Instruction Manual for a Personal Virtual Studio

Table of Contents

Scope

This instruction manual provides steps on how to set up a small-scale virtual studio with the use of Unreal Engine 4. The setup process requires downloading and using third-party software on your computer and phone from Steam, E2esoft, and Network Device Interface (NDI). It also involves downloading two Unreal assets, **GreenScreenMask.uasset** and **Main_TrackerV3.uasset**, from our GitHub repository. These software and assets are crucial in the setup process of a virtual studio. Follow this tutorial video: [Instruction Set Tutorial](#) with the instruction set to get the best result.

Tools and Start-Up

Software for Setting up your Virtual Studio

Download and install the following tools:
1. [Unreal 4.25.4](#)
   a. This can be downloaded from the Epic Games store. The virtual studio is hosted on the Unreal Engine.
2. [Steam VR](#)
   a. This can be downloaded from Steam. Steam VR will be used to connect a VR headset to the virtual studio.
3. [iVCam](#)*
   a. This can be downloaded from E2esoft on both the computer and phone.
      i. There is a QR code found on the bottom left of the iVCam desktop application that needs to be scanned to download the iVCam mobile application.
      ii. The mobile application is also found on the Apple store or Google Play store.
   b. *NOTE*: The hosting computer needs to be fully updated for iVCam to work. If this requirement is not met, the program will continuously display reinstall alerts until the operating system has been updated.
4. Download the [repository](#) as a zip folder and save it on your local drive. Open the folder **Final Assets** and it will contain the two assets needed for the setup, **GreenScreenMask.uasset** and **Main_TrackerV3.uasset**. Take note of the location of this folder because its contents will be extracted later on in this instruction manual.

Software for Live Streaming Unreal to Zoom

5. [NDI Tools](#)
   a. The NDI SDK is required to use the included Webcam Input that broadcasts the Unreal Engine scene to Zoom.
6. [NDI Plugin for Unreal](#)
   a. The NDI Plugin for Unreal is needed to make a sender in the Unreal Engine. The sender is the main component that is needed to broadcast the Unreal Engine scene to Zoom.

Minimum Requirements

Unreal Engine 4 System Requirements

● Desktop PC or Mac
● Windows 7 64-bit or Mac OS X 10.9.2
● Quad-core Intel or AMD processor, 2.5 GHz
● NVIDIA GeForce 970 GTX or AMD Radeon r9 390x HD series card
● 8 GB RAM

Hardware Requirements

● A Virtual Reality Headset
   ○ Example: VIVE Pro Full Kit
● Green Screen*
   ○ *NOTE*: Any solid color background can serve as a "green screen." However, ensure that there are no objects in the scene of the virtual studio that are the same color as the background. For example, do not put a red apple in front of a red background.
● Lighting
   ○ To help remove any shadows or discoloration captured by the camera.
● Tripod or stand
   ○ Tripod or stand-like object that can be used to hold your phone camera up directly in front of the object and the green screen)

Additional Resources

This section is not required for you to set up your virtual studio but is here to provide additional information and help if needed.

Video Tutorials

- [How to Download and Install Unreal Engine 4 on Windows 10](#)
  - This tutorial illustrates how to download and install Unreal Engine 4 on Windows 10. The required version of Unreal Engine 4 for this tutorial is 4.25.4.
- [Downloading and Installing Unreal Engine a Mac](#)
  - This is a tutorial on how to download and install Unreal Engine 4 on MacOS. The required version of Unreal Engine 4 for this tutorial is 4.25.4.
- [NDI Broadcast Setup in Unreal Engine](#)
  - This video walks through the steps on how to set up an NDI Video output broadcast in Unreal Engine.
- [Unreal Virtual Production for Zoom](#)
  - This video shows how to use NDI Tools with Unreal Engine and Zoom to facilitate fully virtual environments for use within Zoom meetings.
- [Instruction Set Tutorial](#)
  - This is a video that goes through all the steps found in this instruction set.

Notes Before Starting

- Set up the green screen
- Connect a virtual headset to the Unreal Engine by starting up Steam VR and going through the calibration process.
- Choose **STANDING ONLY** as the Room Setup
- Follow the remaining steps in SteamVR to ensure tracking for the VR headset


Steps

Virtual Studio Room Setup

The setup for a virtual studio will vary depending on what you are trying to achieve. However, a basic setup will look like the following:

1. A green screen (or any solid color sheet, wall, etc.) Hung up vertically on a leveled surface. Make sure the green screen is within the lines of your VR headsets environment.
2. Place an object or yourself in front of the green screen
3. Use lighting to light up the object and the green screen behind it without any shadows. (This step can be a little tricky as you may need multiple lamps and lights to achieve this. Be mindful of using white or yellow lights - as this can change the coloring of the video input)
4. Place your phone on a tripod or a stand so that the camera is capturing the object and the green screen.

## File Importing and Initialization

1. Before launching Unreal Engine 4.25.4, make sure Steam VR is running (Make sure your VR set is connected to your laptop, and it recognizes one of the controllers). Also, make sure iVCam is running on both the computer and phone and that they are connected to each other. IVCam is working when the phone's camera feed is being displayed on the desktop.
2. In the Unreal Project Browser, create a new project or select a project that was previously made.
3. To create a new project simply choose **Film, Television and Live Events** (another type of project can be used as well) under New Project Categories, and click "Next." For this tutorial, choose the **Blank** template option.
4. With the default settings in Project Settings, name the project and click "Create Project."
5. Enable the **NDI IO Plugin** under *Edit → Plugins → Virtual Production*. Once enabled, Unreal Engine will prompt to restart the software. After restarting, a new tab labeled NDI will appear in Modes.
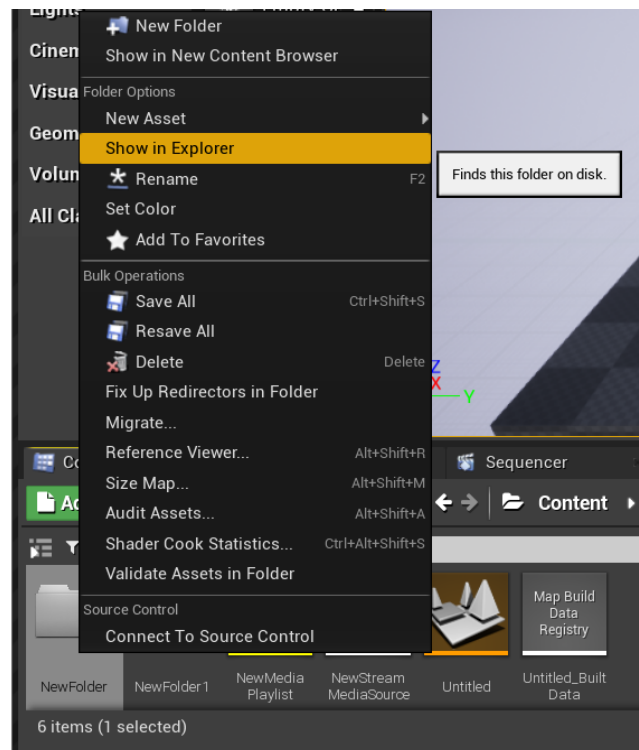
6. Under the Content Browser, create a new folder named "VRTools."
7. Right-click on the folder and select "Show in Explorer". This will open the VRTools folder in File Explorer.
8. Open a separate File Explorer window and navigate to the extracted files found in the **Final Assets** folder, **GreenScreenMask.uasset** and **MainTracker.uasset,** from the GitHub repository folder.
9. Copy and paste the files into the VRTools folder. Close those two File Explorers.*
    a. *NOTE:* If the Unreal Editor does not recognize the new files, right-click the new folder again and select "Fix up Redirectors in Folder."
10. Going back to the Unreal Engine, enter the VRTools folder in the Content Browser and drag the **MainTracker** file into the scene.
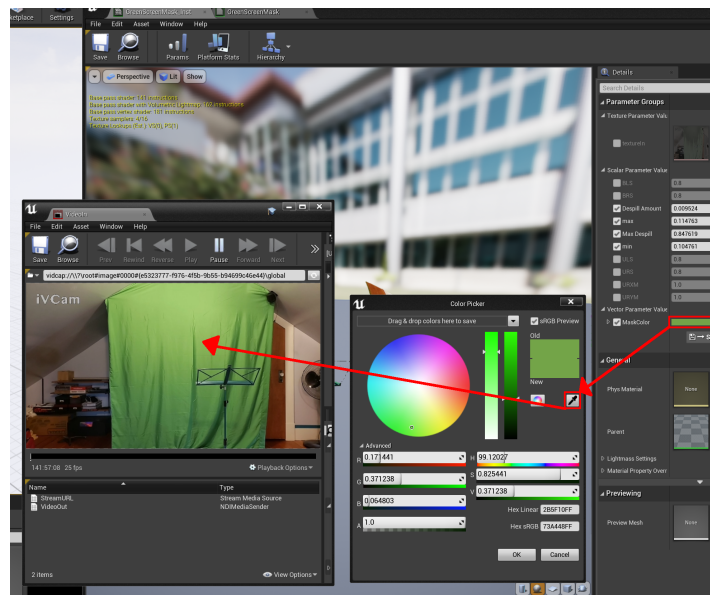
Media Player (Video Timestamp 5:20)

11. Set up iVCam on both your device and your laptop. Ensure that the camera input from your chosen device is being reflected on the iVCam app on your laptop.
12. Create a new media player by going to *Add New→ Media→Media Player* within the VRTools folder. A prompt asking if a video output MediaTexture asset should be created. Be sure to check the box. Label this media player **videoIn**. A media texture will be created and make sure to name it **videoIn_Video** (this should be renamed automatically).

13. Open the Media Player you created in the previous step and look for the small folder button in the upper left corner right below the save button (Shown on the figure to the right). If the iVCam has been launched before, under the video tab, a choice called "e2eSoft iVCam input."

14. On the right of the window, save the generated URL by going to the Playlist tab. Do this by clicking the save icon next to the Playlist tab to save your work. (Shown in the image to the right). Name the generated file **streamURL** and click save. It will also generate a playlist, but that is not needed so you can just click cancel.


6
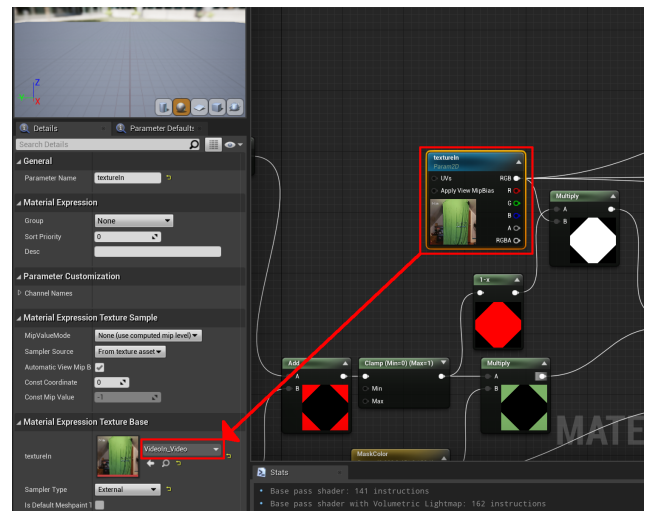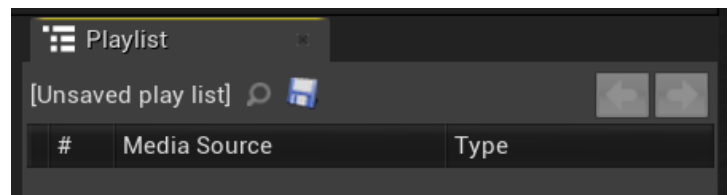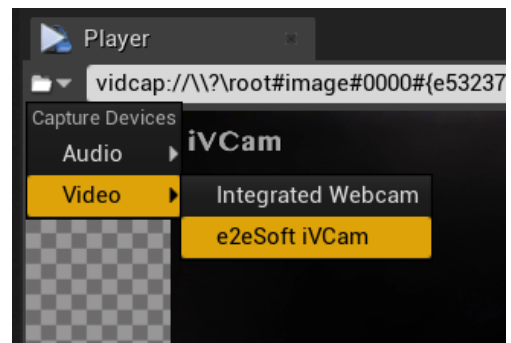
15. Click Save at the top left of the window and be sure to minimize not close it.
16. Open up the GreenScreenMask and click on the "TextureIn" block within the map. Once that is selected navigate to the left side of the window under Material Expression Texture Base and in the Texture, drop-down select the "VideoIn_Video". (Shown in the image above) Click Apply and Save.
17. Make sure that your phone camera is pointing to the green screen (explained in Virtual Studio Room Setup, step # 4) and it is displayed in the "VideoIn_Video" icon in the bottom left content folder.
18. Next, we'll be creating a material instance. Right-click on the GreenScreenMask and click "Create a Material instance".
19. Double-Click on the instance created in the previous step. Check the following boxes:
    a. Despill Amount
    b. Max (Maximum color value)
    c. Max Despill
    d. Min (Minimum color value)
    e. MaskColor
20. Click on the color bar to open up a screen Once that opens up click on the color picker (the pipette) and select the green color from your <VideoIn>_Video. Then click OK.

    a. *Remember you should be able to see the green screen in the little thumbnails in the content browser to choose what green color to make translucent.*
21. Calibrate the material instance with settings under "Scalar Parameter Values" enable all four settings below and try manipulating variables from 19a - 19d. The material sphere should be updated in real-time to accommodate the new settings. (Instructions on how to achieve this can be found at ~8:30 on the Instruction Set Tutorial video)
22. Click Save and minimize the window.
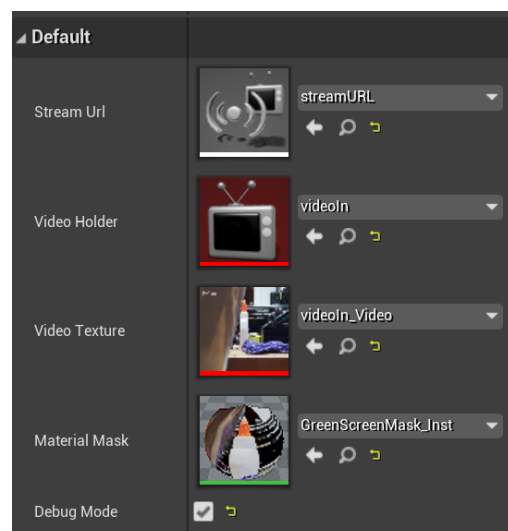
23. Select the Main_TrackerV3 element in the scene and navigate to the Default section under Details on the right of the screen. (Shown in the image to the right) Populate the variables with the following:
    a. Stream URL: **streamURL** (The URL we saved in step 14)
    b. Video Holder: **videoIn** (created from step 12)
    c. Video Texture: **videoIn_Video** (created From step 12)
    d. Material Mask: **GreenScreenMask_Instance** (Created from step 18)
    e. Under Pawns→Auto Posses Player use the drop-down to select "Player 0".
    f. Under the Details list, select "NDIViewportCapture," under "Current Camera Settings," in the drop down for "lens settings," set it to "12mm Prime f/2.8."
24. Use the [Instruction Set Tutorial](#) video to calibrate your Main_TrackerV3 settings at ~9:50.

Video Output (NDI) - Unreal Engine 4 (Video Timestamp 12:00)

25. Before moving to the next step, turn on "Webcam Input" from the NDI SDK downloaded in the previous steps
26. We now need the NDI Viewport Camera, to do so go find the Main_TrackerV3 under the World Outliner and select "Edit Main_TrackerV3". A window will appear.
27. On the left of the opened window underneath Components click on "NDIViewportCapture", then underneath Details scroll down to find the Properties tab.
28. Click on the drop-down menu of the "NDI Media Source" and select "NDI Media Sender" under Create New Source. A window will pop up and simply call the NDI Media Sender "VideoOut" then click save.
29. Go back to the Components table, find "NDIBroadcast" and select it. Then go to Details scrolling down to NDI Media Source underneath Properties and open the drop-down menu to select "VideoOut".
30. Compile and save your changes, then close out of the window.
31. Make sure your VR controllers are on, then click Play to test the project.
    a. *If you need to adjust your camera's view so that it's aligned with your green screen you can move the Borders in the Unreal Engine to fit your green screen.*
32. Now pause the scene, and click on the VideoOut that you made earlier underneath the Content Browser.
33. A window will pop up. Find "Render Target" underneath Content, and click on the drop-down arrow to reveal a menu, and click on Render Target.
34. Another window will open up to create a new Render Target, simply name this "TextureOut" and click Save.
35. Going back to the previous window make sure to Save then close the window.
36. Click the Play button again, and now you're ready to stream your project over Zoom.

Streaming to Zoom (NDI) (Video Timestamp 16:45)

37. While the project is running, Right-click on the Webcam Input icon on the bottom right of your taskbar to make sure your selected output is "Unreal Engine Output".
38. Open up Zoom.
39. At the right upper right corner of Zoom, click on the Settings button (gear icon) and a window will pop up.
40. Go to the Video tab, and navigate to Camera. Click on the drop-down menu and simply select "NewTek NDI Video".
41. In the preview screen, you will see your Unreal Engine.
42. Exit out of the settings menu, be sure to disable debug mode if it is enabled, and you are ready to go into your Zoom meeting with a virtual studio.

Troubleshooting

iVCam
- **iVCam is not connecting to my computer/showing my video feed.**
  - If iVCam is not showing a feed on your laptop when you press the play button, it means that it's getting blocked by your Windows firewall. Make sure to let iVCam have private and public access, to do so go to the ***Control Panel → System and Security → Windows Defender Firewall →*** click on ***"Allow an app or feature through Windows Defender Firewall"***, you'll see a list of programs, click on the ***Change Settings*** button, then scroll down and find ***e2eSoft iVCam*** and make sure it's check box is ticked along with the private and public boxes associated with the application. This should let your phone connect to your desktop through iVCam, and the video feed will appear.

Unreal Engine
- **Video feed disappears in the plane.**
  - If the video feed disappears after adding the texture to the plane that means the camera has disconnected from the engine.
    - Make sure you save before exiting out of the *NewMediaPlayer* window.
    - Or, leave the window open if the video feed stops running when you close that window.
    - Try waking up your phone when it falls asleep, sometimes the video feed blacks out when the phone locks.

Group 8: Develop a Virtual Video Studio
Evan Medinger, Joelle Perez, Sowmya Jujjuri, Xindi Guo
Oregon State University

Special thanks to:
Mike Premi, Project Partner
Raffaele De Amicis, Technical Project Partner
Nathan De Stefano, Teaching Assistant