Video Chat-Unity3D Documentation

Midnight Status

Introduction

Video Chat for Unity3D is an extension class that allows audio and video to be processed and queued for network transmission. The VideoChat class does this via the FromAudio() and FromVideo() function calls, typically in the Update callback of a Unity MonoBehaviour.

For example:

```
void Update() {

	VideoChat.PreVideo();

	VideoChat.FromAudio();

	//Run through all AudioPackets and send them across the network connection of
your choice

	VideoChat.FromVideo();

	//Run through all VideoPackets and send them across the network connection of
your choice
}
```

Depending on the network library employed the transmission and reception can vary. The simplest example uses Unity Networking RPC calls (Both Photon and TNet have similar calls).

For example:

```
//For each AudioPacket...
audioView.RPC( "ReceiveAudio", RPCMode.Others, currentPacket.position,
currentPacket.length, currentPacket.data );

//For each VideoPacket...
videoView.RPC( "ReceiveVideo", RPCMode.Others, currentPacket.x, currentPacket.y,
currentPacket.data );

[RPC]
void ReceiveVideo( int x, int y, byte[] videoData ) {
	VideoChat.ToVideo( x, y, videoData );
}

[RPC]
void ReceiveAudio( int micPosition, int length, byte[] audioData ) {
	VideoChat.ToAudio( micPosition, length, audioData );
}
```

Features

Video Chat-Unity3D can be used as a very simple chat application like Skype or FaceTime. The examples that come with the package demonstrate this type of capability. However, Video Chat can also go far beyond what has typically been done in the past. Because Video Chat can be used in 3D, all manner of effects can be employed to bring Video Chat and the players involved into the game world.

The VideoChat class contains several elements that can help with full 3D integration.

```
//Created in Add() if unassigned
public static GameObject     vcObject;
public static Material       cameraView;

//Created in Init() if unassigned
public static AudioSource    audioIn;
public static AudioSource    audioOut;
public static bool           audioOut3D;
```

The GameObject vcObject (short for Video Chat Object) is a game object from the current scene that VideoChat can render to. If this is unassigned then VideoChat will create a quad for rendering a generic view.

The Material cameraView is the material desired for rendering. VideoChat comes with a a CameraView material that can be assigned (or modified). However, If this field remains unassigned VideoChat will still create the shader and material automatically.

The AudioSource's audioIn and audioOut again will be created automatically if unassigned but can be assigned if so desired. The biggest benefit of this feature currently is the ability to assign audioOut as a 3D object. If the audioOut3D boolean flag is set to true then the clip associated with audioOut will also be in 3D.

Class Reference

VideoChat Class

```
//Created in Add() if unassigned
public static GameObject       vcObject;
public static Material         cameraView;

//Created in Init() if unassigned
public static AudioSource      audioIn;
```

```
//The audio source for audio output
public static AudioSource        audioOut;
//Should the audio source be in 3D?
public static bool               audioOut3D;

public static List<AudioPacket>  audioPackets;
public static List<VideoPacket>  videoPackets;
public static List<WebCamDevice> webCamDevices;

//Has the other user connected and sent us their video specs?
public static bool               videoPrimed;

public static bool               localView;

public static int                frameRate = 30;

//The amount a pixel must change before it’s considered a real change
public static float              pixelDeltaMagnitude = 0.002f;
//The number of chunks read from the video feed in a frame
public static int                chunksPerFrame = 10;
//The number of chunks to send across the network in a frame
public static int                packetsPerFrame = 10;

//The audio quality, 8000 is telephone quality
public static int                audioFrequency = 5000;
//Depending on the network smaller or larger sizes may be desired
public static int                audioPacketSize = 500;

//The magnitude a sound must have before it is considered communication
public static float              audioThreshold = 0.3f;

//Cancel echo caused by speaker feeding the microphone?
public static bool               acousticEchoCancellation;

//The delay before we consider the other user to be done speaking
public static float              acousticDelay = 1.0f;
public static float              acousticTimer;

//The number of pixels in a chunk
public static int                chunkSize = 16;

public static int                requestedWidth;
public static int                requestedHeight;
public static int                webCamWidth;
public static int                webCamHeight;
public static int                renderWidth;
public static int                 renderHeight;
public static int                 startCropX;
public static int                 startCropY;

//The setter here resets the camera. If the value is incremented or
decremented it will switch between available cameras
public static  int deviceIndex {
            get{  return _deviceIndex; }
            set{}
}

public static void Add( GameObject addObject = null, AudioSource addAudioIn =
null, AudioSource addAudioOut = null ) {
//This adds an object to the scene, can be done at start or upon network
instantiation
}

public static void Init() {
//Creates a new list of video packets and resets the deviceIndex
}

public static void FromAudio() {
//Collects audio data, packages it, and adds it to the list of audio packets
}

public static void PreVideo() {
//Prepares VideoChat and sets up the local view (if used)
}

public static void FromVideo() {
//Collects video data, packages it, and adds it to the list of video packets
}

public static void ToAudio( int position, int length, byte[] data ) {
//Applies audio data that arrived over the network
}

public static void ToVideo( int x, int y, byte[] data ) {
//Applies video data that arrived over the network
}
```