#WWDC18

I Have This Idea for an App...💭

Session 203

Jessie Pease, Health Engineering
Tanu Singhal, UIKit Engineering

“To live a creative life, we must lose the fear of being wrong.“

Joseph Chilton Pearce

Our Journey 👣

Dream big and organize our ideas

Our Journey 👣

Dream big and organize our ideas

Learn to navigate Xcode

Our Journey 👣

Dream big and organize our ideas

Learn to navigate Xcode

Build a simple game using Swift

Our Journey 👣

Dream big and organize our ideas

Learn to navigate Xcode

Build a simple game using Swift

Add multiple views within our app

Our Journey 👣

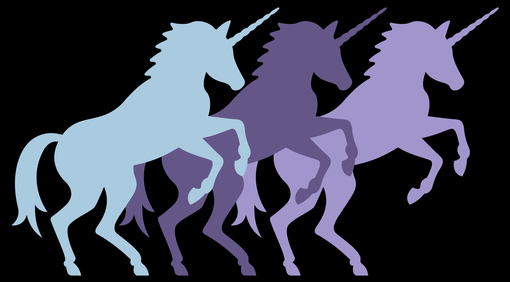
Dream big and organize our ideas

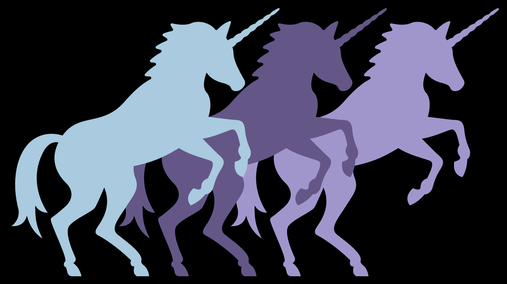
Learn to navigate Xcode

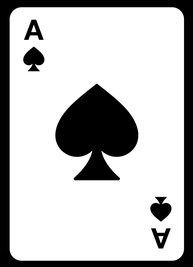
Build a simple game using Swift

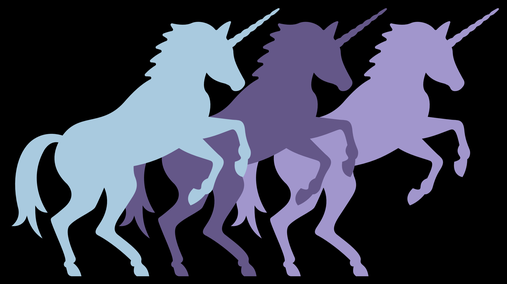
Add multiple views within our app

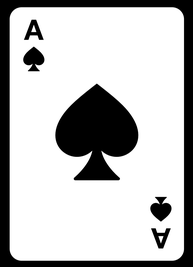
Persist and display data for users

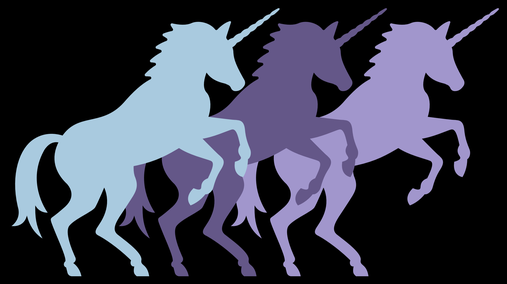


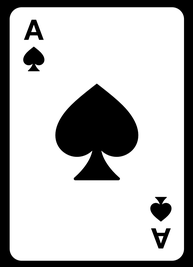


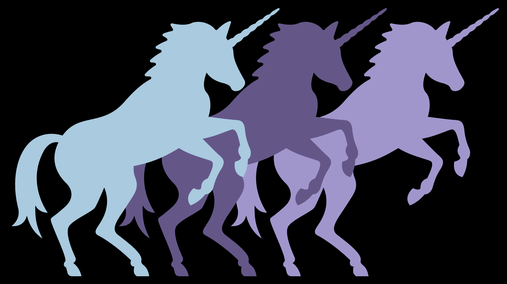

A
A

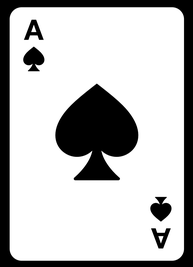


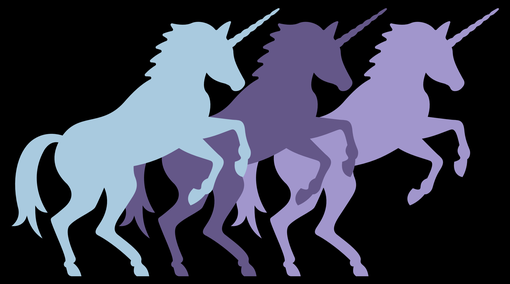

A
A

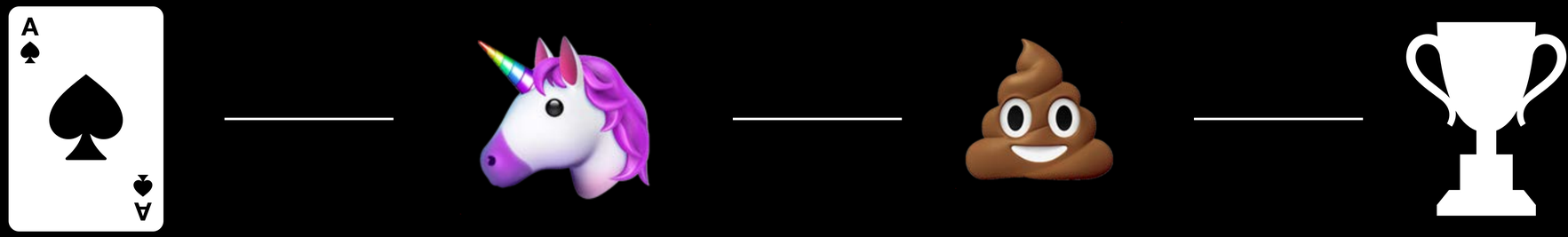


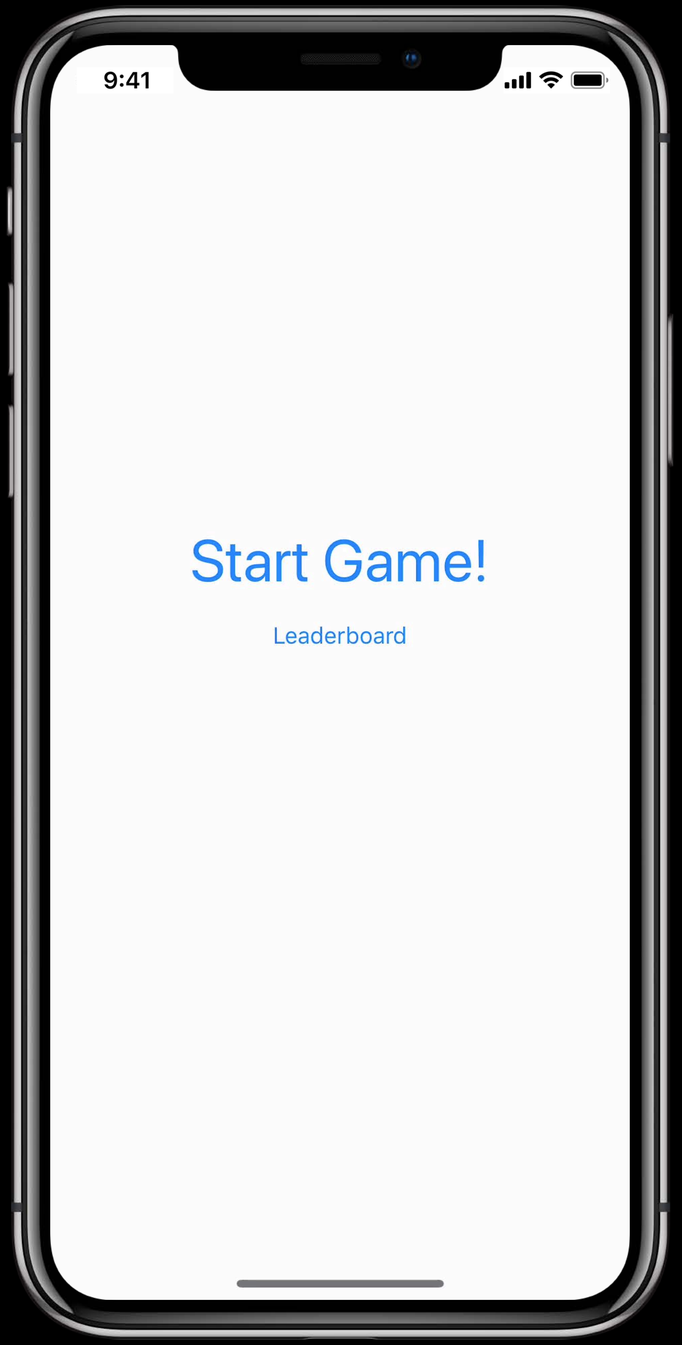

A
A

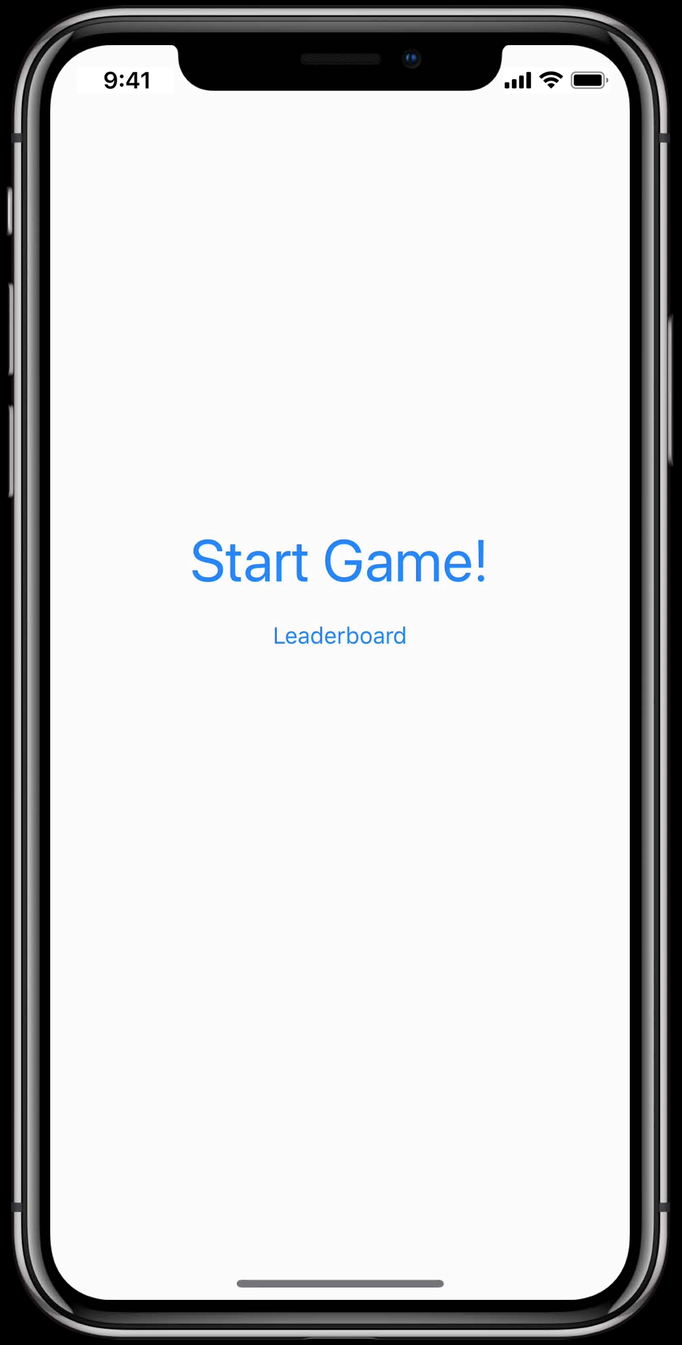


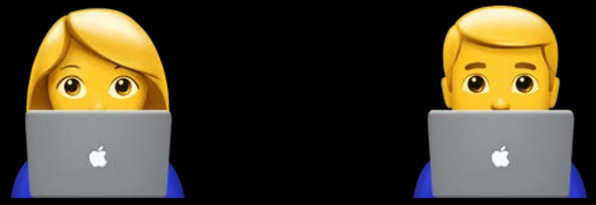

A
A





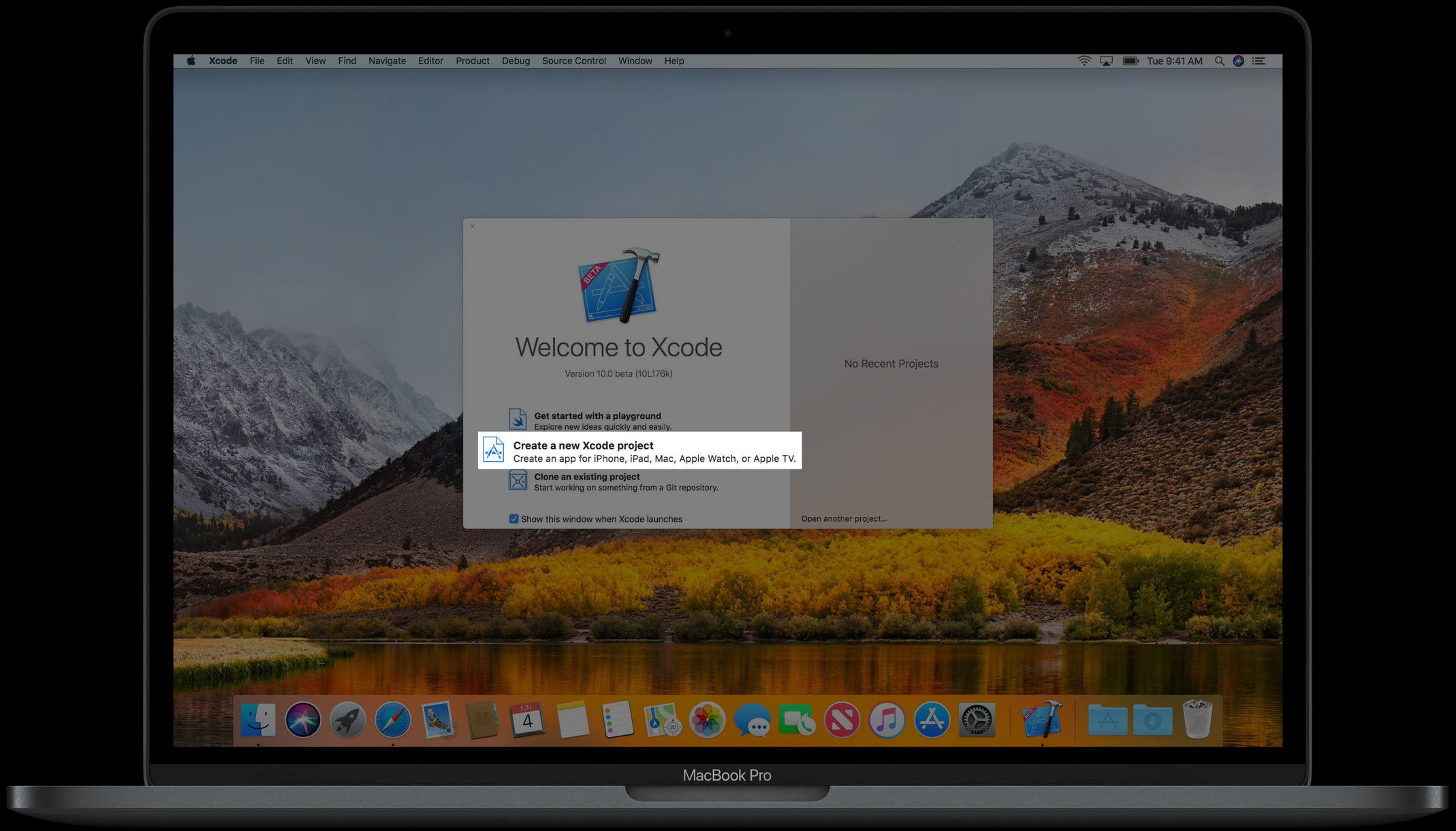

9:41
Start Game!
Leaderboard

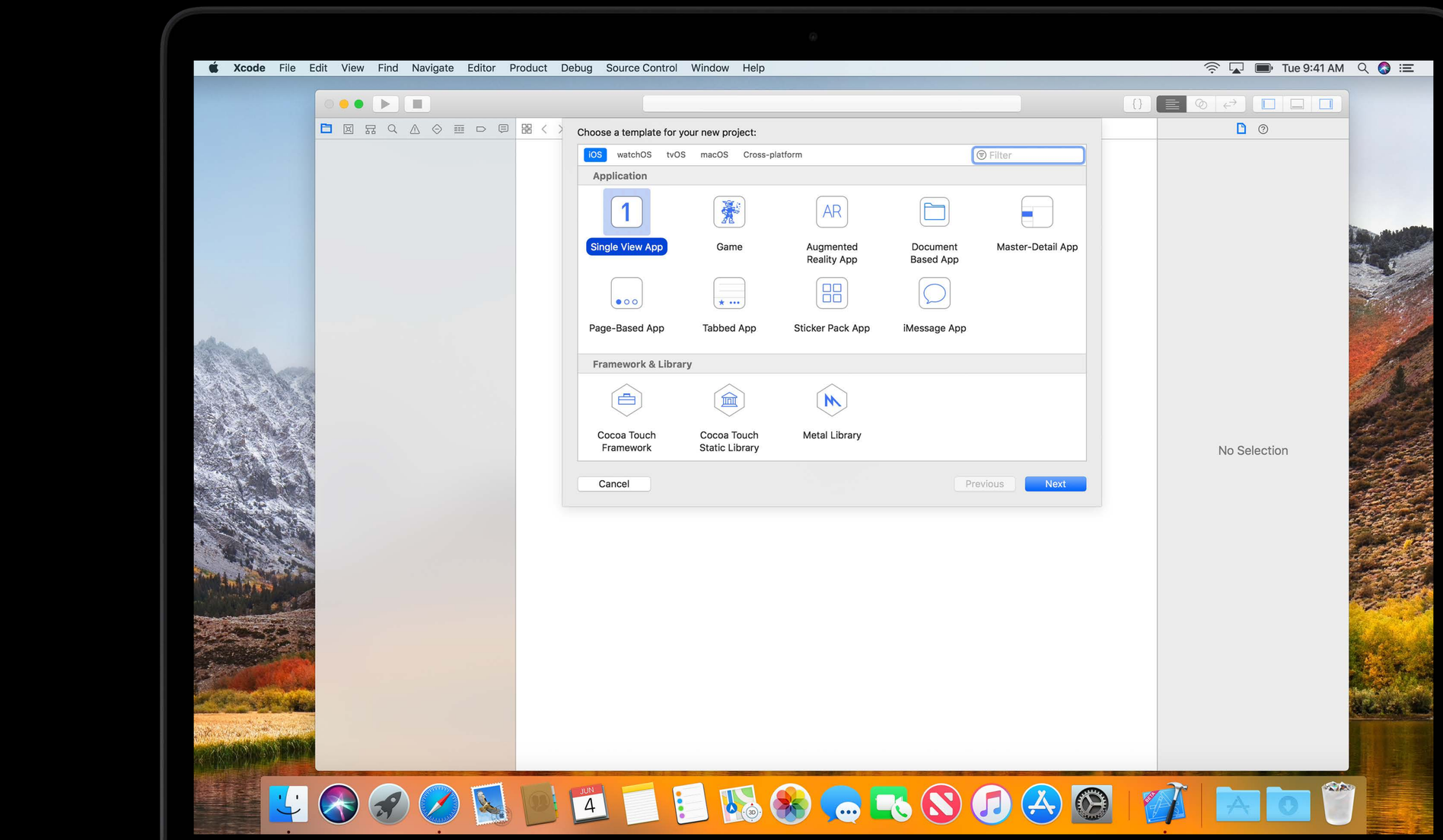

9:41
Start Game!
Leaderboard

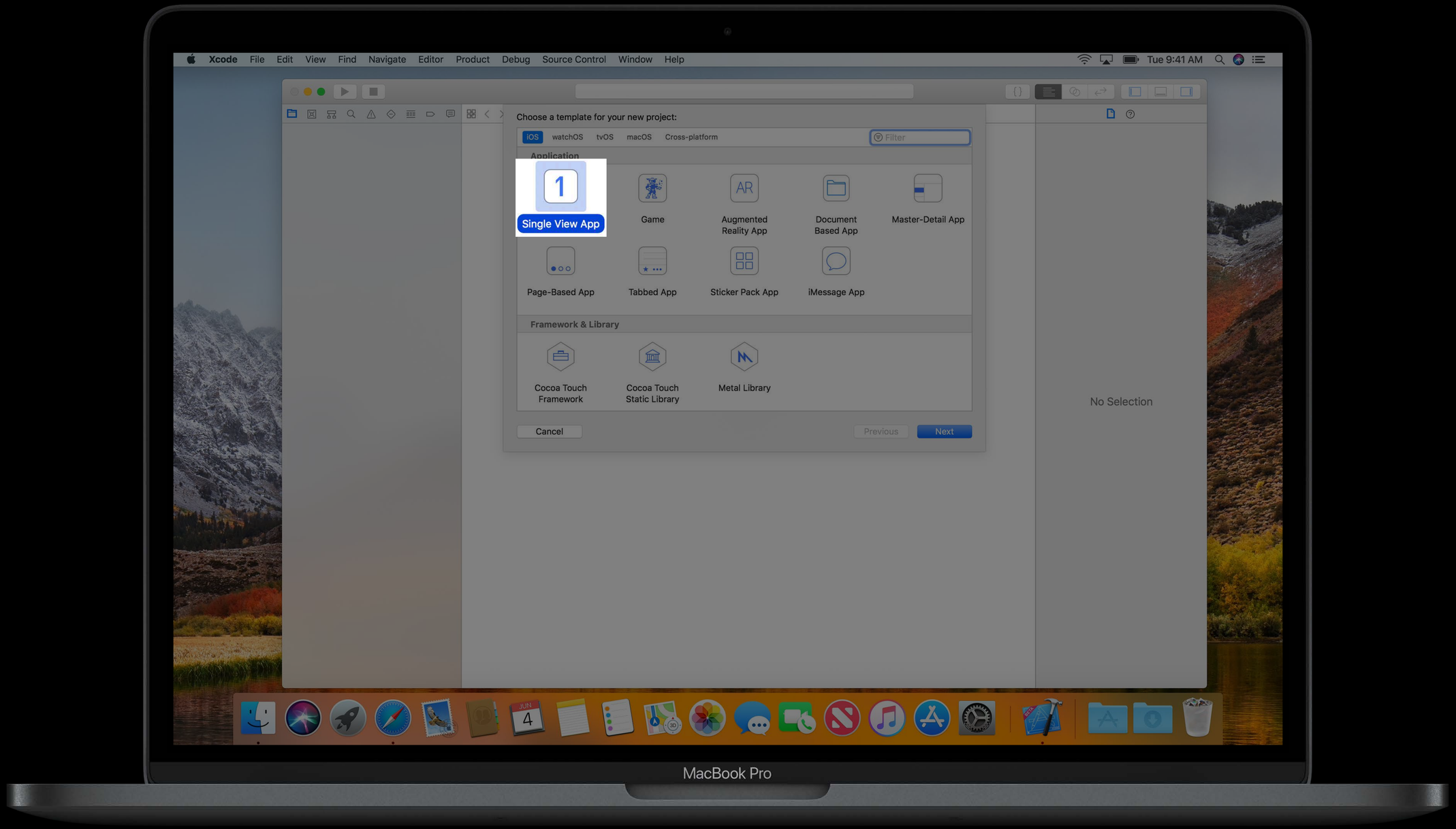


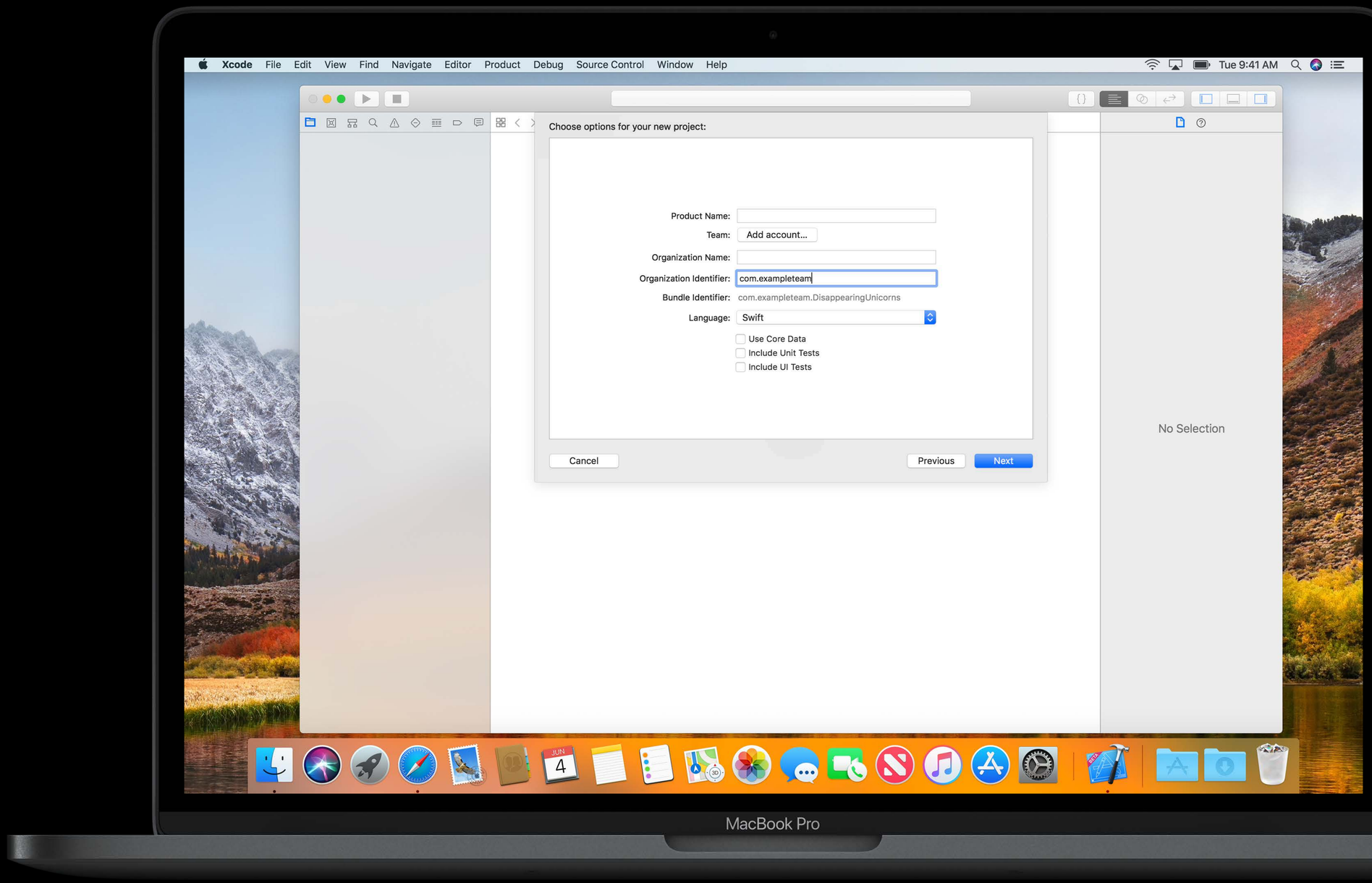
Navigating Xcode

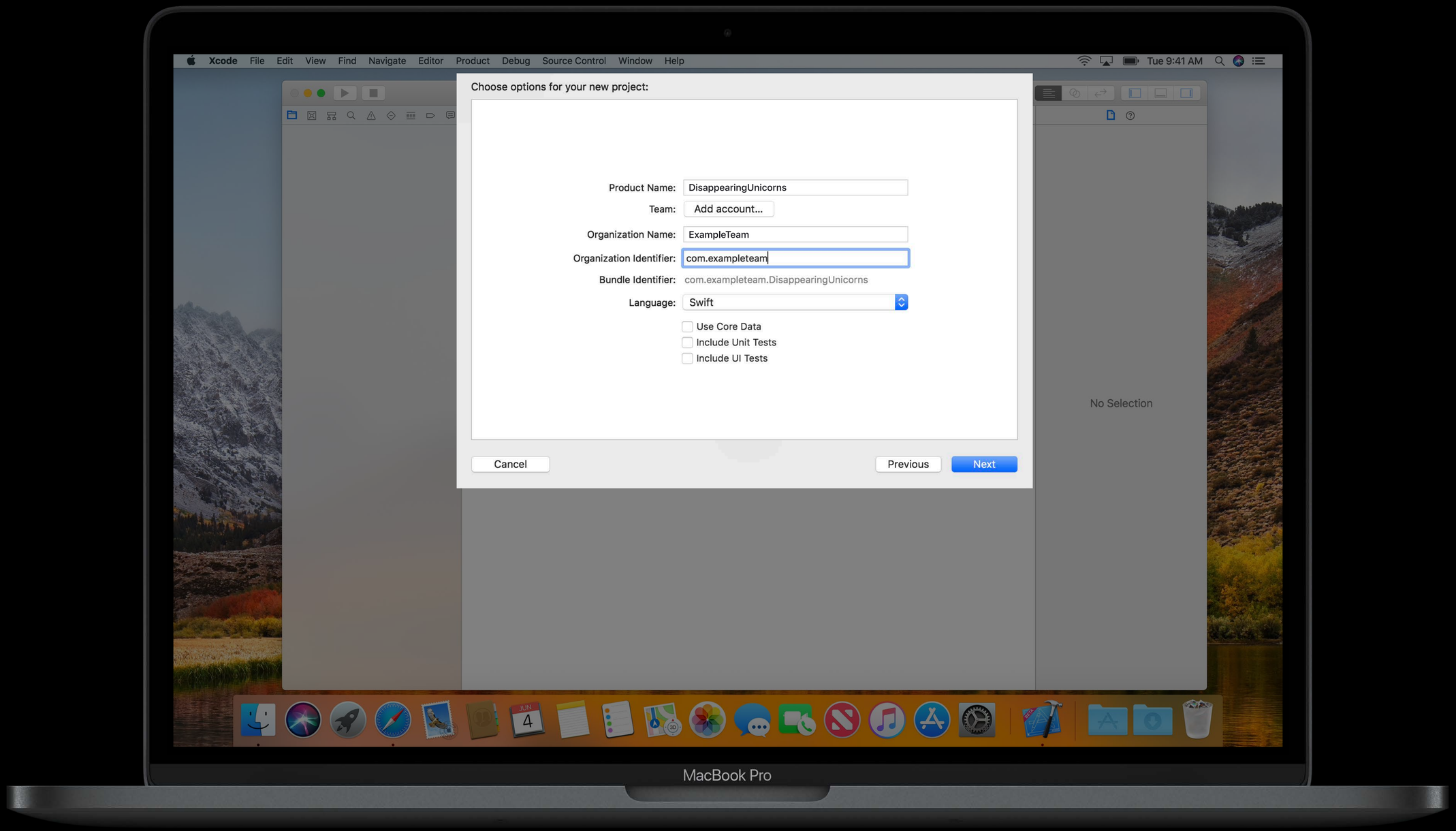

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Welcome to Xcode
Version 10.0 beta (10L176k)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.
Clone an existing project
Start working on something from a Git repository.
Show this window when Xcode launches
No Recent Projects
Open another project...
MacBook Pro

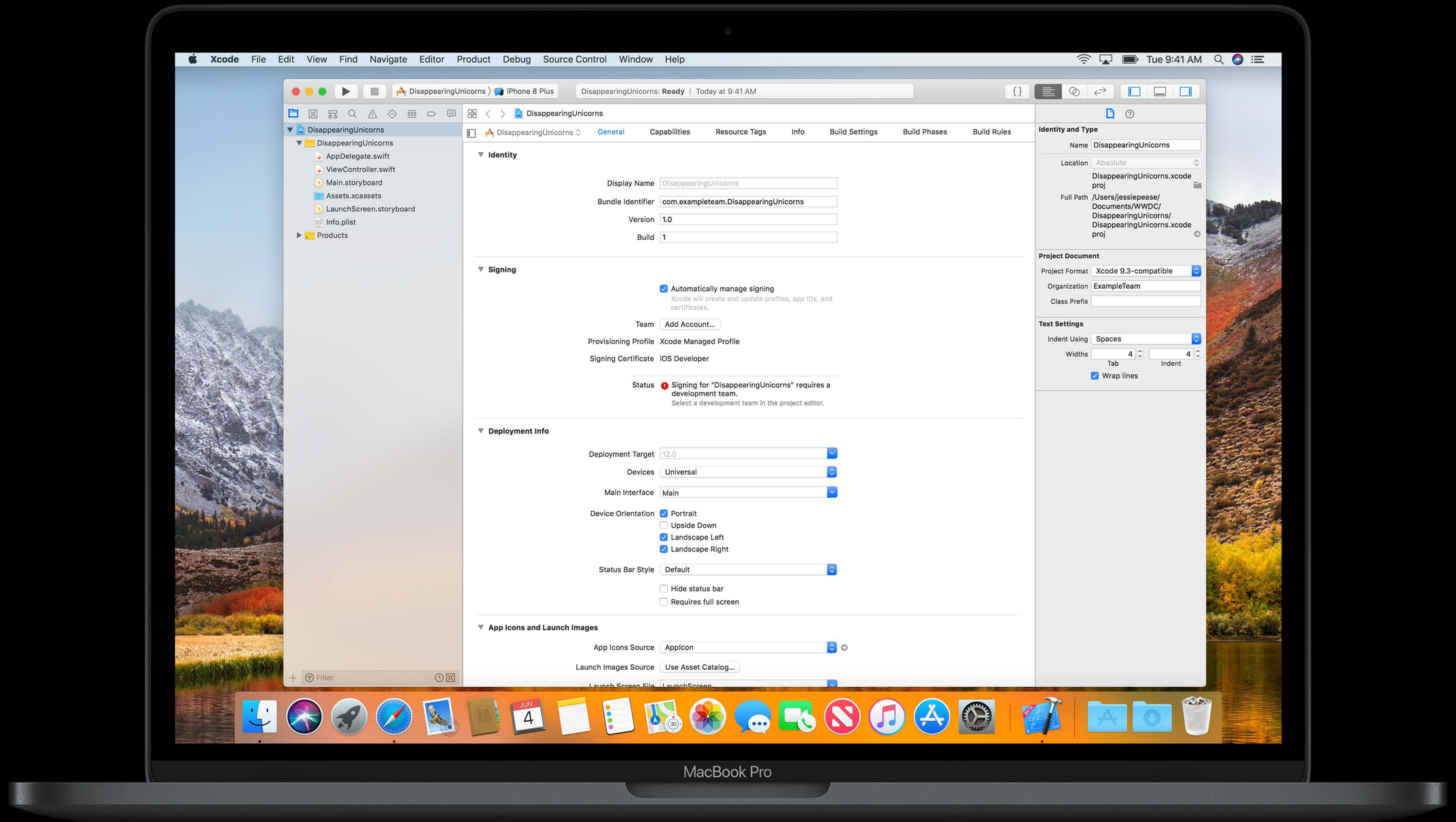

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Welcome to Xcode
Version 10.0 beta (10L176k)
Get started with a playground
Explore new ideas quickly and easily.
Create a new Xcode project
Create an app for iPhone, iPad, Mac, Apple Watch, or Apple TV.
Clone an existing project
Start working on something from a Git repository.
Show this window when Xcode launches
No Recent Projects
Open another project...
MacBook Pro

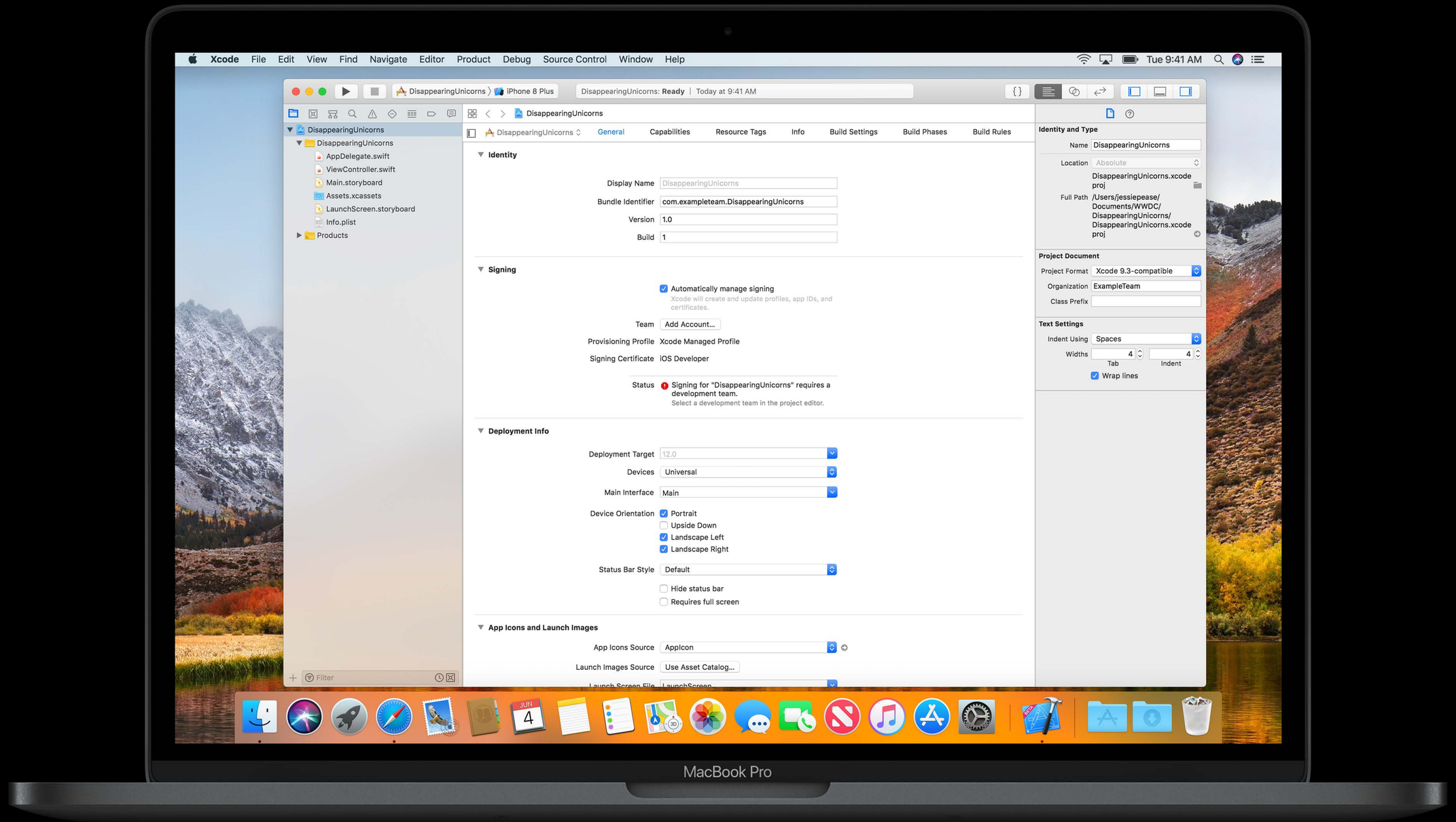

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Choose a template for your new project:
iOS watchOS tvOS macOS Cross-platform
Filter
Application
Single View App
Game
Augmented Reality App
Document Based App
Master-Detail App
Page-Based App
Tabbed App
Sticker Pack App
iMessage App
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next
No Selection
MacBook Pro

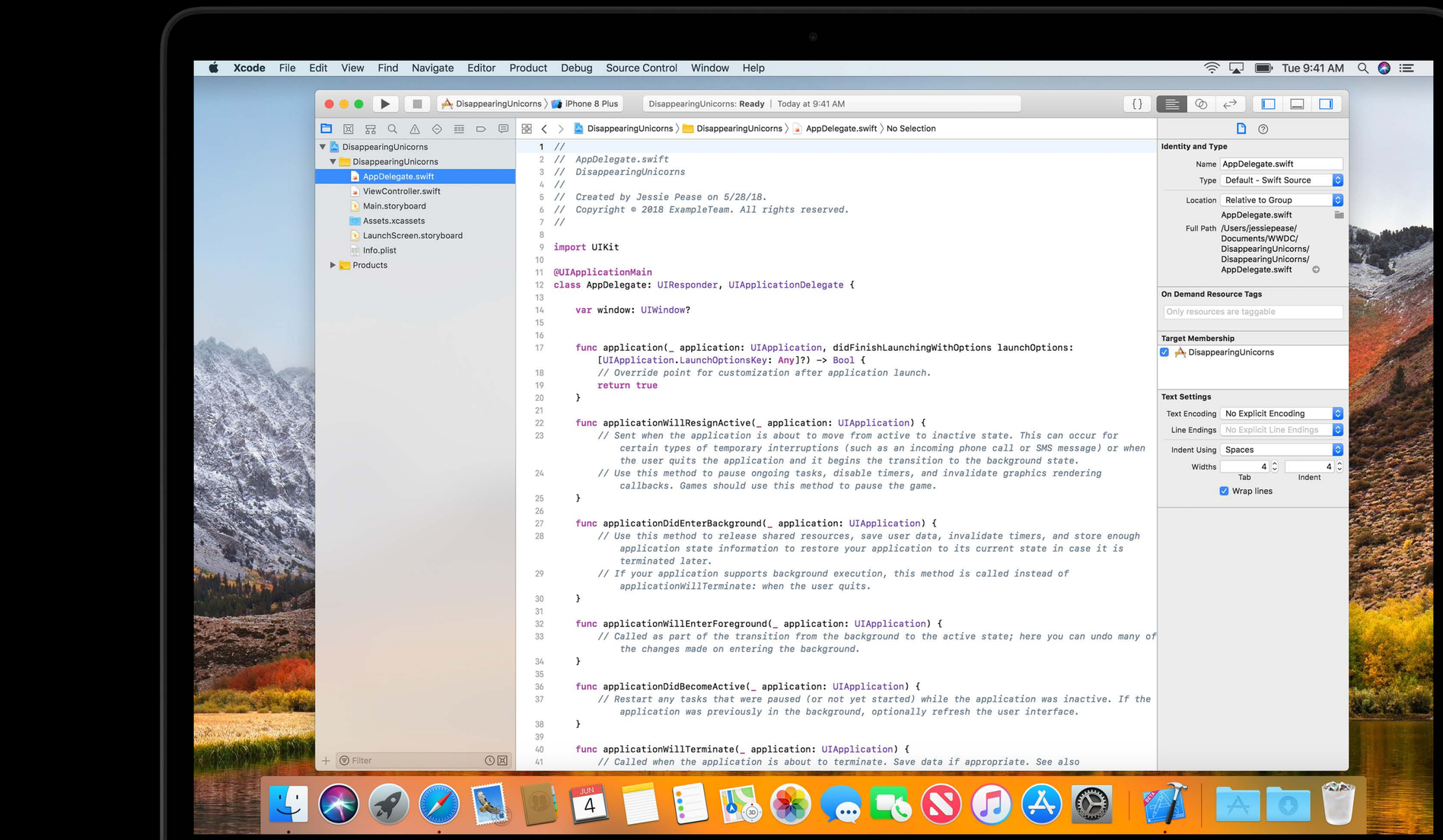

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Choose a template for your new project:
iOS watchOS tvOS macOS Cross-platform
Filter
Application
1
Single View App
Game
AR
Augmented Reality App
Document Based App
Master-Detail App
Page-Based App
Tabbed App
Sticker Pack App
iMessage App
Framework & Library
Cocoa Touch Framework
Cocoa Touch Static Library
Metal Library
Cancel
Previous
Next
No Selection
MacBook Pro

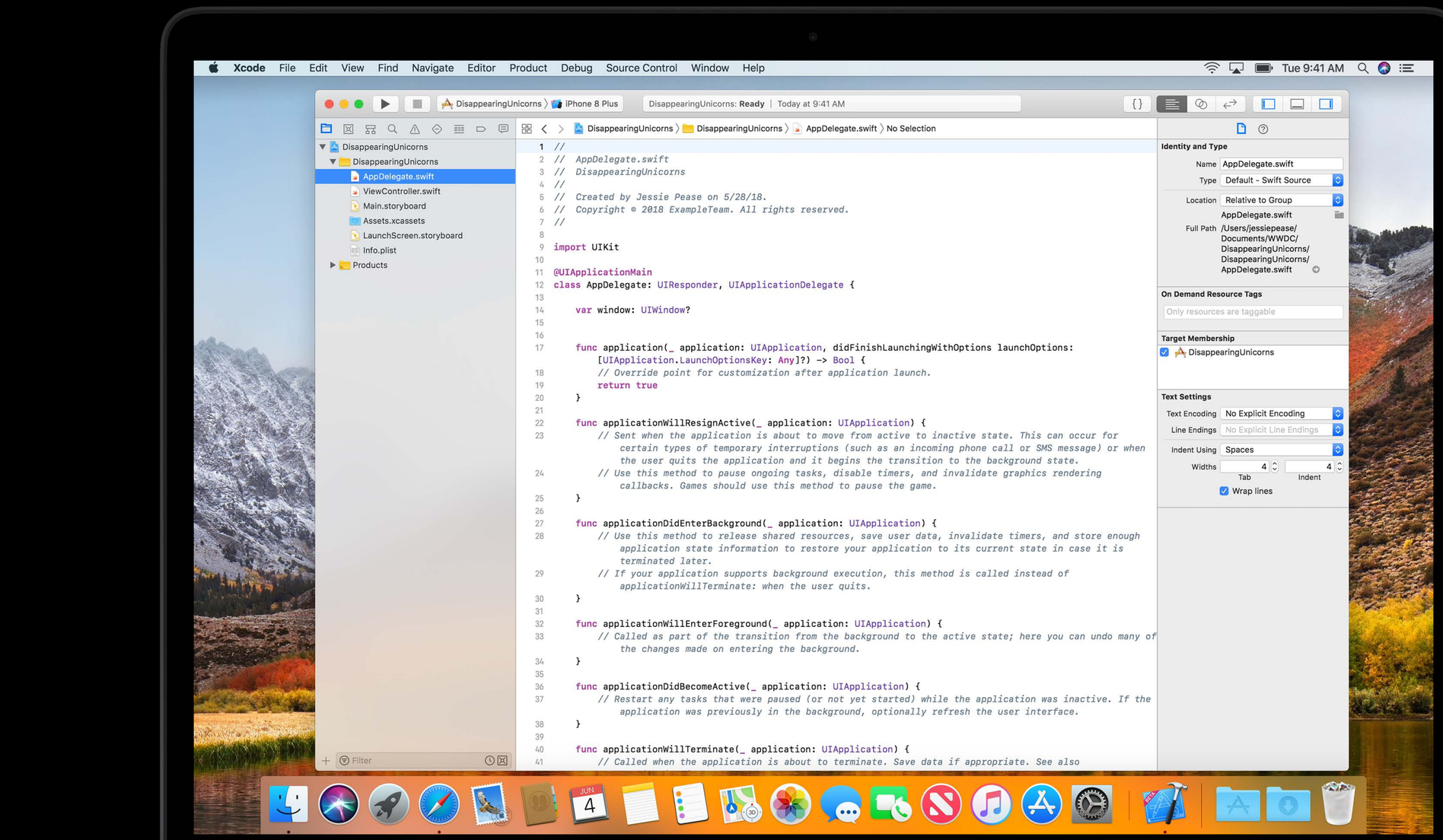

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Choose options for your new project:
Product Name:
Team:
Add account...
Organization Name:
Organization Identifier:
com.exampleteam
Bundle Identifier:
com.exampleteam.DisappearingUnicorns
Language:
Swift
Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next
No Selection
MacBook Pro

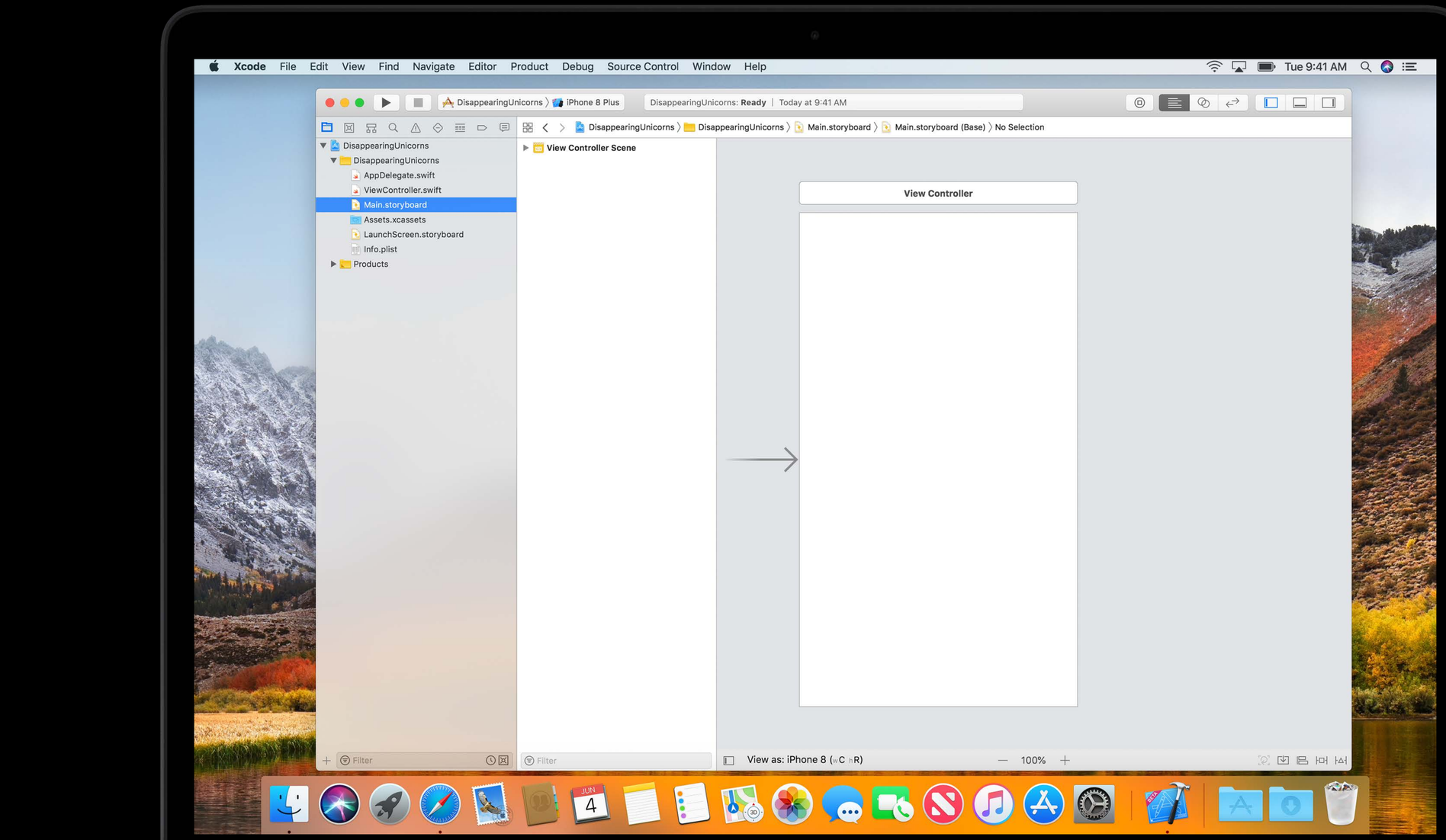

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
Choose options for your new project:
Product Name: DisappearingUnicorns
Team: Add account...
Organization Name: ExampleTeam
Organization Identifier: com.exampleteam
Bundle Identifier: com.exampleteam.DisappearingUnicorns
Language: Swift
Use Core Data
Include Unit Tests
Include UI Tests
Cancel
Previous
Next
No Selection
MacBook Pro

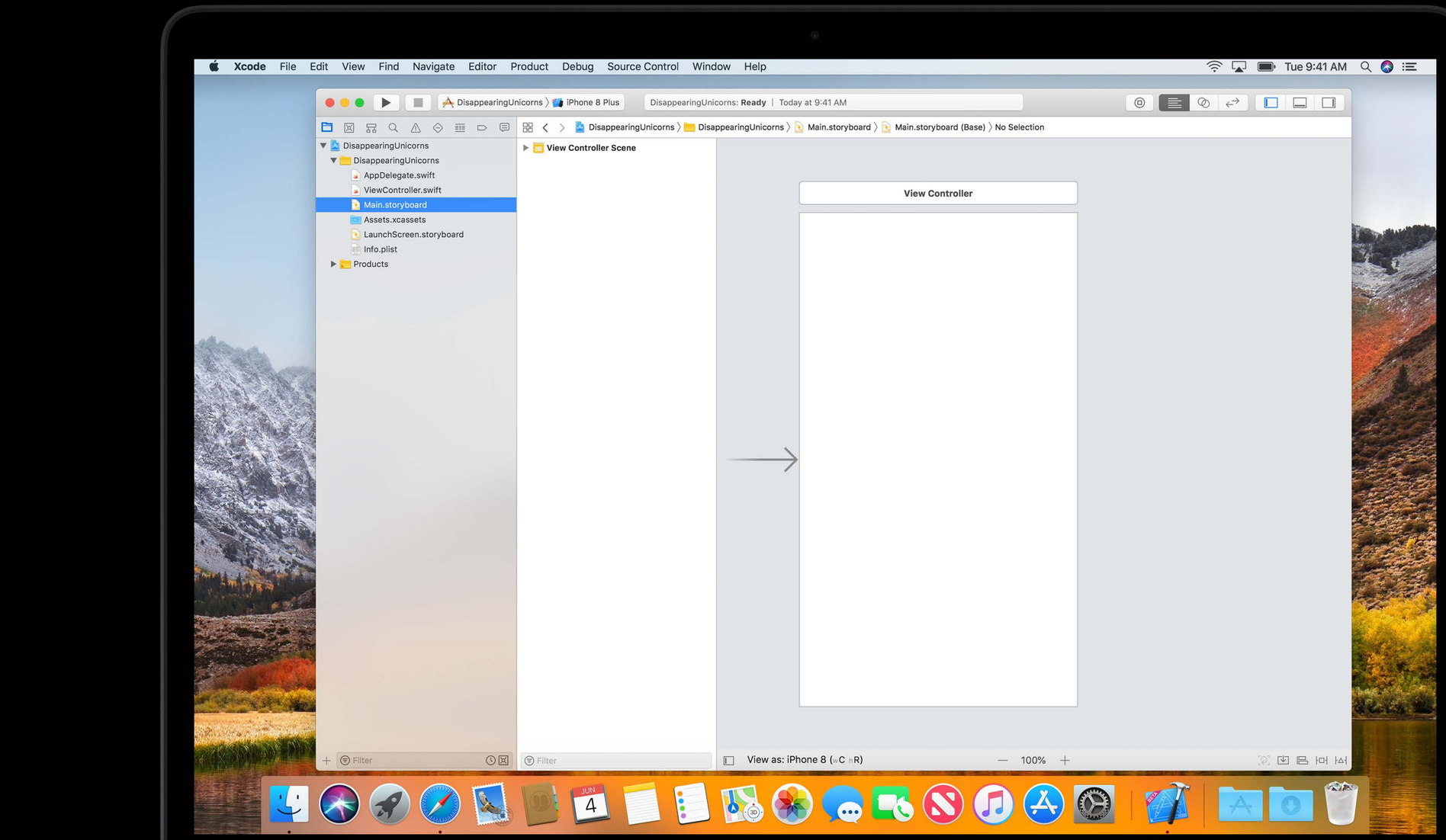

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name DisappearingUnicorns
Bundle Identifier com.exampleteam.DisappearingUnicorns
Version 1.0
Build 1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team Add Account...
Provisioning Profile Xcode Managed Profile
Signing Certificate iOS Developer
Status Signing for "DisappearingUnicorns" requires a development team.
Select a development team in the project editor.
Deployment Info
Deployment Target 12.0
Devices Universal
Main Interface Main
Device Orientation Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source AppIcon
Launch Images Source Use Asset Catalog...
Identity and Type
Name DisappearingUnicorns
Location Absolute
DisappearingUnicorns.xcodeproj
Full Path /Users/jessiepease/Documents/WWDC/DisappearingUnicorns/DisappearingUnicorns.xcodeproj
Project Document
Project Format Xcode 9.3-compatible
Organization ExampleTeam
Class Prefix
Text Settings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines
Filter
MacBook Pro

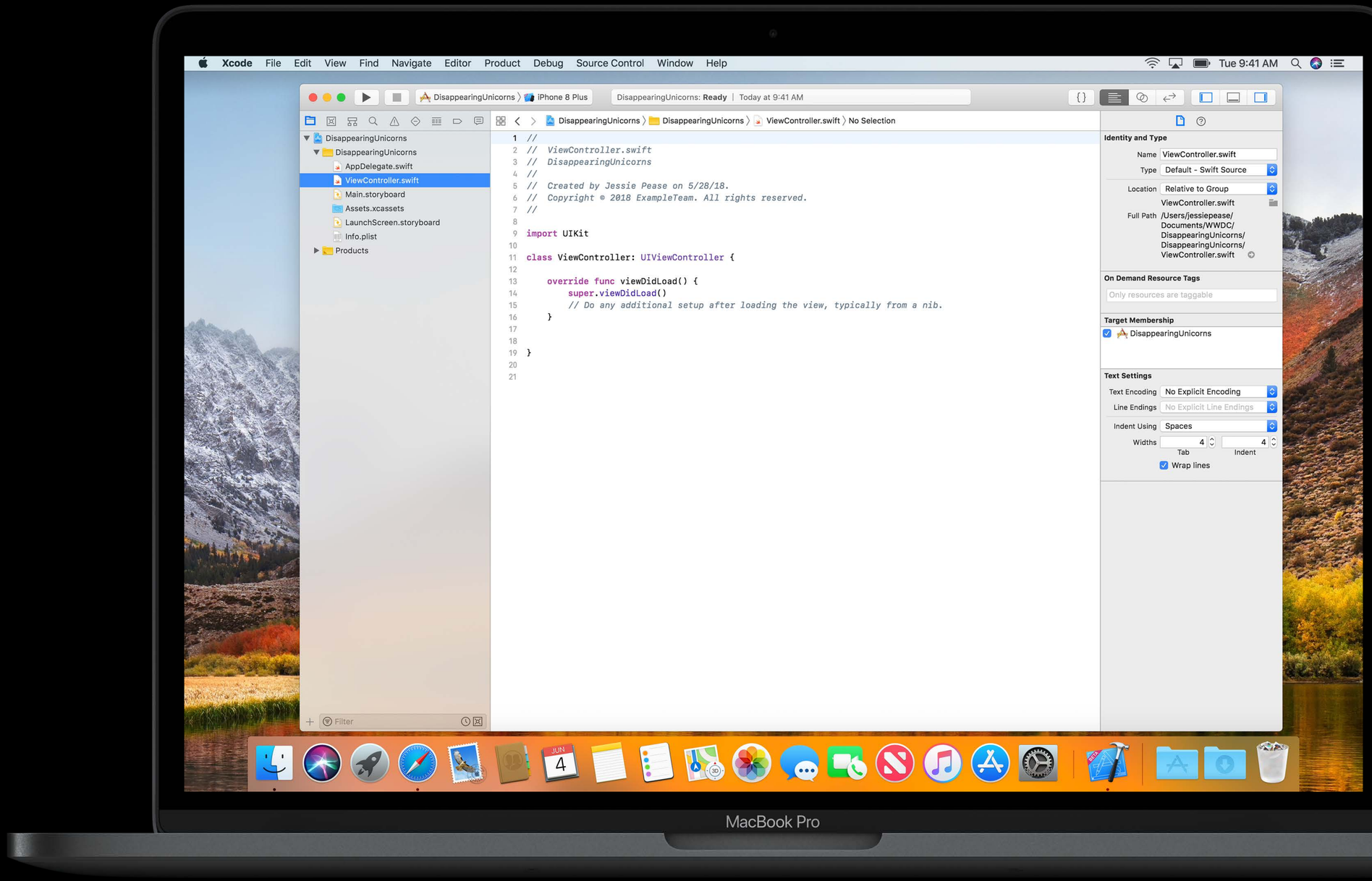

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
General
Capabilities
Resource Tags
Info
Build Settings
Build Phases
Build Rules
Identity
Display Name DisappearingUnicorns
Bundle Identifier com.exampleteam.DisappearingUnicorns
Version 1.0
Build 1
Signing
Automatically manage signing
Xcode will create and update profiles, app IDs, and certificates.
Team Add Account...
Provisioning Profile Xcode Managed Profile
Signing Certificate iOS Developer
Status Signing for "DisappearingUnicorns" requires a development team.
Select a development team in the project editor.
Deployment Info
Deployment Target 12.0
Devices Universal
Main Interface Main
Device Orientation Portrait
Upside Down
Landscape Left
Landscape Right
Status Bar Style Default
Hide status bar
Requires full screen
App Icons and Launch Images
App Icons Source AppIcon
Launch Images Source Use Asset Catalog...
Identity and Type
Name DisappearingUnicorns
Location Absolute
DisappearingUnicorns.xcodeproj
Full Path /Users/jessiepease/Documents/WWDC/DisappearingUnicorns/DisappearingUnicorns.xcodeproj
Project Document
Project Format Xcode 9.3-compatible
Organization ExampleTeam
Class Prefix
Text Settings
Indent Using Spaces
Widths 4 4
Tab Indent
Wrap lines
Filter
MacBook Pro


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 AppDelegate.swift 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Filter
//
// AppDelegate.swift
// DisappearingUnicorns
//
// Created by Jessie Pease on 5/28/18.
// Copyright © 2018 ExampleTeam. All rights reserved.
//
import UIKit
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
var window: UIWindow?
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
// Override point for customization after application launch.
return true
}
func applicationWillResignActive(_ application: UIApplication) {
// Sent when the application is about to move from active to inactive state. This can occur for
certain types of temporary interruptions (such as an incoming phone call or SMS message) or when
the user quits the application and it begins the transition to the background state.
// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering
callbacks. Games should use this method to pause the game.
}
func applicationDidEnterBackground(_ application: UIApplication) {
// Use this method to release shared resources, save user data, invalidate timers, and store enough
application state information to restore your application to its current state in case it is
terminated later.
// If your application supports background execution, this method is called instead of
applicationWillTerminate: when the user quits.
}
func applicationWillEnterForeground(_ application: UIApplication) {
// Called as part of the transition from the background to the active state; here you can undo many of
the changes made on entering the background.
}
func applicationDidBecomeActive(_ application: UIApplication) {
// Restart any tasks that were paused (or not yet started) while the application was inactive. If the
application was previously in the background, optionally refresh the user interface.
}
func applicationWillTerminate(_ application: UIApplication) {
// Called when the application is about to terminate. Save data if appropriate. See also
Identity and Type
Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/jessiepease/Documents/WWDC/DisappearingUnicorns/DisappearingUnicorns/AppDelegate.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
DisappearingUnicorns
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines
MacBook Pro

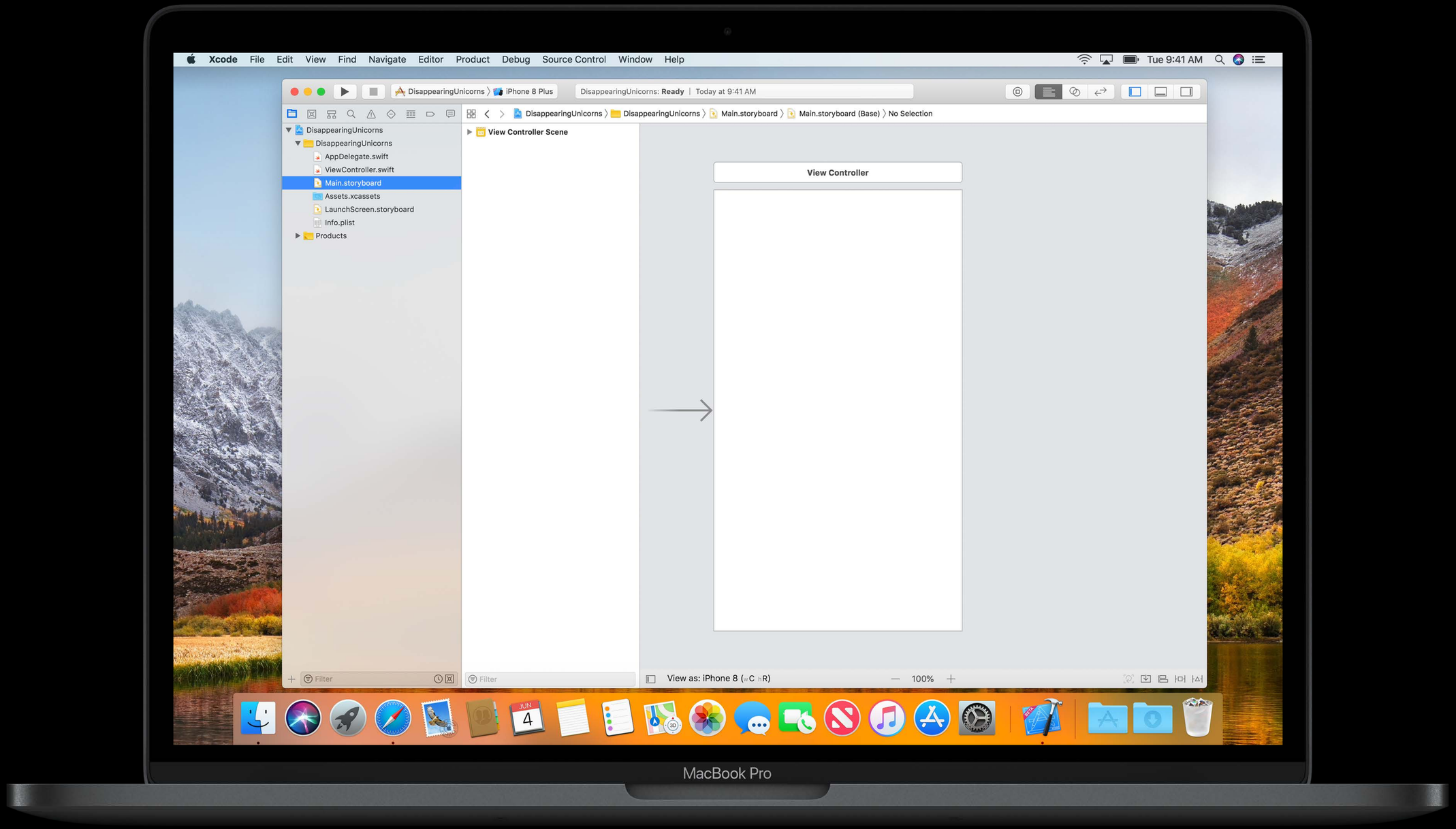

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 AppDelegate.swift 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Filter
//
// AppDelegate.swift
// DisappearingUnicorns
//
// Created by Jessie Pease on 5/28/18.
// Copyright © 2018 ExampleTeam. All rights reserved.
//
import UIKit
@UIApplicationMain
class AppDelegate: UIResponder, UIApplicationDelegate {
var window: UIWindow?
func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
// Override point for customization after application launch.
return true
}
func applicationWillResignActive(_ application: UIApplication) {
// Sent when the application is about to move from active to inactive state. This can occur for
certain types of temporary interruptions (such as an incoming phone call or SMS message) or when
the user quits the application and it begins the transition to the background state.
// Use this method to pause ongoing tasks, disable timers, and invalidate graphics rendering
callbacks. Games should use this method to pause the game.
}
func applicationDidEnterBackground(_ application: UIApplication) {
// Use this method to release shared resources, save user data, invalidate timers, and store enough
application state information to restore your application to its current state in case it is
terminated later.
// If your application supports background execution, this method is called instead of
applicationWillTerminate: when the user quits.
}
func applicationWillEnterForeground(_ application: UIApplication) {
// Called as part of the transition from the background to the active state; here you can undo many of
the changes made on entering the background.
}
func applicationDidBecomeActive(_ application: UIApplication) {
// Restart any tasks that were paused (or not yet started) while the application was inactive. If the
application was previously in the background, optionally refresh the user interface.
}
func applicationWillTerminate(_ application: UIApplication) {
// Called when the application is about to terminate. Save data if appropriate. See also
Identity and Type
Name AppDelegate.swift
Type Default - Swift Source
Location Relative to Group
AppDelegate.swift
Full Path /Users/jessiepease/Documents/WWDC/DisappearingUnicorns/DisappearingUnicorns/AppDelegate.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
DisappearingUnicorns
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines
MacBook Pro

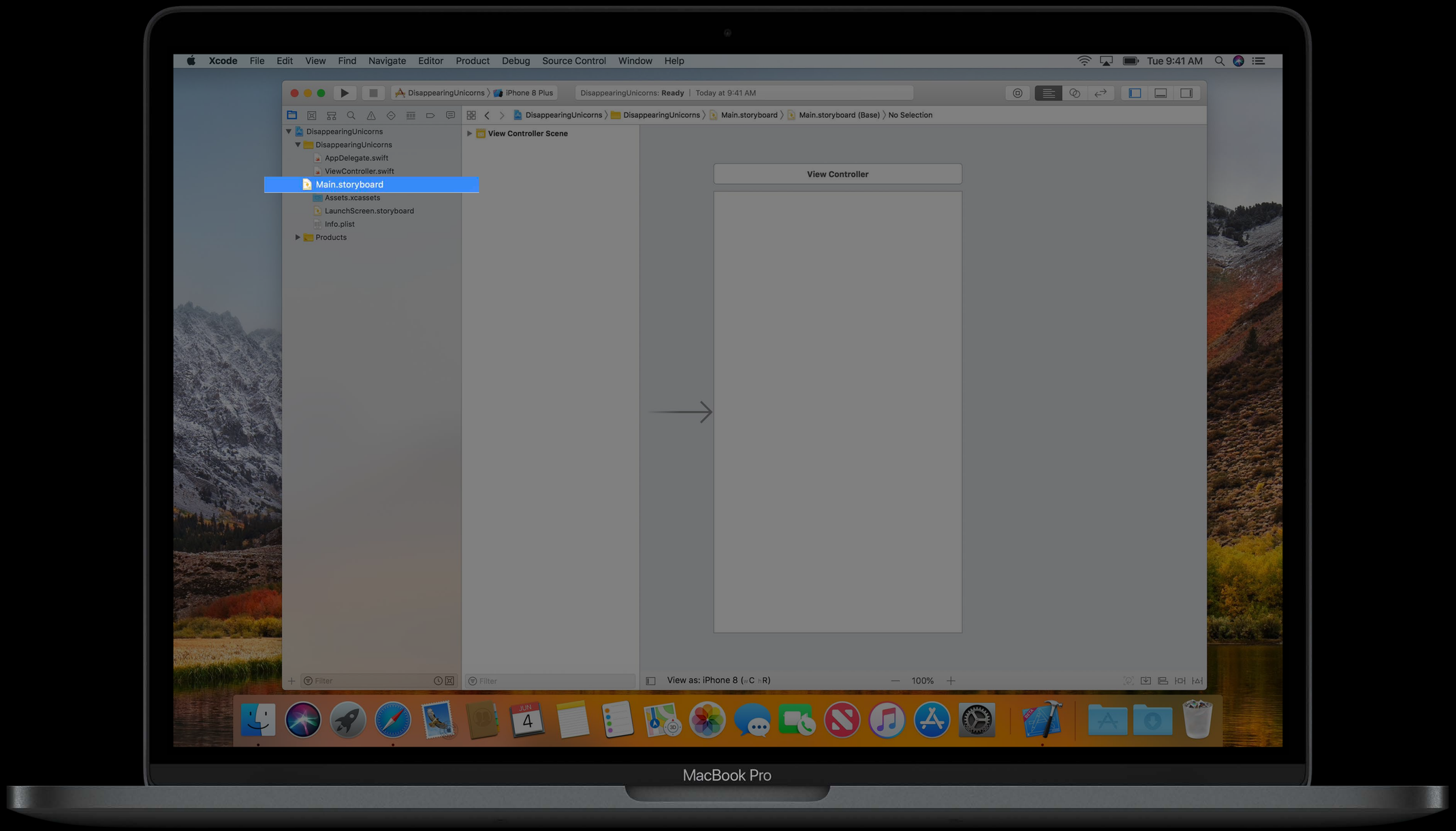

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
JUN
4
MacBook Pro

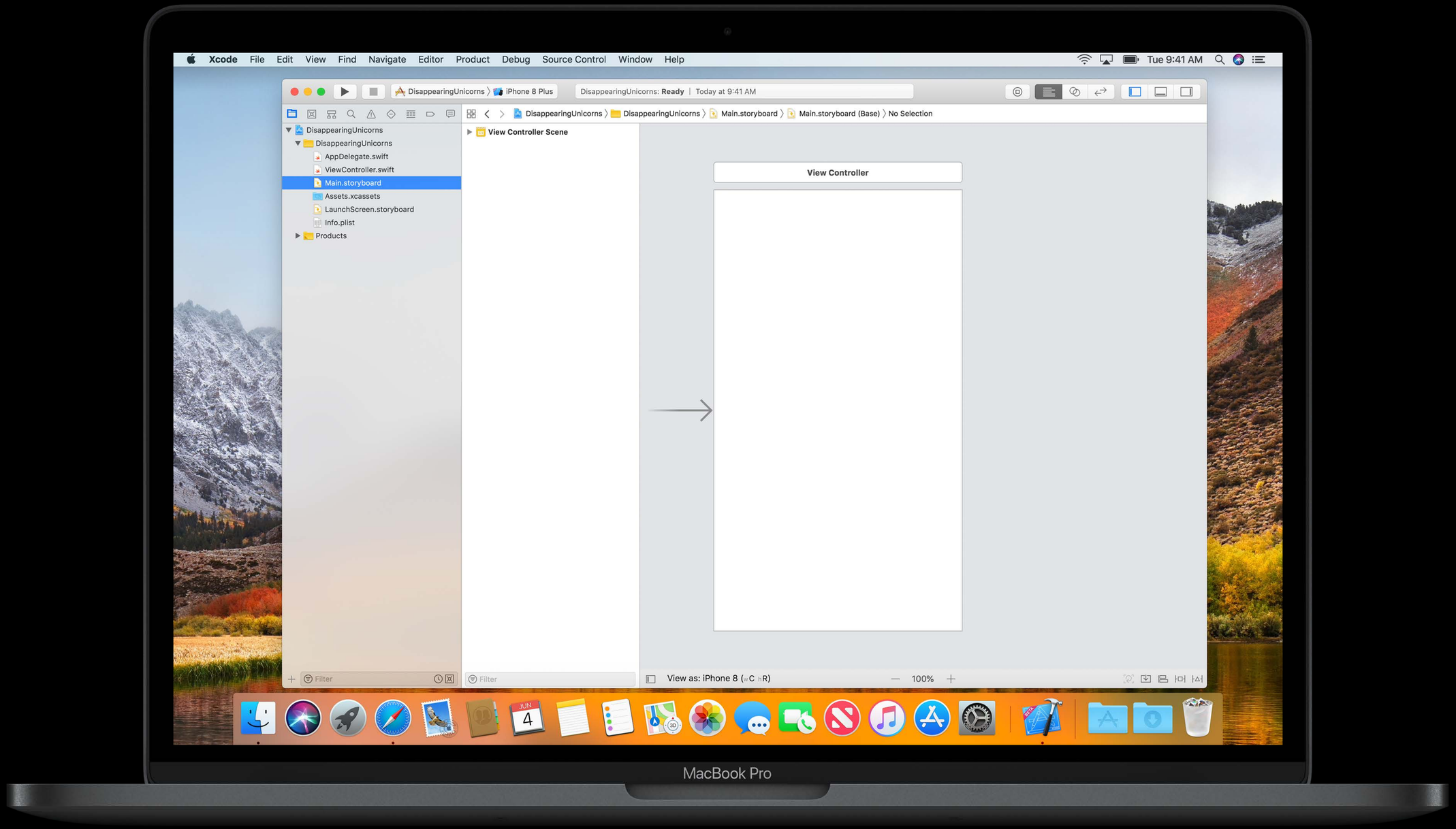

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
JUN
4
MacBook Pro

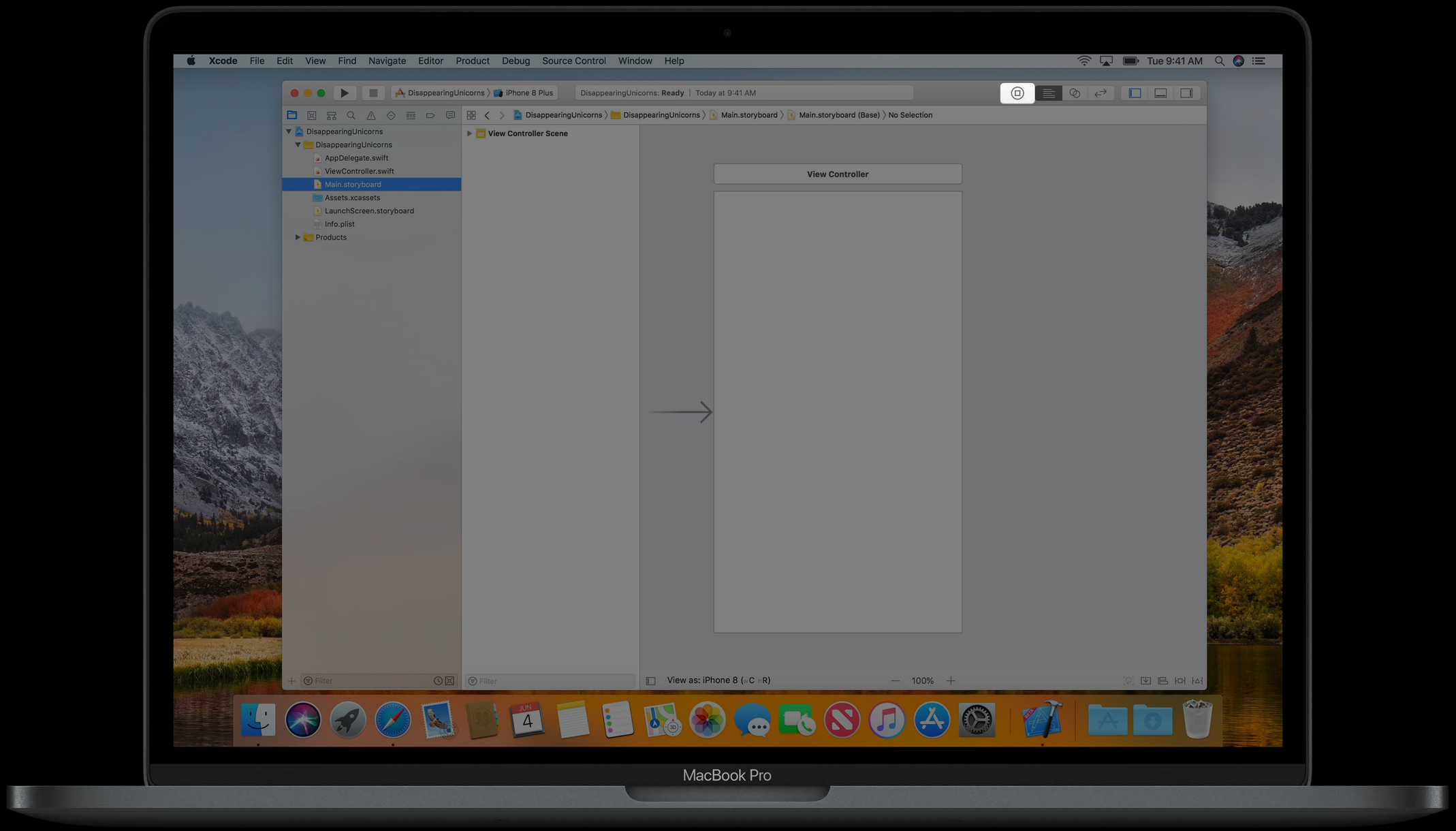

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 ViewController.swift 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
Filter
//
// ViewController.swift
// DisappearingUnicorns
//
// Created by Jessie Pease on 5/28/18.
// Copyright © 2018 ExampleTeam. All rights reserved.
//
import UIKit
class ViewController: UIViewController {
override func viewDidLoad() {
super.viewDidLoad()
// Do any additional setup after loading the view, typically from a nib.
}
}
Identity and Type
Name ViewController.swift
Type Default - Swift Source
Location Relative to Group
ViewController.swift
Full Path /Users/jessiepease/Documents/WWDC/DisappearingUnicorns/DisappearingUnicorns/ViewController.swift
On Demand Resource Tags
Only resources are taggable
Target Membership
DisappearingUnicorns
Text Settings
Text Encoding No Explicit Encoding
Line Endings No Explicit Line Endings
Indent Using Spaces
Widths 4 Tab 4 Indent
Wrap lines
MacBook Pro

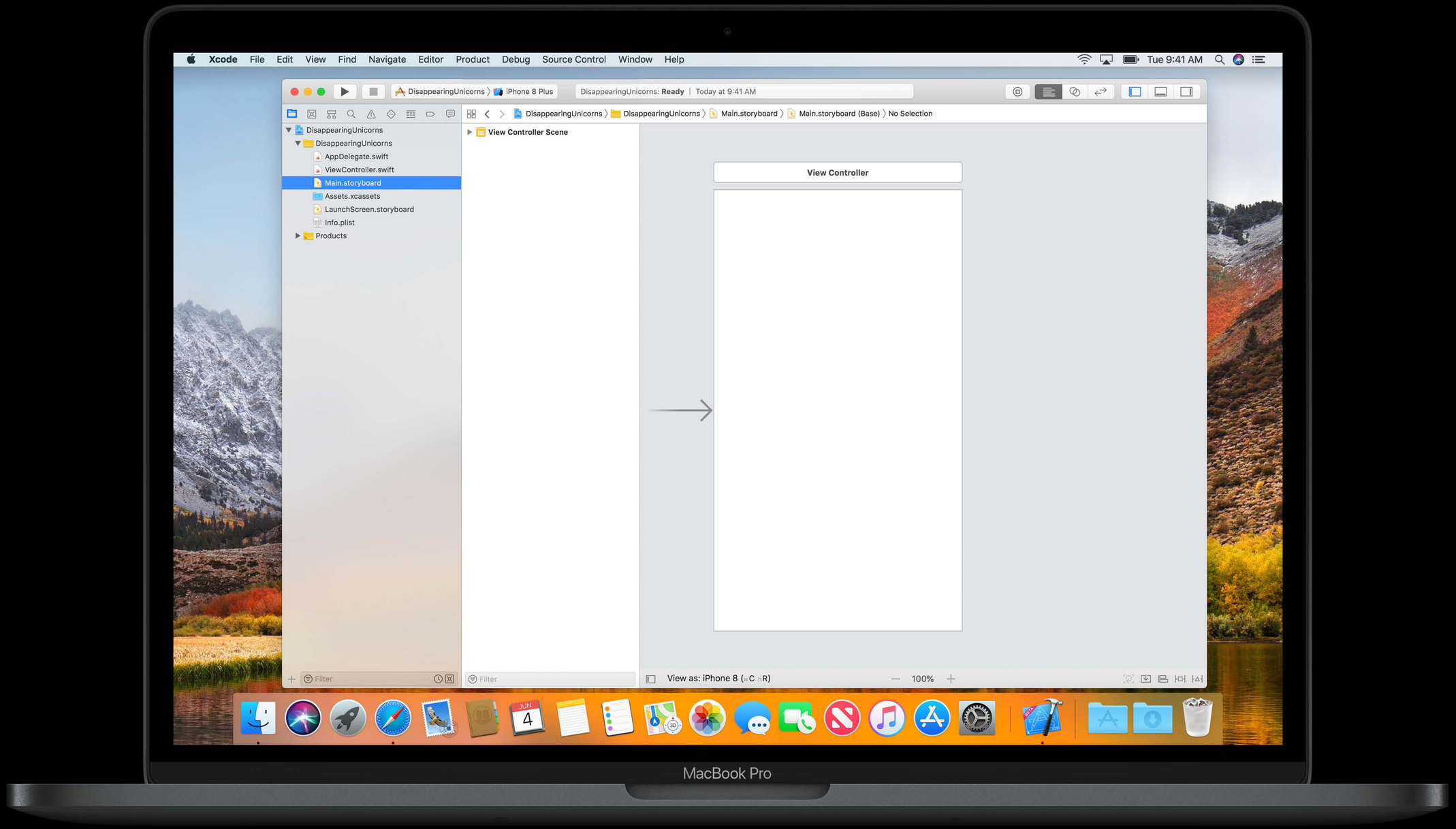
Creating Views with the Storyboard

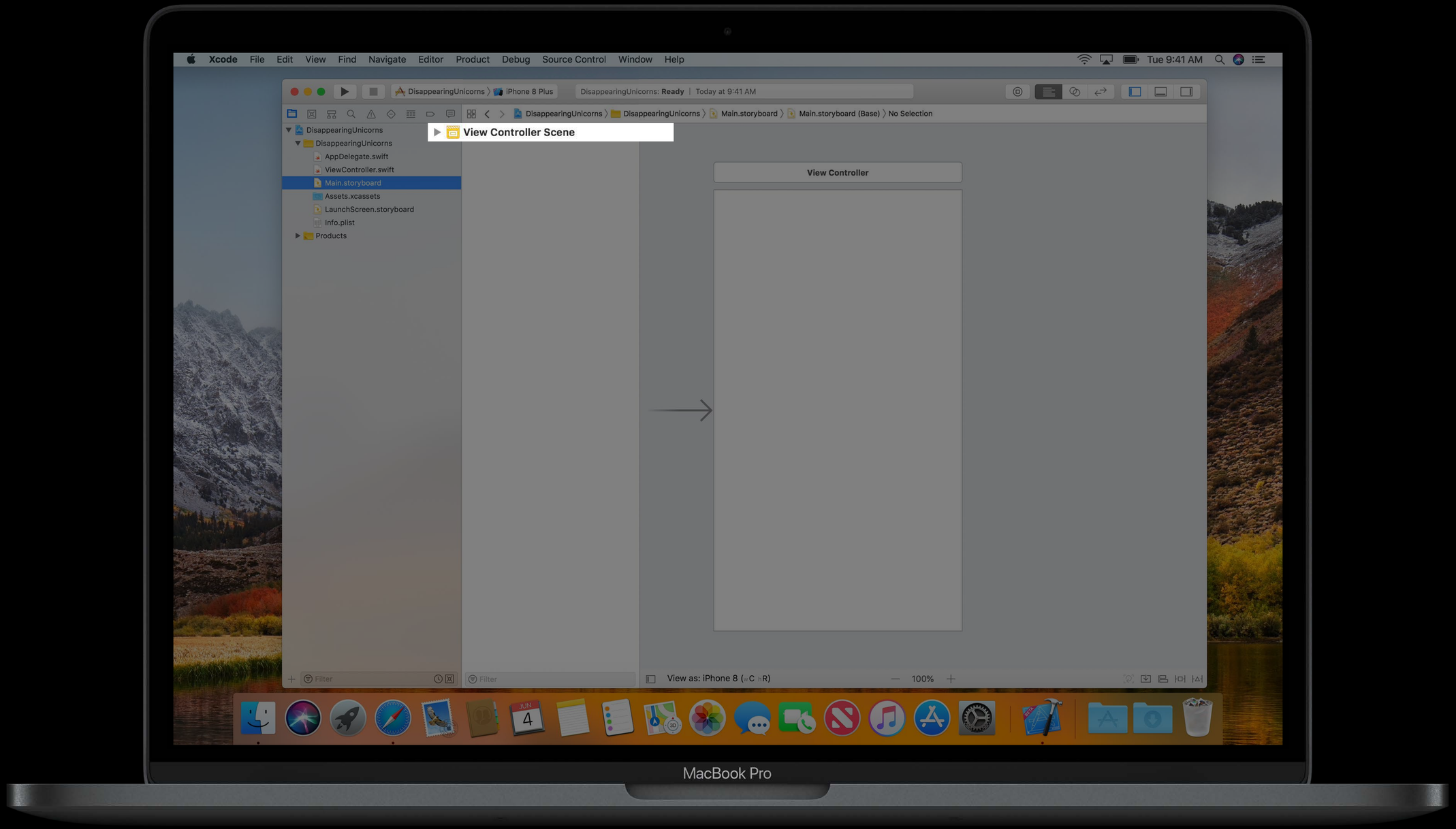

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro

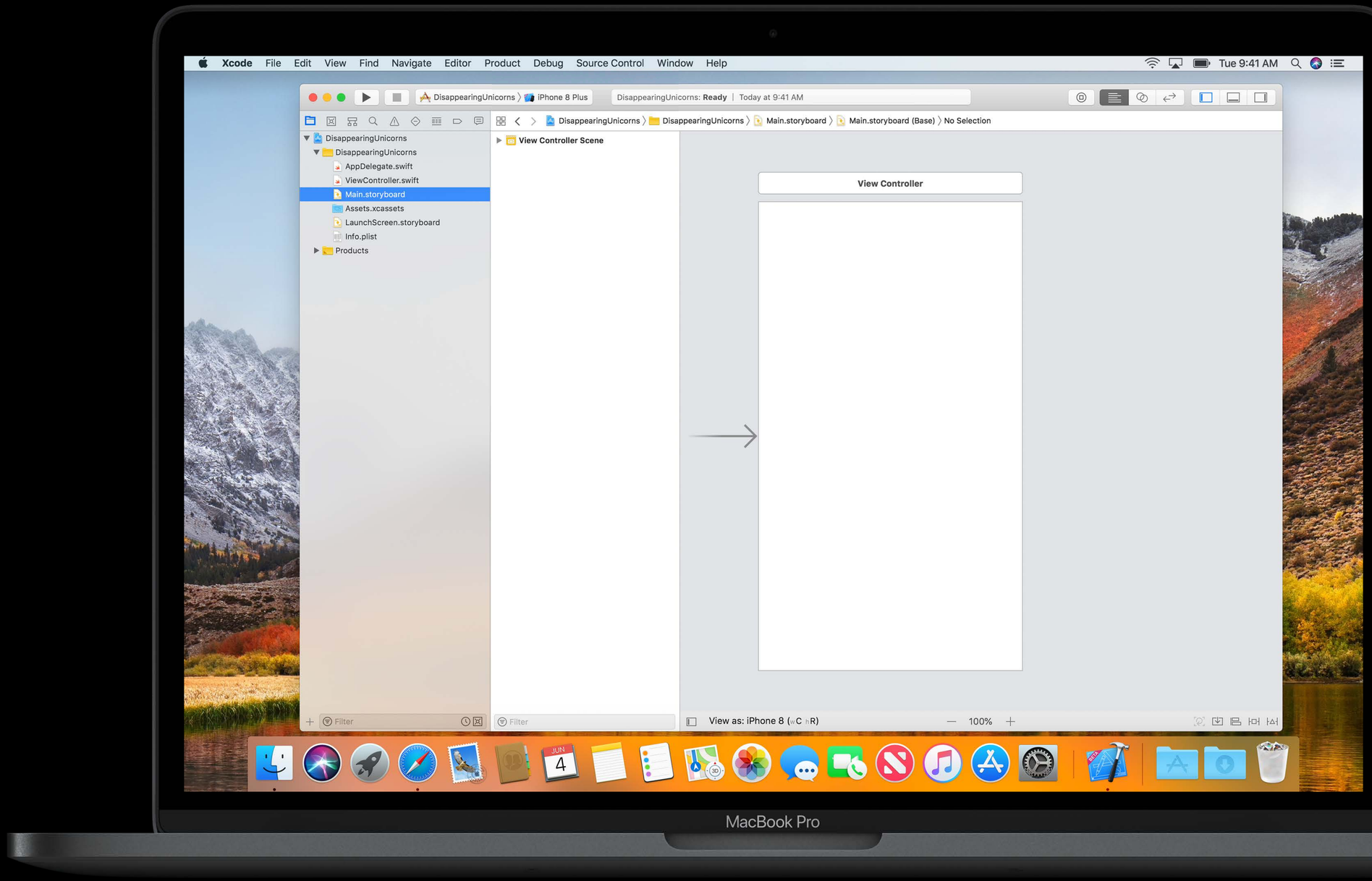

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro

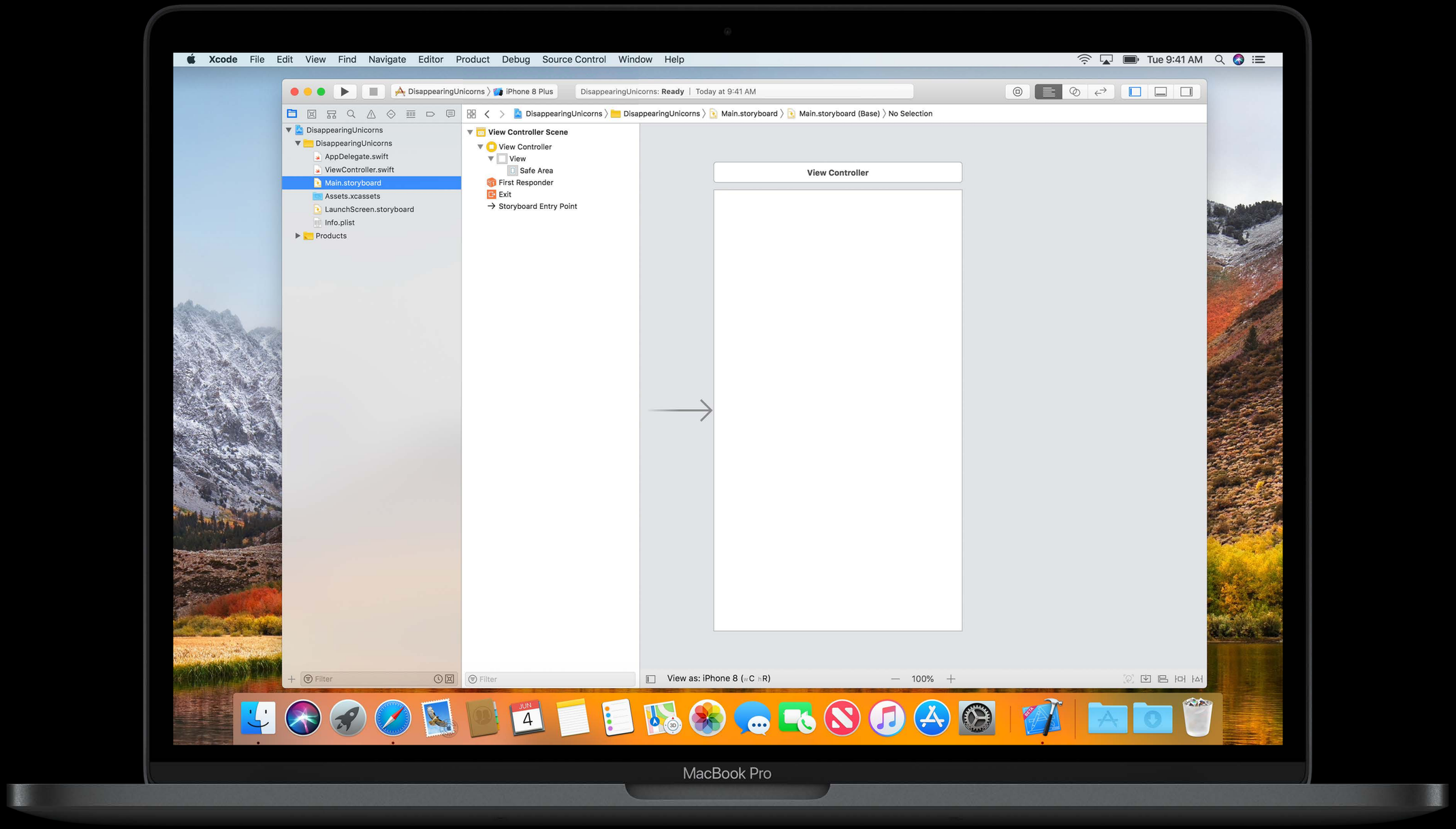

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro


Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro

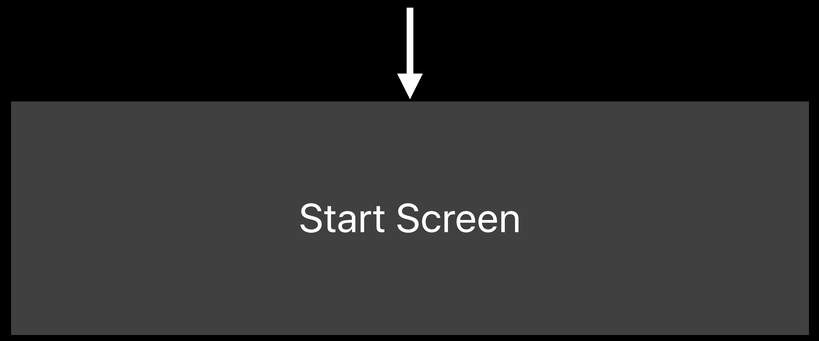

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro

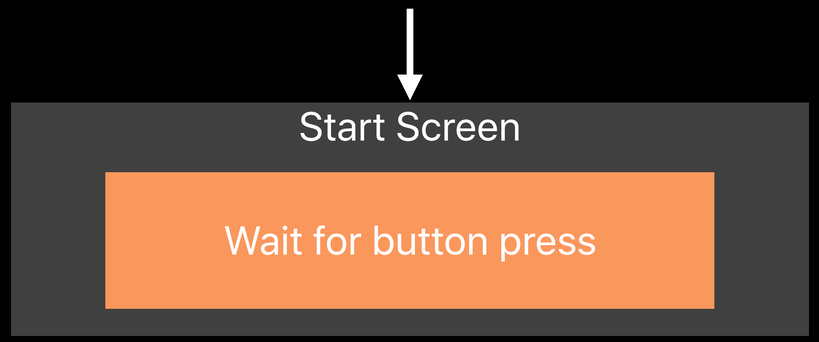

Xcode File Edit View Find Navigate Editor Product Debug Source Control Window Help
Tue 9:41 AM
DisappearingUnicorns 〉 iPhone 8 Plus
DisappearingUnicorns: Ready | Today at 9:41 AM
DisappearingUnicorns 〉 DisappearingUnicorns 〉 Main.storyboard 〉 Main.storyboard (Base) 〉 No Selection
DisappearingUnicorns
DisappearingUnicorns
AppDelegate.swift
ViewController.swift
Main.storyboard
Assets.xcassets
LaunchScreen.storyboard
Info.plist
Products
View Controller Scene
View Controller
View
Safe Area
First Responder
Exit
Storyboard Entry Point
View Controller
Filter
Filter
View as: iPhone 8 (wC hR)
100%
MacBook Pro

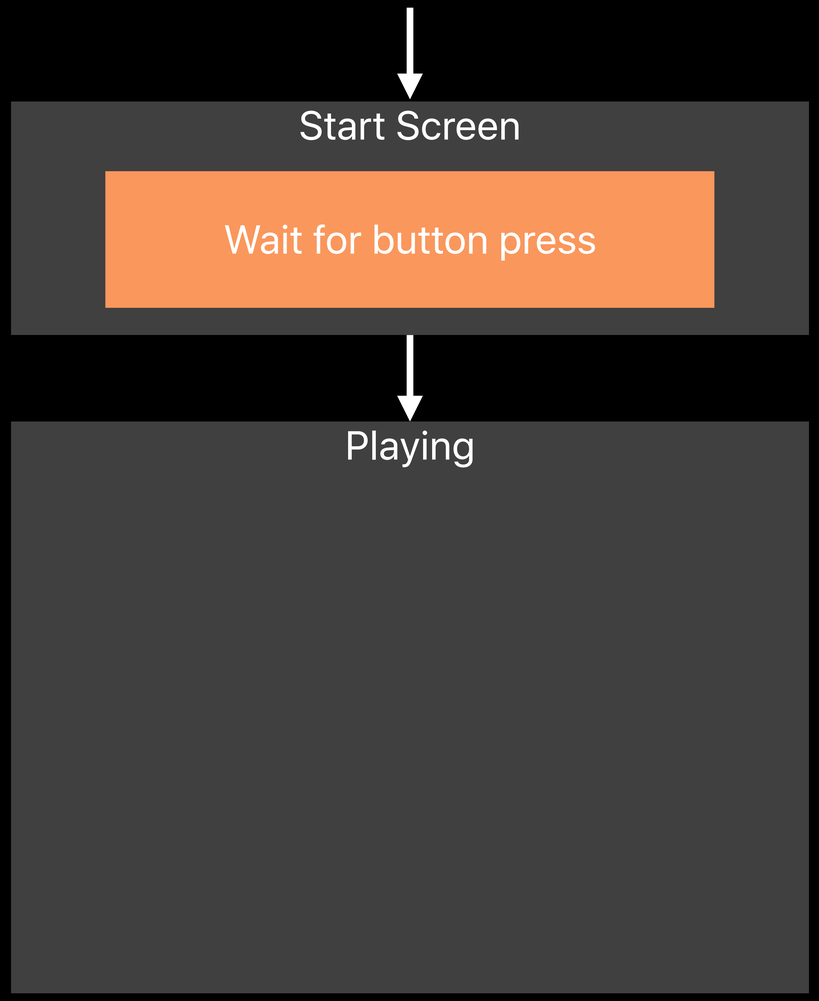
Demo

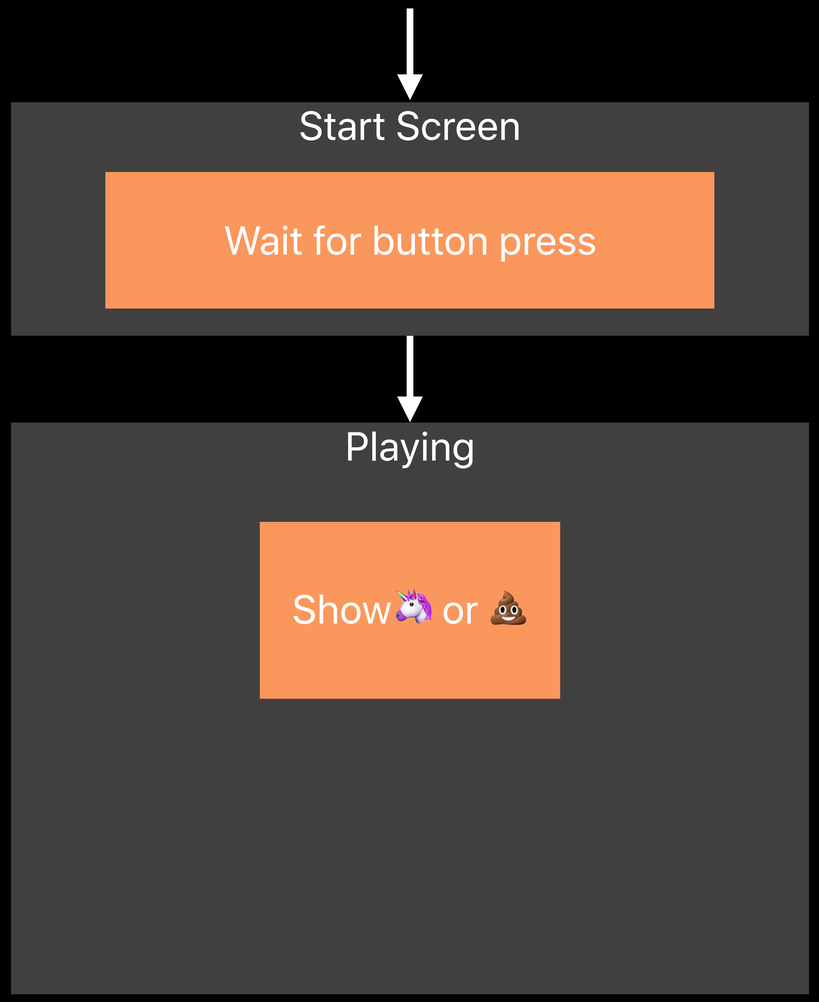
Creating the view for our app

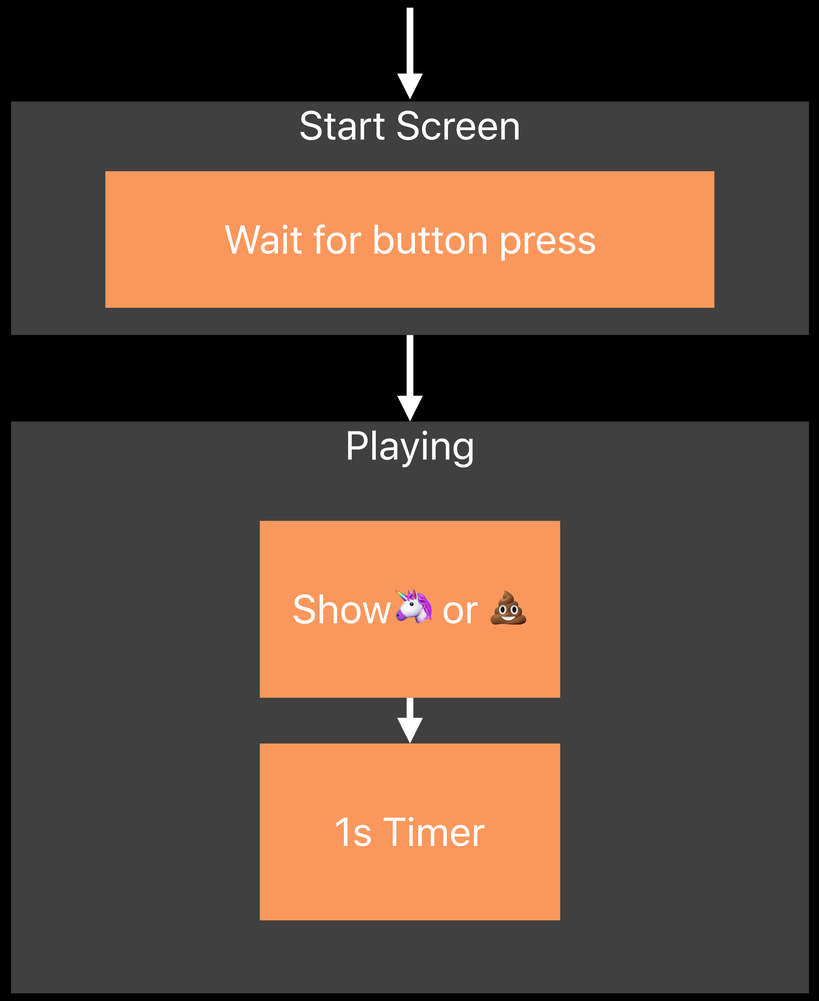
Game Logic

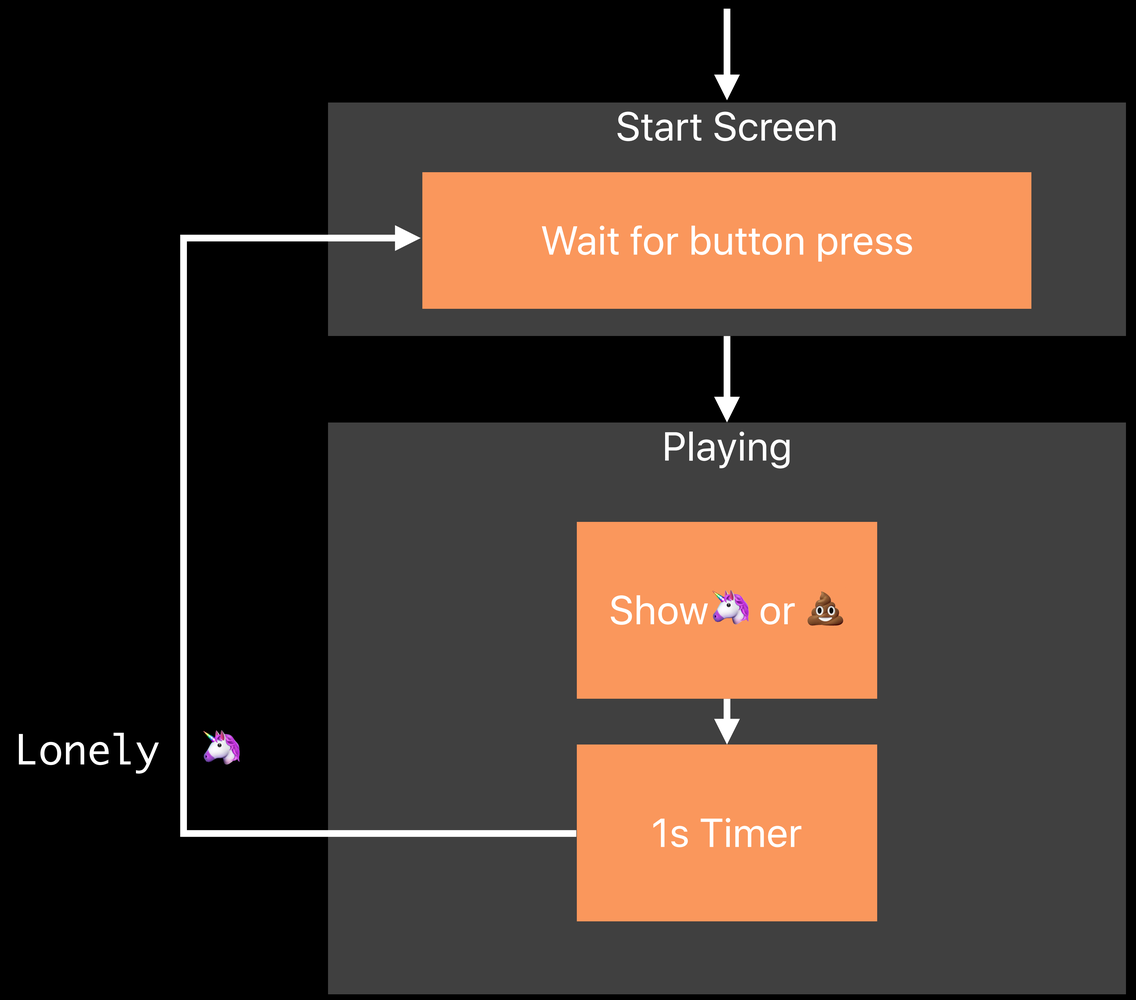

Start Screen

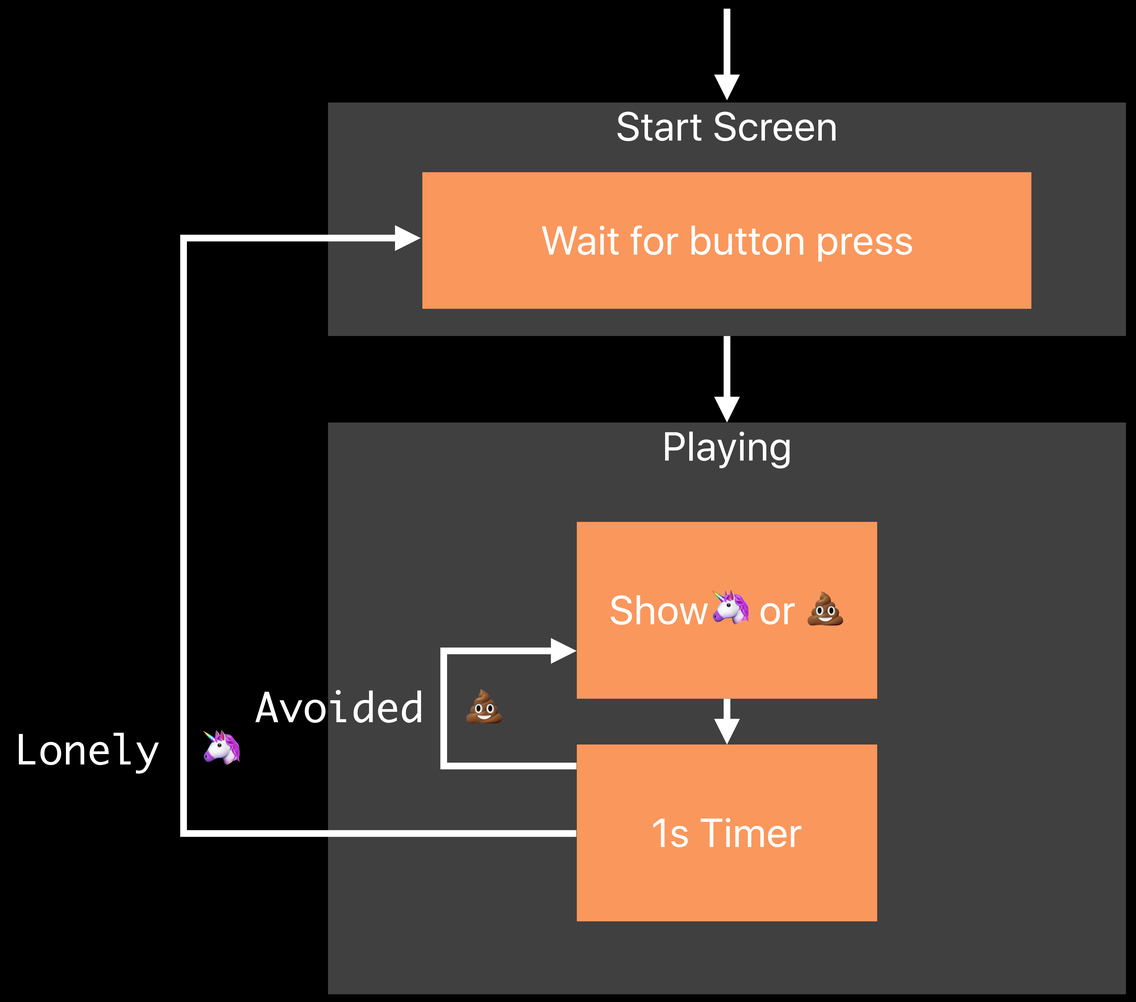

Start Screen
Wait for button press

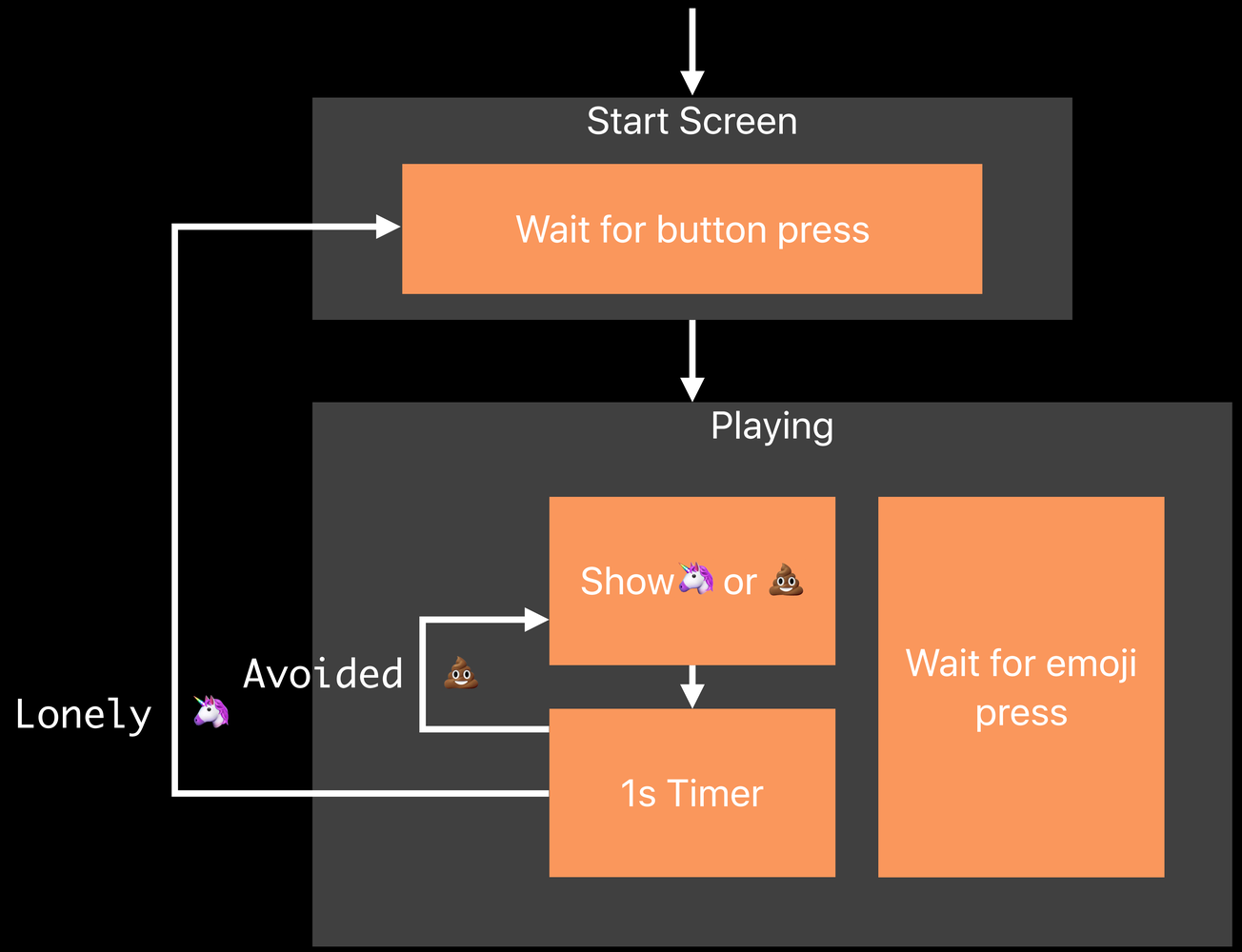

Start Screen
Wait for button press
Playing

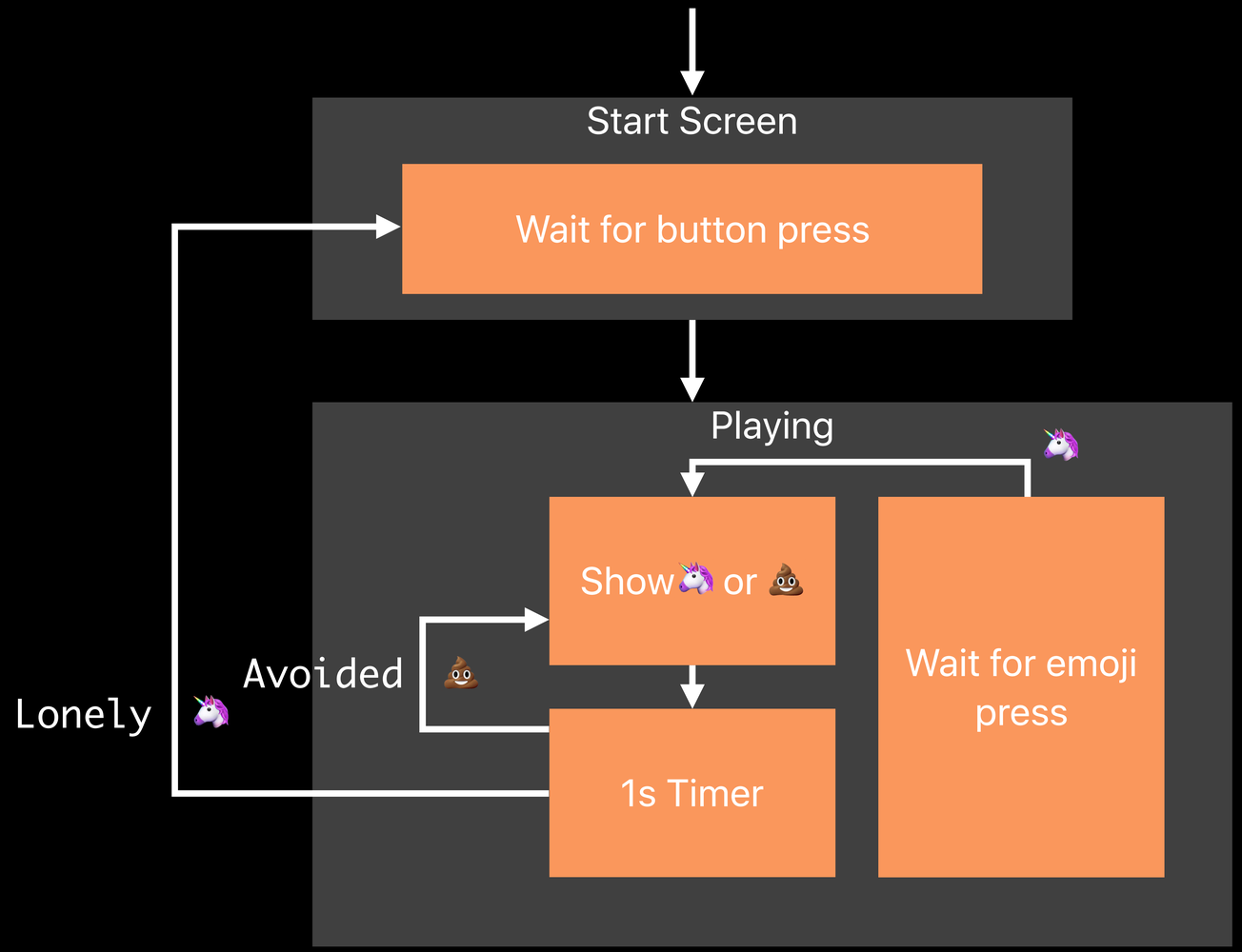

Start Screen
Wait for button press
Playing
Show 🦄 or 💩

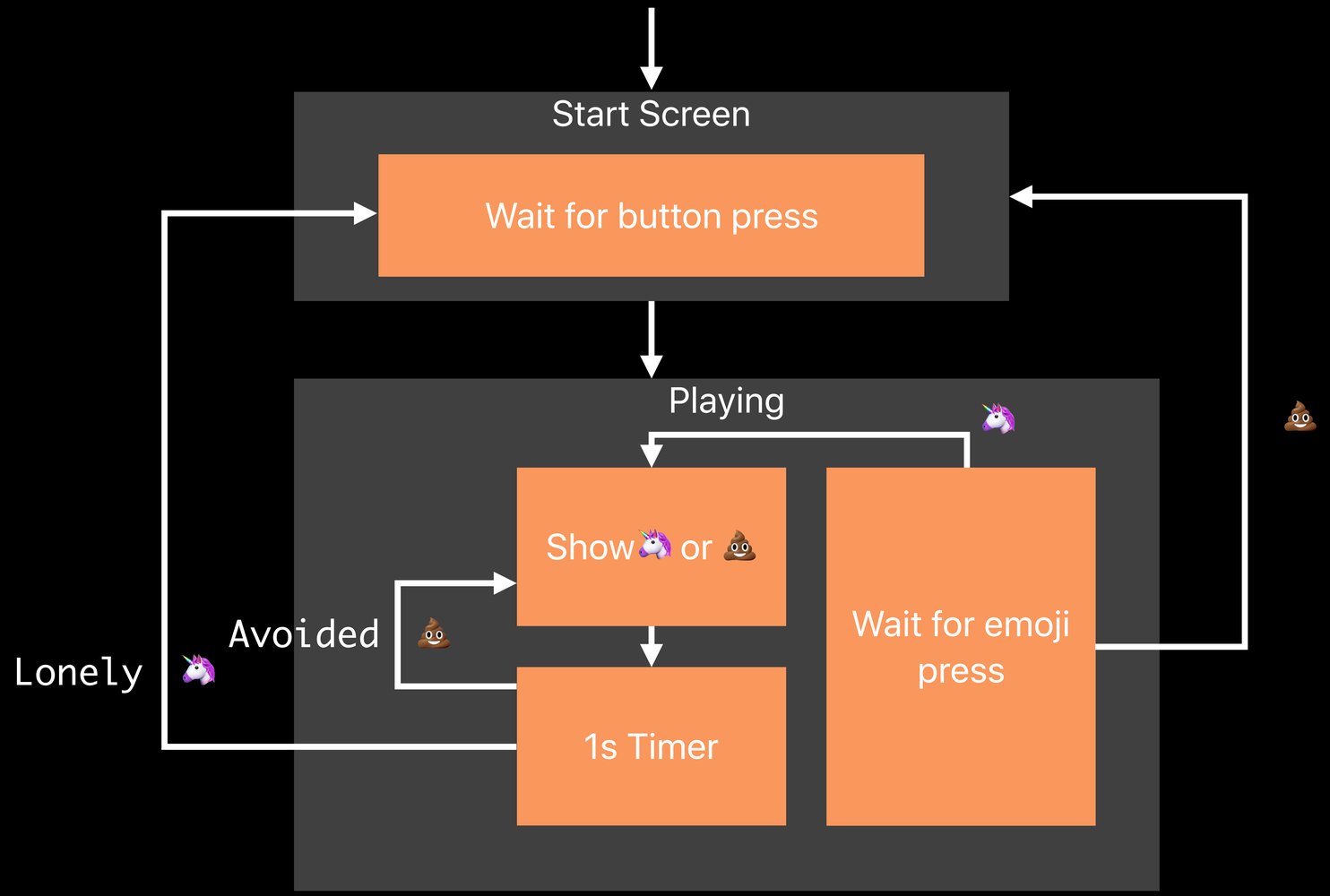

Start Screen
Wait for button press
Playing
Show 🦄 or 💩
1s Timer


Start Screen
Wait for button press
Playing
Show 🦄 or 💩
1s Timer
Lonely 🦄

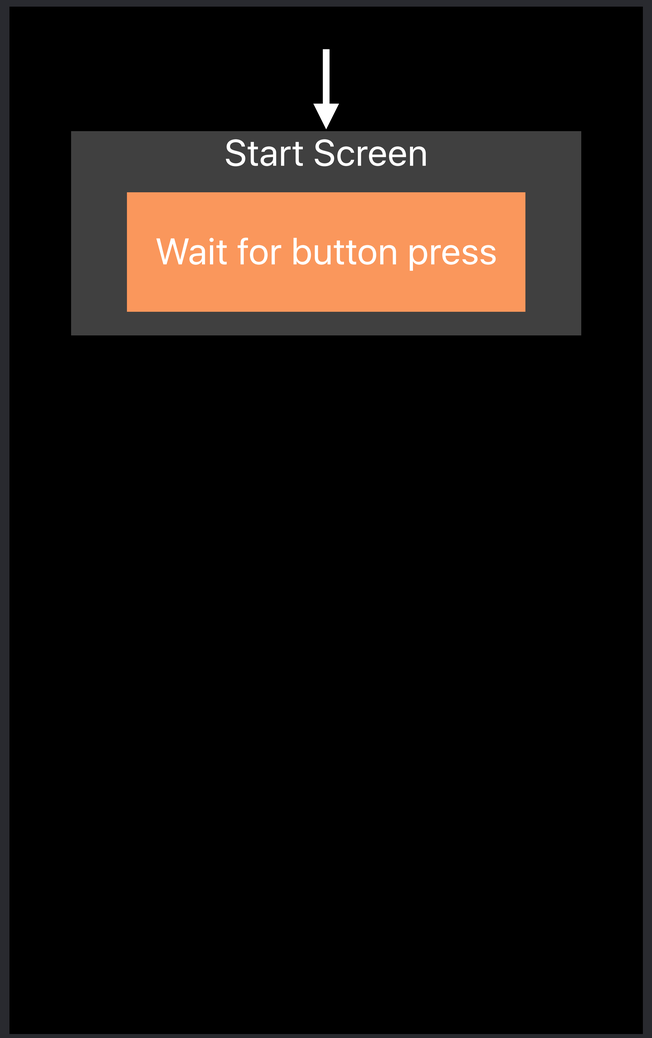

Start Screen
Wait for button press
Playing
Show 🦄 or 💩
Avoided 💩
Lonely 🦄
1s Timer


Start Screen
Wait for button press
Playing
Show 🦄 or 💩
Wait for emoji press
Avoided 💩
Lonely 🦄
1s Timer

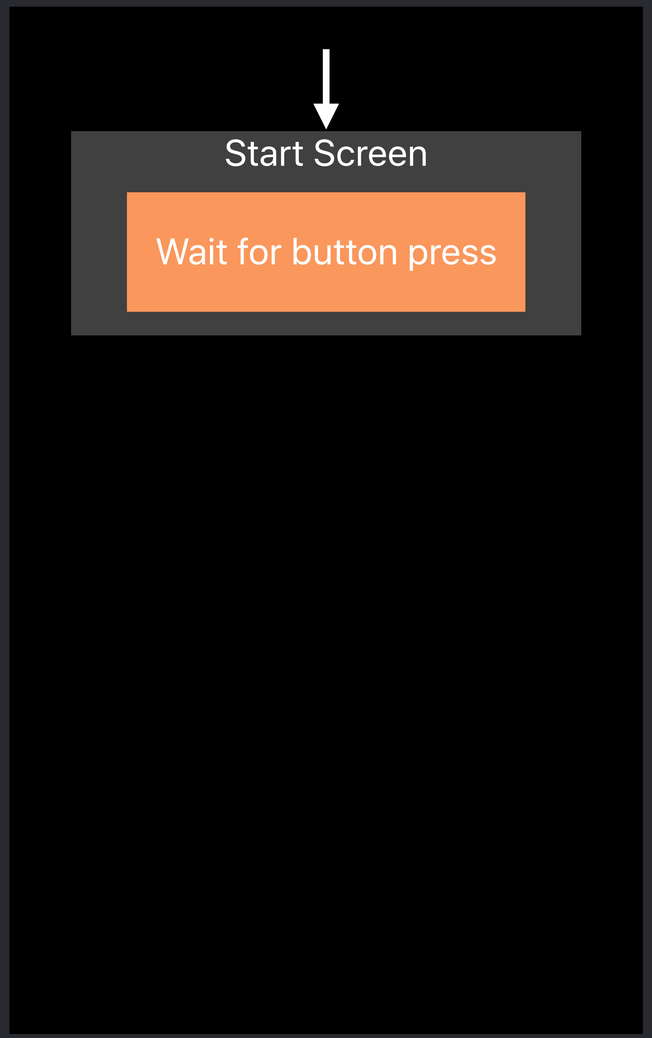

Start Screen
Wait for button press
Playing
Show 🦄 or 💩
Wait for emoji press
1s Timer
Avoided
💩
Lonely
🦄
🦄


Start Screen
Wait for button press
Playing
Show 🦄 or 💩
1s Timer
Wait for emoji press
Avoided
💩
Lonely
🦄
🦄
💩

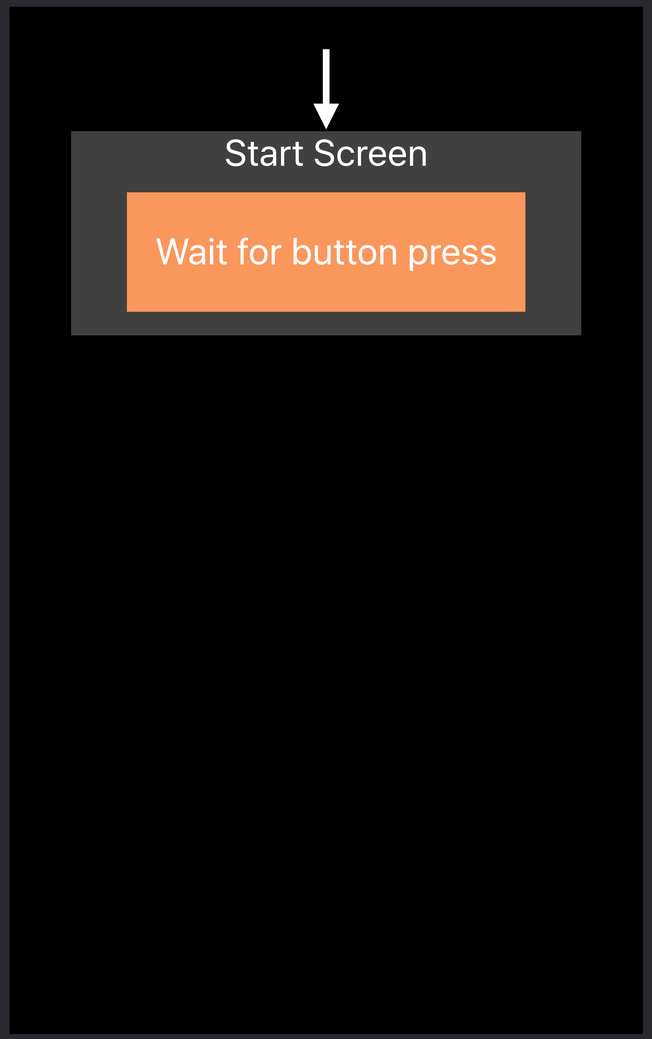
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
```



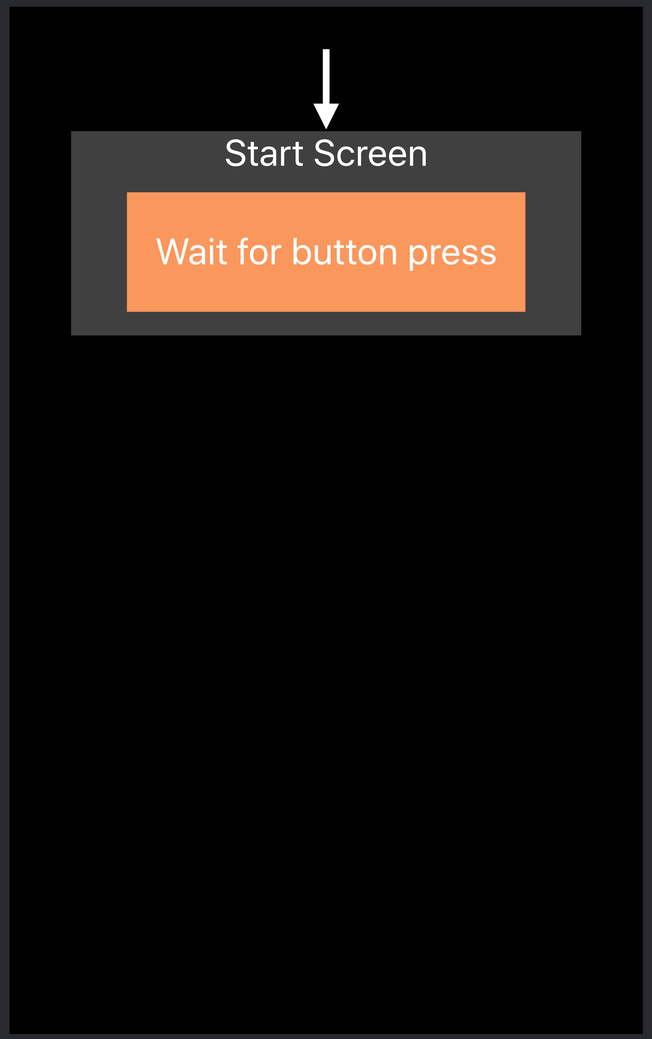
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
```



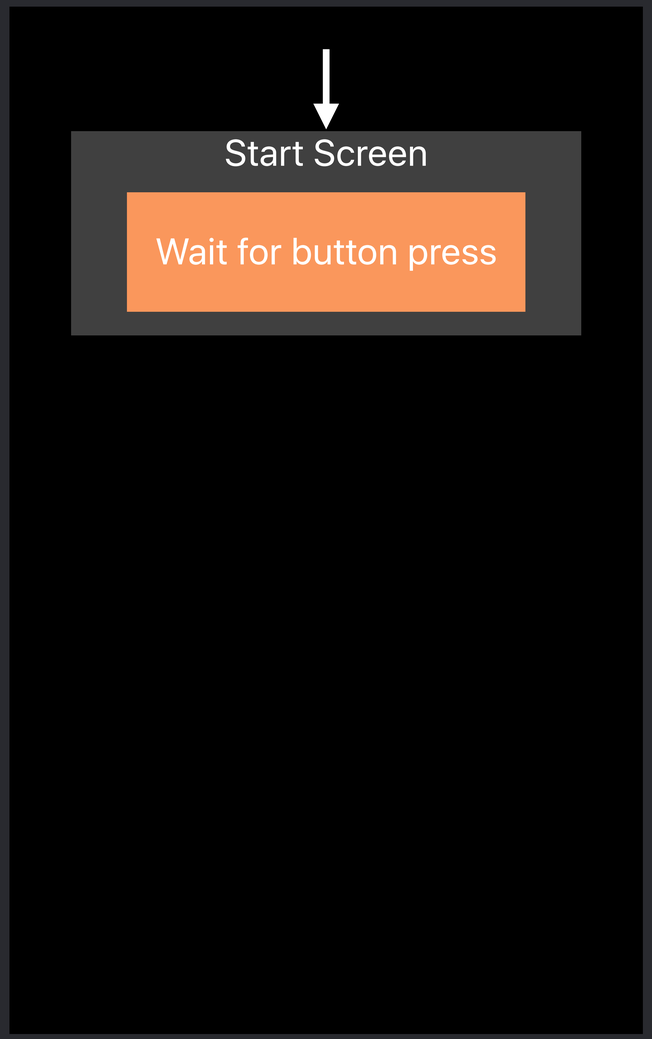
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
```



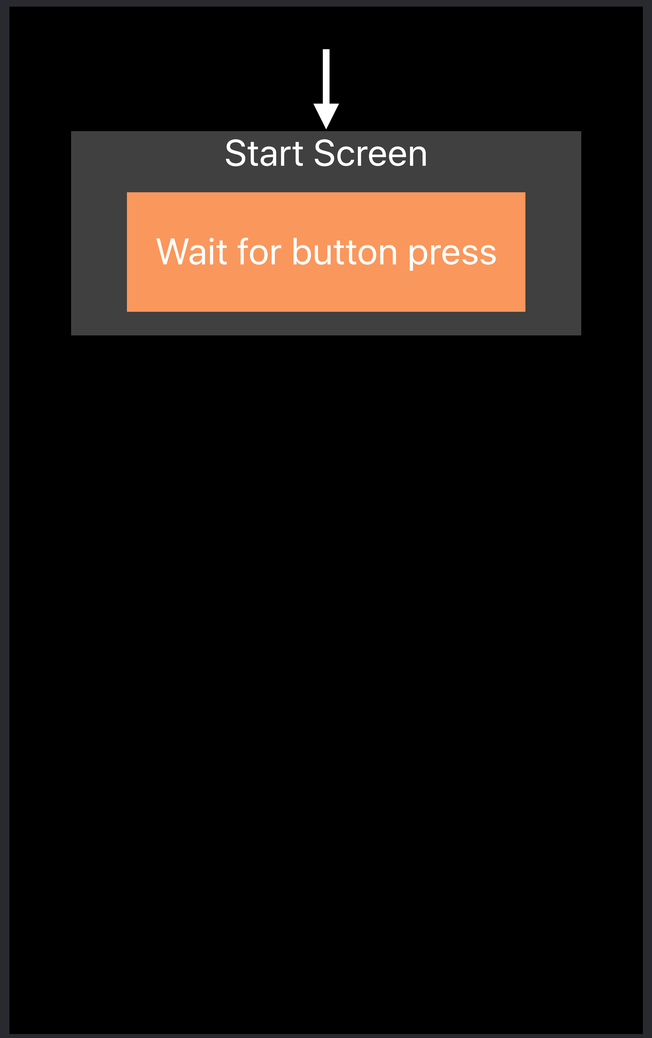
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



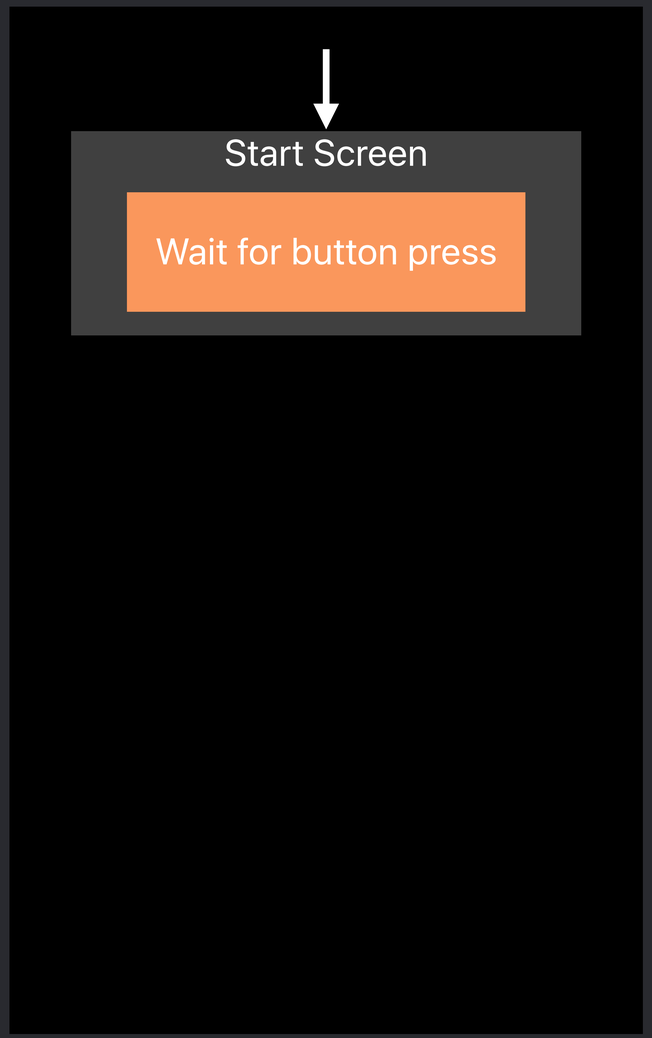
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



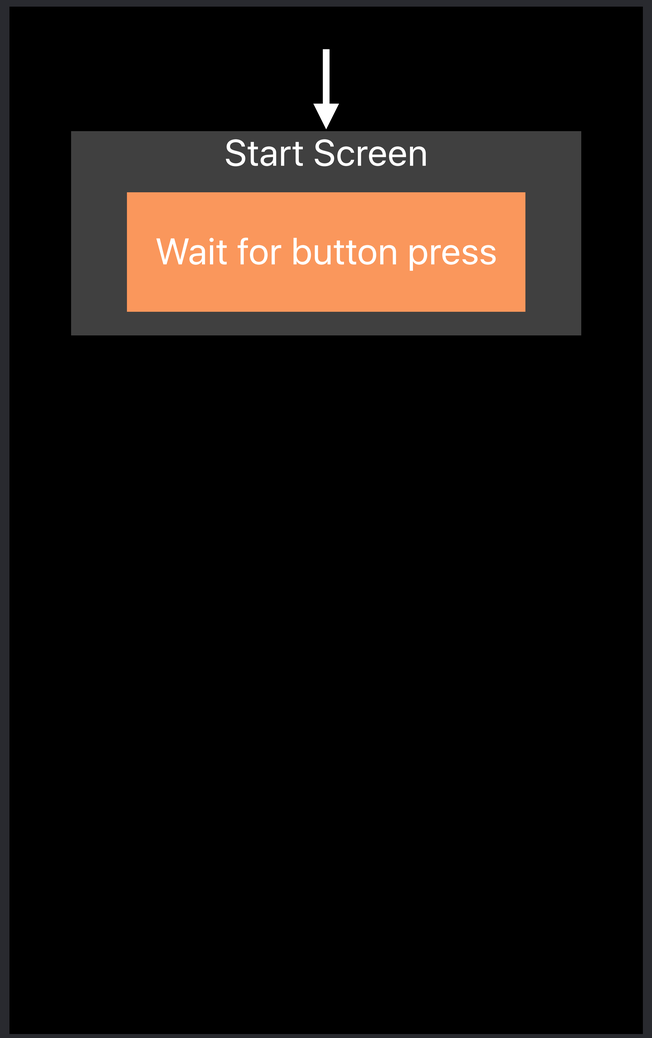
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



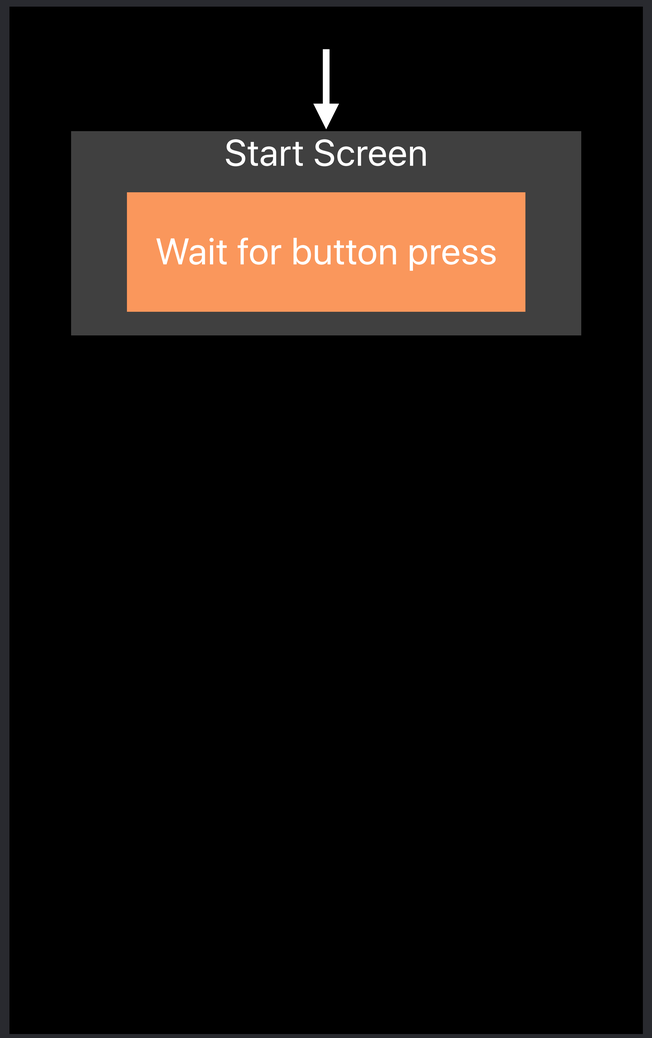
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



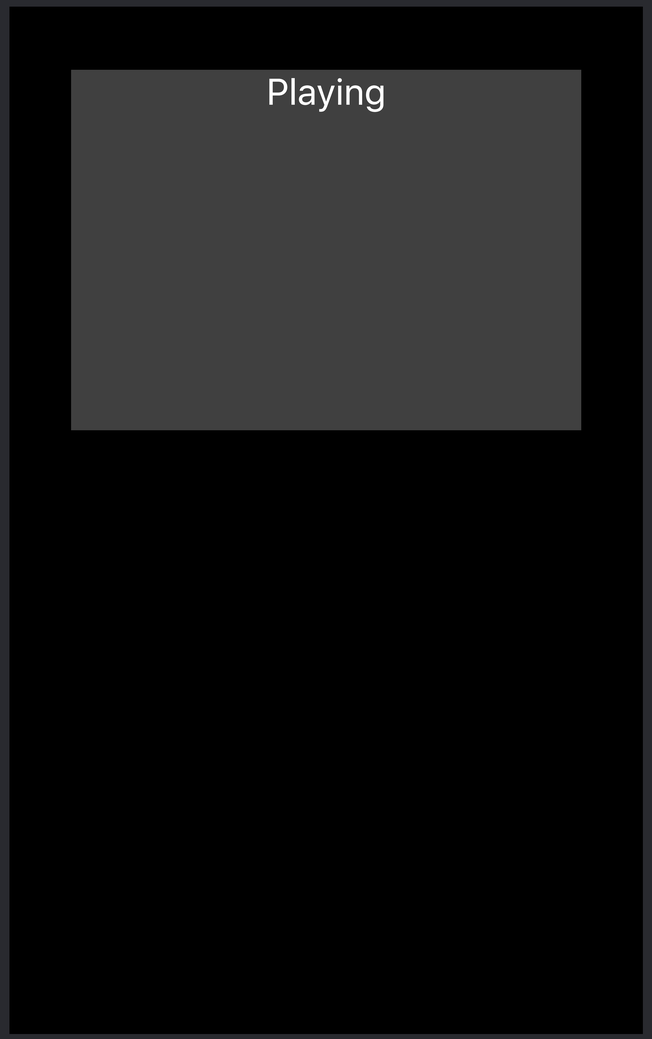
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



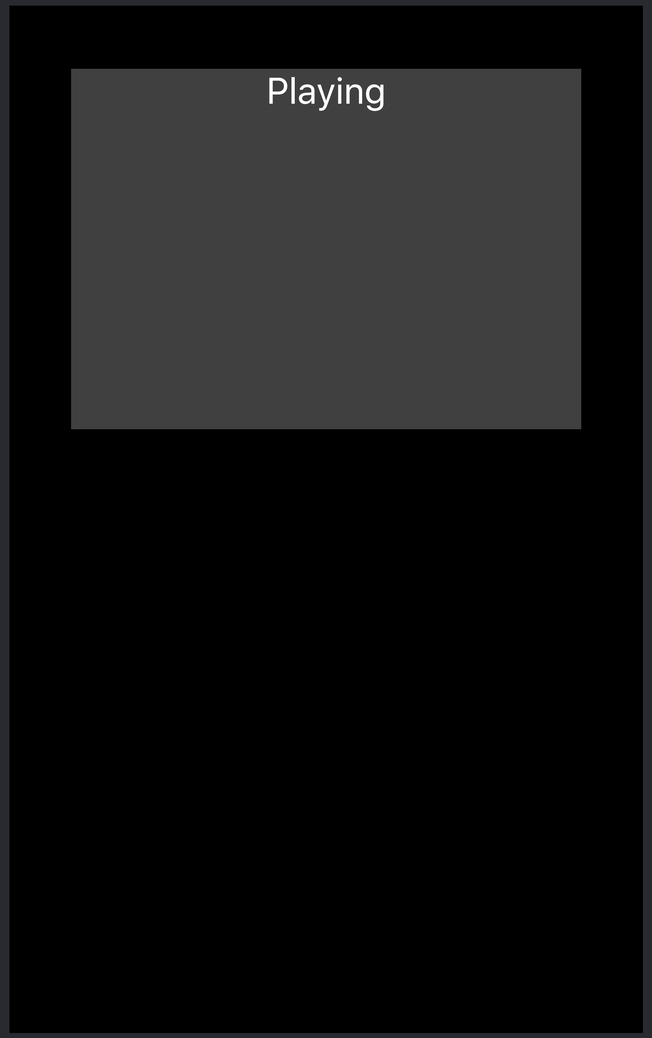
```
//Game Logic - Start New Game

@IBAction func startPressed(_ sender: Any) {
    startNewGame()
}
func startNewGame() {
    startGameButton.isHidden = true
    leaderboardButton.isHidden = true
    gamePoints = 0
    updatePointsLabel(gamePoints)
    pointsLabel.textColor = .magenta
    pointsLabel.isHidden = false
    oneGameRound()
}
```



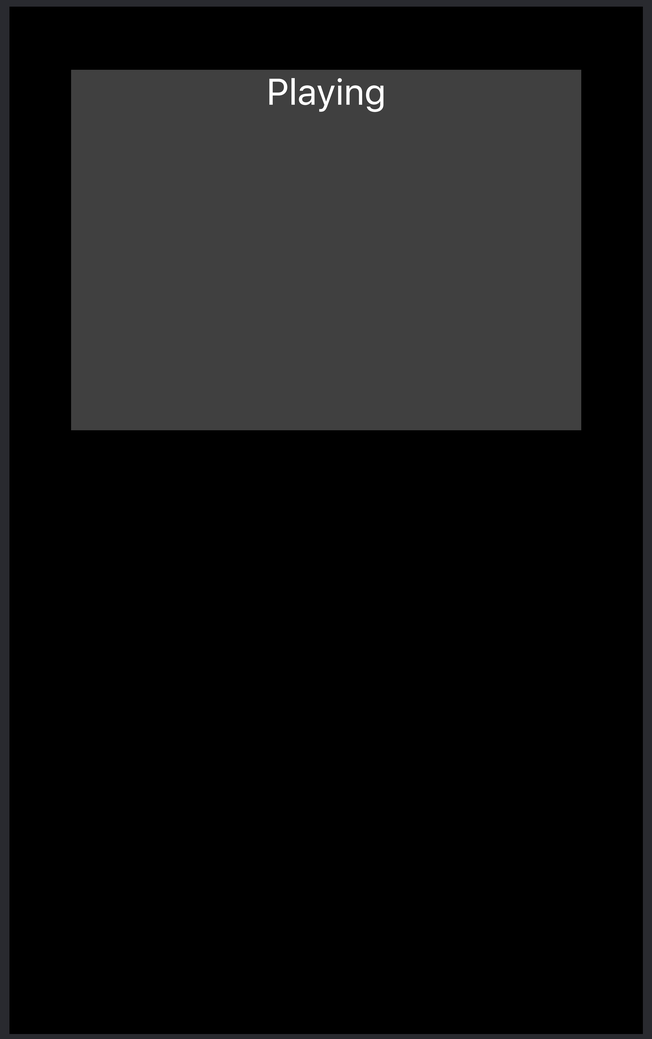
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



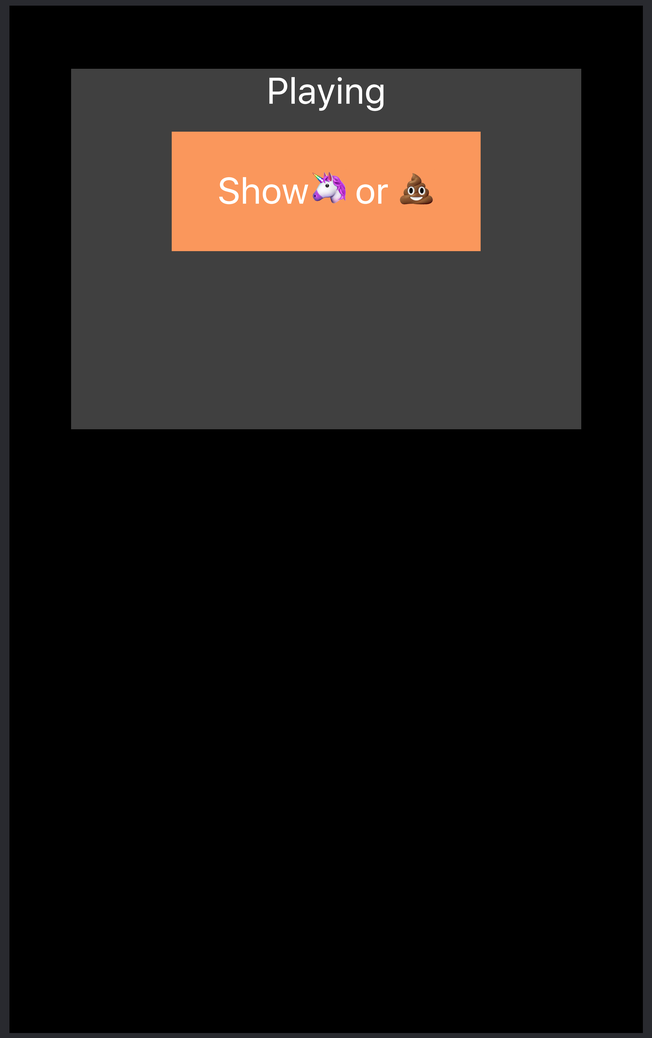
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



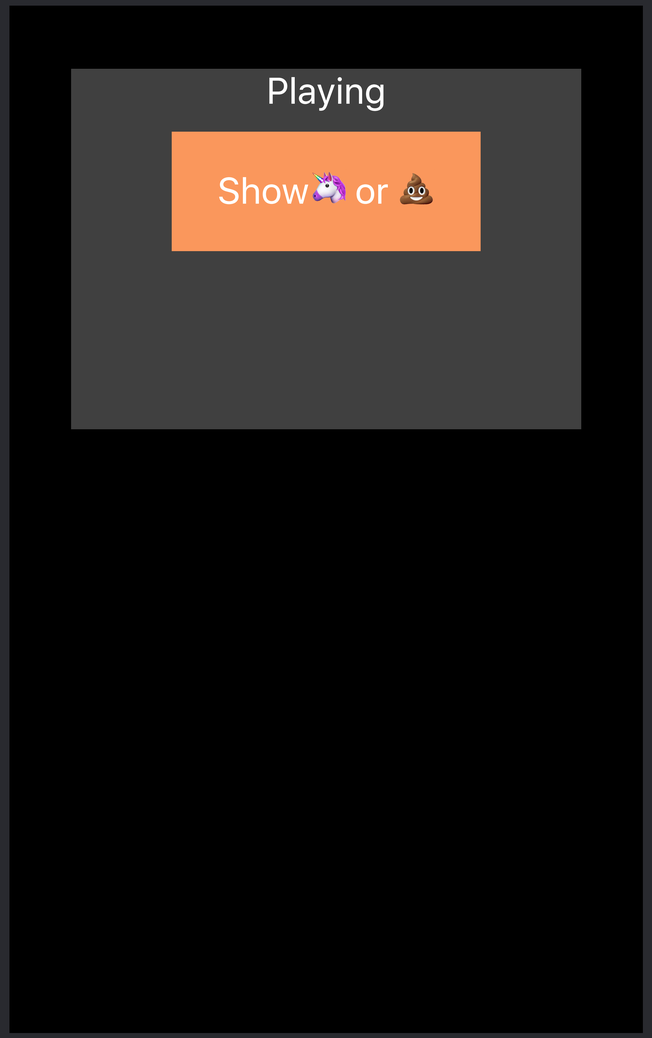
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



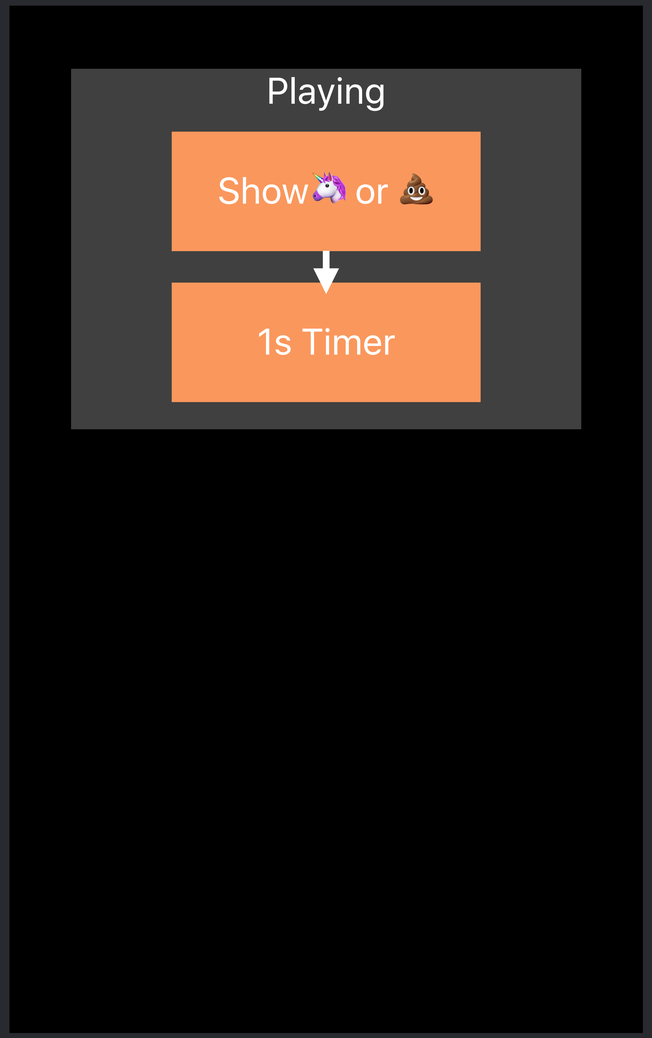
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



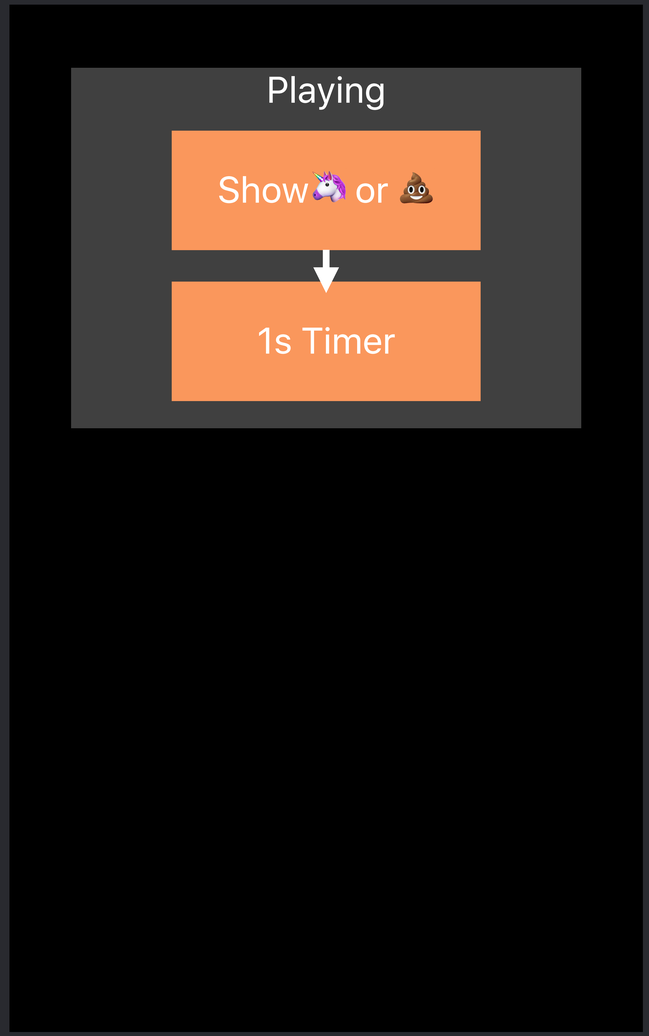
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



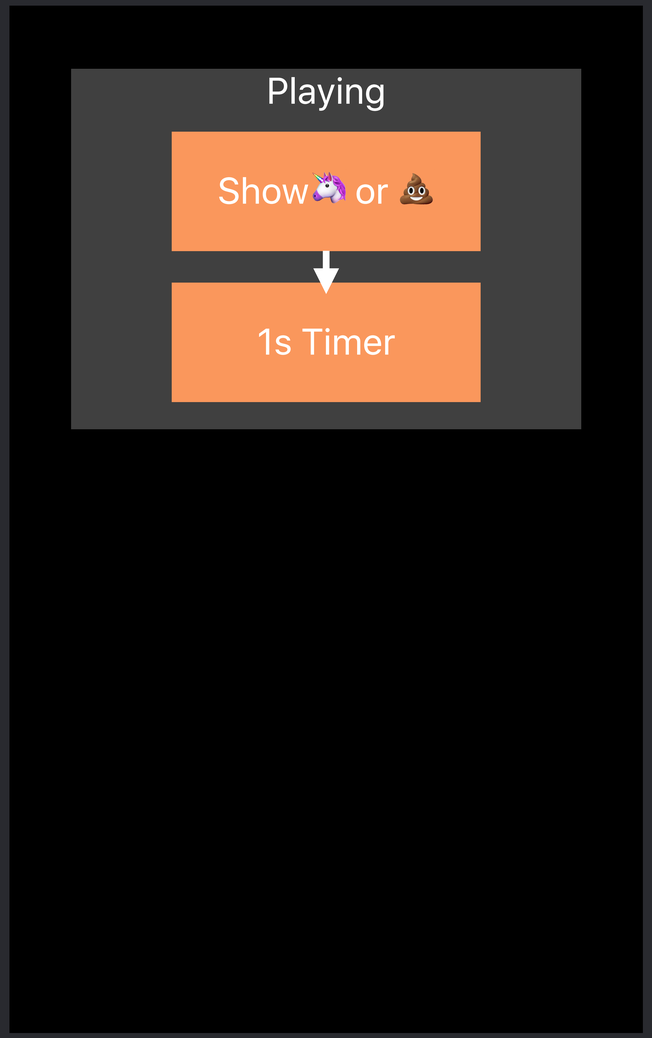
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



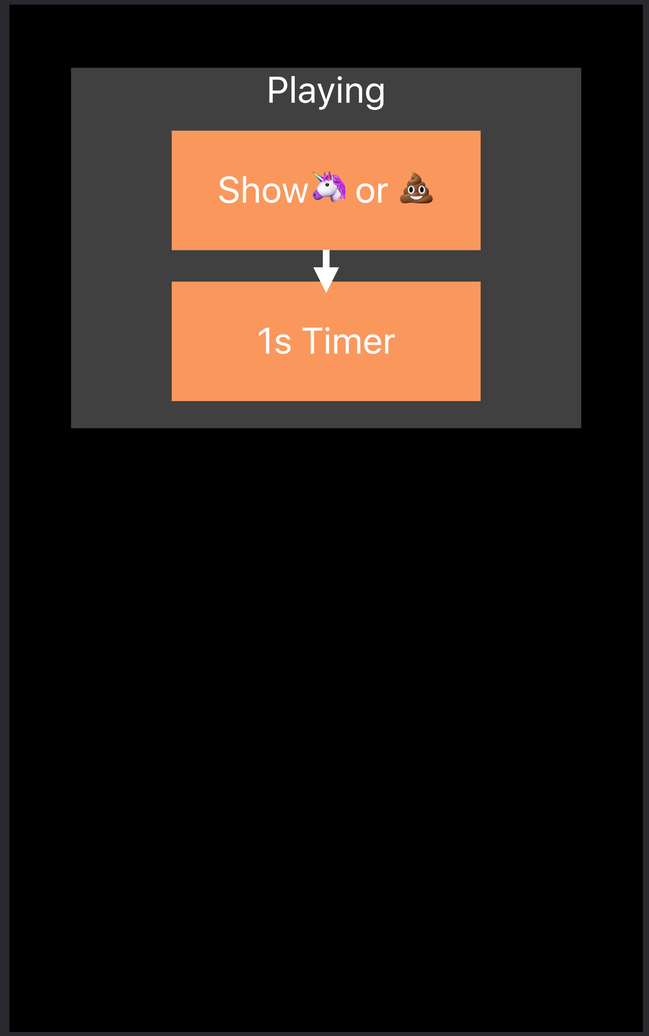
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



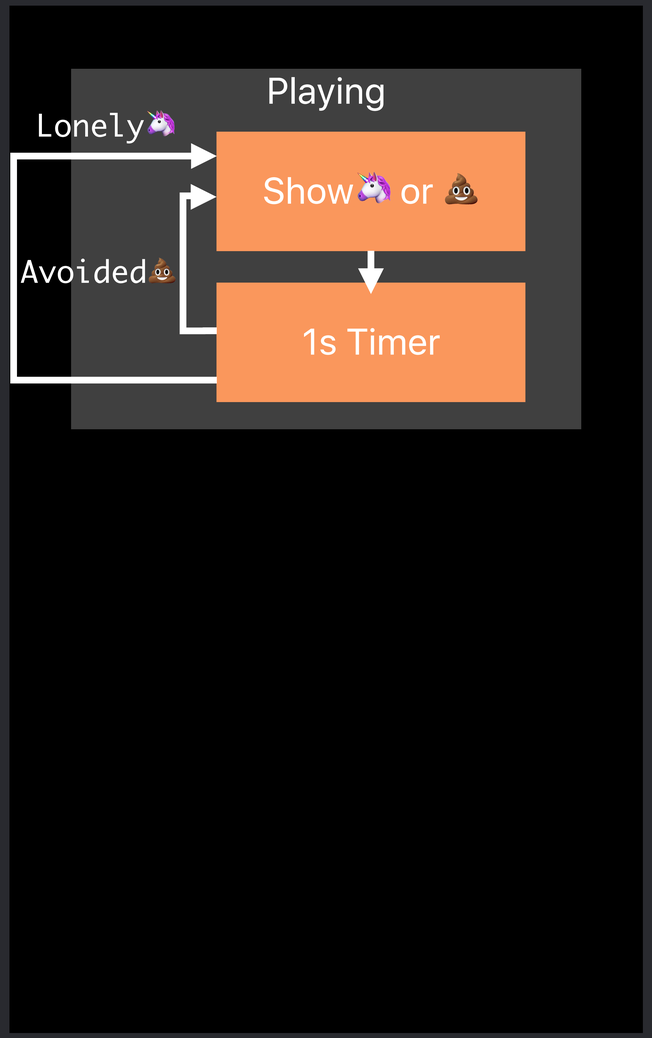
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



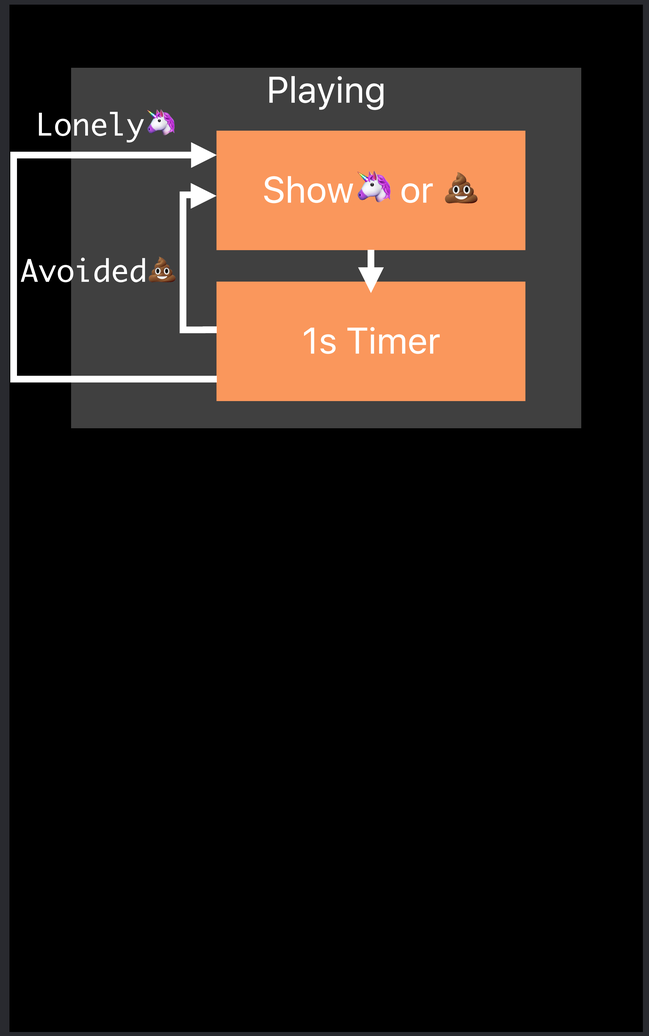
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



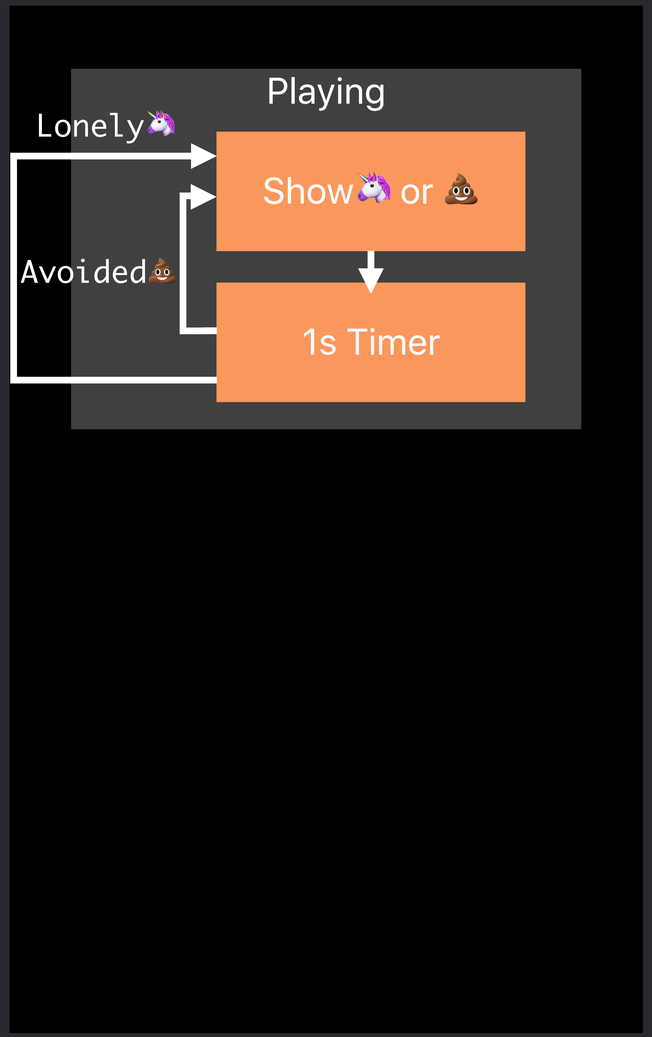
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



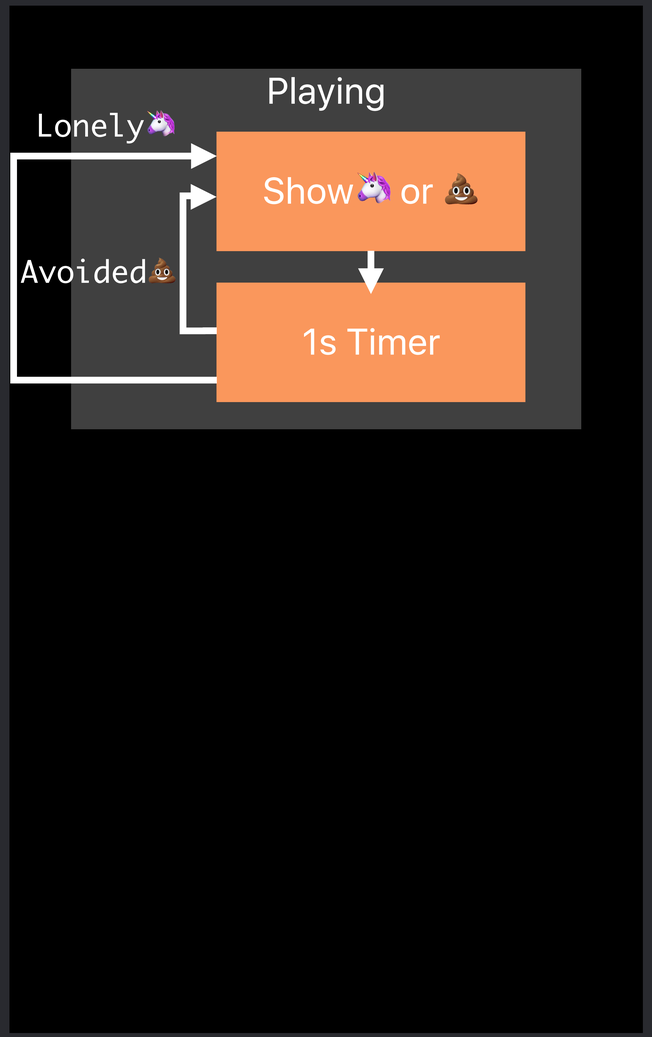
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



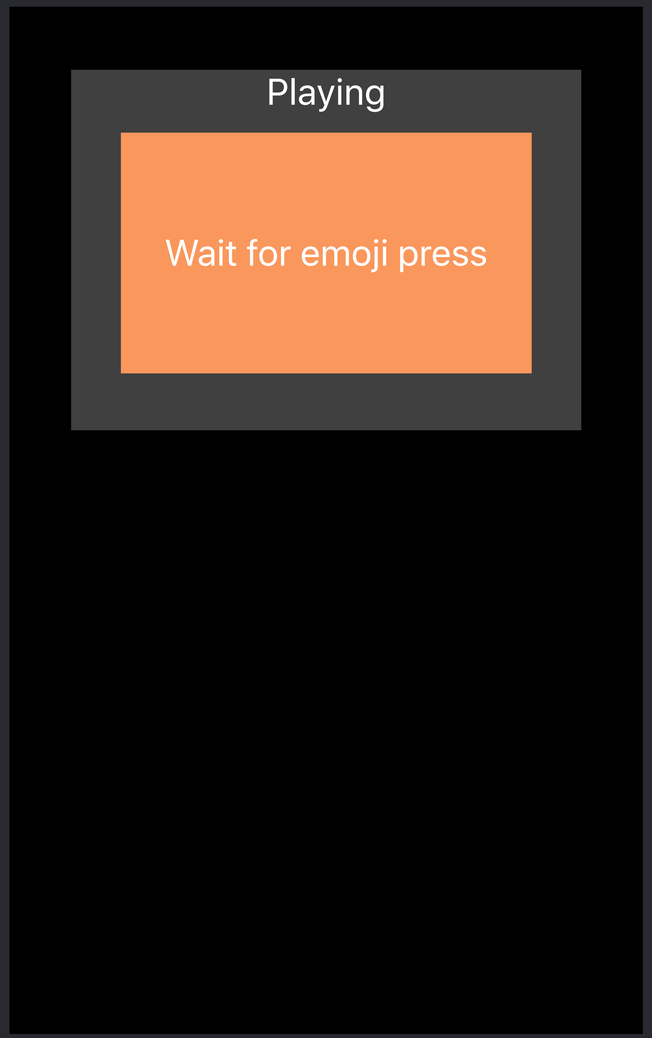
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



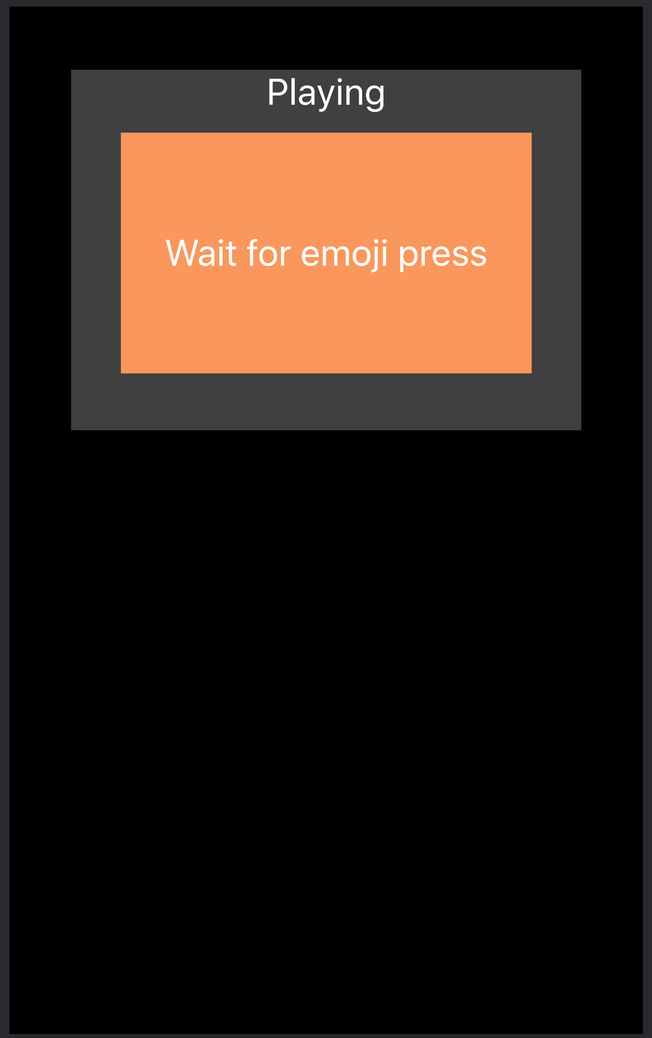
```
//Game Logic - One Game Round

func oneGameRound() {
    updatePointsLabel(gamePoints)
    displayRandomButton()

    timer = Timer.scheduledTimer(withTimeInterval: 1.0,
                                 repeats: false) { _ in
        if self.state == GameState.playing {
            if self.currentButton == self.goodButton {
                self.gameOver()
            } else {
                self.oneGameRound()
            }
        }
    }
}
```



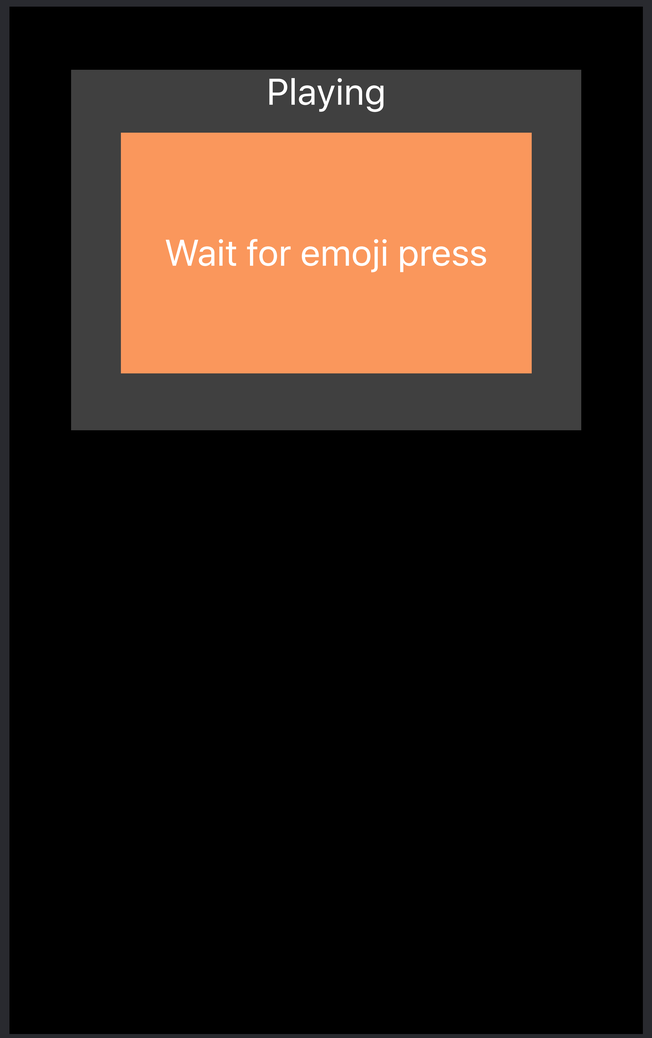
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



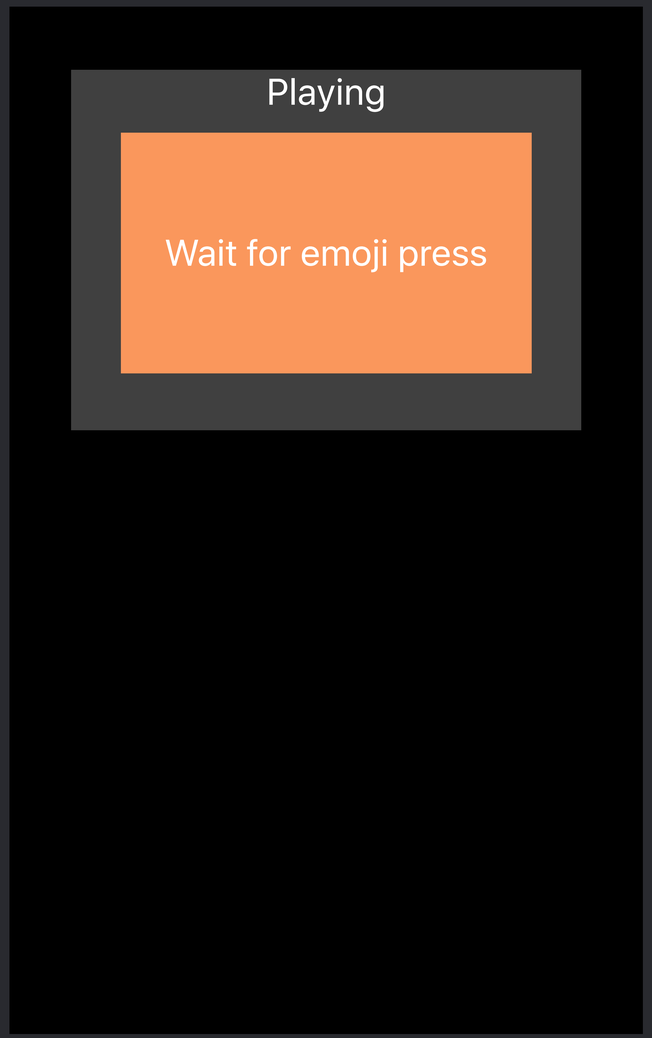
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



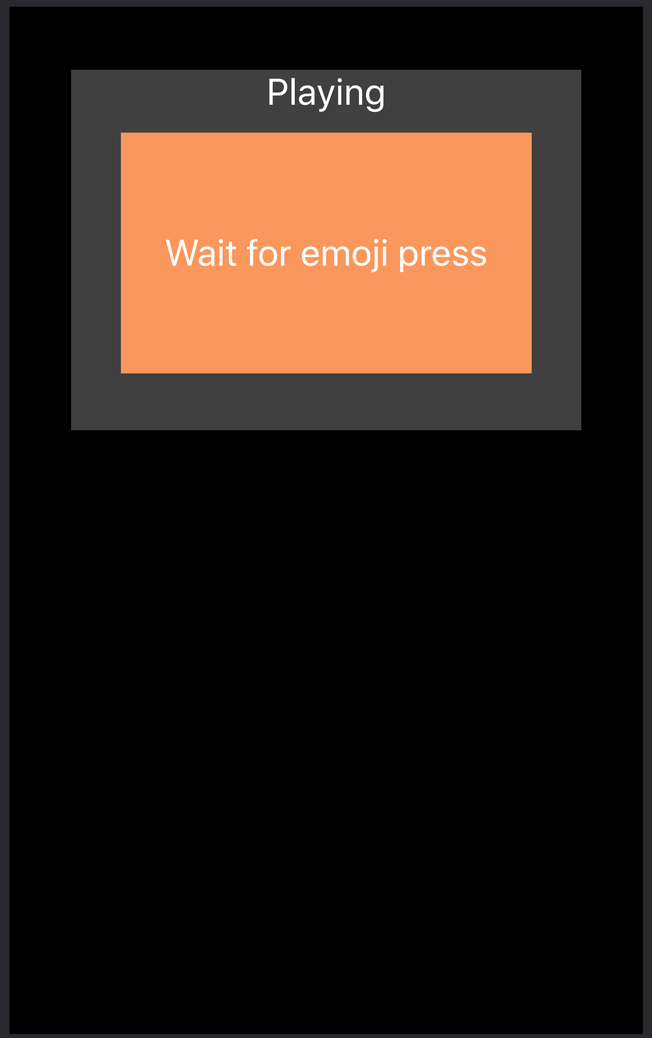
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



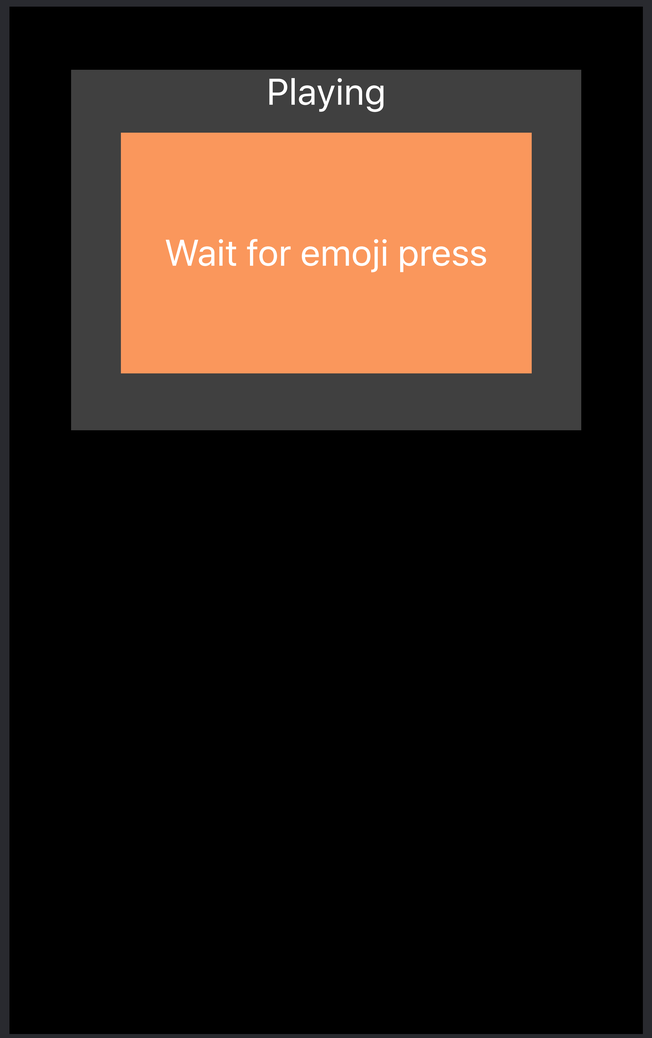
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



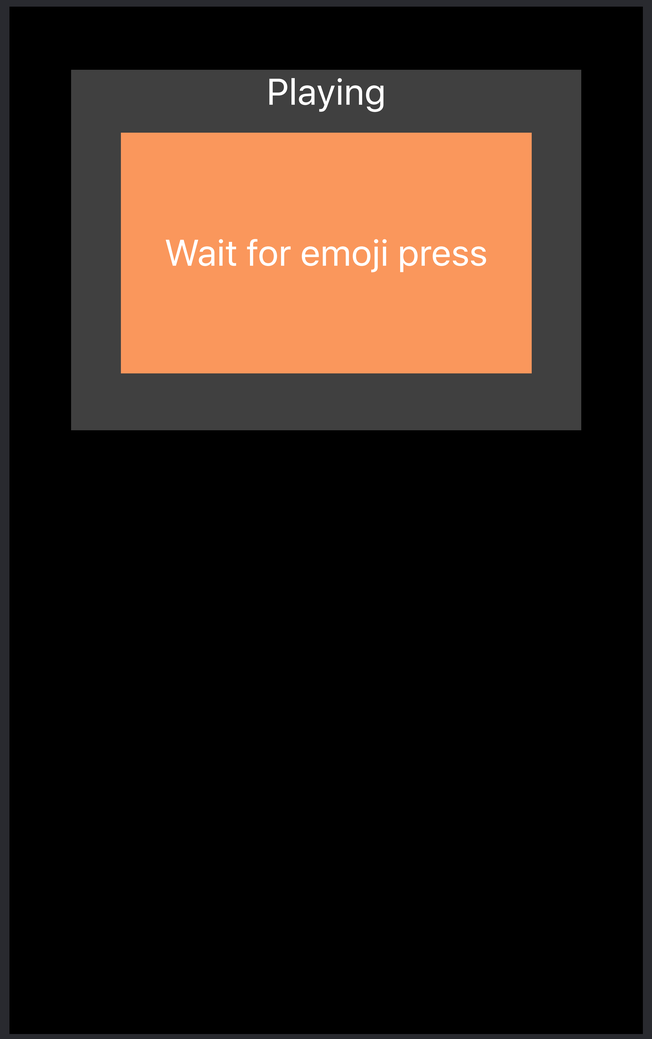
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



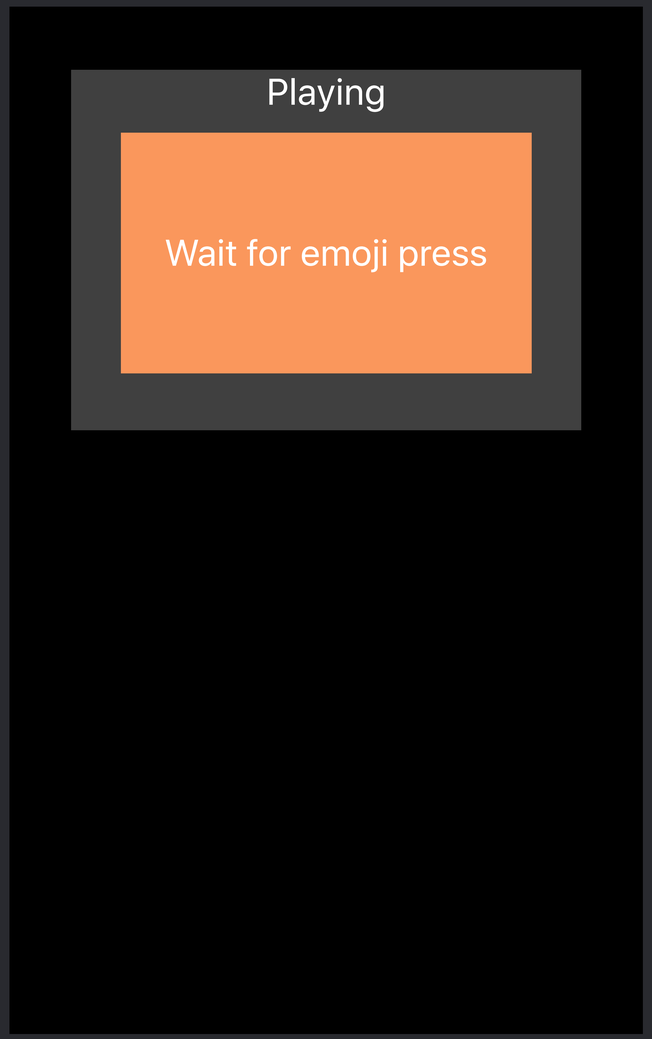
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }
```



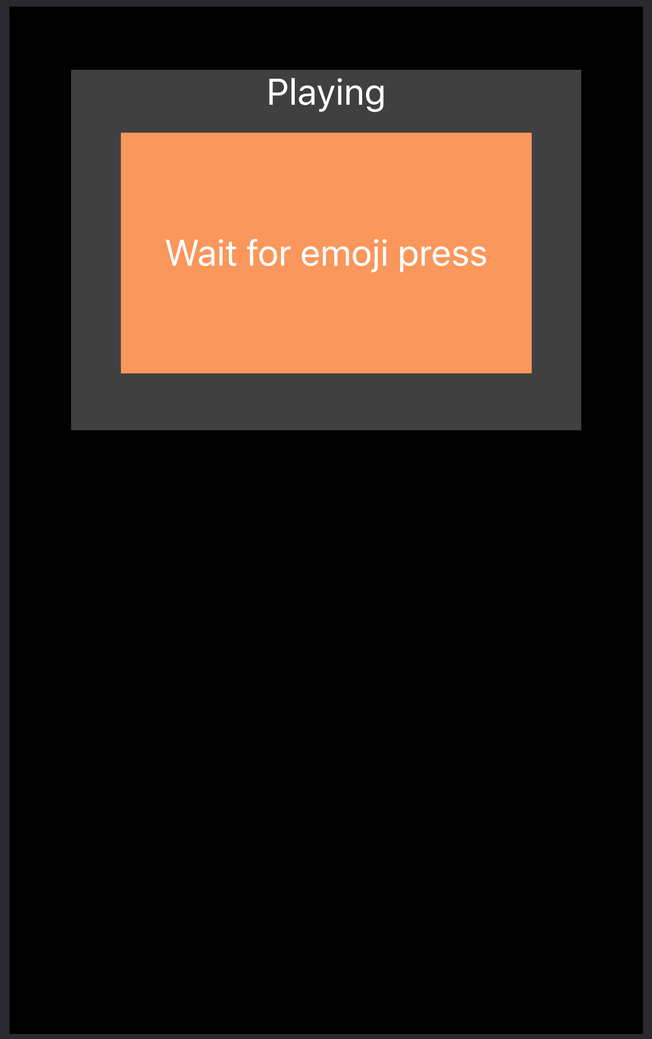
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }


 @IBAction func badButtonPressed(_ sender: Any) {
      badButton.isHidden = true
      timer?.invalidate()
      gameOver()
  }
```



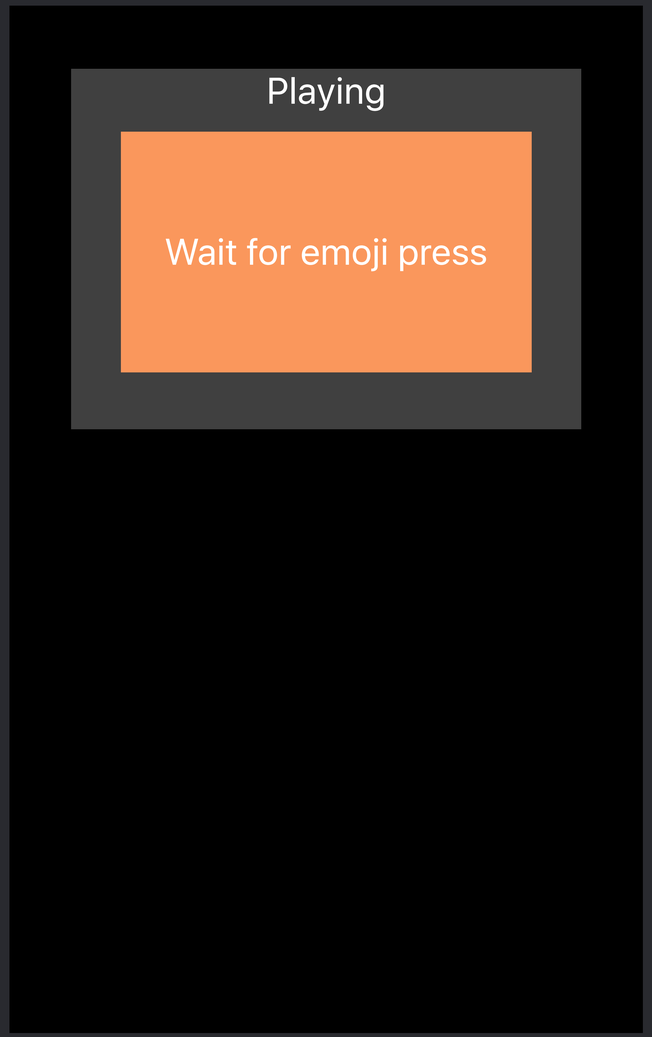
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }


 @IBAction func badButtonPressed(_ sender: Any) {
      badButton.isHidden = true
      timer?.invalidate()
      gameOver()
  }
```



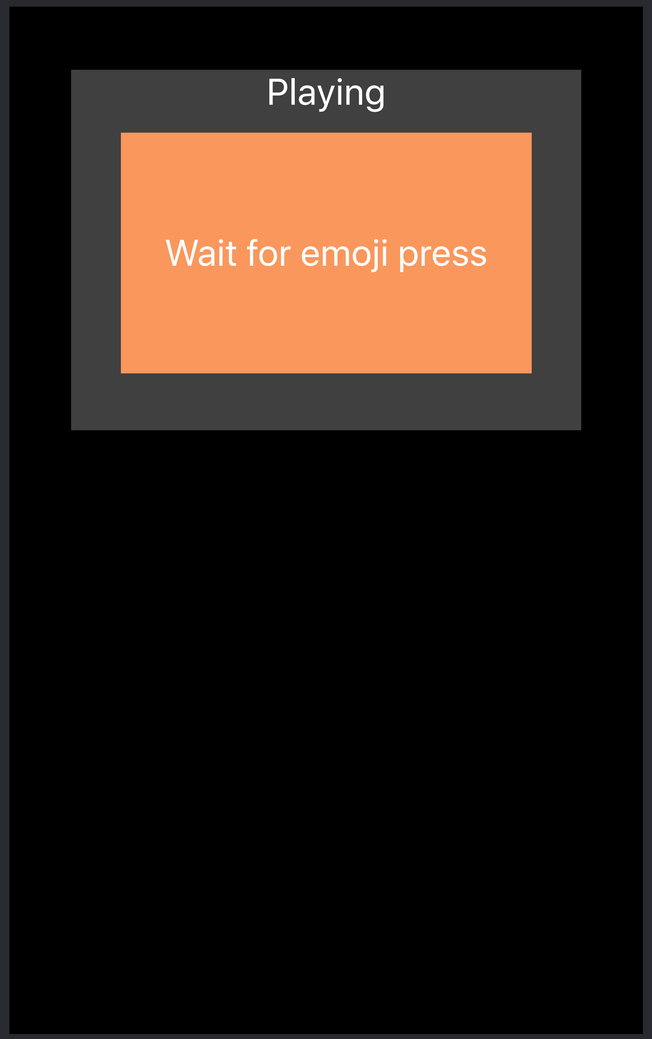
```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }


 @IBAction func badButtonPressed(_ sender: Any) {
      badButton.isHidden = true
      timer?.invalidate()
      gameOver()
  }
```



```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }


 @IBAction func badButtonPressed(_ sender: Any) {
      badButton.isHidden = true
      timer?.invalidate()
      gameOver()
  }
```



```
//Game Logic - Good or Bad button pressed

// Add a point, and update the score when the good button is pressed
    @IBAction func goodButtonPressed(_ sender: Any) {
        gamePoints = gamePoints + 1
        updatePointsLabel(gamePoints)
        goodButton.isHidden = true
        timer?.invalidate()
        oneGameRound()
    }


 @IBAction func badButtonPressed(_ sender: Any) {
      badButton.isHidden = true
      timer?.invalidate()
      gameOver()
  }
```



Demo

Bring in the game logic

Accomplishments 🎉

Accomplishments 🎉

Navigating Xcode to learn how to create fun apps

Accomplishments 🎉

Navigating Xcode to learn how to create fun apps

Creating a simple UI using the Storyboard

Accomplishments 🎉

Navigating Xcode to learn how to create fun apps

Creating a simple UI using the Storyboard

Connecting our UI to code

Accomplishments 🎉

Navigating Xcode to learn how to create fun apps

Creating a simple UI using the Storyboard

Connecting our UI to code

Writing game logic code in Swift

Taking It Further! 🚀

Taking It Further! 🚀

Use SpriteKit to give more life to your good and bad buttons

Taking It Further! 🚀

Use SpriteKit to give more life to your good and bad buttons

Add MusicKit integration to incorporate sound into the game play

Taking It Further! 🚀

Use SpriteKit to give more life to your good and bad buttons

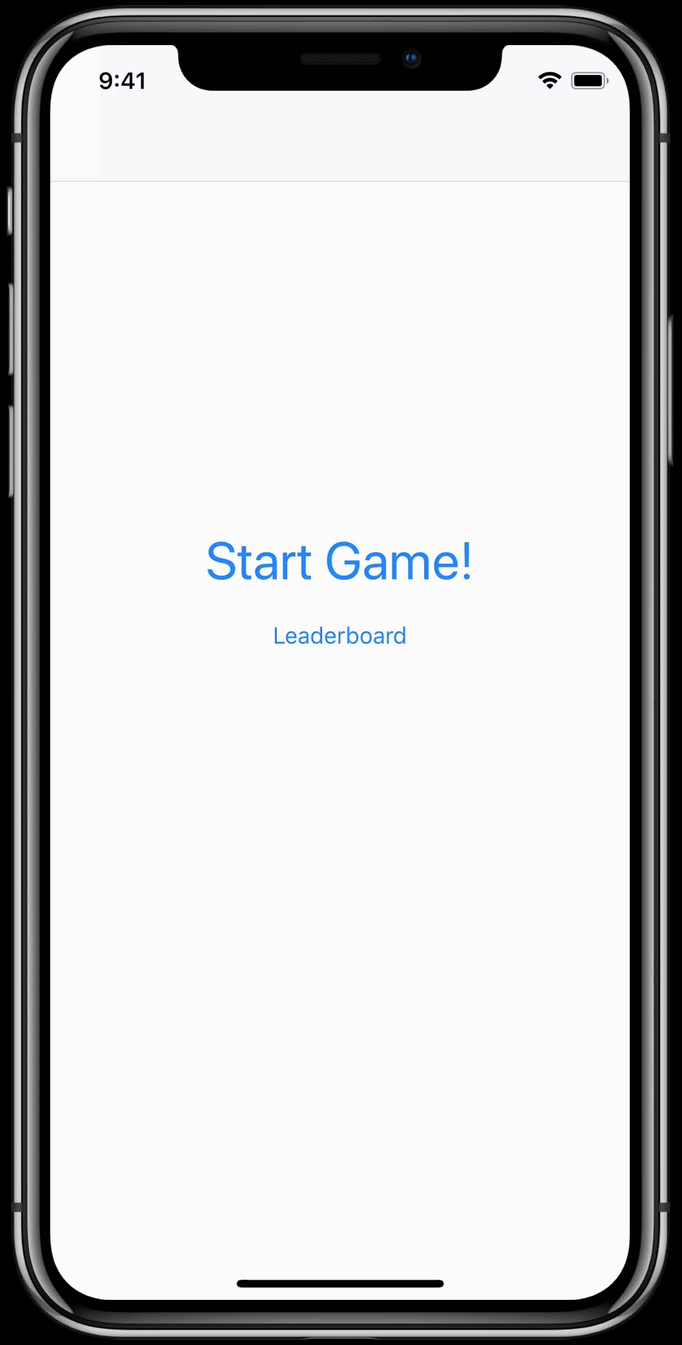
Add MusicKit integration to incorporate sound into the game play

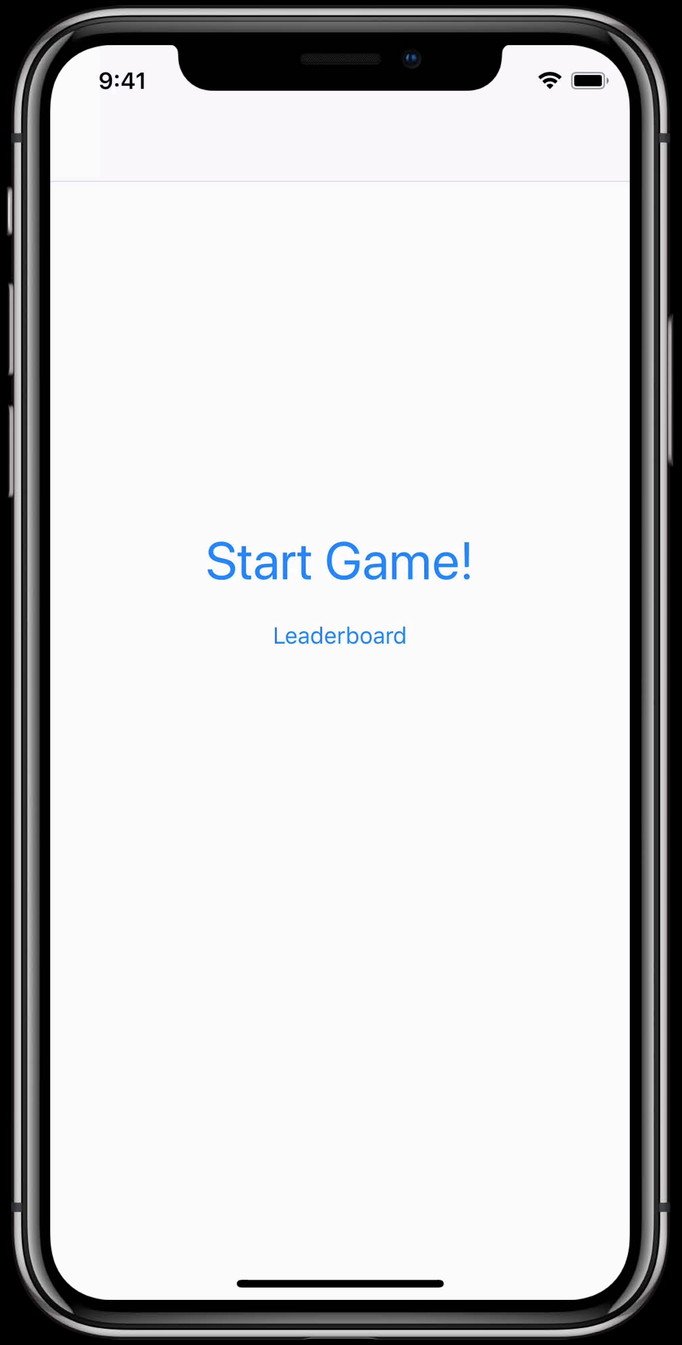
Read from sensors and change the speed of the game based on user movement

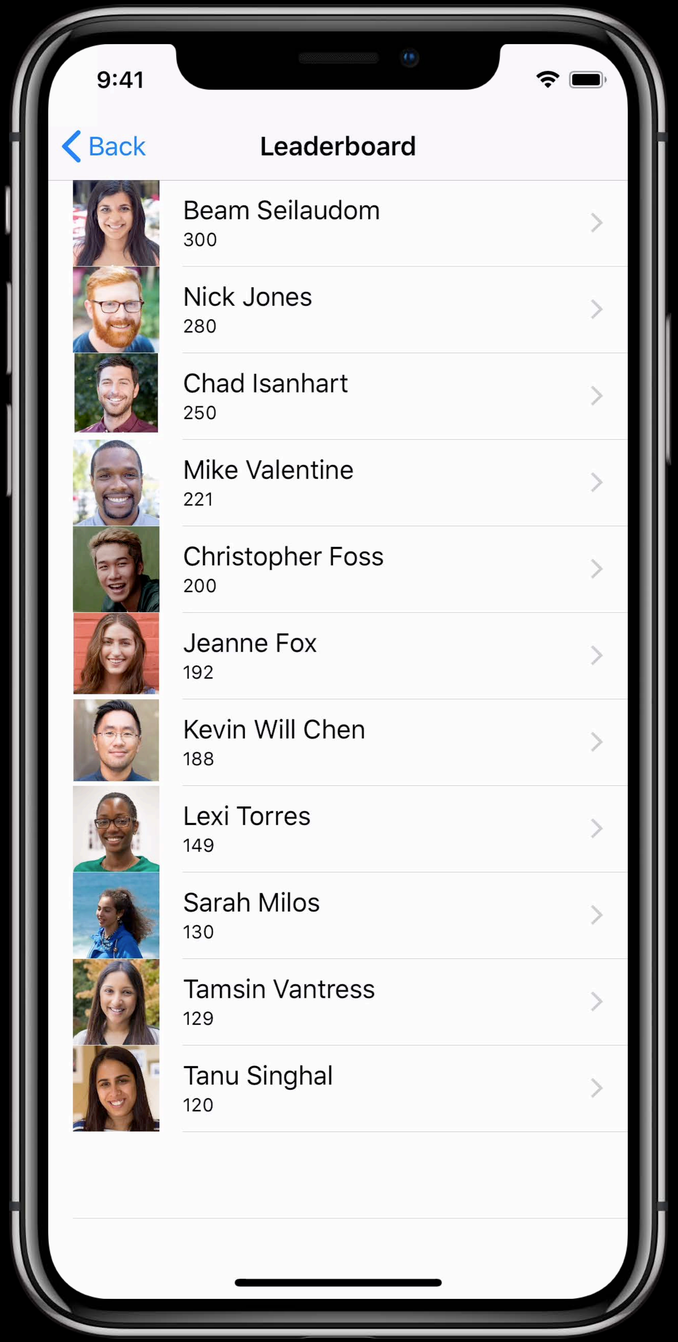
Introduction to SpriteKit	WWDC 2013
Introducing MusicKit	WWDC 2017
Creating Immersive Apps with Core Motion	WWDC 2017

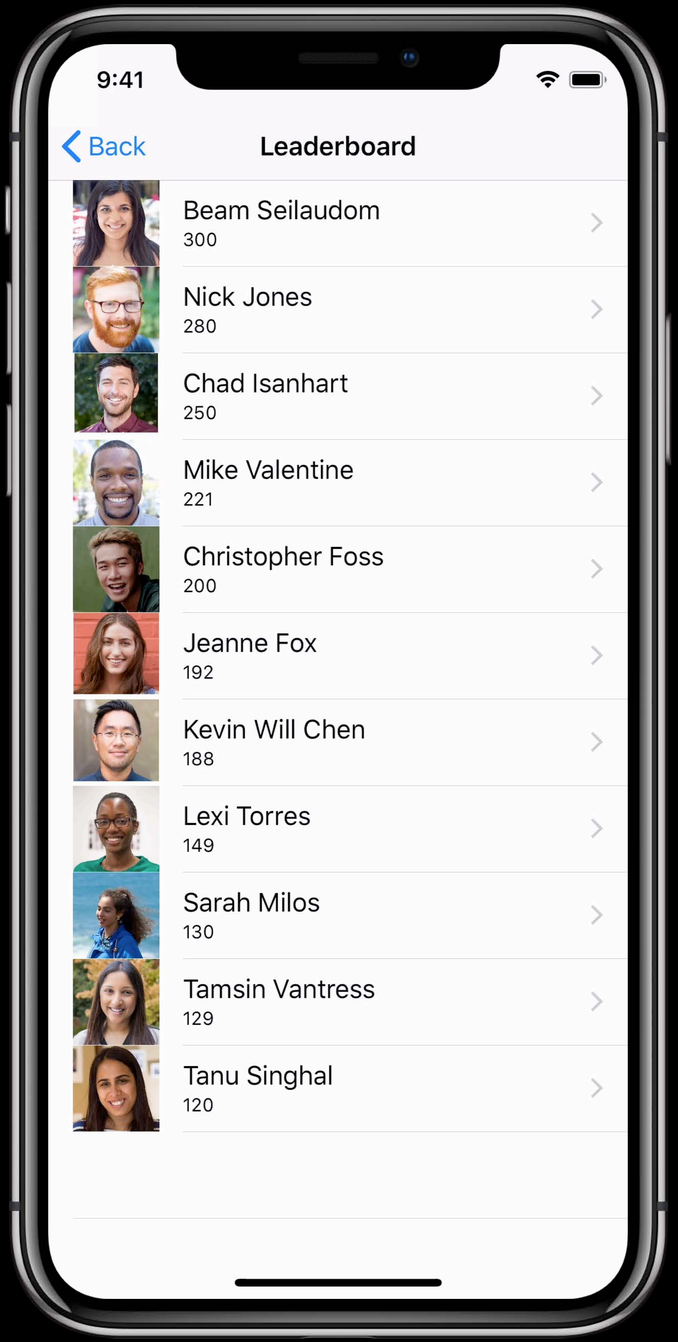
Enhancing Your App

Tanu Singhal


9:41
Start Game!
Leaderboard


9:41
Start Game!
Leaderboard


9:41
Back
Leaderboard
Beam Seilaudom
300
Nick Jones
280
Chad Isanhart
250
Mike Valentine
221
Christopher Foss
200
Jeanne Fox
192
Kevin Will Chen
188
Lexi Torres
149
Sarah Milos
130
Tamsin Vantress
129
Tanu Singhal
120


9:41
Back
Leaderboard
Beam Seilaudom
300
Nick Jones
280
Chad Isanhart
250
Mike Valentine
221
Christopher Foss
200
Jeanne Fox
192
Kevin Will Chen
188
Lexi Torres
149
Sarah Milos
130
Tamsin Vantress
129
Tanu Singhal
120

Areas of Focus

Areas of Focus

Data

Areas of Focus

Data

User Interface

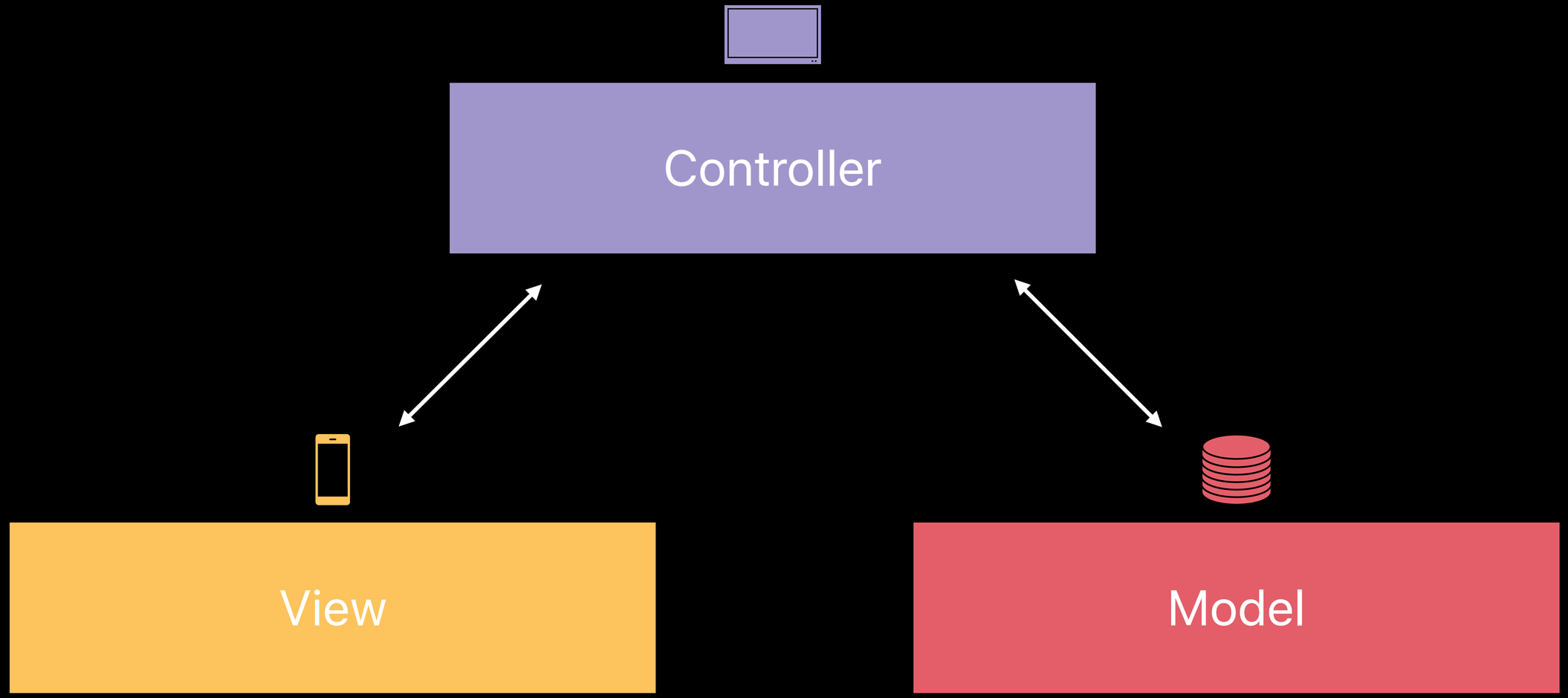
Areas of Focus

Data

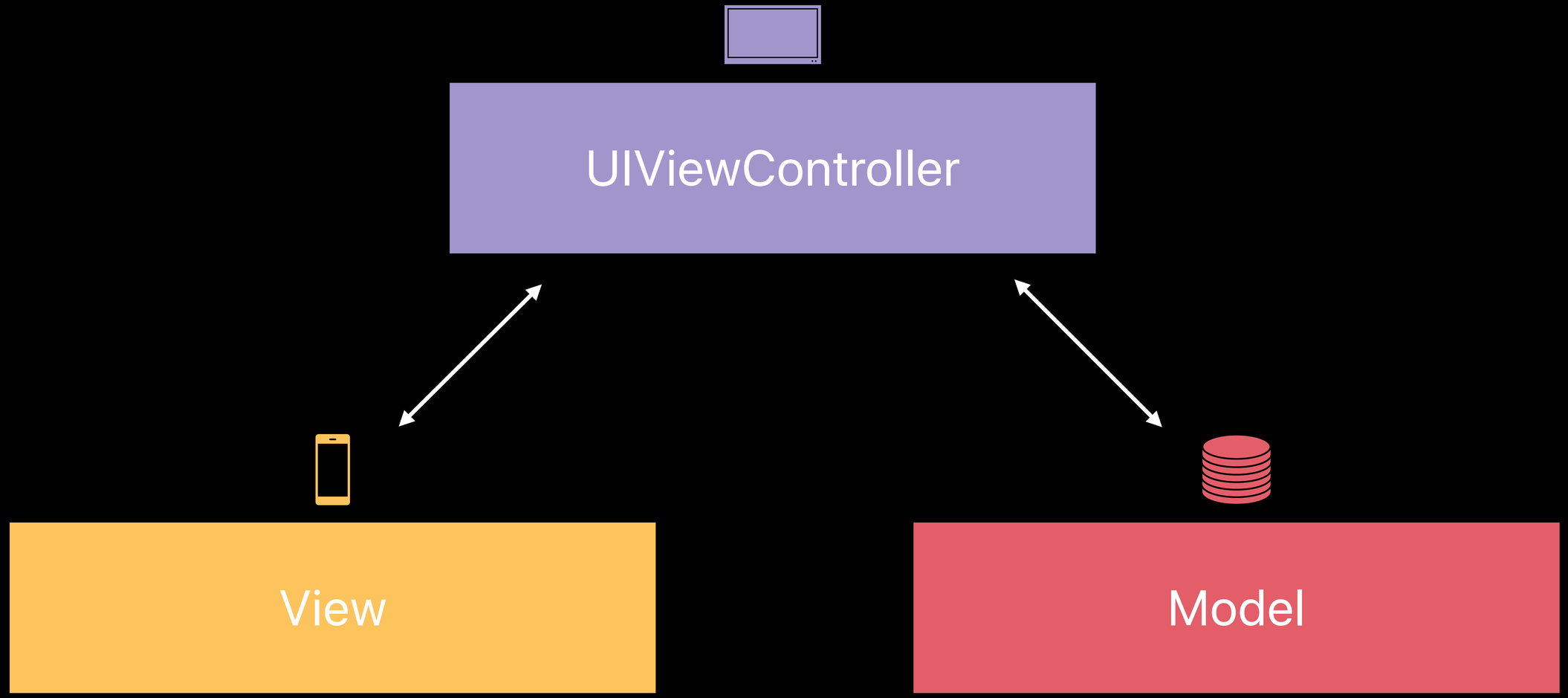
User Interface

Logic

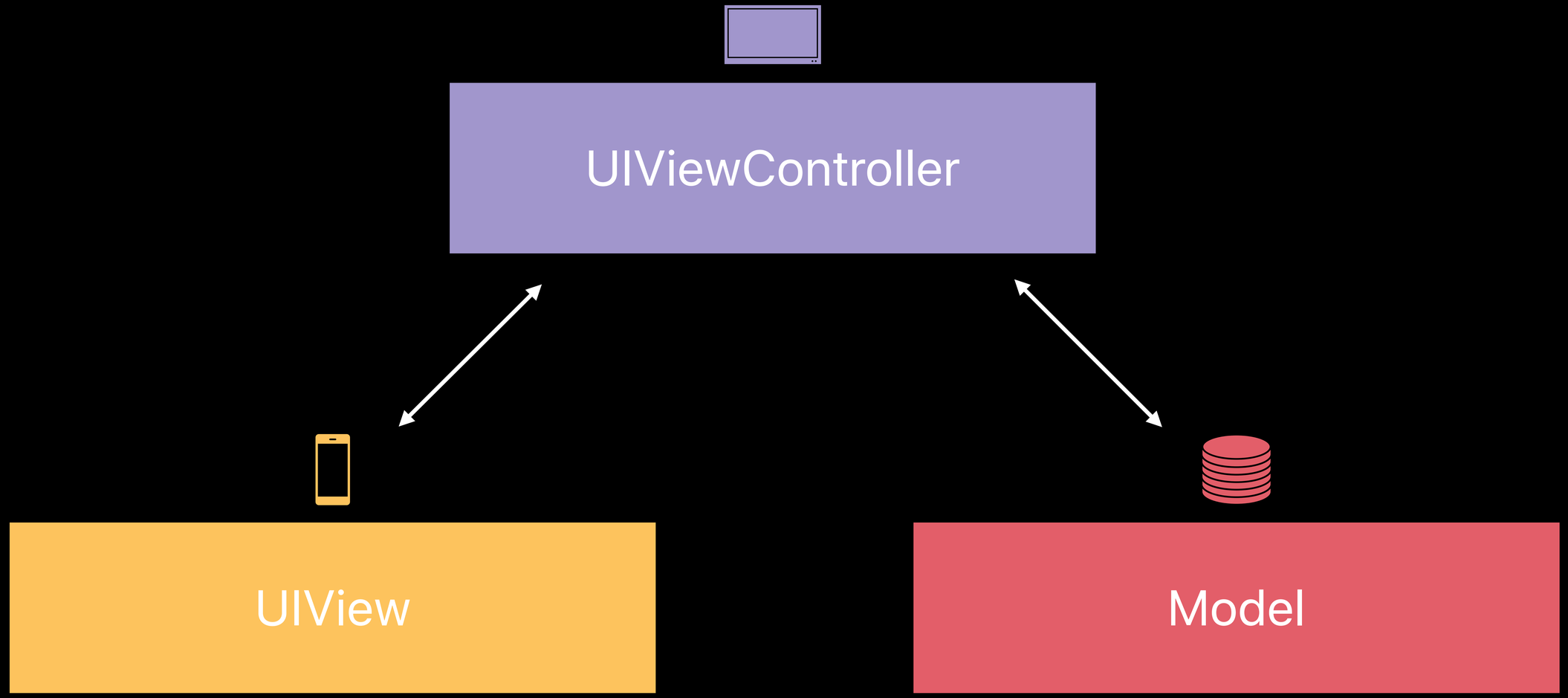
Model-View-Controller



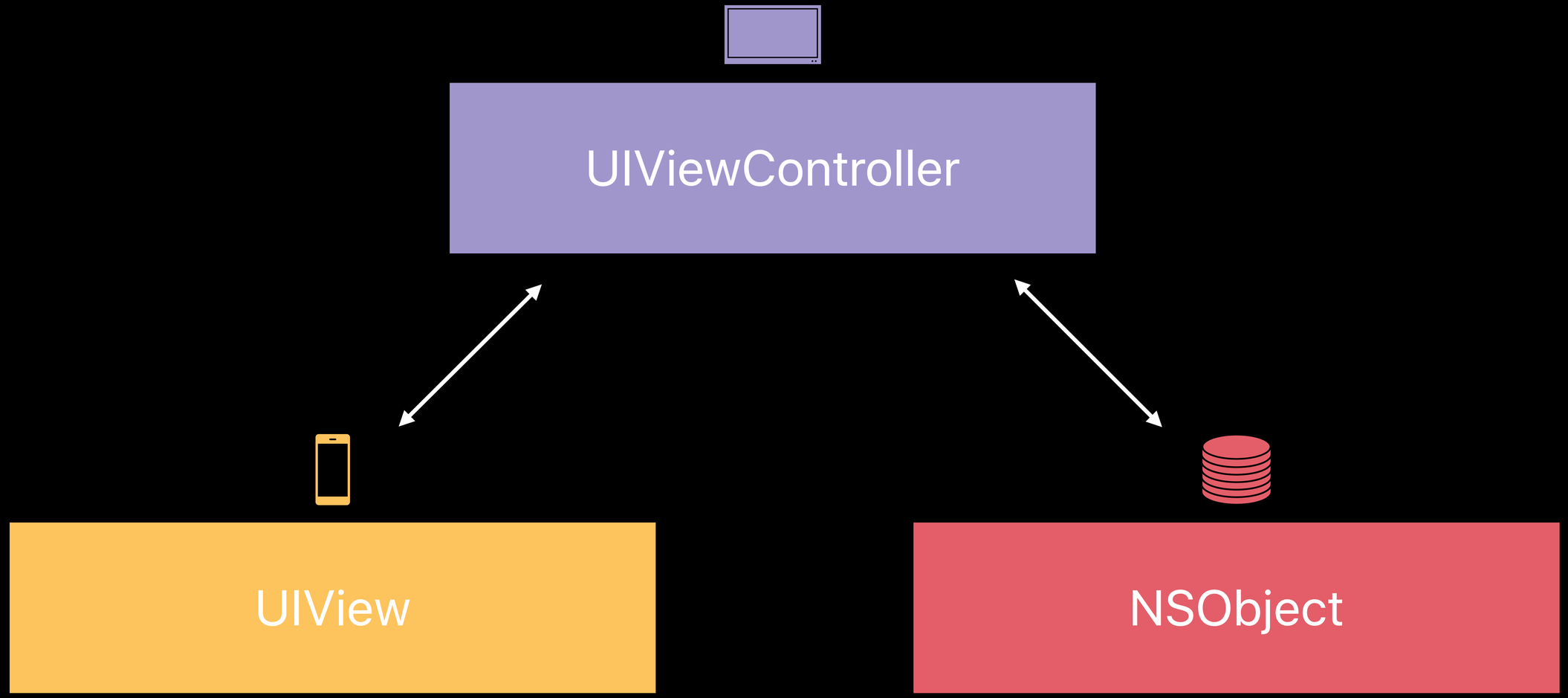
Model-View-Controller



Model-View-Controller



Model-View-Controller



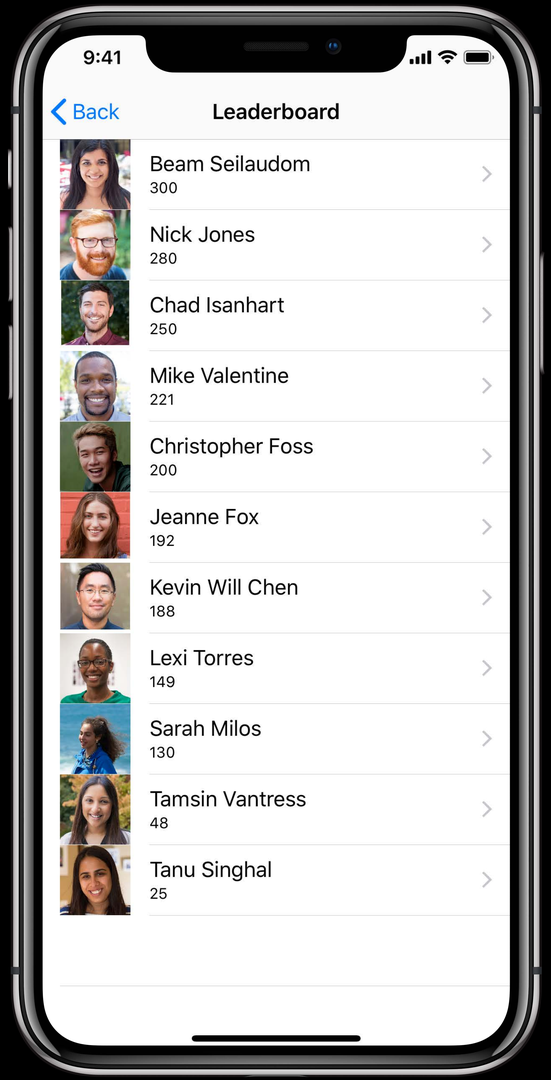
Demo

Setting up a data model

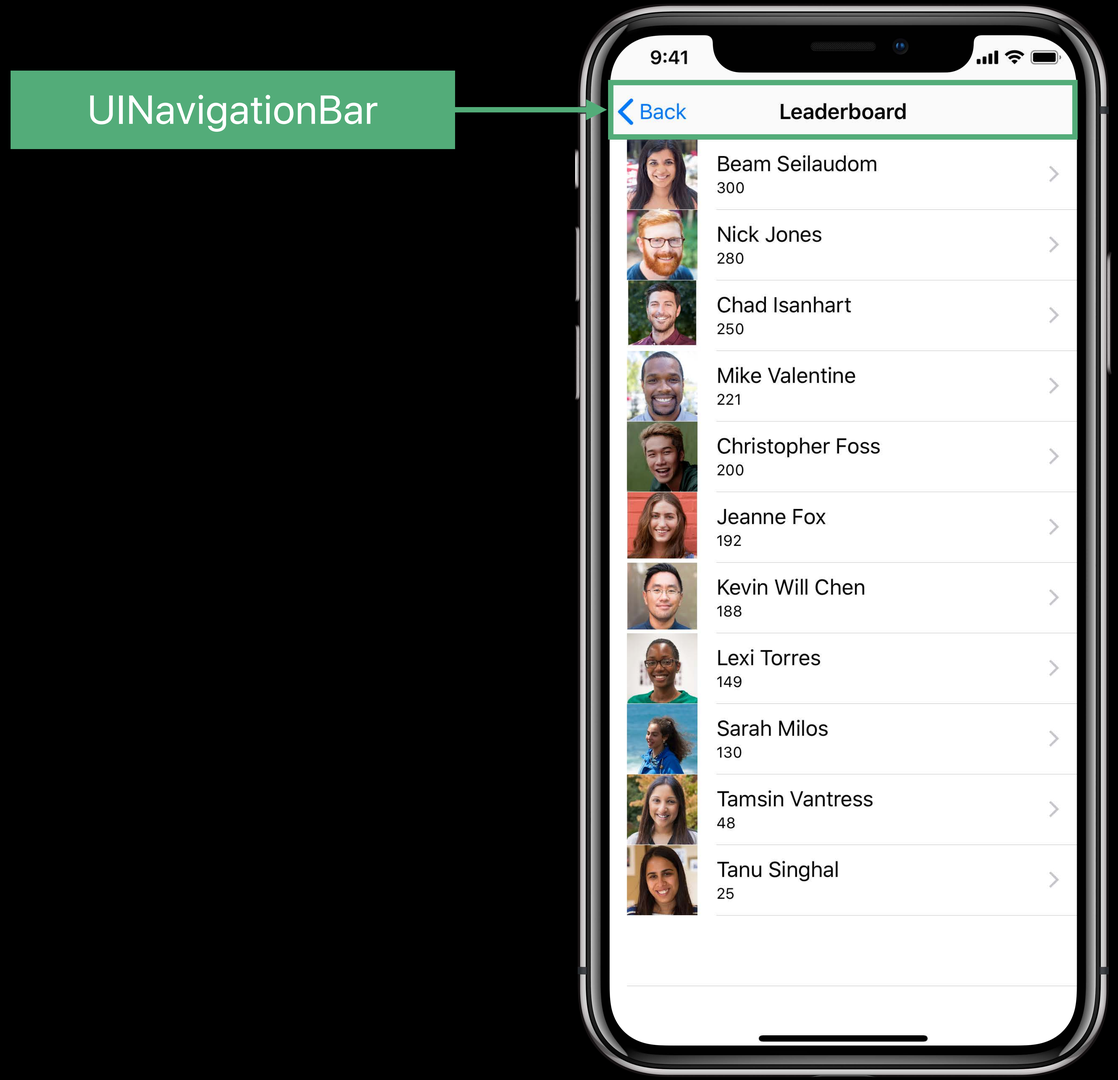
Saving Data

Core Data Best Practices	Executive Ballroom	Thursday 2:00PM
Networking with NSURLSession		WWDC 2015
Introducing CloudKit		WWDC 2014

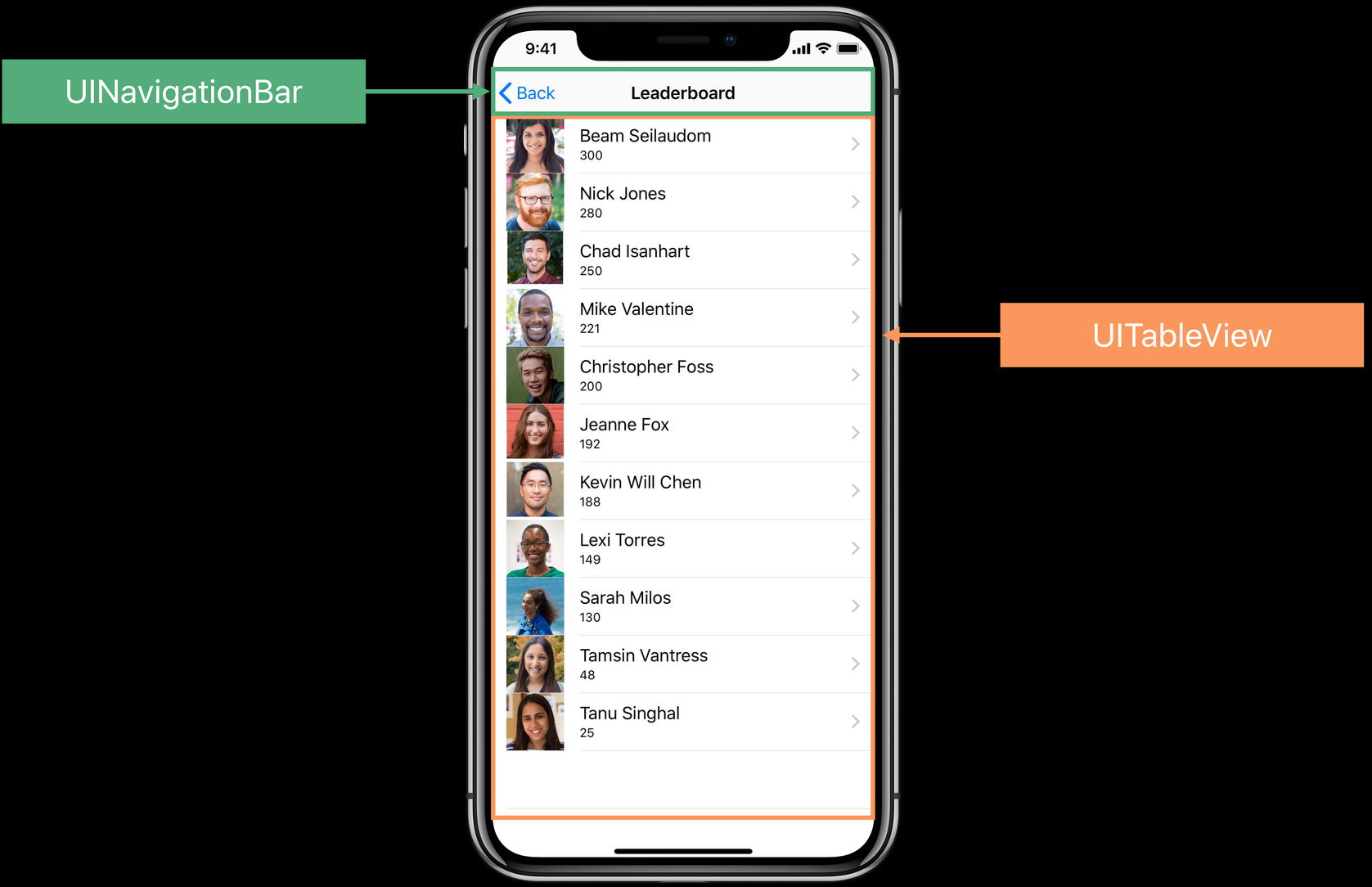
Leaderboard



Leaderboard

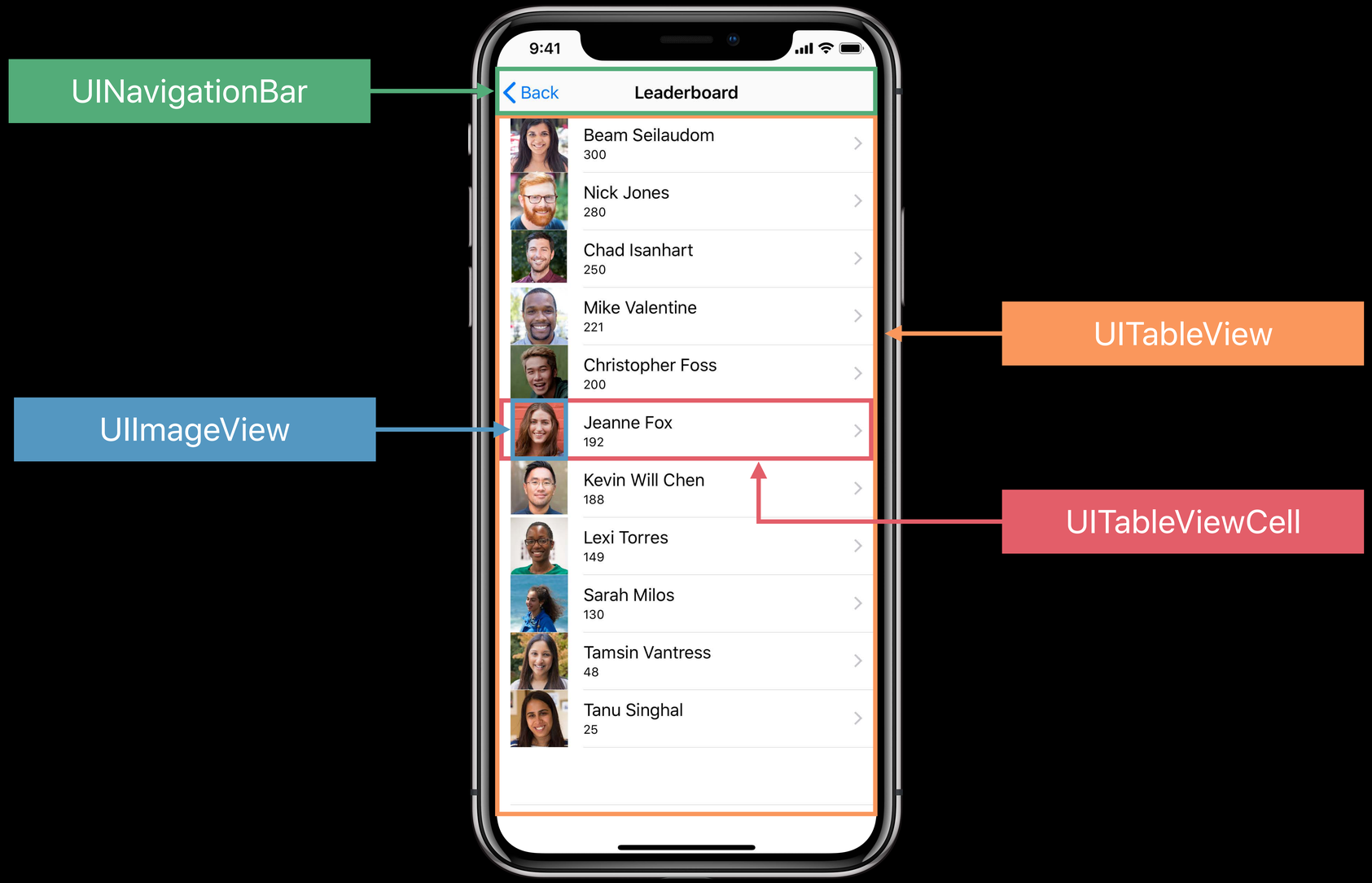


Leaderboard

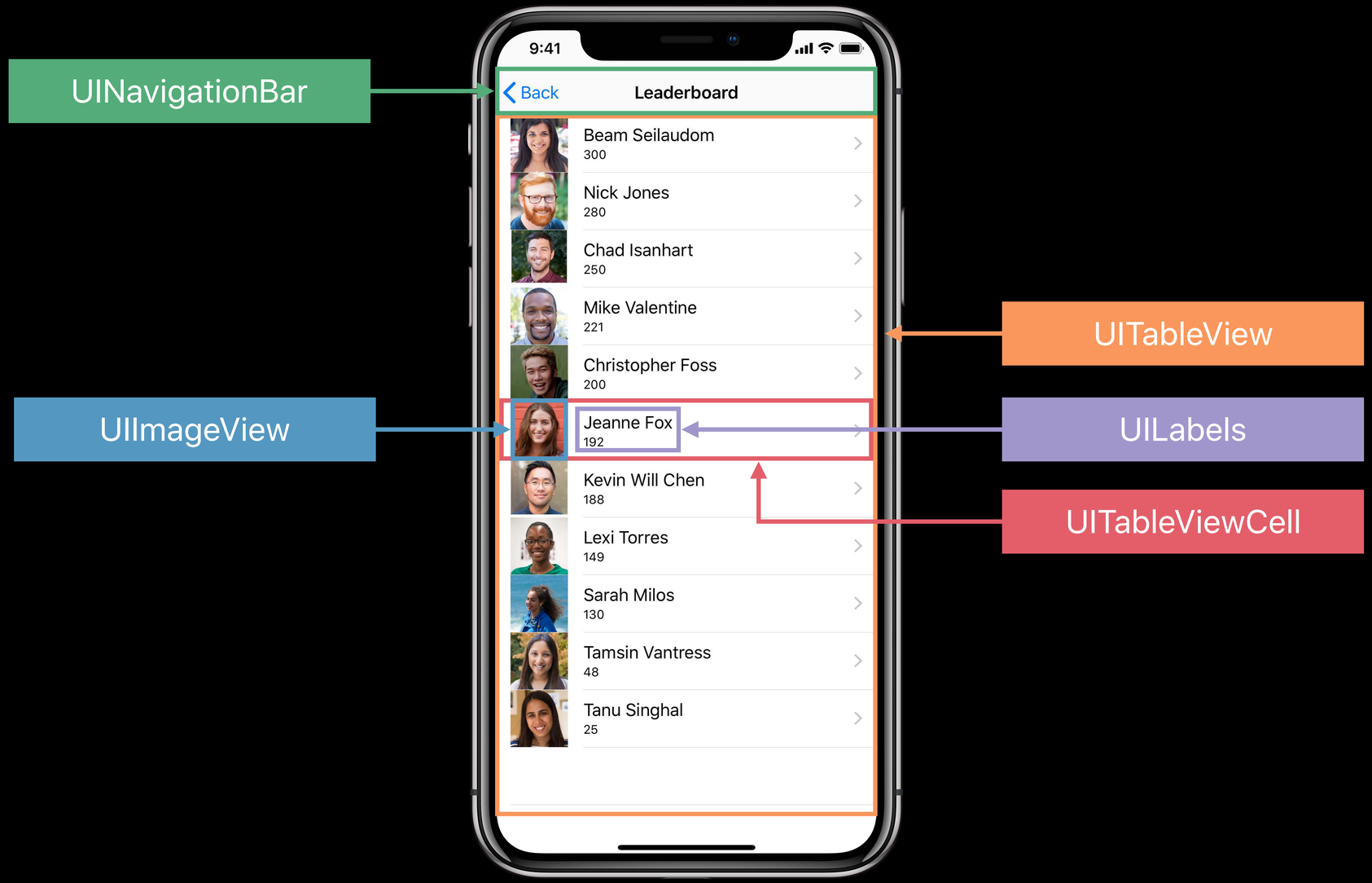


Leaderboard

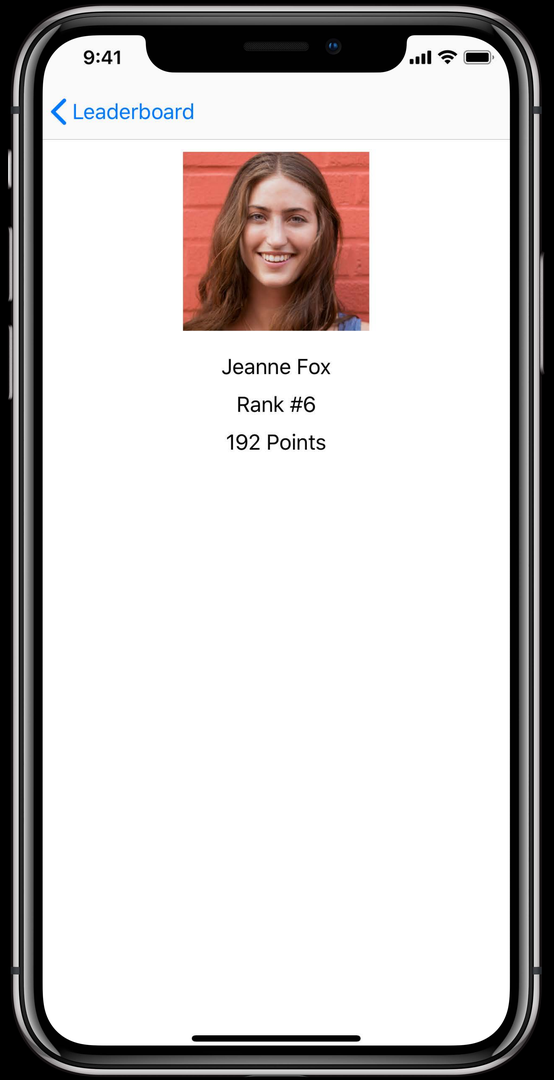
Leaderboard



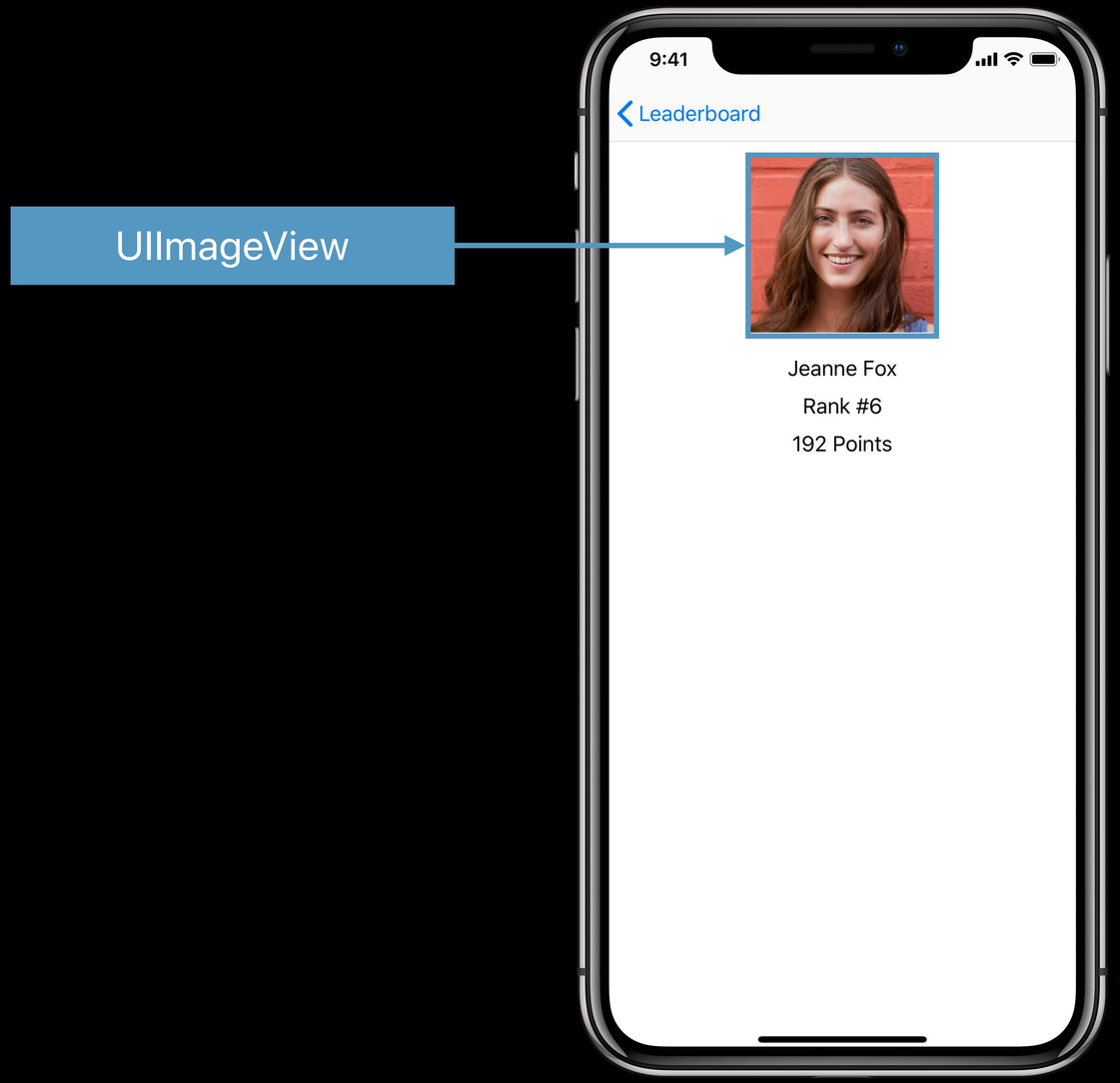
Leaderboard



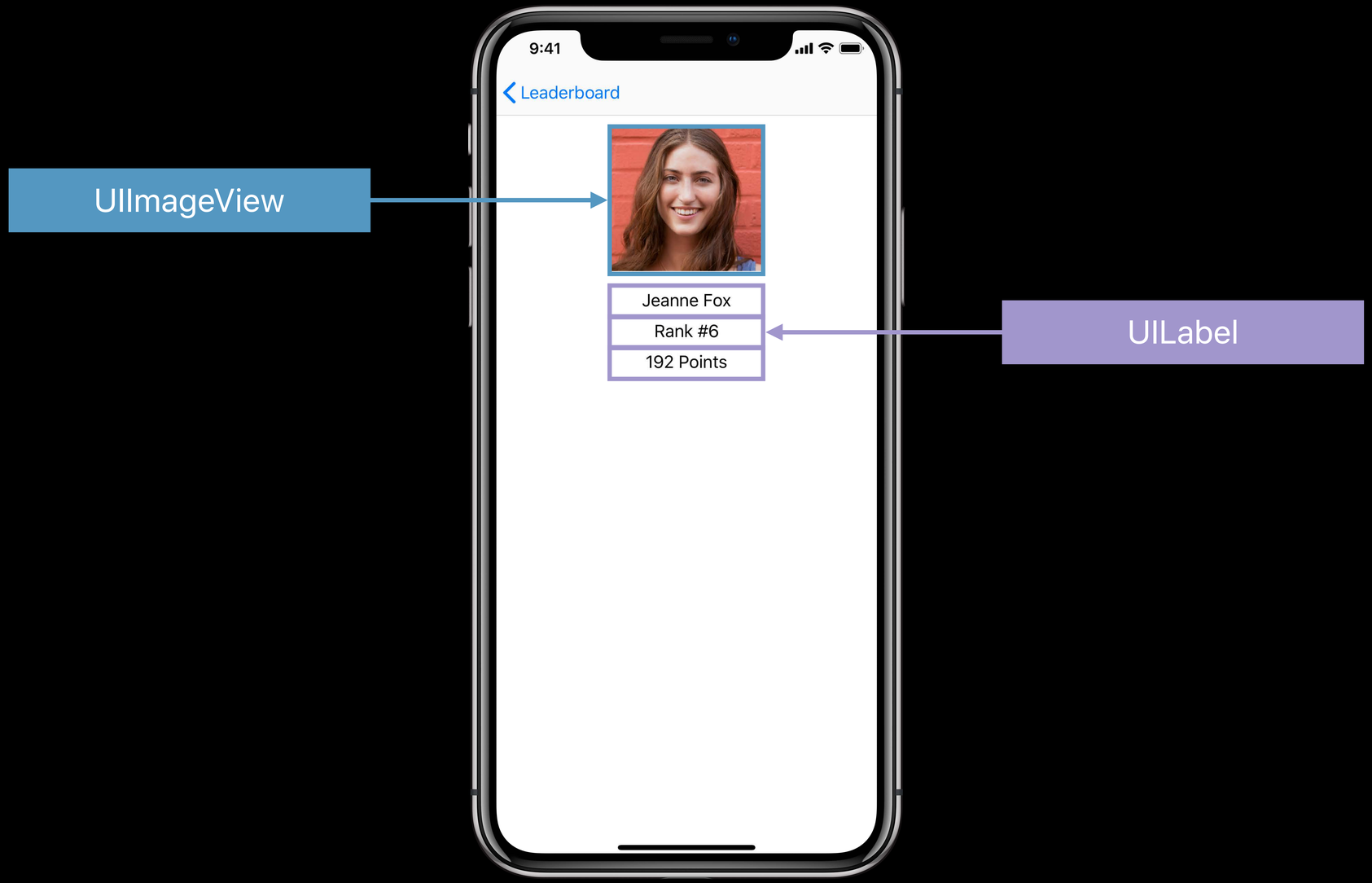
Details



Details



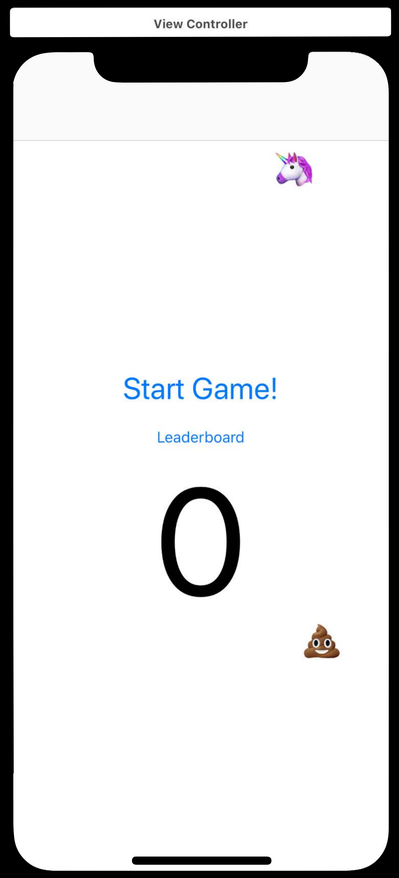
Details


UIImageView
9:41
Leaderboard
Jeanne Fox
Rank #6
192 Points
UILabel

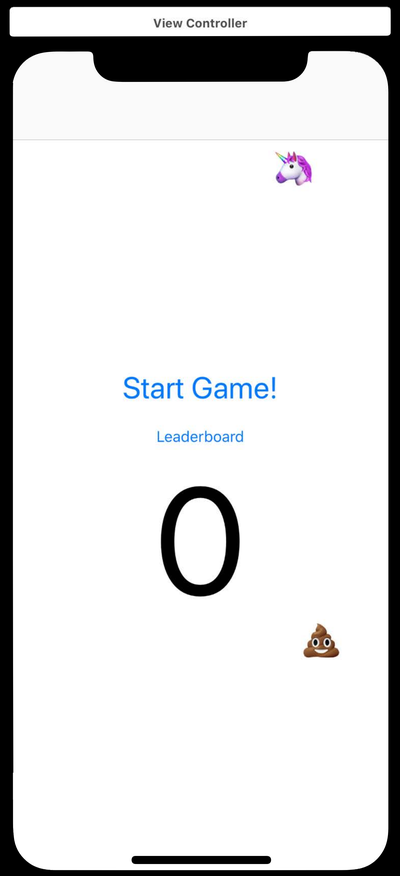
Demo

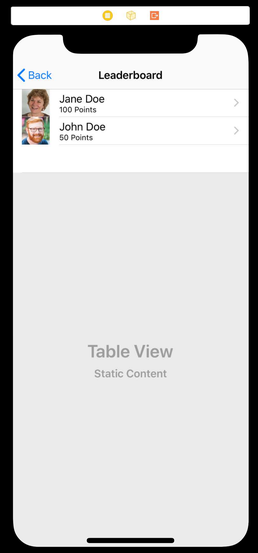
Creating the user interface

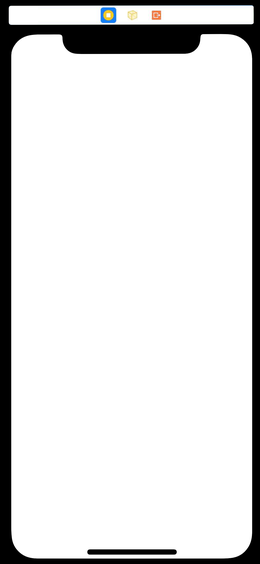
View Controllers

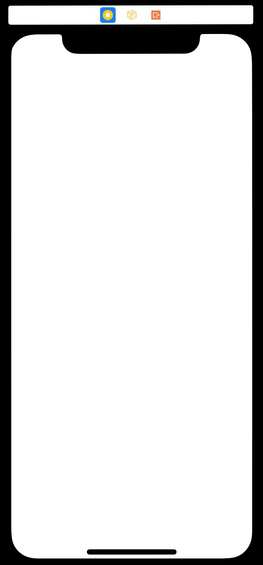


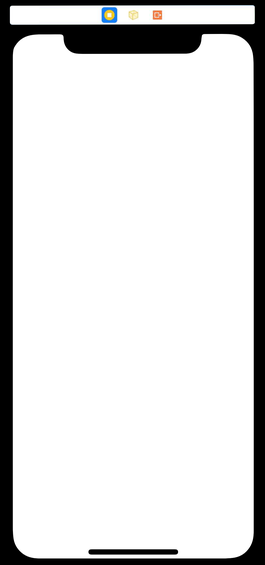
View Controllers



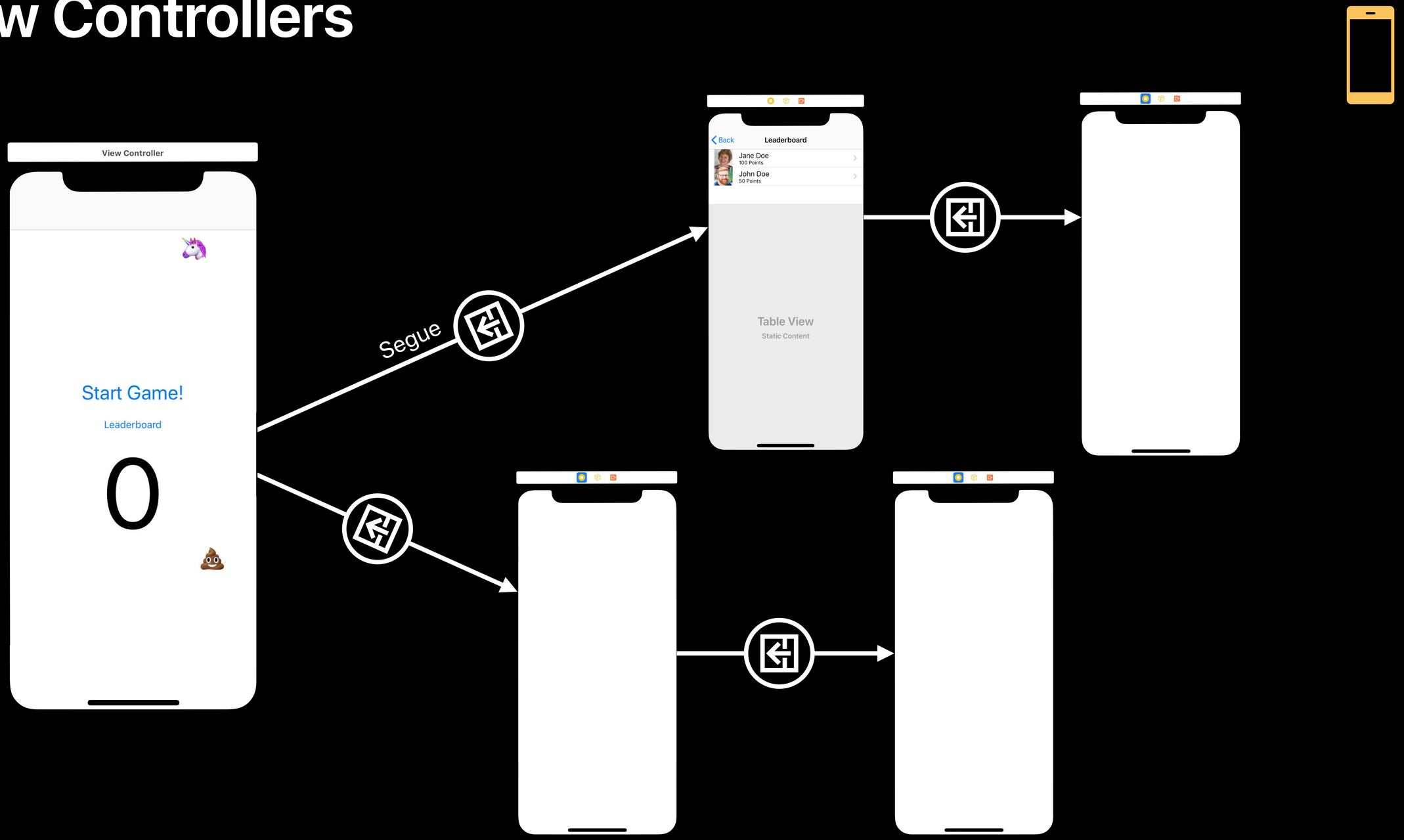




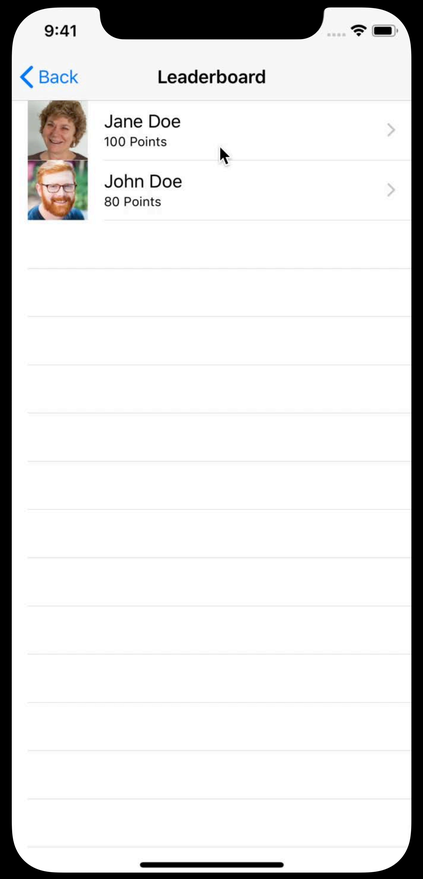




View Controllers

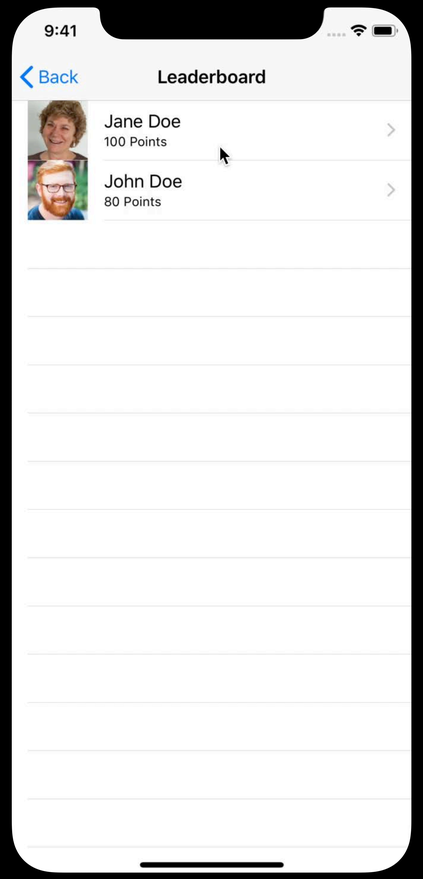


Navigation Controllers and Table View Controllers



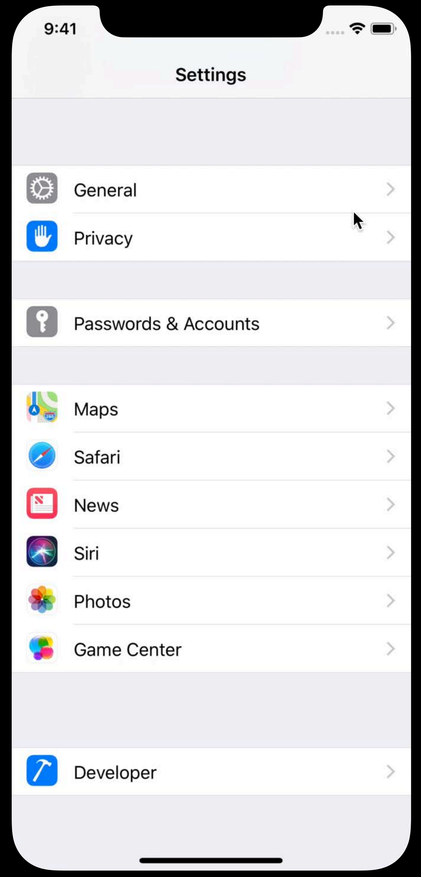
Unicorn Game

Navigation Controllers and Table View Controllers

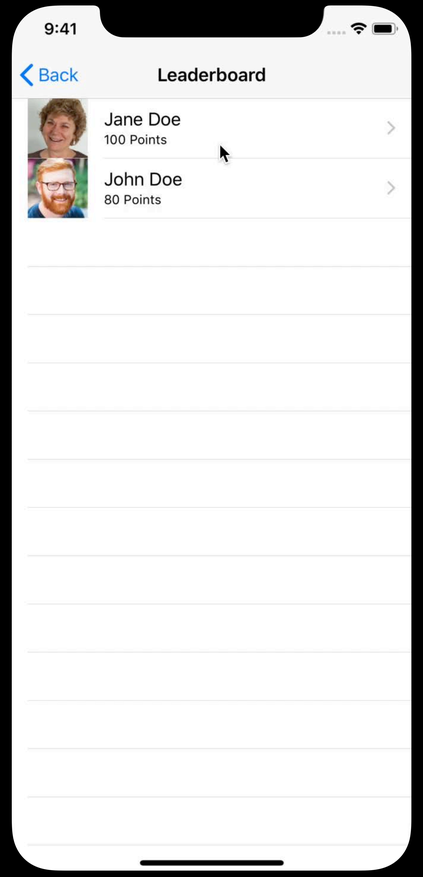


Unicorn Game

Navigation Controllers and Table View Controllers

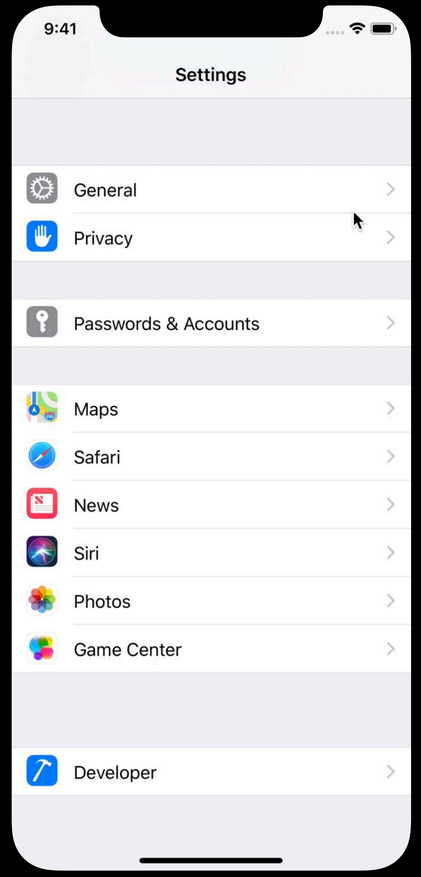


Settings

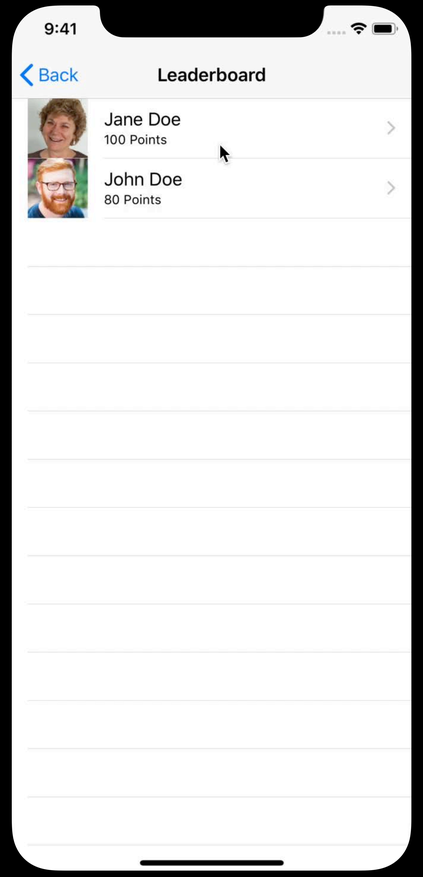


Unicorn Game

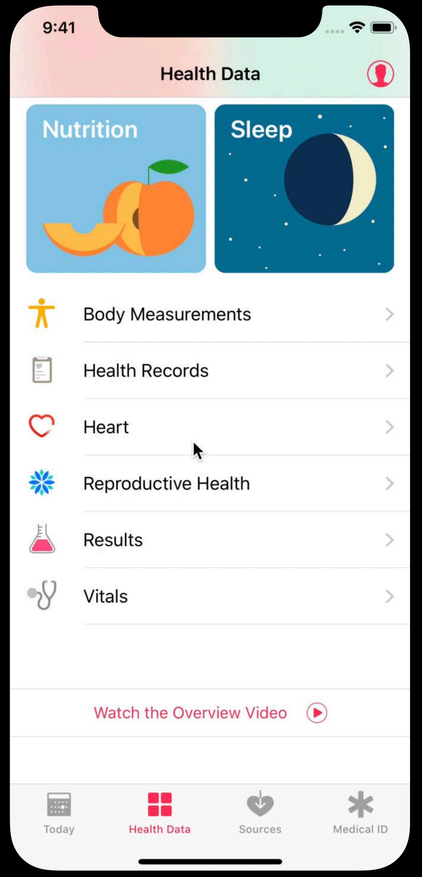
Navigation Controllers and Table View Controllers



Settings



Unicorn Game



Health

View Controllers

View Controller Programming Guide for iOS	Apple Developer Documentation
Table View Programming Guide	Apple Developer Documentation
Navigation Controllers	Apple Developer Documentation

Auto Layout

Mysteries of Auto Layout, Part 1		WWDC 2015
Auto Layout Techniques in Interface Builder		WWDC 2017
High Performance Auto Layout	Hall 2	Wednesday 3:00PM

Demo

View controller and logic

Next Steps

Next Steps

Test your app using XCTest framework

Next Steps

Test your app using XCTest framework

Review App Store Review Guidelines

Next Steps

Test your app using XCTest framework

Review App Store Review Guidelines

Enroll in the Apple Developer Program

Next Steps

Test your app using XCTest framework

Review App Store Review Guidelines

Enroll in the Apple Developer Program

Submit for App Review

Next Steps

Test your app using XCTest framework

Review App Store Review Guidelines

Enroll in the Apple Developer Program

Submit for App Review

Tell the world!

Summary

Summary

Explore Xcode

Summary

Explore Xcode

Build a user interface

Summary

Explore Xcode

Build a user interface

Think about the data

Summary

Explore Xcode

Build a user interface

Think about the data

Create good experiences for all devices

Summary

Explore Xcode

Build a user interface

Think about the data

Create good experiences for all devices

Follow best practices

Summary

Explore Xcode

Build a user interface

Think about the data

Create good experiences for all devices

Follow best practices

Congratulations! You’re an iOS App Developer!

More Information

https://developer.apple.com/wwdc2018/203


WWDC18