Universal Video Texture

Manual

v1.78

Thank you for purchasing the Universal Video Texture Package (UVT for short). Be it Unity Free / Pro or Mobile - This little package will allow you to use a series of sequential images as a video texture while letting you control the FPS rate, play mode, and other features to tweak for your specific needs.

Have fun!

What's new?

Version 1.78

- Unity 3.5.X compatible.
- FIXED conflicts with PlayMaker and Photon.
- FIXED a minor bug causing the external font not to compile with the build (now uses only the default Arial font).
- Renamed Enums.cs to UVT_Enums.cs
- Renamed PlayDirection to UVT_PlayDirection.
- Renamed PlayMode to UVT_PlayMode.

Version 1.75

- Major code reconstruction.
- Changed VideoTexture.cs to VideoTexture_Material.cs to differentiate it from VideoTexture_FullScreen.cs and better reflect its purpose.
- Moved audio management code to AudioManager.cs.
- Moved all enumerators to Enums.cs (all public enums now start with uppercase while their respective local variables start with lowercase).
- Added interactive controls for VideoTexture_Material using customizable controls.
- Added a simple API for controlling all video textures.
- Added Auto Play feature.
- Added Auto Desctruct feature.
- Added Auto Hide When Done feature.
- Added Auto Load Scene When Done feature.
- Added Force Audio Sync feature.
- Added the ability to start PingPong play mode with reverse.
- Added support for HeightMap / Parralax maps.

Version 1.5

- Version 1.5 introduces a new script for playing full screen videos using the built in GUI system with interactive scroll bar and scrubbing.
- Code optimizations.

Version 1.1

- Added Random Playmode - Great for video filters and masks.
- Added the ability to set a playback range (First Frame / Last Frame)
- Added the ability to turn ON/OFF synchronized audio.
- Added BruteForce low memory mode.
- Added Shared Material mode for affecting multiple objects sharing the same video texture with a single script.version 1.1

Playing UVT on an object material
(VideoTexture_Material.cs)

Workflow:

1. Export your chosen video (using your preferred video editor) as a sequence of still images (PNG, BMP, JPG, TIFF, etc..) with running numbers as a prefix or postfix.
2. Import the sequence of images into a “Resources” folder under your project's assets folder.
3. Attach the **VideoTexture_Material.cs** script to your chosen **object** and set the parameters according to your images.
4. Attach a sound source to your object if applicable and the script will play the audio in sync with your video.
5. Play!

Parameters

1. **FPS** – The frame rate of your image sequence.
2. **First Frame** – The first frame fo the sequence.
3. **Last Frame** – The last frame to the sequence.
4. **File Name** – Base file name.
5. **Digits Format** – Format of running digits. For example: 3 digits (filename_**000**.png) ,4 digits (filename_**0000**.png), etc..
6. **Digits Location** – Prefix (**000**_filename.png) or Postfix (filename_**000**.png).
7. **Playmode:**

   Once – Plays the sequence once and freezes on the last frame.

   Loop – Loops infinitely through the sequence from start to finish.

   PingPong – Plays the clip infinitely from start to finish, then backwards from finish to start.

   Random – Plays arbitrary frames from the image sequence at the frame rate chosen.
8. **Pingpong Starts With Reverse** – Starts PingPong play mode backwards from finish to start.
9. **Texture Type** – Defines your texture type in the associated material (Diffuse,

BumpMap, Illumination, etc..).

10. **Number Of Loops** – Number of loops to be played.
11. **Low memory mode** **(Requires Unity 3.5.2)** - Recommended for use with HiRes or long image sequences, especially on mobile platforms:

    Disabled – Turns off memory management.

    Normal – Leaves minimal memory footprint with lowest performance cost (default).

    BruteForce – Forces unloading of all unused assets at the chosen FPS.
12. **Scroll Bar** – The assigned scroll bar game object for controlling the video texture.
13. **CTI** – The assigned current time indicator game object for controlling the video texture.
14. **Time Code** – The time code 3D text object assigned to the video texture.
15. **Control Layer** – The layer assigned for the controls.
16. **Shared Material** – Affects every instance of the assigned material using a single script.
17. **Enable Audio** - Plays the attached audio source (if applicable) in sync with the image sequence.
18. **Force Audio Sync** – Checks and corrects Audio/Video sync with each update (Use only when having Audio/Video sync problems)
19. **Auto Play** – Starts playing the video texture automatically .
20. **Auto Destruct** – Destroys the video texture object and its controls when done (Useful for pop up videos or as a video texture alternative to a particle system).
21. **Auto Load Level When Done** – Automatically loads a specified scene when done (Useful for intros / animated logos, etc..).
22. **Level To Load** – The scene to be loaded when done.

Using interactive controls with your VideoTexture Material

In order to use interactive controls with your video texture you need to assign each control to its corresponding input in the inspector and assign the controls (and your video texture object) to a unique layer as follows:

1. Drag and drop the prefabbed **InteractiveControls** to your scene.
2. Assign the **scroll bar** game object to the **Scroll Bar** input.
3. Assign the **CTI** (Current Time Indicator) game object to the **CTI** input.
4. Assign the **TimeCode** (Text Mesh) to the **Time Code** input.
5. Assign the video texture object and all of the above game objects to a unique layer.
6. Select the layer in the **Control Layer** input.

Notes:

- Clicking the video texture object will toggle playback on/off.
- Clicking and dragging the scroll bar will enable real time scrubbing.
- Position and rotation of your controls should be handled via the parent **InteractiveControls** object.
- The prefabbed controls may be used with any custom mesh and fonts for endless variations and styles.
- In order for the controls system to work properly when using a custom mesh (which may be of different dimensions than the prefabbed scrollbar) – the box collider should be manually adjusted to fit the new mesh.

Playing UVT in full screen using the built in GUI (VideoTexture_FullScreen.cs)

Workflow:

1. Export your chosen video (using your preferred video editor) as a sequence of still images (PNG, BMP, JPG, TIFF, etc..) with running numbers as a prefix or postfix.
2. Import the sequence of images into a “Resources” folder under your project's assets folder.
3. Attach the **VideoTexture_FullScreen.cs** script to your chosen **camera** and set the parameters according to your images.
4. Attach a sound source to your object if applicable and the script will play the audio in sync with your video.
5. Play!

- Clicking the screen toggles between play and pause.
- You can scrub through the video by dragging the current time indicator (CTI) in the scroll bar.

Parameters

1. **FPS** – The frame rate of your image sequence.
2. **First Frame** – The first frame fo the sequence.
3. **Last Frame** – The last frame to the sequence.
4. **File Name** – Base file name.
5. **Digits Format** – Format of running digits. For example: 3 digits (filename_**000**.png) ,4 digits (filename_**0000**.png), etc..
6. **Digits Location** – Prefix (**000**_filename.png) or Postfix (filename_**000**.png).
7. Aspect Ratio – The horizontal/vertical aspect ratio of the video. For instnace:
8. **Playmode:**

   Once – Plays the sequence once and freezes on the last frame.

   Loop – Loops infinitely through the sequence from start to finish.

   PingPong – Plays the clip infinitely from start to finish, then backwards from finish to start.

Random – Plays arbitrary frames from the image sequence at the frame rate chosen.

9. **Pingpong Starts With Reverse** – Starts PingPong play mode backwards from finish to start.
10. **Number Of Loops** – Number of loops to be played.
11. **Low memory mode (Requires Unity 3.5.2)** - Recommended for use with HiRes or long image sequences, especially in mobile platforms:

    Disabled – Turns off memory management.

    Normal – Leaves minimal memory footprint with lowest performance cost (default).

    BruteForce – Forces unloading of all unused assets at the chosen FPS.
12. **Cti Texture** – Texture for the current time indicator.
13. **Background Texture** – Background texture for the player (bottom most layer - also prevents GUI ghosting).
14. **Scroll Bar Texture** – Texture for the scroll bar.
15. **Scroll Bar Length** – Length of the scroll bar in pixels.
16. **Scroll Bar Height** – Height of the scroll bar in pixels.
17. **Scroll Bar Offset** – The scroll bar distance from the bottom edge of the screen in pixels.
18. **Timecode Size** – The size of the timecode font.
19. **Show Scroll Bar** – Displays the scroll bar during playback and enables scrubbing.
20. **Show Timecode** – Displays the current timecode.
21. **Enable Audio -** Plays the attached audio source (if applicable) in sync with the image sequence.
22. **Force Audio Sync** – Checks and corrects Audio/Video sync with each update (Use only when having Audio/Video sync problems)
23. **Auto Play** – Starts playing the video texture automatically .
24. **Auto Hide When Done** – Hides the video texture and its controls when done (Useful for pop up videos / cut scenes)
25. **Auto Load Level When Done** – Automatically loads a specified scene when done (Useful for intros / animated logos, etc..).
26. **Level To Load** – The scene to be loaded when done.

Controlling UVT through script

Version 1.75 introduces a simple API for controlling video textures through script in run time:

Stop() - Pauses playback.
Sync() - Syncs audio and video.
Play() - Starts/Resumes playback.
TogglePlay() - Toggles playback on and off.
ChangeDirection() - Change playback direction (forwards, backwards).
CurrentPlayState() - Returns current playback state.
CurrentPlayDirection() - Returns current playback direction.
CurrentPosition() - Returns current playback position (ranges from 0 to 1).
EnableControls(bool) – Toggles interactive controls on and off:

> *EnableControls(true)* - Turns on interactive controls.
> *EnableControls(false)* – Turns off interactive controls.

Each of the above methods can be used directly inside the VideoTexture script or through reference from another script.

For instance – in the case of **Stop():**

Inside the VideoTexture script you can simply code *Stop();* and the the video texture will pause.
If you want to stop the video texture from another script you can reference it like that:

*(GameObject.Find("**Your UVT Object**").GetComponent("**VideoTexture_Material**") as **VideoTexture_Material)**.Stop()* ;

or

*(GameObject.Find("**Your UVT Object**").GetComponent("**VideoTexture_FullScreen"**) as **VideoTexture_FullScreen**).Stop();*

Naturally the first example should be used when referencing **VideoTexture_Material** while the latter when referencing **VideoTexture_FullScreen.**

A more convenient way to do that would be to declare a local VideoTexture field and use it to cache the VideoTexture reference at the Start() method:

```
VideoTexture_Material myVideoTexture;

Start()
{
        myVideoTexture = GameObject.Find("Your UVT Object").
        GetComponent("VideoTexture_Material") as VideoTexture_Material;
}
```

Now you can access your video texture much more easily:

```
Update()
{
        myVideoTexture.Stop();
//or    myVideoTexture.Play();
//or    myVideoTexture.TogglePlay();
        .
        .
        etc..
}
```

Sample Scenes

- **SampleScene_FullScreen –** Demonstrates a full screen UVT and interactive controls using Unity's built in GUI.
- **SampleScene_Material –** Demonstrates a material UVT with interactive controls using customizable game objects
- **SampleScene_Material_MultiScreen –** Demonstrates two material UVTs with independent controls using the same video texture.

Use the A,W,D,S to move around and the middle mouse button to look around (Not applicable to SampleScene_FullScreen).
Click and drag the scrollbar's current time indicator (CTI) to scrub through the video texture.

General Notes

- The following texture resolutions are highly recommended for general use: 256x256 / 512x256 / 512x512 / 1024x512.
- Unity's editor may not play the audio source in sync during the first iteration due to an initial lag. That shouldn't affect sync in stand alone builds.