

Heyo!

Here's a manual on how to run the broadcaster system. What is the broadcaster system?

[Overview:](#)

[Running the system demo without any inputs \(NBA Celtics Demo\):](#)

[Setting up the non-live demo:](#)

[Intro to interacting with the system:](#)

[Moving a screen:](#)

[Selecting a screen to go to the main screen:](#)

[Selecting a clip from the camera-feeds:](#)

[Moving around graphics on the main screen:](#)

[Making your own graphics elements:](#)

[Using the twitter feed:](#)

[Moving the in-studio camera:](#)

[Running an ML-Talks demo \(live demo with live inputs\):](#)

[Connecting/Troubleshooting the Cameras:](#)

[Setting up the cameras in the VR studio:](#)

[Using the in-built browser:](#)

[Flying around the studio:](#)

[Listening to the audio from the live-audio feed:](#)

[Recording a video of what's happening:](#)

Overview:

When a major event occurs, one does not necessarily have the resources or time to bring in a production truck. During the demonstrations in Tahrir Square, protestors captured photos and videos on smartphones to document and share experiences instantly, creating pop-up amateur broadcasts.

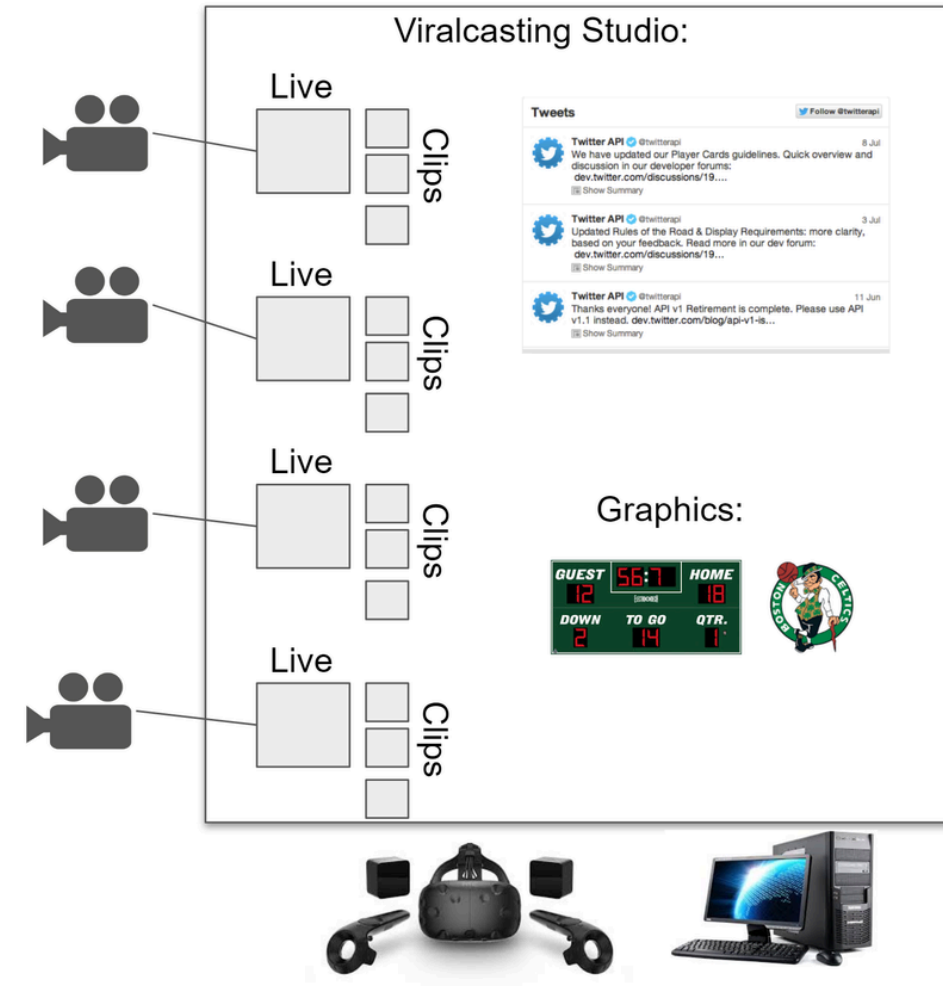
If one wanted to become their own CNN, including cut-aways, multi-camera production, and info-graphics, one would need a production truck and a highly experienced crew. Spatial technologies can help to reduce the cost-of-entry for content creators. In this chapter, I will discuss producing 2D live content without a production truck. This enables new kinds of content from creators we would not normally view. This can also facilitate new kinds of relationships between consumers, prosumers, and the big networks.

The main advantages of the Viralcasting studio over a broadcast truck are:

- customizability - multiple screens that you can position anywhere

- directly manipulate graphics on screen -
- collaborative - clips can be shared
- social media integration

The viralcasting studio is implemented as an application on the Vive VR headset. This headset has two controllers and a headset that are tracked in 3D with sub-mm accuracy. The necessary equipment for the system are a VR headset and a PC. The system is compatible with saved videos, and also works with live-video streams via SDI. This is demonstrated in this high level diagram.



This culminates in the VR studio visible below:

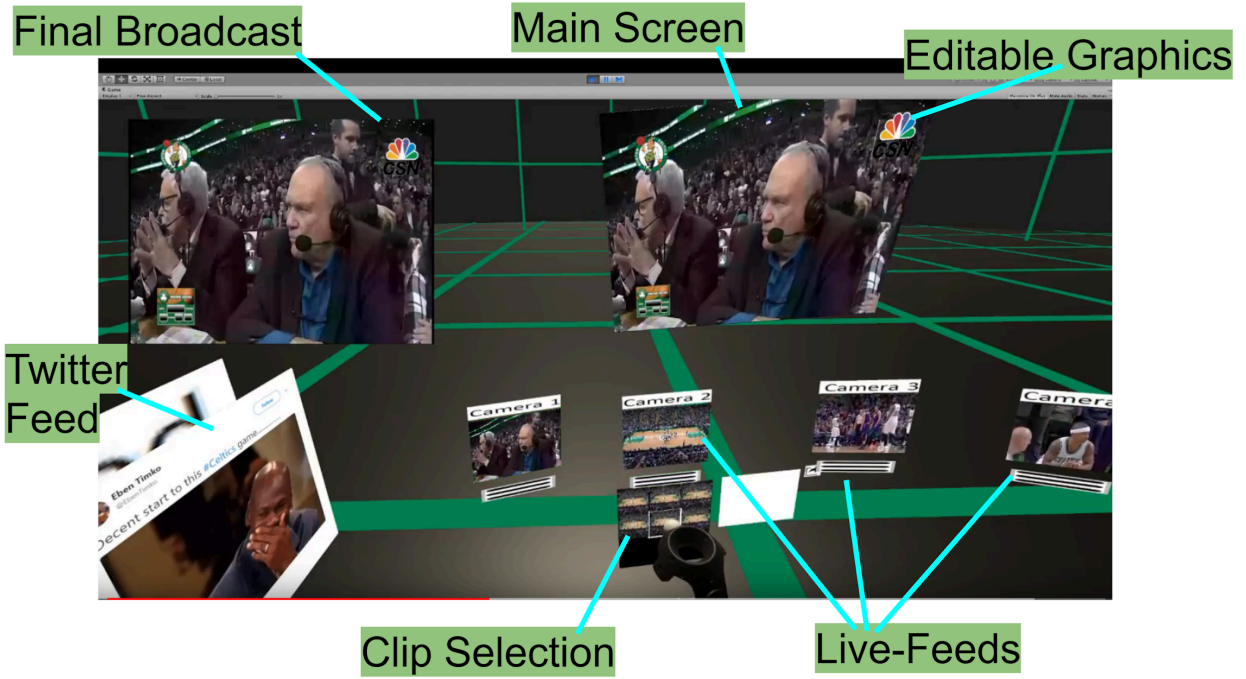


Diagram showing the main components of the user's perspective while using the broadcasting system.

Running the system demo without any inputs (NBA Celtics Demo):

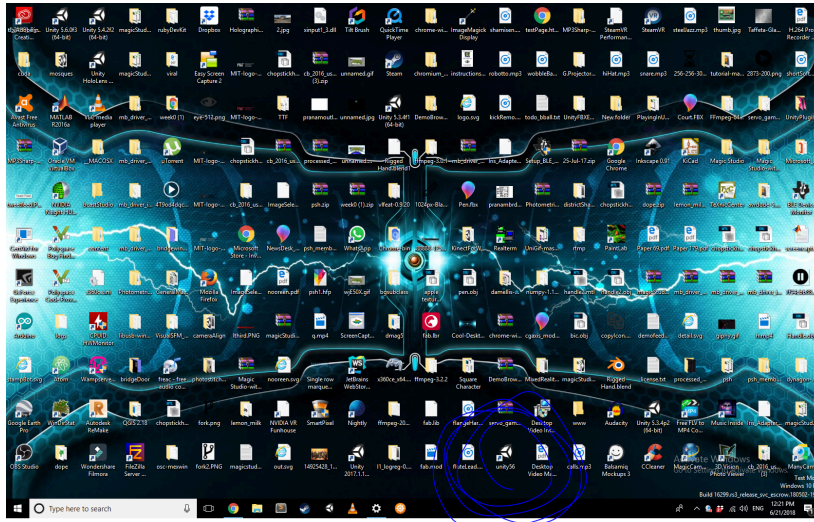
Setting up the non-live demo:

If you want to run a demo to show off how the system works, and you don't want to go through the process of hooking up live-inputs, there is a pre-baked demo with video we took from a Celtics game (thanks NBC!).

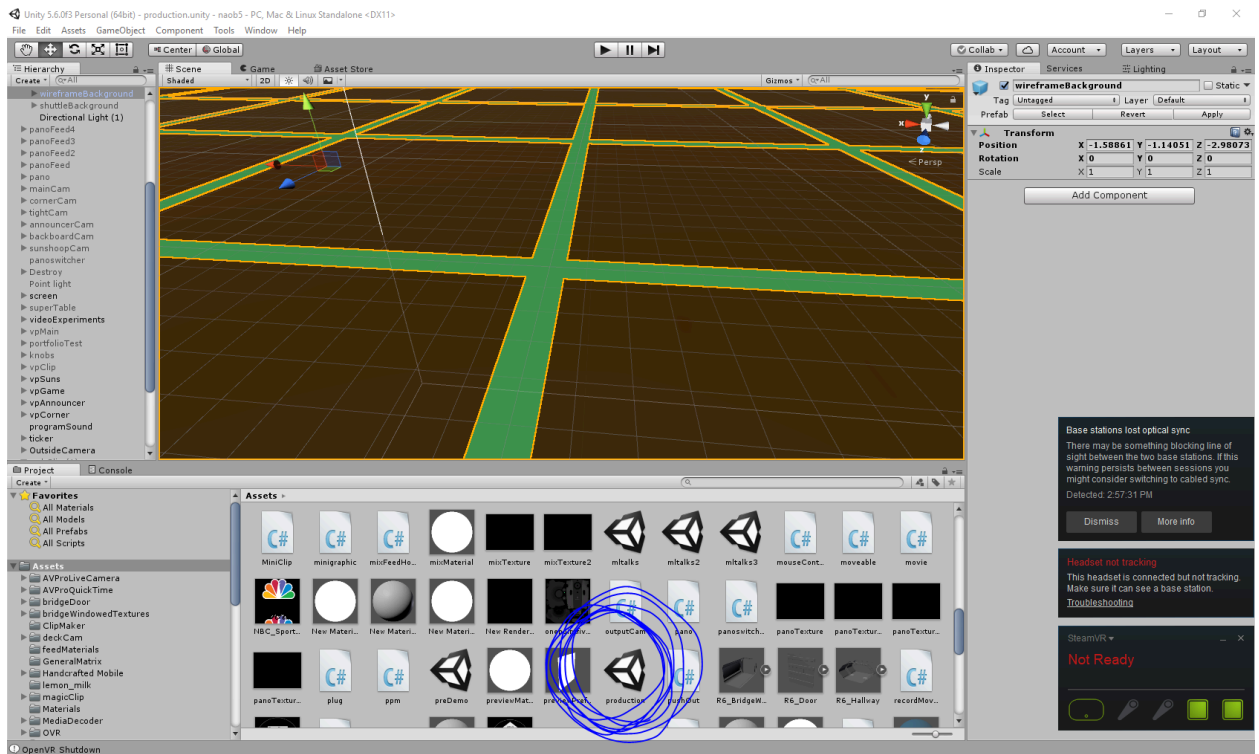
To run this demo, you'll need to run it on the broadcasting computer. This is a PC with a Vive attached to it placed on a movable tray-table-thingy (black in color).

- Turn on the PC:
 - There is a power button on the top left of the PC
 - Click on the "hisham" account, put in the password: "aperture"
- Set up the Vive / make sure it's working:
 - There are guides online, but you'll want to open the "steamVR" app and run room setup. I suggest running a "standing only" setup in front of the system
 - https://www.vive.com/us/support/vive/category_howto/setting-up-for-the-first-time.html (setting up a vive for the first time, you probably won't need to do this unless you need to reinstall steamvr for some reason)
 - Make sure that both controllers have power and that they are both visible and tracking
- Open Unity 5.6
 - Why do I need to use this version of Unity? The project would probably run on newer versions, but it may require Unity to recompile some things. I doubt

anything would break, but whooooo knows what the future holds



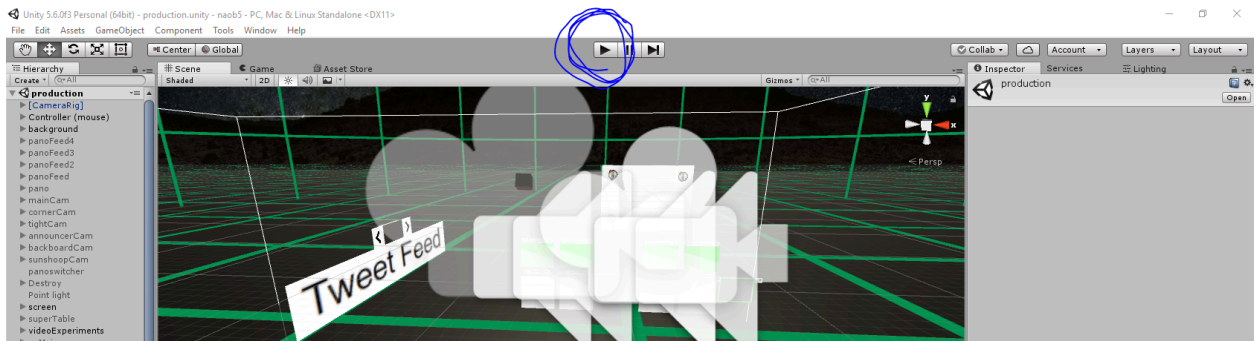
- Open up the NAOB5 project. This is located in: D:\UnityProjects\naob5
- In the project explorer, double click on the Unity Scene called: **“production”**:



- This will load the scene with basketball things:



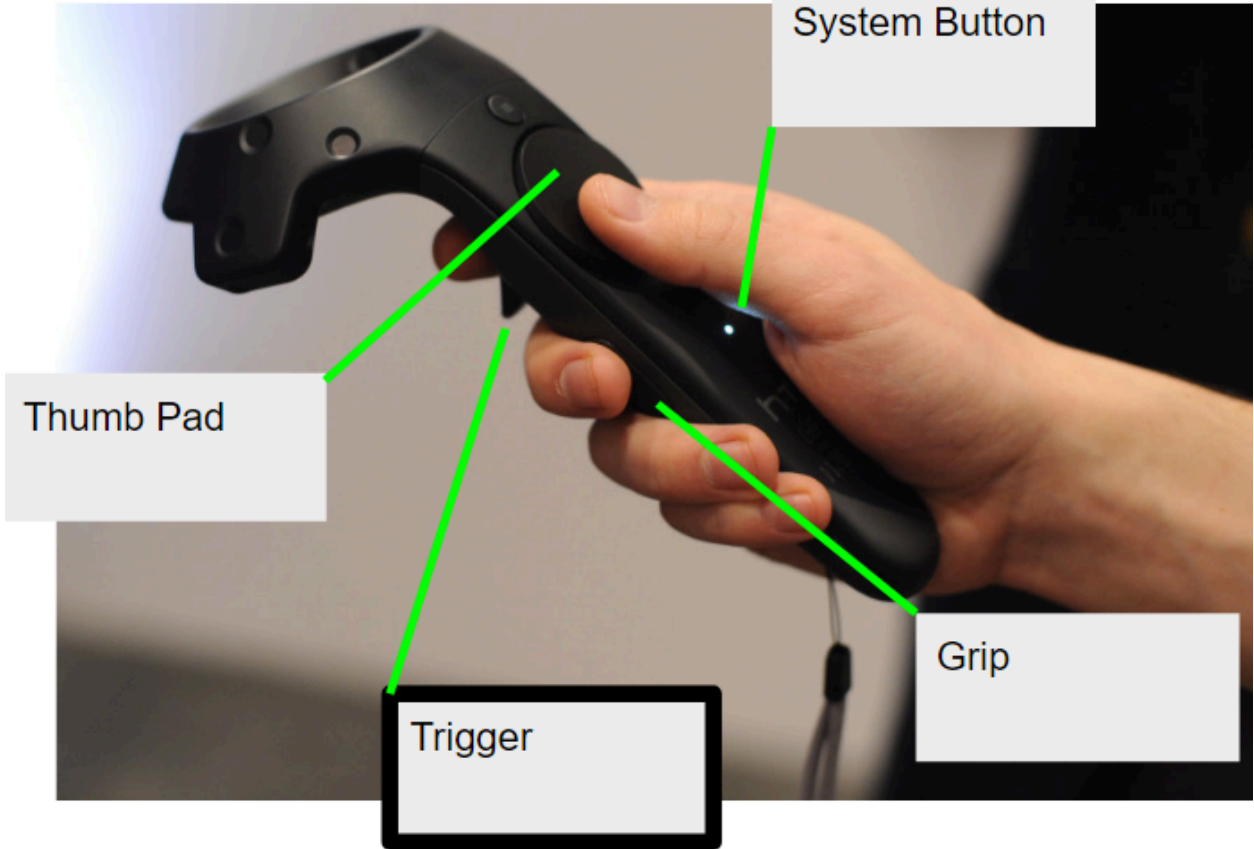
- Hit the play button!



- The application will start. On the monitor, you'll see a view of the output (white at first until a user selects a feed). Inside the headset, you'll be inside the virtual studio.

Intro to interacting with the system:

The rest of the instructions are on how to operate the system so you can utilize all the features. Before we begin, a guide to the vive controllers and how they map to the demo:

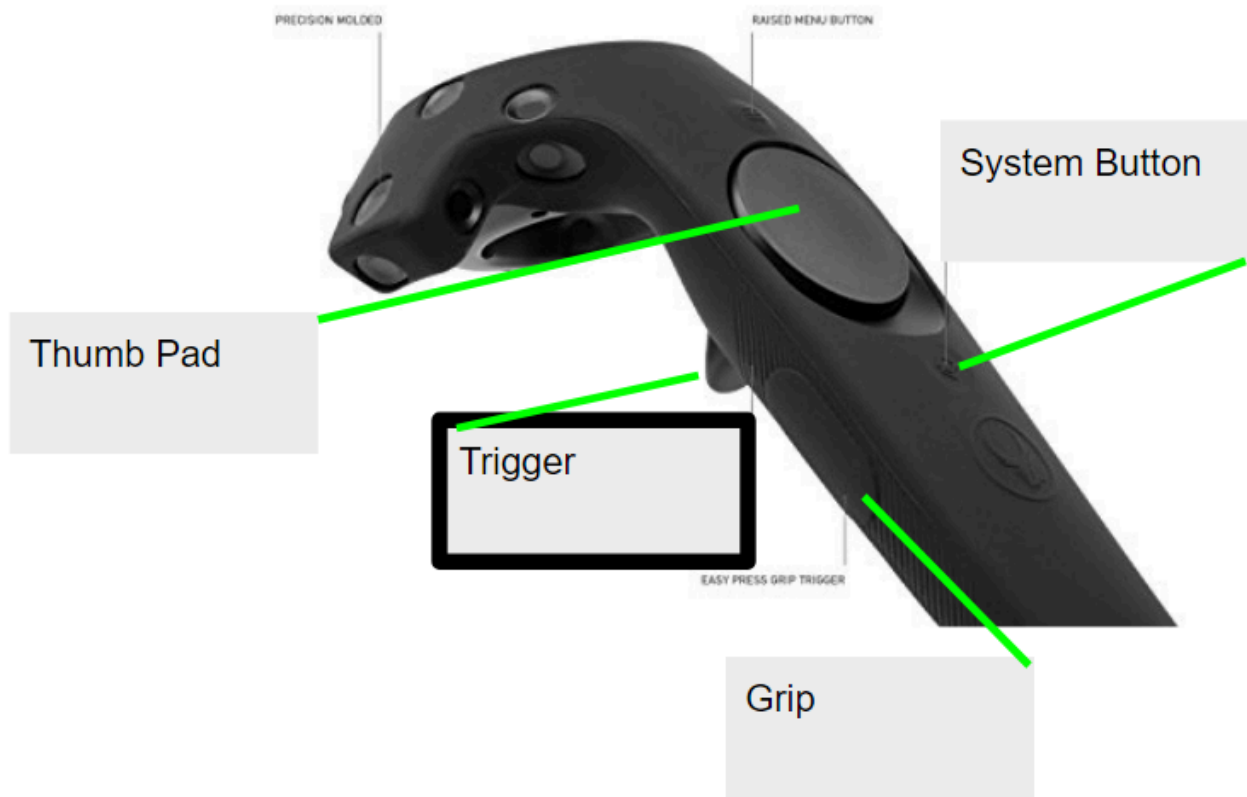


System Button

Thumb Pad

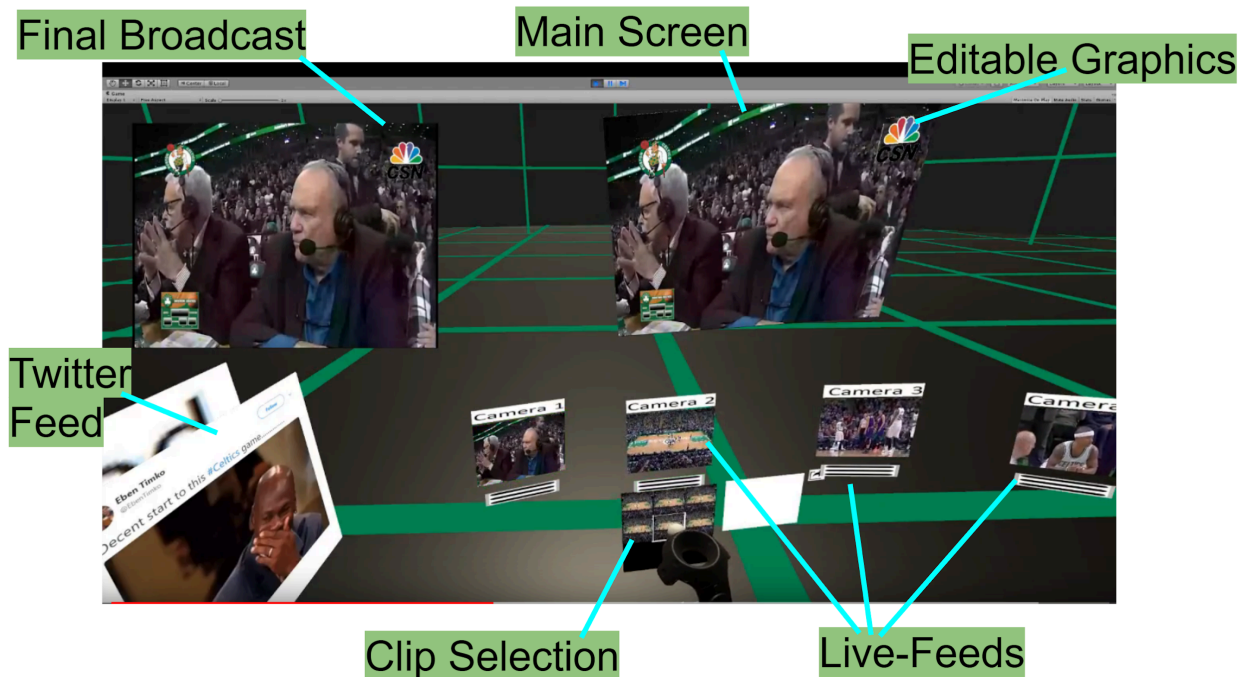
Trigger

Grip



Here's what it looks like on the monitor. Most of the screen is a 3rd person view of the studio. The top left part of the screen is a view of the 2D broadcast you are creating.

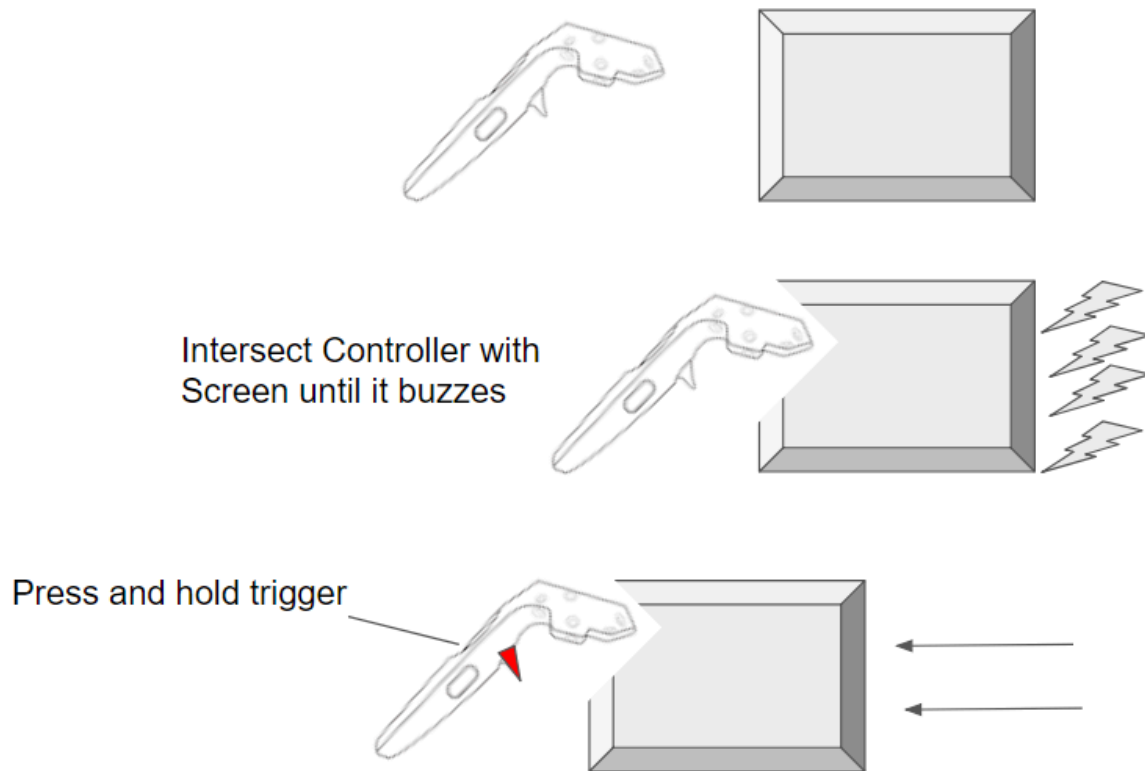
The main element of the system is the screen. The screen can be grabbed and moved around. It contains anything from live-cameras, feeds, clips, and tweets. There is also a "main screen", which is larger. Whatever shows on this main screen shows up in the final broadcast. You can send smaller screens to the main screen by pressing the touchpad when the controller intersects with the screen. The screen will blink to indicate that it's currently being shown on the main screen. Graphics elements also float above the main screen.



Moving a screen:

There are two main interaction paradigms. The first is grabbing. Things can be grabbed by intersecting your controller with the object. The controller will then buzz to let you know that it can be moved around. You can then pull and hold the trigger to grab it. The object will now follow your hand. You can drop it by releasing the trigger. The dropped object will maintain its position in 3D space. You can only grab one object per hand.

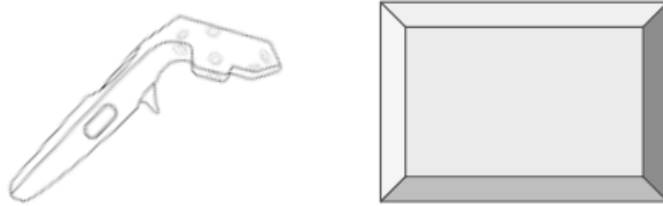
Moving a screen:



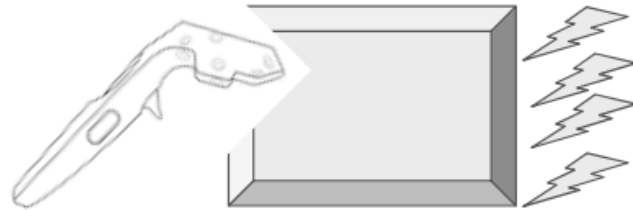
Selecting a screen to go to the main screen:

The other interaction paradigm is to send a screen to the "main screen". You do this by intersecting the controller with the screen, then briefly pressing on the touchpad. The screen will then be sent to the main screen. The screen will blink green to indicate that it is currently being sent to the main screen.

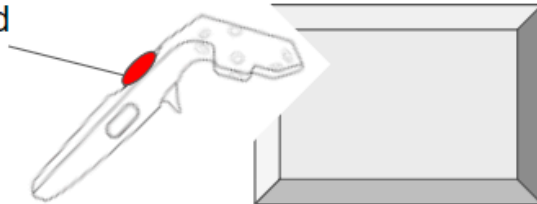
Sending a screen to the main-screen:



Intersect Controller with
Screen until it buzzes



Press Touchpad

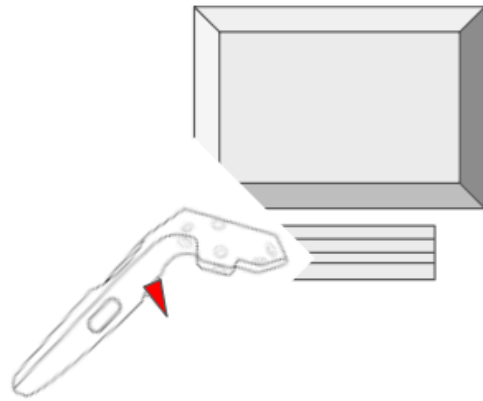


Selecting a clip from the camera-feeds:

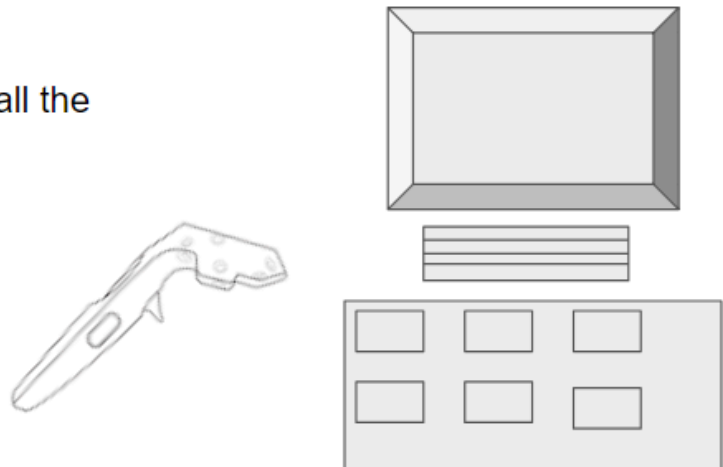
In the non-live demo (NBA demo), the system forms clips live. These clips can be viewed underneath every camera feed. These clips can be viewed, then sent to the main screen. They can also be duplicated into their own screens that can thus be placed into space.

Selecting a clip:

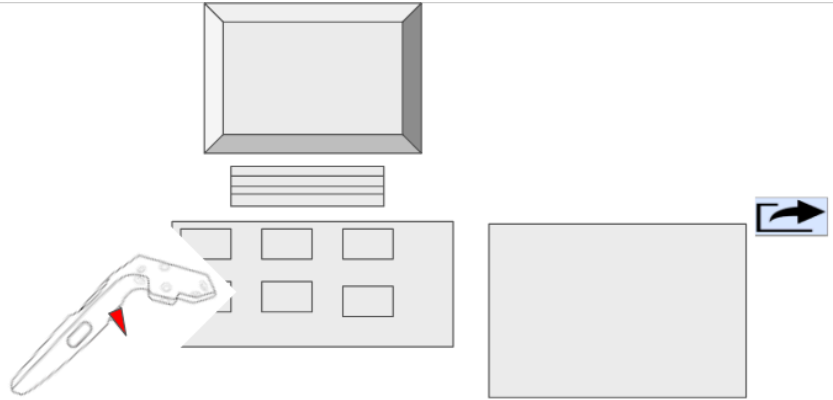
Intersect Controller with Burger underneath screen, then pull the trigger



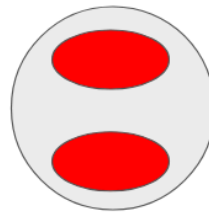
A menu will pop up with all the clips



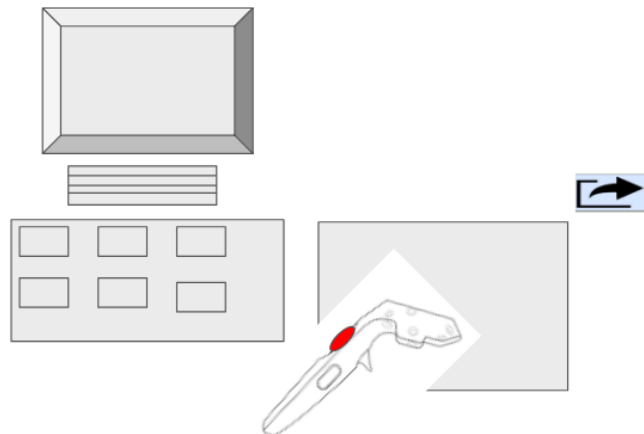
Selecting the clip with the trigger creates a new screen where you can view the clip.



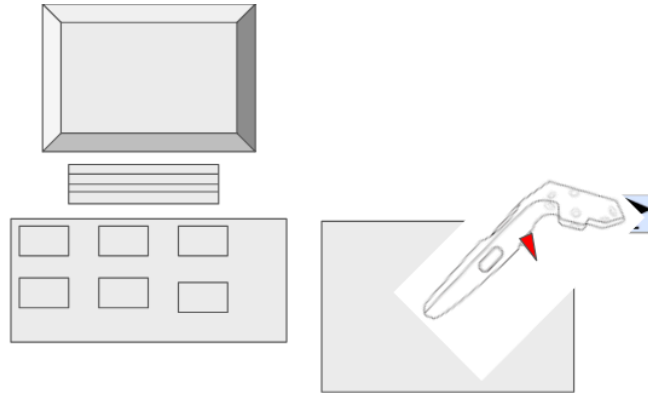
Pressing on the upper side or lower side of the touchpad scrolls through the menu when the controller is close enough to the menu



Pressing the touchpad on the new screen sends it to the main screen.



Clicking the share icon creates a duplicate screen of the clip that you can grab and move around.

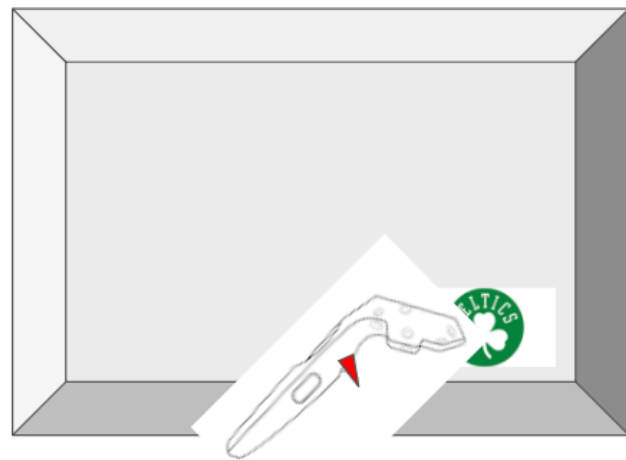


Moving around graphics on the main screen:

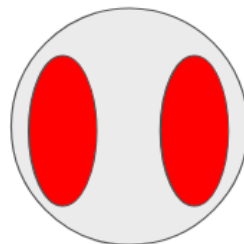
The mainscreen is the screen that is “broadcast”. It’s the one that appears at the front of the demo. The mainscreen can be grabbed and moved around. You can also move around the graphics on the main screen:

The main screen can be moved around by grabbing it from the back using the trigger. Grabbing graphic elements, like this celtics logo, from the front enables you to move the celtics logo around. If you grab the celtics logo and bring it towards you, away from the screen, it will lift off the screen like a sticker. You can put it back by placing it back in front of the screen (at a close enough distance).

Main Screen:



Pressing on the left side or right side of the touchpad scales the onscreen graphic

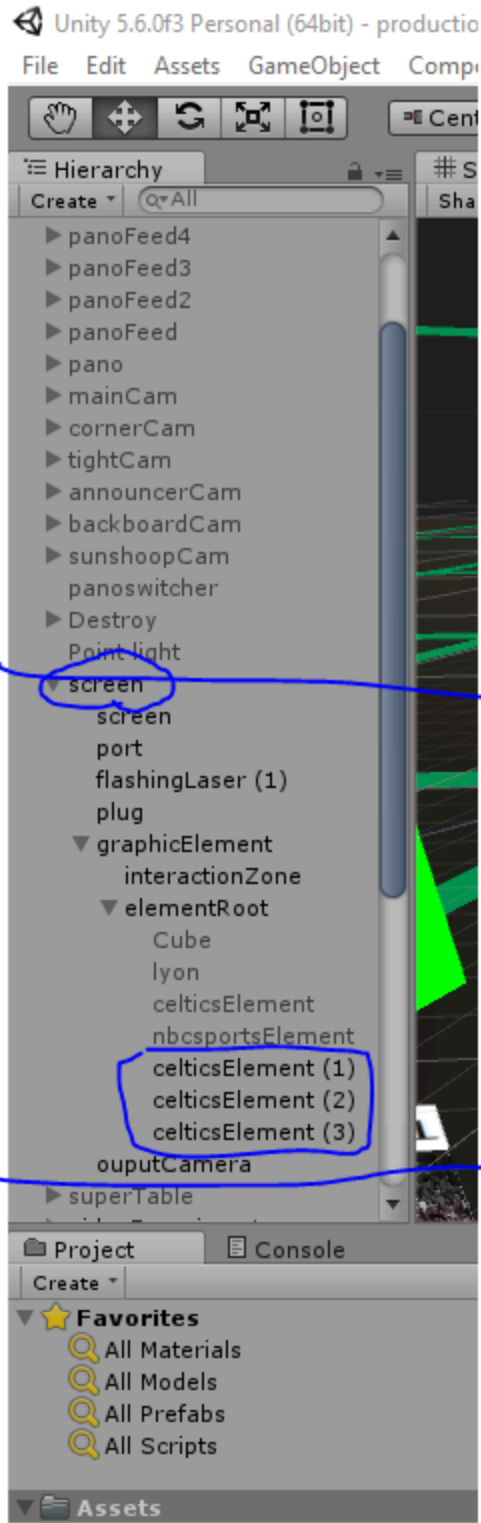


*if the graphic disappears for some reason, it may have gone behind the main screen. Go take a look, or push your hand behind the screen until you feel a buzzing in order to fish for it. Then grab it and put it back in the front.

*Moving graphics currently only works with the right hand. If it's not working, make sure you try it with both hands (controllers).

Making your own graphics elements:

While the app is not running (since any changes while the app is running will not stick), you can edit the scene heirarchy. Within the heirarchy, there is a "screen" element (which refers to the main-screen, i know bad naming). Undernead this is a section called graphic elements. Underneath that is where you can create / destroy graphic elements:



The graphic elements can be seen above as celticsElement(1), etc... Say you want to change a graphics element. Just click one of them, then hit ctrl+c to copy, and ctrl+v to paste. There will then be a new element that gets created. You can then drag and drop an image from the project explorer/assets onto this element to change its photo. You can change it's initial position by

dragging the arrows in 3D or changing the transform position in the inspector in the top right. Note that this will change the local coordinates. Make sure you don't change the shader on this (transparent cutout) unless you know what you're doing or are willing to explore a little with trial and error. Unlike a lot of things in unity, changing the materials of objects can lead to changes that last past when the game is stopped. This makes it work with the transparent layers.

Using the twitter feed:

We have a bunch of tweets that we collected that were sent out the night of the game. You can cycle through these tweets and put them on the main screen as well.

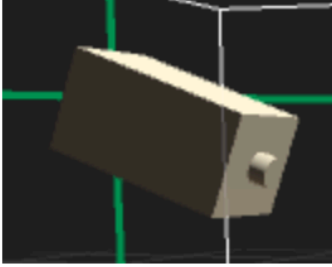
Tweet feed:



Pressing the thumb-pad on any of the tweets sends it to the main screen. Clicking (pulling the trigger) on the left or right buttons slide the tweets down on the carousel.

Moving the in-studio camera:

Moving the In-Studio camera:



The in-studio camera captures a view inside the VR studio. This is put on the main screen. When other people around you are watching the monitor, it is better for them to watch what's happening through the in-studio camera than through the first person view, as the first person camera is constantly moving.

You can change what the in-studio camera is pointing at by grabbing it. You grab the in-studio camera by intersecting the controller with the camera, then pulling and holding the trigger. The camera will then follow your hand until you drop it by releasing the trigger.

Running an ML-Talks demo (live demo with live inputs):

Studio 125 and Diginovations run the setup on most of the ML-Talks. These talks consist of 4 cameras (4x 3G-SDI), 1 graphics feed (1x 3G-SDI), and 1 XLR audio. Most of the cameras come in at 1080i 59.97hz. Setting up the demo will be involved, but once you've done it the first time, you'll be able to do it again.

First, make friends with the folks who will give you the camera feeds + audio feed.

Then, day-of, roll out the system to the area. The PC is installed with two Balackmagic SDI digitizers. One is the Decklink Mini Recorder, which can take in 1 input (either 3G-SDI, or HDMI). The other is Decklink Duo 2, which can take in up to 4 inputs (4x 3G-SDI). They should be set-up to work on the PC, but in the case that they are not, please refer to the manuals to fix things (I'll include a short troubleshooting section, but I don't have access to any SDI broadcast equipment right now, so I won't be able to take screenshots of the actual setup).

The PC also has a XLR-female to 3.5mm audio adapter plugged into the microphone jack (pink) on the motherboard's onboard soundcard. You can connect this to the male XLR cable that the folks at studio 125 will give you. Note, this adapter (or the connection) is not good and causes a buzz.



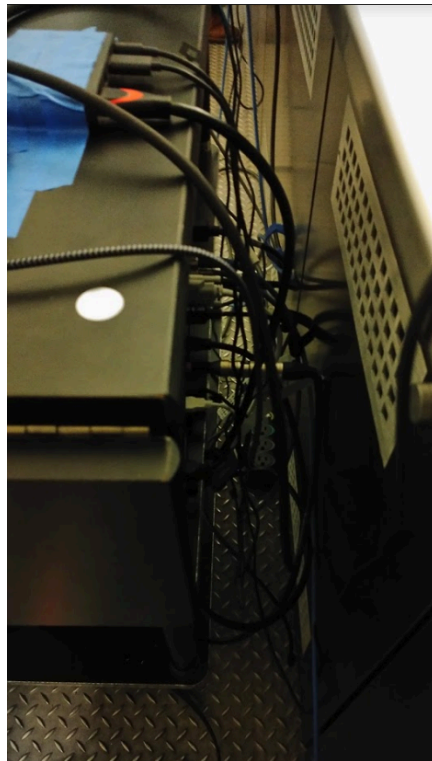
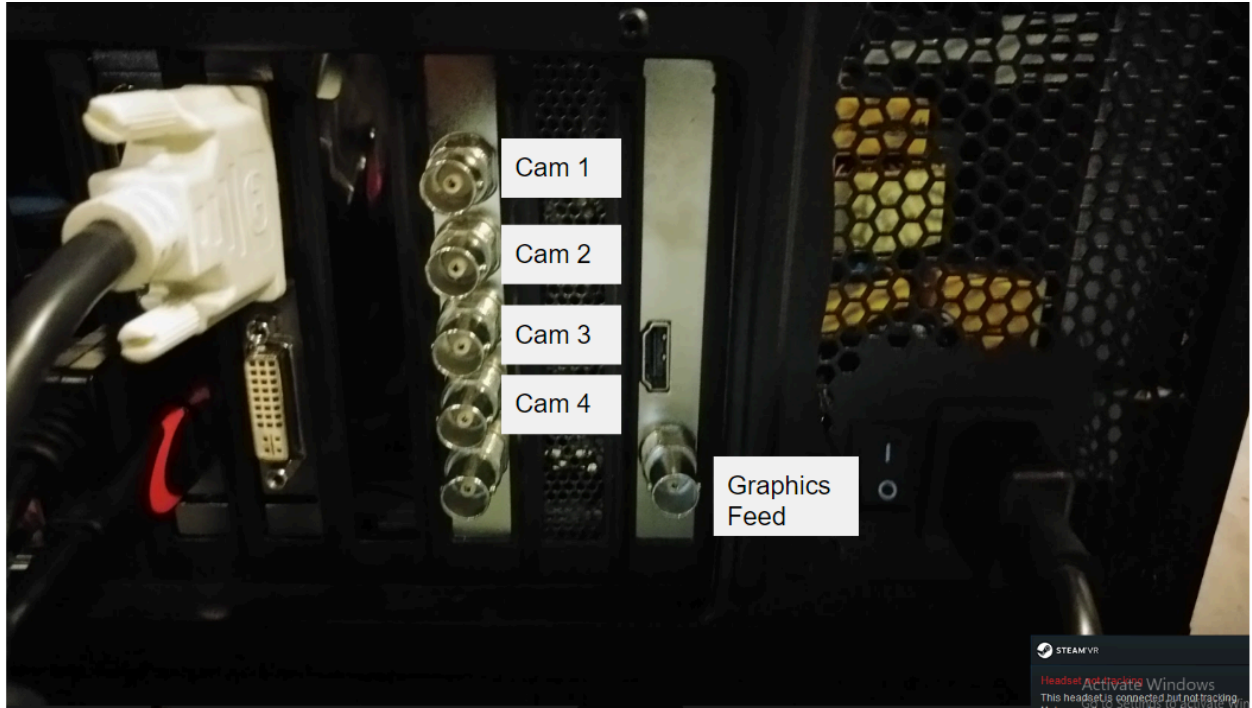
Connecting/Troubleshooting the Cameras:

There will be moments before you are ready with the whole setup that you will need to troubleshoot the cameras. The folks at Studio 125 want to make sure their equipment is all set for you, they are typically really nice like that.

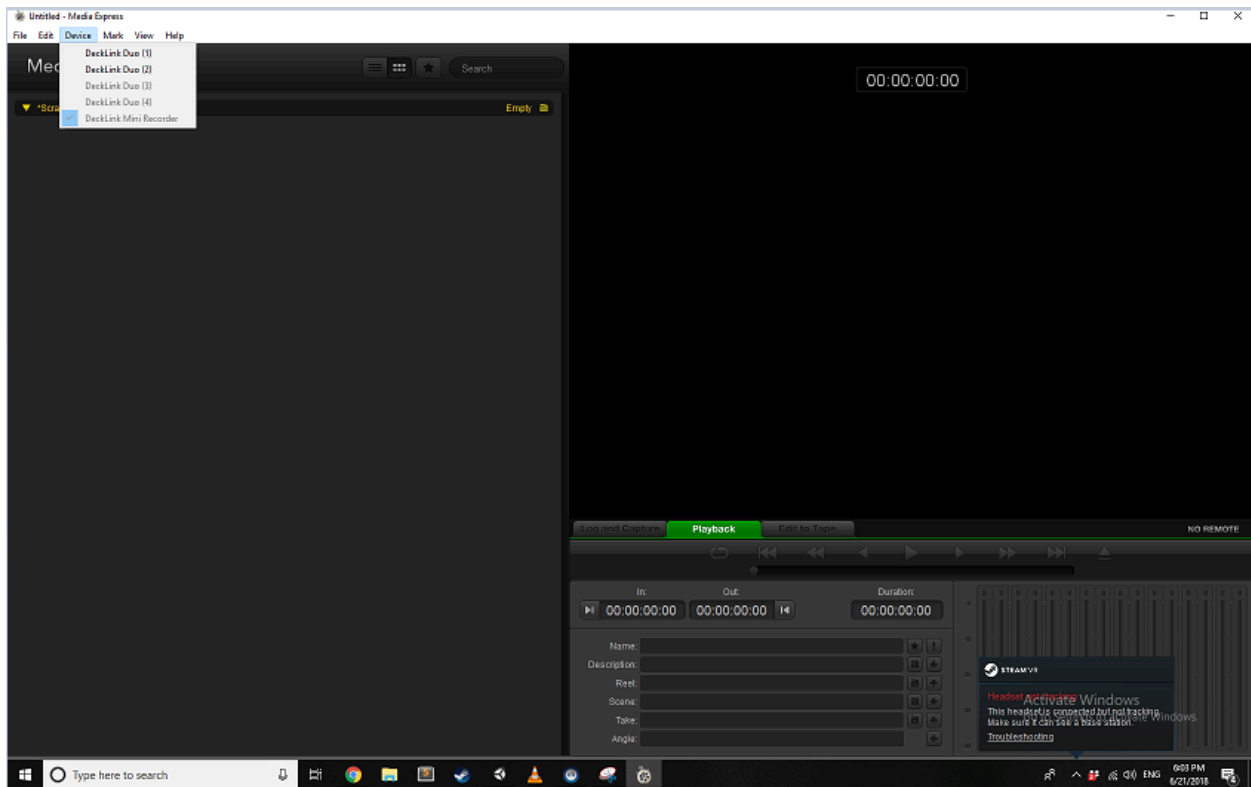
The first step is to connect the SDI cables to the back of the PC:



I usually use this configuration. I am pretty sure the bottom-most (closest to the motherboard) SDI connection on the multi-card is a reference signal, so you should plug into all the others. I could be wrong, but I'm fairly sure.

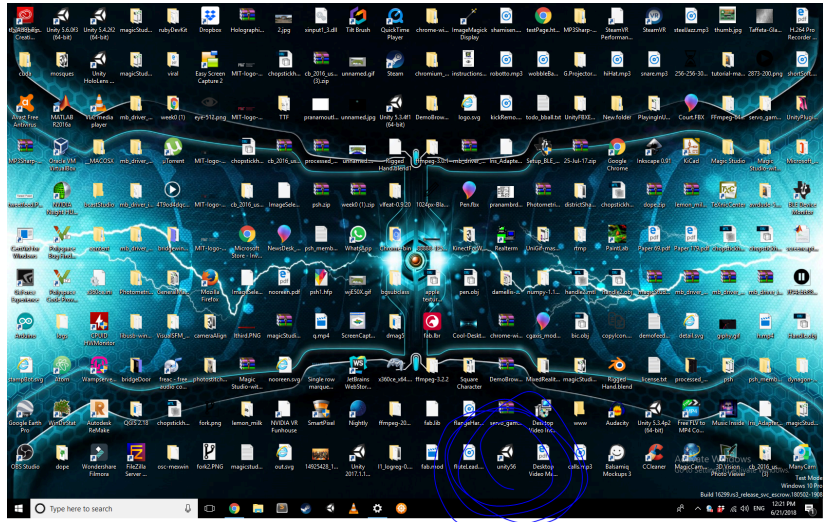


Once these are plugged in, you can check that signal is coming through to all of them by opening up the blackmagic media express application (installed, just hit the windows button and type in blackmagic media express).

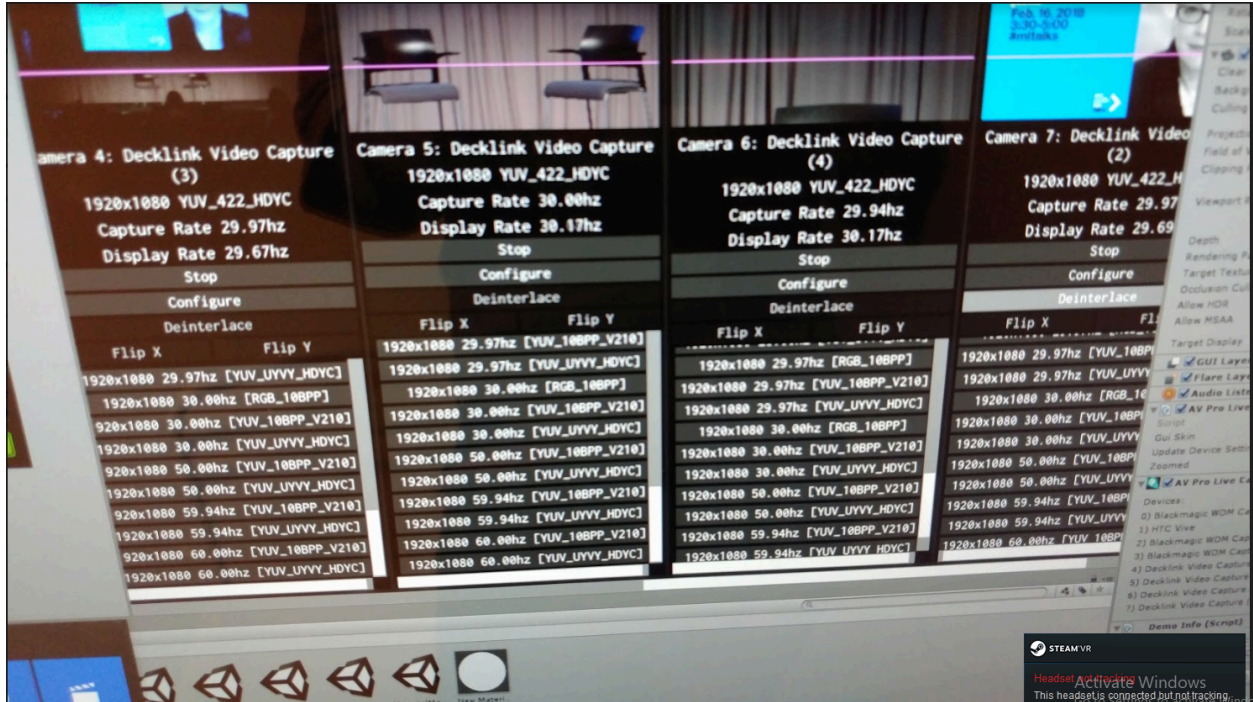


You can then click the various SDI digitizer devices, and switch between playback/log and capture to see that they are all coming through.

Once all the cameras are coming through, you can double check that Unity will be able to see them by opening Unity 5.6, and opening the project: **D:/UnityProjects/decklinktry**



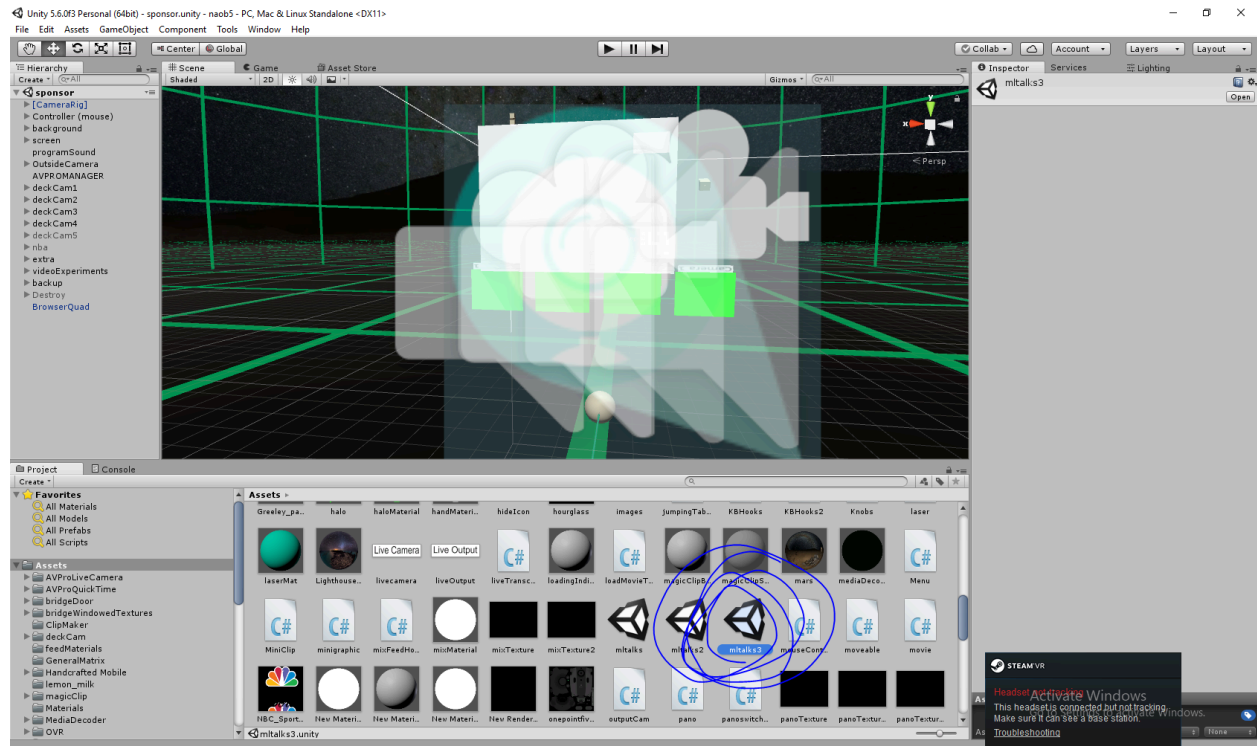
Hit play, and you will be able to see the various cameras coming in below. You can then select various settings to see which ones get the cameras, and not black screens. You will want to test out the devices which are called “decklink Video Capture”.



Once that's done, you will need to setup the cameras in the VR studio:

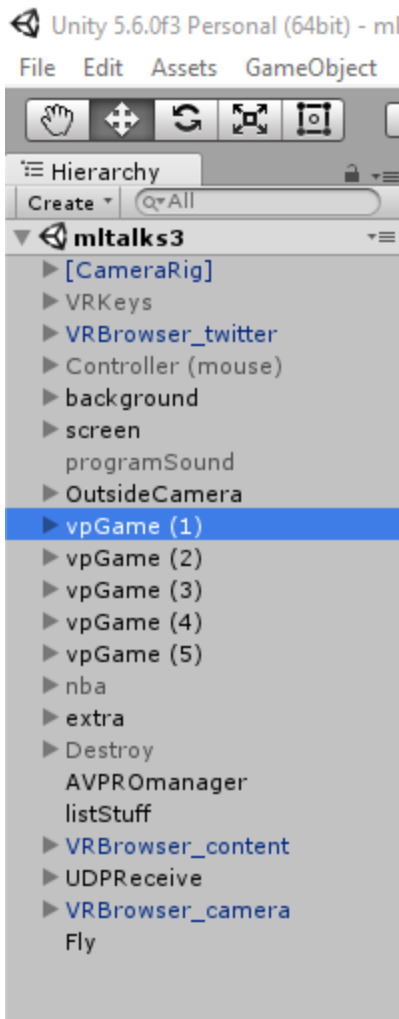
Setting up the cameras in the VR studio:

Open Unity 5.6 and select the project: **D:/UnityProjects/NAOB5**

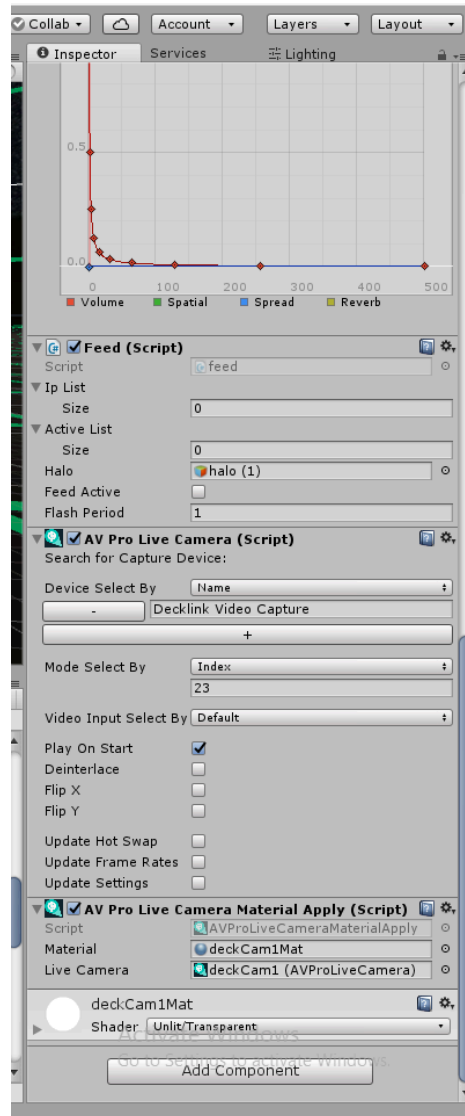


In the assets menu, select the Unity Scene called **mItalks3**. Double click on it to load the scene.

On the left-side, there are a bunch of objects in the heirarchy called vpGame:



You will need to click through each one and assign a digitizer device to it. When selecting each one, you will see the following in the inspector on the right:



You will need to select the correct mode by its index. It will likely be 23 for the cameras what aren't graphics feeds. You can hit run at the top to populate the screens, and view what happens in the "scene" mode. You can then change the index number live to see if you got it correct. Sorry for the trial and error :(

Once you've got all the correct numbers for the 5 different deckcam objects, remember them, hit the "play" button to stop the game from running (if it was already playing). Once not in play mode, you can enter those values into the inspector and save. After they are all put in there, you can put them in and hit "play" again to run.

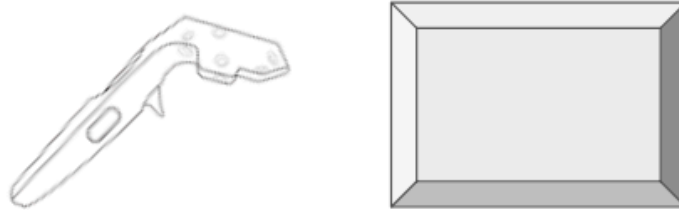
The rest of the app works just like the first section, so i will refer you to the section "**Intro to interacting with the System**". The main difference between the NBA demo and the live demo is that the live-demo does not create replay clips of the live broadcast, and there are web-browsers you can use.

Using the in-built browser:

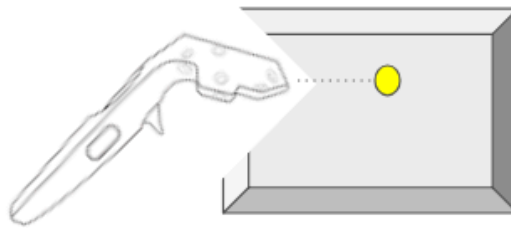
We built an in-game browser for viewing random information and pulling up youtube clips. It works through the zenfulcrum browser asset, which is a chromium browser with limitations to basically only open codecs (webm, etc...). Youtube works though.

In order to interact with the browser, you need to both be looking at it, and using your controller. When it's interactable, you'll see a yellow cursor on the screen.

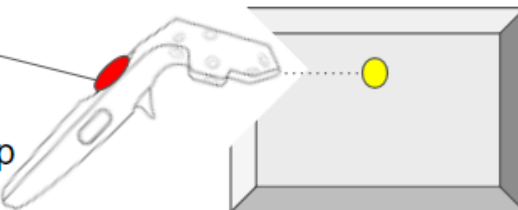
Using the Browser:



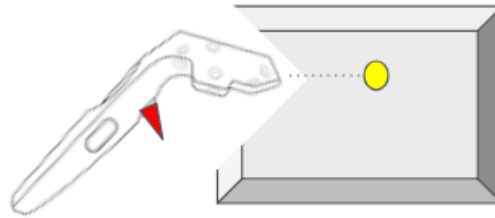
Hover controller over the browser window. A yellow dot will appear where the cursor is.



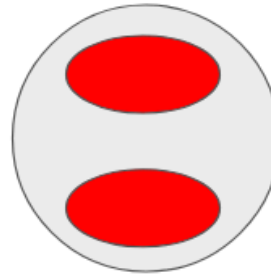
Press Touchpad to do a middle click. This will enable you to scroll by moving your controller up and down (just like on a web-browser)



Hover controller over the browser window. A yellow dot will appear where the cursor is.



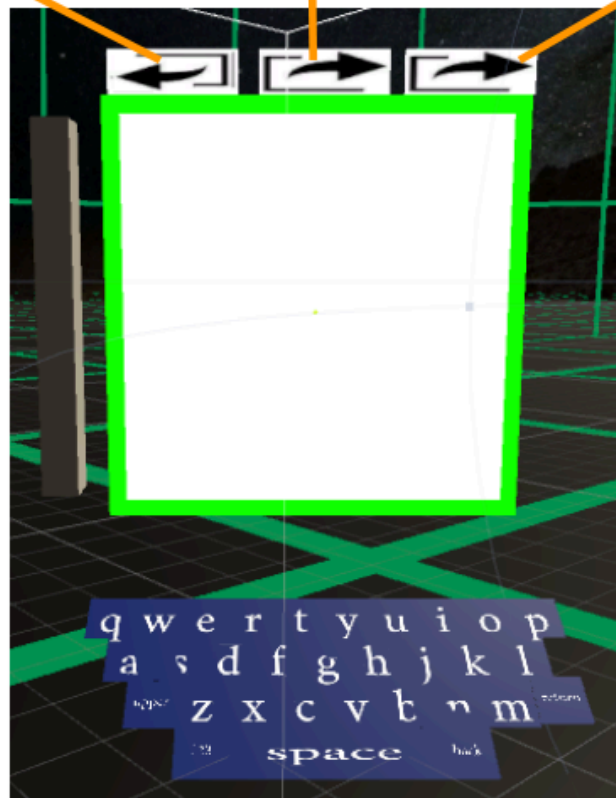
Pressing on the upper side or lower side of the touchpad causes the page to zoom in or zoom out.



Back Button

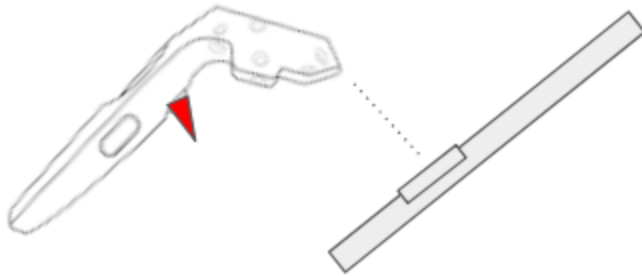
Google Button

Refresh Button



To use the keyboard, over your controller over the key you want, then pull the trigger. This can be tricky, but it should work. You can also switch to capital letters, press backspace, and switch to special characters.

Using the keyboard:

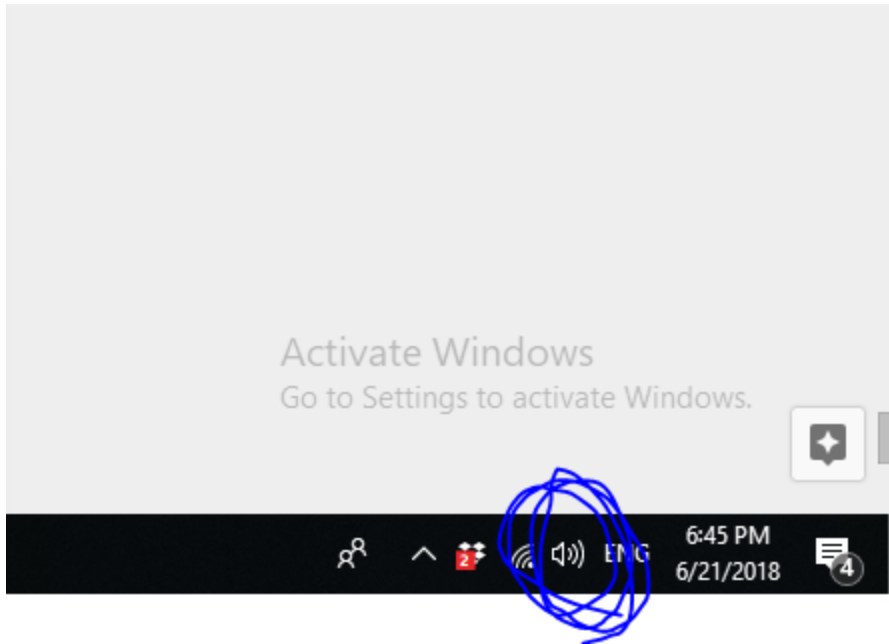


Flying around the studio:

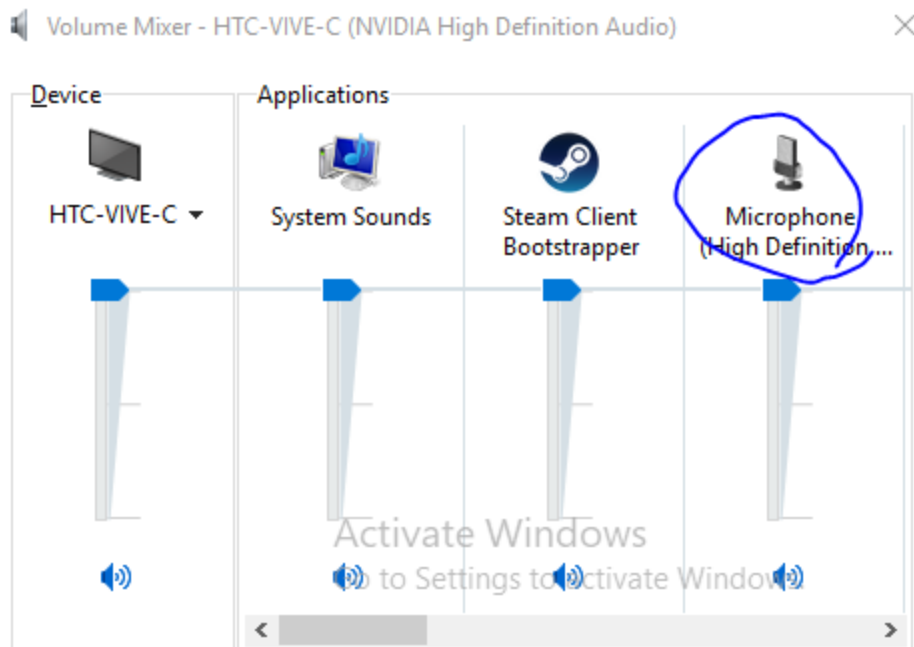
Sometimes you don't have space to move around while doing a live broadcast. Thus you can fly around the studio by pressing and holding the grip of the controller. You can then move the controller any way you would like to move yourself around the VR studio. Be careful though, it's sensitive, but once you get the hang of it, you can go really fast to where you need to be.

Listening to the audio from the live-audio feed:

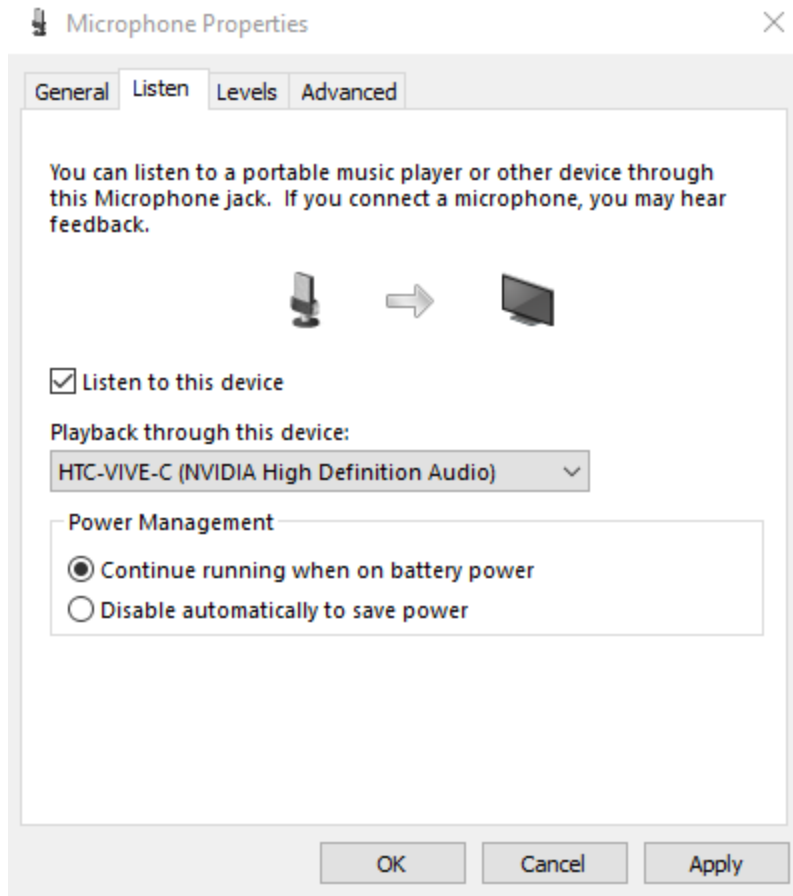
Right-click on the audio icon on the bottom-right of the screen:



Click on “volume mixer”



Click on the microphone icon:

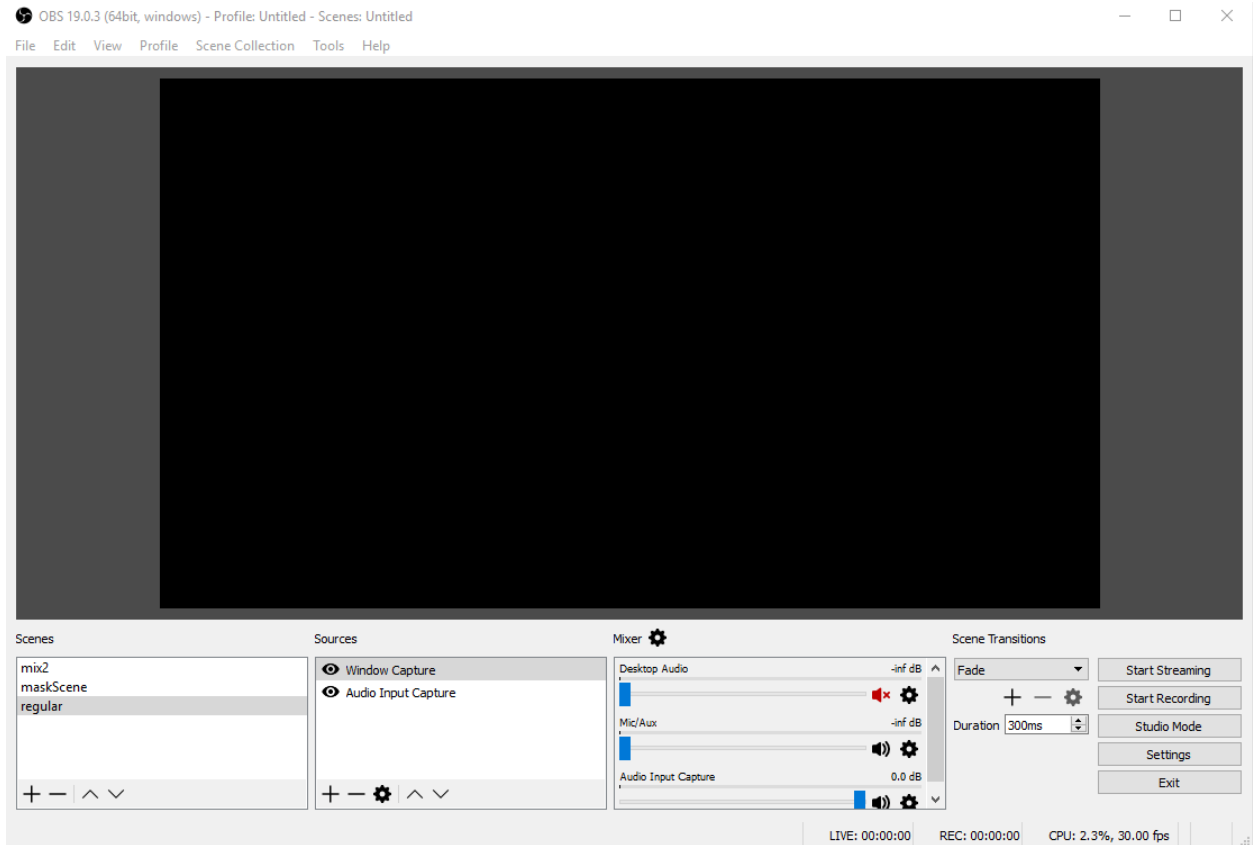


Click on "listen to this device", and make sure the audio is playing back through the audio on the vive (i'm assuming you are pluggin in headphones into the video audio jack attacked to the headset).

Recording a video of what's happening:

At the end of the day, you will need to make a broadcast of what's happening. This is done through Open Broadcaster Studio (OBS). This is "free" software which you can use to screen capture, stream to twitch, or just record to the computer. You can open it by pressing the windows button and typing "obs".

You want to create a window-capture that grabs the unity screen. In addition you want the audio from the microphone (audio capture). Set the OBS settings to save to a video file on the PC. Make sure you save to a format such as FLV. This format allows the computer to randomly die, but still have the data recorded so far be recoverable. This is not true of the mp4 file format.



Make a short video, and watch it, just to make sure everything worked out. Once that's done, hit record, and you will be capturing the broadcast.