using System.Collections;

using System.Collections.Generic;

using UnityEngine;

using UnityEngine.UI;

using UnityEngine.SocialPlatforms;

#if UNITY\_ADS

using UnityEngine.Advertisements;

#endif

public class GlobalControl : MonoBehaviour {

public static GlobalControl Instance;

public bool restart = false;

public int highScore;

public int testHS;

public int setHSCount;

public int HSUICount;

public int gamePurchased;

public bool purchasedMusicBool = false;

public bool musicOn = true;

public bool SFXOn = true;

public int musicOnCount;

public int SFXOnCount;

public bool adCountBool;

public bool adDone;

public int testInt;

public bool iphonex;

public int trophyInt;

public Vector3 Text1YRTPos;

public Vector3 hand1RTPos;

public Vector3 arrowRTPos;

public Vector3 tapTextsPos;

public Vector3 scorePos;

public Vector3 feathersUIPos;

public Vector3 pointsPos;

public Vector3 fAnimsPos;

public Vector3 alertPos;

public Vector3 soundButtonPos;

public Vector3 musicButtonPos;

public Vector3 sfxButtonPos;

public Vector3 purchasedMusicButtonsPos;

public Vector3 defaultMusicPos;

public Vector3 iapButtonPos;

public Vector3 retryButtonPos;

public Vector2 RSPBSize;

public Vector2 PBSize;

public Vector3 RSPBPos;

public Vector3 PBPos;

public int iphoneXModeCount;

void Awake ()

{

Text1YRTPos = new Vector3 (562.5f, 2200f, 0f);

hand1RTPos = new Vector3 (562.5f, 1950f, 0f);

arrowRTPos = new Vector3 (900f, 1100f, 0f);

tapTextsPos = new Vector3 (358.5f, 1160f, 0f);

scorePos = new Vector3 (562.5f, 2065f, 0f);

feathersUIPos = new Vector3 (562.5f, 1675f, 0f);

pointsPos = new Vector3 (1000f, 2000f, 0f);

fAnimsPos = new Vector3 (562.5f, 1675f, 0f);

alertPos = new Vector3 (850f, 2100f, 0f);

//soundButtonPos = new Vector3 (90f, 2000f, 0f);

//musicButtonPos = new Vector3 (260f, 2100f, 0f);

//sfxButtonPos = new Vector3 (680f, 1976f, 0f);

//purchasedMusicButtonsPos = new Vector3 (995f, 1976f, 0f);

//defaultMusicPos = new Vector3 (995f, 1675f, 0f);

//iapButtonPos = new Vector3 (865f, 300f, 0f);

retryButtonPos = new Vector3 (260f, 300f, 0f);

//RSPBPos = new Vector3 (965f, 1200f, 0f);

PBPos = new Vector3 (850f, 300f, 0f);

//PlayerPrefs.DeleteKey ("gamePurchased");

//PlayerPrefs.DeleteKey("HighScore");

//PlayerPrefs.DeleteKey("purchasedMusic");

adCountBool = Advertisement.isShowing;

gamePurchased = PlayerPrefs.GetInt("gamePurchased");

if (PlayerPrefs.GetInt("purchasedMusic") == 1)

{

purchasedMusicBool = true;

}

else if (PlayerPrefs.GetInt("purchasedMusic") == 0)

{

purchasedMusicBool = false;

}

if (musicOnCount == 0)

{

musicOn = true;

}

if (SFXOnCount == 0)

{

SFXOn = true;

}

if (Instance == null)

{

DontDestroyOnLoad(gameObject);

Instance = this;

}

else if (Instance != this)

{

Destroy (gameObject);

}

}

void Update()

{

if (iphoneXModeCount < 1 && GameControl.instance.gameOver == false)

{

IphoneXStuff ();

iphoneXModeCount++;

}

else if (iphoneXModeCount < 1 && GameControl.instance.gameOver == true)

{

gameOverIphoneXStuff ();

iphoneXModeCount++;

}

adCountBool = Advertisement.isShowing;

testHS = PlayerPrefs.GetInt ("HighScore");

if (testHS == 0)

{

highScore = testHS;

if (GameControl.instance.score > highScore)

{

highScore = GameControl.instance.score;

}

}

else if(testHS > 0 && setHSCount < 1)

{

highScore = testHS;

setHSCount++;

}

if (testHS > 0)

{

if (GameControl.instance.score > highScore)

{

highScore = GameControl.instance.score;

}

}

if (GameControl.instance.gameOver == false)

{

HSUICount = 0;

}

if (GameControl.instance.gameOver == true && HSUICount < 1)

{

StopCoroutine ("HSStuffWait");

StartCoroutine ("HSStuffWait");

HSUICount++;

}

}

public void IphoneXStuff ()

{

if (UnityEngine.iOS.Device.generation == UnityEngine.iOS.DeviceGeneration.iPhoneX)

{

iphonex = true;

GameControl.instance.player.transform.position = new Vector3 (0, -2.5f, 0);

if (restart == false && GameControl.instance.titleYellow != null)

{

RectTransform T1rt = GameControl.instance.titleYellow.GetComponent<RectTransform> ();

T1rt.position = Text1YRTPos;

RectTransform T1rt2 = GameControl.instance.titleBlack.GetComponent<RectTransform> ();

T1rt2.position = Text1YRTPos;

RectTransform H1rt = GameControl.instance.hand1.GetComponent<RectTransform> ();

H1rt.position = hand1RTPos;

RectTransform Arwrt = GameControl.instance.arrow.GetComponent<RectTransform> ();

Arwrt.position = arrowRTPos;

RectTransform TTrt = GameControl.instance.tapText1.GetComponent<RectTransform> ();

TTrt.position = tapTextsPos;

RectTransform TTrt2 = GameControl.instance.tapText2.GetComponent<RectTransform> ();

TTrt2.position = tapTextsPos;

}

RectTransform SCrt = GameControl.instance.scoreText.rectTransform;

SCrt.position = scorePos;

RectTransform F0rt = GameControl.instance.Feathers0.GetComponent<RectTransform> ();

RectTransform F1rt = GameControl.instance.Feathers1.GetComponent<RectTransform> ();

RectTransform F2rt = GameControl.instance.Feathers2.GetComponent<RectTransform> ();

RectTransform F3rt = GameControl.instance.Feathers3.GetComponent<RectTransform> ();

F0rt.position = feathersUIPos;

F1rt.position = feathersUIPos;

F2rt.position = feathersUIPos;

F3rt.position = feathersUIPos;

RectTransform P1rt = GameControl.instance.One.GetComponent<RectTransform> ();

RectTransform P2rt = GameControl.instance.Two.GetComponent<RectTransform> ();

RectTransform P3rt = GameControl.instance.Three.GetComponent<RectTransform> ();

P1rt.position = pointsPos;

P2rt.position = pointsPos;

P3rt.position = pointsPos;

RectTransform FA1rt = GameControl.instance.smallFeatherAnim.GetComponent<RectTransform> ();

RectTransform FA2rt = GameControl.instance.mediumFeatherAnim.GetComponent<RectTransform> ();

RectTransform FA3rt = GameControl.instance.largeFeatherAnim.GetComponent<RectTransform> ();

FA1rt.position = fAnimsPos;

FA2rt.position = fAnimsPos;

FA3rt.position = fAnimsPos;

RectTransform Alrt = GameControl.instance.alert.GetComponent<RectTransform> ();

Alrt.position = alertPos;

//RectTransform SOrt = GameControl.instance.SoundOptions.GetComponent<RectTransform> ();

//SOrt.position = soundButtonPos;

RectTransform MSrt = GameControl.instance.MusicToggleButton.GetComponent<RectTransform> ();

RectTransform SFXrt = GameControl.instance.SFXToggleButton.GetComponent<RectTransform> ();

//RectTransform DBrt = GameControl.instance.DummyPurchasedMusicButton.GetComponent<RectTransform> ();

RectTransform PMrt = GameControl.instance.PurchasedMusicToggle.GetComponent<RectTransform> ();

RectTransform DMrt = GameControl.instance.DefaultMusicToggle.GetComponent<RectTransform> ();

//RectTransform IAPrt = GameControl.instance.IAPButton.GetComponent<RectTransform> ();

RectTransform RBrt = GameControl.instance.retryButton.GetComponent<RectTransform> ();

//MSrt.position = musicButtonPos;

//SFXrt.position = sfxButtonPos;

//DBrt.position = purchasedMusicButtonsPos;

//PMrt.position = purchasedMusicButtonsPos;

//DMrt.position = defaultMusicPos;

//IAPrt.position = iapButtonPos;

RBrt.position = retryButtonPos;

}

}

public void gameOverIphoneXStuff()

{

if (UnityEngine.iOS.Device.generation == UnityEngine.iOS.DeviceGeneration.iPhoneX)

{

iphonex = true;

RectTransform RSPBrt = GameControl.instance.RestoreButton.GetComponent<RectTransform> ();

RectTransform PBrt = GameControl.instance.PurchaseButton.GetComponent<RectTransform> ();

RSPBSize = new Vector2 (130f, RSPBrt.sizeDelta.y);

PBSize = new Vector2 (130f, PBrt.sizeDelta.y);

//RSPBrt.sizeDelta = RSPBSize;

//PBrt.sizeDelta = PBSize;

//RSPBrt.position = RSPBPos;

PBrt.position = PBPos;

}

}

public IEnumerator HSStuffWait()

{

if (PlayerPrefs.GetInt ("gamePurchased") == 0) {

yield return new WaitForSecondsRealtime (3.0f);

} else {

yield return new WaitForSecondsRealtime (2.0f);

}

GameControl.instance.highScoreStuff.SetActive (true);

GameControl.instance.trophyScores.SetActive (true);

GameControl.instance.retryButton.gameObject.SetActive(true);

GameControl.instance.highScoreText.text = "HighScore" + "\r\n" + highScore.ToString ();

PlayerPrefs.SetInt ("HighScore", highScore);

if (highScore >= 0 && highScore <= 99)

{

GameControl.instance.wood.SetActive (true);

trophyInt = 1;

}

else if (highScore >= 100 && highScore <= 199)

{

GameControl.instance.bronze.SetActive (true);

trophyInt = 2;

}

else if (highScore >= 200 && highScore <= 299)

{

GameControl.instance.silver.SetActive (true);

trophyInt = 3;

}

else if (highScore >= 300 && highScore <= 499)

{

GameControl.instance.gold.SetActive (true);

trophyInt = 4;

}

else if (highScore >= 500)

{

GameControl.instance.pro.SetActive(true);

trophyInt = 5;

}

PlayerPrefs.Save ();

}

public void ShowDefaultAd()

{

#if UNITY\_ADS

if (!Advertisement.IsReady())

{

Debug.Log("Ads not ready for default placement");

return;

}

ShowOptions options = new ShowOptions();

options.resultCallback = AdCallbackHandler;

Advertisement.Show("video", options);

#endif

}

void AdCallbackHandler (ShowResult result)

{

switch(result)

{

case ShowResult.Finished:

adDone = true;

break;

case ShowResult.Skipped:

adDone = true;

break;

}

}

}