

Motion Graphics Techniques - Lesson 7 Notes

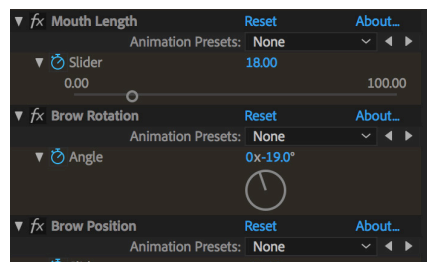
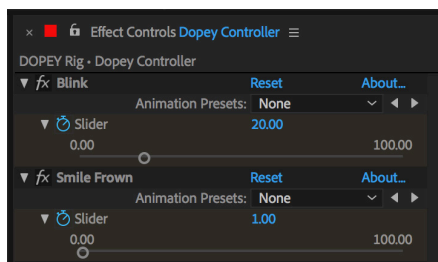
Rigging A Faux 3D Character Face

Character Animation in After Effects

For the last few years character animation has become a dominant force in the motion design world. Although some animators use hand-drawn, or traditional techniques, most studios tend to use After Effects to animate 'puppet' characters made up of Photoshop or Illustrator artwork. Why? Because it's a lot faster to do, and can more easily fit into a typical client budget. Additionally, the results can be charming in the same way as cell animation.

Rigging Characters

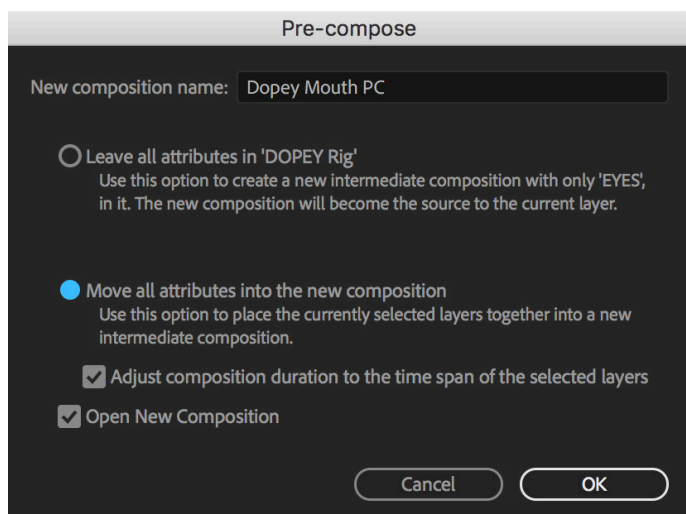
Before you can animate a puppet character in AE, you first need to rig it. Otherwise you would have to animate all of the layers independently, which would be more difficult and time-consuming. An example of very simple rigging would be the robot arms from the Rube Goldberg Mini Golf project, which you connected using parenting and placement of anchor points. More complex rigging uses a lot of expressions, because they allow you to connect and adjust specific properties individually. Expression Controls and Null Objects are also very commonly used.



Rigging allows you to create and control your own properties for a character, and keep them all in one place. I think of this as making a custom tool to animate with.

Making Precomps

In this class we have spent lots of time animating in Precomps. The ones we've used were made in Photoshop as Groups and imported into AE. But there is also a specific way to create precomps directly from a layer (or any number of layers) within an AE composition.



Simply select the layers you want to precomp, and then go to Layer > Precompose (or Shift + Command + C). This opens a pop-up window which asks if you want to leave the attributes (like masks, effects, etc) within the original comp, or move everything into the new composition.

I prefer to have the check box selected that automatically opens the precomp in a new tab.

Motion Graphics Techniques - Lesson 7 Notes

Rigging A Faux 3D Character Face

Variables

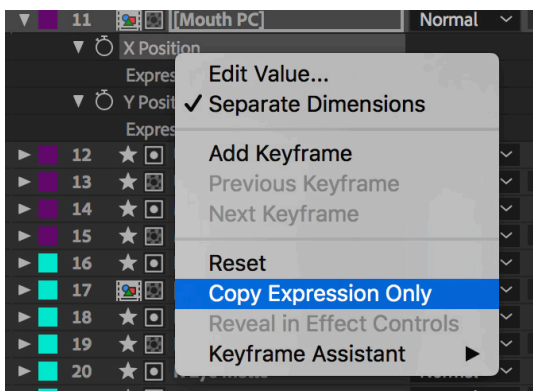
Variables are user-defined letters or words that stand in for numbers in code. They are called variables because they can change (“vary”) what they stand for. For example, in this lesson I use x and y sometimes as variables, but I also use xPosSlider, xPosSlider, negX, etc. Sometimes I use my cats names as variables, if I’m feeling silly. So that it is really clear when I’m defining a variable, I always write “var” before the words. This is a good practice to get into.

```
var xPosSlider = thisComp.layer("Dopey Controller").effect("X Position")("Slider");
var yPosSlider = thisComp.layer("Dopey Controller").effect("Y Position")("Slider");

var x = transform.scale[0] + ( xPosSlider * .1 );
var negX = transform.scale[0] - ( xPosSlider * .1 );
```

Copy Expression Only

Rather than copying code from the little expression box and pasting it into another layer’s expression box, AE gives you a simpler way to copy and paste expressions. Simply select the property (not the layer) with the expression on it, and then right click (or Control click) and navigate to Copy Expression Only. Now you can select a layer, or many layers, and then use the normal Command + V to paste the expression. Easy!



Expressions and Dimensions

Some properties in AE have only one value associated with them, like Opacity or Rotation. The numbers associated with properties are called Dimensions. For example, Scale and Position have two dimensions -- X and a Y axis values. With Position you can “Separate Dimensions” and split those into two different properties. Scale does not allow that. It needs to be given two numbers in order to work. In After Effects, this always needs to be written like this: [Number 1, Number 2]. But as I discussed above, variables can be used in place of numbers, so more often my Scale expressions end like this: [x, y] or [xScale, yScale].

Motion Graphics Techniques - Lesson 7 Notes

Rigging A Faux 3D Character Face

“If Else” Expression

Some expressions need to be ‘conditional’, meaning that sometimes they yield one result and sometimes they yield a different result. The simplest of these is the “if / else” expression. Here is a basic template for a simple conditional:

```
if (true or false statement) DO  
THIS  
else DO THIS OTHER THING;
```

Nose “stretch and squash” conditional expression for Dopey

In this lesson, I built up a somewhat complex expression for stretching and squashing Dopey’s nose and muzzle. I wanted to write it out here so that you can look at it more closely and see the various parts:

```
var xPosSlider = thisComp.layer(“Dopey Controller”).effect(“X Position”)(“Slider”);  
var yPosSlider = thisComp.layer(“Dopey Controller”).effect(“Y Position”)(“Slider”);
```

I pick-whipped to the X and Y Position Sliders

```
var x = transform.scale[0] + ( xPosSlider * .1 );  
var negX = transform.scale[0] - ( xPosSlider * .1 );
```

NOTE: variable x uses a plus and variable negX uses a minus, but otherwise they have the same code

```
if (xPosSlider > 0)  
x = negX  
else x;
```

Here is the first conditional statement. It tells AE that if the slider value is positive, change it to a negative value, otherwise keep the negative value.

```
var y = transform.scale[1] + ( yPosSlider * .1 );  
var negY = transform.scale[1] - ( yPosSlider * .1 );
```

```
if (yPosSlider > 0) y = negY  
else y;
```

I wrote this conditional statement with different spacing, to show that AE is easy-going about most syntax.

```
[x,y]
```

The final values, as defined by the variables x and y

Semicolons, Parentheses and Brackets

After each complete ‘code sentence’, you need to write a semicolon so AE knows you’re defining a new thought. Also you can’t use parentheses and brackets interchangeably -- they are very specific. Parentheses are used for calculations, like (5 * 2) or (x + y), and Brackets are used for Arrays like [100,100] or [x,y]. You’ll get used to it!