nDisplay Control Toolkit

Intro

This is an outline of the development of an Unreal project (v4.26) with nDisplay camera functions to be controlled be Monogram hardware. We will review how functionality is created so the toolkit functionality may be extended, improved and/or migrated to other projects.

Contents

Created by SIRT - Screen Industries Research and Training Centre
2021

Jason Hunter, Wladyslaw Bronowicki, Brianna Mcburney, Cory Salmon, Brian Chiu
v1.1

Disclaimer

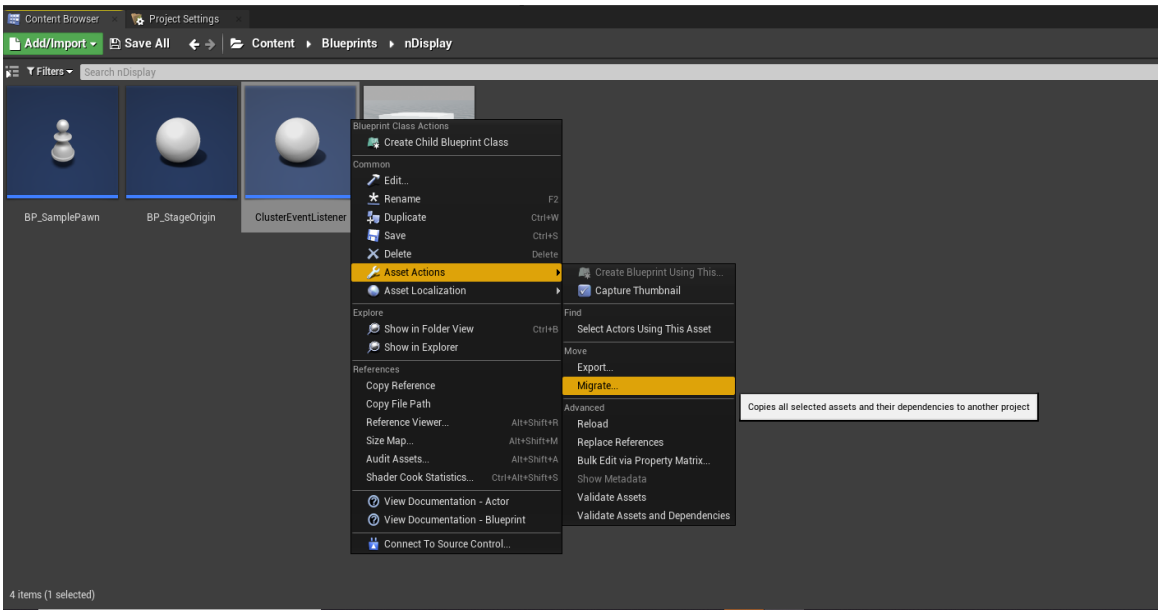
THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS “AS IS.” SIRT CENTRE MAKES NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL SIRT CENTRE BE LAIBLE 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.

Set up Project

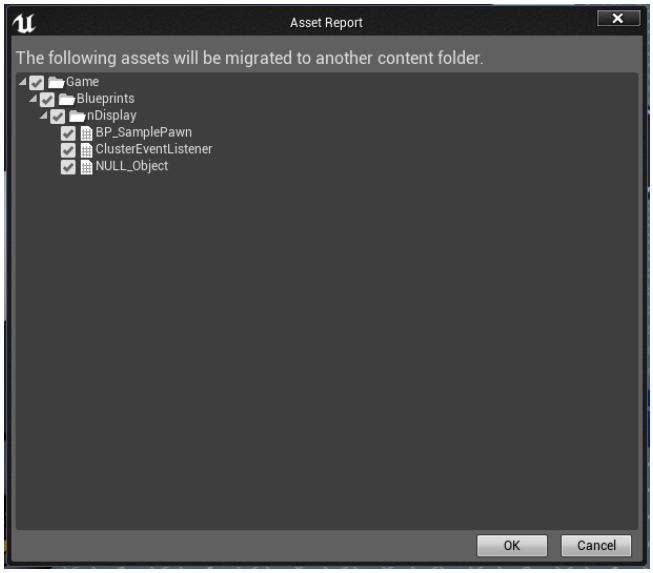
The template used was the [In Camera VFX Example], there is nothing changed except the modification of the BP_SamplePawn class and the addition of the ClusterEventListener class.

Once the project is downloaded ensure that the most recent version of the nDisplayCT plugin is in the plugins folder of the Unreal Engine or the project itself, and make sure it is enabled for the project.

If you are importing the toolkit into an existing project which is already enabled for nDisplay, we recommend Migrating the ClusterEventListener (and all associated assets) to your project.



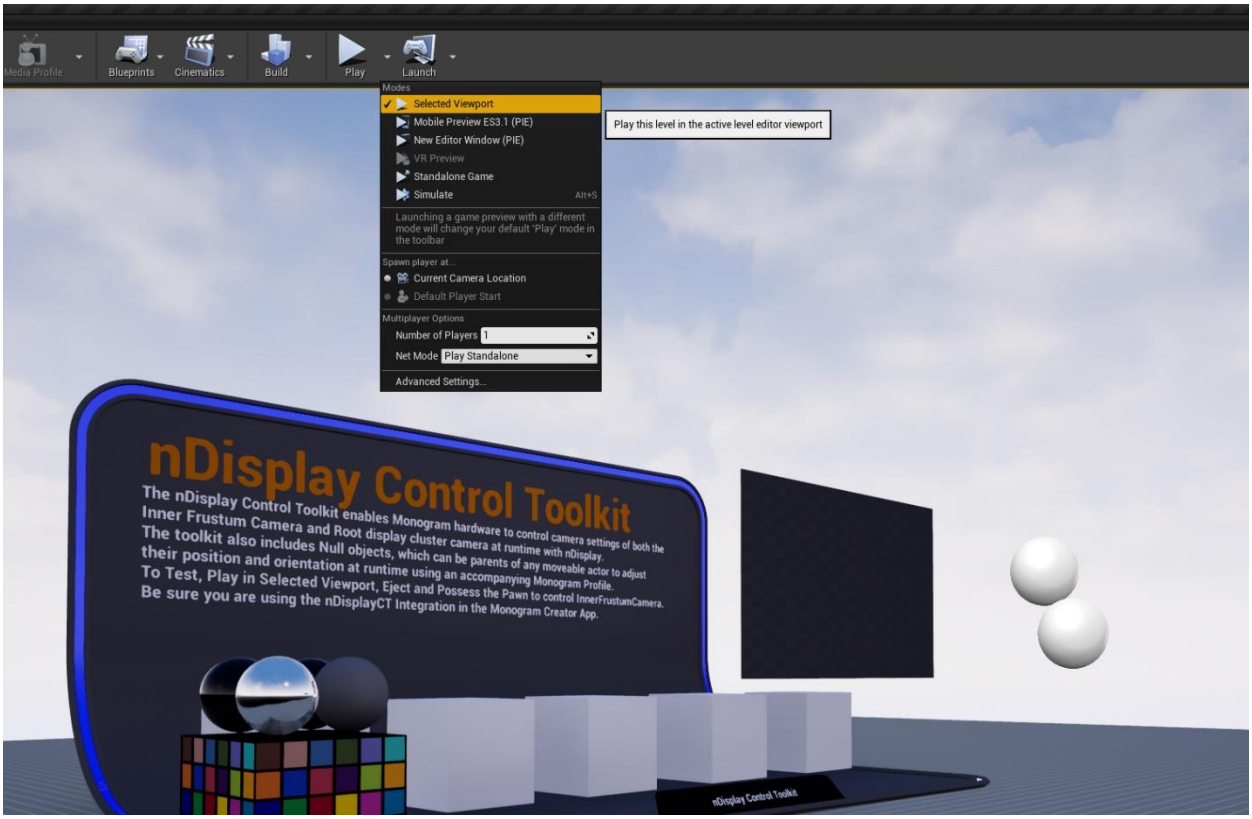
Right-Click the ClusterEventListener from the Content Browser, choose Asset Actions>Migrate…



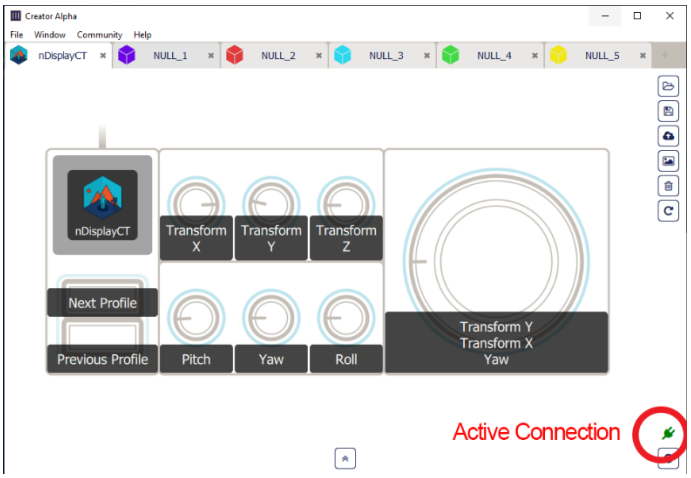
List of Assets which will be migrated

Test Connectivity

Connectivity between Monogram and the Unreal project can be tested running the project using Play>Selected Viewport. Ensure you are possessing the camera or test using Null controls.



Ensure you have an active connection to Unreal in the Monogram Creator App. The toolkit only functions if one (1) instance of Unreal Engine is running. A restart of Monogram Creator can establish a new connection, see Known Issues.



Development

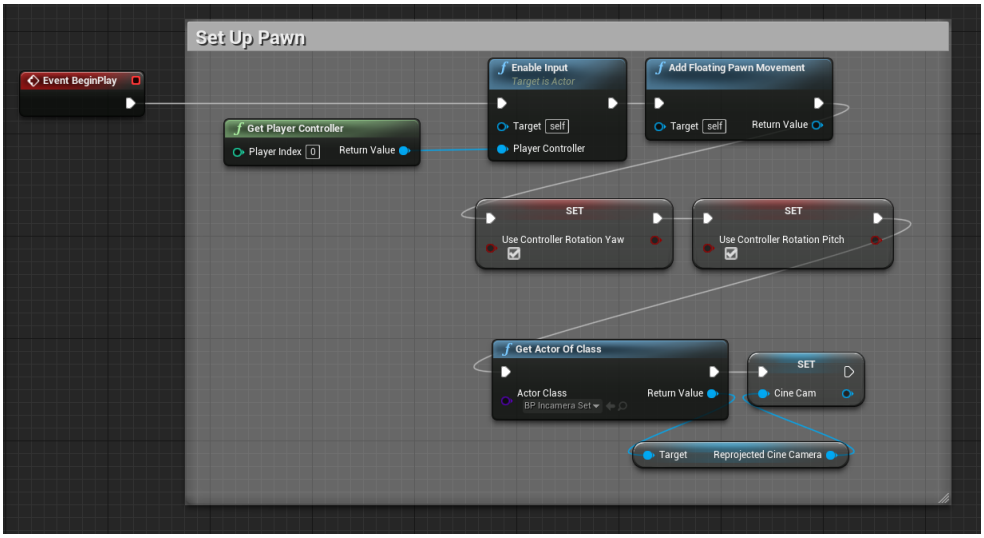
The following section outlines the blueprints which enable functionality and are listed here for development purposes.

BP_SamplePawn blueprint

The Pawn needs to be set up so that it is moveable therefore a *FloatingPawnMovement* component is added to its components window.

Event Graph > Event BeginPlay

A variable called CineCam of type Cine Camera Actor needs to be created and a few actions (see below) need to be added to allow for functions (to be created later) to work.



BeginPlay event in the BP_SamplePawn blueprint event graph

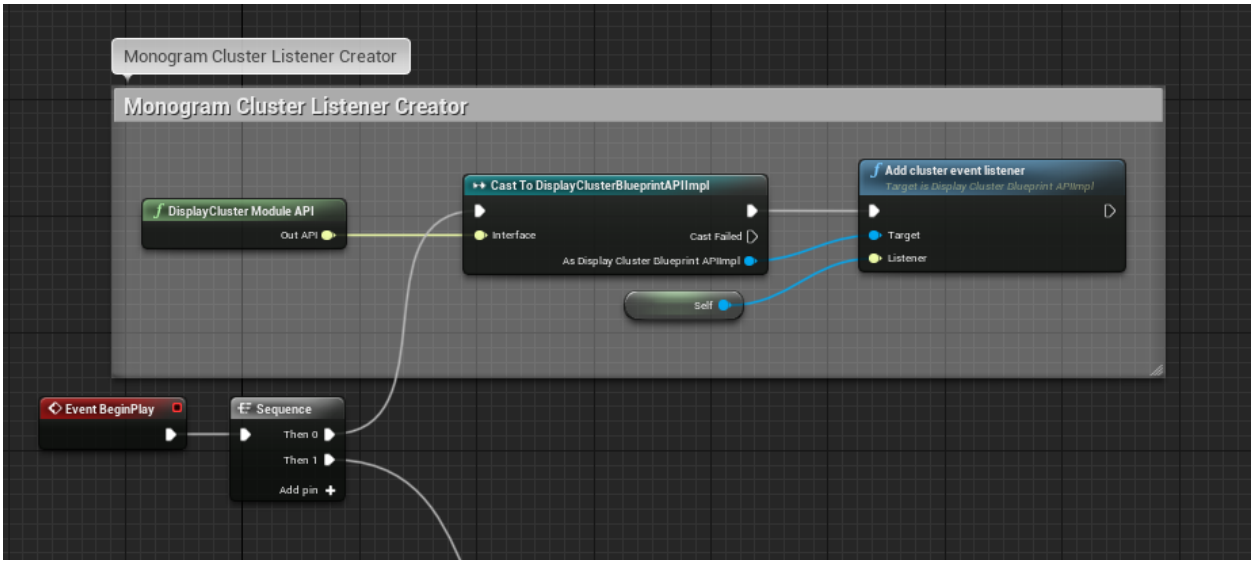
ClusterEventListener blueprint

Cluster Events are used so that when running a project with nDisplay any changes made to the primary node will be reflect on all nodes simultaneously. A class called a listener is required to be the main hub for all cluster events. See Responding to Cluster Events in Blueprints.

- Create new blueprint of type Actor and then open up the blueprint
- Under Class Settings > Interfaces > Implemented Interfaces
    - Add Display Cluster Cluster Event Listener
- Drag the class from the content browser into the level viewport and drop it into the level

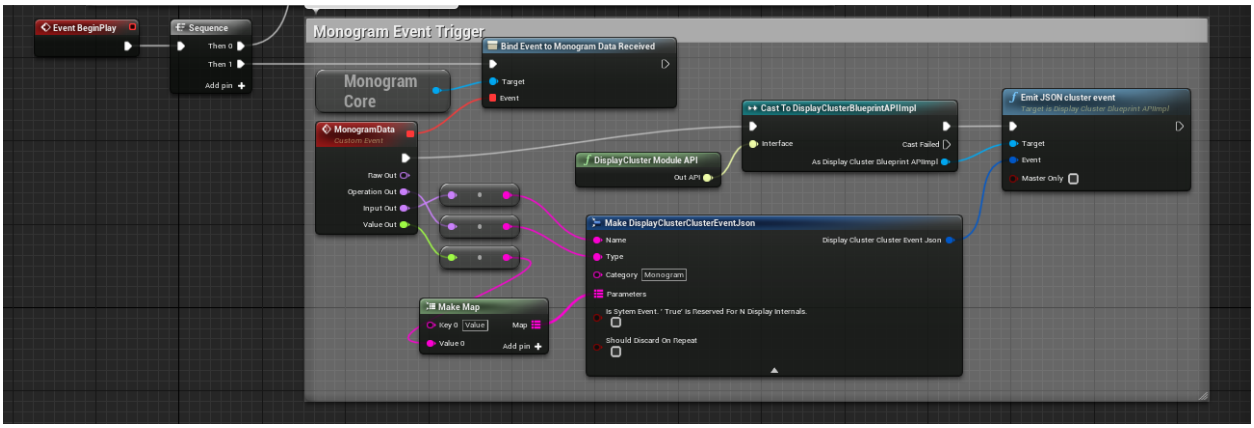
Event Graph > Event BeginPlay

This is to setup the class to wait and listen for any cluster events being created.



BeginPlay event in the ClusterEventListener blueprint event graph

An event is set up and is waiting for data to be passed from the Monogram device.



Event BeginPlay in the ClusterEventListener blueprint event graph. Continuing above.

This part starts with a *Bind Event to Monogram Data Received node* which will wait for data to be sent from Monogram and then pass it to a custom event (i.e. *MonogramData*). This event will break down the data received into the four pins which each represent different bits of information.

Pin Name	Description	Example
Raw out	This is the data exactly the way it is sent over from Monogram.	aperture+-1 (preset, operation, and value)
Operation out	This is the operation to be performed on the value and will determine how you process the value.	When using the dials Monogram will pass a plus sign (+) therefore you need to get the current value and add the passed in one. When using sliders Monogram will pass an equals sign (=) therefore you need to set the current value as the passed in one.
Input out	This is the name of the preset used in Monogram which can be found in the input.json file for the Monogram blueprints plugin.	aperture, focusDistance, transformX, pitch, etc.
Value out	This is the value used to make the changes to the setting.	increasing the pitch by 1 or setting the current aperture to 20.

Now for the portion that can be applied to any cluster event. See Emitting Cluster Events from Blueprints.

The data from Monogram is to be passed to the *Make DisplayClusterClusterEventJson* node. The operation is being passed to the type pin, the input is being passed to the name, and the value is being pass to the parameters using a Map with the keyword 'Value'. The data is combined with the DisplayCluster Module API reference node into an *Emit Cluster Event* node.

Event Graph > Event Destroyed

Best to destroy the listener if the pawn is ever destroyed.

Destroyed event in the ClusterEventListener blueprint event graph

Monogram Creator App

To set up the Monogram integration with Unreal blueprints the plugin must be added to the Creator App by navigating to File>Preferences>Integrations, click the plus sign, and navigate to the plugin folder where the input.json and config.json file are located. Those files are saved in the folder called 'Integrations' within the project folder.

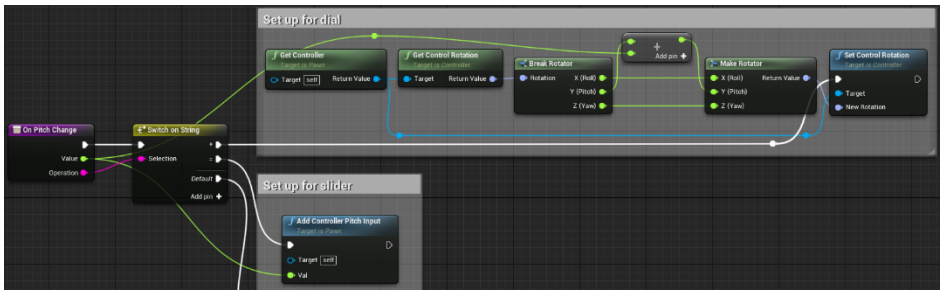
Before creating the functions it's best to decide which camera functions are going to be controlled by which Monogram control. This will be important in the next steps because it will drive how you handle the data being passed into the functions.

BP_SamplePawn blueprint: pt. 2

Functions

Start by creating a function for each of the camera functions that are to be changed. Each function should have inputs for the value and the operation. The operation will be used to determine how the camera function will be updated by the value.

For example, this is a set up to control the pitch. The 'Set up for dial' path is taken when the operation is '+' because the dials are constantly sending in data to increase the current value by (ex. pitch+1 or pitch+-1). The 'Set up for slider' path is taken when the operation is '=' because the sliders are used to set the exact value (ex. pitch=45).

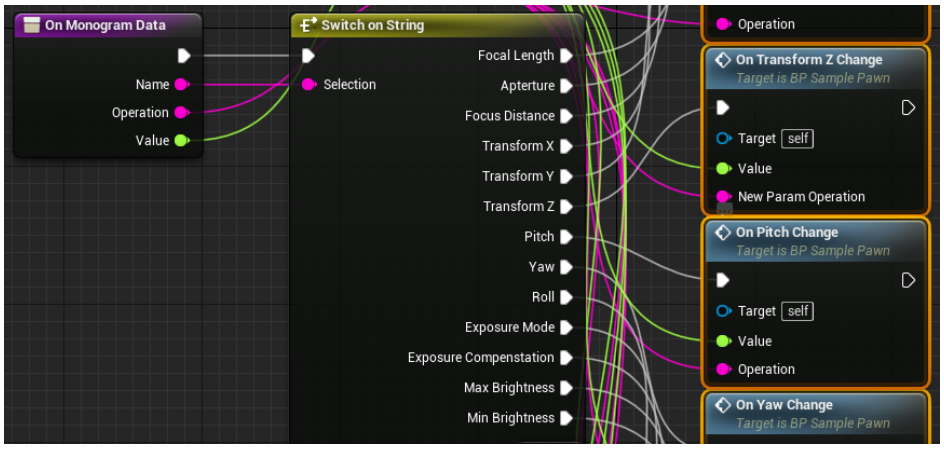


Example function from the BP_SamplePawn blueprint

Note: not all functions need to be set up for multiple operations. This will depend on which types of Monogram controls are to be used and how they pass data to Unreal.

On Monogram Data function

Now that all the functions have been setup they need to be called.



Function called On Monogram Data in the BP_SamplePawn blueprint. It uses a switch node to identify which function the event needs to call to reflect the Monogram data sent.

The data that is being passed into the switch node is the Name from the cluster event therefore each of the pins will have to correspond with a name being passed from the Monogram hardware.

- custom Monogram preset name
  list: focalLength, aperture, focusDistance, transformX, transformY, transformZ, pitch, yaw, roll, exposureMode, exposureCompenstation, maxBrightness, and minBrightness

Each of the pins are to be connected to the corresponding function was just created. Each function will need to be passed in the Value and Operation.

Launching nDisplay

See nDisplay Quick Start.

Currently only one application that uses Monogram can be run at the same time. For example, you cannot have an Unreal project that uses Monogram open in the Editor and also try to run nDisplay and use the Monogram hardware.

Adding New Camera Function/Setting

To start off, the inputs.json file from the plugin needs to be updated with the information about the new setting. Below is an example of how to set it up.

```
{
   "label": "Focal Length",        //** the name that will be displayed by default in the Creator App
   "name": "focalLength",          //** the name that will be sent to Unreal and used in the Switch on String block
   "step": 1,                      //** the value that each turn of the dial will increase/decrease by
   "range": [ 0, 100.0 ]           //** the minimum and maximum possible values (not required)
  },{
   "label": "Aperture",
   "name"...
```

***don't copy the explanations into the json file, this will prevent the json file from working properly in the Creator App*

Next create a new function in the blueprint for the class you are controlling and design the blueprint so it will reflect the desired changes in Unreal. The plugin must be (re)loaded into the Creator App as an integration (see Monogram Creator App above)

Then another pin needs to be added to the *Switch on String* block in the On Monogram Data function (in BP_SamplePawn blueprint) using the exact Name from the inputs.json file. Once a function has been created it needs to be connected to the pin like the other existing functions are and the inputs passed in.

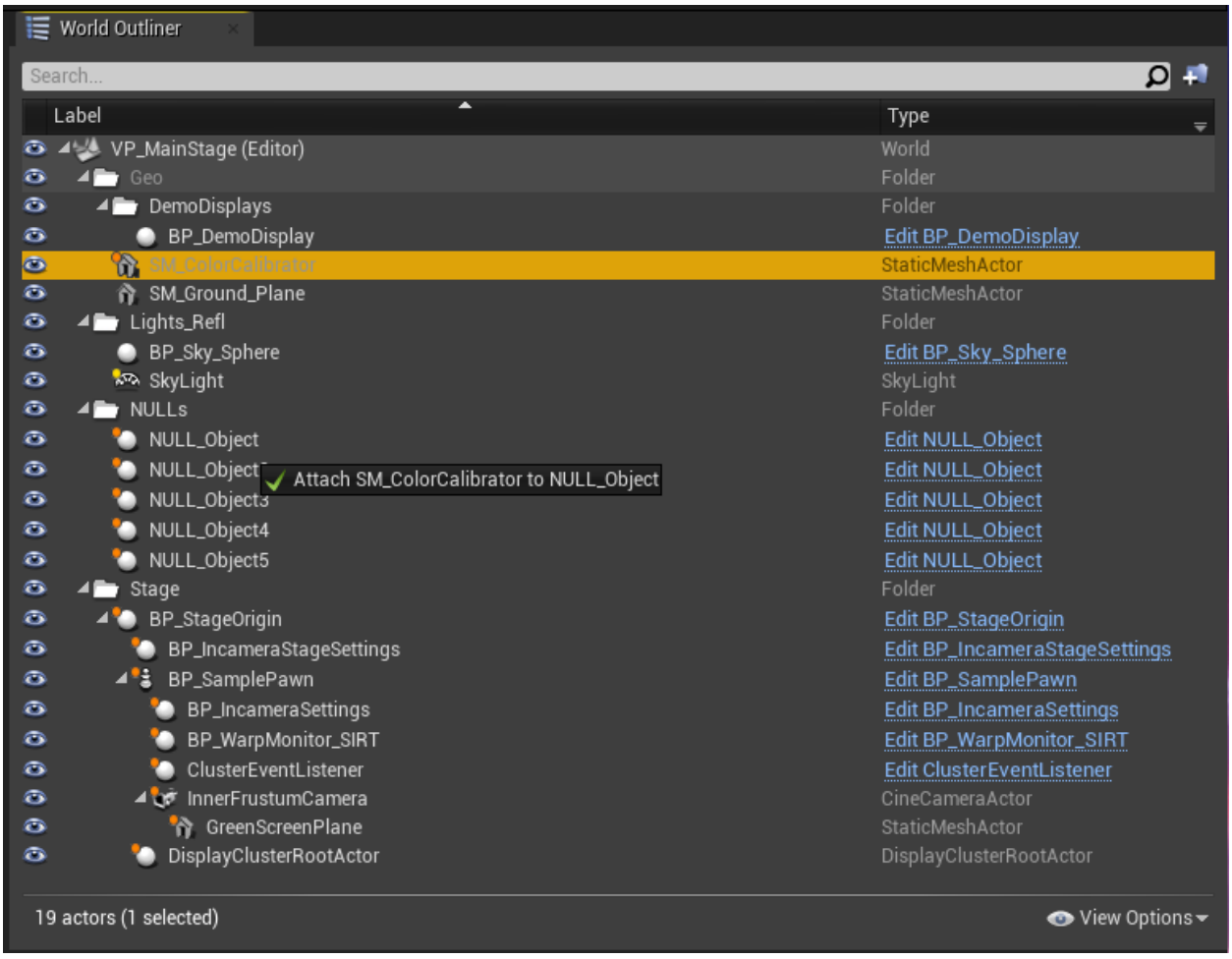
NULL Objects

Monogram NULL Objects offer a solution to the more involved alternative of manual blueprint editing. This offers a drag and drop solution to relocate assets on the fly, potentially also offering a method for input into the custom controls of an object through generic input.

This system is a set of pre-configured pawns to which your assets can be added to as children allowing for easy configuration in Monogram Creator by simply selecting one of the pre-made NULL Object's attributes.

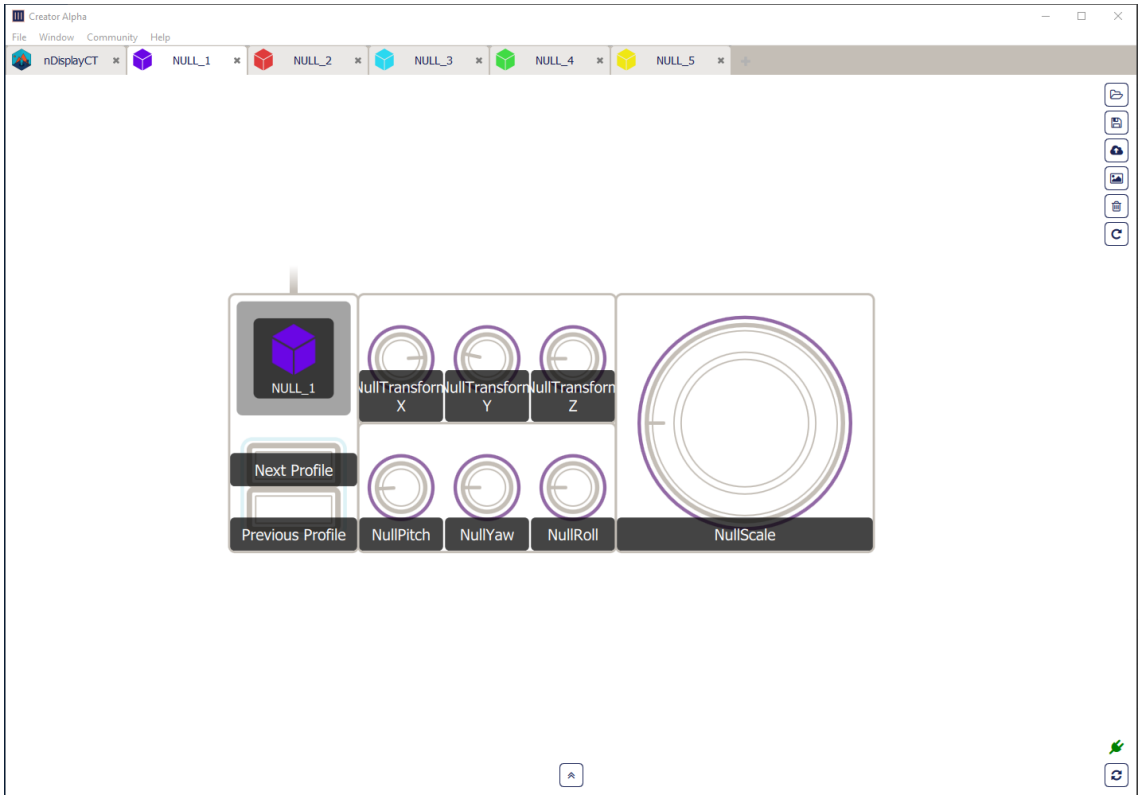
Usage

To control an asset using the Null Object Profile first ensure your asset is set to Moveable, then drag your asset onto one of the NULL_Objects in the World Outliner to make it a child.



While hovering over a NULL_Object you will see the prompt to Attach your asset to the NULL_Object

We've included example profiles for Monogram Creator in the project. The profiles come pre-set with translation and rotation on all three axis, scale and a profile switcher button.



Adding New Null Object

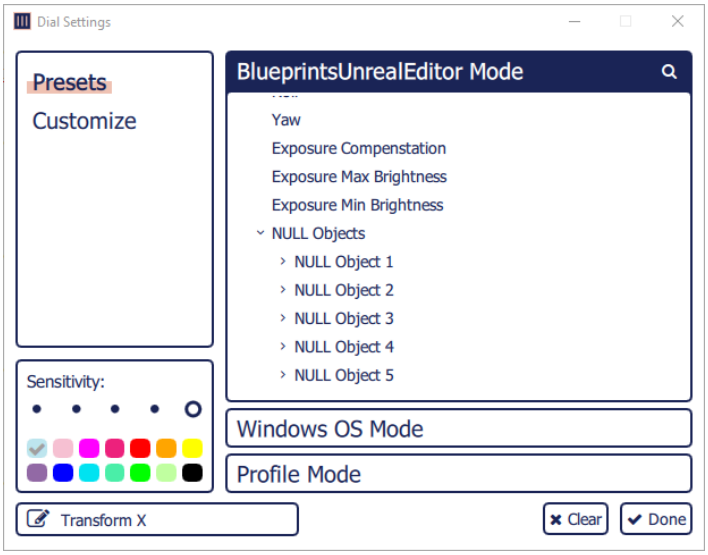
By default the project is set up to work with five null objects but it is very simple to add another one.

- Unreal
  - World Outliner
    - Duplicate one of the existing NULL_Objects
    - Make a new object and set it as moveable
    - Set it as a child of the new NULL_Object
  - Blueprint
    - nothing has to be done here because it automatically looks for all instances of the NULL_Object class and puts them in an array
- Monogram input.json file
  - Copy the following code and add it to the inputs.json file at the bottom before the last ']'. Then based on the last number used at the end of the file, update all the '[number]' to the sequential number.

input.json

```
,{
  "label": "NULL Objects.NULL Object [number].Transform X",
  "name": "[number]_NULL_transformX",
  "step": 5
},{
  "label": "NULL Objects.NULL Object [number].Transform Y",
  "name": "[number]_NULL_transformY",
  "step": 5
},{
  "label": "NULL Objects.NULL Object [number].Transform Z",
  "name": "[number]_NULL_transformZ",
  "step": 5
},{
  "label": "NULL Objects.NULL Object [number].Pitch",
  "name": "[number]_NULL_pitch",
  "range": [ -90.0, 90.0 ],
  "step": 1
},{
  "label": "NULL Objects.NULL Object [number].Roll",
  "name": "[number]_NULL_roll",
  "range": [ -90.0, 90.0 ],
  "step": 1
},{
  "label": "NULL Objects.NULL Object [number].Yaw",
  "name": "[number]_NULL_yaw",
  "range": [ -90.0, 90.0 ],
  "step": 1
},{
  "label": "NULL Objects.NULL Object [number].Scale",
  "name": "[number]_NULL_scale",
  "step": 1
}
```

- Monogram Creator App
  - Create a profile for each object
  - When adding a preset to a control each object will have their own subsection for their controls



Known Issues

Monogram Console is connected but not working	A. Ensure only one (1) instance of Unreal Engine is running B. Close and restart Monogram Creator
Controls work in Editor but not when launched through nDisplay	Close Editor before launching nDisplay
Monogram Creator is connected, but controls don’t work	Ensure your profile is set to nDisplayCT and that assigned functions are from nDisplayCT Mode