Forget Keyframing-Take Real-Time Control of your Animated Assets with 3dsMax

Robert Terry
Stantec

Alonso Rodriguez
Stantec

Learning Objectives

• Explore various animation methods, alternatives, their drawbacks and benefits.

• Learn how to properly set-up, organize and link your models for optimized animation techniques.

• Learn the benefits of expression controllers and how to generate your animations without using keyframes.

• Learn how to wire your animation parameters and create a custom UI that allows you to control multiple animation parameters with a single click like using a “virtual remote control”.

Description

Reacting to client demands with speed and accuracy is why they keep coming back so don’t let last minute changes to your detailed animations force you to work overtime. Eliminate the time spent modifying your animations and start controlling and editing their parameters in real-time. From the dynamic components of a complex thrill ride for theme park pre-viz, a simple ceiling fan for your next VR walkthrough or the interlocking gears of the clocktower in that dramatic scene in your upcoming game, you’re dedicated to making that animated sequence perfect. And while few things up the “wow-factor” like a properly animated scene, few things kill your budget as quickly as time spent editing and re-timing complex keyframed animations. Learn valuable techniques and some tips and tricks in 3ds Max to create a “virtual remote control” that modifies animated parameters with the click of a button. So forget about keyframing-- take control of your animations and never sweat that deadline again.
Speaker(s)

Rob Terry @3dRobT

Rob is a 3D digital artist with 20 years of professional experience as both a production assistant and art director of many wide-ranging architecture projects in the corporate, hospitality, themed entertainment, and simulation industries. A graduate of the University of Massachusetts at Dartmouth and a 3ds Max Certified Professional, Rob uses the latest technologies to help develop customized workflows that facilitate the design process while providing high-quality renderings, animations, and real-time, interactive previsualization as a Design Visualization Specialist for international design firm, Stantec. His clients include Walt Disney Imagineering, SeaWorld Parks & Entertainment, Florida Hospital, Merlin Entertainment, and United States Army Corps of Engineers.

Alonso Rodriguez @AlonsoRodriguez32814

Alonso is a licensed architect, LEED Accredited Professional, and a Certified Revit Professional. He has worked with Hospitality, Mixed-Use, Themed Entertainment, and Department of Defense clients including Disney, SeaWorld, Universal Studios, Hilton Hotels, Florida Hospital and US Army Corps of Engineers. After graduating from Virginia Tech in 2009, Alonso joined Real Designs BIM Consulting in Northern Virginia. That opportunity led Alonso to VOA Associates who had a great reputation in the military sector. Today, he works primarily on hospitality and themed entertainment projects both large and small for domestic and international clients. VOA has recently joined the Stantec family and now works with the extended BIM team.
It’s Simple: Just click the “animate” button.

It’s there…it’s right next to the “make art” button. But not really. Having the proper tools and understanding along with some patience and determination go a long way in ensuring your project gets done right.

The Right Tool/Technique for The Job.
3ds Max offers various tools and techniques to create simple or very complex animations. Every project is different and each situation requires its own evaluation to determine which technique to implement. Creating keyframes along the timeline is simple but even simple animations can get complex quickly, so it’s important to understand the concepts at your disposal...

Animation controllers or procedural object modifiers are based on algorithms that compute the motion for you, making things easier to control, allowing you to spend more time creating instead of editing. An animation controller “is a plug-in that handles the storage and interpolation of all animated values.” You can even build up layers of animation controllers just like adding modifiers to your stack and blend between them.

A noise controller on our object will randomly affect its position along the timeline; a tedious task using keyframes.

List of Animation Controllers

We can access these controllers several different ways. From the main UI, you can assign controllers in the motion panel or from the Animation dropdown in the main menu. In the track view, you can assign controllers from the Edit menu or from the quad menu by right-clicking on the track you want to animate.

For more information about different types of animation controllers, click here.
The 5 “P”s: Proper Planning Prevents Poor Performance

Proper setup and organization is essential. Whenever possible, it’s good practice to link your geometry to a dummy object and add your controllers to it instead of directly to the geometry. To do this, we need to insure our object or objects are linked properly, using appropriately placed pivot points, are hierarchically in order and using the appropriate coordinate system. In this example of a simple ceiling fan, we only need to animate the main housing. If the fan blades and brackets are not part of the housing geometry, they need to be linked to the housing which in turn is linked to the dummy object that has a rotation controller assigned to it. The animation controller in this case is a simple Bezier curve. We need to modify the keyframes to ensure a linear interpolation of the points and then add an additional controller that will ensure the blades spins consistently and indefinitely.
In 3ds Max almost **EVERYTHING** is animatable.

This allows us to extract our objects’ inherent data and use it to our advantage...In many instances we can eliminate the need for keyframes entirely.

Despite the 2 keyframes in our ceiling fan example being easy to assess and to change values, employing this technique to a more complex object, such as in the amusement park ride below, keyframe editing becomes more time consuming and it may make sense to let some additional tools do the heavy lifting for us. Enter the Expression Controller.

*EXAMPLE OF A MODEL WITH MANY SIMPLE ANIMATED ELEMENTS. ALTHOUGH, INDIVIDUALLY THESE MOTIONS ARE SIMPLE, COLLECTIVELY THEY BECOME MORE DIFFICULT TO MANAGE AND EDIT.*

**Expression Controllers**

Using Expression controllers to drive our animations not only saves time and make things easier, but it paves the way to utilize this information to drive other animated elements using wiring parameters or controllers that modify these values in a more efficient and effective way.

The Expression Controller uses mathematical formulas to control animation parameters. Click [here](#) for more information.
Commonly Used Expressions

This topic lists some expressions that you might find useful in situations when you animate.

Circular Path

\[
[ \text{Radius} \times \cos(360 \times \text{Time}), \\
\text{Radius} \times \sin(360 \times \text{Time}), 0 ]
\]

where \( \text{Time} \) is one of the predefined time variables such as NT or S.

If you make the two Radius values unequal, you get an elliptical path.

If you specify a nonzero Z component, the path is no longer planar.

Following Another Object

\([X, Y, Z] + \text{Position}\)

where \( \text{Position} \) is the Position controller of the second object.

The vector \([X, Y, Z]\) can be an offset from the second object. (If it’s \([0,0,0]\), the two objects occupy the same position.) It can also be a vector expression that specifies some movement in itself.

Keeping an Object Between Two Objects

\((\text{Position1} + \text{Position2}) / 2\)

where \( \text{Position1} \) and \( \text{Position2} \) are the Position controllers of two objects.

The divisor 2 constrains the object to be halfway between the two other objects. Other values constrain the object to other locations.

Bouncing Between Other Objects

\((1 + \sin(360 \times \text{Time})) / 2 \times (\text{Pos1} - \text{Pos2}) + \text{Pos2}\)

where \( \text{Time} \) is one of the predefined time variables such as NT or S; \( \text{Pos1} \) and \( \text{Pos2} \) are the Position controllers of two other objects.

The subexpression \((1 + \sin(360 \times \text{Time})) / 2\) is a value that oscillates between 0 and 1 over time. \((\text{Pos1} - \text{Pos2})\) is the vector between the two other objects. Multiplying the two and then adding \( \text{Pos2} \) as an offset locates the object along this vector.
Both the simple ceiling fan and our more complex amusement park ride benefit from this technique. By using an expression, the animated results are the same as when keyframed, but we're better prepared to rig our animation for even more control.

We use the expression:

\[ S \times \text{Speed} \times -360\pi/180 \]

This will rotate our object along the Z axis at a speed determined by the variable we created. In this case, we created a variable called Speed and applied a constant value. We can modify these values easily while being totally independent of animation length. Click here for more information on the expression controller dialog.

In the following example, we use an expression controller to create a circular path of a variable radius that spins at a variable rate with a specific value representing how much it is off-plane (simulating weight) \([\text{Radius} \times \cos(S \times 360 \times \text{Speed}), \text{Radius} \times \sin(S \times 360 \times \text{Speed}), -20]\)

In this case, we added complexity to the animation by introducing elements that spin with our main object and appear to be affected by the centrifugal force of that spin. We apply an expression controller to the goal of a simple IK Chain which determines the radius of the path on which the goal will travel. Our geometry is linked to that chain and duplicated around the main object. Again, we create variable parameters that drive those values and we can introduce variation to simulate weight.

For more Expression Techniques, click here.
Sit Back & Relax, you have a remote for that.

Here is where all the preparation really gives you an added benefit. All of our animations are driven by mathematical equations based on values that make sense to our situation or are governed more or less by the laws of said mathematical equation. There are no keyframes but when the client needs you slow things down, or modify position for better compositions, you still must access the expression dialog and enter in the changes. Better and safer than manually editing keyframes? You bet, but unlike your parents who had to get up to change the channel on their TVs way back when, we can use 3ds max's wiring tools to create a virtual "remote control" to easily edit those animated parameters on the fly and in real-time.

We can access each entities’ expression values by wiring those values to an additional controller by way of an object modifier called ‘Attribute Holder’ which we place in the stack on whichever object you determine to be the ‘remote control’. The Attribute Holder modifier is blank or empty until you specify which attributes you’d like to control. In the case of our swing ride, we’ve determined that we need to control:

- Height
- Rotation Speed
- Centrifugal Force
- Tilt

![Attribute Holder Before and After Populated with Your Custom Animated Elements](image-url)
To add these custom attributes, you need to access the ‘Parameter Editor’ from the Animation drop-down on the main menu. From there you add your parameters and set up both how the attribute looks and works in the ‘Attribute Holder’ modifier. We replace the constant value with a range that drives the value returned by the expression based on either a slider or up/down arrows. We can also give a default value. The alignment and orientation settings dictate how this information will be displayed in the modifier.

Once we wire the expression controller to the appropriate parameter, we’ll be able to control the value of the expressions with one click. We do this by opening the expression controller to access its values.

The values entered in the Parameter Editor (left) will represent how those attributes look and function from within your ‘remote control’ (right).
Before, using the ‘Assign to Constant’ option returned a constant value to our Speed variable. But we want to be able to control this value with sliders or up/down arrows not by accessing dialogs, keyframes or anything that may slow us down, so instead we use Assign to Controller.

This allows us to navigate and select the appropriate attribute. This wires the mathematical formula \( S*Speed - 360*pi/180 \) to the value of the input setup on the remote control. In our case, up/down arrows that vary the speed by a number from -1 to 1 which is a multiplier of the value returned by the expression.
Now when your client asks you to change the parameters of your animation even though it’s already perfect; just kick up your feet because you can edit those animated assets in no time at all.

Now you have: