

Rich Content Creation for Multiple Platforms with Animate CC

Adobe MAX 2017 | Session L254 | Joseph Labrecque



Made by
Thomas Wirtz

Contents

Session Abstract	5
About the Instructor.....	6
Animate CC Publications	7
Beginning Adobe Animate CC	7
Learn Adobe Animate CC for Interactive Media.....	7
Lynda.com / LinkedIn Learning	8
Pluralsight / TrainSimple	8
Adobe Animate CC	9
New in Adobe Animate CC.....	11
January 2017 - Animate CC (2017.1).....	11
June 2017 - Animate CC (2017.5)	12
October 2017 - Animate CC (2018.0)	12
Adobe Flash Player: 2020.....	16
Flash & the Future of Interactive Content.....	17
AIR Roadmap Update.....	17
What to expect when converting Flash to HTML5.....	17
Getting Acquainted with the Workspace.....	18
SEQUENCE I: Creative Foundations	20
Getting Started with Animate CC	20
Choosing a Document Type	21
Stage and Pasteboard.....	22

Frames Per Second	23
Importing Assets	24
Modify Imported Assets.....	26
Saving your Project.....	29
Content Creation within Animate CC.....	29
Timeline and Layers.....	30
Drawing Shapes.....	32
Create Text.....	33
Adding Filters to Text.....	37
Converting Text to Shapes for Vector Art Brushes.....	39
Applying Vector Art Textures	42
Managing Vector Art Brushes.....	46
SEQUENCE II: Animation Concepts.....	48
The Basics of Animation	48
Animating with Shape Tweens	48
Creating Graphic Symbols	54
Frame-by-Frame Animation	56
Graphic Symbol Instance Frame Picker	60
Delaying Visual Assets and Animations.....	62
Animating with Classic Tweens.....	64
Ease Preset Options for Classic Tweens	67
Animating with the Virtual Camera and Advanced Layers Mode	69
Using the Virtual Camera.....	70
Animating the Camera with Motion Tweens	72

Attaching Layers to the Camera.....	77
Specifying Layer Depth.....	80
SEQUENCE III: Multiplatform Publishing.....	82
Exporting Content from Animate CC	82
HD Video and Adobe Media Encoder	83
Animated GIF and other Image Formats	84
Publishing Animate CC Documents.....	86
Adobe Flash Player.....	86
Adobe AIR for Desktop and Mobile.....	87
HTML5 Canvas.....	88
WebGL (preview)	89
Custom Document Types	89
SEQUENCE IV: Adding Interactivity	91
Making your Content Interactive	91
ActionScript vs. JavaScript	91
Converting to HTML5 Canvas	92
Add a Button Component.....	94
The Actions Wizard.....	97
Understanding Platform Differences.....	102
Additional Resources.....	103

Session Abstract

Discover how to get started creating engaging animated and interactive content across multiple platforms in this hands-on lab exploring the professional tools in Adobe Animate CC. You'll walk away with a solid foundation you can apply to your own projects.

Joseph Labrecque, interactive content designer and developer, will show you how to:

- Use the rich capabilities of Animate to create an assortment of assets and animations
- Take advantage time-tested tips and techniques that let you work fast and produce compelling content for your projects
- Make your content interactive by leveraging simple code features within Animate
- Push the creative envelope using the latest new features in Animate

About the Instructor



JOSEPH LABRECQUE

*Senior Interactive Software Engineer - Author - Absintheur
Adobe Education Leader - Adobe Community Professional*

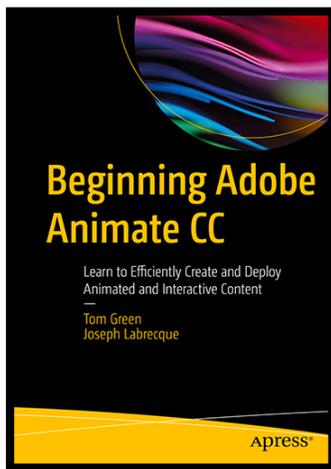
Joseph Labrecque is primarily employed by the University of Denver as a senior interactive software engineer specializing in the creation of expressive desktop, web, and mobile solutions. His work incorporates a strong focus on the Adobe Flash Platform alongside more general web standards initiatives involving the use of HTML5, CSS, JavaScript, and related technologies.

He is also the proprietor of Fractured Vision Media, LLC; a digital media production company, technical consultancy, and distribution vehicle for a variety of creative works. Joseph authors video courses and written works through organizations which include Lynda.com, Peachpit, Pluralsight, Apress, and more.

Joseph is an Adobe Education Leader and Adobe Community Professional.

Animate CC Publications

If interested in learning more about using Animate CC to create content targeting Flash Player, Adobe AIR, HTML5 Canvas, HD Video, WebGL, Apple iOS, and Google Android... Joseph has authored a number of books and video courses through various publishers. Please do have a look at the following publications after MAX concludes!



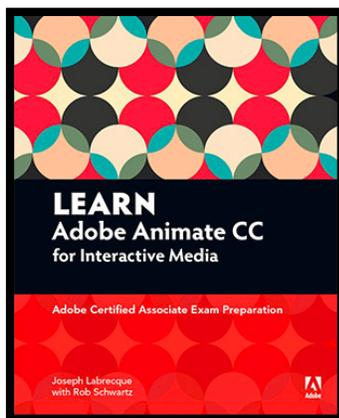
Beginning Adobe Animate CC

Learn to efficiently create and deploy animated and interactive content

Tom Green, Joseph Labrecque

Apress (April 2017)

ISBN: 9781484223758



Learn Adobe Animate CC for Interactive Media

Adobe Certified Associate Exam Preparation

Joseph Labrecque (with Rob Schwartz)

Adobe Press / Peachpit (March 2016)

ISBN: 9780134397818



Lynda.com / LinkedIn Learning

Adobe Animate CC: New Features

Adobe Animate CC: Data-Driven Animation

Adobe Animate CC: Interactive Animation

Learning Adobe Animate CC



Pluralsight / TrainSimple

Animate CC Building AIR Desktop Applications

Animate CC Programming

Animate CC Mobile App Development

Animate CC Game Development

Adobe Animate CC

A NEW AGE FOR ANIMATION.

ANY PLATFORM. ANY DEVICE. ANY STYLE.



Design interactive animations with cutting-edge drawing tools and publish them to multiple platforms — including Flash/Adobe AIR, HTML5 Canvas, WebGL, or even custom platforms — and reach viewers on broadcast TV or virtually any desktop or mobile device.

Adobe Animate is the super-creative application formally known as Flash Professional. While **Flash Professional CC** was mainly focused on producing content for Flash Player and AIR, around two years ago Adobe rebranded the application to **Animate CC** as a platform agnostic creativity application not restricted to any single platform.

Adobe announced the decision to rebrand Flash Professional CC as Animate CC in late 2015:

To more accurately represent its position as the premier animation tool for the web and beyond, Flash Professional will be renamed Adobe Animate CC, starting with the next release in early 2016.

What does all this mean for Flash Player and AIR? Nothing has changed in terms of what can be accomplished with these targets in regard to previous versions of the application. In fact... all of the enhanced drawing tools and additional features that are currently available can absolutely be used when targeting ActionScript-based .fla documents:

Animate CC will continue supporting Flash (SWF) and AIR formats as first-class citizens. In addition, it can output animations to virtually any format through its extensible architecture.

All platforms benefit from new tooling and enhancements present in Animate CC!

New in Adobe Animate CC

Keeping up with new features and enhancements within Adobe Animate CC is quite a task. With the increased resources and importance given to the application over the past few years, we've seen an unbelievable amount of progress with each major release.

Here follow the major feature updates that have been rolled out since Adobe MAX 2016...

January 2017 - Animate CC (2017.1)

Frame-Picker Enhancements: Now it's easier to re-use art assets multiple times to adjust fine movements, like mouth and hand poses. Automatically filter frames based on criteria such as keyframe, all frames, and more.

Global Scripts and 3rd Party JavaScript: Get the flexibility you need to use JavaScript code that applies to all frames in an animation. Also, now you can use the latest JavaScript libraries for animation from within the Animate UI.

HiDPI Support: See icons and text in ultra-sharp detail with more pixels displayed on your screen. Animate CC automatically optimizes your settings for HiDPI monitors.

Mute and Poster Properties: For the HTML5 Video component. (December 2016)

June 2017 - Animate CC (2017.5)

Custom Easing and Ease Presets: Create realistic motion faster by using new Ease presets to define the speed of your Classic and Shape tweens. You can also create custom Eases for Shape tweens.

Texture Atlas Export: Move from game design to development faster by exporting texture atlases or sprite sheets from Animate CC to Unity and other platforms.

Camera Control Improvements: Refined controls for the Virtual Camera pan movement.

HTML5 Component Management: Easily package, share, and install custom HTML5-based components using the Add-ons mechanism.

Brush Performance Improvements: Sketch faster and get higher-quality lines now that you no longer have to wait for a brush stroke to smooth before creating the next one.

In-app Notifications: Get helpful contextual hints and information about new and existing features as you work in the app. (March 2016)

October 2017 - Animate CC (2018.0)

Advanced Layers mode: Advanced Layers Mode is introduced to perform operations such as Layer Depth and Attach to Camera etc., You can turn ON/OFF Advanced Layers mode from Timeline panel or using Layer Depth panel.

Attach Layers to Camera: You can now attach any layer to the Camera from the timeline. When you attach a layer to the Camera, objects in

that layer are pinned to the Camera and always move along with the camera, unaffected by any Camera properties.

Reset options for Camera controls: Resetting the Camera properties and effects are just a click away using the new reset buttons in the Camera properties panel.

Layer depth Panel: You can now change the depth by dragging the color-coded rulers corresponding to a layer in Layer Depth panel. The black dotted line represents the Camera position.

Virtual Camera APIs for Runtime Interactivity: The Virtual Camera can be accessed at runtime for ActionScript 3.0, WebGL, and HTML Canvas documents.

Property level Ease Presets for Classic Tweens: In this rerelease, ease presets can be applied to individual properties when using Classic tweens.

Ease preset enhancements: The ease type dialog has been updated to include a graph to help you select the desired ease preset for the tween span.

Saving custom eases: Give your custom ease a name and save it for use elsewhere.

Scale the Frame Spans as you change FPS: Have you ever had to change your FPS without having to change the animation speed? You can do this easily now using the Scale Frame Spans option as you change FPS to keep to the time intact.

Extending or reducing time for existing spans: Select the frame span and drag the right-side edge of the selected span on timeline, forward or backward. The frames within the timeline are adjusted automatically.

View the Time along with Frames in the Timeline: You can now display the time in seconds along with frame numbers in the Timeline. This feature allows faster conversion from frames to time and also keeps you aware of the FPS you have set throughout the animation process without having to actually glance at the FPS value.

Animating on 1s, 2s, 3s and more: Turn your frame spans into 1s, 2s, 3s or to any desired interval. This works on classic tween span, shape tween span, motion tween span, Keyframe span and even on a Blank Keyframe span.

Pan through timeline from stage: Now you can scrub the Timeline right from the stage using the new Time Scrub tool. This tool is grouped with the Hand tool in Tools panel.

Keyframe navigation: Navigate between keyframes on the active layer using the Control menu or via the UI controls within the Timeline.

Component parameters panel: Animate designers can import their external components to Animate and use them to build their animations. To make this workflow easier, components parameters section is removed from the Properties panel and has been made as an independent panel.

Add actions using wizard for HTML5 Canvas document: You can now add interactivity to HTML5 compositions in two easy steps using the Add using wizard options in Actions panel.

Convert to other document types JSAPI: If you have large number of files that you want to convert at a time into other formats or document types, then you can use the JSAPI.

Convert to other document types: To convert your current document to any other document type, click **File > Convert To** and choose the desired document type.

Library panel enhancement: Animate now remembers the Library panel's Column arrangements including the column order, width and the sorting settings and subsequent Animate sessions will retain the Library panel's state.

Texture Atlas enhancements: Generate texture atlas workflow performance has been improved.

Adobe Flash Player: 2020

JOURNALISTS AND RESEARCHERS
Instead of simply saying "Flash" -
consider the following distinctions:

 "Flash Player" is the web browser technology you are most likely referring to. A browser-based runtime going away by the year 2020.

 "Flash Professional" is the authoring software now known as Animate CC. It enables the creation of rich content for multiple platforms and remains in active development.

Additionally... we must consider the AIR runtime technology for desktop and mobile, the SWF file format, the ActionScript programming language... and much more!

None of which are affected by #Flash2020
#MadeInAnimate by @JosephLabrecque

Adobe has announced that they plan, in cooperation with browser vendors, to end-of-life Flash Player in 2020.

It is important to note a few things here... this is only in regard to Flash Player *not* AIR! The AIR team have some very cool stuff on the roadmap and AIR will continue on, regardless. Additionally, we now have a number of ways to convert content which requires such conversion through Animate CC and other tools.

Flash & the Future of Interactive Content

“Adobe is planning to end-of-life Flash. Specifically, we will stop updating and distributing the Flash Player at the end of 2020...”

<https://blogs.adobe.com/conversations/2017/07/adobe-flash-update.html>

AIR Roadmap Update

“Adobe remains committed to AIR and we believe it continues to be a great desktop and mobile development platform.”

<https://forums.adobe.com/thread/2362234>

What to expect when converting Flash to HTML5

“We had the pleasure recently of connecting with interactive software engineer Joseph Labrecque to learn more about his thoughts on Animate CC, converting Flash to HTML5 and what type of content should be converted.”

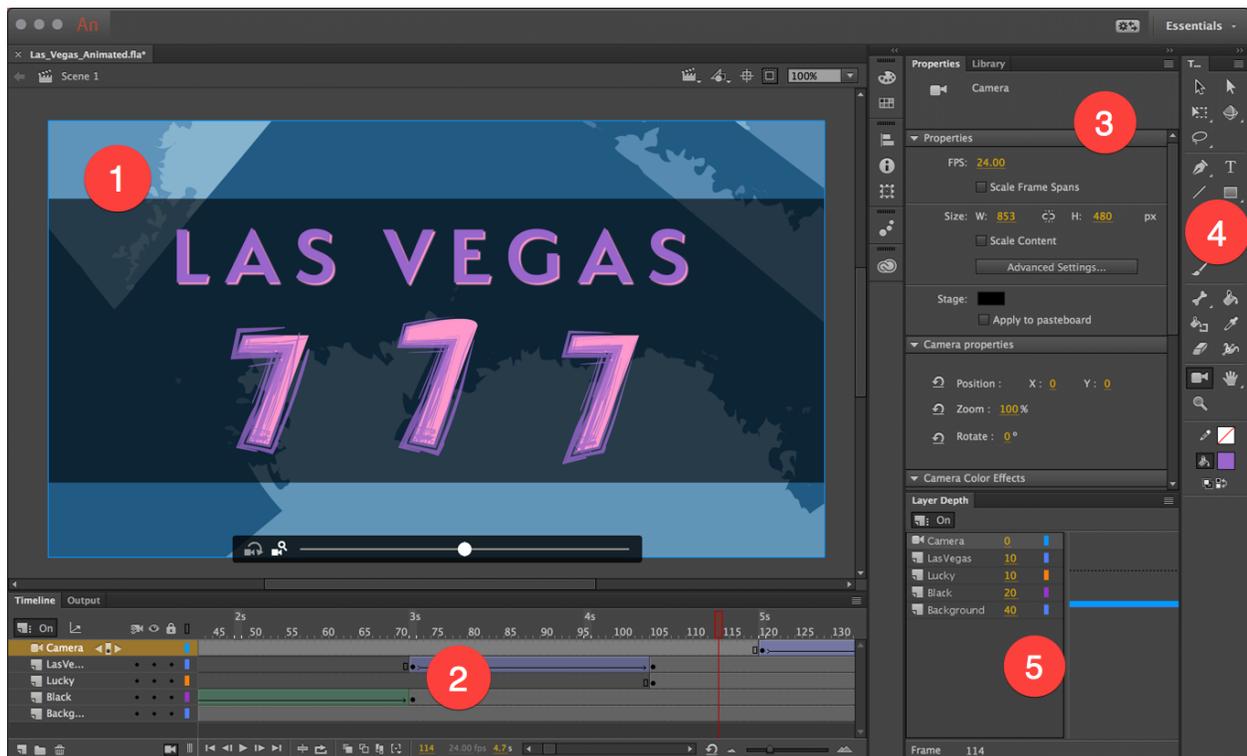
<https://blogs.adobe.com/creativecloud/what-to-expect-when-converting-flash-to-html5/>

Getting Acquainted with the Workspace

The first thing we will do is get a good look at the various tools and panels available to us within Animate.

Some of these may be familiar for those who have worked with Animate in the past, or even other creative tools like Photoshop or Illustrator. Other tools may be brand new – such as the new Camera tool.

Major workspace areas are highlighted below.



1: The Stage and surrounding Pasteboard

Anything within the Stage is visible within your project. The Pasteboard is the surrounding area, which is off-stage and not normally visible.

Animate CC has a new stage clipping feature which hides anything on the pasteboard.

2: The Timeline

The Animate Timeline is where you manage Layer order and animate content across Frames. It is also where the powerful Motion Editor can be accessed.

3: The Library and Properties Panel

The Properties panel will display the properties of any selected object for manipulation. When no objects are selected, it will display properties for the document itself.

The Animate Library is where any of your symbols and imported assets are stored. Vector shapes which have not been made into Symbols will not appear within the Library – only existing on the Stage.

4: The Tools Panel

This is the spot from where all out our creative tooling can be accessed! Brushes, shapes and primitives, the Pen Tool, the new Camera tool, and more!

5: Layer Depth Panel

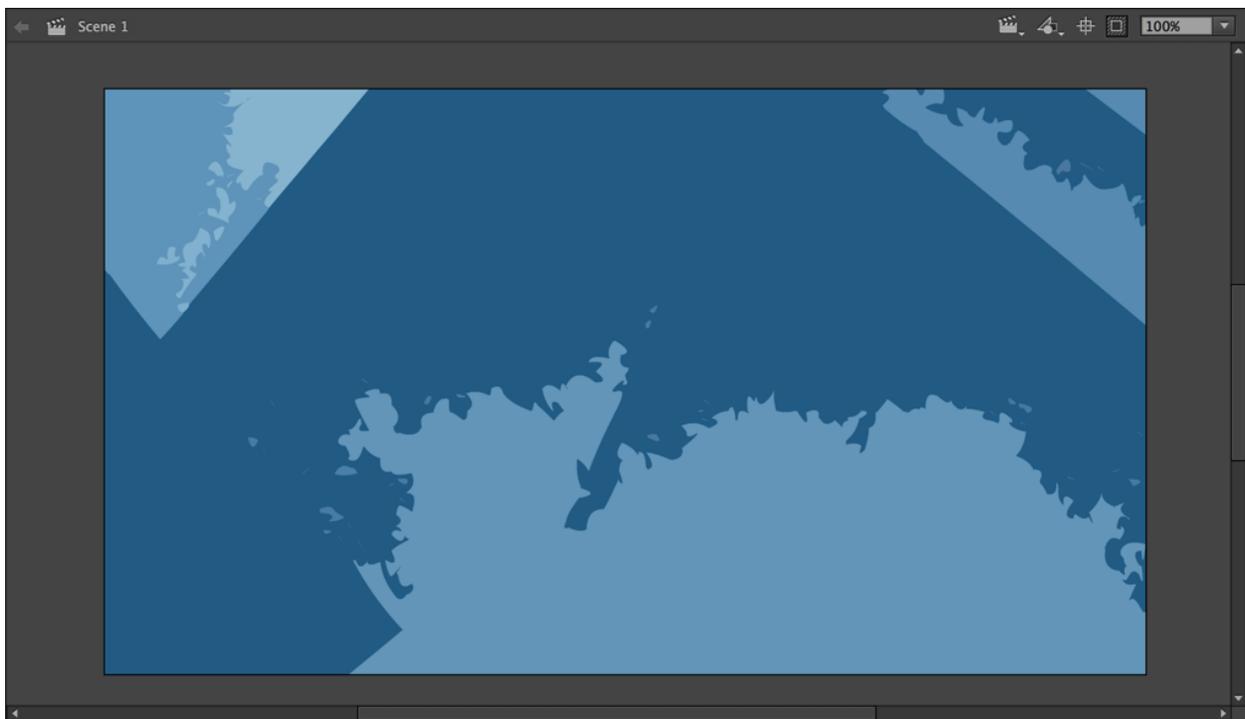
Allows the configuration of z-depth for layers affected by the camera. This enables the illusion of depth as the Camera pans and zooms across your animation.

SEQUENCE I:

Creative Foundations

Getting Started with Animate CC

With any project in Animate CC, the first thing we have to do is make some decisions around which platform we want to target, create a new document targeting that platform, and then configure certain aspects of the project before saving the document.



Here we will create a new document, set its properties, and then import and vectorize a photographic image.

Choosing a Document Type

When creating a new document in Animate, you have a number of choices – but every type will take the form of an .FLA file unless it is strictly code-based. Any document can also be created as an .XFL document – which is a good choice if you’ll be using a version control system as it is basically an uncompressed .FLA file.

Targeting Flash Player? **FLA.**

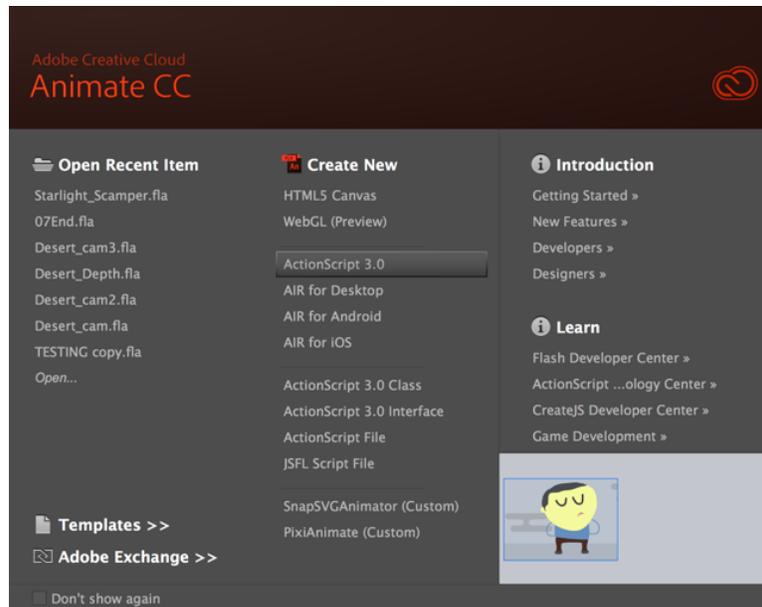
Targeting HTML5 Canvas? **FLA.**

Targeting WebGL, Apple iOS, or Google Android?

FLA. FLA. FLA.

Why is this important? Because you are able to copy assets and animations across all of these different publish targets – since they all use the exact same authoring file type!

What determines whether you create a document targeting HTML5 Canvas, Flash Player, or some other platform is going to depend upon your inevitable output needs. For instance... if you are going to be generating content for After Effects or straight to HD Video – targeting Flash Player makes a lot of sense because you can use all of the creative tooling available to you within Animate. However, if you are creating an interactive project for distribution over the Web – then HTML5 Canvas is the most likely choice.



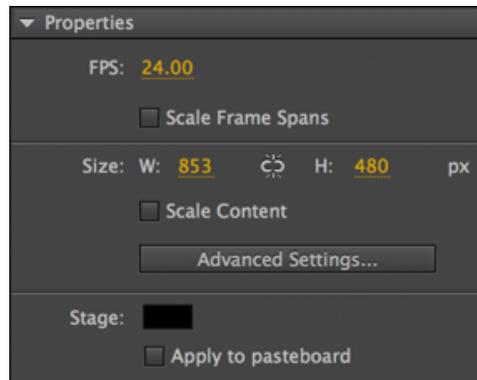
For this project, we'll be using the document type which allows the most creative choices: **ActionScript 3.0** targeting Flash Player.

- From the **Welcome Screen**, click **ActionScript 3.0** under the middle column titled **Create New**. A new ActionScript 3.0 document is then created.

Even though we are working with an ActionScript 3.0 document for this project, we can always either convert the document type later on... or even simply copy assets from one document type to another. Everything in Animate shares the same file format... so everything is sharable between document types!

Stage and Pasteboard

We are going to customize the Stage size and background color for this project. We do so through the use of the Properties panel. This panel will reflect properties of the current document, so long as nothing else is selected on the Stage or within the Timeline.



- In the Properties panel, under the section named **Properties**, change the Stage width to **853** and the Stage height to **480**. This is a common 16:9 ratio.
- Click the color chooser and change the Stage color to **#000000** (black).

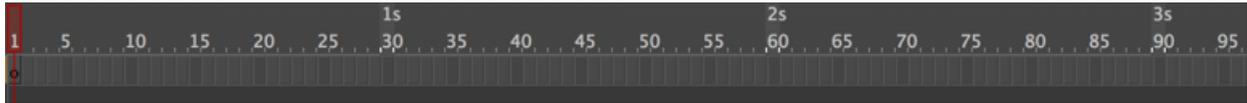
The Pasteboard is the area surrounding the Stage. This can be used to set content to be hidden off-stage and then either appear or animate into view. You can also use this area to place assets you aren't quite sure of yet.

Notice that even though you changed the color of the Stage, the color of the Pasteboard is still grey? Some people, when working with content in Animate, can find this distracting. You'll notice a small checkbox named **Apply to pasteboard**. Checking this will use the selected Stage color across the pasteboard as well, creating a seamless drawing surface.

Frames Per Second

Animate has a frame-based timeline. This means that the overall time it actually takes to perform an animation will be based off of the number of frames along with the framerate – or frames per second – property of the document.

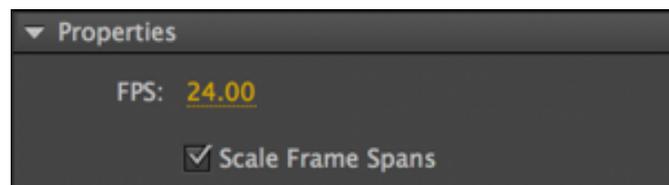
For example, if we have the FPS set to 30 and our total frames in the Timeline is 90, then our resulting animation length will be 3 seconds. Animate now has a display of seconds along the top of the Timeline to help in knowing how long an animation is.



We'll keep our FPS at 24.

- Check and be sure that the FPS value in the Properties panel for our document is at **24**

It used to be rather difficult to adjust your FPS after you begin animating within a document, as the frame spans would not adjust and the entire animation timing would be thrown off. We have a new option when adjusting FPS which is an optional checkbox titled “Scale frame spans”. If this is selected before adjusting a document’s FPS value, Animate will adjust the frame spans accordingly.



It's always a good idea to know your correct FPS when you start off though!

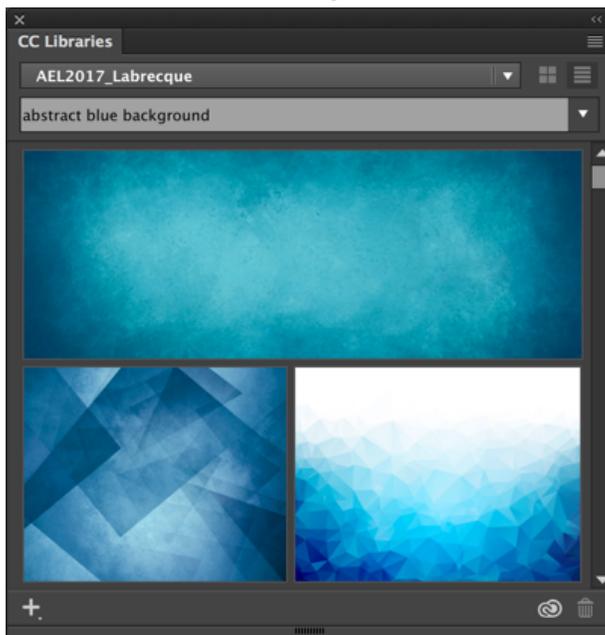
Importing Assets

While Animate is a great vector design and animation tool, it's entirely possible to use bitmap images within a document as well. Animate allows the import of common bitmap image file formats like PNG, JPG, and GIF... but also includes specific importers for both Adobe

Photoshop and Adobe Illustrator – allowing you to intelligently import combinations of bitmap, vector, text, and even certain effects from those document types.

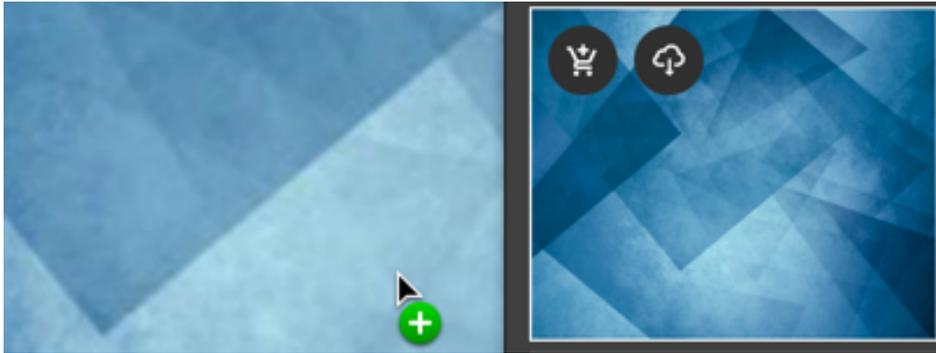
For this project, we will import an image from Adobe Stock and then use that within our document.

- Open the CC Libraries panel. If you cannot find it, you can go to Window > CC Libraries
- In the text input field that says, “Search Adobe Stock”, type in “abstract blue background”

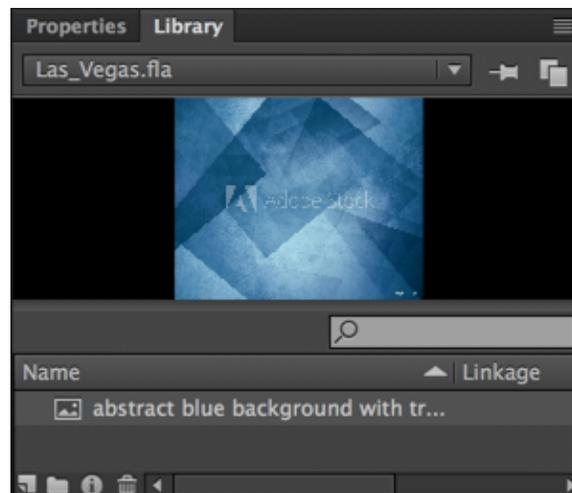


- Animate will go out to Adobe Stock and locate a number of images which match these search terms and display previews within the panel.
- Find one that interests you and click-drag the preview onto your Stage. A low-resolution version of the asset from Adobe Stock will

then appear on the Stage.



If you have a peek within your project Library, you'll see that the preview bitmap image exists there as well. Any imported bitmap images, sounds, and even video (in certain cases) will reside within the document Library.



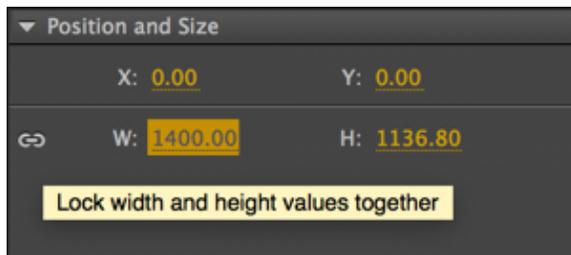
In addition to imported assets, the Library is also used to hold Symbols you create, Fonts, and other additional object types. You'll see how Symbols can be used from the Library a bit later in this session.

Modify Imported Assets

If you notice, the image from Adobe is Stock is much larger than the Stage. That's okay because we are simply using this as raw materials to

create something else. You'll also notice that the image is branded with an Adobe Stock watermark. This is also of no consequence.

- In the Properties panel, with the bitmap selected, be sure the little link icon is closed and not broken. This allows the image to scale in proportion as we resize the width or height, retaining its aspect ratio.

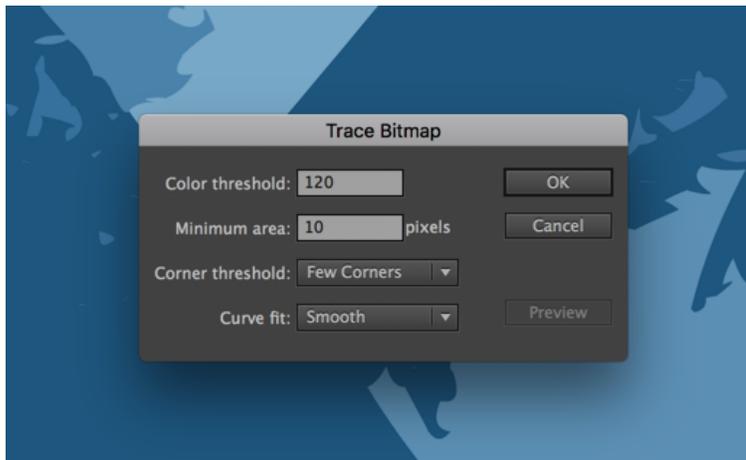


- Resize until you no longer see the watermark on the Stage. Things may look a bit pixelated but that is okay. We'll deal with that next.



- With the bitmap image still selected, go up to the application menu and choose Modify > Bitmap > Trace Bitmap. The Trace Bitmap dialog appears.
- Tweak the different settings here until you get a neat, abstract look. You can continue to click Preview as you tweak the settings

to see how things will end up.



- Choose OK when ready and notice that the bitmap no longer resides on the Stage. In its place is the vector image you just created.

We now have an abstract background to use for our project. You will want to perform some clean up before moving on, so be sure that the Stage is not being clipped and then zoom way out until you see everything. A quick way to do this is to choose Show All from the Zoom dropdown.



The black line you see is the Stage border. Using the Selection tool, you can move the abstract vector background around until what appears on the Stage is exactly what you'd like to see. At that point, you can also

draw selection rectangles across those portions you do not care for and hit DELETE to remove them.



Of course, you don't need to get the size of this background asset even close to perfect. Just enable Stage clipping to hide any distracting overlaps.

Saving your Project

Now, be sure and save your project before moving on to the next sequence. You can name it whatever you like... though I recommend something like "MAX2017_SEQ1.fl" in order to have your files conform with each project sequence.

Content Creation within Animate CC

In the previous section, we saw how to use imported assets within an Animate project. It is also essential to understand how content can be created within Animate itself – and there are a wide assortment of tools available to accomplish this!

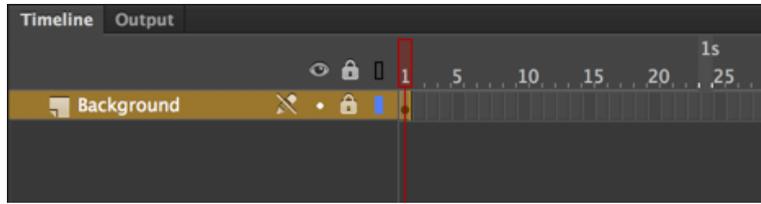


Timeline and Layers

The first thing to address before moving on is the creation and proper naming of layers upon the timeline.

Generally, you'll always want to have any asset you are animating on its own layer. Even with objects which are not being animated – they often benefit from the organizational structures and labeling provided by a well-organized timeline.

- Rename the single, existing layer to “Background”. You can do this by double-clicking the layer name.
- Lock the Background layer by clicking the dot beneath the Lock icon. A lock icon now appears on that layer.

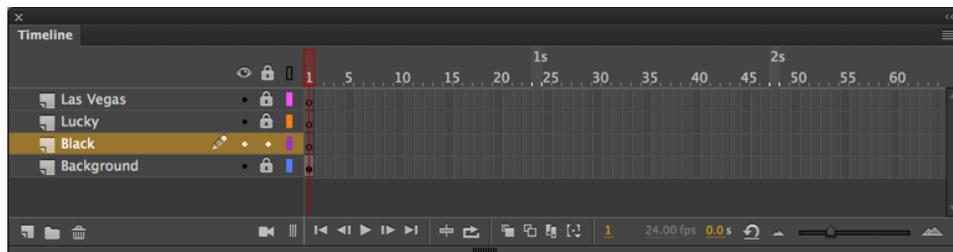


Locking a layer will prevent you from changing the content of that layer in any way. This is good because we are finished with this layer and anything else we do will be included in other layers which we will now create.

Use the **New Layer** icon at the bottom-left of the timeline to insert 3 additional layers – making 4 total layers for now. You’ll next want to double-click each layer name to rename the layers to something more descriptive.

Beginning from the top-most layer, proceed in naming your additional layers as follows:

1. Las Vegas
2. Lucky
3. Black
4. Background

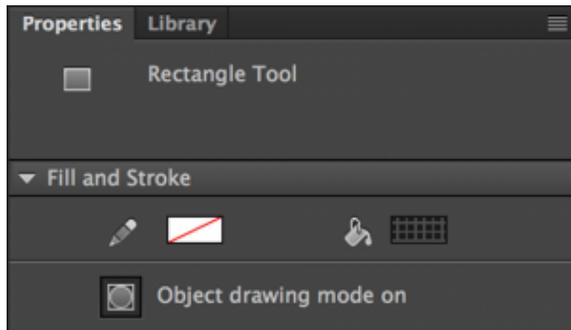


Lock all layers except for **Black** for now. This is the layer we will be working in next.

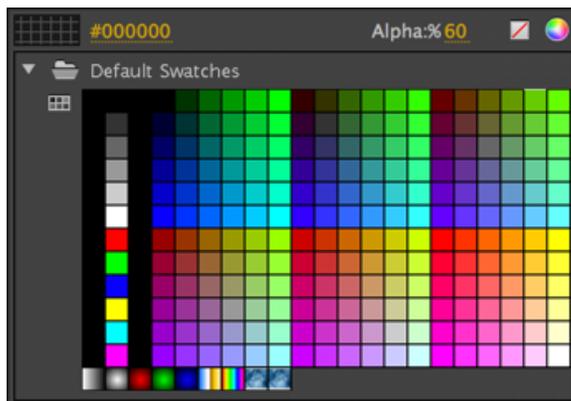
Drawing Shapes

We will now use a number of the shape and selection tools available in Animate to design some

- Select the **Black** layer and choose the **Rectangle tool**.

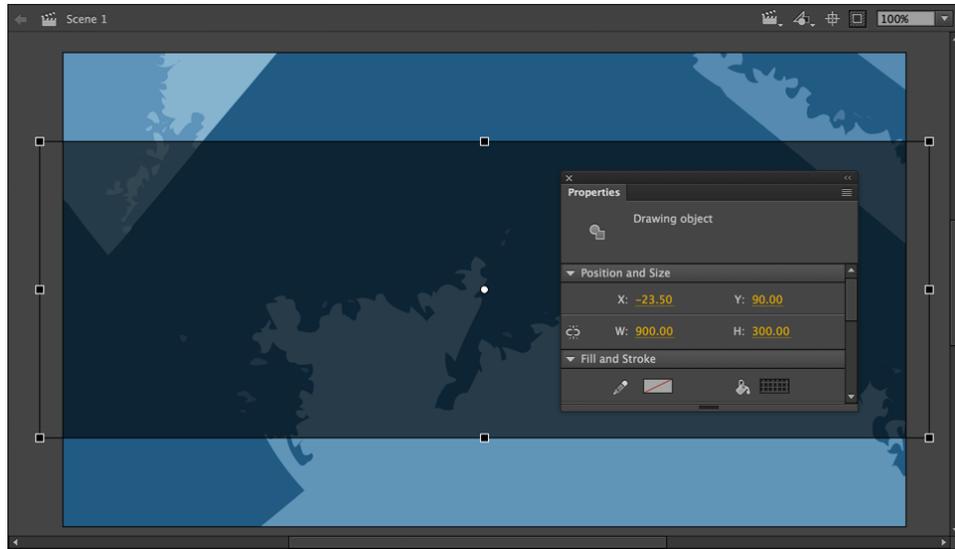


- Create a rectangle in the middle of the stage with no stroke and the fill color of #000000 at 60% opacity.



Don't worry much about size and position just yet.

- Finally, to position and size our rectangle, use either the **Selection tool** or **Transform tool** and size it roughly a bit wider than the Stage and at about 300px in height.
- The **Align panel** can be used to center this object in relation to the Stage, or you can just eyeball it.



Since anything outside of the Stage is not visible when published, it's fine for the shape to extend onto the Pasteboard.

Note that the **Subselection tool** can be used to select paths and points of any vector object in Animate. It may be beneficial to play with this tool a bit before moving on so that you are familiar with it against a basic shape.

Okay – go ahead and lock the **Black** layer for now.

Create Text

Text is a type of vector object within Animate and is created with the **Text tool**. We'll be creating a number of different pieces of text for this animation – so let's first unlock the **Las Vegas** layer.

Before moving on, you may want to grab the Serenity font from Typekit and sync it to your desktop. Of course, you can use whatever font you like!

- With the **Las Vegas** layer unlocked and selected, activate the **Text tool** from the Tools panel. In the Properties panel, be sure that **Static Text** is chosen as the text type. Static Text will always be rendered as vector shapes upon publish – so the user will not need the font you choose.
- Click on the Stage and type “Las Vegas” into the text object that is created.

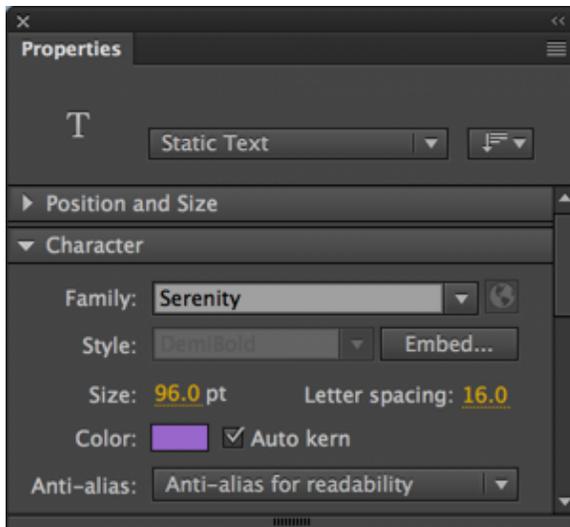


It likely looks awful... let's fix that.

- Using the Selection tool, click on the new text object and have a look at the Properties panel. From here you can set all the properties of your text to make it actually look nice.

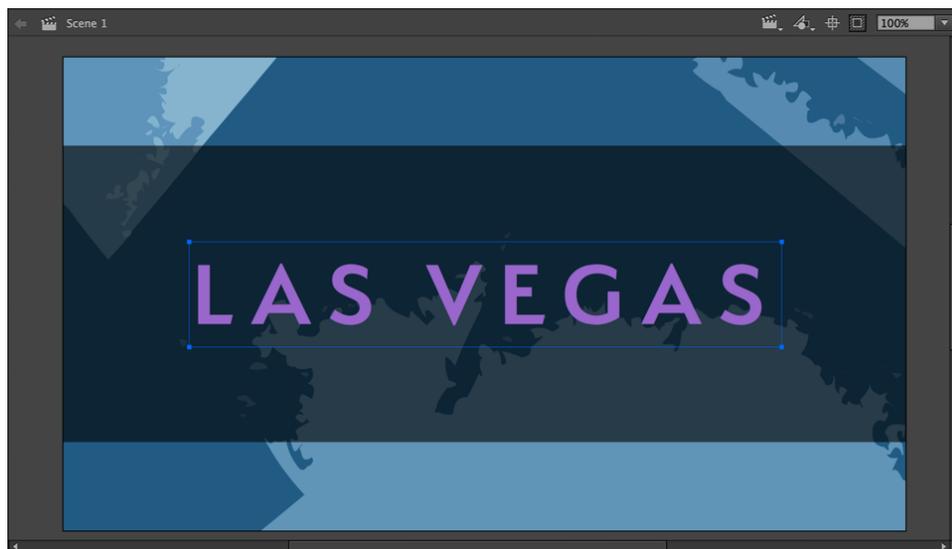
Choose the following values to set:

Family:	Serenity
Size:	96pt
Letter	Spacing: 16
Color:	#9966CC



Feel free to play around with these properties if you wish!

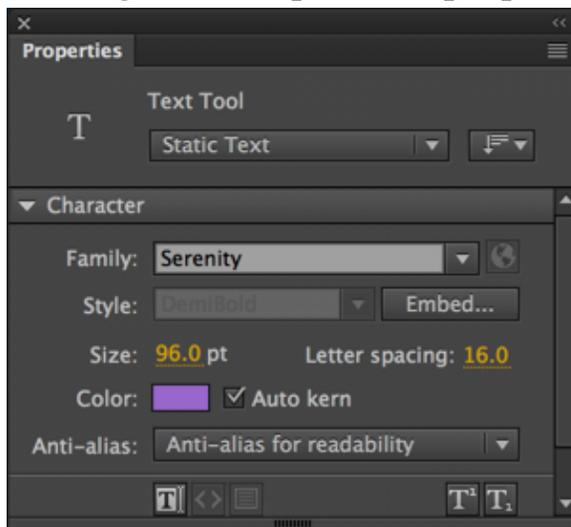
- Once you've set some suitable properties for this text object, use the Align panel to center it on the Stage. Things should be looking pretty good at this point, but we are not done with text just yet.



With the “Las Vegas” text object created, let's now work within the **Lucky** layer to create the remainder of our text elements.

- Move the “Las Vegas” text closer to the top of the black rectangle. This will give us space for the remaining text objects.

- Lock the **Las Vegas** layer.
- With the **Lucky** layer unlocked and selected, choose the Text tool once again. The previous properties are likely still present.



- Directly below our existing text, click on the Stage and type “777” – *LUCKY 7’s!*
- Position the “777” text beneath the “Las Vegas” text and center it horizontally using the Align panel.
- It would be good to increase the font size of this text as well. Using the Selection tool, select the text object and bump the size value up to 140 within the Properties panel.

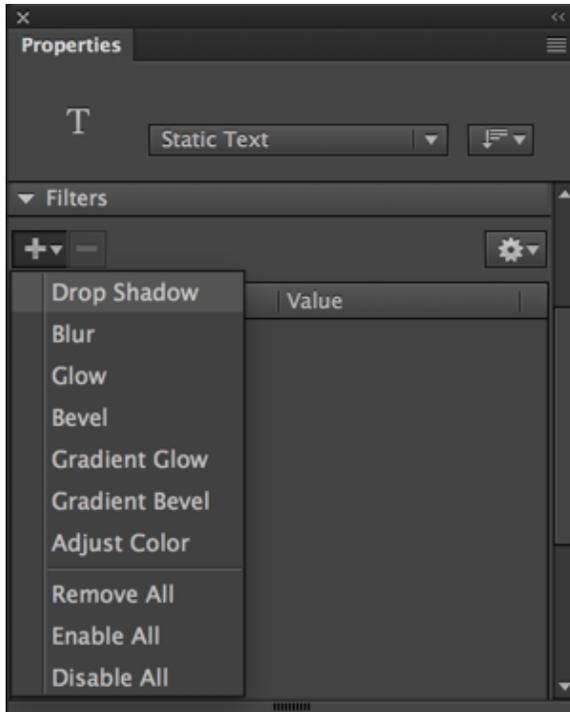


Okay – our text elements have been created and things are looking pretty good... but there are a number of additional things we can do to make this look even more interesting.

Adding Filters to Text

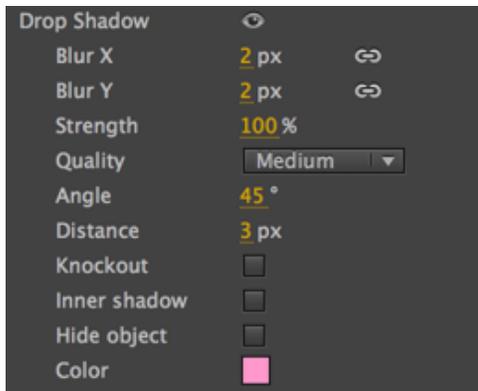
The text we've created looks pretty good right now – even if a bit 1-dimensional... let's add some filter effects to really make it pop and give it a neon-Vegas vibe.

- Using the **Selection tool**, select the “Las Vegas” text and take a look at the **Properties panel**.
- Within the **Filters** section, click the **+** icon to add a **Drop Shadow** filter to this text object. While we'll be using a single filter effect for this piece of text – you can actually stack multiple filters for even more options.



Note that there are a lot of different filter effects that can be added when working with an ActionScript 3.0 document type. If using other document types, you will have less options to choose from.

- We will use a very solid-looking, hot pink drop shadow on our text object. Enter the following values:
 - Blur X/Y: 2 px
 - Strength: 100%
 - Quality: Medium
 - Angle: 45 deg.
 - Distance: 3 px
 - Color: #FF99CC



Offsetting the drop shadow by a distance of 3 pixels at a 45 degree angle makes for a striking effect – providing a bit of depth and contrast to the original text. Looking good!



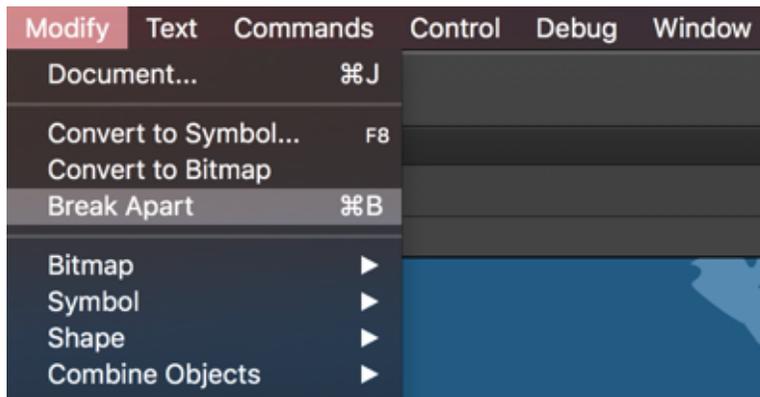
Although... the lucky 7's now appear a bit drab. We'll deal with that in a different way using some neat brush effects next.

Converting Text to Shapes for Vector Art Brushes

Vector Art Brushes can be used to apply all sorts of creative flair to simple strokes within Animate projects. We'll be using these textures to make our "777" text much more interesting.

There is a bit of work that must be done before we actually do this though... **Vector Art Brushes** are used on shapes (strokes, specifically) and not text objects – so we must first break our text apart into simple shapes to use this feature.

- Use the **Selection tool** to select the “777” text object and then either choose *Modify > Break Apart* or use CTRL/CMD+B to break the text apart.



You’ll note that we now have three individual text objects – one for each character of our original text object.

- We must break these apart once more to achieve shapes.



With the three text objects still selected – issue the **Break Apart** command once again.

- The former text objects now exist as simple vector shapes. Choose the **Subselection tool** and draw a selection rectangle across them all

to see all the vector paths and points which compose them.

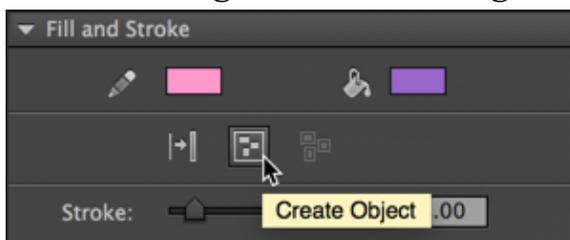


- To use vector art brush textures – we must add a stroke to these shapes since they are now comprised of only fill. For this task, choose the **Ink Bottle** tool from the Tools panel and simply click each path to apply a stroke to the existing fill.



We are using the same pink color previously used for the “Las Vegas” text drop shadow filter effect: #FF99CC.

- For this next part, it would be helpful to ensure that we are working with Drawing Objects and not basic shape data – as it is easy to only select a portion of a shape – and difficult to do so when dealing with a Drawing Object.



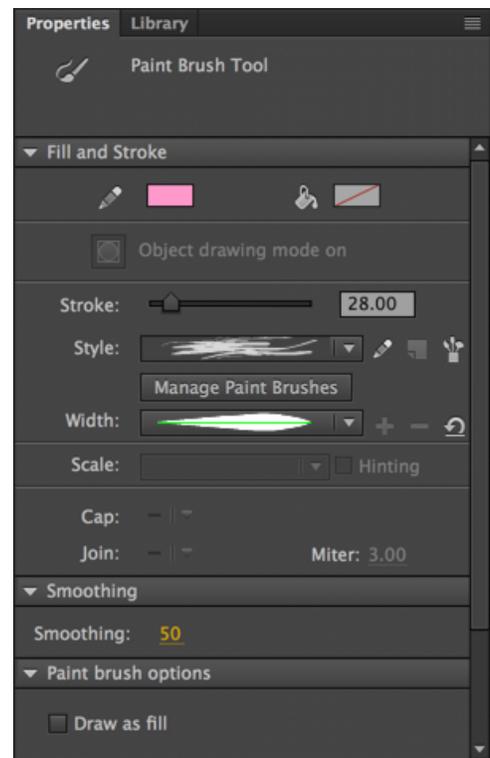
To convert each “7” into a Drawing Object, select each shape individually by dragging a selection rectangle around it and then click **Create Object** in the Properties panel.

- It is now very easy to simply click on any of the three “7” Drawing Objects to move them around the Stage in their entirety. Let’s put about 20-30 additional pixels between the center “7” and the two on each end. This will give more weight to this portion of the design and more space for animating these individual items in the future.

Applying Vector Art Textures

Animate CC has had a **Pencil** tool for strokes and a **Brush** tool for fills for quite a long time. They are great for creating free-form shapes, but the new **Paint Brush** tool is something else altogether. It allows you to draw out paths with rich vector art brushes applied along them.

This art can take the form of **Art Brushes** or **Pattern Brushes**. You can use brushes that come bundled within the **Brush Library**, brushes created with the **Adobe Capture CC** mobile app, or even brushes created within **Animate CC** itself!

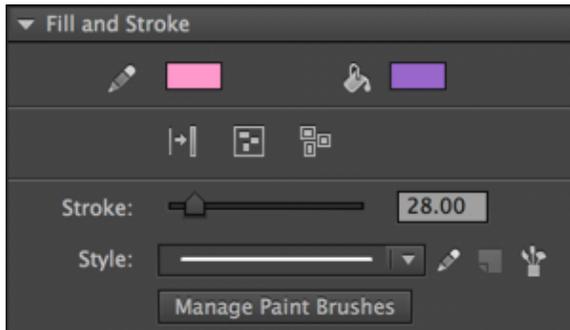


We will not be using the **Paint Brush** tool directly – but can still apply vector art textures used by this tool to any stroke that exists within our document.

- Be sure that all 3 “7” drawing object shapes are selected. This will allow you to change the properties of all three objects at once. Go

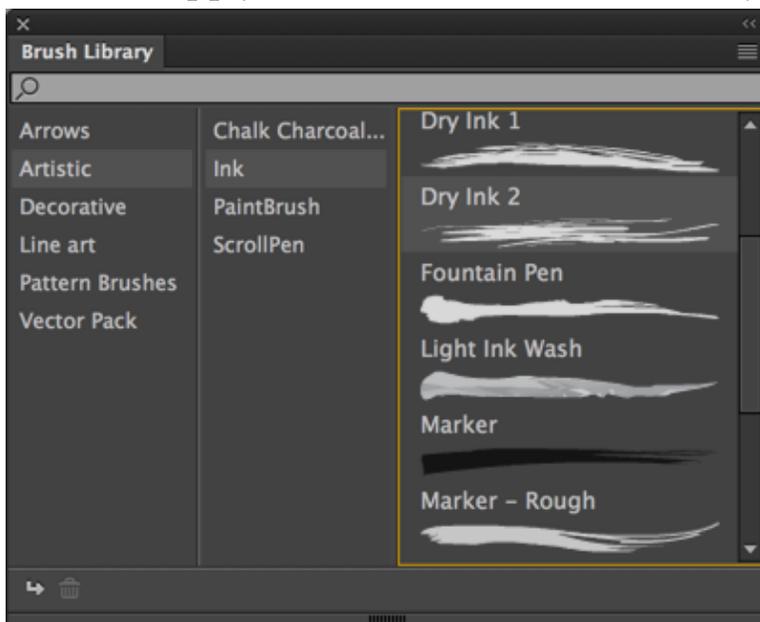
ahead and change the Stroke value to 28 and notice it change across all three objects!

- We'll now open the Brush Library by clicking the little paint can icon within the Properties panel.



You can find it right next to the Style settings within the Fill and Stroke section.

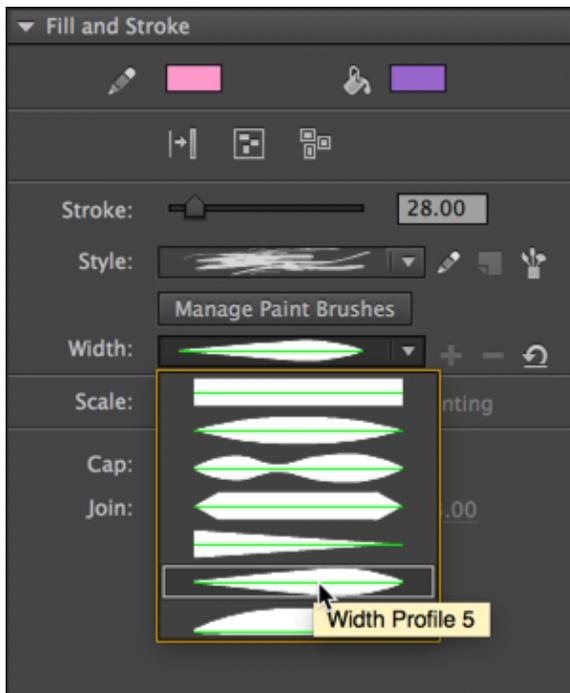
- With the Brush Library now available, choose Artistic > Ink from the categories available. Double-click upon the brush named “Dry Ink 2” to apply this brush to the selected objects.



You'll notice that the brush is now displayed within the Properties

panel under the Style dropdown. You can go ahead and close the Brush Library panel if desired.

- Since we had all three of the “7” drawing objects selected when choosing new brush – it is automatically applied to all three. Without de-selecting anything, choose “Width Profile 5” from the Width dropdown.



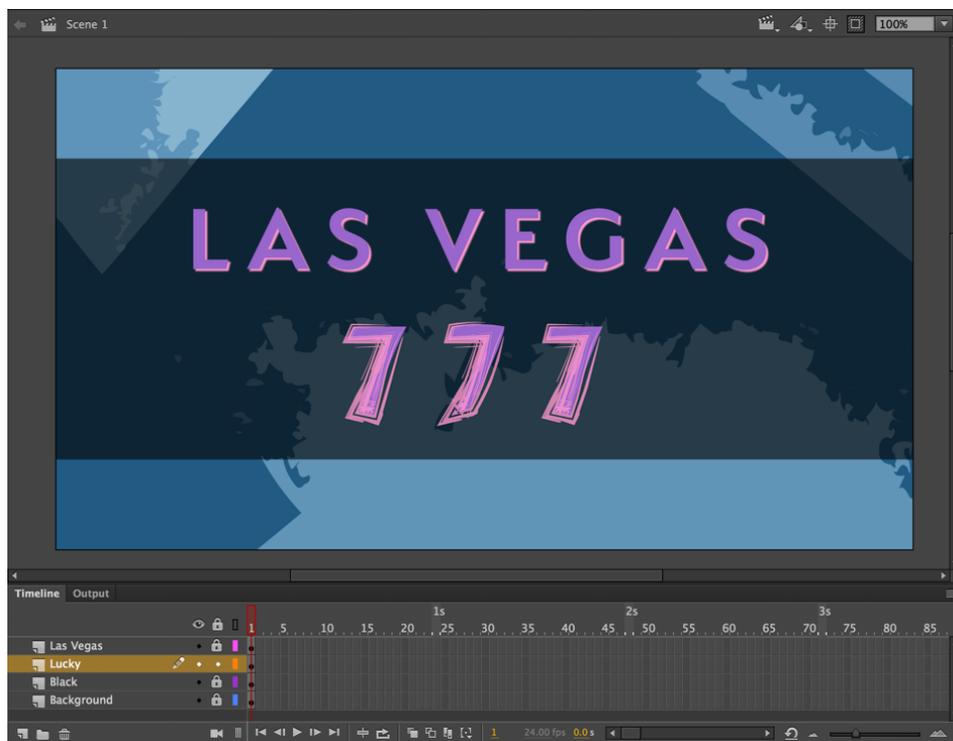
This is a Variable Width Profile and can be adjusted further with the Width tool in the Tools panel, if desired.

- As a last step, let's add a bit more variability to the look of each “7” object. You can experiment with Stroke values, Brushes, and Variable Width properties all you like – but one of the simplest ways of tweaking shapes is to use the Selection tool with nothing at all selected on the Stage.



You'll find that hovering over the stroke path of shapes or drawing objects will cause the cursor to display an arc next to it. This indicates that you can click-drag to modify the shape with the Selection tool. Try it out!

Okay... with all of that completed, we now have a pretty compelling visual composition to work with.



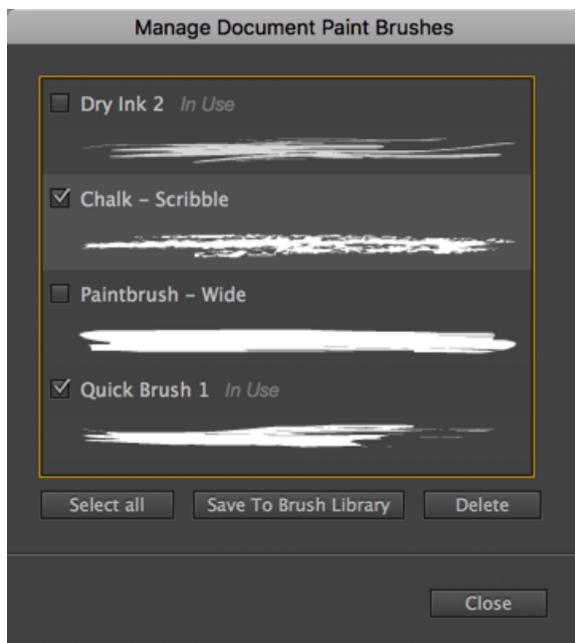
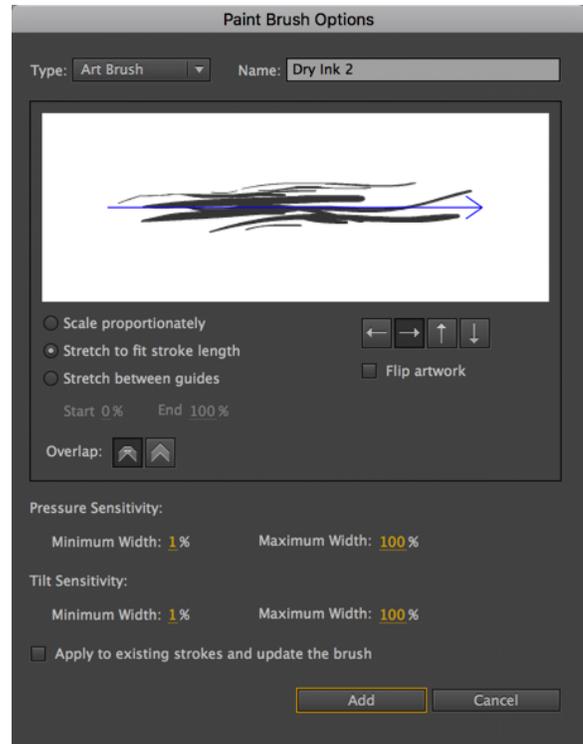
In the next sequence – we'll add some animation to this project to really ramp up the interest!

Managing Vector Art Brushes

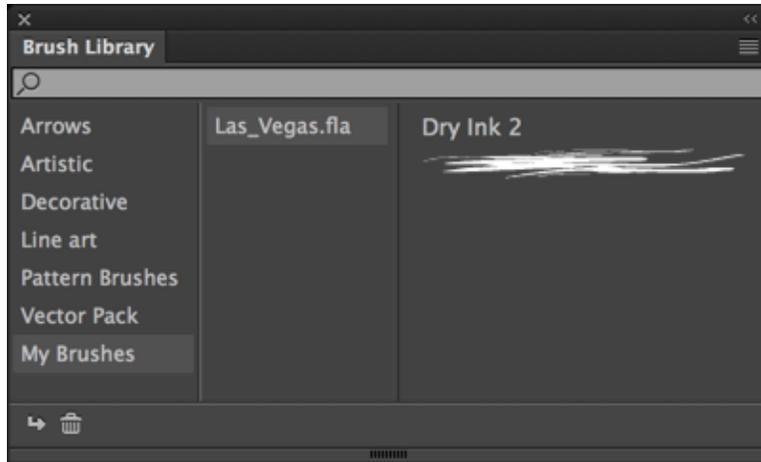
Before moving on, here is a bit of information around managing vector art brushes within your document.

Note that any brushes can be modified from the Properties panel. Change the way brushes are applied to a stroke and even adjust the pressure sensitivity options when using the Paint Brush tool.

You can even switch a brush between Art Brushes and Pattern Brushes – depending upon your needs.



So what happens when you are trying out a bunch of different brushes and they all wind up within your document... how do you clean up such a mess? Thankfully, you are able to manage your brushes from the Properties panel as well – removing any brushes from your project that you will never use.

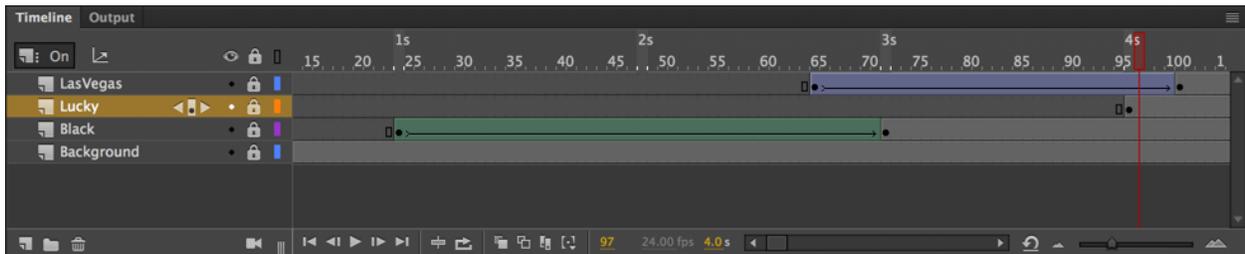


You can also save any brushes you use often into your universal brush library. These can be accessed across documents and can contain often-used brushes and even custom brushes you've created within Animate CC.

SEQUENCE II:

Animation Concepts

The Basics of Animation



Since we have much of our content now created and present within the project. It's now time to begin animating this content in various ways. We'll begin with some simple animation techniques and later move on to utilize more advanced tooling.

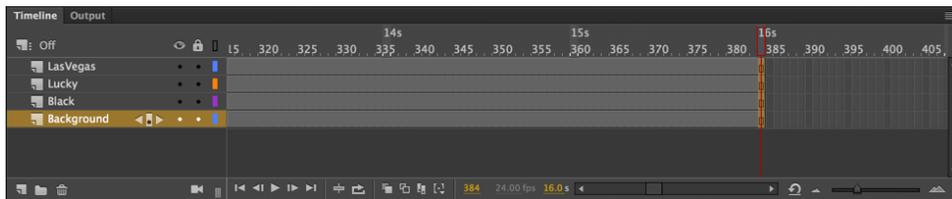
Animating with Shape Tweens

As the name suggests, **Shape Tweens** are only usable upon vector shape objects within Animate. They can be used to tween one shape into another, move shapes around, change shape color across time, or even modify a gradient transform in an animated fashion.

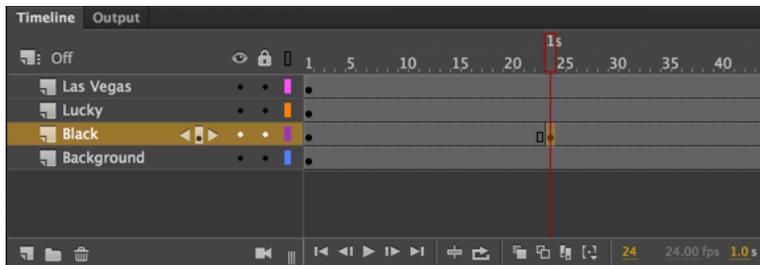
- We already have a nice black shape on the Stage in the **Black** layer. If the layer is currently locked... unlock it.

Now this shape should be the only object on that layer because when creating a shape tween in Animate – the rule is generally one shape per layer – else things can get weird.

- In the Timeline, scroll all the way over to the **16s** mark (frame 384) and click-drag across all four layers at that frame. In the application menu, choose **Insert > Timeline > Frame** to insert a series of frames up to that specific point.



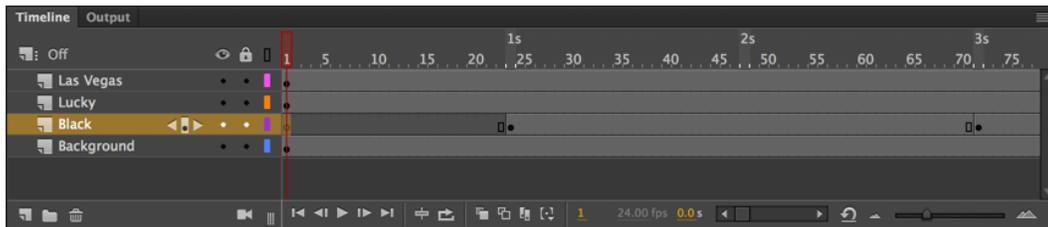
- Now scroll back over to the **1s** mark. Within the application menu, choose **Insert > Timeline > Keyframe** to insert a new keyframe at the 1s mark.



A keyframe appears as a filled circle within the frame. This indicates that it is a keyframe with content within it. Keyframes are used to express some change in the properties of whatever exists upon that layer – for instance, position, color, transform, et cetera. At the moment, the keyframes at frame 1 and the present frame are identical.

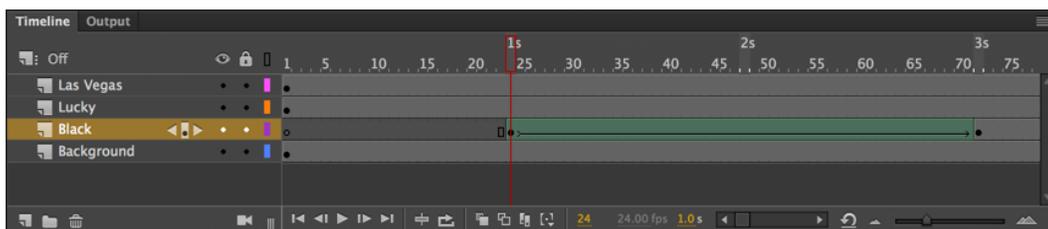
- Add another keyframe at the **3s** mark in that same layer. We are going to create a Shape Tween between these two keyframes.

- Before we create our Shape Tween – we can remove the content from the keyframe at frame 1 since we want our black shape to appear only once the playhead reaches the 1s mark. Click upon the keyframe at frame 1 of the Black layer. This will cause all the content of the keyframe to become selected on the Stage. Now, simply hit the **delete** key on your keyboard to remove everything from that keyframe, creating a **Blank Keyframe**.



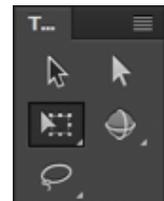
You can see that we now have an unfilled circle on frame 1, and a series of empty frames until we hit the next keyframe at the 1s mark. The black shape will not appear until a full second into the playback.

- Click the span of frames in between the two keyframes just created and choose **Insert > Create Shape Tween** from the application menu.



The frame span turns green and has a solid arrow between the two keyframes indicating a Shape Tween has now been created.

- We want the black shape to grow out from the center of the Stage at the 1s mark until it reaches its present size



at the 3s mark. Move the playhead to the 1s mark and choose the **Free Transform** tool from the Tools panel.

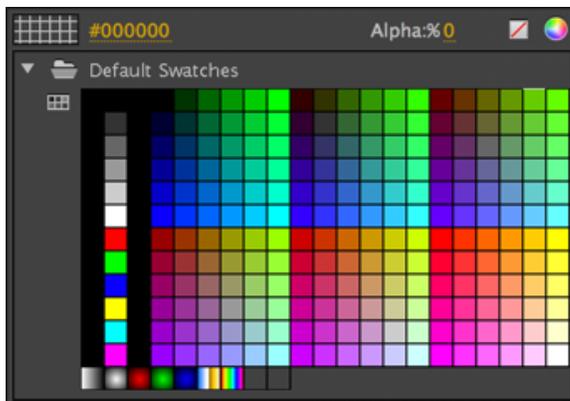
- Click upon the black shape to select it with the Free Transform tool and note that a number of transform handles appear along each side of the shape. Click-drag from the top or bottom toward the center of the Stage to resize the shape until it is between 30 and 40 pixels in height.



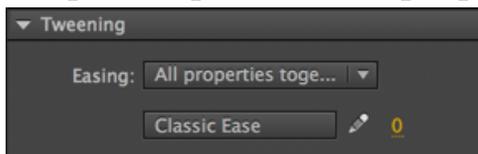
You do not need to be exact, but glancing at the Properties panel will inform you as to the current height.

The small white circle you see is the Transform Point of the object selected. Any transformations done through the Free Transform tool –scale, rotate, skew – will cause the object to transform around this very point.

- With the shape within the keyframe located at the 1s mark still selected, change the color Alpha value within the Properties panel to 0% - this will render the shape completely transparent.

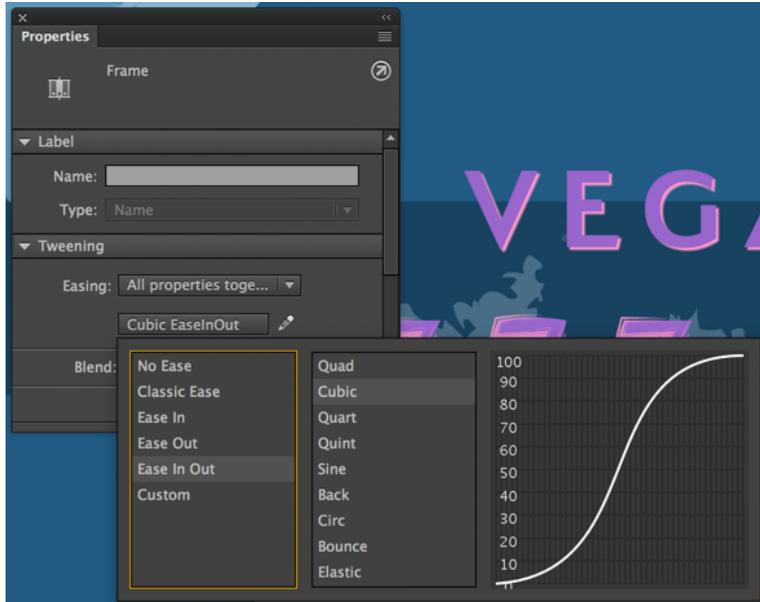


- If you scrub the playhead, you'll see the shape fade in and grow larger from the center of the stage from the 1s mark to the 3s mark on the Timeline. Let's add some easing to our shape tween through the Properties panel.
- Click on any frame that makes up the Shape Tween between the 1s and 3s marks across the Timeline. Now have a peek at the Properties panel to see properties for that frame.



Since the frame is part of a tween – we get the **Tweening** section of the Properties panel.

- From here, click the **Ease Type** dropdown to select an ease preset. There are a number of presets here grouped by category.

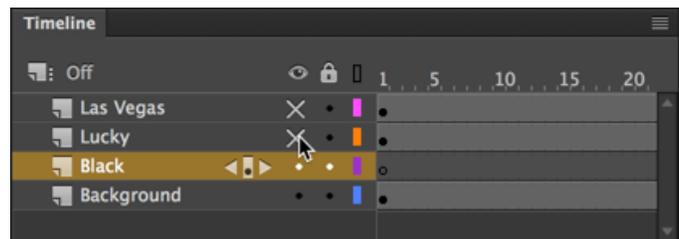


Choose **Ease In Out** and double-click **Cubic** to apply that preset to our Shape Tween.

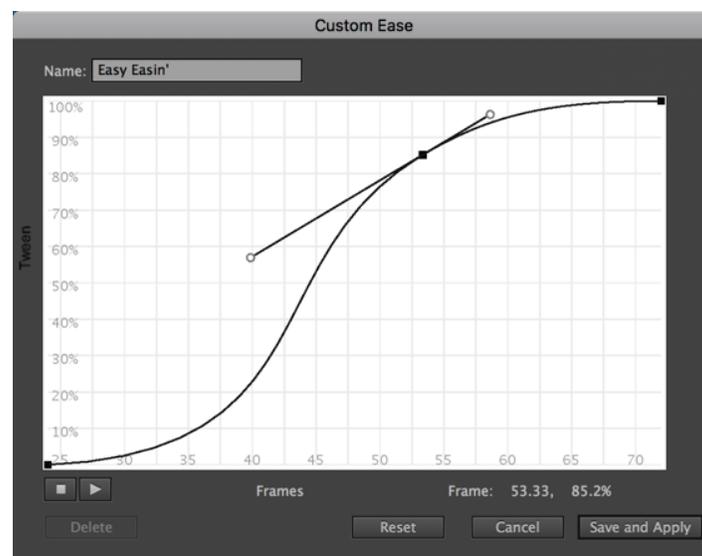
Easing provides a mechanism for more natural acceleration and deceleration between property value changes within a tween. The ease presets make it easy to apply various ease types to our tween.

When scrubbing the playhead to preview the Shape Tween we've created, you will notice that the "Las Vegas" and "Lucky" layers remain visible and are sort of distracting.

If desired, go ahead and hide both of those layers. This will hide them while working in the authoring environment – but if we were to publish right now, Animate would still include them in the published output.



An additional note about Ease Presets... you can modify any of the presets that come bundled with Animate or even create your own from scratch. Just click the pencil icon located next to the **Ease Type** dropdown and the **Custom Ease** panel will appear.



Tweak your ease settings, provide a name, and click “Save and Apply” and it will be available for you to apply to additional tweens in your project!

Creating Graphic Symbols

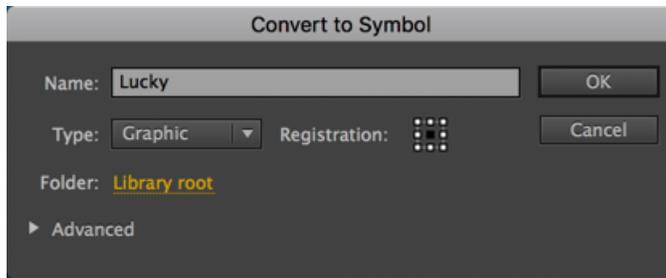
Symbols are re-usable, modular assets in an Animate document that exist within the project **Library**. **Instances** of these symbols are used upon the project Stage. Their internal timeline runs in sync with the main timeline, so it’s easy to pop them onto the Stage and visualize your animation by simply scrubbing the playhead.

We’ll use three instances of a single **Graphic Symbol** to animate our 7s.

- The first thing to do is look over the three “7” drawing objects you currently have on the Stage. Which is your favorite? Use the Selection tool to select and delete the two others.
- Select the remaining “7”.

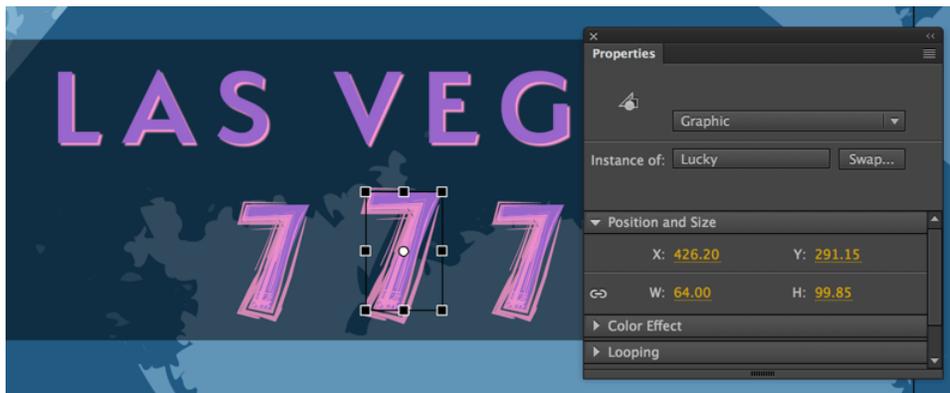


- Create a Graphic symbol by choosing **Modify > Convert to Symbol** from the application menu. The Convert to Symbol dialog appears – this process will take your existing shape artwork and wrap it within a Graphic symbol container.



Be sure that “Graphic” is selected and that you name it “Lucky” – then click OK.

- We now have a single instance of the **Lucky** Graphic symbol on our Stage. You need two more, so either copy/paste the existing instance, or hold the ALT key and click-drag to create additional instances. Instances can also be dragged out of the Library panel and onto the Stage. You have a lot of options here.
- Finally, tweak the positioning and scale of each instance until it looks nice. I’ve centered my instances in a similar fashion to how the 7s were arranged previously. I’ve also scaled up the center-most instance to give it more weight.



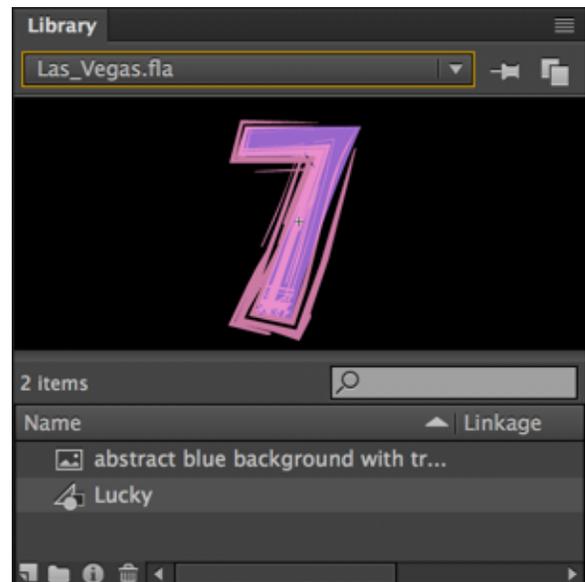
We can see in the Properties panel that each new “7” is an instance of the Graphic symbol “Lucky”.

When a symbol is created, it resides in the project **Library**. The symbol acts as a blueprint to instruct Animate on how to construct an instance

of that symbol. **Instances** are representations of the symbol within the project Stage and Timeline.

You will notice the “Lucky” **Graphic** symbol we created exists within the project Library– as well as the imported bitmap image from Adobe Stock.

You can enter into and edit the contents of a symbol by either double-clicking the symbol itself within the Library panel – or by double-clicking an instance of that symbol upon the Stage. This is what we will be working on next!

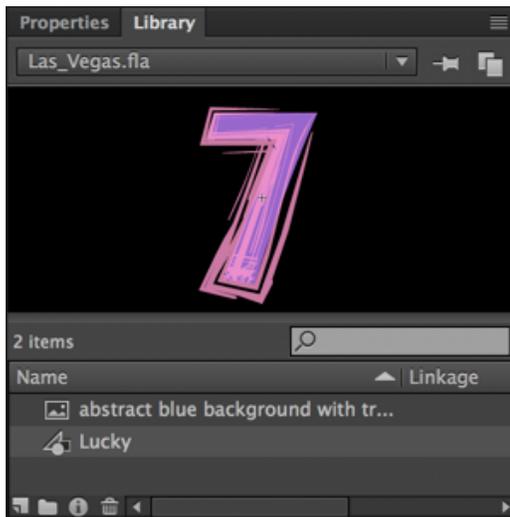


Frame-by-Frame Animation

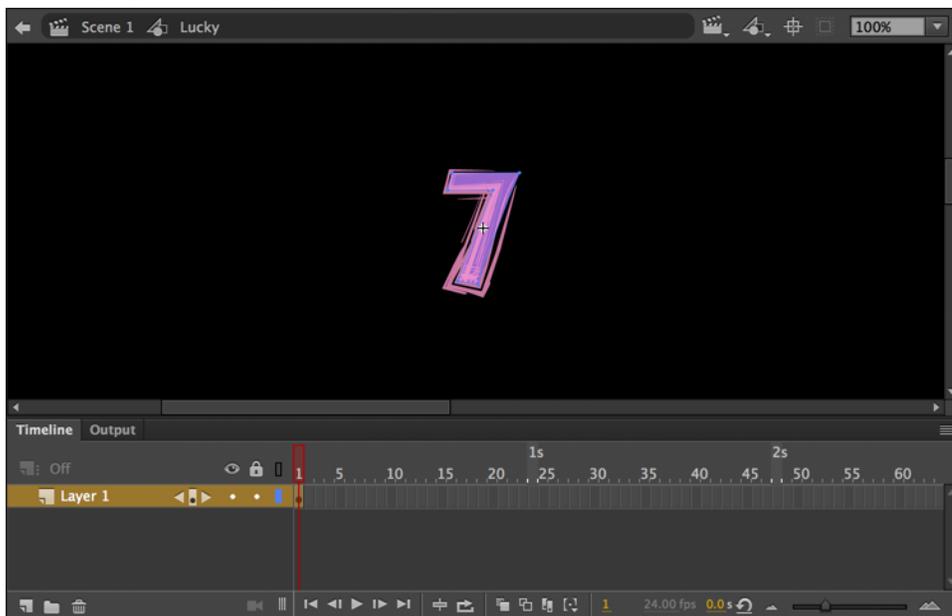
The Frame-by-Frame method of animation is the most manual animation technique... but also the most traditional. There is no tweening involved – just a set of keyframes whose content changes from one to the next to the next...

The result can be jerky, jittery, and altogether random-looking – and this is exactly what we are looking to do with our **Lucky** animation.

- Open the Library panel and double-click the **Lucky** graphic symbol to enter symbol-editing mode.



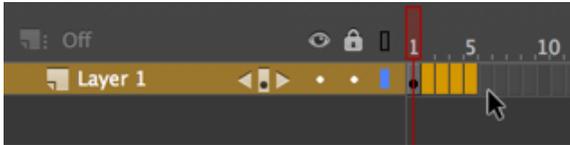
- When within symbol-editing mode, you can manipulate the internal Timeline for that symbol and any artwork contained within it. We know we are within symbol editing mode for the **Lucky** graphic symbol as this is indicated above the Stage.



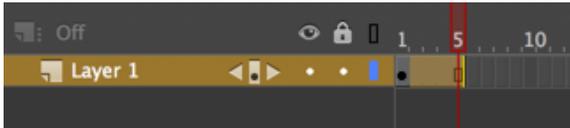
The symbol Timeline works exactly like the main Timeline, with layers, folder, frames, and so on. You'll also notice we can return to

the main Timeline by clicking the **Scene 1** link above the Stage. Don't do that yet though!

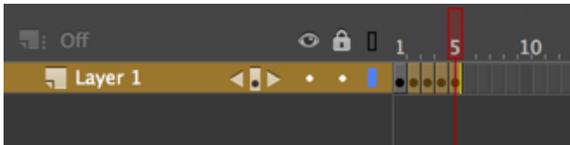
- Let's go ahead and duplicate our keyframe on frame 1 across frames 2-5 as well. Using the mouse, click-drag across these 4 frames to select them all.



- Go up to the application menu and choose **Insert > Timeline > Frame** to create a frame span filling frames 1-5.

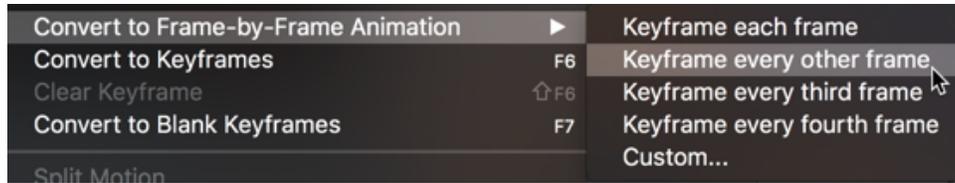


- Now choose **Modify > Timeline > Convert to Frame-by-Frame Animation > Keyframe each Frame** from the application menu.



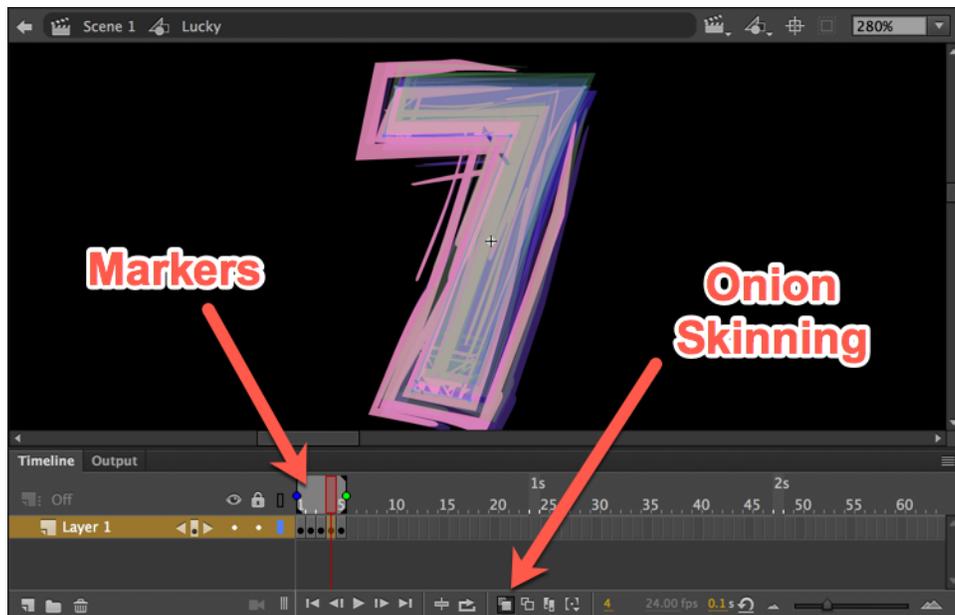
There is now a keyframe residing upon each of the frames. At this point, each keyframe and its contents are all duplicates of the original keyframe on frame 1.

The Convert to Frame-by-Frame Animation command is new. It can be used in many different ways; from converting a tween to a frame-by-frame animation, to establishing a framed structure allowing an animator to more easily animate by 2s, 3s, or whatever frame interval is required.



- Now, we want to step through keyframes 2-5 to make each one a little bit different from the rest. Change position, scale, shape, stroke values, or anything else you'd like in order to make each frame unique – even if slightly so.

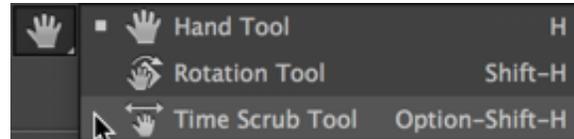
You may wish to enable Onion Skinning to assist you in this task. The controls to enable onion skinning can be found below the timeline. This feature allows you to view a ghosted preview of frame content previous to and after the currently selected frame. It is a tool meant to assist in animating – especially when animating frame-by-frame as we are doing here.



Once you have all 5 keyframes expressing different properties, perform a Test Movie by choosing **Control > Test** from the application menu. You

should have a neat glitch animated effect with each 7 within your project.

Click **Scene 1** above the Stage to return to the main Timeline. Notice that you can also scrub through the Timeline by dragging the playhead directly or by using the new **Time Scrub Tool**.



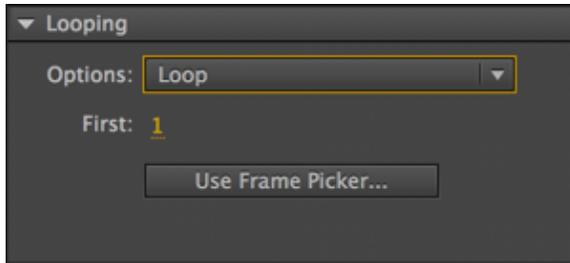
This new tool allows you to scrub the Timeline by click-dragging back and forth along the Stage itself.

Whichever method you choose – note that the Graphic symbol instances on the Stage animate along with the main Timeline. This is a feature unique to Graphic symbols as MovieClip symbols cannot be previewed in this way. In order to see the animation within a MovieClip symbol instance, you must do a Test Movie.

Graphic Symbol Instance Frame Picker

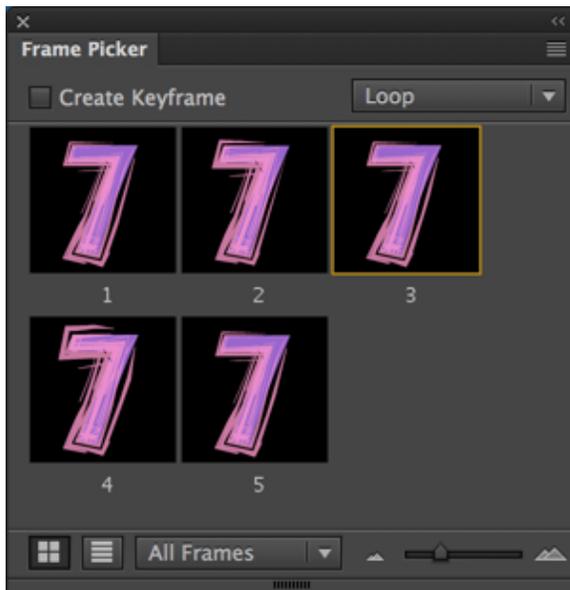
You'll likely have noticed that each instance of the **Lucky** Graphic symbol is playing the exact same frame as each of the others, in sequence. This might look okay depending upon how you've adjusted the properties within each keyframe, but we'll use the **Frame Picker** to stagger these frames across all three instances to introduce greater variability.

- Be sure the **Lucky** layer is unlocked and choose the Selection tool.
- Click on the center instance and have a look at the Properties panel. Pay particular attention to the **Looping** section.



This section is special to Graphic symbol instances and allows you to specify whether the instance timeline loops forever, loops just once, or remains on a specific frame. Be sure that **Loop** is selected in the Options dropdown. We'll leave this instance to begin playback on frame 1.

- Select a different instance of the **Lucky Graphic** symbol. In the Properties panel, click the button labeled **Use Frame Picker...** from the **Looping** section. This will summon the **Frame Picker** panel.



This panel allows you to visually select exactly which frame you'd like the animation to start at – and also provides the options around looping that we have seen previously. Choose a frame other than frame 1 for this instance and be sure it is set to **Loop**.

- Now select the remaining instance and perform the same action – but be sure to choose a different frame number to start on. When you are finished, all three **Lucky** instances should start looping on a different frame, staggering the animation.

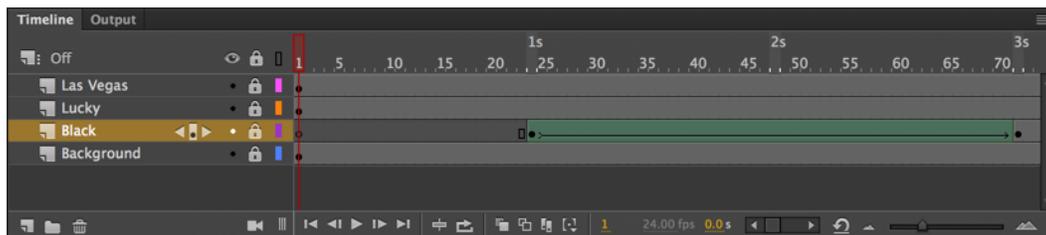
Ok, the **Lucky** instances are all animating independent of one another.



The next thing we need to do is delay the appearance of both the **Las Vegas** layer content and our **Lucky** layer animations.

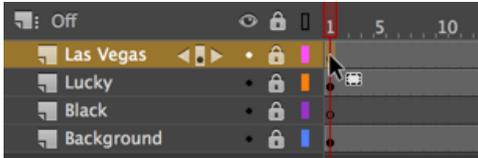
Delaying Visual Assets and Animations

Earlier, we animated our black shape using a Shape tween and had it appear at the 1s mark along the Timeline. Before this point, the black shape simply does not exist and we have a span of empty frames.

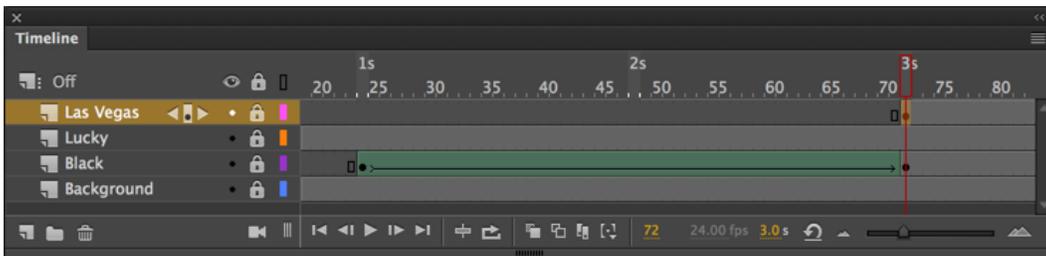


As a result of this, the contents of the **Las Vegas** and **Lucky** layers just exist from frame 1 onward... and we really don't need to see them until much later in the Timeline. The **Black** layer content provides a nice background for them to exist against – and our project looks pretty nasty when absent.

- In the Timeline, select frame 1 within the **Las Vegas** layer by clicking on it. Hovering over the selected layer will reveal a special “box outline” cursor.

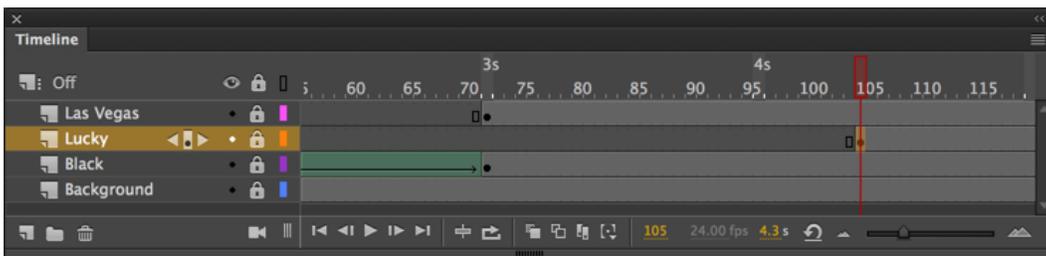


- Now, click-drag the keyframe on frame 1 across to the 3s mark and release the mouse.



The keyframe that previously existed at frame 1 is relocated to frame 72 – as a result of this action, the content of the **Las Vegas** layer will not appear until 3 seconds of animation has elapsed.

- On the **Lucky** layer, we’ll start by performing the same action. Select frame 1 of that layer and then click-drag the keyframe on frame 1 across to frame 105 and release the mouse.



The content of the **Lucky** layer will not appear until frame 105.

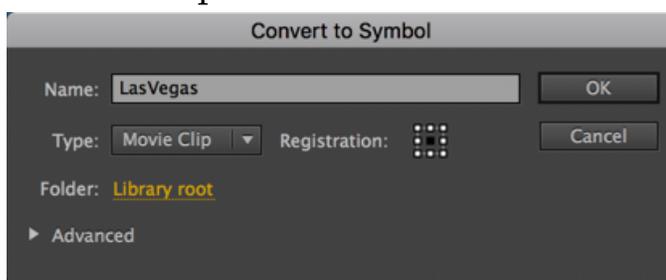
We’ll now move on to animate the “Las Vegas” text using Classic tweens!

Animating with Classic Tweens

Shape tweens are used when dealing with shapes and drawing objects. When dealing with symbol instances, you'll need to employ either **Classic tweens** or **Motion Tweens**. We'll use a Classic tween for this task – but there is no reason you couldn't use a Motion tween. We'll see some information about Motion tweens and the **Motion Editor** after this section.

To convert any piece of content to a symbol we must first select the content.

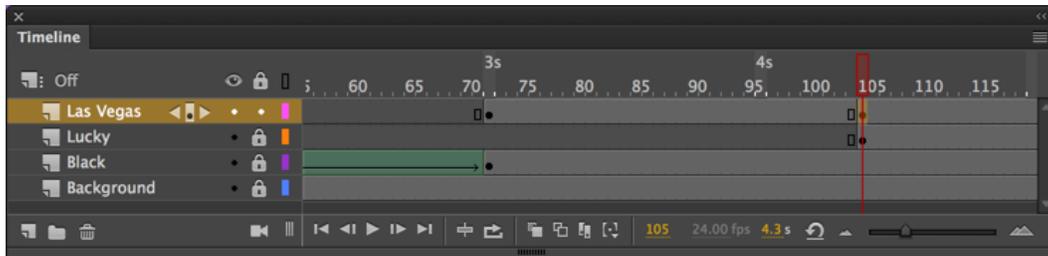
- Using the Selection tool, click on the “Las Vegas” text object at the 3s mark with the **Las Vegas** layer unlocked.
- From the application menu, choose **Modify > Convert to Symbol...** to summon the **Convert to Symbol** dialog.
- Provide a **Name** value of “LasVegas” and a **Type** selection of “Movie Clip” and hit **OK**.



A MovieClip symbol is created and placed in the Library while an instance of this symbol is placed upon the Stage – replacing the previous text object.

A MovieClip symbol has a timeline just like that of the main stage, and is the primary symbol type for creating custom functionality within Animate.

- A keyframe already exists on frame 72. We need to create an additional keyframe at frame 105. Move the playhead to frame 105 on the **Las Vegas** layer and click that specific frame to select it.
- Now go up to the application menu and choose **Insert > Timeline > Keyframe** to insert a new keyframe.

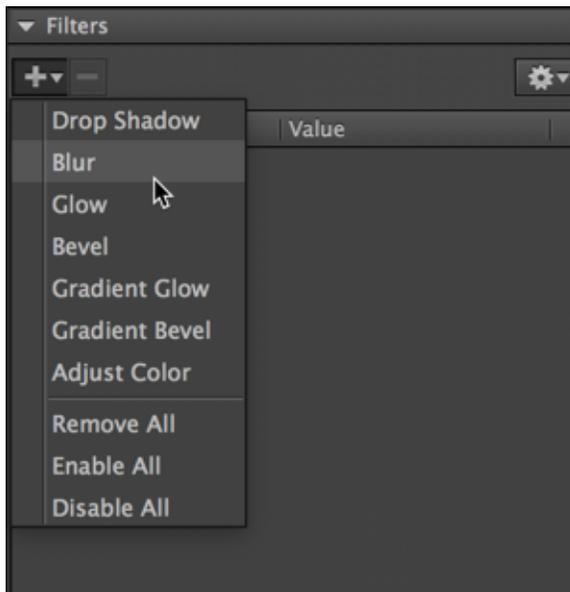


- On frame 72, use the Selection tool to select the MovieClip instance we've created and have a look at the Properties panel.
- Look down to the **Color Effect** section and choose **Alpha** from the **Style** dropdown.



Take the slider value down to 0%.

- Now look down to the **Filters** section. Click the plus icon to add a **Blur** filter to this MovieClip instance.



- With the **Blur** filter added, change the Blur X/Y values to 22px and fix the Quality setting to Medium.

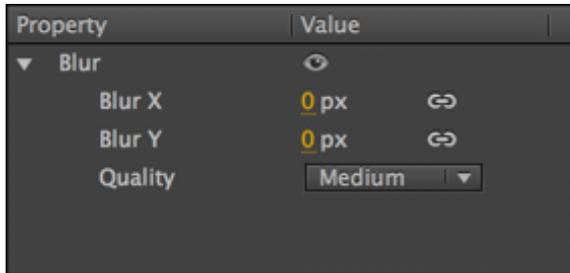


- Now select frame 105 and use the Selection tool to select the MovieClip instance on the Stage and have a look at the Properties panel once more.
- Look down to the **Color Effect** section once more and choose **Alpha** from the **Style** dropdown.

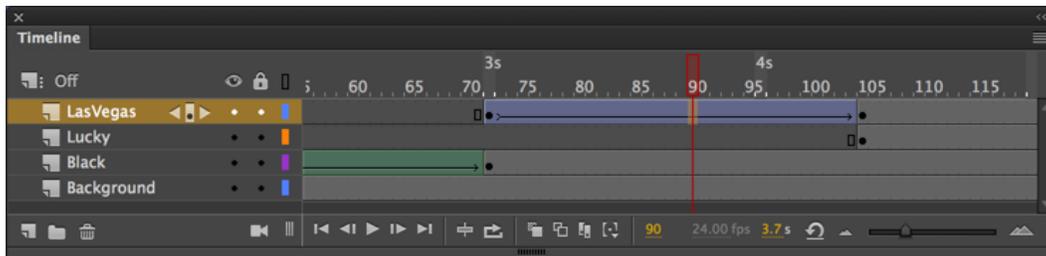


Take the slider value all the way up to 100%.

- Look down to the **Filters** section. There will be a **Blur** filter in place - change the Blur X/Y values to 0px and fix the Quality setting to Medium.



- Now is the time to create a Classic tween to animate between the two keyframes created. Click on any frame between 72 and 105 and from the application menu, choose **Insert > Create Classic Tween**.

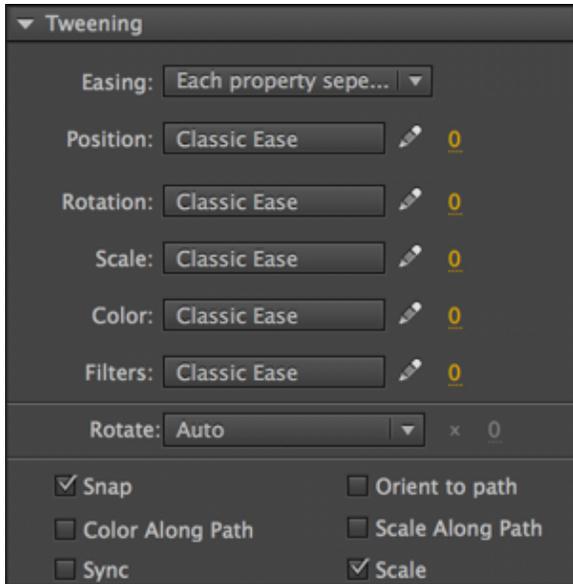


A Classic tween will be created and both Alpha and Filter properties will be smoothly tweened between these keyframes.

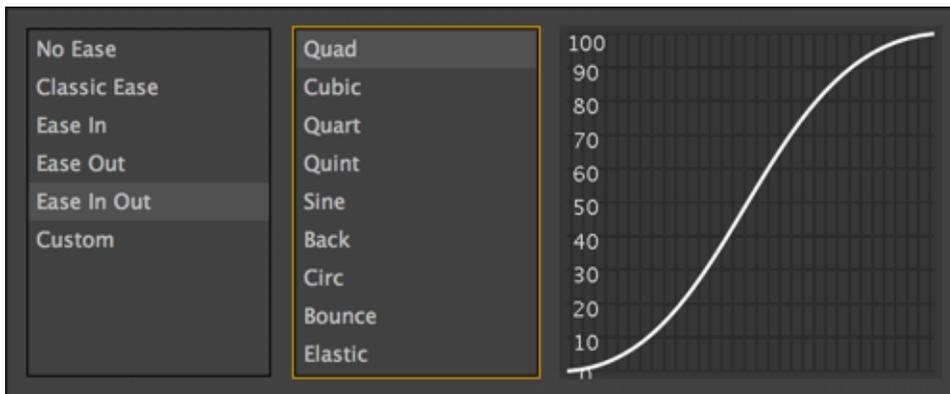
The Las Vegas content now fades in and blurs together across the background elements of our project. There are still some tweaks we can consider though!

Ease Preset Options for Classic Tweens

As we saw when creating a Shape tween earlier, Animate CC now includes **Ease Presets** for both Shape tweens and Classic tweens. When using Classic tweens, you have the additional option of setting ease presets and custom eases for each individual property to be tweened.

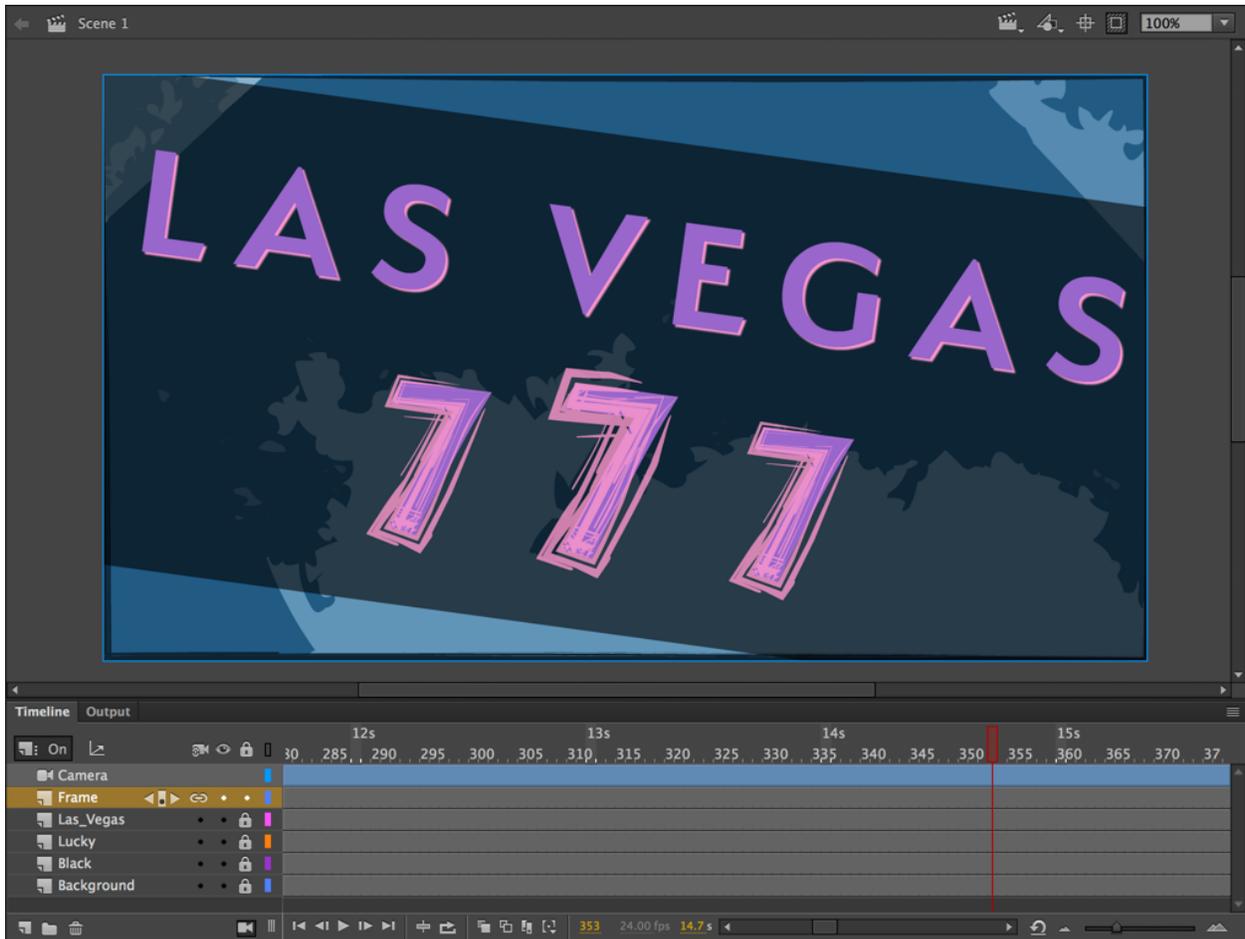


Just choose **Each property separately** from the **Easing** dropdown. Again, this is only possible with Classic tweens and not Shape tweens.



Go ahead and apply some easing to your Las Vegas Classic tween. I'm using a **Quad EaseInOut** as **All properties together** for simplicity.

Animating with the Virtual Camera and Advanced Layers Mode



We'll now completed the animated portion of this workshop by employing Advanced Layers Mode and the Virtual Camera. This allows us to animate the entire project view in a variety of creative ways. Using the Camera tool in Animate CC, we can zoom, pan, and even rotate our entire scene as a whole unit.

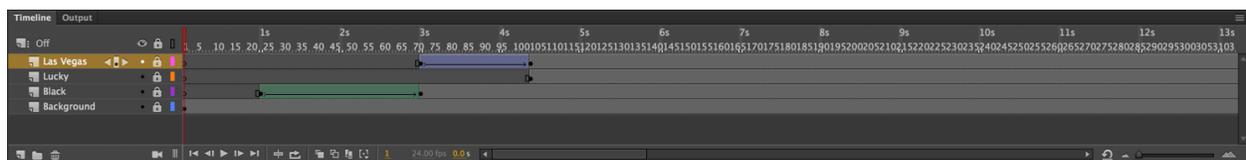
Using the Virtual Camera

The new **Camera** functionality present within Animate CC allows animators to simulate the user of a real-life camera. Previously, animators relied on third party extensions of varying quality and compatibility, or modified their animations to mimic a camera's movement.

Animators can use the following camera features that are integral to any motion film for efficient storytelling:

- Panning with the subject of the frame
- Zooming in the object of interest for dramatic effect
- Zooming out of a frame to remind the viewer of a larger picture
- Modify the focal point to shift the attention between subjects
- Rotating the camera view
- Apply color effects and tint to an entire scene
- Attach layers to the Camera
- Create expressive parallax effects with Layer Depth

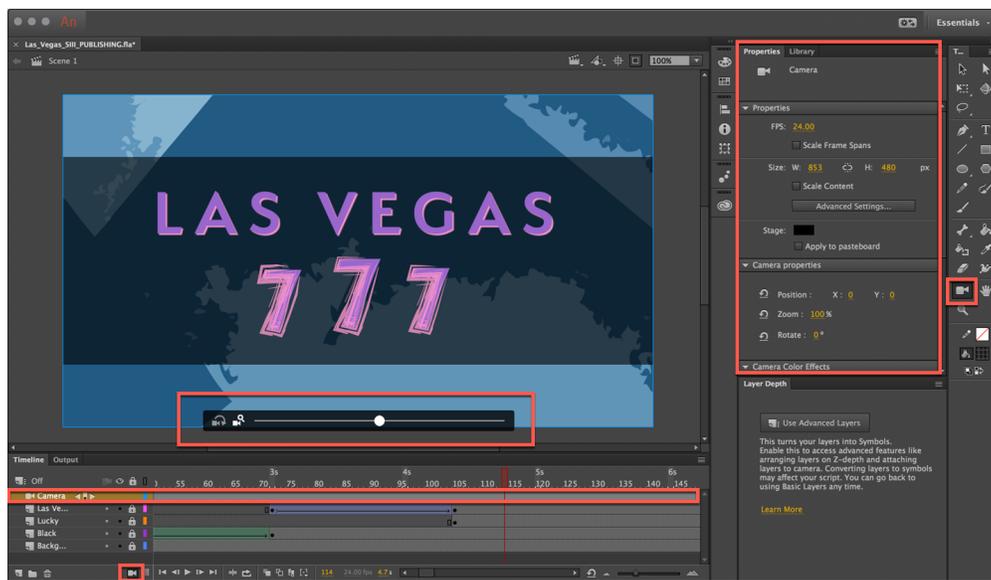
You may have noticed that even though we have frames along the Timeline all the way to the 16s mark, we currently only have individual elements animate up until about 5 seconds in.



Let's use the Camera to animate the entire Stage view for the remaining 10 seconds we've allotted for.

- The first thing to do when employing the **Camera** tool is to choose it from the **Tools** panel. Click the icon  that looks like a camera.

This enables the **Camera** for the document, displaying a special Camera layer within the Timeline, Camera overlays on the stage, and Camera Properties.



The Camera-related items mentioned are highlighted in the figure above.

Note that you can also toggle the Camera on and off with the Camera icon at the bottom of the timeline.

When you set a camera view for your composition, you look at the layers as though you were looking through that camera. The camera layer behaves like a regular object; you can add tweens or keyframes... and animate it like other objects in the document.

With the Camera enabled for our document, we can now create some animated camera movements across the entire Stage.

Animating the Camera with Motion Tweens

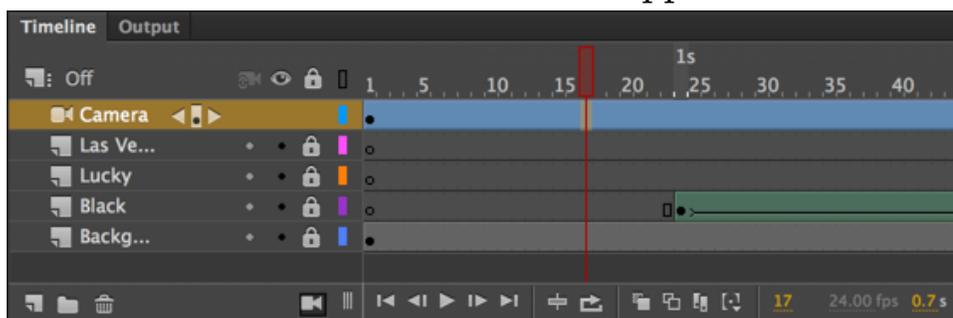
You animate the Virtual Camera in much the same way as you would animate a MovieClip or Graphic symbol instance – that is... you can either set keyframes at various frames and employ Classic tweens... or you can use a **Motion tween**.

A **Motion tween** is probably the easiest tween type to use of the three primary tween types that exist in Animate CC (four counting IK armatures). The reason I state this is that unlike the two other tween types, there is no need to expressly insert keyframes across the Timeline.

To create a Motion tween, select a frame which contains a symbol instance and choose **Insert > Motion Tween** from the application menu. Then simply move the playhead around and tweak the properties of your symbol instances – Animate does the rest!

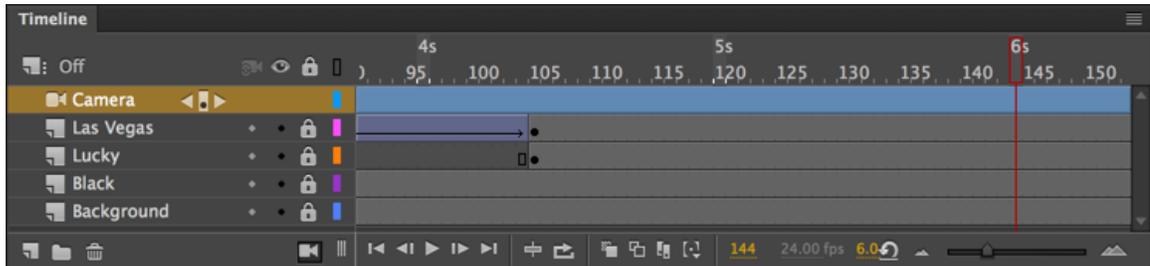
Since we have not dealt with Motion tweens yet, we will be using a **Motion tween** for the Camera animation.

- Right-click anywhere within the **Camera** layer and choose **Create Motion Tween** from the menu that appears.

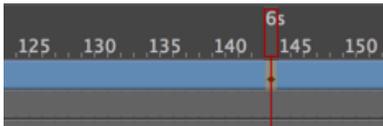


The **Camera** layer will turn blue, indicating a Motion tween has been created. Ready to animate the Camera!

- We want the animation to run as we've established it already with the Camera remaining still and at 100% zoom until about 6 seconds into the animation – so move the playhead to the 6s mark.



- We want the Camera properties to be duplicated on **frame 1** and at **frame 144**. From the application menu, choose **Insert > Timeline > Keyframe** to insert a new keyframe at the 6s mark.



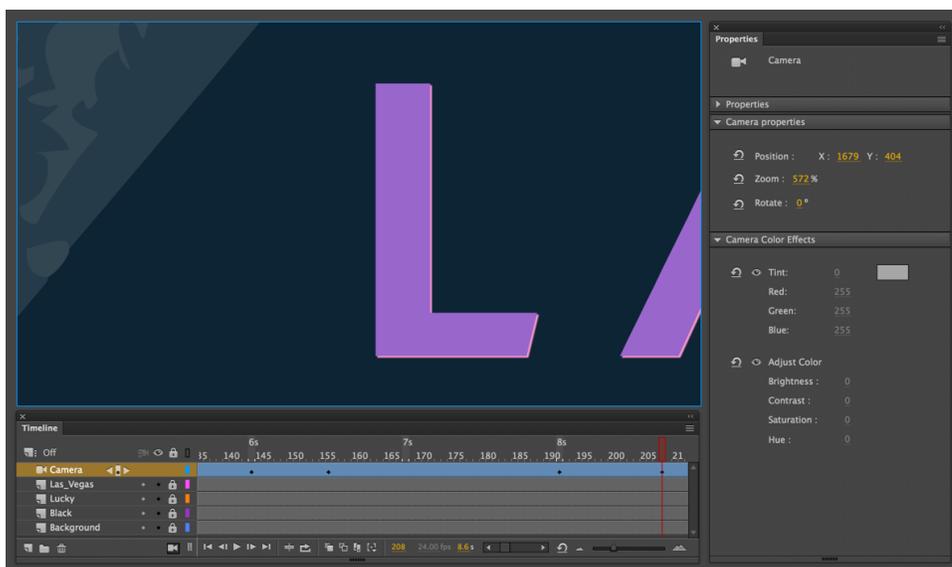
A diamond-shaped keyframe indicator appears. This ensures the Camera properties will not change between the start of the animation up until 6 seconds have elapsed and our initial animation is complete.

- Move the playhead to **frame 156**. Zoom in until the “777” takes up the majority of the screen and rotate it slightly. You may need to pan a bit as well. You can either use the Camera overlay controls or

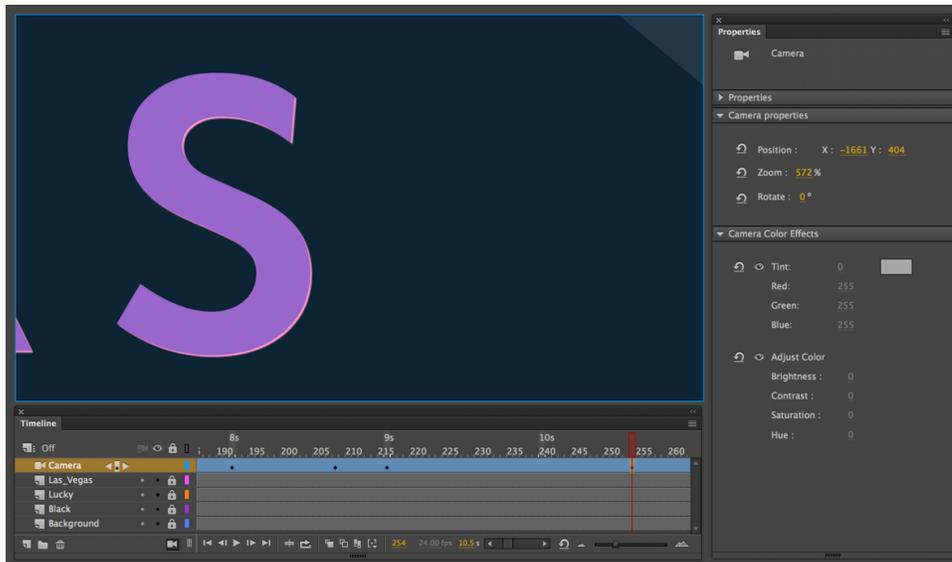
the Properties panel to make these changes as we move along.



- Then move to **frame 192**. Choose **Insert > Timeline > Keyframe** to insert a new keyframe. This will create a pause in the Camera motion.
- Move to **frame 208** and use the Camera controls to position the view at the beginning of the **Las Vegas** instance. Use a combination of pan and zoom to accomplish this. Reset the rotation back to 0.



- Now move the playhead to **frame 216**. Choose **Insert > Timeline > Keyframe** to insert a new keyframe. This will create a pause in the Camera motion.
- Move to **frame 254**. If you hold down the SHIFT key and pan across to the end of the **Las Vegas** instance, the Camera view should pan along a straight line.



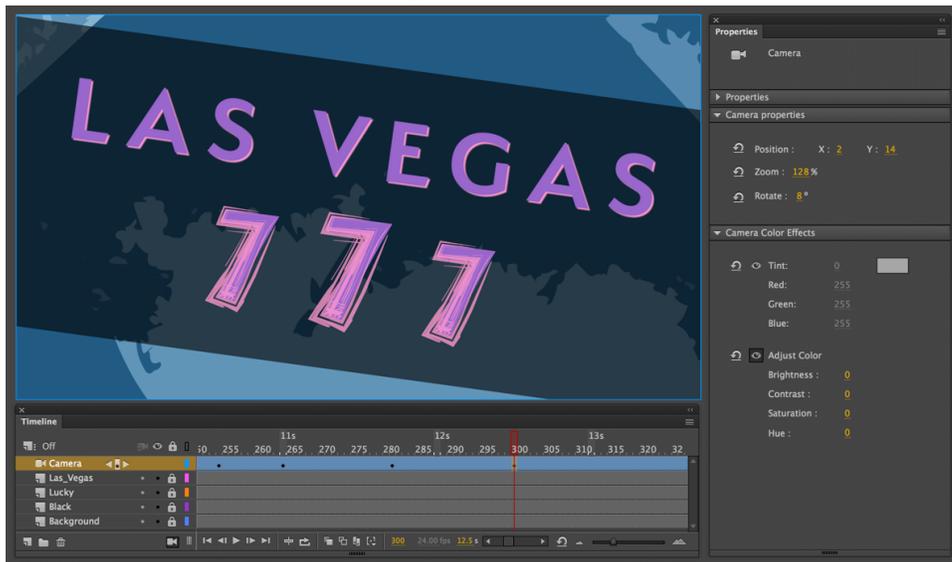
You shouldn't really have to adjust any other properties for this step.

- Now move to **frame 264**. Choose **Insert > Timeline > Keyframe** to insert a new keyframe. This will create another pause in the Camera motion.
- Move to **frame 281** and pull the zoom back so that nearly everything is in view once again. Add some rotation once again to

skew the resulting view.



- Next, move to **frame 300**. Choose **Insert > Timeline > Keyframe** to insert a new keyframe. In the Properties panel, choose to enable **Adjust Color** by clicking the “eye” icon right next to it. Keep all values at the default of 0. This will set up a keyframe from which we can animate color adjustments.



Note that if you are using a document type other than ActionScript 3.0, this option will not be available.

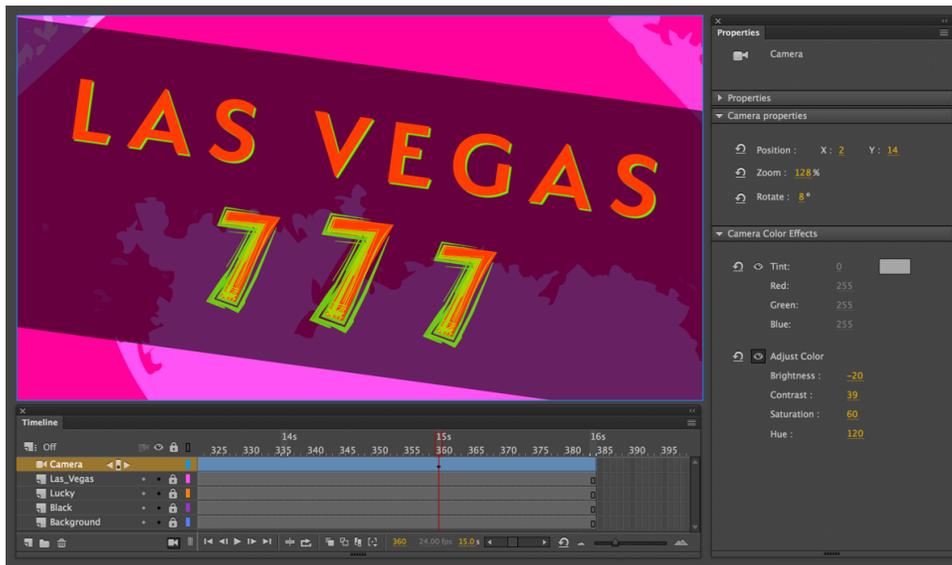
- Okay – last step! Move the playhead to **frame 360**. Under the **Adjust Color** section, set the following values (or just play with them until you are satisfied):

Brightness: -20

Contrast: 39

Saturation: 60

Hue: 120



Things will likely look *very* different when you are finished!

Go ahead and test out your project by choosing **Control > Test** from the application menu. The SWF will compile and open within Animate for you to preview. If playing back from the Timeline, you may want to turn outlines on at this point as Animate may play more slowly with so much going on.

Attaching Layers to the Camera

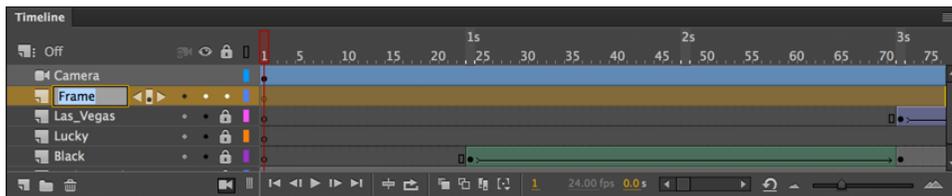
The new release of Animate CC allows us to attach specific layers to the Camera. This enables things like text overlays, design elements, and user

interface objects that are completely unaffected by the movement, effects, and properties of the Camera.

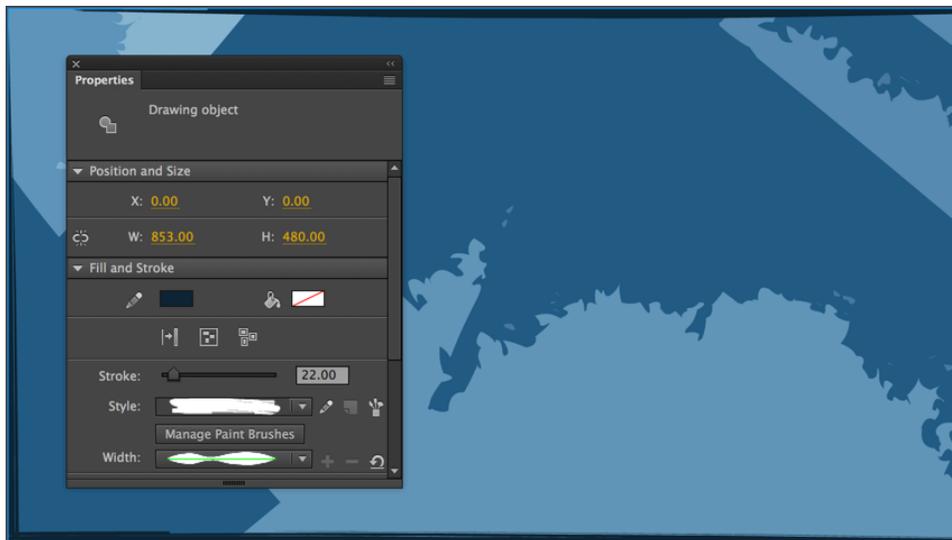
To utilize this feature, we'll need to enable **Advanced Layer Mode** – a new layer mode in Animate that basically converts layers into symbols to enable advanced **Camera** and **Z-Depth** functionality!

Before enabling this new layer mode, let's create a neat framing element we can attach to the Camera.

- First, create a new layer and provide the layer name of “Frame”. We'll be creating content within this layer so keep it unlocked.



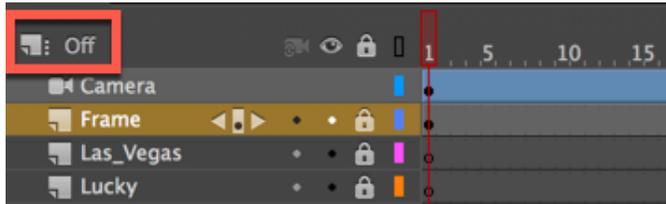
- Choose the **Rectangle tool** from the Tools panel and draw a rectangle across the entire Stage. You can make the frame appear however you like, but it makes sense to have a transparent fill so the content is not obstructed.



I've used a stroke with a value of 22 and a variable width profile

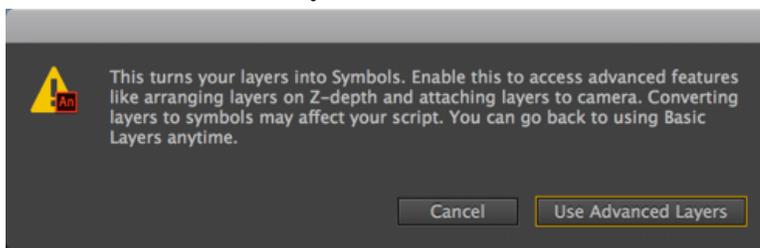
along with a vector art brush and color sampled from the project and darkened via the color picker.

- Time to flip the switch! Activate the new **Advanced Layer Mode** via a toggle located above the layer stack in the Timeline.

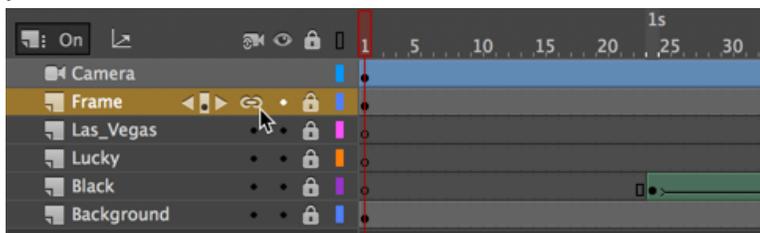


You can go ahead and lock this layer down now, if desired.

- A warning dialog will appear. It explains what **Advanced Layer Mode** does and presents you with a choice. Go ahead and choose **Use Advanced Layers**.



- Now to attach our **Frame** layer to the Camera. All you need to do is click the dot beneath the Attach to Camera column in the layer you wish to attach.

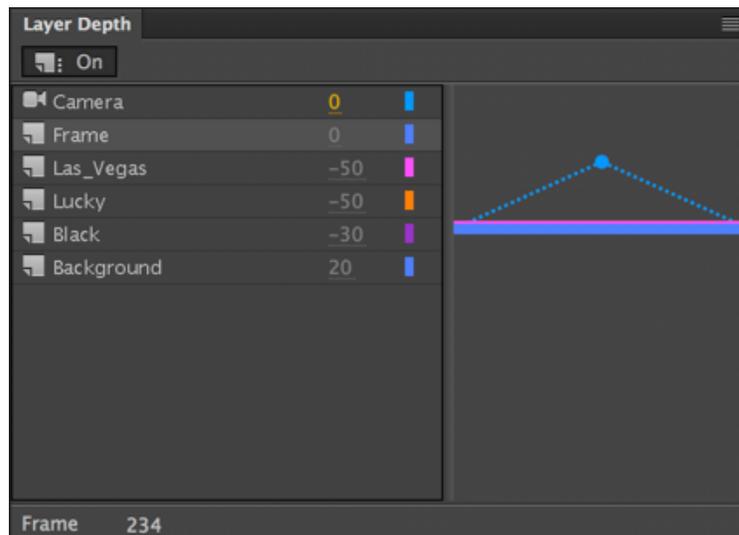


A chain link icon will appear. The **Frame** layer is now attached to the Camera.

Go ahead and scrub the playhead across the Timeline to see this in action. You can attach multiple layers to the Camera and detaching them is just as simple.

Specifying Layer Depth

Activating the **Advanced Layer Mode** will also enable the ability to specify z-depth for each layer (and even folders) via the **Layer Depth** panel. You can access this panel from an icon  that appears next to the **Advanced Layer Mode** toggle switch or from the application menu under **Window > Layer Depth**.



This presents a *major* new set of capabilities within Animate CC. It allows you to specify how near or far any specific layer is in front of or behind the Virtual Camera. In addition, **Layer Depth** is completely tweenable - allowing objects to move across layers as an animation progresses.

Using **Layer Depth** along with the Virtual Camera allows for some really neat effects in your animations that you would have had to create in a much more manual way previous to this release. This allows the

creation of depth and parallax effects easily by simply using the Camera against layers of varying depth.

Not only are all these neat Camera and Layer Depth features part of the Animate CC user interface... but you can hand control of the Camera over to the user or perform programmatic effects with the powerful **VirtualCamera API!** This API is available in ActionScript 3.0, HTML5 Canvas, and WebGL... and even has functionality built into the new **Actions Wizard** for HTML5 Canvas!

SEQUENCE III:

Multiplatform Publishing

Exporting Content from Animate CC

When you are happy with the state of your animated project, you'll want to get it out of Animate CC and into some format which a user can either view or interact with. This could be a video file, native web content via the HTML5 canvas element, or even a fully realized mobile application for Apple iOS.

There are two main ways of doing this – exporting and publishing. As an example of the differences here... you can export an animated GIF from basically any document type you choose – ActionScript 3.0, HTML5 Canvas, WebGL... the document type doesn't really matter much. However, if you want to publish to Flash Player then you must do so from an ActionScript 3.0 document type. If you want to publish an interactive for the native web – you'll likely choose HTML5 Canvas as your document type. This will become more clear as we move on and see some examples – so let's do that now.

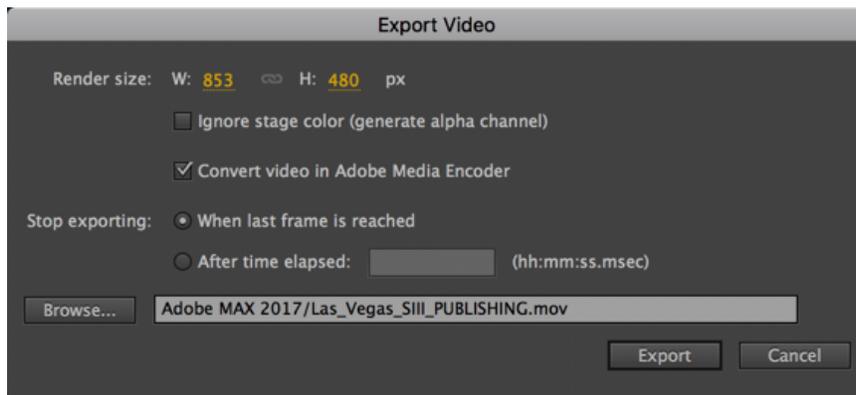
HD Video and Adobe Media Encoder

If we render our project to a web-friendly video format like MP4 – we don't have to worry about the limitations of certain platforms. Just make an awesome little video!

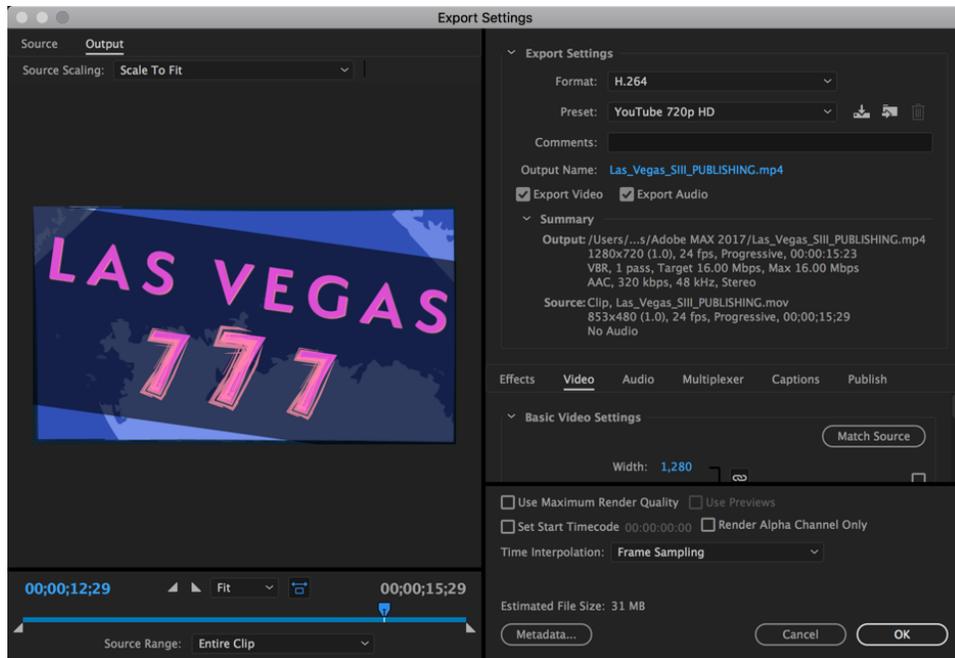


One of the major uses of Animate is to design and produce content for television and video... so we are good! The fact that an ActionScript 3.0 based document type is generally the best choice for video simply due to the number of extra tools and filters that are available should not be overlooked.

Choose **File > Export > Export Video** and then make some decisions around where to export the video and at what resolution. Animate does a frame by frame rendering of the content... so what ends up in the video is precisely what was authored. It even takes into account dynamic animation through the use of code!



Once the video is rendered, we can fire up **Adobe Media Encoder** to render it to any format desired. You can also just import the video to an **Adobe Premiere Pro** or **After Effects** project.



Since video cannot be interactive, you'll want to remove the Actions and UI layers of our project before your render it as video.

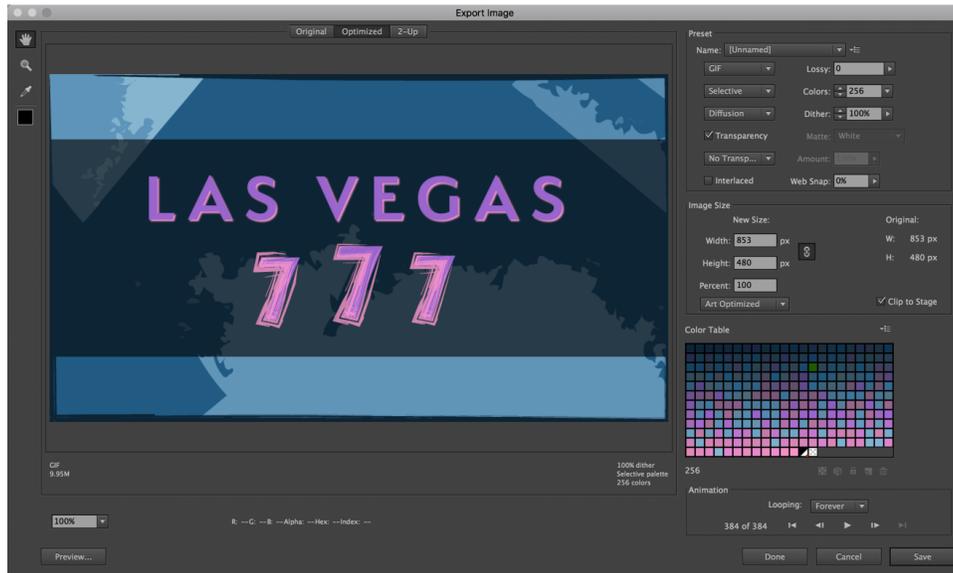
Animated GIF and other Image Formats

Select **File > Export > Export Animated GIF** and a dialog will appear. You can apply individual preset, image, size, color, and animation settings to the optimized and 2-UP views and use the various preset options to set various parameters.



- Select **Matte** to modify and blend the background color across the edges of the asset.
- Select **Transparent** or **non-transparent** to work with the color picker and the transparent option to control the transparency of the asset.
- Set the **Image size, color and animation options**.

The file size values are displayed below the screen and clicking the **Preview** button allows you to preview the animation and navigate to a specific frame. Additionally, you may choose **Select Clip to Stage** in the image size panel to set the clipping boundary to the stage.



When ready, select **Save** to save the animated GIF!

When you save an optimized file using the **Export Animated GIF** option, you can generate an HTML file for the image. This file contains all the necessary information to display your image in a web browser.

Since a GIF file cannot be interactive, you'll want to remove the Actions and UI layers of our project before your render it as video.

Note that in certain export formats, advanced features may need to be disabled in order to produce effective content.

Publishing Animate CC Documents

Now that we've explored some options for exporting content from Animate, let's have a look at the publish options for different platforms. This is very different from the export capabilities which are generally shared among all document types – as the option to publish is restricted to a certain document type... although the ActionScript 3.0 document type can actually publish to a number of platforms including Flash Player, AIR, Windows, macOS, Android, iOS, and more... all from the same document type!

Adobe Flash Player

Animate CC has deep roots in ActionScript and the Adobe Flash Runtimes: Flash Player and AIR. This should come as no surprise... for the majority of its lifetime – this was really the only supported platform!



As a result, choosing any of the AS3-based document types allows for the fullest set of creative tooling and the use of a deeply mature programming language – ActionScript 3.0.

A great example from this workshop project is the Camera. Using the Camera in an ActionScript 3.0 document gives you access to Tints and Color Effects!

SWF Archive

You can also publish all of your layers as separate SWF files to be imported and further composited in an application like After Effects.

For this, you can bundle everything within a **SWF Archive** from **Publish Settings**.

Projectors

Sometimes, it's really useful to have a standalone playback for your Flash Player content. A projector is a small application which is basically your SWF content that is bundled with a standalone version of Flash Player for Windows or macOS.

Adobe AIR for Desktop and Mobile

AIR is hugely popular for building real apps across both desktop and mobile targeting Windows, macOS, Linux, iOS, and Android.

While targeting Flash Player, Video, Canvas, or WebGL... we could just convert the document and just tweak things here or there... creating a native mobile app is a bit of a different story.

We'll probably want to rearrange the various elements on the Stage to accommodate the device resolution, as mobile apps are generally full screen experiences. Additionally, we'd most likely want to create a true app experience around such content, not have a simple animation be the entire app.



Regardless – Adobe AIR is a great target for desktop and mobile app development that you can consider when working in Animate CC!

HTML5 Canvas

Content created in Animate can now to be published directly to a number of modern HTML5 formats. The primary document type which allows this publish target is HTML5 Canvas and this is what we have used throughout this workbook.



When creating content using this publish target, Animate will leverage the CreateJS JavaScript libraries to output an entire animation or interactive project for the HTML5 canvas element. You can even write JavaScript in the Actions panel!

Publish Settings can be easily accessed from the Properties panel and include a LOT of different options. You can actually perform a true publish through the **File > Publish** option in the application menu.

We will be converting our project into an HTML5 Canvas document in the next sequence.

OAM Package

You can also choose to publish a number of other assets from an HTML5 Canvas document from **Publish Settings**, including an OAM bundle you can use in Dreamweaver, Muse, or InDesign.

WebGL (preview)

The secondary native HTML5 document type involves the use of WebGL (Web Graphics Library) technology. Interestingly enough, WebGL also makes use of the HTML5 canvas element... but in a different way.

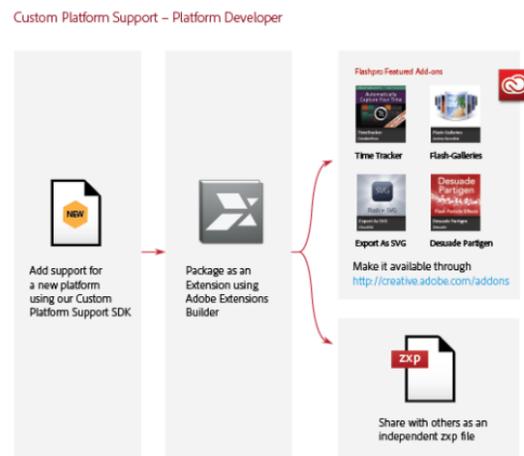


WebGL content is actually GPU-accelerated, meaning that it makes direct use of the hardware GPU (Graphical Processing Unit) instead of sharing rendering tasks with everything else running on the CPU (Central Processing Unit). This makes for a much more effective and powerful rendering target.

While WebGL can be made interactive with JavaScript, the code syntax varies quite a bit from what we've seen using HTML5 Canvas.

Custom Document Types

As we have seen, Animate supports creating rich assets and animations using native document types such as ActionScript 3.0, HTML5 Canvas, and WebGL. The Custom Platform Support feature helps to extend the power of Animate CC to support platforms that are not native to Animate.



Using the Custom Platforms SDK, a developer can integrate his or her own platform as a real FLA based target. By installing a platform support plugin for a new document type, Animate users can create their assets using the rich feature set of Animate and publish it in the output format of the custom platform!

SEQUENCE IV:

Adding Interactivity

Making your Content Interactive

While animation is certainly a valid use for Animate CC, the application is great for implementing interactivity as well. In this sequence, we'll wire in some user interaction using JavaScript and the Actions panel to control the Timeline playback and allow some mouse interaction through Components.

ActionScript vs. JavaScript

For nearly 20 years, the programming language used within Animate (Flash Professional) projects has been ActionScript and solely ActionScript. So why the apparent shift to JavaScript? Well, it isn't exactly as simple as that.

For one thing, Animate CC is *not* a JavaScript-only application. You use JavaScript in projects which require it, such as HTML5 Canvas and WebGL... but you can still use ActionScript on projects which are to be published for Flash Player or AIR, including AIR for Android and iOS – and Adobe continues to update these runtimes.

It's important to note that Animate is not a JavaScript and web-centric application – but is rather a platform and language agnostic application for working across a variety of targets.

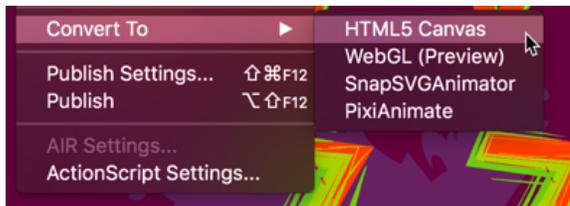
Build your content in Animate CC and use it across all target platforms!

Converting to HTML5 Canvas

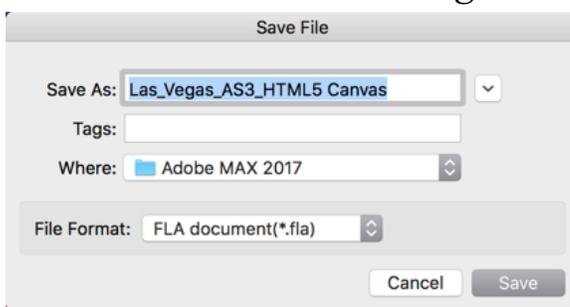
Our project is using an ActionScript 3.0 document type and targets Flash Player. This format has served us well in creating cool effects for video and such... but now we want to distribute this project as native Web content.

Let's go ahead and convert our document to HTML5 Canvas and add some interactivity with JavaScript!

- With the existing ActionScript 3.0 project open, choose **File > Convert To > HTML5 Canvas** from the application menu.



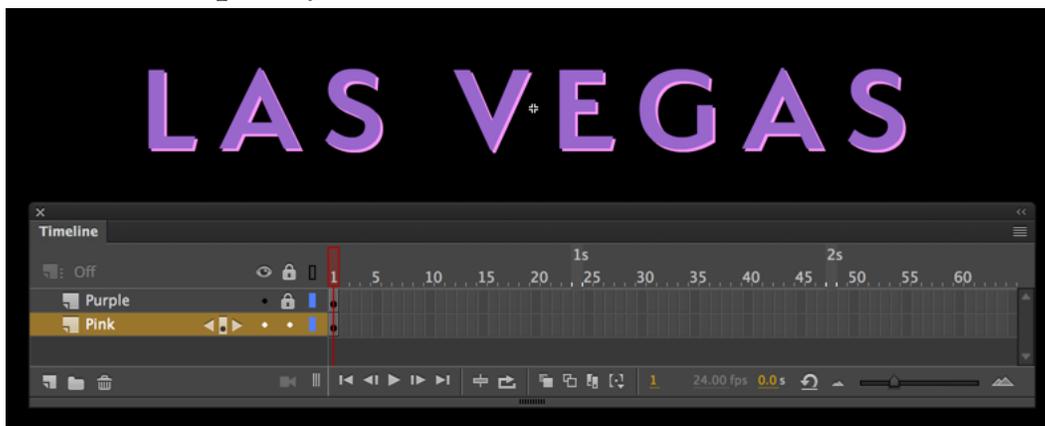
- A **Save File** dialog will appear which prompts you to save a new .FLA document which targets HTML5 Canvas.



Name it whatever you like and click **Save**.

- A new document is created alongside the existing one. Note that your original document is completely preserved and not changed in any way.

- You will notice a warning in the **Output** panel:
For StaticText, Drop Shadow filter is not supported in HTML5 Canvas document, and has been removed.
- Okay – so there are certain differences between different platforms to deal with. No big deal. Enter the **Las Vegas** symbol from the **Library** panel – you’ll be in symbol editing mode.
- We’ll now duplicate the layer by dragging it into the new layer icon below the Timeline.
- Rename the top layer to “Purple”, the bottom layer to “Pink”, and lock the **Purple** layer.



- Using the **Selection** tool, select the text object on the Stage and change the color to a hot pink.
- Now, use your arrow keys to offset the **Pink** layer text content so it appears similar to the previous drop shadow effect. There is always a way!

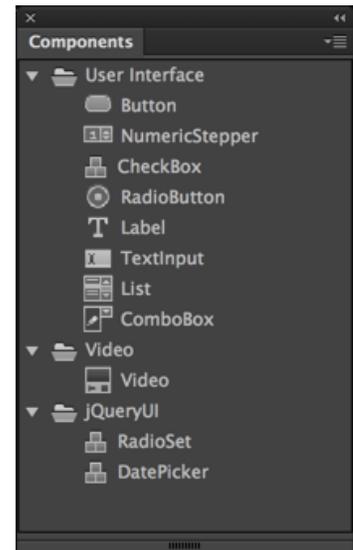
Go ahead and exit out of symbol editing mode and back to **Scene 1** – the main Timeline and Stage.

If you test the animation in a web browser, you'll notice the "Las Vegas" animation remains blurred and never comes into focus. This is because when using filter effects in HTML5 Canvas, the initial effect state is cached and does not animate. To get around this, simply remove the effect from your filters stack.

Add a Button Component

When using HTML5 Canvas as a target document type, you can now use components.

If you open the Components panel using **Window > Components** from the application menu, you will see a list of components for use with the active target platform. Since we are using HTML5 Canvas as our target document type – we see components for use with HTML5 Canvas.

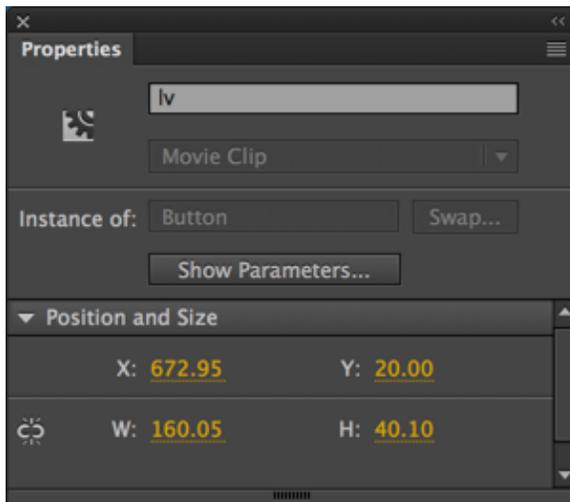


To use any of these components, we can drag them onto the Stage and then, with the component instance selected, modify the component properties through the **Properties** panel just as we do with regular symbols.

We will be using two components for this project... **Button** and **CSS**:

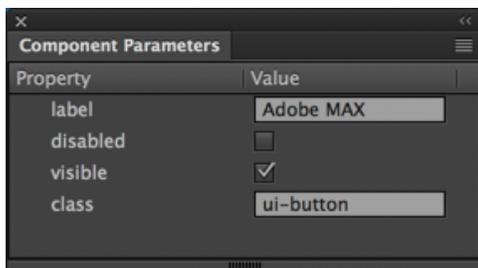
- Create a new layer named **Button** to hold our component instances.
- Drag a **Button** component to the Stage, ensure it is selected, and provide an instance name of **lv** in the **Properties** panel. Instance names work with component instances just like **MovieClip** instances.

- With the **lv** instance still selected, set the width to **160** and the height to **40**.

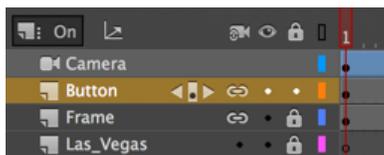


Then, click the **Show Parameters...** button to show the **Component Parameters** panel.

- Change the label value to “**Adobe MAX**” and keep the class value at “**ui-button**” – this will be important for the CSS later on.



- We want the **Button** component instance to be attached to the Camera view – so be sure and do this before moving on.



- Now drag an instance of the **CSS** component onto the stage. It doesn't require an instance name and can be placed anywhere you

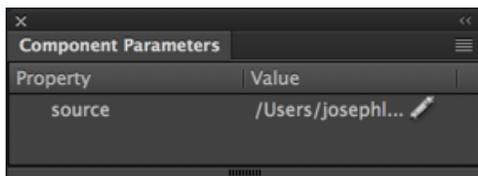
like – as it will not be visually rendered on publish. The **CSS** component only exists to include an external **.CSS** file.

We'll now need to link up a **.CSS** file into our **CSS** component instance. One already exists named *custom.css* in the provided files with the following class defined:

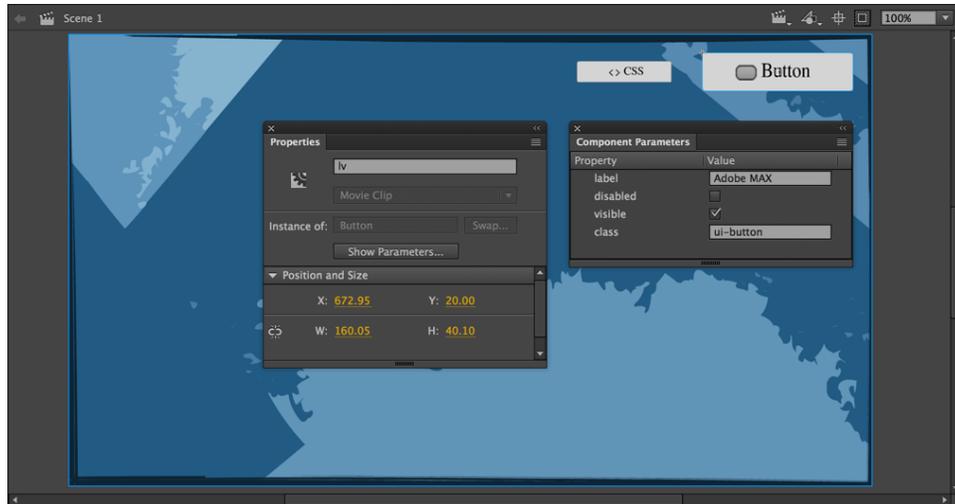
```
.ui-button {  
    background-color: #E730D8;  
    color: #E0B2DD;  
    font-size: 3rem;  
}
```

Let's apply this **.CSS** file to our project to style our **Button** component instance.

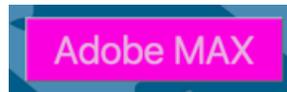
- Click the **CSS** component instance on the Stage and look to the **Component Parameters** panel. The attributes will shift to reflect those of this particular component.
- The only editable property is that of **source**. Use the file browser to locate *custom.css* and the **Component Parameters** panel will then reflect this inclusion.



There are now two component instances on the Stage and both have been configured for our project. Note that we haven't written any interactive code as of yet.



If you perform a **Control > Test** you will see how the **Button** component instance receives and displays the styles from our **.CSS** file and also reflects the label value we've provided.



Note that any components used in an HTML5 Canvas document will exist in a DOM layer above the canvas element. There is no way around this.

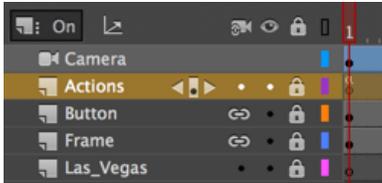
The Actions Wizard

The final few things needed for this project is to allow the project to stop when it reaches the end of the Timeline, and to add some interactivity to our **Button** component instance.

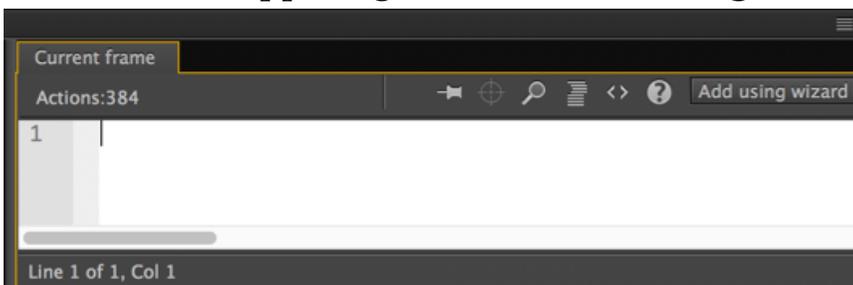
In the past, we'd have to either know how to write this code by hand or through manipulation of the **Code Snippets** panel. With the newest release of Animate CC, we now have access to the new **Actions Wizard** functionality to assist us with writing interactive JavaScript.

Let's set this up now!

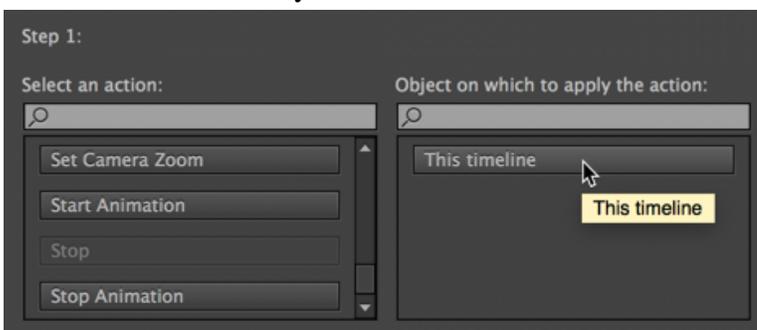
- Add a new layer named **Actions** and lock it. It is best practice to have a layer such as this that is only for holding code – so locking it makes sure we do not mistakenly add any visual assets.



- At the 16s mark, add a blank keyframe. Keep this frame selected.
- Open the Actions panel from the application menu by choosing **Window > Actions**.
- Within the Actions panel, do not write any code. Instead, click the button in the upper right labeled **Add using wizard**.

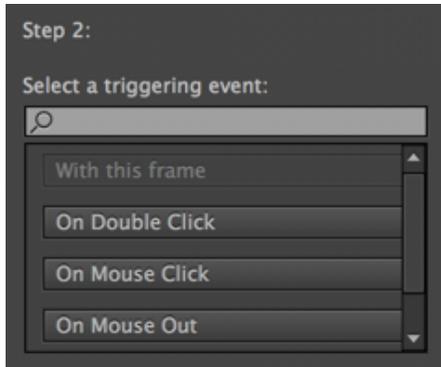


- This will summon the **Actions Wizard**! Scroll down within the available actions and choose **Stop**. For the object to apply the action to, the only choice is **This timeline** – so click that as well.



Once completed, click **Next**.

- For **Step 2**, we must define a triggering event. Choose **With this frame** to execute the code when the playhead reaches the current frame.



Click **Finish and add** to complete our interaction.

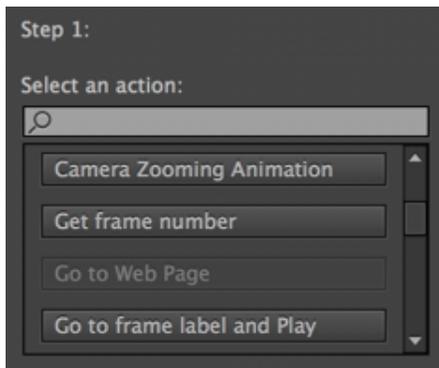
- The **Actions** panel now reflects the code established by the **Actions Wizard** on the selected frame. Notice it also writes some comments describing how the code functions:

```
/*  
Stop a Movie Clip/Video  
Stops the specified movie clip or video.  
*/  
this.stop();
```

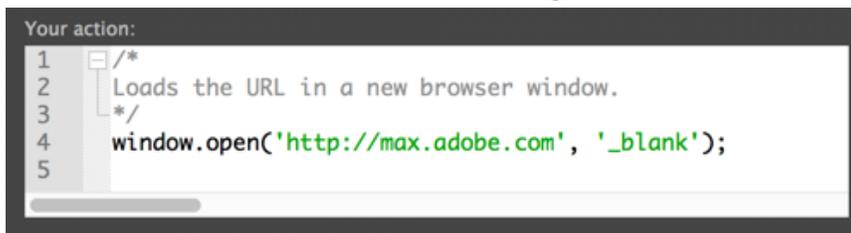
Half way there... We'll now move on to write some code which enables the user to interact with this project by clicking on the **Button** component instance we've created.

- Select frame 1 in the **Actions** layer and once again click **Add using wizard** in the **Actions** panel.

- Choose the button that reads **Go to Web Page**.

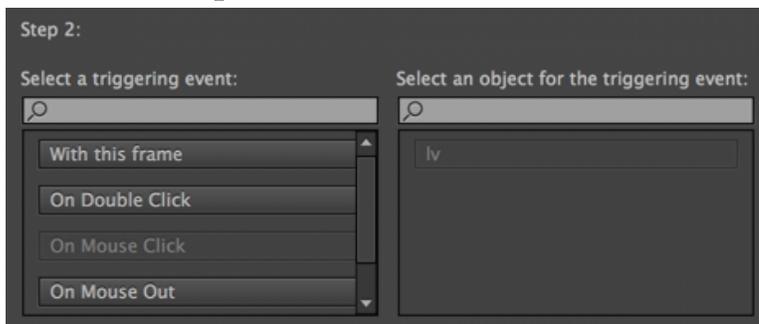


- Look above to see the code which has been injected. We are going to tweak this a bit before moving on.



Change “*http://www.adobe.com*” to “*http://max.adobe.com*” so the user is directed to the **Adobe MAX** website. Click **Next** to move on.

- On **Step 2**, we will first select **On Mouse Click** as a triggering event and then choose the **lv** instance as our triggering object. Remember that this is the instance name we have given to our **Button** component instance.



Now hit **Finish and add** to commit the code.

- You'll now be in the normal view of the Actions panel and will see the following JavaScript code:

```
/*  
Clicking on the specified symbol instance  
executes a function.  
*/  
/*  
Loads the URL in a new browser window.  
*/  
$('#dom_overlay_container').on('click', '#lv',  
function() {  
/*  
Loads the URL in a new browser window.  
*/  
window.open('http://max.adobe.com', '_blank');  
}.bind(this));
```

Perform a **Control > Test** to interact with your HTML5 Canvas project. The animation will stop at the end, as instructed.



You'll also be able to click the button at any time to visit the **Adobe MAX** website.

Understanding Platform Differences

Sometimes, if your interactive elements do nothing or your animation is not running correctly, it is because of JavaScript errors. To see whether your HTML5 Canvas project is throwing errors, you can refer to the browser console.

Let's look at how to access the console from **Google Chrome**. Many other browsers have similar features.

- Run your project by choosing **Control > Test** from the application menu.
- In the **Chrome** web browser, right-click anywhere within the viewport and choose **Inspect**.

In the tooling area that appears, click on the **Console** tab. Any errors and warning and such will be displayed within.

Additional Resources

To keep up on all things Animate, here are a few resources:

- **Adobe Animate Team Blog**
<https://blogs.adobe.com/creativecloud/tag/adobe-animate-cc/>
- **Adobe AIR and Adobe Flash Player Team Blog**
<http://blogs.adobe.com/flashplayer/>
- **Adobe Animate CC Learn and Support**
<https://helpx.adobe.com/animate.html>

To contact Joseph Labrecque, note the following links:

- **Website**
<http://josephlabrecque.com/>
- **Twitter Profile**
<https://twitter.com/JosephLabrecque>
- **Blog - In Flagrate Delicto!**
<http://inflagratedelicto.memoryspiral.com/>
- **Lynda.com Author Page**
<http://www.lynda.com/JosephLabrecque>
- **Pluralsight Author Page**
<https://www.pluralsight.com/authors/joseph-labrecque>
- **Amazon.com Author Profile**
<http://amazon.com/author/josephlabrecque>



THANK YOU!

HAVE FUN WITH ADOBE ANIMATE!