



UNREAL SETUP GUIDE

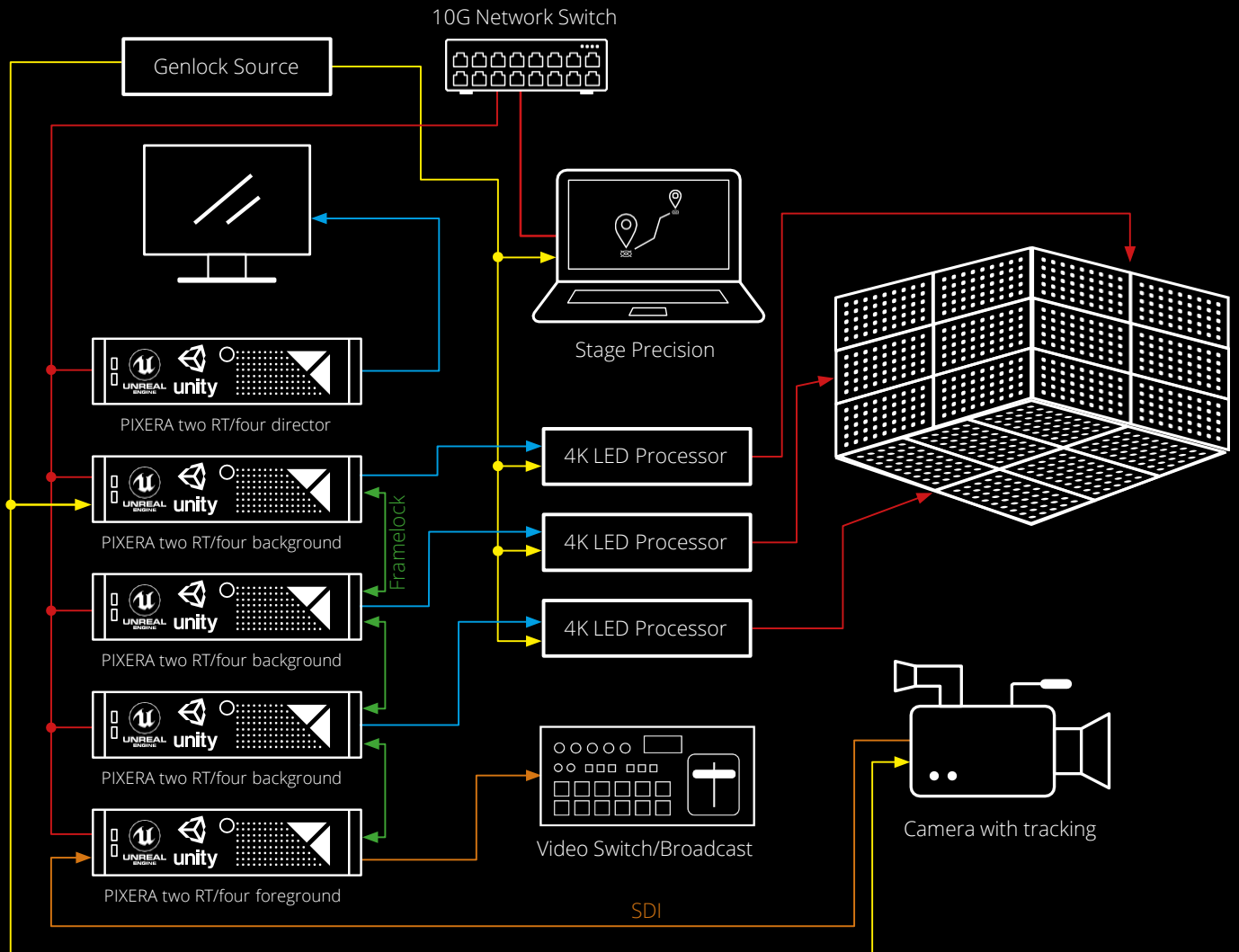
PIXERA 1.8.92

CONTENT

System Design	3
1. How to prepare your UNREAL project for PIXERA	4
1.1 Installation of UNREAL 4.27.2.....	4
1.2 Create a project in UE4editor	6
1.3 System preparation	8
1.3.1 PIXERA - UNREAL Plugin for object movement.....	8
1.3.2 Package UNREAL Project for PIXERA.....	9
1.3.3 UNREAL Project in Dash Game Mode.....	10
1.3.4 UNREAL Project in Multiuser Mode.....	11
1.4 UNREAL Texture sharing.....	13
2. Virtual Production setup in PIXERA	22
2.1 Foreword.....	22
2.2 XR Stage in PIXERA	23
2.3 Importing Unreal Projects.....	24
2.4 Screen presence and nDisplay Config File	26
2.5 Moving objects in PIXERA.....	29
3. STUDIO CAMERA SETUP	30
3.1 Studio Camera Settings	31
3.2 Virtual Set Extension.....	32

SYSTEM DESIGN

This chart shows a proper technical setup in a Virtual Production environment:



1. HOW TO PREPARE YOUR UNREAL PROJECT FOR PIXERA

1.1 INSTALLATION OF UNREAL 4.27.2

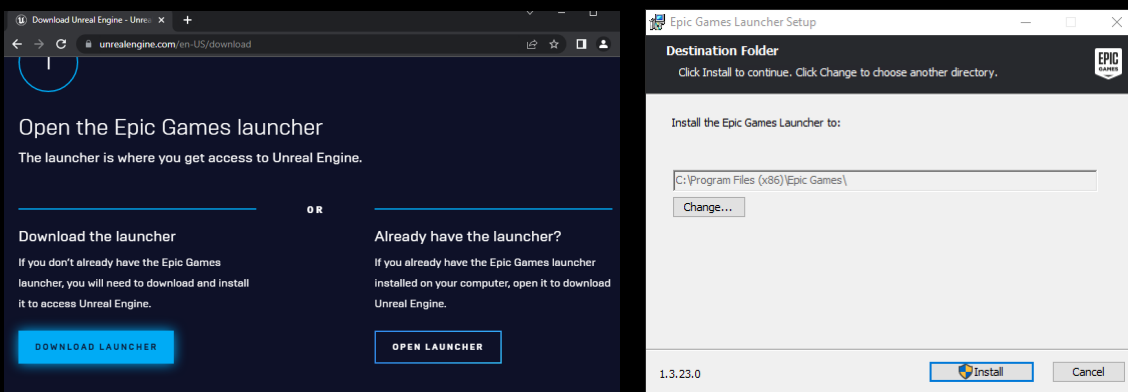
1) What you need:

- ✚ Unreal Engine 4.27.2 on Master and Clients
- ✚ pixera_unreal_engine_interop_4.27.2_P.exe Installer
- ✚ PIXERA_Connect_1.9_4.27_P.zip Plugin
- ✚ .NET 4.8 SDK
(<https://dotnet.microsoft.com/download/dotnet-framework/thank-you/net48-developer-pack-offline-installer>)

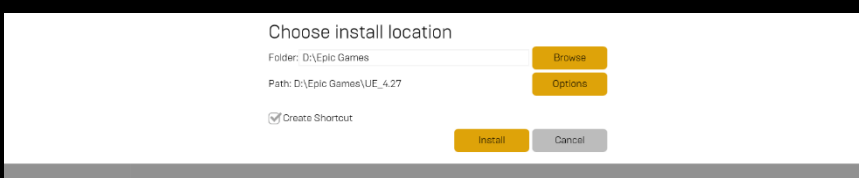
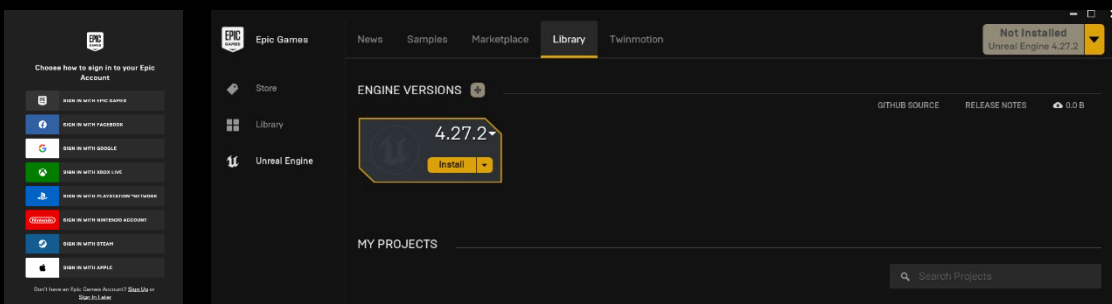
If you want to pack your project:

- ✚ Visual Studio 2019 (<https://visualstudio.microsoft.com/de/downloads/> Community License is fine)
- ✚ Full Unreal Engine form AV Stumpf on request (Unreal_Engine_4-27_O.zip)

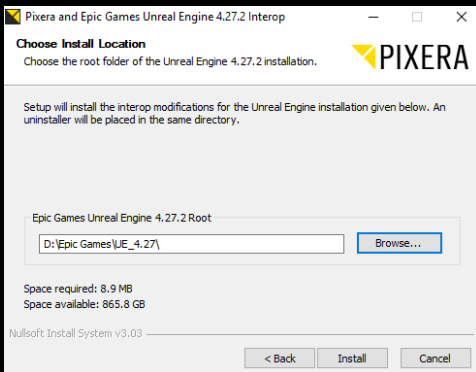
2) Download and install Epic Games launcher



3) Login and install unreal Engine



5) Install pixera_unreal_engine_interop_4.27.2_P.exe Installer



Please make sure to select the right directory

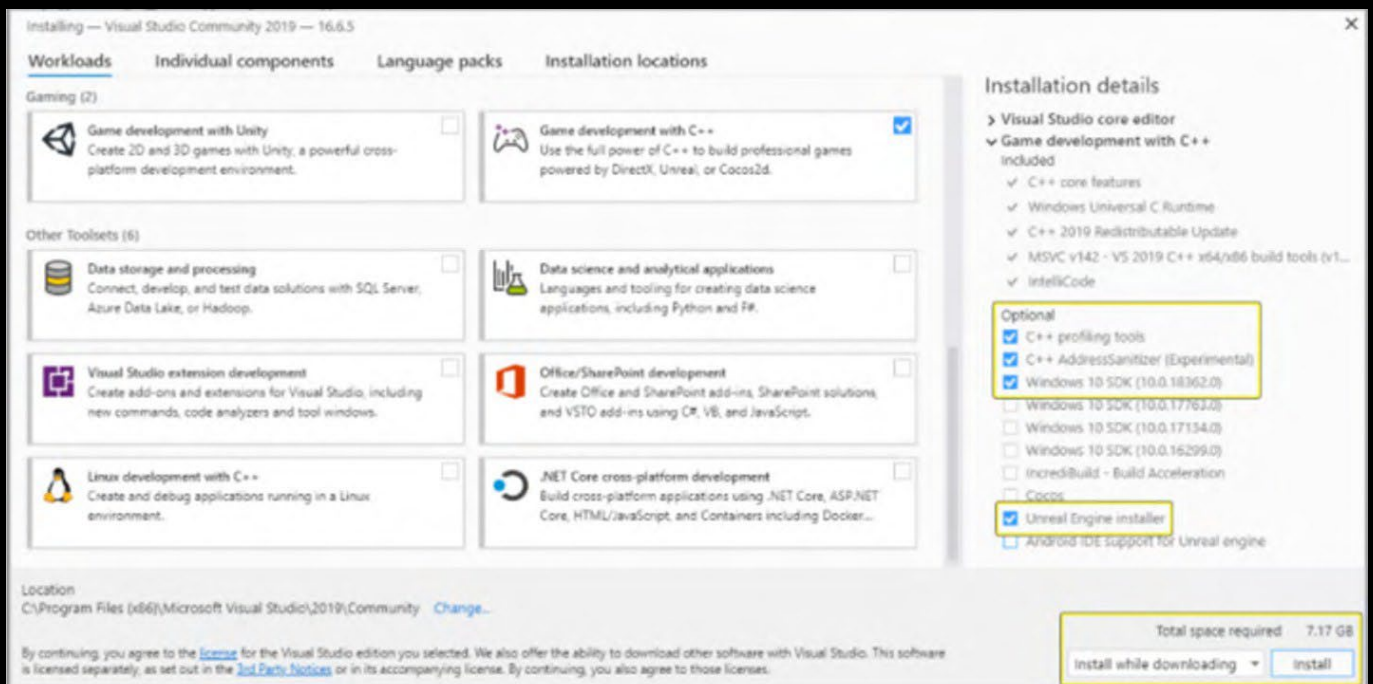
6) Install .NET

Optiona for packed projects:

7) Unzip the Engine in your preferred folder.

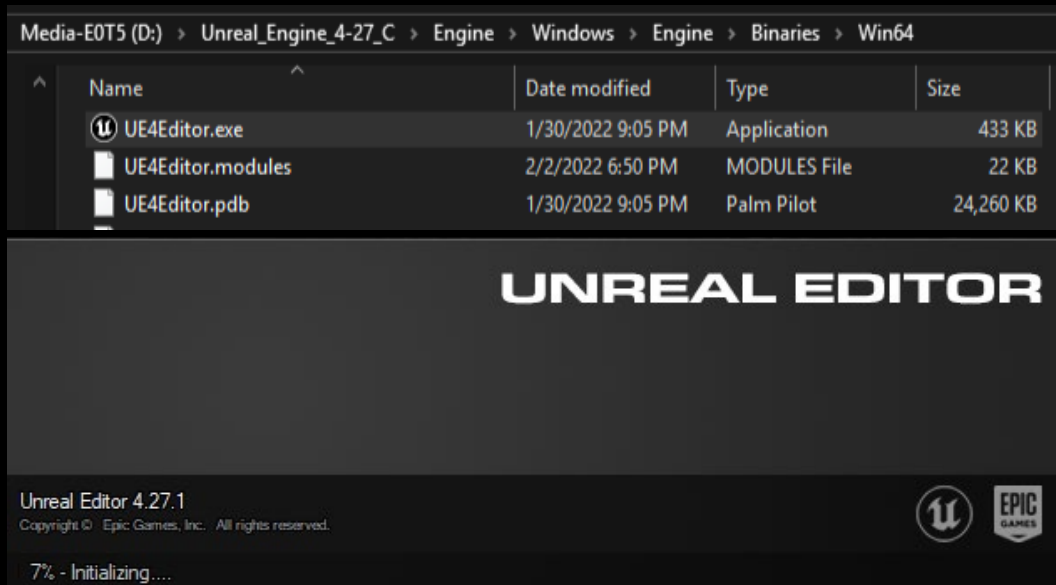
8) Install Visual Studio including following Options (only necessary if you want to pack a project):

- Add **Game development with C++** (under Workloads) and following options:
 - C++ profiling tools
 - C++ AddressSanitizer
 - Windows 10 SDK (10.0.18362 or newer)
 - Unreal Engine Installer



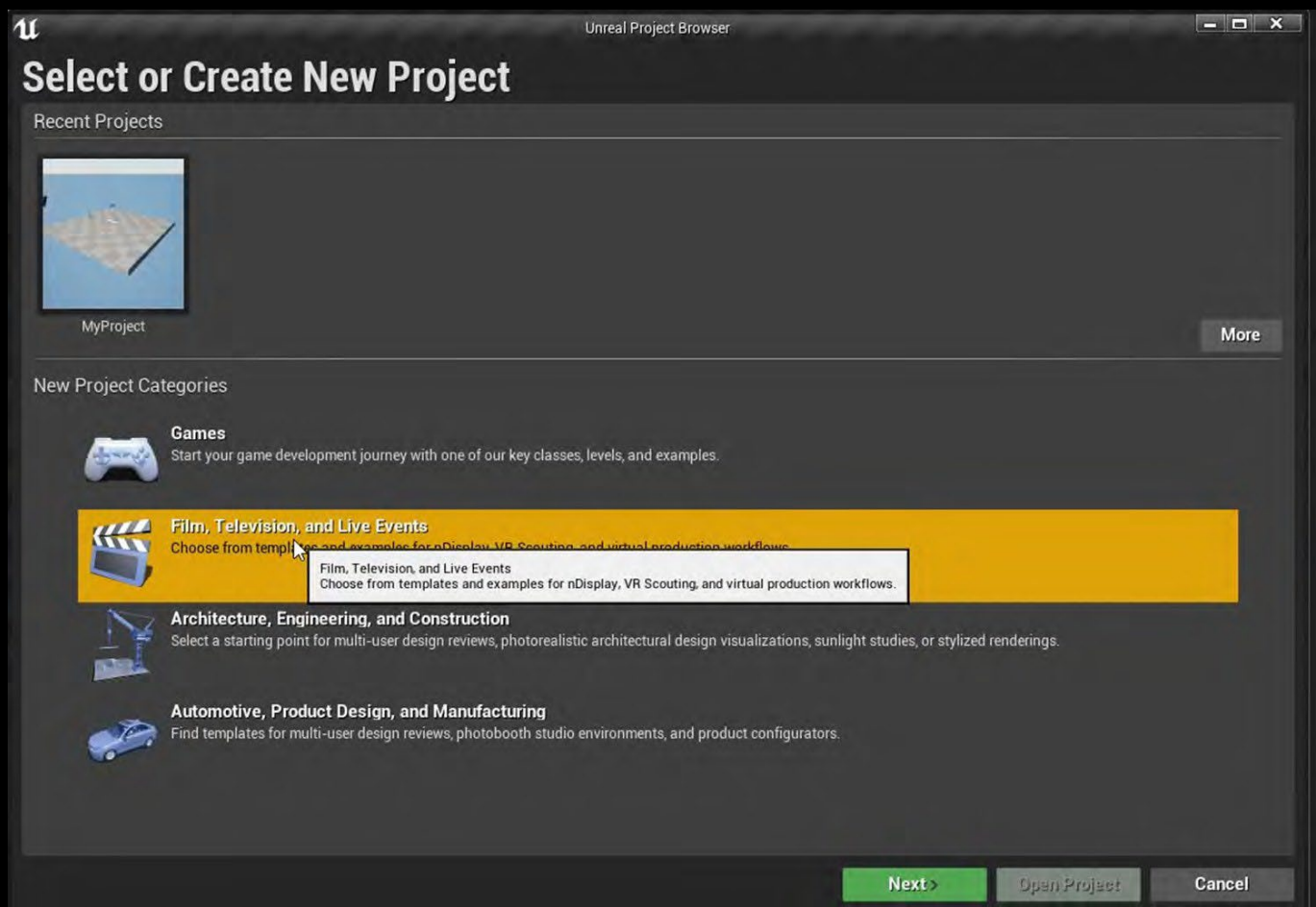
1.2 CREATE A PROJECT IN UE4EDITOR

Open UE4Editor.exe (Unreal_Engine_4-2D_C\Engine\Windows\Binaries\Win64\)

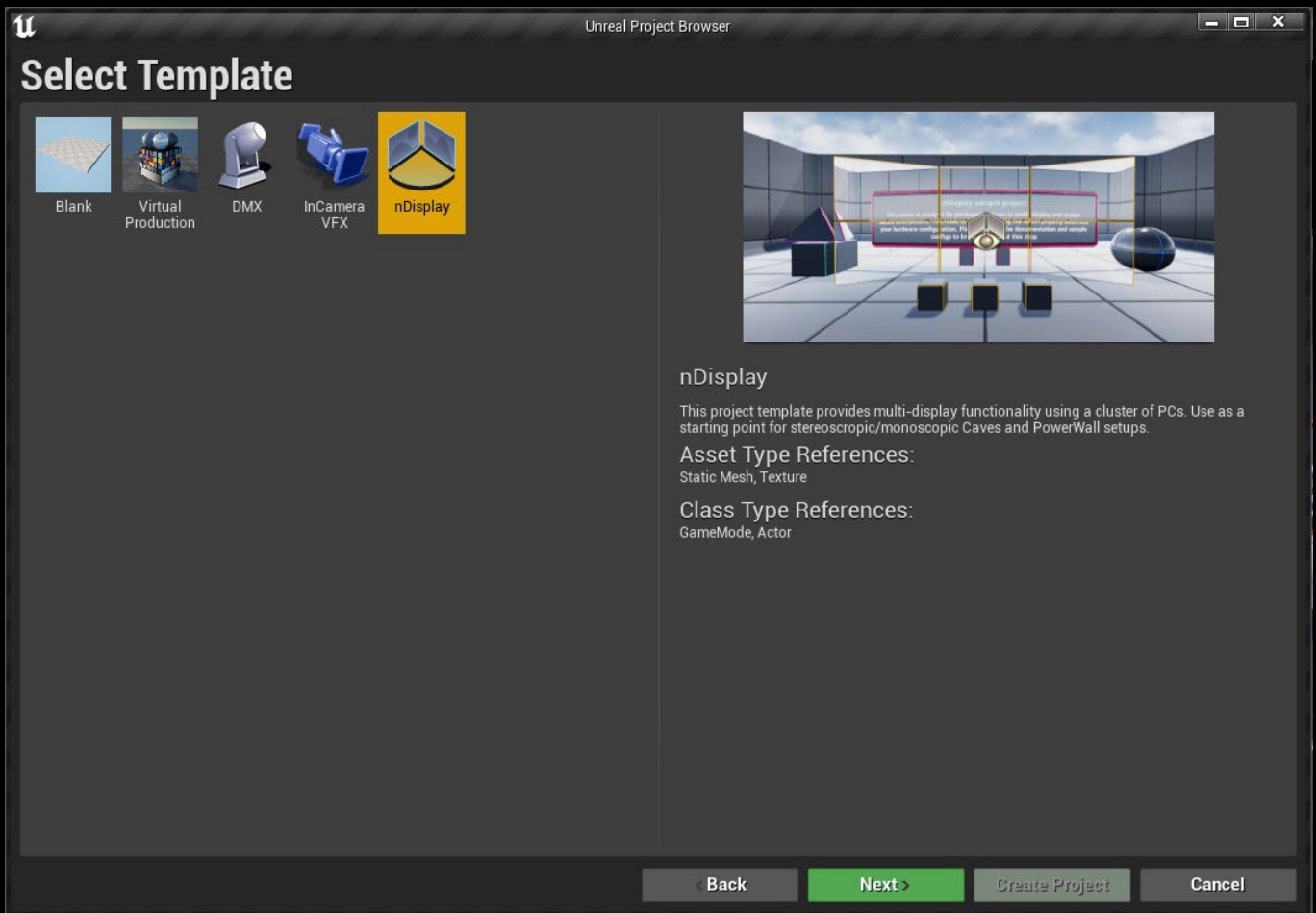


NOTE: Stay patient – first startup will take some minutes...

Select from New Project Categories „Film, Television, and Live Events“



Click next, select the „nDisplay” template and click next again.



Select your location and click on „Create Project”.



The first initialization will take some minutes...

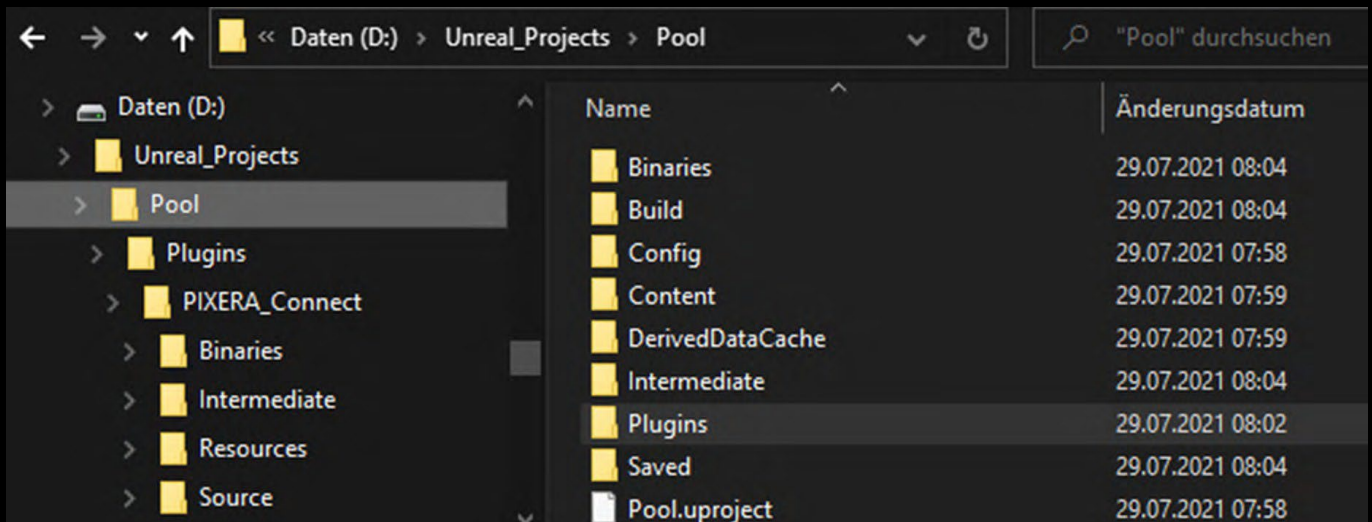
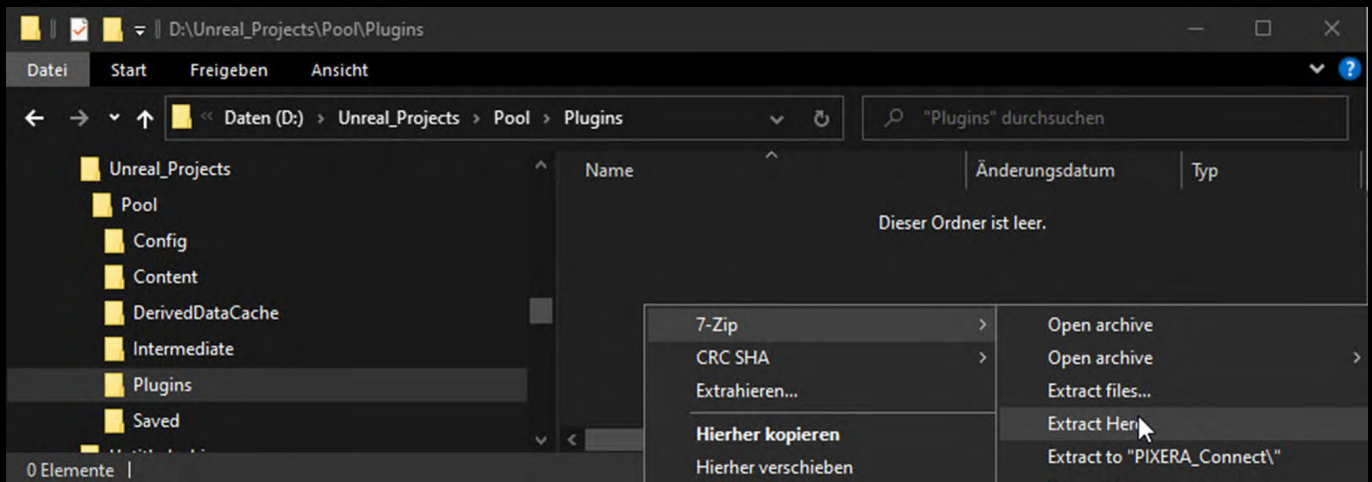
1.3 SYSTEM PREPARATION

1.3.1 PIXERA - UNREAL PLUGIN FOR OBJECT MOVEMENT

To be able to move objects of the Unreal scene directly in PIXERA you have to copy the **PIXERA_Connect** plugin to its correct directory:

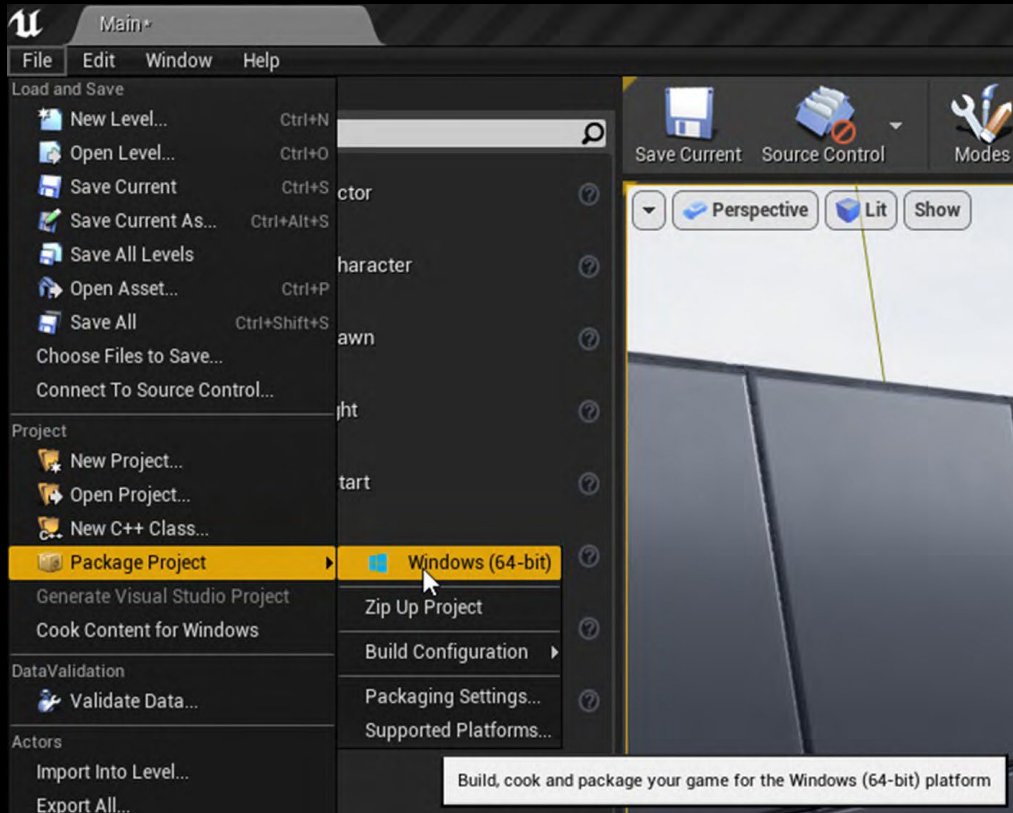
Create a folder called „Plugins“ next to your project file (D:\Unreal_Projects\Pool\Plugins)

Extract **PIXERA_Connect_1.9_4.27_P.zip** directly (without subfolders) to created Plugins folder.



1.3.2 PACKAGE UNREAL PROJECT FOR PIXERA

To package the project in unreal click on **File → Package Project → Windows (64-bit)** and select the directory where you want to pack it. (i.e. D:\Unreal_Projects\Pool_export)



1.3.3 UNREAL PROJECT IN DASH GAME MODE

To run your Unreal project in dash game mode there are no further steps in Unreal necessary. Just import your project, assign your screens, create nDisplay config, distribute to clients and copy RXInterop.dll. Then you are already able to run you project.

1.3.4 UNREAL PROJECT IN MULTIUSER MODE

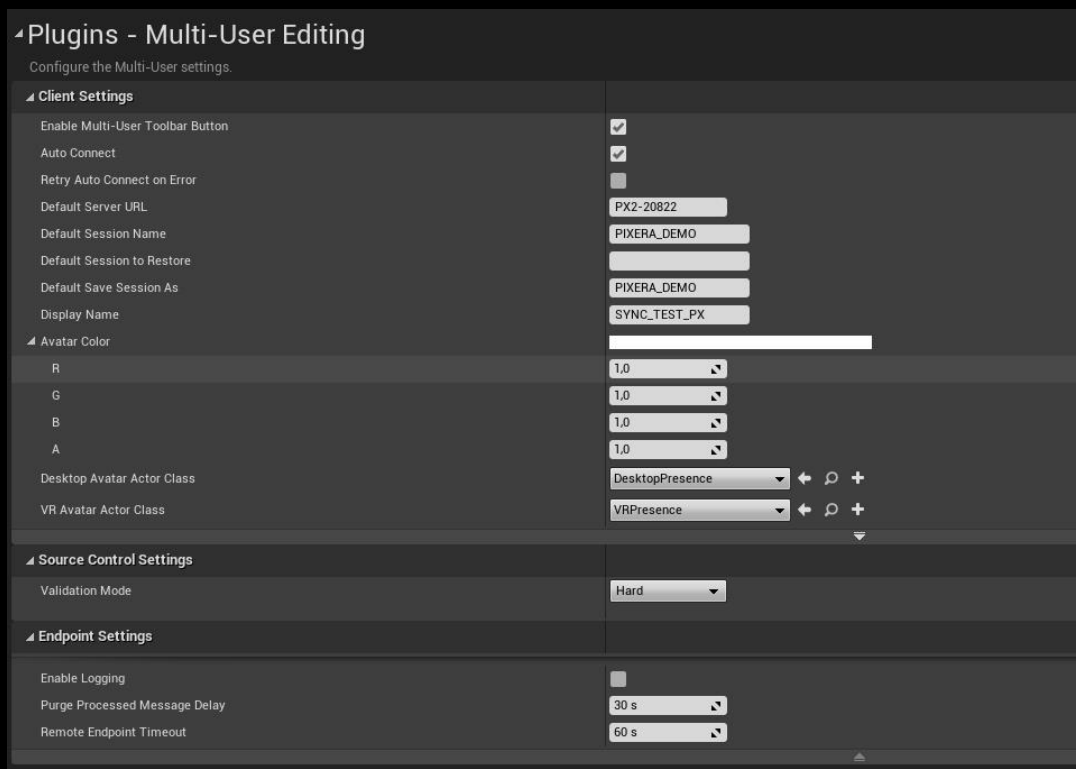
In Unreal the multi-user plugin has to be enabled.



After saving and restarting your Unreal scene, navigate to „Project Settings“ (Edit → Project Settings..), navigate to „Plugins“ → „Multi-User Editing“.

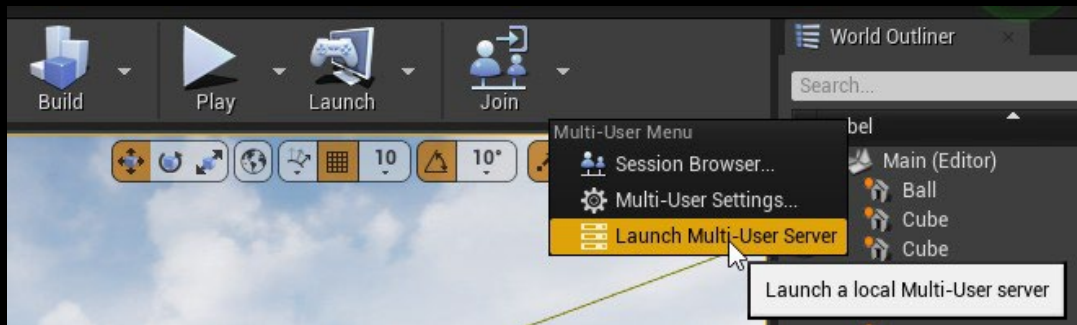
“**Enable Multi-User Toolbar Button**” and “**Auto Connect**” have to be enabled.

Further please enter the Windows computer name of your Unreal Server in “**Default Server URL**” and assign a name for “**Default Session Name**”.



Save your project and re-open it.

Click on the arrow next to „Join“ in the toolbar and select **Launch a local Multi-User server**.

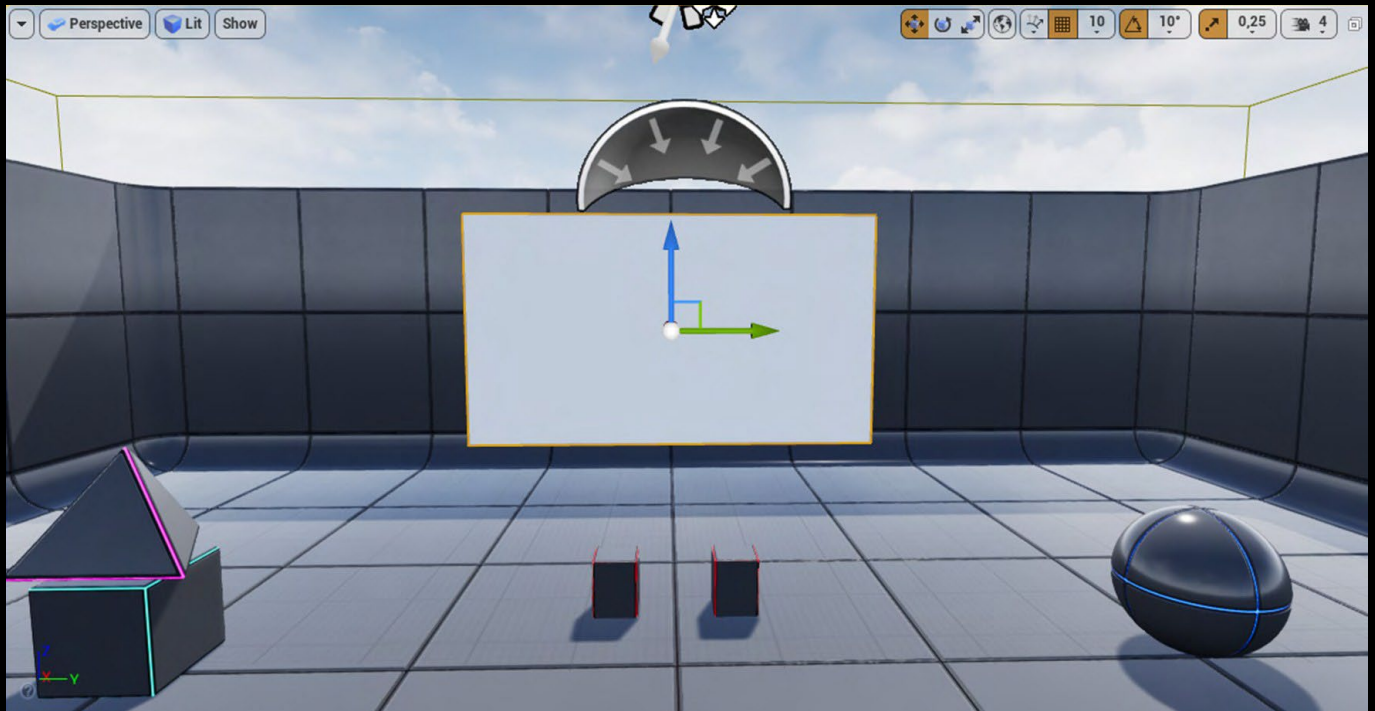


A new window pops up, afterwards close Unreal again.

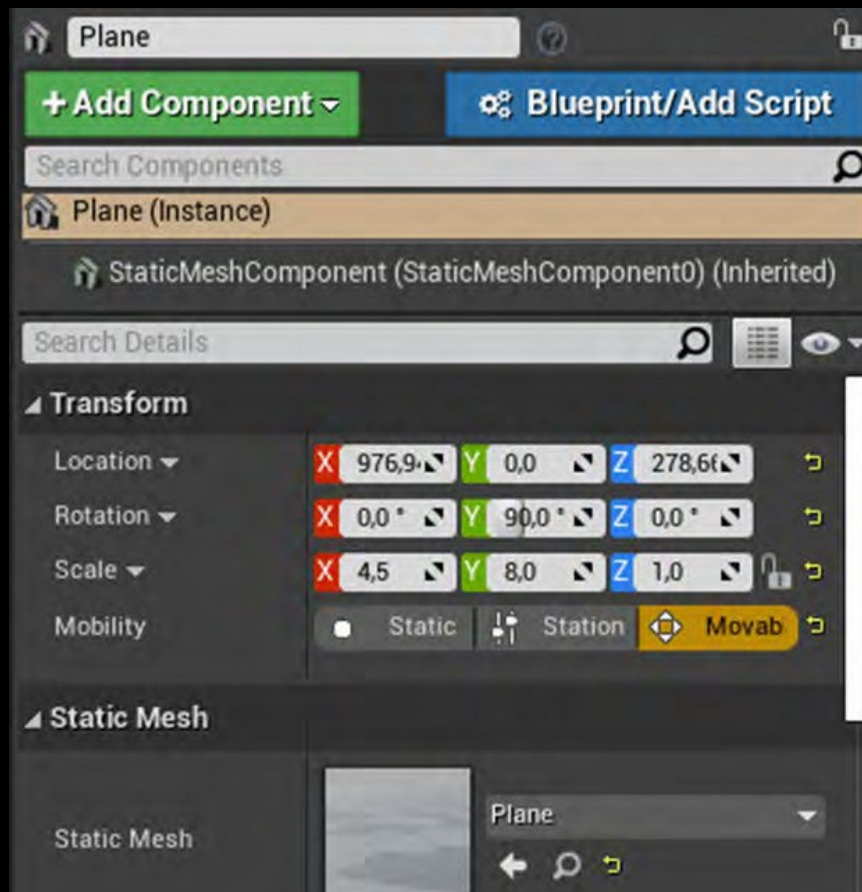
Now you can import the project to PIXERA - please find instruction in the paragraph **Import to PIXERA**

1.4 UNREAL TEXTURE SHARING

Add a plane (can be another shape as well) to your project and place it wherever you like.

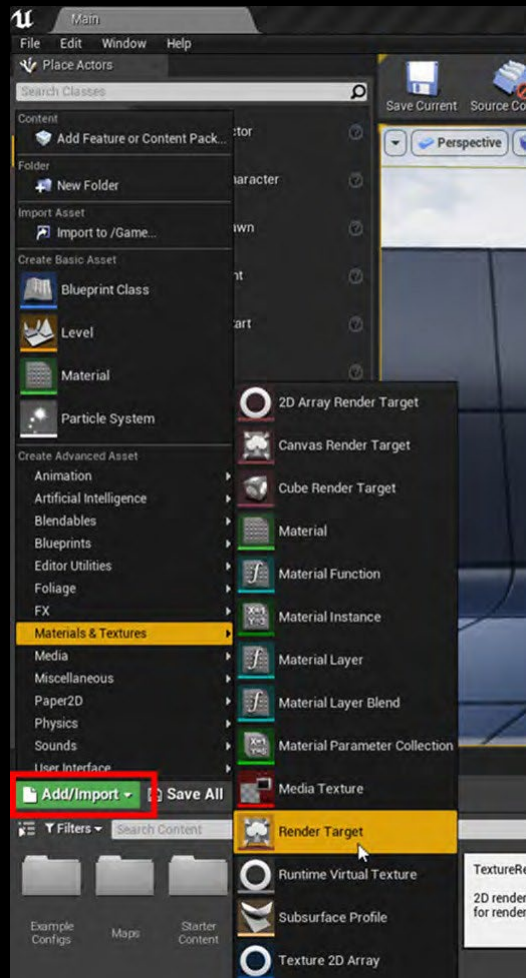


If you want to control position out of PIXERA, set it movable.

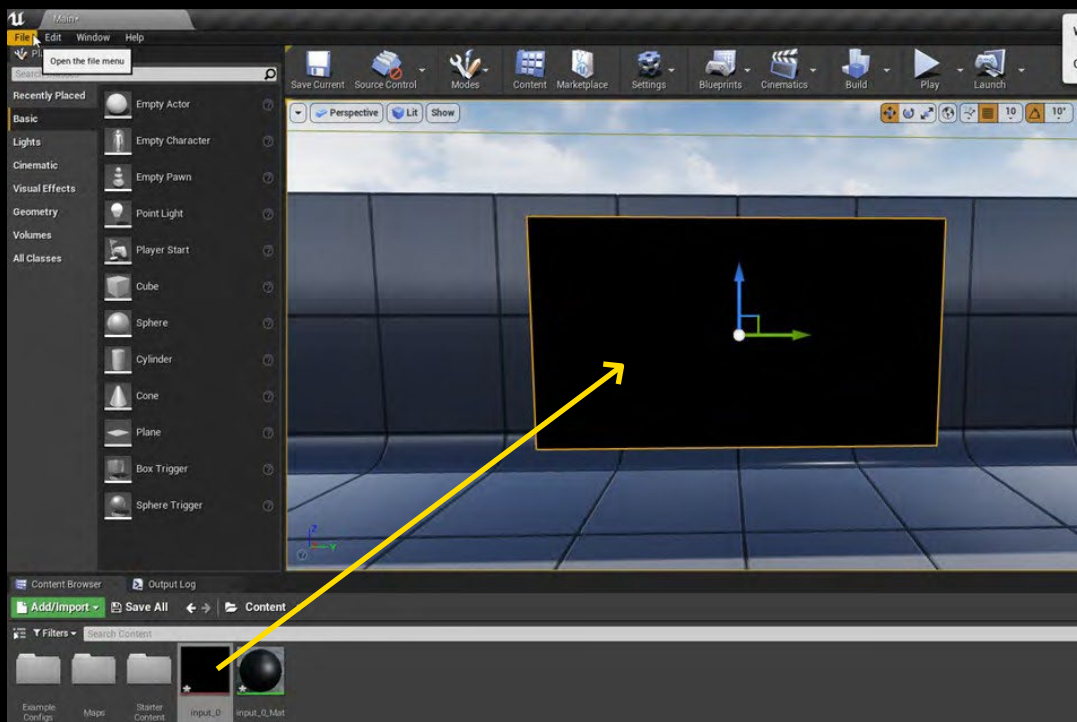


Add a render Target by clicking on **Add/Import** and navigate to **Material & Textures – Render Target**.

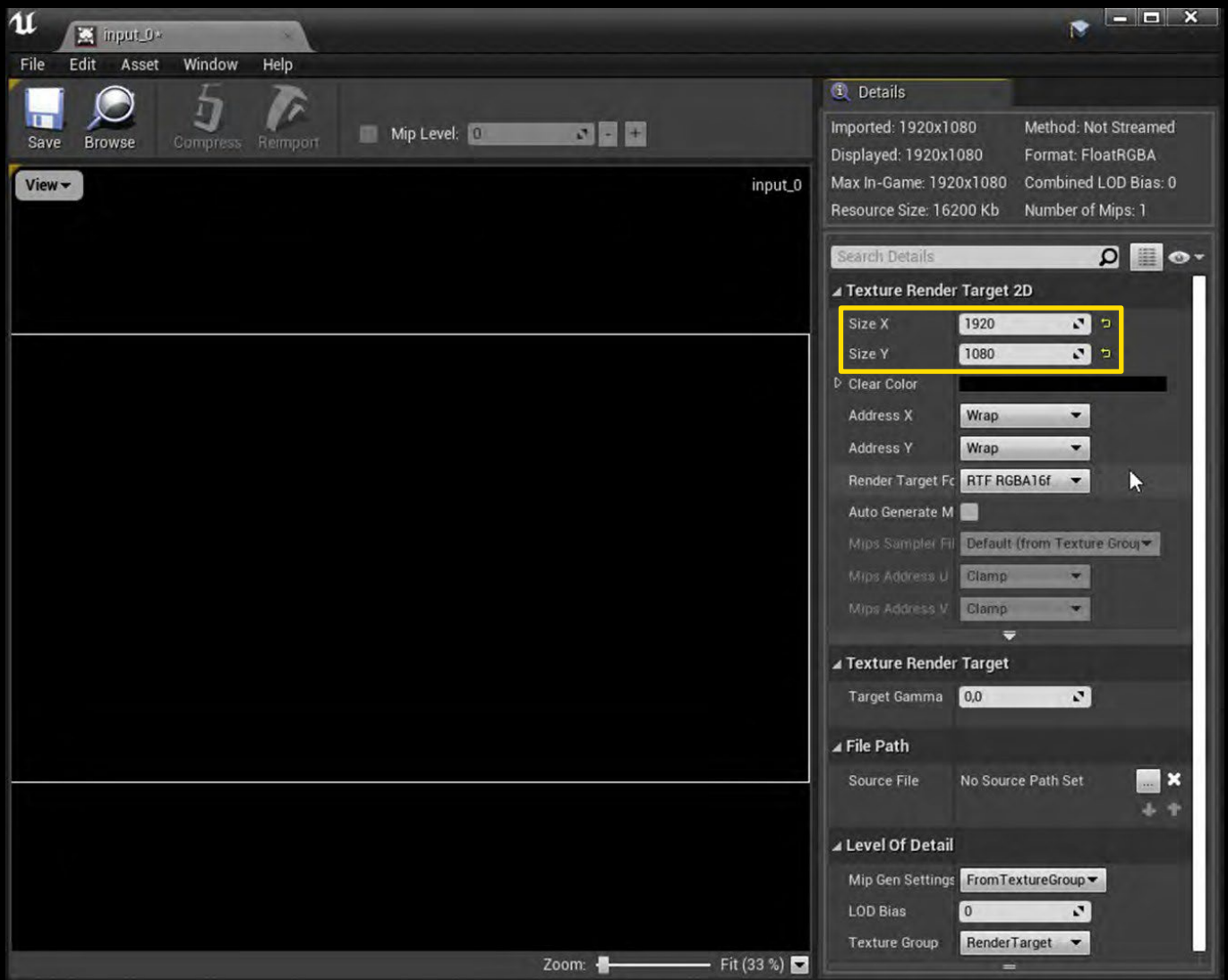
Rename it if you want.



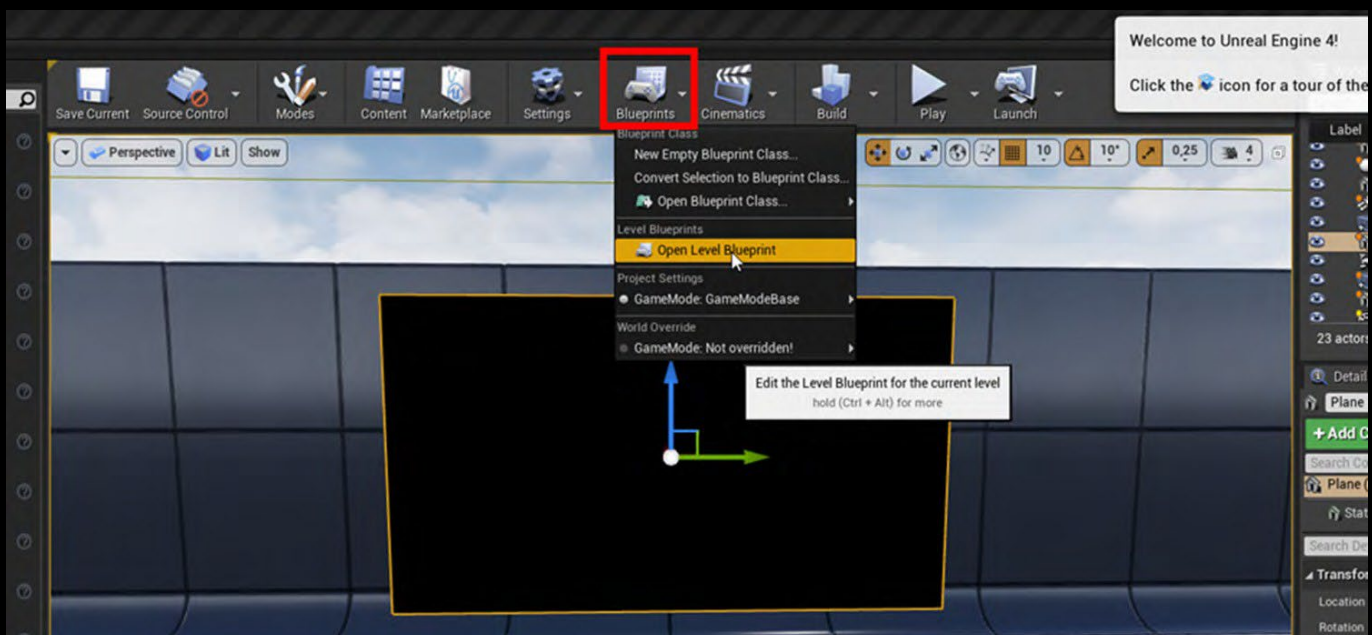
Drag and drop the created render target to your plane. A new material will be created automatically.



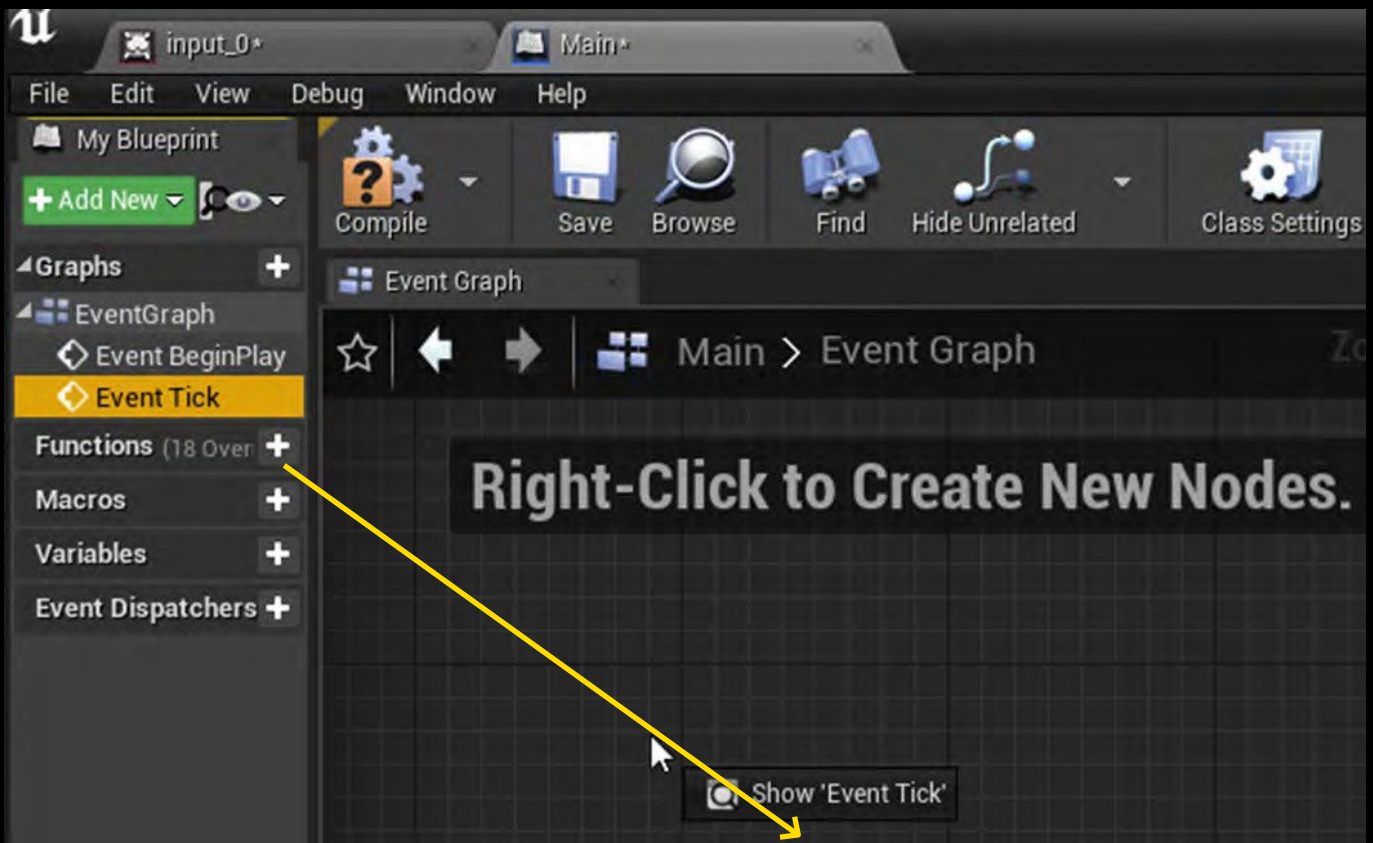
Double click on the created render target to adjust the resolution - **Save**.



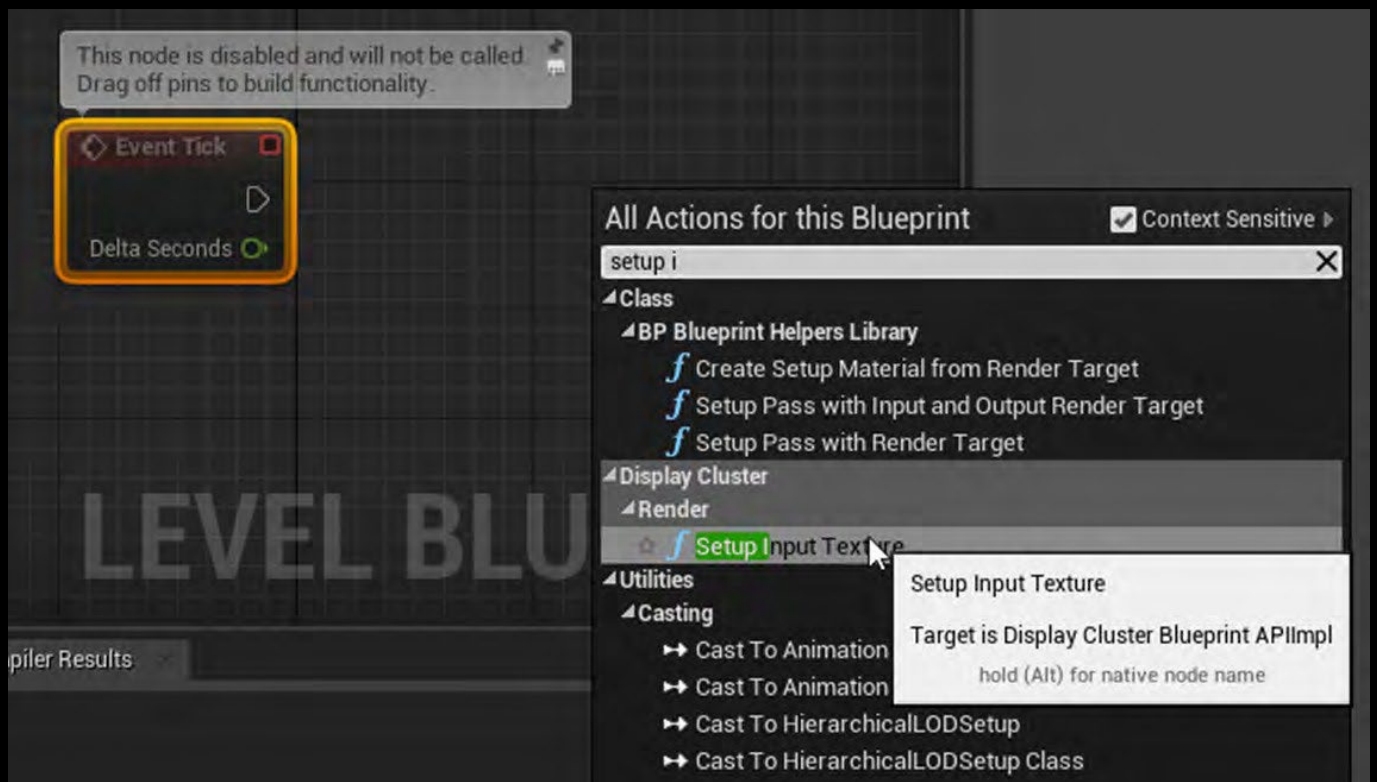
Open **Blueprints - Level Blueprint**.



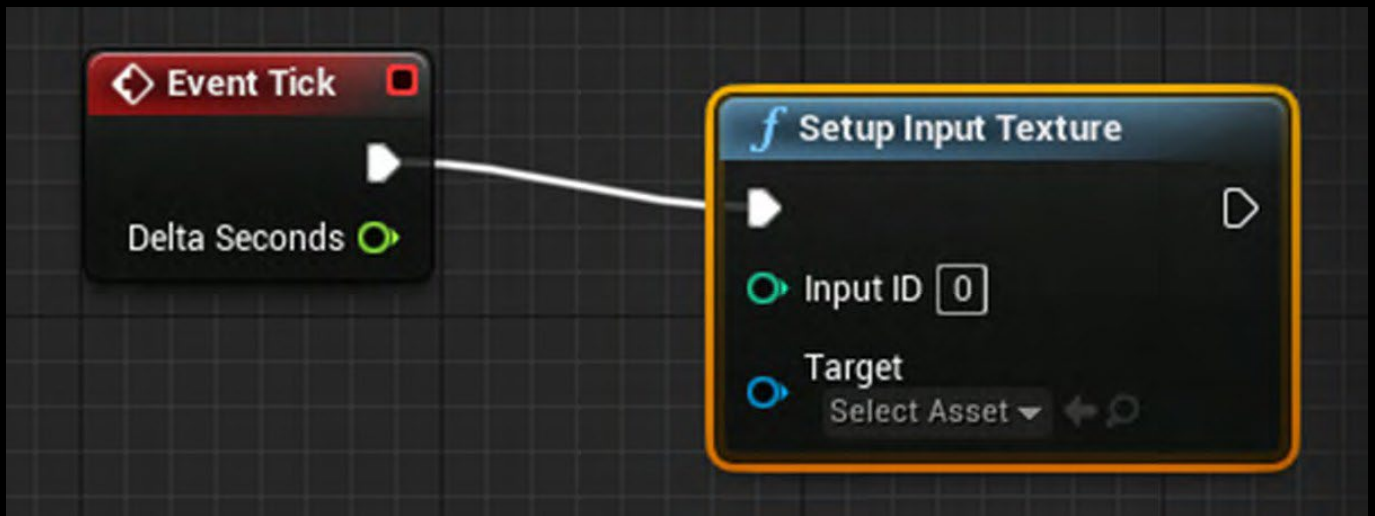
Add **Event Tick** from left hand site to your Event Graph (workspace in the center).



Right click in the event graph and add **Setup Input Texture**.

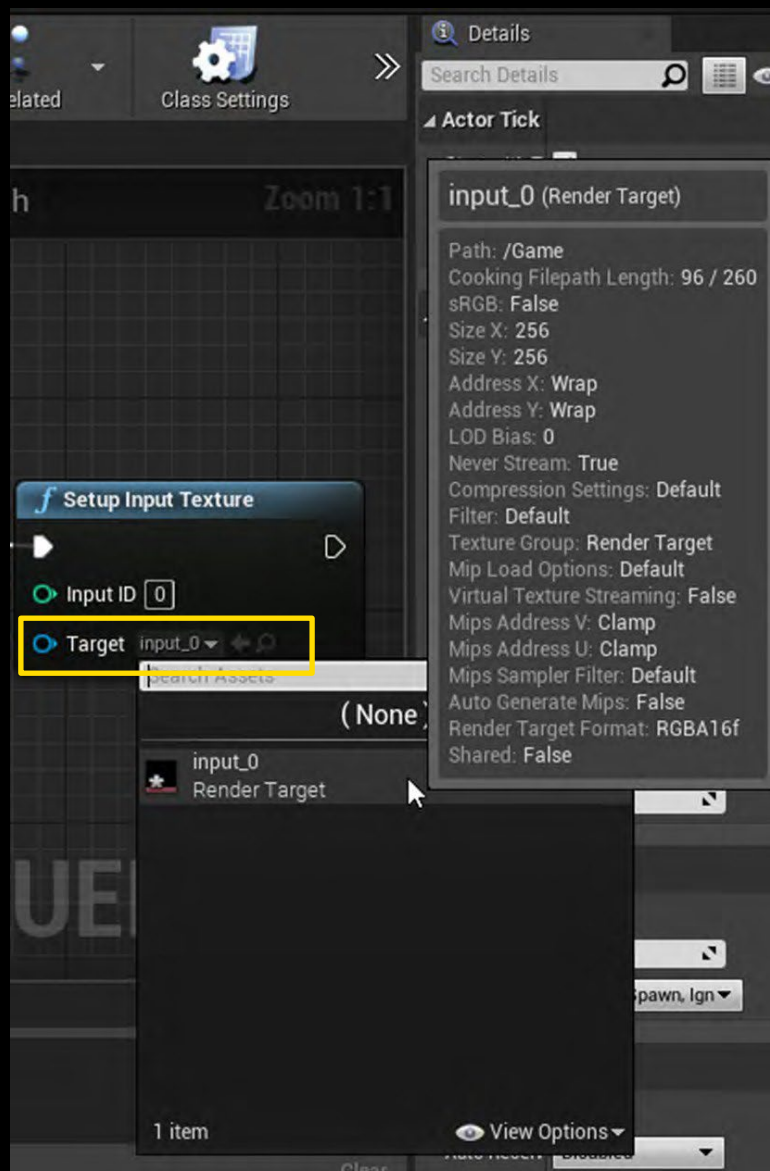


Connect those two.

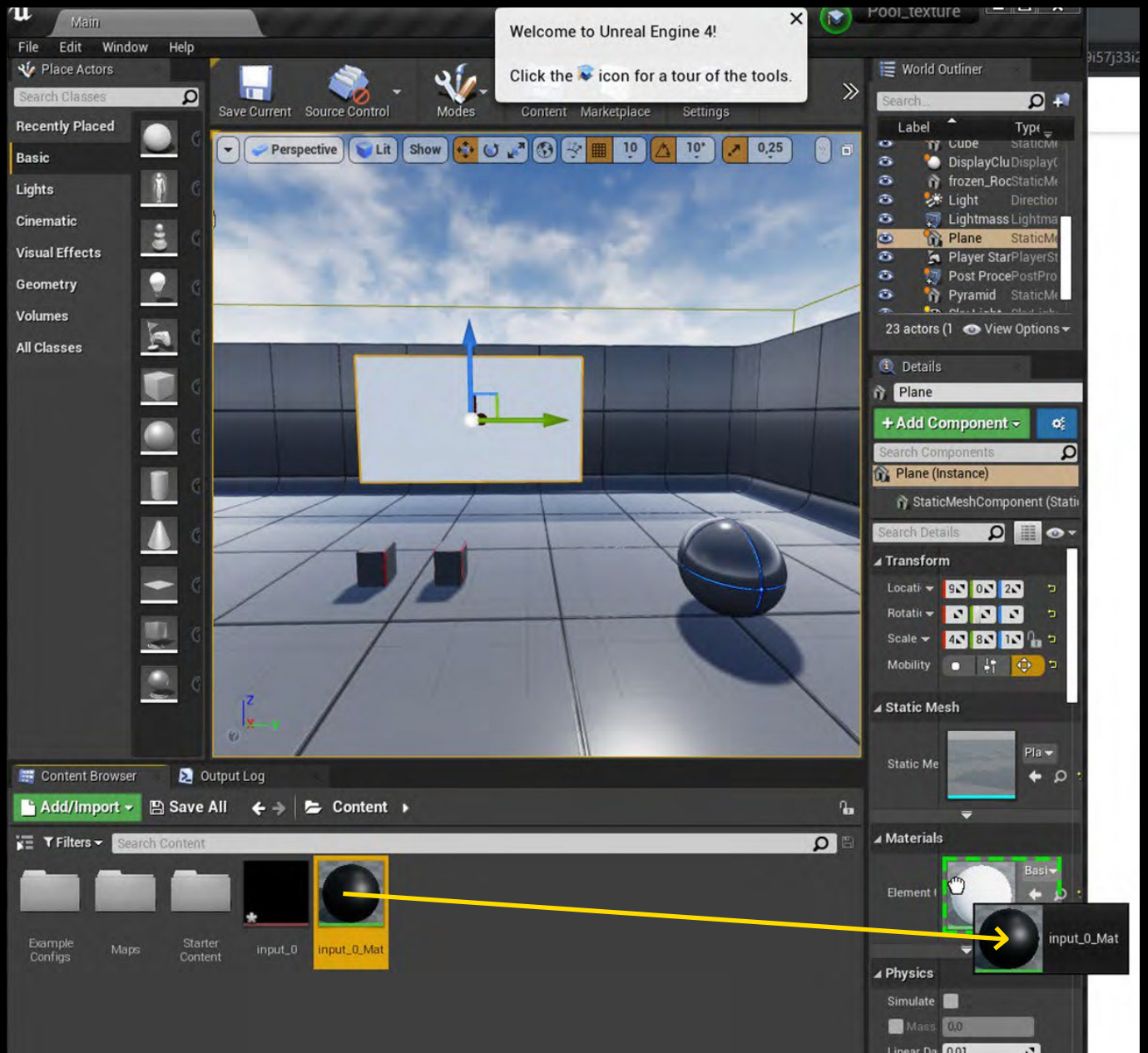


NOTE: The Input ID in "Setup input Texture" refers to sampler ID in PIXERA's PUX file.

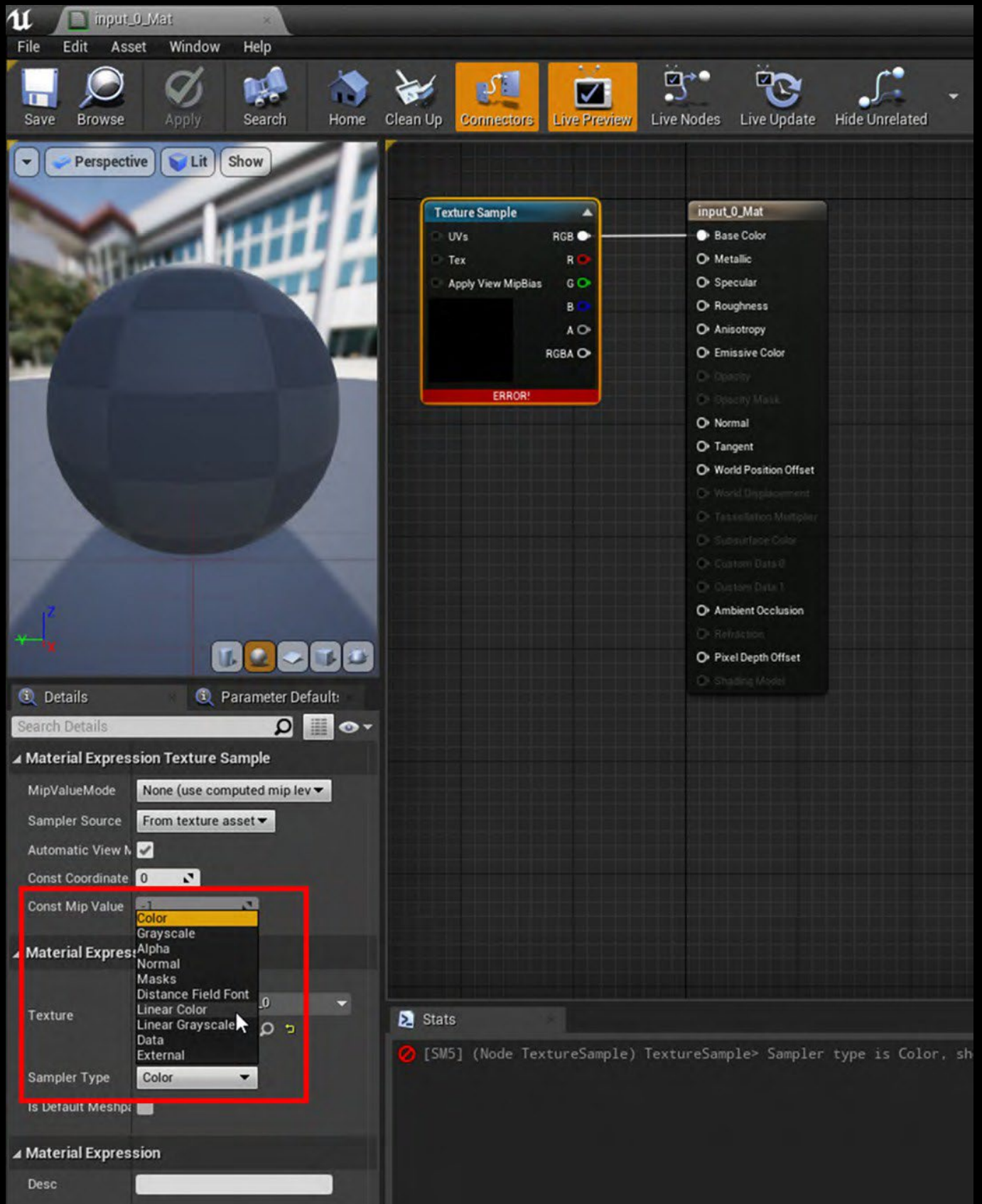
Select your created render target and **Save** it



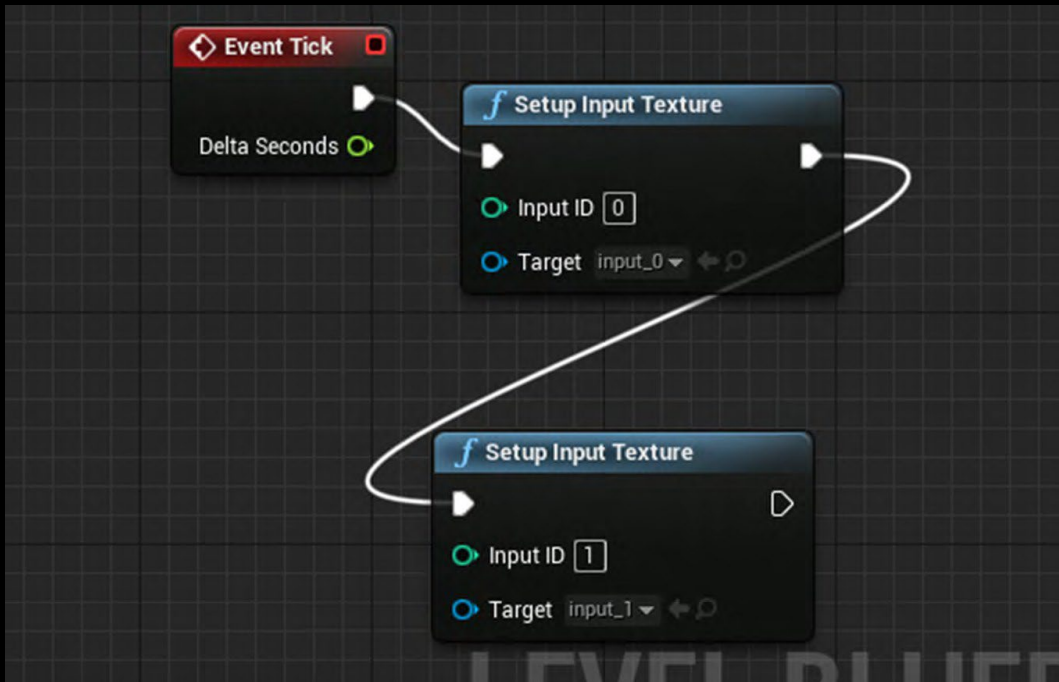
Drag and drop the material into the material in the pane properties on the right:



Double-click on the material, select the **Texture Sample** and change the Sampler type to **Linear Color**. Save afterwards. If you want the plane to emit color as well, you can link RGB output of **Texture Sample** to **Emissive Color** input of the material as well.

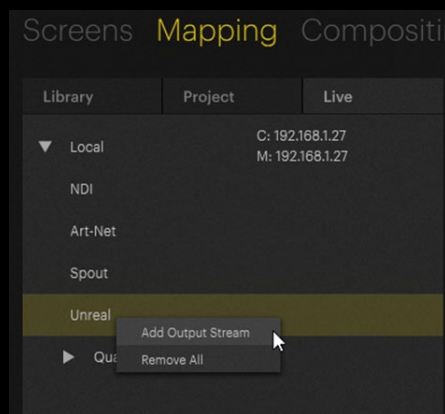


If you want to have two targets to share, repeat the steps and connect them as shown below:

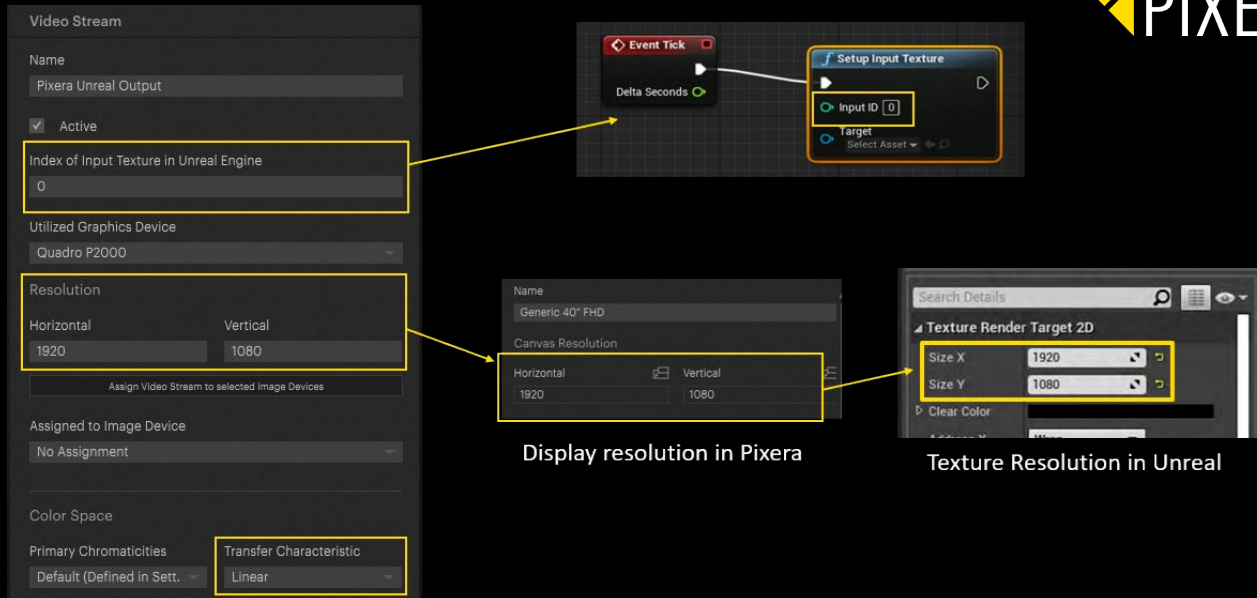


After importing your Unreal project in PIXERA you have to add a Display with the resolution of your created texture in Unreal.

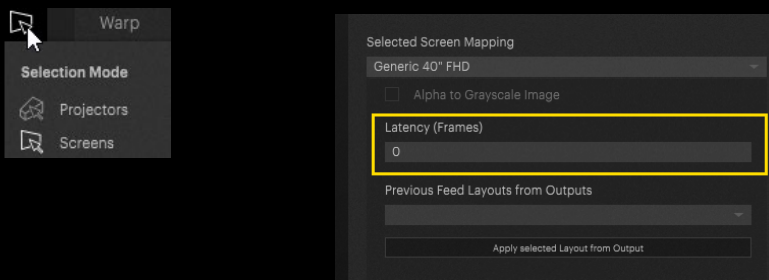
- Please use a separate Screen Group for this Display.
- Further you have to add an Unreal Output Stream for each client you want to display the shared texture.



- All those outputs have to be assigned to the newly added Display and the resolutions have to be the same in Pixera (Display resolution and Unreal Output resolution) and Unreal Texture.
- Index of Input Texture in Unreal Engine refers to the Input ID in Unreal.
- **Transfer Characteristics** have to be set to Linear.



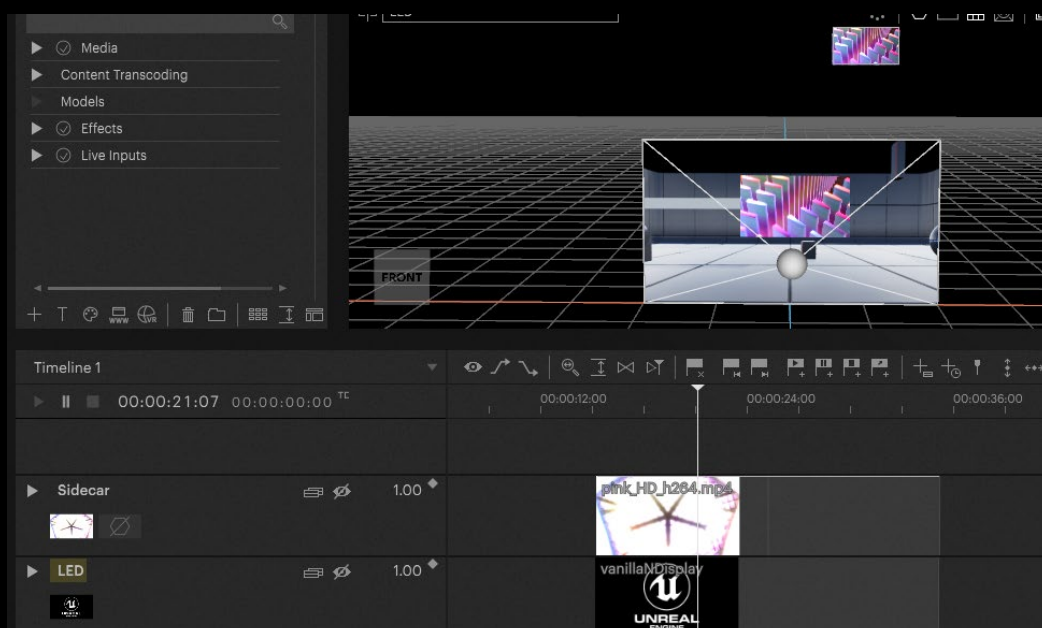
- For set extension or foreground you can add a additional Latency to the specific output in the Inspector by selecting the Display in Mapping Tab (change selection mode to be able to select a Display).



After importing the Unreal asset Pixera automatically streams the content of the Display into Unreal.

Just assign a layer to the display with the assigned "Unreal Output Stream" and start to add content onto it. The whole compositing of this display is then visible on the shared texture in Unreal

NOTE: In this example, the layer used was named "Sidecar" - but it is a standard layer, no special naming conventions are needed.



2. VIRTUAL PRODUCTION SETUP IN PIXERA

2.1 FOREWORD

Virtual Production features are available for official PIXERA hardware as well as the Server license. It is not available for the Director and Player license.

By using the newest PIXERA Version, it is possible to utilize Unreal Engine, Unity as well as Notch in PIXERA for virtual production setups and other projects that would benefit from these powerful tools.

This workflow Guide focuses on the technical setup of Unreal Engine in PIXERA. It is a step by step guide. As long as you follow everything, you should be able to get a test setup running.

Here is a list of features we were able to test and can confirm they are working properly:

- ✔ Unreal Engine Preview
- ✔ Unreal Engine Payout scaled over 3 servers in sync
- ✔ Unity Preview
- ✔ Unity Payout (currently we do not have any sync mechanism implemented for Unity. We are investing further R&D resources and will post regular updates if sync is possible, as there might be a chance)
- ✔ Notch Preview & Payout (Notch currently only supports one Viewport Render, therefore realising a setup is tricky and requires utilization of Camera and back projection)
- ✔ Stage Precision direct API and SPNet tracking integration with the Studio Camera feature in PIXERA. Tracking with Vive, OptiTrack, FreeD, Antilatancy, Sytpe, Mo-Sys, is tested so far

The following applications are necessary for a proper Unreal Engine setup in PIXERA:

- ✔ Pool_texture_1.5.exe (or any other packed Unreal project)
- ✔ PIXERA 1.8.45 or later

Please note that it is not necessary anymore to launch nDisplay Listener or Launcher and to copy Rxinterop.dll manually.

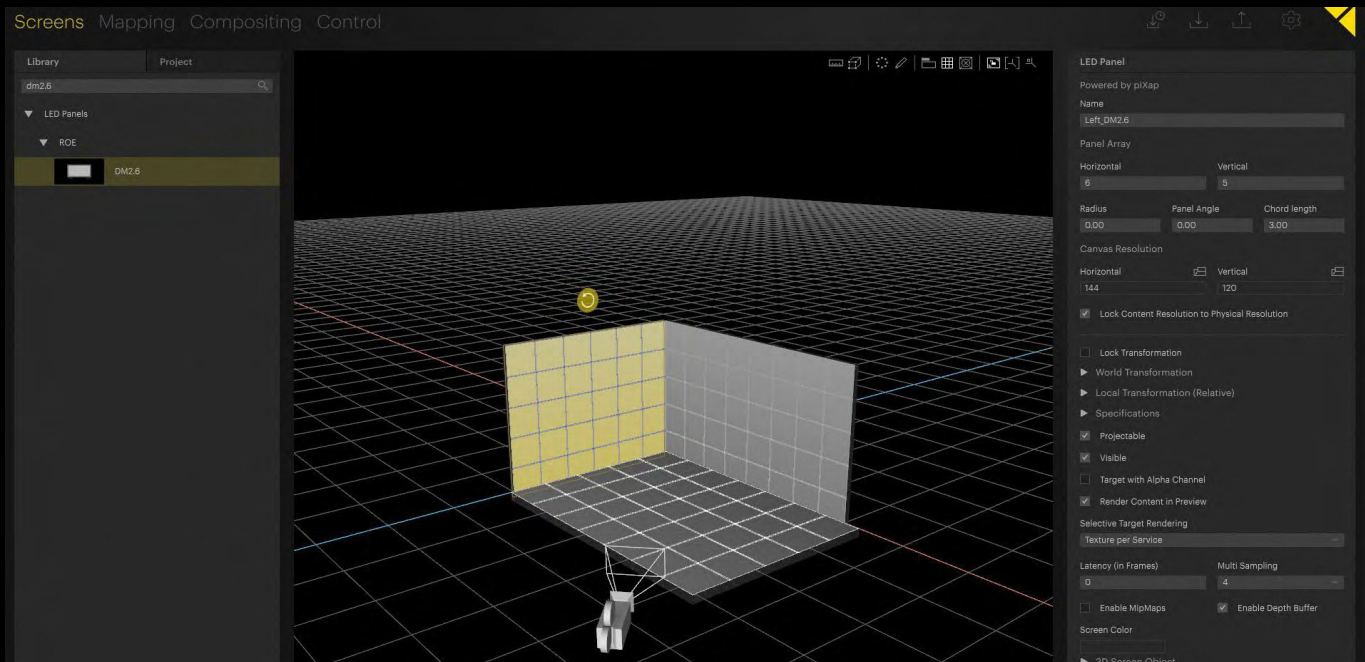
2.2 XR STAGE IN PIXERA

The steps are explained on a simple setup.

In this example we use for the left LED a ROE DM2.6 in a 6x5 panel array, the right LED is a ROE DM2.6 as well in a 8x5 array and for the floor we use a ROE BM4 in a 7x5 array.

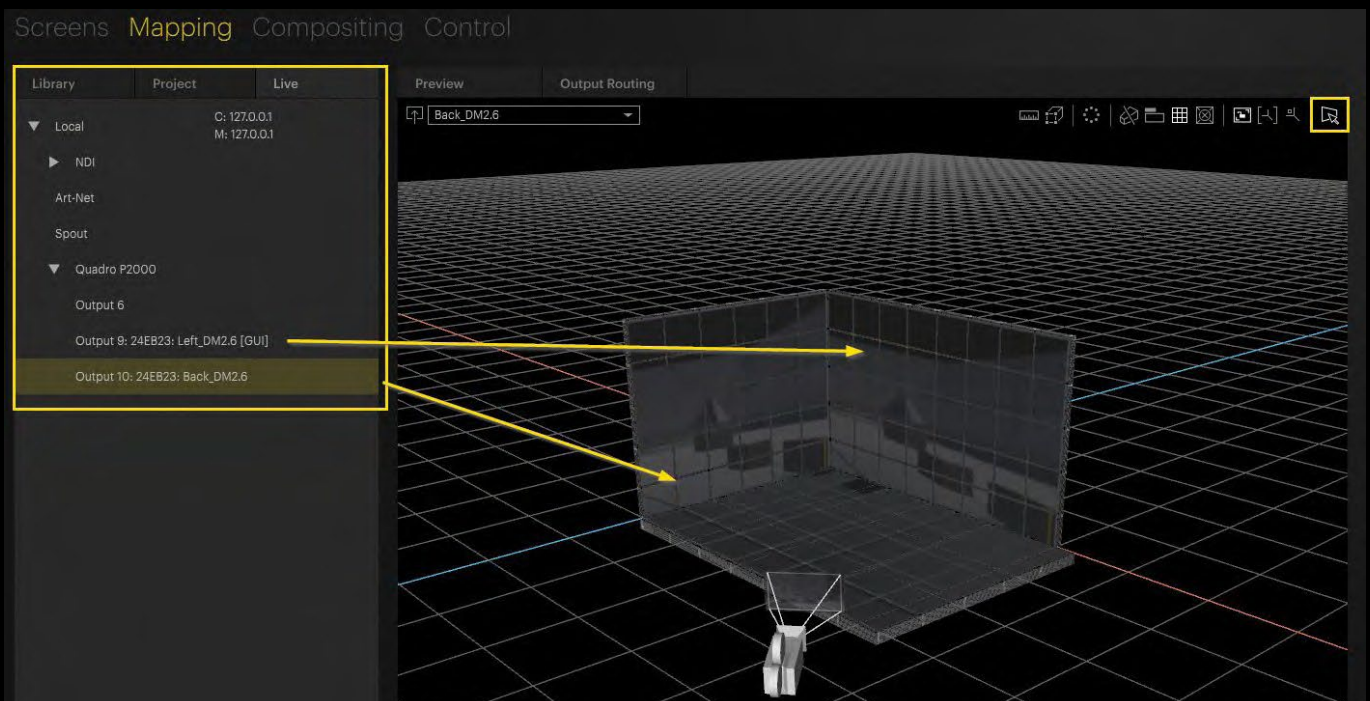
For a proper setup, go ahead and connect physical outputs to your Screens/Displays/LEDs.

Though for test purposes you don't have to have a physical output assigned to your setup. It is possible to only work from within the preview. For tests scenarios this is totally sufficient.



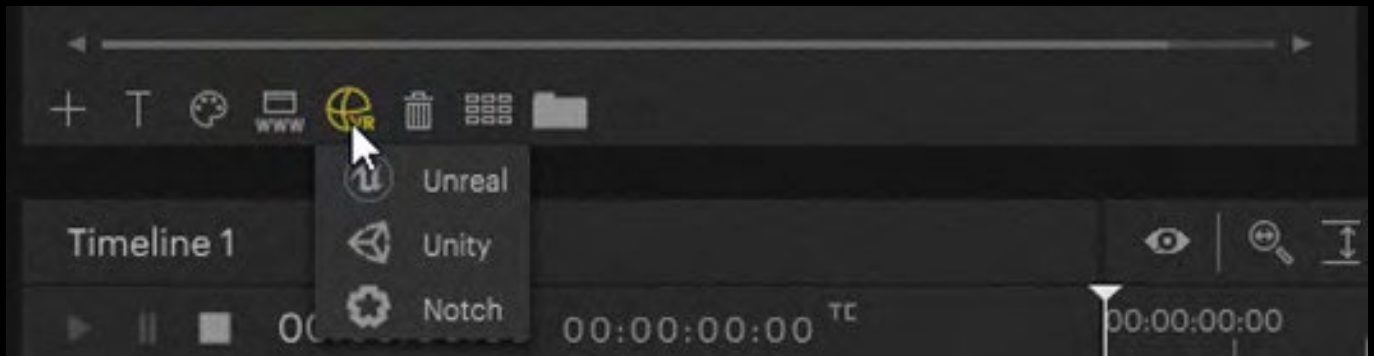
In case you want to play out on real outputs, you can find the output configuration in the Mapping tab. This setup looks different on each computer, as it depends on the system you are currently using. PIXERA will display the graphics card outputs of all your systems in the network. Here is an example of a possible network setup:

(make sure to have the selector on the top right set to „Screens“ instead of projectors, otherwise you won't be able to select and assign outputs to your displays)



2.3 IMPORTING UNREAL PROJECTS

Unreal has to be imported by using the new VR/AR button in the interface:



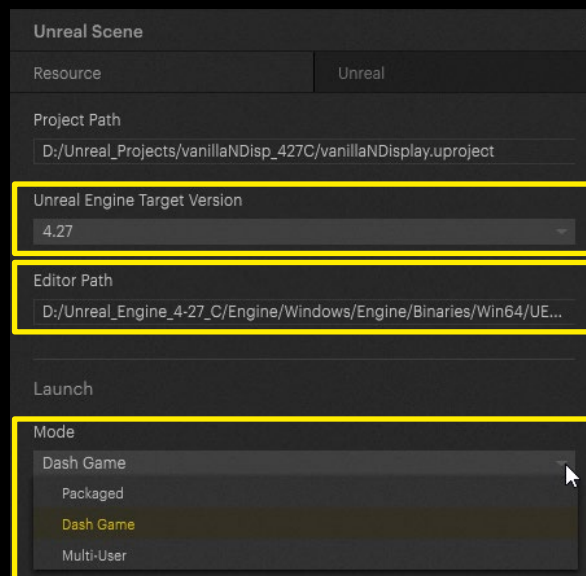
We do offer the same demo scene for download on our website here:

<https://pixera.one/de/downloads/>

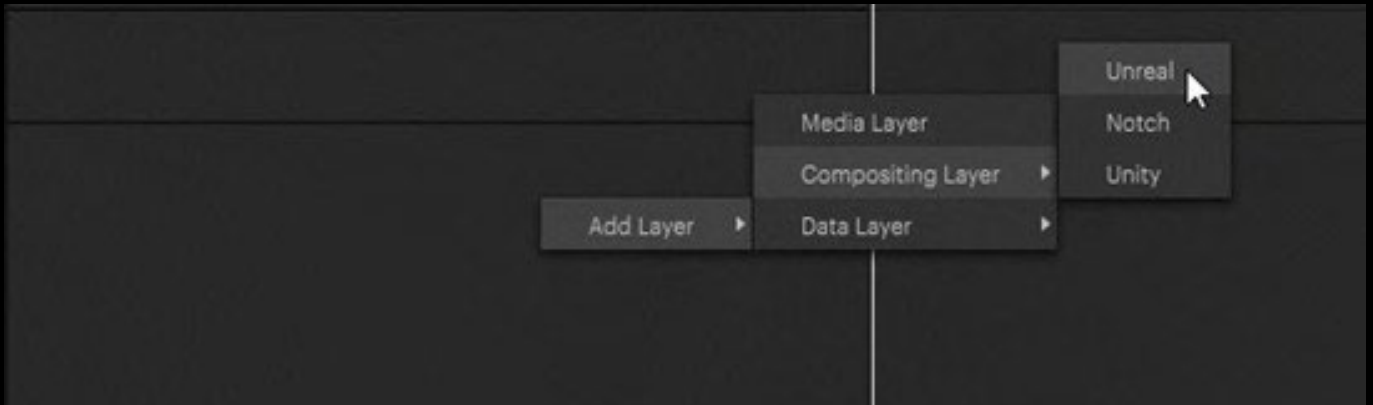
There are three options how to import your Unreal project

- Packed project (.exe File packed in Unreal)
- Dash Game (.uproject file)
- Multi-User (.uproject with Multi-User Plugin)

In inspector you have to select the corresponding mode to be able to launch it the right way. When using .uproject file you have to set the path for UE4Editor in **"EditorPath"** as well. This can be set as initial value in PIXER settings as well. Further please note that you can switch between Unreal Engine 4.26 and 4.27.



Once imported, drag and drop the scene on the empty timeline. If you already have regular layers in use, please be cautious. We implemented a new layer type for 3D engine applications:

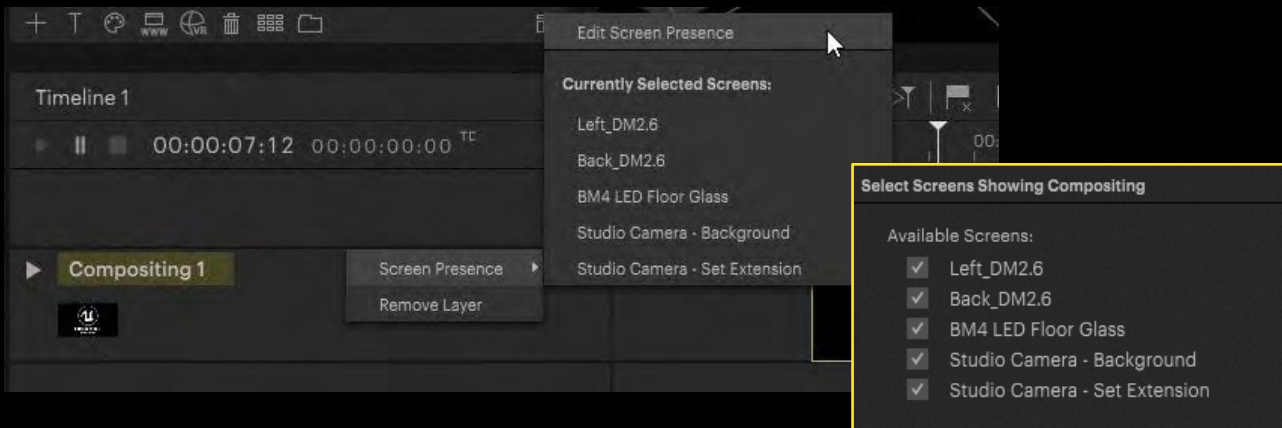


By simply dragging – and – dropping Unreal on an empty timeline, PIXERA will automatically create the correct Compositing Layer for you:

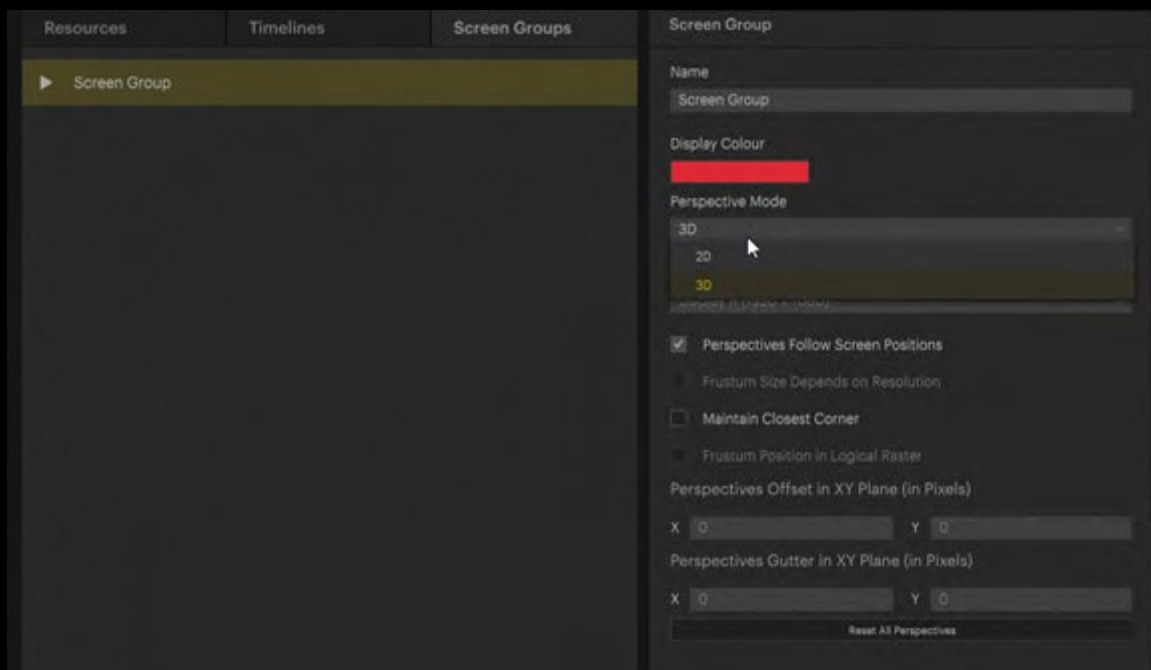


2.4 SCREEN PRESENCE AND NDISPLAY CONFIG FILE

After assigning the scene to a layer, the so called **Screen Presence** has to be edited. By right-clicking on the layer head the drop down menu can be accessed.

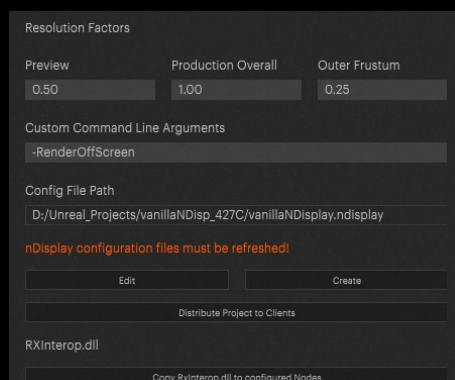


Inside the **Screengroup**, settings change from 2D to 3D:

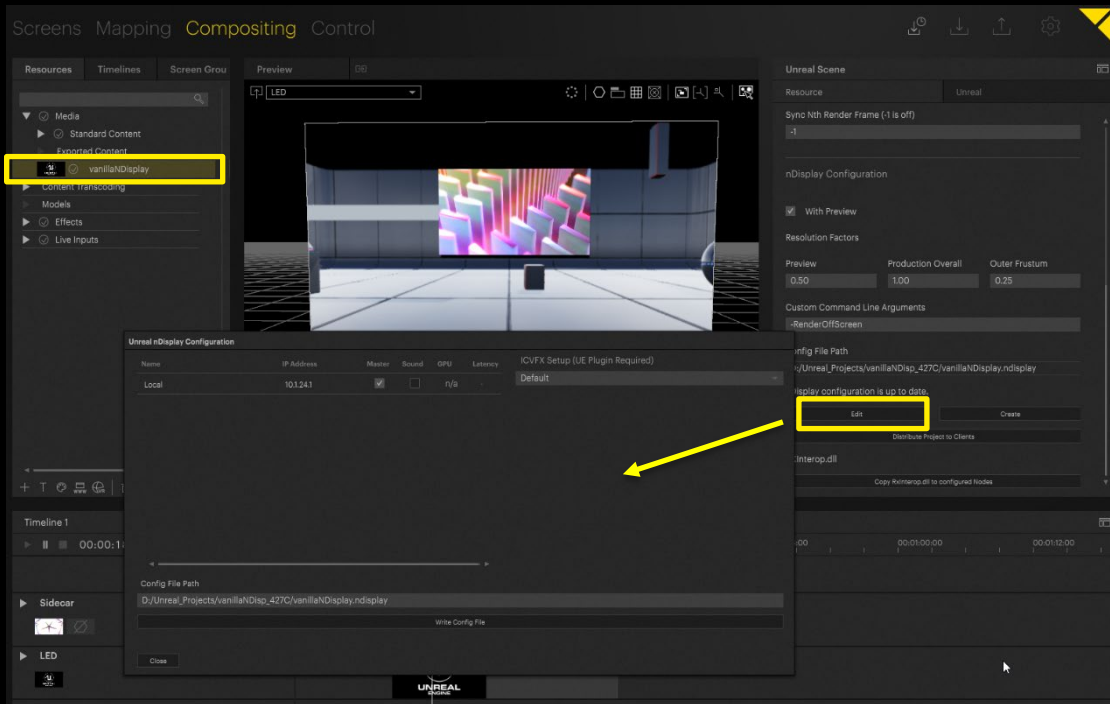


Preview Resolution Factor in inspector allows for lowering the resolution within the preview. We strongly suggest to turn down this value for the Preview machine. The value is displayed in percentages. 1.00 equals 100% whereas 0.50 equals 50%.

With **Outer Frustum Resolution Factor** you can lower the resolution of your LED output to minimize system load.






After finishing those steps you can distribute your project to your clients with **Distribute Project to Clients** button, and create a nDisplay config file by pressing **Create** button at the bottom in inspector. In **Edit** config file you can select for which display should be part of the nDisplay configuration. Please don't forget to distribute RxInterop.dll after installing a new Version of PIXERA or importing a new packed project.



PIXERA will automatically generate a separate config file for the Render-Nodes and one for the Preview.

After these steps, the config files should have been created next to your Unreal project file (either *.exe or *.uproject)

	vanillaNDisplay.ndisplay	21.02.2022 12:59	NDISPLAY-Datei	2 KB
	vanillaNDisplay_preview.ndisplay	21.02.2022 12:59	NDISPLAY-Datei	2 KB
	vanillaNDisplay.uproject	14.02.2022 12:11	Unreal Engine Proj...	1 KB




NOTE: Use Distribute Project to clients to spread your project. The Unreal project is copied to the exact same file path as it is on your master. So please make sure that the same drive letter is available on your clients.

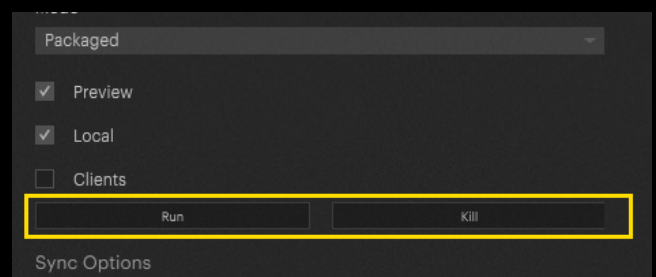
NOTE: Please remember, you should refresh whenever you change something output screen or hardware related

It is of course also possible to open the config files, in order to get a deeper understanding, although it is not necessary.

The last step is to **run the project**.

Therefore you have to select which instances of Unreal you want to launch. You can choose between:

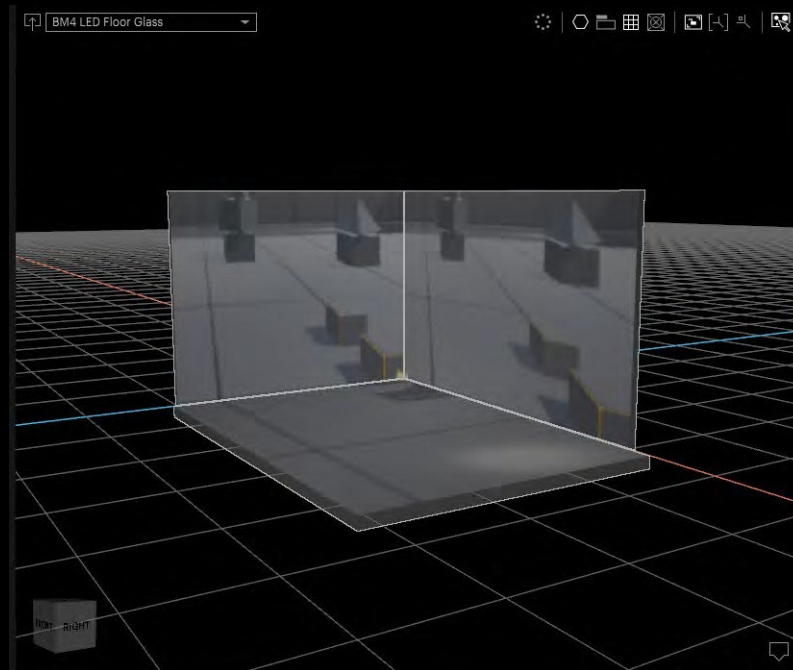
-  Preview (for previewing in your PIXERA Workspace)
-  Local (if outputs are running on your Director machine)
-  Clients



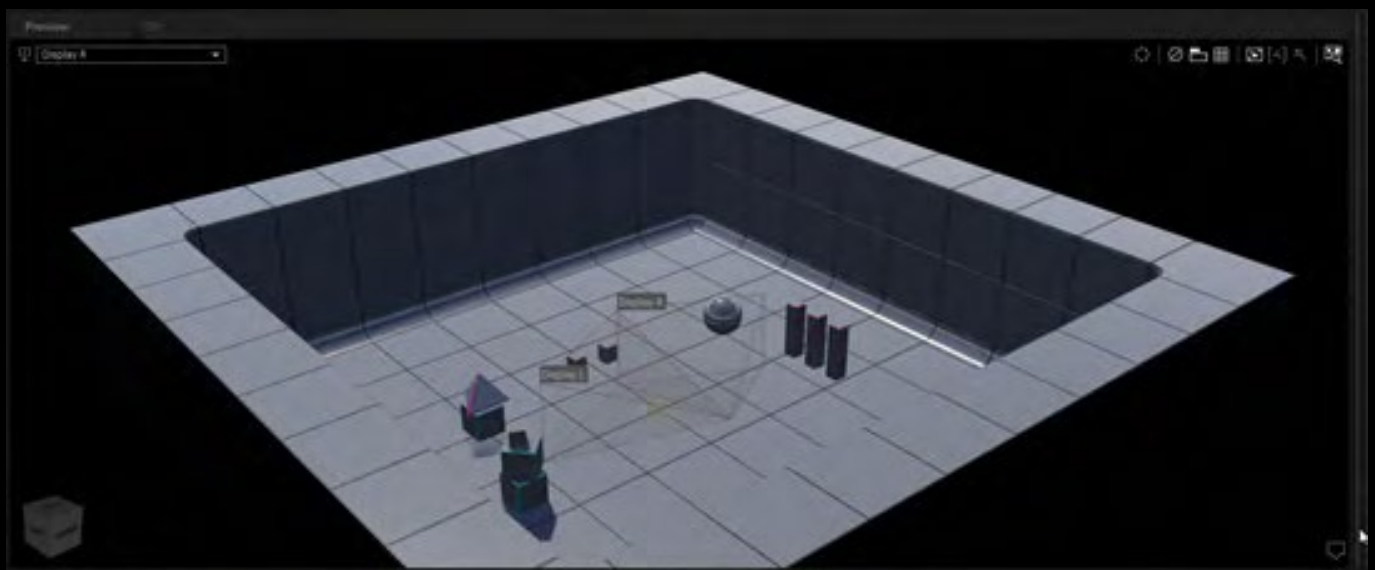
PIXERA will automatically launch nDisplay in the background and run your scene on the selected instances when clicking **Run** button.

NOTE: If changes to the system setup have been made, you first have to **Kill** the scene, create a new config and **Run** your scene again.

In the „outer compositing“ which represents the „real setup“ you can now see how the Unreal scene is rendered to Displays



The complete Unreal scene is rendered inside PIXERA. You can press “D” on the keyboard or dive in via the “dive in” button in the compositing tab, in order to get into the 3D scene:

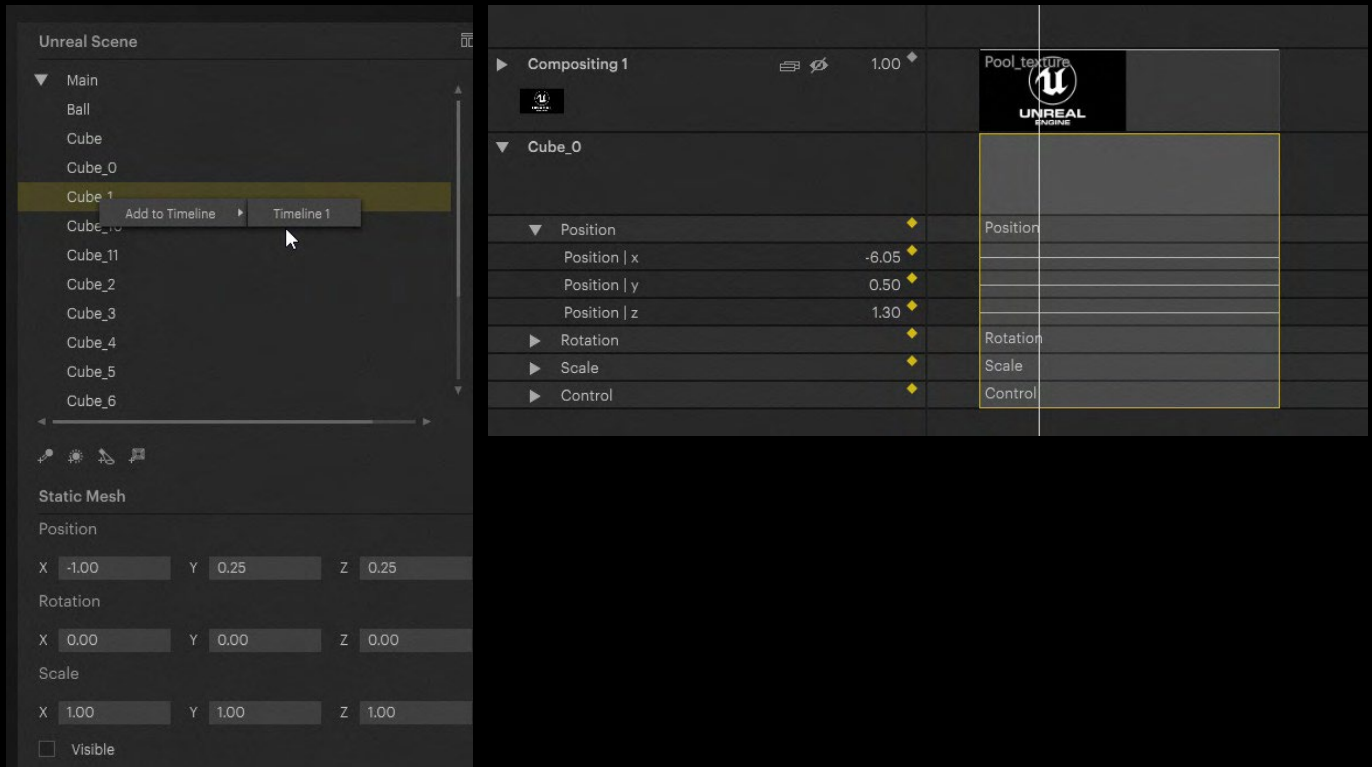


The displays act as viewports looking into the Unreal scene. Changing the position of the screen in PIXERA changes the viewport position in Unreal as well. In this example, I moved the screens up on the X axis as well as adding rotation:

2.5 MOVING OBJECTS IN PIXERA

If the used scene was created with our Unreal Plugin, you also have the possibility to manipulate objects in Unreal via PIXERA.

In inspector you can find a list of all available items. You can either manipulate them in inspector itself or you can add them to the timeline by right clicking an object. Further you can multiselect multiple object and add them to the timeline as group. This might be interesting e.g. when changing multiple lights simultaneously.

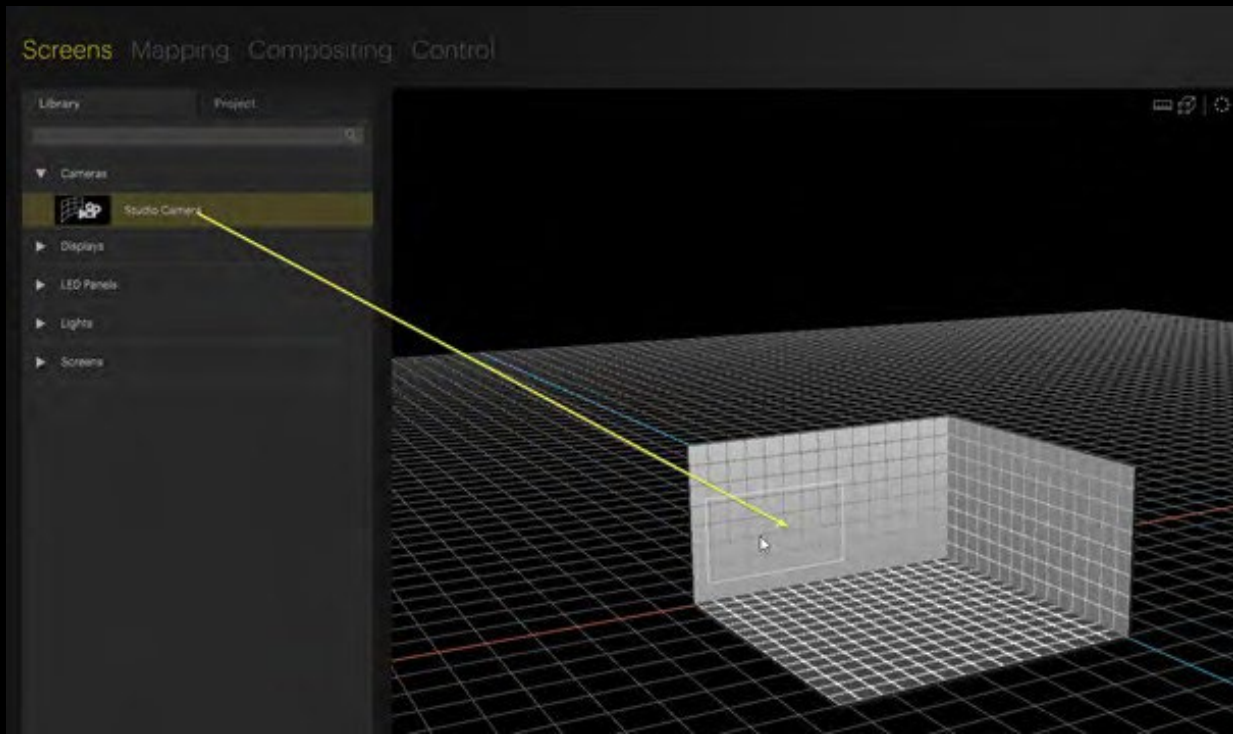


After a new layer is added to the timeline create a new object by doubleclicking in the layer. Then you can add keyframes to manipulate the object in your timeline.

3. STUDIO CAMERA SETUP

The Studio Camera is needed if you are using a Camera and want to either back project the camera frustum to your LED Wall or control entire Perspective of your LED Wall, if you want to use foreground in your live input signal or for using set extension. Camera parameters can be manipulated in Realtime via tracking System.

Navigate to Screens Tab – Cameras and drag & drop your Studio Camera to your workspace.



Select your camera in workspace and adjust the properties in Inspector.

Please note, that you have to edit the screen presence settings of your layer after adding or changing parameters of your Studio Camera.

3.1 STUDIO CAMERA SETTINGS

Studio Camera

Name

Cam

Resolution

Horizontal

1920

Vertical

1080

Screens Associated with Camera:

floor LED

side LED

back LED

Edit Screens

Background

☒ Active
 ☒ Visible

☒ Project on Associated Screens

Overscan

1.10

☒ Scale Resolution

Target Resolution

2112 x 1188

☐ Apply Lens Distortion

Near

0.10

Far

500.00

Auto Color

► Color Correction

Output

Background is re-projected to Associated Screens

☒ Control Perspective Position of Associated Screens

Live Input

☒ Active
 ☒ Visible

► Color Correction

Live Input with Camera Feed

3G-SDI - SDI 1

Delay (in Frames)

0

Foreground

☒ Active
 ☒ Visible

Delay (in Frames)

0

☒ Apply Lens Distortion

Near

0.10

Far

500.00

☒ Discard Background

► Color Correction

Output

No Output assigned

Screens Associated with Camera:

Select the Screen where you want to project your camera frustum Background

Background:

Activate **Background** and **Project on associated Screens** to be able to project your frustum to the screens.

Use **Overscan** to increase the size of your inner frustum

Activate **Control Perspective Position of associated Screens** to link the perspective of the screens to the camera position. This needs to be done to have realistic reflections in real world for camera movements.

Live Input:

Select and activate the Live Input of your camera on your client that is used for mixing Real and virtual world. Delay represents Delay of your tracking system.

Foreground:

Activate Foreground to mix Camera image and virtual foreground together.

Adjust the delay of your camera to synchronize for and background animation.

Near and Far plane must be set separately for the foreground to adjust the depth visibility.

Activate **Discard Background** to make your Unreal scene transparent behind the far plane to be able to see the Live Input.

No output must be set here unless you want to route just the foreground on a separate output.

3.2 VIRTUAL SET EXTENSION

Virtual Set Extension

☒ Active

Update Screen Mask

Delay (in Frames)

5

Virtual View Offset

0.00

☒ Apply Lens Distortion

Mask Softness

Strength

10.00

Power

2.00

► Color Correction

Output

No Output assigned

Output for Live Input and VR

Output

No Output assigned

▼ Tracking

☐ Use with Direct API Input

API Smoothing Factor

0.00

Reload Engine IDs

☐ Pause Direct API Input

▼ Lens Parameters

☒ Use Lens Parameters From Tracking

FOV

40.00

Aspect Ratio

1.778

Panel Width (mm)

9.60

Radial Distortion Coefficients

K1

0.0000

K2

0.0000

K3

0.0000

Tangential Distortion Coefficients

P1

0.0000

P2

0.0000

Center Shift

Cx

0.00

Cy

0.00

► World Transformation

► Local Transformation (Relative)

Activate to extend your camera view beyond the physical LED wall. This means that you can see the virtual world in your camera image beyond the boundaries of your physical LED wall.

Update Screen Mask must be applied after making changes of physical position and size of your screens in PIXERA.

Delay should be the same as in foreground settings.

Mask Softness can be applied to make a softened transition from physical live video to virtual world at the edges.

No output must be set here unless you want to route just the virtual set extension on a separate output.

Output for Live Input and VR:

Select the output where you want the route the mixed signal of live input, foreground, and virtual set extension.

Tracking:

Activate **Use with Direct API Input** to get maximum performance for you tracking. Click **Reload Engine IDs** after any changes you have done in the Camera or general setup settings to make sure, that the tacking input is passed to the right objects.

Please note, that most of the Camera parameters are driven by the tracking system. To adjust them manually, use **Pause Direct API Input** checkbox to temporary disable tracking input and adjust values manually.

Lens Parameters:

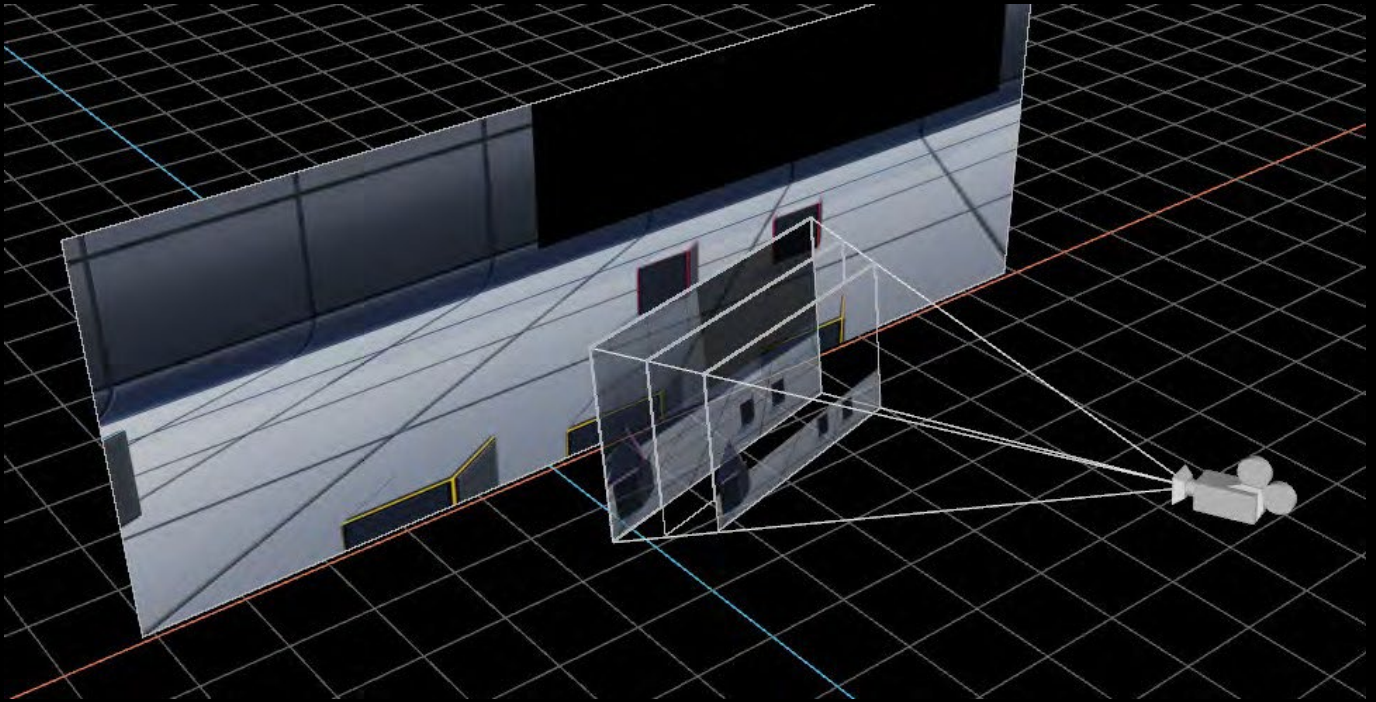
Those parameters can either be set manually or automatically by your tracking system.

Transformation:

Those values can be adjusted for manual positioning of your camera.

After all settings are done, please **refresh** your nDi-splay configuration.

Your Studio Camera frustum should now follow your manual or tracked position.



After setting up your Camera ,Screen presence, generating config files, and running your project you can see the Studio Camera with three planes in your preview and if activated on your output.

- ▼ The plane next to your Camera represents the foreground.
- ▼ The plane in the middle is your Live Input Plane.
- ▼ The 3rd plane is for background that is projected to your assoiciated screens.

On your assigned render output live input, forground and set extenseion are rendered togehter to get one combined image.