using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.SceneManagement;

using UnityEngine.UI;

using System.IO;

//This script is called when camera button is pressed.

public class TakeScreenshot : MonoBehaviour {

[SerializeField]

GameObject blink; //Mimics a camera flash on the screen

public GameObject UI; //UI elements to disappear in screenshot

public GameObject header; //UI elements to disappear in screenshot

public Button GalleryButton; //Navigates user to Gallery scene.

public string[] files = null; //This is where the photos are stored

public GameObject afterScreenshotPanel; //Immediately shows photo just taken

void Start()

{

//Get all the photos that currently exist

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

}

//Turn off all UI and begin taking the screenshot

public void TakeAShot()

{

UI.SetActive(false);

header.SetActive(false);

StartCoroutine(CaptureIt());

}

//Deletes the photo that was just taken/that shows up in the "afterScreenshotPanel".

//Connected to the “Retake” button.

//Different from the delete function used in the Gallery scene.

public void DeletePhoto(){

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

string pathToFile = files[files.Length - 1];

if (File.Exists(pathToFile))

{

File.Delete(pathToFile);

afterScreenshotPanel.SetActive(false);

UI.SetActive(true);

header.SetActive(true);

}

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

//This Begin() function updates the gallery button with the most recent photo

GalleryButton.GetComponent<GalleryButtonImage>().Begin();

}

//Photos are automatically saved. This serves a purpose of turning the UI back on.

//Connected to the “Save” button in the “afterScreenshotPanel”.

public void SavePhoto(){

afterScreenshotPanel.SetActive(false);

UI.SetActive(true);

header.SetActive(true);

}

//The code that is called to take the photo

IEnumerator CaptureIt()

{

string timeStamp = System.DateTime.Now.ToString("dd-MM-yyyy-HH-mm-ss");

string fileName = "Screenshot" + timeStamp + ".png";

string pathToSave = fileName;

ScreenCapture.CaptureScreenshot(pathToSave);

PlayerPrefs.SetString("FilePath", pathToSave);

yield return new WaitForEndOfFrame();

Instantiate (blink, new Vector2(530f, 974f), Quaternion.identity);

yield return new WaitForSeconds(0.5f);

afterScreenshotPanel.SetActive(true);

GalleryButton.GetComponent<GalleryButtonImage>().Begin();

}

}