RENDERHEADS

AVPro Live Camera

Unity plugin for high-end cameras and video capture cards

Version 2.9.1

Released 17 February 2021

Contents

1. Introduction

AVPro Live Camera is a plugin that brings high quality cameras and video capture devices to Unity.

2. System Requirements

- Unity Pro 5.6 - 2020.x
- Desktop Microsoft Windows platform (32-bit and 64-bit)
- Windows XP SP3 and higher.

2.1 Platforms not Supported

- WebGL
- WebPlayer
- Mobile, Android, iOS, Windows Phone
- Mac
- Linux
- Windows Metro / Store Apps (Note: Windows Metro apps don't support DirectShow which this plugin is built upon. This plugin only supports Windows desktop apps)

3. Installation Steps

1. Import the **unitypackage** file into your Unity project.

4. Features

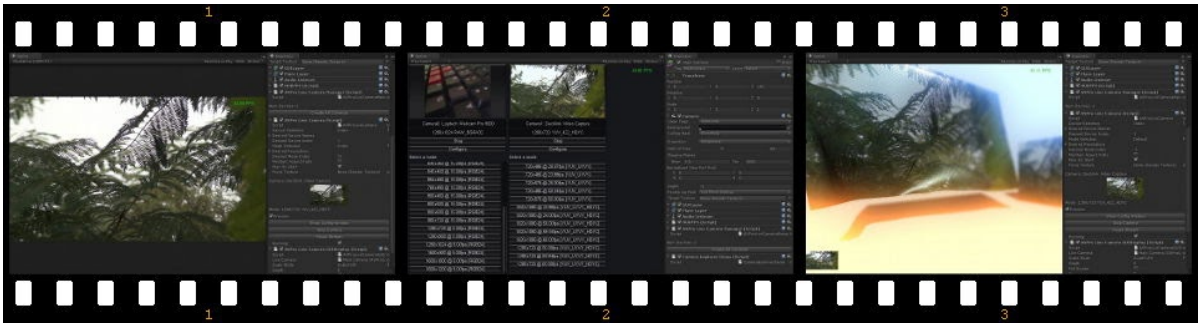
- *NEW* Unity 2020 support
- *NEW* XSplit-VCam transparency support
- High performance
- Easy to use
- Supports high-end cameras & formats
- Support for Blackmagic Design's DeckLink capture cards**
- Support for TV capture cards
- Supports Unity 5.6 and above
- Works in the editor and also in stand-alone builds
- 5 demos included
- No scripting required (for most usage)
- NGUI component included

** For more professional DeckLink support see our AVPro DeckLink plugin

Useful for:

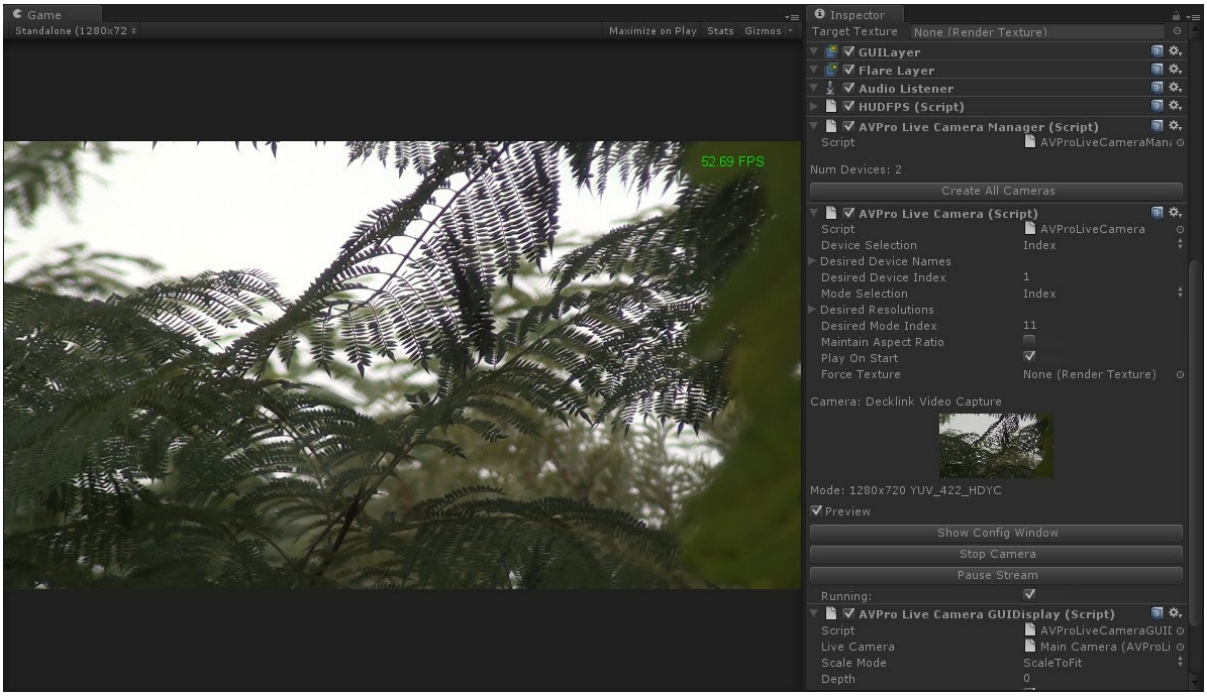
- Interactive installations
- Scientific research
- Computer vision
- Serious games
- Training and simulation
- Kiosks
- Video apps

5. Demos



The plugin includes several demos, some which require scripting and some which don't.

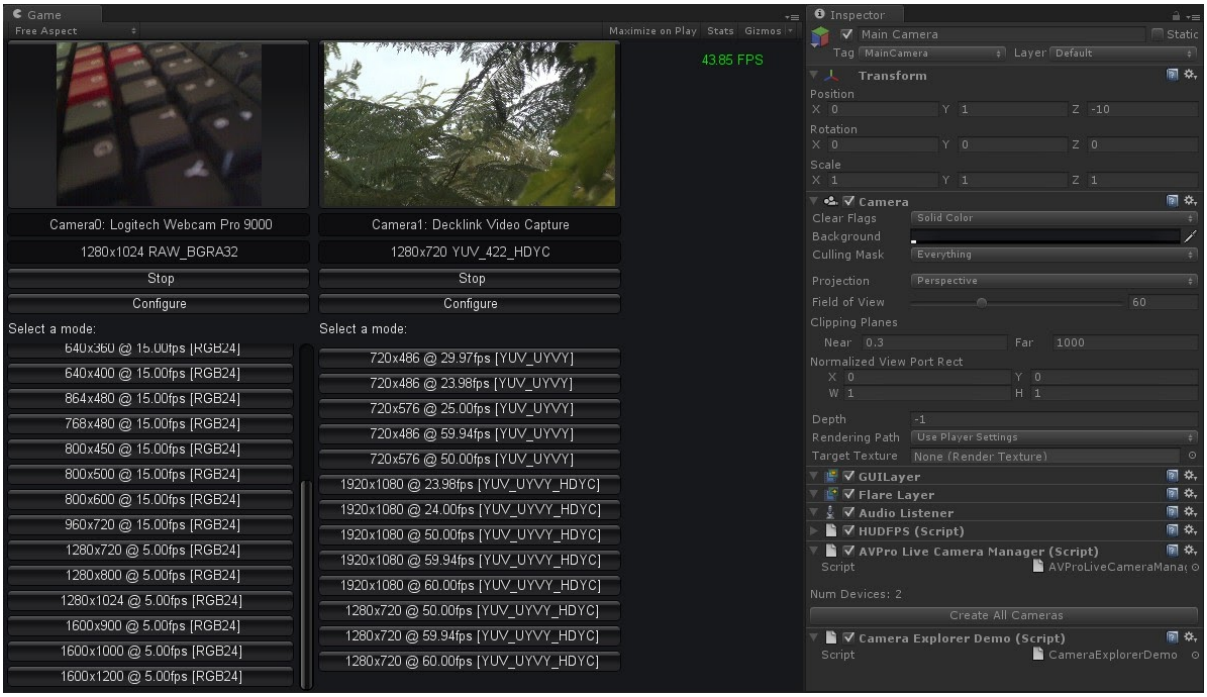
5.1 Default Camera Demo



This is the most basic demo as it requires no scripting at all. The demo simply draws the default camera to the screen. To create the demo 3 components were used:

1. **AVProLiveCameraManager** - This component is always required.
2. **AVProLiveCamera -** This component represents the default camera.
3. **AVProLiveCameraGUIDisplay -** Actually draws the camera to the screen via Unity's IMGUI system. You could use multiple of these components to render the camera image to multiple places on the screen.

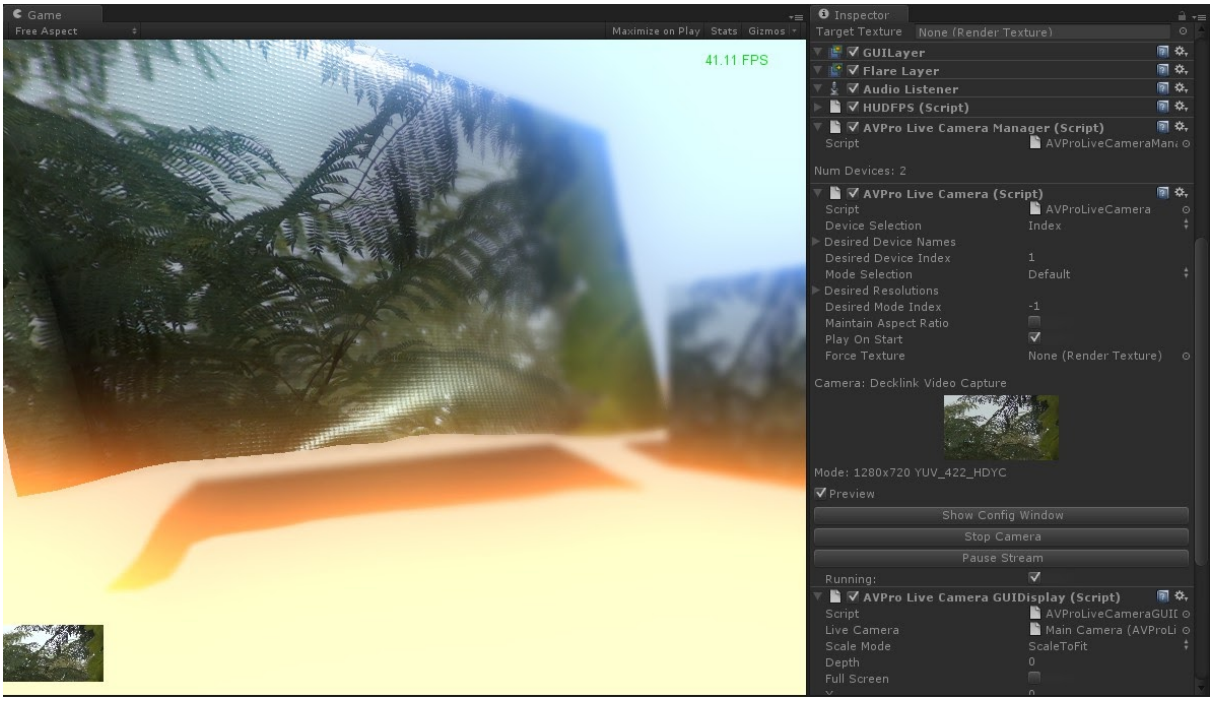
5.2 Camera Explorer Demo



This demo doesn't use any of the drag 'n drop components (except the required **AVProLiveCameraManager** component) and demonstrates how easily the a demo can be written using a few lines of scripting. The demo is created by a single 120 line script file, most of which is GUI related.

The demo allows you to explore all of the cameras connected to the system.

5.3 Material Mapping Demo



This demo shows how to integrate AVProLiveCamera into your 3D scene. No scripting is required as you can just use either of the 2 included components:

AVProLiveCameraMaterialApply - Replaces the main texture of a material with the texture from the camera.

AVProLiveCameraMeshApply - Replaces the main texture of all of the materials on a mesh with the texture from the camera.

5.4 uGUI Demo

This demo shows how to display a AVProLiveCamera texture feed into a Unity uGUI canvas. It uses a component called AVProLiveCameraUGUIComponent which can be added to any object in the canvas. There is also a menu shortcut under the GameObject > UI menu

5.5 Background Demo

This demo shows how to display a AVProLiveCamera texture feed in the background behind all other Unity content. This can be useful for augmented reality.

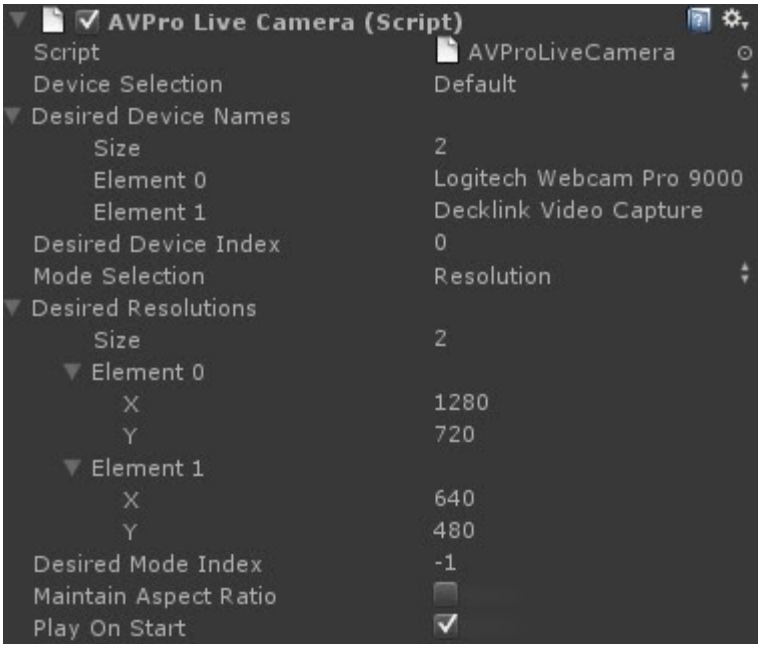
6. Components

This asset includes a number of Unity script components that allow use of the asset without any scripting.

6.1 AVProLiveCameraManager

There must always be exactly one **AVProLiveCameraManager** in your scene when you use this plugin. It is also important that this component starts before the other **AVProLiveCamera** components. There is usually nothing to configure in this component but you may want to disable 'Support Hot Swapping' which is used to gracefully handle dynamic device connection/disconnection - this isn't usually needed and adds extra overhead. .

6.2 AVProLiveCamera



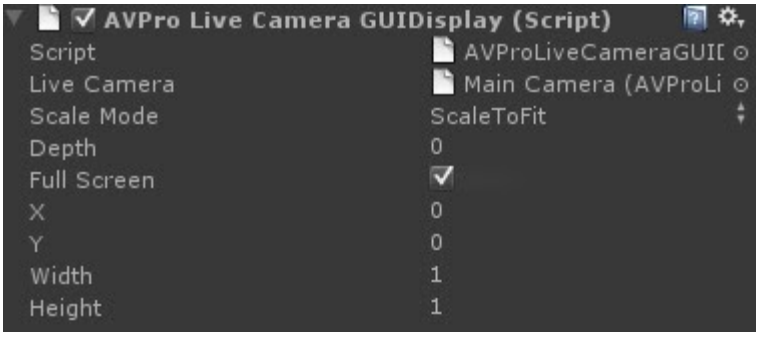
This component represents a single camera device. It has options to allow you to set which camera it uses. The default is just to load the default camera in the default mode, however you can also search cameras by name or by index on your system. The resolution of the camera can also be set either as the default, from a list of resolutions or by index. When using a list of device names or resolutions, it will try to find the closest match, prioritising those at the top of the list.

When the scene is playing this component also previews the currently running camera and allows you to control the stream which is useful for debugging:

6.3 AVProLiveCameraUGUIComponent

This component allows you to easily display an AVProLiveCamera in the new Unity uGUI system.

6.4 AVProLiveCameraGUIDisplay



This component displays an AVProLiveCamera on the screen using Unity's IMGUI system. Simply select the **AVProLiveCamera** component you want to display in the "Live Camera" option. Next you can set the placement of the item on the screen or use the fullscreen default.

6.5 AVProLiveCameraMaterialApply

Use this component to apply an **AVProLiveCamera** to a material in your scene.

6.6 AVProLiveCameraMeshApply

Use this component to apply an **AVProLiveCamera** to all of the materials used by a mesh in your scene.

6.7 AVProLiveCameraGrabber

This component demonstrates how you can grab the camera texture as a Color32 array. This can be useful for passing the data to a 3rd party library, for example for Augmented Reality (AR) or OpenCV.

7. Scripting

The scripting can easily be understood by examining the script in the **CameraExplorerDemo** scene which uses the Unity Wrapper Layer**.**

7.1 Layers

The plugin scripts are split into 3 logical layers of abstraction:

1. Low-Level
2. Unity Wrapper
3. Unity Components

7.2 Low-Level Layer

The low-level aspect of this plugin consists of a single static class AVProLiveCameraPlugin that wraps the native DLL functions, exposing it to C# scripting. Most of the plugin functionality is derived from functions in this class. Is it possible to use this class alone however we have built 2 more layers on top to make it easier to use.

7.3 Unity Wrapper Layer

This layer consists of classes with high-level functionality that are built from the low-level layer and expose functionality in a more easy to understand manner. These classes are Unity specific as they use certain Unity classes, eg Texture2D. It is possible to use these scripts in your own scripts or you can use the higher level Unity Components Layer. Classes include:

- AVProLiveCameraDevice
- AVProLiveCameraDeviceMode

- AVProLiveCameraFormatConverter
- AVProLiveCameraPixelBuffer

7.4 Unity Components Layer

These scripts are all based on Unity's MonoBehaviour class and are thus drag-and-drop type components. The scripts are built on the middle wrapper layer classes and allow use of the plugin without having to write scripts. Some of these scripts also have Editor components to customise their appearance. Classes include:

- AVProLiveCameraManager
- AVProLiveCamera
- AVProLiveCameraUGUIComponent
- AVProLiveCameraApplyUITextureNGUI
- AVProLiveCameraGUIDisplay
- AVProLiveCameraGrabber
- AVProLiveCameraMaterialApply
- AVProLiveCameraMeshApply

7.5 Script Order

Sometimes the script execution order is important and we recommend this order for our component scripts:

The most important is the Manager script which should always be one of the first in your list. Any of our own scripts that refer to the AVPro Live Camera scripts may have to have their script order explicitly set so they run after the AVPro Live Camera scripts.

8. Support

If you are reporting a bug please include any relevant files (screenshots, logs, scripts etc) so that we may remedy the problem as fast as possible.

If you have a bug to report, or a feature request, please use our GitHub Issue tracker:

- Issues: https://github.com/RenderHeads/UnityPlugin-AVProLiveCamera

For general questions and product information please see:

- Forum: http://forum.unity3d.com/threads/137233-AVPro-Live-Camera
- Website: http://renderheads.com/product/av-pro-live-camera/
- Email: unitysupport@renderheads.com

9. About RenderHeads Ltd

RenderHeads Ltd is an award winning creative and technical company that has been designing and building cutting edge technology solutions since its formation in 2006. We specialise in creating unique interactive audio-visual software for installations at auto shows, museums, shows and expos.

9.1 Services

- Unity plugin development
- Unity game / interaction / augmented reality development
- Unity consulting

9.2 Jobs at RenderHeads

We're always looking for creative technical people to join our team. RenderHeads is an innovative software company that develops interactive experiences lead by high-end technology and beautiful graphics. We have created installations for Nike, Ford, Nissan, Dell, Intel, PwC, Shell, Cisco, and others. Our work also appears in the National Maritime Museum Greenwich, Museum of Science and Industry Manchester, Sheikh Abdullah Al Salem Cultural Centre Science Museum Kuwait and Guiness Storehouse Ireland.

We create games for museums and brands, bespoke AR, VR and huge video wall experiences for shows, real-time visualisations for events and audio-visual products for developers and the broadcast industry. All of the software we develop is unique, and our clients have high expectations. We work with some of the latest hardware and software technologies, and each project gives us a new opportunity to explore the edge of possibility.

RenderHeads offer a flexible working opportunity with remote or in-office working. We have offices in Cape Town, South Africa and Glasgow, Scotland, and team members working remotely from around the world.

If you're looking for a new opportunity to push the limits and create awesomeness, please read through the requirements below and email us at jobs@renderheads.com. Send us your CV and links to showreel, past projects or code examples.

General Requirements

- You must be able to show a history of making software, either by professional experience or personal projects
- Pragmatic software development approach - we ship software frequently
- Either very strong technical skills, or a mix of creative and technical skills
- Good communication skills - most of our communication is online, via Slack/Skype/Hangouts and email/Google Docs

Positions we have available:

Video Software Developer

- You would be developing in-house camera and video related software, including our AVPro series of Unity plugins, related to cross-platform video playback, streaming, capture and encoding
- Required experience:
  - Strong C++
  - Multi-threaded programming
- Ideal experience:
  - C# and Unity
  - AVFoundation / Media Foundation / DirectShow / libAV / ffmpeg / gstreamer
  - DeckLink / NDI SDK
  - Low-level MP4 / H.264 / H.265 / HEVC / HLS / DASH / RTSP / RTP
  - Shader development
  - Camera / broadcast experience
  - 360 video workflows
  - 360 audio technologies

Interactive Software Developer

- You would be creating software for a wide variety of technically challenging and creative projects, and implementing cool interactive experiences using the latest technologies
- Features of our typical projects:
  - Most development is in Unity
  - Visualisation and effects
  - Educational games
  - VR experience development
  - Integration with cameras, sensors and hardware
  - UI and animation development
- Required experience:
  - C# and Unity
  - SVN / Git
- Ideal experience:
  - iOS / Android development
  - An interest in UI and UX

- An interest in improving workflow / tools
- Shadertoy and other graphics rendering tech
- Ability to travel (for on-site installations)

Senior Software Developer

- You would oversee the progress of the other developers in your team: making sure everyone is on track, helping to instill good development practices, helping to grow everyone's experience and skills
- You would tackle some of the more difficult programming problems and make sure that the final software is of high quality
- You would also help to scope and plan the approaches for new projects, working closely with the other senior members
- Required experience:
  - Many years of software development
  - Many projects/products released
  - Optimisation
  - Unity and C#
- Ideal experience:
  - Graphics programming
  - An interest in improving workflow / tools (Jenkins, CI, Dev ops)

9.3 Our Unity Plugins

Many of the apps and projects we develop require features that Unity doesn't yet provide, so we have created several tools and plugins to extend Unity which are now available on the Unity Asset Store. They all include a **free trial or demo version** that you can download directly from the website here:
http://renderheads.com/product-category/for-developers/

9.3.1 AVPro Video

Powerful cross-platform video playback solution for Unity, featuring support for Windows, OS X, iOS, Android and tvOS. This is our newest plugin.

9.3.2 AVPro Movie Capture

Video capture to AVI files direct from the GPU and encoded to files using DirectShow codecs. Features include 4K captures, lat-long (equirectangular) 360 degree captures, off-line rendering and more. Windows only.

9.3.3 AVPro Live Camera

Exposes high-end webcams, tv cards and video capture boards to Unity via DirectShow.

Windows only.

9.3.4 AVPro DeckLink

Integrates DeckLink capture card functionality into Unity, allowing users to send and receive high-definition uncompressed video data to and from these capture cards.

Appendix A - FAQ (Frequently Asked Questions)

A1.1 Installation

1. **How do I fix the error: “DLLNotFoundException”?**

   You need to move/copy the “Plugins” folder from your “AVProLiveCamera” folder into the root of your folder structure. This means the “Plugins” folder should be moved to your “Assets” folder.

2. **How do I fix the error: “DLLNotFoundException” where it seems to be looking for the 64-bit DLL?**

   Sometimes Unity gets confused and will try to load the 64-bit DLL in the editor (which is only 32-bit). To fix this:

   - Open Build Settings
   - Select Web Player platform and press Switch Platform
   - Select PC and Mac Standalone (with Target platform set to Windows not Windows 64-bit) and press Switch Platform

A1.2 Scripting

1. **I have compiled the scripts into a DLL and am now experiencing some unexpected behaviour.**

   Some of our scripts have Unity version-specific preprocessor defines which determine how they compile (eg UNITY_4_0). Usually when you build an external DLL these defines are missing and so the incorrect version of the code can be compiled. You need to add the appropriate compiler defines to your build.

A1.3 Performance

1. **The plugin isn’t playing back smoothly, what could the cause be?**

   The first thing to check is your hardware to make sure it’s suitable. You’ll need a decent GPU and CPU with the specifications related to the resolution of camera you’re trying to play back.

   If your video plays smoothly in the editor but is jerky when you make a build then disabling Unity’s multi-threaded rendering can help. Go to Player Settings, switch the Inspector to Debug disable and disable “MT Rendering”.

   You should also check the Asset Store to make sure you’re using the latest version of this plugin, and using the latest version of Unity.

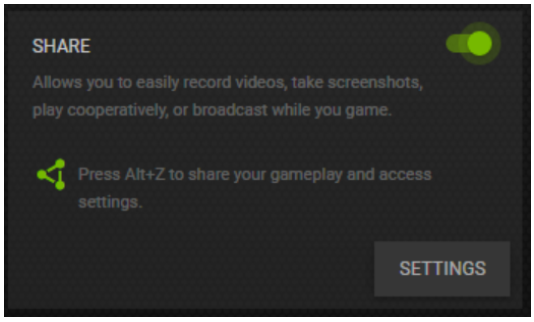
2. **The camera display frame rate is lower than expected, what could the cause be?**

You should check the settings of the camera to make sure it isn't doing unnecessary processing which reduces the frame rate. Disable options like "low light compensation", disable auto exposure and reduce exposure settings. You can access camera settings via the CameraExplorer demo (click "Configure"), or usually via the camera setup application on the Start Menu.

Another possibility is that the data bus is overloaded (especially if you're using multiple camera/capture devices and high resolutions and/or high frame-rates). If you're using USB you may need to plug devices into different ports so they are on a different USB bus. The same could apply for firewire devices.

You could also check that the checkboxes for Hot Swapping support and Settings Update are not checked in the AVProLiveCameraManager and AVProLiveCamera components. On some device drivers polling for device connection status and checking the settings can really slow things down and could be done manually instead.

3. **Why do the device enumeration / startup take a long time?**

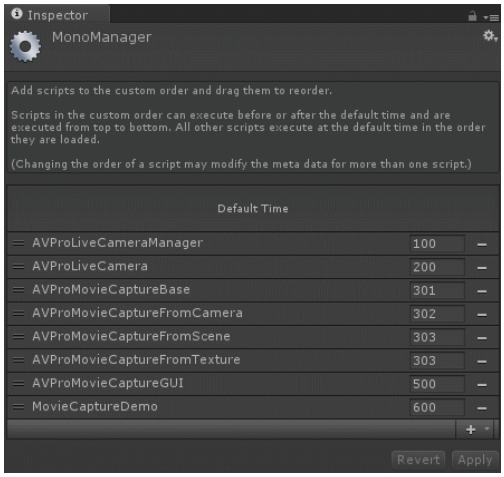


We have seen this happen when the Nvidia Geforce Experience is installed and the Share option is enabled. Disabling the share option or uninstalling this software resolves this issue.

A1.4 Other

1. **How can I fix issues when also using the AVPro Movie Capture plugin?**

   We've had reports that if you adjust the script execution order then the two plugins will work together:



2. **How can I get Axis IP Streaming Cameras to work?**

   Axis have a program called Axis Streaming Assistant which you need to install first. This will let you set up the cameras so that they can be used locally on your machine.

3. **Can I stream from IP cameras?**

   We don't natively support IP cameras, but if the manufacurer has DirectShow drivers then it should work. We know Axis cameras have software called Axis Streaming Assistant for this purpose.

   For a more general solution you could install this "DirectShow Video Source Filter for JPEG and M-JPEG IP Cameras" here: http://alax.info/blog/1216. Make sure to install either the 32-bit or 64-bit version depending on the system you're targeting. Usually 64-bit is best.

4. **How can I crop the live camera feed?**

   AVPro Live Camera doesn't yet have any native functionality to crop, but there are a few options. If you are displaying the image using the DisplayUGUI component, then you can adjust the UVRect field to crop the display. Alternatively you could take the OutputTexture property from the LiveCamera component and crop it via shader using Graphics.Blit or Graphics.DrawTexture to another RenderTexture and then display that texture.

5. **Is XSplit-VCam or any virtual webcams with transparency supported?**

   Yes we support virtual webcams devices with an alpha channel. To use this feature simply make sure that you start the device in a pixel format that has an alpha channel. This can be done in the Inspector by changing "Mode Selection" to "Resolution" and then ticking the "Transparency" option, or selecting the format "RAW_BGRA32". Also make sure that the "Allow Transparency" option is ticked on the AVProLiveCamera component. If you are applying the camera to a Mesh/Material then you need to make sure that material is using a shader that supports transparency.

6. **Why is by webcam /device rendering invisible?**

   This could be because the webcam / device is set to fully transparent in the alpha channel. If you are not expecting transparency from your device then untick the option "Allow

Transparency" on the AVProLiveCamera component.

7. **Why does OBS Virtual Camera / VirtualCam crash or not work correctly?**

   This virtual device is quite buggy in our experience. We have noted that if you just try to start the device it will not produce frames. Instead you must select one of the modes. In our testing there were only I420 and NV12 format modes. The I420 modes worked, but if you switched between modes it would crash internally with a stack overflow exception. The NV12 modes didn't produce any frames.

   There is another virtual camera plugin for OBS which appears in the menu and produces virtual devices called "OBS-Camera". We found that this virtual device worked much better.

8. **What devices are supported?**

   All devices that are compatible with DirectShow should be supported. Some devices however use exotic pixel formats which require custom support. Most consumer webcams are supported. Video capture cards, tv cards and frame grabbers are generally supported though we recommend downloading the demo and testing it with your hardware. The following is a list of hardware that we or others have successfully tested:

   - Webcams
     i. Logitech Brio
     ii. Logitech Pro 9000
     iii. Logitech HD C920
     iv. Microsoft LifeCam Cinema
   - High-end Cameras
     i. Lumenera
     ii. PointGrey Flea 3 USB
     iii. Optitrak v120 slim
     iv. Basler Ace Camera (using the GenICam standard)
     v. Axis Streaming IP Cameras (using the Axis Streaming Assistant)
   - Virtual Webcams
     i. XSplit-VCam
   - TV Cards
     i. ?
   - Video Capture / Frame Grabbers
     i. Blackmagic Intensity Pro
     ii. Blackmagic Intensity Shuttle
     iii. Blackmagic DeckLink 8K Pro
     iv. Blackmagic DeckLink SDI 4K
     v. Blackmagic DeckLink Duo
     vi. Blackmagic DeckLink Quad
     vii. Blackmagic Decklink Mini Recorder
     viii. Elgato HD60S
     ix. Epiphan DVI2USB3.0
     x. Datapath Vision capture cards
     xi. Avermedia HD Capture Cards - Requires installing the Stream Engine from Avermedia website which contain the DirectShow drivers. Please note that there is no 64-bit support for these drivers so only 32-bit Unity Editor can be used and 32-bit Windows app can be built.

xii. Deltacast cards - We have heard people had success after installing the DirectShow drivers which are available here: http://www.deltacast.tv/support/download-center

If you have tested the plugin on different hardware do let us know your results.

Appendix B - Version History

- **Version X**
  - ← Your suggestion here.
  - Add vsync locked updating?
  - Add dropped frame counter?
  - Add support for mip-map generation?
  - Support planar pixel formats?
  - Improve code documentation?
  - Audio support for tv-capture cards?
  - Add http://alax.info/blog/1216
  - Why can hot swapping enabled make FPS drop after 10-20minutes?
  - Fallback to WebcamTexture for other platforms
  - Make it more visible which mode is selected via logging
  - #ifdef for UNITY_WEBPLAYER?

- **Version 2.9.1 - 17 February 2021**
  - Bug fixes
    - Fixed crash issue when selecting I420 pixel format modes due to buggy Logitech color conversion filter. This issue presented when using the "OBS Virtual Camera" filter which was pretty buggy itself. See FAQ for notes about OBS.

- **Version 2.9.0 - 20 August 2020**
  - Improvements
    - Added support to set the YCbCr range mode to either Limited (default) or Full to get accurate representation from more devices
    - Added support for selecting which DirectShow pin type (preview or capture) is preferred. Capture pin is now the default whereas it was Preview previously. This was done to be more correct and to fix frame rate and mode selection issues on some capture cards (Elgato, Magwell)
    - Inspector UI improved to be more intuitive and offer greater control
    - Shaders refactored
  - Changes
    - Unity 5.6 is now the minimum version supported
  - Bug fixes
    - Fixed issue where the clock mode wasn't passed down to the plugin
    - Fixed issue linear colour space issue where colours would become crushed and lose precision especially at the dark end

- **Version 2.8.4 - 29 July 2020**
  - Improvements
    - All devices now run without a clock,so they run as fast as possible. A new ClockMode option was added to support this, with ClockMode.None being the default, and ClockMode.Default using the default system clock. This fixes low frame rate issues with ZCam and Elgato hardware.
  - Bug fixes
    - Fixed issue where the specified frame rate wouldn't be selected if it was only available in a specific pixel format (eg MJPG)

- **Version 2.8.2 - 15 June 2020**
  - Improvements
    - More frame rate options exposed
    - ApplyMesh and ApplyMaterial better API and behaviour, with texture property selector and editor hints for HDRP usage
  - Bug fixes
    - Fixed a bug where switching webcams would sometime cause the UGUI component to stop showing the new texture frames, even though new frames were coming through
    - Fixed gamma for ZCam device as it was using a shader without gamma correction causing the image to be too bright
    - Fixed issue where disabling the gameobject with LiveCamera component would prevent component from working even when re-enabled

- **Version 2.8.0 - 21 May 2019**
  - Improvements
    - Added support for XSplit-VCam virtual webcam
    - Added support for virtual webcams that support transparency
    - Improved resolution selection UI

- **Version 2.7.5 - 18 October 2018**
  - Bug fixes
    - Fixed the device GUID string

- **Version 2.7.4 - 6 July 2018**
  - Improvements
    - Mode searching improved by adding option to set desired frame-rate and pixel format
    - MeshApply and MaterialApply components no longer generate garbage per-frame
    - QuickDeviceMenu now has an option to toggle visibility to stop per-frame garbage generation
  - Bug fixes
    - Fixed a bug in QuickDeviceMenu that would cause the wrong device to be highlighted

- **Version 2.7.2 - 28 June 2018**
  - Improvements
    - Improved performance from new compiler optimisations
    - UGUIDisplay component no longer generates garbage
  - Bug fixes
    - Fixed issue where certain rogue drivers could cause a crash
    - Fixed a linear colour space issue since Unity 2018

- **Version 2.7.0 - 27 February 2018**
  - Improvements
    - Improved performance so that high-frame rate cameras (60hz and above) are supported
    - Refactored some demos and code and wrapped all scripts in a namespace
    - Tested and validated support for Unity 2018.1.0b8
  - Bug fixes

        - Fixed a crash bug on Avermedia capture cards, eg Live Gamer HD 2, when using their Recentral 2 software
    - Upgrade Notes
        - You should add the namespace declaration:
          “using RenderHeads.Media.AVProLiveCamera;” if you are referencing one of the AVPro Live Camera scripts, or:
          “using RenderHeads.Media.AVProLiveCamera.Demos;” for referencing one of the demo scripts

- **Version 2.6.2 - 3 July 2017**
    - Fixed an issue where using Flip X would cause image quality issues with streams using YUV formats due to different pixel ordering

- **Version 2.6.1 - 12 June 2017**
    - Added support for unicode device names
    - Improved mode selection by index to make it less confusing
    - Added support for Unity 2017

- **Version 2.6 - 7 April 2017**
    - Added support for Logitech Brio webcam
    - Added support for high-frame rate cameras
    - Added support in automatic mode selection to pick highest frame rate
    - Fixed demo IMGUI textures for linear colour space

- **Version 2.5 - 15 February 2017**
    - OpenGL Core support fixed
    - Improved enumeration speed of camera settings

- **Version 2.4 - 10 January 2017**
    - Fixed IMGUI texture display when using Linear color-space
    - Improved performance by inlining some shader functions
    - Added FAQ about Axis cameras

- **Version 2.3 - 1 July 2016**
    - Unity 5.4 supported
    - Improved Linear colour space support
    - Fixed some YUV conversion producing zero alpha channel
    - Added support for Vista and above DMO colour converter to support more YUV formats
    - Fixed editor UI so that it remembers settings
    - Can now edit multiple Live Camera components at once
    - Improved demo compatibility by removing cloth and Standard Assets
    - Minor UI improvements

- **Version 2.2 - 19 February 2016**
    - Added support for Unity 5.3 and 5.4 beta
    - Dropped support for Unity 4.5 and below. 4.6 is the new minimum
    - Fixed Unity 5.2+ bugs in uGUI components
    - Fixed Unity 5.3.0+ bug in Graphics.Blit()
    - Improved uGUI component
    - Added new uGUI demo

    - Added new background demo
    - Updated documentation for hot-swapping performance notes
    - Made hot-swapping support disabled by default
    - Made ‘internal format conversion’ enabled by default

- **Version 2.12 - 15 June 2015**
    - Fixed Unity 5.1 support

- **Version 2.1 - 15 May 2015**
    - Fixed Blackmagic Decklink 4k bug (again)
    - Added new Stop() method as Pause() causes Decklink hardware to buffer frames
    - Improved support for devices in general
    - Camera Explorer demo now have settings updates disabled by default but can toggle them on manually

- **Version 2.0 - 4 February 2015**
    - Fixed Blackmagic Decklink 4k bug
    - Supports Unity 4.6 and Unity 5.0 (beta)
    - Added Unity 4.6 uGUI component
    - Added NGUI component
    - Improved documentation
    - Dropped support for Unity 3.x, now requires Unity 4.1 minimum
    - Simplified code after dropping legacy Unity 3.x support

- **Version 1.94 - 20 May 2014**
    - Fixed bug where cameras left for hours would freeze
    - Improved format conversion performance
    - Improved texture memory usage
    - 16 byte memory alignment

- **Version 1.92 - 1 May 2014**
    - Fixed textures not being released in editor
    - Added names to textures

- **Version 1.9 - 11 March 2014**
    - Exposed camera settings (brightness, exposure, focus etc)
    - Exposed horizontal and vertical image flip option
    - Added script to automatically install (copy) the plugin DLLs
    - Improved best video mode selection

- **Version 1.8 - 27 August 2013**
    - Added Lumenera camera support
    - Added support for MJPEG camera mode
    - Added support for RGB24 camera mode
    - Added support for 8-bit mono camera mode
    - Added support for colour convertable camera modes
    - 64-bit builds no longer require DLL copying
    - Fixed yuy2 bug in frame grabber

- **Version 1.7 - 28 May 2013**
    - Improved Camera Explorer demo

    - Fixed bug affecting IDS-Imaging cameras
    - Optimised frame conversion
    - High-speed cameras (>60fps) now supported.
    - Improved internal frame buffering
    - Added grabbing from internal frame buffer
    - Added counter for captured and displayed FPS
    - Added documentation about layers of abstraction

- **Version 1.62 - 29 April 2013**
    - Fixed DemoInfo script error
    - Fixed lost device when dragging/resizing window
    - Fixed 64-bit crash bug when dragging/resizing window

- **Version 1.6 - 15 April 2013**
    - Added deinterlacing
    - Added Unity 4.1 non-pow2 texture support - optimisation
    - Added support for hot-swapping camera devices
    - Added mirror display to GUI Display component
    - Added Script Order to documentation

- **Version 1.52 - 18 March 2013**
    - Added Unity 4.1 support
    - Fixed some platform #if issues

- **Version 1.5 - 4 March 2013**
    - Added Color32 grabber component
    - Optimised pixel format conversion
    - Fixed conflict with other AVPro plugins by using unique GL.IssuePluginEvent()
    - Fixed some shaders display half the texture

- **Version 1.4 - 15 December 2012**
    - Unity 4.0 support added.
    - Unity 4.0 native DirectX texture updates supported, boosting performance.
    - Various minor improvements

- **Version 1.2 - 10 September 2012**
    - Added support for devices with crossbars including TV-cards and capture cards with multiple inputs (eg AVerMedia).
    - Faster device enumeration (startup).
    - Added colour to AVProLiveCameraGUIDisplay component
    - Inspector UI improvements.
    - Fixed a texture size bug in OpenGL mode.

- **Version 1.12 - 12 July 2012**
    - Fixed minor null reference bug.
    - Minor tweaks to file names to make them unique.

- **Version 1.1 - 4 July 2012**
    - Added Windows 64-bit support.
    - Renamed files to prevent future name collisions.

- **Version 1.0 - 20 June 2012**
    - Initial release submitted to Asset Store.