



# DMX via Stream Panel Guide

## **Introduction:**

Welcome to VR Stage Lighting (formerly known as “Custom VR Stage Lighting”), VR Stage Lighting is a year-long project that started out as a means to research and develop a performant/reliable way to send DMX512 (Digital Multiplex with 512 pieces of information) data to VRChat to control spot lights and other lighting effects. This guide was created to help users get started in using VR Stage Lighting’s DMX mode and the settings needed to get used to the pipelines designed to make DMX powered shows in VRChat.

## **Requirements:**

Here's a list of the required software/assets/hardware needed to get started:

- Unity 2018.4
- VRChat SDK3 for worlds (Udon)
- UdonSharp
- Usharp Video Player (<https://github.com/MerlinVR/USharpVideo>)
- PostProcessing Stack V2
- uDesktopDuplication 1.6.0  
(<https://github.com/hecomi/uDesktopDuplication/releases/tag/v1.6.0>)  
(for working with in-Editor; must be 1.60, Unity 2018.4 causes 1.70 to break for some reason)
- Open BroadCast Software (or streaming software of your choice)
- VR Stage Lighting Grid Node (For DMX control, more info below)
- At least one extra screen that can support 16:9 resolutions.
- The latest release of VR Stage Lighting
- Software/Hardware

## **Installation:**

1. Open Unity and download/install VRChat SDK3, UdonSharp, Usharp Video Player, PostProcessing Stack V2, and uDesktopDuplication 1.6.0 via their respective unity package files.
2. Download/Install the latest release of VR Stage Lighting via the included unity package files.
3. Download/install OBS or streaming software of your choice.
4. Download VR Stage Lighting Grid Node.

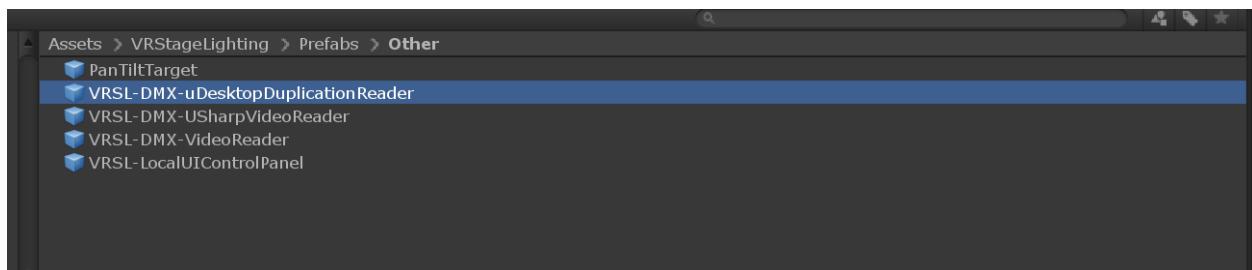
# Getting Started: (DMX VIA Stream Panel through Editor)

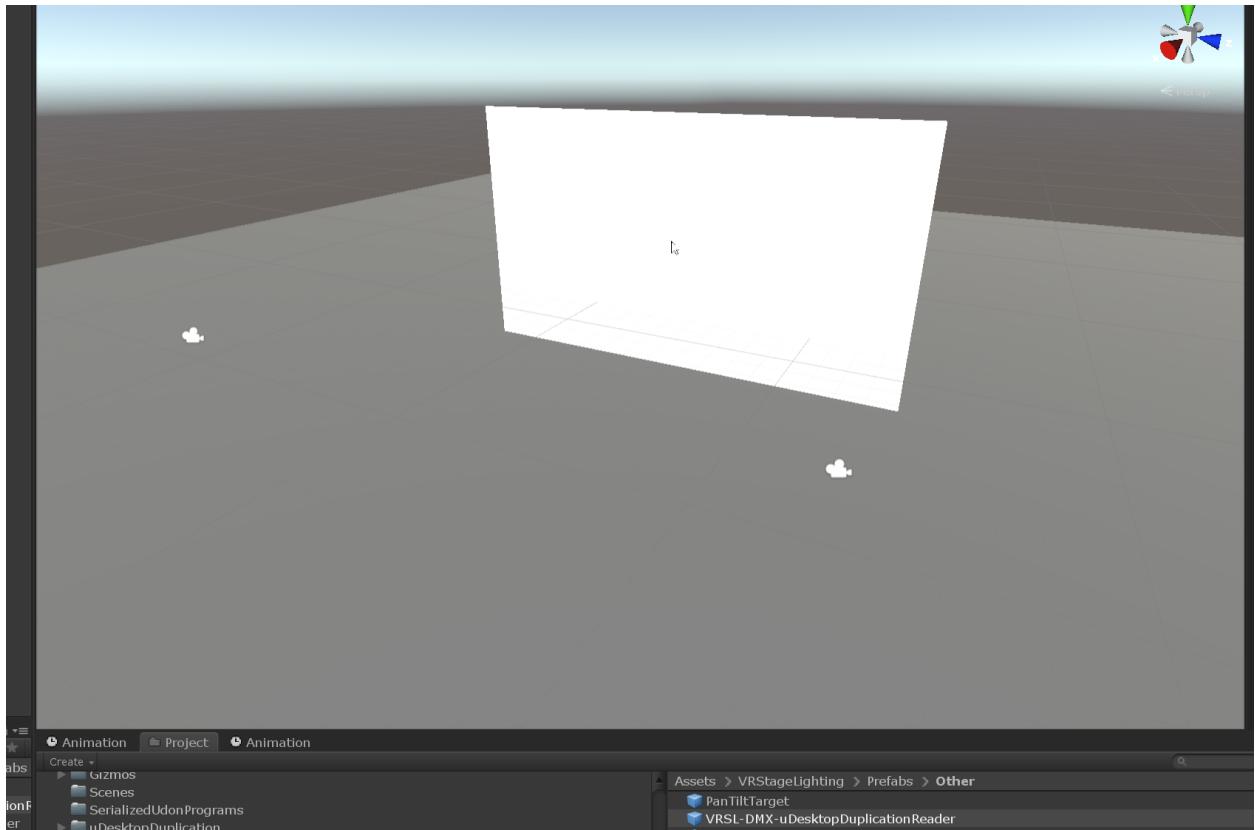
In this section, we will show how to use the “VRSL-DMX-uDesktopDuplicationReader” prefab to send DMX signals to the editor, emulating how it would work for a stream. This is an important thing to learn as it will allow you to design/see how your lights will look and move around in real time in response to your DMX grid without an latency as everything is local. If you just want a quick view of how the system works, go ahead and hit “Test & Build” in the VRC SDK while having the “VRStageLighting-ExampleScene-DMX-USharpVideoExample” scene open!

**NOTE:** *If you want to go ahead and jump straight into controlling lights, you can open the example scene, “VRStageLighting-ExampleScene-DMX-DesktopViewerExample.unity”, Which will have steps 1-3 done for you. It will also serve as a good example of how to properly perform steps 1-3 for your own worlds. The scene is located under Assets/VRStageLighting/ExampleScenes.*

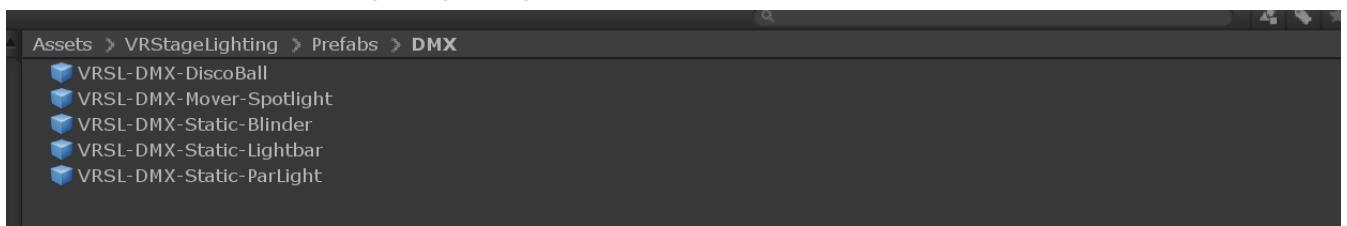
1. Open Unity with everything installed properly and drag “VRSL-DMX-uDesktopDuplicationReader” prefab into the scene of your choice.

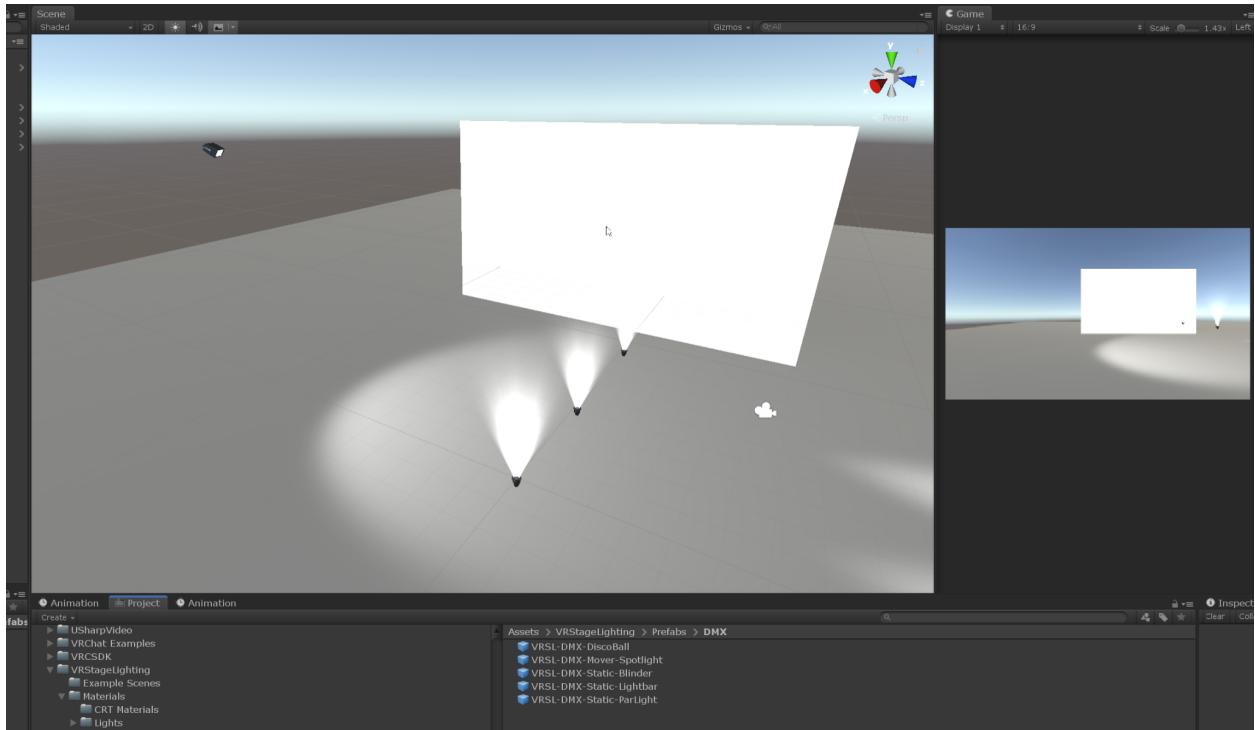
This prefab will be located under Assets/VRStageLighting/Prefabs/Other.



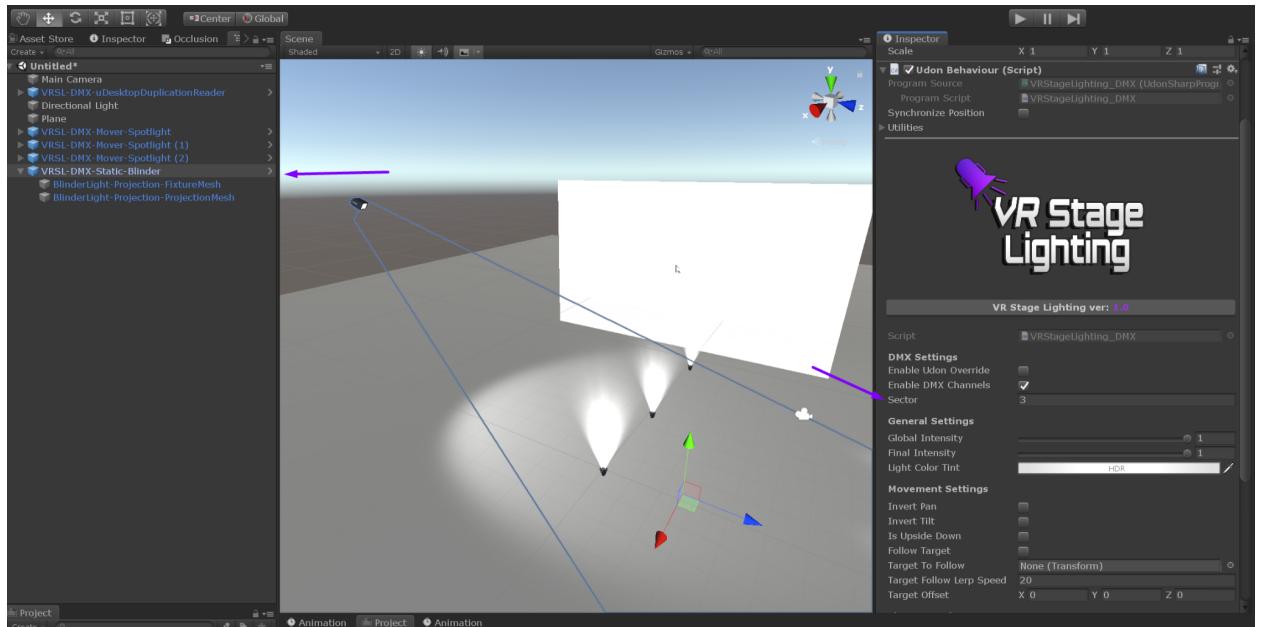


2. Add some DMX compatible light fixture prefabs. They will be located in Assets/VRStageLighting/Prefabs/DMX.



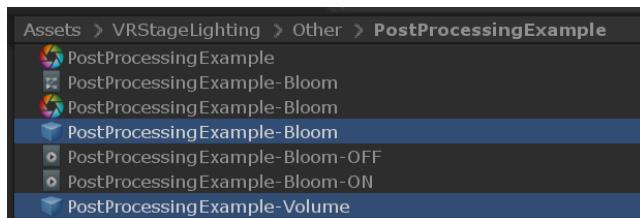


3. Assign sectors to your fixture prefabs through their udon scripts. Each sector represents the 13 channels that will control the fixture.

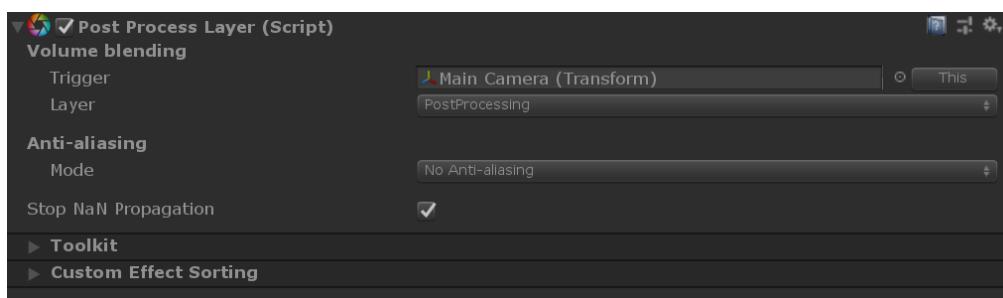




3.5: To enable post processing, add these post processing prefabs to enable bloom and color correction top the scene :



Then go to your scene's main camera and add a “Post Processing Layer” to it with these settings:

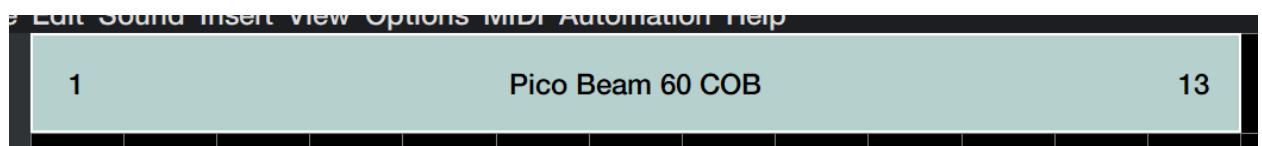


4. Open your Arnet compatible software of choice. (Recommendation: SoundSwitch)

5. Add/Patch multiple 13 channel fixtures that are according to this spec:

1. Pan (Left/Right Rotation)
2. Fine-Pan
3. Tilt (Up/Down Rotation)
4. Fine-Tilt
5. SpotLight Radius/Cone Width (Will show up as "Motor Speed" in some software)
6. Intensity
7. Strobe (0-009 is off, 010-255 is slow to fast)
8. Red Intensity
9. Green Intensity
10. Blue Intensity
11. Unused (Labeled as "White Intensity" in some software)
12. GOBO selection (1 through 6; Labeled as "Programmes" in some software)
13. GOBO Spin Speed (Work-In-Progress; Labeled as "Speed/Sensitivity");

In SoundSwitch, a fixture that fits this spec is the "Pico Beam 60 COB (Mode 2) (Chn = 13)". The documentation for this fixture can be found here on page 47: [PicoBeam 60 COB RGBW moving head user manual](#)



Every fixture type in VRSL is based around this spec, which will result in some channels not being used.

## IMPORTANT NOTES ABOUT DMX MAPPING FOR THE VRSL GRID NODE:

-Due to limitations of this system, the range supported channels for Universe 1 is CH1-CH512 and the supported channels for Universe 2 is CH1- CH346. This is due to how the grid system is currently made, expansion settings are planned for future releases.

-Since 13 channels doesn't fit evenly into 512, it is highly recommended to end Universe 1 on Channel 517 and begin Universe 2 on Channel 9. This is due to the fact that adding a 13 channel fixture on Channel 508 will result in it bleeding over to Universe 2.

- U1 Channel 508 to U2 Channel 8 will be reserved for auxiliary channels in future updates. Here is what is currently mapped to those channels by default:

--Channel 508: Disco Ball Intensity.

--Channel 509: GPU Particle Confetti (Coming Soon)

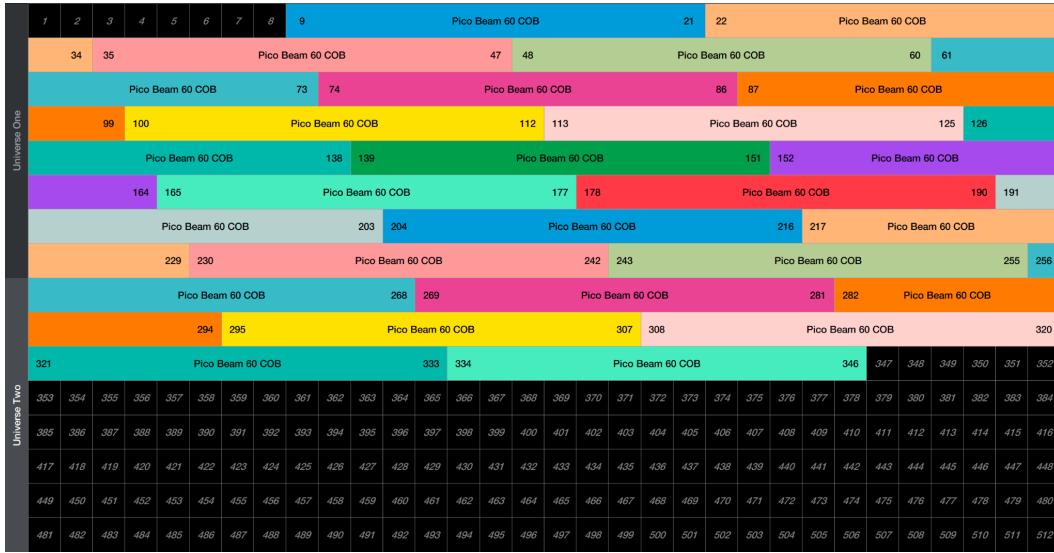
--Channel 510: Global Mover Speed (More Information Below)

--Channel 511-512: Fog Machines (Coming Soon).

--U2 Channel 1-8: Currently Unused; More Planned Features.

Here is the current max number of fixture combinations as illustrated in SoundSwitch:



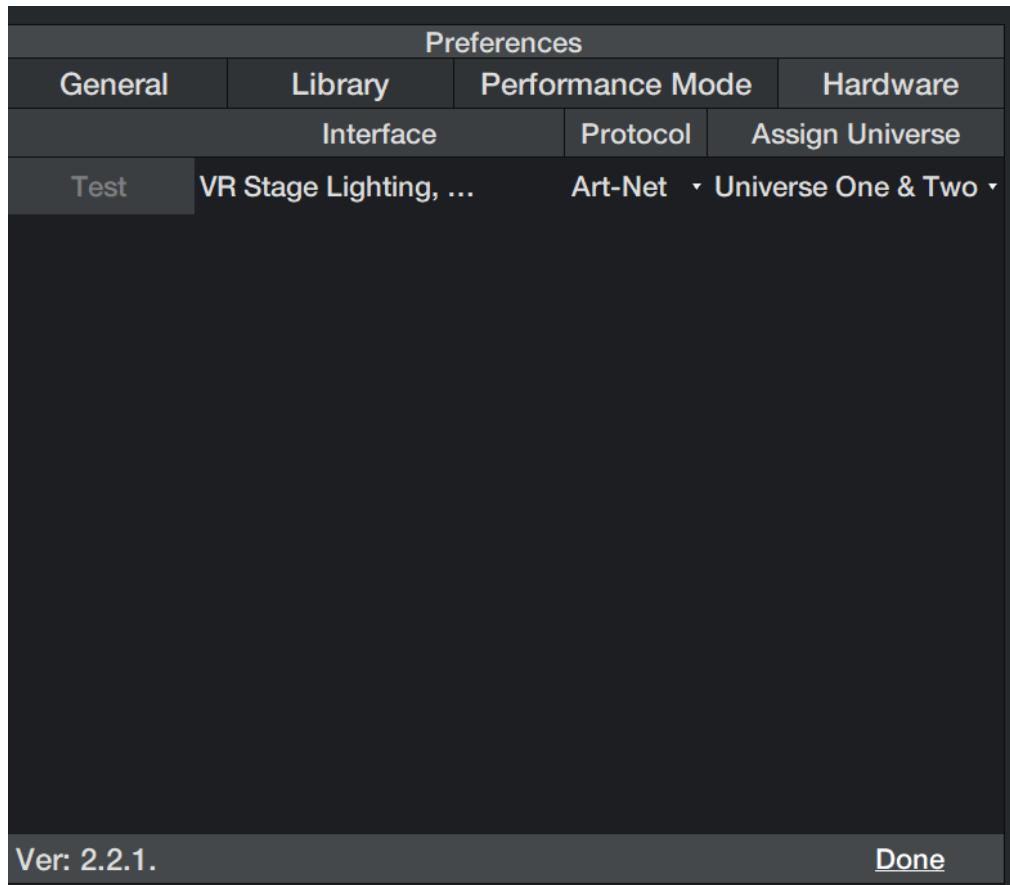


(all 13 Channel fixtures in this image are Pico Beam 60s, the first few are just renamed for organization purposes as this is based off the Orion Venue).

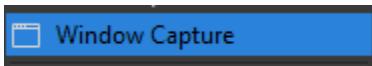
-Due to these limitations, it is highly recommended to be efficient with how you map your fixtures. Using the Invert Pan and Invert Tilt functionalities on the VRSL fixtures in Unity to create mirrored setups are one of the best ways to minimize the number of unique fixtures needed.

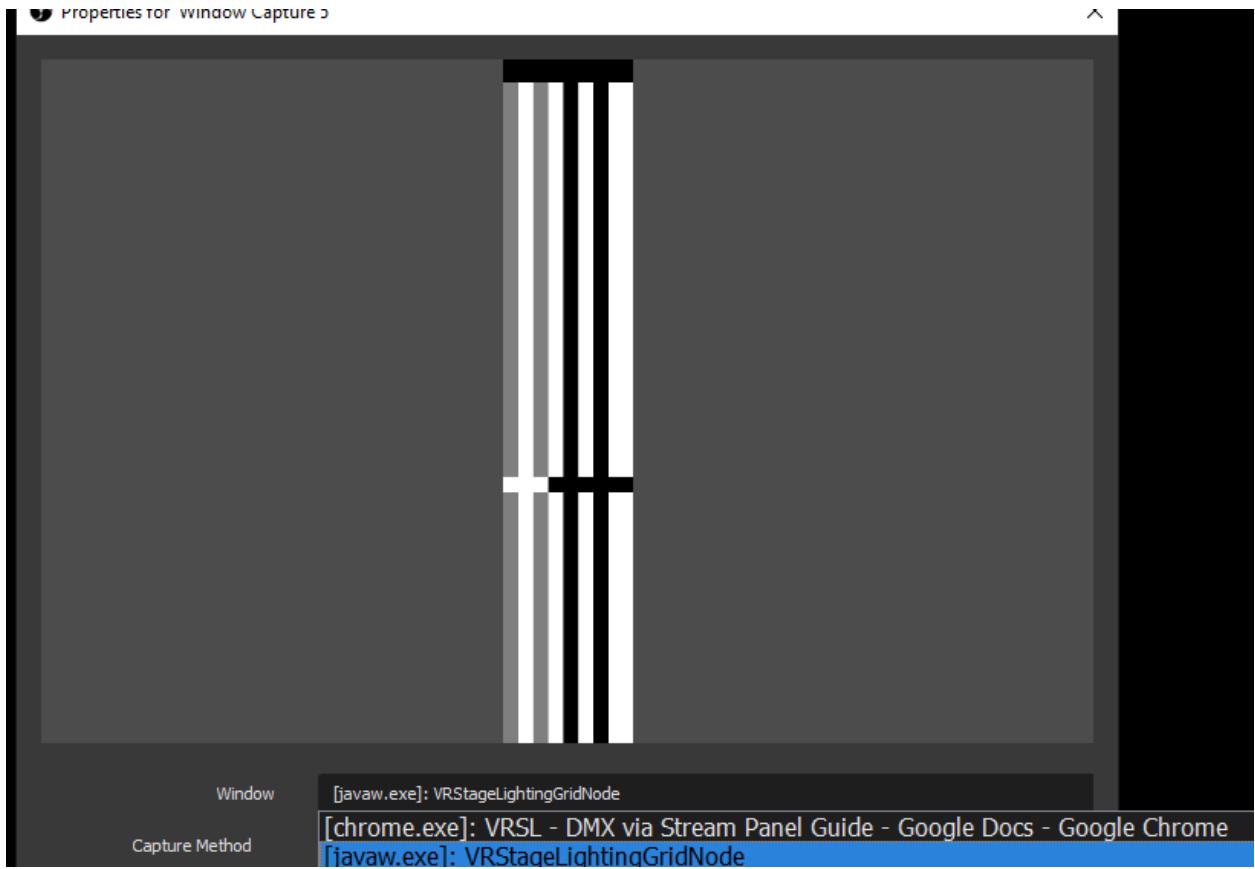
6. Open up “VRStageLightingGridNode.exe” software that was purchased. This software is an ArtNet Node that is detectable by software/hardware that can output ArtNet. In this example, we will show how to connect it to SoundSwitch. For other software/more information, please refer to the documentation that came with the software.

7. The default settings should have the node searching for ArtNet Devices on the loopback/localhost IP address. As such, SoundSwitch should be able to detect it and in SoundSwitch’s settings, it should appear as such under Preferences>Hardware:



8. After going into Edit mode on a track or Performance mode on one of your venues in SoundSwitch, you should see the grid node light up with whatever SoundSwitch is outputting, this means the connection has been established. If you are having trouble getting connected, please refer to the documentation provided with the node itself to configure/debug.
9. Open OBS and create a new Scene with a 16:9 aspect ratio. Add a new window capture source and set “VRStageLightingGridNode.exe” as the source.





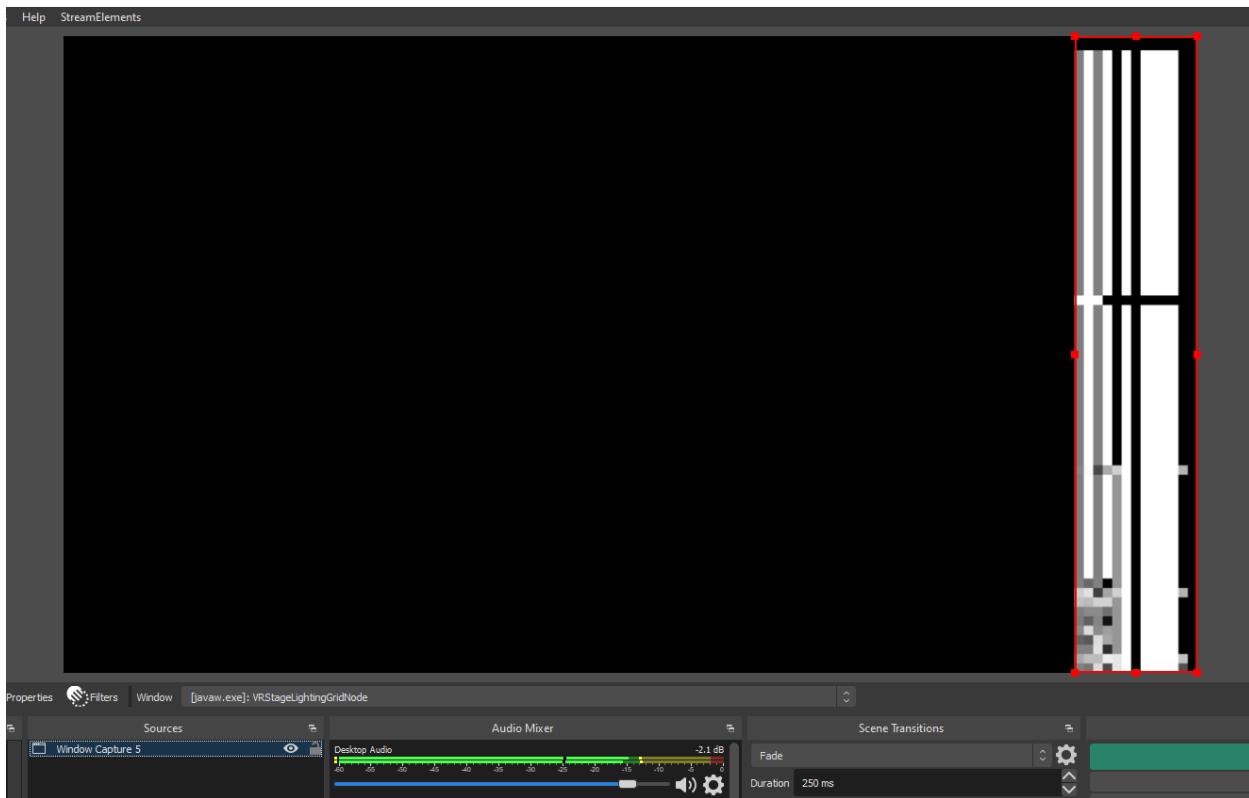
If the full grid isn't showing, set the capture method to this:

Capture Method

Windows Graphics Capture (Windows 10 1903 and up)

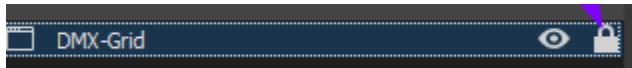
If needed, resize the grid to fit the entire height of the scene. The resolution of the grid is 208 x 1080px, so ensure that the ratio of the scaling isn't changed (I.E. the width and height must scale evenly.) No scaling should be required if the scene is already 1920x1080, however, the screen the grid is running on could influence the size OBS detects from the source.

(\*NOTE\*: Due to limitations, the grid will pause updates while minimized, so ensure that the grid is not minimized. The grid can work while behind another window however.)

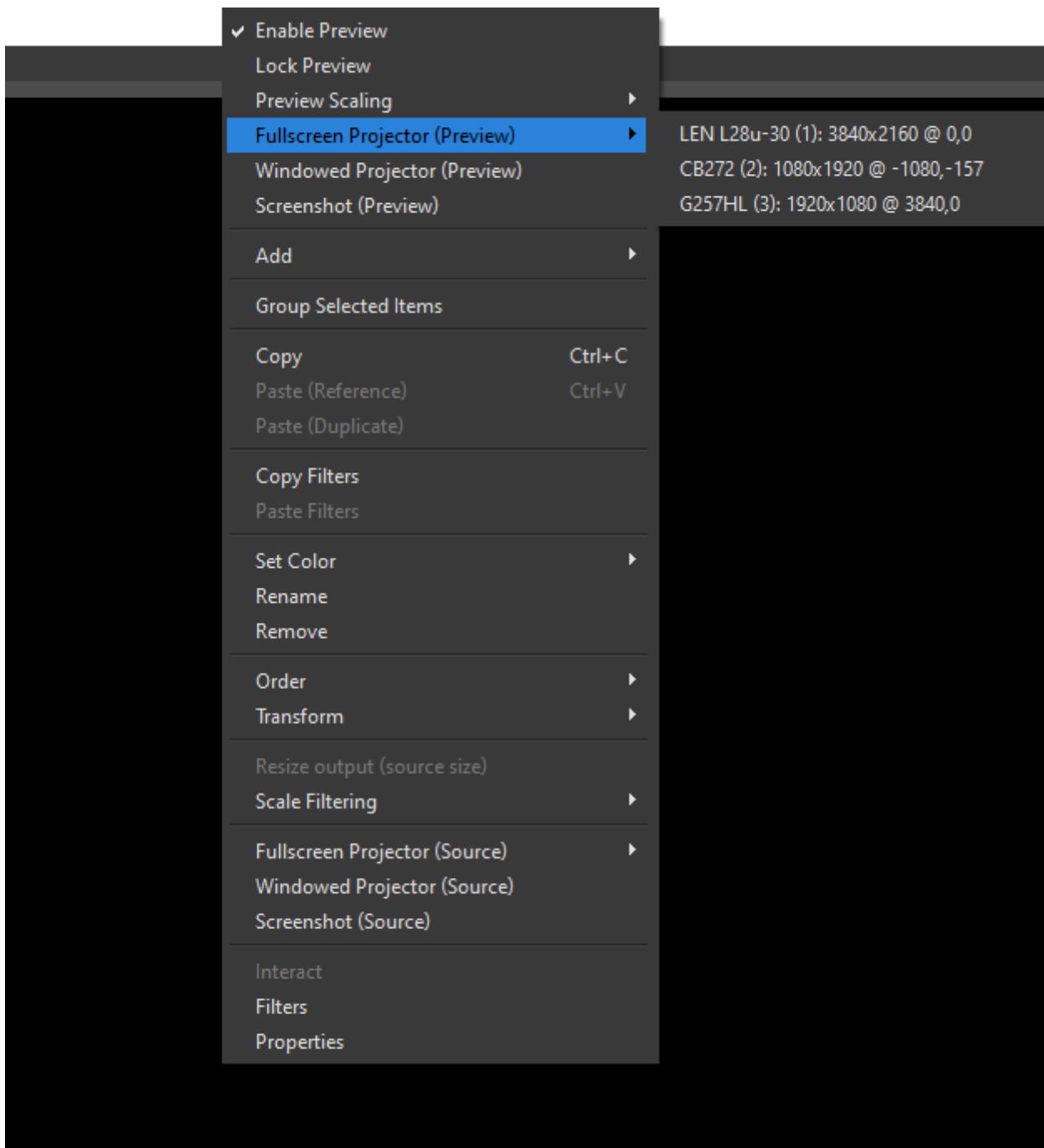


Add the source all the way to the right side of the scene as shown above.

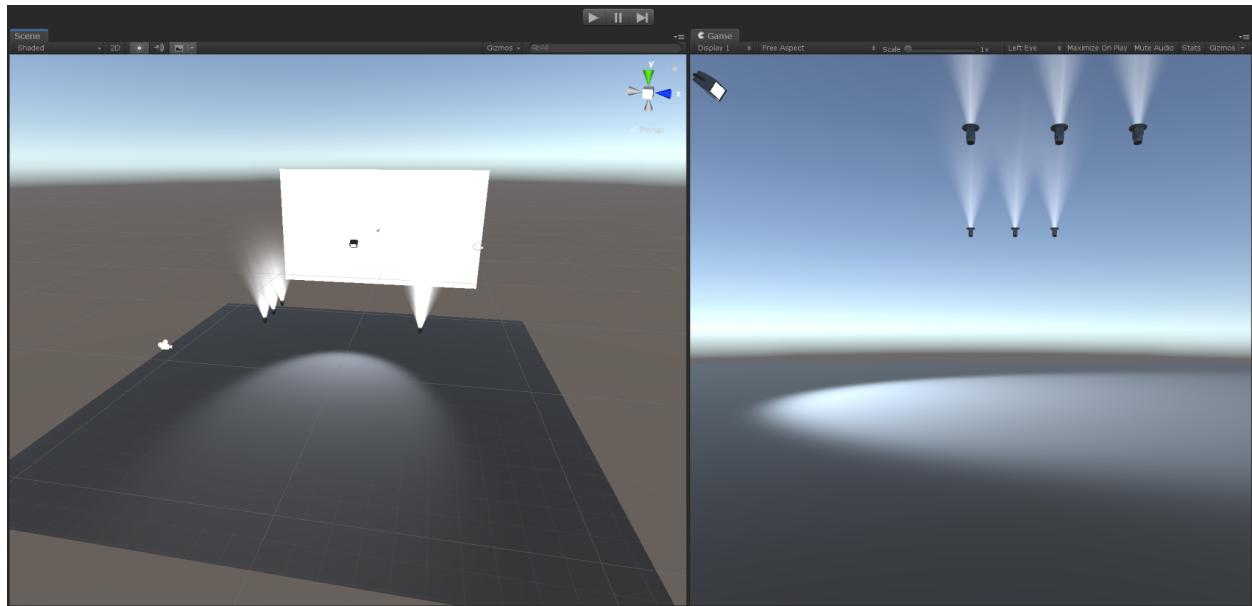
10. Lock the source and ensure any other sources you intended to place go under this source (I.E. Ensure the grid is ever blocked in the scene or the lights will not be able to read the data properly). Rename it for organization purposes.



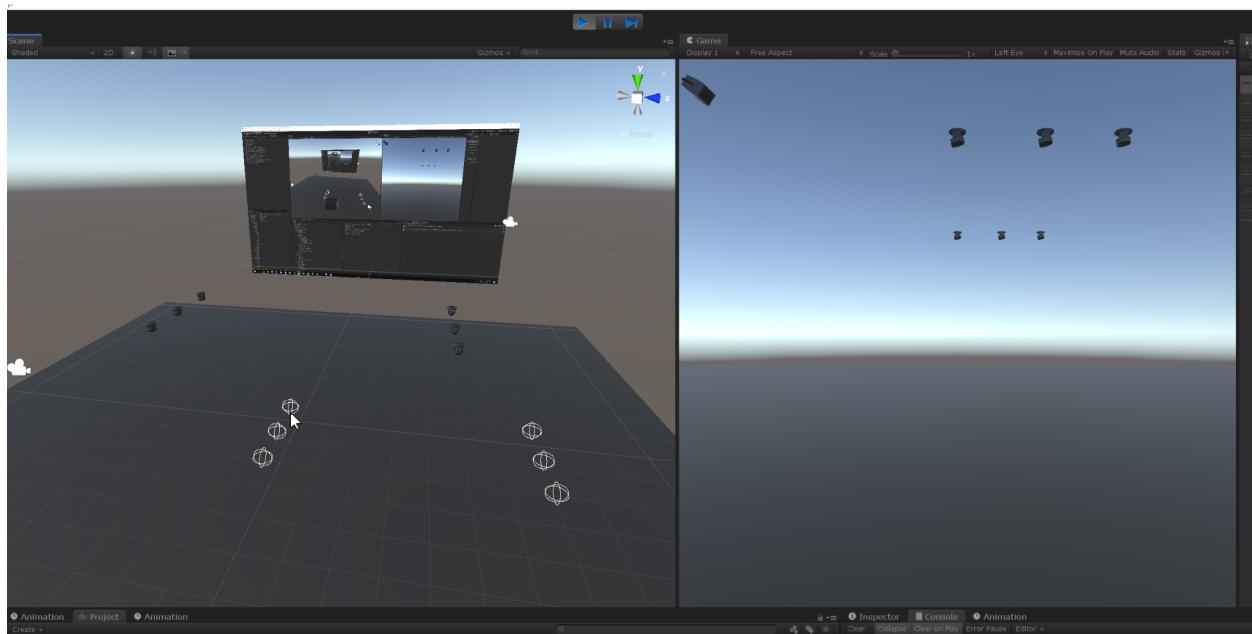
11. Bring the scene into full screen preview on a separate monitor. In OBS, this setting can be found here after right clicking on the scene itself:



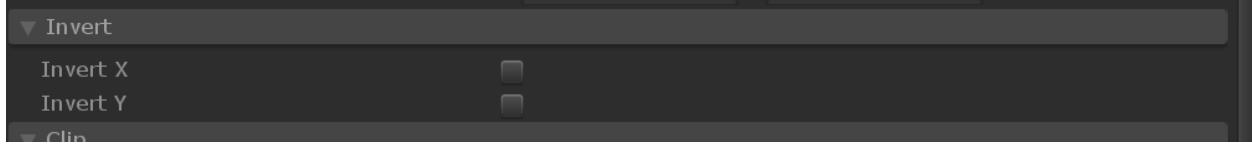
12. Go back to unity, and ensure that bot the game tab and scene tab are visible at the same time in a similar manner as shown below (this is only to ensure that uDesktop will update properly):



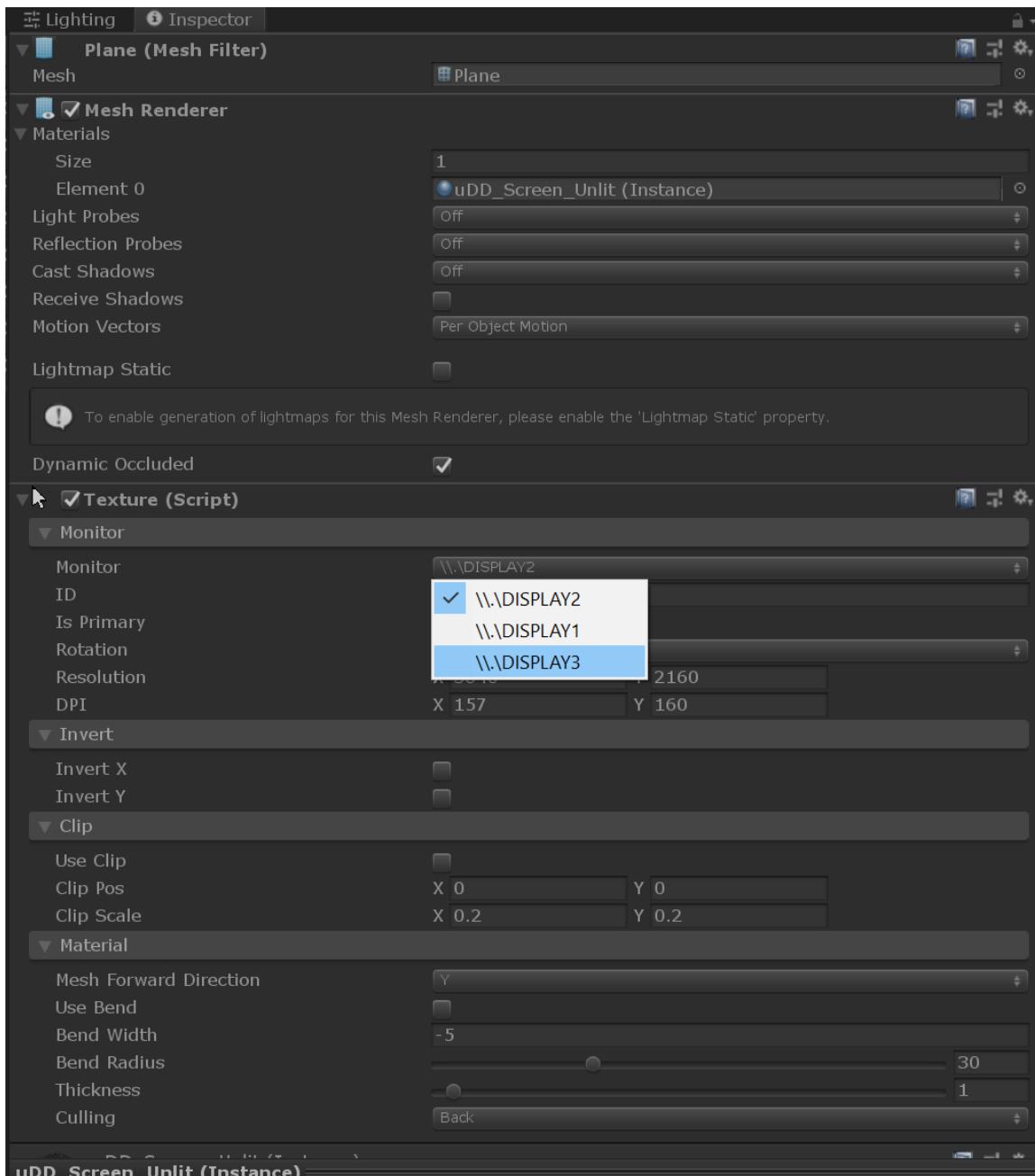
13. Hit Play. You should see your own screen pop up on the screen prefab and all of the lights going out.



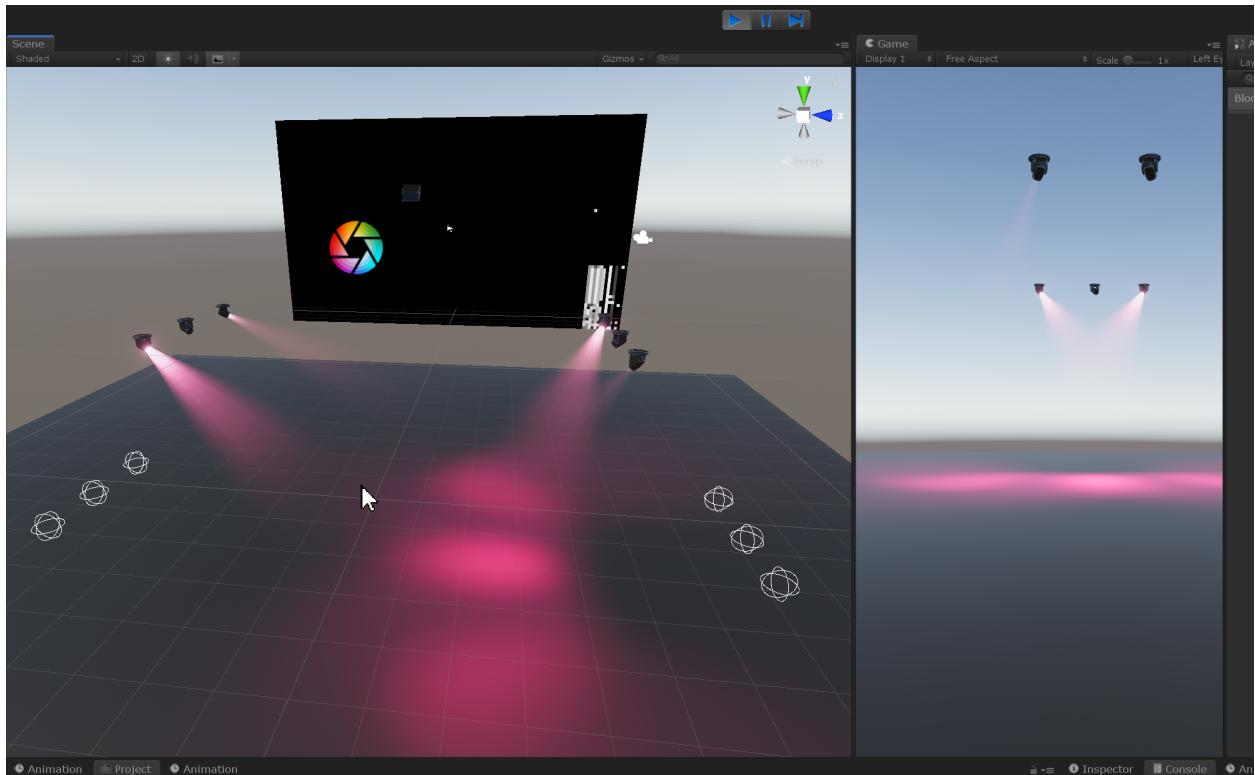
13.5: If the image is in the wrong orientation, click on the screen mesh object and go to the “Texture” script and hit one of these options:



14. Go to the “Texture” script on the screen mesh and under the “Monitor” section, select the monitor that is currently full screen previewing your OBS scene.



15. You should now see your OBS preview inside of unity on the screen. The lights will react accordingly depending on what the last DMX output was. In this example, I have opened a project I have already created in SoundSwitch prior to this tutorial and am somewhere in the middle of it, so what you see may vary.



16. **Congrats**, you are ready to go! From here, you can start working on your project in SoundSwitch or start creating cues in whatever DMX compatible software/hardware of your choice! This method enables you to see the changes in unity in realtime and how it would reflect inside of VRChat.

If you wish to change some settings about the fixtures at runtime, such as sector or pan/tilt inversion, go ahead and make the changes to the settings you want on your fixture in the Udon inspector, then hit this button:

Note: The changes made after hitting this button won't save once you exit play mode. You'll have to make the changes when not in play mode, then re-enter playmode once you want to save them.

# (DMX VIA Stream Panel through VRChat (Streaming))

Note: Due to limitations of this system, this system cannot stream to VRChat without some level of latency. This is due to the fact that the system relies on the VRChat stream panel to get the data into VRChat. This guide will show you how to set up streaming this into VRChat as well as things to do to compensate for the latency.

1. Read the guide “DMX VIA Stream Panel through Editor” and make sure you have a decent understanding of how that system works as this guide builds off of that.
2. Set up your stream based on the scene created from the previous guide. Ensure that the grid is *never* covered or blocked in any way.
3. It is recommended to set your output settings to 720p 30fps instead of 1080p or anything higher as VRChat video players will automatically downsize any stream or video to 720p 30fps. This is so that you can increase the bitrate of the video and improve the quality of the responsiveness to the lights.
4. Technically, from here, you can stream to twitch or other streaming service of your choice as normal and load the stream in a VRS<sup>L</sup> compatible world of your choice, however, from here are some optional but recommended tips to improve the experience:
  - Use a software that can change according to your DJ’s software. An example of this is SoundSwitch, which allows DJs to prescribe light shows per song, and will play the correct light show when that song is loaded into the proper deck. Features such as autoloops for non-scripted songs are also available in SoundSwitch. Check out SoundSwitch here!: <https://www.soundswitch.com/>
  - If you are wanting to do lighting the traditional way with software such as grandMA2, you will want to use a “Dummy Rig” of the venue you are performing at.

- A “Dummy Rig” should be a simple unity scene with the light prefabs set up with their proper settings intact in their proper positions, essentially “mimicking” the actual rig that would be inside of VRchat. Dummy props can also be used here. This Dummy rig is essentially designed to allow lighting engineers to see what the lights would look like in VRChat in real time. The Dummy Rig should be provided by the venue owner via a Unity Package File. More info about making Dummy Rigs is available in the next guide.
- For using a Dummy Rig as a lighting engineer, open Unity and have the Dummy Rig set up with the udesktop prefab set up and run the scene and have a preview of the stream on a separate monitor, just like the previous guide.
- You will be able to control the lights in real time, but you will also be able to stream your video output to VRChat at the same time. This way, you will know how the lights are behaving in realtime and can adjust as such.
- The ideal, professional setup for this would be using some type of stream relay system. Where the DJ would send their audio to their lighting team via an OBS Relay, and the lighting team would then send their lighting data and the DJ’s stream to a VJ/Stream hoster, and the VJ would then output the final stream with their graphics attached. This would allow splitting the workload and mimic a similar pipeline to a professional concert.

# **(DMX VIA Stream Panel through VRChat (world building))**

1. Read the guide “DMX VIA Stream Panel through Editor” and make sure you have a decent understanding of how that system works as this guide builds off of that.
2. Remove the “VRSL-DMX-uDesktopDuplicationReader” prefab in the scene and replace it with the “VRSL-DMX-USharpVideoReader” prefab.
3. Upload the world to VRchat as a test world. Go into the test world and ensure that the player can play videos and streams.
4. Set up your world how you would want it, ensuring all the props you need are in the right place and such.

## **5. DUMMY RIGS**

- As stated before in the previous guide, a Dummy Rig is a mock up version of your VRSL setup in your world that you can distribute to your performers.
- The Dummy Rig only needs the prefabs that were placed in the original world, with similar lighting and post processing so that the lighting engineers have an idea of how the lighting is set up.
- Placing fake props out of boxes or other simple geometry will help bring more accuracy for the engineers of how the system is set up in VRChat.
- Ensure that you export your Dummy Rig as a separate scene in a unity package file and can be easily accessible to your venue performers.

#### **NOTE ABOUT THESE GUIDES:**

These are more simplified guides, and more direct tutorials and guides will be released as this system matures. The system itself is still in heavy work in progress, and there will be available communities and discords to ask questions and the like around. Please be patient and thank you for using/supporting VRSL!