//

// Menu.swift

// Spike Hopper

//

// Created by Sevan Productions on 9/1/15.

// Copyright (c) 2015 Sevan Productions. All rights reserved.

//

import Foundation

import SpriteKit

import GameKit

class MainMenu: SKScene, SKPhysicsContactDelegate, GKGameCenterControllerDelegate {

//shows leaderboard screen

func showLeader() {

let vc = self.view?.window?.rootViewController

let gc = GKGameCenterViewController()

gc.gameCenterDelegate = self

vc?.present(gc, animated: true, completion: nil)

}

func saveHighscore(score:Int) {

//check if user is signed in

if GKLocalPlayer.localPlayer().isAuthenticated {

print("authenticated")

let scoreReporter = GKScore(leaderboardIdentifier: "grp.spikehopperleaderboard") //leaderboard id here

print("ScoreReporter: \(scoreReporter)")

scoreReporter.value = Int64(HighScore) //score variable here (same as above)

let scoreArray: [GKScore] = [scoreReporter]

GKScore.report(scoreArray, withCompletionHandler: {(error : Error?) -> Void in

if error != nil {

print("error")

}

else{

print("reported correctly")

}

})

}

}

//hides leaderboard screen

func gameCenterViewControllerDidFinish(\_ gameCenterViewController: GKGameCenterViewController)

{

gameCenterViewController.dismiss(animated: true, completion: nil)

}

//variable for the background image

var mainBackground = SKSpriteNode(imageNamed: "Background")

var mainBackgroundbig = SKSpriteNode(imageNamed: "Backgroundbig")

//variable for the play button

var playButton1 = SKSpriteNode(imageNamed: "playgame")

//vairable for the score button

var scoreButton = SKSpriteNode(imageNamed: "gamecenter")

//variable for the spike hopper text

var logoText = SKSpriteNode(imageNamed: "logo")

//triggered when the view loads

override func didMove(to: SKView) {

saveHighscore(score: HighScore)

//main background setup

if screenBounds.size.width == 1366 && screenBounds.size.height == 1024 {

mainBackground.removeFromParent()

self.mainBackgroundbig.anchorPoint = CGPoint(x: 0, y: 0)

self.mainBackgroundbig.position = CGPoint(x: 0, y: 0)

self.mainBackgroundbig.zPosition = 0

self.addChild(mainBackgroundbig)

}else{

self.mainBackground.anchorPoint = CGPoint(x: 0, y: 0)

self.mainBackground.position = CGPoint(x: self.frame.minX, y: self.frame.minY)

self.mainBackground.zPosition = 0

self.addChild(mainBackground)

}

//play button setup

self.playButton1.anchorPoint.y = 0.5

self.playButton1.position = CGPoint(x: self.frame.minX + 100, y: self.frame.minY + 100)

self.playButton1.zPosition = 30

self.addChild(playButton1)

//score button setup

self.scoreButton.anchorPoint.y = 0.5

self.scoreButton.position = CGPoint(x: self.frame.maxX - 100, y: self.frame.minY + 100)

self.scoreButton.zPosition = 30

self.addChild(scoreButton)

//logo text setup

self.logoText.xScale = 0.8

self.logoText.yScale = 0.8

self.logoText.position.x = self.frame.midX

self.logoText.position.y = self.frame.maxY - 100

self.logoText.zPosition = 30

self.addChild(logoText)

}

//triggered when the user touches the screen

override func touchesBegan(\_ touches: Set<UITouch>, with: UIEvent?) {

// stuff

for touch: AnyObject in touches {

let location = touch.location(in: self)

//triggered when the user touches the play button

if self.atPoint(location) == self.playButton1 {

//Sets up transition

let transition = SKTransition.fade(withDuration: 1)

//loads the scene in the class "GameScene"

let scene2 = GameScene(size: self.size)

let skView2 = self.view as SKView!

skView2!.ignoresSiblingOrder = true

skView2!.showsPhysics = false

scene2.scaleMode = .resizeFill

scene2.size = skView2!.bounds.size

skView2!.presentScene(scene2, transition: transition)

//Sets the score back to 0

score = 0

}

if self.atPoint(location) == self.scoreButton {

showLeader()

saveHighscore(score: HighScore)

}

}

}

}