Latest Manual URL

https://docs.google.com/document/d/1slQ6CiH0NkkO8xIbc81xAmmyzbnSbz9fnlUemrBLRUo/

Version 1.x Manual URL

https://docs.google.com/document/d/1VELi43m4RbyMBNHbQ0AahYXsT5Pb2ymOit8ImTVdDzc/

Technical Support URL

Discord: https://discord.gg/6UGu7sy

Revision History

v2.0.0

- Rebuild from the ground up based on user feedback
- JSON calibration files are using a new format, v1.x JSON that need to be updated to v2.x can be submitted to us for updating. Currently a JSON updater is not included as will be bundled in a future update.
- Updated UI
- Mesh control points are controlled differently now, split into Corners, Rows, Columns and Point selection adjustments (custom point group selection in a future update)
- Added white balance controls into UI and shader
- Added keyboard UI and mesh selection controls
- Reduced main scripting inspector UI to avoid confusion and added extra help box messages
- Allow customizable hotkeys

v1.2.0

- Bug Fixes:
    - Avoids null references when recreating cameras
    - Removed more GetComponents and replaced with direct references for more efficiency
- Added Horizontal and Vertical Perspective Support
- Added warning messages when Game window does not match projector resolution
- New Qoobit Discord Support Channels added. Please join at: https://discord.gg/6UGu7sy

v1.1.0

- Bug Fixes:
    - Some fields not being serialized because of editor changes not dirtying scene

    - Loaded calibrations not properly being initialized into UI
    - Control point vertex calculations for edge cases
- Rebuilt code without requiring Substance textures. We had been having a lot of feedback with different versions of Substance not working with Unity 2018, so we reimplemented the shader as a standalone not requiring Substance. This means that in this update, it will be backwards compatible with Unity 2017. Adjusting fades should be more responsive and less buggy with this new implementation.

v1.0.5

- Added hard limits to the **Projection Warp System** editor to prevent inputting unsupported values
- **Overlap** and **Projection Camera Space** are also hidden if there is only 1 camera being used
- Major code refactoring for increased efficiency removing many instances of *GetComponent*
- Moved many prefabs and substances to the resources folder so that they can be referenced and instantiated in the new workflow behind the scenes
- Added version number to the calibration json in case future versions have different formats
- Added sample scene for use with Unity Video Player.

v1.0.4

- Major update to support Substance in Unity for Unity 2018+. Primarily removed the use of deprecated **ProceduralMaterial**.

v1.0.3

- Updated **Control Point** and **Projection UI** prefab to be initially disabled to prevent initial blip when launching from build.

v1.0.2

- Added support to launch a default json configuration file if it is located in the same directory as the exe. The filename must match **default_calibration.json**.

v1.0.1

- Minor bug fixes

v1.0.0

- Initial submit

What does it do?

This Unity Asset allows you to manage multiple unity cameras to produce large, seamless horizontal or vertical projection panoramas with one controller. It can be used for flat projection surfaces or curved panorama surfaces.

Prerequisite

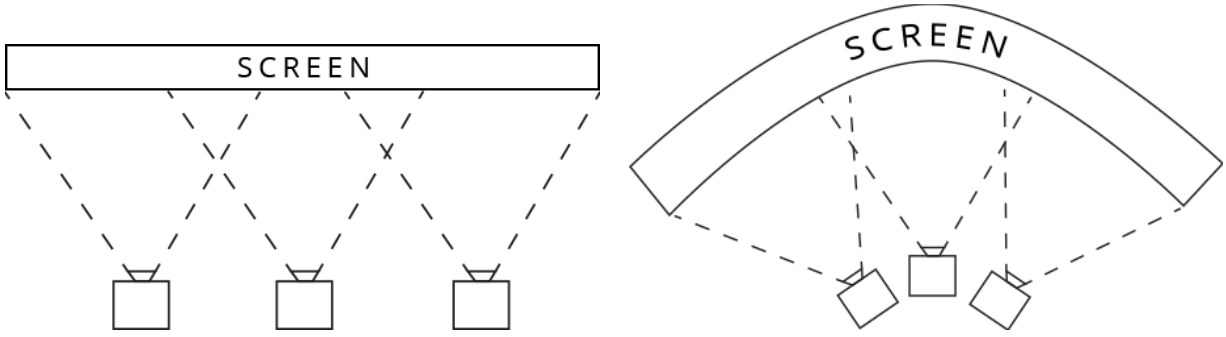
The plugin requires at least one projector, with all projectors having the same resolution settings. It is recommended that all projectors used are of the same model for best color matching.

Supported Projector Configurations

Place the projectors at a distance where the **image slightly overshoots the projected surface**. If possible, **try to align the top or bottom edge** to the projection surface for minimal tweaking for best results. Also, make sure that there is at least 10% overlap between each projected image. Be sure to reorder the cameras in sequence from left to right inside the display settings.

Horizontal Panorama

Bird's Eye View

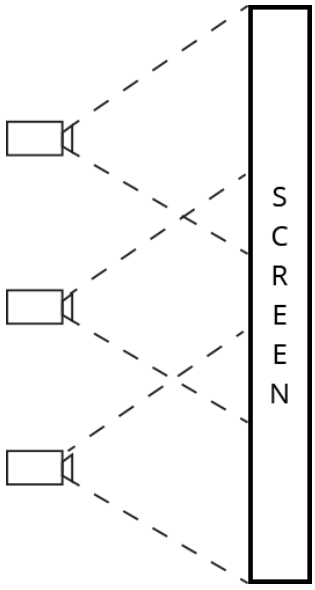


Horizontal Line **Horizontal Curved**

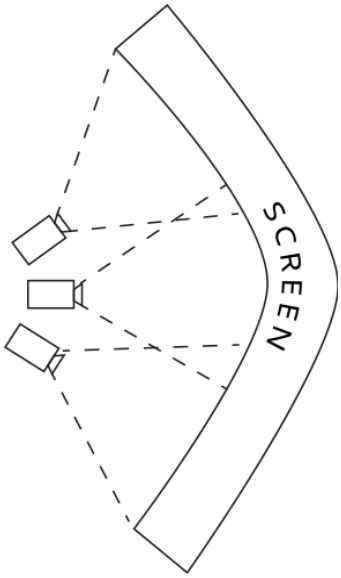
The projectors need to be lined up either horizontally in a straight line or curved with adequate overlap between each projector image on the left and right.

Vertical Panorama

Side View



Vertical Line



Vertical Curved

The projectors need to be lined up either vertically in a straight line or curved with adequate overlap between each projector image on the top and bottom.

How it Works

We've scripted a system prefab for drag and drop use in your scene. The **ProjectionWarpSystem** script attached to the prefab controls the shared properties of all the cameras being used.

Please view our tutorial video for the functions of this version:
https://www.youtube.com/watch?v=_6Qgq60m8DI

(v1.x Tutorial: https://www.youtube.com/watch?v=_6Qgq60m8DI)

We begin by entering the basic global settings that will be using for the setup.

Projection Warp System Component Values

Initialization File	JSON file to use at start up. If left blank, none will be used and the current mesh settings will be used. If there is a **default_calibration.json** file in the same folder as your exe build file, it will be loaded automatically if this field is left blank.
Arrangement	Are the projectors horizontally or vertically placed in the physical world? Is what you're trying to draw using an orthographic or perspective camera?
Projector Resolution	The resolution settings on the projectors. Be sure that these are synced across all projectors (and Game windows).
X Divisions/Y Divisions	The number of divisions to use for the render target for shape adjustments. Minimum value of 1 division for each axis.
Projector Count	The number of projectors being used. This will create this amount of Unity cameras and render texture projection targets.
Near	Near clipping plane for all source cameras
Far	Far clipping plane for all source cameras
Overlap	The amount of overlap between each camera. This will reduce the overall area of the panorama, but is necessary for making seamless blend transitions. In HORIZONTAL_ORTHOGRAPHIC and VERTICAL_ORTHOGRAPHIC arrangements, this refers to the amount of positional offset for each of the cameras to create the overlap. In HORIZONTAL_PERSPECTIVE And VERTICAL_PERSPECTIVE, this refers to the amount of angular offset for each of the cameras to cover the desired field of view
Projection Camera Spacing	The amount of space between each render texture mesh. If 0 is used, it is advised to set layer masks for each Projector Image or to use a non zero value to avoid any potential overlaps between adjacent meshes.
Field of View (Perspective Only)	Field of view you wish to cover with the system. In HORIZONTAL_PERSPECTIVE, the value represents the

	horizontal field of view angle you wish to cover in your scene. In VERTICAL_PERSPECTIVE, the value represents the **vertical field of view** angle you wish to cover.
Selected Mesh	Index of the currently selected mesh for editing. -1 means not selected.
Show Grid	Shows the base mesh grid for current mesh
Show Selected Grid	Shows the selected grid lines for current mesh
Show Control Points	Show the base mesh control points
Show Selected Control Points	Show the selected control points
Top Blend Range	0 to 1 value for the top edge blend distance (0 = no gradient, 1 = edge to center gradient). The gradient transparency runs linearly from 0 to 1.
Top Blend Choke	0 to 1 value for the top edge starting position (0 = top of mesh, 1 center of mesh)
Bottom Blend Range	0 to 1 value for the bottom edge blend range (0 = no gradient, 1 = edge to center gradient). The gradient transparency runs linearly from 0 to 1.
Bottom Blend Choke	0 to 1 value for the bottom edge starting position (0 = bottom of mesh, 1 center of mesh)
Left Blend Range	0 to 1 value for the left edge blend range (0 = no gradient, 1 = edge to center gradient). The gradient transparency runs linearly from 0 to 1.
Left Blend Choke	0 to 1 value for the left edge starting position (0 = left of mesh, 1 center of mesh)
Right Blend Range	0 to 1 value for the right edge blend range (0 = no gradient, 1 = edge to center gradient). The gradient transparency runs linearly from 0 to 1.
Right Blend Choke	0 to 1 value for the right edge starting position (0 = right of mesh, 1 center of mesh)

Tint	White balance multiplicative color for color correction. Default is white.
Selected Vertex	-1 if no vertex selected for current mesh. Editing mesh defines which corner, row or column is being selected.
Mesh Edit Mode	Values: **NONE, CORNERS, ROWS, COLUMNS, POINTS** **CORNERS** mode will only have 4 control points which will linearly stretch the mesh to the corner anchor points. The others will allow offset adjustment on a per point basis relative to the interpolated base positions defined by the corners.
Reference Game Objects	Do not edit. It is used for faster referencing of related dependency scripts

NOTE When calibrating, please remember to check the **Projector Warp GUI** item under the **Visibility** heading to display the UI for making adjustments. Alternatively, you can press the ` key to toggle the GUI in play mode. It will only be shown on display 1 because Unity UI has bugs dealing with mouse positions across multiple displays.

Projection Mesh Component Values

Plane Distance	Distance of projection mesh from orthogonal camera. Make sure there is adequate space so that the cubes for the control points are visible (recommended value 0.4)
Index Appear Duration	Amount of time the projector's index is displayed on screen when switching displays for edit
Left Fade Range	Shader Mask Left Fade Range
Left Fade Choke	Shader Mask Left Fade Choke
Right Fade Range	Shader Mask Right Fade Range
Right Fade Choke	Shader Mask Right Fade Choke
Top Fade Range	Shader Mask Top Fade Range
Top Fade Choke	Shader Mask Top Fade Choke

Bottom Fade Range	Shader Mask Bottom Fade Range
Bottom Fade Choke	Shader Mask Bottom Fade Choke
Tint	White balance multiplicative color for color correction. Default is white.
Show Grid	Shows the base mesh grid for current mesh
Show Selected Grid	Shows the selected grid lines for current mesh
Show Control Points	Show the base mesh control points
Show Selected Control Points	Show the selected control points

After setting up the parameters above, a render texture and mesh will be generated for each projection camera.

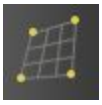
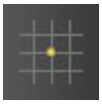
Hotkeys

Hotkeys can be remapped by editing the values in the **Projection Warp System Keyboard Input** component.

F1	Help toggle
`	Toggle UI display
F	Show / hide mouse cursor
1 - 8	Select Projector # to edit. The selected display with have its index shown for 3 seconds before fading out.
W, A, S, D	Choose selection / move selection. When using point group edit mode, these keys are used to select a specific point, row or column, or move the selected control points.
F5	Reset selection offset. When you have selected a point group (blue indicates selection), you can reset the offset back to (0,0).
R	Reset mode. Disables point group selection, blend, white balance and help modes.
Space	Select / deselect current point group

T, Y, U, I	Corner, row, column, point group selection mode
O	Edge blending mode
P	White balance mode
G	Show / hide base grid
H	Show / hide selected grid
B	Show / hide control points
N	Show / hide selected control points

UI Controls (Play Mode Only)

	Currently selected projector. Click to jump to next projector
	Show / hide mouse cursor
	Corner point group selection
	Row point group selection
	Column point group selection
	Single point selection
	Edge blend mode

	White balance mode
	Help / Hotkeys

Saving and Loading Calibrations

A JSON file is used to save and load the calibration settings for all the cameras in the system.

In Editor

To save out the calibration file, goto the **Projection Warp System** component and click the **Save Calibration** button. You will be prompted with a dialog to save the file.

To load a calibration file, goto the **Projection Warp System** component and click the **Load Calibration** Button. You will be prompted with a dialog to search for a previously saved JSON configuration file.

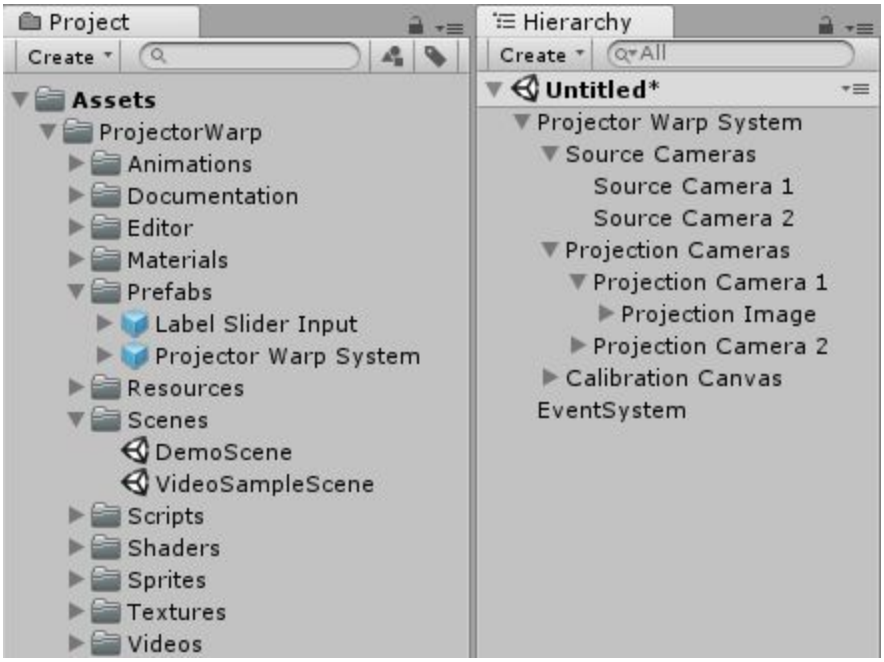
In Play Mode

To save out the calibration file, enter a filename in the input field and click the **Save** button. It will be saved in the folder of your build .exe file.

To load a calibration file, enter a filename relative to your build .exe file click the **Open** Button.

Pipeline

1. Set the physical projectors in the desired arrangement and try to lock them in place so they don't move around. Synchronize the settings across the projectors and make sure they have adequate overlap in the physical space.
2. Drop the **Projection Warp System** prefab into the current scene and position the GameObject into the region you wish to use for the projection. Animations should be done only on the **Projection Warp System** prefab. All other transforms underneath are being controlled by this script.



3. Enter the global settings for the cameras. Begin by selecting the camera arrangement.

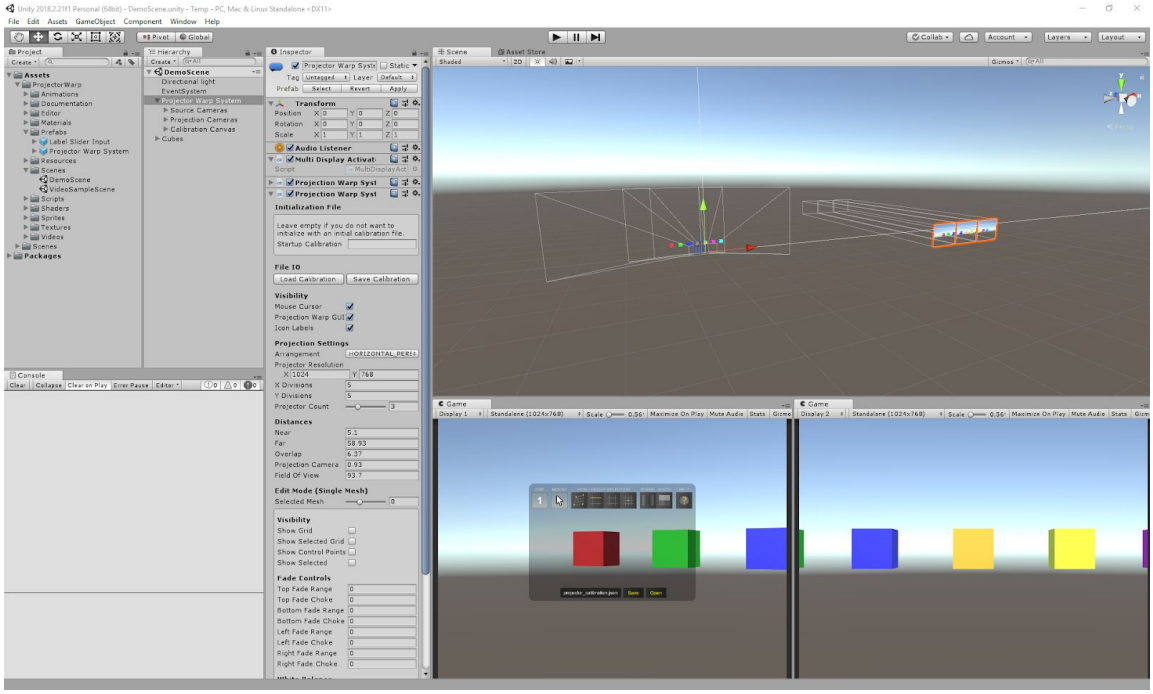
   The **Projector Resolution** should be set to the exact resolution of the projectors being used. By default they have been set to 1024x768. Be sure to set all your Game windows to the same resolution or you will see a warning.

   The **X Divisions** and **Y Divisions** are arbitrary determined by you and should be set to a resolution that suitably represents the projection surface. The more divisions you have, the more manual tweaking will be needed on the control points at the later stages. However, with more control points, you will have a more accurate deformation.

   Camera Count needs to be set to the number of projectors being used. They will be centered to the parent prefab.

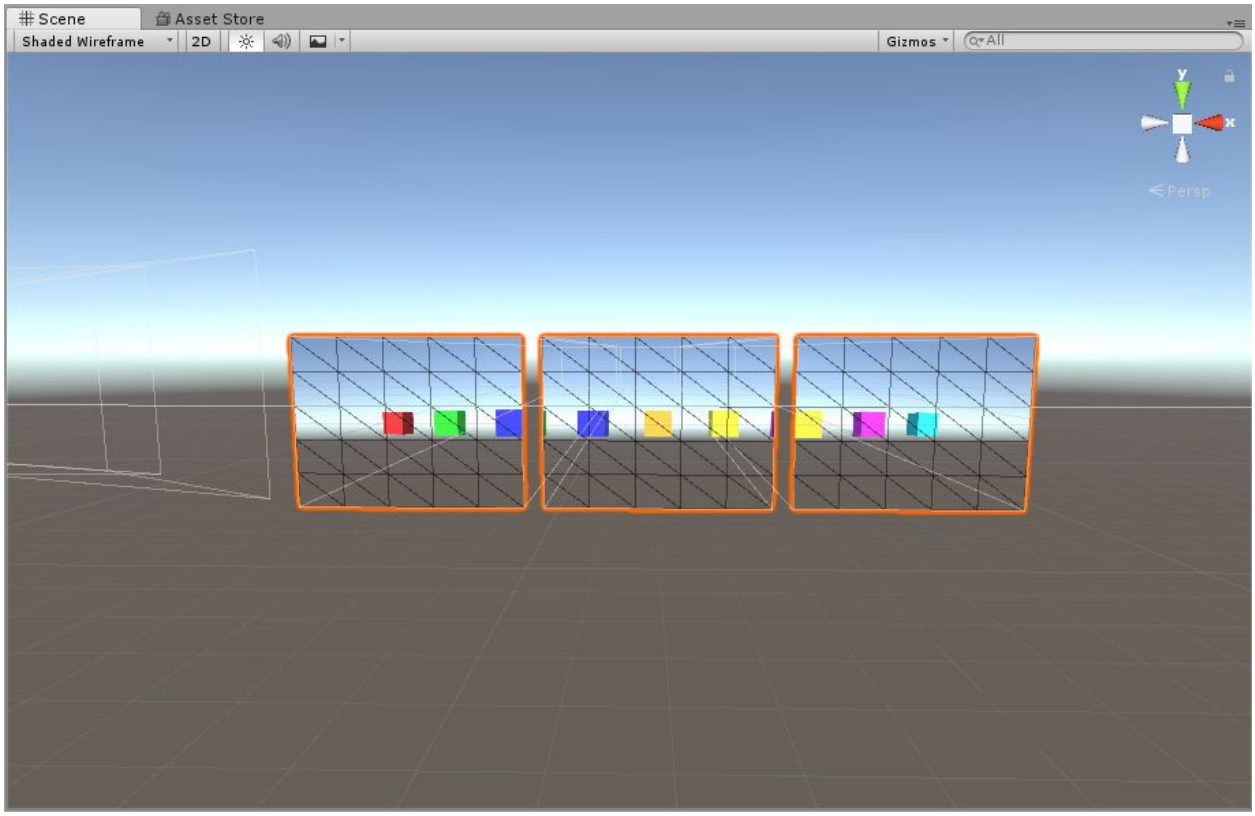
   Near and **Far** are the clipping planes that will be used for all the Orthographic Cameras.

   Overlap is for setting the amount of overlap region between each camera in the panorama. Remember that in **ORTHOGRAPHIC** modes these are positional offsets and in **PERSPECTIVE** modes these are rotational offsets. Make sure that a non zero positive value is set for best results. A good point of reference is to try and get at least 10% of the image overlapped.

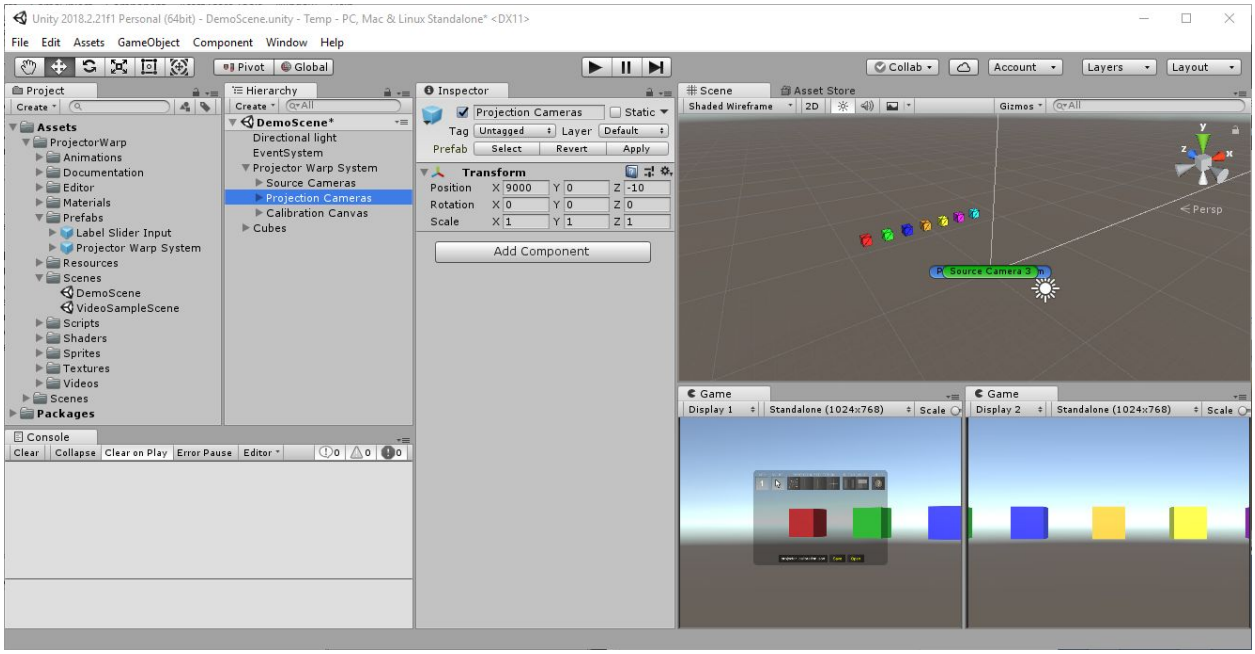


WARNING	After these global values have been set, avoid making changes to them since they will re-generate the meshes for the projections. This includes, but is not limited to, changing the X Divisions, Y Divisions and Camera Count.

4. After setting the values above, each camera will be rendered to their own **Render Texture** which are then projected onto a deformable plane listed under **Projection Cameras**. Since these planes are rendered onto different displays, it is a good idea to move them out of the viewing range of the **Source Cameras**. To move them, just move the **Projection Cameras** GameObject.

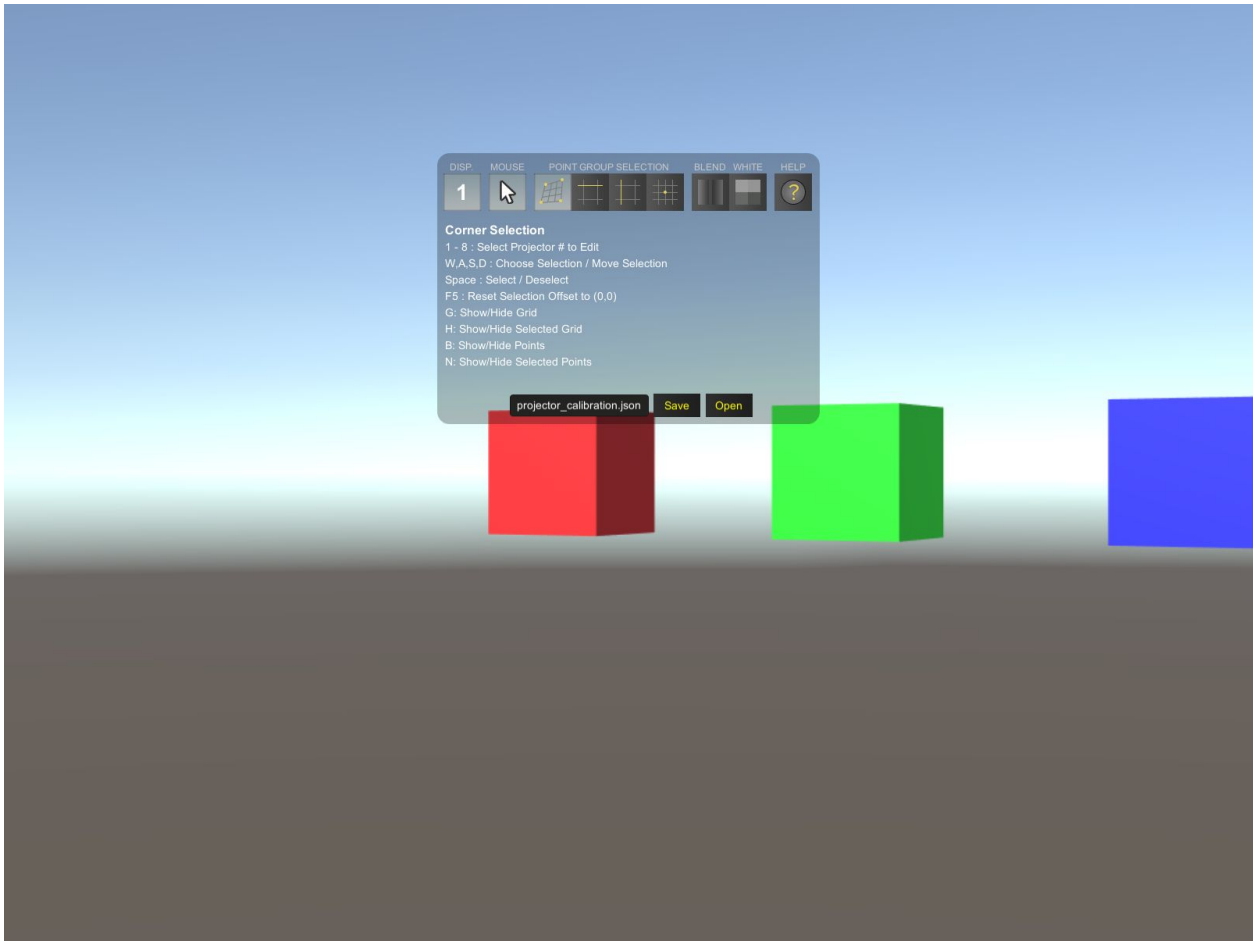


In the event, that they cannot be moved out of the viewing range of the **Source Cameras**, create layers and assign them to the **Projection Image** GameObjects so that they will not be rendered through the **Source Cameras**.

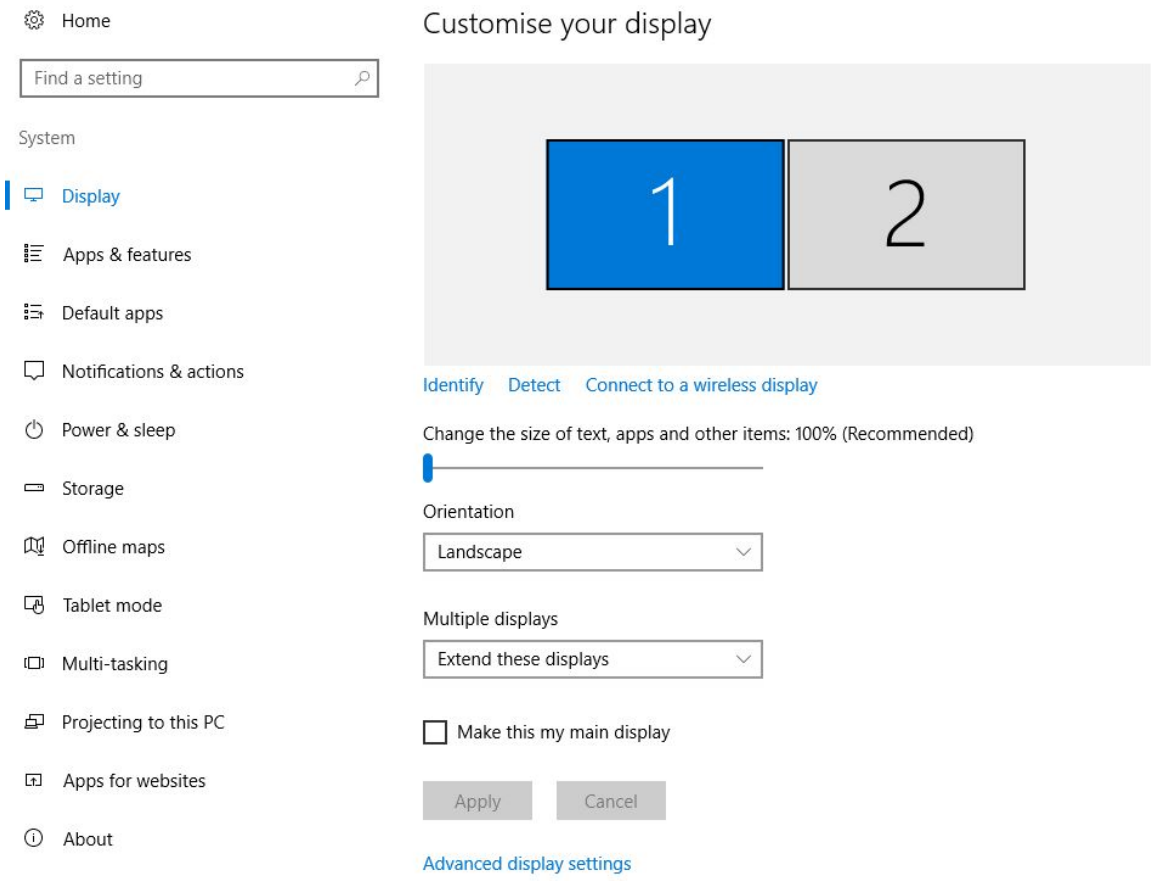


The **Projection Camera Space** can also be used to space apart the generated mesh planes so that their images do not overlap on the cropped out areas of the displays.

5. At this point we need to build out and run the program since the Unity Editor doesn't allow simultaneous displaying of multiple full screens during play mode. In editor mode, you can still swap between displays, but it's not very useful for trying to line up the images. After building and running, your scene will be presented in its default state.



Double check that all displays are in the correct sequence. If not goto the display settings and rearrange them accordingly.



The controlling UI has all been consolidated on display 1 since Unity Screen Space UI doesn't work properly beyond display 1.

If you think the UI dialog box is too small, you can change its scale in the editor and rebuild for a larger UI experience.

TIP	Try to physically move the projectors to match a general point of reference for the horizon. This usually means trying to match the images so that the top or bottom edge lines up with the projection surface as much as possible. Slightly overshooting the surface with the projector images is recommended since you can pull the image back inside the viewing range.

6. Pressing the ` key will reveal the UI and **F1** will reveal all the hotkeys.

   Pressing **G, H, B,** or **N** on the keyboard will toggle the grid, selected grid, control points and selected control points for each selected mesh on each display.

These are a useful reference for finely tuning the overlap regions and knowing exactly which points have been selected. Hiding the grids and control points afterwards are a good way to see how good the overlap region is.

Use the **Space** key to selected the current highlighted point group and **W, A, S, D** to move the point group around. By default they will move at increments of 1. Holding down **Shift** will scale the increments down to 0.1 and **Ctrl + Shift** will adjust increments to 0.01.

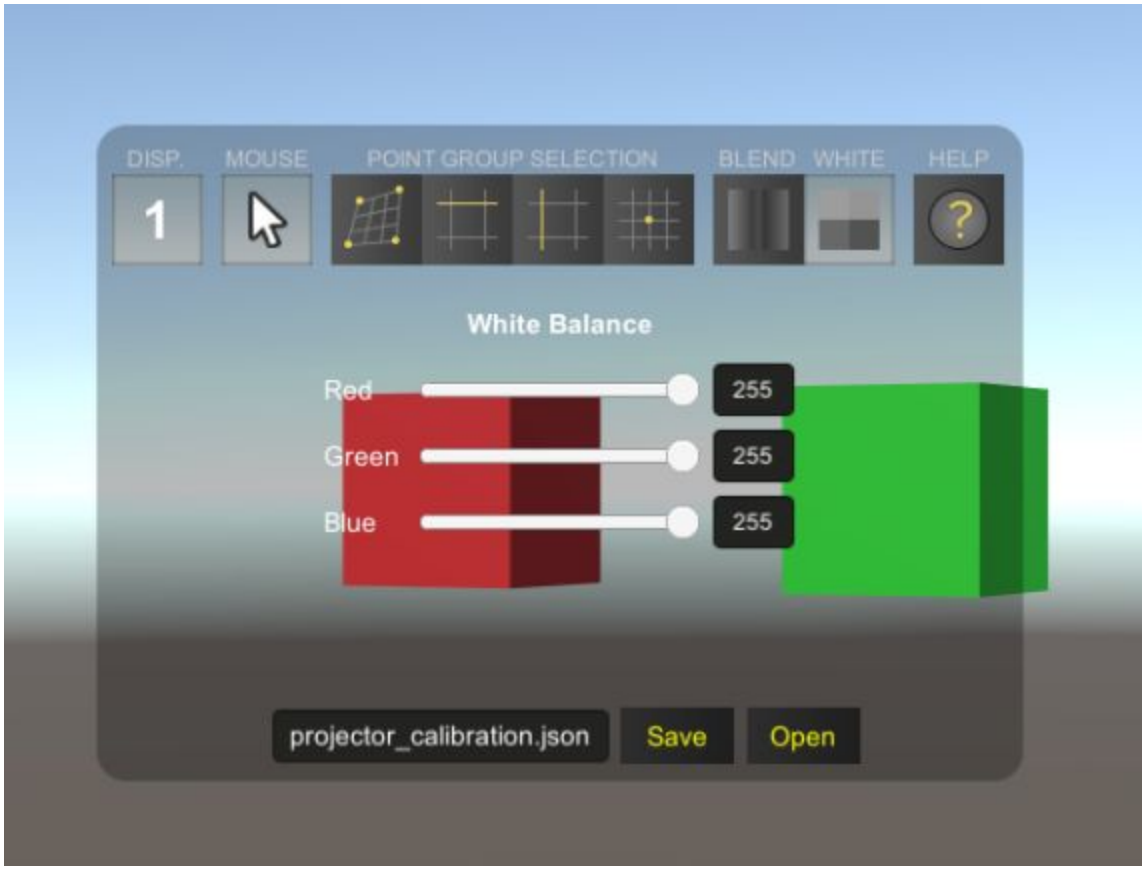
Use **1 - 8** to switch between meshes for editing.

T, Y, U, I can be used to switch between point group selections

7. After lining up the images properly, it should be looking pretty good with minimal to no blurs in the overlapping regions. Except now, the overlapping regions will be displaying overly bright. We will address that in this step. Switch to blend settings by pressing **O** on the keyboard or clicking the  button on the UI.

There will be **range** and **choke** slider/input values that can be adjusted to create a desired gradient. These values are linked to a shader that is responsible for procedurally generating the alpha gradient used for the blend.

8. After getting a good blend, you should not see overbright areas, however if using non calibrated or off brand projectors, you may have color discrepancies. To address this white balance problem, press **P** on the keyboard or click the  button on the UI.



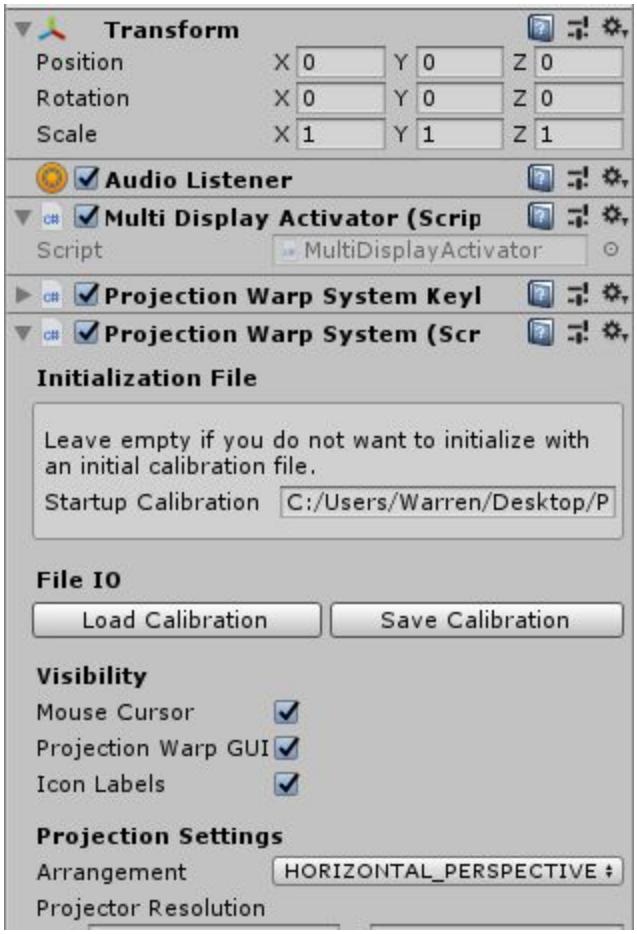
There will be RGB slider/input values that can be adjusted to try and match the colors on the two projectors.

9. SAVE!

The most important step! After making the adjustments above, it would be a real pain if you forgot to save (since changes made at runtime are not saved automatically). Clicking the **Save** button will store all the changes into a JSON file that can be used later. Double check that the JSON has been saved and exists in the proper location and close the

executable.

10. Go back to Unity and click the **Load Calibration** button in the inspector and select the JSON file that was just exported. This will bring the settings in as a default and you're good to go. Any future builds will utilize these calibration settings.



TIP If you're running the exe on a computer without Unity and need to update the calibrations, rename your calibration filename to **default_calibration.json** and keep this file in the same directory as your export executable. If the plugin finds this file at startup, it will opt to use this instead of the provided json file.

Potential Future Upgrades

- Cube Projector Mapping
- SteamVR Integration

If you have any other features you would like to suggest or if you find any bugs or problems, please send us an e-mail at info@qoobit.com or visit us at our Discord channel for more immediate responses: https://discord.gg/6UGu7sy

Thanks for using our asset!