

아이폰 SDK 프로그래밍

아이폰 SDK의 이해 및 설치

2008년 3월 드디어 아이폰 SDK가 발표됐다. 아이폰 SDK의 발표는 사용자들이 직접 자신의 폰에서 사용할 응용프로그램을 개발할 수 있다는 점에서 의미가 있다. 이런 취지를 살려 이 글에서는 아이폰 SDK를 처음 접하기 위해 필요한 내용들을 함께 살펴본다. 먼저 SDK를 설치하는 방법을 설명하고 SDK에 포함되어 있는 개발 툴에 대해서도 소개한다.

3

연 재 순 서

- 1회 | 2008. 7 | 아이폰 SDK의 이해 및 설치
- 2회 | 2008. 8 | Objective-C 기본 예제 분석
- 3회 | 2008. 9 | SDK 프로그래밍 기법
- 4회 | 2008. 10 | 실전 예제 프로그램 작성

강덕진 fromdj2k@gmail.com | 임베디드 시스템 소프트웨어 분야에서 근무하고 있고 MS 윈도우 커널과 WDM 등이 주 관심분야이다. IA32 기반의 운영체제를 개발해 수상한 바 있고, 이 운영체제를 발전시켜 언젠가 사람들을 위해 릴리즈한다는 목표를 갖고 있다. 현재 개인 홈페이지 fromdj.pe.kr을 운영하고 있다.

지난 2007년 6월 29일 미국에서는 아이폰(iPhone)이 정식으로 발매됐다. 당시 세계의 언론들은 길게 줄지어 서서 아이폰을 사려고 기다리는 미국인들을 신기한 눈으로 보도했다. 심지어 최초로 아이폰을 구입한 사람은 해외 토크의 주인공이 되기도 했다.

사실 아이폰은 발표 이전부터 많은 사람들이 궁금해 하던 '핫 아이콘'이었다. 디자인이 어떤 형태가 될 것이라는 이야기를 비롯해 어떤 기능이 있을 것이라는 수많은 추측과 소문이 난무했

다. 실제로 제품이 나왔을 때도 사람들의 이런 기대를 저버리지 않았다. MacOS를 아이폰으로 포팅하고, 거기에 애플의 인터넷 브라우저인 사파리를 이용해 거의 완벽한 웹 브라우징(풀 브라우징이라고 한다)을 지원한다. 뿐만 아니라 구글 어스와 유튜브 애플리케이션 등을 비롯해 마치 애플 매킨토시를 그대로 사용하는 듯한 느낌을 갖게 해줬다.

그 중에서도 사람들에게 가장 이목을 끈 기술은 역시 '멀티 터치'라는 기술이다. 기존의 터치스크린은 이미 오래전부터 사용되고 있었다. 대부분의 PDA 제품들은 물론이고 사용자 인터페이스를 갖는 많은 임베디드 시스템에 이미 터치스크린은 광범위하게 도입되었다. 하지만 아이폰에 장착된 터치스크린은 하나의 점만 인식한다는 점을 탈피했다. 사용자들은 멀티 터치를 이용한 조작, 즉 두 손가락으로 화면을 늘이거나 줄이는 새로운 형태의 손맛을 느낄 수 있게 되었다. 또한 실제 제품의 외관도 매우 단순했다(많은 사람들이 애플 제품의 성공요인이 단순함에 있다는 데 동의하고 있다. 물론 하얀색의 이어폰도 큰 역할을 했지만).

제품 전면의 거의 대부분은 커다란 LCD 화면이 차지하고 있고, 그 화면 아래에 버튼이 하나 있을 뿐이다. 일반적으로 전화기에 달려 있던 다이얼 버튼이 사라진 것이다. 사람들은 화면에 있는 버튼을 눌러 전화를 걸거나 문자를 주고 받았다. 사실 애플의



〈화면 1〉
최초의 아이폰 구매자

아이(i) 계열 시리즈의 시초는 아이팟(iPod)이다(아이팟을 모르는 독자는 아마 없을 것이다). 미국 뉴욕의 현대미술관에는 아이팟 1세대가 전시되어 있다고 한다. 하지만 아이팟이 최초의 MP3 플레이어도 아니고, 또한 최고의 기술을 갖춘 플레이어도 아니라는 것은 누구나 잘 알고 있다. 당시 수많은 MP3 플레이어 제품이 있었지만 아이팟이 선풍적인 인기를 끌 수 있었던 것은 (어디까지나 주관적인 생각이다) 기존의 틀을 한 단계 넘어서는 매력과 사용자의 요구를 교묘히 간파한 특성 때문일 것이다.



〈화면 2〉 처음 공개된 후부터 애플의 메인 페이지에 한 달 이상 자리 잡고 있던 SDK 로드맵

아이폰 3G의 등장

6월에 열린 WWDC2008에서 애플은 아이폰 3G를 공개했다. 그리고 좀더 많은 나라에서 아이폰을 쓸 수 있게 되었다고 발표했는데, 아쉽게도 여기에서 한국은 빠져 있었다. 필자는 빠른 시일 내에 한국에도 아이폰이 정식으로 서비스되기를 기대한다. 하지만 다행스럽게 한글 키보드가 정식으로 지원된다는 반가운 소식이 있었다.

새로 발표된 아이폰의 큰 특징이라면 먼저 3G를 지원함으로써 전송속도가 개선된 점과 MS Exchange를 지원함으로써 기업 환경과 통합할 수 있어서 회사 업무용으로 활용할 수 있게 된 점을 들 수 있다. 또한 MobileMe를 통해 개인이 사용하는 모든 컴퓨터에서 메일, 연락처, 사진 등을 하나로 동기화해 사용할 수 있는 시스템을 추가한 점도 주요한 특징이다.

2008년 3월 SDK 발표

2008년 봄 드디어 아이폰 SDK가 발표되었고 이와 관련된 내용이 〈화면 2〉와 같이 애플 홈페이지의 첫 화면을 꽤 오랜 기간 당당히 차지하고 있었다. 이것이 의미하는 바는 분명해 보인다. 아이폰 SDK의 발표는 사용자들이 직접 자신의 폰에서 사용할 응용프로그램을 개발할 수 있다는 의미가 된다. 즉 누구나 자신

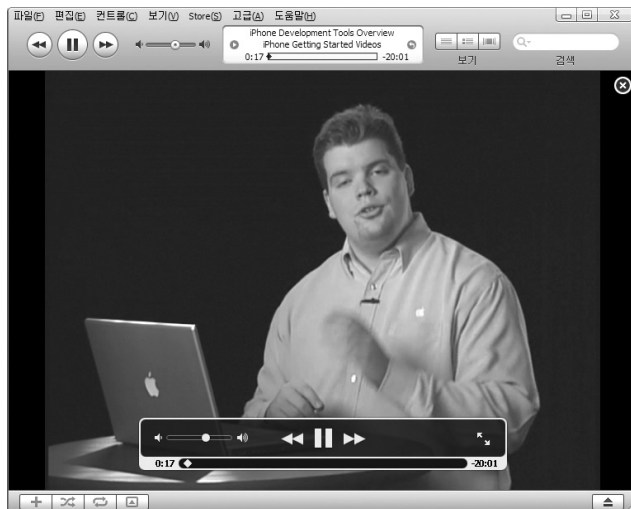
의 필요에 맞는 프로그램을 만들어서 자신의 폰에서 사용하거나 다른 사람과 공유할 수 있고, 뿐만 아니라 돈을 받고 판매할 수도 있게 된 것이다.

사실 요즘 휴대폰에는 아이폰만큼의, 또는 그 이상의 기능을 가진 것들이 넘쳐난다. 기능적인 면에서 아이폰은 절대 최고는 아닐 것이다. 하지만 얼마나 첨단 기술이 들어가 있는지와 얼마나 많은 기능을 집어넣었는가는 그다지 중요한 문제가 아니다. 대신 얼마나 사용자에게 어필하고 편리함을 주느냐가 가장 중요할 것이다.

아울러 이 글을 읽는 개발자들에게는 이러한 프로그래머블(Programmable)함이 곧 큰 축복이다. 필자도 아이폰 SDK가 발표되자마자 거금(?)을 들여 아이폰을 구입했고 그것을 가지고 이것저것 뜯어보려는 게 요즘의 일과다. 필자는 앞으로의 연재를 통해 아이폰 SDK를 써서 자신만의 애플리케이션을 개발해 보는 방법과 이와 관련된 여러 아이디어를 함께 공유해 보고자 한다.

SDK 소개

먼저 SDK가 어떻게 구성되어 있는지 알아보도록 하자. 먼저 애플의 아이폰 개발 홈페이지인 <http://developer.apple.com/iphone/index.action>에서 SDK에 대한 귀중한 자료들을 다수 얻을 수 있다. 사실 이 사이트 말고는 마땅히 자료를 구할 곳이 많지 않다. 사용자 등록을 하지 않으면 자료에 접근할 수 없기 때문에 우선 회원가입을 하도록 하자. 이 페이지에는 많은 문서 자료 외에 〈화면 3〉과 같이 아이튠즈(iTunes)와 연계된 동영상 강의도 제공되고 있다. 그리고 현재도 계속 새로운 문서들이 추가되고 있으므로 꾸준한 관심을 가지고 살펴보길 바란다.



〈화면 3〉 동영상 강의도 들을 수 있다

그럼 아이폰을 개발하기 위해 설치해야 할 SDK를 살펴보자. SDK는 <표 1>과 같은 구성요소들로 이루어져 있다.

개발 툴	Xcode	기본 개발 툴. 코드의 작성 빌드 및 실행을 지원한다. 디버깅도 Xcode를 이용해서 진행한다.
	Instruments	애플리케이션을 프로파일링하거나 분석할 수 있다. 메모리 사용량, 네트워크 사용량 등을 모니터링 가능하게 한다.
	Dashcode	웹 애플리케이션을 위한 개발 도구
	Interface Builder	사용자 인터페이스를 디자인할 수 있는 도구.
	iPhone 시뮬레이터	아이폰 장치 없이도 테스트해 볼 수 있는 시뮬레이터.
참고 문서	Shark	시스템 콜 단위까지도 프로파일링할 수 있는 도구.
	SDK 문서	도움말 및 관련 참고자료
	예제 코드	Xcode 프로젝트 예제 코드

<표 1> SDK의 구성 요소

우선 아이폰 SDK는 기존 애플 매킨토시 컴퓨터의 개발환경과 동일한 환경을 이용한다. 즉 <표 1>에서 설명한 Xcode 등의 모든 툴은 아이폰을 위한 애플리케이션은 물론이고, MacOS에서 실행되는 프로그램도 만들 수 있는 개발 툴이다. 아이폰 SDK를 다운로드해서 설치하면 Xcode와 같은 개발 툴이 설치되고, 거기에 관련 참고 문서들도 함께 설치된다. 그리고 아이폰 SDK를 위한 예제 코드들을 홈페이지에서 다운로드해 실행시켜 볼 수도 있다. 그럼 SDK를 설치하는 과정을 알아보자.

SDK 설치

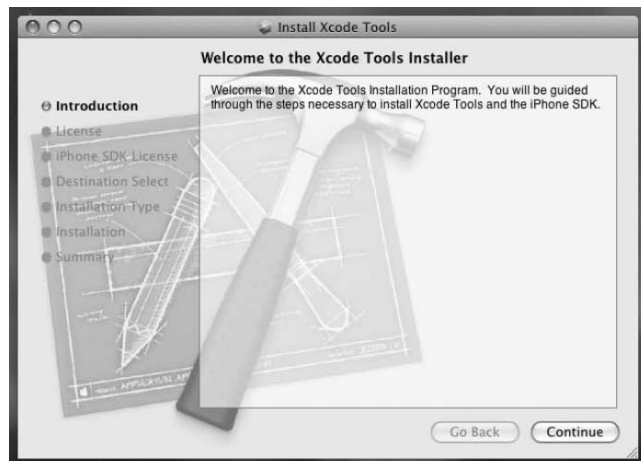
아이폰 SDK는 방금 전에도 언급했던 애플 개발자 홈페이지에서 다운로드할 수 있다. 우선 SDK를 다운로드하면 .dmg 확장자를 가진 파일이 다운로드될 것이다. 이 확장자는 애플 MacOS에서 디스크 이미지 형식으로 사용되는 확장자 명이다. 즉 우리에게 익숙한 .iso와 같이 디스크의 이미지를 저장할 때 사용되는 형식이다.

여기에서는 설치에 앞서 먼저 확인해야 할 사항이 있다. 약간 안타까운 소식이기도 한데, 아직까지 아이폰 SDK는 애플의 MacOS에서만 사용할 수 있다(그리고 앞으로도 그럴 것 같다). 즉 대부분의 사용자들이 주로 사용하는 윈도우 환경에서는 개발할 수 없다. 더군다나 운영체제인 MacOS X(이른바 '맥 오에스 텐')의 최신 버전인 10.5.3을 설치해야만 개발을 진행할 수 있다. 실제로 이 제한 조건은 국내 개발자들에게는 상당한 장벽이 될 것 같다. 사실 국내에서는 매킨토시 사용자가 많지 않으므로 이 글을 읽는 대부분의 독자들도 갑작스레 심각한 고민에 빠질 것이다. 하지만 뜻이 있는 곳에 길이 있고 자신이 원한다면 못할 것

이 없다고 믿는다.

이 글에서 필자는 독자들이 동작하는 MacOS X 환경을 갖고 있다고 가정한다. 다행스럽게도 한 가지 좋은 소식이 있는데 아이폰이나 아이팟 터치를 반드시 사야하는 것은 아니란 점이다. 별도의 시뮬레이터가 있으므로 아이폰이나 아이팟 터치가 없어도 충분히 테스트를 진행해 볼 수 있다.

MacOS X에서 다운로드한 dmg 파일을 더블 클릭하면 자동으로 디스크로 인식되고 별도의 창이 하나 뜰 것이다. 거기서 설치를 위한 응용프로그램을 실행하면 <화면 4>와 같이 설치가 진행될 것이다.



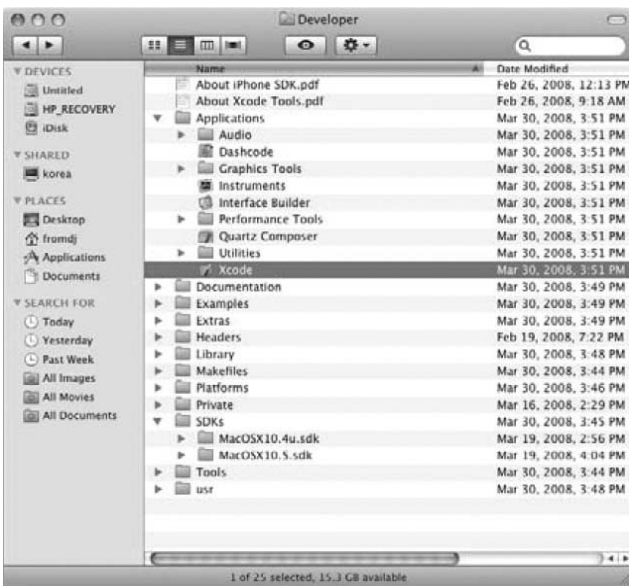
<화면 4> SDK의 설치 과정

설치는 의외로 간단하다. 기본 값으로 두고 Continue만 누르면 몇 분 안에 설치가 완료된다. 기본 설치 디렉토리는 /Developer/이다. 한편 MacOS X의 경우 우리에게 아주 익숙한 윈도우의 레지스트리 개념이 따로 없다. 그래서 프로그램을 제거하는 것도 매우 간단하다. 해당 디렉토리를 삭제하기만 하면 된다. 필자도 이 사실을 알고서는 어렵פות이 도스 시절이 떠올랐다.

일단 이것으로 설치가 매우 간단히 마무리되었다. 여기서 한 가지 확실히 해둘 것은 반드시 MacOS(Leopard) 10.5.3을 사용해야 한다는 것이다. 물론 그 이하 버전을 설치하더라도 SDK가 설치되고 프로그램들이 실행될 수도 있지만 아이폰 SDK 코드는 올바르게 동작하지 않을 것이다.

설치가 완료되면 <화면 5>와 같은 디렉토리 구조가 생성될 것이다. 여기서 간단히 몇 가지 중요한 디렉토리를 살펴보도록 하자. 먼저 최상위 디렉토리에는 SDK에 대한 간단한 소개와 Xcode 툴에 대한 문서가 들어 있으므로 한번 읽어두면 도움이 될 것이다. 그 다음의 디렉토리는 Applications로, 우리가 실제 사용할 개발 툴들이 모두 여기에 들어 있다. Xcode, Interface Builder, Dashcode, Instruments 등의 툴들이 여기에 있음을

기억하자. 다음은 개발에 필요한 도움말들이 들어 있는 Documentation 디렉토리이다. Xcode 개발 툴에는 도움말 시스템이 통합되어 있어서 간단하게 여기에 있는 문서들을 검색할 수 있게 한다. 그 다음은 Examples 디렉토리로 다양한 예제 프로그램들을 가지고 있는데, 아이폰 예제가 아니라 MacOS에서 동작하는 애플리케이션 예제를 포함한다. 그 다음으로 Headers 디렉토리를 볼 수 있는데, 소스에서 참조할 헤더 파일들이 이곳에 모여 있다. 그럼 지금부터는 주로 사용하게 될 개발 툴에 대해 살펴보자.

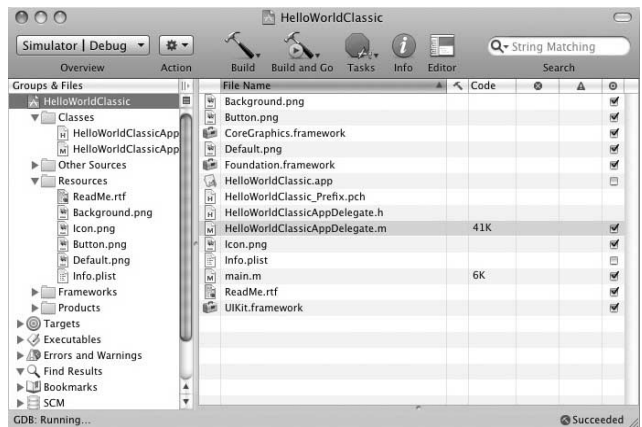


〈화면 5〉설치가 완료된 디렉토리

Xcode

아이폰 SDK의 기본 개발환경은 Xcode라는 프로그램을 이용한다. 모든 개발자들은 자신만의 IDE 개발 툴을 가지고 있다. 물론 Ultraedit와 같은 일반 에디터를 쓰는 독자도 있지만, 대부분의 경우 전용 툴을 이용할 것이다. 마이크로소프트 관련 개발자라면 비주얼 스튜디오 계열을, 자바 개발자라면 이클립스, 그리고 ARM 개발자라면 CodeWarrior 등을 많이 사용할 것이다.

이와 마찬가지로 애플 MacOS나 아이폰을 위한 프로그램을 개발할 경우에는 Xcode라는 툴을 주로 사용한다. Xcode는 코드의 편집은 물론이고 프로젝트도 관리할 수 있으며 프로젝트를 빌드하고 디버깅하는 것도 모두 하나의 프로그램에서 가능하다. 또한 도움말 문서도 통합되어 있어서 실제로 Xcode 하나만 있으면 개발을 진행할 수 있다. 그리고 실제 아이폰 장치를 가지고 있지 않더라도 SDK에 내장되어 있는 아이폰 시뮬레이터를 이용하면 실제 장비가 없어도 프로그램을 테스트해 볼 수 있다. 〈화면 6〉은 애플 홈페이지에서 제공하는 'Hello world' 프로젝트를



〈화면 6〉Xcode 기본 화면

Xcode에서 열어본 화면이다.

〈화면 6〉을 보면 프로젝트를 관리하는 윈도우를 볼 수 있다. 이 창의 툴바에는 큰 버튼들만 몇 개 있다. 이 가운데 맨 처음의 콤보 박스는 타겟을 설정하는 부분이다. 즉 이 프로젝트를 디버그 모드로 빌드할지 아니면 릴리즈 모드로 결정할지와 같은 빌드 옵션에 대한 항목을 선택하는 것이다. 오른쪽에 있는 항목은 프로젝트를 구성하고 있는 파일들을 보여주는데, 파일을 더블 클릭하면 편집할 수 있다. 그리고 툴바 중간에 큰 아이콘들이 있다. 각각 프로젝트를 빌드하기, 빌드하고 실행시키기 등을 위한 버튼이고 자주 사용하게 될 것이다. 여기까지의 과정을 통해 Xcode를 사용할 수 있는 준비를 마쳤고, 실제적인 애플리케이션을 만들 수 있는 준비를 마쳤다고 볼 수 있다. 하지만 개발함에 있어서 프로그램을 작성하고 빌드해 실행시키는 것 외에도 더 깊은 단계의 분석이 필요할 때가 있다. 이런 작업의 가장 기본적인 형태가 시스템 프로파일링이다.

프로파일링이란 프로그램이 실행될 때 그 행적을 조사해서 프로그램이 내부적으로 어떤 일을 하고 어떤 작업에서 가장 오랜 시간이 걸리는지를 분석할 수 있는 기능이다. 그리고 메모리 누수를 감지할 수 있는 분석도구나 CPU 성능을 모니터링할 수 있는 도구 등이 반드시 필요한 것도 추가로 고려할 수 있다. 특히 단지 취미용으로 프로그램을 작성하는 경우가 아니라 실제로 사용자에게 배포할 목적의 프로그램을 만든다면 최대한의 성능과 안정성을 확보하기 위해 이러한 분석 툴을 반드시 이용해야 한다.

아이폰 SDK에서 이런 목적으로 이용되는 툴이 바로 Instruments와 Shark이다. 특히 Instruments의 경우에는 그것이 실제 동작되는 모습을 보면 분석된 결과가 시각적으로 나타날 수 있다. 이 툴들을 이용하는 것은 여기서는 따로 설명하지 않고 다음 시간에 예제 프로그램을 분석하면서 함께 실행시켜 보

기로 한다. 사실 지금 설명하는 모든 개발 툴들은 언어나 플랫폼을 가리지 않고 적용되는 공통적인 것들이다. 예를 들어 필자가 리눅스에 익숙하다면 리눅스의 gcc, gdb tool-chain에 익숙할 것이다. 이 툴 체인에서도 시스템 프로파일링 디버깅 등을 모두 지원한다. 사실 이 기능들은 개발에 있어서 가장 기본이 되는 조건인 것이다. 이처럼 아이폰 SDK 역시 이런 기본적인 것들을 모두 제공하고 있다.

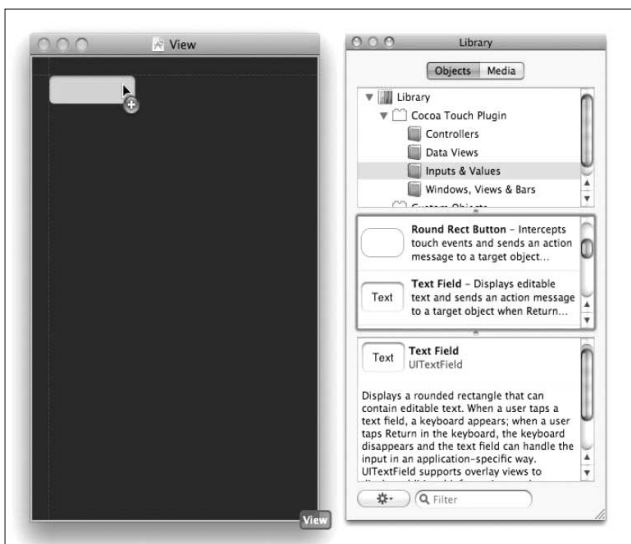
인터페이스 빌더

GUI 방식의 프로그램을 개발할 때 가장 시간이 많이 소요되는 부분은 바로 사용자 인터페이스를 디자인하고 구현하는 부분이다. 그래서 많은 개발 툴에서는 GUI를 빠른 시간에 디자인하고 그 결과를 바로 코드에 반영할 수 있도록 많은 노력을 기울이고 있다. Xcode의 개발 툴에도 인터페이스를 디자인하고 관련 코드를 자동화할 수 있는 '인터페이스 빌더'라는 툴이 포함되어 있다. 하나의 인터페이스 단위는 .nib(현재는 같은 내용을 텍스트 형식으로 저장한 파일인 .xib를 사용하도록 되어 있다. 아이폰의 경우에는 반드시 .xib만 이용해야 한다)라는 확장자로 저장된다.

〈화면 7〉은 CalcView.xib라는 인터페이스를 디자인하는 모습을 보여준다. 인터페이스 빌더는 UI를 구성할 요소들을 생성하고 배치하는 단순하면서도 복잡한 반복 작업을 최소화시켜준다. 그리고 이 툴을 통해 화면을 이루는 구성 요소들과 코드 상의 오브젝트들과의 연결 구조도 설정할 수 있다. 앞으로 이어질 연재를 통해 인터페이스 빌더를 실제로 이용해 보도록 하자.

아이폰 시뮬레이터 vs 실제 아이폰

조금 전에 언급한 바와 같이 아이폰 SDK를 이용해 프로그램



〈화면 7〉 인터페이스 빌더

을 개발한다고 해서 반드시 아이폰 장치가 필요한 것은 아니다. SDK에 아이폰 시뮬레이터가 내장되어 있으므로 기본적으로 테스트를 진행할 수 있기 때문이다. 하지만 아이폰 시뮬레이터는 실제 장치와 완전히 일치하지 않을 뿐 아니라 여러 가지 성능 분석, 즉 CPU 사용량 분석이나 네트워크 분석 등은 실제 장치를 이용해야 효율적으로 분석할 수 있음을 알아두자.

하지만 메모리나 객체 할당에 대한 분석은 시뮬레이터에서도 안정적으로 진행할 수 있다. 그리고 시뮬레이터에서는 테스트할 수 없는 기능들도 있으니 주의하길 바란다. 최신 SDK가 계속 릴리즈되면서 이러한 제약들이 사라지고 있으니 현재의 상태를 확인하기 위해서는 아이폰 시뮬레이션(iphone simulation) 가이드를 참고하자. 아울러 실제 아이폰을 가지고 있더라도 자신이 작성한 애플리케이션을 아무런 제약 없이 실행시켜 볼 수 있는 것은 아님을 기억하자. 애플 홈페이지에서 자세한 내용을 확인할 수 있겠지만, 개인 개발자라고 하더라도 개발자 등록을 거쳐 99달러를 지불해야만 실제 아이폰에서 테스트 및 배포를 할 수 있다.

Hello world!

모든 새로운 개발 환경이나 개발 언어에서 항상 맨 먼저 나오는 게 있다. 사실 이제는 좀 지겨울 때도 되었는데, 이른바 'hello world'가 그것이다. 사실 가장 좋은 방법이니 어쩔 수는 없다. 백문이 불여일견이고 시작이 반이라고 했다. 필자를 비롯한 많은 개발자들이 그렇지만, 일단은 뭔가 실행되는 과정을 한번 보는 것이 큰 도움이 되는 게 사실이다. 그래서 우리도 hello world를 한번 실행시켜 보도록 한다. 연재의 첫 회에서는 hello world까지 살펴보는 것으로 내용을 마치도록 하고, 2회부터는 심화된 예제를 통해 실제 프로그래밍을 수행하는 데 필요한 시스템 구성 요소들을 좀더 자세히 설명할 것이다(필자도 hello world를 보기까지 참으로 오랜 시간이 걸렸음을 여러분에게 고백한다. 따라서 여러분도 hello world를 한번 도전해 보길 바란다).

먼저 〈화면 8〉과 같이 애플 홈페이지의 예제 샘플 페이지에 방문해 보자. 여기에는 정말 많은 예제 프로그램들이 올라와 있으므로 시작하는 개발자들에게는 좋은 참고 자료가 된다. 예제를 선택해 들어가면 페이지에서 직접 소스 코드들을 살펴볼 수도 있고, 다운로드할 수도 있다. 일단 파일을 다운로드하고 폴더를 하나 만들어 압축을 해제한다. dmg로 다운로드한 경우에는 바로 드라이브처럼 인식되지만 별도의 디렉토리를 만들어 반드시 복사해야 한다(복사하지 않으면 readonly 속성이라 빌드가 진행되지 않음).

그럼 압축을 해제하면 나타나는 파일들 가운데 주목해 볼 만한



〈화면 8〉 애플 홈페이지에 있는 예제들

것들을 살펴보자. 먼저 HelloWorld.xcodeproj 디렉토리 안에 있는 project.pbxproj 파일이 프로젝트 파일이다. Xcode에서 이 프로젝트 파일을 열어 빌드하고 실행시키면 된다. Info.plist 파일은 애플리케이션의 이름이나 아이콘 정보 등을 담고 있는 속성 파일이다. HelloWorld.xib 파일은 인터페이스 빌더로 작성한 화면 디자인 정보를 담고 있는 파일이다. 그 외에 소스 코드는 .m이라는 확장자, 헤더 파일은 기존 c와 동일하게 .h로 되어 있다. 먼저 Xcode를 실행한 후에 project.pbxproj 파일을 열어 프로젝트를 열도록 하자. 그러면 〈화면 6〉과 같이 Xcode 프로젝트 창을 볼 수 있다.

이 글에서는 소스 코드까지 분석하지는 않겠지만 관심 있는 독자는 한번 .m 파일들을 살펴보도록 한다. 아마 다소 생소한 문법

들을 볼 수 있을 것이다. 프로젝트를 빌드하기 전에 프로젝트 창의 맨 오른쪽 상단에서 타깃을 설정해야 하는데, 아이폰 시뮬레이터를 사용해야 할 것이므로 Simulator|Debug를 선택하도록 하자. 그 후에 Build and Go 버튼을 눌러서 빌드하고 아이폰 시뮬레이터에서 실행시켜 보자. 아이폰 시뮬레이터가 실행된 후 Hello World 프로그램이 다시 실행되는 모습을 볼 수 있다.

새로운 자극, 아이폰

첫 시간에는 아이폰 SDK를 처음 접하기 위해 필요한 내용들을 함께 살펴봤다. 먼저 SDK를 설치하는 방법을 다뤘고, SDK에 포함되어 있는 개발 툴에 대해서도 간단히 설명했다. 그리고 마지막으로 Hello world를 실행해 보는 과정도 함께 소개했다.

필자는 많은 개발자들이 아이폰 SDK에 대해 큰 관심을 가지고 있으리라 믿는다. 그런 까닭에 이 글이 개발자들에게 얼마나 도움이 될 것인지 의문이 들고 걱정이 되기도 했다. 어느 수준의 독자를 대상으로 해야 하는지, 또한 어떤 내용들을 다루어야 많은 사람들이 만족할지 고민스러웠기 때문이다.

결국 누군가는 이 글을 재미있게 읽겠지만 누군가에는 너무 쉬운 내용일 수도 있다. 하지만 이 글이 뭔가 흥미롭고 재미있는 것을 찾는 프로그래머들에게 어떠한 형태의 자극이 되었다면 그보다 큰 만족은 없을 것 같다. 이어지는 글에서는 실제 프로그래밍을 하기 위해 필요한 기술들에 대해 하나둘 설명하기로 한다. +

참고자료

1. 애플 아이폰 개발 홈페이지 : <http://developer.apple.com/iphone>

micro
Software

1년 후에도 내용이 살아있는 잡지

전망 좋은 창가에서

전망 좋은 창가에 서서 밖을 내다본 적 있으세요?

창 저 너머의 탁 트인 세상을 보노라면 마음 한구석의
시름과 두려움이 사라져 어느새 용기마저 솟아나곤 합니다.

〈마소〉가 꿈꾸는 역할 또한 그렇습니다.
개발자와 IT 산업을 위한 전망 좋은 창이고자 합니다.

기술 트렌드를 따라가기가 버겁고
반복되는 업무에 지쳐간다면
〈마소〉의 곁에 조금 더 다가오세요.

마음이 한결 뿌듯해집니다.
도전하는 용기가 생깁니다.
미래가 그려질 것입니다.