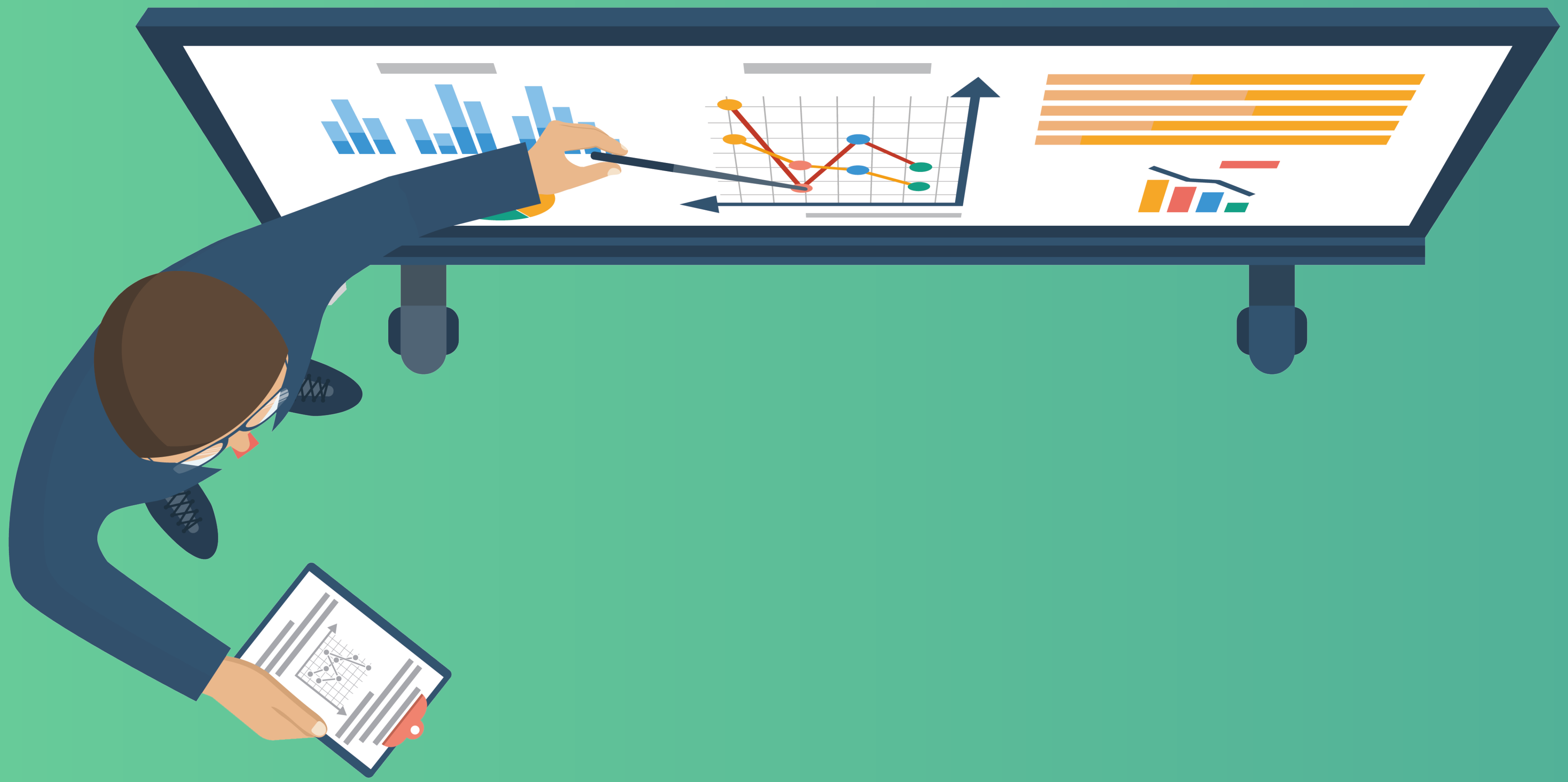


appropos academy

iOS Basics with Swift

appropos
academy

Introduction

- ➜ Name
- ➜ Experience
- ➜ Expectation

Schedule

	Day 1	Day 2	Day 3
9:00 - 10:30	Introduction + Setup	Storyboards	Networking
10:45 - 12:15	Swift I	Navigation + Lists	Dependencies + Wishlist
13:15 - 14:45	Swift II	Assets + Scheduling	Testing + Platform
15:00 - 16:30	Apps + Xcode	Finishing the Game	Distribution + Feedback

GitHub Repository:

github.com/ernesto-elsaesser/ios-training

Let’s start!

Please interrupt me if ...

- ... you have a question
- ... you want me to repeat something
- ... you need a break

iOS

What’s different?

Limitations

- Screen
- Energy
- Mobile Data
- Memory
- Input

Expectations

- Stability
- Performance
- Design
- Responsiveness
- Usability (HIG)

https://developer.apple.com/design/human-interface-guidelines/ios/overview/themes/

Swift

A New Language

After Apple unveiled the Swift programming language, it quickly became one of the ***fastest growing*** *languages in history. Swift makes it easy to write software that is incredibly* ***fast and safe*** *by design. Now that Swift is* ***open source****, you can help make the best general purpose programming language available everywhere.*

For students, learning Swift has been a great introduction to modern programming concepts and best practices. And because it is now open, their Swift skills will be able to be applied to an even broader range of platforms, from ***mobile*** *devices to the* ***desktop*** *to the* ***cloud****.*

- **The Swift Team** - https://docs.swift.org

A Safe Language

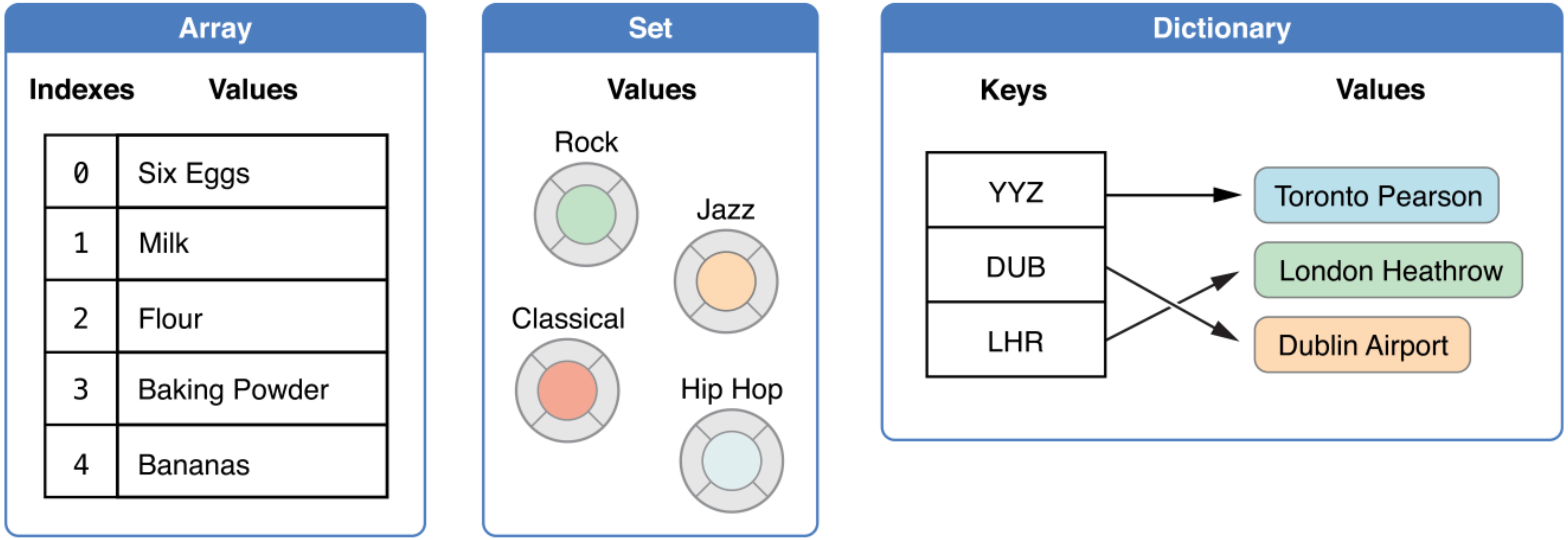
Swift defines away large classes of common programming errors by adopting modern programming patterns:

- *Variables are always initialized before use*
- *Array indices are checked for out-of-bounds errors**
- *Integers are checked for overflow*
- *Optionals ensure that null values are handled explicitly*
- *Memory is managed automatically*
- *Error handling allows controlled recovery from unexpected failures*

- **The Swift Programming Language (Swift 5)** - https://docs.swift.org/swift-book/

appropos academy

Collection Types



Array

Indexes	Values
0	Six Eggs
1	Milk
2	Flour
3	Baking Powder
4	Bananas

Set

Values

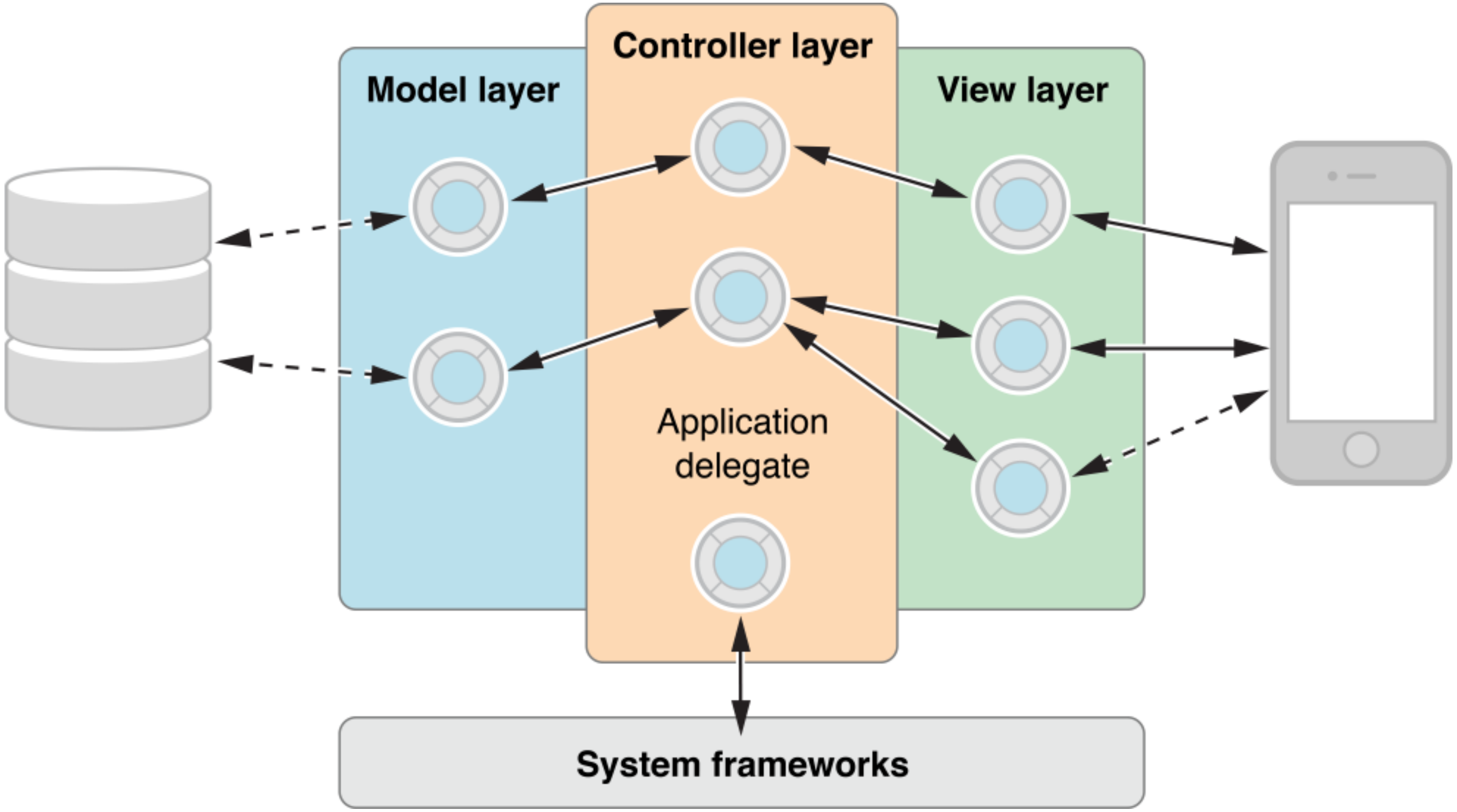
Dictionary

Keys	Values
YYZ	Toronto Pearson
DUB	Dublin Airport
LHR	London Heathrow

The Swift Programming Language (Swift 4.2) https://docs.swift.org/swift-book/LanguageGuide/CollectionTypes.html

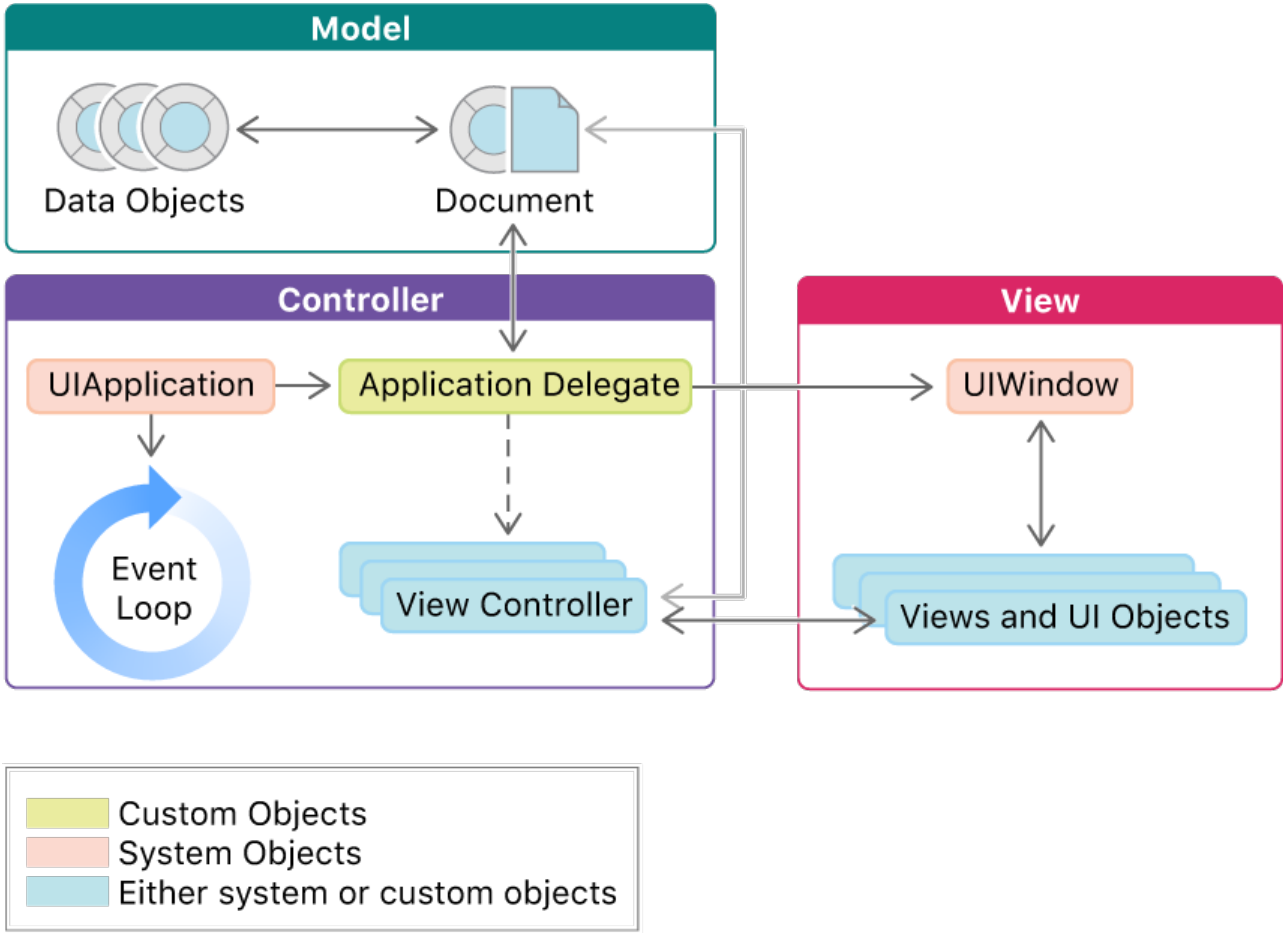
Apps

Model View Controller



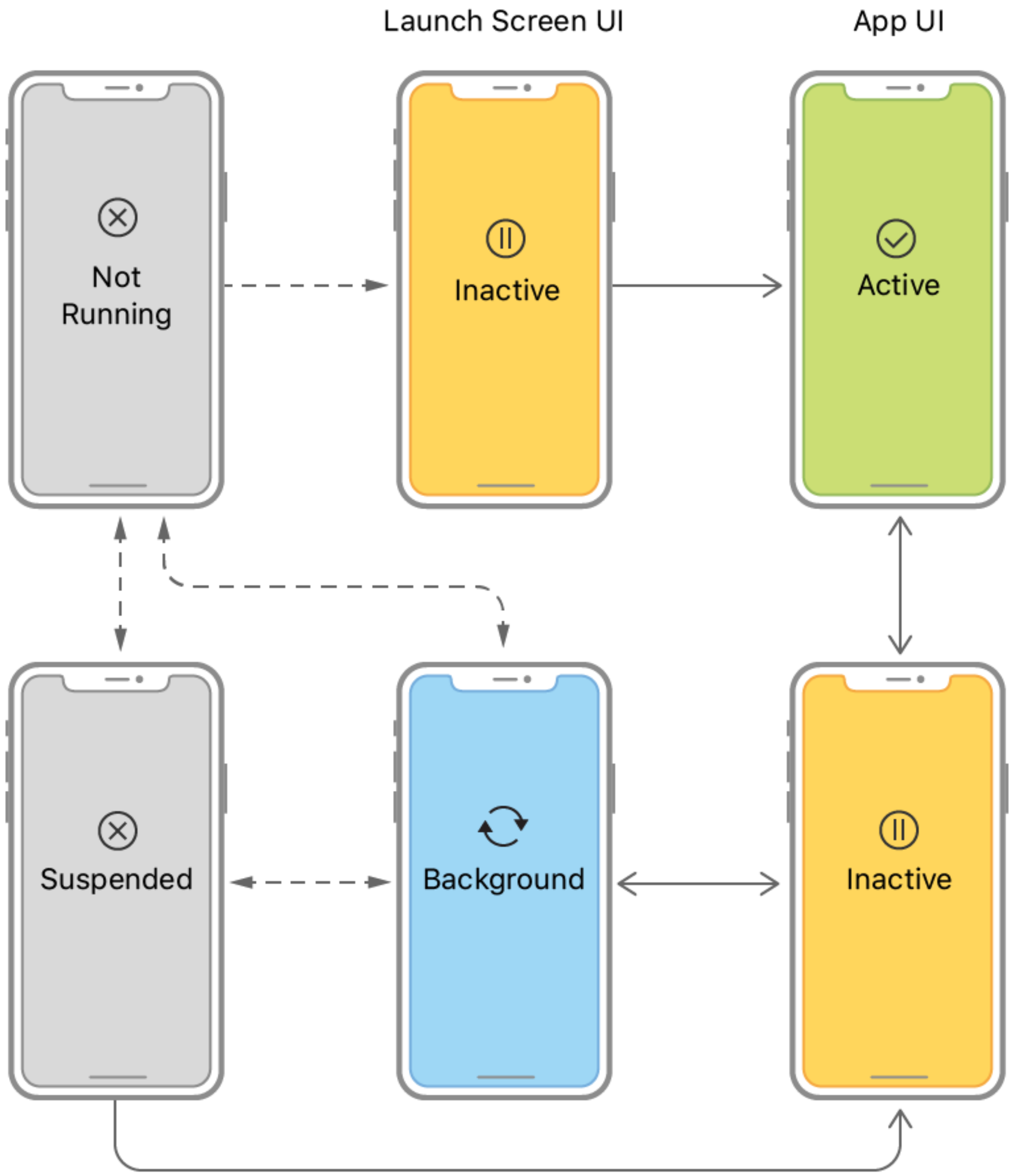
appropos academy

App Architektur



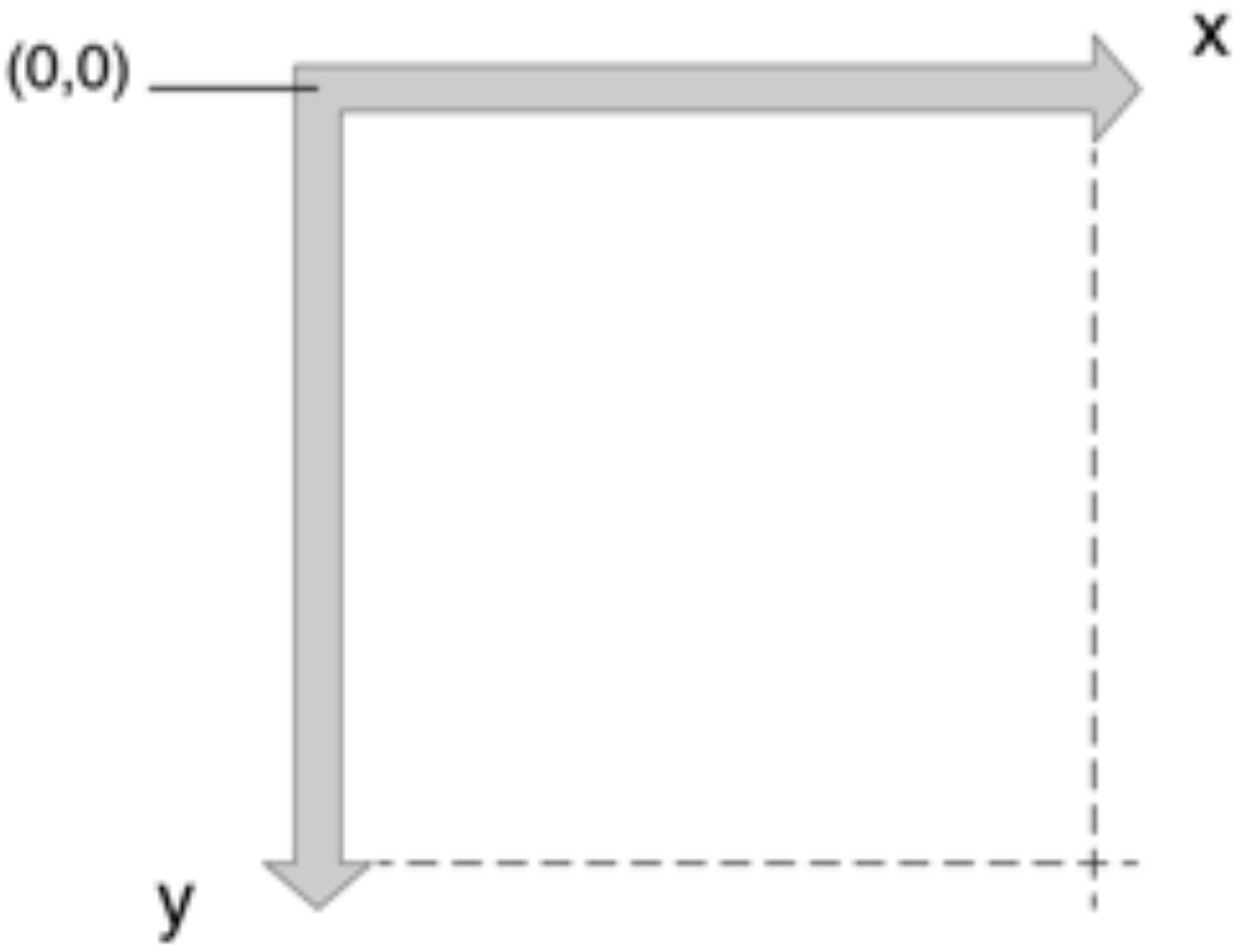
UIKit Doumentation https://developer.apple.com/documentation/uikit/about_app_development_with_uikit

App Lifecycle



UIKit Documentation https://developer.apple.com/documentation/uikit/app_and_scenes/managing_your_app_s_life_cycle

The UI Coordinate System



Drawing and Printing Guide for iOS https://developer.apple.com/library/archive/documentation/2DDrawing/Conceptual/DrawingPrintingiOS

Let’s code!

a very simple game

Task 1

- UIViewController with button and label
- tapping the button appends an “!” to the label text

Task 2

- UIViewController with 3 **game buttons** and a start button
- start button tapped > **mark** a random **game button**
- **marked** button tapped > **unmark** the button

Hint: Int.random(...)

Task 3

- UITableViewController with 5+ **cells**
- **cell** gets **marked** on tap (change text)
- **cell** gets **unmarked** on another tap (change text)

Task 4

- UITableViewController with 5+ **image cells**
- **image cell** gets **marked** on tap (show image)
- **image cell** gets **unmarked** on another tap (hide image)

Dispatch Queues

FIFO queues to which your application can submit tasks in the form of block objects

- ***Main queue:*** *runs on the main thread and is a serial queue*
- ***Global queues:*** *four concurrent queues that are shared by the whole system. Priorities: high, default, low, and background (is throttled in any I/O activity to minimize negative system impact)*
- ***Custom queues:*** *self-created queues, can be serial or concurrent (blocks in these queues actually end up in one of the global queues)*

Concurrency Programming Guide https://developer.apple.com/library/archive/documentation/General/Conceptual/ConcurrencyProgrammingGuide/Introduction/Introduction.html

Dispatch Queues

Blocks are scheduled synchronously or asynchronously

- ***Synchronously:*** *code waits until that block finishes execution*
- ***Asynchronously:*** *code continues executing while the block runs elsewhere*

Attempting to synchronously execute a block on the main queue results in a ***deadlock****!*

Concurrency Programming Guide https://developer.apple.com/library/archive/documentation/General/Conceptual/ConcurrencyProgrammingGuide/Introduction/Introduction.html

Task 5

- taps can only **unmark** **image cells**
- a random **image cell** becomes **marked** after some time
- repeat until all **image cells** are **marked**

Hint: asyncAfter(...)

Task 6

- random **marking** accelerates with every tap
- UIAlertController pops up when all **image cells** are **marked**
- bonus: app icon, launch screen, save longest run (taps)

Package Management

CocoaPods	Carthage	Swift Package Manager
old stable default option deeply integrated complex configuration	newer lightweight more manual steps not always supported repository = package	newest official well integrated doesn't support iOS yet

List of selected libraries: https://github.com/matteocrippa/awesome-swift

The iOS Platform

Hardware

Connectivity	Sensors	Other
• GSM/HSPA/LTE • WLAN • Bluetooth (BLE) • GPS • NFC*	• Face ID / Touch ID • Barometer • Three-axis gyro • Accelerometer • Proximity sensor • Ambient light sensor	• Camera • Speakers • 3D Touch

Software

Content	Graphics	Data
WebKit	Metal	CloudKit
MapKit	OpenGL	EventKit
ARKit	SceneKit	HealthKit
iAd	SpriteKit	Core Data
Devices	**Other**	
CoreBluetooth	SiriKit	Core Motion
iBeacon	PassKit	UserNotifications
WatchKit	Game Center	AirPlay
HomeKit	Core ML	StoreKit

Apple Developer Documentation https://developer.apple.com/documentation/

Distribution

Distribution Process

- Enroll into Apple Developer Program
- Create App Store listing (Bundle ID, Texts, Images, ...)
- Submit release build
- Review by Apple (automated + manual, up to 5 days)
- Approval, rejection or request for changes
- Publishing (up to 1 day until visible everywhere)

Pricing

Apple Developer Program (prices per year)

Apple ID (0$/y) / Individual (99$/y) / Organization (99$/y) / Enterprise (299$/y)

https://developer.apple.com/support/compare-memberships/

Business Models (fees per transaction)

Free / Freemium (30%) / Paid (30%) / Subscription (15-30%)

https://developer.apple.com/app-store/business-models/

Feedback

➜ Theory

➜ Practice

➜ Support Material