RENDERHEADS

AVPro Live Camera

Unity plugin for high-end cameras and video capture cards

Version 2.2

Released 19 February 2016

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 4.6 - 5.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.

2. Move the DLL plugin files to the appropriate folder. This should happen automatically via the installation script when you first to to run/play in the editor. The manual steps for this are:
    a. In Unity version 4.x copy the 32-bit DLL to the **Assets/Plugins/x86** folder and the 64-bit plugins to the **Assets/Plugins/x86_64** folder.
    b. In Unity version 5.x the above can be done, or the plugin inspector can be used

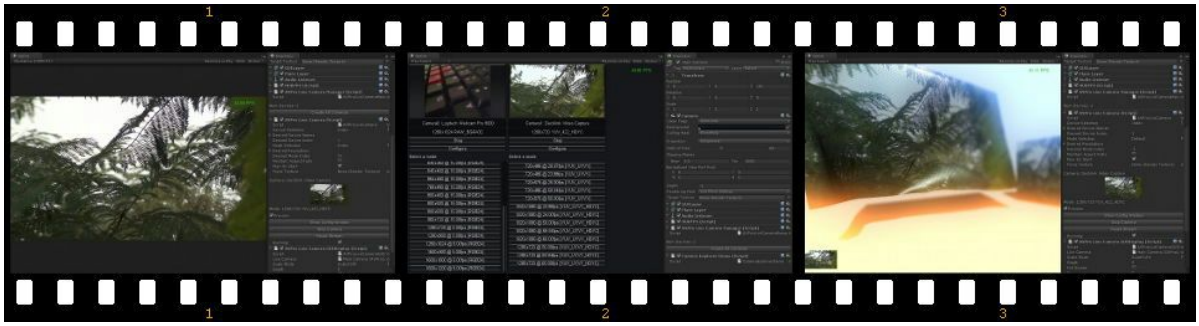
4. Features

- *NEW* Unity 5.3 support.
- High performance.
- Easy to use.
- Supports high-end cameras & formats.
- Support for Blackmagic Design's DeckLink capture cards.
- Support for TV capture cards (eg AVerMedia)
- Supports Unity 4.6 and above.
- Works in the editor and also in stand-alone builds.
- 5 demos included.
- No scripting required (for most usage).
- NGUI component included

Useful for:

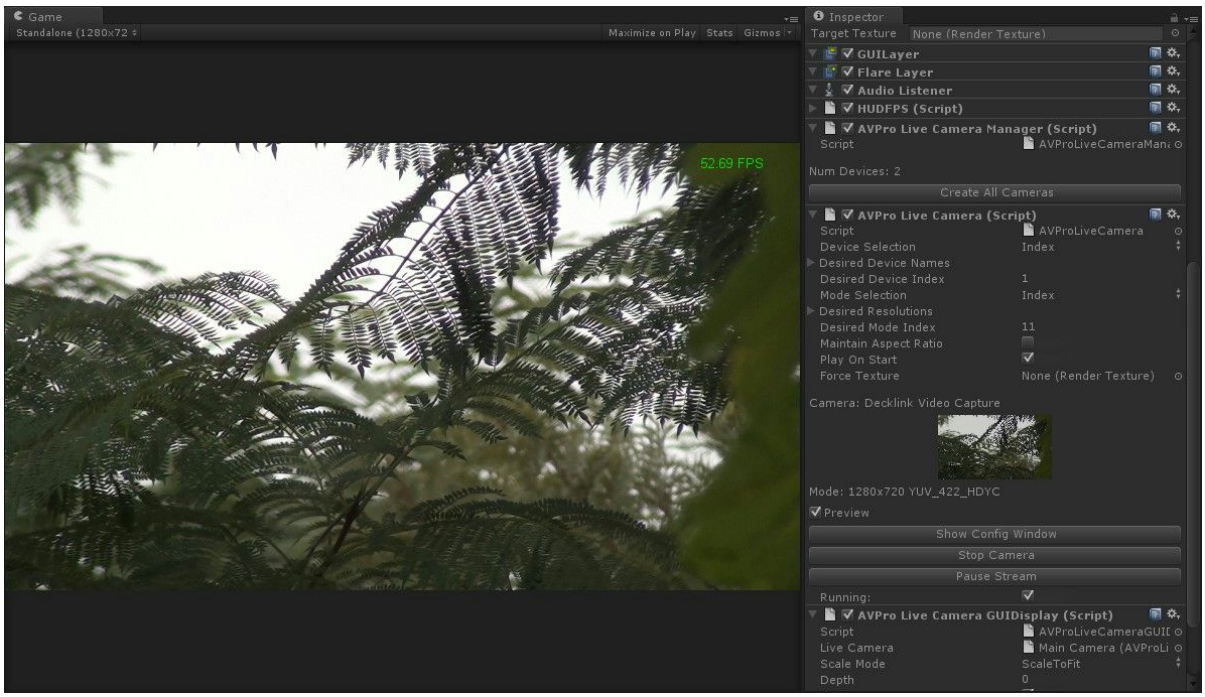
- Interactive installations
- Scientific research
- Computer vision
- Serious games
- Training and simulation
- Kiosks
- Video apps

5. Demos



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

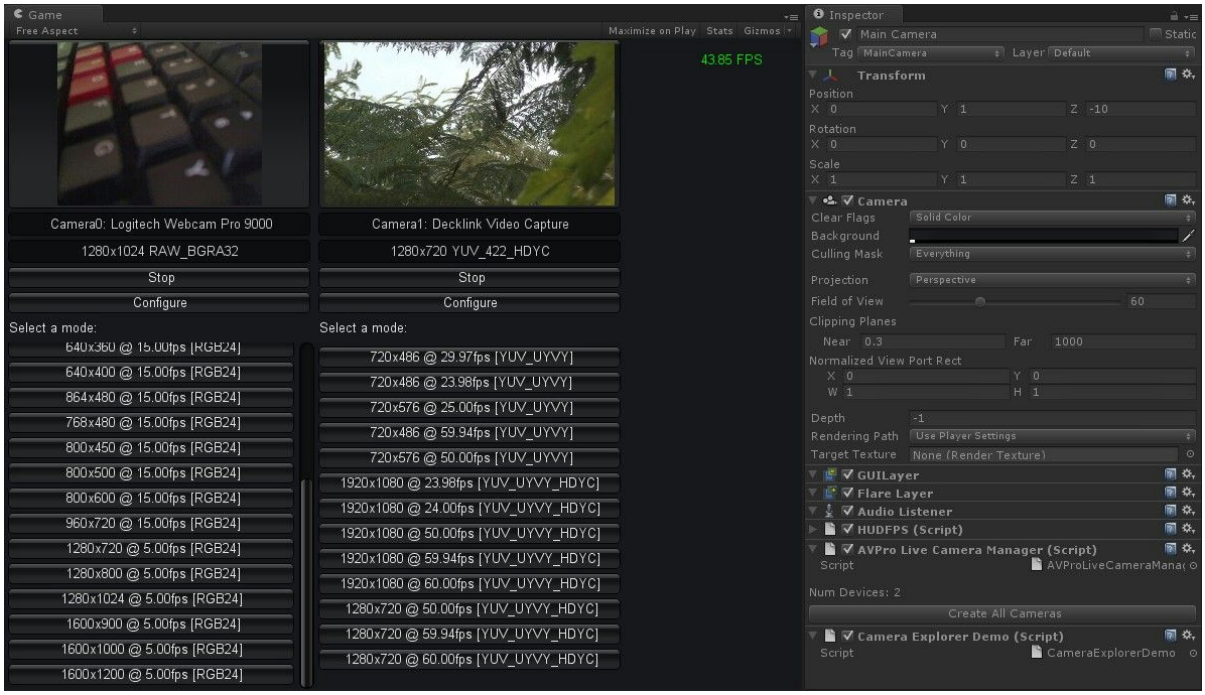
5.1 Default Camera Demo



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.

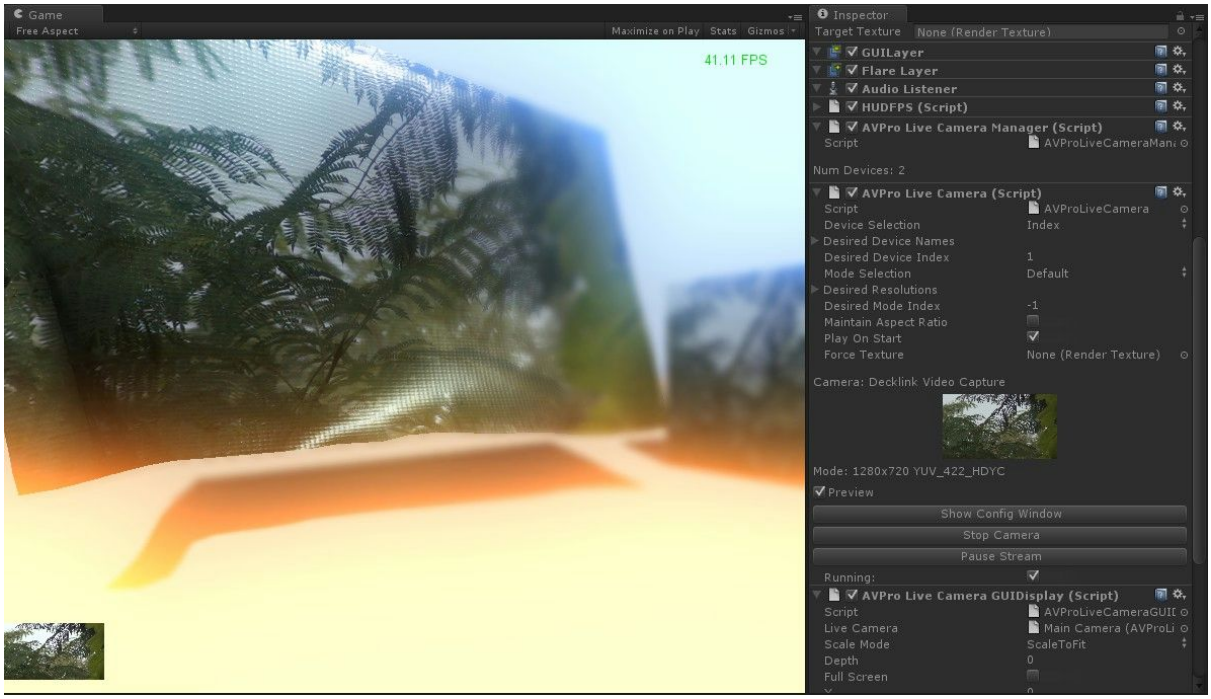
5.2 Camera Explorer Demo



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.

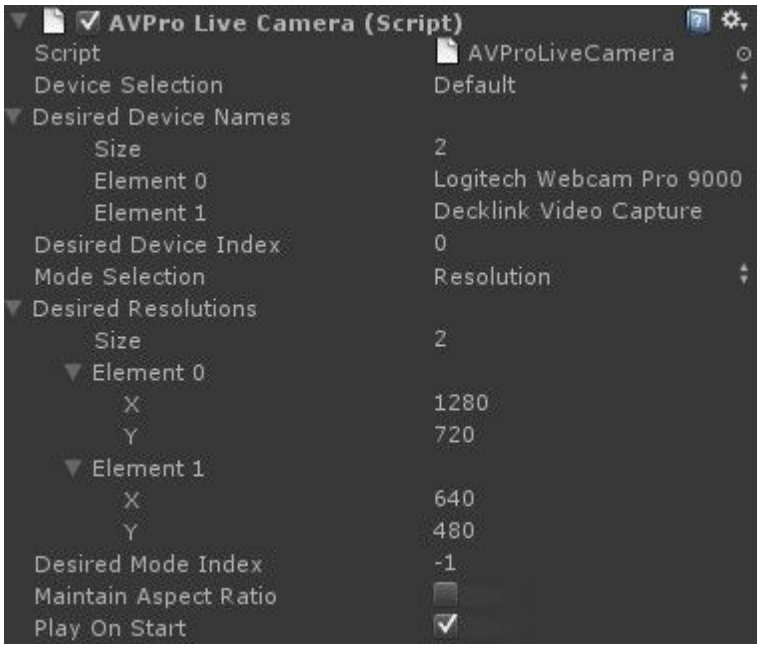
6. Components

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



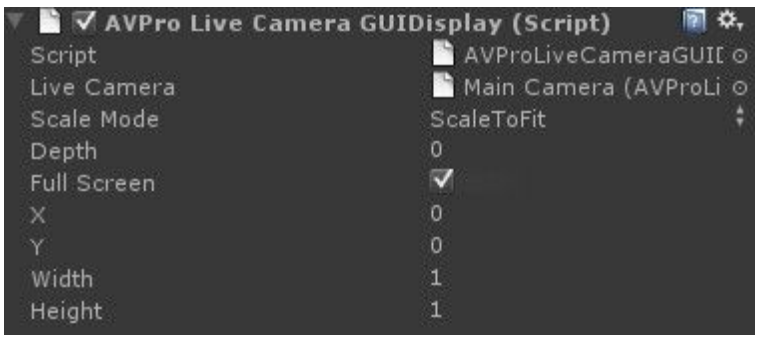
This component represents a single camera device. It has options to allow you to set which camera it uses. The default is just to load the default camera in the default mode, however you can also search cameras by name or by index on your system. The resolution of the camera can also be set either as 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).

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 in need of support or have any comments/suggestions regarding this product please contact us.

Website: http://www.renderheads.com/contact/
Forum: http://forum.unity3d.com/threads/137233-AVPro-Live-Camera
Email: unityplugins@renderheads.com

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.

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 Our Unity Plugins

[AVPro QuickTime]

[AVPro Windows Media]

[AVPro Movie Capture]

[AVPro Live Camera]

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.

A1.4 Other

1. **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 Pro 9000
     ii. Logitech HD C920
     iii. Microsoft LifeCam Cinema
   - High-end Cameras
     i. Lumenera
     ii. PointGrey Flea 3 USB
     iii. Optitrak v120 slim
   - TV Cards
     i. ?
   - Video Capture / Frame Grabbers
     i. Blackmagic Intensity Pro
     ii. Blackmagic Intensity Shuttle
     iii. Blackmagic DeckLink SDI 4K
     iv. Blackmagic DeckLink Duo
     v. Blackmagic DeckLink Quad
     vi. Blackmagic Decklink Mini Recorder
     vii. Epiphan DVI2USB3.0
     viii. Avermedia Live Gamer HD Capture Card

   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?
  - Faster format conversion path for native formats?
  - Support planar pixel formats?
  - Improve code documentation?
  - Audio support for tv-capture cards?
  - Add http://alax.info/blog/1216
  - Add support for CLSID_CColorConvertDMO?
  - Why does hot swapping enabled make FPS drop after 10-20minutes?

- **Version 2.2 - Friday 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 - Monday 15 June 2015**
  - Fixed Unity 5.1 support

- **Version 2.1 - Friday 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 - Wednesday 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 - Tuesday 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 - Thursday 1 May 2014**
    - Fixed textures not being released in editor
    - Added names to textures

- **Version 1.9 - Tuesday 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 - Tuesday 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 - Tuesday 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 - Monday 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 - Monday 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 - Monday 18 March 2013**
    - Added Unity 4.1 support
    - Fixed some platform #if issues

- **Version 1.5 - Monday 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 - Saturday 15 December 2012**
    - Unity 4.0 support added.
    - Unity 4.0 native DirectX texture updates supported, boosting performance.
    - Various minor improvements

- **Version 1.2 - Monday 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 - Thursday 12 July 2012**
    - Fixed minor null reference bug.
    - Minor tweaks to file names to make them unique.

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

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