



Введение в разработку iOS

Занятие №1



Евстратов Геннадий

План лекции



- Знакомство
- Что необходимо для курса
 - Знания
 - Софт и железо
- Рассказ про архитектуру и особенности iOS
- Демонстрация!

- Структура
- В конце курса реальное приложение
 - Интересно
 - Живые задачи
 - Полезно
 - Портфолио
 - Стажировки
 - Практика!

0 курсе 2/6 - Структура



- Лекции
- Практические занятия
- 4 домашних задания
- 2 рубежных контроля
- Защита проекта

0 курсе 3/6 - Лекции



- Немного теории
- Пояснение, откуда эта теория берется
- Объяснение, что с этим в итоге делать

0 курсе 4/6 - Практические занятия



- Куда нажимать
- Что писать
- Почему писать (см. лекции)

0 курсе 5/6 - Контроль



- Один проект на всё время
- Несколько человек на проект
- Защита проекта
- Реальный навык командной работы, уау!

0 курсе 6/6 - Преподаватели



- Геннадий
- Дмитрий
 - «Мой мир»
 - Moosic
 - Скоро к нам присоединится

Необходимо для старта 1/2



- Базовые знания программирования
- Понимание ООП
- Опыт (хотя бы небольшой) программирования
 - C
 - C++
 - Java
 - Ruby
- Примерно понимание многопоточности

Необходимо для старта 2/2



- Mac OSX
 - Xcode / Appcode
 - Dash для документации
 - Git для контроля версий
 - Hackintosh
- iPhone / iPad
 - Лучше настоящий
 - Есть симулятор

Введение в архитектуру 1/4



- iOS это unix-like (darwin)
 - POSIX
 - Приложения пользователя изолированы
- Фреймворки
 - Низкий уровень
 - CFNetwork
 - CoreGraphics
 - ...
- UIKit + верхний уровень

- Sandbox
 - В общем случае нет доступа к другим приложениям
 - Полная изоляция
- Фоновый режим
 - Аудио
 - VoIP
 - Геолокация
 - Скачивание данных
 - Обработка уведомлений

- Apple называет это MVC
 - И мы будем
- Model
 - Данные и работа с ними
- View
 - Кнопки, надписи, всё, что видно
- Controller
 - То, что всё это связывает вместе

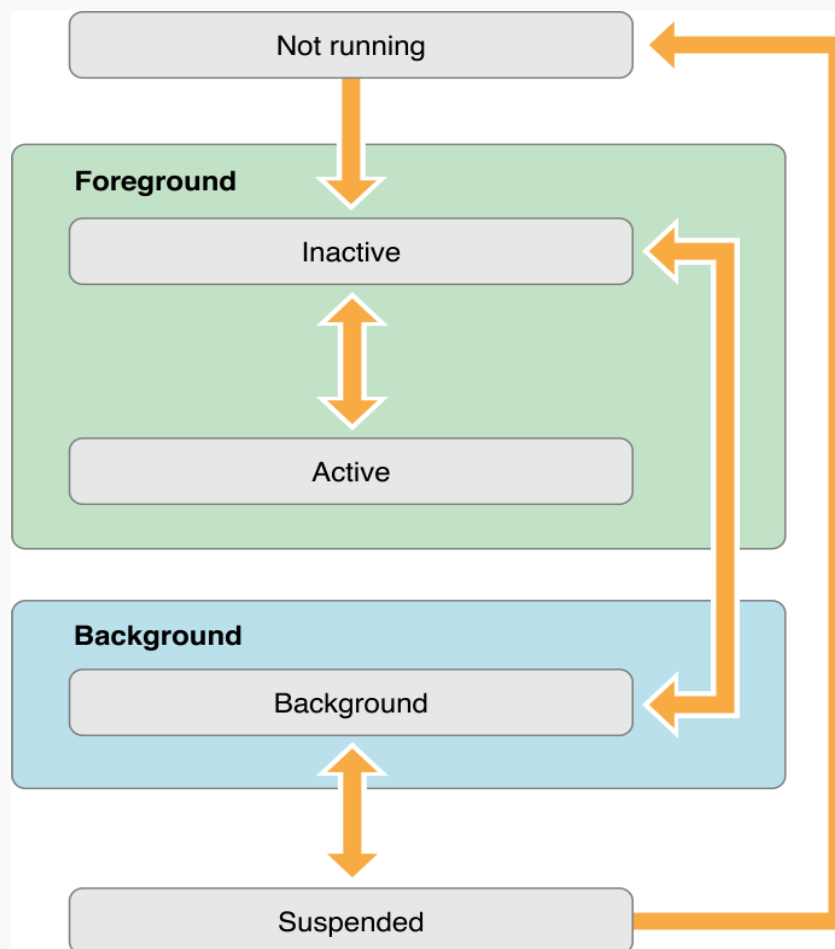
- Все работает в рамках RUNLOOP
 - Обработывает события
 - Рисует всё на экране
- Запуск приложения
 - App delegate получает события про это
 - Загружает первый Controller
- Тот обрабатывает жизненный цикл своего View
- И дальше по этой же схеме

Состояния приложения 1/2



- Active
 - Это когда оно на экране
- Inactive
 - На экране, но не получает событий
 - Например вылезла нотификация или что-то подобное
- Background
 - Не на экране, но выполняет что-то в фоне
 - Выполнять можно только определённые действия
- Suspended
 - Не на экране, и ничего не делает, просто занимает память
 - Система может его удалить из памяти без предупреждения

Состояния приложения 2/2

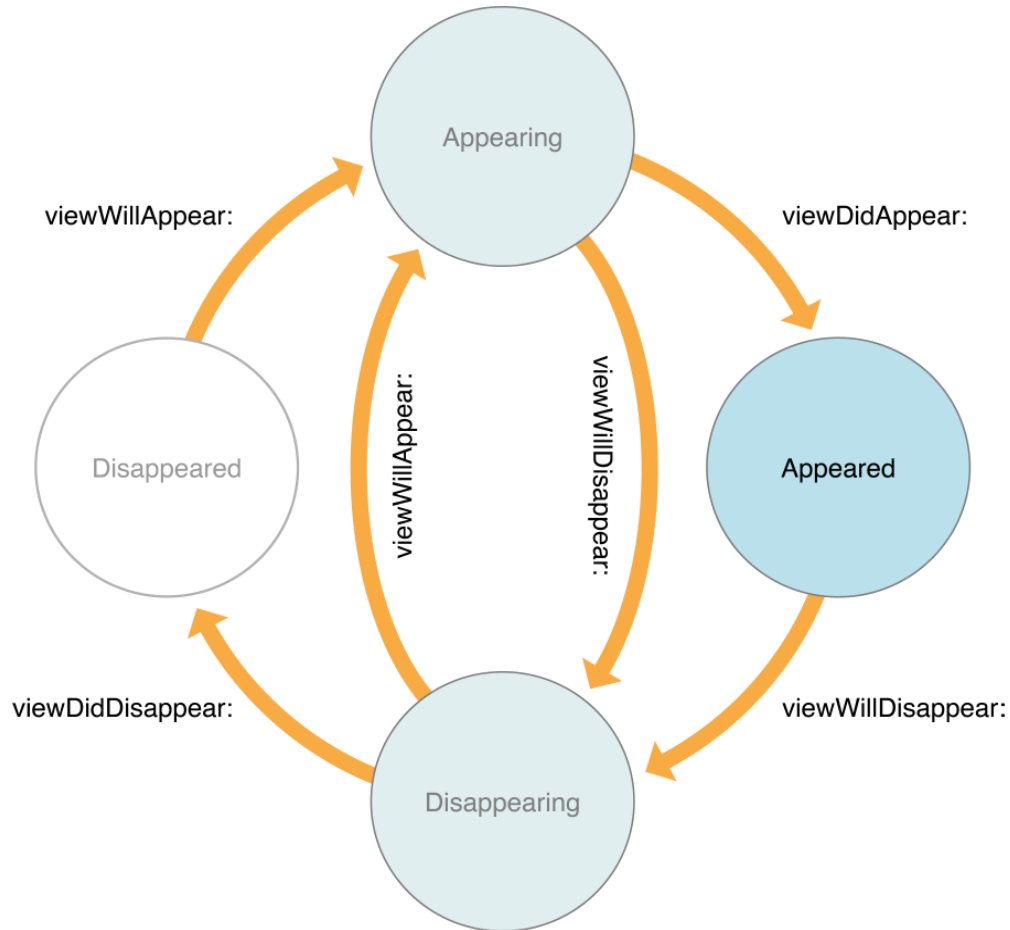


Состояния view 1/2



- Только что загрузился (viewDidLoad)
- Видимый
- Невидимый
 - не путать с тем, видно его реально на экране или нет

Состояния view 2/2



Как делается интерфейс



- Interface Builder — часть Xcode
- Xib
 - Все в отдельных файлах
 - Сами грузим интерфейсы
- Storyboard
 - Всё в одном файле
 - Указываем, что и как куда переходит
- Все можно сделать исключительно кодом
 - ОЧЕНЬ много печатать
- По большей части подходы комбинируют, нет единого правильного

Как делается интерфейс 1/3



- Interface Builder — часть Xcode
- Xib
 - Все в отдельных файлах
 - Сами грузим интерфейсы
- Storyboard
 - Всё в одном файле
 - Указываем, что и как куда переходит
- Все можно сделать исключительно кодом
 - ОЧЕНЬ много печатать
- По большей части подходы комбинируют, нет единого правильного

Как делается интерфейс 2/3



Основные понятия

- IBOutlet
 - Показывает, какой элемент какому полю класса соответствует
 - Можно делать массив из них, IBOutletCollection
- IBAction
 - Показывает, какой метод вызвать, когда что-то случается

Как делается интерфейс 3/3



Как всё расположить на экране?

- Autolayout
 - Система линейных неравенств
 - Решается алгоритмом Cassowary
- Руками задавать положение
- Подходы можно комбинировать

Будем загружать картинку из интернета и обновлять её
при нажатии на кнопку!

ОЧЕНЬ ИНТЕРЕСНО