**MP8 - Manual**

# Known issues:

* Incompatible Unity versions: 2018.2.0 to 2018.2.7. Solution: Install at least version 2018.2.8

# Run the example scenes:

Include all the 3 scenes found in MediaPlayer8/Scenes, with **\_Main.unity** being the first one (the entry point).

* Android: Build for Android and test on the physical device (no emulator support for now)
* Windows: (coming soon)
* IOS: (coming soon)

# Quick Start:

* Add a **SimpleMediaPlayer8** component to a **GameObject** with a renderer
* Create a material (or use the included MediaPlayer8/Materials/movie.mat) that has blank texture set (or use included MediaPlayer8/Textures/movie.png) and apply it to the renderer of the GameObject that you want the video information to show on
* Drag the used texture onto the Rendering->"Target Texture" property in the **SimpleMediaPlayer8**'s inspector
* Add a **ControlPanel** component to any GameObject and use the inspector to add one or more media sources. This component is placed oursede the BaseScripts fodler, because it's not a necessary functionality for the media player to work, rather it's an utility to demo the package.
* Create a button with the suggestive text "Next" and assign ControlPanel.LoadNext() to it's onClick event via Inspector. If you have multiple sources, clicking Next will cycle through all of them, loading them one by one as expected.
* Build & Run & Grab some popcorn

# Package Structure

The most important folders are:

* MediaPlayer8/BaseScripts - the core scripts, project-independent
* MediaPlayer8/Scripts - the scripts used to build the example scenes
* MediaPlayer8/BaseAssets/Plugins - the native plugins for each platform (currently, only android armeabi-v7a)
* MediaPlayer8/Scenes - \_Main.unity is the starting scene, that'll load one of Example.unity or AdaptiveCacheExample.unity (assuming the "AdaptiveCache" upgrade is present)
* MediaPlayer8/StreamingAssets - this folder includes "short\_video1080p.m4v" that must be moved to the \*root\* StreamingAssets in order for the local playback in the example scene to work

# Core Modules

1. **MediaPlayer8:**

* Base abstract class for *SimpleMediaPlayer8* and *AdaptiveCacheMediaPlayer8*
* Handles the native texture creation & updating
* Stores a reference to a *ControllerProxy* in the property *NativeController*, which provides the full set of commands & callbacks to- and from the native player
* Call *Initialize* first time and you'll receive a callback to *onDone* or *onError<string>*
* If successful, call *Load* with an URL to your media file/stream or use "path/to/file/file.extension" to play the file in "<projectPath>/Assets/StreamingAssets/path/to/ffile/file.extension"
* Use MediaPlayer.NativeController to access common functionality

1. **SimpleMediaPlayer8**
   * Extends **MediaPlayer8** and supports playback for all the streaming formats listed in the asset's description and local files, audio or video
   * Exposes *SimpleControllerProxy*, through which commands/callbacks are made
   * Exposes *SimpleParameters*, which are simply the player's configuration
2. **AdaptiveCacheMediaPlayer8:**
   * Also extends **MediaPlayer8**, but only supports H.264 container format
   * The purpose of this component is to stream H.264 content without any interruptions due to network slow speed. The "adaptive" part takes care of disk+ram caching & network speed predictions so after the initial waiting time, the video will play without any re-buffering (given there is enough disk and/or ram available). No disk or ram resources are used beyond those needed to achieve no interruptions.
   * Another feature of the "adaptive" upgrade is fragmented caching: nothing is wasted; if you re-wind, the content is played from disk (there is a limit for disk cache's size, of course)
   * Only available in the upgrade package (even if the C# code includes it, the native part it's not, if the package was not upgraded)
3. **ControllerProxy:**
   * Currently, this only includes functionality for Android, but will act abstractly when other platforms will be supported
   * Includes functionality like *SetPlayWhenReady(bool), RequestSeek01(double), GetBufferLevel01RelativeToDuration(), GetProgress01()* etc.
   * The *Events* property can be used to subscribe to common events
   * *Dispose()* is automatically called in the **MediaPlayer8**'s *OnDestroy()*

# Additional Info

You can see that there are 2 submodules: Simple and AdaptiveCache. Each of them has it's own implementation of **MediaPlayer8**, **ControllerProxy**, **Parameters** etc.

For general playback, use the *Simple* module

If you want disk caching and any benefits of the *AdaptiveCache* module, use it instead