



Xcode

Giftbot

Xcode

통합 개발 환경 (IDE, Integrated Development) = Editor + Compiler + Debugger + ...

e.g. **Visual Studio, Eclipse, Delphi, Net Beans** 등

macOS에서만 실행 가능

공식 지원 언어 : Swift, Objective-C, C, C++, Fotran, Ruby, Python, Java 등

History

- Xcode 이전 프로젝트 빌더라는 프로그램으로 사용되던 것이 2003년 OS X 와 함께 공개
- Xcode 3 - OS X Leopard (10.5), iOS 개발 가능
- Xcode 4 - 인터페이스 빌더가 내부 모듈로 탑재, ARC 소개, LLVM 컴파일러로 넘어가는 과도기
- Xcode 5 - LLVM 이 메인 컴파일러로 변경, nib 위주에서 스토리보드 위주로 변화
- Xcode 6 - Swift 등장 및 Playground 지원. Xcode 라이선스 무료화
- Xcode 7 - Swift 2 지원. OS X, iOS, watchOS 앱을 만들 수 있도록 개발환경 지원
- Xcode 8 - Swift 3 지원. 스토리보드 개선
- Xcode 9 - Swift 4 지원. 시뮬레이터 다중 실행, 무선 디버깅 등 지원

Welcome to Xcode



Welcome to Xcode

Version 10.1 (10B61)

No Recent Projects



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

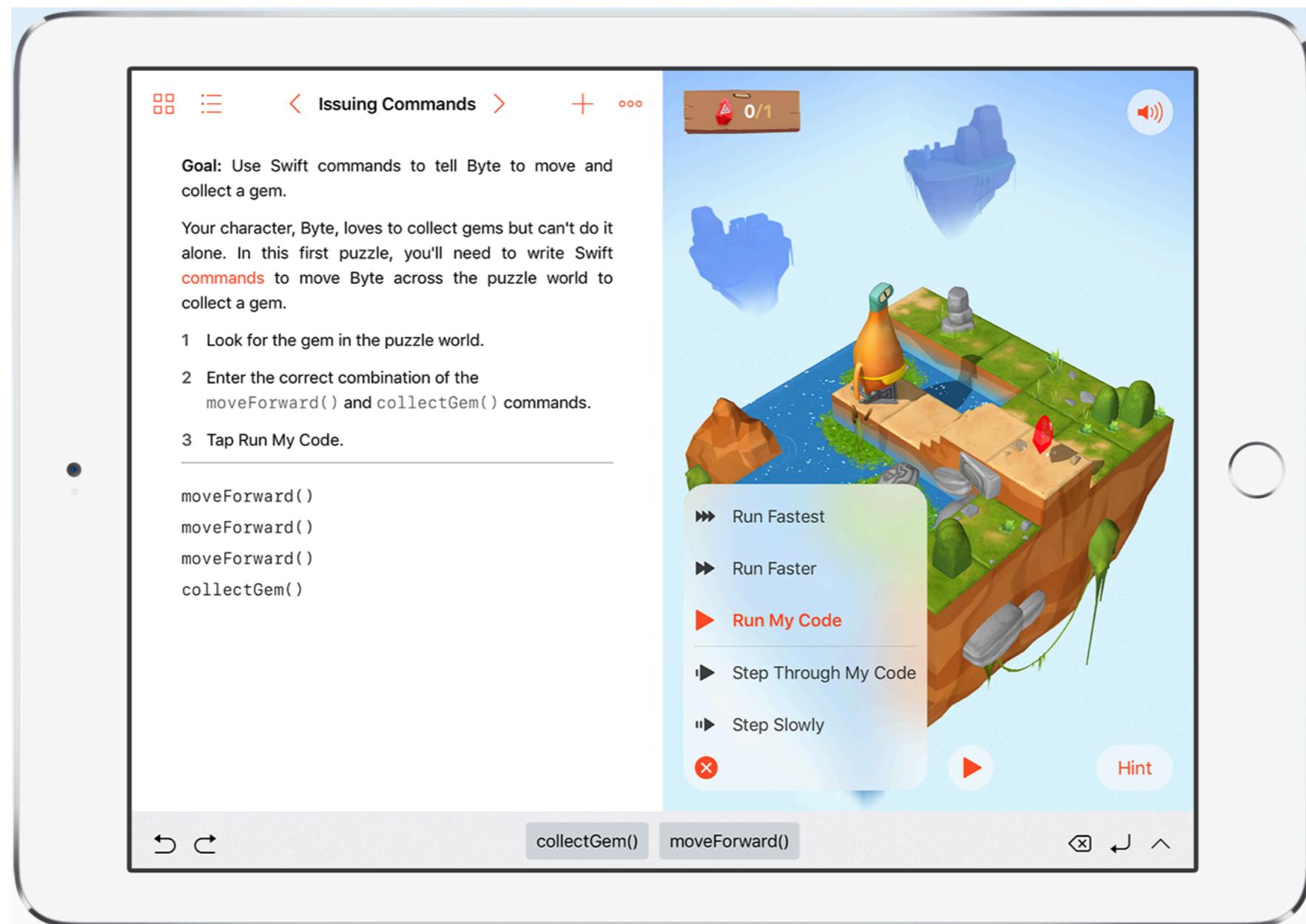
Start working on something from a Git repository.

Open another project...

Playground

Playground

Create small programs called “playgrounds” that instantly show the results of the code that you write



Get started with a playground



Welcome to Xcode

Version 10.1 (10B61)

No Recent Projects



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.

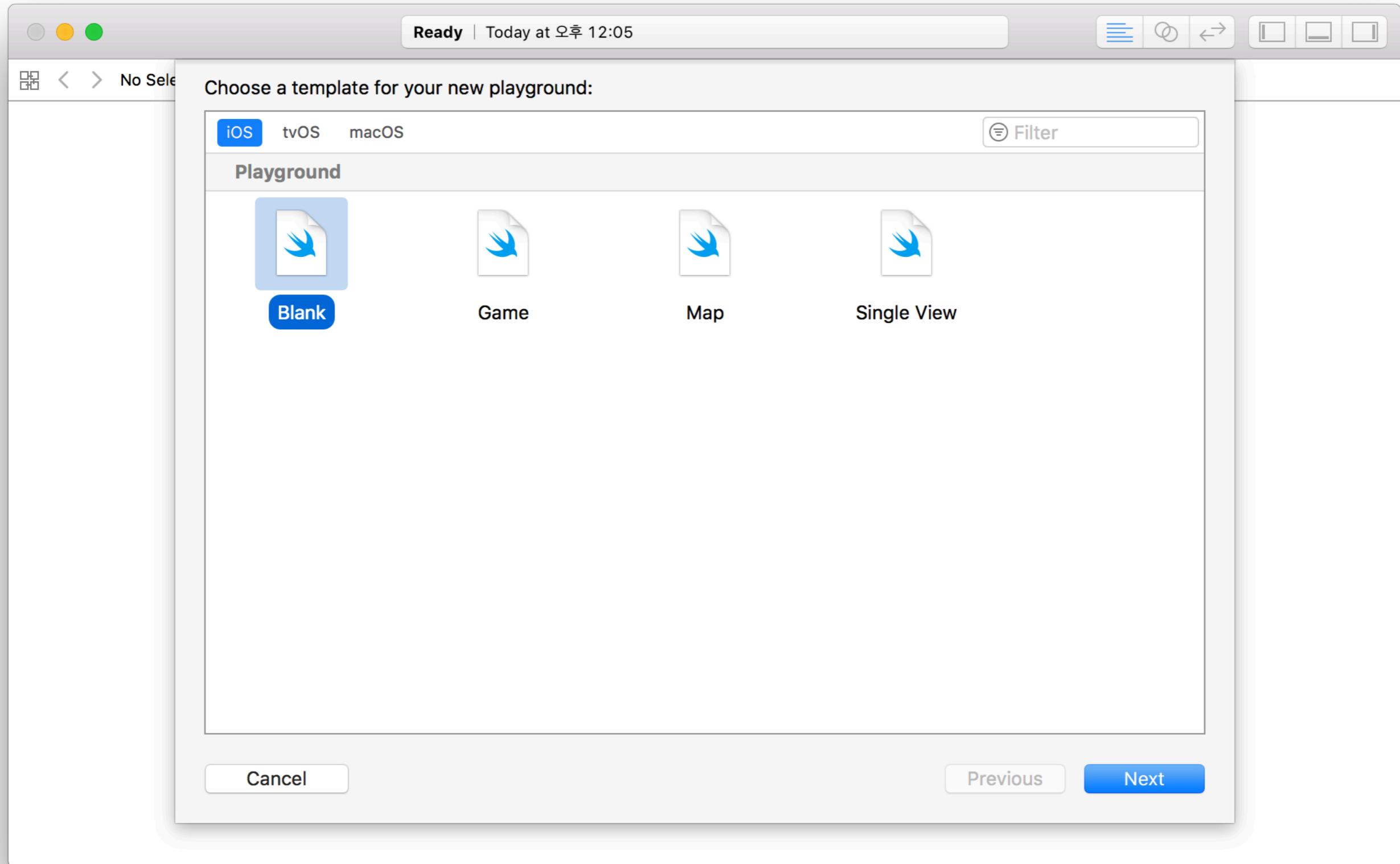


Clone an existing project

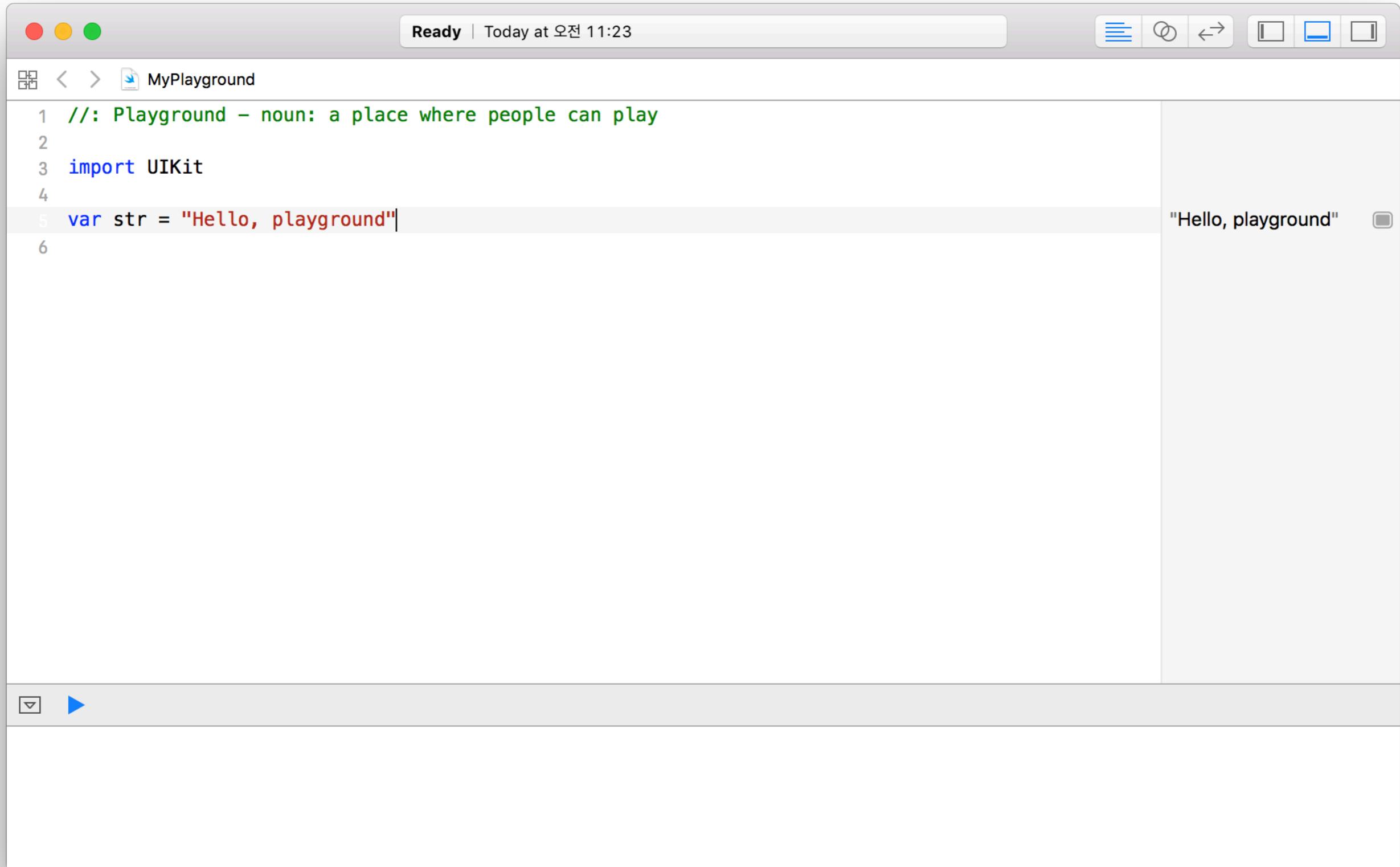
Start working on something from a Git repository.

Open another project...

Templates



Blank



The screenshot shows the Xcode playground interface. The top bar displays "Ready | Today at 오전 11:23". The title bar says "MyPlayground". The main editor area contains the following Swift code:

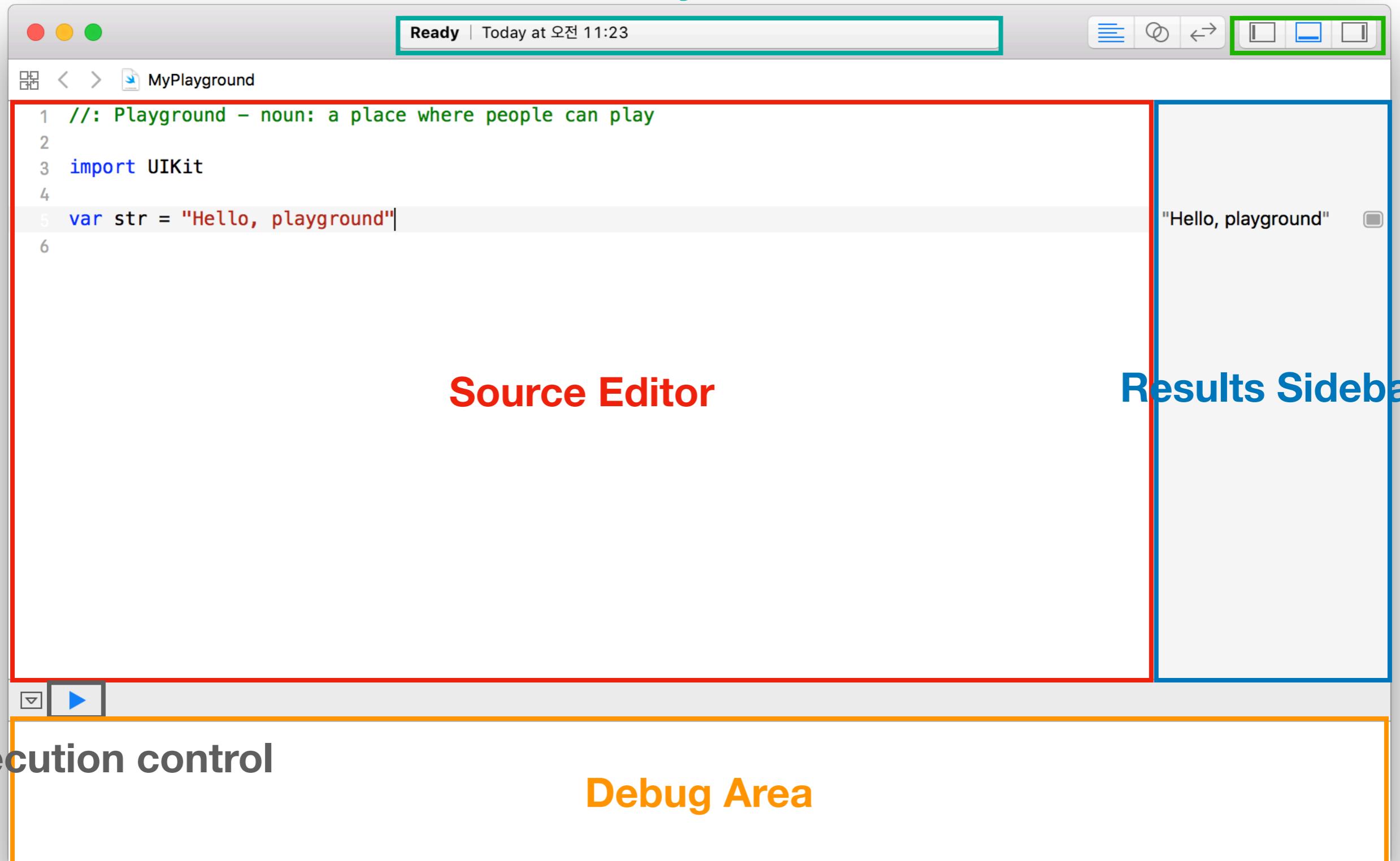
```
1 //: Playground - noun: a place where people can play
2
3 import UIKit
4
5 var str = "Hello, playground"
6
```

To the right of the code, the playground's output window shows the result: "Hello, playground". The bottom of the screen features the Xcode navigation bar with a disclosure triangle icon and a blue arrow icon.

Playground Area

Activity Viewer

Panel Controls



Game

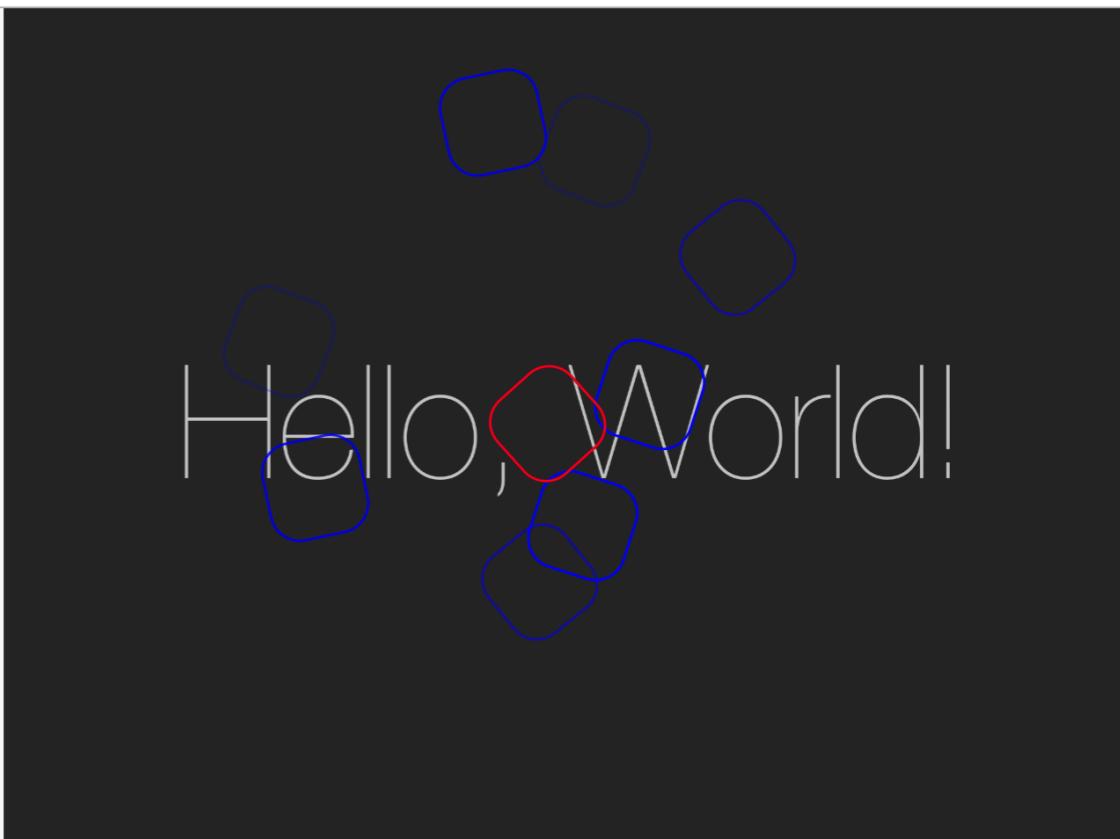
A SpriteKit based Playground

```

2 import PlaygroundSupport
3 import SpriteKit
4
5 class GameScene: SKScene {
6
7     private var label : SKLabelNode!
8     private var spinnyNode : SKShapeNode!
9
10    override func didMove(to view: SKView) {
11        // Get label node from scene and store it for use later
12        label = childNode(withName: "//helloLabel") as? SKLabelNode
13        label.alpha = 0.0
14        let fadeInOut = SKAction.sequence([.fadeIn(withDuration:
15            2.0),
16                                         .fadeOut(withDuration:
17                                             2.0)])
18        label.run(.repeatForever(fadeInOut))
19
20        // Create shape node to use during mouse interaction
21        let w = (size.width + size.height) * 0.05
22
23        spinnyNode = SKShapeNode(rectOf: CGSize(width: w, height: w),
24                               cornerRadius: w * 0.3)
25        spinnyNode.lineWidth = 2.5
26
27        let fadeAndRemove = SKAction.sequence([.wait(forDuration:
28            0.5),
29                                         .fadeOut(withDuration:
30                                             0.5),
31                                         .removeFromParent()])
32
33        spinnyNode.run(.repeatForever(.rotate(byAngle:
34            CGFloat(Double.pi), duration: 1)))
35        spinnyNode.run(fadeAndRemove)
36    }

```

Live View Game.playground (Live View)



The screenshot shows the Xcode interface with a playground file open. The left pane displays the Swift code for a SpriteKit scene. The right pane shows the live preview of the game scene. In the preview, the text "Hello, World!" is displayed in white, and several blue-outlined circles are scattered around it. The Xcode interface includes standard window controls, a toolbar, and a sidebar with various icons.

Map

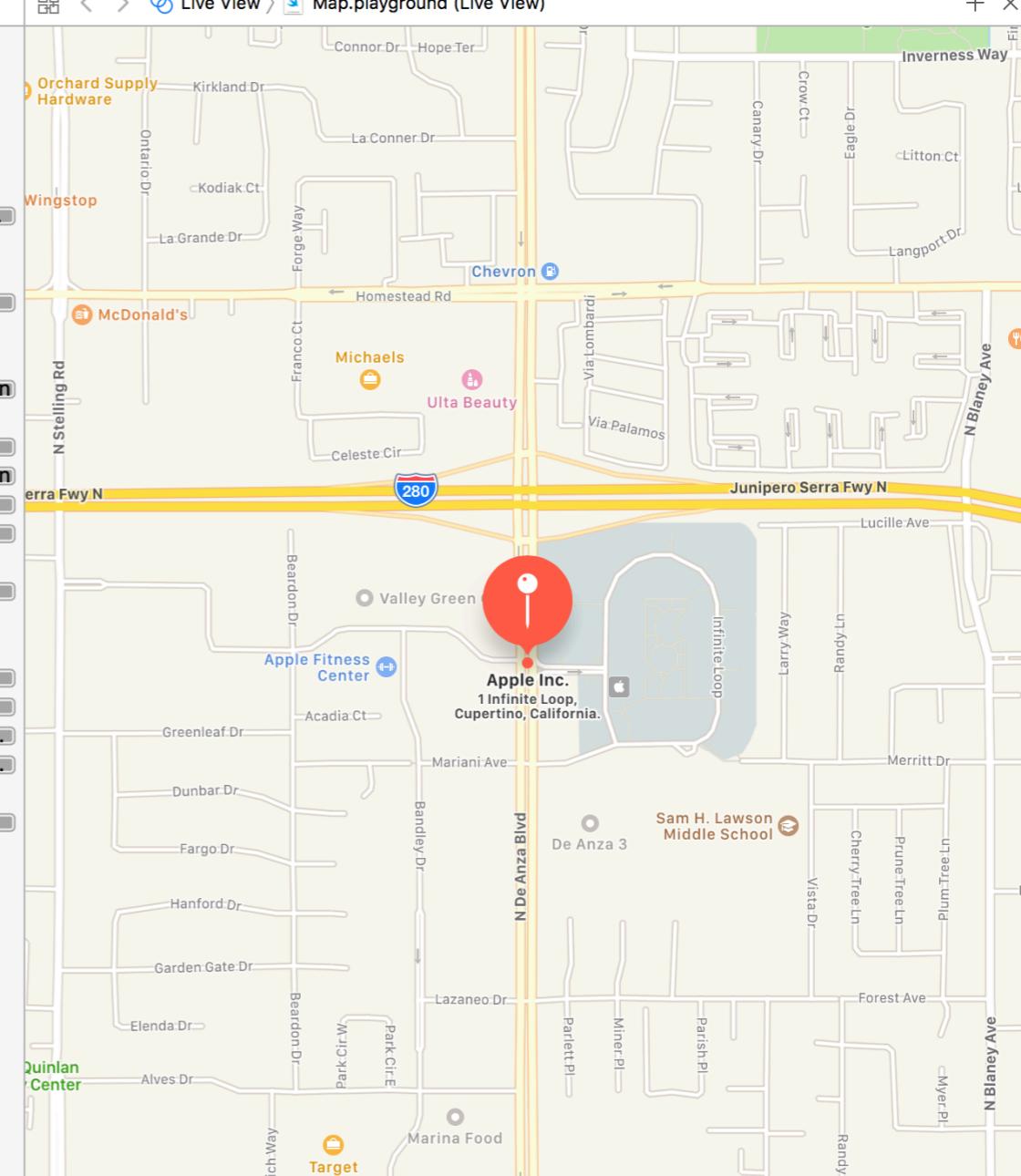
Running Map

```

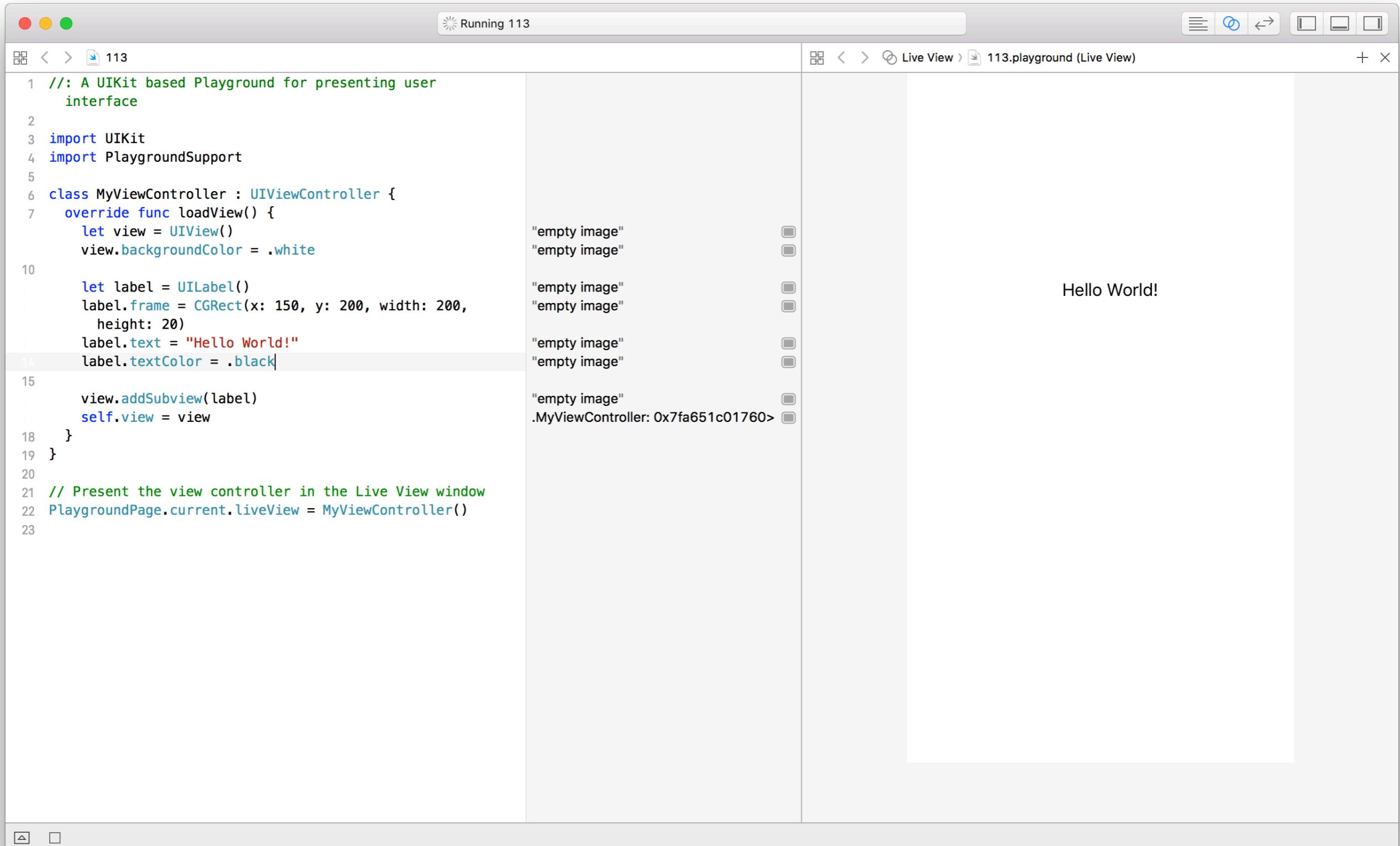
1 //: A MapKit based Playground
2
3 import MapKit
4 import PlaygroundSupport
5
6
7 let infiniteLoopCoord = CLLocationCoordinate2DMake(37.331695, -122.0322854)
8
9 // Now let's create a MKMapView
10 let mapView = MKMapView(frame: CGRect(x:0, y:0, width:800, height:800))
11
12 // Define a region for our map view
13 var mapRegion = MKCoordinateRegion()
14
15 let mapRegionSpan = 0.02
16 mapRegion.center = infiniteLoopCoord
17 mapRegion.span.latitudeDelta = mapRegionSpan
18 mapRegion.span.longitudeDelta = mapRegionSpan
19
20 mapView.setRegion(mapRegion, animated: true)
21
22 // Create a map annotation
23 let annotation = MKPointAnnotation()
24 annotation.coordinate = infiniteLoopCoord
25 annotation.title = "Apple Inc."
26 annotation.subtitle = "1 Infinite Loop, Cupertino, California."
27
28 mapView.addAnnotation(annotation)
29
30 // Add the created mapView to our Playground Live View
31 PlaygroundPage.current.liveView = mapView
32

```

Live View > Map.playground (Live View)



Single View



The screenshot shows a Xcode playground window titled "Running 113". The left panel contains Swift code for a `MyViewController` class. The right panel shows the "Live View" where the code runs, displaying a single `UILabel` with the text "Hello World!".

```
1 //: A UIKit based Playground for presenting user
  interface
2
3 import UIKit
4 import PlaygroundSupport
5
6 class MyViewController : UIViewController {
7     override func loadView() {
8         let view = UIView()
9         view.backgroundColor = .white
10
11     let label = UILabel()
12     label.frame = CGRect(x: 150, y: 200, width: 200,
13         height: 20)
14     label.text = "Hello World!"
15     label.textColor = .black
16
17     view.addSubview(label)
18     self.view = view
19 }
20
21 // Present the view controller in the Live View window
22 PlaygroundPage.current.liveView = MyViewController()
23
```

The "Live View" window displays the output of the code execution. It shows a single `UILabel` with the text "Hello World!". The label has a black border and is centered on the screen. The background is white, and there are some empty image placeholders visible on the right side of the window.

Xcode Project

New Project

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

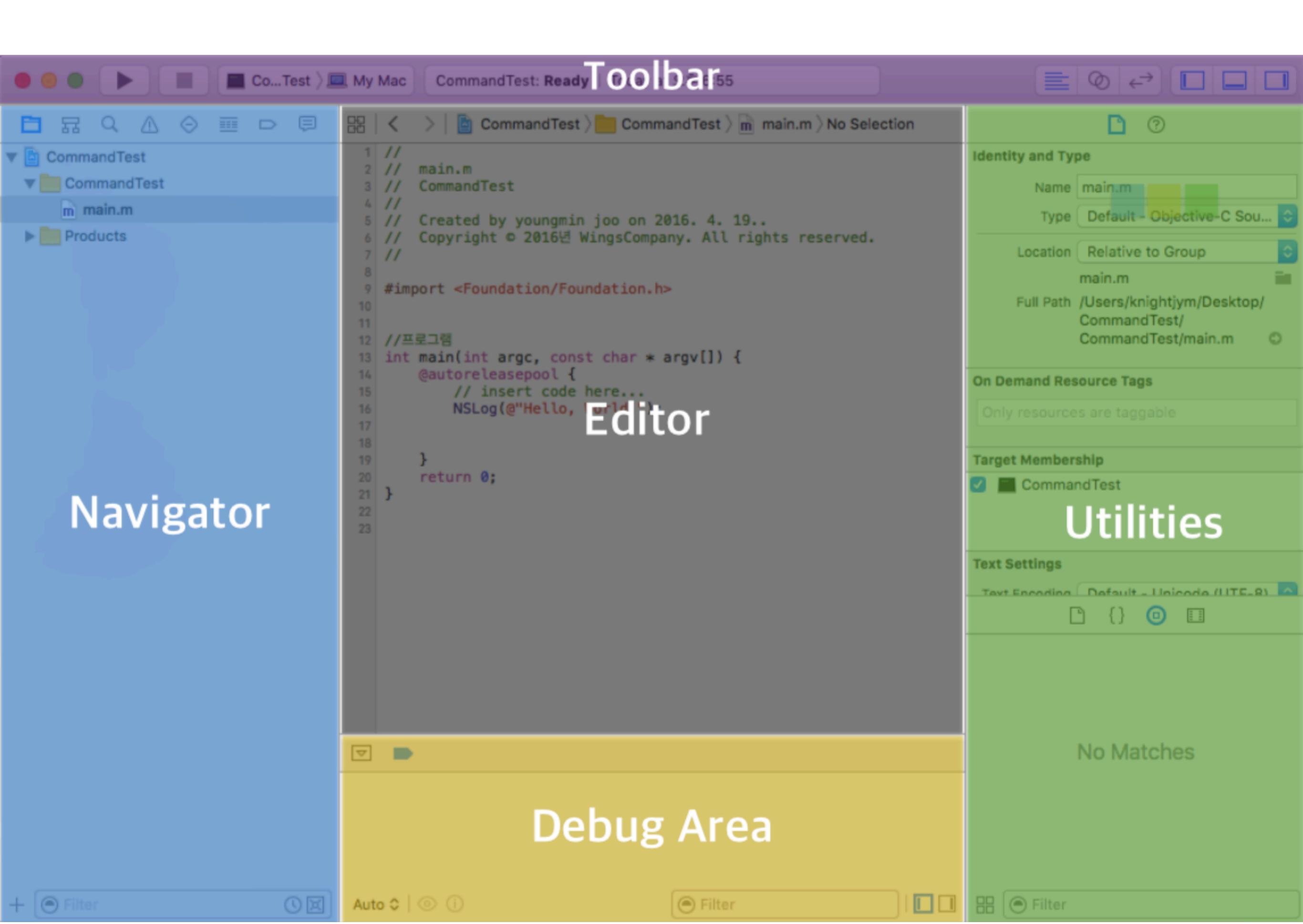


Metal Library

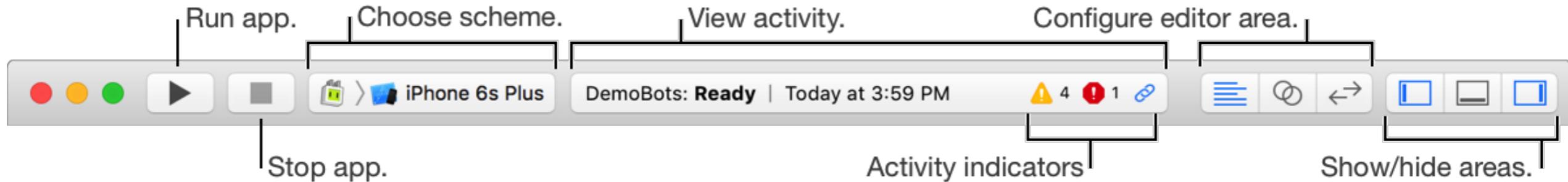
Cancel

Previous

Next

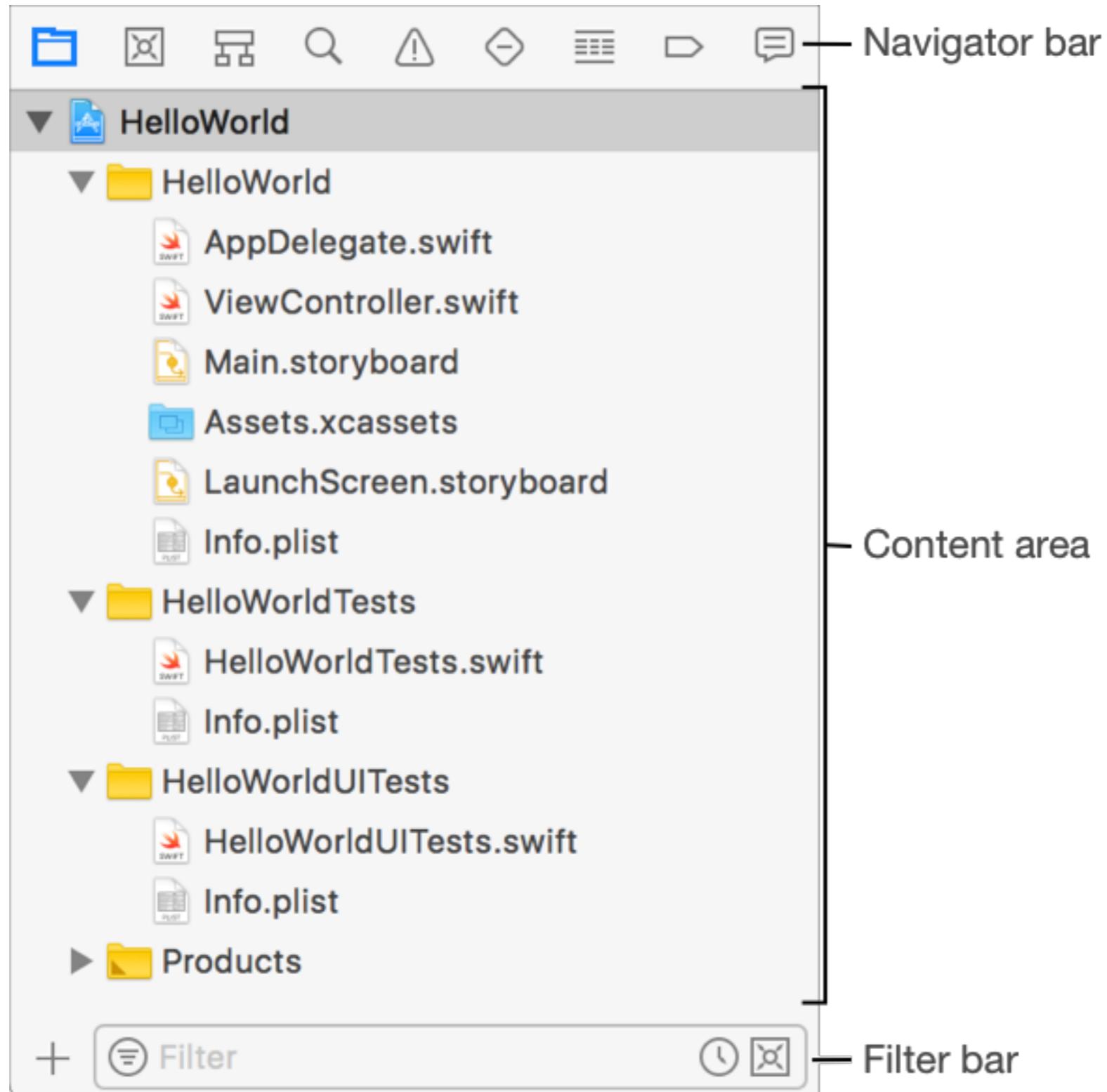


Toolbar



Indicator	Name	Description
⚠	Warning	Appears when there are issues with the project. Clicking on the warnings indicator opens the Issues navigator.
❗	Error	Appears when there are errors that will prevent the project from building. Clicking on the errors indicator opens the Issues navigator.
⚠	Ambiguity	Appears when the view hierarchy in the current view debugger session contains ambiguous constraints.
🔗	Alternative toolchain	Appears when an alternative toolchain is selected. Clicking on the button opens the Toolchains pane of Components preferences.

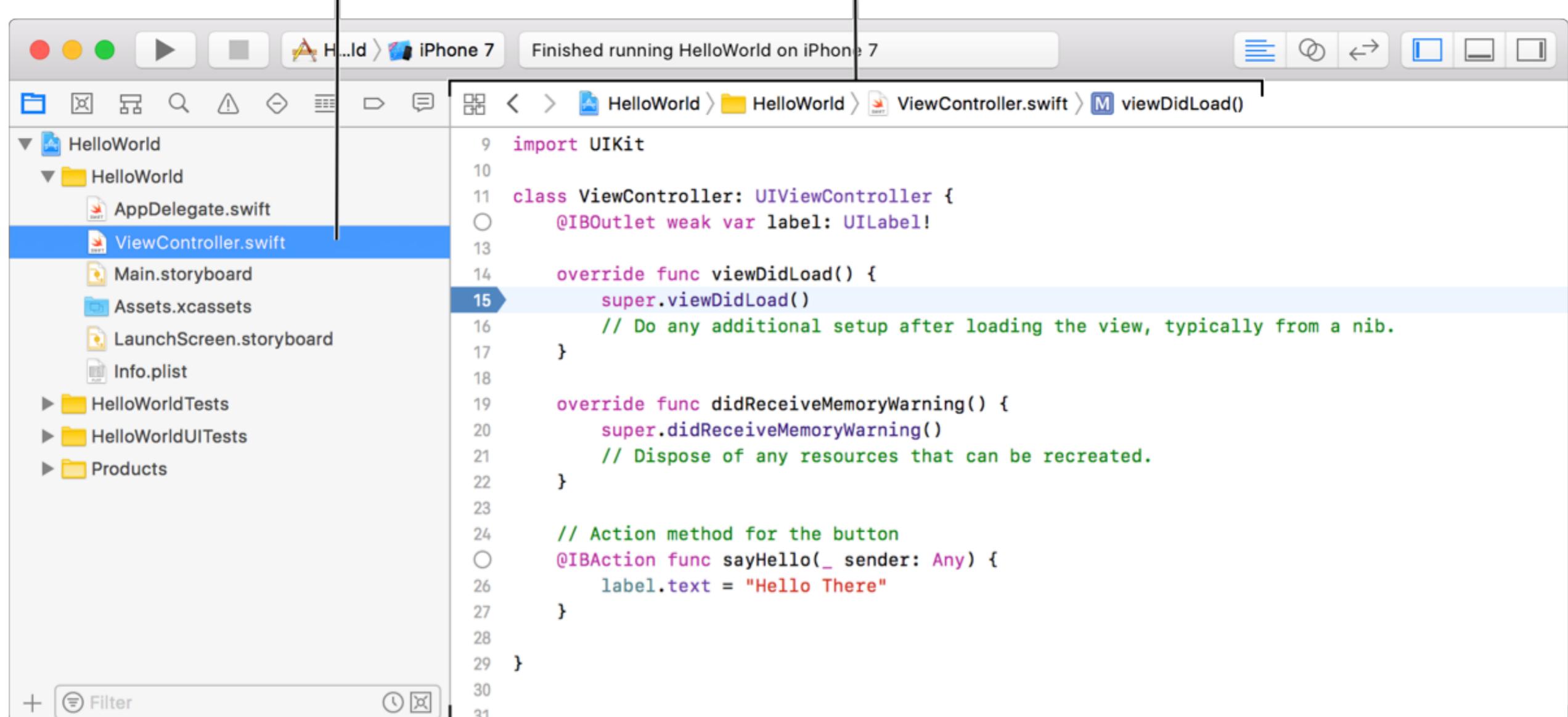
Navigator Area



Navigator Area

Select a source file.

Jump to a location in the file.

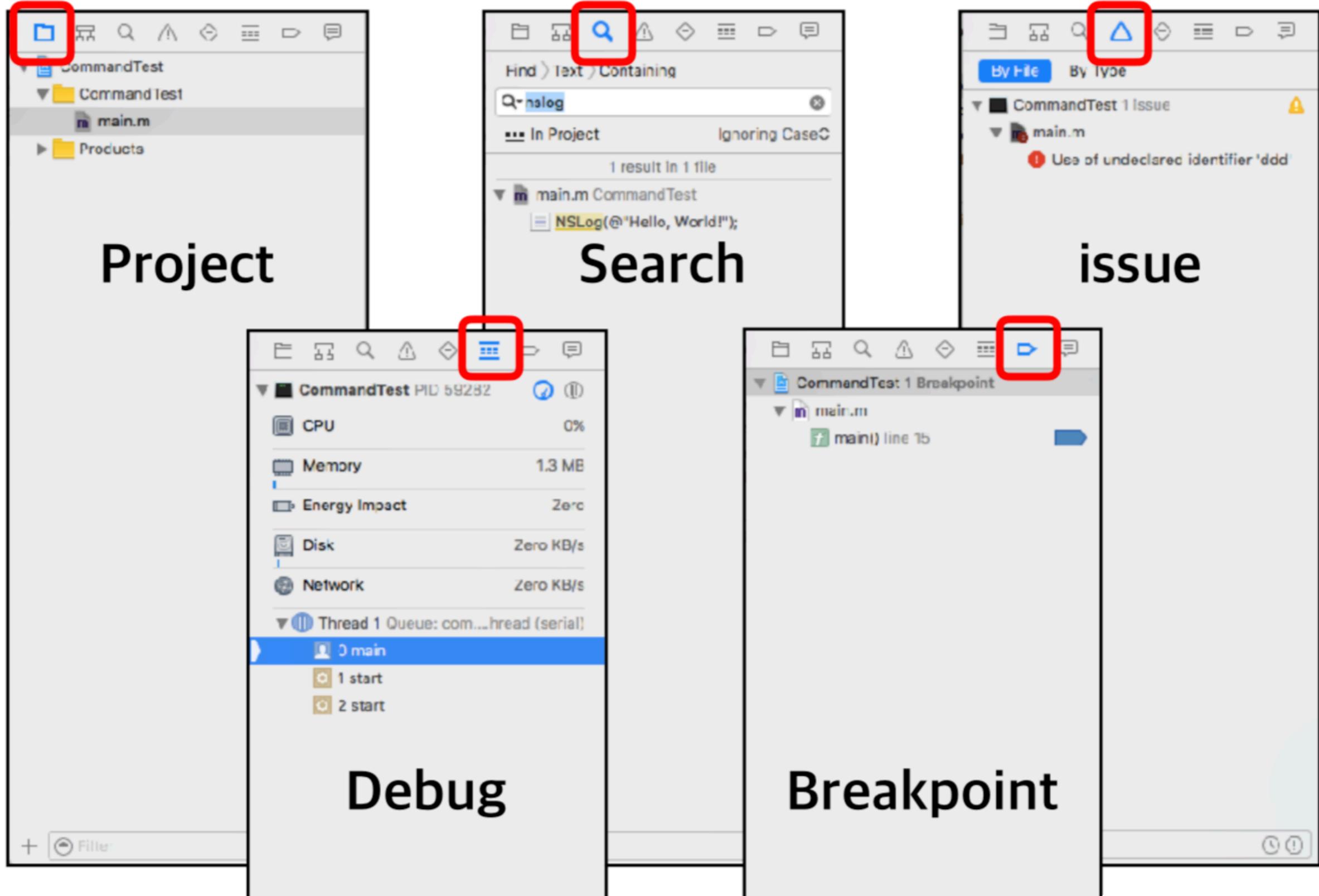


The screenshot shows the Xcode interface with the Navigator area highlighted. In the Navigator, 'ViewController.swift' is selected under the 'HelloWorld' group. The Source editor below shows the Swift code for 'ViewController.swift', specifically the implementation of the 'viewDidLoad()' method. The code is color-coded for syntax. A blue arrow points from the 'ViewController.swift' entry in the Navigator to the corresponding line in the Source editor.

```
9 import UIKit
10
11 class ViewController: UIViewController {
12     @IBOutlet weak var label: UILabel!
13
14     override func viewDidLoad() {
15         super.viewDidLoad()
16         // Do any additional setup after loading the view, typically from a nib.
17     }
18
19     override func didReceiveMemoryWarning() {
20         super.didReceiveMemoryWarning()
21         // Dispose of any resources that can be recreated.
22     }
23
24     // Action method for the button
25     @IBAction func sayHello(_ sender: Any) {
26         label.text = "Hello There"
27     }
28
29 }
```

Source editor

Navigator Bar



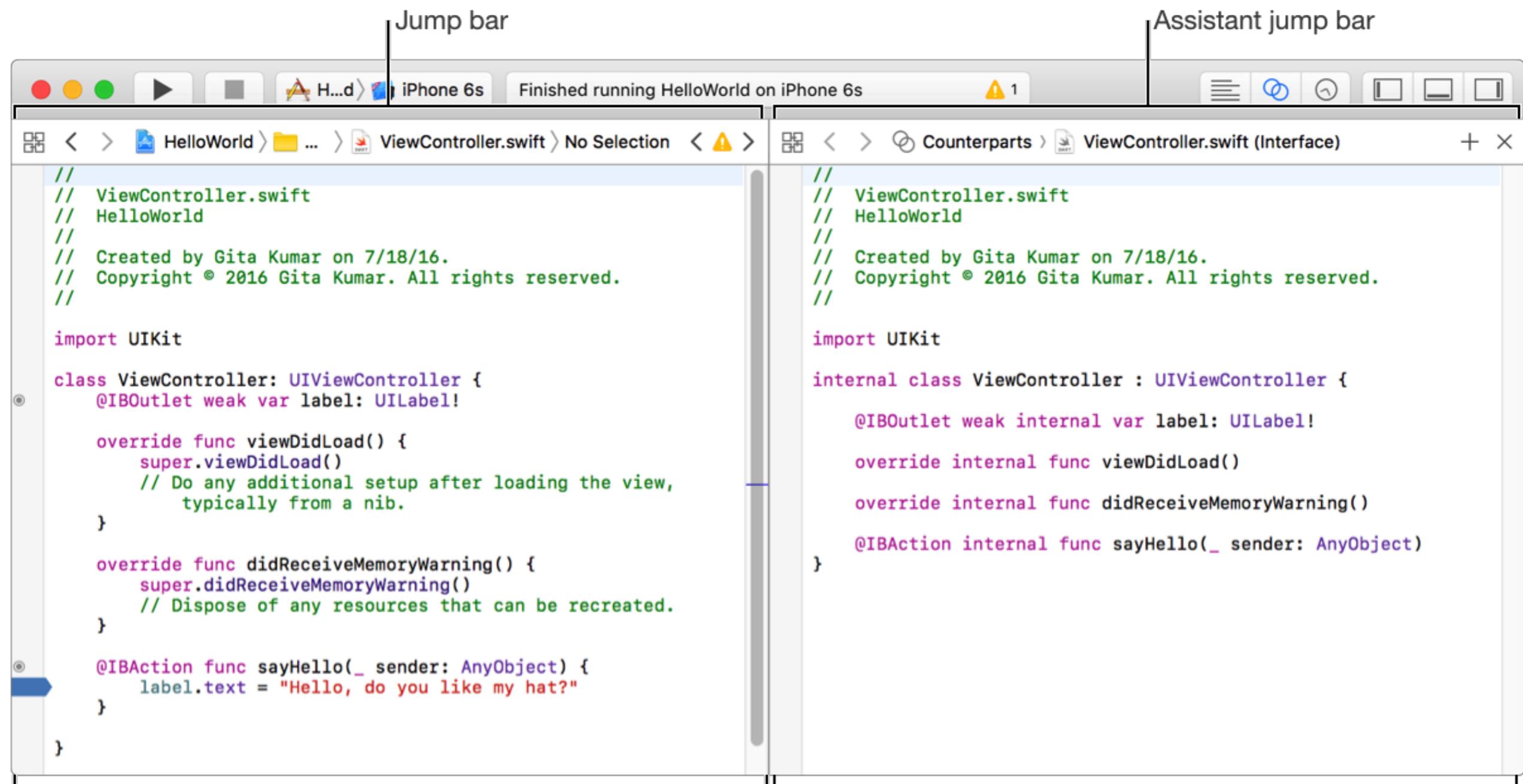
Navigator Bar

Icon	Navigator	Description
	Project navigator	Add, delete, and group files, and open files in the editor.
	Source Control navigator	View your source control working copies, branches, commits, tags, and remote repositories.
	Symbol navigator	Browse the symbols in your project or a specified scope.
	Find navigator	Find any string in your project and frameworks.
	Issue navigator	Display issues found when opening, analyzing, and building your project.
	Test navigator	Create, manage, and run unit tests, and also review their status.
	Debug navigator	Examine running threads and stack information.
	Breakpoint navigator	Add, delete, and edit breakpoints.
	Report navigator	View the history of building, running, and debugging your project, and of source control tasks.

Editor Area

Jump bar

Assistant jump bar



```
// ViewController.swift
// HelloWorld
//
// Created by Gita Kumar on 7/18/16.
// Copyright © 2016 Gita Kumar. All rights reserved.

import UIKit

class ViewController: UIViewController {
    @IBOutlet weak var label: UILabel!

    override func viewDidLoad() {
        super.viewDidLoad()
        // Do any additional setup after loading the view,
        // typically from a nib.
    }

    override func didReceiveMemoryWarning() {
        super.didReceiveMemoryWarning()
        // Dispose of any resources that can be recreated.
    }

    @IBAction func sayHello(_ sender: AnyObject) {
        label.text = "Hello, do you like my hat?"
    }
}
```

Standard editor

```
// ViewController.swift
// HelloWorld
//
// Created by Gita Kumar on 7/18/16.
// Copyright © 2016 Gita Kumar. All rights reserved.

import UIKit

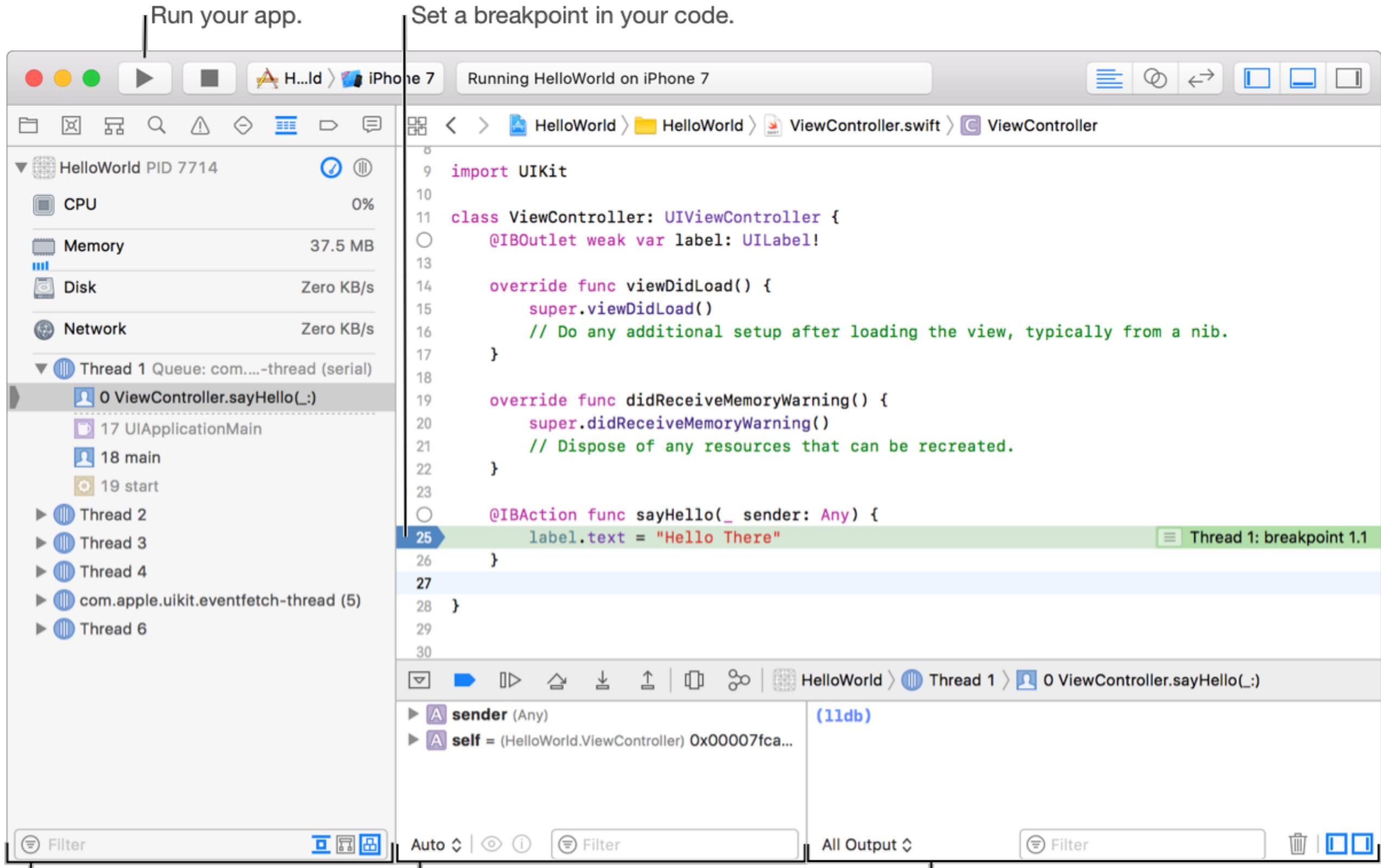
internal class ViewController : UIViewController {

    @IBOutlet weak internal var label: UILabel!

    override internal func viewDidLoad()
    override internal func didReceiveMemoryWarning()
    @IBAction internal func sayHello(_ sender: AnyObject)
}
```

Assistant editor

Debug Area



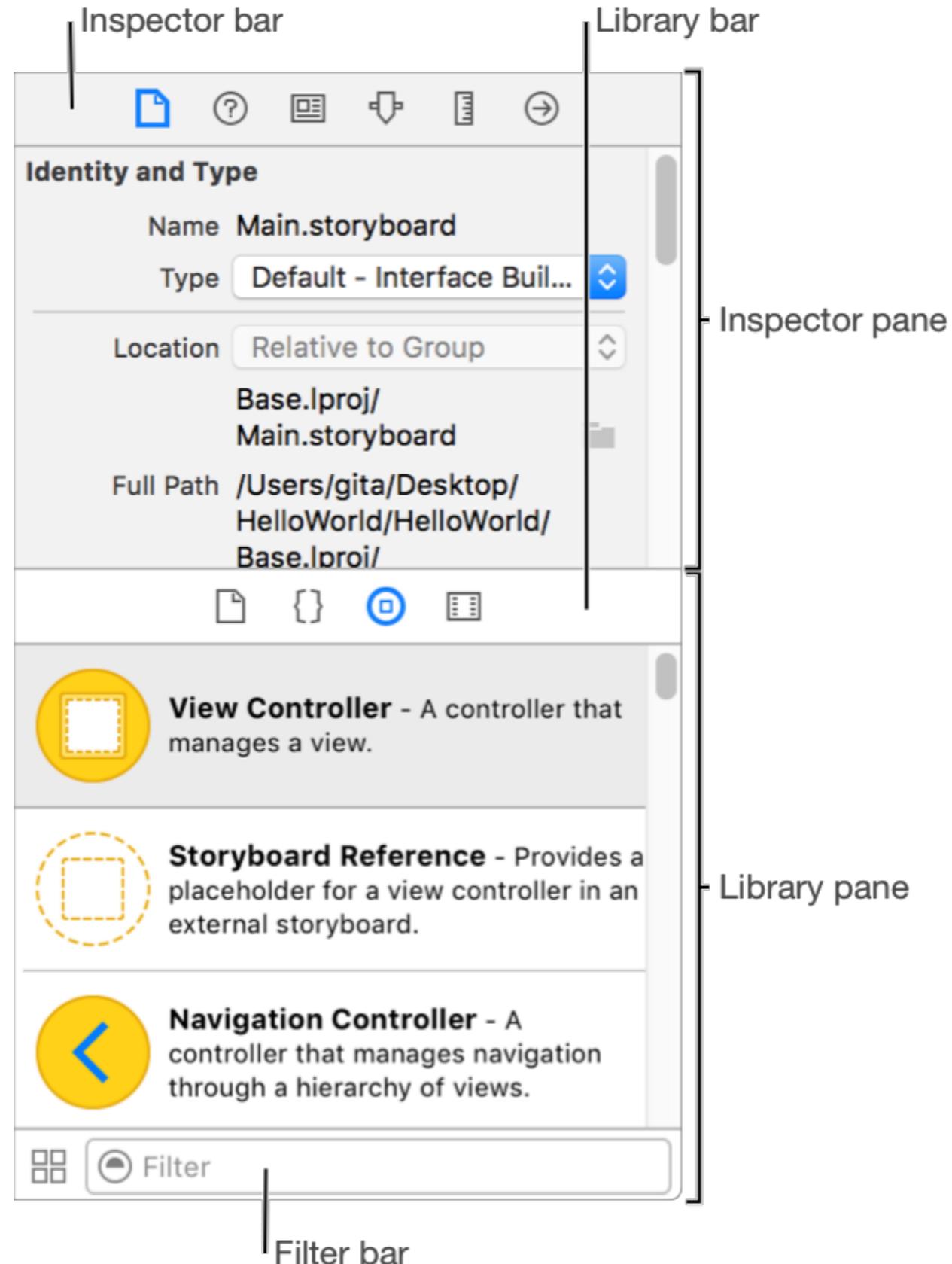
Debug Area



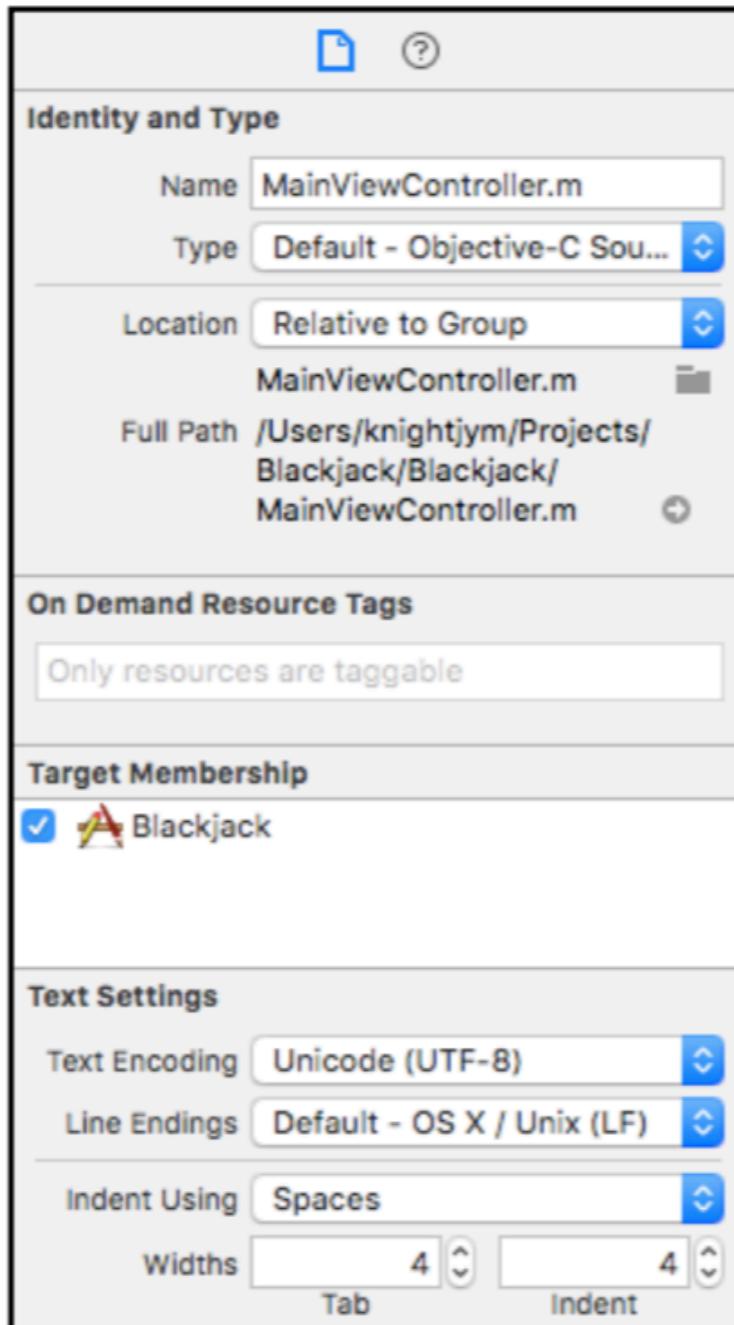
View variable details.

Choose a type of output.

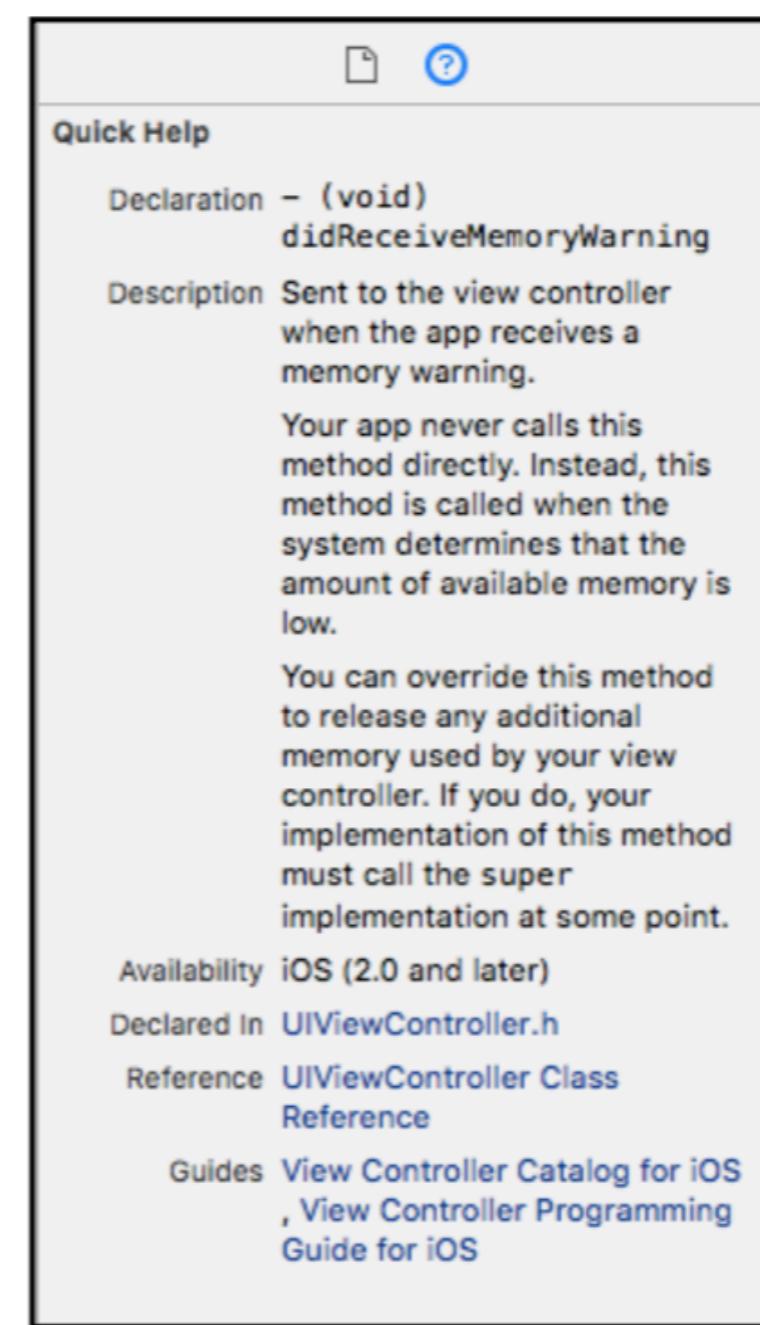
Utilities Area



Utilities Area

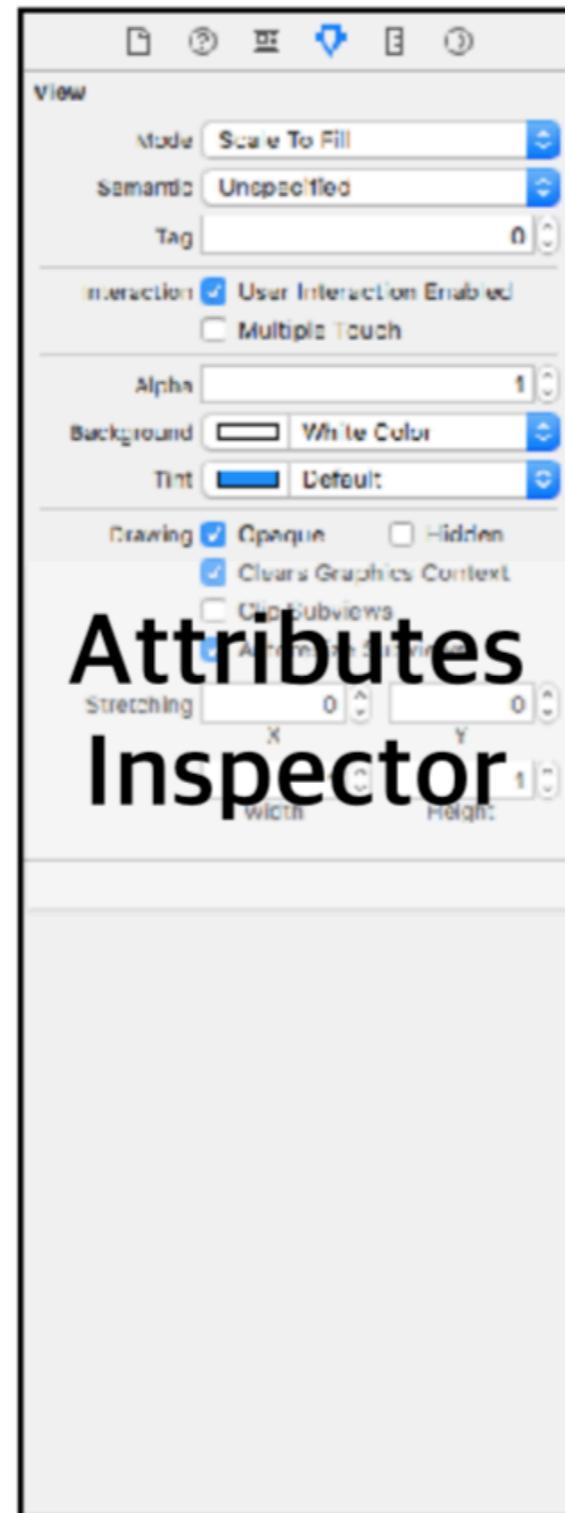


File Inspector



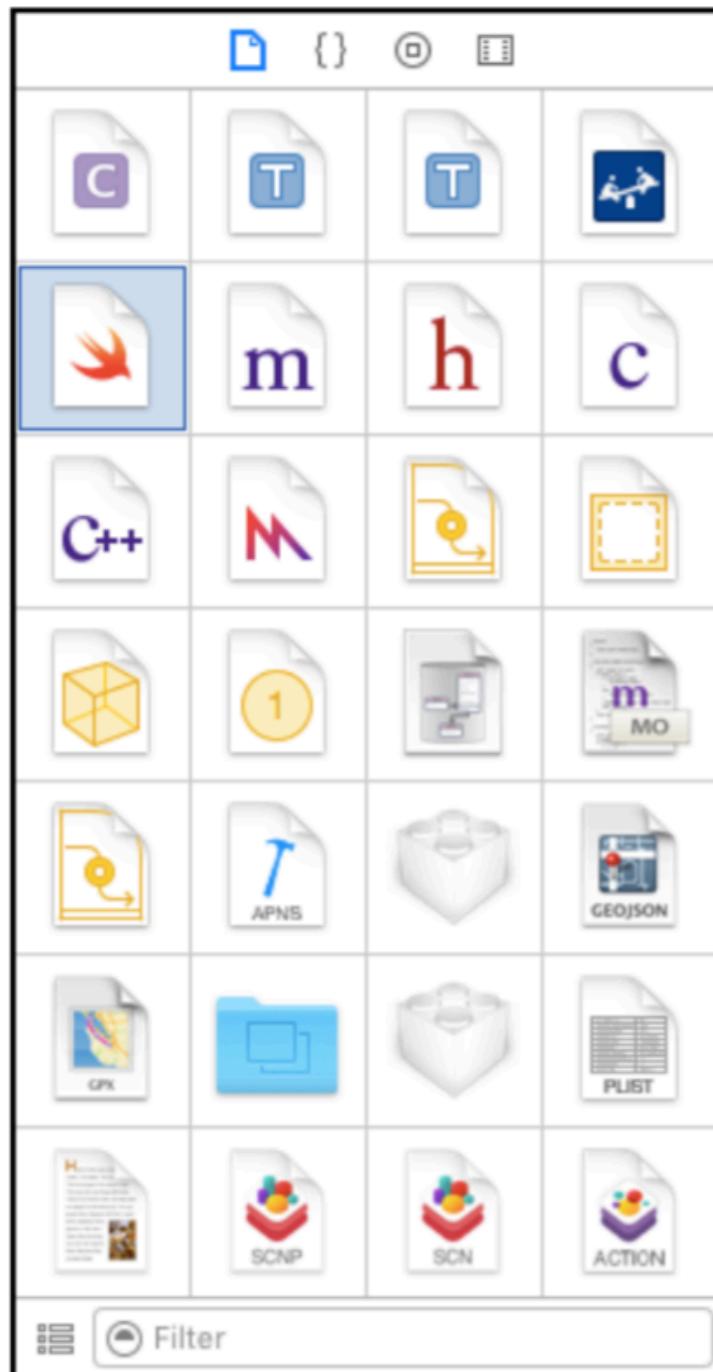
Quick Help Inspector

Utilities Area

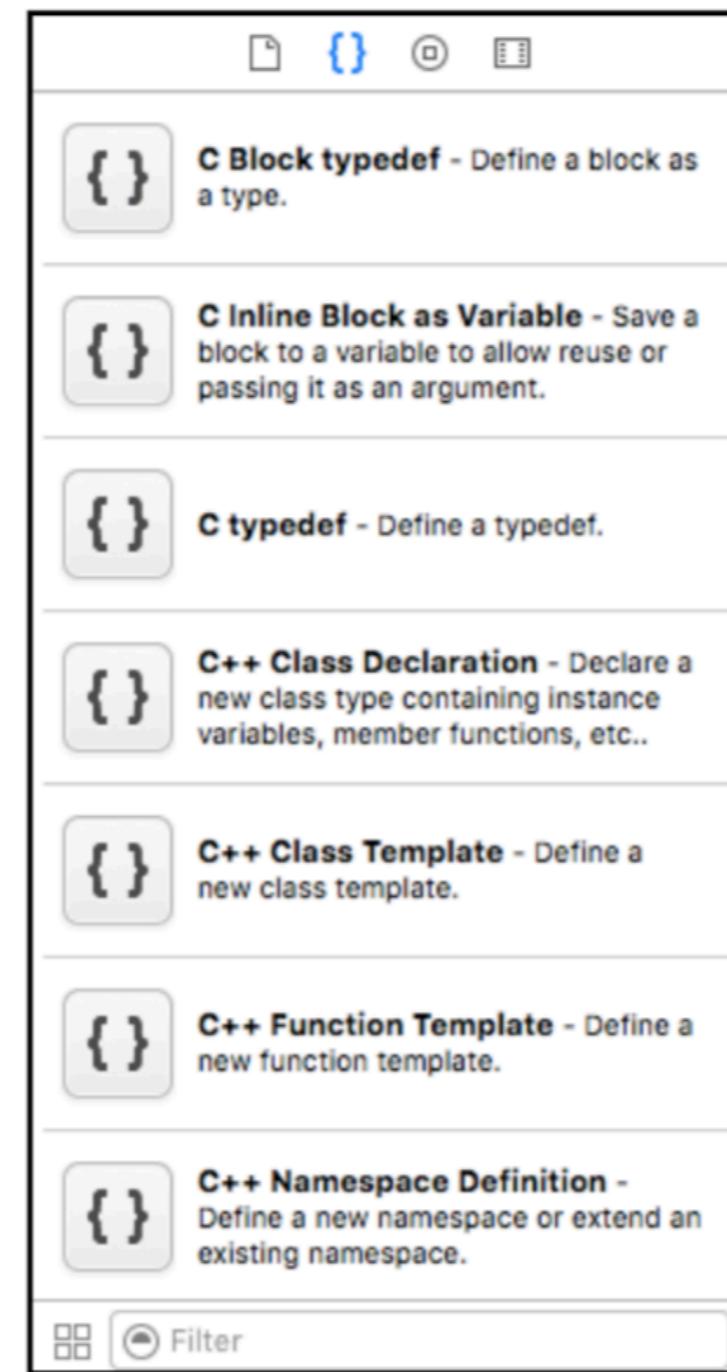


Utilities Area

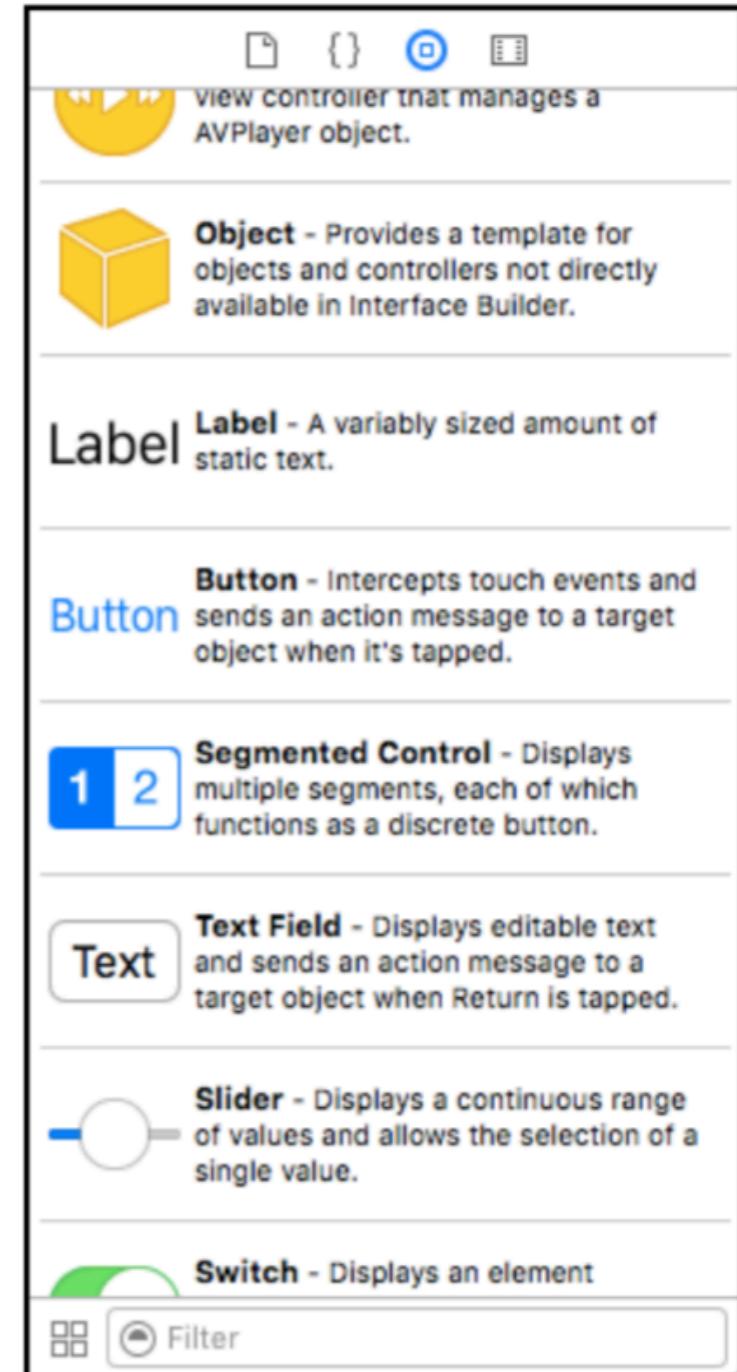
File Template



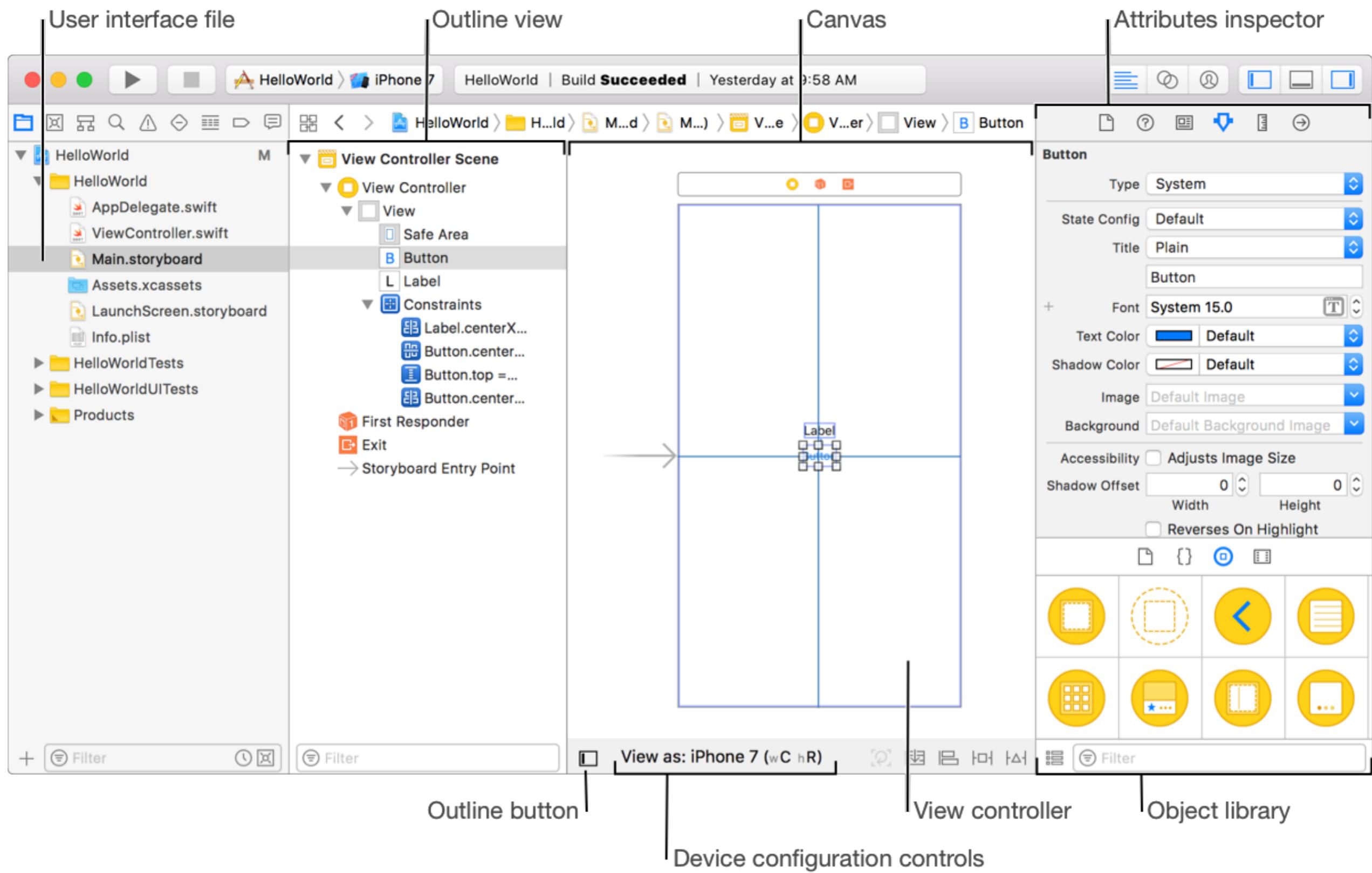
Code Snippet



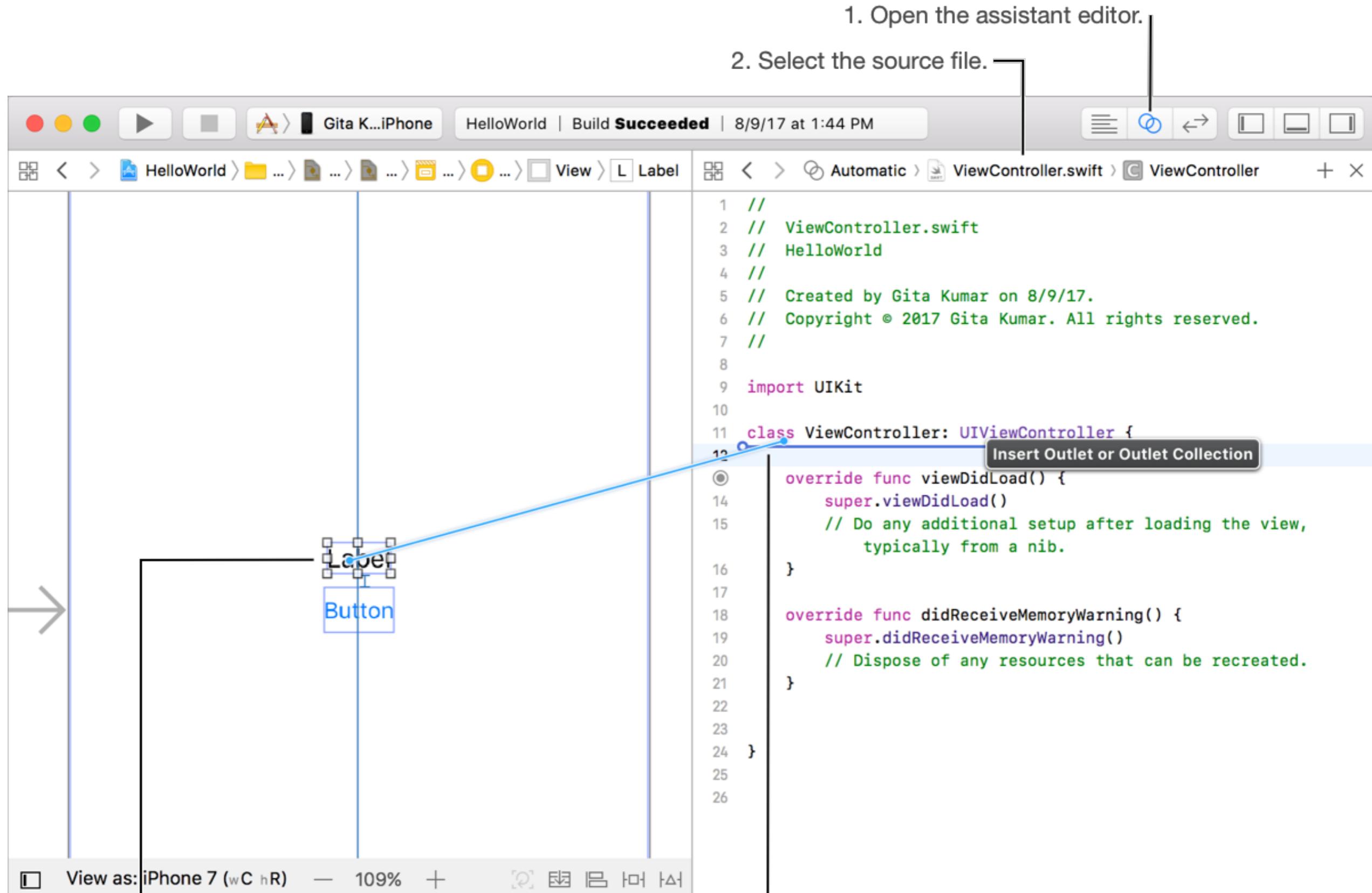
Object



Storyboard



Storyboard



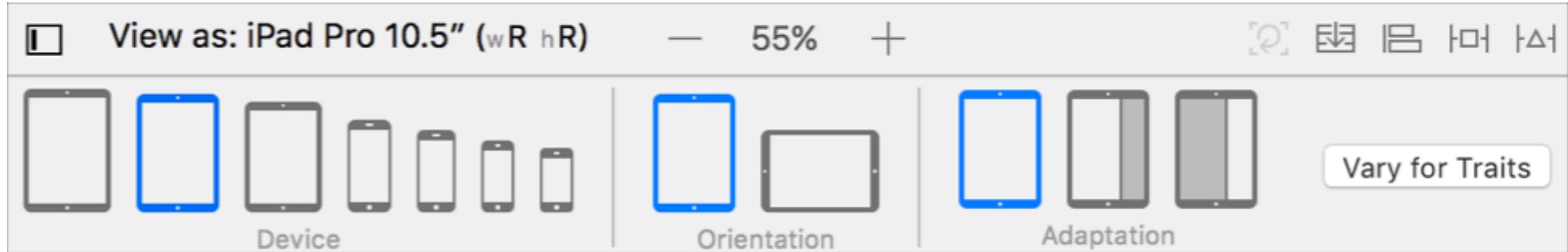
3. Control-drag from the object to the source file.

1. Open the assistant editor.

2. Select the source file.

4. Release the pointer in the source file.

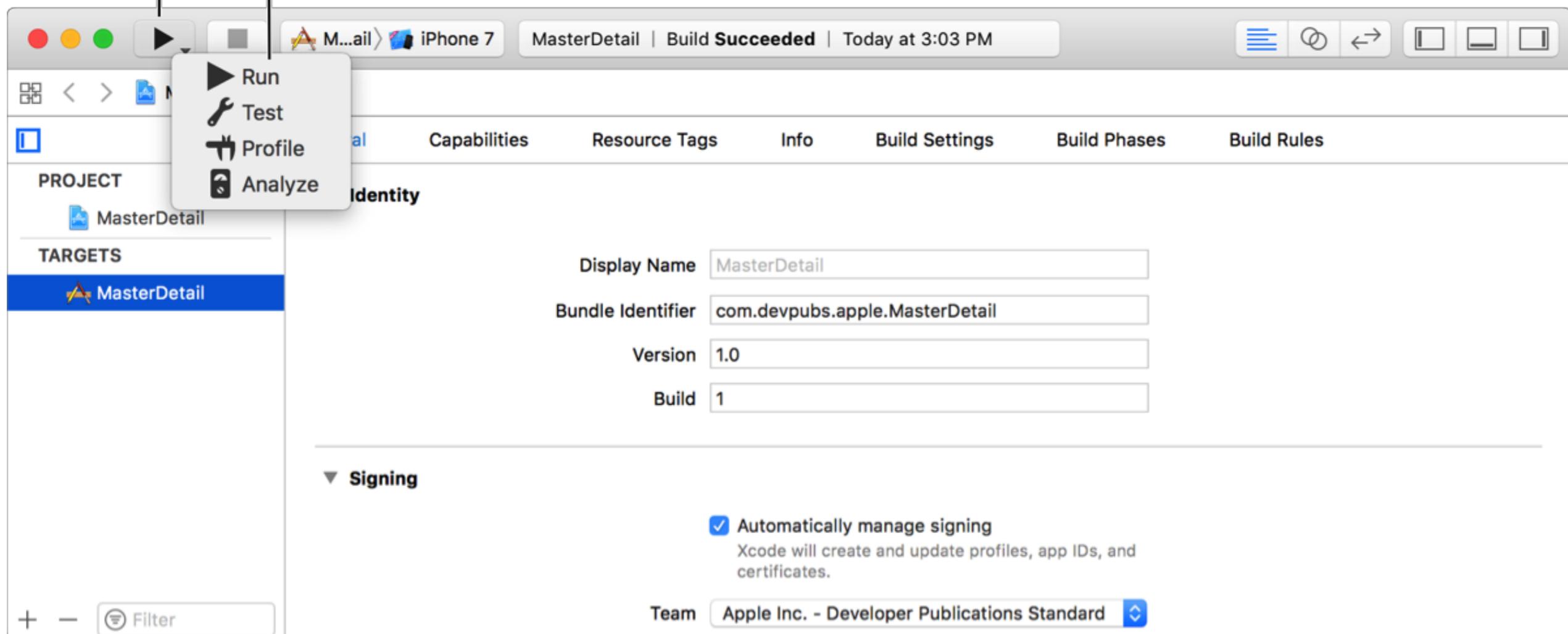
Storyboard



Run apps

Click and hold the run button.

Choose an action from the pop-up menu.



Simulator

