UNITY PLUGIN MANUAL

v0.4.0

Contents

Welcome

Welcome to the 8i Unity Alpha programme!

We’re looking forward to finding out how you use our custom plugin and getting your feedback. There are a few things to cover before we let you loose, so have a read.

What is the 8i Plugin?

The 8i Unity Plugin allows developers to add 8i’s fully volumetric video to any Unity experience. It’s like embedding traditional 2D video content, except you get to walk around ours as it plays!

Why are we doing it?

- To introduce developers to our technology, its possibilities and constraints
- To gather feedback from you
- To build an 8i community of early adopters, developers, content creators and VR enthusiasts

Requirements

- Unity 5.2.3 or greater
- Windows 7 or later

Getting Started With Unity

Using Unity

If you're brand new to Unity then there are some excellent tutorials for learning the interface and it's features here;
https://unity3d.com/learn/tutorials

VR Development in Unity

If you are new to VR development in Unity, there are some great tutorials here;
https://unity3d.com/learn/tutorials/topics/virtual-reality

Installation

Downloads

All the files you need to access, including the plugin, example content and documentation can be found in the dropbox folder that was shared with you.

Adding the 8i Unity Plugin

1) Ensure you have Unity 5.2.3 or greater installed
2) Create a new Unity Project
3) Extract the '8i' folder from the plugin zip file into your project

Updating the 8i Unity Plugin

1) Close the Unity Editor
2) Completely remove the old plugin by deleting the '8i' folder from your project.
   No files should be left behind that had previously been installed for the 8i plugin
3) Extract the '8i' folder from the plugin zip file into your project

Using the 8i Unity Plugin in your Project

Examples

There are a set of example scenes which show the different features of the 8i Unity Plugin within the '8i/examples' folder.

Unity Virtual Reality Support

Using Unity's built in Virtual Reality support is as simple as making enabling it, and adding the HvrRender component to the rendering camera.

Steps

1. The 'Virtual Reality Supported' checkbox in PlayerSettings must be ticked.
2. The Main camera's 'Target Eye' should be set to 'Both'

3. The Main camera must have a 'HvrRender' component attached to it.

8i Objects and Components

Scene Objects

●HvrActor

This component acts most like Unity's built in Mesh Filter component. Where instead of a mesh, it takes as input a folder from the Unity Project which contains .hvr frames.

How to Create

- Right click in Unity Scene Hierarchy
- Select '8i/Create HVR Actor'

Parameters

Data	Slot which a folder containing .hvr frames can be dragged to from the Project. Doing this will automatically create a HvrAsset, set it up with the data folder and then be assigned to the HvrActor
Actor Render Method	The rendering method the actor will use to draw into the scene. Options ● Standard ○ Supports per-actor color-grading ○ Supports custom shaders ● Direct ○ Faster than the 'Standard' mode ○ Does not support color grading. ○ Does not support custom shaders ○ Renders directly into the framebuffer Note This setting will be overridden if the rendering camera's HvrRender component is not set to 'Standard'
Asset Render Method	The rendering method the HvrActor's HvrAsset will render with. Options ● Point Blend ○ Renders the actor with smooth points which soften the look of the actor ● Point Sprite ○ Renders the actor with hard edges. ○ Renders faster than PointBlend
Occlusion Culling	Whether when rendering the HvrActor the HvrRender should use Unity's Occlusion Culling system to check whether the object is visible. A wireframe sphere will be drawn in the Editor SceneView to show the size and location of the bounding sphere which is used to check the visibility of the HvrActor. Options ● Occlusion Radius Offset Allows you to offset the size of the bounding sphere radius.

●HvrActor Clone

This component allows you to clone a specified HvrActor and render copies of it with a reduced performance cost.

The performance savings are on the CPU and come from letting the clone share the source's HvrAsset and reducing the total disk read and file decompression. Three clones are still just as expensive to render as three regular HvrActors.

How to Create

- Right click in Unity Scene Hierarchy
- Go to '8i/Create HVR Actor Clone'
- Assign the 'Source Actor' by dragging an HvrActor into the slot, or searching the scene by clicking on the icon next to the slot.

Parameters

Source Actor	Slot for assigning the HvrActor to clone.
Actor Render Method	Functionality matches HvrActor.
Asset Render Method	This functionality is disabled, as the source HvrActor defines this setting.
Occlusion Culling	Functionality matches HvrActor.

Note: You cannot clone a clone.

●HvrAsset

The purpose of a HvrAsset object is to store the data for an HvrActor to play. A HvrAsset acts most like a MovieTexture, AudioClip or an Animation in Unity.

Unless manually creating HvrActors using code, this object will not be encountered when creating scenes as they are automatically created behind the scenes for HvrActors when actors are assigned data..

Rendering

●HvrRender

In order to render HvrActors a HvrRender component must be attached to a Camera.

There are three different render modes that the HvrRender component can be set to.
While Standard is the default and recommended mode, there are two others which each have different properties and performance costs.

	Standard	**Composite**	**Direct**
Performance	Bronze	Silver	Gold

How it renders	Renders the scene in multiple passes, where it renders each actor that is visible one by one. This allows for a lot of customization as each HvrActor can have unique shaders and be color-graded individually.	Renders the scene in a single pass by compositing into the scene by using a fullscreen shader pass.	Rendering the scene in a single pass by rendering directly into the framebuffer.
HvrColorGrading support	Allows individual color grading of HvrActors if the HvrActor 'Unity Render Method' is set to 'Standard'. If set to 'Direct' the color grading component will be ignored.	Only supports color grading if the HvrColorGrading component is attached to same camera as the HvrRender component. Invididual HvrColorGrading components on HvrActors are ignored.	Does not support color grading.

Performance information can be viewed on the component and will show the cost of rendering the current view in milliseconds. Lower is better.
Keep in mind that render performance is closely tied to the number of actors in the scene, the level of detail in the actors and the rendering resolution.

Notes

- Hvr rendering in the Unity Editor Scene-View is handled automatically and the camera uses the 'Standard' render mode

●HvrColorGrading

This component allows the user to apply color grading to the rendering of HvrActors.

To color grade a single HvrActor, just add a HvrColorGrading component to the same GameObject and set its render method to 'Standard'.
Keep in mind that if the rendering camera's HvrRender's render method is not set to 'Standard' this component will be ignored.

To color grade all the HvrActors in the scene, add a HvrRender and HvrColorGrading component to a Camera and set the HvrRender render mode to 'Composite'.

Useful Components

●PlayHvrActor

Assigning an HvrActor to this component will trigger it to begin playing as soon as the scene loads.

●TriggerHvrActor

By attaching this component to a GameObject that has a Collider and assigning a HvrActor, the HvrActor will begin playing when the Collider is hit by any collidable object.

●HvrActorProjectorShadow

Can be used in conjunction with a Projector component to project a texture onto a surface based on the bounds of an HvrActor. This is useful for creating blob shadows.

●HvrActorAudioSourceSync

Attaching a AudioSource and HvrActor to this component will automatically cause the AudioSource's playback to match the HvrActor.

Create by

1. Select (or create) a Unity GameObject
2. Click on 'Add Component' in the inspector
3. Go to '8i/HvrActor Audiosource sync'
4. Slot the HvrActor and Audiosource into the component. From now on the audio source will match the playback time of the HvrActor.

Note

- The HvrActorAudioSourceSync does not take an audio clip itself as a value. It expects the audiosource to have an audioclip assigned.
- Once you have assigned a Audiosource and HvrActor, you will not need to manage this component - If the actor is playing, the audio source will be playing.
- If the AudioClip assigned to the AudioSource is shorter than the HvrAsset's duration it will play as long as it can, if it is longer it will stop once the HvrActor stops playing.

Platform Support Android

Getting assets onto your Android device

Android is similar to desktop systems in that .hvr files are read from disk when being used to play a HvrActor. The Android HVR playback system expects that all static data is stored in a directory named “8i” in the devices external public storage.
There are two different ways you can get .hvr files into the correct location for playback on your Android device.

1. **Build External Assets Folder (manual copy)**
   Under the “8i” menu in Unity, you can use the “Build External Assets Folder” to build the required “8i” folder (full of .hvr files) and then copy this folder directly to the devices external public storage yourself. You can do this by plugging your device into a computer, and accessing the device’s file system. The public directory should contain other folders such as “Movies”, “Media”, “Music”, “Download” and others.

2. **Build and Run (Assets Packed)**
   Under the “Android” sub-menu of the “8i” menu in Unity, there is an option to “Build and Run (Assets Packed)”. This will create an APK that contains all of your data, and will add an “unpacking” scene to the project, which populates the “8i” external public folder as required whenever your application is opened. Please note that if your project uses a large amount of data, Unity may fail to build a packed APK.

   It is suggested that you use “Build and Run (Assets Packed)” for the first build after you have added a new asset to a scene (or are distributing your APK). You should otherwise use Unity’s standard build system. Once an asset has been unpacked, it will remain unpacked and accessible during future runs and builds of the application. Uploading an APK with packed assets to a device will take significantly longer than uploading an APK produced by the standard Unity build system (as the asset packed APK will be much larger).

We are aware of the APK size limits imposed by the Google Play Store. Support for .obb files is likely to be added in the future.

Performance on Android

Android performs much slower than desktop systems. It is recommended that hvr frames with point counts of 600k or less are used, with the recommended point count being around 300k.

It is recommended to use the 'Direct' HvrRender render method on Android as it is the best performing renderer.

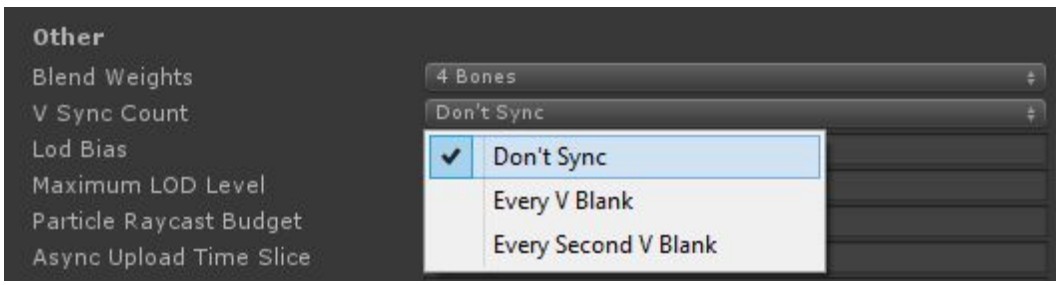
Third Party Support

Using the Plugin with SteamVR or OculusSDK

Below are some VR specific changes that need to be made to ensure compatibility with the 8i Unity Plugin.

Disable VSync in the Quality Settings

1. Open the QualitySettings menu (Found under 'Edit/Project Settings/Quality')
2. For each quality level, set the 'VSync Count' option to 'Don't Sync'



Cinema Director

Cinema Director is a third party plugin for creating cutscenes in Unity, it is available on the Unity Asset Store here.
https://www.assetstore.unity3d.com/en/#!/content/22171

Within the '8i Unity Plugin' download folder there is a zip folder under '8i/Integrations' which includes our integration for the Cinema Director plugin. The script is fully functional and can be used with the current version of Cinema Director (as of writing, version: 1.1.3.0, Jan 25, 2016).
In order to install the integration, extract the 'CinemaDirector' folder from the zip into your project

For an introduction to Cinema Director, tutorial videos are can be found below.
https://www.youtube.com/playlist?list=PLkTFhf2jQXOkn0Un8ej8THMVG8ccZH2y8

Playing an HvrActor

1. Open the 'Creator' window for Cinema Director by going to 'Window/Cinema Suite/Cinema Director/Create Cutscene'
2. Set the Actor Track Groups to '1'
3. Drag the GameObject that has a HvrActor component attached into the 'Actor 1' slot
4. Press 'Create Cutscene'
5. If it is not visible already, open the Director window by going to 'Window/Cinema Suite/Cinema Director/Director'
6. In the Director window click the '+' icon next to 'Actor 1 Group' and add a 'Actor Track'

7. Click the '+' button next to the 'Actor Track' and select '8i/Play Hvr'
8. Click the play symbol in the Director window to preview the cutscene.

Slate Cinematic Director

Cinema Director is a third party plugin for creating cutscenes in Unity, it is available on the Unity Asset Store here. It is an alternative to the Cinema Director plugin above.
https://www.assetstore.unity3d.com/en/#!/content/56558

Within the '8i Unity Plugin' download folder there is a zip folder under '8i/Integrations' which includes the integration. The script is fully functional and can be used with the latest version (as of writing, version: 1.4.5, Jul 15, 2016).
In order to install the integration, extract the 'Slate Cinematic Sequencer' folder from the zip into your project

Playing an HvrActor

1. Create a new Cutscene by going to 'Tools/ParadoxNotion/SLATE/Create New Cutscene'
2. In the SLATE window that appears, click the 'Add Actor Group' button on the left side of the screen
3. Drag the GameObject which has the HvrActor component attached into the Object slot on

the Actor.

4. Right click in the timeline and add the '8i/Play HvrActor' timeline object.
5. Press the 'Play' button

Note - The Slate Cinematic Director creates a camera for the cutscene when one is created. This camera will not have the HvrRender component attached. You will need to add the HvrRender component to the 'Render Camera' object under '★ Director Camera Root' in the hierarchy if you wish to render the scene using this camera.
You may also delete the 'Camera Track' from the cutscene via the SLATE window if you do not want to use this camera at all.

Troubleshooting

HVR Actors are not rendering

Windows

The Graphics API may not be supported

Go to 'Edit/Project Settings/Player' and make sure that the PlayerSettings 'Graphics API for Windows' is set to either DirectX11 or OpenGL2. These are the only supported graphics APIs at this time.

Android

The Graphics API may not be supported.

Make sure that under the PlayerSettings that the targeting GraphicsAPI is GLES3 and that your device supports GLES3.

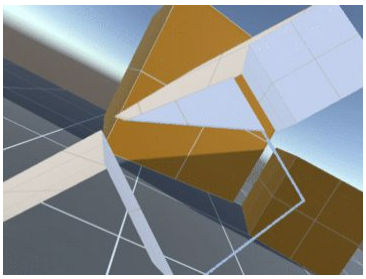
Under the PlayerSettings 'Split Application Binary' may be enabled.

There is a known issue that on some devices, that when the 'Split Application Binary' option is enabled the HvrRender the shaders may not load correctly from the resources.
Go to 'Edit/Project Settings/Player' and make sure that the option 'Split Application Binary' is not checked.

VR

VSync

If when using a Vive, the actors do not render or there is a upside down and white copy of the world visible in the headset the issue may be due to VSync being enabled.
Go to build QualitySettings 'Edit/Project Settings/Quality and make sure that all Quality Settings have the 'V Sync count' option set to 'Don't Sync'.



The Plugin did not import correctly

1. Check that the plugin was fully extracted from the '8i Unity Plugin'.
2. Make sure the Unity version is compatible with this version of the plugin.
3. Check the console to see whether there are any errors blocking Unity from compiling.

Help

Support Email

support@8i.com

Private Community Chat

8itoolbelt.slack.com (contact us for an invite, if you haven't received one) - we'll be monitoring this from our LA and NZ offices over the alpha testing period