

:: flashwebsite**intro**

This tutorial is a continuation of tutorial #7, and the final tutorial for the semester. This tutorial is split into two parts: the goal of first part is to create a custom animation that can play while the preloader is loading the contents of our site; the goal of the second part is to replace our “rollback” nav highlight with a more animated floating navback.

part 1: creating a custom preloader animation

In this part of the tutorial, we are going to simply customize the preloader animation. In tutorial # 6 we implemented a preloader that we found on the internet. I chose this tutorial because the code was clean and simple. However, the loading bar was a bit confusing. It didn’t animate like a tween, as we are used to seeing, but instead via an “onClipEvent” and a frame loop.

```
onClipEvent (enterFrame) {
    _xscale = _root.percentDone;
}
```

Instead of looking for a new preloader online that utilizes a tweened preloader animation, we are instead going to customize the existing preloader and add a tweened animation to it. There are many options for how to customize our existing preloader, but most stem from two basic techniques that I will go over in this tutorial. I will include several preloader animations in the .fla that will be posted on the class site and hopefully the range will demonstrate a range of possibilities. In the end, your preloader depends on your artistic vision and your creativity.

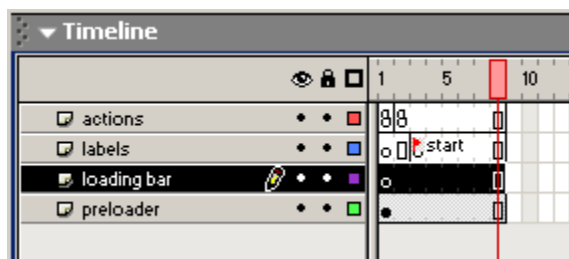
The two techniques for preloader customization using our existing preloader are the following:

- **The 100 framed tweened movie clip technique**
- **The nested looping movie clip technique**

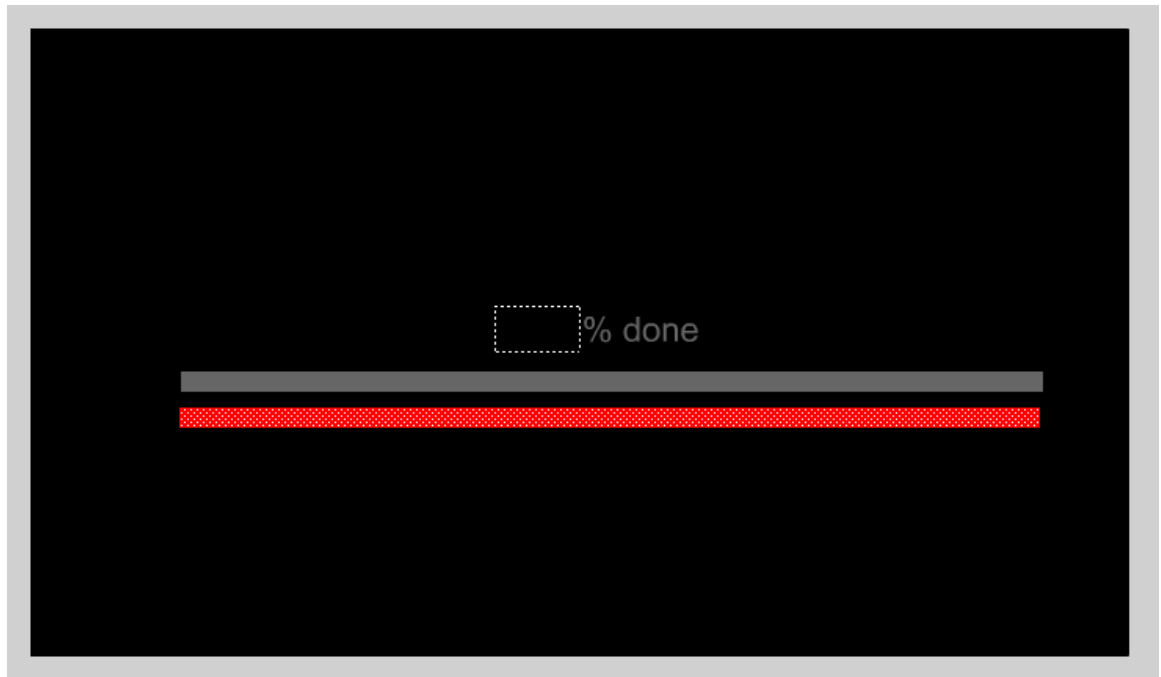
part 1a: the 100 framed tweened movie clip technique

In this first part of the tutorial – part 1a – we are going to redo the loading bar, BUT... instead of using an “onClipEvent” on a movie clip, we are going to do a simple tweened animation.

1. First, open up art257_f2005_tutorial_fs07.fla and let’s go to the “preloader” scene make a new layer called “loading bar” or “mc_loadingbar” (since we will eventually be making a movie clip).



2. On this layer, draw your own loading bar. For the purposes of this tutorial, I am going to draw a red rectangle that closely resembles the loading bar that already exists.

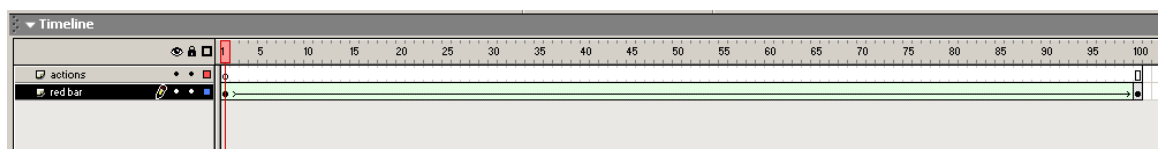


3. Next, let's convert this red rectangle to be a movie clip called "mc_loadingbar" (INSERT > CONVERT TO SYMBOL).
4. Whenever you create a new symbol. You should get in the habit of immediately going to your library and putting it into the appropriate folder. In this case, open your library palette and move the newly created mc_loadingbar into the movie clips folder.
5. Next, let's double-click on the mc_loadingbar movie clip directly on our canvas, and let's edit mc_loadingbar.

The first thing that I want to do is add an actions layer. We won't be using it at all, but it's always a good practice to get into.

6. Then I am going to rename my layer with the red rectangle on it to be "new loading bar".
7. Next, I am going to add a keyframe of the red rectangle at frame 100. This is going to be the keyframe for when the movie is 100% loaded.
8. Next, I am going to return to my original keyframe at frame #1 and shrink it's width down to be very skinny.
9. In order to make this loading bar grow/animate, I need to give it a tween. Since we are using simple shapes, I am going to select frame #1 and give it a shape tween.

When you are done, your loading bar should grow when you scrub the timeline and your timeline should look like this:



10. Next, we need to make our preloader “talk” to this movie clip. But first we need to name the “instance” of our movie clip, right? So let’s return to our “preloader” scene and let’s name our instance of our movie clip. To do this, you must select the movie clip ON THE CANVAS then go to your properties palette and name it mc_loadingbar where it says “Instance Name”.
11. Now let’s look at our actionscript at frame #1 and frame #2. As you can see, this is a simple 2-framed loop.

You’ll also notice that there is a variable called “percentDone”. This variable is a number from 1-100 – it grows every time the loop returns to frame #1 – depending upon how much of the site is loaded. We are going to use this number to display the correct frame of our new movie clip mc_loadingbar.

To do this, we are going to tell mc_loadingbar to gotoAndStop at whatever frame the movie is loaded (from 1 – 100 percent). So, if the movie is 75% loaded, then the variable percentDone should equal 75. Therefore, mc_loadingbar should be at frame # 75. Make sense?

Let’s add the actionscript...

12. At frame #1 on your actions layer add this line of code in between the variable declarations and the conditional “if” statement:

mc_loadingbar.gotoAndStop(percentDone);

so your final actionscript for frame #1 should look like this:

```
totalBytes = Math.round(getBytesTotal() / 1024);
loadedBytes = Math.round(getBytesLoaded() / 1024);
percentDone = Math.round((loadedBytes / totalBytes) * 100);
mc_loadingbar.gotoAndStop(percentDone);
if (_root._framesloaded >= _root._totalframes)
{
    gotoAndPlay("start");
}
```

13. Finally, test your movie out. In order to see the preloader work, you will need to turn on your bandwidth profiler (VIEW > BANDWIDTH PROFILER) when you test your movie in flash. You will also need to adjust the speed at which you are debugging. If you are having problems, you can always add some temporary content (large images, for instance) to an empty frame somewhere in your flash file simply to give you file more K size for testing purposes.

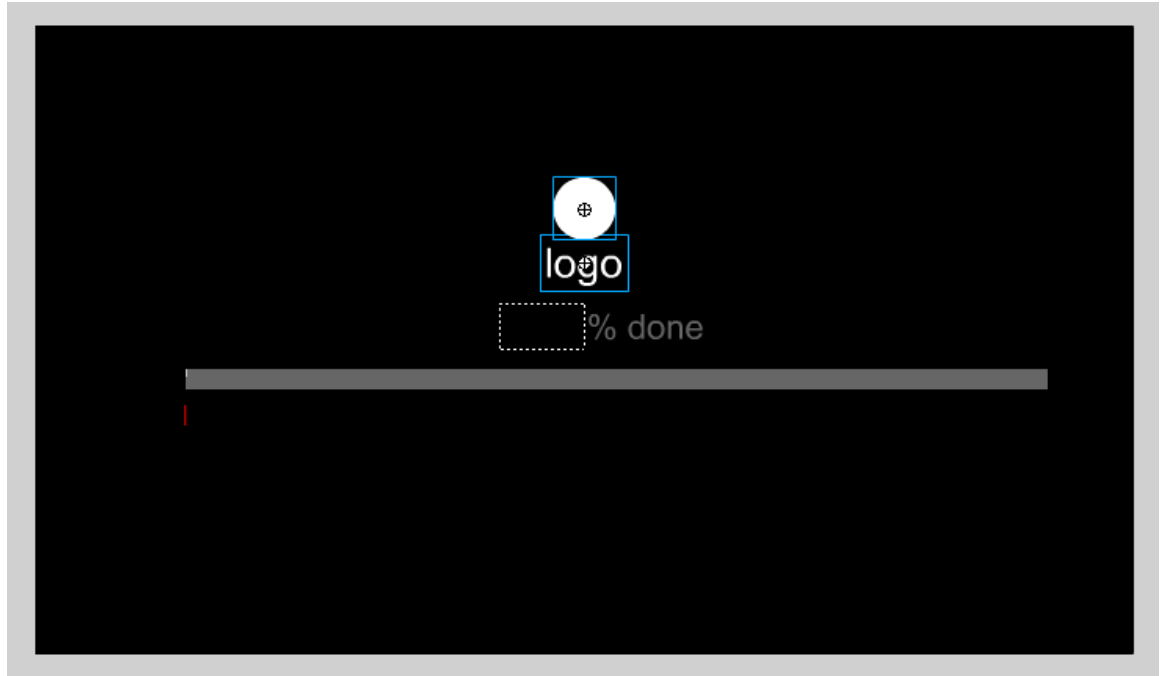
You can use this technique to do a wide variety of animations – whatever you want – so long as it occurs over the course of 100 frames. In the .fla on the web site, I will include an alternative way to make some letters “fill up” using a mask fill similar to how we did it last week – but this time we are using a simple tween animation, which should make more sense to implement.

part 1b: The nested looping movie clip technique

In this first part of the tutorial – part 1b – we are simply going to add a looping movie clip to the timeline. Since we already have a fake logo with a circle in it, I am going to create a bouncing ball/logo animation.

1. Let’s go to the “preloader” scene make a new layer called “mc_bouncingball”.

2. Since I want to animate the circle of the logo, I am going to copy and paste both the logo and the text from the “intro” scene and paste it onto my new layer in my “preloader” scene.
3. Next, I am going to position it on the canvas so as to not overlap any of my existing preloader elements.



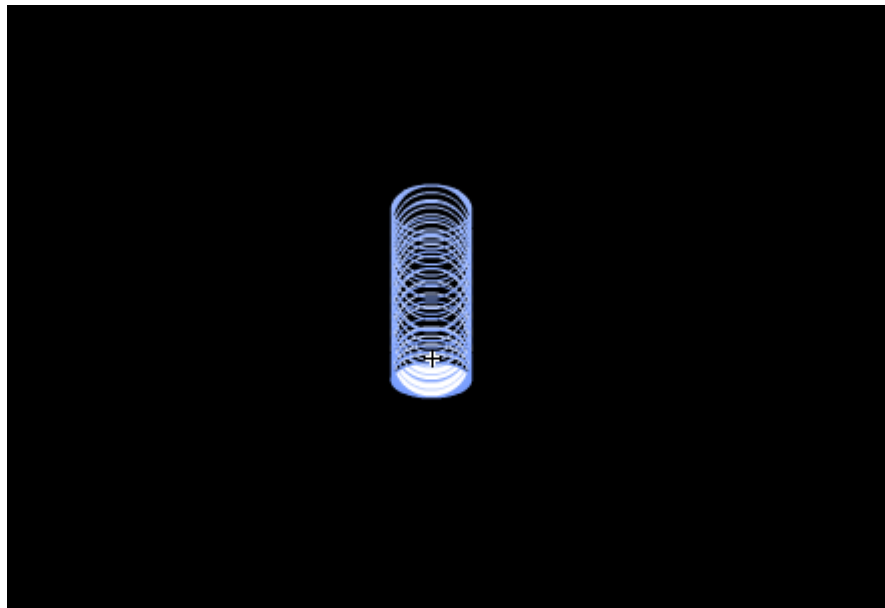
4. Next I need to convert just the circle to become a movie clip that I will make bounce. To do this, I will select just the circle shape and go to INSERT > CONVERT TO SYMBOL.

I will name it mc_bouncingball.

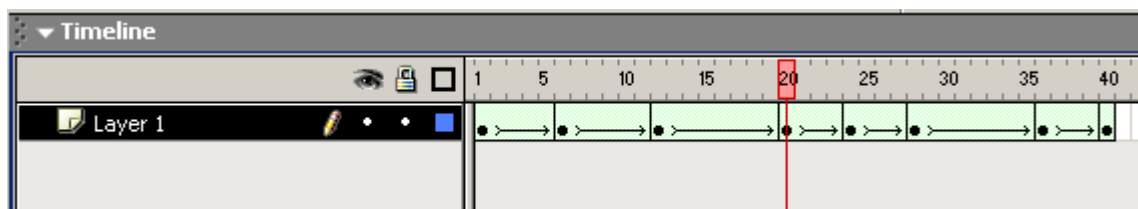
5. Next, I am going to edit it by double clicking it right on the canvas.

Once inside, I am going to animate my bouncing ball – over however many frames that I so desire (in my case I will do it over 40 frames) – and I will make sure that the last keyframe of my animation is the same as my first keyframe, so that the animation will appear like a “looped” animation.

Here’s an onion-skinned view of my animation:



and here's what my timeline looks like:



6. Finally, test your movie and you should see your movie clip of the ball bouncing running on a loop. Once the movie is loaded, it simply disappears.

With a little actionscript, you can easily make an “ending” to your movie clip if you would like for it to transition out. I can show this in class if people like.

part 2: a floating nav back

This part is incomplete and coming soon. Return to this file over the next few days.