4DVIEWS

Unity Plug-in

Documentation

Unity plug-in for 4DS

Version 3.11

What's new in version 3.11

New Features:

- Decryption of 4ds files previously encrypted with 4DCoder
- Compatibility with latest OS and devices

Compatibility

This version of the plug-in is compatible with:

- Windows 10, x64, and UWP x86, x64, arm64
- Mac OS >= 15 x64 and arm64
- Android >= 9.0, arm64, armv7
- iOS >= 17, arm64

What the plug-in CAN do

- Play 4ds files with the new compression algorithm
- Play files which are on the device, as before
- Play files which are on a web server, over HTTP protocol

What the plug-in CAN'T do

- Play 4ds files with the old compression format
- Play distant files over other protocol than HTTP

Before starting

Some video tutorials on this Unity plug-in usage are available on the 4DViews website, in the Creators Platform section : https://creators.4dviews.com/viewTutorial?id=2

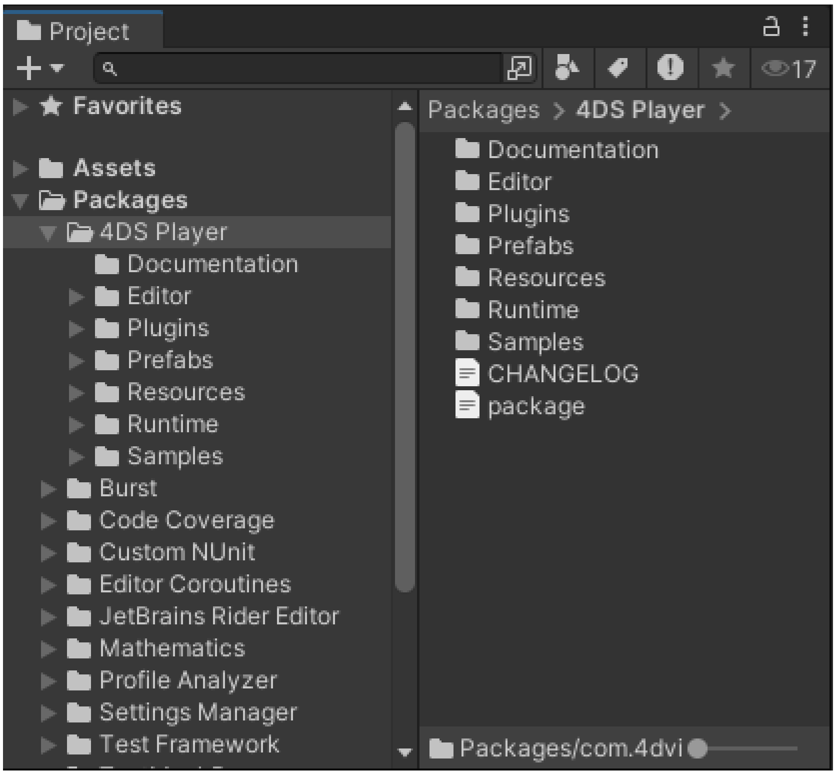
Installation

The plug-in is distributed under a zip file, containing license, documentation and a Unity package.

To install the plug-in, please unzip this file, then run Unity and choose in the menu Assets > Import Package > Custom Package select the Unity package file, or simply drag and drop the unity package file in the assets window of unity.

The Unity Package Manager will take care of importing the plugin and its dependencies.

Once the package is imported, the Packages directory now contains a new directory called 4DS Player:



The 4DS Player directory contains the prefab, material and internal scripts to manage the plug-in.

The Editor directory contains other scripts specific to the Unity editor window.

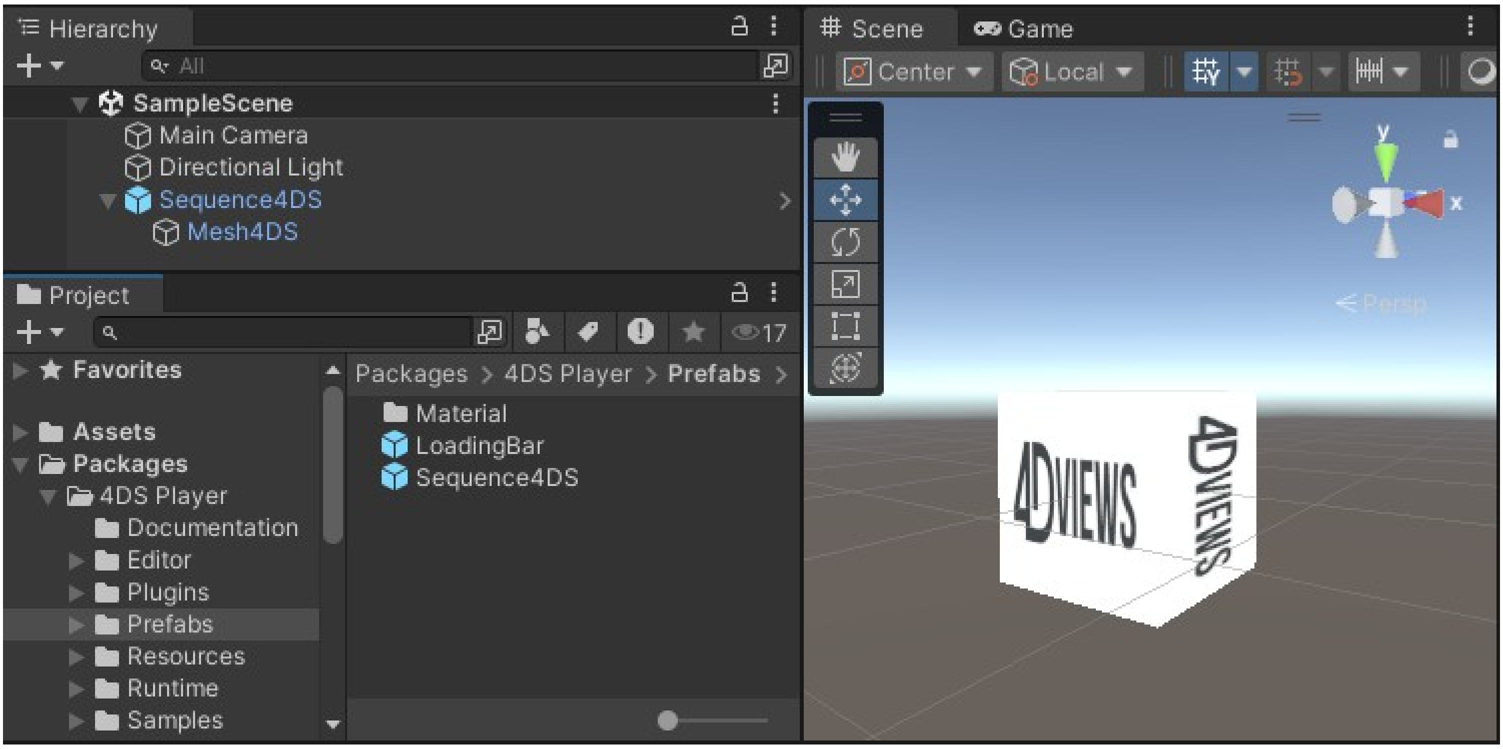
The Plugins directory contains all the platform native libraries.

Start using the plug-in

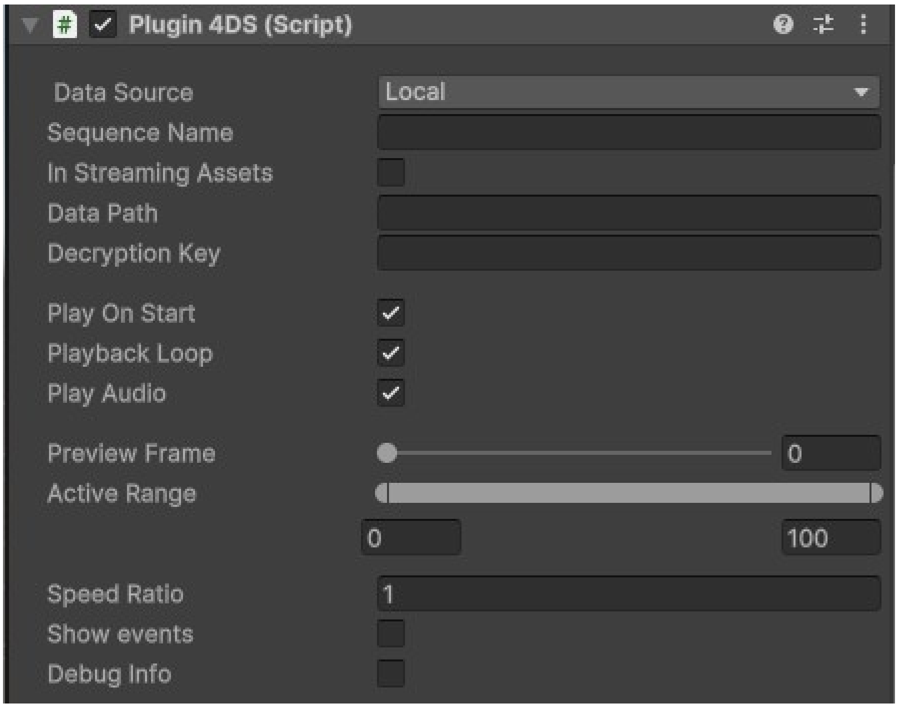
To add a sequence in the scene, go into the Packages and choose 4DS Player > Prefabs > Sequence4DS.

Then drag and drop Sequence4DS in the scene view or in the Hierarchy window.

The 4DS sequence template should appear.



It's now time to specify a sequence to read :



Plugin 4DS (Script)

Data Source: Local
Sequence Name:
In Streaming Assets: □
Data Path:
Decryption Key:

Play On Start: ☑
Playback Loop: ☑
Play Audio: ☑

Preview Frame: 0
Active Range: 0 – 100

Speed Ratio: 1
Show events: □
Debug Info: □

- **Data Source** switches "Local" to "Network" make appear new fields about the server settings.
  1. **To load local 4ds file,** choose "Local" and simply drag'n' drop the file in the "Sequence name" field. If you want to package the data inside your app, it's needed to first copy it in assets/StreamingAssets, then drag'n'drop it from there.
  2. **To load distant 4ds file** from a http server, choose "Network" and set up the full path to the 4ds file, like "http://www.mywebsite.com/path/mysequence.4ds". To enable the storage of the downloaded data in RAM, check "Keep Data in Cache"
- **Decryption Key** : In case the 4ds file has been encrypted using 4Dcoder, set here the decryption key.
- **Play On Start** : if checked, the sequence will start playing automatically with the app.
- **Auto Loop**: if checked, the sequence will loop automatically.
- **Preview frame :** choose the frame displayed as preview in the 3D viewer.
- **Active Range :** select the frame range of the sequence which will be played.
- **Debug Info :** display some information about the sequence in the Game window.

If Network Data Source is chosen, some advanced options are available:

- **Keep Data In Cache :** if checked the downloaded data will be kept in memory. It avoid downloading it again when looping the sequence, but it implies a big RAM consumption.

- **Request Download Size :** size (in MB) of each http request to the server. Depending on the http server and network, it may be useful to change this setting to adjust the balance between big chunks download with less requests and small chunks download with more requests.
- **Cache Max Size :** maximum size of the downloaded data cache. Used only if the Keep data In Cache option is checked.

Deployment

Basicallly the build and deployment on device does'nt require any manipulation.

But some devices may need some specific settings.

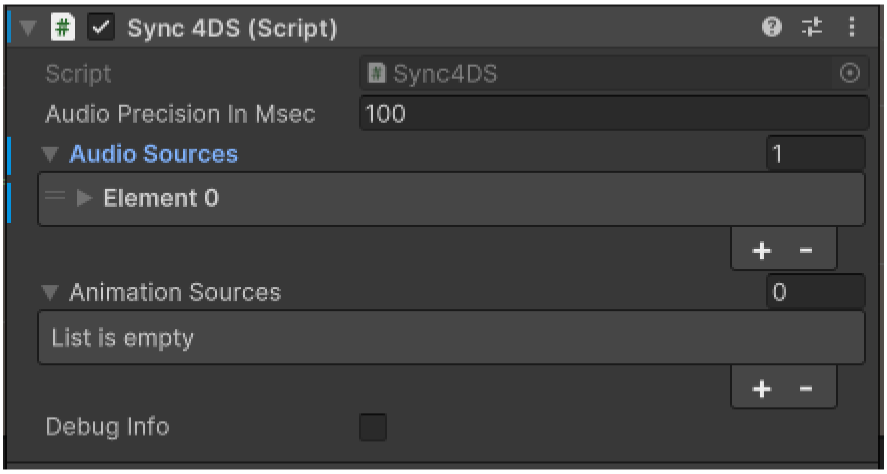
On **VisionOS** :

- Since VisionOS 2.6, new settings have appeared. For a good deployment it is needed to **disable** both of the folowing options in **Project Settings > Polyspatial** :
  - GPU Mesh Transfer
  - GPU Texture Transfer

Go further

FDV Sync

This script is a helper to synchronize audio or animation with the 4D sequence playback.



Sync 4DS (Script)

Script: Sync4DS

Audio Precision In Msec: 100

Audio Sources: 1

Element 0

Animation Sources: 0

List is empty

Debug Info: □

Audio

drag and drop the audio source you want to synchronize in the field.

- **Start on Frame** designs the frame of the 4D sequence you want the audio to start.

- **Audio Precision** is the allowed offset between the sequence and the audio. A value too small can result in a jerky sound. A value too big can result in a delay between what you see and what you hear.

Note : If the 4ds file contains an internal audio track, it is automatically added and synchronized with the 4d sequence.

Animation

drag and drop the animator you want to synchronize in the field.

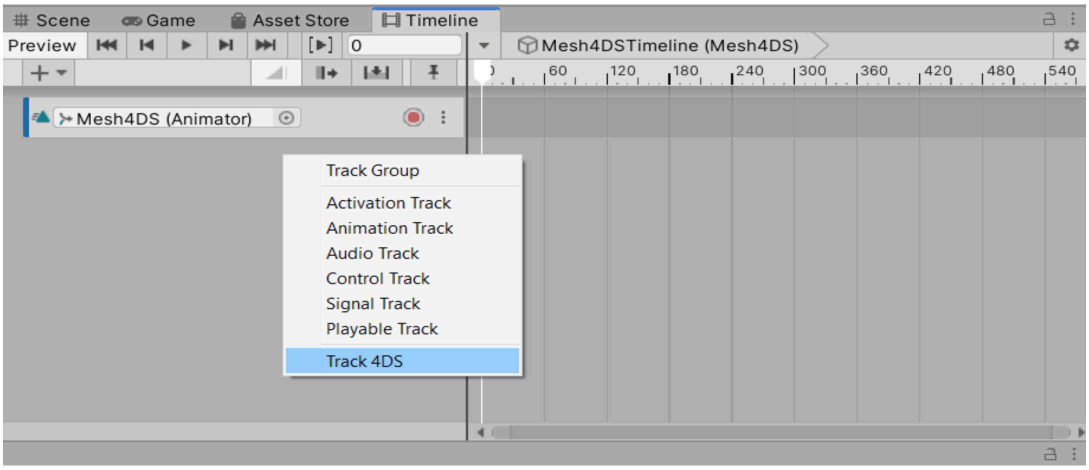
For a good synchronization, it is recommended for the length of the animation to be the same as the 4DS sequence. You can use the FDV Timeline to help in this.

Note : If the 4ds file contains an internal tracking track, it is automatically added and synchronized with the 4d sequence.

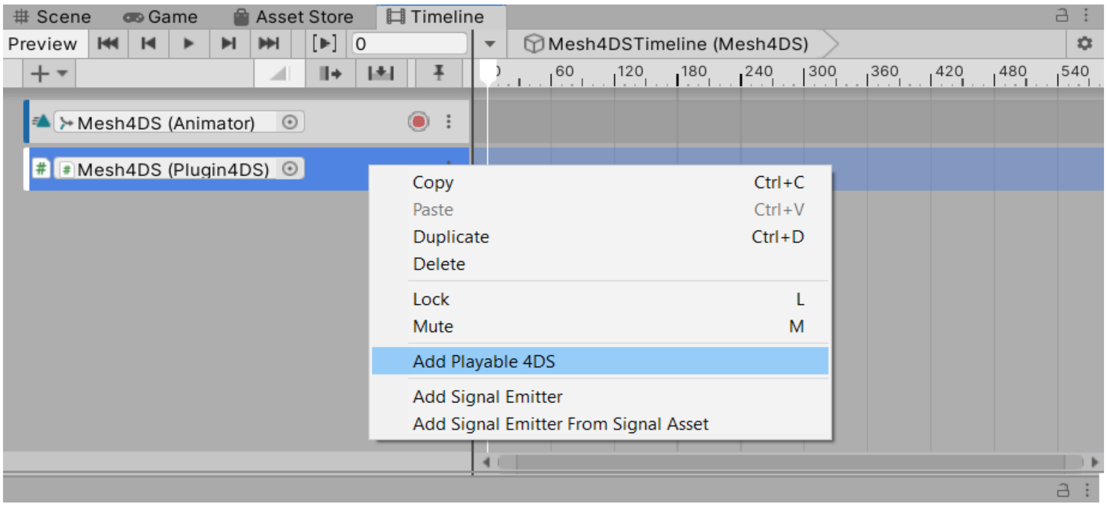
Timeline

The plugin 4DS is compatible with the new Unity Timeline window.

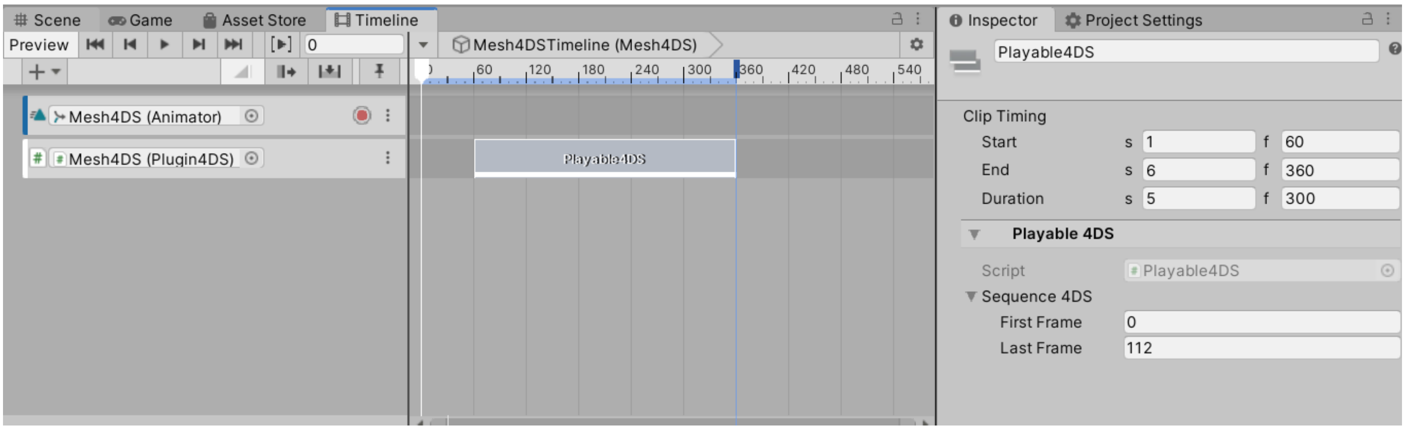
To use it, first create a timeline from the mesh4DS of your sequence.



Then, by right clicking on the left part, choose to create a new Track4DS.



Set the Mesh4DS in the field, and right click on this track to Add a Playable 4DS.



Playable4DS

Clip Timing

Start	s	1	f	60
End	s	6	f	360
Duration	s	5	f	300

Playable 4DS

Script: Playable4DS

Sequence 4DS

First Frame: 0

Last Frame: 112

You now have a clip in the track, corresponding to your sequence.

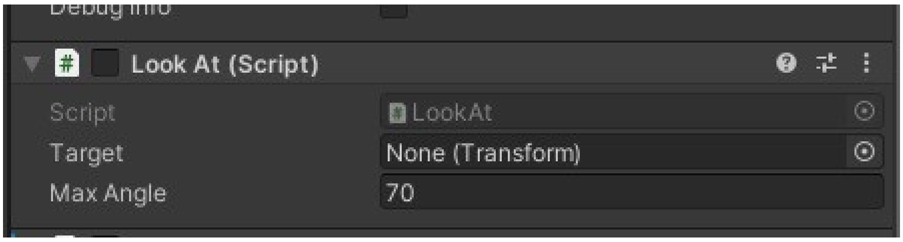
You can then adjust its parameters for the clip Timing, and for the sequence itself.

Look At

The plug-in now integrates the Look At feature, meaning a mesh character can rotate its head targeting an object or the the camera itself to keep its gaze on the user for instance.

WARNING : this feature requires some meta data inside the 4ds file which can only be computed using 4DViews 4Dfx software. Please contact 4Dviews for more information on how to get this software and licensing.

When a 4ds file containing some Look At meta data is imported in Unity, it automatically enable the Look At script in the Mesh4DS inspector.



Debug Info

Look At (Script)

Script: LookAt

Target: None (Transform)

Max Angle: 70

- **Target** specifies the object to be targeted to compute head's rotation.
- **Max Angle** specifies the maximum rotation that can be applied to the head.

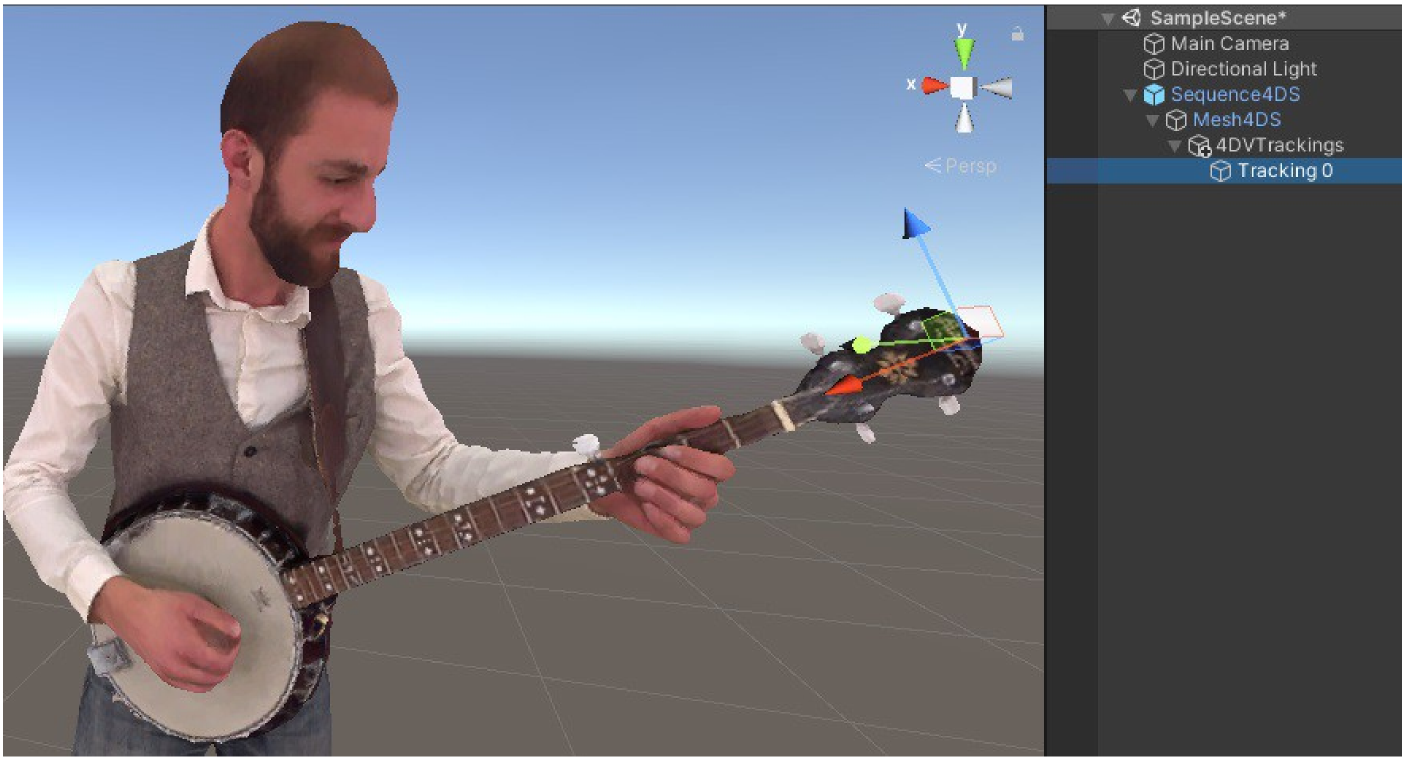
Tracking

If the 4ds file contains some tracking meta data, this will automatically be imported as an animation in Unity.

This animation is separated from the 4ds files itself, but is synchronized with it. It will appear in the hierarchy graph as a child of the Mesh4DS.

You can then add an object as child of the tracking transform, it will follow it's animation,

WARNING : This feature requires some meta data inside the 4ds file which can only be computed using 4Dviews 4Dfx software. Please contact 4Dviews for more information on how to get this software and licensing.

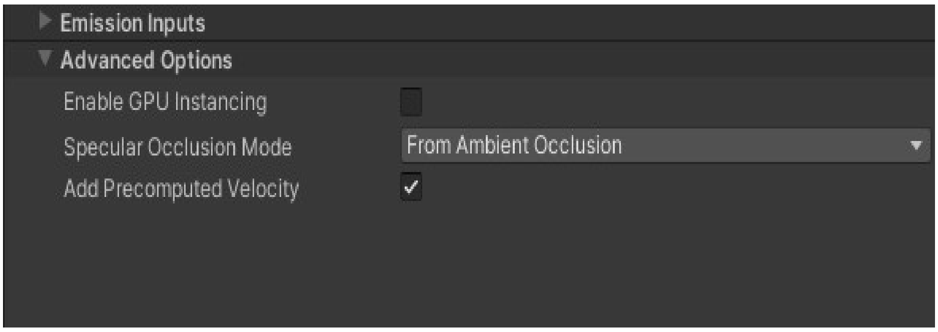


Motion Blur (HDRP)

If the 4ds file contains velocities data, Unity can generate motion blur on the sequence.

For this purpose, you need to :

- Add a Post Process Volume component to the Mesh4DS
- Enable Motion Blur on it
- On the HDRP material of your sequence, you have to check the option '*Add Precomputed Velocities'.*



▶ Emission Inputs

▼ Advanced Options

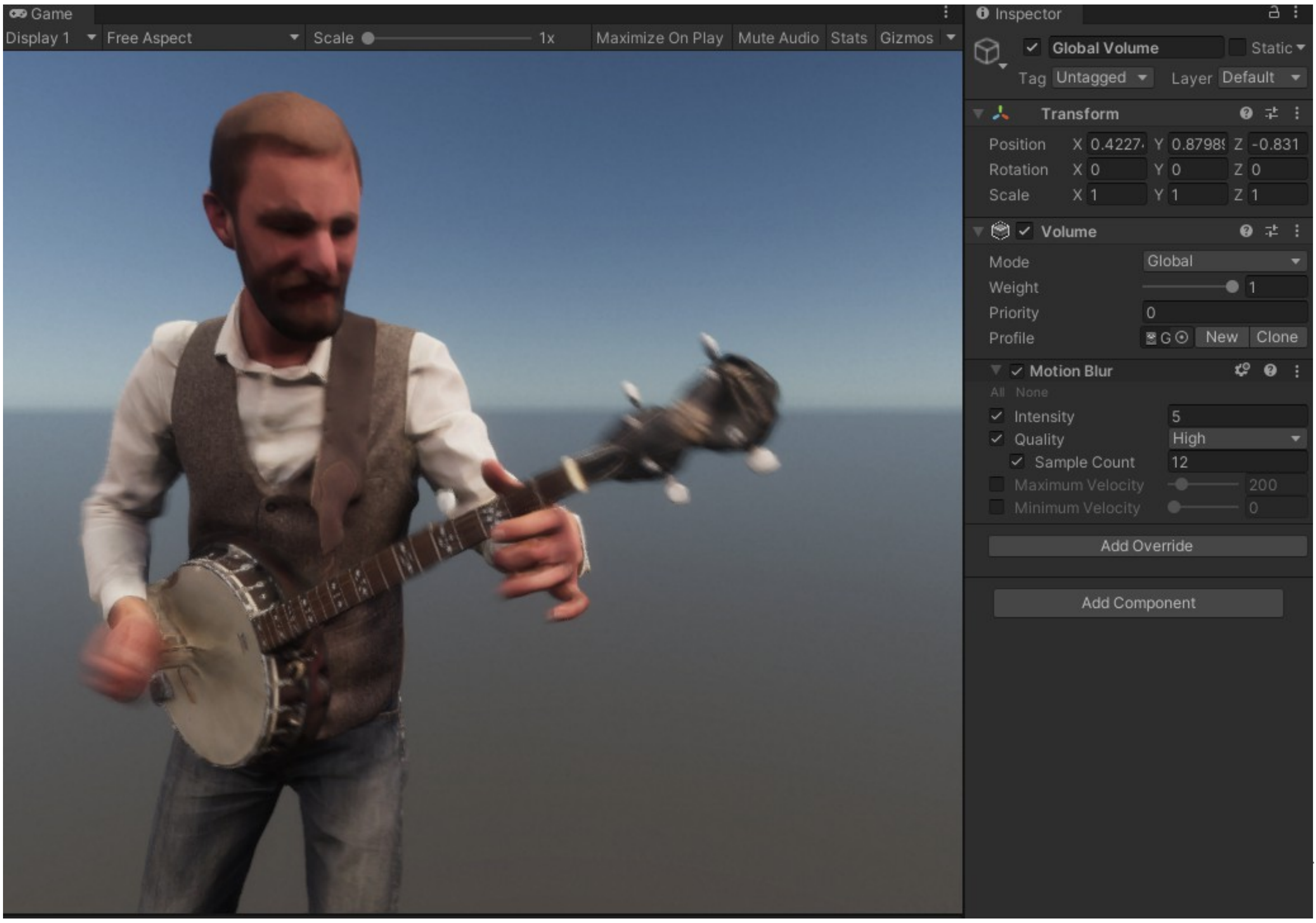
Enable GPU Instancing □

Specular Occlusion Mode: From Ambient Occlusion

Add Precomputed Velocity ☑

WARNING : this works only with **High Definition Render Pipeline (HDRP)**.

WARNING : this feature requires some meta data inside the 4ds file which can only be computed using 4Dviews 4Dfx software. Please contact 4Dviews for information on how to get this software and licensing.



☑ Global Volume □ Static

Tag: Untagged Layer: Default

Transform

Position	X	0.4227	Y	0.8798	Z	-0.831
Rotation	X	0	Y	0	Z	0
Scale	X	1	Y	1	Z	1

☑ Volume

Mode: Global

Weight: 1

Priority: 0

Profile: G New Clone

☑ Motion Blur

All None

☑ Intensity: 5

☑ Quality: High

☑ Sample Count: 12

□ Maximum Velocity: 200

□ Minimum Velocity: 0

Add Override

Add Component

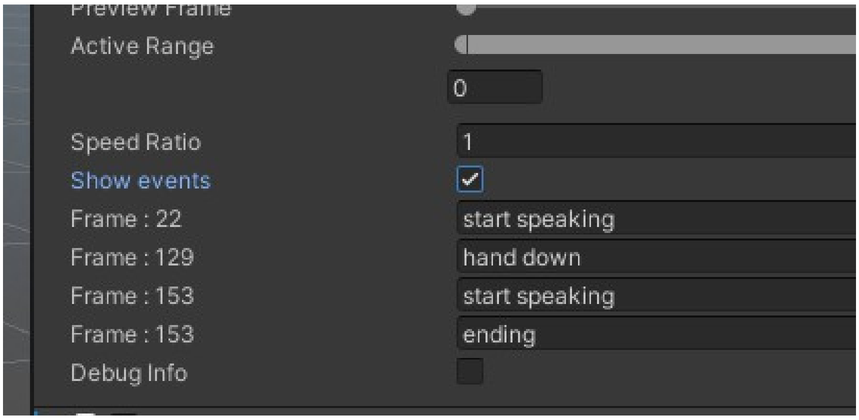
Events

The 4ds file can contain some meta data called Events. These events are key frames set by the user to specify that something happens at this specific time frame.

WARNING : this feature requires meta data inside the 4DS file which can only be computed using 4DViews 4DFX software. Please contact 4DViews for information on how to get this software and licensing.

For every events in the sequence, the plugin will invoke a Unity Event containing the frame number and the name of this event.

You can see the frame and the name of each event in the file by checking the *Show Events* box in the plugin inspector.



Preview Frame

Active Range 0

Speed Ratio 1

Show events ☑

Frame : 22 start speaking

Frame : 129 hand down

Frame : 153 start speaking

Frame : 153 ending

Debug Info □

To handle these events, it is necessay to do some script to add a listener to the **OnUserEvent** of the plugin.

In addition to the OnUserEvent, the plugin will invoke **OnFirstFrame** and **OnLastFrame** events for the first and last frame of the active range.

Here is an example script:

```
void Start()    {
   //Get the Plugin4DS object
   Plugin4DS plugin = gameobject.GetComponent<Plugin4DS>();

   //Add a listener to the OnUserEvent of the Plugin4DS
   plugin.OnUserEvent.AddListener(printEvent);

   //Add a listener to the OnFirstFrame of the Plugin4DS
   plugin.OnUserEvent.AddListener(firstFrameEvent);

   //Tell the Plugin4DS to jump to the frame corresponding to the event "start speaking"
   plugin.GotoFrame("start speaking");
}

//Function called each times OnUserEvent is generated. Arguments are the frame number and the name of the event
public void printEvent(int frame, string name)    {
    Debug.Log("EVENT frame " + frame + "  name = " + name);
}

//Function called each time OnFirstFrame is generated. Argument is the frame number
public void firstFrameEvent(int frame)    {
    Debug.Log("EVENT First frame is " + frame);
}
```

Material

The plug-in uses a fdvMaterial to display the texture of the mesh.

This material is an instantiation of standard Unity shader. It is possible to change it.

By default, this material instantiates a Vertex Lit Legacy Shader. You can Change the shader to modify the material properties.

WARNING: if the shader used does not contains a texture field, the texture will not be displayed on the mesh. The shader's texture field should be named `_BaseMap, _BaseColorMap, or _UnlitColorMap`

It is also possible to write your own shader, as long as it respect Unity's norms and can be applied to a triangle mesh.

Color Replacement Shaders

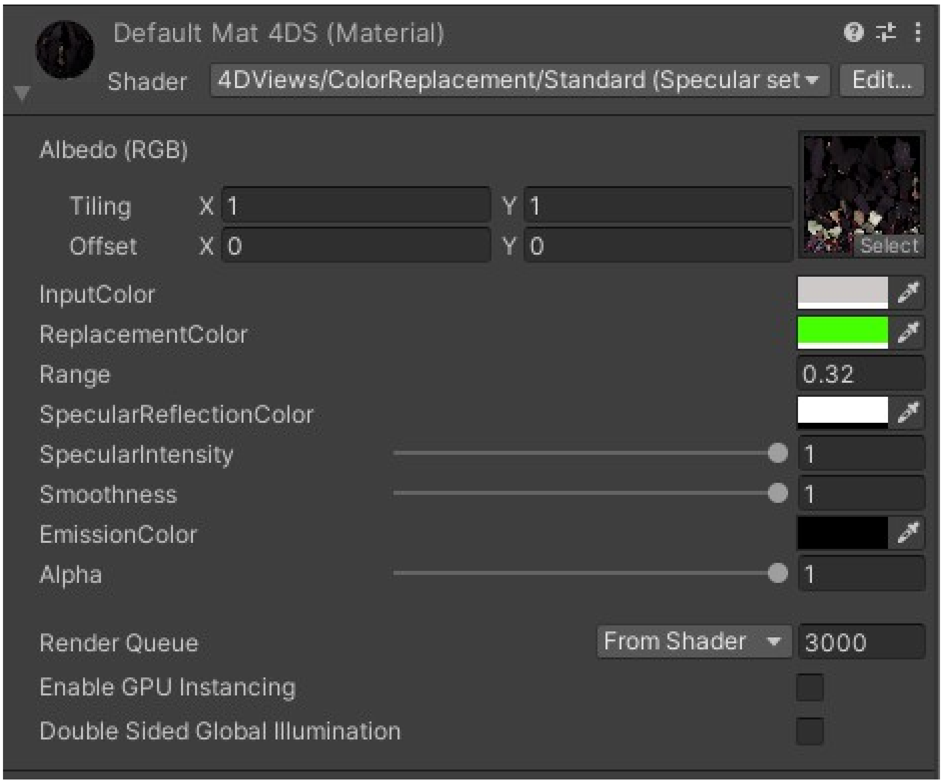
As an example of what is possible with shaders and 4ds data, two shaders are provided that can be affected to the sequence's material:

- Color Replacement Standard
- Color Replacement Specular

These shaders are available under 4DViews/ColorReplacement/ in the shader selection window.

The goal of the shader is to key a certain color of the mesh, and change it to another one.

By adjusting the settings, you can recreate metallic material, or even glass.



Default Mat 4DS (Material)

Shader 4DViews/ColorReplacement/Standard (Specular set ▾ Edit...

Albedo (RGB)

Tiling X 1 Y 1

Offset X 0 Y 0

Select

InputColor ⬜

ReplacementColor 🟩

Range 0.32

SpecularReflectionColor ⬜

SpecularIntensity 1

Smoothness 1

EmissionColor ⬛

Alpha 1

Render Queue From Shader ▾ 3000

Enable GPU Instancing □

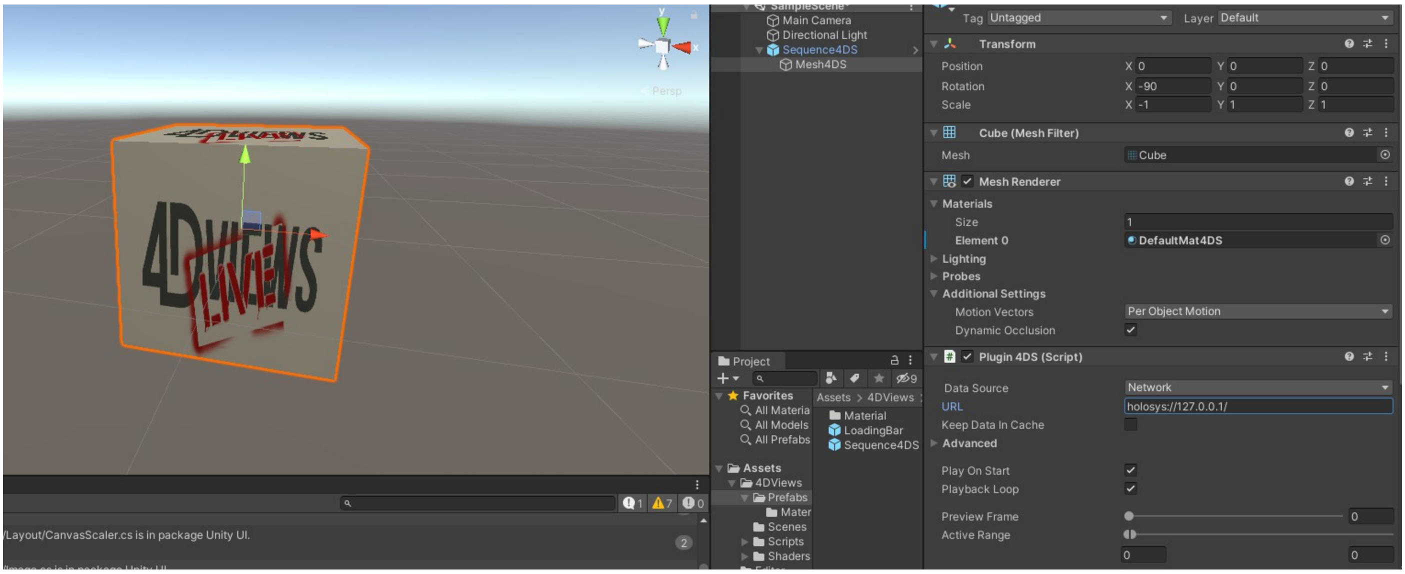
Double Sided Global Illumination □

L!ve

The live mode is only available with a compatible HOLOSYS version. Please contact 4DViews if you want more information about it.

To enable the live client in the Unity plug-in, just do as if you want to open a distant file in the network section. Then as the address put :

– holosys://*yourServerIPaddr*/



/Layout/CanvasScaler.cs is in package Unity UI.

Tag Untagged Layer Default

Transform

Position	X 0	Y 0	Z 0
Rotation	X -90	Y 0	Z 0
Scale	X -1	Y 1	Z 1

Cube (Mesh Filter)

Mesh Cube

☑ Mesh Renderer

Materials
Size 1
Element 0 DefaultMat4DS
Lighting
Probes
Additional Settings
Motion Vectors Per Object Motion
Dynamic Occlusion ☑

☑ Plugin 4DS (Script)

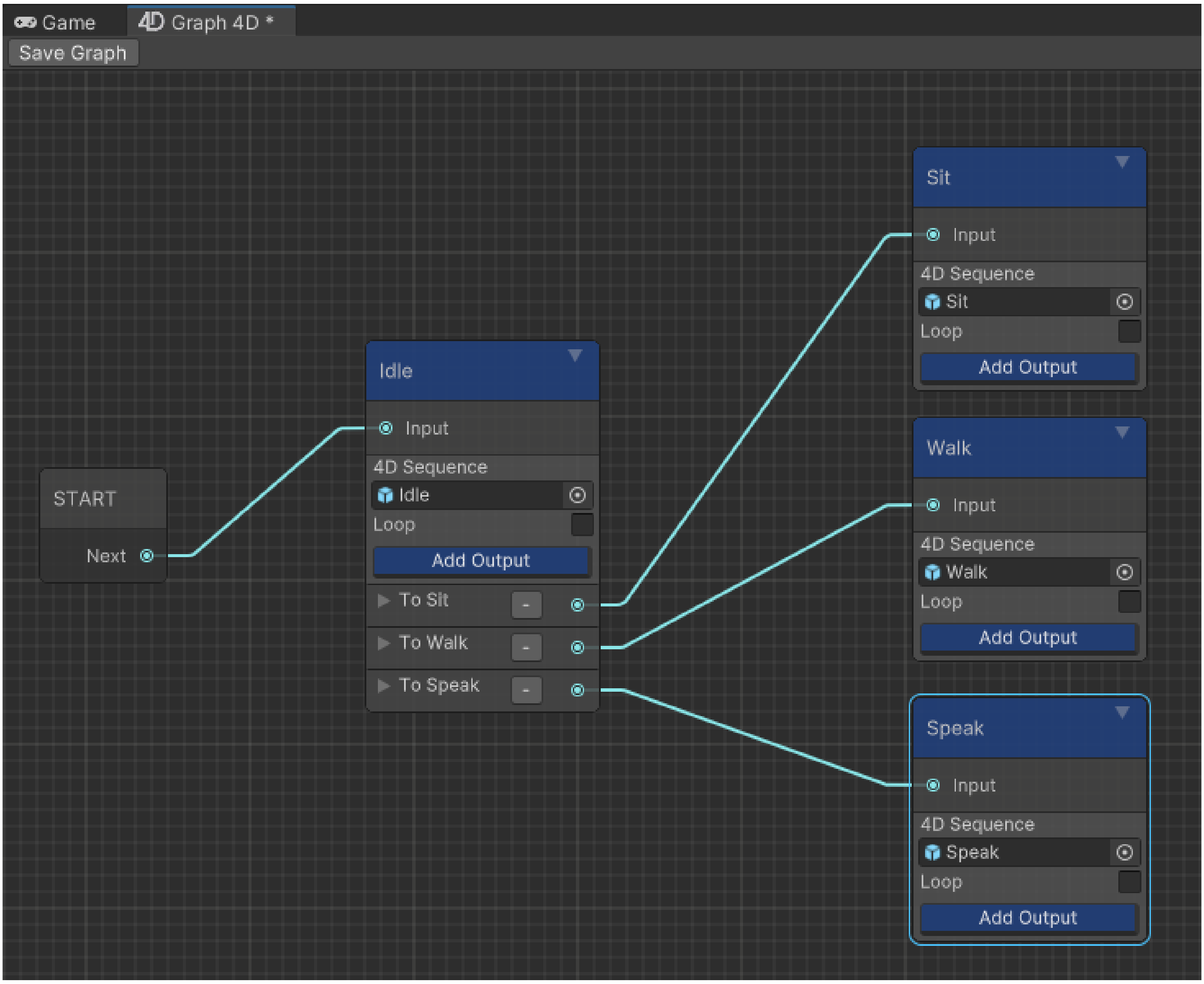
Data Source Network
URL holosys://127.0.0.1/
Keep Data In Cache □
Advanced
Play On Start ☑
Playback Loop ☑
Preview Frame 0
Active Range 0 0

Graph 4D

The Graph4D script allows you to seamlessly play several 4DS files and decide at runtime which file to play thanks to a node editor.

This tool is particularly useful if you want to add some user interaction in your app.

It is available as a script you can put on any object. Once it has been added, click on « edit graph » in the inspector.



In the graph, each node represents a 4D sequence to play. The links between them represent the possible transitions between the nodes.

A right click in the graph editor enables you to create a new node.

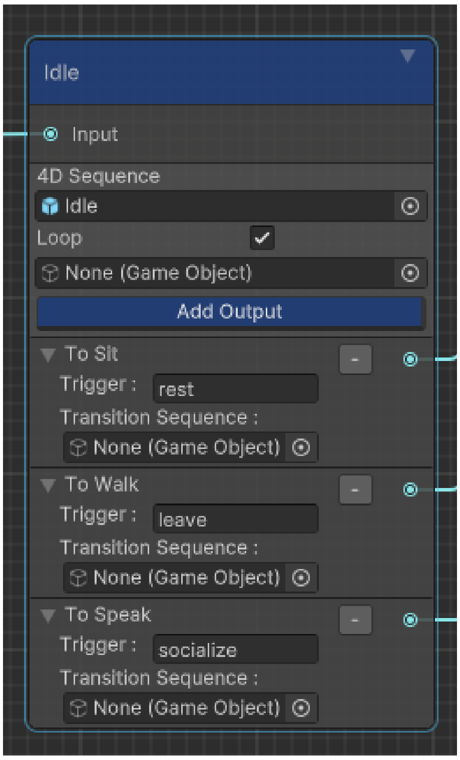
Then, clicking on the « Add Output » button on it, creates a new output connection.

Finally, dragging from one of its outputs to an other node's input creates a new link.

At runtime, the Graph4D plays the sequence associated to the first node, the one linked to START.

Then, depending on which trigger is activated, one of the next node is played. Activating a trigger can only be done by script (see following description).

Inside a node, several options and settings are available :



Idle

Input

4D Sequence: Idle

Loop ☑

None (Game Object)

To Sit
Trigger : rest
Transition Sequence : None (Game Object)

To Walk
Trigger : leave
Transition Sequence : None (Game Object)

To Speak
Trigger : socialize
Transition Sequence : None (Game Object)

- **4D Sequence** : The Mesh4DS to play for this node.
- **Loop** : if checked, the sequence 4D will loop until an output is triggered.
- **[optional] Loop Sequence** : a 4D sequence to play as a transition with the loop (between the end and the new start of the node sequence).

For each output :

- **Trigger** : a name used to trigger the output. It is used as an identifier in script to activate this output. If a trigger doesn't have a name, it is used as the default output (it will be played if no other node transition is triggered)
- **[optional] Transistion Sequence** : a 4D sequence to play between the end of the node's and the begining of the next one.

Group Bloc

Groups blocs enable you to define and name sections of your graph. You can drop nodes inside (and move them out using Shift+click).

They don't have any influence at runtime. They are only here to help making the graph clearer.

WARNING:

- If a node contains several unnamed triggers OR if the node loops and have an unnamed trigger, the behaviour of the graph is undefined.
- It’s not recommanded to use several times the same GameObject inside a Graph. If a 4DS file should appear multiple times in the graph, please create one GameObject per node.

Public API

Parameter :

- *`Play On Start`* : if checked, the Graph will start playing automatically at the app startup.

Method :

- *`void Play()`* : start the Graph execution.
- *`void Stop()`* : stop the Graph execution.
- *`bool IsPlaying()`* : return true if the Graph is playing.
- *`void SetTrigger(string name)`* : set *name* as triggered.
- *`void UnsetTrigger(string name)`* : *set name* as untriggered.

Events :

- *`OnGraph4DStart / OnGraph4DStop`* : launched at the start or stop of the Graph.
- `OnSequence4DStart / OnSequence4DStop` : launched at the Sequence start or stop.