using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.Video;

using UnityEngine.SceneManagement;

//How the videos are loaded and displayed

public class StoreVideoPaths : MonoBehaviour

{

public GameObject videoPanel; //videos UI elements are instatiated as this

public GameObject parentCanvas; //video UI parent. videoPanel is instantiated here

public GameObject videoHolder; //video video players are instatiated as this

public GameObject videoParent; //videoHolder parent. videoHolder instantiated here

public GameObject videoZoom; //expanded video screen

public GameObject galleryCanvas; //gallery home screen

public GameObject photoCanvas; //photo album screen

public GameObject videoCanvas; //video album screen

int videoNumber; //which video in clicked on gallery is stored here

Transform clone; //this is the screen that shows the full screen video

public void Start()

{

DontDestroyOnLoad(this.gameObject);

if(GameObject.Find(gameObject.name) && GameObject.Find(gameObject.name) != this.gameObject)

{

Destroy(GameObject.Find(gameObject.name));

}

//gets list of videos stored in the Application.persistentDataPath

SerializationExample.Load();

StartCoroutine(LoadScene());

}

//this script exists in both the AR and gallery scene but should only be activated when in the gallery scene

public IEnumerator LoadScene()

{

while(SceneManager.GetActiveScene().name != "Gallery ")

{

yield return null;

}

yield return new WaitForSeconds(1.0f);

parentCanvas = GameObject.Find("VideoUIParent");

videoParent = GameObject.Find("VideoParent");

videoZoom = GameObject.Find("VideoZoom");

galleryCanvas = GameObject.Find("GalleryCanvas");

videoCanvas = GameObject.Find("VideoCanvas");

photoCanvas = GameObject.Find("PhotosCanvas");

StartCoroutine(Prepare());

}

public IEnumerator Prepare()

{

//Get rid of all videos currently in the scene

foreach (Transform child in parentCanvas.transform)

{

GameObject.Destroy(child.gameObject);

}

SerializationExample.Load(); //load all the URLs saved to the static list

yield return new WaitForEndOfFrame();

//display videos on instantiated panels

for (int i = 0; i < SerializationExample.videoFiles.Count; i++)

{

if (SerializationExample.videoFiles.Count > 0)

{

//UI element for videos

GameObject videoUI = Instantiate(videoPanel) as GameObject;

//giving an original name

videoUI.name = i.ToString();

videoUI.transform.parent = parentCanvas.transform;

//gameObject with video player to be applied to UI element respectively

GameObject videoObject = Instantiate(videoHolder) as GameObject;

videoObject.name = i.ToString();

videoObject.transform.parent = videoParent.transform;

//set up all urls to all video players

videoObject.GetComponent<VideoPlayer>().url = SerializationExample.videoFiles[i];

//load videos

videoObject.GetComponent<VideoPlayer>().Prepare();

yield return new WaitForSeconds(1);

//set raw images as the video texture

videoUI.GetComponent<RawImage>().texture = videoObject.GetComponent<VideoPlayer>().texture;

//play videos for just a moment to get a thumbnail

videoObject.GetComponent<VideoPlayer>().Play();

yield return new WaitForSeconds(0.5f);

videoObject.GetComponent<VideoPlayer>().Pause();

}

}

}

//Allows for user to click on a video thumbnail and view full size

public void ClickToZoom()

{

//get panel name and convert name to int

//get list position i

//convert video of that url to texture

//apply texture to VideoZoom raw image

videoNumber = PlayerPrefs.GetInt("VideoNumber");

//UI element that is full screen size

videoZoom.GetComponent<RawImage>().enabled = true;

//this element gets the same video assigned to it as the one the user clicked on

clone = videoParent.transform.GetChild(videoNumber);

Debug.Log("the clone name you clicked is: " + clone.name);

videoZoom.GetComponent<RawImage>().texture = clone.GetComponent<VideoPlayer>().texture;

clone.GetComponent<VideoPlayer>().Play();

PlayerPrefs.SetString("VideoPath", clone.GetComponent<VideoPlayer>().url);

}

//go back to video gallery/escape full screen view

public void Back()

{

clone.GetComponent<VideoPlayer>().Pause();

}

//delete a video

public void Delete()

{

clone.GetComponent<VideoPlayer>().Stop();

//look in list for current URL and remove it

SerializationExample.Remove(clone.GetComponent<VideoPlayer>().url);

//reload videos in gallery to account for the deleted video

StartCoroutine(Prepare());

videoZoom.GetComponent<RawImage>().enabled = false

GameObject SSViewer = GameObject.Find("ScreenShotViewer");

//if no videos left, go to photo album instead of showing an empty video gallery

//if no videos or photos left, navigate back to AR scene

SSViewer.GetComponent<GalleryGrid>().MediaCount();

}

}