Hardware-Accelerated Video Decode in Unity* 3D

By Bryan Mackenzie

As it stands, Unity* does not provide hardware-accelerated decode for video playback. Therefore, playing video on a low-end device could suffer dropped frames, low frame rates, and poor quality. The provided project illustrates how to merge the hardware decode path within Unity. The main components within this solution are the creation of an external video texture mapped to Unity, the creation of the media player, and the calls exposed to Unity.

Video Texture Creation

To leverage the media player component, an external texture is created and shared with the Unity rendering component. To gain access to the Unity rendering component we use the IUnityInterface to retrieve the IUnityGraphicsD3D11 interface, which provides access to the Unity rendering component used to call CreateTexture2D, creating a staged texture on the Unity device. CreateTexture2D then receives the ID3D11Texture2D interface.

The D3D device is used to call CreateShaderResourceView, passing-in the ID3D11Texture2D that will serve as the input to the shader. The CreateShaderResourceView also receives the ID3D11ShaderResourceView, which provides access to the shader-resource-view interface, used to access data of the passed-in resource; in this case, the ID3D11Texture2D texture.

At this point, a shared texture is created and a handle to the shared texture is returned. The handle is then used to open the shared resource (ID3D11Texture2D), where the IDXGISurface surface of the texture is retrieved. CreateDirect3D11SurfaceFromDXGISurface is then called to create an instance of IDirect3DSurface from an IDXGISurface, whereupon a video frame is available. The frame is then copied to the IDirect3DSurface video surface which, as stated, is shared with the Unity rendering pipeline.

Media Player

To begin, we first create a new instance of the Windows Media* playback media player interface, which provides access to media playback functionality such as play, pause, seek, and so on. Once we have the interface, we load content by calling CreateFromUri, which creates a MediaSource* from the passed-in Uri, where MediaSource provides a means of accessing media data.

We then associate the MediaSource to a MediaPlaybackItem, which acts as a wrapper around a MediaSource, exposing audio and video tracks within the MediaSource.

Unity* Integration

The purpose of this project is to detail the integration of hardware-accelerated video in Unity, how it is achieved, and how you can leverage the code to integrate accelerated video into your project. A wrapper is created to expose the relevant calls to create, load, and interact with the media. While not within this article's scope, more information on creating and exposing native dynamic libraries to Unity can be found here. The project does not expose all available features of the media player, but does expose core functionality to provide a jump-off point to further expand upon with new features. The following functionality is currently exposed and available for integration:

CreateMediaPlayback: Create media player instance

CreatePlaybackTexture: Return native texture to a pointer

LoadContent: Load from Uri

SetPosition: Set current playback position (seek)

GetPosition: Get current playback position

GetDuration: Retrieve the video duration

GetPlaybackRate: Retrieve current playback speed

StateChangedCallback: Media playback change event callback

Play: Play video

Stop: Stop video

Pause: Pause video

Notices

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting www.intel.com/design/literature.htm.