SCPlayerPro

Contact us:http://www.sttplay.com

Content

Thank you

What can we do?

As the application field of Unity becomes wider and wider, it is also a very friendly thing for the multimedia industry. However, many video files need to be demonstrated, and some plugins cannot meet the needs of customers, Therefore, we have developed an interesting plug-in specifically for this situation, namely SCPlayerPro. We hope to provide some help to friends who are engaged in the development of the multimedia industry. At the same time, we will continue to improve our own capabilities.

Like most media players, we deal with audio and video playback. The difference with other playback plug-ins is that we will be compatible with most video formats and some non-popular pixel formats. The current release version is suitable for windows and android platforms, supporting local files and network files. We will continue to maintain and update to support media types and cross-platform versatility. We will support common external devices such as cameras, virtual cameras, and microphones, of course, rtsp, rtmp, hls and other online streams are indispensable. We will also make adjustments based on user feedback, and we will strive to give users the most satisfactory answer.

On this basis, we packaged it, that is, the core decoding plug-in SCCore.dll. In terms of video playback, we did not use DX9 and DX11 for direct video rendering, but used Texture2D in Unity for intuitive rendering. Considering the needs of relatively special users, users may want to obtain the original data of video frames, so they must get the data of each frame to the CPU. Although this will affect certain performance, it can better meet the needs of users. These effects are negligible. In terms of audio playback, we have also encapsulated, and the same factors as video, we can also obtain the original PCM audio data. Users can easily obtain the original data after audio and video decoding for secondary use.

Features

- **Support most audio and video files**

  Support video file format: mp4, flv, mov, rmvb, wmv, avi, rm, rmvb, 3gp, 3g2, mpg, mpeg... ...

Support audio file format: mp3, flac, wma, wav, aac… …

- **It is friendly to support network streaming**

  The current version supports rtsp, rtmp, hls, http, https… …

- **It is friendly to support camera**

  In addition to supporting traditional USB cameras, virtual cameras are also supported

- **It is friendly to the video support of different resolutions**

  The image can be parsed normally if the image is large or small, and the image will not be abnormal due to memory alignment

- **pixel formats that are not supported by the plug-in are detected and automatically converted to BGRA pixel format**

  The plug-in is not omnipotent. When it encounters an unsupported pixel format, it will automatically convert to the BGRA pixel format, which greatly increases the compatibility of the pixel format.

- **Support multiple pixel formats, and uniformly set the memory to be aligned to 1 byte**

  Output pixel format support YUV420P, YV422P, YUV444P, YUYV422, UYVY422, NV12, NV21, RGB, BGR, RGBA, BGRA, ARGB, ABGR, GRAY. and uniformly set the memory to be aligned to 1 byte.

- **The user can manually specify the output pixel format to ensure normal use in special occasions**

  Users can set according to their needs.

- **Supports hardware acceleration of multiple device types, and users can specify the device types for hardware acceleration**

  Users can set according to their needs.

- **Friendly and compatible with videos containing transparent channels**

  Many player plug-ins do not support, we also solved this pain point

- **Manually or enable audio and video tracks**

  Users may not need all streams, we also provide to enable or disable audio and video streams

- **Render to Rawimage or mesh**

  It is easy to draw on mesh or ugui

Bnefit

Developers can easily obtain the original audio and video data and perform secondary processing, and can also easily and simply realize the playback of ordinary videos or special videos, you can simply control the opening, closing, and replay of the video, which is very friendly to developers who often control video state switching. We have encapsulated UnitySCPlayerPro, users do not need to care about internal processing and data exchange, users only need to simply call the interface function to achieve control over UnitySCPlayerPro. We can also process large-resolution videos efficiently. We can use hardware accelerated video decoding, and we can also change the output pixel format according to our own needs. We can friendly support the smooth playback of 8K videos.

Support resolution

480p 60 fps

720p 60 fps

1080p 60 fps

4K 60 fps

8K 30 fps

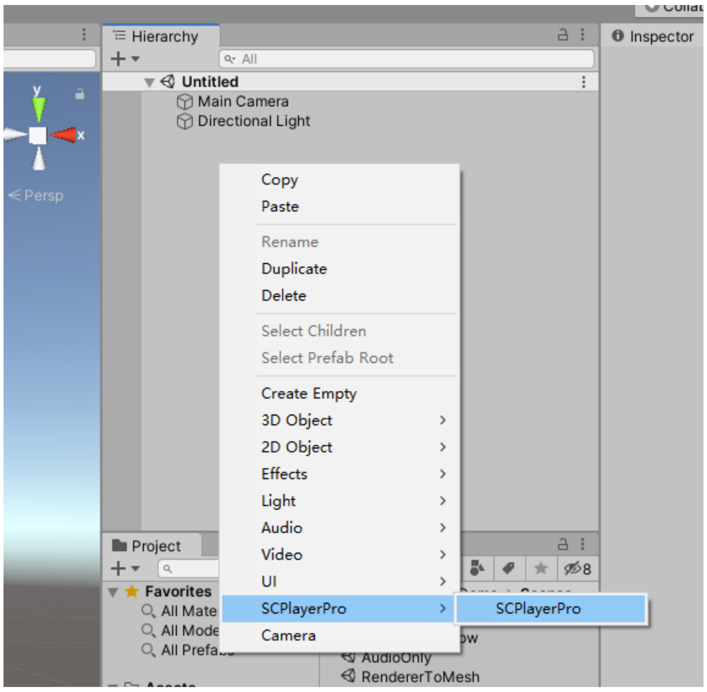
The above indicator is a range value, which can also be well supported in these ranges

Technical specifications

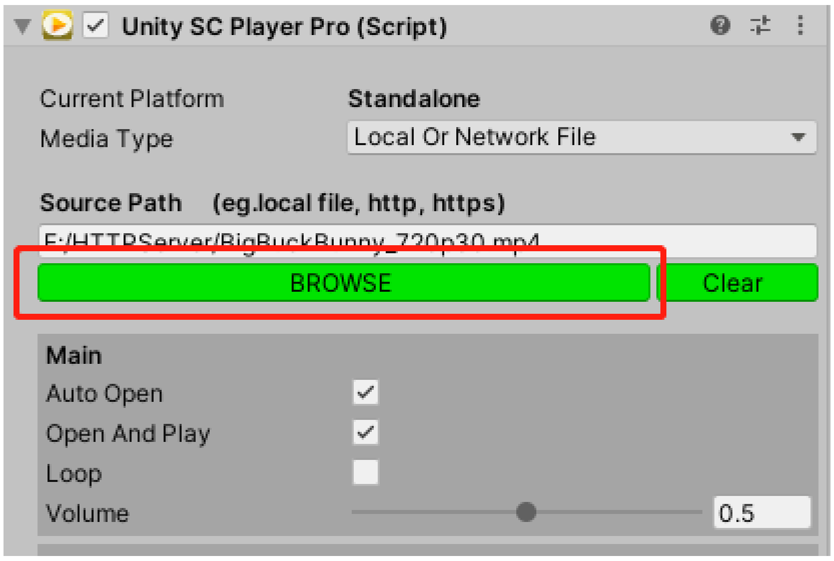
The operating environment of the plug-in is based on Unity2019.4.x, We have no special requirements for the .Net version. The plug-in uses a 3D model 360vr.fbx, this model is used to play panoramic video, we have inverted its normal and UV, if the user needs it can be used directly.

Simple start

1.Right-click on the blank of the “Hierarchy” panel, select SCPlayerPro and then select the SCPlayerPro option to create a player.



2. Select the SCPlayerPro object, select the BROWSE button on the SCPlayerPro component to select a media file


Unity SC Player Pro (Script)

Current Platform Standalone
Media Type Local Or Network File

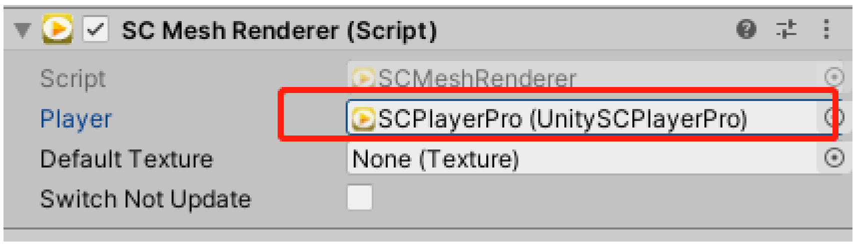
Source Path (eg.local file, http, https)
F:/HTTPServer/BigBuckBunny_720p30.mp4

Main
Auto Open ✓
Open And Play ✓
Loop
Volume 0.5

3. We also need a render object, here we create a cube

4.Select the cube to add a SCMeshRenderer component

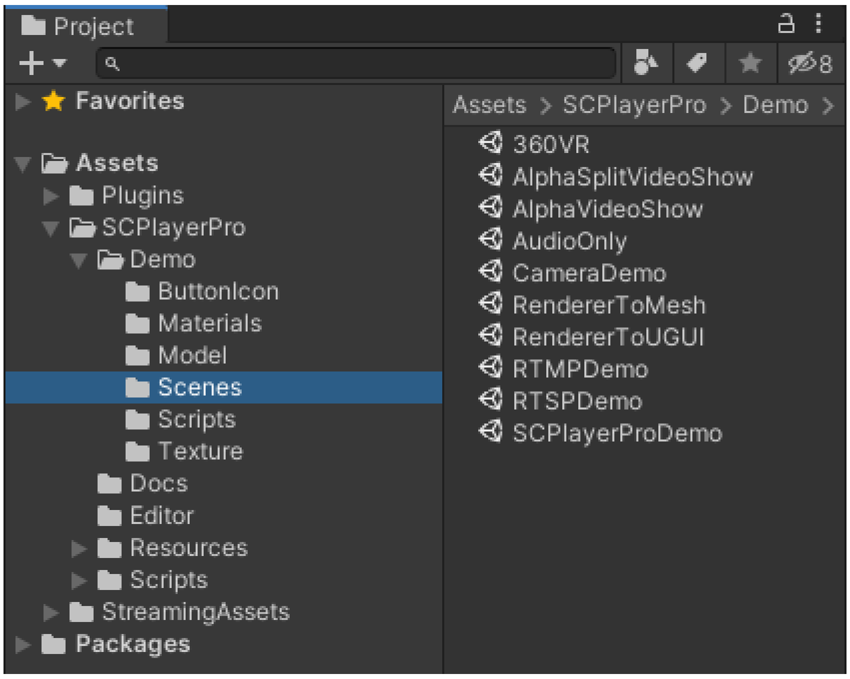
5. Drag the previously created SCPlayerPro to the player of SCMeshRenderer


SC Mesh Renderer (Script)

Script: SCMeshRenderer
Player: SCPlayerPro (UnitySCPlayerPro)
Default Texture: None (Texture)
Switch Not Update: ☐

6.OK, Let's click the play button to test

Demo scenes



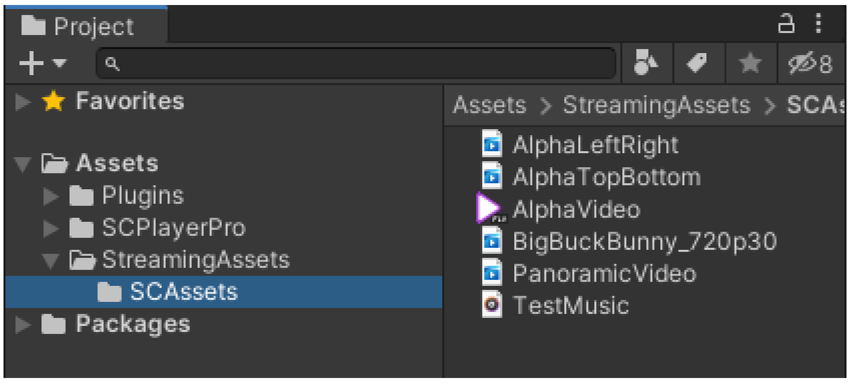
360VR
AlphaSplitVideoShow
AlphaVideoShow
AudioOnly
CameraDemo
RendererToMesh
RendererToUGUI
RTMPDemo
RTSPDemo
SCPlayerProDemo

Here are some example cases for reference

Here we did not upload the StreamingAssets folder, because the video file is relatively large, here is a screenshot of the catalog. Currently, video files in the StreamingAssets directory are used in many demos. Before using, you need to choose the correct file path according to your own situation.

Note that the SCAssets folder is a special folder, and the file path inside can be called by SCMGR.GetUrlFromSCSCAssets(), which is common on the current platform.

For example SCMGR.GetUrlFromSCSCAssets("AlphaLeftRight.mp4");



- **SCPlayerProDemo**

  A complete player, most of the functions can be reflected in this player. (Click the gear button in the upper right corner to set various parameters in the ui interface)

- **RendererToUGUI**

  We can draw the video on the UGUI canvas through this scene, of

course, the canvas is based on Rawimage

- **RendererToMesh**

  We can draw a video on the 3D model through this scene

- **AudioOnly**

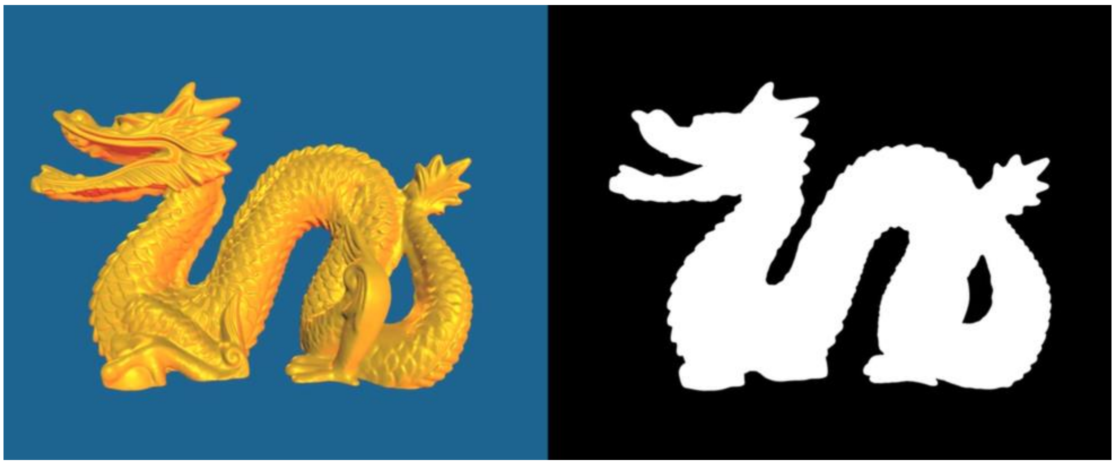
  This scene played a wonderful piece of music

- **AlphaVideoShow**

  This scene shows the playback of transparent videos that most players do not support

  AlphaSplitVideoShow

  This scene shows the playback of transparent videos，however, this type of video does not contain transparent channels, but uses RGB color values to represent transparent channel data through a certain part of the image.



For example, this kind of video, the left is the normal video, the right is the transparent channel data.

- **360VR**

  Play the panoramic video and draw it on a special sphere

(In addition to SCPlayerProDemo, other demos can use SampleController gameObject to set the file path and whether to use the StreamingAssets folder.)

- **RTSPDemo/RTMPDemo**

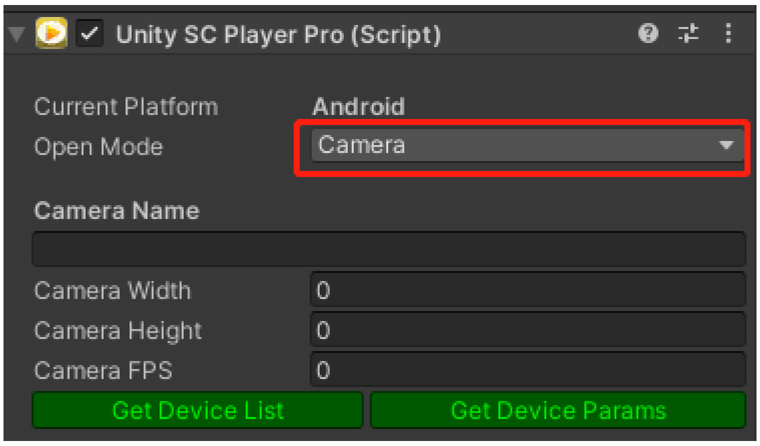
  This scene shows how to turn on a network video stream.

- **CameraDemo**

  This scene shows how to turn on the camera.

(Need to be executed in run mode)

1. Set OpenMode to Camera.



Unity SC Player Pro (Script)

Current Platform Android

Open Mode Camera

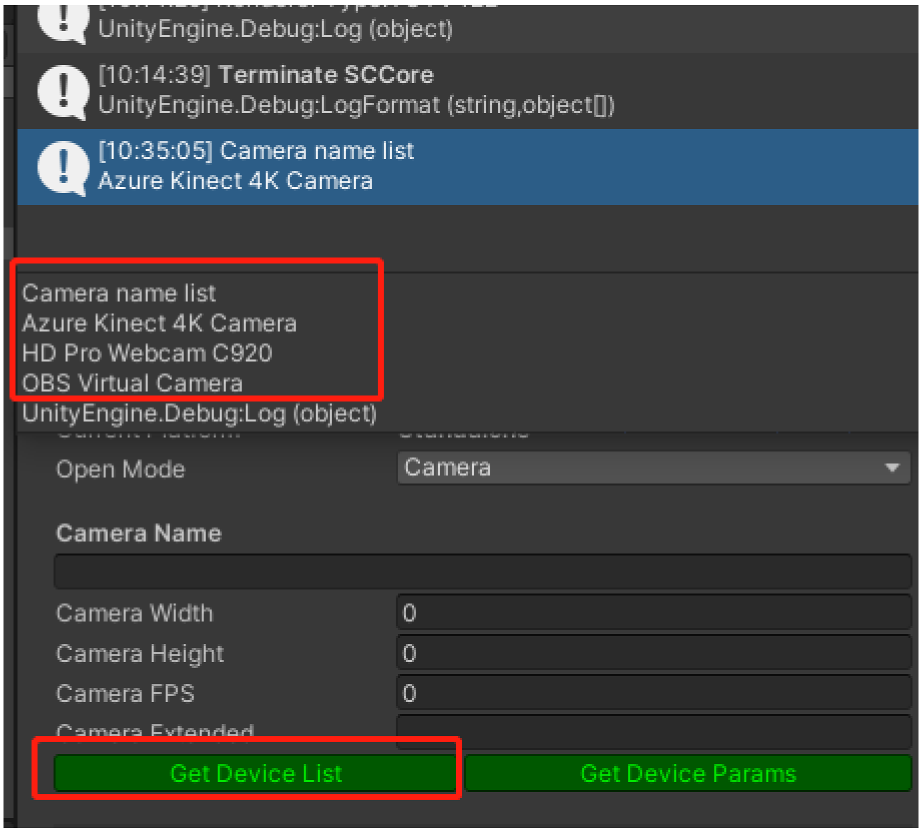
Camera Name

Camera Width 0

Camera Height 0

Camera FPS 0

2. Clicking on GetDeviceList to get all camera names on the current device (including virtual cameras)



```
UnityEngine.Debug:Log (object)
[10:14:39] Terminate SCCore
UnityEngine.Debug:LogFormat (string,object[])
[10:35:05] Camera name list
Azure Kinect 4K Camera

Camera name list
Azure Kinect 4K Camera
HD Pro Webcam C920
OBS Virtual Camera
UnityEngine.Debug:Log (object)
```

Open Mode: Camera

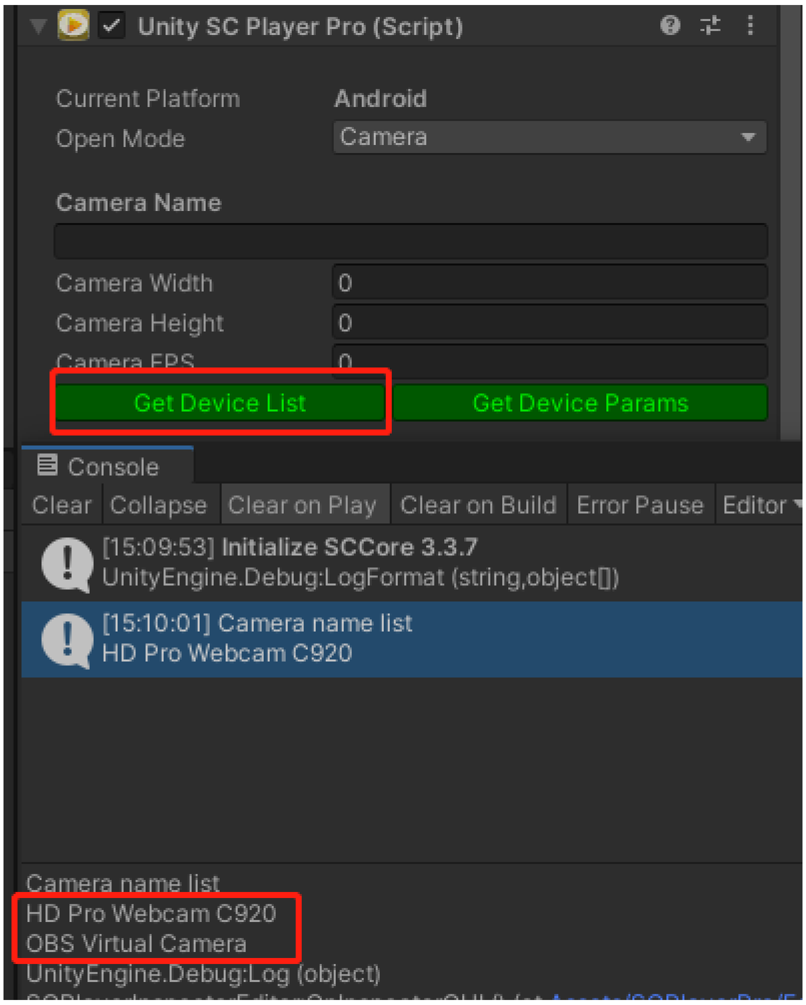
Camera Name:

Camera Width: 0

Camera Height: 0

Camera FPS: 0

Camera Extended:



Current Platform: Android

Open Mode: Camera

Camera Name:

Camera Width: 0

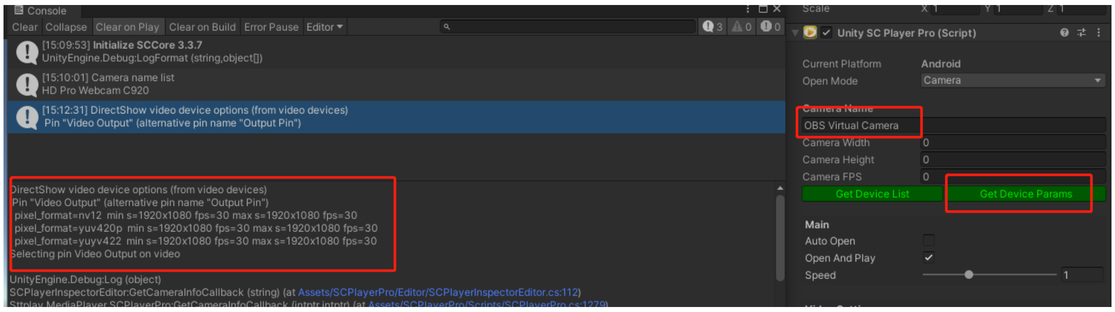
Camera Height: 0

Camera FPS: 0

```
[15:09:53] Initialize SCCore 3.3.7
UnityEngine.Debug:LogFormat (string,object[])
[15:10:01] Camera name list
HD Pro Webcam C920

Camera name list
HD Pro Webcam C920
OBS Virtual Camera
UnityEngine.Debug:Log (object)
```

3. Here I got three camera names, and now I just need to copy the camera name into the CameraName input box to use it.

4. About the Camera Options parameter.

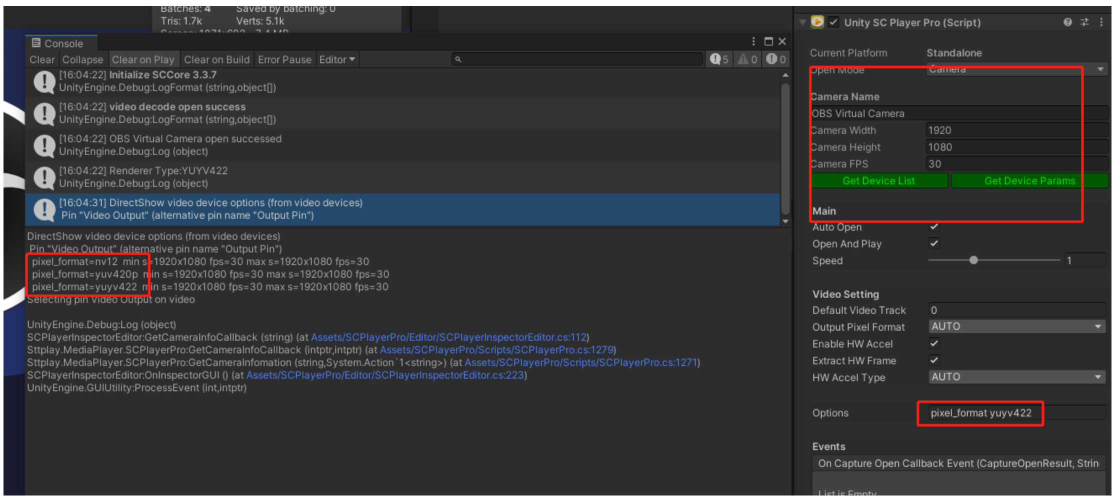


```
[15:09:53] Initialize SCCore 3.3.7
UnityEngine.Debug:LogFormat (string,object[])
[15:10:01] Camera name list
HD Pro Webcam C920
[15:12:31] DirectShow video device options (from video devices)
 Pin "Video Output" (alternative pin name "Output Pin")

DirectShow video device options (from video devices)
Pin "Video Output" (alternative pin name "Output Pin")
 pixel_format=nv12  min s=1920x1080 fps=30 max s=1920x1080 fps=30
 pixel_format=yuv420p  min s=1920x1080 fps=30 max s=1920x1080 fps=30
 pixel_format=yuyv422  min s=1920x1080 fps=30 max s=1920x1080 fps=30
Selecting pin Video Output on video

UnityEngine.Debug:Log (object)
SCPlayerInspectorEditor:GetCameraInfoCallback (string) (at Assets/SCPlayerPro/Editor/SCPlayerInspectorEditor.cs:112)
```

Unity SC Player Pro (Script)

Current Platform	Android
Open Mode	Camera
Camera Name	OBS Virtual Camera
Camera Width	0
Camera Height	0
Camera FPS	0

After entering the camera name, you can click the GetDeviceParams button to get more parameters for the camera. This list allows the camera to output a specific format by setting the relevant parameters.



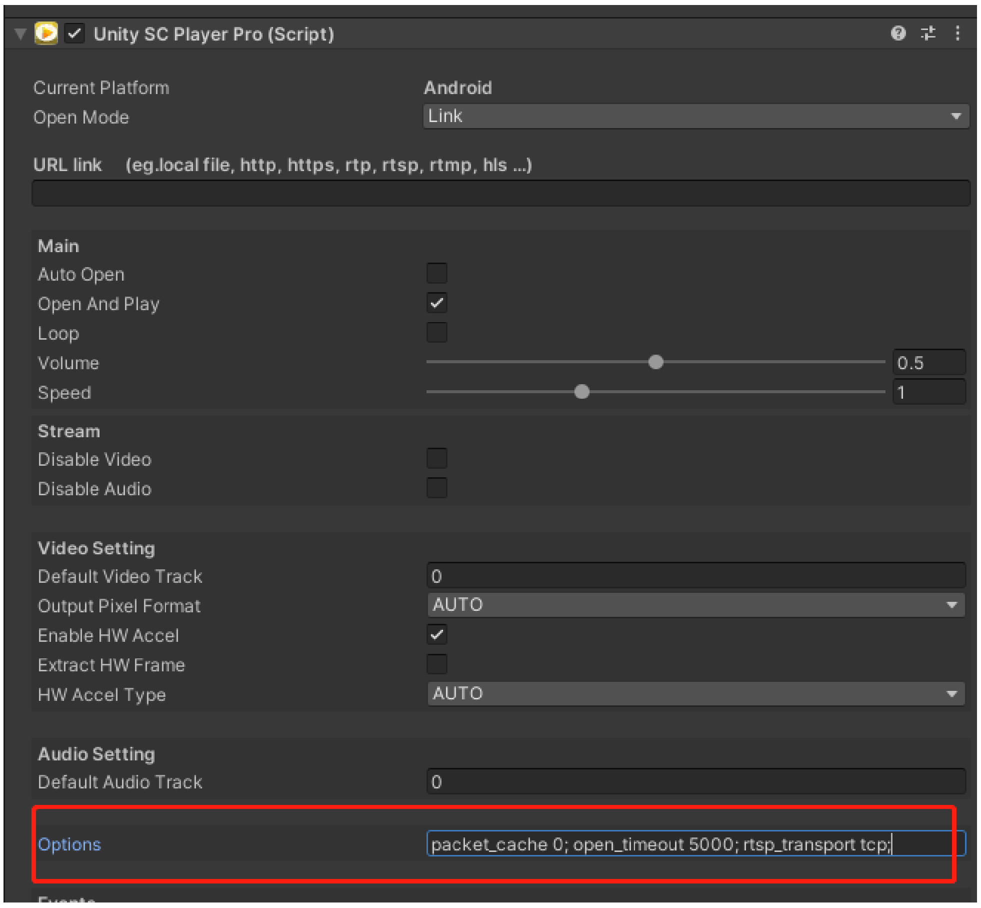
```
[16:04:22] Initialize SCCore 3.3.7
UnityEngine.Debug:LogFormat (string,object[])
[16:04:22] video decode open success
UnityEngine.Debug:LogFormat (string,object[])
[16:04:22] OBS Virtual Camera open successed
UnityEngine.Debug:Log (object)
[16:04:22] Renderer Type:YUYV422
UnityEngine.Debug:Log (object)
[16:04:31] DirectShow video device options (from video devices)
 Pin "Video Output" (alternative pin name "Output Pin")

DirectShow video device options (from video devices)
 Pin "Video Output" (alternative pin name "Output Pin")
  pixel_format=nv12  min s=1920x1080 fps=30 max s=1920x1080 fps=30
  pixel_format=yuv420p  min s=1920x1080 fps=30 max s=1920x1080 fps=30
  pixel_format=yuyv422  min s=1920x1080 fps=30 max s=1920x1080 fps=30
Selecting pin Video Output on video

UnityEngine.Debug:Log (object)
SCPlayerInspectorEditor:GetCameraInfoCallback (string) (at Assets/SCPlayerPro/Editor/SCPlayerInspectorEditor.cs:112)
Sttplay.MediaPlayer.SCPlayerPro:GetCameraInfoCallback (intptr,intptr) (at Assets/SCPlayerPro/Scripts/SCPlayerPro.cs:1279)
Sttplay.MediaPlayer.SCPlayerPro:GetCameraInfomation (string,System.Action`1<string>) (at Assets/SCPlayerPro/Scripts/SCPlayerPro.cs:1271)
SCPlayerInspectorEditor:OnInspectorGUI () (at Assets/SCPlayerPro/Editor/SCPlayerInspectorEditor.cs:223)
UnityEngine.GUIUtility:ProcessEvent (int,intptr)
```

Unity SC Player Pro (Script)

Current Platform	Standalone
Open Mode	Camera
Camera Name	
OBS Virtual Camera	
Camera Width	1920
Camera Height	1080
Camera FPS	30
Main	
Auto Open	✓
Open And Play	✓
Speed	1
Video Setting	
Default Video Track	0
Output Pixel Format	AUTO
Enable HW Accel	✓
Extract HW Frame	✓
HW Accel Type	AUTO
Options	pixel_format yuyv422
Events	
On Capture Open Callback Event (CaptureOpenResult, Strin	

UnitySCPlayerPro options

packet_cache 10;	Keep the buffer frame at around 10 frames
open_timeout 5000;	Set the network stream opening timeout to 5 seconds
fflags nobuffer;	Refer to the fflags parameter in ffmpeg
lowdelay;	Decoder low latency mode
rtsp_transport tcp;	Set the transmission protocol of rtsp stream



Unity SC Player Pro (Script)

Current Platform	Android
Open Mode	Link
URL link (eg.local file, http, https, rtp, rtsp, rtmp, hls ...)	
Main	
Auto Open	
Open And Play	✓
Loop	
Volume	0.5
Speed	1
Stream	
Disable Video	
Disable Audio	
Video Setting	
Default Video Track	0
Output Pixel Format	AUTO
Enable HW Accel	✓
Extract HW Frame	
HW Accel Type	AUTO
Audio Setting	
Default Audio Track	0
Options	packet_cache 0; open_timeout 5000; rtsp_transport tcp;

Script References

Our core program is UnitySCPlayerPro, here is the detailed instructions

UnitySCPlayerPro.cs

Important fields or attribute

mediaType: File type to open, If you choose LocalOrNetworkFile, then the supported files are local files, network files, For example, file: ///, http, https, but not hls stream

Closed: Mark whether player is closed, open failure is also considered not close

OpenSuccessed: Mark whether player successfully opened media

url: URL When the choice of mediatype is different, url has different meanings, for specific support, please refer to mediaType

disableVideo: Whether to disable video

disableAudio: Whether to disable audio

enableHWAccel: Whether to enable hardware acceleration , not all videos support hardware acceleration. If you enable this option, hardware acceleration will be tried first, and if it fails, the CPU will be used for decoding.

HWAccelType: Hardware device type when video hardware accelerates decoding, Not all of the current platforms are supported, if the current option does not support, set as the default.

outputFmt: Pixel format of output SCFrame

autoOpen: Whether to open the media when UnityPlayer starts

openAndPlay: Play directly after opening or stay at the first frame

loop: Whether the media is played in a loop, This option is valid only when the mediaType is LocalOrNetworkFile

volume:Media volume

IsPaused: Whether the marker is in a paused state

onOpenEvent: Called when opening

onCloseEvent: Called when closing

onCaptureOpenCallbackEvent: Called when StreamCapture demux succeeds or failed

onFirstFrameRenderEvent: Called after the first frame is drawn, if there is no video stream, this event will not be called

onRendererFrameEvent: Called after each frame of video is drawn , if there is no video stream

onStreamFinishedEvent: Called when the video has finished playing, whether looping or not

CurrentTime: Current playback timestamp, valid when the mediaType is LocalOrNetFile

Duration: The total duration of the media, valid when the mediaType is LocalOrNetFile

VideoRenderer: Video renderer

Important method

Open: Open the media file by changing the method, You can pass a url, if you don't pass it, use an default url

Close: No matter what state SCPlayerPro is in, it will be closed

Pause:Pause the video

Play:Play the video

Replay: Replay from the beginning, you can also specify whether to pause

SeekFastPercent: Seek video according to percentage, seek to key frame

SeekFastMilliSecond: Seek video according to millisecond, seek to key frame

ReleaseCore: Release all resources of player, After the release is complete, the object can no longer be used, and the user does not need to manage

SCUGUIRenderer.cs

Important fields or attribute

player: Need to choose a for video drawing

defaultTexture: You can set a default image to RawImage

switchNotUpdate: The screen does not refresh when player is closed or when the video source is switched

SCMeshRenderer.cs

Important fields or attribute

player: Need to choose a for video drawing

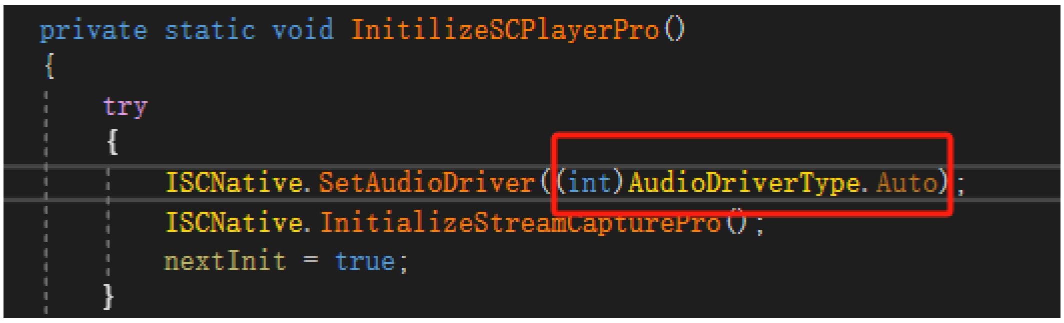
defaultTexture: You can set a default image to RawImage

switchNotUpdate: The screen does not refresh when SCPlayerPro is closed or when the video source is switched.

There are not many parts that we need to manually control. This is also our major feature. It is simple and easy to use. The above script description can satisfy our playback and display of any video.

Important Notice

1.On some machines, there may be no audio output issues via AudioDriverType.Auto initialization, which is related to the settings of the computer's audio output device. Of course, you can also try to change the initialization method by changing the AudioDriverType in function InitilizeSCPlayerPro of Assets\SCPlayerPro\Scripts\SCMGR.cs, here you can choose DirectSound or Winmm.



```
private static void InitilizeSCPlayerPro()
{
    try
    {
        ISCNative.SetAudioDriver((int)AudioDriverType.Auto);
        ISCNative.InitializeStreamCapturePro();
        nextInit = true;
    }
```

Thanks