

04. CocoaPods и работа с анимациями

Понятие Cocoa

Cocoa (в пер. с англ. – какао) – объектно-ориентированный API для операционной системы macOS производства компании Apple. Это один из пяти основных API, доступных в Mac OS X.

Приложения, использующие Cocoa, обычно разрабатываются с помощью среды разработки Apple Xcode и Interface Builder с использованием языков программирования: C, Objective-C и Swift. Однако, среда Cocoa также доступна и при разработке на других языках, таких как Ruby, Python и Perl.

С точки зрения конечного пользователя, Cocoa-приложения - это приложения, написанные с использованием программной среды Cocoa. Такие приложения обычно имеют характерный внешний вид, поскольку эта среда во многом упрощает поддержку принципов Apple Human Interface Guidelines.

Cocoa является продолжением программных сред NeXTSTEP и OPENSTEP, которые разрабатывались компанией NeXT в конце 1980-х годов. Apple приобрела NeXT в декабре 1996 года, и начала работу над операционной системой Rhapsody, которая должна была стать прямой наследницей OPENSTEP. Предполагалось, что она будет включать в себя Blue Box, для обеспечения эмуляции приложений Mac OS. База библиотек и поддержка формата исполняемых файлов OPENSTEP получила название Yellow Box. Rhapsody эволюционировала в Mac OS X, и Yellow Box превратилась в Cocoa. В результа-

те этого, названия классов Сосоа начинаются с букв NS (от NeXTStep): NSString, NSArray и т. п.

Core Animation

Core Animation – технология, разработанная компанией Apple для создания анимаций и представляет собой набор Objective-C классов для рендеринга графики, проектирования, и анимации. Динамические, анимированные пользовательские интерфейсы трудно создавать, но Core Animation позволяет создавать эти интерфейсы легче.

Анимации с Core Animation автоматизированы и могут быть получены с минимальным вмешательством разработчика. Когда разработчик изменяет атрибут компонента, Core Animation автоматически видоизменяет его промежуточными шагами (цвет, прозрачность и т. д.) между начальным и конечным значениями, визуально улучшая приложения и уменьшая количество исходного кода, который бы потребовался при использовании стандартных средств и технологий, предоставляемых Сосоа.

Используя Core Animation можно создавать динамические пользовательские интерфейсы для своих приложений без использования низкоуровневых API, графики, таких как OpenGL, чтобы получить респектабельную по производительности анимацию.

Пример кода для базовой анимации

1. Размещаем UIView и UIButton на выюконтроллере, расставляем констрейнты и создаем аутлеты и айбизекшин для кнопки.

9:41 AM



Core Animation

Run Core Animation

2. Создаем новый файл **UIButtonExtension**

```
import Foundation
import UIKit

extension UIButton {

    func pulsate() {

        let pulse = CASpringAnimation(keyPath: "transform.scale")
        pulse.duration = 0.6
        pulse.fromValue = 0.95
        pulse.toValue = 1
        pulse.autoreverses = true
        pulse.repeatCount = 2
        pulse.initialVelocity = 0|5
        pulse.damping = 1

        layer.add(pulse, forKey: nil)
    }
}
```

3. Переходим в айбизекшин

```
@IBAction func runCoreAnimation(_ sender: UIButton) {

    sender.pulsate()

    UIView.animate(withDuration: 1, delay: 0.5, options:
[.autoreverse, .repeat], animations: {
        self.blueView.frame.origin.x -= 20
    })
}
```

Понятие CocoaPods и фреймворков

Cocoapods – это менеджер зависимостей для проектов построенных на Swift и Objective-C. В cocoapods собрано более 48 тысяч библиотек, которые используются в более чем 3 миллионах приложений.

Работа с CocoaPods в терминале

Если ранее ни разу не ставили *CocoaPods*, то открываем терминал и запускаем команду `$ sudo gem install cocoapods`. Данная команда загрузит и установит все необходимые библиотеки для работы с *CocoaPods*.

1. `ls` – команда просмотра содержимого каталога, сокращение от английского слова «list». Введённая без параметров команда `ls` покажет вам названия всех видимых файлов и папок внутри текущего каталога.
2. Как перейти в другую папку? Очень просто. Запомните вторую самую главную команду: `cd`, означает Change Directory – сменить папку. Для этой команды недостаточно ввести её имя – так вы просто скажете, что хотите поменять папку, но не скажете, куда хотите перейти. Поэтому после `cd` нужен пробел и путь к каталогу, в который вы хотите переместиться.
3. Находясь в директории нашего проекта вводим команду `pod init` для создания нового под файла
4. Открываем созданный файл для редактирования `open -a Xcode Podfile`
5. Вводим путь для установки под файла `pod 'Spring', :git => 'https://github.com/MengTo/Spring.git'`

6. Запускаем процесс установки фреймворка `pod install`

Фреймворк **Spring** для работы с анимацией

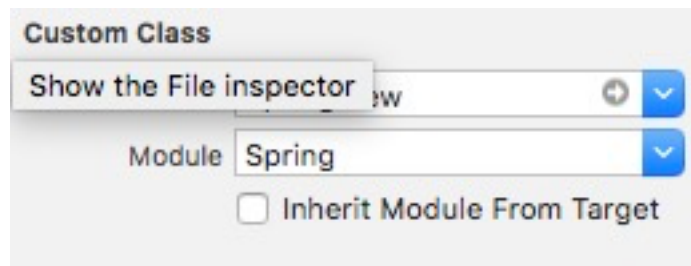
1. Создаем новые вью и кнопку, привязываем им соответствующие классы **SpringView** и **SpringButton**. Расставляем необходимые констрейнты и создаем аутлеты с соответствующими типами данных (Spring). (Если атлет привязать не получается, то необходимо собрать проект)

Core Animation







Spring Animation

Run Core Animation














Run Spring Animation









2. Создаем новый айбизекшин для кнопки
3. Открываем атрибуты **Spring View**:
















Spring View

Autostart	On	
Autohide	Default	
Animation	shake	
Force	2	
Delay	0,5	
Duration	1	
Damping	--	
Velocity	--	
Repeat Count	--	
X	--	
Y	--	
ScaleX	--	
ScaleY	--	
Rotate	--	
Curve	easeIn	

4. Открываем атрибуты **Spring Button**:



Spring Button

Autostart	On	
Autohide	Default	
Animation	flash	
Force	--	
Delay	--	
Duration	--	
Damping	--	
Velocity	--	
Repeat Count	--	
X	--	
Y	--	
ScaleX	--	
ScaleY	--	
Rotate	--	
Curve	Default	

5. Переходим в класс **ViewController** и импортируем фреймворк **Spring**.
6. Если не включили автостарт в сториборде для наших элементов, то для запуска анимации при старте приложения в методе `viewDidLoad` пишем:

```
springButton.animate()  
springView.animate()
```

8. айби экшин для спринг вью

```
@IBAction func runSpringAnimation(_ sender: SpringButton) {  
  
    springView.animation = "pop"  
    springView.curve = "easeIn"  
    springView.force = 2  
    springView.duration = 1  
    springView.delay = 0.3  
    springView.animate()  
}
```