
부록 가

C o c o a P r o g r a m m i n g f o r M A C O S X

Hello, iPhone!

최근에 국내에서 코코아 프로그래밍에 관심이 커진 이유는 맥 사용자 층이 늘어나 맥 개발의 핵심인 코코아 프로그래밍에 관심이 많아진 것도 있지만, 아이폰 때문이기도 하다. 그래서 특별히 원서에는 없는 내용이지만, 아이폰 개발 입문을 위한 짧은 장을 추가하기로 하였다. Xcode 3.1만 있으면 아이폰이 내 손에 없더라도 아이폰 시뮬레이터를 통해 바로 아이폰 프로그래밍을 해볼 수 있다. 일단 한번 따라해 보자.

Xcode 확인하기

만약 자신의 Xcode가 3.0 버전이라면 아이폰 SDK가 포함되지 않아서 아이폰 개발을 할 수 없다. 우선 Xcode의 버전을 확인하고 업데이트하자. 아이폰 SDK가 포함된 Xcode는 <http://developer.apple.com/iphone/>에서 다운로드하면 된다. 해당 사이트는 아이폰 Dev Center로, 앞으로 아이폰 개발과 관련하여 개발자 프로그램부터 애플리케이션 등록, 아이폰 개발 관련 문서 열람 등을 하러 자주 가게 될 것이다. 사과리에 북마킹해 두자.



그림 1 아이폰 Dev Center 첫 화면. 여기에서 Xcode 최신 버전을 다운받는다.

Hello, iPhone!

보통 프로그래밍 언어를 소개할 때 만드는 일반적인 ‘Hello, World!’ 출력 애플리케이션 대신, 우리는 아이폰에 ‘Hello, iPhone!’을 출력하는 애플리케이션을 만들 것이다. 우선 Xcode를 실행하여 File -> New Project로 새로운 프로젝트 생성 윈도우를 띄우자. Xcode 3.1을 사용하던 사용자라면 이전에도 몇 번 보았겠지만, Xcode 3.0에서 업그레이드한 사용자라면 iPhone OS라는 항목이 새로 생겨 있을 것이다. 기본 템플릿으로 애플리케이션 유형을 여섯 가지 제공하는데, 이를 이용하면 애플의 UI를 사용하는, 전형적인 아이폰 스타일의 애플리케이션을 쉽게 만들 수 있다. 우리는 5번째에 있는 View-Based Application 프로젝트 유형으로 새로운 프로젝트를 생성한다. 프로젝트 이름은 Hello라고 하자.



그림 2 새로운 뷰 기반 프로젝트 생성하기

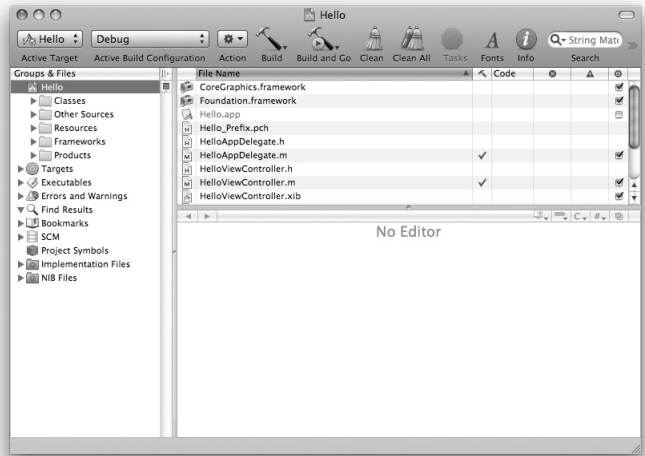


그림 3 생성한 프로젝트

처음 프로젝트 생성한 화면이 나타나면 무조건 Build and Go를 눌러서 한번 실행해 보자. 놀라지 말자. 그림 4와 같이 커다란 아이폰이 나타나면서 회색 바탕화면이 나타날 것이다. 이는 아이폰 애플리케이션 시뮬레이터로, 아이폰을 가로로

회전시킨다든가, 기본적으로 아이폰에 내장된 사파리, 주소록, 사진, 환경설정 등의 애플리케이션과 우리가 만든 애플리케이션이 여기에 내장된다. 실제 아이폰 혹은 아이팟 터치를 구매하기전까지는 이 아이폰 시뮬레이터에서 애플리케이션을 개발하면 된다.



그림 4 아이폰 시뮬레이터

인터페이스 빌더

인터페이스 빌더를 통하여 Hello, iPhone!을 뷰에 추가할 것이다. 코코아 프로그래밍에서 했던 것처럼 Xib(1) 파일을 였다. Xcode에서, 왼쪽의 Groups & Files에서 Resources 항목에 HelloViewController.xib을 더블 클릭한다.

(Xib - Xib는 Nib과 같이 뷰 객체들을 아카이빙한 것이다. 하지만 기존 Nib은 바이너리 형태로 SVN과 같은 버전 컨트롤 시스템에 적합하지 않아 이를 XML 형태로 변경한 것이 Xib이다. 소스코드를 컴파일할 때 이 Xib도 같이 컴파일되어 Nib으로 바뀐다. Xcode 3를 사용한다면 Nib대신 Xib을 사용하길 강력히 추천한다!)

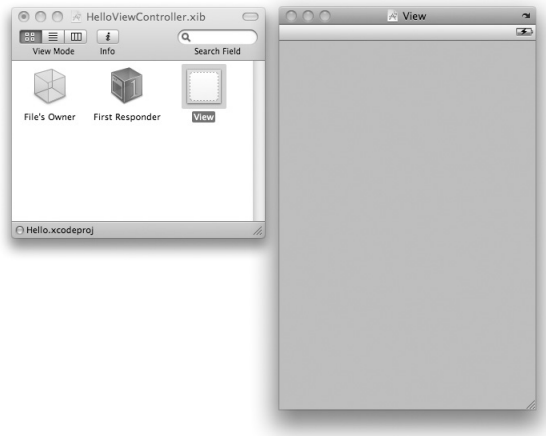


그림 5 HelloViewController.xib

HelloViewController.xib을 열었으면 독 윈도우에서 View를 더블 클릭하여 View 윈도우를 띄운다(그림 5). 해당 뷰에 이제 Hello, iPhone!을 표시할 UILabel 객체를 하나 추가한다.



그림 6 라이브러리 윈도우에서 UILabel 찾기

라이브러리 윈도우에서 UILabel을 검색하여 추가한다. UILabel을 더블 클릭하고 Hello, iPhone!을 입력하자. Hello, iPhone!이 한가운데 있으면 더 보기 좋을 것 같

다. 이를 정렬하기 위해 UILabel을 선택하고 인스펙터 윈도우를 열어서 Label size 탭으로 이동한다. 해당 탭에 맨 아래에 Placement 항목의 아이콘 두 개를 클릭하면 뷰에서 가로, 세로 정렬을 해준다.



그림 7 View 에 Hello, iPhone!을 표시하기

이제 빌드하고 실행해 본다. 그림 8과 같은 화면이 나오면 성공이다.



그림 8 Hello, iPhone!을 출력하기

여기서 이제 커맨드 키를 누른 채 오른쪽 화살표나 왼쪽 화살표를 누르면 아이폰 시뮬레이터가 회전을 한다. 이렇게 회전할 때 Hello, iPhone! 문자열이 같이 회

전하면 조금 더 값진 예제가 될 것이다. 이제 이 작업을 해보자.

가속도계

아이폰이나 아이팟 터치에 사파리를 이용해 본 분들은 이미 알겠지만 사파리에서 세로로 보다가 기기를 왼쪽이나 오른쪽으로 돌리면 가로로 브라우징 화면이 전환된다(그림 9, 그림 10).



그림 9 세로로 사용 중인 사파리



그림 10 가로로 전환하면 이렇게 된다.

이런 기능은 바로 아이폰에 내장된 가속도계(accelerometer)를 이용하여 현재 아이폰이 어느 위치로 돌아갔는지를 판단하여 UI에서 반응을 하는 것이다. 게임

같이 가속도계 자체를 실제 입력 장치로 사용하지 않는 한 우리가 가속도계 값을 직접 사용하는 경우는 드물다. 일반적인 뷰들은 화면이 세로에서 가로로 변환해도 되는지를 묻는 메시지를 받는데, 다음과 같은 메서드를 통해 뷰를 가로로 전환할지 여부를 결정한다. `HelloViewController.m`에 다음 코드를 찾아보자.

```
- (BOOL)shouldAutorotateToInterfaceOrientation:
    (UIInterfaceOrientation)interfaceOrientation
{
    // Return YES for supported orientations
    return (interfaceOrientation == UIInterfaceOrientationPortrait);
}
```

해당 코드의 `return` 부분을 `return YES`로 바꾸고 애플리케이션을 실행하여 아이폰 시뮬레이터를 회전시켜 보자. 그림 11과 같은 결과가 나온다면 제대로 동작하는 것이다.



그림 11 Hello, iPhone!이 가로로 표시되도록 하기

자, 완성되었다. 물론 굉장히 간단한 내용이지만, 아이폰 프로그래밍 맛보기라고 하기엔 충분할 것 같다. 지금까지 코코아 프로그래밍을 열심히 공부했으므로 아이폰에서 개발하는 데 큰 어려움이 없으리라 확신한다. 다음 장에서는 맥과 아이폰에서 코코아가 어떻게 다른지를 알아보겠다.

부록 나

C o c o a P r o g r a m m i n g f o r M A C O S X

맥 개발자, 아이폰 입문하기

우리는 「부록 가」에서 간단한 아이폰 애플리케이션 예제를 작성해 보았다. 사실 나(웁긴이)는 여기에 간단한 예제를 하나 더 추가해야 할지, 아니면 아이폰 개발에 관한 다른 사실을 알려주는 게 좋을지 꽤 많은 시간을 고민했다. 물고기를 주는 것 보다는 물고기 잡는 법을 알려 주는게 좋겠다는 생각에서 코코아 프로그래밍을 활용할 수 있도록 맥과 아이폰의 차이점에 대해서 간략하게 설명하고자 한다. 하나씩 아이폰 환경에 대해서 알아보자.

시작하기에 앞서… 어디부터 시작해야 할까?

일단 코코아에 익숙하지 않은 분이라면 Objective-C와 코코아부터 공부하길 꼭 권하고 싶다. 애플의 여러 가이드 문서를 살펴보아도 필히 Objective-C와 코코아 문서를 보라고 권장한다. 물론 아이폰을 임베디드라는 플랫폼으로 볼 수도 있지만, 일반적인 윈도 모바일이나 리눅스 임베디드와는 완전히 다르다고 말할 수 있다.

Objective-C와 코코아에 대해 낯설어서 왜 이렇게 동작을 하는지가 잘 이해되지 않아 쉽게 해결 가능한 문제임에도 한참 헤메는 경우를 주위에서 종종 보았다. 이 책을 읽는 독자들이라면 여기까지 잘 따라 왔을 테니 애플의 문서들을 찬찬히 훑어보며 공부하면 될 것이다. 코코아에 익숙하고 맥 애플리케이션 개발에 익숙하다면 특별한 도움 없이도 아이폰 개발을 자연스럽게 익힐 수 있을 것이다.

물론! 모바일 기기라는 여러 하드웨어적인 제한이 있어서 어려운 부분도 있겠지

만 아이폰 프로그래밍을 해보면 어떻게 맥에서 사용하던 것들을 이렇게 그대로 옮겨 놓을 수 있었을까 하며 감탄하게 될 것이다. 빨리 아이폰 애플리케이션을 개발하고 싶다고 급하게 아이폰 관련 서적부터 찾지 말고 꼭! Objective-C와 코코아를 먼저 공부하자.

관련 문서는 <http://developer.apple.com/iphone>에서 회원 가입을 하고 아이폰 개발자 포털에 접속하면 아이폰 입문에 관련된 애플의 문서들을 여럿 볼 수 있다. 지금까지 나온 많은 책을 보고, 또한 OSXDev에서 아이폰 스터디를 같이 진행하며 느낀 점인데, 애플의 문서가 지루하고 딱딱하기는 하지만 아직까지 아이폰에 관해서는 애플의 문서만큼 좋은 책은 없다. 간략한 비디오도 제공하니 한번씩 보자.

아이폰 OS

아이폰의 OS는 Mac OS X를 기반으로 만들어진 운영체제다. 애플의 XNU 커널을 이용하고 Mac OS X와 마찬가지로 BSD 기반으로 이루어진 나름 작은 유닉스라고도 할 수 있다. 국내에 앱스토어(아이폰과 아이팟 터치를 위한 애플리케이션 스토어)가 생기기 이전에는 국내 사용자들은 소위 말하는 해킹된 아이팟 터치를 이용하여 여러 애플리케이션을 직접 설치하여 사용하였는데, 아파치 같은 웹서버도 빌드되어 실행이 되었다.

아이폰 OS는 Mac OS X와 비슷한 구조로 이루어져 있다. 애플이 내세우는 코어 시리즈들로 구성된 기반인데, 코어 OS 위에 코어 서비스, 미디어, 제일 상위에 아이폰을 위해 다시 작성된 코코아 터치가 올라가 있다. 그림 1은 기존의 Mac OS X의 모습이고 그림 2는 아이폰의 모습이다. 비교해 보자.

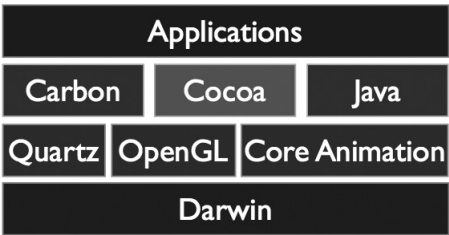


그림 1 Mac OS X 구성도

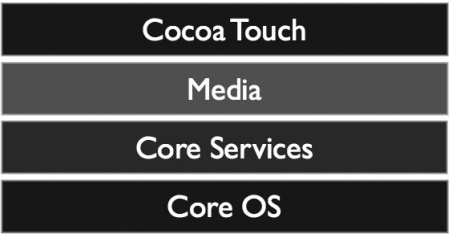


그림 2 아이폰 OS 구성도

코어 OS와 코어 서비스 부분은 기본적인 시스템 API 등을 제공한다. 앞에서도 말했듯이 아이폰 OS는 Mac OS X와 같이 유닉스 기반이다. 따라서 POSIX 표준을 준수하는 API들을 제공한다. 네트워크 소켓에 관련된 부분이라든가 스레드 같은 부분은 POSIX 표준으로 바로 작성할 수 있다. 꼭 기존 맥 개발자가 아니더라도 유닉스 개발 환경에 익숙하다면 아이폰 개발 환경에도 어느 정도 쉽게 익숙해질 것이다.

다음으로 미디어가 있는데, 이 부분에는 C 언어와 Objective-C 언어를 섞어서 사용하는 프레임워크들이 제공된다. 막강한 3D 기능을 사용할 수 있는 OpenGL ES 부터, 이를 코코아에서 편하게 쓸 수 있게 도와 주는 코어 애니메이션 프레임워크, 그리고 소리 재생과 관련된 코어 오디오 프레임워크 등이 지원된다. 아이폰은 또 다른 게임 플랫폼으로도 떠오르고 있는데, 바로 이렇게 OpenGL과 OpenAL, 그리고 코코아에서 쉽게 사용 가능한 코어 애니메이션 등의 지원 덕분이다.

마지막으로 볼 것이 바로 우리가 가장 관심을 가져야 할 부분인 코코아 터치 프레임워크다. 코코아 터치 프레임워크는 우리가 지금까지 공부한 코코아와 거의 비슷한 구조다. 파운데이션 프레임워크와 UI킷 프레임워크로 구성되어 있다. 앞에서 우리는 UI를 사용하기 위해 앱킷을 이용하였는데, 이 부분이 아이폰 OS에서는 UI킷으로 변경되고, 몇 가지 기능을 축소하고 아이폰에 필요한 기능들을 추가하여 코코아 터치라는 이름으로 다시 태어났다.

파운데이션 프레임워크

기존 코코아에서 사용한 파운데이션 프레임워크에서 아이폰 OS에 맞게 제작되며 빠진 내용을 요약하면 다음과 같다.

- 아이폰 OS에서는 코코아 바인딩을 사용할 수 없다.
- 가비지 컬렉터를 사용할 수 없다.
- 애플스크립트는 사용할 수 없다.
- 취소하기 관리자를 사용할 수 없다.
- XML 파싱을 위한 NSXMLParser 클래스는 제공하지만 이 외의 다른 XML 관련 클래스들을 제공하지 않는다.

코코아에서는 정말 유용하게 사용했던 바인딩과 가비지 컬렉터가 빠지기 때문에 입력해야 할 코드 양이 늘어난다. 하지만 Objective-C 2.0의 새로운 문법들을 지원하므로, 프로퍼티 등을 잘 활용하면 도움이 될 것이다.

이 외에는 기존 파운데이션 프레임워크와 거의 동일하게 동작하기 때문에 부담 없이 아이폰 애플리케이션 개발에 발을 담글 수 있을 것이다.

앱킷 vs UI킷

우리는 지금까지 코코아 애플리케이션을 개발하면서 앱킷을 통해 UI를 구현하였다. 아이폰에서는 이에 대응하는 것이 바로 UI킷이다. 우선 그림 3을 보며 아이폰 애플리케이션의 동작 흐름에 대해 살펴보자.

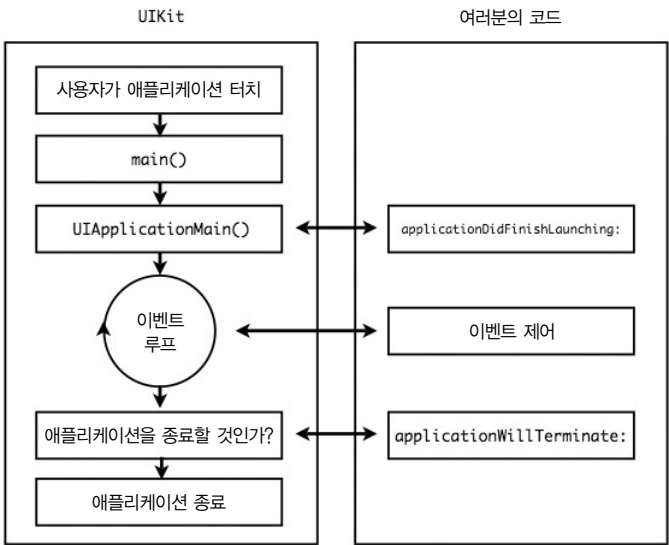


그림 3 아이폰 애플리케이션의 동작 흐름

사용자가 애플리케이션 아이콘을 터치하여 애플리케이션이 실행되면 여느 C 프로그램과 마찬가지로 main() 함수가 호출이 된다. 그리고 여기서 코코아 애플리케이션과 똑같이 UIApplicationMain 함수를 통해 런루프(RunLoop)가 돌아가고, 이 런루프를 통해 이벤트를 주고 받으면서 애플리케이션이 동작된다. 당연한 이야기처럼 들리겠지만, 아이폰에서는 윈도우를 하나만 띄울 수 있다. 그래서 기존의 앱킷과는 약간 다르게 클래스 상속을 한다. 코코아에서 NSWindow NSViwe와 NSObject->NSResponder를 통해 상속하지만 코코아 터치에서는 NSObject->UIResponder->UIView->UIWindow라는 특이한 상속을 한다. 아이폰에서는 어차피 화면에 하나의 뷰 밖에 띄울 수 없다는 제한점이 있어서 UIView에서 모든 이벤트를 받아서 처리한다는 개념을 가지고 있다. 잠깐! Hello, iPhone! 예제에서 이미 눈치 채신 분들도 있겠지만 코코아 터치에서는 NS대신 UI라는 접두사를 사용하여 클래스들을 만들었다. 맥에서 사용하는 코코아와 아이폰에서 사용하는 코코아 터치를 구분하려는 것 정도로 받아들이면 되겠다.

윈도

그렇다면 한 화면에 윈도가 하나 밖에 없는데, 하나의 애플리케이션 안에 여러 윈도를 띄우고 싶다면 어떻게 뷰 전환을 해야할까? 애플은 왼쪽, 오른쪽으로 슬라이딩하며 이동하는 UI를 방법으로 제시하였다. 이것은 내비게이션 컨트롤러를 이용하여 구현할 수 있다. 간단하게, 기본 내장된 주소록 애플리케이션을 보면, 처음에 실행할 때 전체 목록이 화면에 나타난다. 여기서 사용자 한 명을 선택하면 기존 화면이 왼쪽으로 미끄러지면서 오른쪽에서는 선택한 사용자의 상세 내용이 담긴 화면이 나타난다. 이렇게 여러 뷰를 미리 만들어 두고 각각 상황에 맞는 뷰를 전환하며 사용자에게 보여 준다. 물론 이렇게 미끄러지는 모습은 애플에서 기본으로 제공하는 애니메이션 효과다. 코코아 프로그래밍에서도 뷰 전환을 위해 코어 애니메이션을 이용하는 예제가 있었다. 아이폰 개발에서도 마찬가지로 코어 애니메이션을 통하여 이렇게 여러 뷰를 이동할 때 원하는 효과를 준다. 그 좋은 예로, 사파리 웹 브라우저의 세로 브라우징에서 가로 브라우징으로 전환할 때도 뷰 전환 애니메이션을 이용한다.

테이블 뷰

앱스토어를 이용해 보신 분은 알겠지만, 앱스토어의 게임을 제외한 애플리케이션

의 99%가 테이블 뷰로 이루어진 애플리케이션이라고 해도 될 만큼 아이폰에서 테이블 뷰는 중요하다. 사용자에게 데이터를 열거하여 보여줄때 아이폰 화면처럼 작은 공간에서 효과적으로 표시할 방법은 테이블 뷰밖에 없기 때문이다. 앞에서 소개했던 내비게이션 컨트롤러와 여러 테이블 뷰의 조합만으로도 멋진 애플리케이션이 탄생할 수 있다. 코코아 프로그래밍에서 테이블 뷰를 공부할 때 대충 보았다면 코코아 터치를 공부할 때는 정말 꼼꼼히 보자.

이벤트 제어

아이폰은 입력 장치가 터치, 게다가 멀티 터치를 지원하는 특이한 입력 장치를 가지고 있다. 물론, 터치 키보드라는 키보드 입력 장치도 있긴 하지만, 이는 단순히 텍스트 필드에 값을 입력하는 도구라고만 생각하자. 코코아 프로그래밍에서는 마우스 버튼 누름, 드래그 등 마우스에 관한 이벤트만 해도 여러 가지가 있었다. 반면 아이폰에서는 터치 시작, 터치 이동, 터치 끝이라는 세 가지의 이벤트로 이루어져 있다. 만약에 사용자가 한꺼번에 손가락 5개로 화면을 누르면 시작 이벤트가 주르륵 날아온다. 어찌 보면 단순하지만 또 처리하기 굉장히 난감할 수도 있어서 이러한 터치 이벤트를 직접 대기열을 만들어서 쌓는 경우도 있는데, 이런 일은 없어야 한다.

화면 그리기

코코아 프로그래밍과 마찬가지로 코코아 터치에서도 퀴즈, 코어 그래픽스, 코어 이미지, 코어 애니메이션 등을 이용할 수 있다. 물론 OpenGL도 지원한다. 이미 수많은 게임들이 이러한 프레임워크들을 이용하여 앱스토어에서 판매되고 있다. 자신만의 커스텀 UI를 만드는데, 단순히 UIView, UIWindow만을 서브클래스하여 만들기에 부족함이 있다면 애초에 메인 화면을 OpenGL 뷰 등으로 만들고 모든 뷰를 OpenGL로 직접 그려도 상관없다. 여기서 한 가지 조언하자면, 이렇게 완전히 커스터마이징한 UI로 애플리케이션을 만들어도 상관은 없으나 간혹 앱스토어에 등록할 때 애플에서 제시한 UI 가이드라인에 맞지 않는다며 등록되지 않는 경우도 있으니 참고하자.

마치며

간략하나마 코코아와 코코아 터치, 맥 애플리케이션 개발과 아이폰 애플리케이션 개발 환경의 차이점에 대해서 알아보았다. 이것 말고도 아이폰에서는 메모리 관리라든가, 전화가 왔을 때의 처리 등 신경 써야 할 부분이 몇 가지 더 있다. 하지만 이러한 내용들을 여기서 모두 다 소개하는건 원래의 책 목적에도 맞지 않을 뿐더러, 내용도 상당하기 때문에, 더 자세한 내용을 원하는 분들은 애플의 개발 문서를 참고하길 권한다. 다시 한번 강조하지만 아직까지는 애플 문서보다 좋은 책은 없다고 확신하다. 아이폰의 막막한 개발 장벽(Objective-C와 코코아)에 겁을 먹고 시작하지 못한 분들에게는 이번 코코아 프로그래밍 개정판과 이렇게 추가로 작성한 아이폰 관련 부록을 통해 조금이나마 도움이 되길 바란다.