

Screen Creator

Intro

This is an outline of the usage of the Unreal Engine plugin Screen Creator to generate walls for use with game development, ICVFX, nDisplay, and other potential use cases.

Contents

Intro.....	1
Requirements & Downloads	3
Plugin Setup	3
Screen Creation Basics.....	4
nDisplay Usage	6
Multiple Node Setup	10
Editor Usage.....	12
Adding Additional Screens	14
Development.....	16
ScreenCreator.....	16
UScreenCreatorComponent.....	16
ScreenCreatorEditor.....	16
FScreenCreatorComponentDetails	16
Known Issues.....	18

Disclaimer

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS." SIRT CENTRE MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SIRT CENTRE BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL CLAIMS, DAMAGES, OR OTHER LIABILITY HOWEVER CAUSED, WHETHER IN AN ACTION OF CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE), ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

Requirements & Downloads

Unreal Engine 5 (5.0.3 or 5.1.1)

Source files - Bitbucket Repository for building from source:

https://bitbucket.org/sirt_projects/screencreator

Plugin Setup

Download and place the **ScreenCreator** folder into your **project's** Plugins folder.

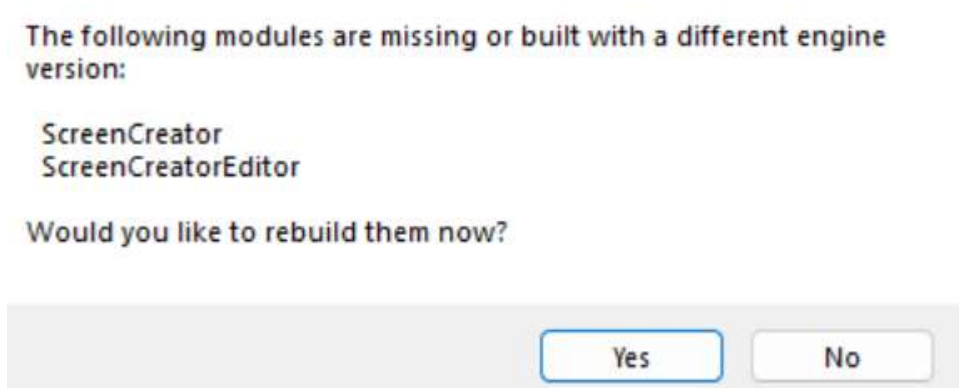
Note: If the Binaries/Intermediates folders are missing in the ScreenCreator plugin folder, you will need to rebuild the plugin code:

You will need to set up:

- Visual Studio 2019/2022 set up for Unreal Engine C++ Development:
<https://docs.unrealengine.com/5.1/en-US/setting-up-visual-studio-development-environment-for-cplusplus-projects-in-unreal-engine/>
- .Net 6.0 - <https://dotnet.microsoft.com/en-us/download/dotnet/6.0>

Then follow these steps:

UE 5.1.1 - Place the plugin into your desired project's Plugins/ folder. Open the project and you will be prompted to rebuild the plugin modules, select **Yes** and the project will open after a short moment.



UE 5.0.3 - Open the project without the plugin in the project's Plugins/ folder. Add a C++ file through the editor options at the top **Tools > New C++ Class...** Select all the default options, and decline opening the IDE for modifying the script. Once completed, close the project and place the plugin into the Plugins/ folder. Open the project and you will be prompted to rebuild the plugin modules, select **Yes** and the project will open after a short moment.

Enable the plugin within the Plugins window inside the project. **The plugin can now be safely copied into your engine's Engine/Plugins/ folder for the so that it is available for all projects using the same Unreal version.

Screen Creation Basics

This tool was generated based on an initial use case of dynamic generation of LED Walls/Warp Meshes for use with Virtual Production (nDisplay) in Unreal. As such, the wall generation is based on a tiled approach, as LED walls are comprised of modular tiled panels. Properties are specified based on tile width/height, as well as number of columns and rows. Finally, the rotation angle between columns can be specified in order to accommodate for curved LED wall setups.

The following properties are used to control the size, shape, and details of a wall:

Tile Width: Width of an individual tile (Unreal units)

Tile Height: Height of an individual tile (Unreal units)

Rows: Number of rows in the screen

Columns: Number of columns in the screen

Rotation Angle: Angle of rotation between each column (degrees). Can be positive or negative for convex and concave shapes.

Double Sided: Toggles the screen to be double sided, rendering the material from either angle of viewing. Does not need to be used if a double-sided material is set for the Screen Material.

Screen Material: Sets the material to be displayed on the screen. The same material will be applied to both sides if Double Sided is selected. The Default Material will be used if no material is selected. If using a double-sided material, ensure that Double Sided is not enabled.

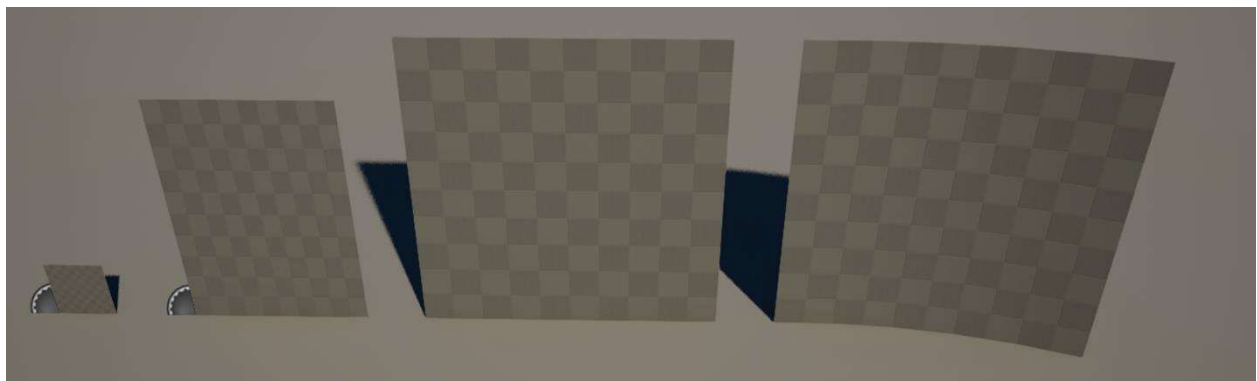
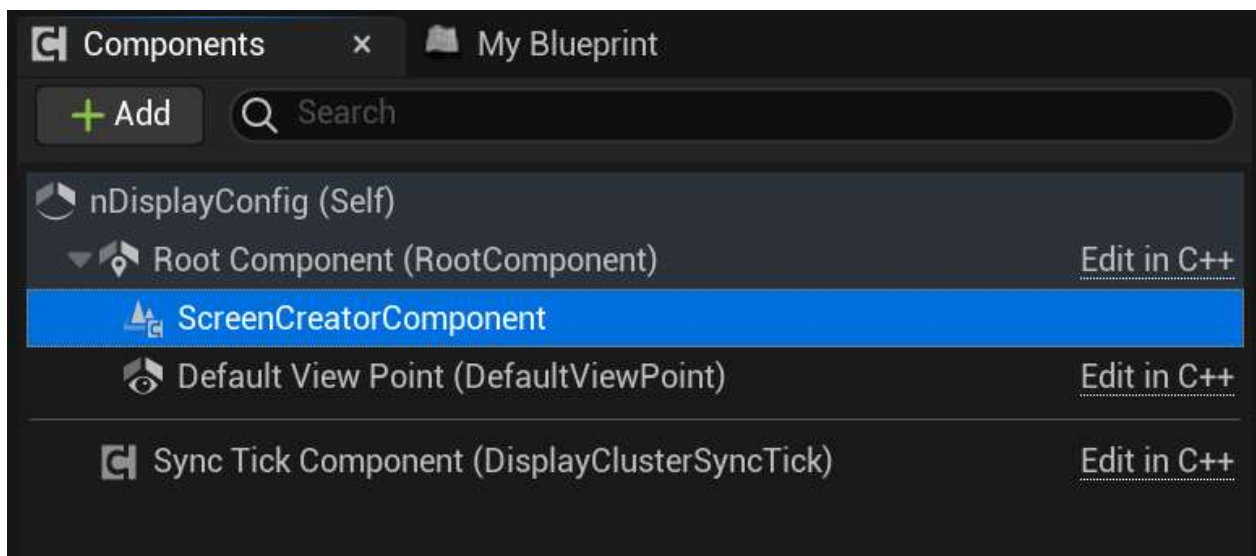
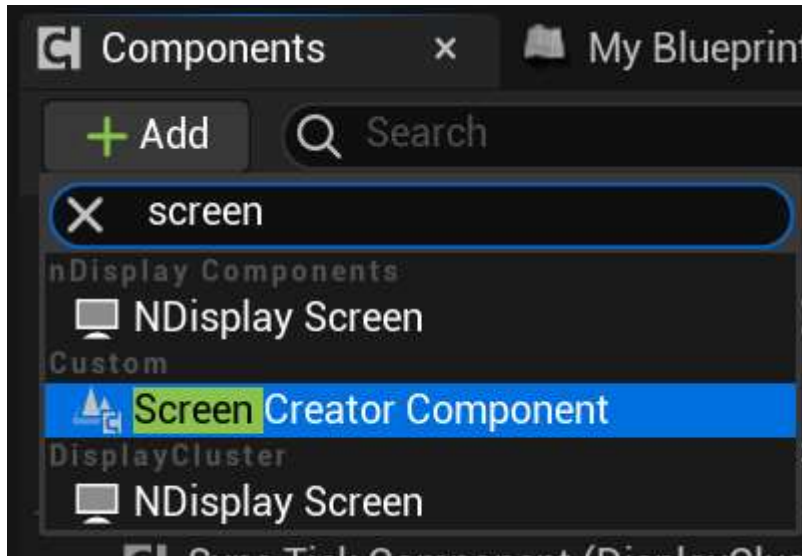


Image: Example screens in order left to right: 1x1 screen, 3x4 screen, 5x5 screen, 5x5 screen with rotation angle of +5 degrees

The wall gets dynamically built from the bottom left corner of the mesh in the positive X direction.

nDisplay Usage

Create or open an existing nDisplay Config asset. In the components window on the left, add **Screen Creator Component** as a child of the Root Component.



Select the component, and in the Details pane, you can modify the size of the screen with the settings under the **Screen Creator** section:

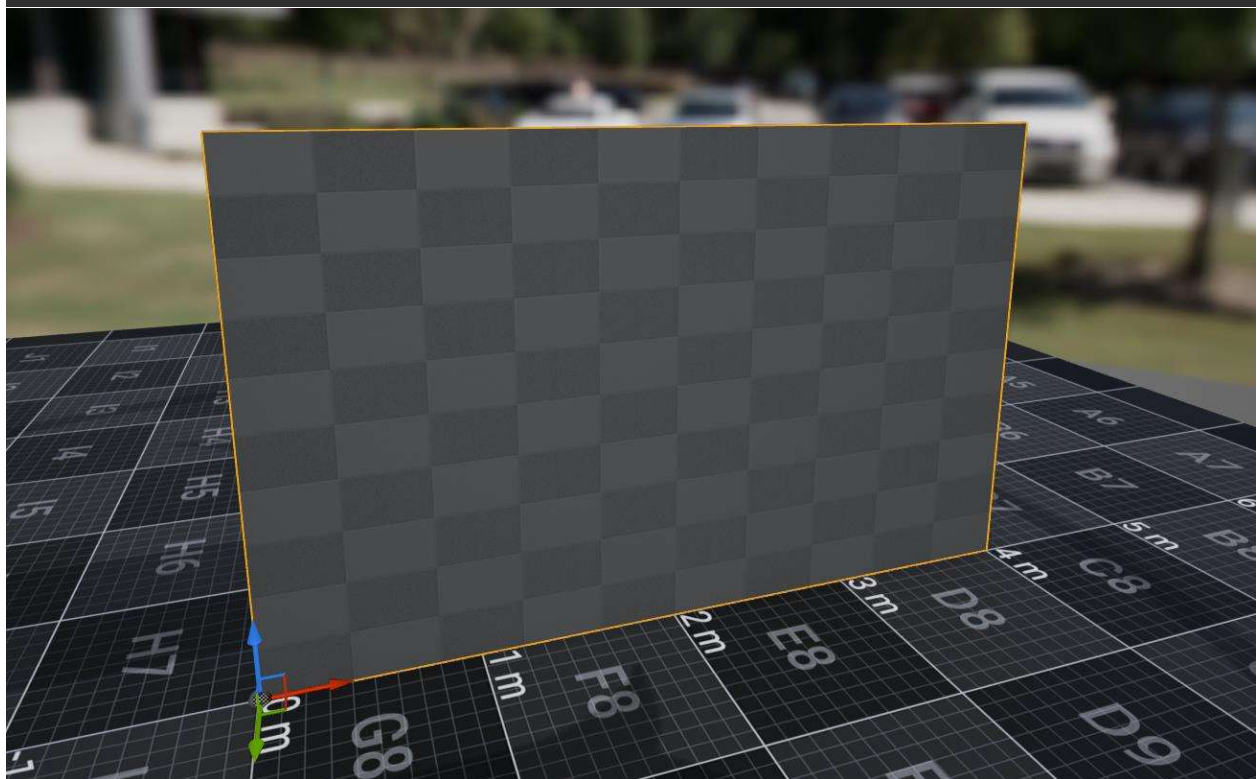
▼ Screen Creator

Tile Width	<input type="text" value="0.0"/>
Tile Height	<input type="text" value="0.0"/>
Columns	<input type="text" value="0"/>
Rows	<input type="text" value="0"/>
Rotation Angle	<input type="text" value="0.0"/>
<input type="button" value="Copy Next Position"/>	<input type="button" value="Copy Next Rotation"/>

When the properties are updated, the screen immediately updates its appearance in the viewport to reflect the changes.

▼ Screen Creator

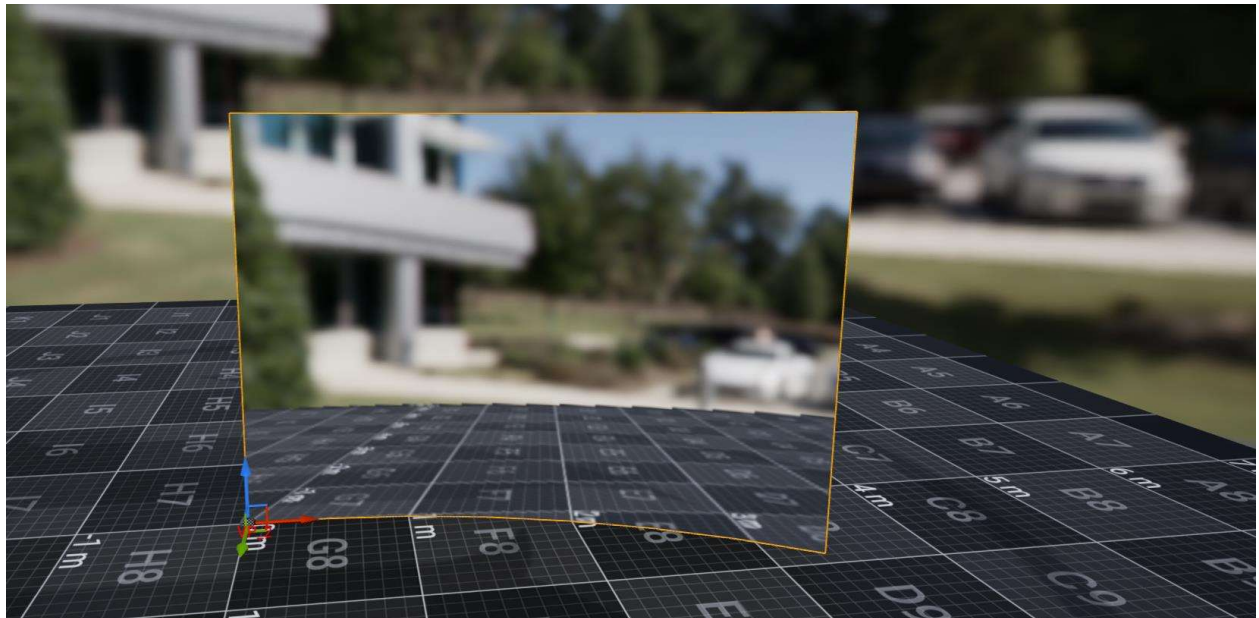
Tile Width	<input type="text" value="50.0"/>
Tile Height	<input type="text" value="50.0"/>
Columns	<input type="text" value="8"/>
Rows	<input type="text" value="5"/>
Rotation Angle	<input type="text" value="0.0"/>
<input type="button" value="Copy Next Position"/>	<input type="button" value="Copy Next Rotation"/>



For usage with nDisplay, the components can be found under the **Mesh** category for the Projection Policy type:

▼ Projection Policy	
Type	Mesh ▼
Mesh	ScreenCreatorComponent ▼
Section Index	0
Warp UVs	-1
Chromakey UVs	-1

Once your component screen is selected as the target mesh, confirm that the projection against the wall matches your setup required.

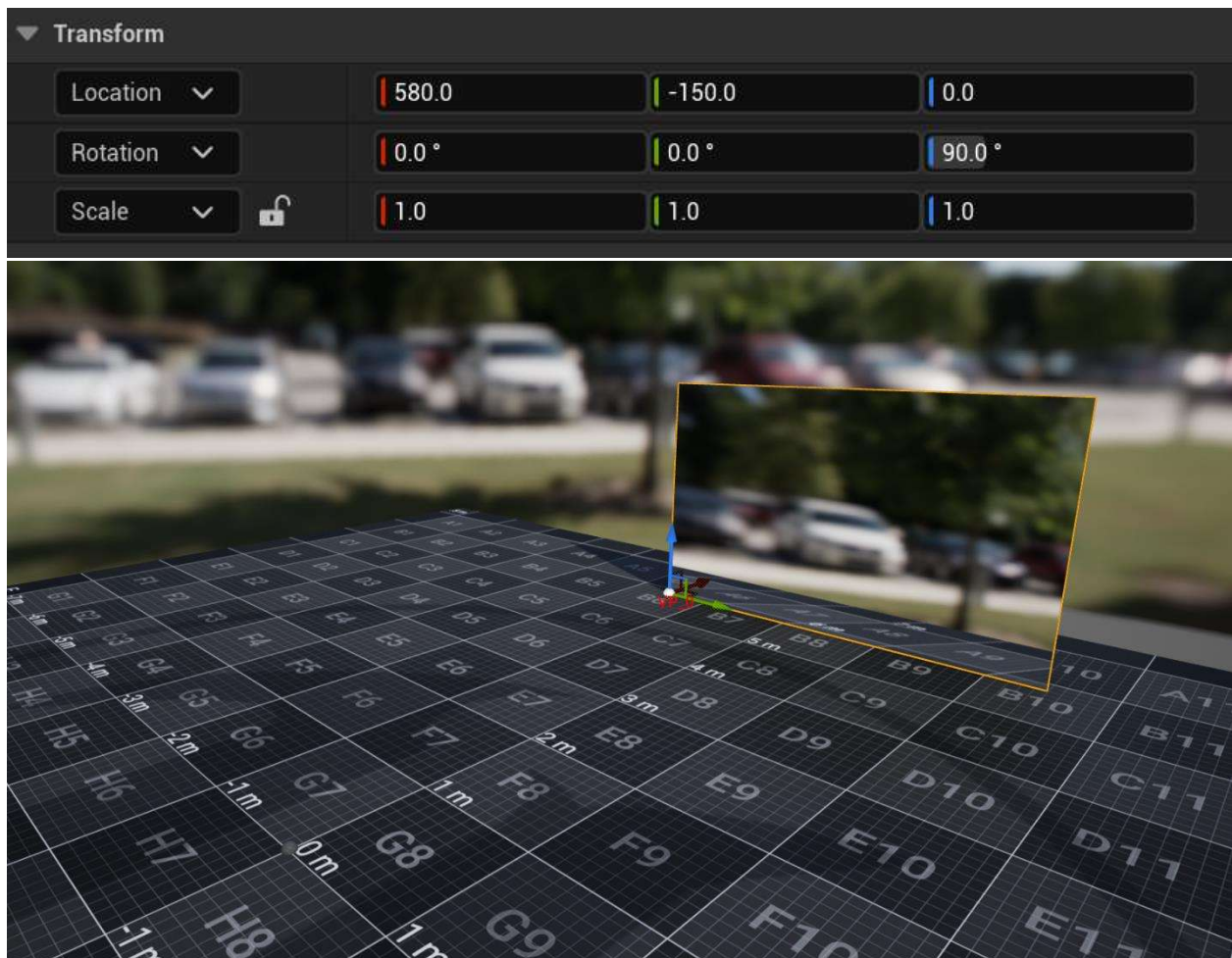


Note: The screen setup is slightly different than the nDisplay Screens provided by default for nDisplay. nDisplay Screens have their anchor point at the centre of the mesh, while the ScreenCreatorComponents have their anchor points at the bottom left corner of the wall. This is done as it's much simpler to measure the physical position of the wall from the tracking origin of your space to the bottom left corner of your LED wall. If replacing the workflow of nDisplay screens with the Screen Creator, you may be required to calculate the different position of your wall accordingly.

For placement of the wall to match your physical wall setup, measure the distance in physical space from your **tracking origin to the bottom left corner of your wall in CM/M**. This distance

in X and Y values represent where the virtual wall should be placed. Additionally, rotate the virtual wall in the Z direction to match the rotational offset of your physical wall setup.

This is an example setup where the wall measured 5.8 metres in the X, and -1.5 metres in the Y in physical space, and the wall is perpendicular to the X direction.



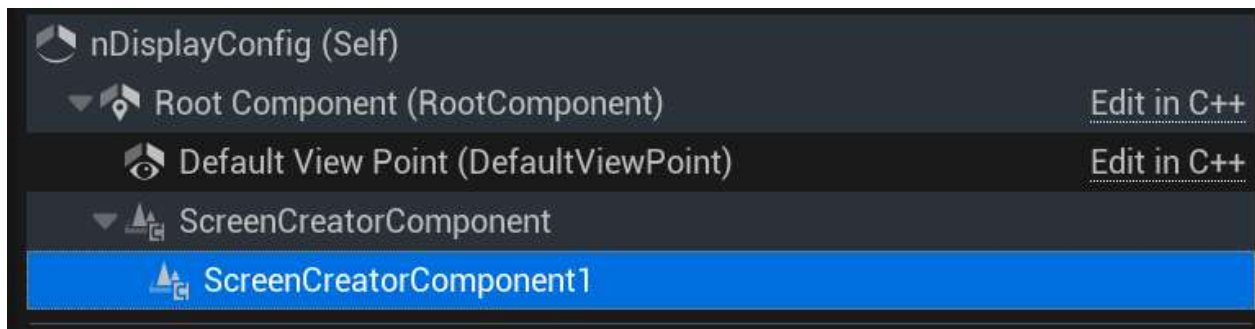
Multiple Node Setup

To help assist with usage for multiple node setups for a continuous LED Wall for nDisplay, buttons providing the next position & rotation relative to the current screen are provided for you.

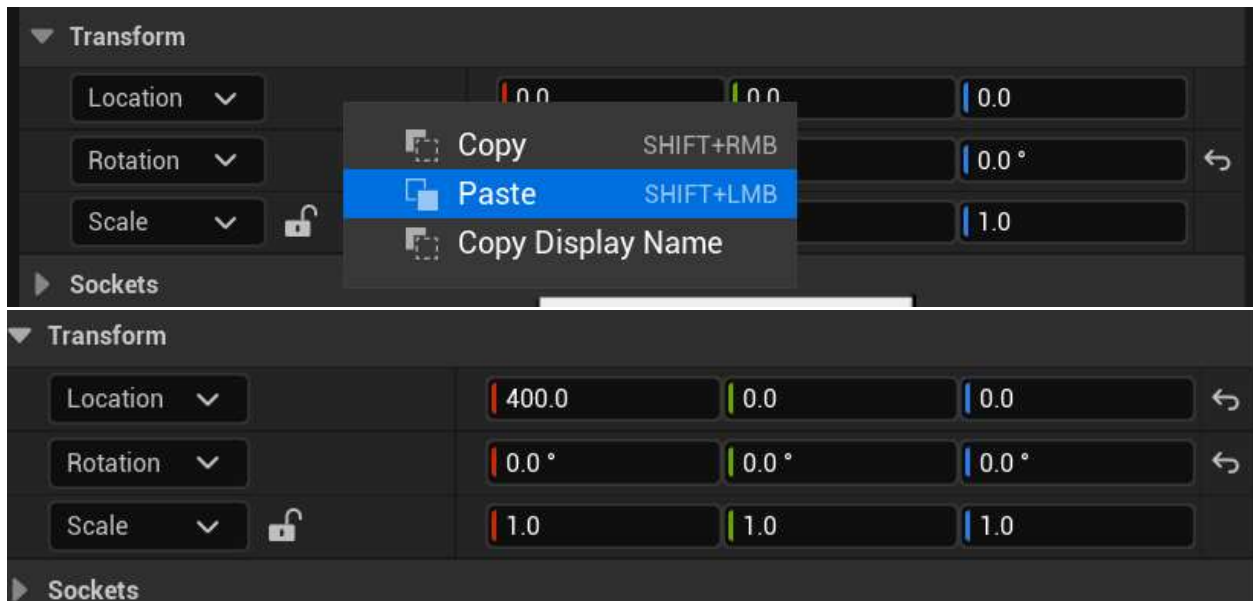
Select the component you want to continue the wall generation on. In the details pane, you can find the following buttons under the **Screen Creator** section:



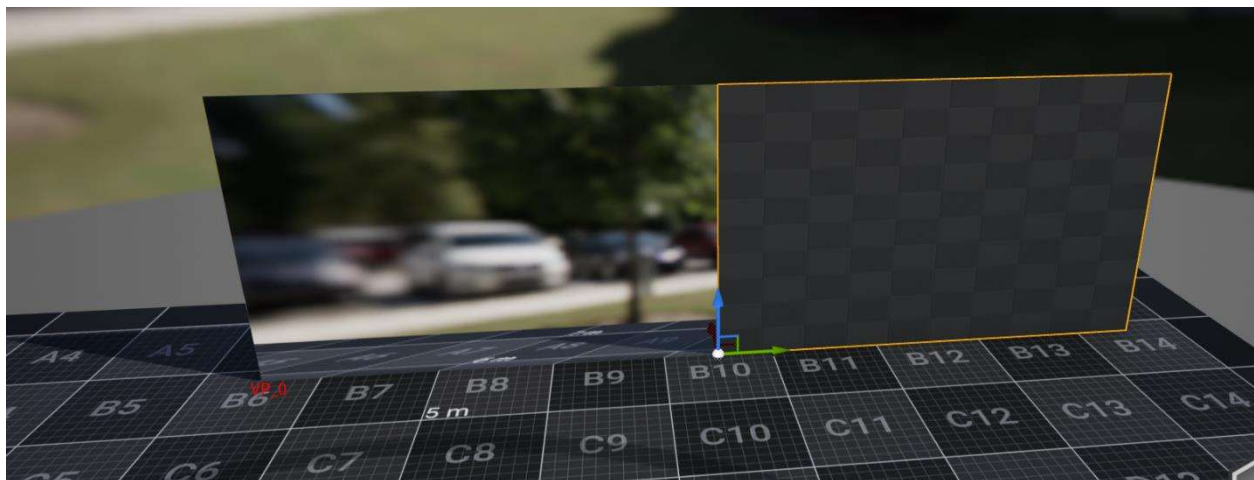
These buttons will copy the values for the next position and rotation respectively for any additional walls to create a continuous mesh. Duplicate the existing **Screen Creator Component** and set it as a child of the existing component.



Use the copy buttons from the parent Screen Creator Component to copy and paste the values into the child Screen Creator Component's transform by right clicking on the **Location** and **Rotation** fields. The properties of the wall (tile size, rows, columns, etc.) will be copied over from the original and will need to be modified if changes between node usage is required.



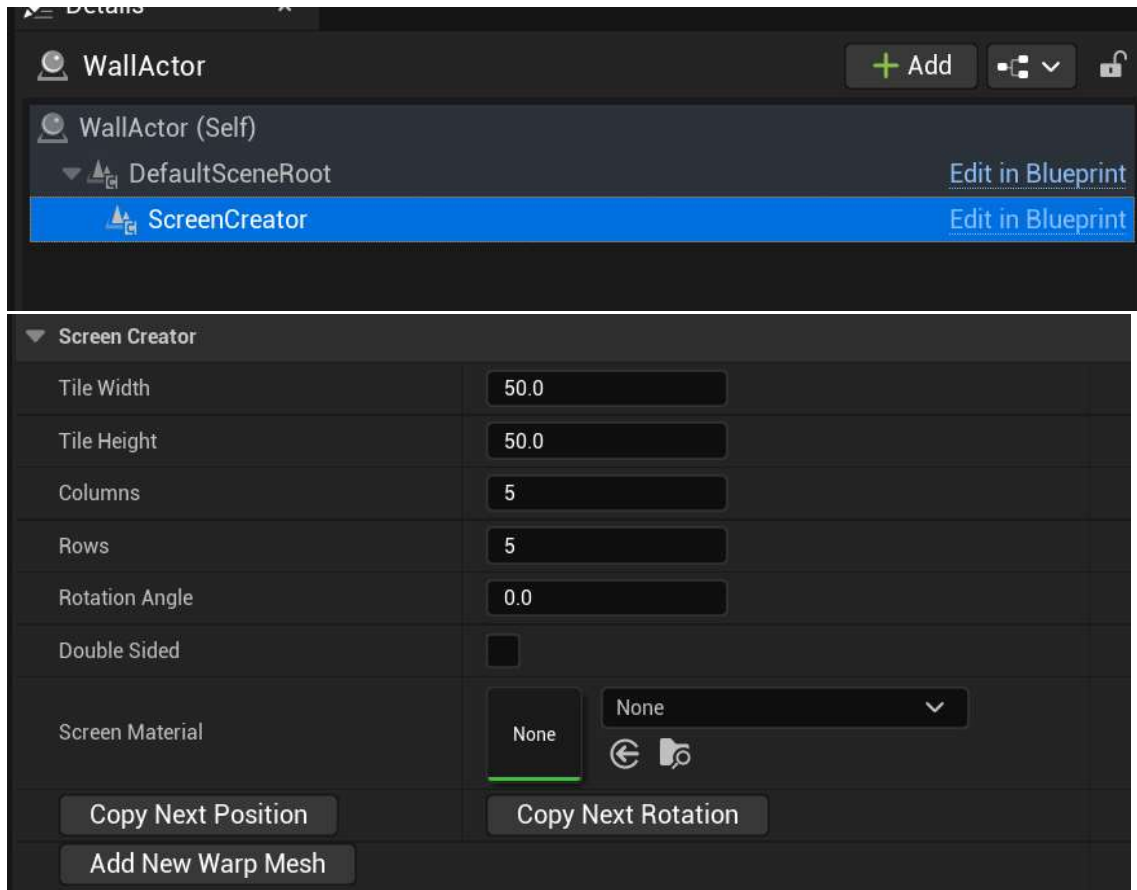
In the viewport, you will see the additional wall will be placed as if it was a continuous mesh:



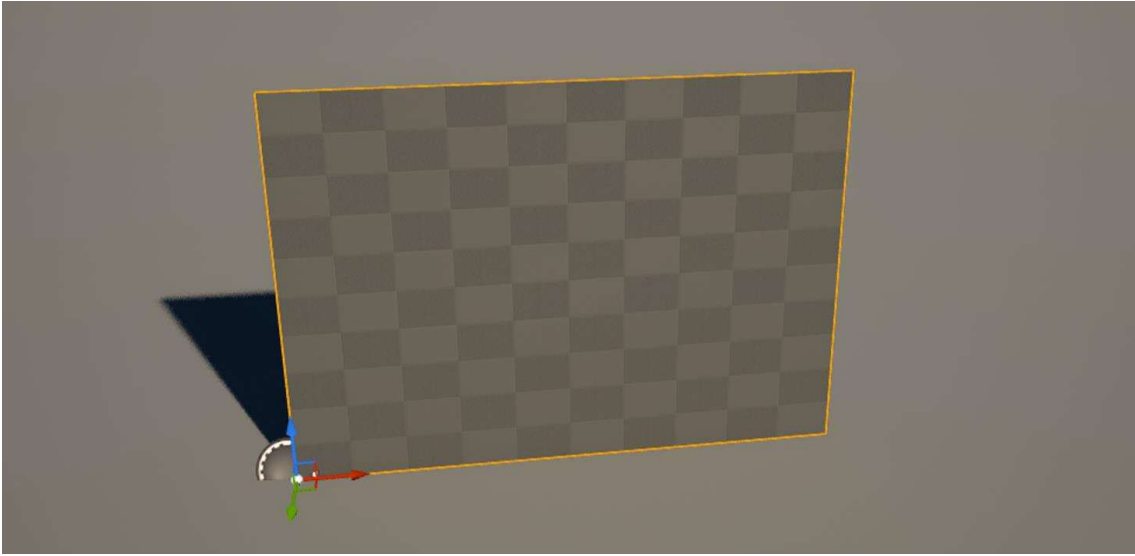
Continue the process for any additional nodes required. Ensure to set up your additional nodes with the Mesh projection policy to properly project onto the walls.

Editor Usage

Create an actor and attach the **Screen Creator Component** to the actor. In the Details pane, under **Screen Creator**, set the properties of your screen component:



As the properties are updated in the Details pane, the screen will automatically generate based on the set information.

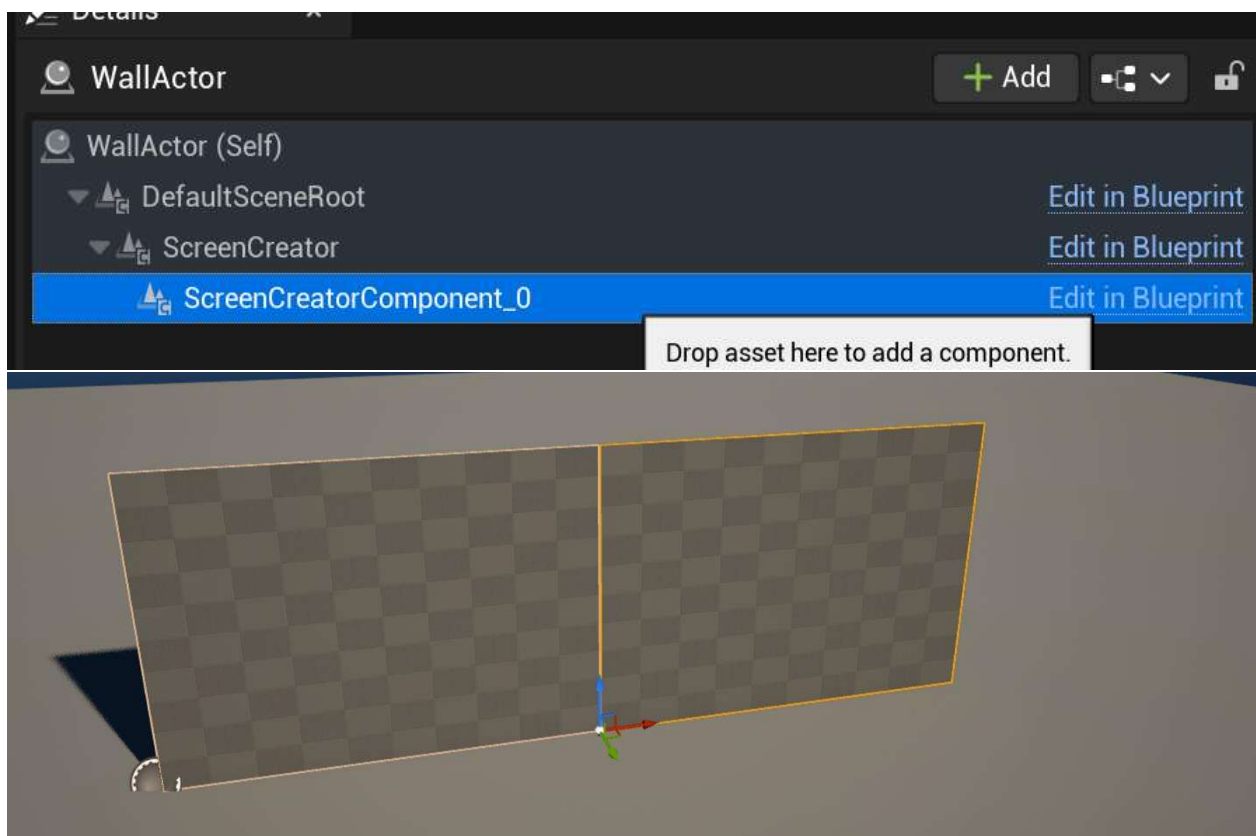


Adding Additional Screens

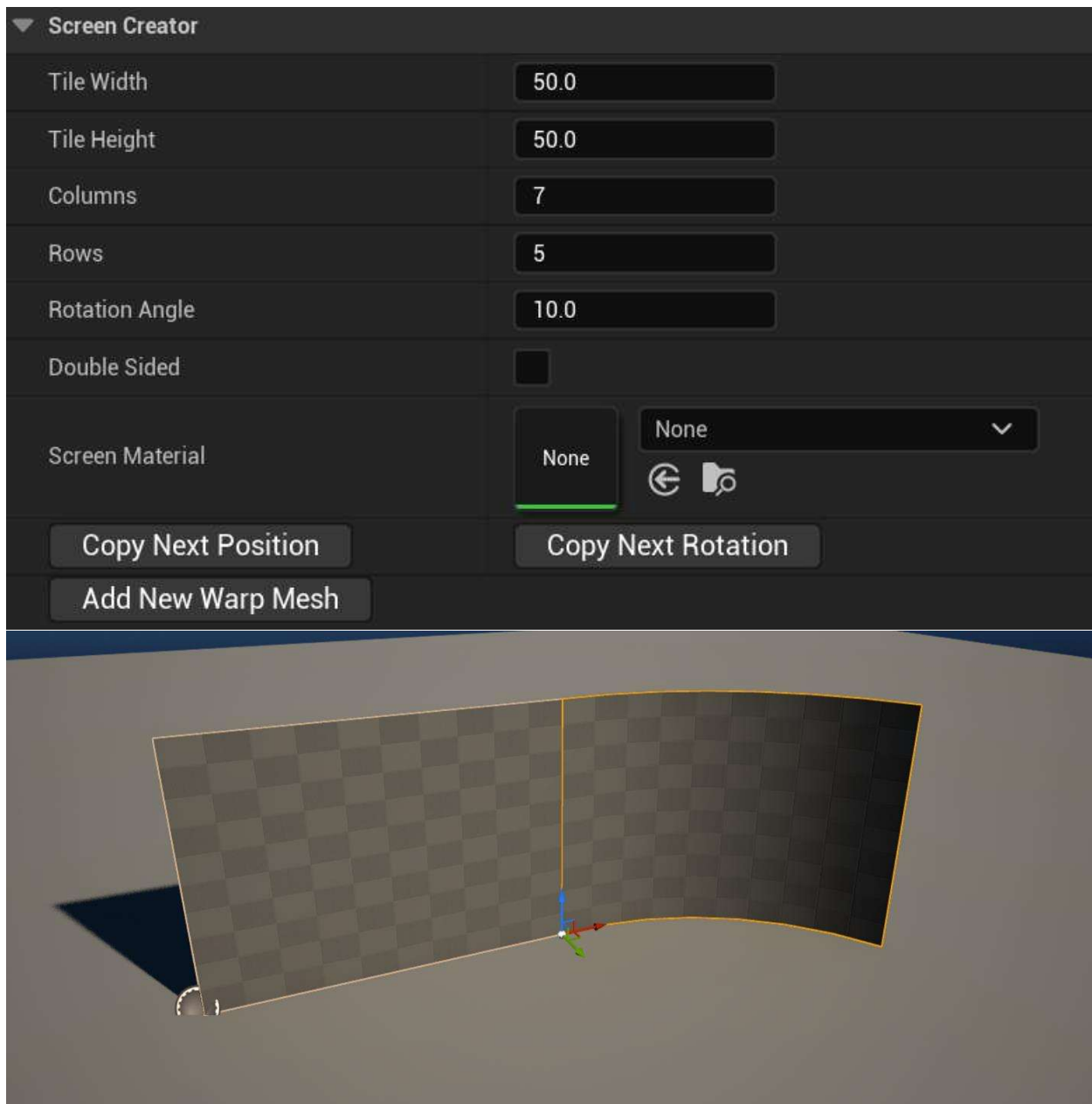
When adding additional screens, the screen is measured from the bottom left corner of the generated wall. To assist with placement of screens for continuous structures, while in editor, the **Add New Warp Mesh** function will place a new component at the end position of the previous screen.



When pressed, an additional component is created in the Actor's list. The component will be initially invisible in the viewport, but clicking the component in the Actor component list will generate the screen with the same details used from the parent component the screen was created from.



Change the properties of the additionally generated wall as required and the viewport will update with the screen changes



Repeat the process for any additional screens to be used.

Development

The following section covers the code used for the plugin and are listed here for development purposes.

The plugin is split into two modules: **ScreenCreator** and **ScreenCreatorEditor**. ScreenCreator handles the wall generation, and the Editor module is used to handle the editor functionality for copying next position & rotation for a wall.

ScreenCreator

UScreenCreatorComponent

Component responsible for the generation of the procedural mesh based on the provided properties. Inherits from **UProceduralMeshComponent**.

Function	Description
void GenerateWarpMesh()	Generates the screen using the supplied properties. Triggers when PostEditChangeProperty triggers.
void AddNewWarpMesh()	Creates a new screen at the position/rotation of the end of the current screen as if it was one continuous mesh.
FVector CalculateEndPosition()	Calculates the end position of the wall segment. Used primarily for curved walls as adding additional walls as a continuous mesh requires geometric calculations.
FRotator CalculateEndRotation()	Calculates the end rotation of the wall segment. Used primarily for curved walls as adding additional walls requires a rotation to be set relative to the parent.
void PostEditChangeProperty(FPropertyChangedEvent& PropertyChangedEvent)	Function for USceneComponents to determine if there are any changes to object properties, and to regenerate the screen if there are.

ScreenCreatorEditor

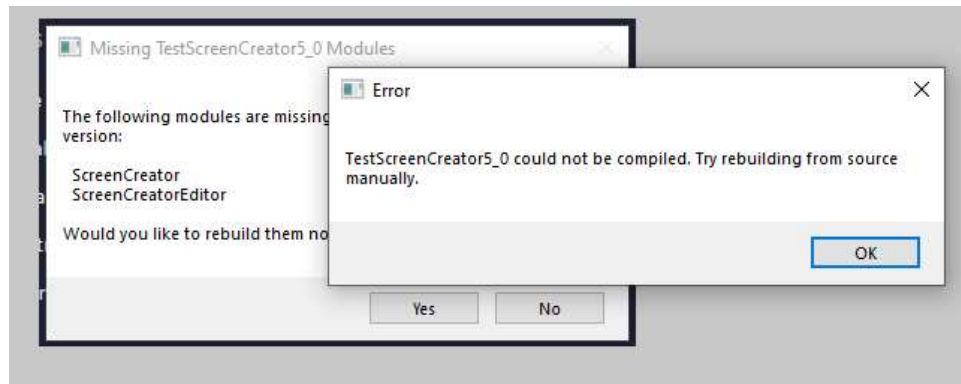
FScreenCreatorComponentDetails

Editor script used to display the custom UI elements for copying the next position and rotation for additional wall generation.

Function	Description
FReply OnCopyPositionDataClicked()	Handles copying of the next position value of the selected ScreenCreatorComponent.
FReply OnCopyRotationDataClicked()	Handles copying of the next rotation value of the selected ScreenCreatorComponent.
FReply OnCreateNextScreenClicked()	Handles the creation of an additional wall. Currently disabled due to limitations of dynamic adding of components within an editor window (such as nDisplay)
UScreenCreatorComponent* GetFirstSelectedScreenCreatorComponent() const	Retrieves the first selected ScreenCreatorComponent in the editor window. Used for determining valid components for obtaining next position & rotation data

Known Issues

During installation/rebuilding of the plug-in, you may get the following message:



You will need to rebuild the solution in visual studio. Refer to Visual Studio and VS for Unreal Development documentation for more information.