

Setting up Stereoscopic Rendering with nDisplay for LAVA's 3D Display Wall

DRAFT

Jason Leigh

Laboratory for Advanced Visualization & Applications

Version 12/14/2025



This document provides a step-by-step guide for setting up stereoscopic rendering on LAVA's 3D display wall using Unreal Engine's nDisplay, covering both new projects and integrating nDisplay into existing ones. It explains how to configure nDisplay assets, Switchboard launching, and optional low-level setup details.

Before you begin, be aware that nDisplay has not been implemented for the Mac.

There are two ways to start:

1. Start with [Part A](#) to build a **new** Unreal 3D Display Wall project.
2. Start with [Part B](#) if you want to **add** LAVA's nDisplay capability to an **existing** Unreal project.

Then continue with the following two parts:

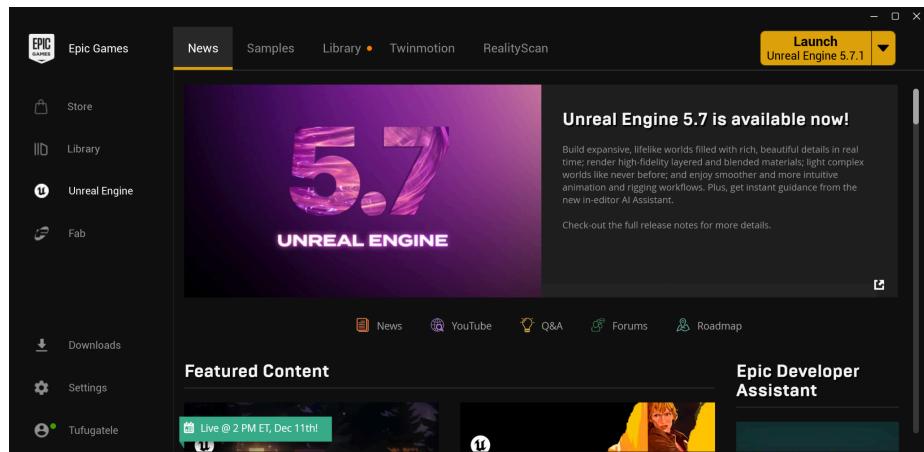
- [Part C](#) - Adding LAVA's nDisplay Asset to Existing Unreal Project
- [Part D](#) - Set up the Switchboard and Switchboard Listener to launch nDisplay
- [Part E](#) - Tips and Caveats

If you wish to understand how the LAVA nDisplay asset is implemented, go through Part A, [Part E](#), D and F.

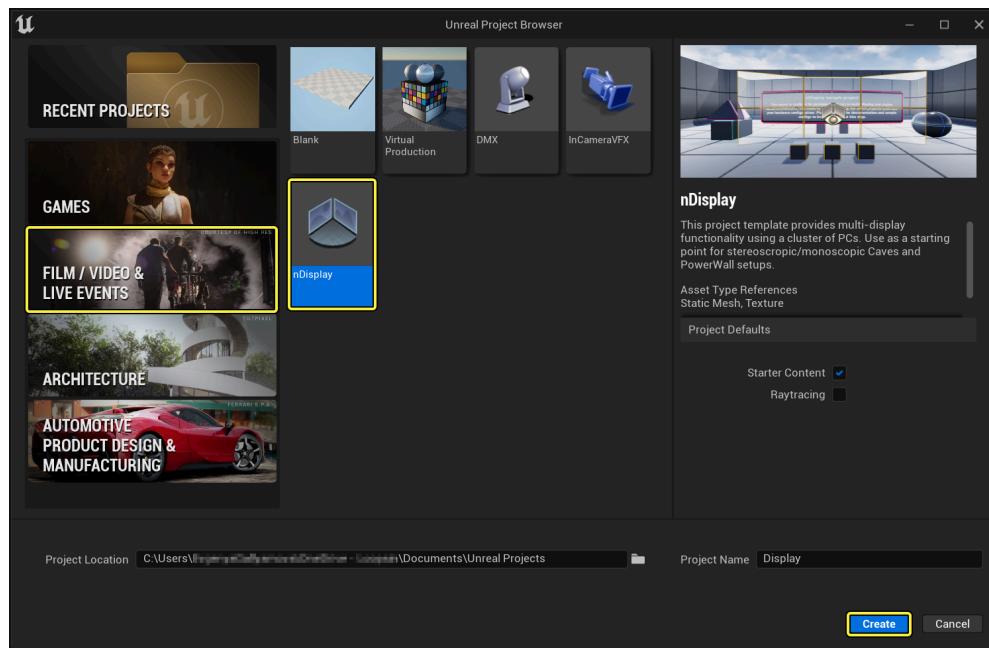
Part A. Getting Started with nDisplay from Unreal nDisplay Project Template

Unreal's nDisplay project template is essentially their simulation template with the nDisplay plugins already installed. To install the nDisplay project, follow these steps:

1. Go to Epic Games website and download their hub. From the hub install the latest version of Unreal.
2. Having installed Unreal, launch it by clicking on the Launch Unreal Engine button



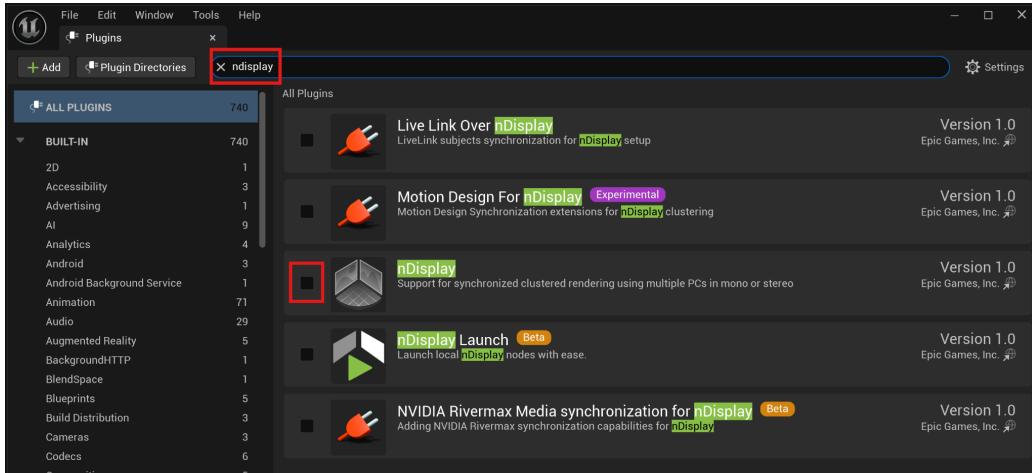
3. Create an nDisplay template project as follows: Click on **New Project**. Then click on **Film/Video & Live Events**. Choose the **nDisplay** template project and enter a suitable project name.



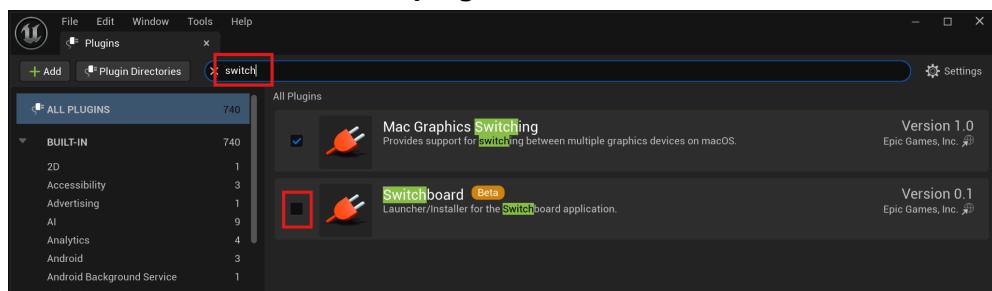
4. Look for the NDC_Basic actor in the Outline tab of your project. If you intend to go onto Part C, you can delete the NDC_Basic actor. If you plan to go on to Part E do not delete NDC_Basic actor.

Part B. Adding nDisplay and Switchboard Plugins to an Existing Unreal Project

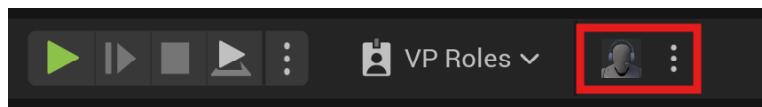
1. From your Unreal editor, go to the **Edit** menu and click **Plugins**.
2. Search for the **nDisplay plugin** and enable it. You may have noticed that there is an **nDisplay Launch** plugin too. Do not enable that. It's something you can explore later.



3. Now search for the **Switchboard plugin** and enable that too.

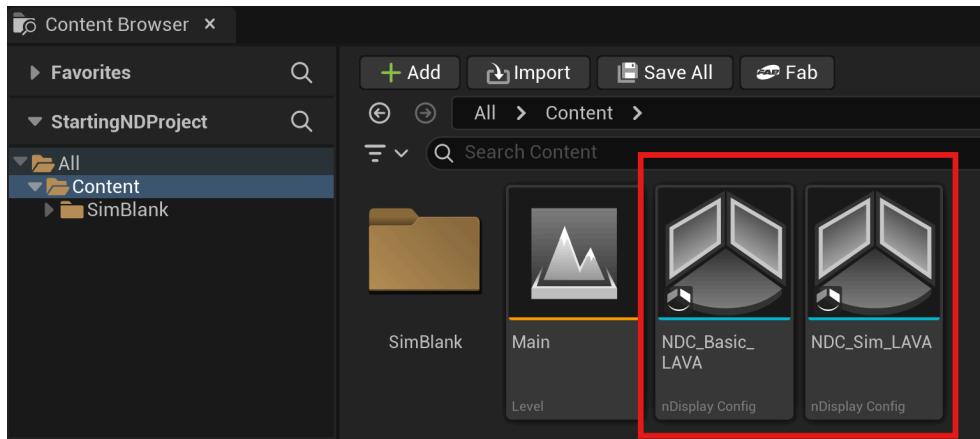


4. By now, you will likely also notice a prompt at the bottom of the plugin window asking you to restart Unreal. Go ahead and click on **Restart**.
5. You should see this at the top of your Unreal editor. Part D will explain how to use those buttons.



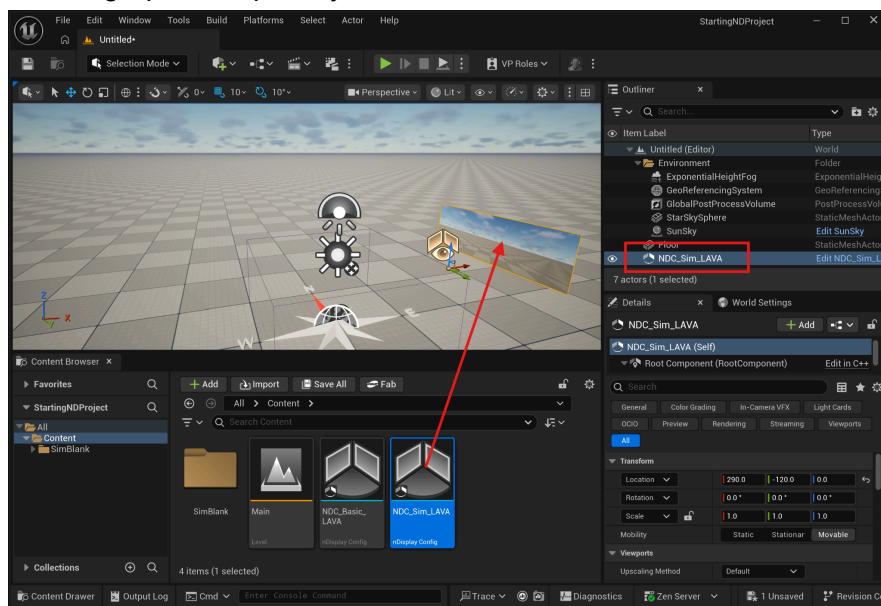
Part C. Adding nDisplay Support and LAVA nDisplay Asset to Existing Unreal Project

1. Download the **NDC_Basic_LAVA.uasset** file and the **NDC_Sim_LAVA.uasset**. (<https://github.com/doctorspiffy/Unreal-nDisplay-3D-Wall>) Copy and paste them into the **Content** folder of your UnReal project. You'll see them appear in your Unreal editor's **content browser tab**.



2. Drag either NDC_Basic_LAVA or NDC_Sim_LAVA asset into the Unreal editor's **scene** window to turn it into an actor, and then position and orient the resulting nDisplay preview window as appropriate. Note the following difference between NDC_Basic_LAVA and NDC_Sim_LAVA:

- The **NDC_Basic_LAVA** asset contains the nDisplay configuration to drive LAVA's display wall which requires you to render two side-by-side stereo pairs of 4800x1620 each- making a total resolution of 9600x1620. Such resolution will require substantial graphics memory and may choke a development laptop.
- The **NDC_Sim_LAVA** asset contains an nDisplay configuration of a scaled down display wall that you can use to make development possible on computers with lesser graphics capability.



- Now go to the Unreal Editor's **File** menu and click **Save All**.

Part D. Set up the Switchboard and Switchboard Listener to launch nDisplay

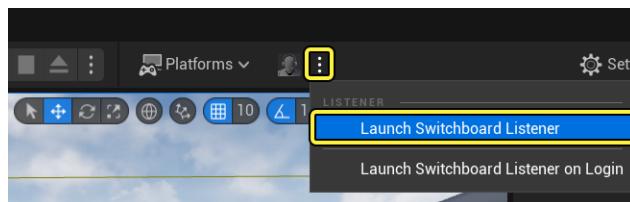
If you attempt to run your program by clicking on Play in the Unreal Editor, it will run the program but not by using nDisplay. To run via nDisplay, you need to use the Switchboard.

The switchboard controls the devices (computers) that drive the displays in your VR-room set up (whether it's a single display wall or a CAVE). Switchboard does this by communicating with Switchboard Listeners, potentially running on separate computers, to launch copies of your project. For a single display wall, both the Switchboard and Switchboard Listener will reside on the same computer. For more information on nDisplay visit:

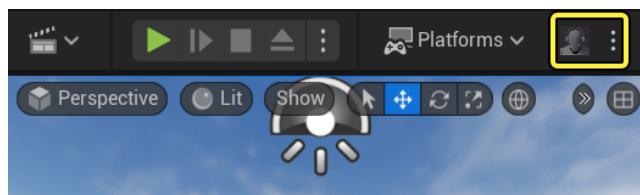
<https://dev.epicgames.com/documentation/en-us/unreal-engine/ndisplay-overview-for-unreal-engine>

The following are instructions for setting up Switchboard and the Listener:

- In the Unreal editor, go to the Toolbar, click the 3 dots next to Switchboard and choose **Launch Switchboard Listener**.



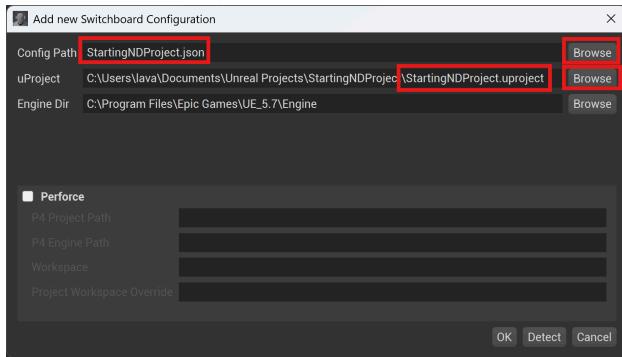
- Switchboard Listener launches and may immediately minimize itself. You may also be asked to provide a password. If you are not asked for the password then your organization may already have set one, so ask your administrator. Actually, if you look at the Settings menu in the listener there is an option for you to change the password if necessary. Later this password will be used to authorize the connection between the Switchboard and the Switchboard Listener.
- Anyway, continuing- once the switchboard listener is open, minimize it (don't close it).
- In the Unreal editor Toolbar, click the Switchboard button to launch the Switchboard application on your computer.



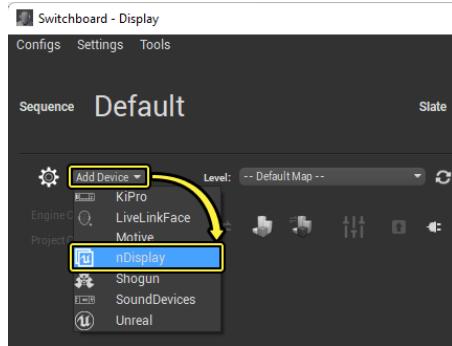
When launching Switchboard for the first time, a command prompt window may appear, installing required dependencies before the Switchboard window opens. Allow the installation to proceed.

- When the switchboard opens, the **Add new Switchboard Configuration** window will likely appear. A switchboard **configuration** is needed for each Unreal project that will

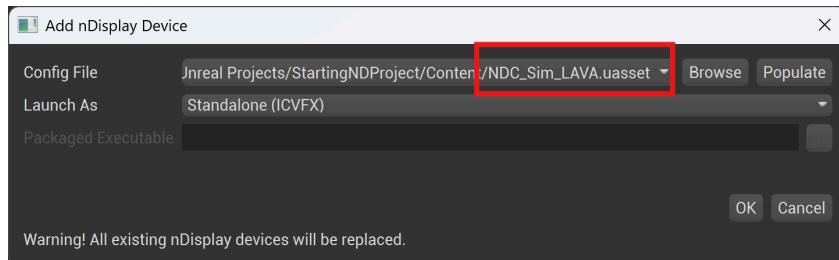
need to use nDisplay. If this window does not appear go to the Switchboard's **Configs** menu and click **New Config**.



6. Give a name to **Config Path** field- by clicking on the Browse button next to the field.
7. Then in the **uProject** field, browse to your project's .uproject file.
8. Then click OK to proceed.
9. In Switchboard, click **Add Device** in the top left and choose **nDisplay** to open the Add nDisplay Device window.

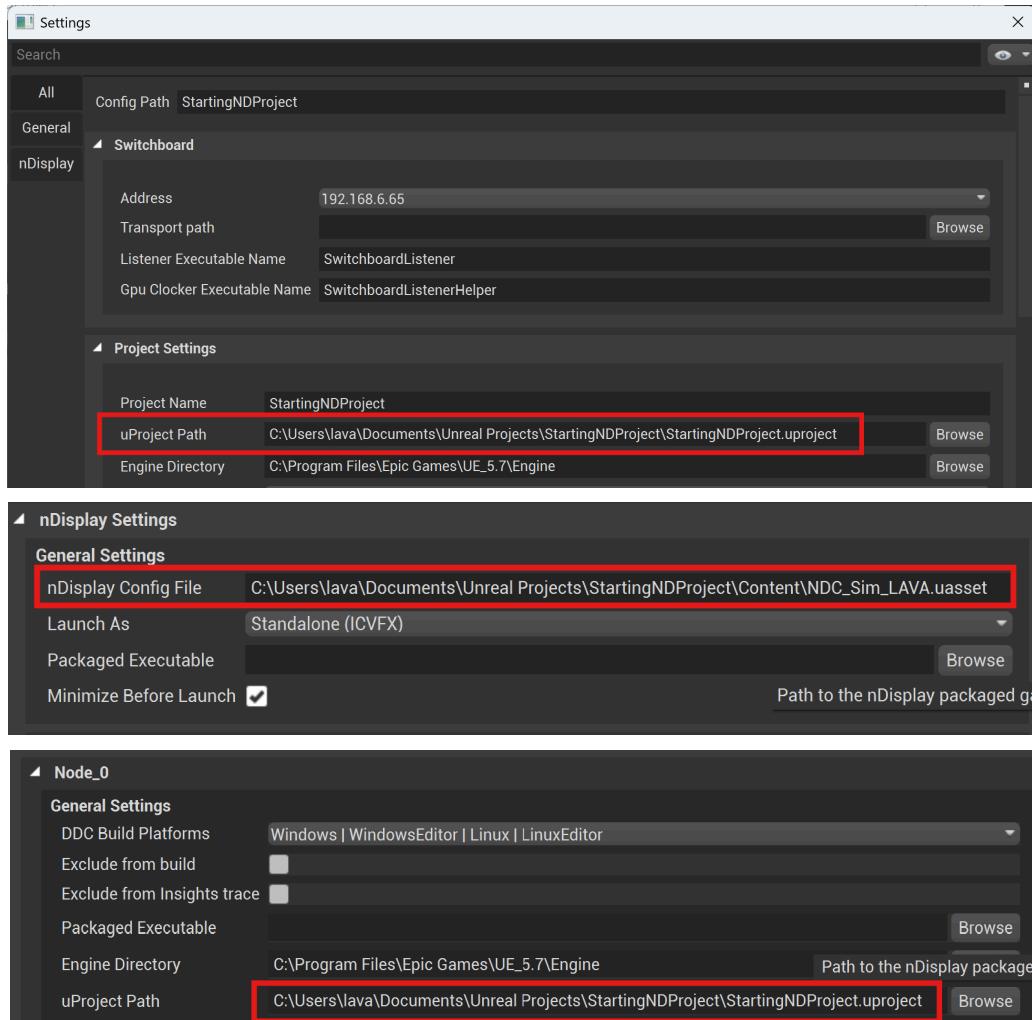


10. Browse for and select the nDisplay .uasset configuration file in your project (either **NDC_Basic_LAVA.uasset** or **NDC_Sim_LAVA.uasset**), and click **OK**. If you are doing this for the first time, I recommend using NDC_Sim_LAVA. NDC_Basic_LAVA is the configuration for the actual display wall. NDC_Sim_LAVA is intended as a lower resolution simulator that can run on weaker laptops/PCs. Later you will want to repeat the above steps in Part D to create a separate device configuration for NDC_Basic_LAVA. Having both configurations will make it easier for you to switch between testing your app in simulation mode and running your app in full blown resolution of the display wall.

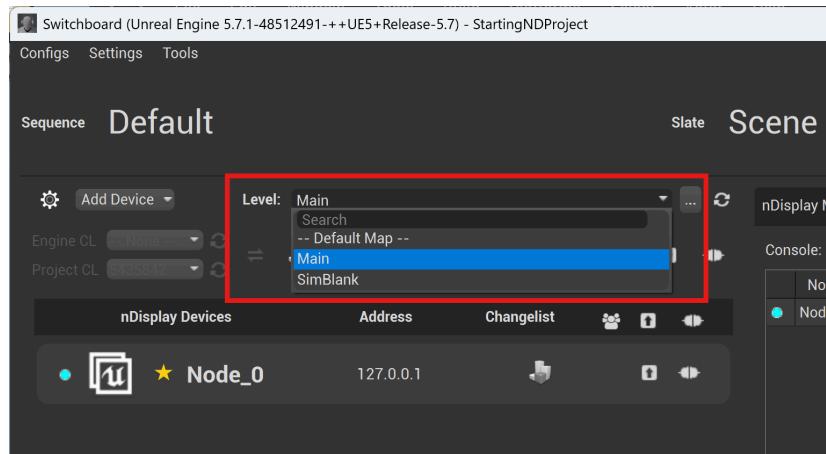


11. Anyway, continuing- now click on the **Settings** menu in the Switchboard and make sure the **Project Settings** sub-panel, the **nDisplay Settings** sub-panel, and the **Node_0**

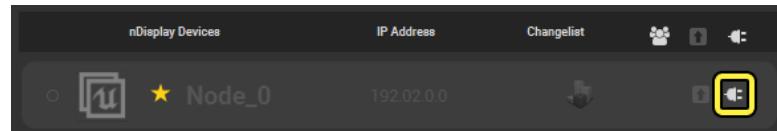
sub-panel contain the correct information. Below are examples of what you might see. Notice the paths show the location of your project file and the nDisplay uasset.



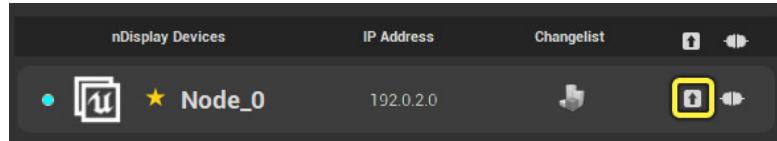
12. Close the **Settings** window.
13. In the Switchboard window, make sure **Level** is set to the correct “game” level of your app.



14. Click the **Connect to listener** button to connect to the **Switchboard Listener**. You may be asked to provide the password that you chose earlier.



15. Start the **nDisplay instance**. This will immediately minimize all windows and launch the nDisplay instance showing the scene. Note: it will take some time for the scene to finally appear, so be patient- especially on slower computers or for large projects.



16. Once launched, you should be able to interact with your program. For example, if you started with the nDisplay template project you should be able to use First-Person-Shooter controls (WASD + mouse) to move about the environment.

17. Press the ESC key to exit the application.

Final note: When you use your Switchboard again at a later date, you can quickly bring up the appropriate configuration for your project by clicking on the **Config** menu and selecting the previously created configuration. Don't just assume that the configuration you see loaded by default is the correct one, as it may be the configuration for someone else's project since your display wall is used by multiple people for multiple projects.

Part E. Set up nDisplay Actor from Scratch for Stereoscopic Rendering on LAVA's Display Wall

This part of the tutorial is optional to most users. It explains how to build LAVA's nDisplay actor from Part C from scratch. However you may also want to read this if you intend to modify the configuration in some way.

Fundamentally what LAVA's nDisplay asset does is, for each eye of your stereo image, assign each **view point** to a viewport, and nDisplay Screen, where the **viewport** reflects the resolution of the wall, and the **nDisplay screen** reflects the physical dimensions of the wall.

View Point → View Port → nDisplay Screen

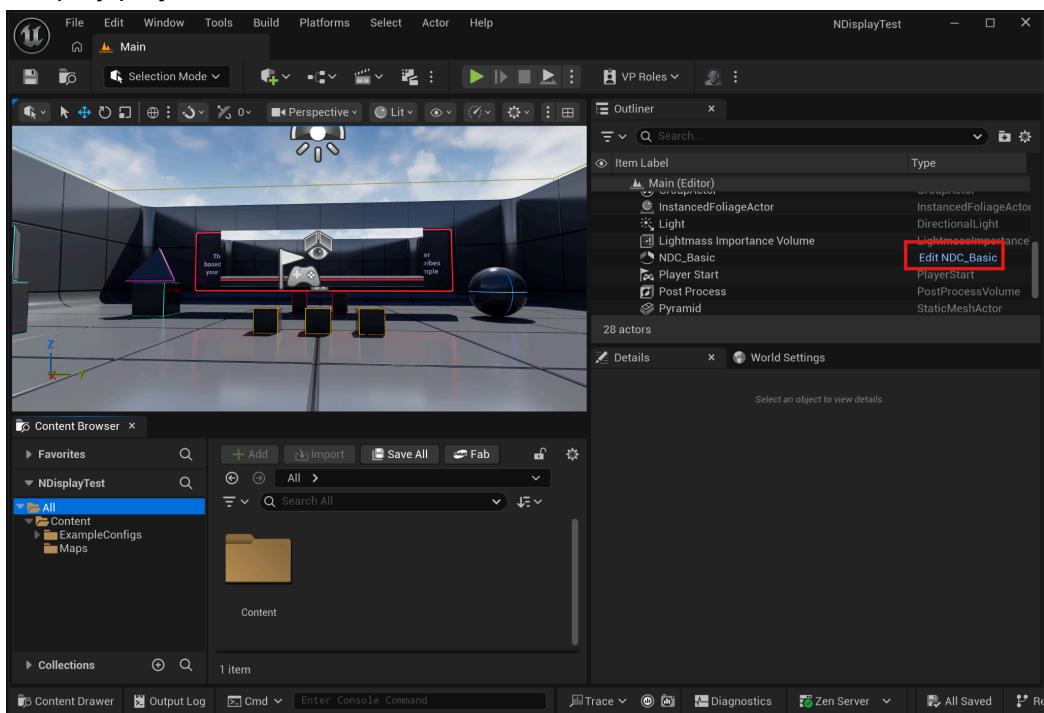
A view point is essentially the location of the user relative to the display wall.

A view port is the region into which the graphics will be rendered.

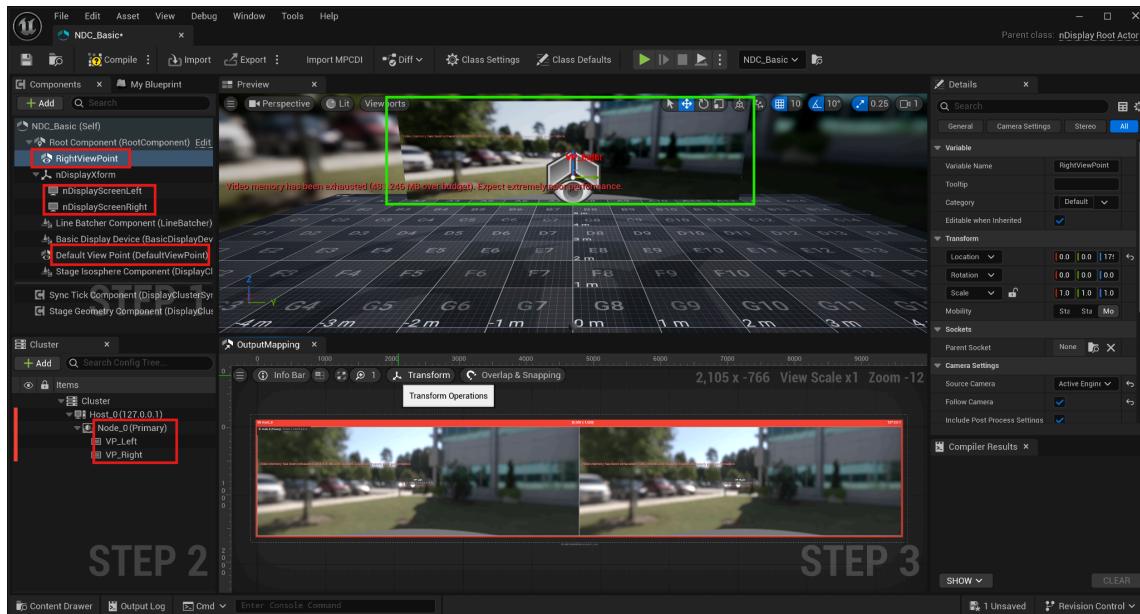
The nDisplay screen is the virtual representation of the physical display wall.

The steps needed to do this are:

1. Continuing from Part A, edit your nDisplay actor (NDC_Basic) in the Outliner tab of your nDisplay project.

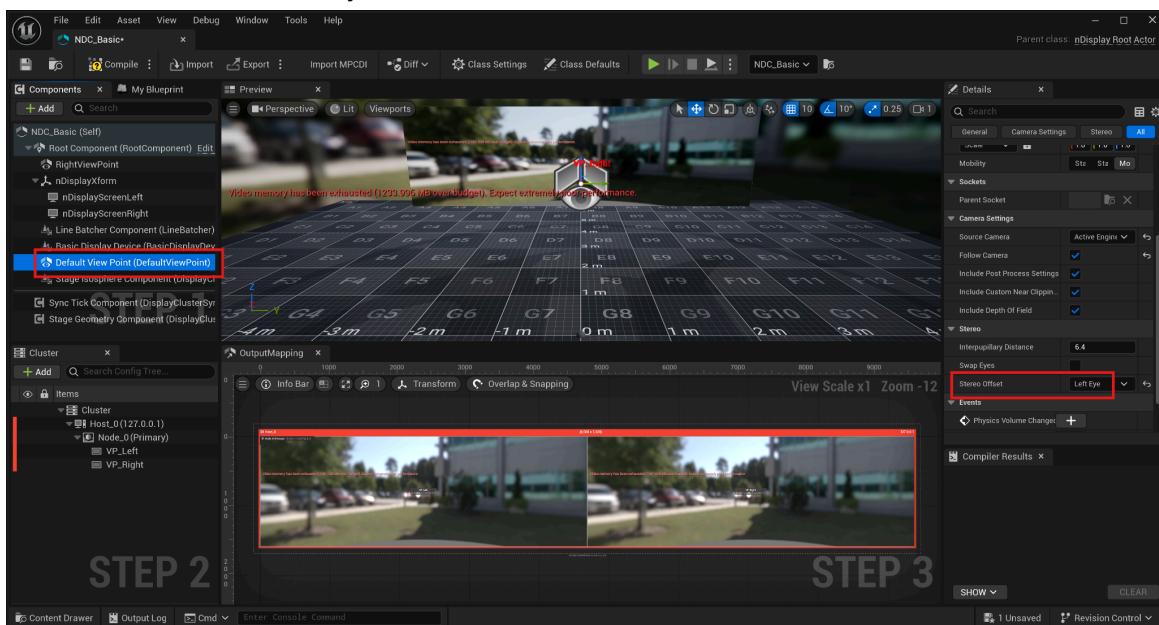


2. This is what the NDC_Basic's editor interface window look like:

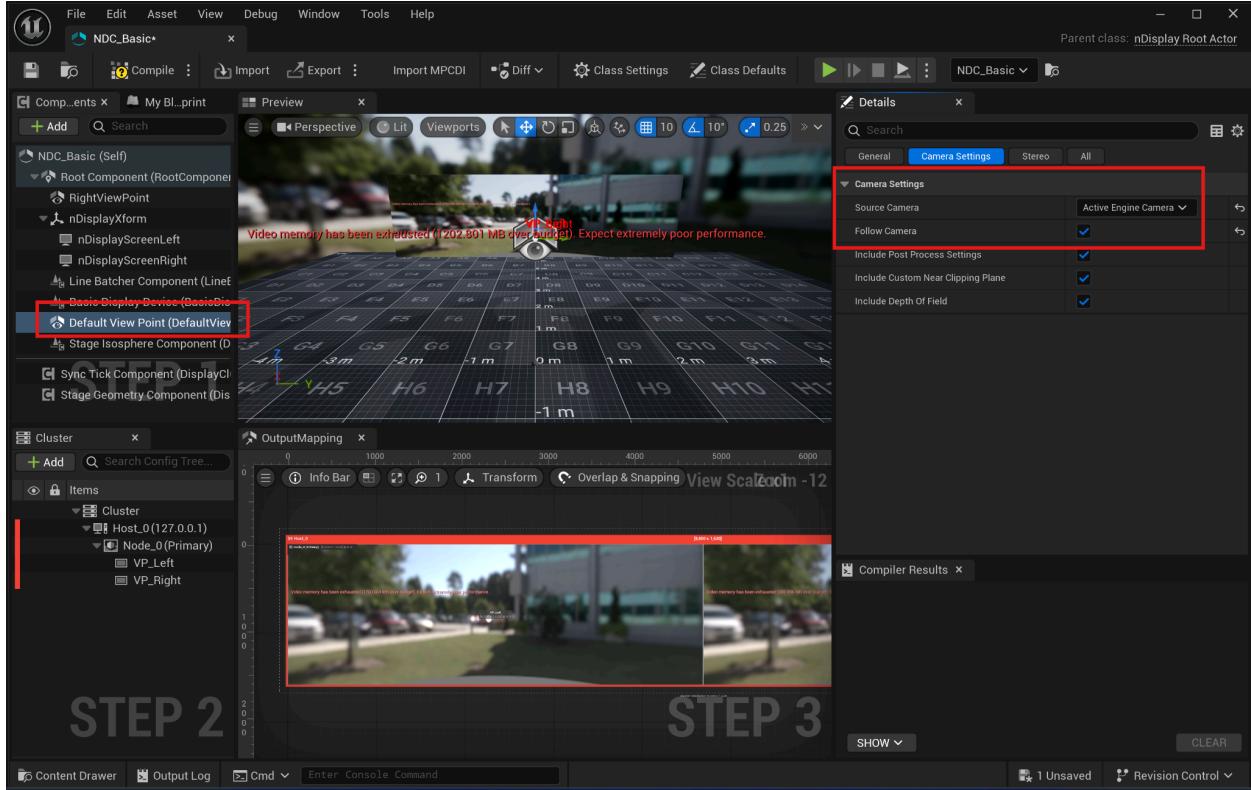


Marked in RED on the left are the nodes we will be modifying. The top left tab shows the components of the nDisplay actor. The bottom left tab shows the nDisplay cluster settings. The Preview window in the center top shows in green the nDisplay screens (which overlap each other because the left and right eye images share the same sized screens). The bottom Output Mapping window shows how the viewports map to Node_0's frame buffer.

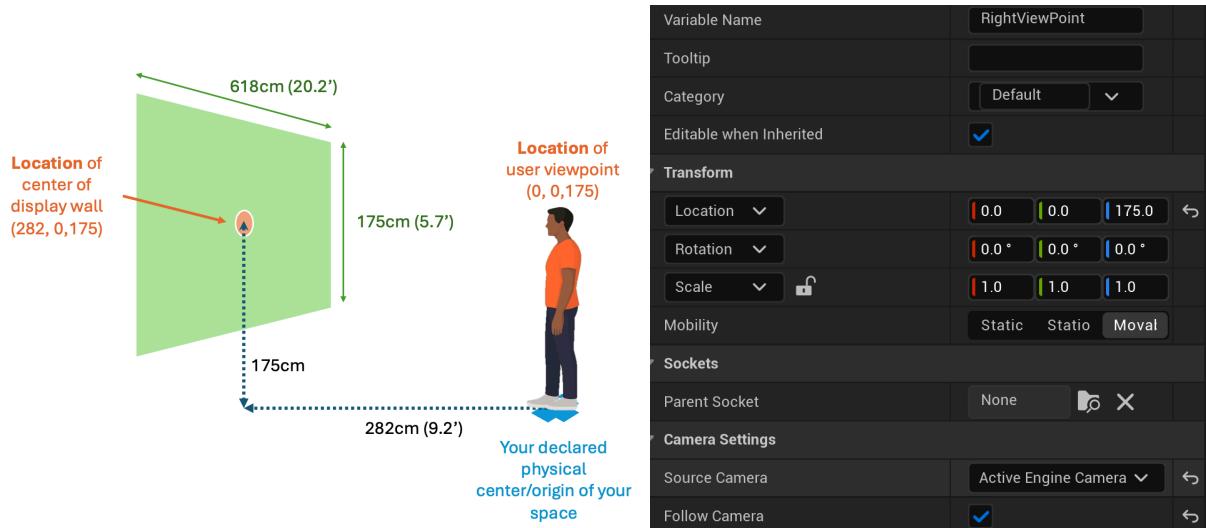
3. We're going to use the **Default Viewpoint** for the **left** eye. So click on Default Viewpoint in the Components panel and in the details panel on the right look for Stereo and set the Stereo Offset to Left Eye.



- We'll create a new viewpoint (and call it **RightViewPoint**) for the **right eye**, by duplicating the **Default Viewpoint** and setting stereo to **right eye**. To duplicate the Default Viewpoint, right click on it to bring up a menu and select **Duplicate**.
- Make sure for both viewpoints, the **camera setting → source camera** is set to **Active Engine Camera**, and also make sure **Follow Camera** is checked.

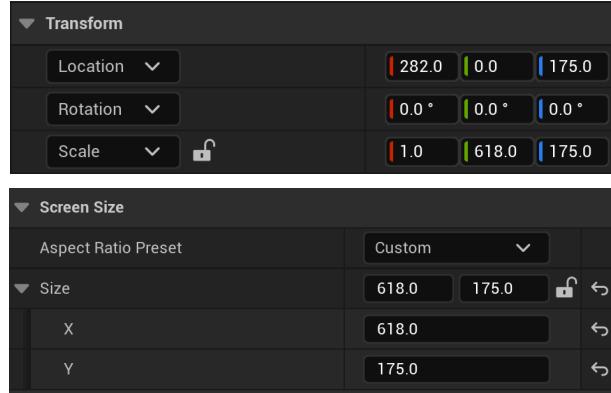


- Set the **Default ViewPoint** and **RightViewPoint**'s location to **0, 0, 175** (**5'7" ~ 175cm**). This also assumes the adult is standing at the center of their physical world.

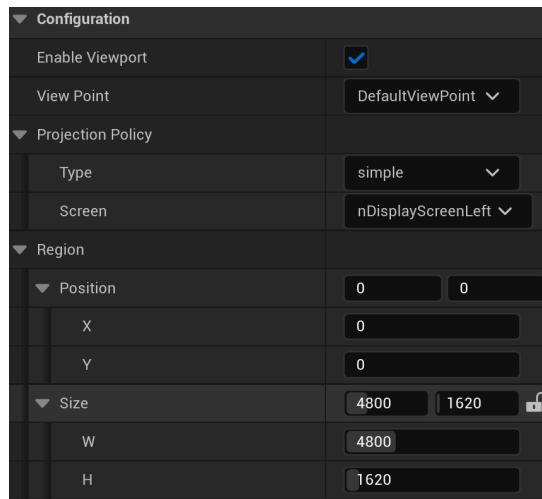


- Rename **VP_0** to **VP_Left** by right clicking on **VP_0** and selecting **Rename**.
- Create a second viewport by duplicating **VP_Left** (call it **VP_Right**).

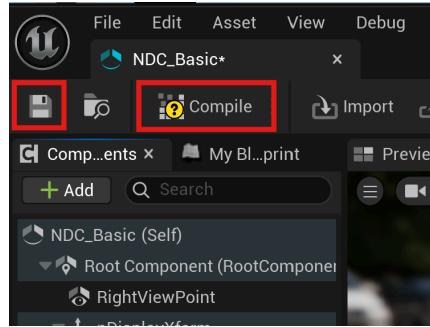
9. Now rename **nDisplayScreen** to **nDisplayScreenLeft**, and duplicate it and call it **nDisplayScreenRight**.
10. Set both **nDisplayScreenLeft** and **nDisplayScreenRight**'s **Transform** and **Screen size** to the values shown below:



11. This sets the physical screen size to 618cm x 175cm (which is 20.2' x 5.7'). The screen is also set to 282cm (9.2') in front of the viewer.
12. For **Node_0** rendering size, set it to the full stereo width (in pixels) of the display wall and the individual viewports (VP_Left and VP_Right) to half the size:
 - For Node_0 Size is 9600x1620
 - For VP_Left Size is 4800 x 1620
 - For VP_Right Size is 4800x1620
 - For VP_Right position set it to 4800x0
13. In **VP_Left** set the **Configuration/View Point** to **DefaultViewPoint**
14. Set its **Projection Policy/Type** to **Simple** and **Projection Policy/Screen** to **nDisplayScreenLeft**



15. Similarly, for **VP_Right** set the **Configuration/View Point** to **RightViewPoint** and set the **Projection policy** to **Simple**, and **Projection Policy/Screen** to **nDisplayScreenRight**.
16. After making all the edits, click on Compile and Save in the top left of the editor window. Then close the editor window.



If you're wondering what the settings for the NDC_Sim_LAVA are, Node_0 is set to 1920x1080, VP_Left and VP_Right are set to half those values.

So in summary:

Left eye: Default View Point → VP_Left → nDisplayScreenLeft

Right eye: RightViewPoint → VP_Left → nDisplayScreenRight

Part F. Caveats

Surround Screen must be enabled for proper operation

LAVA's 3D display has a physical resolution of 4800x1620. The PC outputs an extended desktop over two of the Nvidia 4080s graphics cards outputs (providing a total desktop resolution of 9600x1620), and provides it to a Colorlight Z6 LED controller that then takes those two video signals and turns them into a field sequential signal synchronized with the 3D glasses.

Here's the problem: in order for Unreal's nDisplay to display across the full 9600x1620 screen, you need to enable Surround Screen mode in Window's Nvidia controller. Otherwise the image will only fill half the screen (effectively making it a monoscopic image). So then one might ask, why not just leave your desktop with surround screen permanently enabled? The problem is, if you maximize a regular desktop app window, it will extend beyond the extent of the physical screen making the right half of the window difficult to access.

So then, in order to resume normal use of the desktop for regular Windows applications you need to take the graphics card out of surround screen mode.

It appears Unity does not have this limitation and can work in both Surround Screen and non-Surround Screen mode.

Packaged nDisplay Application is a todo

The above tutorial assumes you will always run your nDisplay application from your Unreal editor. It is possible to build the application to a package and run the executable. I have not had time to explore this. Suffice it to say, in order to build to an executable you will need to first install Microsoft Visual Studio. But note: when you do attempt to compile with it, you have to make sure UnReal uses Visual Studio 2022. At a later date I will put out some instructions on how to do this.