

Практична робота 14. Проектування мобільних додатків для iOS. Мова Swift

Теоретичні відомості

Swift — багатопарадигмова компільована мова програмування, розроблена компанією Apple для того, щоб співіснувати з Objective C і бути стійкішою до помилкового коду. Swift була представлена на конференції розробників WWDC 2014. Мова побудована з LLVM компілятором, включеного у Xcode 6 beta. Безкоштовний посібник мови програмування Swift доступний для завантаження у магазині iBooks.

Компілятор Swift побудований з використанням технологій вільного проєкту LLVM. Swift успадковує найкращі елементи мов C і Objective-C, тому синтаксис звичний для знайомих з ними розробників, але водночас відрізняється використанням засобів автоматичного розподілу пам'яті і контролю переповнення змінних і масивів, що значно збільшує надійність і безпеку коду.

При цьому Swift-програми компілюються у машинний код, що дозволяє забезпечити високу швидкодію. За заявою Apple, код Swift виконується в 1.3 рази швидше коду на Objective-C.

Отже, знадобиться комп'ютер Apple і IDE Xcode, яке є в магазині App Store у вільному доступі (безкоштовно).

Програмне середовище розробки Xcode – це сервіс, який включає: редактор коду, редактор графічних інтерфейсів, систему контролю версій і інструменти для налагодження і викладення, написаних нами програм. Якщо коротко, це програма, яка дозволяє вирішити 90% всіх завдань у розробці програм під сімейство операційних систем від Apple.

Отже, запустивши Xcode, ми бачимо вітальне вікно, де є можливість для відкриття, створення або клонування проєкту. Нам потрібно створити новий, натиснувши “Create a new Xcode project”.

x



Welcome to Xcode

Version 9.2 (9C40b)



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.

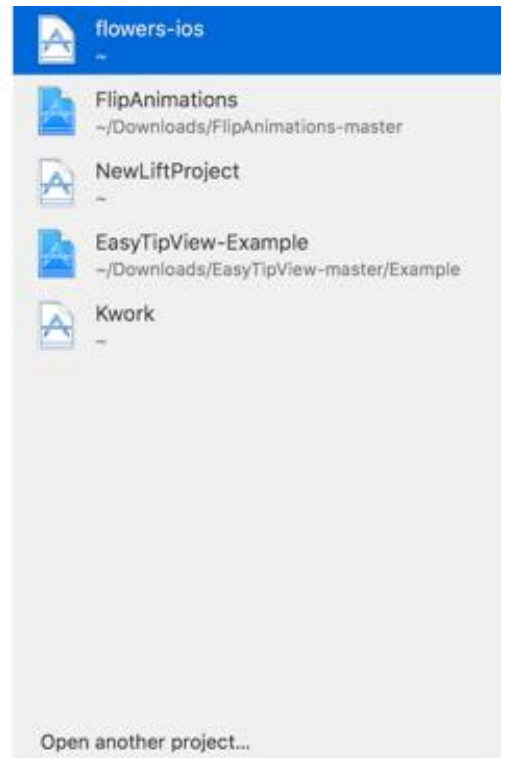


Clone an existing project

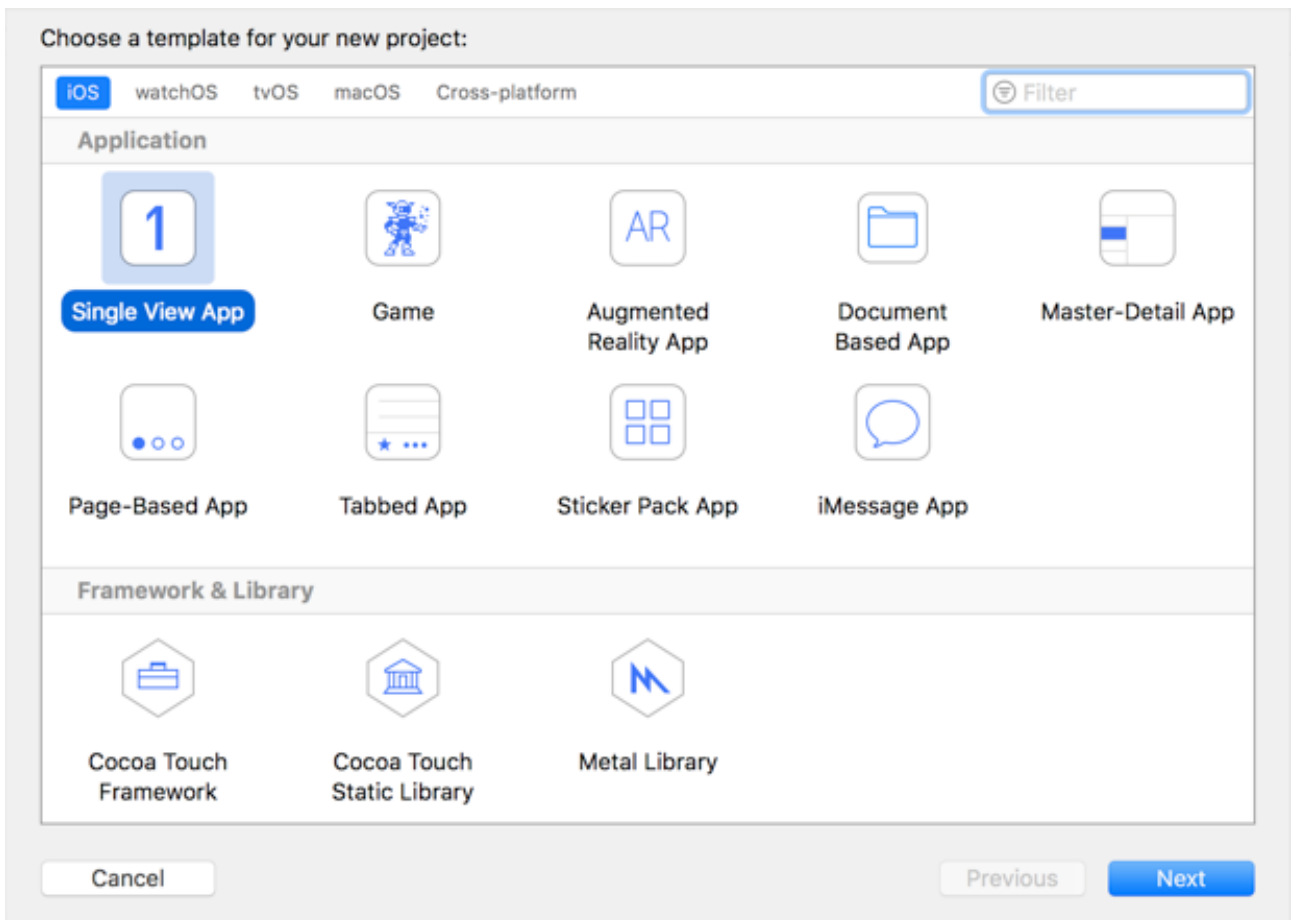
Start working on something from an SCM repository.



Show this window when Xcode launches

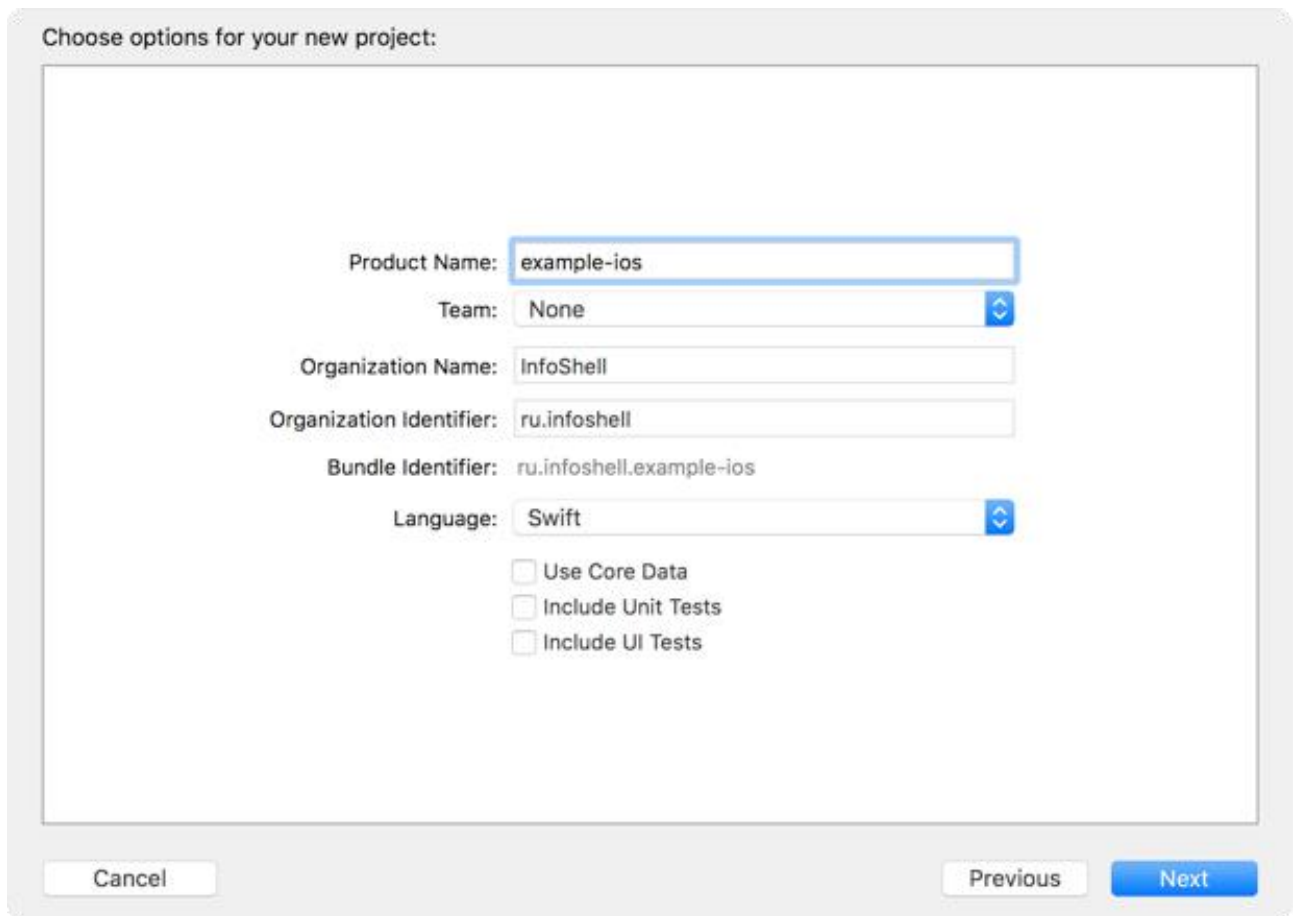


Наступним кроком ми бачимо це вікно:



Тут нам пропонують вибрати платформу та тип програми. Нам потрібно програмне забезпечення для створення iOS програми (у верхній частині діалогового вікна) та Single View App. Все починається із Single View, а потім розростається до 50 View App.

Наступним кроком нам потрібно ввести назву проекту, організацію, в якій ми хочемо зробити програму iOS, вибрати мову та увімкнути/вимкнути додаткові функції.



Choose options for your new project:

Product Name: example-ios

Team: None

Organization Name: InfoShell

Organization Identifier: ru.infoshell

Bundle Identifier: ru.infoshell.example-ios

Language: Swift

Use Core Data

Include Unit Tests

Include UI Tests

Cancel Previous Next

Product Name – ім'я програми;

Team – команда розробки. Встановимо None. Це поле буде потрібно на етапі викладки та тестування, щоб сертифікувати програму обліком розробника iOS;

Organization Name – назва організації якої належать права на вихідний код програми. Ім'я організації та деякі інші параметри будуть виділені заголовком у кожному файлі;

Organization Identifier – ідентифікатор організації. Як правило, це зворотний домен офіційного сайту. Якщо компанія називається example та адреса в інтернеті у неї example.ru, то її ідентифікатор дорівнюватиме ru.example;

Bundle Identifier – ідентифікатор пакета. Унікальний ID, який використовуватиметься при викладанні продукту в App Store і буде фігурувати під час підпису програми;

Language – мова програмування. Є дві опції: Objective-C та мова Swift;

Use Core Data — параметр, який додасть до нашого проекту бази даних. Вона зазвичай використовується для великих обсягів даних. В рамках нашої сьогоденної роботи ми не маємо на увазі, що взагалі зберігатимемо будь-які дані, тому відключаємо цей параметр;

Include Unit Tests (юніт-тести) – автоматизоване тестування коду. Його виконання допомагає побачити помилки, які можуть виникнути під час змін останньої версії готового digital-продукту. Але оскільки коду у нас у принципі не буде, необхідно відключити цей параметр;

Include UI Tests - автоматизоване тестування інтерфейсу користувача. Відключаємо цей параметр, так як у нас буде лише один елемент інтерфейсу для користувача, і його неможливо протестувати;

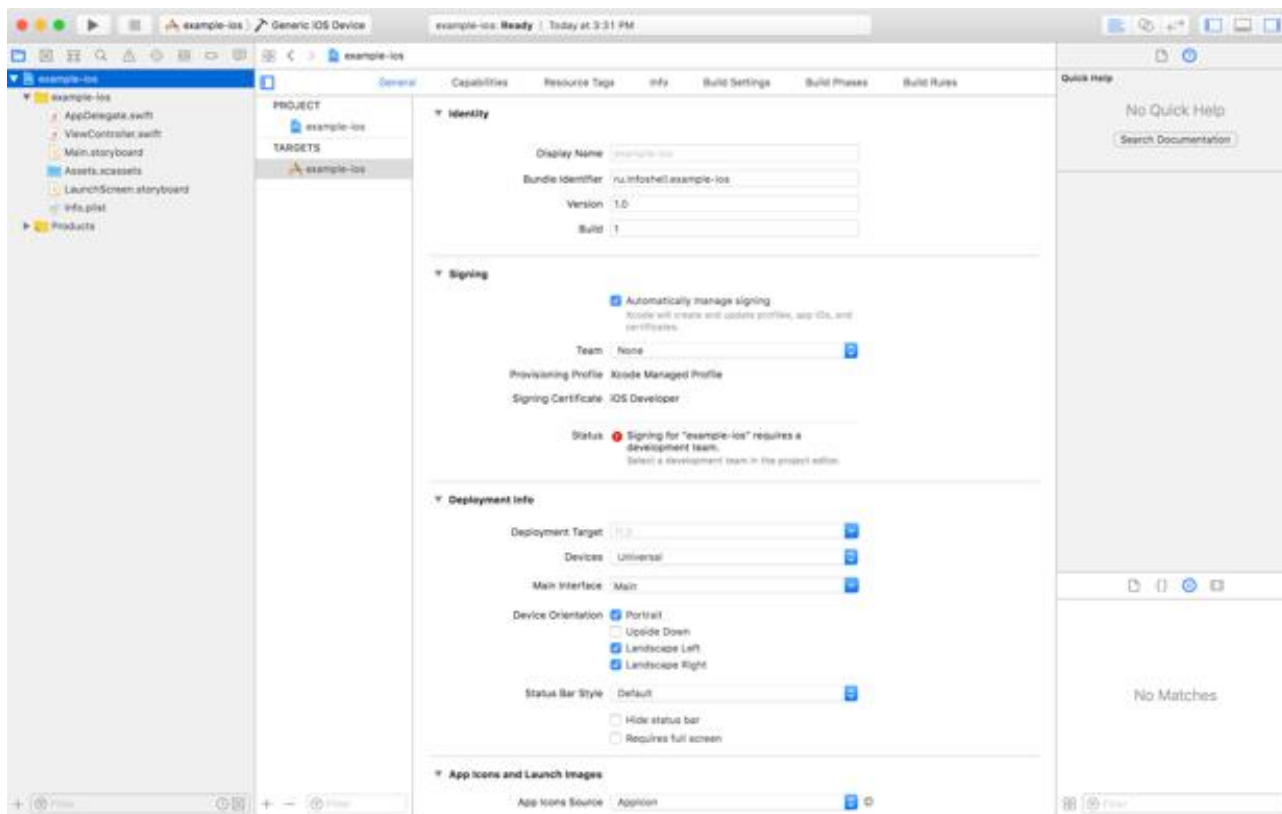
Заповнивши інформацію, натискаємо Next і вибираємо місце проекту на нашому комп'ютері, тиснемо Create.

У нас відкривається наш основний робочий простір і давайте розберемося, що тут є. Інтерфейс досить складний, тому розділимо його на кілька частин: вгорі, ліворуч, праворуч, внизу та посередині.

У верхній частині робочого простору ви бачимо кнопки запуску або зупинки програми, поле налаштування запуску, яке складається з двох частин: що запускаємо, на чому запускаємо.

Також ми бачимо рядок стану, в якому можна спостерігати поточне завдання, яке виконується в середовищі розробки iOS. Правіше ми можемо побачити налаштування виду редактора: стандартний редактор, редактор з асистентом та перегляд змін у системі контролю версій, а також є кнопки, щоб

показати/приховати частини середовища, які ми розглядатимемо далі: знизу (інформація налагодження), справа (навігація) та зліва (утиліти).



Зліва ми бачимо кілька кнопок, які перемикатимуть функціонал лівої панелі, а саме:

- Навігація по файлам та папкам
- Система контролю версій
- Ієрархія класів у проекті
- Пошук за проектом
- Повідомлення про попередження та помилки при складанні
- Автоматизовані тести
- Налаштування
- Точки зупинки
- Логи збірок

Праворуч у нас є дві частини: верхня та нижня. Вся права панель – утиліти. У верхній частині ми побачимо налаштування вибраного файлу та/або елемента

інтерфейсу користувача. У нижній частині утиліт ми бачимо елементи інтерфейсу, які ми можемо використовувати.

У нижній частині програмного середовища розробки ми можемо побачити налагоджувальну інформацію, логі програми та інші корисні дані, необхідні для перевірки правильної роботи мобільного додатку iOS. Відкрити нижню частину можна за допомогою передостанньої кнопки у верхній частині програми, а також при запуску дана панель відкривається автоматично.

І в основній (центральной) частині середовища ми можемо спостерігати редактор, в якому пишеться код і ведеться робота над інтерфейсом програми.

– Identity

- Display Name — Ім'я програми, що відобразиться на домашньому екрані мобільного пристрою.
- Bundle ID — унікальний ідентифікатор, який використовуватиметься в App Store
- Version — Поточна версія проекту, яка ведеться за допомогою семантичного версіонування.
- Build — Номер збирання, який слід збільшувати на одиницю з кожною установкою програми на пристрій

– Signing

- Automatically manage signing — автоматичне керування підписом програми
- Team – найменування вашої команди iOS розробників
- Provisioning profile — профіль для підпису
- Signing Certificate — сертифікат, який дозволяє підписувати програму в Apple

– Deployment Info

- Deployment Target — мінімальна версія iOS, на яку може бути встановлений продукт, що розробляється
- Devices — Типи пристроїв
 - iPhone

- iPad
- Універсальний
- Main Interface — Storyboard файл, з якого буде починатися програма після
- Device Orientation — Можливі орієнтації пристрою
 - Портретна
 - Догори ногами
 - Альбомний проти годинника
 - Альбомний за годинником
- Status Bar Style - Колір статус бару (верхня панель інформації з часом, оператором та батареєю)
 - Світлий
 - Темний
 - Приховати
- App Icons and Launch Images
 - App Icons Source — значок програми
 - Launch Screen File — екран, який з'являється під час запуску

Створення iOS додатку

Насамперед, нам потрібно знайти файл інтерфейсу, який називається Main.storyboard. Його можна знайти зліва на панелі навігації.

Відкривши файл, бачимо екран мобільного додатка. Тепер потрібно перетягнути елемент під назвою Label на цей екран. Цей елемент знаходиться зліва внизу.

Перетягуємо його на екран у будь-яке місце. Потім на панелі утиліт, шукаємо поле, в якому написано "Label".

Змінюємо Label на Hello World. Після цього ми побачимо, що наш елемент на екрані не показує повністю нашого тексту, який ми вписали, тому що у нього був текст спочатку менший, і він прийняв його розміри.

Object - Provides a template for objects and controllers not directly available in Interface Builder.

Label Label - A variably sized amount of static text.

Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Segmented Control, DatePicker

Filter

Label

Text Plain

Label

+ Color [Black] Default

+ Font System 17.0

Dynamic Type Automatically Adjusts Font

Alignment [Left] [Center] [Right] [Justified] [Natural]

Lines 1

Behavior Enabled Highlighted

[Icons]

He...

→

Label

Text Plain

Hello World

+ Color [Black] Default

+ Font System 17.0

Dynamic Type Automatically Adjusts Font

Alignment [Left] [Center] [Right] [Justified] [Natural]

Lines 1

Behavior Enabled Highlighted

Baseline Align Baselines

Line Break Truncate Tail

Autoshrink Fixed Font Size Tighten Letter Spacing

+ Highlighted [Black] Default

+ Shadow [Stroke] Default

Shadow Offset 0 Width -1 Height

View

Content Mode Left

Semantic Unspecified

Tag 0

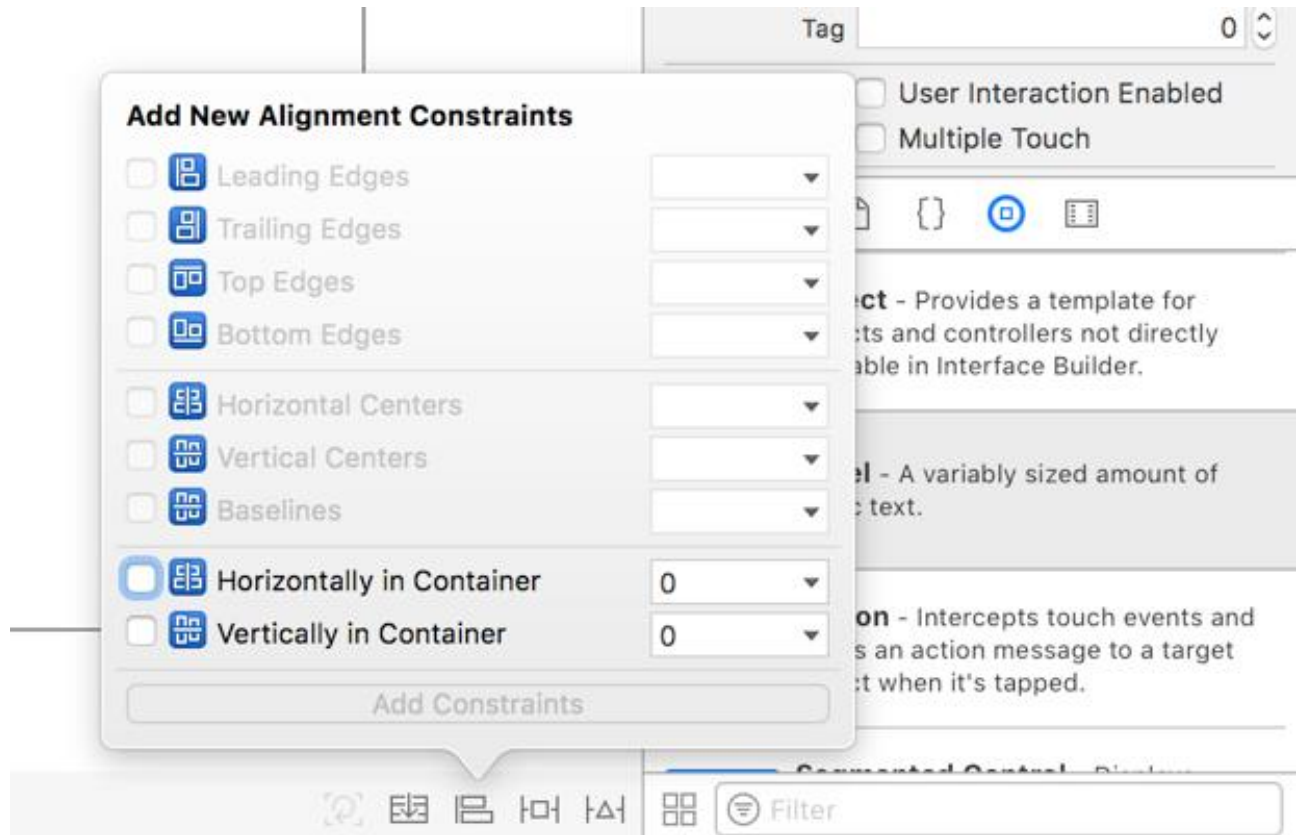
Interaction User Interaction Enabled Multiple Touch

Object - Provides a template for objects and controllers not directly available in Interface Builder.

Label Label - A variably sized amount of static text.

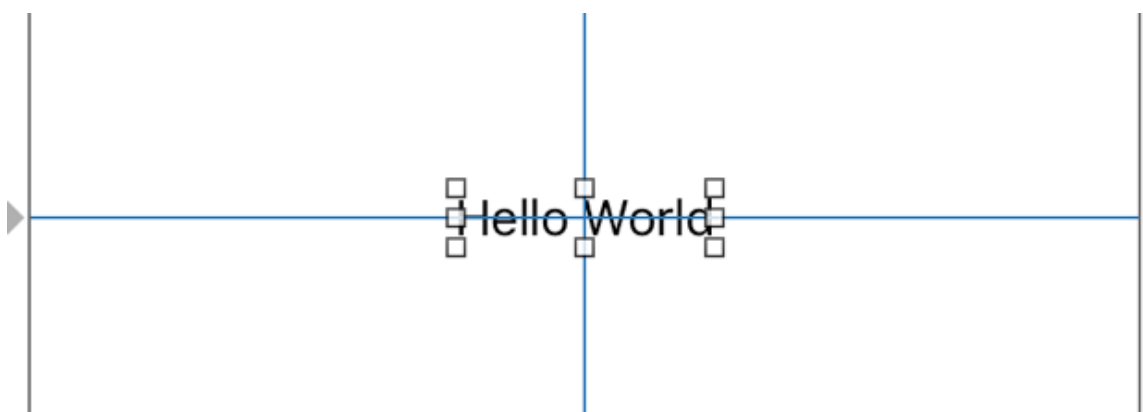
Button Button - Intercepts touch events and sends an action message to a target object when it's tapped.

Фіксуємо цей елемент по центру екрана, щоб він набув правильної форми і наш текст повністю відобразився. Для цього нам потрібно знайти другу кнопку внизу зліва, яка називається *Align* і натискаємо на неї.

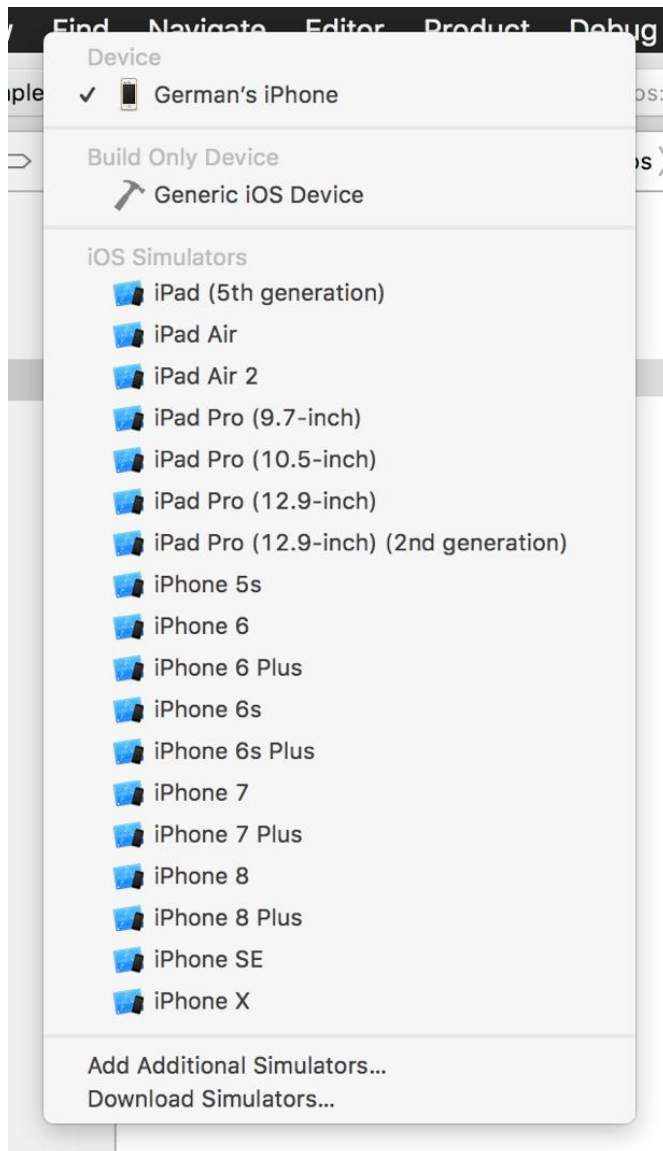


Тут бачимо деякі параметри розташування нашого елемента. Нам потрібні два останні параметри. Активуємо їх та натискаємо “Add 2 Constraints”.

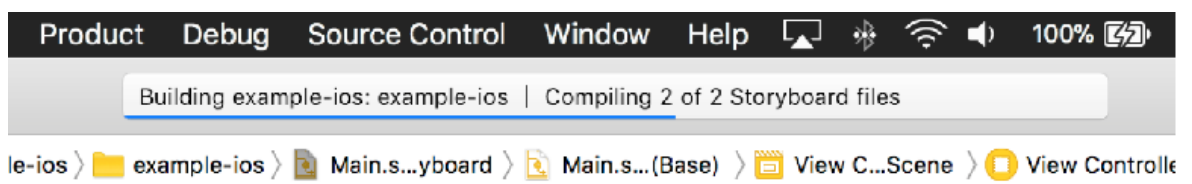
Після цього ми можемо спостерігати, як наш елемент зафіксувався посередині екрана і прийняв форму, в якій відображається текст повністю.



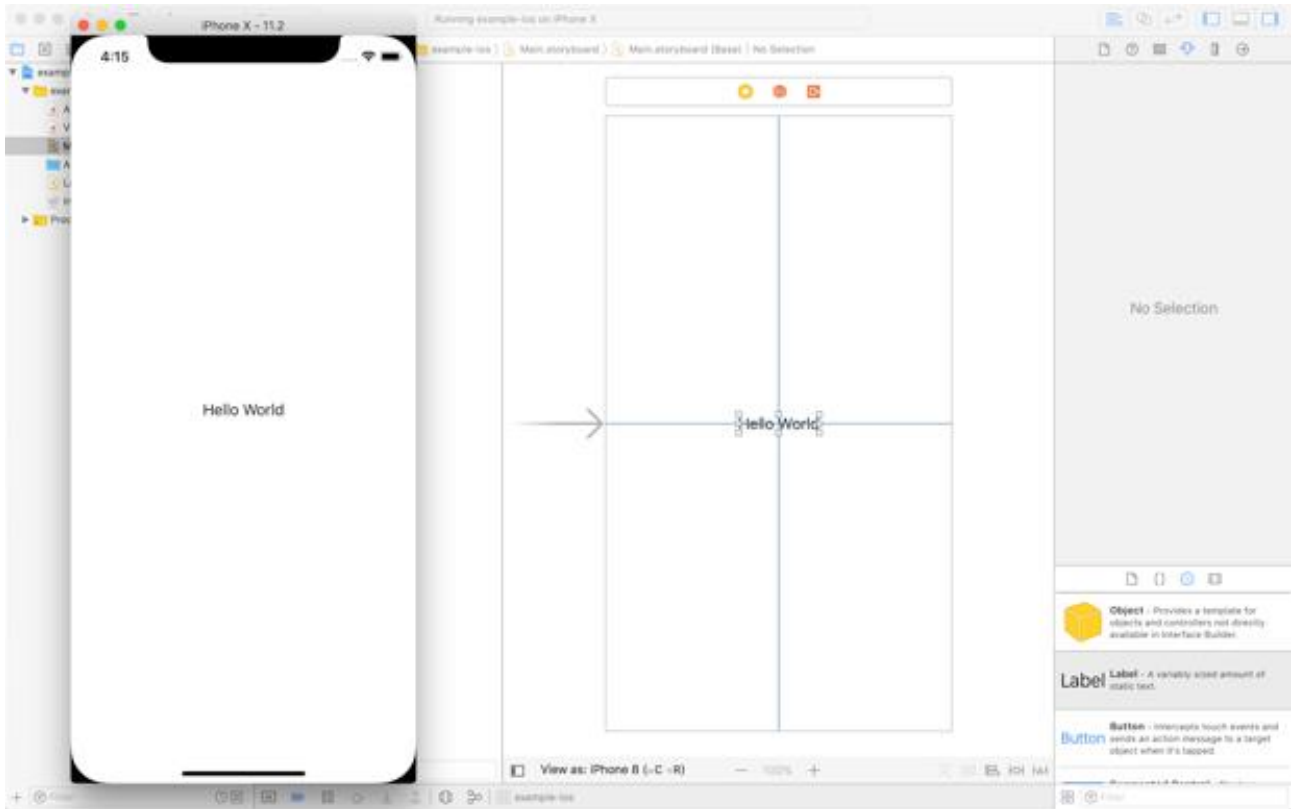
Тепер, нам потрібно запустити мобільний додаток на пристрої, і подивитися, чи все правильно відобразиться. Для цього, ми натискаємо вгорі на вибір призначення запуску програми і бачимо список, що випадає:



У цьому списку бачимо пристрої та їх емулятори, на яких ми зможемо здійснити запуск. Вибираємо будь-яке і тиснемо Run. Почнеться складання програми та Xcode покаже нам завдання, які він виконує в даний момент часу:



Після успішної збірки проекту ми побачимо, що Xcode перейде в режим очікування запуску і з'явиться вікно симулятора смартфона, яке завантажуватиметься. Після того, як симулятор завантажиться, ми зможемо спостерігати нашу готову програму на пристрої.



Завдання

Розробити додаток під iOS, використовуючи мову Swift згідно варіанту завдання (визначається викладачем).