

**SUBMITTED BY:**

**NAME:** SHRAVYA RUMANDLA

**ROLL NUMBER:** A00120746

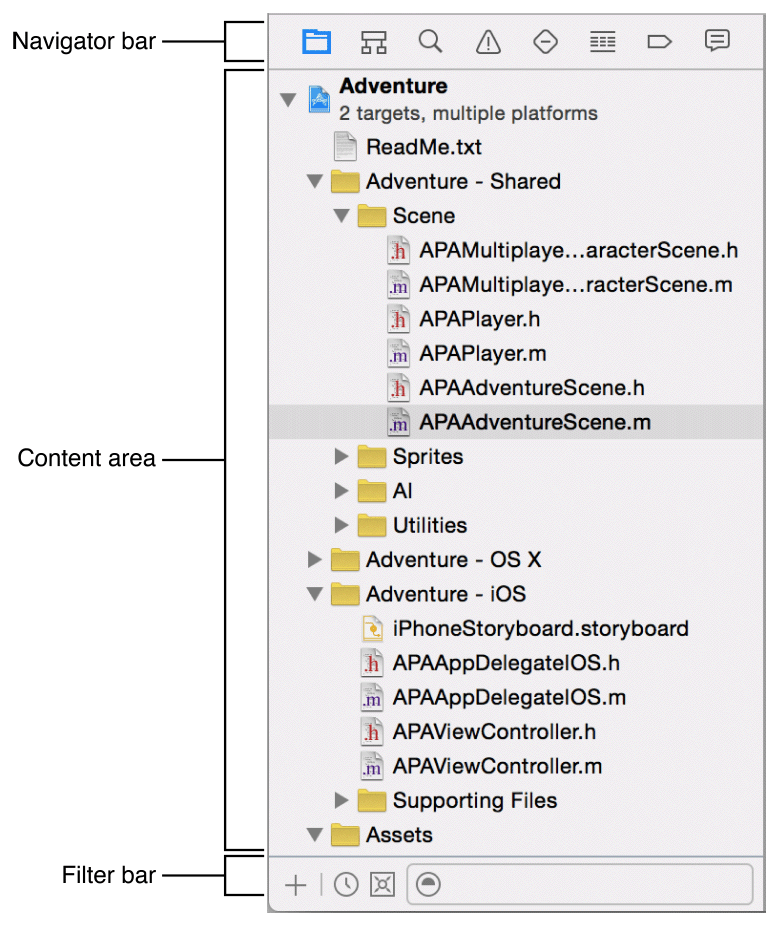
**INTRODUCTION**

Xcode is Apple's application framework (IDE), which you may use to create apps for the iPad, iPhone, Apple Watch, and Mac. Xcode has tools to help you manage the whole development process, from app creation to testing, optimization, and submission to the App Store. The Features of Xcode

* Single-Window Interface
* Assisted Source Code Editing
* Graphical UI Design
* Integrated Debugging
* Testing and Continuous Integrations
* Automatic Saves and Source Control Management
* Integrated Documentation
* App Distribution to Testers and the App Store

**Navigating Your Workspace**

From the navigation section, you may access files, symbols, unit tests, diagnostics, and other aspects of your project. You select the navigator most matched to your job from the navigation bar. Every navigator's content section provides access to important parts of your projects, and the filtering bar on every navigator enables you to limit the material that is shown.

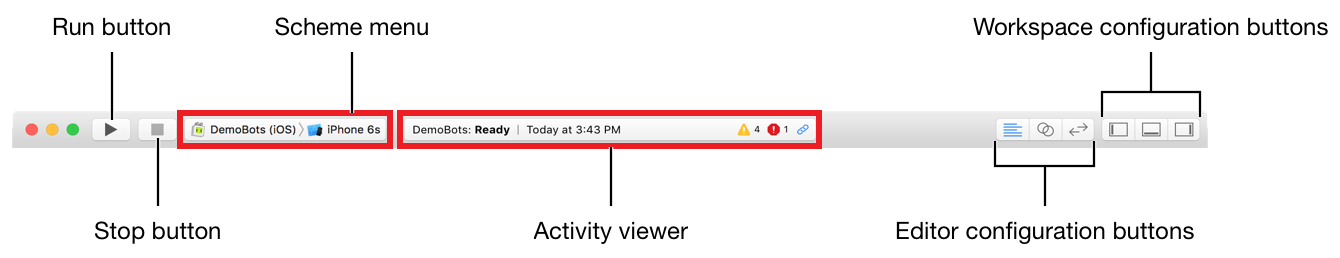


With in filter bar, type text. Just the entries in the content section that contain the search phrase are displayed in the text input box. Most navigators have icons on the left side of the filter bar that may be used to narrow down the material that is displayed. On the left side of certain filter bars is an Add button (+) that you may use to add a bit to the subject areas. Communicating with bots is done using the button on the left of the filter bar in the report browser. Control and Monitoring Bots from the Report Navigator goes through how to use bots from the report navigator in greater depth.

**Using the Workspace Toolbar**

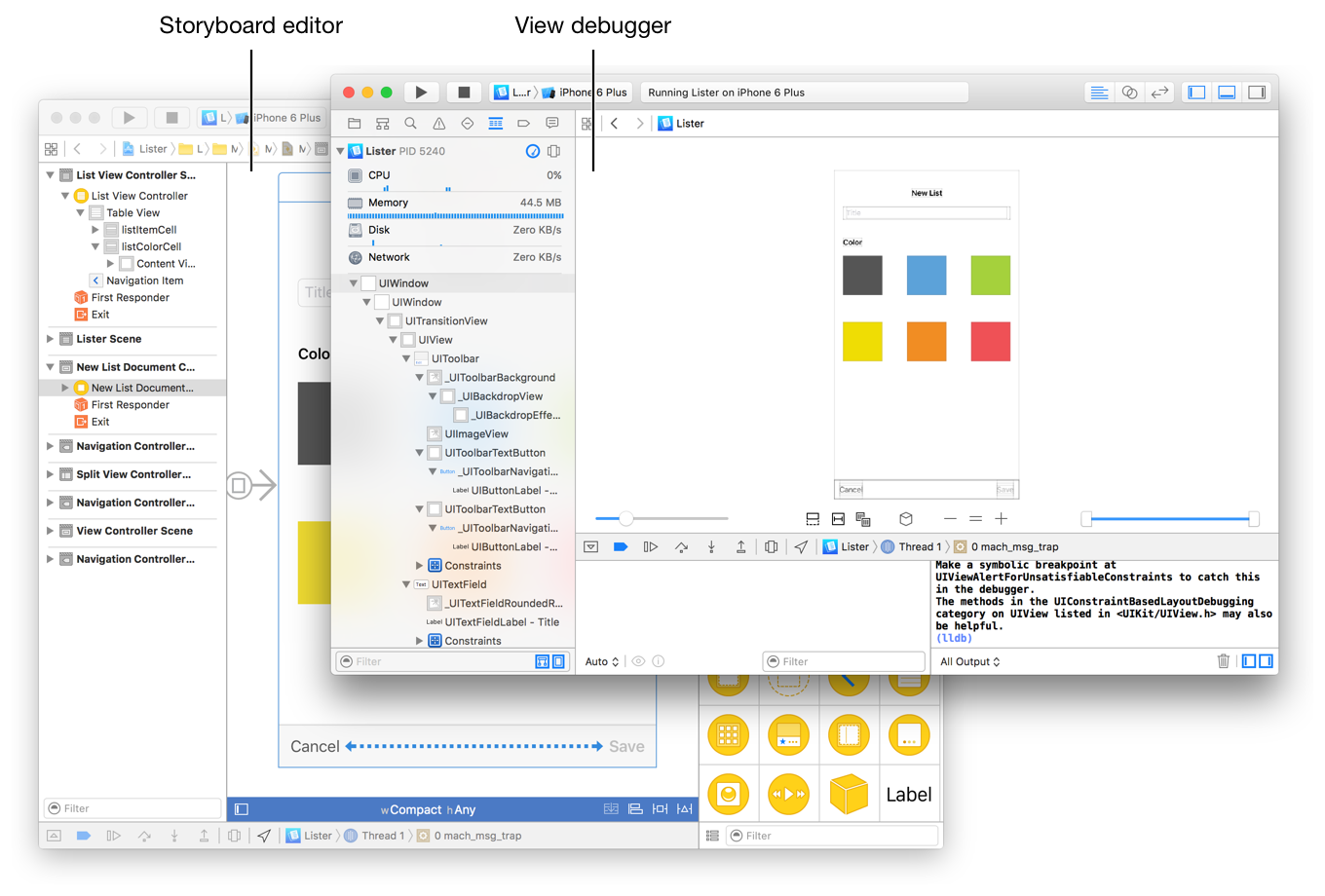
This workspace window's toolbar at the top gives immediate access to commonly used functions. Your products are built and run using the Run button. The Pause button brings your currently running code to a halt. You may specify the items you wish to produce and run using the Scheme menu. The activity viewer displays status messages, build progress, and other information to indicate the progress of presently running processes.

You may customise the editor area using the editor configuration buttons. See Configuring the Editor Area for additional details. As illustrated in the Workspace Window Overview, the workspace settings buttons conceal or reveal the optional navigation, debug, and services sections.



**Using Multiple Workspace Windows**

Choose File > New > Window to create several workspace windows. Each window may be personalised separately from the others



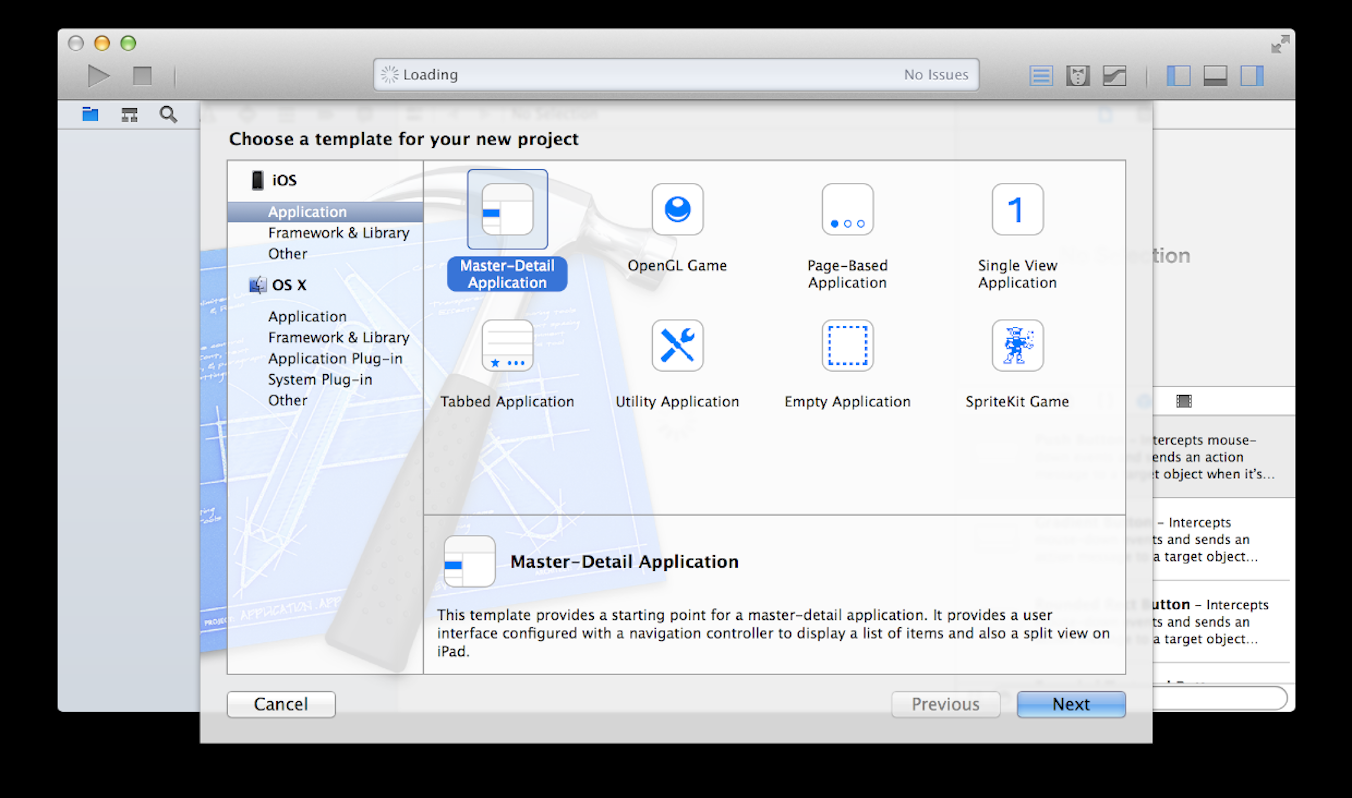
**The Use of Tabs**

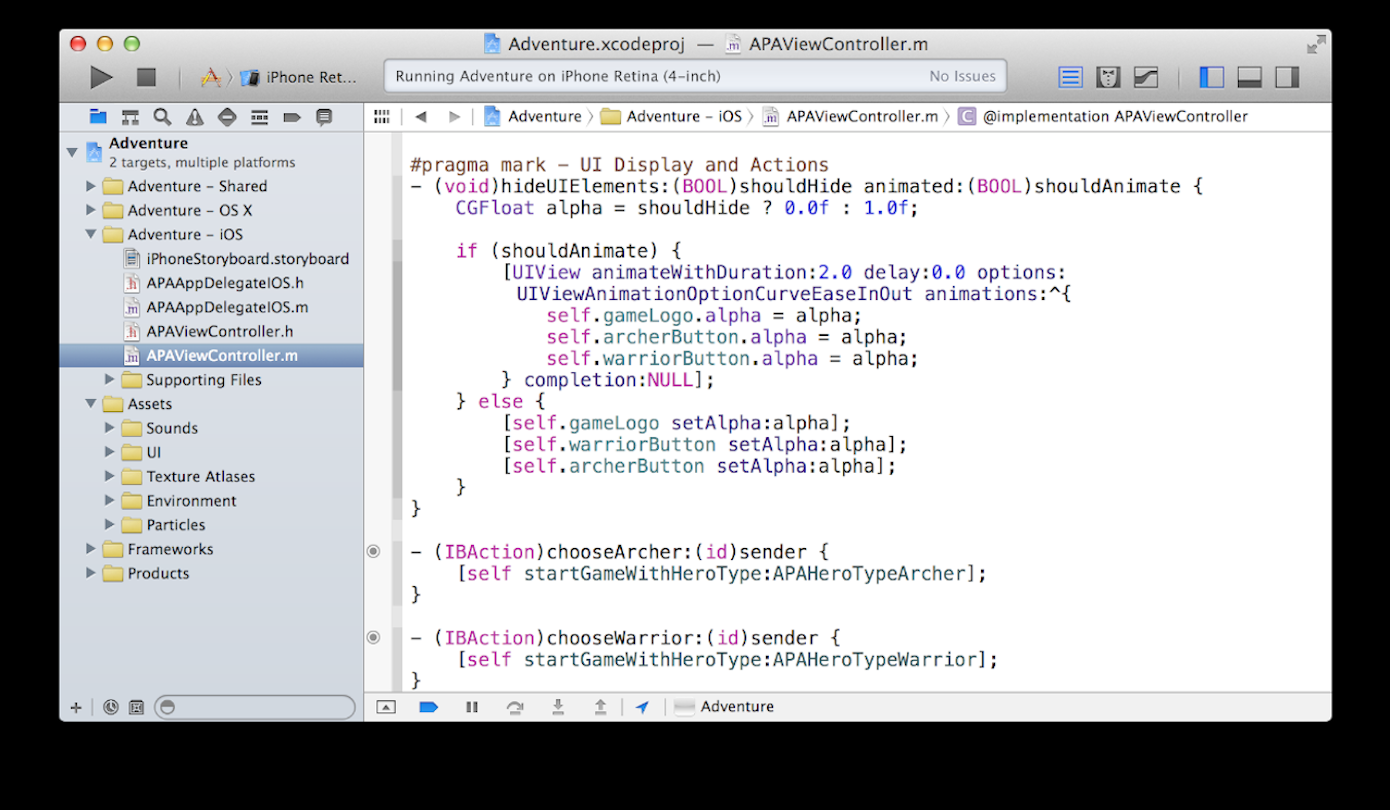
Use tabs to create various, workflow-specific workspace window layouts. Choose File > New > Tab or click the Add (+) button in the tab bar to add tabs to a workspace window. To make a tab the current workplace window style, click it.

Working with Projects

In Xcode, you must build a project to maintain all of the essential files and resources organised. Choose File > New > New Project to begin a project. Xcode announces a new workspace window and prompts you to select a project template. Xcode comes with built-in themes for creating popular iOS, watchOS, and OS X app designs.

In the project navigator, you can see the names of project files. The contents of a file are displayed in the relevant editor or client when you choose it in the project navigator. The Adventure project is seen in the screenshot above. The project browser selects an implementation file , and the contents of the file display in the code editor





**Using Alternative Toolchains**

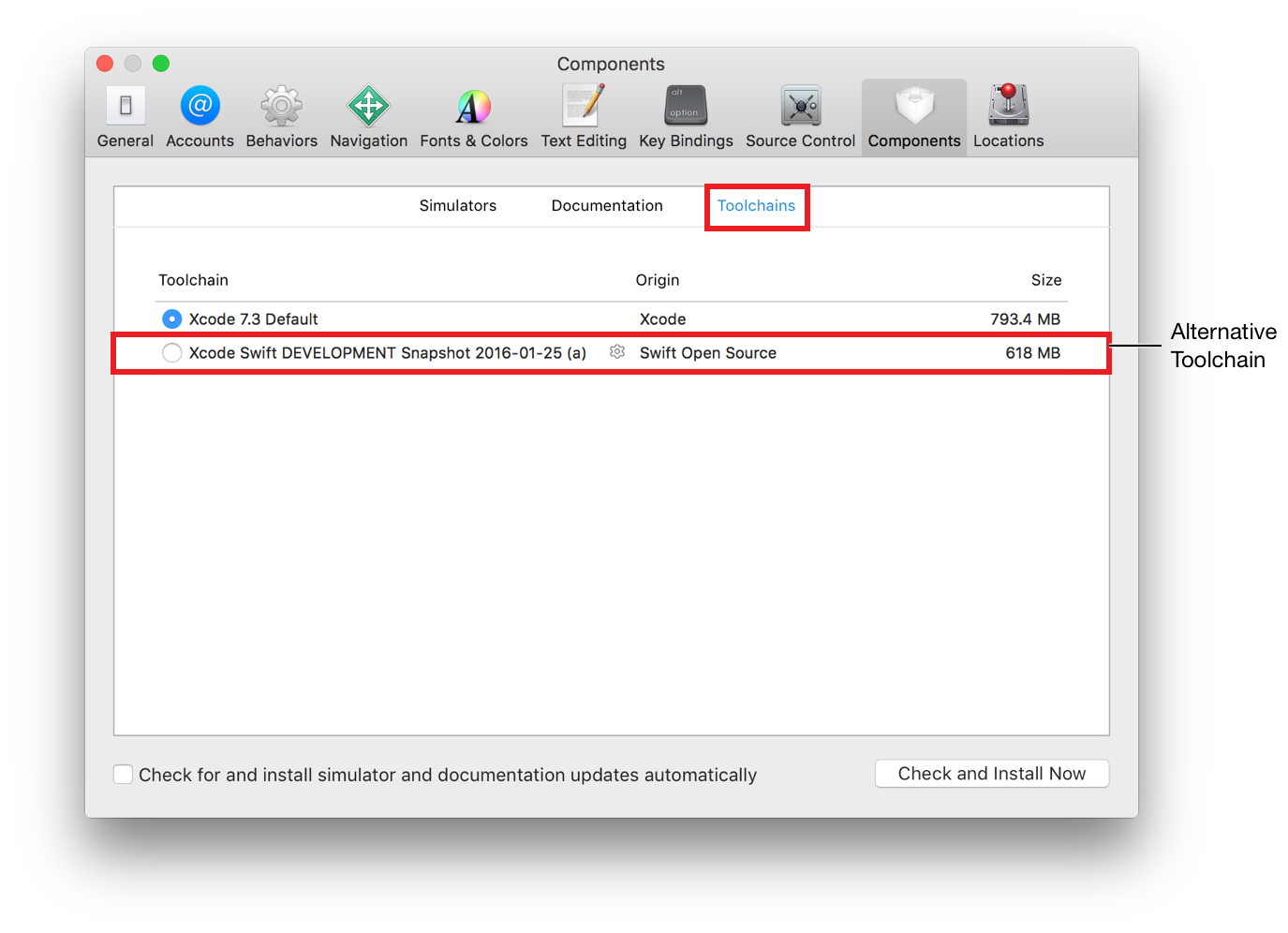
Alternative Xcode toolchains may be found either in upper library (/Library/Developer/Toolchains) or a client library (/Library/Developer/Toolchains). Downloading a toolchain in a personal library 's available solely to that user, but installing it in the top level library provides access to all customers on that machine that share accounts.

Viewing and Managing Installed Toolchains

Alternative toolchains show in the Contents preferences' Toolchains pane. Whenever alternative toolchains are downloaded, the Toolchains window becomes visible.

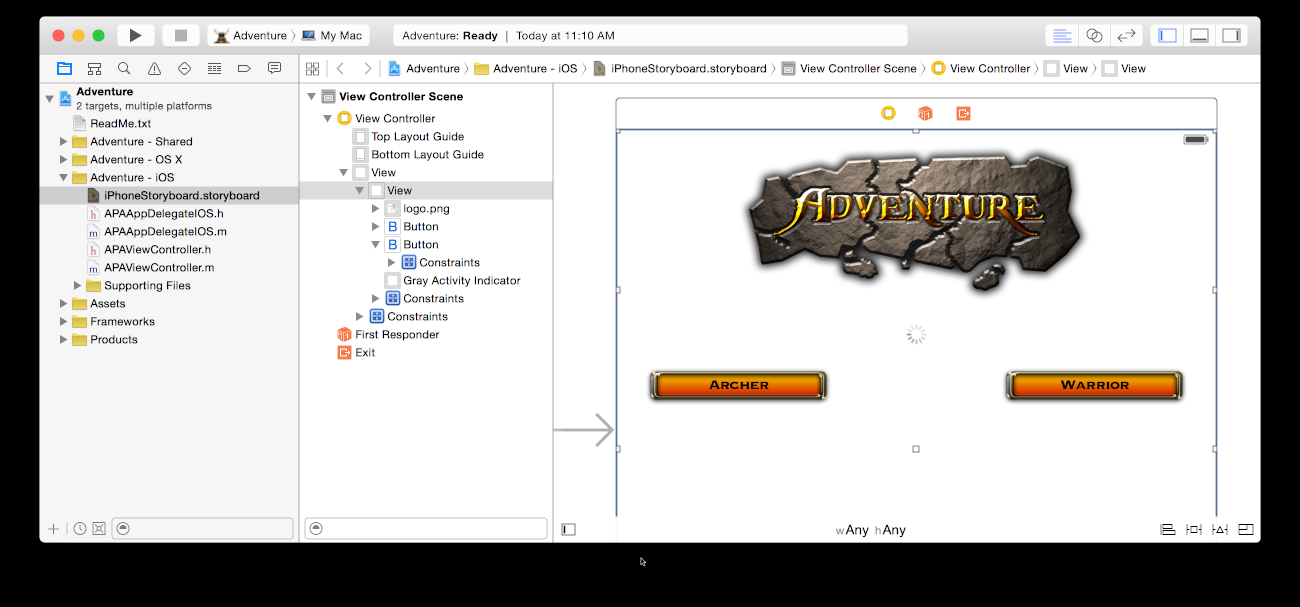
You may do the following in the Toolchains pane:

* View the alternative toolchains that have been downloaded.
* Differentiate between different toolchains.
* Verify a toolchain's code signature.
* In the Finder, identify the identity of a toolchain.
* Remove a toolchain that has been downloaded.

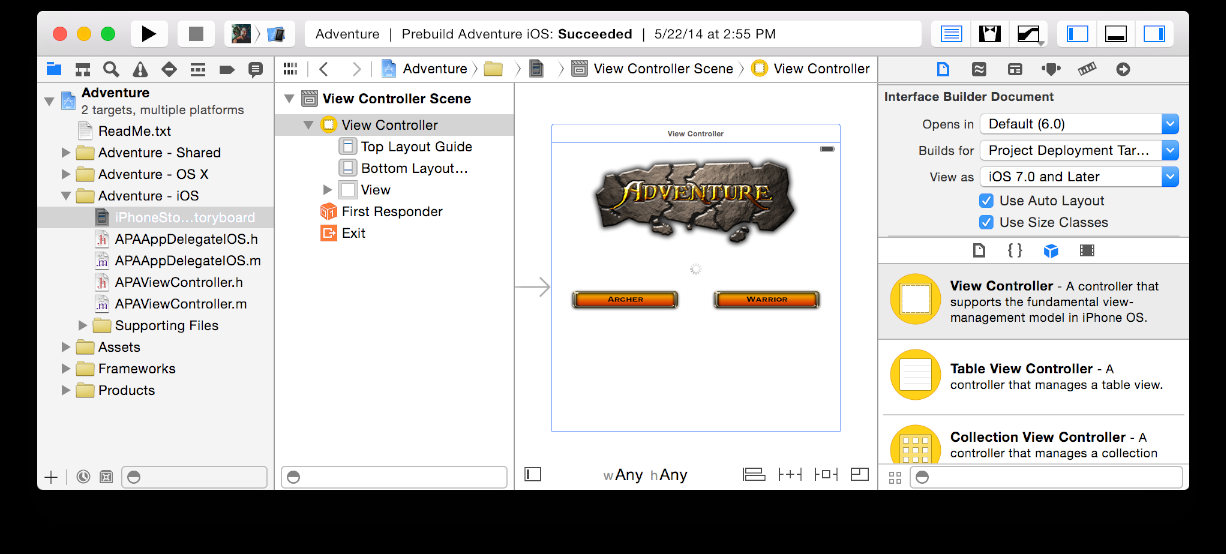


**Using Interface Builder**

In Interface Builder, you design the user interface for your programme. Choose an interface file in the project navigator, and the data of the document appear in Interface Builders in the workspace window's editing area. The filename suffix for a user interface file is .storyboard or.xib. A single view controller or menu bar is generally specified in a xib file. A storyboard defines a collection of view controllers as well as the transitions among them. A storyboard, unlike a xib, may include a large number of current programs as well as the transitioning between them. When you are using Xcode's built-in templates to build new projects, it includes basic interface files.



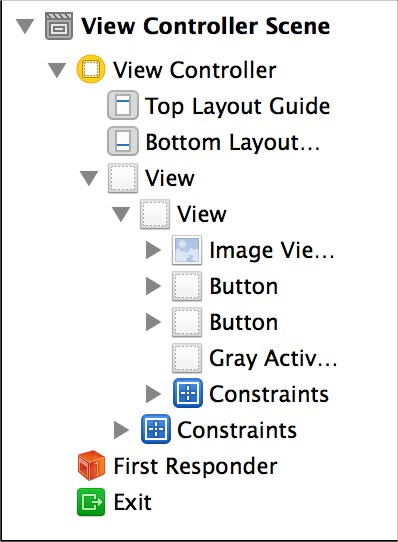
Drag items from the tools section into the Interface Builder canvas to add user interface components. You can then organise the elements, define their properties, and link these to the code in your source files. In the helper editor, you may write the code that implements the functionality of your app's user interface elements as you design them in Interface Builder.



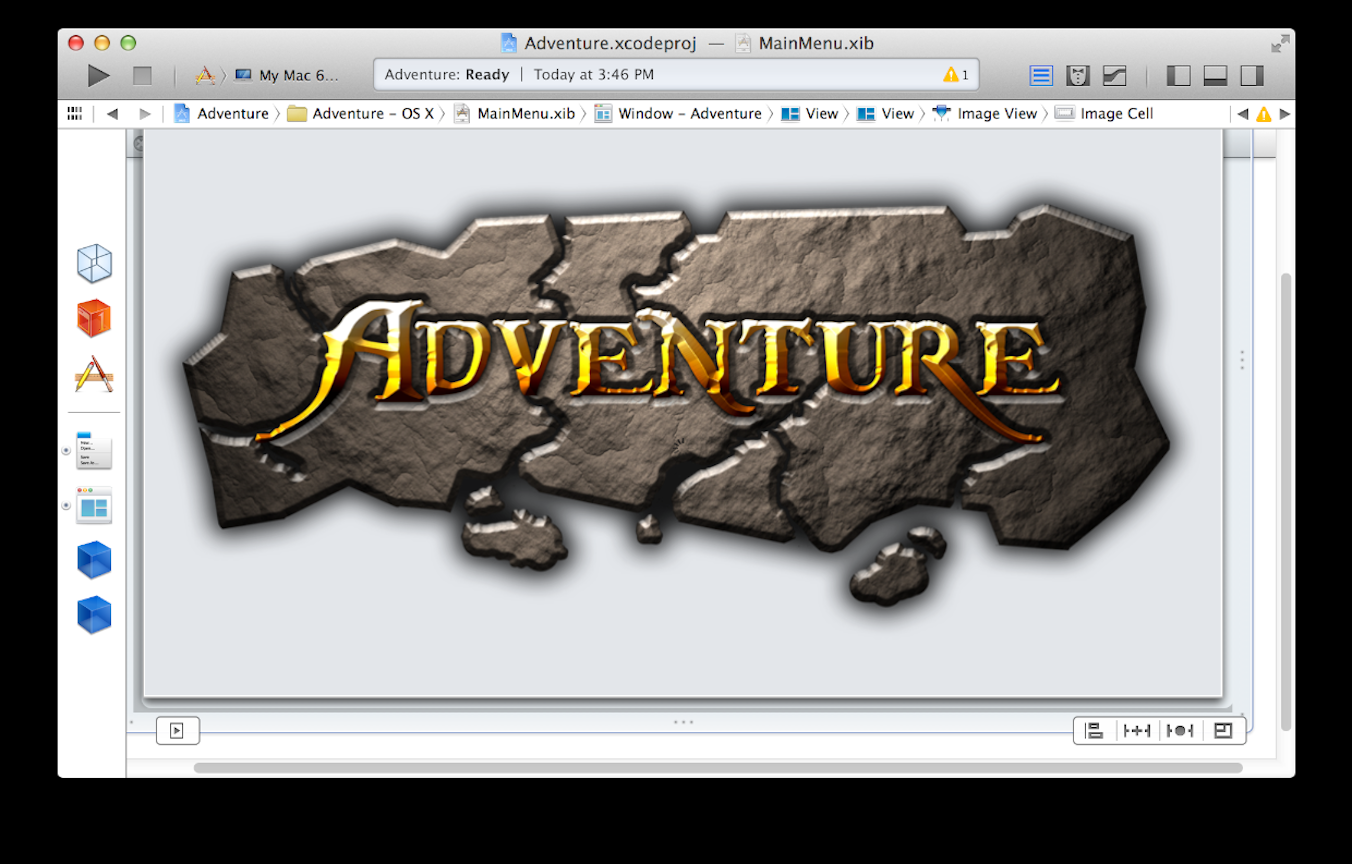
**Parts of Interface Builder**

The dock (on the left) and the canvas (on the right) are the two main sections of Interface Builder (on the right). The dock displays a list of the items in the ui file. The canvas is just where you arrange these items in the user interface of your programme.





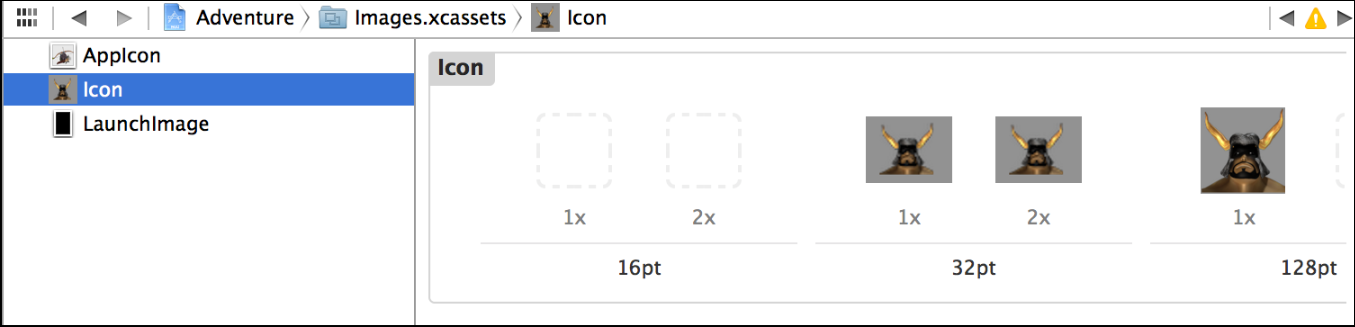
By selecting the Hide and Show Documents Outline control on the bottom left of the Interface Builder canvas, you may display the rising entities in an icon perspective instead of the outline view for xib files.



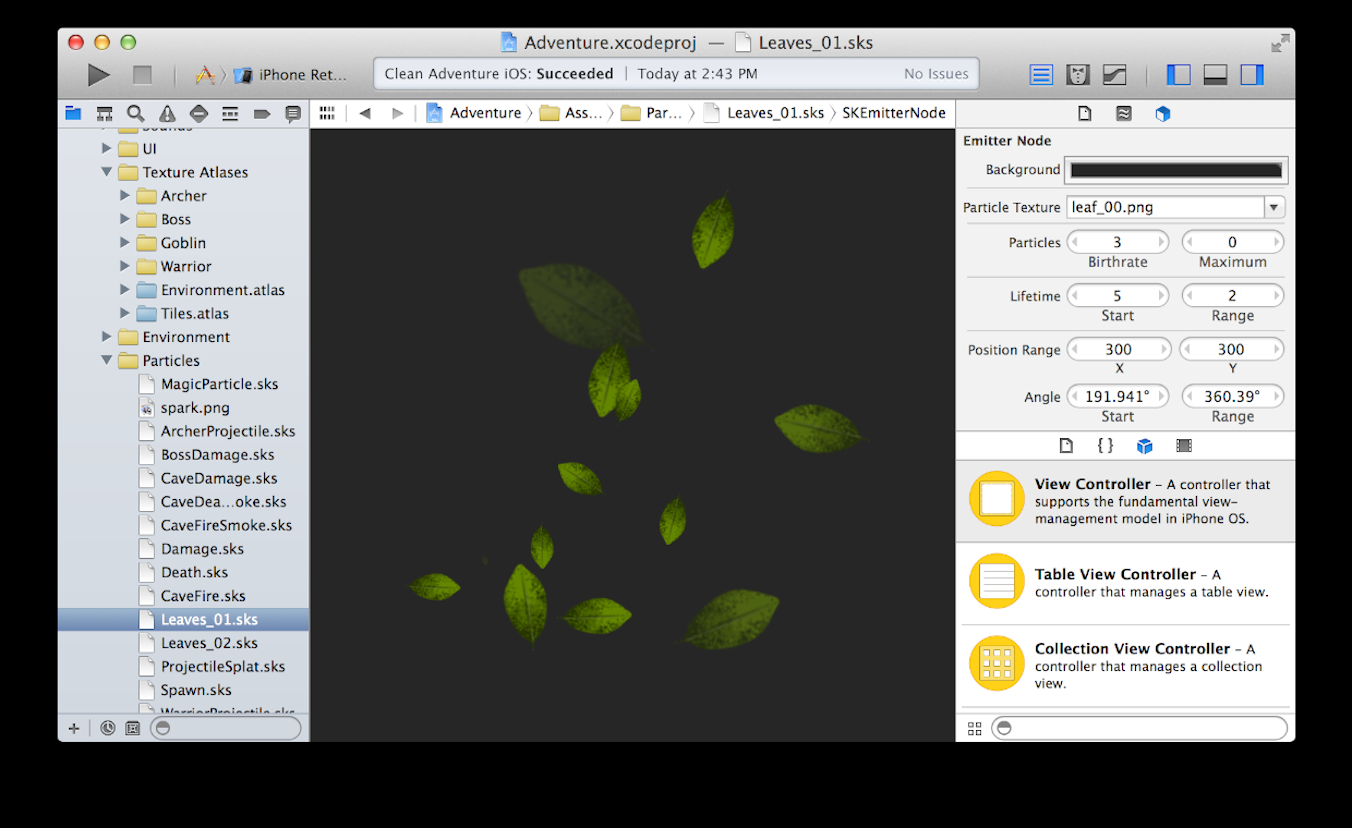
**Adding Images**

In addition to Interface Builder, Xcode provides a number of tools to help you design and manage user interface elements for your project.

You design a variety of graphics for your app, such as icons, unique art, and start backgrounds for various iOS devices. A few of these pictures are necessary for submission to the App Store. The asset catalogue aids in their management.



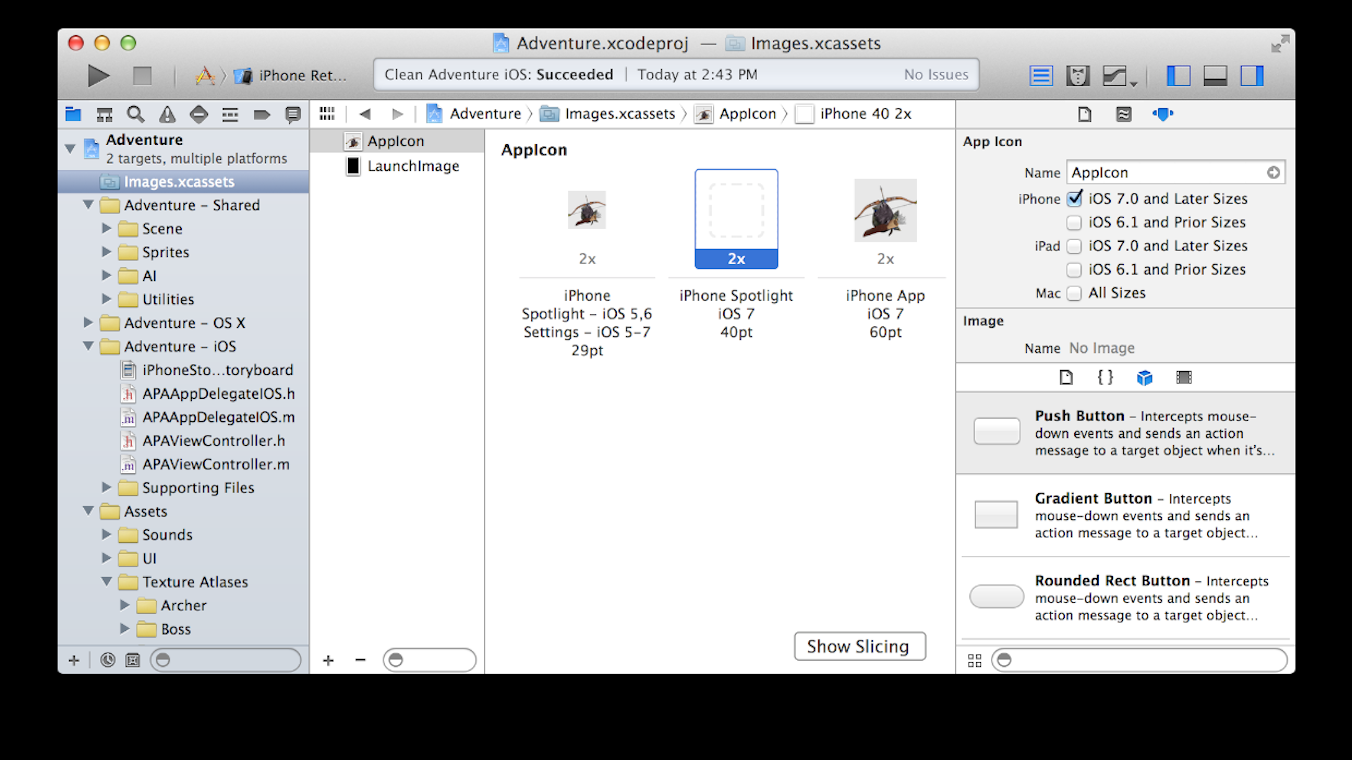
You may improve your app by introducing motion effects including particles moving like snow, sparks, and smoke using the particle emitter editor. These techniques are very handy in iOS and Mac games.



**Work with Image Assets in the Asset Catalog**

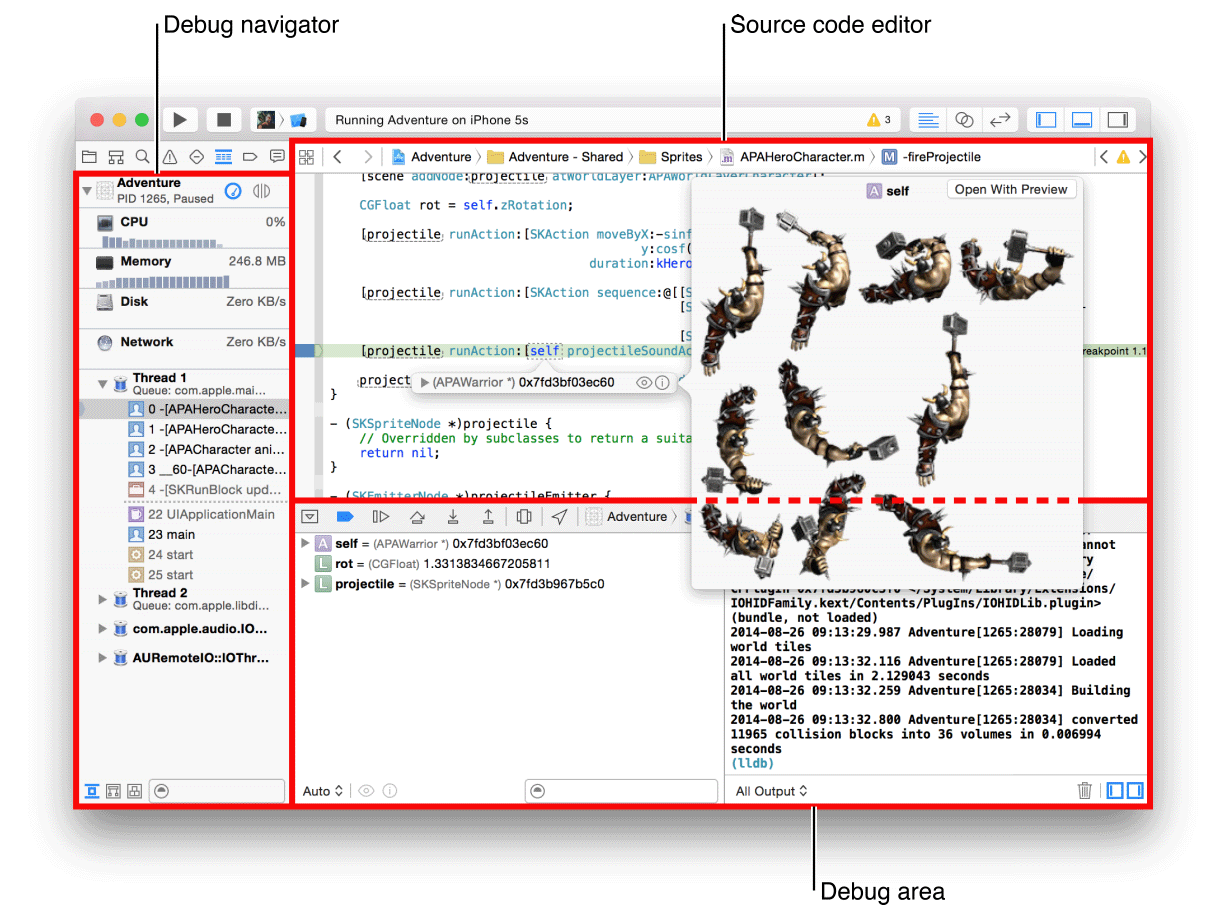
Xcode produces an asset collection named Assets.xcassets when you start a new project. Xcode opens the asset catalogue in the editing area when you choose it from the project navigator.

A list of picture sets may be found in the asset catalogue. Every image collection, such as AppIcon in the example, has all of the picture variants required to support different devices and size factors.

  
Image feature sets, such as those for buttons and other elements in your programme, can be created. Click the Add button (+) at the bottom of the picture set list to create a new image set or integrate photos into an existing one. To arrange the products in the catalogue, you may upload files. Sprite Maps, Watch Details, and Data files are all supported by asset catalogues.

**Using the Debugger**

Xcode executes your project and begins a troubleshooting session after you select the Enter key in the workspace toolbar and your app compiles properly. You may use tools like data hints and Fast Check for the values of variables to troubleshoot the app right in the code editor. You can investigate the current status of your running app and manage its operation using the debug area and debug navigator.



You must limit your app's effect on our customers' systems in order to produce a high app. To get insight into your app's resource usage, utilise the debug gauges in the debug navigator, and then when you find an issue, use Instruments to monitor and evaluate your app's efficiency. Use the energy guidelines to keep your battery capacity as long as possible.

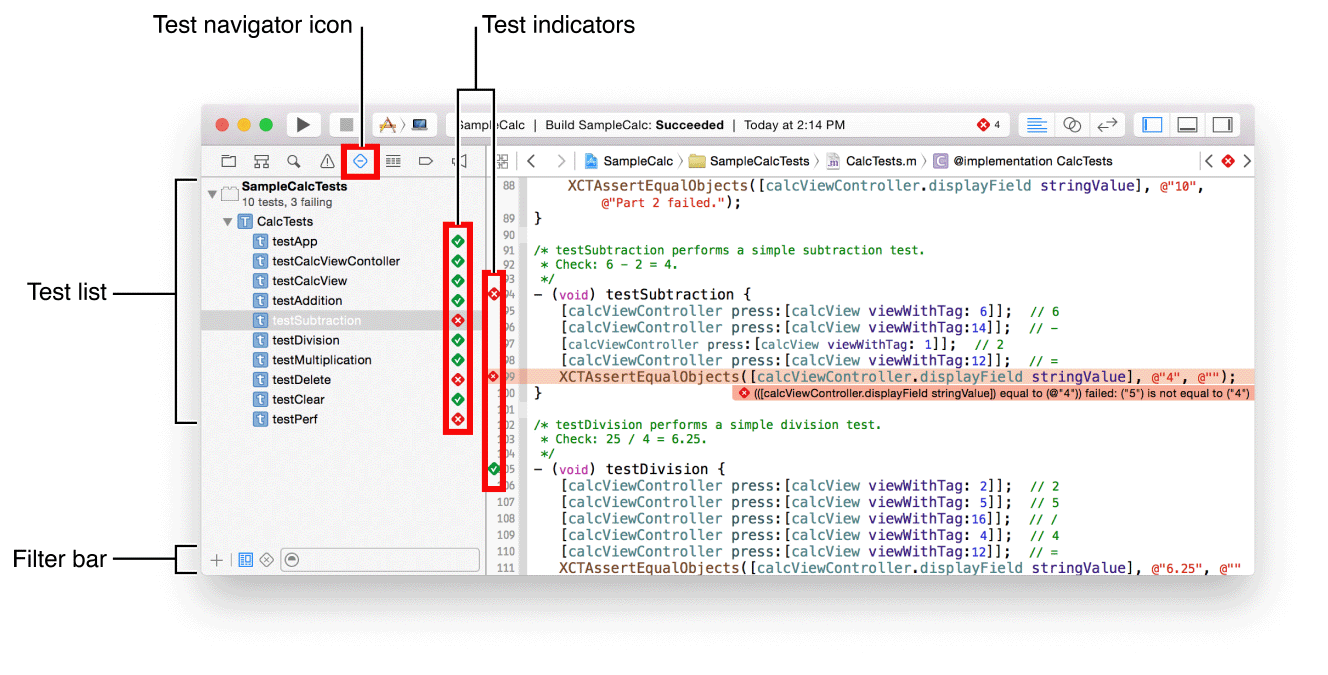
Use Simulation to identify big issues during design and early testing if you're creating an iOS or watchOS app. You may set up Xcode to assist you in concentrating on your debugging chores. Whenever your code reaches a halt, for instance, you may have Xcode play an alert sound and display the debug area, the debug navigator, and your code at the transition point in a windows tab called Debug.

**Using Unit Tests**

Three different forms of testing are supported by Xcode. The focus of functional testing is on the code's functionality. The aim of performance testing is to determine how long it takes to complete a task. The transmission of data through the user interface is the subject of ui testing. You write functions for functional and performance testing. Each method creates a testing environment, runs the targeted sections of the programme, then deconstructs the test environment.

Unit testing is by far the most frequent kind of functional testing. A unit of code is the lowest tested component of a project, such as a method in a class or a group of methods that perform the same task. Unit tests are frequently used to discover regression analysis induced by project code modifications. Some programmers start by writing unit tests, then create functions that pass the tests.

Select Product > Test to run all tests. To see the progress and results of the survey, click the Test Navigator button. By selecting the Add button (+) in the bottom-left corner of the test navigator, you may add a test target to a project (or a class to a test). Select a test from the test list to see the code for that test. The original code editor opens when you double-click the document.



Click to the right of a test suite's name to execute it. Select Product > Perform Action > Execute Test Methods from the test navigator to run a selection of test methods. Click the down arrow to the right of the method name to execute a single test method. To execute all checks in the current scheme, go to Product > Test.

A green diamond with a check signifying completion displays to the right of the test title when it passes. When a failure occurs, a red diamond with an X next to the test name indicates failure, and the problem is shown in the issue navigator. Select the Problem Navigator icon in the navigator bar to display the issue

Select the Failed Test button at the bottom of the test browser to see just the failed tests. To inspect a failed function in the source code editor, click it. To rerun the test, click the failed test indication (a red diamond with an X) once you've addressed the cause for the failure.

**Using File Saving**

As you work, Xcode saves changes to source, project, and workspace files automatically. Because Xcode records and stores your changes in memory, this functionality does not require any setup. When you do the following, Xcode saves the modifications to disc:

* Create and maintain your app.
* Files should be committed to a source code repository.
* Finish the project.
* Xcode should be closed.

By selecting File > Save, you may save changes to disc manually.

You may revert files and whole projects to a prior state in Xcode, as well as delete any modifications you've made. To keep track of changes at a finer level, you utilise source control management.

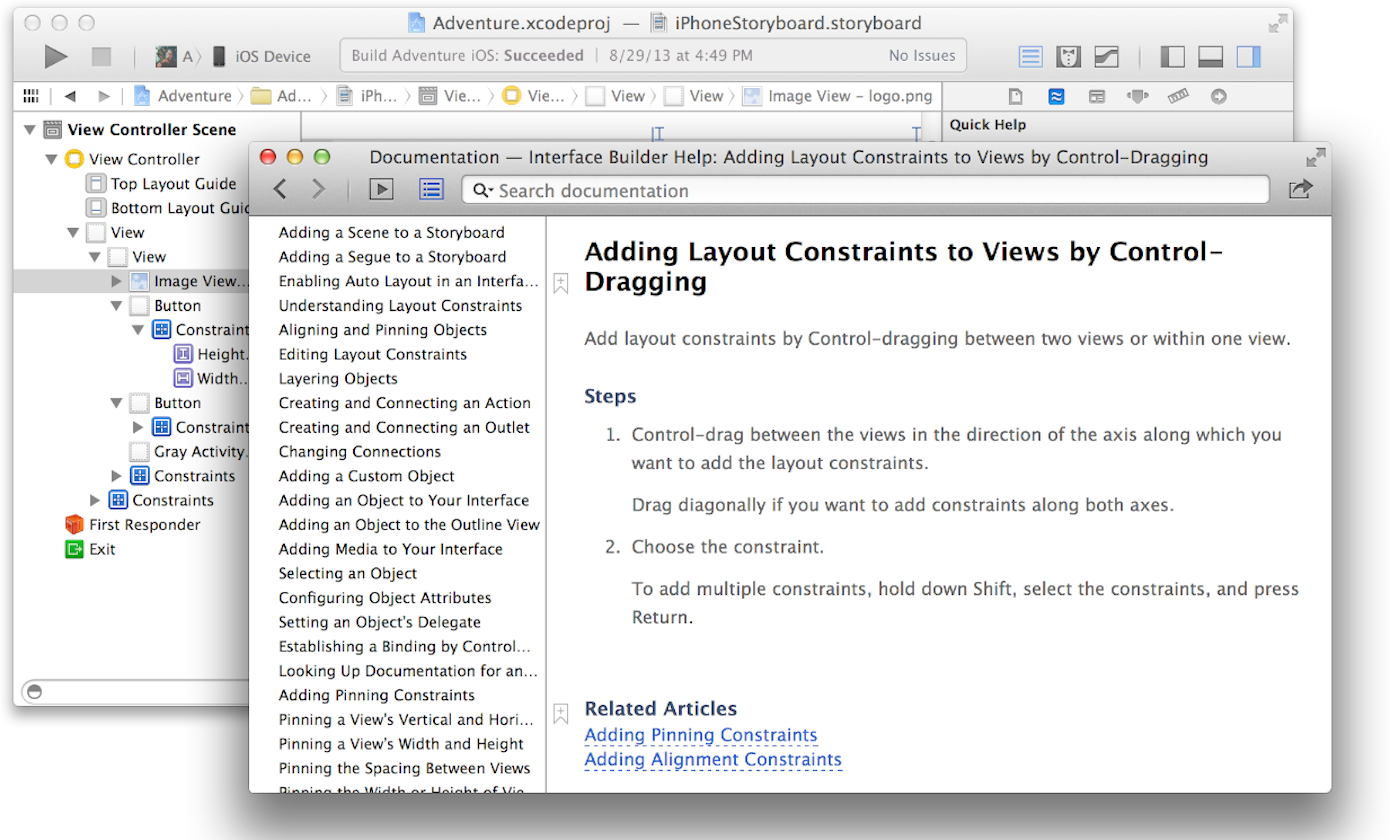
**Reverting to the Last Saved Version of a File**

Choose File > Revert to Saved to undo all modifications you've done to a file since it was last committed to disc. Only the file with the editing focus is affected by the Revert to Saved command. By clicking a file's editing pane or choosing it in the project navigator, you may give it editing attention. For instance, suppose you try out a new user interface design and then choose to go back to the old one.

The Revert to Stored function always restores the file's content to their previous saved state on disc. Use the Undo option in the Menu settings to undo adjustments one at a time if you want.

**Finding Step-by-Step Instructions**

In Xcode, you may get step-by-step tutorials for doing typical tasks. To display a quick list of the most popular operations, control-click sections of the Xcode user interface. To see a wider selection, select Show All Help Topics. When you choose an action from the list, the Xcode documentation viewer screen contains a help article.



**The Help Menu**

Fast access to help articles is also available via the Help menu. When you type a searching word or phrase into the menu, a selection of relevant help articles and linked documents displays.

