

MFG226300

# VR From Scratch

Ryan Johnson  
D3 Technologies

## Learning Objectives

- Learn what software and hardware options are out there for your Virtual Reality needs.
- Learn how to import/export your designs into 3ds Max Interactive and out to your VR gear.
- Learn how to add navigation elements to travel in and around a facility.
- Learn some methods for building interaction with your model.

## Description

This discussion will look in on the process of getting started in developing for Virtual Reality using 3ds Max Interactive. There I was. I had no prior experience developing VR, and a short lead time for the requested content. What software do I use? What hardware would I need? What about you - are you wondering what it will take to get your facility design into VR for immersive navigation? Are you wondering how to get started in making your machine or product interactive in VR? Where do you go when you hit a roadblock? This course covers all those topics from a beginner's perspective so you can learn from someone else's mistakes and victories.

## Speaker

I'm a Senior Implementation Consultant for D3 Technologies, an Autodesk Partner and Authorized Training Center. I specialize in Autodesk software training and consulting for a variety of customers across the US.

## Intro

Last year at AU I was sitting in my first 3ds Max Interactive class. I was excited about the possibilities I saw, and I wanted to jump in right away.

This year I have a presentation on VR development. How did I get here?? Better question: What are YOU doing here?

Since that time last year things happened quickly. I proposed a VR project to a client. They were interested! Next, I submitted this class proposal to Autodesk, and they accepted it! You may have noticed the class description explains this topic is “from a beginner’s perspective” – I don’t know how much more of a beginner I could be. However, I’ve learned a lot in my short time with this project. This is also a unique opportunity for me, because it’s probably the only time I can ever truly present this topic from this perspective. I hope I can answer some of the questions you might have if you’re in the same shoes.

I am not a full-time VR or game designer. This quickly became evident as I worked on this project in my spare time, including nights and weekends. Way too many weekends. I’ll be upfront about this, the more you can commit to your VR project, the better it will be. If you only give it time in-between other projects, expect corresponding results.

You may also notice I said you should “learn from my mistakes” – and I made plenty. This is supposed to be a fun presentation. We will cover many topics, some simple & some complex. We will see places 3ds Max Interactive shines – and several places where I ran into roadblocks. The goal is to propel you past this painful 6-month experience I’ve had. Above all – have fun! You should be able to pick up some useful knowledge.

I’ll be using 3ds Max Interactive and developing for the HTC Vive. Some of the content may apply loosely to developing using other templates for other headsets because, Flow Editor.

This field is moving quickly. Already since starting this project some of my data has become obsolete. To keep up to speed make sure you’re keeping up with the tech news on VR/AR/MR technology (or whatever it will be called in a few years.)

## So much hardware, so much software!!

You need to decide the type of experience you want to create before you start. This can affect your choice of hardware & software to use during your development.

### Hardware

The “reality” hardware field is growing at a rapid pace. There are several types of hardware out there, some tether to a computer or even use a smartphone screen to deliver their experience. I would divide options into these categories, with some examples of each:

- Immersive: these systems allow a user to be immersed in the environment. They are however usually more limited in processing power, so require simpler environments. They generally do not allow the user to physically walk around in their environment.
  - o Google Cardboard
  - o Google Daydream
  - o Samsung Gear
  - o Oculus Go
- Interactive: these systems all true room-scale experiences, where the user can move physically walk around the environment. They are generally higher-power than the previous group. Some are tethered to a computer, others can run their own processing.

#### Virtual Reality

- o HTC Vive
- o Oculus Rift

#### Windows Mixed Reality

- o Lenovo
- o Acer
- o HP
- o Dell
- o Samsung

#### Augmented Reality

- o HoloLens
- o Magic Leap



## Software

We are not discussing 3D modeling here. Assuming you have a 3D model, how do you get your 3D model into a state you can view & interact with it?

Game development software. That's right, game design. To build an interactive or immersive experience, you're basically building a game. Some software options out there are listed below. This field is also changing rapidly.

- 3ds Max Interactive
- Unity
- Unreal
- CryEngine
- Lumberyard



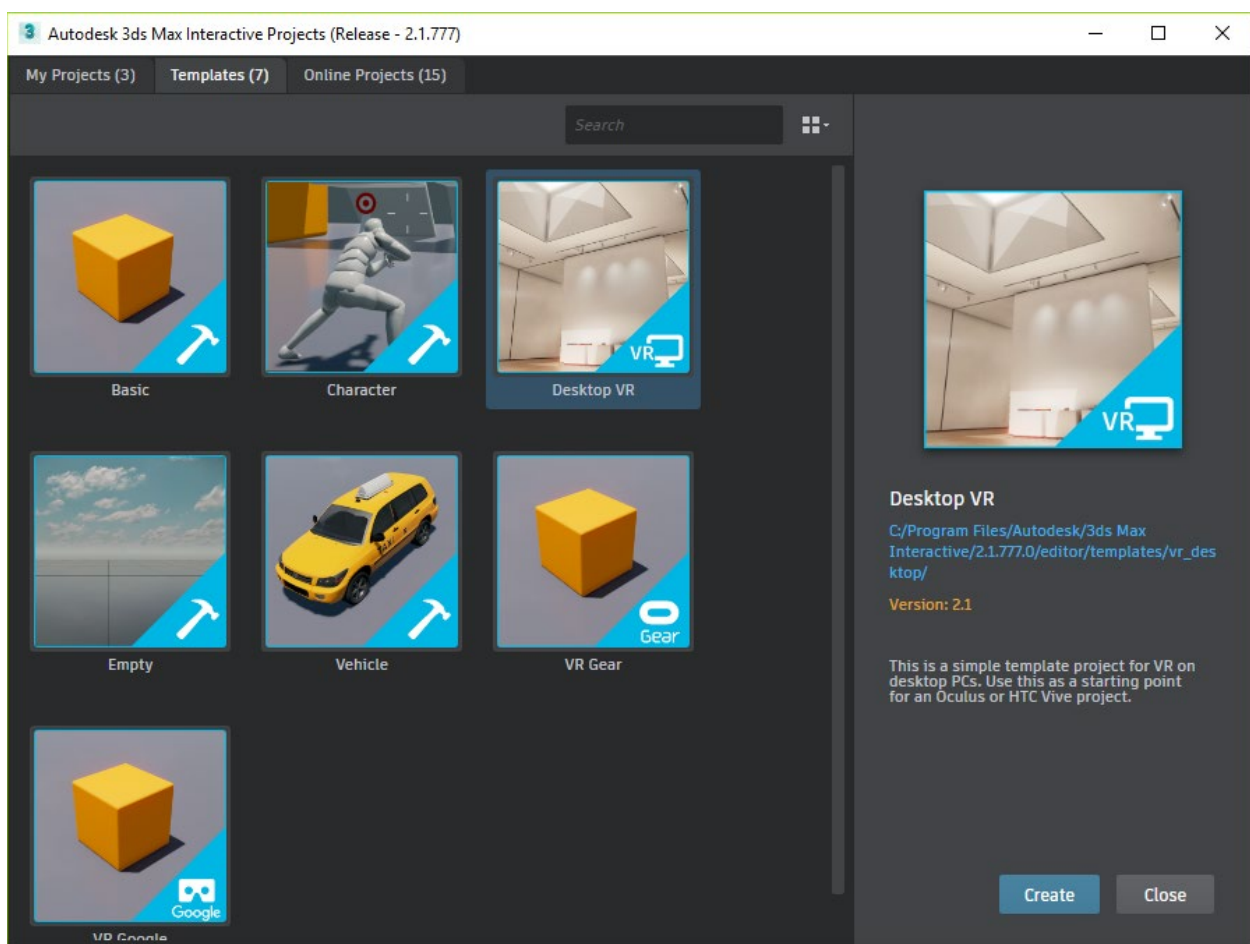
Of course, this is Autodesk University. There are several sessions on Unity and Unreal here, but since I'm a Product Design & Manufacturing Collection user, in this presentation I will be using 3ds Max Interactive along with an HTC Vive VR system. The fact that I already had it available with the PDMC and that it tied so well to 3ds Max were both huge reasons for me to go in that direction.

## Getting your stuff into 3ds Max Interactive

Since you do not create models in 3ds Max Interactive (3ds MI from here on out, because that's a lot to type), the first step is getting your stuff into 3ds MI. Before we get there though...

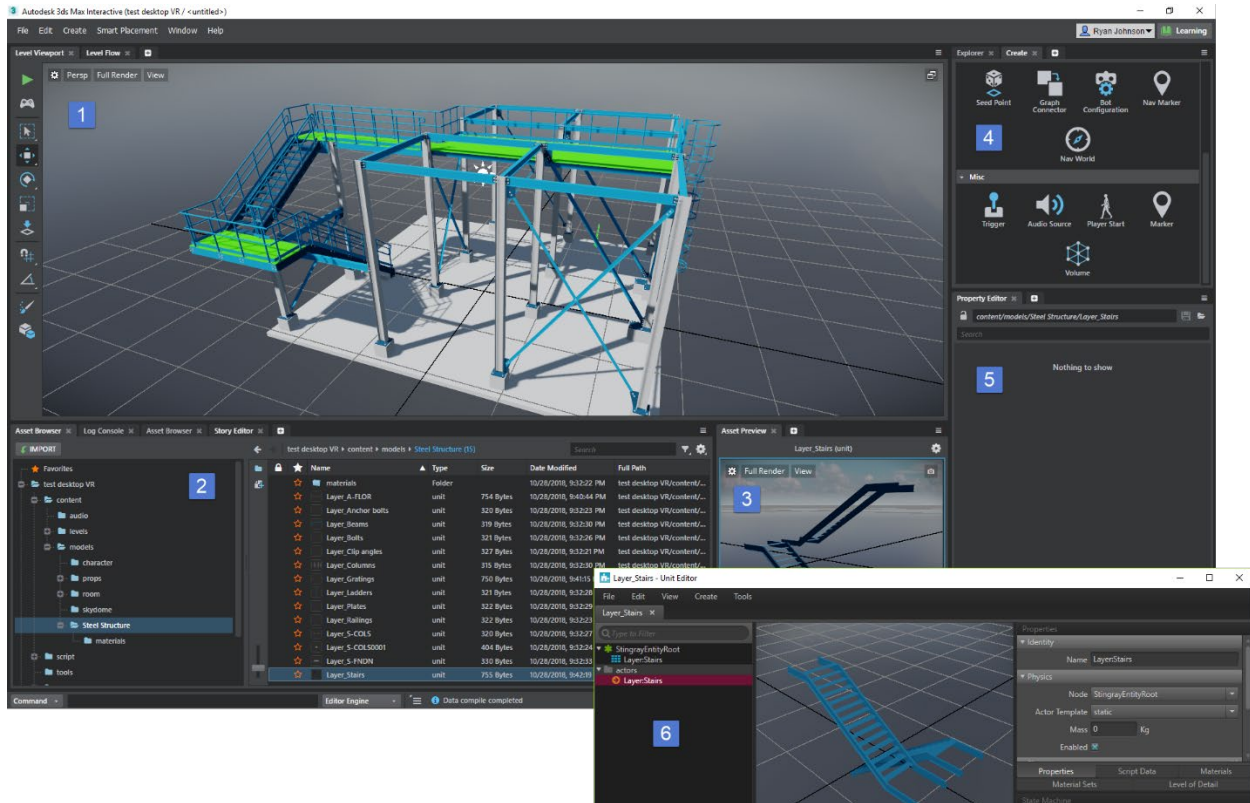
### Getting 3ds Max Interactive ready for your models

When using 3ds Max Interactive, you'll pick a starting template depending on what platform you'll be developing for. In this case I'll be using the Desktop VR template, which has several built-in pieces for desktop VR such as teleporting and laser-pointing. Using the right template is essential for the beginner. Without this template you'd have to program a lot of these elements from scratch.



3DS MAX INTERACTIVE – PROJECT MANAGER

## 3ds Max Interactive UI



3ds Max Interactive has several important parts of the UI we'll be discussing in the following pages. Most of these panes can contain multiple items depending on which tab is active. You can also add tabs with the + button in many areas. After each number I'll discuss the common tabs for that area

- 1- **Level Viewport** – interactive view of the level you're building  
**Level Flow** – used for building complex interactions and events
- 2- **Asset Browser** – contains all assets used in the project  
**Log Console** – shows event logs, useful for troubleshooting errors  
**Story Editor** – used for building simple animation inside 3ds Max Interactive
- 3- **Asset Preview** – preview of currently selected asset
- 4- **Create tab** – used for creating new assets in 3ds Max Interactive (not 3D modeling)  
**Explorer** – similar to the Asset Browser, shows active assets in a level
- 5- **Property Editor** – used to modify currently selected asset's properties
- 6- **Unit Editor** – used for modifying units or setting up unit flow



## Adding Your Model – via 3ds max

The first and easiest method is to start in 3ds Max and send a scene and its assets into 3ds MI automatically. If you're already familiar with 3ds Max, this will allow you to do the preliminary setup in 3ds Max. It also allows you to model and manipulate objects and still get them quickly into 3ds MI.

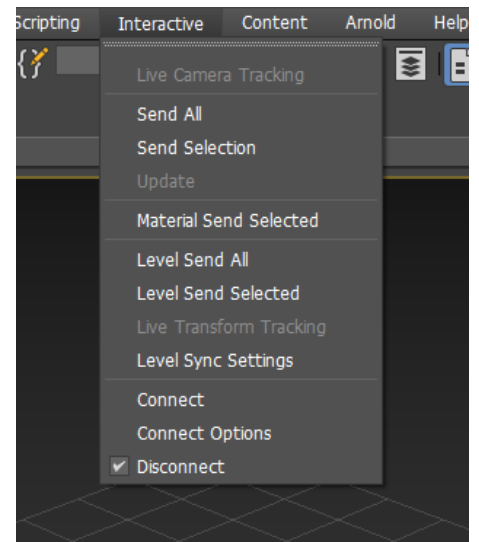
Sending the scene is simple. Use the Interactive menu > Level Send (All or Selected). Level Send Selected lets you send and update specific objects only if you're manipulating the scene after initial testing. These tools will automatically send the scene to the currently active level in 3ds MI.

### Materials

When I first started sending models to 3ds MI, the materials were a mess. All the presentations and examples I saw seemed to indicate it should be simple and automatic, however only a fraction of my material colors made it, and definitely no maps. After some digging I found out that 3ds MI supports:

- V-Ray materials (3<sup>rd</sup> party)
- Standard materials

You could possibly convert other materials like old Mental Ray or Arnold materials using the scene converter, but I just opted to get or build some simple Standard materials in 3ds Max for my scenes.

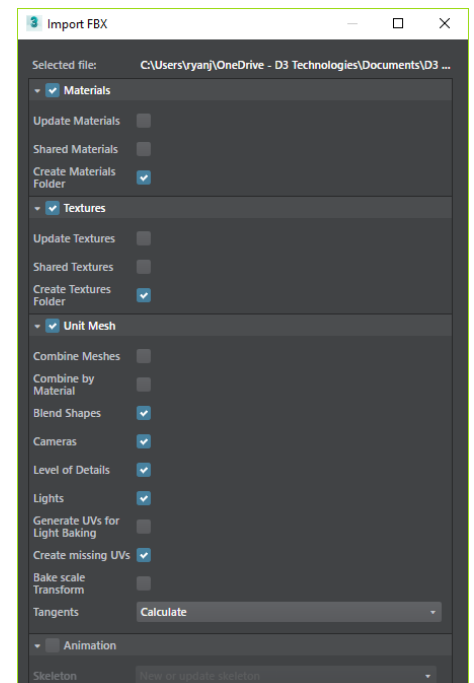


## Adding Your Model – via FBX

If you will not be using 3ds Max in your workflow, you can also import FBX models directly into 3ds MI. For me this added a lot of extra work because I did not want to mess with learning materials in 3ds MI. Also, I needed to add extra models, animation, and other stuff to the scene which I could easily do in 3ds Max. However, you may have a complete FBX scene and wish to dump it straight into 3ds Max.

If you are planning on using this workflow, you have an opportunity here to organize your project well, or create a huge mess.

I recommend you use the Asset Browser (docked at the bottom of 3ds MI by default) to locate your [Project] > content > models folder. You can right-click the folder pane and "create folder." Right-click > Import Asset to import a new FBX asset. You will have a variety of options here on what to include with the FBX import, including animation.



## **Materials**

After importing from FBX, you may need to work with materials in 3ds Max Interactive. With the right options selected during import, you will have a nicely organized folder inside each model folder containing all your materials. Simply select any material in the Asset Browser to adjust its properties in the Property Editor.

**Adding Cloud Models – via Online Asset Browser** – You can also access some models and materials custom-built for 3ds MI using the Online Asset Browser. You'll find it at the bottom of the Asset Browser. Just double-click on any asset in the browser to download it for use in your project.

You can right-click any online asset that has been downloaded and choose "locate in project" to be redirected in the asset browser to the location of that item.

## **A note about Optimizing Models**

I'm not going to spend much time on optimization. I spec'd out a powerful VR computer that could handle what I threw at it. However, for very large scenes or complex models – not optimizing can result in the VR not being able to keep up with the rendering. The resulting lag time in your level can cause motion sickness very quickly. Remember – a VR experience is essentially rendering in real-time, so make sure your hardware can handle it!

So, you used one of the above methods. That's it! You can don your headset now and look around at your model.

If your headset is connected and ready to go, just click the green "play" button at the top left of 3ds MI. This will launch your level in an interactive state for you to test.

This is one of the coolest parts of 3ds MI by far. You don't have to compile some program and run it separately. You can just click play to connect to your headset and try the changes you just made in real-time. This is a huge time-saver when you're in the middle of developing content.

But wait, I'm stuck in this single position. I want to move around in my scene! What's next?



## Navigating through a scene

There are a couple navigation methods you can use in your VR masterpiece.

### Old-school moving

You can use a trigger, a controller, or keyboard keys to fly or walk around in the model. I WILL NOT BE TEACHING YOU THIS! I got sick after 5 minutes flying around Google Earth and other demo games where the camera or character motion did not match my own. It's a very quick way to get motion sickness. I decided I would not put even my worst enemy through that.

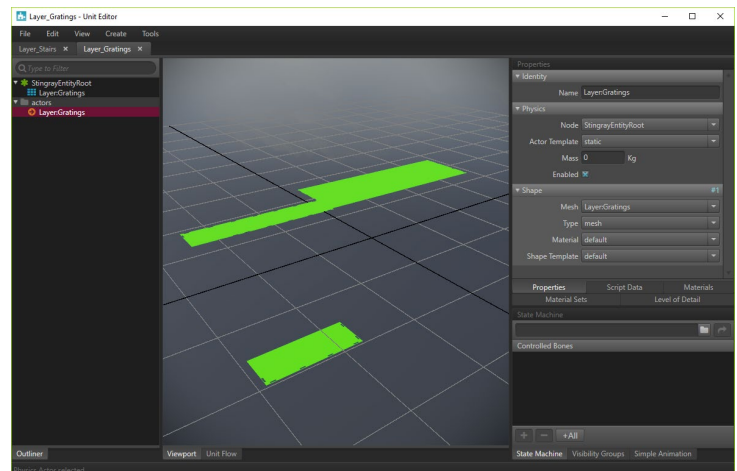
### Teleporting

This is what we'll be doing. Teleporting is built into the Desktop VR templates in 3ds Max, so there are really not too many steps to complete here. We'll be teleporting around our scene in no time.

### Physics Actors

Physics actors are necessary any time you want to interact with anything in your scene. In this case, we want our teleporter "target" to be able to land on the object we want to teleport onto. So, it needs to have a physics actor assigned to it. Follow these steps to enable teleporting on an object (generally a floor or platform).

- 1- Locate the asset, right-click > open in Unit Editor
- 2- In the Unit editor, right-click the object > Create Physics Actor
- 3- Make sure the Actor Template is set to "static."
- 4- Pick the appropriate Shape Type. Simplified shapes are preferable as they require less calculations. Mesh will attempt to regenerate using the existing mesh, but results in a much heavier or more costly mesh.
- 5- Save and close the Unit Editor.



That's it! Go ahead and click the Play button and see if you can teleport.

If you don't already know from all the gaming you did during your "research" phase, you usually teleport with the HTC Vive by clicking and holding the thump pad. You'll see a green teleport target (or red if you can't go there). When you release the button, you'll teleport there.

## Interacting with your stuff

This is the biggest topic of this presentation. While navigating is also very important, there are just not many steps. Building action, reaction, interaction is the biggest part of the project in my experience.

### Picking Stuff Up

This is one basic interaction which is always fun. Since it's fairly easy, we'll start here.

Just like Teleporting, picking objects up requires them to have Physics Actors assigned to them. In this case though, the objects will not be static.

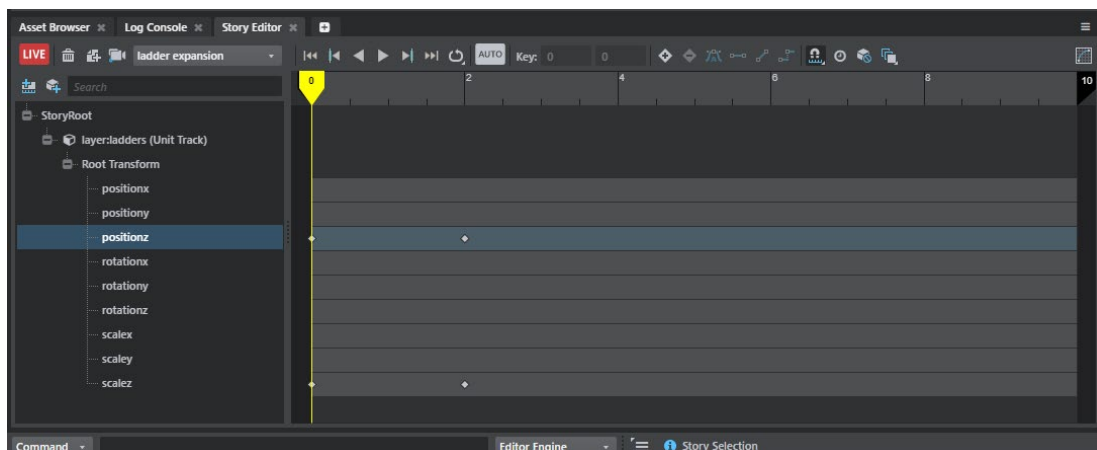
- 1- Find the object you'd like to pick up, and go to its Unit Editor.
- 2- Assign a Physics Actor.
- 3- This time, however, we'll make the Actor Template *dynamic*.
- 4- If possible, use a simplified type other than mesh.
- 5- Save the unit and close the Unit Editor.

If you did everything correctly, you should now have an object you can pick up with your controller. Hit the play button and try it out. With the HTC Vive you'll touch the object with your controller (it will highlight when you've touched it and you should feel some haptic feedback in the controller). Once you're touching it, squeeze the trigger to latch on to the object and pick it up.



- Note – If you started with an Inventor Assembly or a Groups in 3ds Max, you cannot assign one physics actor for the group. Use Shrinkwrap in Inventor to get a single object or attach in 3ds max to combine objects into a single mesh for a single physics actor.

### Story Editor

The story editor allows to build simple animations or actions in 3ds Ml and trigger them using Flow Editor, which we'll get into next. First let's create a simple Story. Start by turning on the Story Editor tab at the lower left. Click the + and select Story Editor.



This is the first of several areas where 3ds MI works by having you preselect something before taking an action. This is a very smooth workflow once you're used to it, but may take a while to remember. For the story editor, we need to first select the asset we want to create a story of.

- 1- Select the asset you want to animate in the Level Viewport
- 2- In Story Editor, click the "add story" button  and rename the story in the Property Editor
- 3- Expand the Root Transform node and select the transform(s) you want to animate
- 4- Click the "add key" button 
- 5- Slide the time slider (yellow tab) forward on the timeline
- 6- Change the value in the Property Editor
- 7- Click the "add key" button to add a new key with the current value

You now have a simple story created. Scrub the animation using the yellow slider or press the play button in the Story Editor to test it out.

## Flow Editor

The Flow Editor is one of the most powerful parts of 3ds Max Interactive. The Flow Editor is how you can build complex things without having to know how to write any code! Based on the template you chose and the flows you create, 3ds MI will convert it all to code in the end which drives your experience. We'll look at several types of interaction in the following examples. We'll be using Level Flow for flows on the level level and Unit Flow for flows on the unit level.

### Tying stuff to your HMD view

This is a cool tool to have in your toolbox. When you want an object's to always be oriented to a user, you can have Unit Flow tie the object's actual rotation to the HMD pose values.

Go to the Unit Editor for an object, then click the Unit Flow tab. In Unit Flow we just need to add a few nodes and wire them together. Just right-click to add a node.

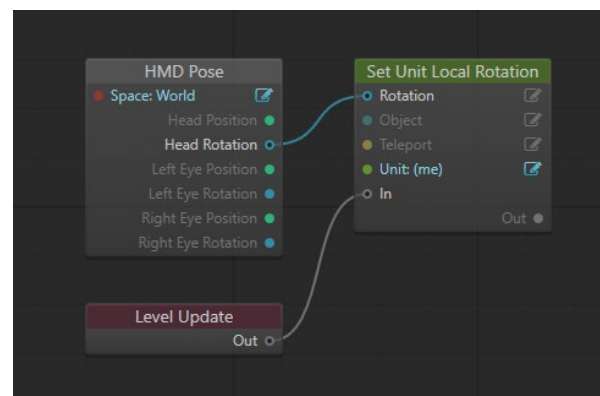
In most demos or instructional videos I've seen, the Flow Nodes are added so quickly you can't really tell where they got them. There are tons of them to dig through, so I'll try to detail exactly which ones you'll need in each step.

Here's what you'll need to add here:

- VR Tools > HMD Pose
- Unit > Set Unit Local Rotation
- Event > Level Update

How to connect them in Unit Flow:

Once that's done, you're good to go!!  
Click the Play button and give it a try



## Activating a story

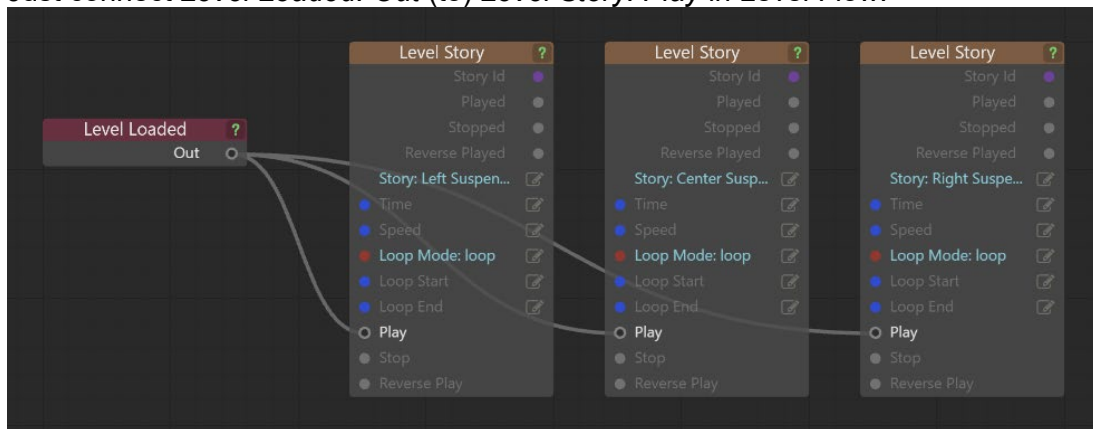
It's good to build any complex flow in stages, where you can verify one part is working before adding another layer of complexity to it. First, we'll take a look just at getting a story to play on level start, without any other triggers.

This time, instead of going to Unit Flow for a specific object, we'll be using Level Flow. You'll find this tab right next to the main Level Viewport by default.

What we need to add here:

- *Event > Level Loaded*
- *Level > Level Story*
  - o Click *Story* to select the proper story
  - o Click *Loop Mode* and select a loop mode
  - o You can also modify the *Speed* and other options here
- o There's another way to do this last step. Remember that often if you have an object already selected, it's much easier to add in Flow. In the Story Editor tab, select Story Root. Now when you right-click in Flow Editor you'll see a shortcut to add a Level Story node with the story already specified.

Just connect *Level Loaded: Out (to) Level Story: Play* in Level Flow:



Now when you play the level, your story should immediately begin playing! And if you selected *Loop Mode: loop*, it will keep playing, and playing, and playing. Let's say you want to play and stop the story on demand? Well, we need to add another layer here.

## Activating with a Keyboard Key

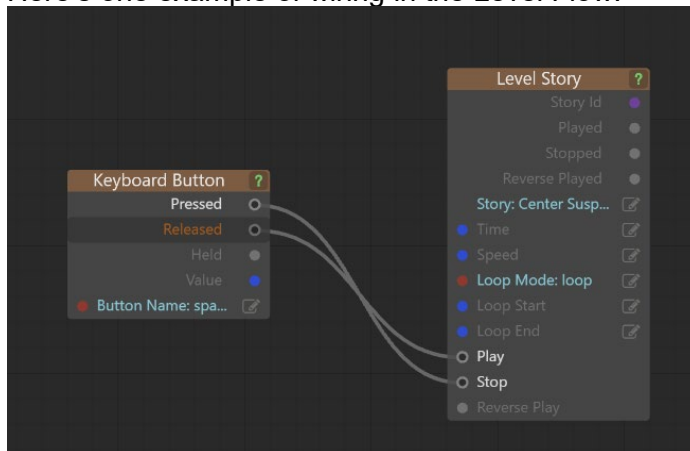
Activating a story with a keyboard key is a quick first pass at making sure you will be able to trigger an animation and that it will play and stop as you intended. We're going to add a Keyboard Key to our Level Flow.

Here's what to add:

- Input > Keyboard *Button*
  - o Select *Button Name* and specify which Keyboard key you want to press. *Space* is what I usually use for testing. It's a nice easy one to find by touch when you have a headset obscuring your real-world vision.

Now you just need to wire the correct button action to the Level Story node you'd like to be triggered with that button. You can even add multiple Keyboard Button nodes to see what happens with your story at each.

Here's one example of wiring in the Level Flow:



### Activating with Volume Trigger

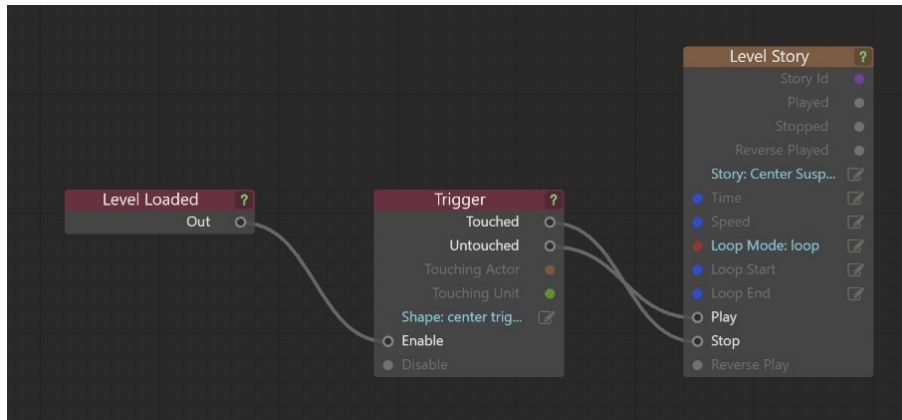
Activating with a Volume Trigger is a really cool option that I planned to use all over my project. However, it seems out of the box that I cannot get this working properly. I'm sure there's some simple thing I need to do to make it work. However, the point of this presentation was VR From Scratch. So, I don't want to have to open any code or mess with anything that's not working as intended.

For now, I will skip getting in depth on this topic. As I get feedback on the project and if I can get it working properly without a ton of manual coding, I'll update this handout with what I learn.

Here's what you *would* need to add:

- *Event* > *Level Loaded*
- *Event* > *Trigger* (or preselect the Trigger and use the shortcut to add to Flow)
- Tie this to the *Level Story* you already added which you know works

Sample Level Flow which I hoped would get this working:



There are some more details about setting the Trigger to the correct template and stuff. I'll get to those details if I get to update this.

### Activating with Laser Pointer

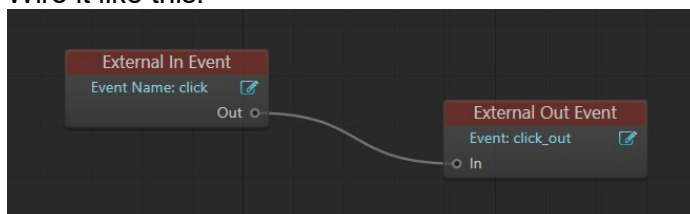
This is another triggering mechanism which you can build based off the Desktop VR template. For the HTC Vive, when you squeeze the trigger of the controller you'll see a laser beam which extends through space if you're pointing at any object that has a Physics Actor when you trigger. Getting the Trigger to actually give you action takes a few extra steps, but it's a really cool thing once you get it working.

First, you'll need to open the Unit Flow for the object you want to trigger on (think like a button you want to shoot with the laser). This is a little outside the box, but we need to make a new event we can listen for out in the Level Flow. The mechanisms to tie this to actual trigger presses are all built into the Desktop VR templates, so just follow the steps and you'll be good.

What you need to add:

- *External > External In Event*
  - o Name it *click*
- *External > External Out Event*
  - o Name it *click\_out*

Wire it like this:

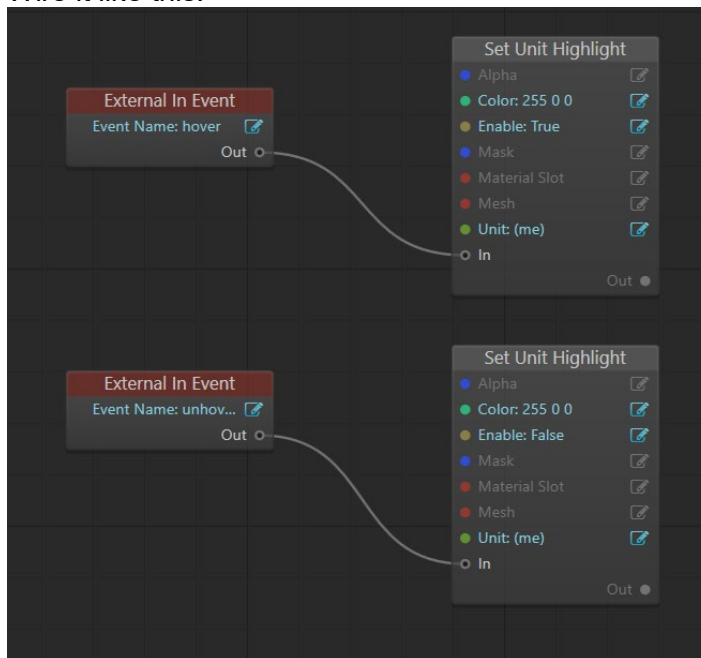


That's all you really need, but you can add a couple more for a cool effect where the object will be highlighted when you're hovering on it with the laser.



- *External > External In Event*
  - o Name it *hover*
- *External > External In Event*
  - o Name it *unhover*
- *Unit > Set Unit Highlight*
- *Unit > Set Unit Highlight*

Wire it like this:



Once you have the Unit Flow set up, it's time to go back to the Level Flow.

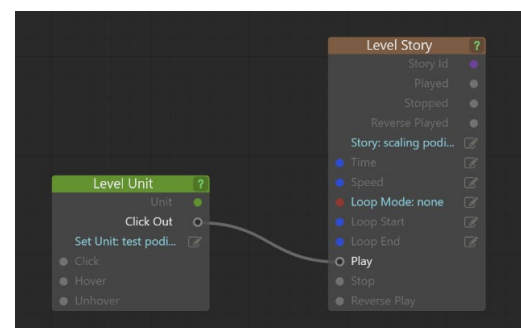
Add this:

- *Data > Level Unit*
  - o Set *Unit* to the object you just added all that stuff to in the Unit Flow

You'll notice that the Level Unit node automatically populates now with all the In Events we added in the Unit Flow!! This is cool, it gives us a way to tie things down into that Unit Flow and back out.

Just wire up the *Click Out* node to the *Play* node of your Level Story that's already there.

Go ahead and press Play to test it out! You should now have a story that starts playing as soon as laser unclicks the object you chose. Obviously, there are other ways to make this work too.



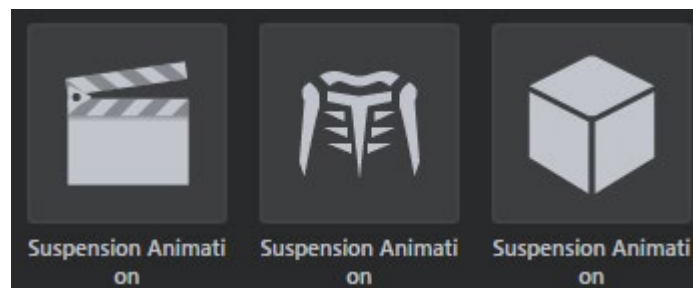
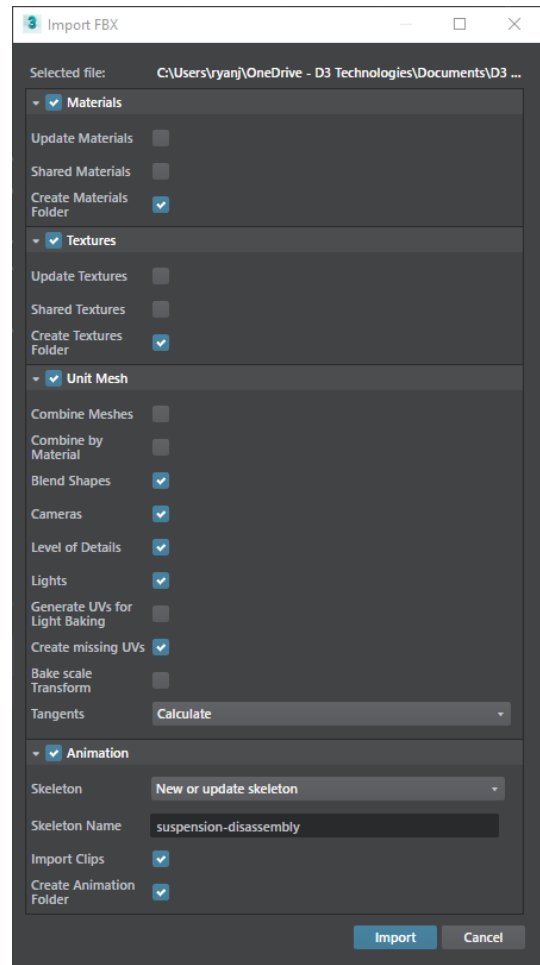
## Activating an animation

In the entire previous section, we added animation completely in 3ds Max. This next method can be used when you want to import a more complex animation from an FBX file.

In this case we will start by building the animation in a separate scene file, using objects that don't exist in our scene yet.

- 1- In the Asset Browser, create a folder for the animation.
- 2- Start Import and choose your FBX file.
- 3- Include geometry and animation, make sure to check "create new skeleton."
  - a. All the objects below should be created. If they weren't, something went wrong.
  - b. This is the real factor – although every demo and tutorial I saw had this working simply with a couple clicks, I had to work at it and reimport/recreate scenes several times before I got it working.
  - c. Although technically probably not a requirement, importing the objects and animation from a separate scene straight into 3ds Max made this process much simpler for me.
- 4- You should automatically have a new model and also a new Story to go along with it!

Once the animation is imported, you can use the same options for triggering it as we covered above with Stories.



## Conclusion

Hopefully this gives you a good jumping-off point! You now know more about what's possible, and what it will take to get you there.

There was a lot we didn't even talk about here, like switching the default level that loads when you start your project, or compiling your stuff into a program which you can distribute and run without 3ds Max Interactive. These topics are pretty easy to find info on and for time I wanted to stick to the objectives I set out to cover.

Besides the technical aspects we've covered on 3ds Max Interactive, and based on my experience as a beginner jumping into this world, I can say the following things will be very important to remember.

- **PLAN AHEAD!** Don't get stuck with a deadline you can't fulfil.
- **Set Proper Expectations.** Make sure whoever you're working with understands you will need time set aside to focus on this project. It will be difficult to finish if you're working on it here or there as you get time.
- **Set aside time and stick to your timeline.** Do not procrastinate or assume you can catch up later if you get busy with other stuff. Your project plan is the tool for you to stay focused and on-task

Above all, have fun!!

Are you stuck? Need help? Following are a few of the common resources I've used during my time with 3ds Max Interactive.

- Great AU Presentations from other awesome developers
- The 3ds Max Interactive help system:  
[http://help.autodesk.com/view/3DSMAX/2019/ENU/?guid=interactive\\_help\\_getting\\_started\\_html](http://help.autodesk.com/view/3DSMAX/2019/ENU/?guid=interactive_help_getting_started_html)
- The 3ds Max and Stingray product forums: <https://forums.autodesk.com/>
- Connections and colleagues I can hit up on LinkedIn or send a quick email to.  
Connect with me on LinkedIn: <https://www.linkedin.com/in/ryaanj/>
- YouTube tutorials
- 3ds Max Interactive Getting Started tutorials – these automatically appear when you're first launching 3ds Max Interactive