

Video Codec SDK



1. Description

Video Codec SDK supports

UnityEditor,Android,iOS,Windows,UWP(Hololens1&2),MR/AR/VR Glasses,Mac,Linux.

Multiple video types:device camera/unity camera/custom texture.

The interface is simple, the video/audio capture and codec are separated, you can insert the sdk to your own network easily.

Support Email:1786570525@qq.com

2.Project Setting

2.1 Player Settings -> Other Settings -> Configuration

->Scripting Runtime Version ->. Net 4.x . Mono

2.2 Create a project with Unity 2019.4 and above.

2.3 Edit-> ProjectSettings -> Audio ->DSP -> Good Latency

3.Integrate the SDK into your own project

You can intergrate the sdk in your own network. The SDK provides

video/audio capture, codec.

4. Main Interface

```
/// <summary>
/// Initialize audio
/// </summary>
void InitMic(int index = 0);
/// <summary>
/// Initialize video
/// </summary>
void InitVideo(int index = 0);
/// <summary>
/// Set the type of video capture
/// </summary>
/// <param name="type">type of video capture</param>
/// <param name="captureCamera">If the capture type is unitycamera,
you need to identify a unitycamera.</param>
bool SetVideoCaptureType(VideoType type, Camera captureCamera =
null);
/// <summary>
/// Set the resolution of the video
/// </summary>
/// <param name="resolution">resolution of the video</param>
void SetResolution(VideoResolution resolution);
/// <summary>
/// Get the size of the current video resolution
/// </summary>
/// <returns></returns>
Vector2 GetResolutionSize();
/// <summary>
/// Set the compression quality of the video
/// </summary>
/// <param name="quality">quality of the video</param>
void SetVideoQuality(VideoQuality quality);
/// <summary>
/// Start capturing audio and video
/// </summary>
/// <returns>Capture results</returns>
CaptureResult StartCapture();
/// <summary>
/// Stop capturing audio and video
/// </summary>
```

```

void StopCpture();
/// <summary>
/// Get the current audio package
/// </summary>
/// <returns></returns>
AudioPacket GetAudio();
/// <summary>
/// Get the current video package
/// </summary>
/// <returns></returns>
VideoPacket GetVideo();
/// <summary>
/// Decode the audio of the peer
/// </summary>
/// <param name="id">peer id</param>
/// <param name="packet">peer audioPacket</param>
void DecodeAudioData(AudioPacket packet);
float[] DecodeAudioFloatData(AudioPacket packet);
/// <summary>
/// Decode the video of the peer
/// </summary>
/// <param name="packet">peer videoPacket</param>
Texture2D DecodeVideoData(VideoPacket packet);
/// <summary>
/// Send your customTexture
/// </summary>
/// <param name="tex">the Texture2D to be sent</param>
void UpdateCustomTexture(Texture2D tex);
/// <summary>
/// Add extra float data to the current video frame, optional
/// </summary>
/// <param name="data"></param>
void AddVideoFloatData(List<float> data = null);
/// <summary>
/// Set up audio capture available
/// </summary>
/// <param name="enable"></param>
void SetAudioEnable(bool enable);
/// <summary>
/// Set up video capture available
/// </summary>
/// <param name="enable"></param>
void SetVideoEnable(bool enable);
/// <summary>

```

```

    /// Switch device camera
    /// </summary>
    void SwitchCam();
    /// <summary>
    /// Set device front camera
    /// </summary>
    /// <returns></returns>
    bool SetCamFrontFacing();
    /// <summary>
    /// Get the current volume of the peer
    /// </summary>
    /// <param name="id">peer id</param>
    /// <returns></returns>
    float GetPeerAudioVolume(int id);
    /// <summary>
    /// Get peer videoInfo
    /// </summary>
    /// <param name="id"></param>
    /// <returns></returns>
    VideoInfo GetPeerTexture(int id);
    /// <summary>
    /// Get the current volume of yourself
    /// </summary>
    /// <returns></returns>
    float GetSelfAudioVolume();
    /// <summary>
    /// Get self videoInfo
    /// </summary>
    /// <returns></returns>
    VideoInfo GetSelfTexture();
    /// <summary>
    /// Start record audio
    /// </summary>
    /// <param name="limit">Maximum recording time, timeout
    automatically stop</param>
    /// <returns></returns>
    bool StartRecordAudio(int limit, OnRecordFinished onFinished);
    /// <summary>
    /// Stop record audio
    /// </summary>
    /// <returns></returns>
    void StopRecordAudio();
    /// <summary>
    /// Play record audio

```

```
/// </summary>  
/// <param name="recordData"></param>  
void PlayRecordAudio(byte[] recordData);
```

5. Update

This sdk will continue to be updated , we will listen to the developer's suggestions and improve the sdk, and also make a series of video tutorials to help developers use it more easily.

[Email:1786570525@qq.com](mailto:1786570525@qq.com)