using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using System.IO;

using UnityEngine.SceneManagement;

public class GalleryGrid : MonoBehaviour

{

//displays photos in gallery

public GameObject panel; //object being instatiated for each photo

public GameObject canvas; //parent object for instantiated photos

public string[] files = null; //where screenshots are stored

int whichScreenShotIsShown = 0; //a particular screenshot

Sprite sprite; //where screenshot will be placed

public GameObject fullScreenPhoto; //full size photo

public GameObject buttonsCanvas; //Holds the delete/share buttons

public GameObject nextButton; //scroll back and forth in full screen

public GameObject previousButton; //see above

int childNumber; //which photo to show in fullscreen mode

public GameObject gallery; //gallery home page

bool whitePanel = true; //any instantiated panel without a texture

public GameObject photoCanvas; //photo album home page

public GameObject videoCanvas; //video album home page

//these gameobjects are used when there may or may not be any photos or videos left.

public GameObject backButton;

public GameObject photoBack;

public GameObject photoBackHome;

public GameObject videoBack;

public GameObject videoBackHome;

//Get photos and check to see if they exist

void Start()

{

files = Directory.GetFiles(Application.persistentDataPath + "/", "\*.png");

if (files.Length > 0)

{

GetPictureAndShowIt();

}

}

//Get rid of instantiated objects without a texture

public void GetRid()

{

foreach (Transform child in canvas.transform)

{

GameObject.Destroy(child.gameObject);

}

whitePanel = true;

GetPictureAndShowIt();

}

public void GetPictureAndShowIt()

{

files = Directory.GetFiles(Application.persistentDataPath + "/", "\*.png");

//Instantiate UI element for each photo and assign each photo as a texture

for (int i = 0; i <= files.Length; i++)

{

GameObject child = Instantiate(panel) as GameObject;

child.name = i.ToString();

child.transform.parent = canvas.transform;

string pathToFile = files[whichScreenShotIsShown = i];

Texture2D texture = GetScreenshotImage(pathToFile);

sprite = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height), new Vector2(0.5f, 0.5f));

panel.GetComponent<Image>().sprite = sprite;

if (whitePanel == true)

{

GameObject.Destroy(canvas.transform.GetChild(0).gameObject);

whitePanel = false;

}

}

}

//Get each photo and load them

Texture2D GetScreenshotImage(string filePath)

{

Texture2D texture = null;

byte[] fileBytes;

if (File.Exists(filePath))

{

fileBytes = File.ReadAllBytes(filePath);

texture = new Texture2D(2, 2, TextureFormat.RGB24, false);

texture.LoadImage(fileBytes);

}

return texture;

}

//Enter full screen mode

public void ZoomIn()

{

//if clicked child i of canvas, get file i texture path, change texture to sprite

fullScreenPhoto.SetActive(true);

buttonsCanvas.SetActive(true);

childNumber = PlayerPrefs.GetInt("child");

string pathToFile = files[childNumber - 1];

Texture2D texture = GetScreenshotImage(pathToFile);

Sprite sp = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height), new Vector2(0.5f, 0.5f));

fullScreenPhoto.GetComponent<Image>().sprite = sp;

PlayerPrefs.SetString("ImagePath", pathToFile);

}

//Exit full screen mode

public void ZoomOut()

{

fullScreenPhoto.SetActive(false);

buttonsCanvas.SetActive(false);

}

//Delete a photo

public void DeleteImage()

{

if (files.Length > 0)

{

string pathToFile = files[childNumber - 1];

if (File.Exists(pathToFile))

{

File.Delete(pathToFile);

ZoomOut();

SceneManager.LoadScene("Gallery");

}

files = Directory.GetFiles(Application.persistentDataPath + "/", "\*.png");

}

}

//scroll through pictures in full screen mode

public void NextPicture()

{

if(childNumber + 2 == files.Length)

{

nextButton.SetActive(false);

}

else

{

nextButton.SetActive(true);

}

if (childNumber + 2 <= files.Length)

{

string pathToFile = files[childNumber += 1];

Texture2D texture = GetScreenshotImage(pathToFile);

Sprite sp = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height), new Vector2(0.5f, 0.5f));

fullScreenPhoto.GetComponent<Image>().sprite = sp;

previousButton.SetActive(true);

}

}

//Scroll through photos in full screen mode

public void PreviousPicture()

{

if (childNumber - 1 == 0)

{

previousButton.SetActive(false);

}

else

{

previousButton.SetActive(true);

}

if (childNumber - 1 >= 0)

{

string pathToFile = files[childNumber -= 1];

Texture2D texture = GetScreenshotImage(pathToFile);

Sprite sp = Sprite.Create(texture, new Rect(0, 0, texture.width, texture.height), new Vector2(0.5f, 0.5f));

fullScreenPhoto.GetComponent<Image>().sprite = sp;

nextButton.SetActive(true);

}

}

public void MediaCount()

{

//if no photos and no videos

if (files.Length == 0 && SerializationExample.videoFiles.Count == 0)

{

//Return to AR scene

backButton.GetComponent<LoadASync>().LoadAO();

}

//if photos but no videos, go to photo canvas

else if (files.Length > 0 && SerializationExample.videoFiles.Count == 0)

{

photoCanvas.GetComponent<Canvas>().sortingOrder = 2;

videoCanvas.GetComponent<Canvas>().sortingOrder = 0;

gallery.GetComponent<Canvas>().sortingOrder = 0;

photoBack.SetActive(false);

photoBackHome.SetActive(true);

}

//videos but no photos, go to video canvas

else if(files.Length == 0 && SerializationExample.videoFiles.Count > 0)

{

photoCanvas.GetComponent<Canvas>().sortingOrder = 0;

videoCanvas.GetComponent<Canvas>().sortingOrder = 2;

gallery.GetComponent<Canvas>().sortingOrder = 0;

videoBack.SetActive(false);

videoBackHome.SetActive(true);

}

}

}