

# Become an iPhone Developer

*Darren McKie*

# Contents

- Bedroom Game Developers
- What can the iPhone Do?
- How to make a good iPhone game
- What do you need to become an iPhone Developer?
- Xcode Development
- Development Resources

# Bedroom Game Developers

# Modern Game Developers

- Modern developers have large teams of people
  - Designers
  - Programmers
  - Artists
  - Sound Engineers & Musicians
  - Testers
- Modern games can cost up to £12 million to produce

# Bedroom Developers

- Historically it was possible for one person to develop a computer game
- These people were often teenagers developing computer games in their bedrooms



# Famous 'Bedroom' Games

- Manic Miner (1983)
  - Mathew Smith



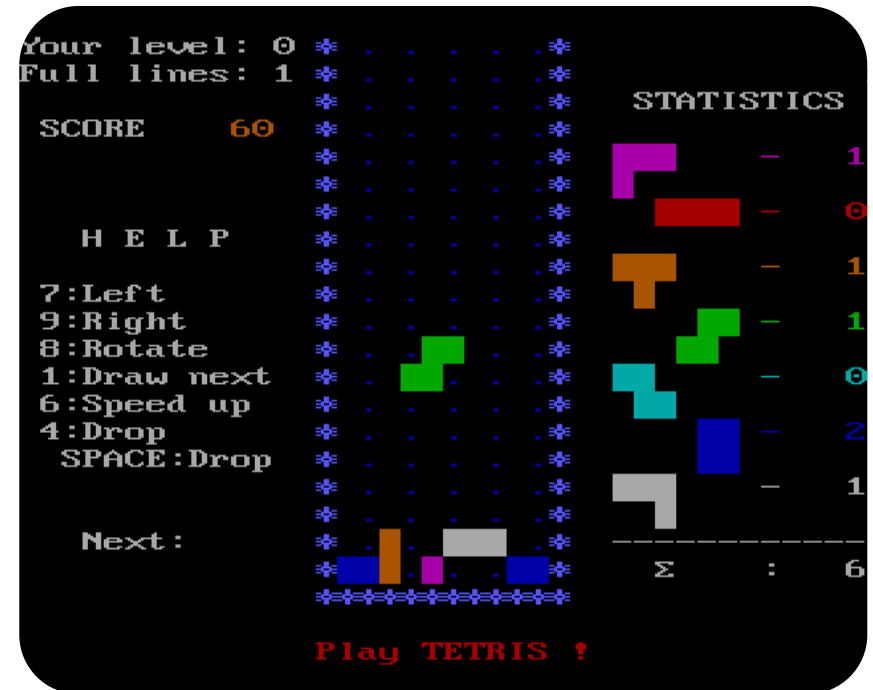
# Famous 'Bedroom' Games

- Jet Set Willy (1984)
  - Mathew Smith



# Famous 'Bedroom' Games

- Tetris (1984)
  - Alexey Pajitnov



# Analysing Retro Games

- It was possible for one person to develop these retro games because:
  - There were no special Hardware Development Kits required
  - There were very limited graphic and audio capabilities
  - There were very limited processing power
- All of these factors are the same for modern mobile phone development

What can the iPhone Do?

---

# iPhone / iPod Touch

- Multi-touch screen interface (480 x 320)
- A 3-axis accelerometer
- Camera (iPhones)
- Wi-Fi
- Bluetooth
- GPS (iPhone 3G & 3GS)
- Digital compass (iPhone 3GS)
- OpenGL ES (at least v1.1)

# Game Capabilities

- OpenGL ES graphics
  - v1.1
  - v2.0 GLSL Shaders
- Bluetooth
  - Peer-to-peer gaming
- 3-axis accelerometer
  - Used for user input, e.g. driving games



Invaders – [dmckie.co.uk](http://dmckie.co.uk)

# How to make a good iPhone game

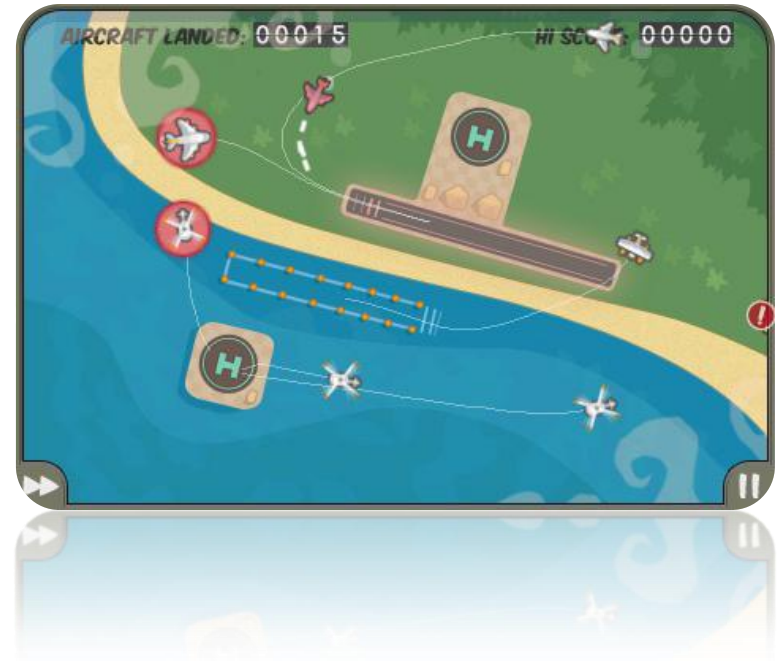
# Flight Control (by Firemint)

- Aim:
  - To direct planes to land safely
  - Avoid planes crashing into each other
  
- Gameplay:
  - Simply touch a plane and then change its flight path by moving you finger



# Flight Control

- Graphics:
  - Single screen playing field
  - 2D
  - Cartoon representation



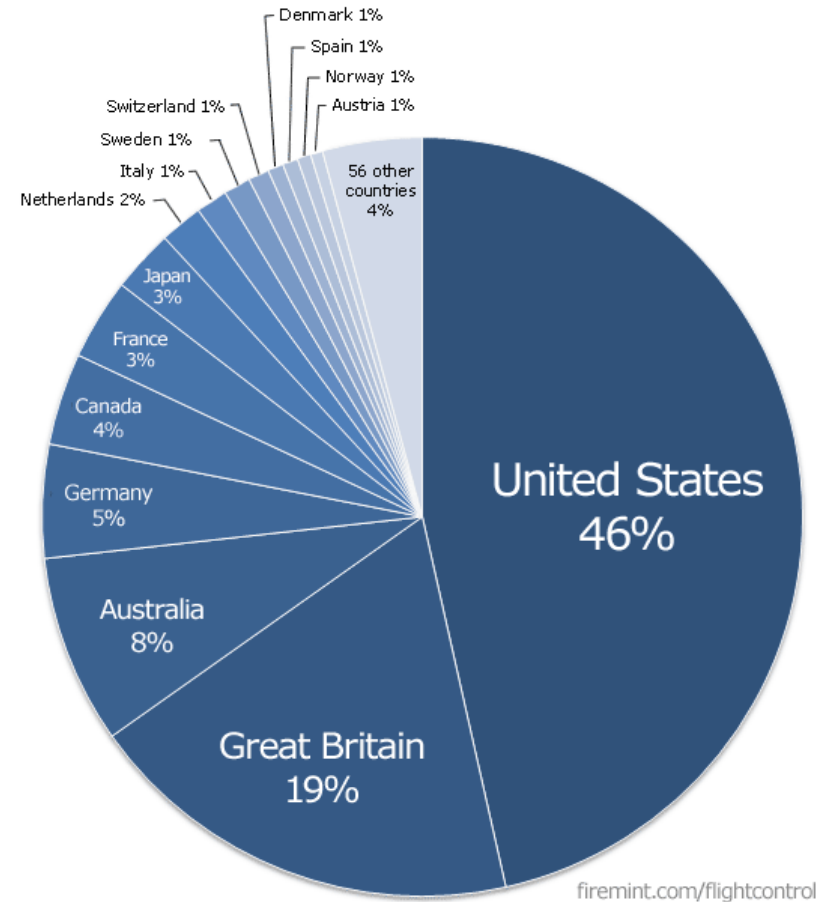
# Flight Control - Sales

- Hit #1 in at least 20 countries
- Over 2 million game sales
- Game costs \$0.99
- \$1.4 million revenue
  - Apple take 30% cut of sales
  - Subject to tax

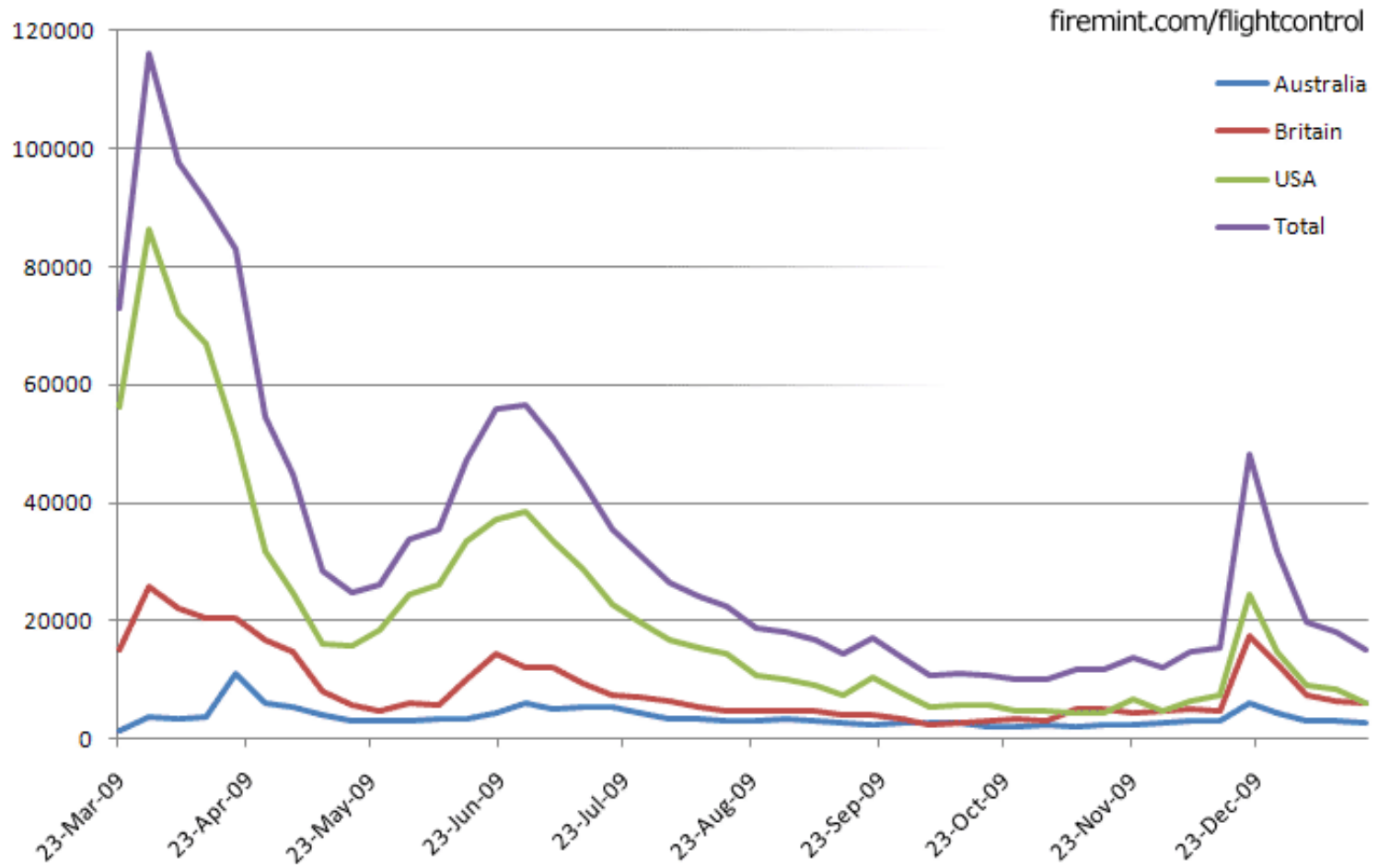


# Flight Control - Country Sales

- United States = 46%
- Great Britain = 19%
- Australia = 8%
- Germany = 5%
- Canada = 4%
- France = 3%
- Japan = 3%
- 64 other countries = 12%



# Flight Control – Weekly Sales



# How Can a Game Become Popular?

- Developers rely heavily on word of mouth and reviews
  - so the game has to sell itself
  - Apple gives you 50 free download vouchers to give to reviewers
- A simple name for the game is vital
  - the “browse and buy” audience responds well to simple game naming and titling
  - screenshots are also important

# How Can a Game Become Popular?

- The game must meet people's expectations rather than necessarily blowing players away
  - simple gameplay over gimmicks
- A simple description and video of the game is important to show gamers that the title is simple and fun to play, as opposed to being too difficult or too complex
- Correct pricing of the game
  - Rob Murray from Firemint who created Flight Control said “the right price is the price that you're just a little bit uncomfortable with”

# AOGI?!



What do you need to become  
an iPhone Developer?

# Required Hardware

- An Apple Mac with Mac OS X v10.5 or later
  - No special Hardware Development Kit required
- *Optional*: Either an iPhone or an iPod Touch
  - If you want to publish your games

## Required Membership

- You must register with Apple as a developer
- Free to register if you are only going to use the emulator
  - <http://developer.apple.com/programs/register/>
  - Download all of the required software and emulator
  - Access to all developer documentation
- £59 per year to become a full developer
  - <http://developer.apple.com/iphone/program/>
  - Allows for development directly on a device (i.e. iPhone or iPod Touch)
  - Publish your game to the App Store and make money ££

---

# Required Software

- Xcode
  - Free integrated development environment (IDE) allows you to develop games
  - Design your game's user interface
  - Comes with an emulator that allows for your game to run directly on the Mac without the need for a real device

# The Objective-C Language

- This is the native language of the iPhone
- You must use objective-c to handle all of the native functionality, e.g. Multi-touch, camera, loading/saving.
  - Class declaration in header files (.h)
  - Class definition in program files (.m)

# The C++ Language

- This can be used in conjunction with objective-c classes
- This allows you to re-use your code on other platforms
  - Class declaration in header files (.h)
  - Class definition in program files (.cpp)

---

# The Objective-C++ Language

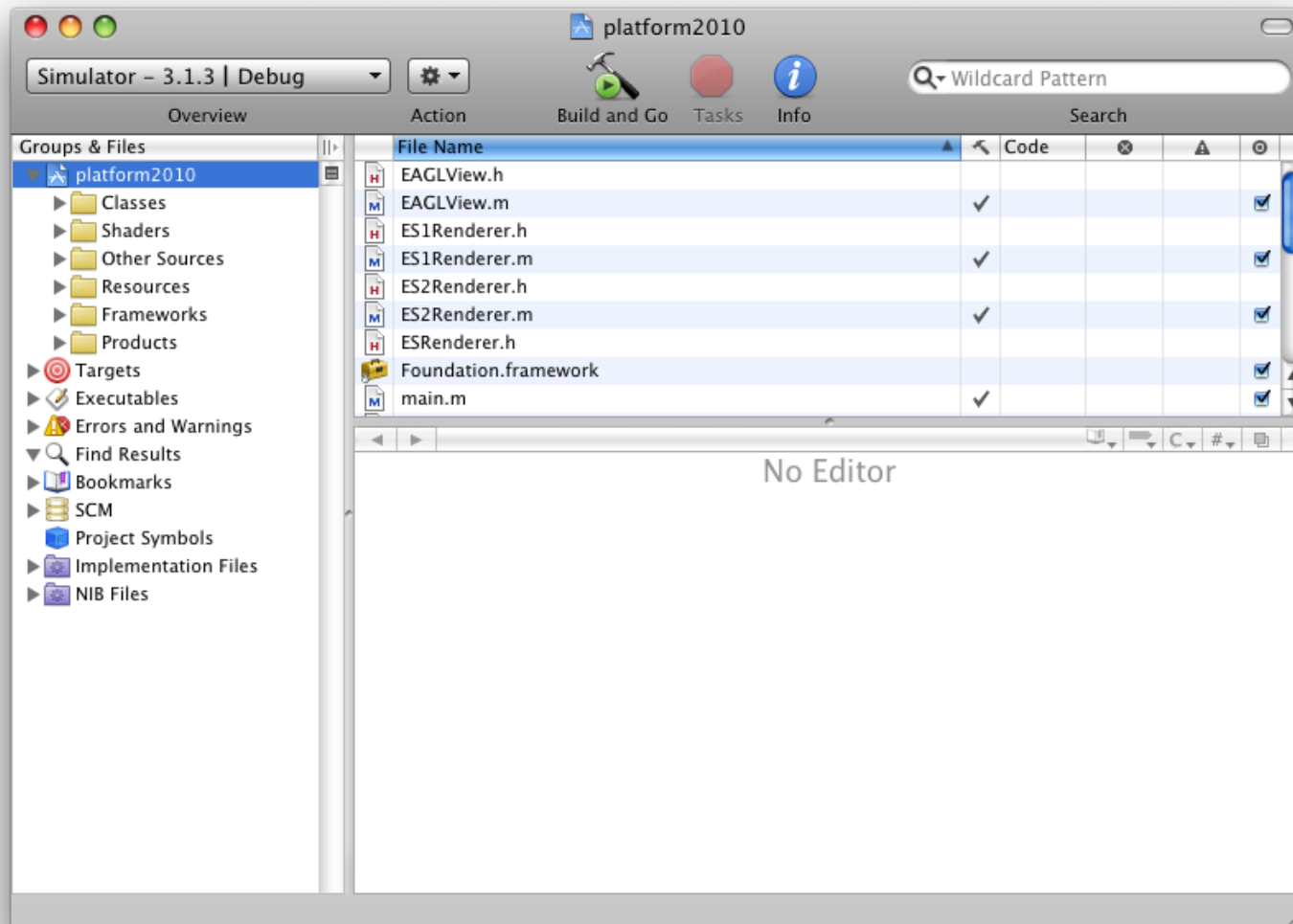
- Objective-C and C++ code can be used together in the same class, often known as objective-c++
  - Class declaration in header files (.h)
  - Class definition in program files (.mm)

# Xcode Development

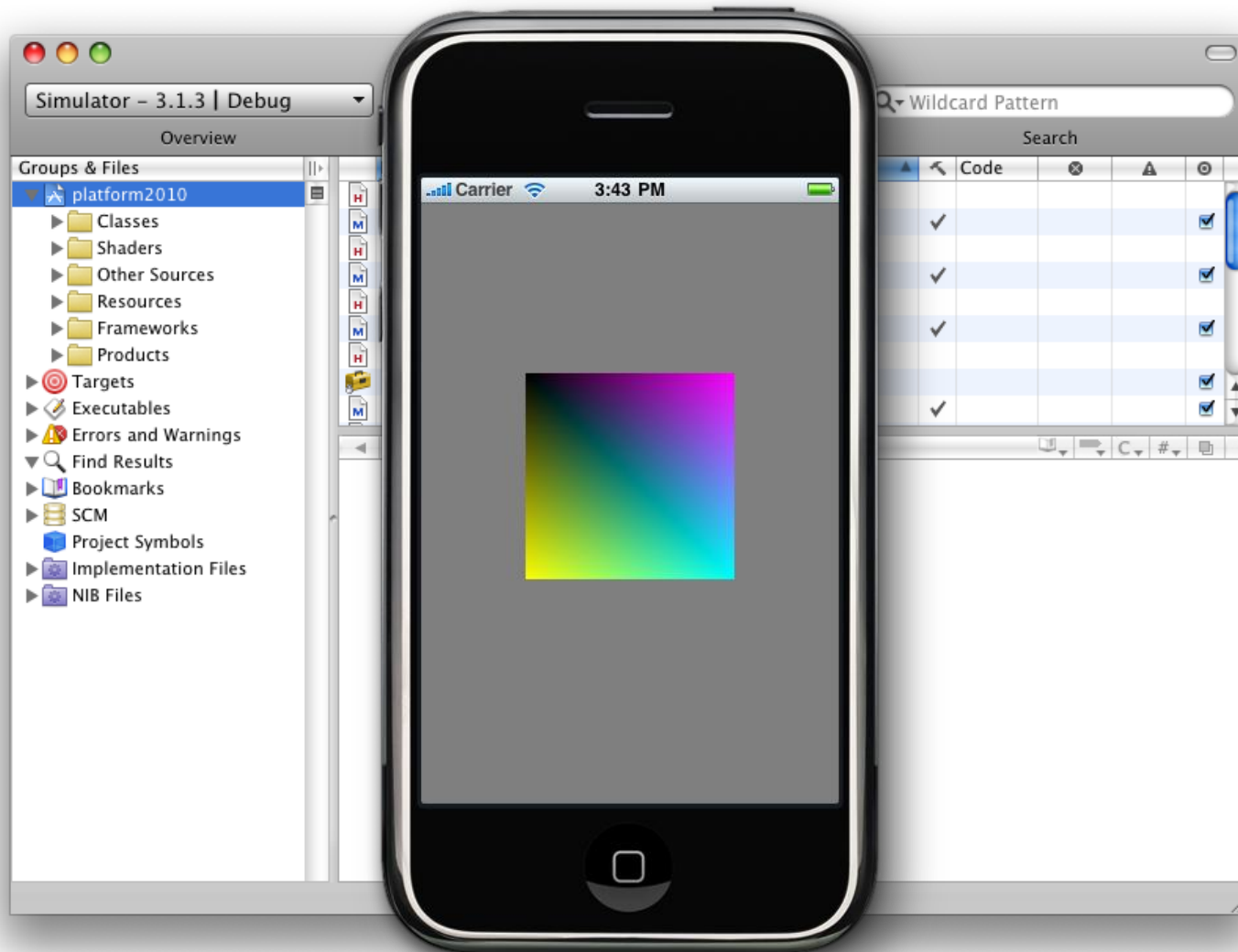
# Create an iPhone Project



# Create an iPhone Project

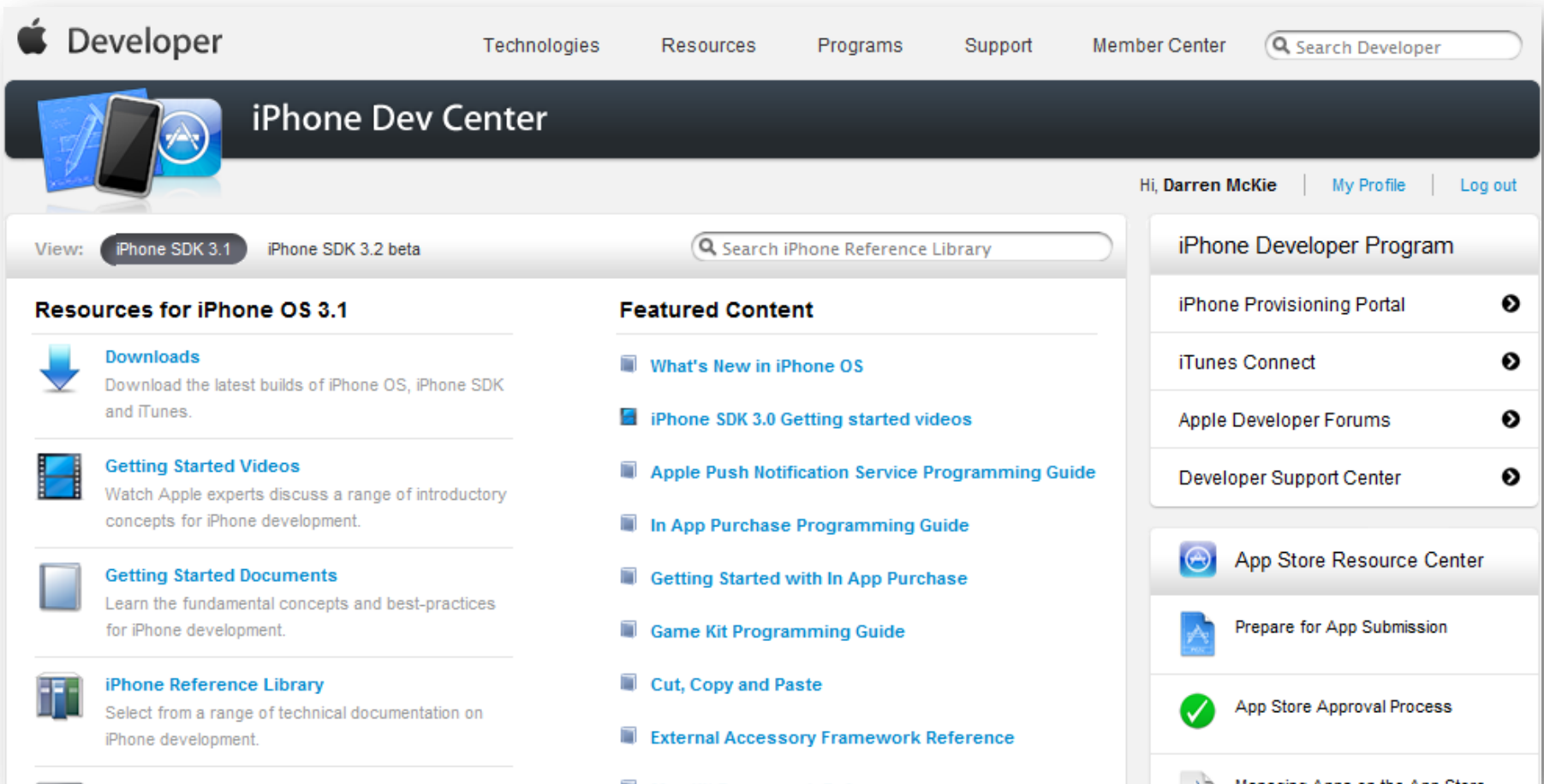


# Create an iPhone Project



# Development Resources

# iPhone Developer Center



The screenshot shows the Apple Developer website's iPhone Dev Center. At the top, there is a navigation bar with the Apple logo, the word "Developer", and links for Technologies, Resources, Programs, Support, and Member Center. A search bar labeled "Search Developer" is on the right. Below this is a dark blue header for "iPhone Dev Center" with an icon of an iPhone and development tools. On the right side of the header, it says "Hi, Darren McKie" with links for "My Profile" and "Log out".

The main content area has a sub-navigation bar with "View: iPhone SDK 3.1" (selected) and "iPhone SDK 3.2 beta", and a search bar for the "iPhone Reference Library".

On the left, there are four resource sections for "iPhone OS 3.1":
 

- Downloads**: Download the latest builds of iPhone OS, iPhone SDK and iTunes.
- Getting Started Videos**: Watch Apple experts discuss a range of introductory concepts for iPhone development.
- Getting Started Documents**: Learn the fundamental concepts and best-practices for iPhone development.
- iPhone Reference Library**: Select from a range of technical documentation on iPhone development.

In the center, there is a "Featured Content" section with a list of links:
 

- What's New in iPhone OS
- iPhone SDK 3.0 Getting started videos
- Apple Push Notification Service Programming Guide
- In App Purchase Programming Guide
- Getting Started with In App Purchase
- Game Kit Programming Guide
- Cut, Copy and Paste
- External Accessory Framework Reference

On the right, there is a sidebar titled "iPhone Developer Program" containing:
 

- iPhone Provisioning Portal
- iTunes Connect
- Apple Developer Forums
- Developer Support Center
- App Store Resource Center
- Prepare for App Submission
- App Store Approval Process (with a green checkmark icon)
- Managing Apps on the App Store

# iPhone Developer Center



## Getting Started Videos

Watch Apple experts discuss a range of introductory concepts for iPhone development.



## Getting Started Documents

Learn the fundamental concepts and best-practices for iPhone development.



## iPhone Reference Library

Select from a range of technical documentation on iPhone development.



## Coding How-To's

Learn how to incorporate features of iPhone in your application.



## Sample Code

Use these samples to inspire development of your own great applications.



## Apple Developer Forums

Discuss iPhone development with other developers

Apple Push Notification Service Programming Guide

In App Purchase Programming Guide

Getting Started with In App Purchase

Game Kit Programming Guide

Cut, Copy and Paste

External Accessory Framework Reference

Map Kit Framework Reference

Media Player Framework Reference

Core Data Tutorial for iPhone OS

Develop



RSS

Videos

# iPhone Developer Center – Coding How-To's

**▼ Coding How-Tos**

Short descriptions of how to accomplish focused tasks.

---

Documents 7 of 781  ?

Title	Topic	Framework	Date
▶ <a href="#">Data Management Coding How-To's</a>	Data Management		2009-05-27 Minor Change
▶ <a href="#">User Experience Coding How-To's</a>	User Experience		2009-04-30 Minor Change
▶ <a href="#">Graphics &amp; Animation Coding How-To's</a>	Graphics & Animation		2009-04-30 Minor Change
▶ <a href="#">Audio &amp; Video Coding How-To's</a>	Audio & Video		2009-04-15 Minor Change
▶ <a href="#">Security Coding How-To's</a>	Security		2009-04-13 Minor Change
▶ <a href="#">Networking &amp; Internet Coding How-To's</a>	Networking & Internet		2009-04-13 Minor Change
▶ <a href="#">Tools Coding How-To's</a>	Development Environments		2009-04-12 Minor Change

# iPhone Developer Center – Sample Code

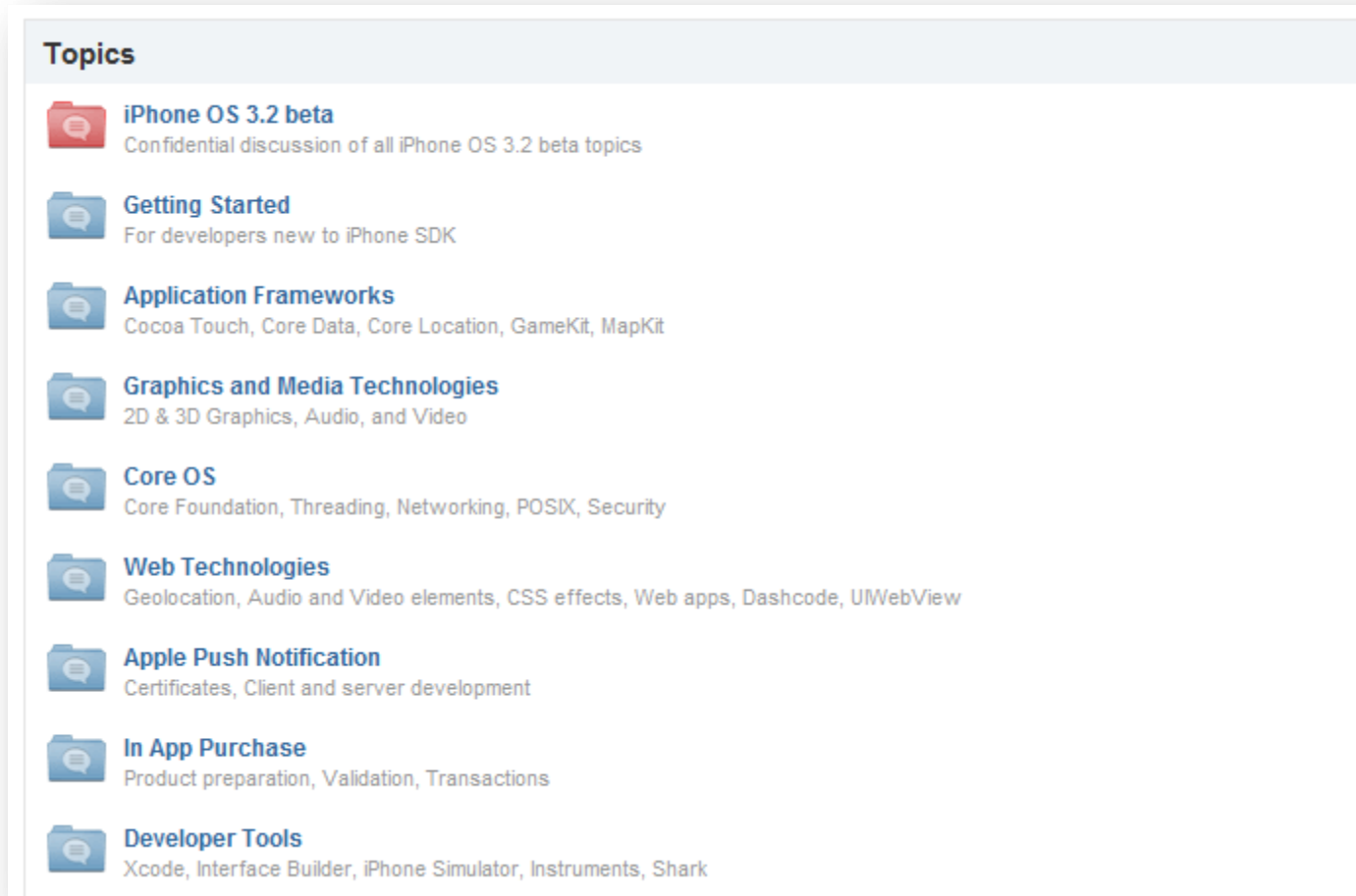
## ▼ Sample Code

Study sample code for implementation examples of Phone OS features. Each sample code project is build-able and executable source example of how to accomplish a task for a specific technology. They show the correct sequence of calls and parameter data types to provide a generalized method for API use that developers can modify for their specific needs.

Documents 86 of 781  ?

Title	Topic	Framework	Date
▶ <a href="#">MapCallouts</a>	User Experience	Map Kit	2010-03-19 First Version
▶ <a href="#">GKRocket</a>		Game Kit	2010-03-19 First Version
▶ <a href="#">avTouch</a>	Audio & Video Audio	AV Foundation	2010-03-15 Content Update
▶ <a href="#">Table Search</a>	User Experience Tables	UIKit	2010-03-11 Content Update
▶ <a href="#">AdvancedURLConnections</a>	Networking & Internet Protocol Streams	Foundation	2010-02-26 First Version
▶ <a href="#">iPhoneMixerEQGraphTest</a>	Audio & Video	Core Audio	2010-02-24 First Version

# iPhone Developer Center – Forum



The screenshot shows a list of forum topics under the heading "Topics". Each topic is represented by a folder icon with a speech bubble, followed by the topic name and a brief description.

- iPhone OS 3.2 beta**  
Confidential discussion of all iPhone OS 3.2 beta topics
- Getting Started**  
For developers new to iPhone SDK
- Application Frameworks**  
Cocoa Touch, Core Data, Core Location, GameKit, MapKit
- Graphics and Media Technologies**  
2D & 3D Graphics, Audio, and Video
- Core OS**  
Core Foundation, Threading, Networking, POSIX, Security
- Web Technologies**  
Geolocation, Audio and Video elements, CSS effects, Web apps, Dashcode, UIWebView
- Apple Push Notification**  
Certificates, Client and server development
- In App Purchase**  
Product preparation, Validation, Transactions
- Developer Tools**  
Xcode, Interface Builder, iPhone Simulator, Instruments, Shark

# Developer Communities

- Great way to ask questions and learn a new language
- Great way to help others who are starting out
  - This also helps to promote both yourself and the things that you are working on

# Online Tutorials

- <http://www.cocoalab.com/?q=becomeanxcoder>
- <http://blog.taragana.com/index.php/archive/top-10-tutorials-to-develop-iphone-apps/>

# Questions?

