AVPro Movie Capture
Unity Plugin

Version 2.02 - Released 18 March 2013

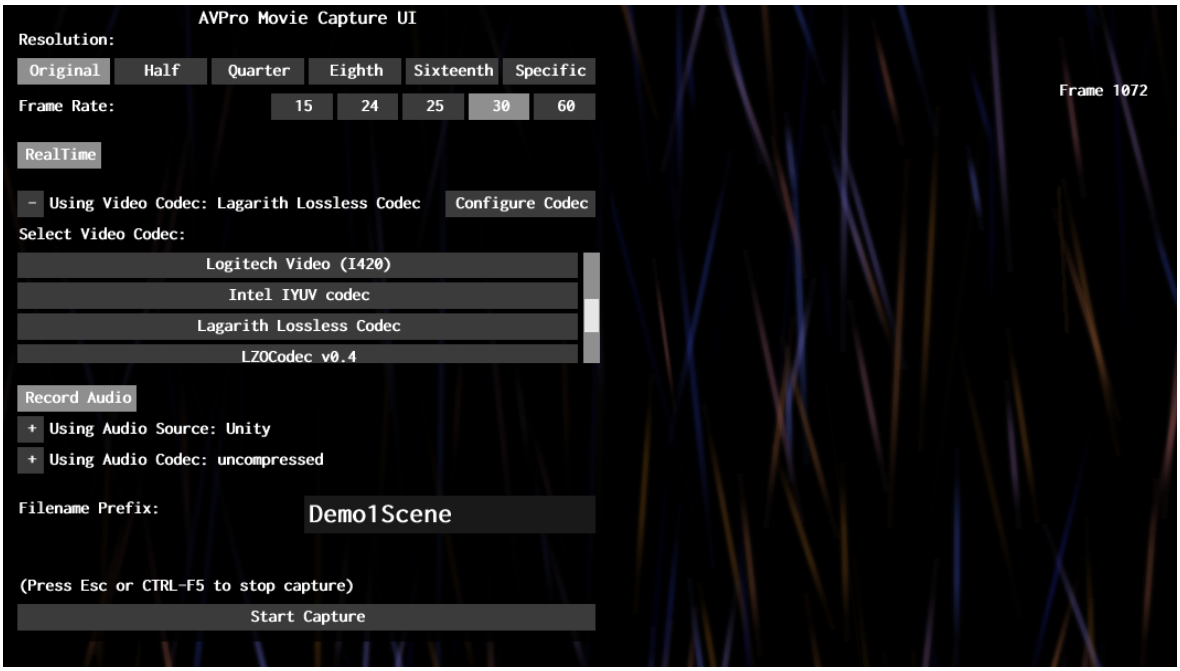
Real-time capture of Unity Camera to AVI file.

RENDERHEADS
DIGITALLY PERFORMED

Contents

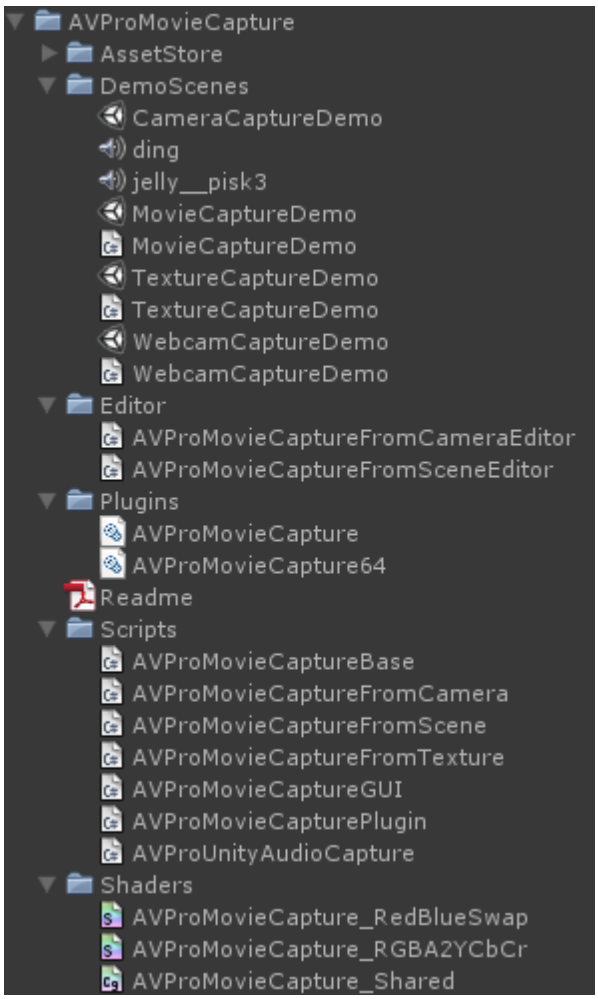
1. Introduction

“AVPro Movie Capture” is a plugin for Unity that allows recording video directly to disk as an AVI file.



Whether we’re just testing out an idea for a new effect, playing around with some parameters or producing demos for our clients, we often find it useful to be able to quickly and easily capture a video from within Unity. Previously we used screenshots and captured videos using tools like Fraps, however we wanted something completely integrated into Unity and so AVPro Movie Capture was born.

The asset package consists of the following elements:



- DemoScenes
  - **CameraCaptureDemo.unity** - A simple demo showing how to use the AVProMovieCaptureFromCamera component.
  - **MovieCaptureDemo.unity** - A simple demo scene showing how to use the AVProMovieCaptureFromScene component.
  - **TextureCaptureDemo.unity** - A simple demo scene showing how to use the AVProMovieCaptureFromTexture component.
  - **WebcamCaptureDemo.unity** - A simple demo scene showing how to use the AVProMovieCaptureFromTexture component.
- Plugins
  - **AVProMovieCapture.dll** - The main plugin DLL that talks to DirectShow.
  - **AVProMovieCapture64.dll** - The main 64-bit plugin DLL that talks to DirectShow.
- Scripts
  - **AVProMovieCapturePlugin.cs** - Wrapper interface to access capture functions in the DLL.
  - **AVProMovieCaptureBase.cs** - Base class
  - **AVProMovieCaptureGUI.cs** - Helper component that displays a GUI exposing the capture options of AVProMovieCaptureBase

    - **AVProMovieCaptureFromCamera.cs** - Drag ‘n drop component to allow easy capturing from a camera but not GUI.
    - **AVProMovieCaptureFromScene.cs** - Drag ‘n drop component to allow easy capturing of the entire scene including GUI.
    - **AVProMovieCaptureFromTexture.cs** - Drag ‘n drop component to allow easy capturing of a dynamic texture.
    - **AVProUnityAudioCapture.cs** - Drag ‘n drop component for capturing Unity audio into a buffer for saving to the AVI file.
- Shaders
    - **AVProMovieCapture_RedBlueSwap.shader** - Internal shader used to swap red and blue channels.
    - **AVProMovieCapture_RGBA2YCbCr.shader** - Internal shader used to convert RGBA to YCbCr YUY2 format.

2. Installation

System Requirements:

- Unity Pro 3.5 and above.
- The plugin only supports Microsoft Windows (32 and 64-bit builds)
- Windows XP SP3 and higher.
- Codecs for any video formats you want to record to.

Installation Steps:

1. Import the unitypackage file into your Unity project.
2. Move the “Plugins” folder into the root of your project.

Note: Ensure you have the relevant video codecs installed for formats you want to record to.

3. Features

- High performance.
- Easy to use.
- Use any video codec you want.
- Can capture alpha channel for creating transparent videos.
- Records audio directly from Unity or from a Windows recording device.
- Unicode file name support.
- Works in the editor and also in stand-alone builds.

Useful for:

- Games - recording gameplay.
- Development - easily record videos for clients or to share online.
- Testing - create videos of bugs to aid debugging.
- Interactive Installations - making videos of each user session.

4. Components

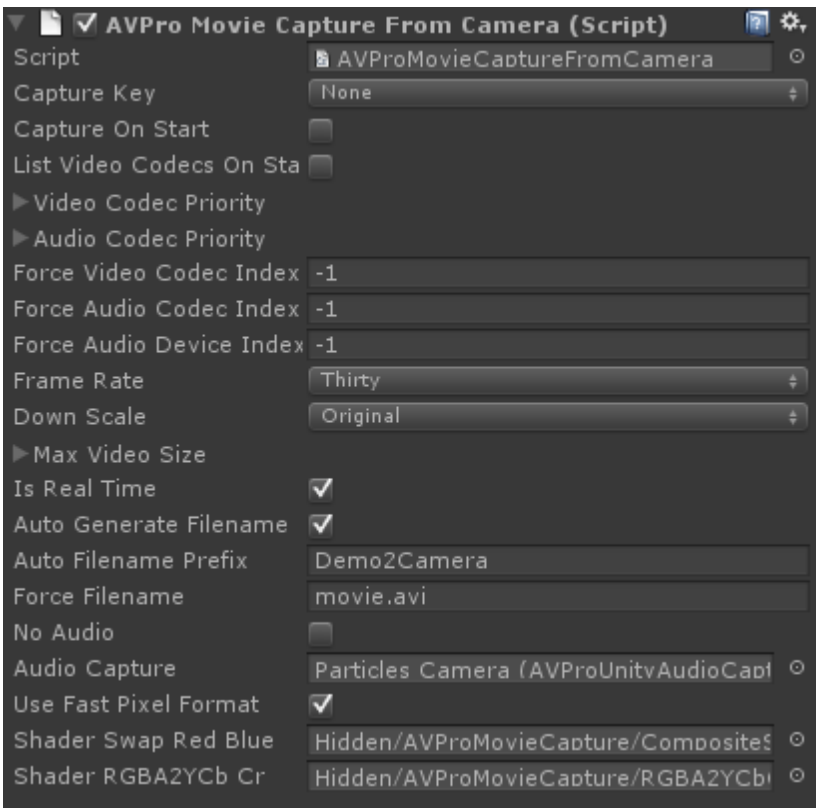
There are 2 main components that come with this plugin.

AVProMovieCaptureFromCamera & AVProMovieCaptureFromScene

In Unity 3.5 and above AVProMovieCaptureFromScene is faster than AVProMovieCaptureFromCamera as it accesses the graphics API directly. For older versions of Unity, AVProMovieCaptureFromCamera has the best performance, however it cannot capture the GUI.

AVProMovieCaptureFromCamera Component

This component is attached to a camera and captures the 3D output from that capture. It cannot capture GUI (for this use the AVProMovieCaptureFromScene component). Simply drag the “AVProMovieCaptureFromCamera ” script to the camera you want to capture or select it from the “AVPro Movie Capture” components menu. Make sure “AVProMovieCaptureFromCamera” component is the last component on your camera.



The component can be set to start/stop recording when a specific key is pressed, or it can be set to start recording when the application starts.

The option “Use Fast Pixel Format” translates internally to using 1 of 2 pixel formats:

RGBA32 (standard) or YCrCb422_YUY2 (fast).

Using YCrCb422_YUY2 is usually faster to encode as it is half the size of RGBA32 and also usually the native format for most encoders. RGBA32 should be used when compatibility with a codec is needed and when lossless encoding is required.

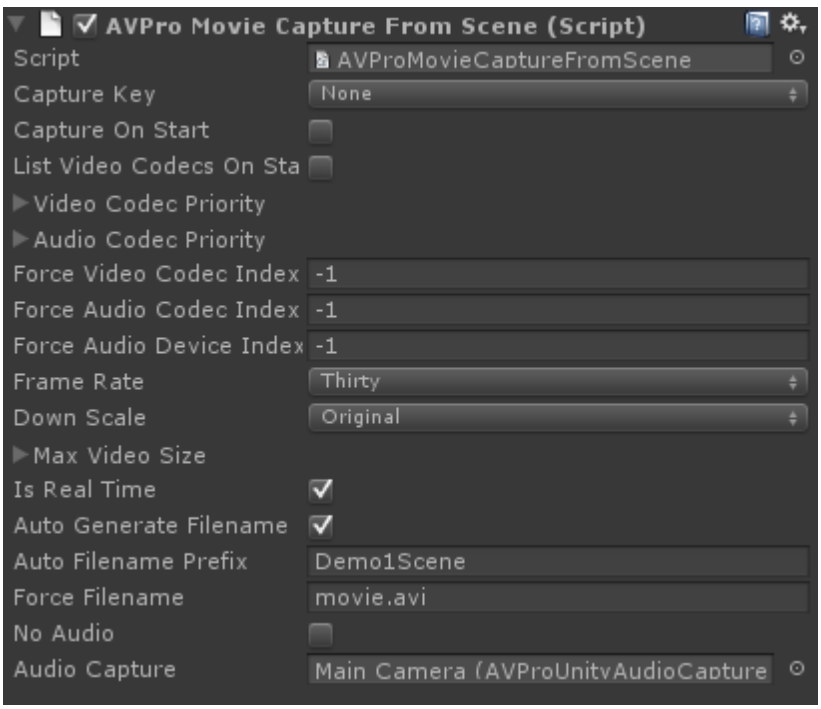
The option “List video codecs on start” will print the list of available video codecs installed on the machine to the console. This is useful to be able to know which codecs you can use.

“Video codec priority” is an array of strings the user is free to edit. Each string is the name of a codec. When the component runs it will try to select the codec from the the list that it finds first on the system.

“Force video codec index” will override the “Video codec priority” list and allows the user to easily select a codec from the list of system codecs. This index is not an index into the above list of codec names but into the list returned in the console. This value is default to “-1” which means ignore and use the priority codec list.

AVProMovieCaptureFromScene Component

This is the preferred component for capturing as it grabs directly from the GPU buffers bypassing Unity (requires Unity 3.5) and is thus faster. Unlike AVProMovieCaptureFromCamera this component captures the entire scene including the GUI. It must be added to a camera object in the scene.



The options are the same as AVProMovieCaptureFromCamera .

5. Tips

For best results we recommend:

1. Install and use the Lagarith video codec. It is a free lossless codec with excellent performance. It does produce quite large files though so you may need to convert it to another format before sharing/uploading. You need to configure the Lagarith codec and enable multi-threading and null frames.

2. If you need to convert videos from one codec to another use VirtualDub or FFMPEG from the command-line.

3. Install and use the X264 VFW video codec. It is fairly fast and produces videos of a very small file size. If you're recording at a high resolution though you'll need a very fast CPU for this codec.

4. If a specified codec couldn't be found, a warning is generated and uncompressed video will be produced.

5. For best performance create a build and do your captures from the build running in full-screen mode.

6. Custom Usage

If you want to go beyond the component you can access the functionality of the AVProMovieCapture DLL directly or edit AVProMovieCaptureBase.cs to make your changes.

The DLL has the following functions which are wrapped in AVProMovieCapturePlugin.cs:

bool Init();
Global initialisation for the plugin. Returns false if unsuccessful.

void Deinit();
Global deinitialisation for the plugin.

int GetNumAVIVideoCodecs();
Returns the number of video codecs on the system.

bool GetAVIVideoCodecName(int index, StringBuilder name);
Returns true if successful. The name of the system codec at index is returned as a

StringBuilder. StringBuilder should be created with size 512.

int GetNumAVIAudioInputDevices();
Returns the number of audio input devices on the system.

bool GetAVIAudioInputDeviceName(int index, StringBuilder name);
Returns true if successful. The name of the system audio inptu device at index is returned as a StringBuilder. StringBuilder shoudl be created with size 512.

int CreateRecorderAVI(string filename, uint width, uint height, int format, bool isTopDown, int videoCodecIndex, int audioInputDeviceIndex);
Creates a recorder instance to generating AVI files. An integer is returned which is a unique value specific to this instance of the recorder.

void Start(int handle);
Starts recording. Handle is the handle of the recorder instance.

void Pause(int handle);
Pauses recording.

bool IsNewFrameDue(int handle);
Let's us know whether the encoder is ready for another frame.

bool EncodeFrame(int handle, System.IntPtr data);
Sends frame to be encoded. "data" points to an array of widthxheight and with a bitdepth of 32 for RGBA32 videos and 16 for YUY2 videos.

void Stop(int handle);
Stops recording. Handle is the handle of the recorder instance.

void FreeRecorder(int handle);
Releases the instance of the recorder.

7. FAQ (Frequently Asked Questions)

1. **Does this plugin record the audio from Unity?**

   Yes, if you use the AVProUnityAudioCapture component you can record audio directly from Unity.

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

   You need to move/copy the “Plugins” folder from your “AVProMovieCapture” folder into the root of your folder structure. This means the “Plugins” folder should be moved to your “Assets” folder. Unfortunately this is a limitation in the way Unitys Asset Store handles plugins.

3. **Where are my movie captures stored?**

   By default the components auto-generate a filename each time you run a capture. These files are stored in the root of your project (the folder above “Assets”). You can always disable auto-generation of filenames in the component and specify your own file name and location for a capture.

4. **How do I record in-game audio and microphone audio at the same time?**

   Currently this plugin only supports recording audio from a single Windows audio device. There is a “trick” you can use though. In Windows 7 (and perhaps Vista) you may be able to set your microphone to play through the speakers by going to:

   Control Panel -> Sound -> Recording -> Select your microphone -> right click -> Properties -> Listen -> check "Listen to this device".

   You should then hear your microphone recording through your speakers. It’s recommended to use headphones during recording to prevent feedback from the speakers into the microphone.

5. **How do I get this plugin working with 64-bit Windows builds?**

   Currently in Unity there is no way to automatically support both 32-bit and 64-bit plugins at build time so you need to rename the plugin file manually once you’ve made a build. In your build “plugins” folder simply delete “AVProMovieCapture.dll” and rename “AVProMovieCapture64.dll” to “AvProMovieCapture.dll”.

6. **How do I prevent Unity from freezing after doing a recording using the Xvid MPEG4 codec?**

   Open the configuration for the Xvid MPEG4 codec and go to the “other options” page and make sure “Display encoding status” is not selected.

7. **I’m using Autodesk ScaleForm and it’s glitching, how do I get it to record properly?**

   ScaleForm doesn’t seem to like it when render settings (window size, vsync count) are changed while the app/game is running. This plugin removes vsync during recordings which breaks the ScaleForm rendering. Just run your app/game using a

quality settings that doesn't have vsync.

8. Version History

- **Version 2.x - ?**
  - ← Your suggestion here
  - Add support for specifying regions?

- **Version 2.02 - Monday 18 March 2013**
  - Added Unity 4.1 support
  - Fixed some platform #if issues

- **Version 2.0 - Monday 12 March 2013**
  - Added audio recording directly from Unity
  - Fixed GL.IssuePluginEvent() conflict bug with other AVPro plugins
  - Fixed DX11 recording in Unity 4.0
  - Fixed bug in audio codec listing
  - Renamed and restructured code

- **Version 1.8 - Tuesday 18 December 2012**
  - Added audio codec enumeration
  - Added Unity 4.0 support
  - GUI improved
  - Added more demos

- **Version 1.6 - Thursday 6 September 2012**
  - Added ability to pause movie capture
  - Scene capture resolution can differ from Screen resolution
  - Inspector: displays capture rate and has buttons to control capture
  - Less CPU usage
  - Optimisation: removed software RB channel swap
  - Optimisation: removed per-frame memcpy
  - Optimisation: removed vertical flip
  - GUI layout improved
  - Lots of source cleaning up

- **Version 1.5 - Monday 6 June 2012**
  - Improved smoothness of captures significantly.
  - 64-bit Windows support added.
  - Added GUI to easily set up recordings (taken from previous demo scene).
  - Added code to detect dropped frames during encoding.

- **Version 1.4 - Thursday 15 March 2012**
  - Much faster capturing due to new Unity 3.5 native API features.

- **Version 1.3 - Friday 17 February 2012**
  - Added audio for testing audio recording.
  - Autodetection of loopback audio devices.

- **Version 1.2 - Saturday 4 February 2012**
  - Added ability to capture GUI.
  - Added audio capture.
  - Added resizing to half, quarter, eighth resolution.
  - Improved capture performance and smoothness by only preparing the frame capture data when the encoder requires it.
  - Automatic disable of vsync helps performance.
  - Rounding to multiple of 4 resolution to hep codec compatability.
  - Added ability to set target frame rate (15, 24, 30, 60).
  - Added ability to set own file name.
  - Video codecs can now be configured.
  - Fixed various minor bugs.

- **Version 1.1 - Tuesday 24 Janurary 2012**
  - Removed Vista/Win7 dependency (WMV).

- **Version 1.0 - Thursday 17 January 2012**
  - Initial release submitted to Asset Store.

9. 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/
Email: contact@renderheads.com

If you are reporting a bug please include any relevant files so that we may remedy the problem as fast as possible.
