NEXSTREAMING

NexPlayer™ Plugin for Unity

Version 11.0

Technical Reference Manual

Generated May 9th, 2017

Index

NexPlayer™ Plugin for Unity Engine

Legal Notices

Disclaimer for Intellectual Property

Copyright

Abstract

NexPlayer™ Plugin for Unity provides an interactive video playback for Android, iOS and Windows devices through Unity. NexPlayer™ Plugin for Unity has been built to be reliable and robust without any sacrifice in performance, and has proven compatibility with international standards. NexPlayer™ for Unity works exclusively in conjunction with the native NexPlayer SDK and can benefit from all NexPlayer™ features, including intelligent ABR, PD/Local Playback, HTTP Live Streaming (HLS), DASH have a customizable feature set and API and many more.

This documentation is a work in progress.
Additional details and sample code will continue to be added.

Please also be aware that for testing and development purposes, the Android/iOS emulator should not be used with the NexPlayer™ as there are known differences and issues between the emulator and actual devices, including the fact that the OpenGL renderer does not run properly within the emulator environment.
Frequent testing on actual devices is strongly recommended during development and all apps should be tested on actual devices prior to release.

NexPlayer™ Plugin for Unity Capabilities and Limitations

Protocols Summary

Platform	Supported Graphics APIs	HLS	PD	DASH	Local	Inside App (Streaming Assets)
Windows 8 and above (x86 and x64)	Direct3D11 (aka DX11) with feature level 9.3 and above	O	O	O	O	O
Windows Editor	Direct3D11 (aka DX11) with feature level 9.3 and above	O	O	O	O	O
Android (armeabi-v7a and x86)	OpenGLES2, OpenGLES3	O	O	O	O	O
iOS	OpenGLES2, OpenGLES3, Metal	O	O	O	O	O
WebGL	WebGL 1, WebGL 2	O	O	O	X	O

Features

- Support protocols for ABR algorithm, including HLS and DASH
- Support for progressive download (eg. online .mp4)
- Complete API including:
  - Play / Pause
  - Seek
  - Video resolution
  - Last millisecond buffered
- Useful callbacks including:
  - Information about the buffering state
  - Track change
  - State of the playback
- Widevine DRM on Android and iOS for DASH videos

Requirements

NexPlayer™ Plugin for Unity supports Android, iOS, Windows and WebGL.

In Android the following is **required**:

- Use of OpenGL ES version 2.0 or 3.0
- Use of the Android version 2.3.1 or above

In iOS the following is **required**:

- Use of OpenGL ES version 2.0
- Use of iOS or above 8.0 or above

In Windows and the Windows Editor the following is **required**:

- Use of Direct3D11
- Use of Windows 8.0 or above

In WebGL (supported functionalities can be checked at https://html5test.com/) the following is **required**:

- HTML5 video element support
- WebGL 1 or WebGL 2 support
- Media Source Extension to use HLS or DASH*
- Ability to control the playback of the video element at any point**

* DASH doesn't work on Safari on Mac due to a reported bug
** Chrome on Android doesn't allow to autoplay videos or to control the playback outside of a touch callback (which Unity doesn't provide). This affects the use of Chrome on Android

NexPlayer360™

NexPlayer™ Plugin for Unity includes many of the features of NexPlayer360 SDK, such as:

- Touch input, including movement, zoom and rotation of the camera
- Gyroscope input to move the camera
- Mouse input to move the camera
- Automatic Ground Leveler to stabilize the video
- Custom shaders to map 2D, 3D Over/Under and 3D Left/Right 360 videos
- Customizable key controls
- VR supported scenes

Please bear in mind that to move the camera with the movement of the phone, a gyroscope and the support by the Unity Gyroscope API is needed.

Integration Guide

A fully working example of NexPayer™ for Unity is provided in the Unity Package. It can be imported into an Unity project by double clicking it.

An example scene is available in the example Unity project in the Unity folder ./Assets/NexPlayer/Scenes/ in the file NexPlayer.unity.

A 360 example scene is available in the example Unity project in the Unity folder ./Assets/NexPlayer/NexPlayer360/Scenes/NexPlayer360.unity in the file NexPlayer.unity.

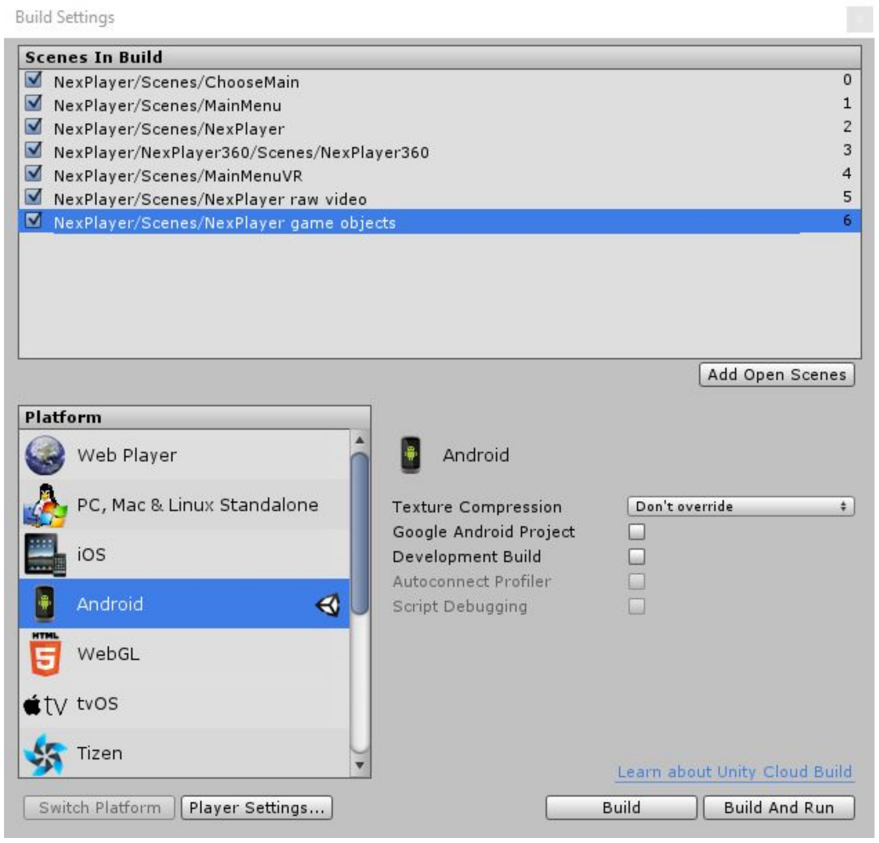
In order to integrate NexPlayer™ the compatible graphics APIs need to be selected. That can done manually in the “Player Settings” section of Unity for each platform. If the helper component NexEditorHelper.cs is attached to any GameObject it will include a graphics UI to automatically detect any conflict regarding the graphics API, and it will promptly solve it.

A sample way of integrating NexPlayer™ with any GameObject with a Unity Material can be used using the component NexPlayer.cs.

To allow any remote video on Android “Internet Access” needs to be set to true in the Unity player settings. Also to view HTTP videos on iOS “Allow downloads over HTTP” needs to be enabled. A quick and easy way to enable these settings is using the helper component (NexEditorHelper.cs).

Sample scenes

Add the following scenes to the Unity build:


Build Settings

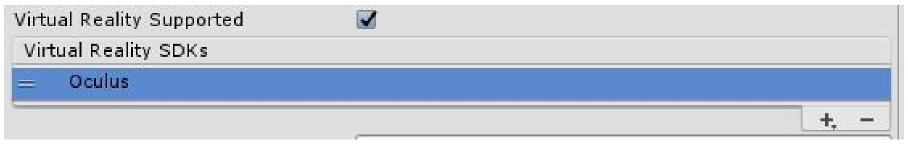
Scenes In Build

✓	NexPlayer/Scenes/ChooseMain	0
✓	NexPlayer/Scenes/MainMenu	1
✓	NexPlayer/Scenes/NexPlayer	2
✓	NexPlayer/NexPlayer360/Scenes/NexPlayer360	3
✓	NexPlayer/Scenes/MainMenuVR	4
✓	NexPlayer/Scenes/NexPlayer raw video	5
✓	NexPlayer/Scenes/NexPlayer game objects	6

Android

Texture Compression: Don't override
Google Android Project ☐
Development Build ☐
Autoconnect Profiler ☐
Script Debugging ☐

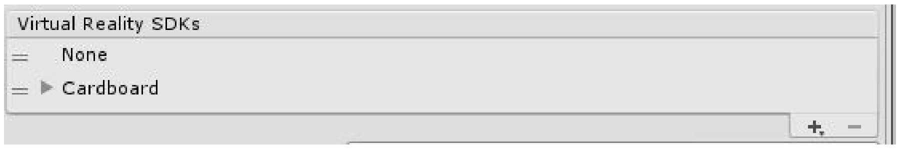
Select Virtual Reality in the Player Settings to generate a build for the Gear VR. Remeber to add the (remember to generate the OSIG for your device https://developer.oculus.com/osig/):


Virtual Reality Supported ✓

Virtual Reality SDKs

Oculus

Select None and Cardboard to try an APP that transitions to VR at runtime (in Unity 5.6 and above):


Virtual Reality SDKs

None

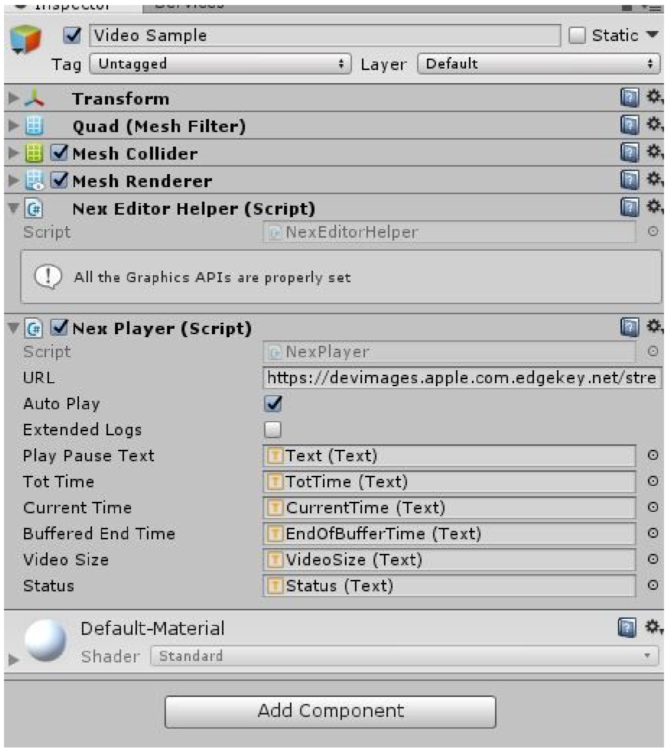
Cardboard

Snippets

Examples of how to use the main features of NexPlayer can be found in this section.

Playing a video

An example of using NexPlayer can be found in the script NexPlayer. It's recommended to also attach the Nex Editor Helper script that will make sure the correct graphics APIs are selected.


Video Sample
Static
Tag Untagged
Layer Default
Transform
Quad (Mesh Filter)
Mesh Collider
Mesh Renderer
Nex Editor Helper (Script)
Script NexEditorHelper
All the Graphics APIs are properly set
Nex Player (Script)
Script NexPlayer
URL https://devimages.apple.com.edgekey.net/stre
Auto Play ✓
Extended Logs
Play Pause Text Text (Text)
Tot Time TotTime (Text)
Current Time CurrentTime (Text)
Buffered End Time EndOfBufferTime (Text)
Video Size VideoSize (Text)
Status Status (Text)
Default-Material
Shader Standard
Add Component

The NexPlayer script needs to be attached to some game object that has a material and a texture. The URL and the text fields used to update the status can be personalized.

A custom implementation of NexPlayer can also be done manually. The script NexPlayer can be used as a reference. First the player needs to be created, an action should be registered to receive the callbacks, the player should be initialized, and the coroutine needs to be started:

```
void Start()
{
    // Creation of the NexPlayer instance
    player = NexPlayerFactory.GetNexPlayer();
    // Register to the events of NexPlayer
    player.OnEvent += EventNotify;
    // Initialize NexPlayer with an URI
    player.Init(URL, true, false);

    // The coroutine needs to be started after the player is created an initialized
    StartCoroutine(player.CoroutineEndOfTheFame());
}
```

The update method of the player needs to be called at the Update callback of the MonoBehaviour object:

```
void Update()
{
    player.Update();
}
```

The previously used Action needs to be declared. It will provide a number of helpful callbacks. Properly assigning the texture to the material used by the game object should be done in the correct callback:

```
void EventNotify(NexPlayerEvent paramEvent, int param1, int param2)
{
    switch (paramEvent)
    {
        case NexPlayerEvent.NEXPLAYER_EVENT_TEXTURE_CHANGED:
            // It's important to change the texture of every Unity object that should
display the video frame when this callback is called
            GetComponent<Renderer>().material.mainTexture = player.GetTexture(); break;
    }
}
```

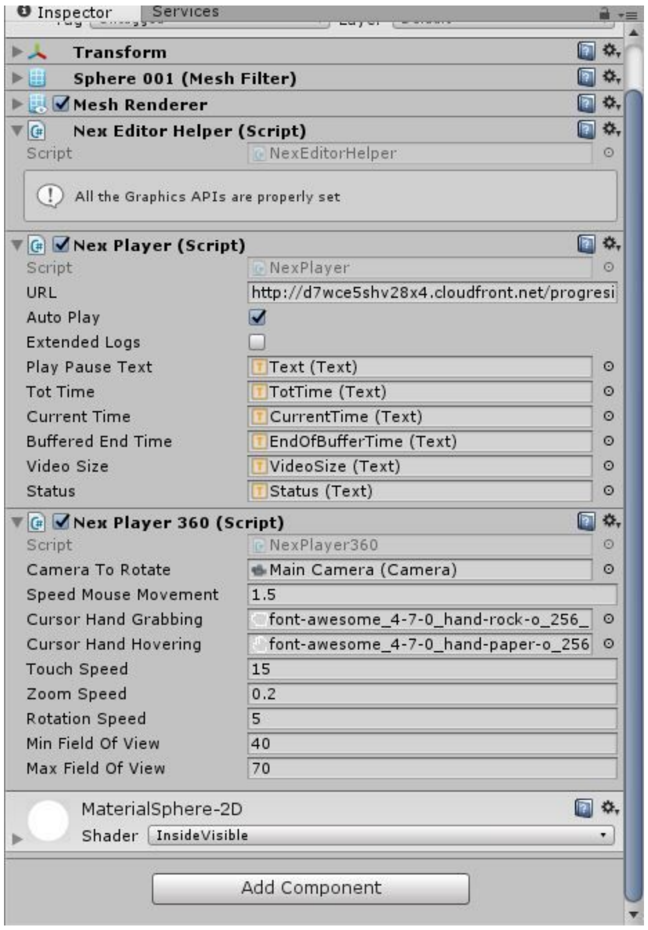
To release the player, call the ClosePlayback method and wait for the NEXPLAYER_EVENT_CLOSED callback:

```
public void ToogleQuit()
{
   player.ClosePlayback();
}

void EventNotify(NexPlayerEvent paramEvent, int param1, int param2)
{
    ...
    switch (paramEvent)
    {
        ...
        case NexPlayerEvent.NEXPLAYER_EVENT_CLOSED:
            {
                Application.Quit();
            }
            break;
    }
}
```

Present 360 video

Using the script NexPlayer360.cs in a gameobject provides the more important functionalities of a 360 viewer. It has touch input (movement, rotation and zoom with multi touch support), gyroscope control, and mouse control. Furthermore it uses the “Automatic Ground Leveler” feature to stabilize the video for the user. It can be fully customized in the editor:



Nex Editor Helper (Script)
Script NexEditorHelper
All the Graphics APIs are properly set

Nex Player (Script)
Script NexPlayer
URL http://d7wce5shv28x4.cloudfront.net/progresi
Auto Play ✓
Extended Logs
Play Pause Text Text (Text)
Tot Time TotTime (Text)
Current Time CurrentTime (Text)
Buffered End Time EndOfBufferTime (Text)
Video Size VideoSize (Text)
Status Status (Text)

Nex Player 360 (Script)
Script NexPlayer360
Camera To Rotate Main Camera (Camera)
Speed Mouse Movement 1.5
Cursor Hand Grabbing font-awesome_4-7-0_hand-rock-o_256_
Cursor Hand Hovering font-awesome_4-7-0_hand-paper-o_256
Touch Speed 15
Zoom Speed 0.2
Rotation Speed 5
Min Field Of View 40
Max Field Of View 70

MaterialSphere-2D
Shader InsideVisible

That script can also be used as a reference for a custom integration. To integrate the AutomaticGroundLeveler an instance should be created, and the method step method should be called appropriately:

```
void Awake()
{
    agl = new AutomaticGroundLeveler();
}


void Update () {
    //Move the camera with a custom login

    //Stabilize the camera
    agl.AutomaticGroundLevelerStep(cameraToRotate.transform, latestAttitude, rotating);
}
```

Documentation

An alternative HTML version of the latest version of this documentation is available at the website http://nexplayer-plugin-for-unity.s3-website-eu-west-1.amazonaws.com.

NexPlayerBase Class

NexPlayerBase abstract class that determines the available methods for all the supported platforms. Use this with the NexPlayerFactory.GetNexPlayer();

The NexPlayerBase type exposes the following members.

◢ Methods

Name	Description
ClosePlayback	Close the player playback. The NEXPLAYER_EVENT_CLOSED event will be launched after closing the player
CoroutineEndOfTheFrame	Coroutines that needs to be started after player is initialized, and stopped before setting the player to null. Using this in this way it's mandatory and extremely important
GetAudioStreamCount	Returns the number of streams
GetAudioStreams	Returns an array of all the available audio streams. This method is not available for all the platforms

GetBufferedEnd	Returns the last millisecond buffered on the player. This is useful to display a seek bar with a secondary progress, indicating the buffered content
GetCurrentAudioStream	Returns the current audio stream. This method is not available for all the platforms
GetCurrentTime	Returns the current time of the playback
GetCurrentTrack	Returns information about the current track. This method is not available for all the platforms
GetStatusPlayer	Returns the status player
GetTexture	Returns a Unity Texture that is updated with the video frames. Update this on your Unity Objects when the event NEXPLAYER_EVENT_TEXTURE_CHANGED is triggered
GetTotalTime	Returns the total time of the of the playback
GetTracks	Returns information about all the possible tracks. This method is not available for all the platforms
GetTracksCount	Returns the number of tracks
GetVideoHeight	Returns the current height of the video. This changes when the event NEXPLAYER_EVENT_TRACK_CHANGED is triggered
GetVideoWidth	Returns the current width of the video. This changes when the event NEXPLAYER_EVENT_TRACK_CHANGED is triggered

Init(String)	Initializes the player with an URI, and auto-plays it without the extended logs enabled
Init(String, Boolean, Boolean)	Initializes the player with an URI, it set the autoPlay and useExtendedLogs
Init(String, Boolean, Boolean, String)	Initializes the player with an URI, it set the autoPlay, useExtendedLogs and a Widevine server Key URI for the DASH content. This method is not available for all the platforms
Pause	Pauses the playback
Resume	Resumes the playback
Seek	Seeks in the playback, moving the playback to the specified millisecond
SetAudioStream	Set a stream to be used. The possible audio streams can be obtained from the method GetAudioStreams. This method is not available for all the platforms
SetTrack	Set a tack to be used. Using this disables ABR. The possible tracks can be obtained from the method GetTracks. This method is not available for all the platforms
StartPlayBack	Starts the video playback. Call this after the event NEXPLAYER_EVENT_INIT_COMPLEATE if auto play is disabled
Stop	Stop the playback

Update	This Update method should be called on the Update callback of a Unity MonoBehaviour. Using this in this way it's mandatory and extremely important

◢ Fields

Name	Description
OnEvent	The event where registered actions will be notified when there is a player event. Register to this after getting the instance from NexPlayerFactory.GetNexPlayer() and before initializing the player

NexPlayerFactory Class

NexPlayer factory that creates NexPlayerBase instances.

The NexPlayerFactory type exposes the following members.

◢ Constructors

Name	Description
NexPlayerFactory	Initializes a new instance of the NexPlayerFactory class

◢ Methods

Name	Description

GetNexPlayer	Creates an instance of NexPlayerBase for the currently selected platform. This allows to use one API for all the supported platforms. It's recommended to use this and not instantiate directly a class that extends NexPlayerBase.

NexPlayerStatus Enumeration

Possible status of NexPlayer

◢ Members

Member name	Value	Description
NEXPLAYER_STATUS_CLOSED	1	The player is closed and not initialized
NEXPLAYER_STATUS_OPENED	2	The player is opened, but not playing
NEXPLAYER_STATUS_PLAYING	3	The player is playing the video
NEXPLAYER_STATUS_PAUSED	4	The player is paused
NEXPLAYER_STATUS_BUFFERING	5	The player is buffering new content

NexPlayerEvent Enumeration

Possible events that NexPlayer sends to every Action subscribed to the EventNotify OnEvent

◢ Members

Member name	Value	Description
NEXPLAYER_EVENT_INIT_COMPLEATE	1	The Player has been initialized, but it's not playing
NEXPLAYER_EVENT_PLAYBACK_STARTED	2	The player has stated playing the video
NEXPLAYER_EVENT_END_OF_CONTENT	3	The player has reached the end of the video playback
NEXPLAYER_EVENT_ON_TIME	4	This will be called at regular intervals, and can be useful to update the UI
NEXPLAYER_EVENT_BUFFERING_STARTED	5	The player has stated buffering
NEXPLAYER_EVENT_BUFFERING_ENDED	6	The player has buffered enough content and has resume the playback
NEXPLAYER_EVENT_TEXTURE_CHANGED	7	The internal texture has changed, and NexPlayerBase.GetTexture() should be called to retrieve the texture that should be set in all the objects that will display the video
NEXPLAYER_EVENT_TRACK_CHANGED	8	The track of the playback has changed. This is especially useful for protocols with several resolution

		tracks https://en.wikipedia.org/wiki/Adaptive_bitrate_streaming
NEXPLAYER_EVENT_PLAYBACK_PAUSED	9	The playback has been paused
NEXPLAYER_EVENT_CLOSED	10	The player has been closed. The Close coroutine can now be stopped and the App can exit gracefully
NEXPLAYER_EVENT_ERROR	11	There was an error in the playback

NexPlayerError Enumeration

Possible values of param1 when an error event is generated

◢ Members

Member name	Value	Description
NEXPLAYER_ERROR_GENERAL	1	General unknown error
NEXPLAYER_ERROR_SRC_NOT_FOUND	666	The URI is not supported or not found. Make sure the URI actually exists and check that necessary permissions are set (eg. if you are playing a video on Android platform set the Player Setting "Internet Access" to "Required" (streaming videos) and the "Write Permission" to "External" (local videos)

Frequently Asked Questions

- **What files are included in the NexPlayer™ Plugin for Unity?**

All the necessary files to integrate it with any Unity project, sample scenes for a normal and 360 video, this documentation and the release notes. The files are distributed in the following directories:

```
./NexPlayer
        /NexPlayer_Plugin_for_Unity_Reference_Manual.pdf
        /NexPlayer_Plugin_for_Unity_Release_Notes.txt
        /License.txt
        /NexPlayer360
                /Resources
                        /cardboard.png
                        /font-awesome_4-7-0_compass_396_50_ffffff_none.png
                        /font-awesome_4-7-0_hand-paper-o_256_0_ffffff_none.png
                        /font-awesome_4-7-0_hand-rock-o_256_0_ffffff_none.png
                        /gray_dot.png
                        /SpherePoints.FBX
                /Scenes
                        /NexPlayer360.unity
                /Scripts
                        /Nex360DLL.dll
                        /NexPlayer360.cs
                        /NexPlayer360KeyControls.cs
                        /NexVRInteractable.cs
                        /NexVRInteractableSeekBar.cs
        /Prefabs
                /Main Camera.prefab
                /Sample Controls.prefab
                /SF Button.prefab
                /SF Scene Elements.prefab
                /SF Title.prefab
                /StereoCameras.prefab
        /Resources
                /font-awesome_4-7-0_arrow-left_396_50_ffffff_none.png
                /font-awesome_4-7-0_link_32_28_ffffff_none.png
                /font-awesome_4-7-0_pause_256_100_d9d5d4_none.png
                /font-awesome_4-7-0_play_256_100_d9d5d4_none.png
        /menu.png
        /Nexplayer Unity.png
        /Roboto-Bold.ttf
        /Roboto-Regular.ttf
        /seekbar_background.png
        /stars.jpg
        /UIButtonDefault.png
        /Materials/
                /InsideVisible.shader
                /InsideVisible-3D-LEFT.shader
                /InsideVisible-3D-OVER.shader
                /InsideVisible-3D-RIGHT.shader
                /InsideVisible-3D-UNDER.shader
                /MaterialLeft.mat
                /MaterialMono.mat
                /MaterialOver.mat
                /MaterialRight.mat
                /MaterialUnder.mat
                /Nexplayer Unity.mat
/Scenes
        /ChooseMain.unity
        /MainMenu.unity
        /MainMenuVR.unity
        /NexPlayer game objects.unity
        /NexPlayer raw video.unity
        /NexPlayer.unity
/Scripts
        /Editor
                /iOS Build.cs
                /NexCustomEditor.cs
                /SunShaftsEditor.cs
        /UI
                /ChooseMain.cs
                /Main.cs
                /NexMainCube.cs
                /NexSeekBar.cs
                /NexUIController.cs
                /NexVideoObject.cs
                /PanelManager.cs
                /StereoMode.cs
                /TiltWindow.cs
        /NexEditorHelper.cs
        /NexPlayer.cs
```

```
                    /NexPlayerAndroid.cs
                    /NexPlayerBase.cs
                    /NexPlayerFactory.cs
                    /NexPlayeriOS.cs
                    /NexPlayerWebGL.cs
                    /NexPlayerWindows.cs
          /VRMenu
./Plugins
          /Android
          /iOS
          /x86
          /x86_64
```

- **What is adaptive bitrate streaming or ABR?**

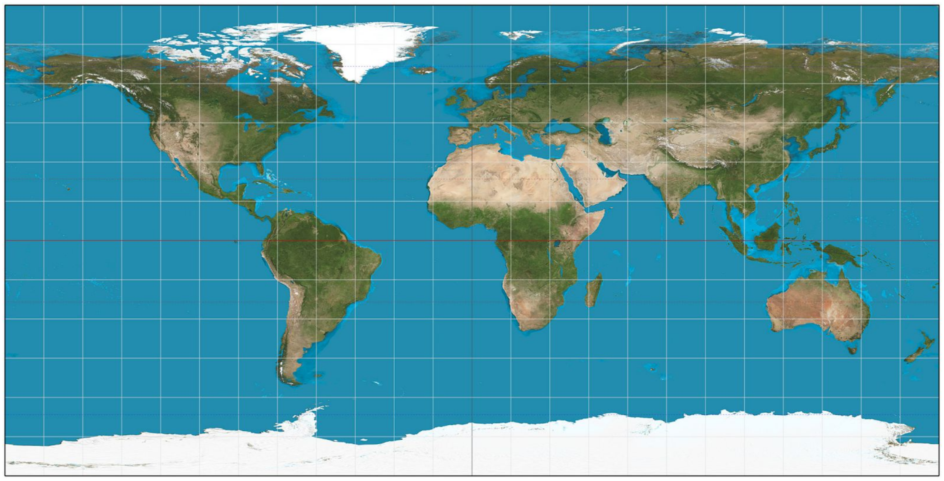
It's a technique used in streaming videos online that offers multiple videos resolutions for the same content to the player. The player then changes to the correct resolution resolutions depending on the network conditions and the device at runtime. NexPlayer supports both HLS and DASH.

- **Can I play .mp4 that are hosted online?**

Yes, that's called progressive download, and NexPlayer fully supports it.

- **How does 360 video work?**

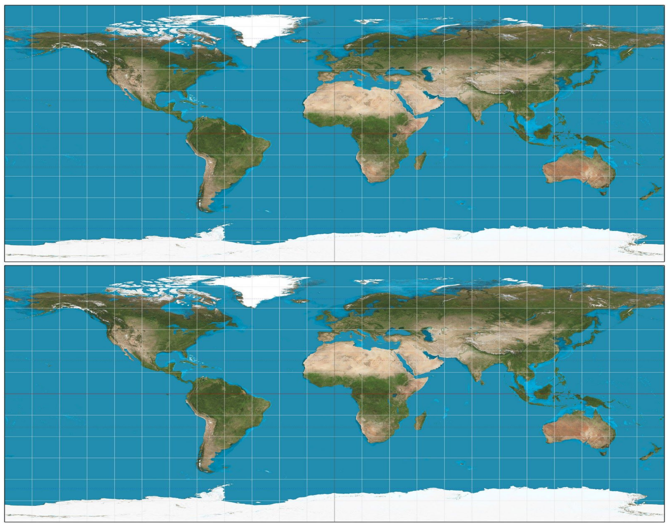
A 360 video is contained into a single video. Usually 360 videos are recorded with several cameras, and later those videos are "stitched" together into a single equirectangular video. This video is the one that needs to be provided to the player. This is an example of a equirectangular image, that could be an individual frame:



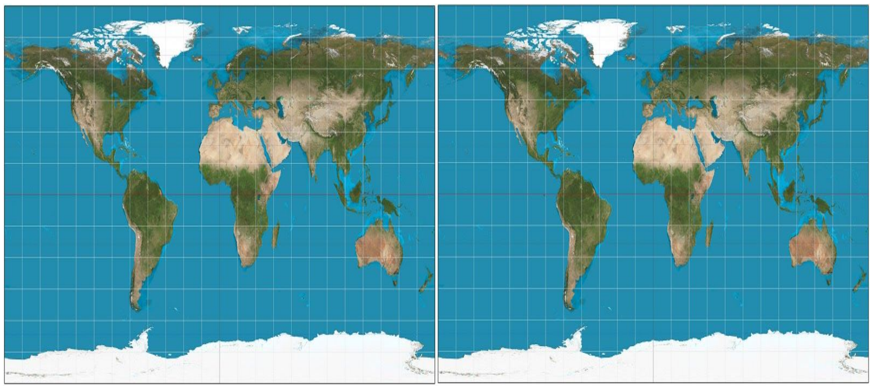
Each frame is then displayed in such a way that reproduces the real world.

- **What formats of 3D 360 videos are supported?**

360 3D videos are useful for VR Apps, as they generate the sensation of depth in the video, showing different images to each eye. The two types of 3D 360 video formats are supported. There are two formats, Over-Under where the top half of the video is used for the frames of one eye, and the bottom part are used for the other eye. One example of a frame would be:



The other format is Side-by-Side, where the left half of the video is used for one eye and the other half if used for the other eye. One example of a frame would be:



There are helper shaders and materials in the folder ./NexPlayer/NexPlayer360/Resources that allow to show 3D 360 videos into a Unity App.

- **How can I stabilize a 360 video for the best user experience?**

Moving the camera properly in a 360 video can be a difficult task that can lead into having the video askewed for the user. NexPlayer with NexPlayer360 provides a way to stabilize dynamically the video for the user. This functionality is called “Automatic Ground Leveler”. An example can be found in the class ./NexPlayer/NexPlayer360/Scripts/NexPlayer360.cs.

- **Can I play DRM protected videos?**

Yes, NexPlayer supports the free DRM Widevine for DASH videos on Android and iOS.

- **What version or Unity should I use?**

The use of the latest Unity version is recommended.

Unity can be downloaded at https://store.unity.com/.