FTV197415-L

The Mechanics of Motion Automation

Steven Schain
Post Production Supervisor | M & E Content Producer
4D Technologies | CADLearning

Learning Objectives

- Wire parameters for rotational motion animation in 3ds Max.
- Simulate pivot constraints using 3ds Max tools suitable for animation.
- Rig a mechanical assembly using bones and inverse kinematics.
- Create a keyframe animation to drive the assembly.

Description

Mechanical motion is all around. From a simple door hinge to a car cylinder piston, it is easy to forget that even the simplest real-world motions can be complex operations in 3D animation software. This class will explore several methods for automating complex motion animation in 3ds Max 2020. Designers will learn how to wire parameters, then import the assembly into 3ds Max, as well as how to create similar constraints using 3ds Max tools. You will learn strategies for building a hierarchy based on the animation requirements, basic parameter wiring, and setting up links and Inverse Kinematics to mimic specific constraints. The class will step through multiple short lessons that allow you to create the animation of specific moveable parts. Finally, attendees will see what advantages each tool has for different steps in the animation of complex motion.

About the Speaker

Steven Schain is the post-production supervisor for all CADLearning products from 4D Technologies, as well as the content development manager of CADLearning's Media & Entertainment and Design products for Autodesk, Inc. software, including 3ds Max, Maya, Inventor and Fusion 360. In 1998, Autodesk recognized Steven as one of only 16 Autodesk Training specialists worldwide. He has since contributed to Autodesk's certified courseware for 9 releases of 3ds Max, was a co-developer of Autodesk's ACI Program and 3ds Max's fundamental standards and is currently an Autodesk Certified Instructor. As a premier Autodesk trainer, he has continued teaching end users, companies, and many others, including The Walt Disney Company, Guess, and the United States Army. As a 8-year veteran of Autodesk University and a featured speaker for AU 2018, Steven has taught classes ranging from creating particle fountains in 3ds Max, to classes on 3D printing and entrepreneurship.

sschain@cadlearning.com

Contents

Introduction

Whether it's for serious production development, simple visual design, or just for fun, as an animator or artist, you can use Autodesk's 3ds Max software to create animatable mechanical models for anything from simple doors or gears, all the way to highly complex mechanical assemblies. There are many instances in which it can be helpful to animate a design, to see how the parts will function together. While simulation can be performed in a program like Autodesk Inventor or Fusion 360, it is often important to have a higher level of control over the animation of an assembly.

Animation is a tool for showing the operation of an assembly in a way that still pictures and diagrams cannot convey. 3ds Max is the ideal environment for creating animated versions of your designs. You can either import your design into 3ds Max or create it completely from scratch. Once created, you can add helper objects, create hierarchical structures, and animate your design using multiple types of animation.

Controlling animated elements and animation timing is what makes 3ds Max a better solution than simply performing a simulation in Autodesk Inventor or Fusion 360. While a simulation can portray the motion of a complex assembly, presenting a compelling animation involves being able to coordinate the animation of multiple objects within the same scene. Once the animation is complete, you need to decide what the output will be for the animated sequence.

In many cases, an animation is rendered to a sequence that can be played back at full speed. The level of quality of the final rendering will usually be based on certain requirements, such as the purpose of the animation or the delivery method. If you're running a test to see if the animation looks the way you want it to, you can save a preview rendering of the viewport. When it comes time to create a final rendering, Arnold or ART can create a high-quality, realistically rendered animation.

This class will discuss and outline the steps you would take to animate several different assembly types. Each assembly type represents a different need because how its parts are controlled is unique.

Animatable Assemblies

Autodesk Inventor and Fusion 360 are powerful design tools that allow you to create a wide array of mechanical assemblies. You can also create assemblies within 3ds Max that fill your needs. Assemblies do not have to be complex either; they can be as simple as a hinged arm, or as complex as a bucket excavator assembly.

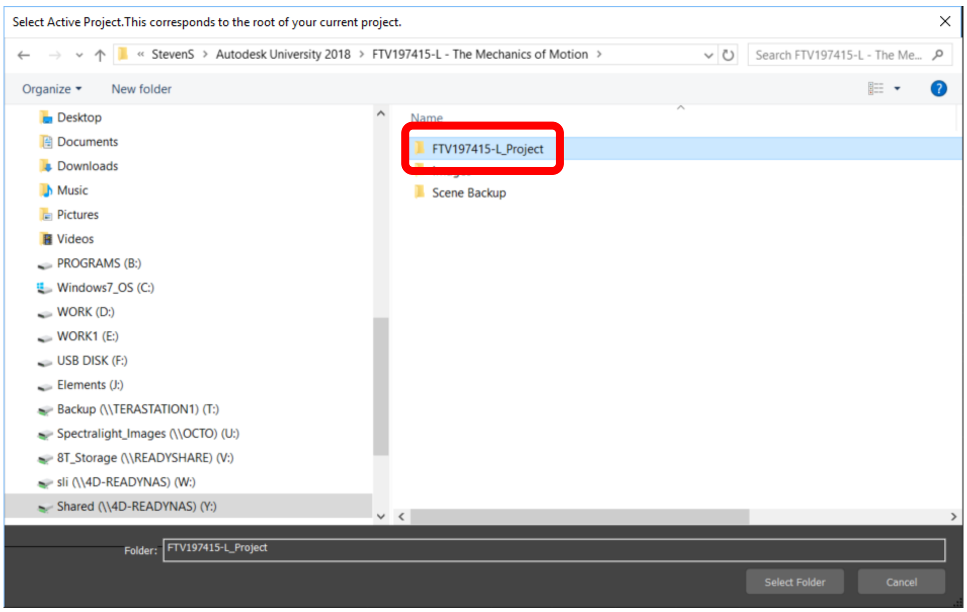
Each assembly type can be animated using a number of different techniques, which employ a variety of tools available in 3ds Max. In this class, you will set up three different types of assemblies, using three different techniques that all tie in to the same excavator assembly.

The first exercise is how to wire parameters together to link the animation of one object to another using a simple equation. The second is how to create a hinged piston animation using a helper object and a constraint to maintain the piston's alignment. The third uses bones and Inverse Kinematics to create an operating set of joints on an excavator assembly. Lastly, you will use wired on-screen manipulators to add animation to a rigged excavator.

Setting the Project Directory

Before you do anything, you are going to configure a project path. For this lab, I have provided a preconfigured project directory that you are going to use. If you don't work with project directories, I recommend that you start using them. Project directories are a very good way to organize individual projects.

1. In the **File** menu, choose **Set Project Folder…**
2. In the **Browse for Folder** dialog, navigate to the "**FTV197415-L_Project**" directory, then click **OK** to set it as the current project directory.



Now, when you load and save files, you can save them to the **scenes** directory within that project folder.

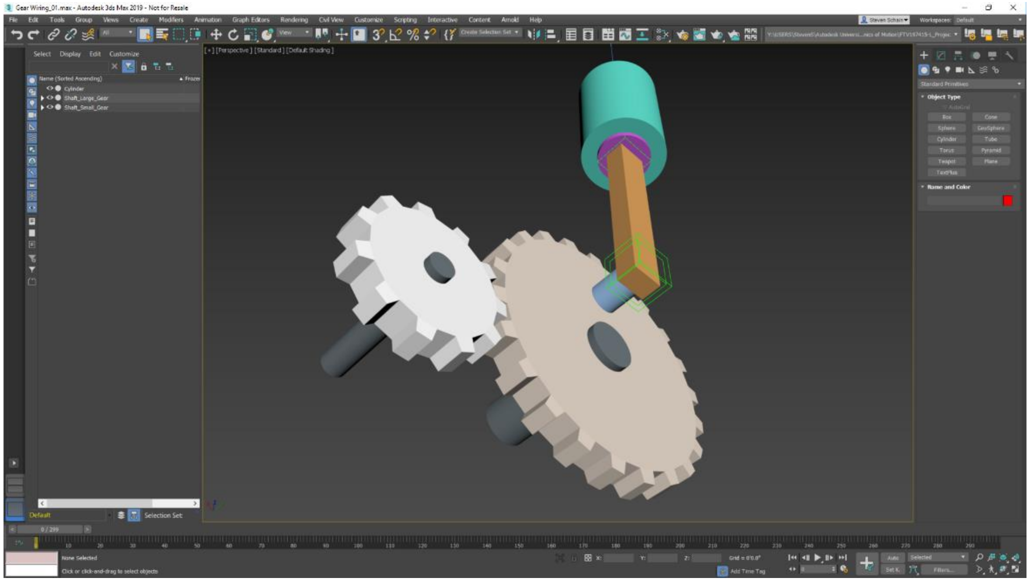
Wiring Parameters for Connected Animation

Creating connected animation in 3ds Max can be accomplished in several different ways. Depending on your needs, you can use controllers, expressions, or you can wire parameters together. *Wiring* is a tool that allows you to define a relationship between two parameters. The parameters can be on one object. For example, the size of a sphere can be controlled by its X position. Or, wiring can be between two different objects so that the behavior of one object affects the behavior of the other.

In this exercise, you will wire the Y rotation of two objects to create a relationship between the objects. The objects in this case are a set of gears mounted on individual shafts. The gears are configured with a 2-to-1 ratio, with the smaller gear and larger gear meshing together.

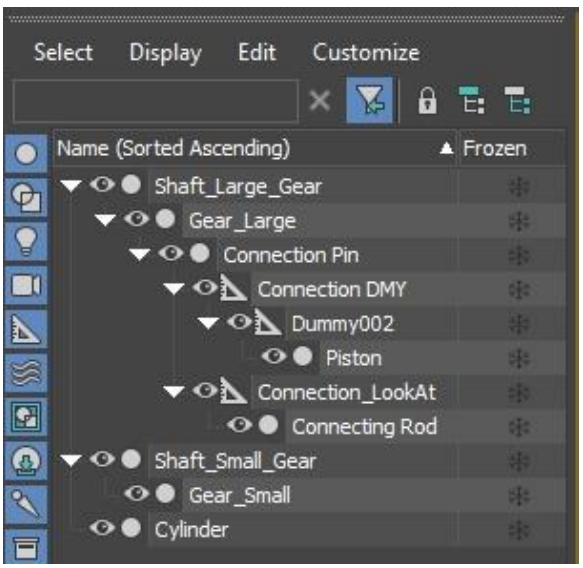
Wiring the Gear Shafts

Wiring parameters for the gear shafts can be done one of several ways. The easiest method is to use the right-click quad menu and select the parameter to be wired right in the viewport. Open the file **Gear_Wiring_01.max** for this exercise.

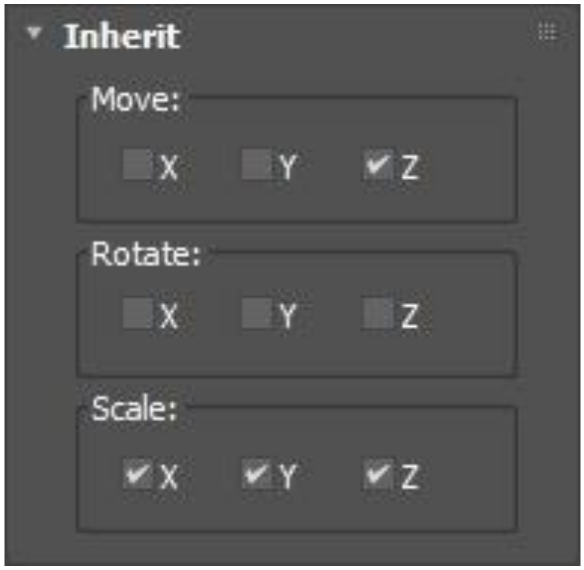


The scene is a set of simple gears, with the larger gear connected to a piston rod and piston that functions when the larger gear rotates. The smaller gear is the main driver of the animation, but in this case, the wiring will be set up so that rotating either gear will work.

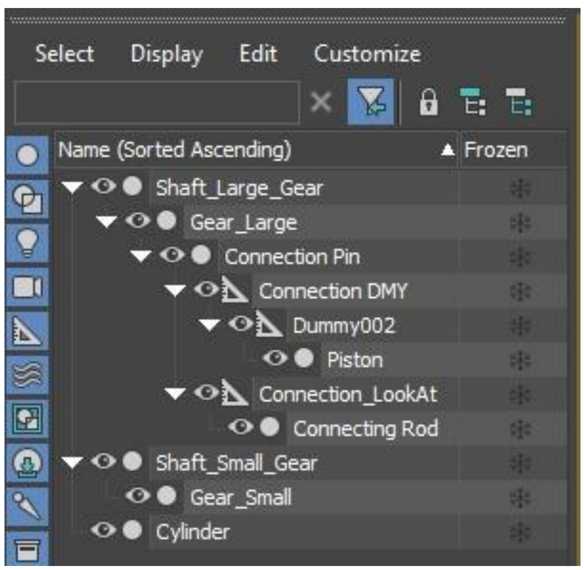
The way the scene is configured, the gear shafts are the parents of the gears. This way, if you rotate the shaft, the gear will rotate with it. An additional hierarchy is set up for the piston and connecting rod, to make it function automatically when the large gear rotates.



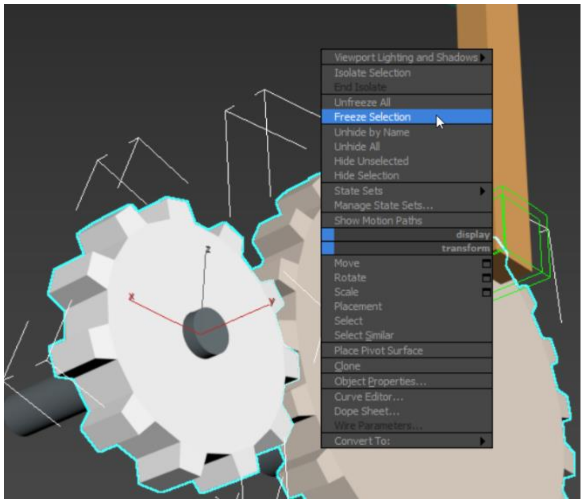
The piston and connecting rod are made up of several dummy objects and a LookAt constraint. By turning off the X and Y Move, and X, Y, and Z Rotate Inheritance, the motion of the **Connection DMY** object will be limited to just the Z-axis. This allows it to move up and down only when the gear rotates.



The LookAt constraint on the **Connection_LookAt** dummy object ensures that the attached **Connecting Rod** object will remain in proper orientation to the Piston as the gear rotates.

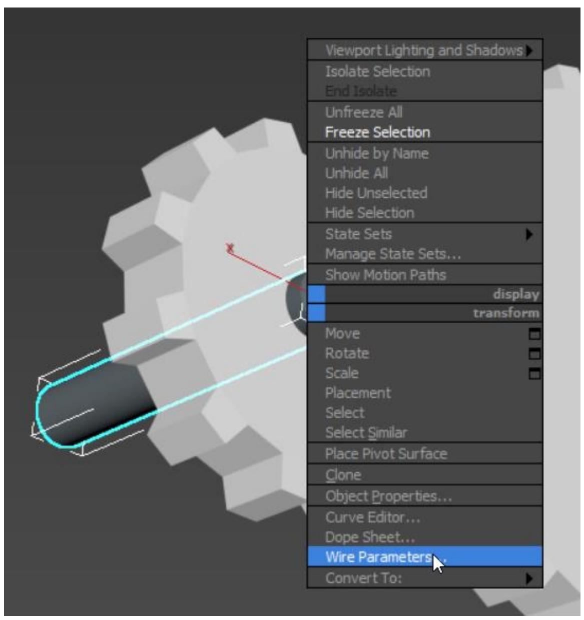


1. To make it easier to wire the gear shafts, freeze the gears. Select the small gear.
2. Hold the **Shift** key and select the large gear.
3. Right-click and choose **Freeze Selected** from the Quad menu.



With the gears frozen, it's easier to select the gear shafts to work with them.

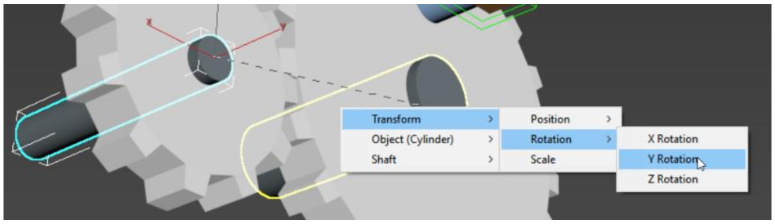
4. Select the **Shaft_Small_Gear** object.
5. Right-click and select **Wire Parameters** from the Quad menu.



6. From the parameters options popup, choose **Transform** > **Rotation** > **Y Rotation**. A rubber band line will appear.

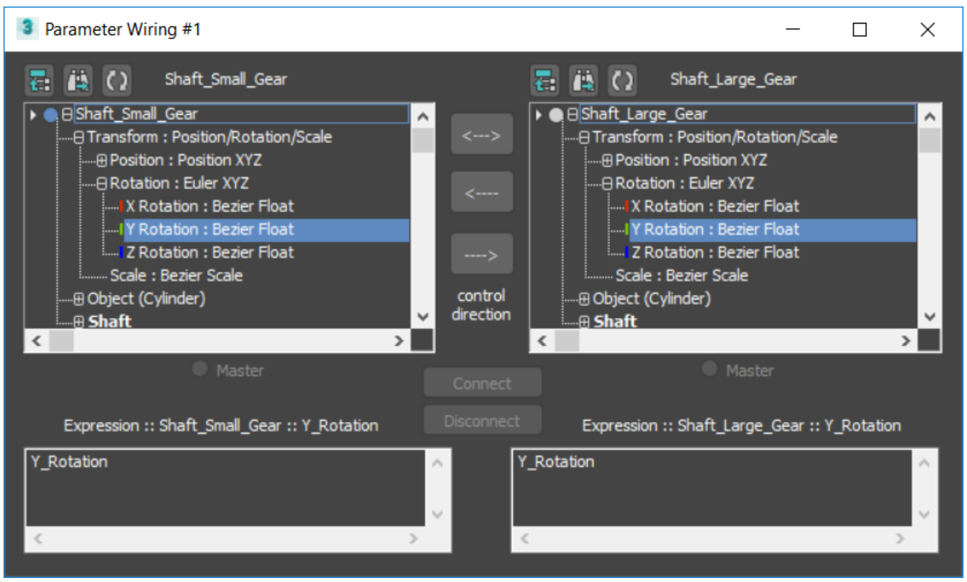
7. With the rubber band line active, select the **Shaft_Large_Gear** object.

8. From the parameter options popup, choose **Transform** > **Rotation** > **Y Rotation**.



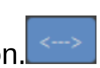
The Parameter Wiring Dialog

Once you connect the parameters, you are presented with the Parameter Wiring dialog box. This dialog box allows you to connect a parameter from one object to a parameter on a second object. By using the right-click menu, you make it easier to access those parameters for each object.



The parameter wiring dialog is split with the first object selected on the left, and the second object selected on the right. The middle contains a set of three buttons between the parameter trees, which allows you to control the direction of the parameter wiring. The top button is the two-way connection option, allowing both objects to control each other. Below that are the two single-direction controls, with the first object being controlled by the second, or the second object being controlled by the first.

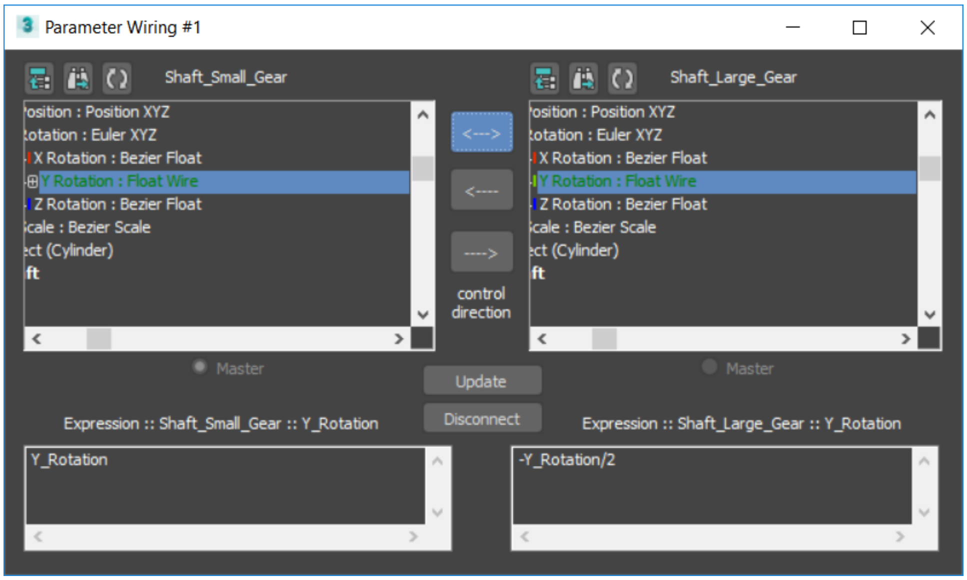
At the bottom of each parameter tree are two expression text boxes. Currently, the expressions are the same **Y_Rotation** for both the left and right sides. However, this animation requires the small gear to rotate in a 2-to-1 ratio, as compared to the large gear, and it needs to rotate in the opposite direction.

9. In the parameter wiring dialog, click the **Two-way connection** option. 

10. Leave this **Shaft_Small_Gear** object as the master for this animation. However, since this is a two-way connection, animating either shaft will rotate both.

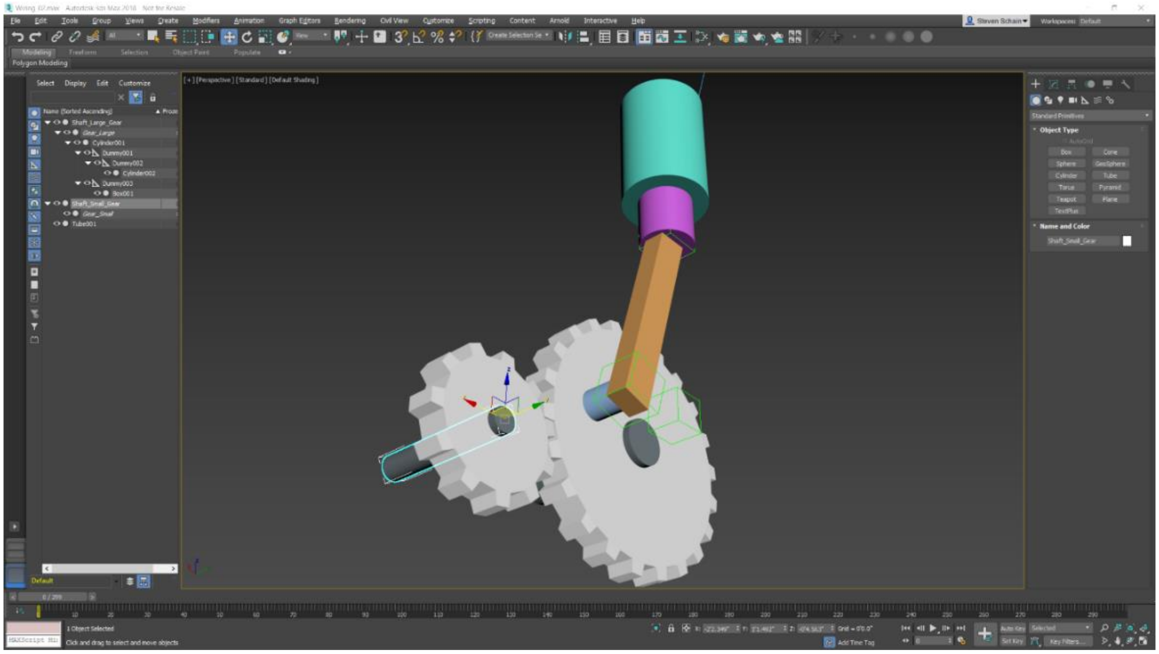
IMPORTANT: *In order to cause the larger gear shaft to rotate at half the speed of the smaller gear and in the opposite direction, you need to edit the expression for the large gear shaft. It's important to understand the relationship between expressions. By setting the small gear shaft as the master, you must edit the expression for the large gear shaft.*

11. In the expression for the **Shaft_Large_Gear** object, enter "**-Y_Rotation/2**."
12. Click **Connect** to connect the two parameters, wiring them together.



13. Close the parameter wiring dialog.
14. From the main toolbar, click **Select and Rotate**.
15. Select the **Shaft_Small_Gear** object.
16. Rotate the object around the Y-axis.

Notice that the large gear rotates once for every two rotations of the small gear. And, the rotation of the large gear causes the piston to function.



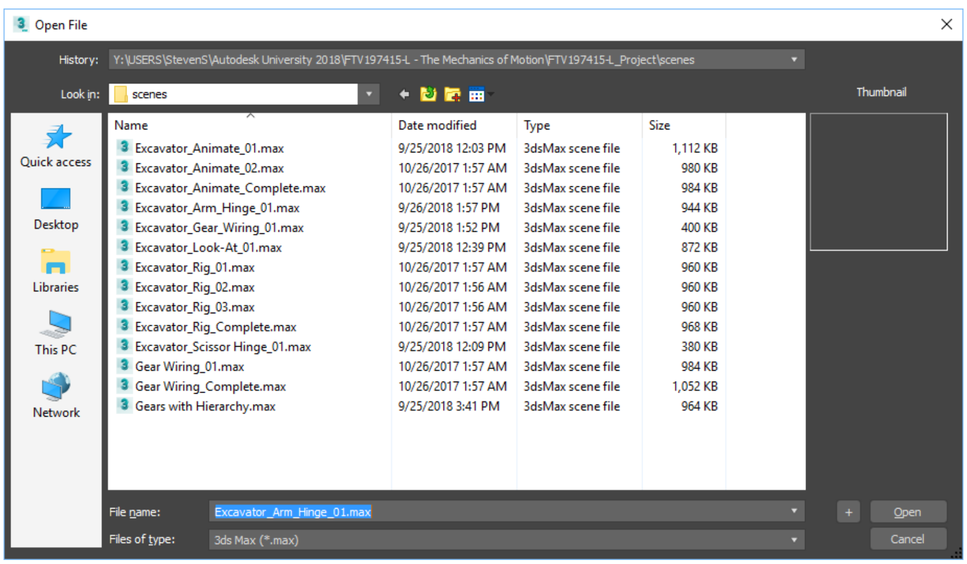
Configuring Pistons on an Industrial Hinge

You are going to start with a scene that is already set up with a backhoe model. The idea is to create a basic rig that allows the animation of the pistons connected to the arms. When animating a part, like a piston set, it makes it easier if it can be made to be animated using some type of helper and a constraint. In this case, dummy objects are the objects that will provide the motion for the attached excavator piston objects.

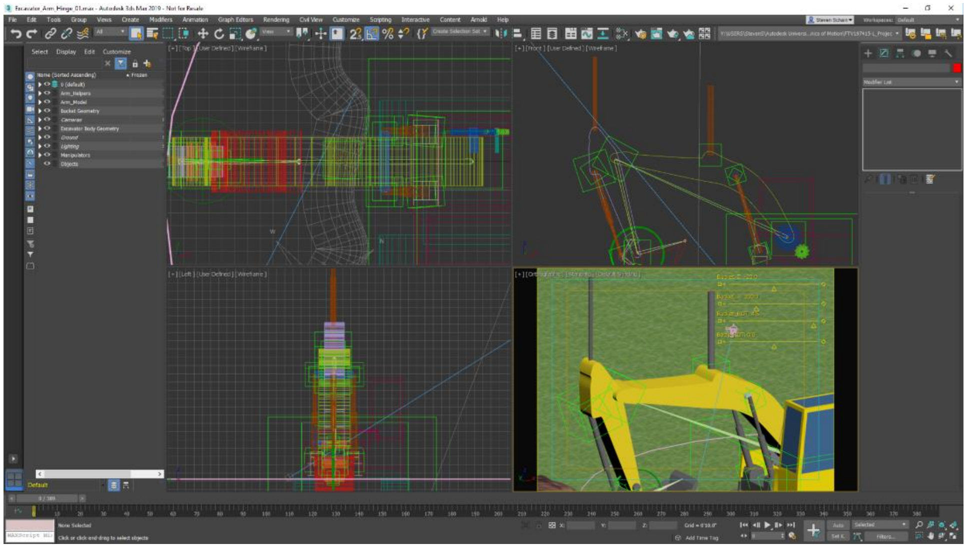
While having an organized file will certainly help when working on any file, that will not always be the case. If you are importing a model from Revit or Inventor, you may not have the time to reorganize the layers and objects to fit a particular configuration. However, if you can, it can make working in the files easier.

Often, it is helpful to have certain elements preset from a default file. For example, having certain layers for elements like cameras, shapes, and environmental elements preset can help in scene organization.

1. From the **File** menu, click **Open**.



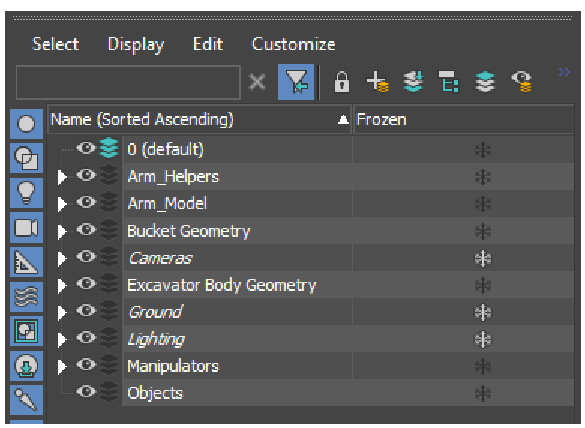
2. Open the file, **Excavator_Arm_Hinge_01.max**.



You can have several pre-built environments that you use for rendering your designs. There may be times when you need to make changes, but a basic environment gives you a good starting point. The other reason to start with a preconfigured scene like this, is because you can have some layers preset that help isolate the lighting and cameras, as well as background and staged objects.

Layers

Looking at the **Layers** view in the **Scene Explorer** for the file, you'll see that there is a Ground layer, a Cameras layer, and a Lighting layer. Also, notice that those three layers are frozen, which means that you cannot pick them in the viewport.



The advantage to working with frozen layers is that you don't have to worry about accidentally selecting the lights or moving cameras. Notice that the default layer, 0, is the current layer, so anything you build will be created on that layer.

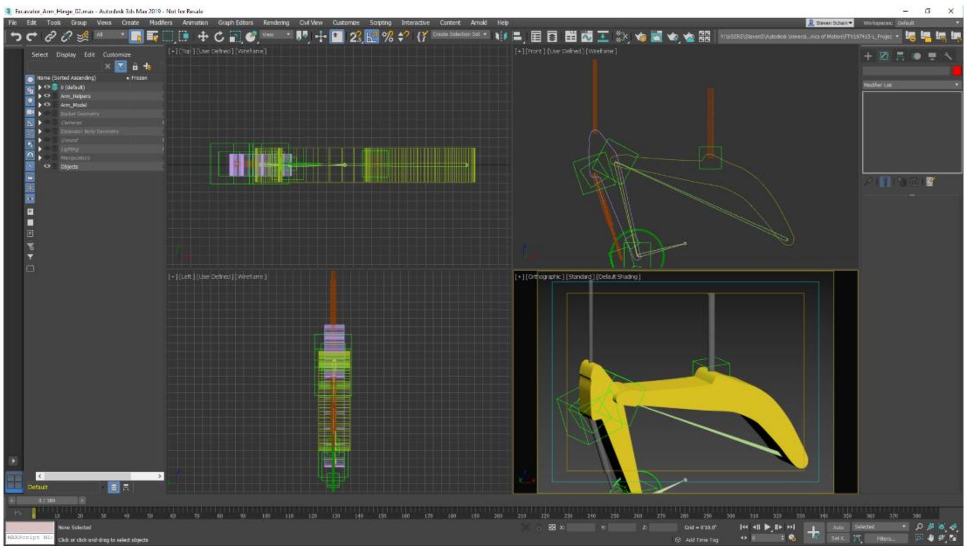
Preparing the Scene

3ds Max allows you to work in a way that you are comfortable, by giving you the flexibility of showing and hiding objects based on either layers or object hierarchy. First, you are going to hide all the elements of the scene that will not be needed for this portion of the exercise. Once the door assembly has been configured, you'll unhide the objects and restore the scene for rendering.

This scene is set up so that you can hide certain layers in order to easily work with the door.

Hiding Scene Layers

3. Make sure the **Scene Explorer** is set to **Sort by Layer**. 
4. In the layers list, turn **off** the visibility for the Bucket Geometry, Cameras, Excavator Body Geometry, Ground, Lighting, Manipulators, and Objects.



ONLY THE ARM AND PISTONS REMAIN IN THE SCENE ONCE ALL THE OTHER OBJECTS HAVE BEEN HIDDEN.

Hierarchies for Animation

Hierarchies are used when determining how certain objects, or assemblies animate. A *hierarchy* is created by linking one object, the child, to another, the parent. The parent can then control the child and allow for easy creation of more complex animations. When organizing objects for animation, you first want to analyze the way the objects will be animated, and whether it will require any constraints or inverse kinematics.

With the pistons, you will be adding two Look-At constraints, one for each half of the piston assembly. However, with 3ds Max, you cannot just have two objects look at each other. For this reason, there are several extra helper objects in the scene for each part of the piston to look at. These need to be linked properly to their respective parent parts as well.

If you look at the two piston parts that are connected to the first and second arm sections, there are only a few parts: the piston cylinders and three dummy objects. However, all the parts must be connected in such a way that makes them work together. While there are a number of ways to connect these objects, you need to keep in mind how they will interact with each other. Remember—you are adding a Look-At constraint to each of the cylinders, and they cannot look at the opposing cylinder.

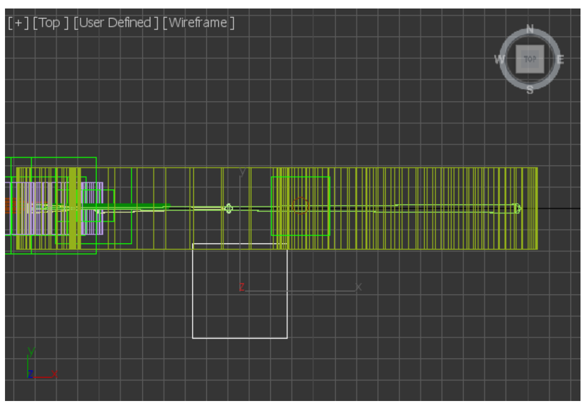
To start here, use: **Excavator_Arm_Hinge_02.max**

Adding Look-At Helpers

In order to be able to properly configure the pistons to maintain their alignment, there are two additional helper objects that are required for the scene. One is for the upper part of the piston, and the second is for the lower part. These will need to be aligned to the pivot point of each part, then linked to the associated part of the arm.

5. Click the **Create Tab** in the **Command Panel**.
6. In the **Create Tab**, choose **Helpers**.
7. Set the **Scene Explorer** to **Sort by Layer**. 
8. Set the **Arm_Helpers** layer as the current layer by clicking the layer icon to the left of the layer label.

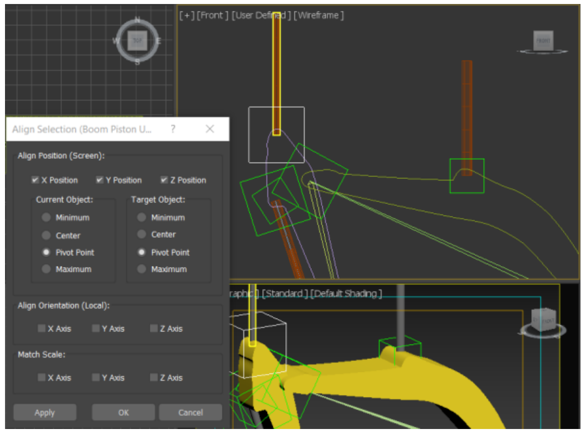
   This will create the new dummy object on the Arm_Helpers layer.
9. Click the **Dummy** helper object.
10. To create the dummy object, click and drag in the **Top** viewport. Make the dummy object a little wider than the arm geometry.



11. Once the dummy is created, click the **Modify** panel.
12. In the **Name** type-in, change the name to "**Dmy_Boom Piston Link**" and press **ENTER**.

Positioning the first helper

13. With the **Dmy_Boom Piston Link** selected, click the **Align** tool. 
14. In the **Front** viewport, click on the **Boom Piston Upper** object. The Align Selection dialog opens.

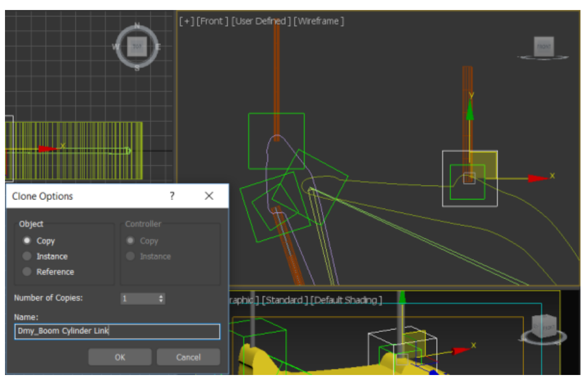


1. In the **Align Selection** dialog, make sure the **Align Position X**, **Y**, and **Z** axis options are all checked.
2. Set both the **Current Object** and **Target Object** alignment options to **Pivot Point**.
3. Click **OK**.

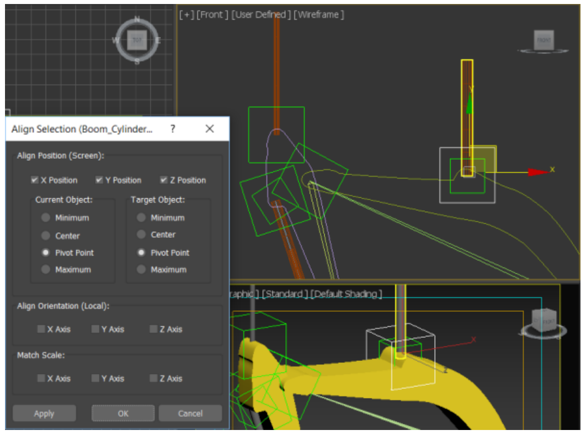
Copying the Second Helper

Instead of creating a second dummy helper, it's simpler to make a copy of the first one, then align it to the correct location.

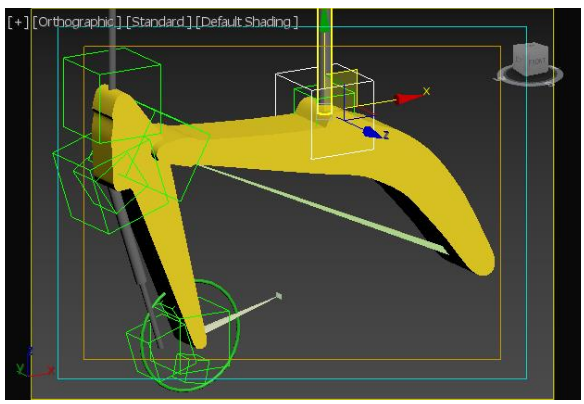
4. From the main toolbar click **Select and Move**.
5. In the front view, hold the **SHIFT** key, then click and drag the **Dmy_Boom Piston Link** object over to the pivot point of the **Boom_Cylinder Upper** object and release the mouse button.



6. In the **Clone Options** dialog, set the **Object** option to **Copy**.
7. Change the **Name** to "**Dmy_Boom Cylinder Link**", the click **OK**.
8. With the **Dmy_Boom Cylinder Link** selected, click the **Align** tool and click on the **Boom_Cylinder Upper** object.



9. In the **Align Selection** dialog, make sure the **Align Position X**, **Y**, and **Z** axis options are all checked.
10. Set both the **Current Object** and **Target Object** alignment options to **Pivot Point.**
11. Click **OK**.



To start here, use: **Excavator_Arm_Hinge_03.max**

Setting up the Hierarchy

The next step is to set up simple hierarchies using the helper objects and the geometry in order to be able to properly set up the Look-At constraints. In the current setup, there are three dummy objects associated with the upper piston objects. The idea behind this is to have the piston and cylinder linked to their respective helpers, with the helper objects getting the Look-At constraints, instead of the cylinders themselves. The three helpers allow for additional flexibility in how the motion is configured.

In this section, you will create a parent-child relationship between the objects that make up the piston.

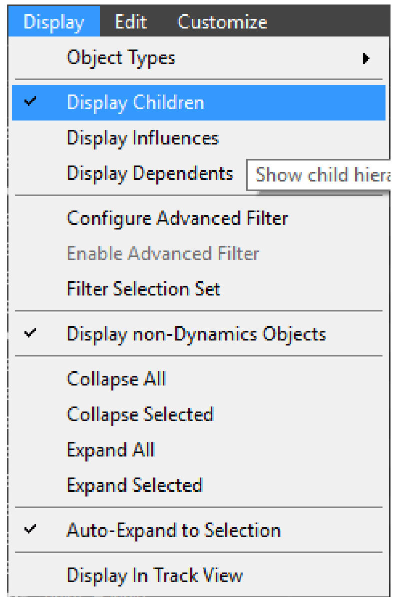
Linking the Piston Parts and Helpers

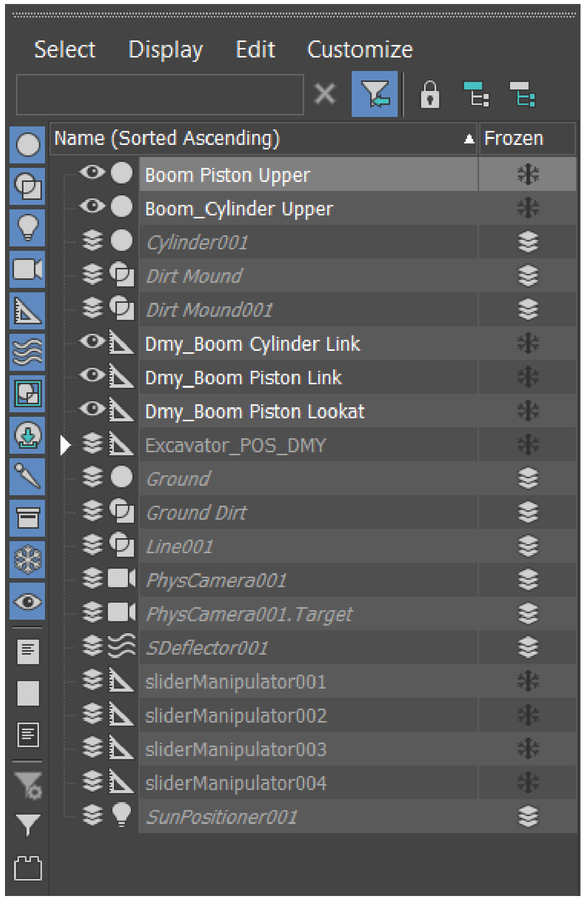
To link the objects, you will use the Select and Link tool. This allows you to simply drag and link objects in the scene to create the relationships.

15. In the **Scene Explorer**, select **Sort by Hierarchy**.

*If needed, expand the **Name** column so that you can see the complete name by clicking on the vertical separation line to the right of the name category heading and dragging to the right.*

16. Select the **Display** menu and make sure the **Display Children** option is checked.





The Scene Explorer gives you a listing of all the objects in the current scene, which allows you to visually identify the types of objects and how they fit within the hierarchies within the scene.

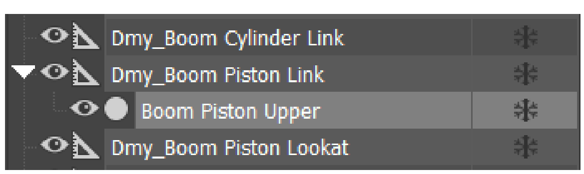
17. To begin creating the hierarchy for the piston, select the **Boom Piston Upper**.

18. Click the **Select and Link** tool in the main toolbar. 

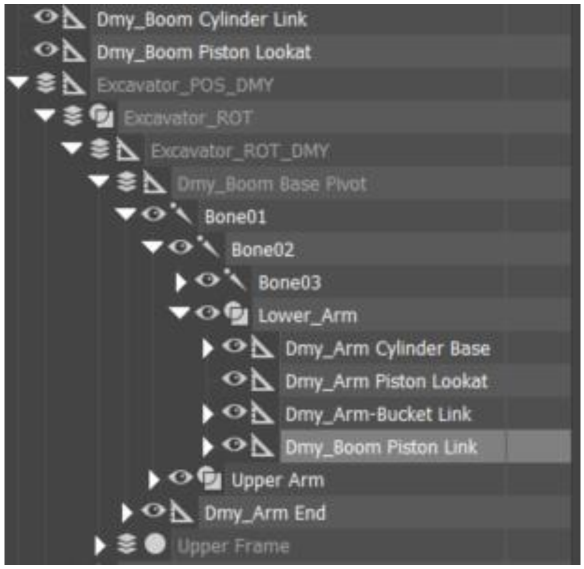
    (Optionally, you can use a drag-and-drop operation instead.)

19. Starting on the **Boom Piston Upper** object, drag and drop it onto the **Dmy_Boom Piston Link** object and release the mouse button.

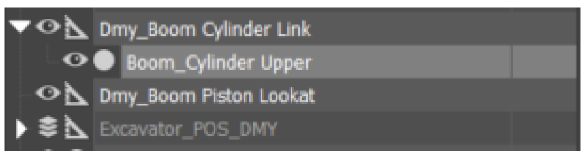
    If this is done correctly, the Dmy_Boom Piston Link will flash once, and you will see the change in the hierarchy in the Scene Explorer:



20. Now select the **Dmy_Boom Piston Link** object, and carefully drag and drop it onto the **Lower Arm** object and release the mouse button.

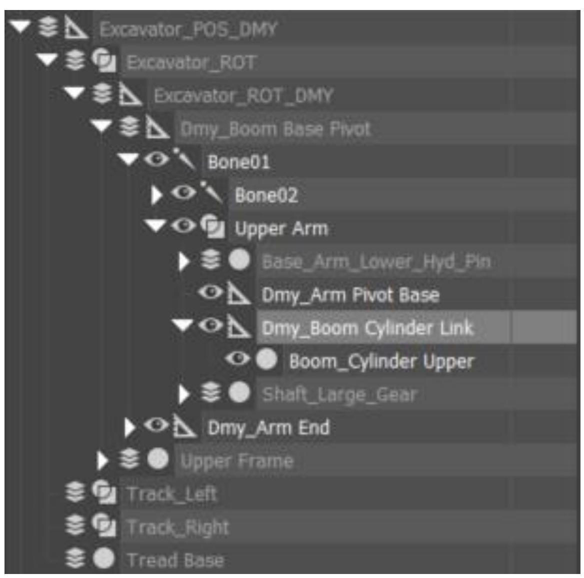


21. Select the **Boom_Cylinder Upper** object, and carefully drag and drop it onto the **Dmy_Boom Cylinder Link** object.

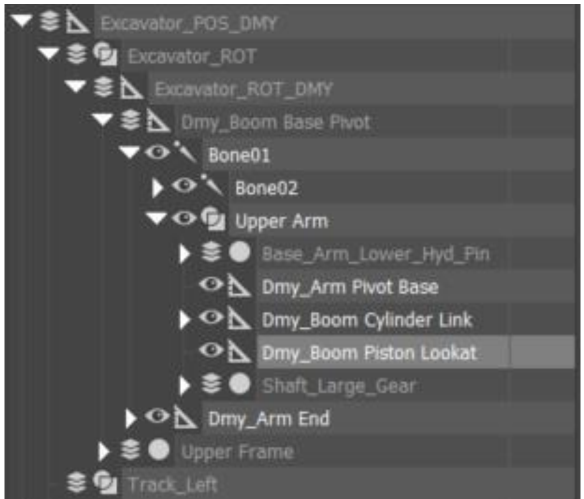


If you need to zoom the viewport, scroll the mouse wheel up to get closer to the objects you are working on.

22. Select the **Dmy_Boom Cylinder Link** object, and drag and drop it onto the **Upper Arm** object.



23. Select the **Dmy_Boom Piston Lookat** object and drag and drop it onto the **Upper Arm** object.



24. To stop the Select and Link tool, click the **Select Object** tool. 

This completes the hierarchy for the helper objects in the scene. Now, the pistons can be configured with Look-At constraints, to automate the animation of the pistons based on the motion of the upper and lower arms.

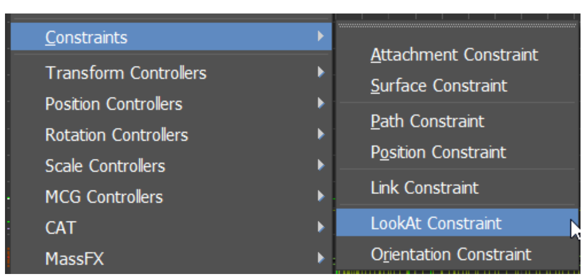
To start here, use: **Excavator_Arm_Hinge_04.max**

Adding the Look-At Controllers

Once the hierarchy is established, you can begin to add the Look-At constraints. The Look-At constraint is a specialized controller that can be easily added to an object to keep it oriented towards another object in the scene. By having the two cylinders properly aligned, using a Look-At constraint on them will create the look of a piston that works automatically when the parent objects are animated.

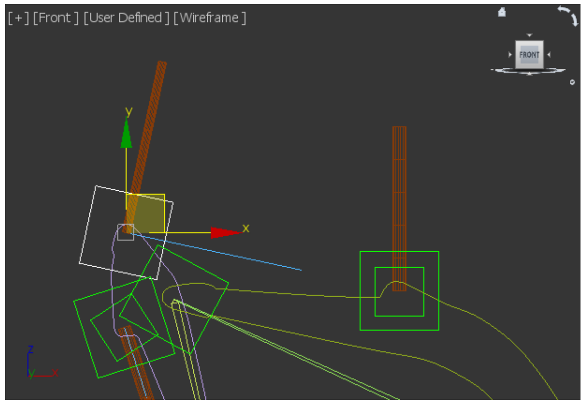
To accomplish this, you will use the dummy objects and add the constraints to them, not the cylinders directly. Since the cylinders are children of the dummy objects that will have the constraints assigned, they will reorient with the helper objects.

25. In the **Front** viewport, select the **Dmy_Boom Piston Link** object.
26. From the **Animation** > **Constraints** flyout menu, choose **Look-At Constraint**.



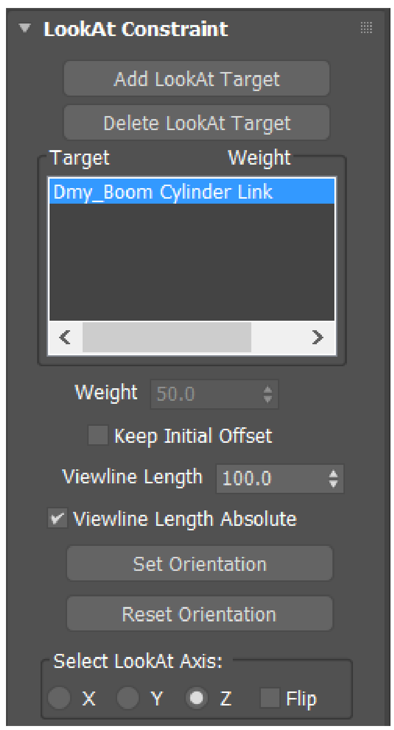
A rubber band line appears from the origin of the dummy object once the constraint is chosen.

27. Click the **Dmy_Boom Piston Lookat** to add the **LookAt Controller** to the helper.

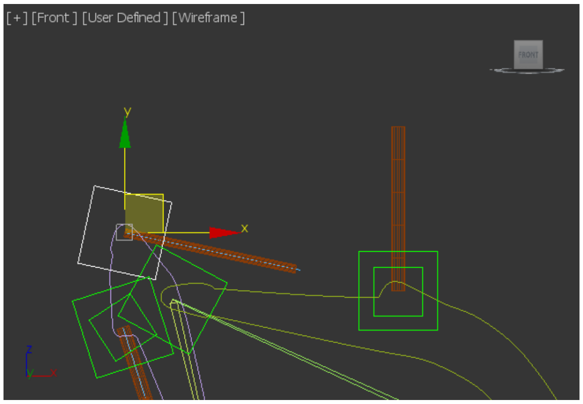


Once the constraint is assigned, the dummy reorients itself, and the Motion panel becomes active. Now, the orientation needs to be modified so the piston is looking at the target object.

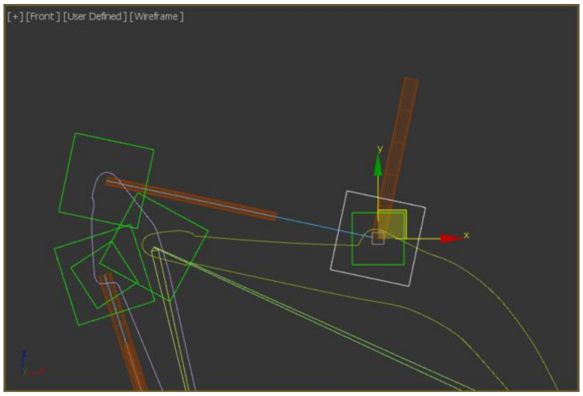
28. In the **Motion** panel, scroll to the **LookAt Constraint** rollout; set the **LookAt Axis** to **Z**.



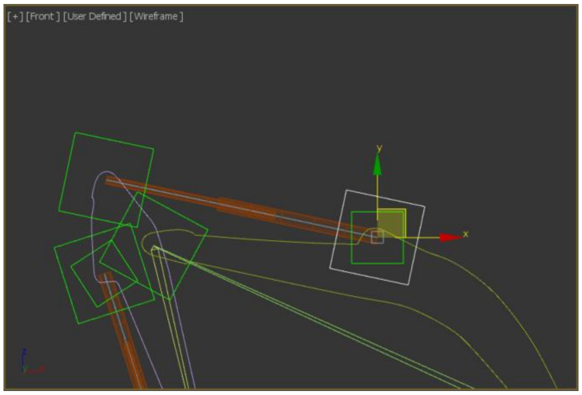
When the Z-axis is selected, the piston will be oriented properly.



29. Select the **Dmy_Boom Cylinder Link** object.
30. From the **Animation** > **Constraints** flyout menu, choose **Look-At Constraint**.
31. Click the **Dmy_Boom Piston Link** object, to add the **LookAt** controller.



32. In the **Motion** panel, scroll to the **LookAt Constraint** rollout; set the **LookAt Axis** to **Z**.



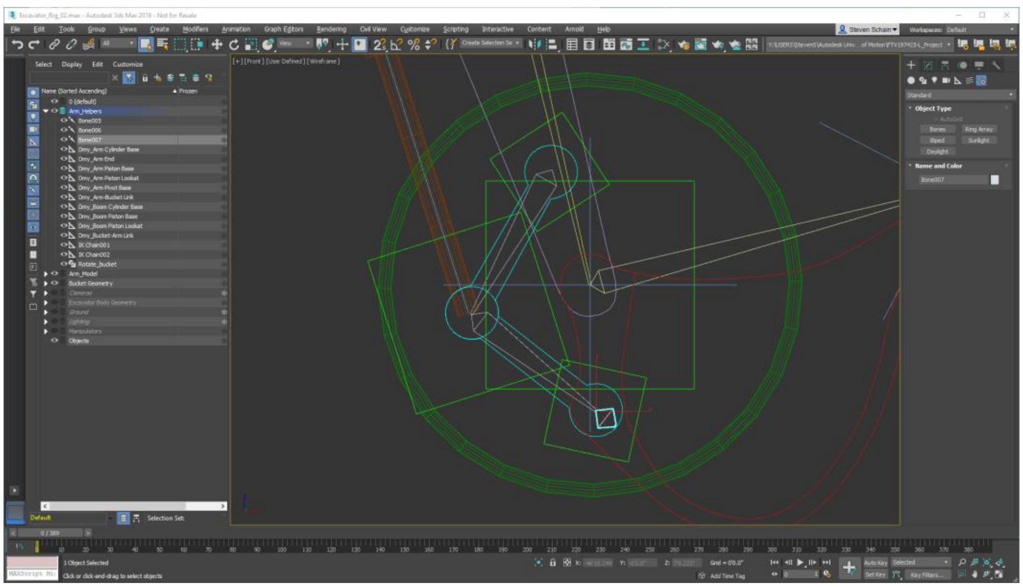
From this point, you can animate the excavator arms without concern that the piston will not function properly. By using LookAt constraints, you can automate the motion of an object in reference to another object, making assembly automation simpler.

Rigging an Excavator Bucket Scissor Joint

Sometimes, when creating more complex rigs, a combination of methods needs to be used. Using helper objects can accomplish a wide range of tasks for setting up animation rigs. So can using constraints, such as LookAt and Link constraints for adding secondary motion. However, one of the most-used rigging components is *inverse kinematics*, or IK for short.

Inverse kinematics works very well with 3ds Max's bone objects. The combination can be used to create complex armatures with a range of flexibility. In this case, an IK setup will be used to activate the scissor joint for an excavator arm bucket rig.

The scene for this lesson is preconfigured with a rig that allows for direct animation of the upper part of the excavator stand-in. This rig represents the low-polygon stand-in version that the high-resolution rig would be attached to for the final animation.
Open the file **Excavator_Rig_01.max** for this exercise.



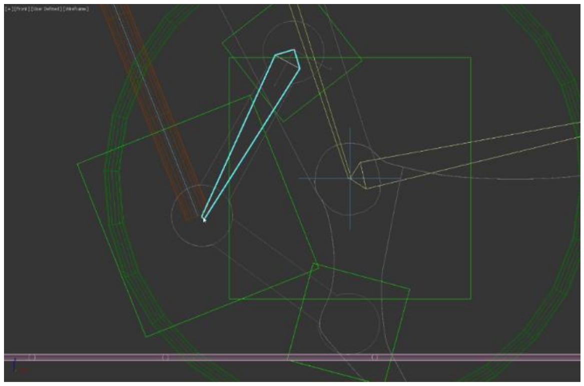
Adding the Bones

To get the scissor joint to work, bones and an IK chain must be added to the excavator rig.

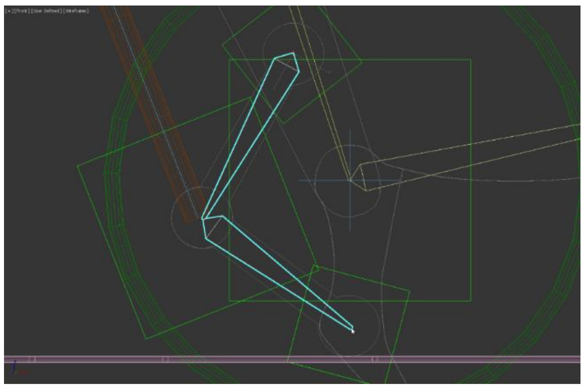
1. From the **Command** panel, click the **Create** tab, then click the **Systems** option. 
2. In the object type rollout, click **Bones**.

In the **Front** viewport, notice how the bucket link objects are connected. You want to make sure you create the bones so that they are lined up with the joints that correspond to the ends of the bucket link objects.

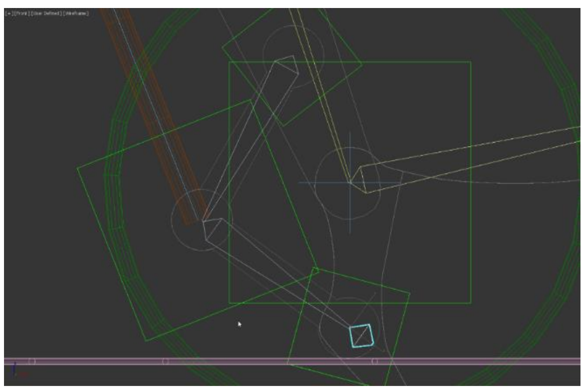
3. Click towards the top of the screen in the Front viewport at the center of **Bucket Link A1** top joint, to create the start of the bone chain.
4. The end of the first bone should be at the pivot point where the bucket links meet. Click there to create the first bone and start the second.



5. To place the end of the second bone, click at the pivot point for **Bucket Link A2** towards the bottom of the screen.



6. Right-click to create the second bone and add the bone end. Then press ESC to exit the bone creation tool.



You now have a complete bone chain that can be used for this scissor joint.

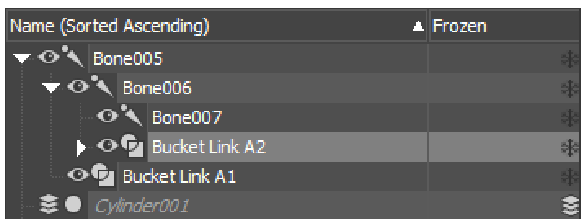
To start here, use: **Excavator_Rig_02.max**

Linking the Assembly

Once the bones have been created, the next step is to link the assembly objects to their appropriate bone. This can be done easily in the viewport using the Select and Link tool.

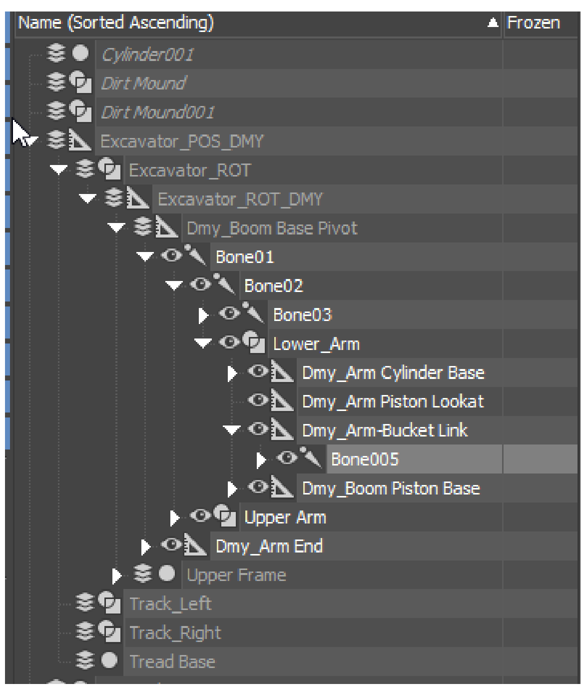
7. From the **Main** toolbar, click **Select and Link**. 
8. In the **Front** viewport, select **Bucket Link A1**, hold the left mouse button, and drag over the first bone in the chain. You'll see a rubber band line appear with a link icon.
9. Release the mouse button, and the bone will flash once, indicating it is linked as the parent of the Bucket Link A1 object.

10. Now select Bucket Link A2, hold the left mouse button, and drag it over the second bone in the chain.
11. Release the mouse button, and the bone will flash once, indicating it is linked as the parent.



Lastly, the first bone in the chain needs to be linked as a child of the **Dmy_Arm-Bucket Link** helper object.

12. Select the first bone in the chain. Holding the left mouse button, drag the cursor over the **Dmy_Arm-Bucket Link** helper object.



13. Click **Select Object** in the **Main** toolbar to get out of Select and Link mode. 

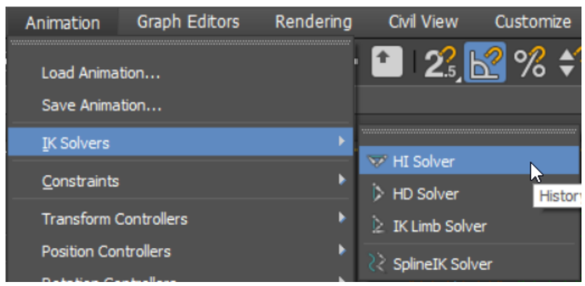
    This configures the bone chain and the hierarchy required to make the scissor joint work.

To start here, use: **Excavator_Rig_03.max**

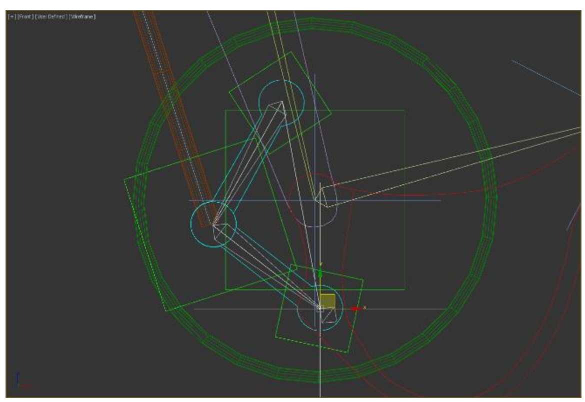
Creating the IK Chain

Bones in 3ds Max are very flexible and can be used in a variety of ways. One of the ways it can be used is to create an automated flexible component for a rig. This can be accomplished easily by adding IK to the bone chain.

14. In the **Front** viewport, select the first bone in the chain.
15. From the **Main** menu, choose **Animation** > **IK Solvers** > **HI Solver**. A rubber band line will appear from the origin of the first bone.

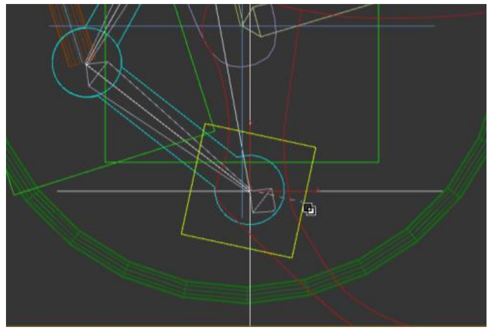


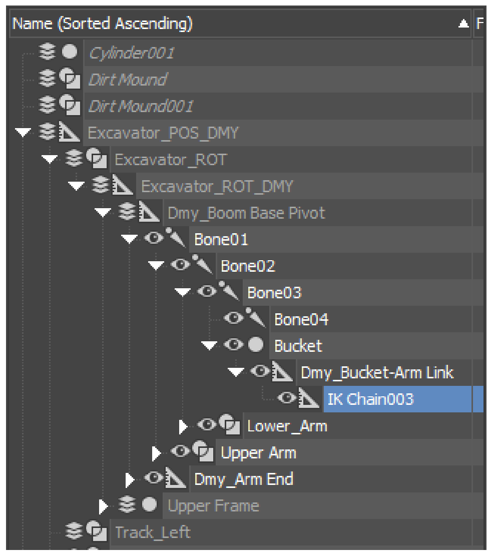
16. Click the last bone in the chain, the very small end bone, to create the IK chain.



Next, you need to link the IK chain end effector, the crosshair at the end of the bone chain, to the **Dmy_Bucket-Arm Link** helper object.

17. Make sure the IK chain end effector is selected.
18. Click **Select and Link** from the **Main** toolbar. 
19. Click and drag from the end effector to the **Dmy_Bucket-Arm Link** helper object. The helper object will flash once to let you know that it has been linked as the parent of the end effector.



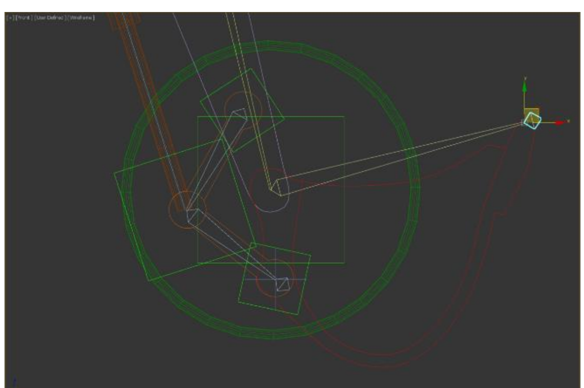


20. Click **Select Object** in the **Main** toolbar to get out of Select and Link mode. 

Rotate the Bucket

Once everything is linked, and the bones are set up with an IK chain, it's time to test out the assembly.

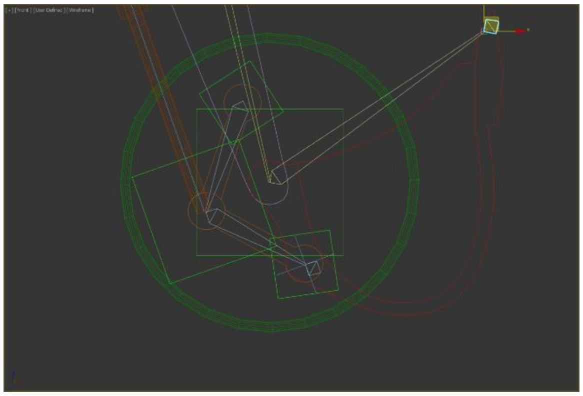
21. In the viewport, select last bone in the bucket arm bone chain.

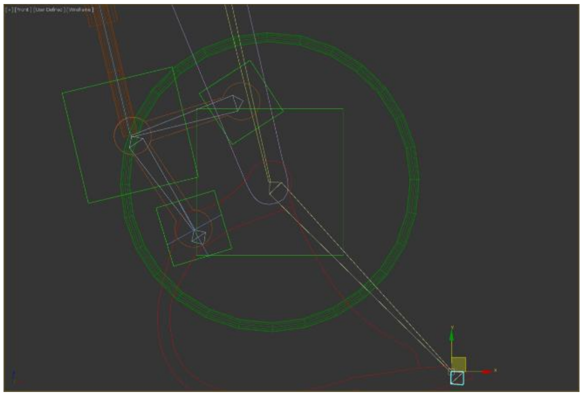


22. From the **Main** toolbar, click **Select and Move**. 

23. Move the bone down to close the scissor joint, and up to open it.

Notice that as you move the bone the scissor joint opens and closes as a result of the IK.





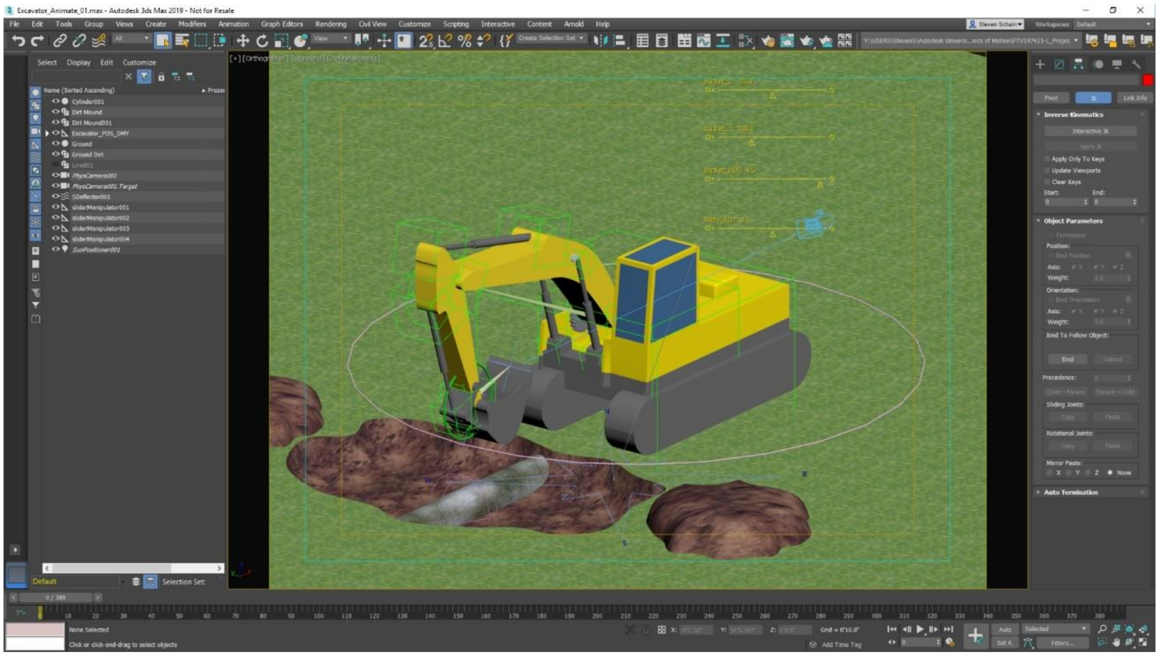
The bucket rotated to its two extremes. Pay attention to the position and orientation of the scissor joint objects as the bucket rotates. They operate automatically, now that they have been incorporated into the excavator rig.

Animating a Rigged and Wired Assembly

individually, hierarchical linking, wiring, and inverse kinematics are handy tools available to animators within 3ds Max. However, when combined together, they can make for a very powerful and flexible animation system.

The file in this exercise is configured using this combination in conjunction with on-screen manipulators. Specifically, sliders are configured to animate several aspects of the excavator's motion. With these manipulators, the animator has a simple dashboard from which they can create more complex animated sequences.

This exercise will consist of using the manipulators to create keyframes for the bucket X and Z, the bucket rotation, and the body rotation for the excavator.
Open the file **Excavator_Animate_01.max** for this exercise.



Creating Keyframe Animation

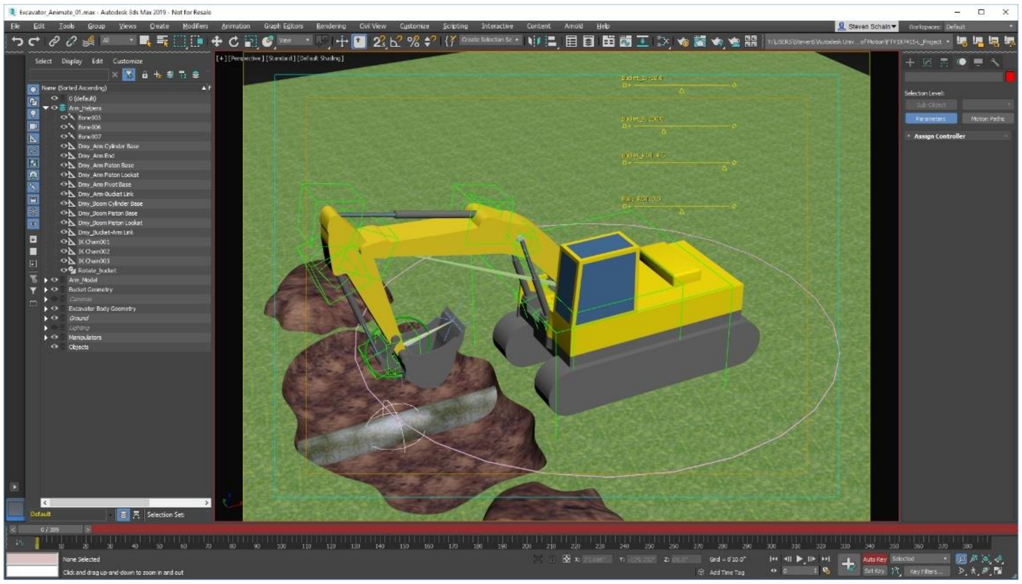
Using the manipulators, you can create keyframes that will allow you to animate the upper portion of the excavator. Slider Manipulators are special objects that function in the screen space of the viewport, not in the 3D model space. By using these manipulators, you can create a remote-control panel for whatever it is you're animating.

Manipulators are operated by using the **Select and Manipulate** tool from the main toolbar. Once enabled, the sliders become active for each manipulator on-screen. The sliders can then be keyframed, animating the excavator assembly.

1. Click **Auto Key** in the lower right of the interface, to activate the auto keyframe animation feature within 3ds Max. 

2. From the **Main** toolbar, click **Select and Manipulate.** 

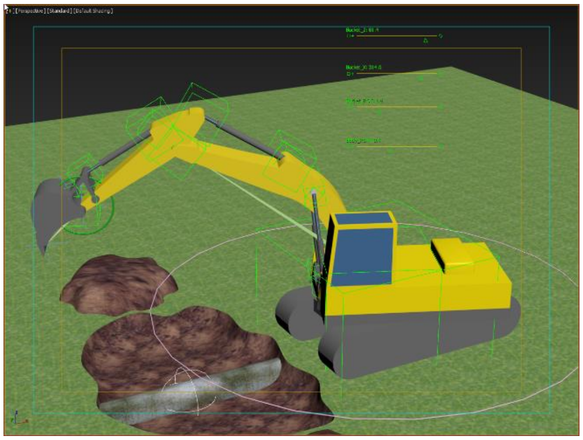
With Auto Key active, the time slider background and viewport frame turn red. This is an indicator that you are in Auto Key mode. Also, notice that the manipulators have changed color—the slider arrows are now green and are able to be adjusted.



3. Drag the **body rotate** slider to about **-0.4**.

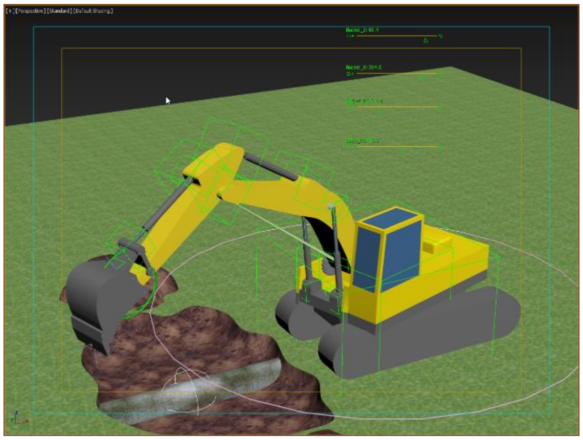
   One of the benefits of using manipulators is that you can set minimum and maximum values. You can also set the position and what value the slider will snap to as you drag it. Also note that the rotation is currently in radians, not degrees. And one radian equals approximately 57.3°. While you can adjust for this when wiring the parameters, for visual animation, it's not really necessary.

4. Drag the **bucket Z** to the right to about a value of **90**, raising the bucket.
5. Drag the **bucket X** to the right to a value of about **275**, pushing out from the main body.
6. Drag the **bucket rotate** to the left to about **1.9**, extending the rotation of the bucket.

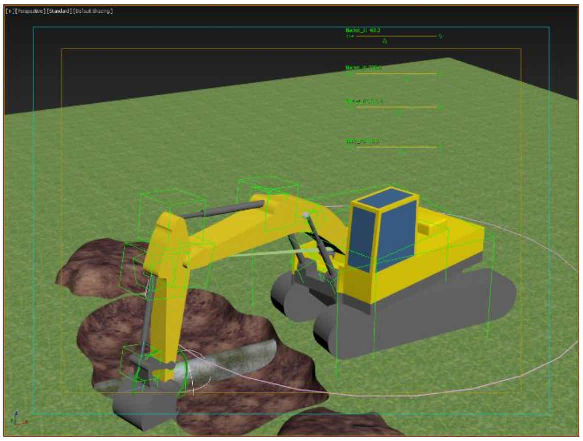


The excavator rig positioned for the start of the animation.

7. Drag the **time slider** to frame **100**.
8. Drag the **body rotate** slider to **0.3**, and then drag the other three manipulator sliders over one notch. This will cause a keyframe to be made at frame 100 for this value.

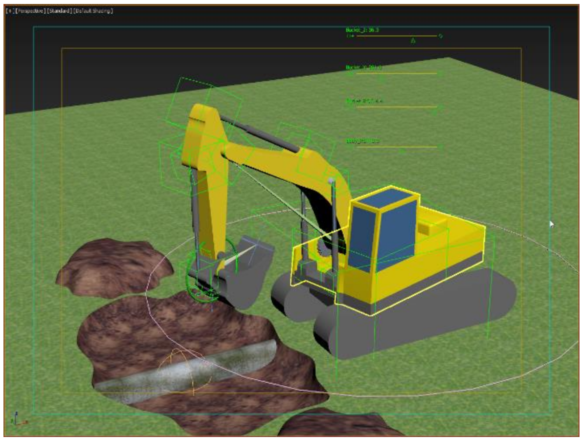


9. Drag the **time slider** to frame **150**.
10. Drag the **bucket Z** slider to the left to about **-63**, and the **bucket rotate** slider to about **2.8**.



11. Drag the **time slider** to frame **200**.
12. Drag the **bucket rotate** slider to about **4.0**, the **bucket X** slider to about **230.0**, and the **bucket Z** slider to about **-19**.

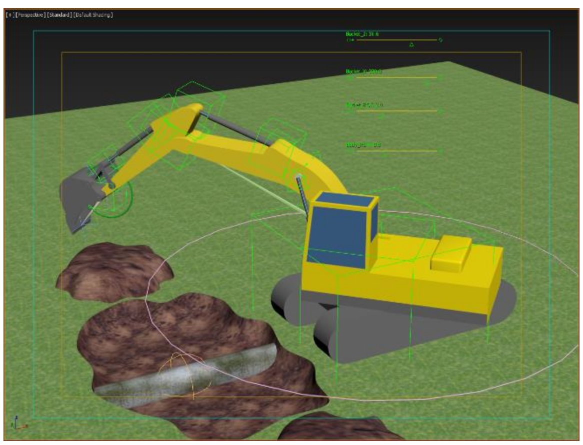
13. Scrub the **timeline** from frame **0** to frame **200** and watch the animation.
14. Slide the **time slider** to frame **270**.
15. Adjust the **bucket Z** value to about **37.6**, the **bucket X** to about **200**, and the **bucket rotate** to **4.4**.



Notice a pattern? Change the frame, adjust manipulator slider value, repeat. Creating this animation dashboard makes it easy to animate complex assemblies with minimal effort.

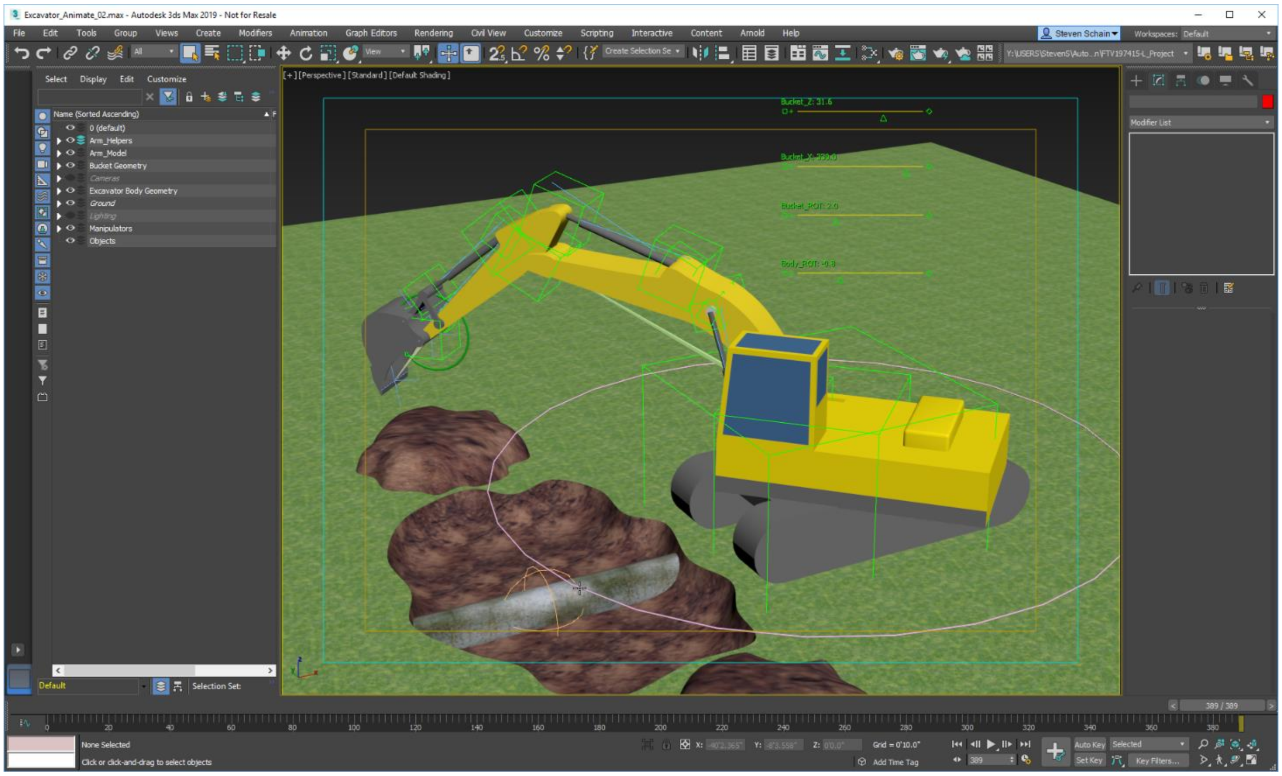
Now, finish up the animation and preview what has been created.

16. Slide the **time slider** to frame **340**.
17. Adjust the **bucket Z** value to about **32**, the **bucket X** to **330**, and the **bucket rotate** to **2.0**. and **Body Rotate** to **-0.8**.



18. One last keyframe. Slide the **time slider** to frame **200**.
19. Adjust the **body rotate** back to **0.3**. This sets up a keyframe at frame 200 for the rotation of the body, keeping it stationary between frame 100 and frame 200.
20. Turn **off** Auto Key.
21. In the animation controls, click **Go to Start**.

22. Click **Play** to play back the animation.



THE FINAL VERSION SHOWS THE EASE OF USING SLIDERS FOR CREATING ANIMATED ASSEMBLIES.


For more Information:
www.cadlearning.com
sschain@cadlearning.com
The Blast Magazine
facebook.com/CADLearning
@3DProfessor
linkedin.com/in/sschain/