firstly. Select the logo in the stage, right mouse button, select “Convert to Symbol...” on popup menu.

In the “Convert to Symbol” dialog box, please make sure:

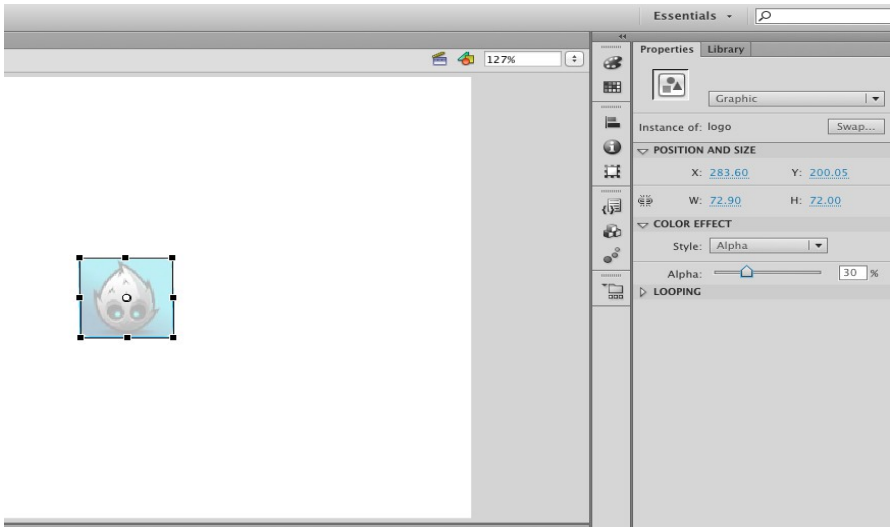
1. Type must be Graphic;
2. Registration must be center. I always assume that the origin of symbol is located at the center. If the transformation of your animation is wrong, maybe because the registration is not center.



Once you have converted the image into Graphic Symbol, you can apply any animation effect on it. Let's add some basic transformation to the logo which includes scale up & down, skew, rotation and translation.

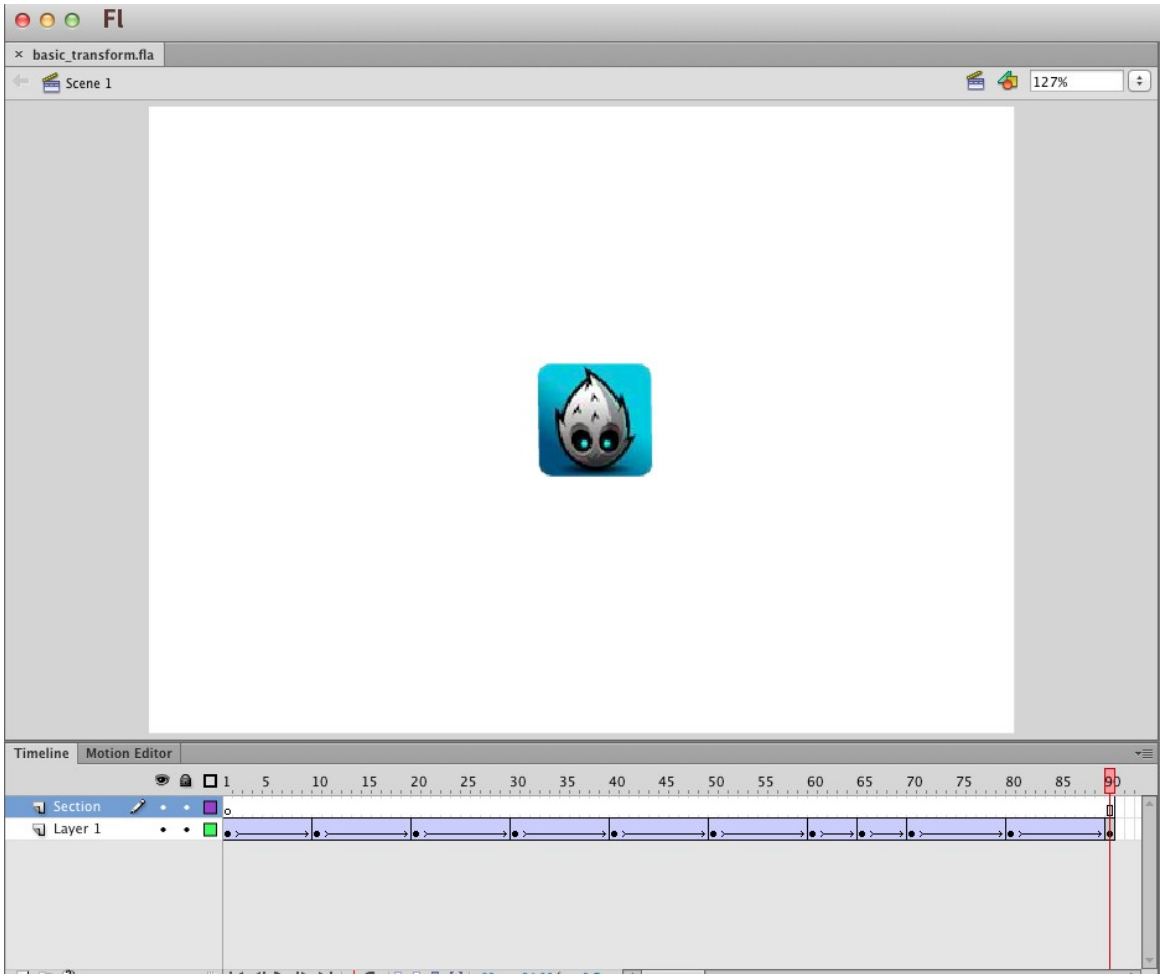


Super Animation Converter also support Alpha effect. So let's add fade out/in effect for the little blu logo.

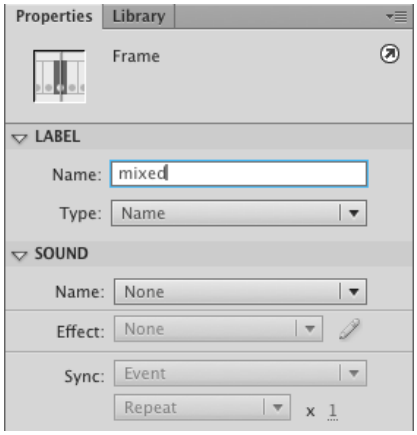


OK, our animation seems ready now. But let's assume a scenario: a character may have multiple actions, like idle action, battle action. We want to play “battle” action when character is battling, otherwise play “idle” action. Usually, artist like to create all these actions in one animation file. So in order to separate animation, Super Animation Converter introduces “**Animation Section**”.

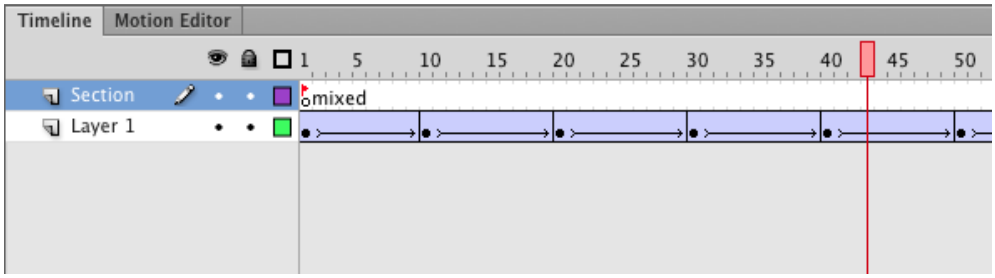
Create a new layer, rename it as “Section”, and place it on the top. Super Animation Converter use this layer to separate animation.



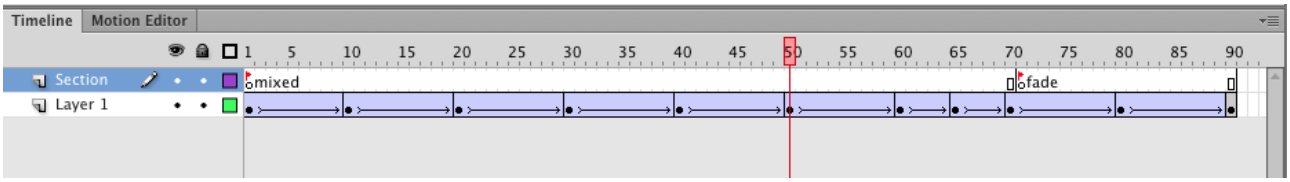
“Section” layer should not include any drawable item. We use “**Label**” property of “**Blank Keyframe**” to specify the start of an animation section. Since we already have one blank keyframe at the 1st frame of “Section” layer, just select it, name it as “mixed”.



Enter, you will see that there is a little red flag and a name “mixed” on the 1st frame, which means you have added the section label successfully.

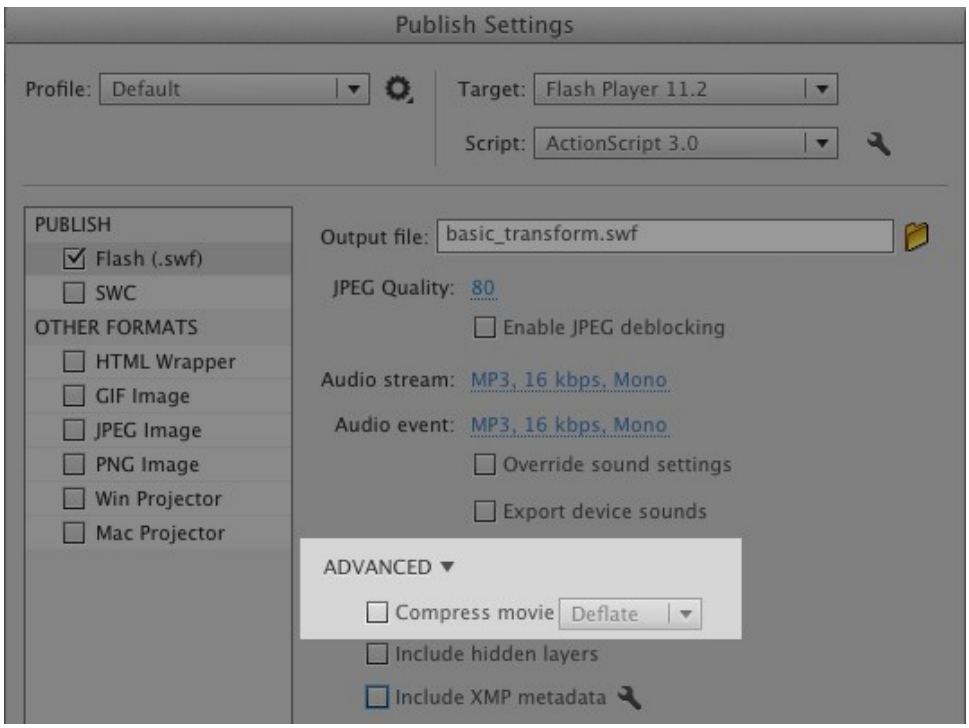


Let's separate the second section. Assume that the second section begins at the 71st frame, select the 71st frame on the “Section” layer, right mouse button, the select “Insert Blank Keyframe” on the popup menu. Then edit the label name of the new added blank keyframe, name it as “fade”.

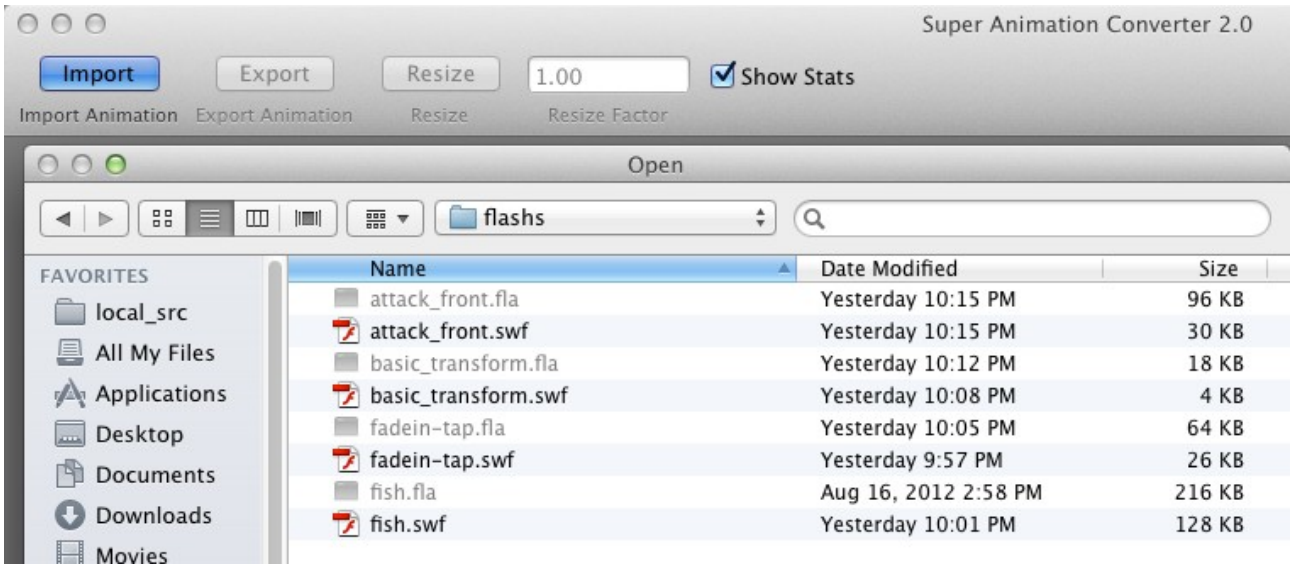


Now we've separated the animation into two sections. The first section named “mixed”, which begins at the 1st frame, ends at the previous frame of “fade” label. The second section named “fade”, which begins at the frame of “fade” label, ends at the tail of the whole animation.

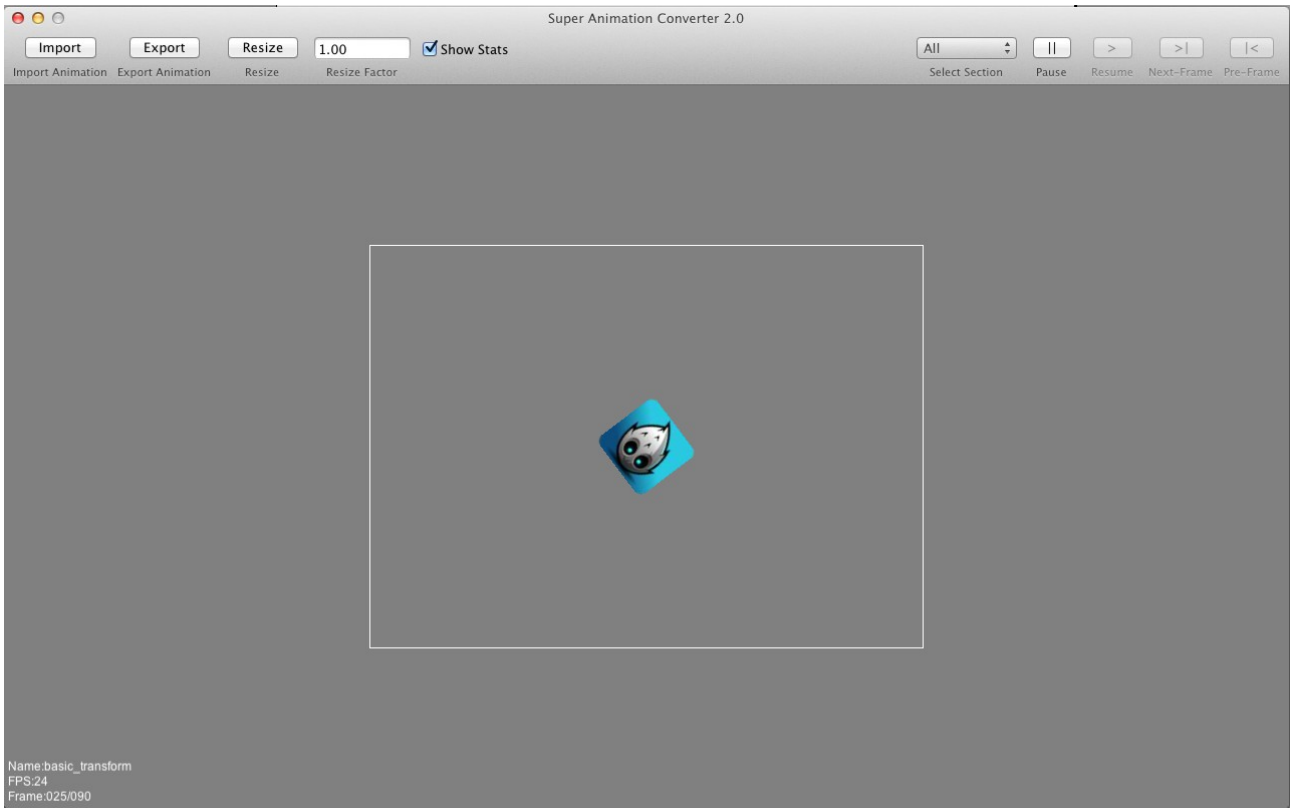
Ok, everything is done, it's time to publish our animation as SWF file. Select “File/Publish Settings”, make sure on thing in the “Publish Settings” dialog box: “Compress movie” is NOT selected in “ADVANCED”. Then click “Publish” button, a new file named “basic_transform.swf” will be created in the folder of “basic_transform.fla”.



Run Super Animation Converter, click “Import”, import “basic_transform.swf”.



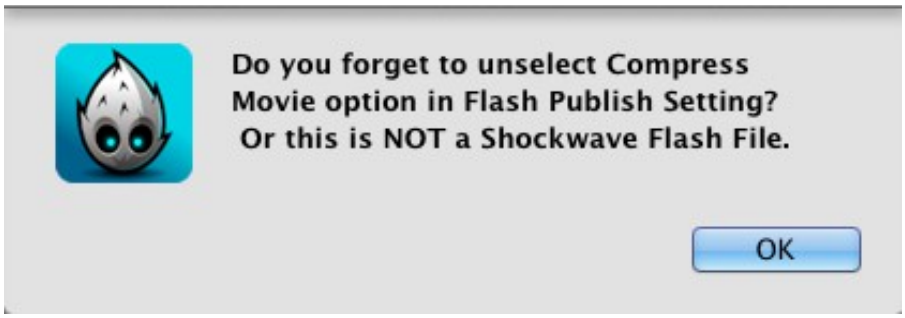
If everything goes well, you can see the animation start to run.



Let's have a quick look at the Super Animation Converter. On the left bottom corner, there are some stats data, includes animation file name, FPS and the current frame number, etc. On the right top corner, there are some operation buttons and a pop-up list named “Select Section”. Click the “Select Section” pop-up list, you can see all the animation sections. Super Animation Converter play the whole animation repeatedly by default. You can select any animation section to play in the pop-up list.

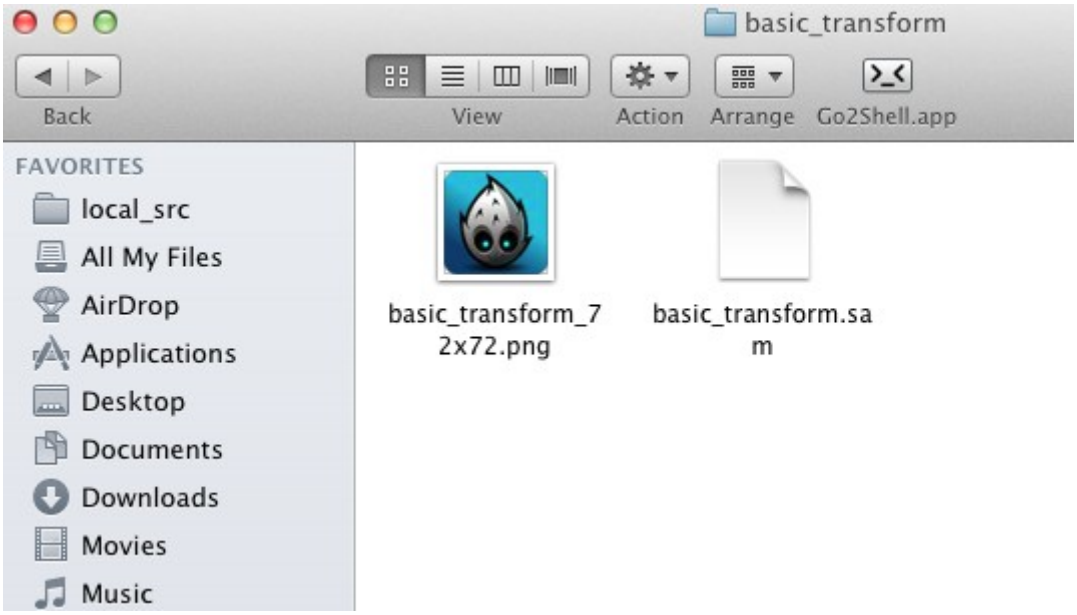
I think you will also notice that there is a white frame on the grey background. This white frame stands for the stage dimension in the Flash. And I will use it as the content size of "SuperAnimNode" in the cocos2d game engine.

If something goes wrong after you click "Import" button, Super Animation Converter will you, like



Which indicates that you didn't unselect "Compress movie" in the "Publish Settings".

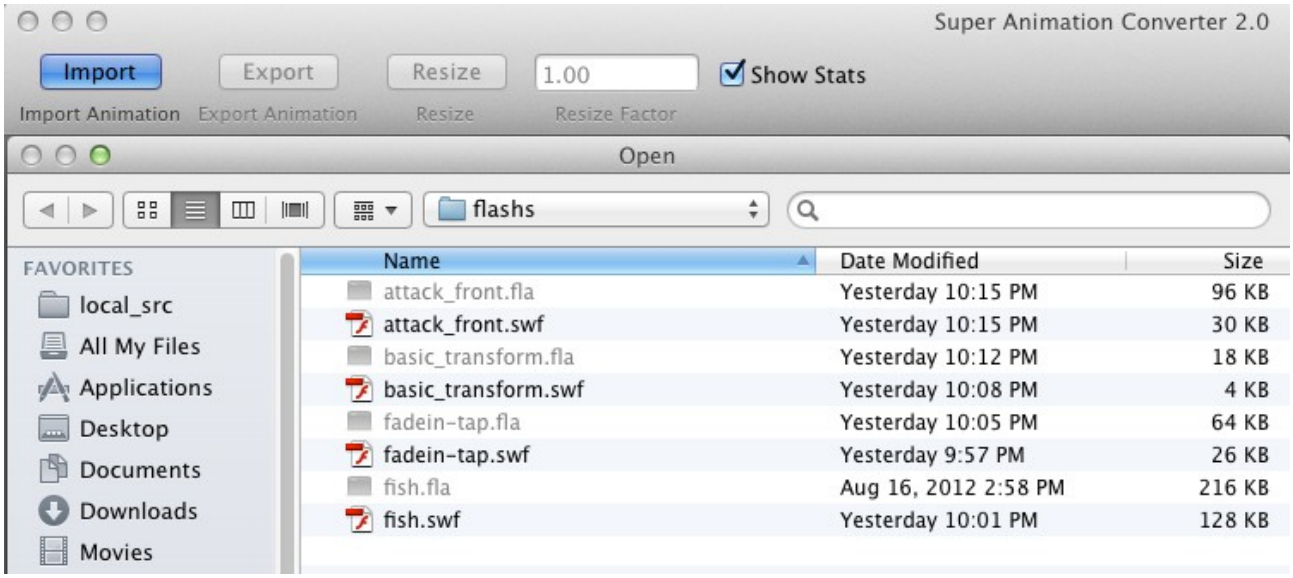
Ok, if everything goes OK, it's time to export the animation. Just click the "Export" button, then select a folder.



As you see in the previous screenshot, we only have one image in the output since only one image is used in the animation. Besides, we have a **.sam**(short for Super Animation) file in the output, which includes transformation data of each frame in the animation. These two files are all we need in Cocos2d game engine. You may say that the animation is too simple. Yes, but you can do anything you want in Flash as long as you follow the rules I mentioned. You will see a complicate animation in my sample code, a fish, which can spit out some bubbles:-).

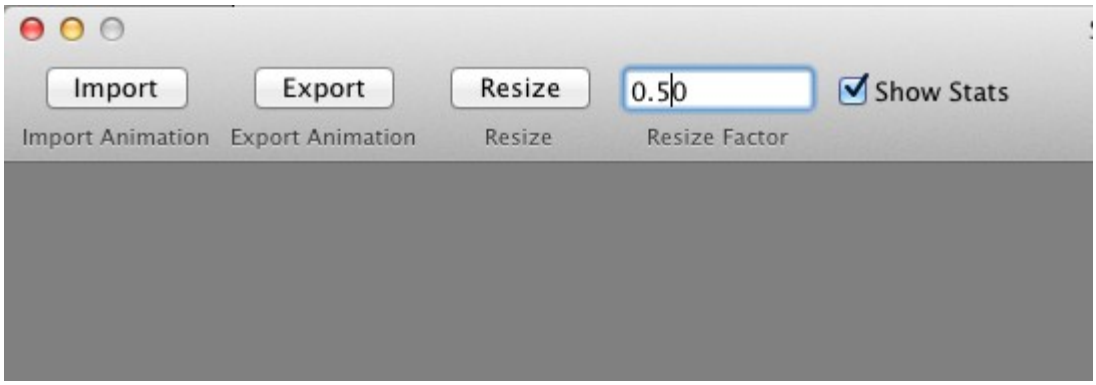
How to Resize Animation

1 Import the **SWF** you want to resize

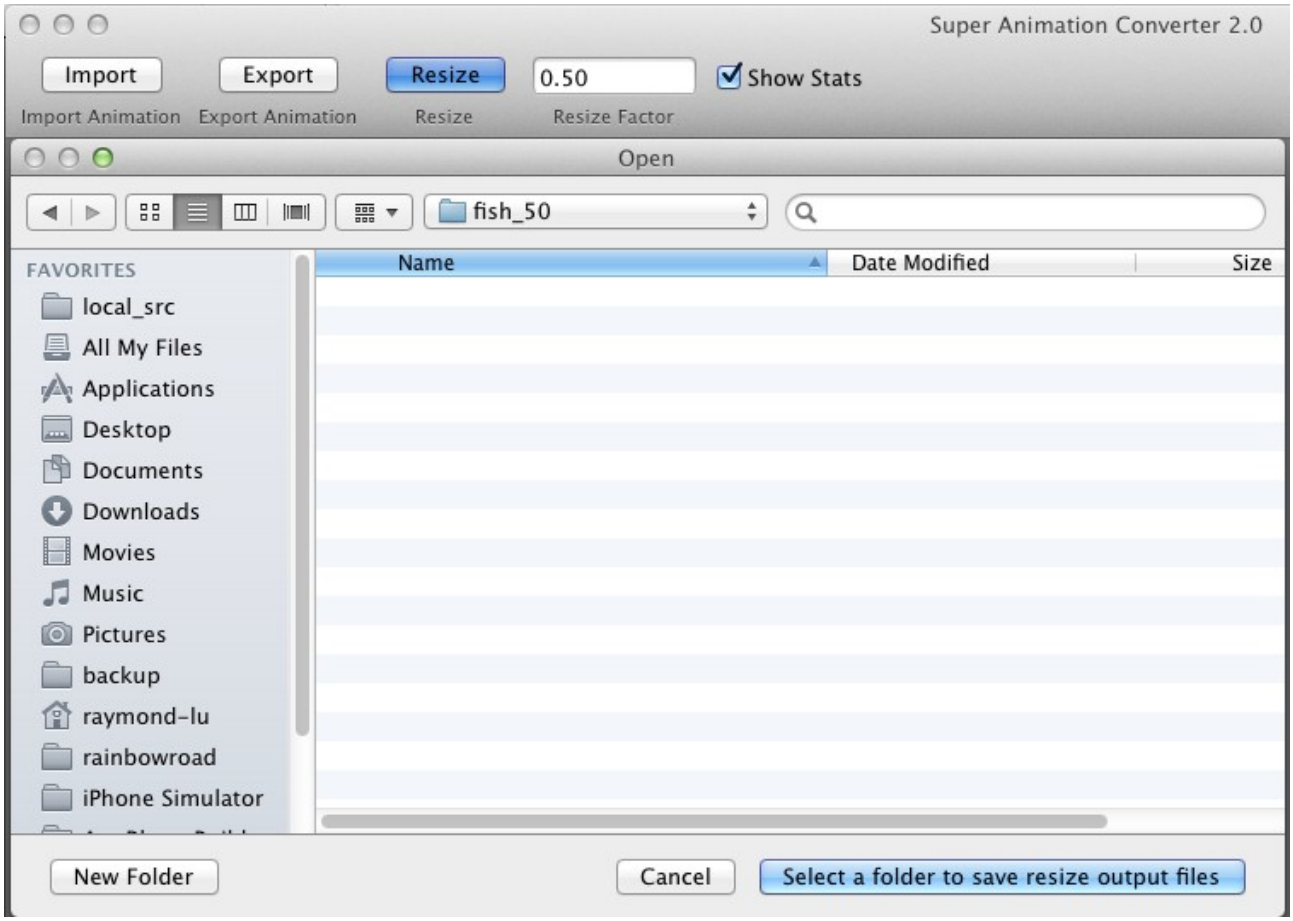


2 The Resize button & Resize Factor textfield will be enabled

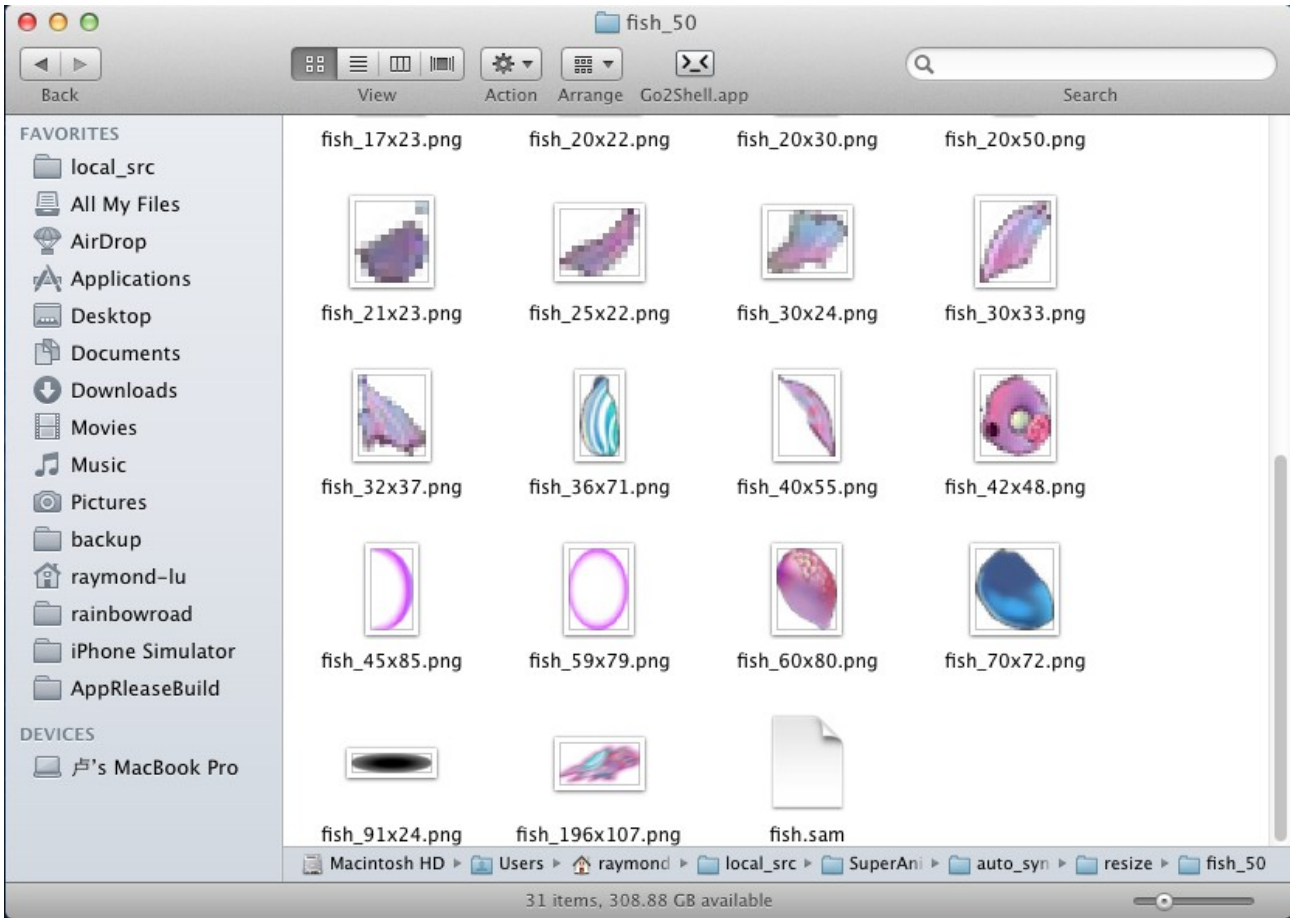
3 Input the Resize Factor you want to set to the animation, for example 0.50



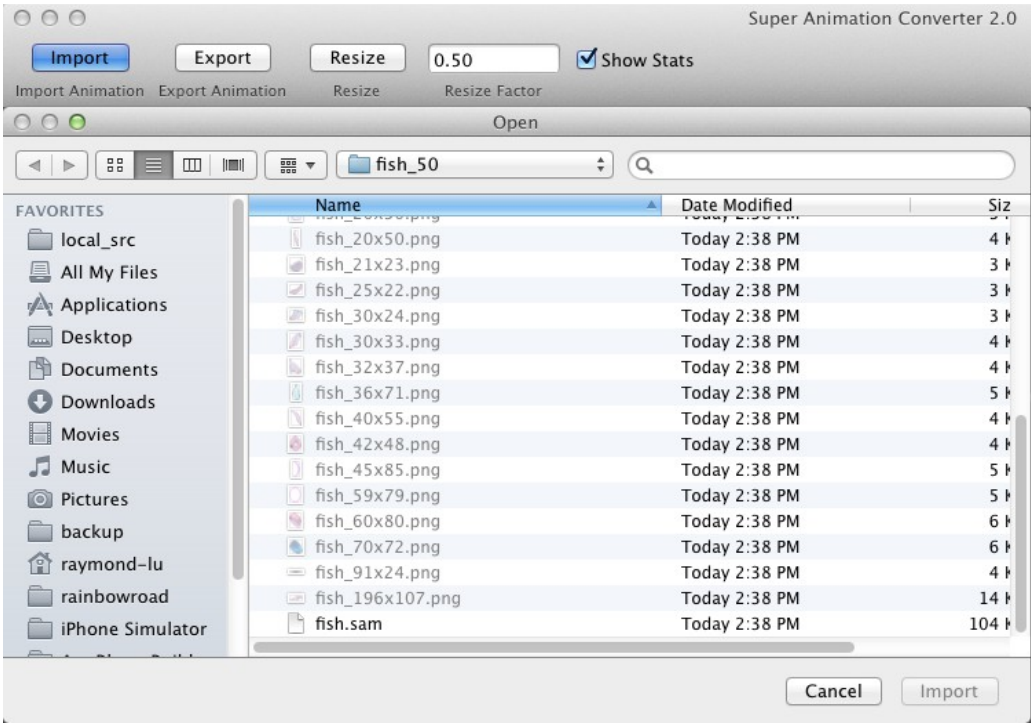
4 Press Resize button, you will need to select a folder to save resized animation



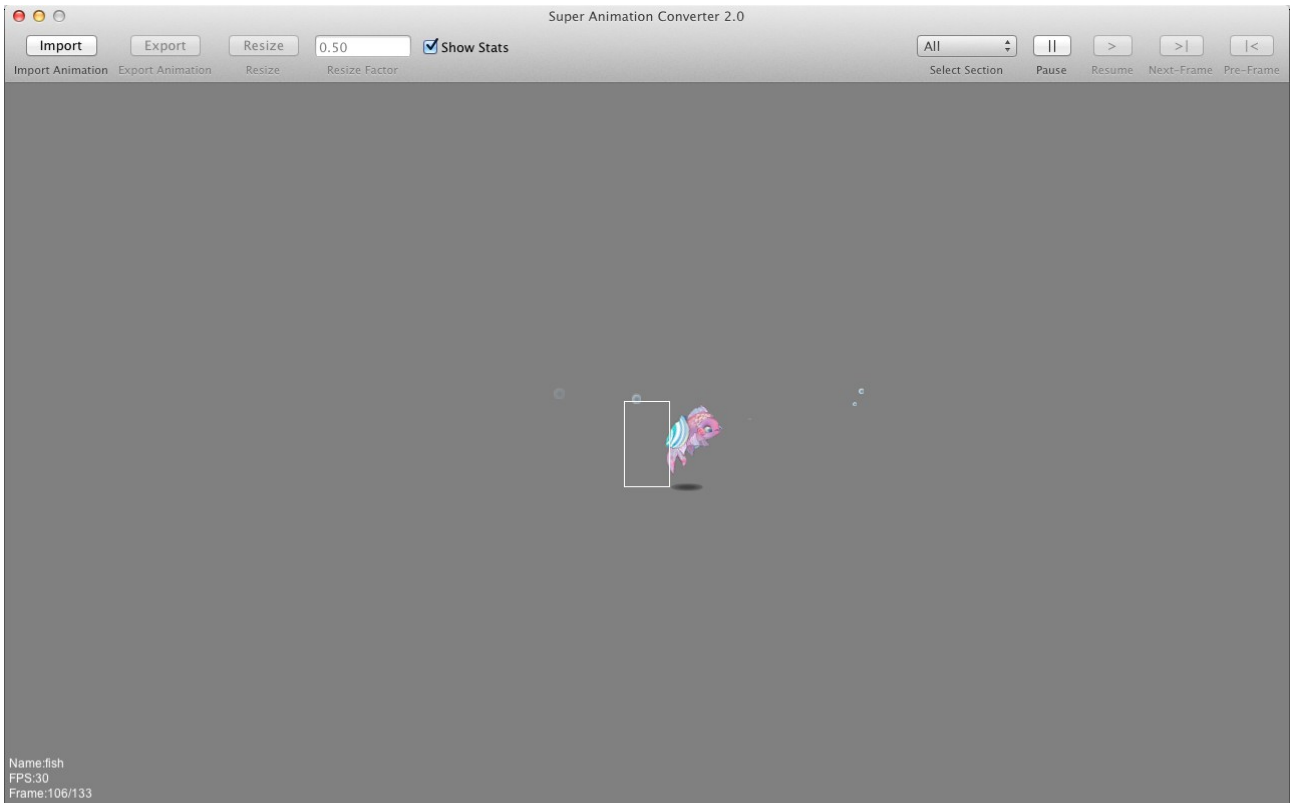
5 After a moment, go to the folder, you will see all images & sam file of the resized animation



6 If you want to check the resized animation, import it

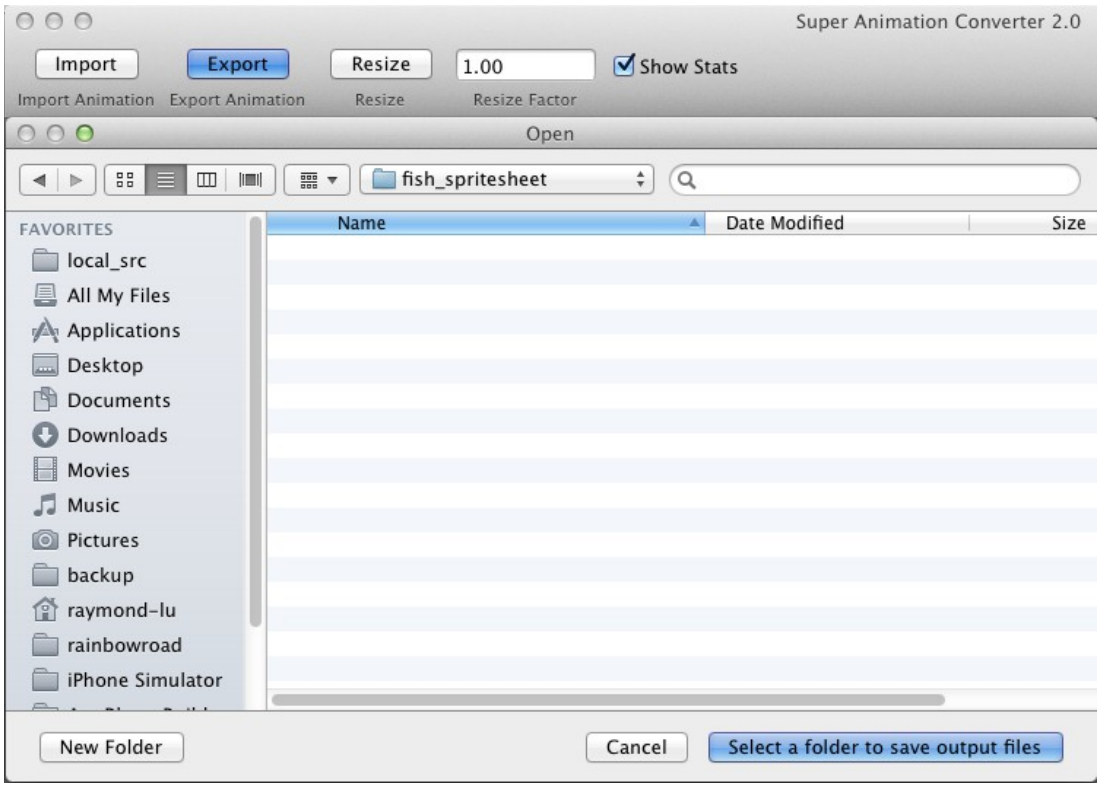


7 Here it is the small size of the fish animation



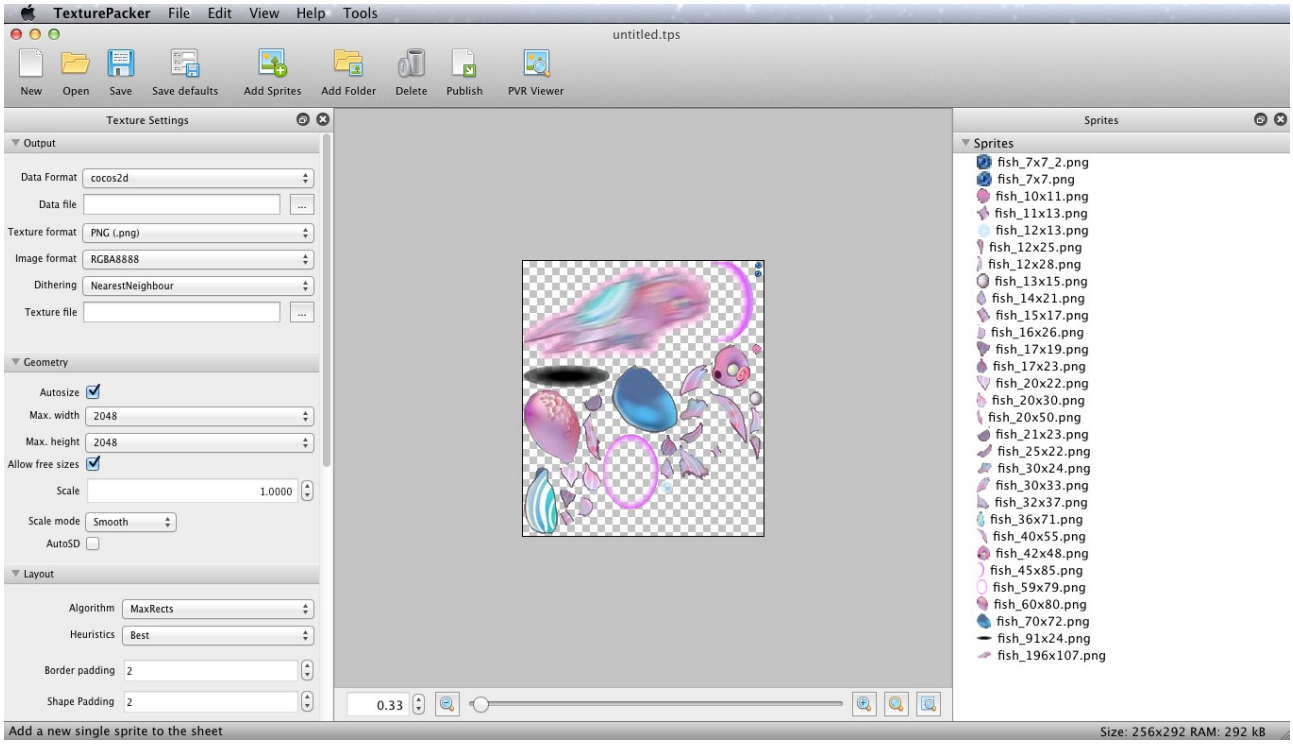
How to Use Sprite Sheet

1 Export the animation as usual

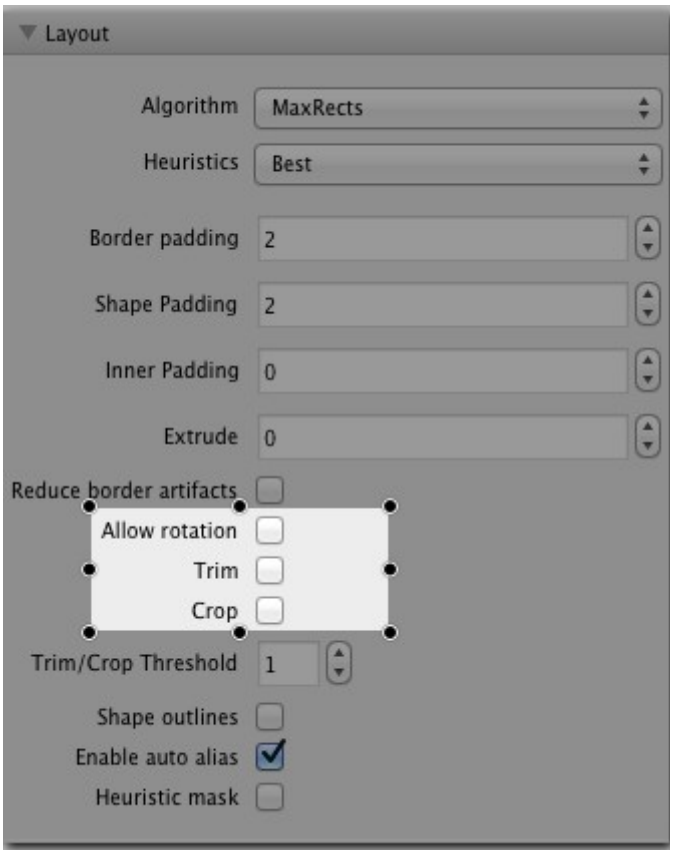


2 Close super animation converter

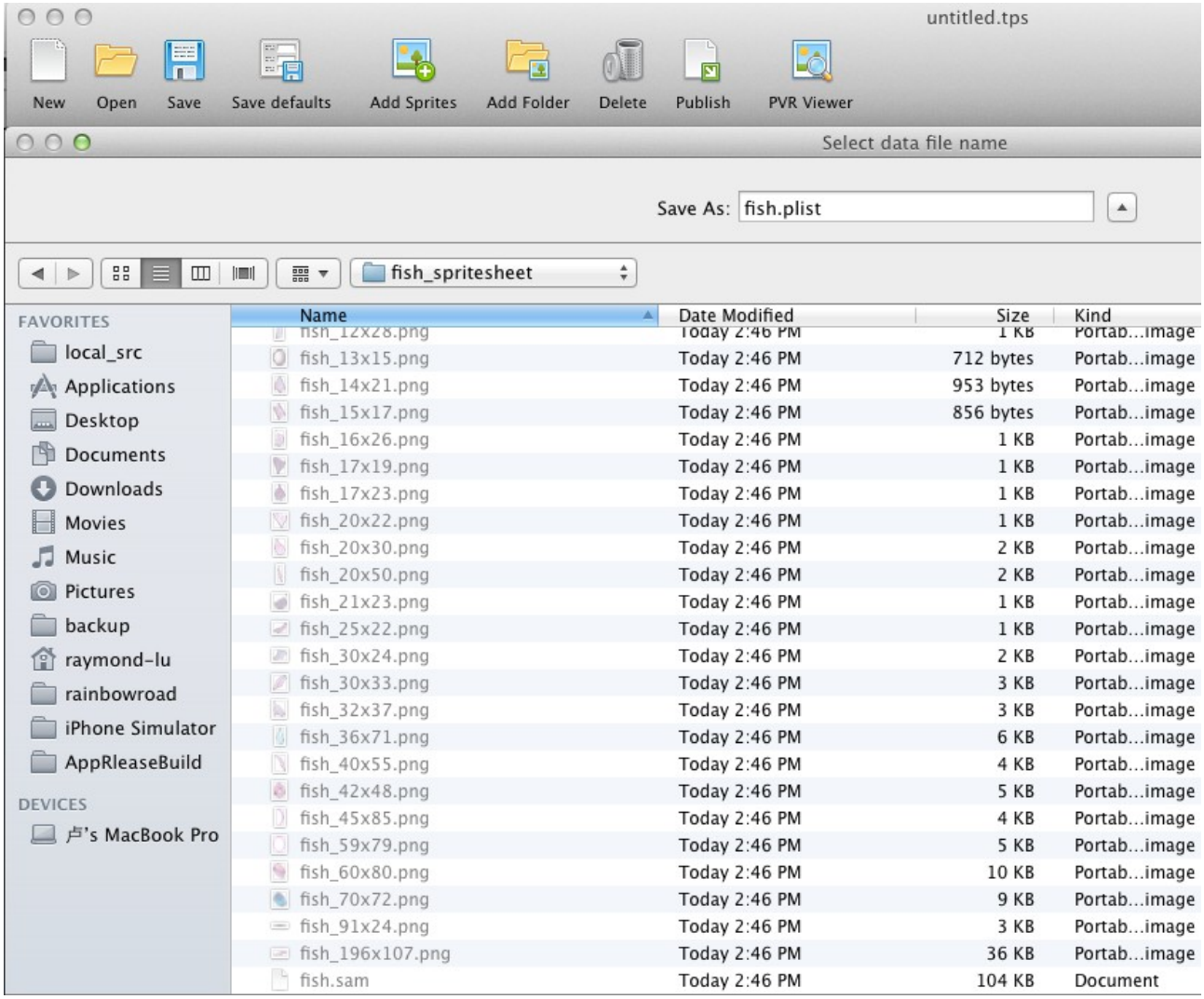
3 Open “TexturePacker”, add all images into one sprite sheet



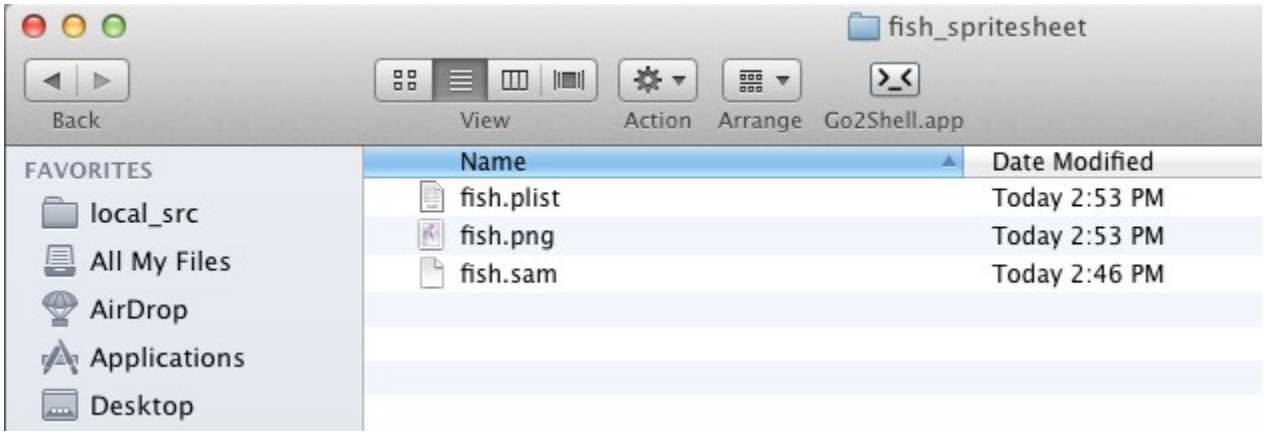
4 Please uncheck the “**Allow rotation**” in “Layout” before publish, **ATTENTION!!!!!!**



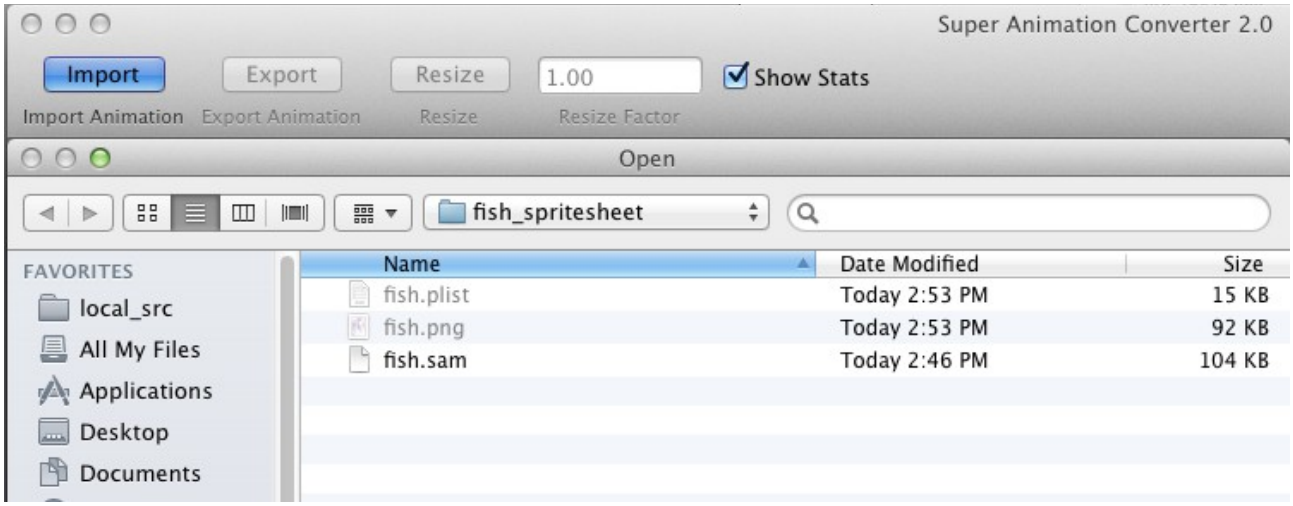
5 Publish the sprite sheet to the same folder as the animation, and the file name of plist **MUST** be the same as the sam file



6 Remove all other images files except the sprite sheet, DONE!



7 If you want to check whether it works, import that sam file into converter



8 The animation should be the same as before