1장 헬로, 안드로이드

여러분이 숙련된 모바일 엔지니어이든, 데스크톱 혹은 웹 개발자이든, 아니면 풋내기 프로그래머이든 간에, 안드로이드는 혁신적인 모바일 장치용 어플리케이션을 작성하기 위한 새로우면서도 흥미로운 기회를 선사한다.

안드로이드[1]는 그 이름에도 불구하고, 인류의 재앙에 빠진 지구를 잔인하게 척결하는 막을 수 없는 차가운 로봇 전사 군대를 만들어주지는 않는다. *안드로이드*는 휴대폰의 외관, 분위기, 기능을 모양 지을 수 있는 모바일 어플리케이션을 작성하기 위한 API 라이브러리 셋과 더불어, 운영체제, 미들웨어middleware, 그리고 핵심 어플리케이션을 포함하고 있는 오픈 소스 소프트웨어 스택이다.

작고 스타일리쉬 하면서 다양한 기능을 갖춘 최신 휴대폰은 이제 카메라, 미디어 플레이어, GPS 시스템, 터치 스크린을 결합한 강력한 도구가 되었다. 이렇게 기술이 발전됨에 따라 모바일 장치는 단순히 전화를 거는 것 그 이상이 되었지만, 그 안에 탑재된 소프트웨어와 개발 플랫폼은 보조를 맞추기에만 급급했다.

최근까지 휴대폰은 전용 개발 도구가 필요한 특정 회사의 운영체제 위에 구축된 매우 폐쇄적인 환경이었다. 이들 휴대폰은 본래부터 탑재되어 있는 네이티브 어플리케이션native applications을 서드파티third parties가 제작한 어플리케이션보다 우선시했다. 이는 점점 더 강력한 모바일 하드웨어 구축을 희망하는 개발자들에게 인위적인 장벽을 가져왔다.

안드로이드에서는 네이티브 어플리케이션과 서드파티 어플리케이션이 모두 같은 API를 사용해 작성되며, 동일한 런타임run time 위에서 실행된다. 이 API는 하드웨어 접근, 위치기반 서비스location-based services, 백그라운드 서비스background services, 지도기반 액티비티map-based activities, 관계형 데이터베이스relational databases, 장치간 피어-투-피어 메시징peer-to-peer messaging, 2D 및 3D 그래픽스를 주요 기능으로 갖는다.

이 책을 통해 여러분은 이들 API를 사용하여 여러분만의 안드로이드 어플리케이션을 만드는 방법을 배우게 될 것이다. 이번 장에서는 몇 가지 모바일 개발 지침을 살펴보고, 안드로이드 개발 플랫폼을 통해 사용할 수 있는 기능을 소개한다.

안드로이드는 강력한 API와 탁월한 문서 그리고 활발한 개발자 커뮤니티를 가지고 있으며, 개발 또는 배포 비용이 들지 않는다. 이러한 사실은 모바일 장치의 인기가 점차 높아짐에 따라, 여러분이 쌓아온 개발 경험이 무엇이든 관계없이 혁신적인 휴대폰 어플리케이션을 만들기 위한 흥미로운 기회가 된다.

---

[1] [역주] android [ændrɔid] n. 인조 인간

가볍게 살펴보는 배경지식

트위터Twitter [2]와 페이스북Facebook [3]이 없던 시절, 구글Google이 아직 설립자들의 눈에서 빛나고 있고 공룡이 지구를 거닐던 무렵의 휴대폰은 그저 서류가방에 넣을 수 있을만한 작은 크기에 몇 시간가량 지속될 수 있는 배터리를 가진 휴대형 전화기에 불과했다. 이 휴대형 전화기는 전화선을 물리적으로 연결하지 않고서도 전화를 걸 수 있는 자유를 주었다.

점점 더 작고 스타일리쉬 하면서 강력한 휴대폰은 이제 없어서는 안될 만큼 흔하다. 하드웨어의 발달은 더 많은 주변장치를 가지면서도 보다 작고 효율적인 휴대폰을 만들었다.

카메라와 미디어 플레이어를 시작으로, 휴대폰은 이제 GPS 시스템, 가속도 센서, 터치 스크린을 가지고 있다. 이러한 하드웨어 혁신은 휴대폰이 소프트웨어 개발을 위한 비옥한 토양임을 입증하는 반면, 휴대폰에서 사용할 수 있는 어플리케이션은 전반적으로 하드웨어에 비해 지체되어왔다.

그리 멀지 않은 과거에는

역사적으로 볼 때, 대개 하위 수준의 C나 C++로 코딩 하는 개발자는 그 코드가 실행될 특정 하드웨어를 이해할 필요가 있었는데, 이 하드웨어는 단일 장치이거나 한 제조사에서 나온 일련의 장치들일 수 있다. 이 닫힌 접근법은 하드웨어 기술이 발전해감에 따라 보조를 맞추기에 급급했다.

보다 최근에는, 개발자에게 좀더 넓은 대상 고객층을 제공하기 위해 심비안Symbian [4]같은 플랫폼이 개발됐다. 이들 시스템은 사용 가능한 하드웨어를 보다 더 잘 활용하는 풍부한 어플리케이션을 제공하도록 모바일 개발자를 독려한다는 점에서 보다 성공적임이 입증됐다.

이들 플랫폼은 장치 하드웨어에 대한 접근 몇 가지를 제공하지만, 복잡한 C/C++ 코드 작성과 사용하기 어렵기로 악명 높은 전용 API의 사용을 과도하게 요구한다. 이러한 어려움은 서로 다른 하드웨어 구현상에서 동작해야만 하는 어플리케이션 개발 시 더 커지며, 특히 GPS 같은 특정 하드웨어 기능을 위한 개발 시 그러하다.

최근 휴대폰 개발에서 이룬 가장 커다란 진척사항은 바로 자바Java를 사용하는 미들릿MIDlets의 도입이다. 미들릿은 자바 가상 머신Java virtual machine 위에서 실행되며, 하부에 있는 하드웨어를 추상화하고, 개발자로 하여금 자바 런타임을 지원하는 폭 넓은 다양한 하드웨어 위에서 동작하는 어플리케이션을 만들 수 있도록 해준다. 하지만 불행히도 이러한 편의성은 장치 하드웨어로의 제한된 접근을 대가로 얻어진다.

---

[2] [역주] http://www.twitter.com/ 미국의 댓글 형식 블로그.

[3] [역주] http://www.facebook.com/ 우리나라의 싸이월드와 비슷한 미국의 소셜 네트워크social network 웹사이트.

[4] [역주] 심비안 소프트웨어Symbian Software, Ltd.가 개발한 모바일 운영체제(심비안 소프트웨어는 2008년 노키아Nokia에 인수됐다). 보다 자세한 정보는 심비안 파운데이션Symbian Foundation 홈페이지(http://www.symbian.org/)에서 확인할 수 있다.

모바일 개발의 경우, 서드파티 어플리케이션이 폰 제조사가 작성한 네이티브 어플리케이션에 비해 상이한 하드웨어 접근과 실행 권한을 받는 것은 통상 있는 일이며, 미들릿은 이들보다도 더 적은 권한을 받는다.

자바 미들릿의 도입은 개발자의 대상 고객층을 확대했지만, 하위 수준의 하드웨어 접근과 샌드박스sandbox를 통해 보호되는 실행의 부재는, 대부분의 모바일 어플리케이션이 손 위에서 이뤄지는 핸드헬드 플랫폼handheld platform의 고유한 이동성을 이용한다기 보다는, 자그마한 화면에서 동작하도록 설계된 데스크톱 프로그램임을 의미했다.

안드로이드의 미래

안드로이드는 점점 더 강력한 모바일 하드웨어를 위해 설계된 모바일 운영체제의 새로운 물결과 나란히 앉아있다. 현재 윈도우 모바일Windows Mobile과 애플Apple의 아이폰iPhone은 모바일 어플리케이션을 위한 보다 풍부하면서도 간편한 개발 환경을 제공하고 있다. 하지만 안드로이드와는 달리, 이들은 네이티브 어플리케이션을 서드파티에서 만든 것보다 종종 더 우선시하는 전용 운영체제 위에 구축되어있으며, 어플리케이션과 네이티브 폰 데이터간의 통신을 제한한다. 안드로이드는 오픈 소스 리눅스 커널 위에 구축된 열린 개발 환경을 제공함으로써 모바일 어플리케이션을 위한 새로운 가능성을 제시한다. 일련의 API 라이브러리를 통해 모든 어플리케이션에서 하드웨어를 접근할 수 있으며, 어플리케이션 상호작용 역시 면밀한 제어 하에 완벽히 지원된다.

안드로이드에서는 모든 어플리케이션이 평등하다. 서드파티 안드로이드 어플리케이션과 네이티브 안드로이드 어플리케이션은 모두 같은 API를 사용해 작성되며, 동일한 런타임 위에서 실행된다. 사용자는 어떤 네이티브 어플리케이션이든 삭제 후 서드파티 개발자가 만든 것으로 대체할 수 있으며, 심지어 다이얼러dialer와 홈 스크린home screens 역시 이런 식으로 대체될 수 있다.

안드로이드에 대한 오해

이미 성숙해있는 진영에 찬물을 끼얹는 것과 같은 안드로이드의 등장으로 인해, 과연 안드로이드가 정확히 무엇인지에 관해 다소 혼란이 존재해왔던 이유를 쉽게 살펴볼 수 있다.

- **안드로이드는 자바 ME 구현이 아니다.** 안드로이드 어플리케이션은 자바 언어로 작성되지만, 자바 ME 가상 머신 내에서 동작하지는 않으며, 자바로 컴파일 된 클래스와 실행이미지가 있는 그대로 안드로이드에서 실행되지는 않을 것이다.

- **안드로이드는 리눅스 폰 표준화 포럼LiPS, Linux Phone Standards Forum [5]이나 오픈 모바일 얼라이언스OMA,**

[5] [역주] http://www.lipsforum.org/ 리눅스기반의 모바일 장치 대중화를 위해 설립된 국제 포럼. 2008년, 모바일 장치용 리눅스기반 운영체제 및 플랫폼 제공을 위한 컨소시엄인 리모 파운데이션LiMo Foundation (http://www.limofoundation.org/)에 통합됐다.

Open Mobile Alliance [6]의 일부가 아니다. 안드로이드는 오픈 소스 리눅스 커널 위에서 동작하며, 표준을 정의하는 이들 단체와 비슷한 목표를 갖지만, 모든 것을 갖춘 안드로이드의 소프트웨어 스택 방식은 이들 단체가 집중하고 있는 바를 훨씬 뛰어넘는다.

- **안드로이드는 (UIQ나 S60[7] 같은) 단순한 어플리케이션 계층이 아니다.** 안드로이드 역시 어플리케이션 계층을 포함하고는 있지만, 이와 더불어 하부에 놓인 운영체제와 API 라이브러리 그리고 어플리케이션 자체를 둘러싼 소프트웨어 스택 전체를 말한다.

- **안드로이드는 휴대폰이 아니다.** 안드로이드는 휴대폰 제조사를 위한 참조 설계reference design를 가지고는 있지만, 아이폰처럼 "안드로이드 폰"이라는 이름을 가진 단 하나의 휴대폰이 존재하는 것은 아니다. 안드로이드는 이렇게 하지 않고 다른 많은 하드웨어 장치를 지원하게끔 설계되어있다.

- **안드로이드는 아이폰을 겨냥한 구글의 대응책이 아니다.** 아이폰은 한 회사(애플)에서 발표한 완전히 독점적인 하드웨어 및 소프트웨어 플랫폼인 반면, 안드로이드는 오픈 핸드셋 얼라이언스OHA, Open Handset Alliance에서 제작하고 지원하는 오픈 소스 소프트웨어 스택으로서, 요구사항을 만족하기만 한다면 어떠한 휴대폰에서든 동작되도록 설계됐다. 구글의 이름을 단 안드로이드 폰과 관련된 많은 추측이 있는데, 설령 구글이 안드로이드 폰을 만든다손 치더라도 이는 그저 일개 한 회사가 만든 안드로이드 플랫폼의 하드웨어 구현일 뿐이다.

모바일 개발을 위한 오픈 플랫폼

구글은 안드로이드를 다음과 같이 설명하고 있다.

> *특정 회사에 종속되어 모바일 혁신을 더디게 만든 독점적인 장애물 없이 휴대폰을 실행하는 모든 소프트웨어와 모바일 장치를 위한 최초의 진정한 종합 오픈 플랫폼.*
>
> http://googleblog.blogspot.com/2007/11/wheres-my-gphone.html

안드로이드는 아래의 것들을 포함한 여러 가지 필수적인 부분과 종속적인 부분으로 구성된다.

- 모바일 장치가 이 소프트웨어 스택을 지원하기 위해서는 어떠한 기능이 요구되는지를 설명하는 하드웨어 참조 설계

- 하드웨어, 메모리 관리, 프로세스 제어와 관련하여 모바일 장치에 최적화된 하위 수준의 인터페이스를 제공하는 리눅스 운영체제 커널

---

[6] [역주] http://www.openmobilealliance.org/ 휴대폰 산업을 위한 열린 표준을 개발하는 표준화 단체.

[7] [역주] UIQ와 S60은 심비안 OS 기반의 오픈 소프트웨어 플랫폼이다. 이들에 대한 보다 자세한 정보는 UIQ 테크놀로지 홈페이지(http://www.uiq.com/)와 S60 플랫폼 홈페이지(http://www.s60.com/)에서 확인할 수 있다.

- 어플리케이션 개발을 위한 SQLite, WebKit, OpenGL, 미디어 관리자 등의 오픈 소스 라이브러리
- 안드로이드 어플리케이션을 실행하고 호스팅 하는데 사용되는 런타임. 이 런타임은 Dalvik 가상 머신과 특정 안드로이드 기능을 제공하는 코어 라이브러리를 포함하고 있으며, 모바일 장치에서의 사용을 위해 작고 효율적으로 동작하도록 설계됐다.
- 윈도우 관리자, 콘텐트 공급자, 위치 관리자, 전화, 피어-투-피어 서비스 등의 시스템 서비스를 어플리케이션 계층에 노출하는 어플리케이션 프레임워크
- 어플리케이션을 호스팅하고 띄우는데 사용되는 사용자 인터페이스 프레임워크
- 미리 설치되어 스택의 일부로 탑재된 어플리케이션
- 어플리케이션을 만드는데 사용되는 도구, 플러그 인, 문서를 가지고 있는 소프트웨어 개발 킷

폰이 시장에 출시되고 나면 상황이 달라지겠지만, 아직까지는 안드로이드 스택 전부가 오픈 소스로 공개된 것은 아니다. 여러분이 개발하는 안드로이드 어플리케이션 역시 꼭 오픈 소스일 필요는 없다.

진정으로 안드로이드를 주목하지 않을 수 없게끔 만드는 것은 바로 그가 가진 오픈 철학으로서, 이는 사용자 인터페이스나 네이티브 어플리케이션 설계에 있는 어떠한 결함도 확장이나 대체물 작성을 통해 수정될 수 있음을 보장한다. 안드로이드는 개발자로서 여러분이 상상하는 그대로의 모양과 분위기 그리고 기능을 찾는 휴대폰 인터페이스 및 어플리케이션을 만들 기회를 제공한다.

네이티브 안드로이드 어플리케이션

안드로이드 폰에는 보통 아래와 같은 어플리케이션이 미리 설치되지만, 여기에 국한되지는 않는다.

- G메일Gmail과 호환되는 이메일 클라이언트(G메일만 쓸 수 있는 것은 아님)
- SMS 관리 어플리케이션
- 구글의 온라인 서비스와 긴밀히 통합되는 달력과 연락처 목록을 가진 완전한 PIM(개인 정보 관리personal information management)
- 스트릿뷰StreetView, 기업 검색business finder, 운전 방향, 위성 사진, 교통 상태를 포함한 모든 것을 갖춘 모바일 구글 맵Google Maps 어플리케이션
- WebKit기반 웹 브라우저

- ❑ 인스턴트 메시징Instant Messaging 클라이언트
- ❑ 뮤직 플레이어와 픽쳐 뷰어
- ❑ 서드파티 어플리케이션을 다운로드하기 위한 안드로이드 마켓Android Marketplace 클라이언트
- ❑ DRM[8]이 해제된 음악을 구매하기 위한 아마존Amazon MP3 스토어

모든 네이티브 어플리케이션은 안드로이드 SDK를 사용해 자바로 작성되며 Dalvik 위에서 실행된다.

네이티브 어플리케이션에 의해 저장되고 사용되는 데이터(연락처 세부 정보 같은)는 서드파티 어플리케이션에서도 사용 가능하다. 이와 마찬가지로, 걸려오는 전화나 새로운 SMS 메시지 같은 이벤트 역시 여러분의 어플리케이션에서 처리할 수 있다.

새로운 안드로이드 폰에서 사용할 수 있는 어플리케이션의 정확한 구성은 하드웨어 제조사나 통신 사업자 혹은 대리점에 따라 다를 가능성이 있는데, 이는 통신 사업자가 출하되는 장치에 탑재되는 소프트웨어에 대해 중요한 영향력을 가지는 미국에서는 특히나 더욱 그렇다.

안드로이드 SDK의 특징

개발 환경으로서의 안드로이드가 가진 매력은 그가 제공하는 API에 있다.

안드로이드는 네이티브 어플리케이션과 서드파티 어플리케이션을 차별하지 않는 어플리케이션 중립적인 플랫폼으로서, 여러분에게 출하된 폰에 탑재되어 있는 네이티브 어플리케이션만큼이나 중요한 부분을 차지하는 어플리케이션을 만들 기회를 준다. 안드로이드의 특징 가운데 가장 주목할만한 것 몇 가지를 살펴보면 다음과 같다.

- ❑ 라이선스 사용료나 개발/배포 비용이 들지 않음
- ❑ Wi-Fi 하드웨어 접근
- ❑ 전화 또는 SMS 메시지를 주고받거나 모바일 네트워크를 통해 데이터를 송수신할 수 있게 해주는, 전화나 데이터 통신을 위한 GSM, EDGE, 3G 네트워크
- ❑ GPS 같은 위치기반 서비스를 위한 광범위한 API

---

[8] [역주] Digital Rights Management디지털 저작권 관리의 준말로, 디지털 콘텐츠의 무단 사용을 막아 콘텐츠 제공자의 권리와 이익을 보호해주는 기술 및 서비스를 통틀어 일컫는다.

- ❑ 카메라와 마이크를 사용한 재생 및 기록을 포함한 풍부한 멀티미디어 하드웨어 제어
- ❑ 가속도 센서와 나침반 하드웨어를 위한 API
- ❑ IPC 메시지 전달
- ❑ 공유 데이터 저장소
- ❑ WebKit을 기반으로 한 통합 오픈 소스 브라우저
- ❑ 맵 컨트롤을 유저 인터페이스의 한 부분으로 통합해 사용하는 어플리케이션을 위한 완벽한 지원
- ❑ 구글 톡Google Talk을 사용한 피어-투-피어P2P, peer-to-peer 지원
- ❑ 패스path기반 2D 그래픽스 라이브러리와 OpenGL ES를 사용한 3D 그래픽스 지원을 포함하는 모바일에 최적화된 하드웨어 가속 그래픽스
- ❑ 다양한 오디오/비디오나 정지 이미지 포맷을 재생하고 기록하기 위한 미디어 라이브러리
- ❑ 어플리케이션 컴포넌트의 재사용과 네이티브 어플리케이션의 대체를 장려하는 어플리케이션 프레임워크

카메라, GPS, 가속도 센서 등의 하드웨어 접근

안드로이드는 장치 하드웨어의 사용이 수반되는 개발을 단순화하는 API 라이브러리를 가지고 있다. 이는 여러분의 소프트웨어에 대해 서로 다른 장치를 위한 특정 구현을 만들 필요가 없도록 하여, 안드로이드 소프트웨어 스택을 지원하는 장치라면 어디에서는 기대한대로 동작하는 안드로이드 어플리케이션을 만들 수 있도록 보장한다.

안드로이드 SDK는 위치기반 하드웨어(GPS 같은), 카메라, 네트워크 연결, Wi-Fi, 블루투스Bluetooth, 가속도 센서, 터치 스크린, 전력 관리를 위한 API를 포함하고 있다. 안드로이드의 하드웨어 API가 가지는 몇 가지 가능성에 대해서는 10장에서 보다 자세히 살펴본다.

네이티브 구글 맵, 지오코딩, 위치기반 서비스

네이티브 맵 지원은 안드로이드 장치의 이동성을 극대화하는 다양한 맵기반 어플리케이션을 만들도록 해준다. 안드로이드는 사용자 인터페이스의 한 부분으로 인터랙티브한 구글 맵을 가지면서, 안드로이드의 풍부한 그래픽스 라이브러리를 사용해 맵을 프로그래밍적으로 제어하고 주석을 달 수 있도록 하는 기능을 갖는 액티비티를 만들도록 해준다.

안드로이드의 위치기반 서비스는 장치의 현재 위치를 측정하기 위해 GPS와 구글의 GSM 셀기반 위치 기술cell-based location technology 같은 기술들을 관리한다. 이들 서비스는 특정한 위치감지 기술에 종속적이지 않도록 추상화되어 있어서, 여러분으로 하여금 어느 한 특정 기술을 고르도록 하기 보다는 최소한의 요구사항(이를 테면, 정확도라든가 비용)을 지정하게끔 한다. 이는 휴대폰이 어떤 기술을 지원하는지에 관계없이, 여러분의 위치기반 어플리케이션이 동작할 것임을 의미하기도 한다.

안드로이드는 맵과 위치를 결합하기 위해, 특정 주소에 해당하는 맵 좌표와 맵상의 특정 위치에 해당하는 주소를 알아낼 수 있도록 해주는 순방향 및 역방향 지오코딩geocoding [9] API를 가지고 있다.

맵, 지오코더geocoder, 위치기반 서비스 사용에 관한 자세한 내용은 7장에서 배운다.

백그라운드 서비스

안드로이드는 화면에 보이지 않은 채 백그라운드에서 실행되도록 설계된 어플리케이션과 서비스를 지원한다.

요즘 휴대폰은 다양한 기능을 가지고 출시되지만, 폰의 제한된 화면 크기는 대게 한번에 하나의 인터랙티브한 어플리케이션만을 화면에 보일 수밖에 없다는 결과를 낳는다. 백그라운드 실행을 지원하지 않는 플랫폼은 여러분이 지속적으로 관심을 기울일 필요가 없는 어플리케이션의 생존력을 제한한다.

백그라운드 서비스는 사용자의 직접적인 관여 없이도 화면에 보이지 않은 채 자동 처리를 수행하는 어플리케이션 컴포넌트를 만들 수 있게 한다. 백그라운드 실행은 여러분의 어플리케이션이 이벤트 중심으로 동작할 수 있도록 하고 또 주기적인 업데이트를 지원할 수 있게끔 하는데, 이는 게임 점수나 시장 가격 모니터링, 위치기반 경보 발생, 혹은 걸려오는 전화와 SMS 메시지에 우선순위를 매기거나 미리 차단하는 데에 제격이다.

백그라운드 서비스를 최대한 활용하는 방법에 대한 보다 많은 내용은 8장에서 배운다.

데이터 저장과 검색을 위한 SQLite 데이터베이스

아담한 크기로 인해 저장 능력이 제한된 장치에는 빠르고 효율적인 데이터 저장과 검색이 필수다.

안드로이드는 SQLite를 사용해 각 어플리케이션을 위한 작고 가벼운 관계형 데이터베이스를 제공한다. 여러분의 어플리케이션은 관리되는 관계형 데이터베이스 엔진managed relational database engine을 사용해 데이터를

---

[9] [역주] 길거리 주소나 우편 번호 같은 지리 데이터로부터 그와 연관된 지리 좌표(종종 위도와 경도로 표현됨)를 찾는 과정을 가리켜 지오코딩이라 한다. 이때, 지리 데이터로부터 그와 연관된 지리 좌표를 찾는 것을 순방향 지오코딩forward geocoding이라 하고, 반대로 지리 좌표로부터 그와 연관된 지리 데이터를 찾는 것을 역방향 지오코딩reverse geocoding이라 한다.

안전하고 효율적으로 저장할 수 있다.

각 어플리케이션 데이터베이스에는 기본적으로 샌드박스가 적용되지만(즉, 데이터베이스에 있는 내용은 그 데이터베이스를 만든 어플리케이션만이 사용할 수 있다), 콘텐트 공급자는 이들 어플리케이션 데이터베이스의 관리되는 공유managed sharing를 위한 메커니즘을 제공한다.

데이터베이스, 콘텐트 공급자, 그리고 기타 안드로이드에서 사용 가능한 데이터 지속 옵션은 6장에서 자세히 다룬다.

공유 데이터와 어플리케이션간 통신

안드로이드는 여러분의 어플리케이션에서 정보를 다른 곳에 사용하기 위해 전송하는 세 가지 기법인 알림, 인텐트, 콘텐트 공급자를 가지고 있다.

*알림Notifications*은 전통적으로 모바일 장치가 사용자에게 경보를 발생시키는 표준적인 방법이다. 이 API를 사용하면 가청 경보를 울리고, 진동을 발생시키고, 장치의 LED를 번쩍이게 할 수 있을 뿐만 아니라, 상태바에 있는 알림 아이콘을 제어할 수 있는데 이에 대해서는 8장에서 살펴본다.

*인텐트Intents*는 어플리케이션 내에서의 메시지 전달과 어플리케이션간의 메시지 전달을 위한 메커니즘을 제공한다. 인텐트를 사용하면 원하는 액션(전화 걸기나 연락처 편집 같은)을 다른 어플리케이션이 처리하도록 시스템 전역에 방송할 수 있다. 인텐트는 안드로이드의 중요한 핵심 구성요소로서 5장에서 자세히 다룬다.

마지막으로, *콘텐트 공급자Content Providers*는 여러분의 어플리케이션이 가지는 전용 데이터베이스에 대한 관리되는 접근managed access을 제공하는 수단이다. 연락처 관리자Contact Manager 같은 네이티브 어플리케이션을 위한 데이터 저장소는 콘텐트 공급자로서 노출되므로, 여러분은 이들 데이터 저장소를 읽거나 수정하는 여러분만의 어플리케이션을 만들 수 있다. 6장에서는 네이티브 공급자에 대한 내용과 여러분만의 공급자를 만들고 사용하는 방법 설명을 포함하여, 콘텐트 공급자에 대해 자세히 다룬다.

구글 톡을 이용한 P2P 서비스

이전 버전의 SDK에 근거해볼 때, 이후 발표되는 버전에서는 안드로이드의 피어-투-피어(P2P) 통신 서비스를 사용해 다시 한번 여러분의 어플리케이션에서 다른 안드로이드 폰으로 구조화된 메시지를 전송할 수 있게 될 것으로 기대된다.

안드로이드 P2P 서비스는 XMPP(Extensible Messaging and Presence Protocol)의 특화된 버전을 사용한다. 이는 구글의 구글 톡 인스턴트 메시징 서비스에 기반해, 여러분의 장치와 기타 다른 안드로이드 폰간에 지연 시간latency이 짧고 반응 시간이 빠른 통신을 보장하는 영구적 소켓 연결을 생성한다.

이것이 사용 가능해지면, 여러분은 일반적인 인스턴트 메시징이나 별개의 장치에 있는 어플리케이션 인스턴스들간에 데이터를 전송하기 위한 인터페이스를 위해 구글 톡 서비스를 사용할 수 있게 될 것이다. 이는 실시간 멀티플레이어 게임이나 소셜 어플리케이션social applications 같이 여러 사용자가 참여하는 인터랙티브한 어플리케이션에 큰 재미를 더하게 된다.

P2P 서비스는 또한 지인(知人)이 온라인상태인지를 확인하는데 사용되는 현재 상태 알림presence notification을 제공한다. P2P 서비스는 그 자체만으로도 무척이나 매력적이지만, 안드로이드의 다른 기능하고도 매우 잘 동작한다. 친구들끼리 서로의 위치를 전송하는 백그라운드 서비스와 이에 대응하여 친구들의 위치를 보여주거나 친구들이 근처에 있을 때 알려주는 어플리케이션을 생각해보자.

안드로이드 1.0에서는 보안상의 문제로 인해 구글 톡을 이용한 데이터 메시지 전송이 불가능하다. 인스턴트 메시징 클라이언트는 사용 가능하며, 추후 발표되는 SDK에서는 XMPP와 호환되는 IM과 데이터 메시징이 사용 가능해질 것으로 기대된다.

광범위한 미디어 지원과 2D/3D 그래픽스

보다 큰 화면과 더 밝고 높은 해상도를 가진 디스플레이는 휴대폰이 멀티미디어 장치가 되는데 한 몫 했다. 안드로이드는 사용할 수 있는 하드웨어를 최대한 활용하기 위해, 2D 캔버스 드로잉을 위한 그래픽스 라이브러리와 OpenGL을 이용한 3D 그래픽스를 제공한다.

또한 안드로이드는 MPEG4, H.264, MP3, AAC, AMR, JPG, PNG, GIF 포맷 등의 정지 영상, 비디오, 오디오 파일 처리를 위한 광범위한 라이브러리를 제공한다.

2D와 3D 그래픽스는 11장에서 상세히 다루며, 안드로이드 미디어 관리 라이브러리는 10장에서 다룬다.

최적화된 메모리와 프로세스 관리

안드로이드의 프로세스와 메모리 관리는 조금 특이하다. 자바와 .NET처럼 안드로이드도 어플리케이션 메모리 관리를 위해 자신만의 독자적인 런타임과 가상 머신을 사용한다. 하지만 이들 프레임워크와는 달리 안드로이드는 프로세스 수명process lifetimes도 관리한다. 안드로이드는 보다 높은 우선순위를 갖는 어플리케이션을 위한 리소스 마련을 위해, 필요한 만큼 프로세스들을 중지하고 종료함으로써 어플리케이션이 좋은 반응성을 가지도록 보장한다.

이 상황에서 우선순위는 사용자가 현재 상호작용중인 어플리케이션에 따라 결정된다. 여러분의 어플리케이션이 갑작스런 종료에 대비하되 여전히 반응할 수 있는 상태로 남아 업데이트되거나 필요 시 백그라운드로 재시작 될 수 있게끔 보장하는 것은, 어플리케이션이 자신의 수명을 제어할 수 없는 환경에서는 중요한 고려사항이다.

안드로이드 어플리케이션의 수명 주기life cycle에 대해서는 3장에서 보다 자세히 배운다.

오픈 핸드셋 얼라이언스 소개

*오픈 핸드셋 얼라이언스(OHA)*는 하드웨어 제조사, 이동통신 사업자, 소프트웨어 개발사를 포함한 30여 개 이상의 기술 회사가 모인 연맹이다. 특히 탁월한 모바일 기술 회사인 모토로라Motorola, HTC, T-모바일T-Mobile, 퀄컴Qualcomm이 눈에 띈다. 이들은 OHA를 다음과 같이 표현한다.

> *개방에 대한 헌신, 미래를 위한 공동의 비전, 그 비전을 현실로 만들기 위한 구체적인 계획. 모바일 혁신의 촉진과 소비자에게 보다 풍부하고 덜 비싸며 더 나은 모바일 경험을 제공하기 위하여.*
>
> http://www.openhandsetalliance.com/oha_faq.html

OHA는 혁신적인 모바일 개발을 위해 소프트웨어 개발사나 휴대폰 제조사가 필요한 고품질의 플랫폼을 라이선스 비용 없이 발 빠르게 제공함으로써, 소비자에게 더 나은 모바일 소프트웨어 경험을 전하길 소망한다.

결국 모바일 플랫폼으로서의 안드로이드의 성공은, 안드로이드 폰이 널리 채택되도록 북돋을 매력적인 모바일 서비스와 휴대폰이 필요하다는 점에서 OHA 파트너사의 성공에 크게 좌우될 것이다. 그러는 동안 개발사는 더 많은 모바일 기술 회사가 OHA의 일원이 되도록 장려하는, 안드로이드를 위한 혁신적인 새로운 모바일 어플리케이션을 만들 기회를 가진다.

안드로이드의 발전 추이

최초의 안드로이드 폰인 T-모바일 G1은, 2008년 10월 미국과 같은 해 11월 영국에서 각각 출시됐다. 오픈 핸드셋 얼라이언스는 더 나아가 안드로이드를 지원하는 추가 휴대폰과 서비스를 2009년 초에 배포하기 위해 노력하고 있다.

단 하나의 하드웨어 구현을 위해 만들어진 모바일 OS와는 달리, 안드로이드는 터치 스크린 폰에서부터 화면이 없는 장치에 이르기까지 매우 다양한 하드웨어 플랫폼을 지원하도록 설계됐다.

이 밖에도 라이선스 비용이나 전용 소프트웨어가 존재하지 않는다는 점과 더불어, 휴대폰 제조사가 안드로이드와 호환되는 자사의 휴대폰을 공급하는데 드는 비용이 상대적으로 낮다. 일단 인기 있는 안드로이드 어플리케이션을 실행시킬 수 있는 하드웨어 요구가 충분한 양에 도달하게 되면, 더 많은 장치 제조사가 점점 더 그 요구를 만족하는 하드웨어를 생산할 것으로 기대된다.

안드로이드의 개발 이유

만일 모바일 어플리케이션을 개발해본 경험이 있다면, 굳이 말하지 않아도 아래와 같은 사실에 대해 잘 알고 있을 것이다.

- ❑ 굳이 안드로이드를 사용하지 않아도, 안드로이드로 할 수 있는 많은 것이 이미 가능하다.
- ❑ 하지만 기존의 방법은 힘들다.

안드로이드는 현대 모바일 장치의 실재에 기반한 모바일 프레임워크를 대표한다.

단순하면서도 강력한 SDK, 무료 라이선스, 탁월한 문서, 왕성한 개발자 커뮤니티와 함께, 안드로이드는 사람들이 휴대폰을 사용하는 방법과 이유를 변화시키는 소프트웨어를 만들기 위한 훌륭한 기회다.

안드로이드는 30여 개 이상의 OHA 멤버가 지지하며, 업계의 주요 소문으로 둘러싸여있다.

시장의 측면에서 볼 때, 많은 나라에서 휴대폰 소유 대수가 컴퓨터 소유 대수를 웃도는 것을 비롯해, 휴대 장치의 증가는 세계적인 현상이다. 모바일 광대역 및 Wi-Fi의 이용성 증가와 맞물린 *스마트폰*(전화를 비롯해, 카메라, 인터넷 접속, 미디어 플레이어, Wi-Fi, GPS 서비스를 채용하고 있는 다기능 장치)의 인기 상승은 고급 모바일 어플리케이션 시장 증대를 가져왔다.

안드로이드 도입을 주도하는 것

개발자가 소프트웨어를 보다 손쉽게 작성할 수 있도록 하는 것이 고객에게 더 나은 모바일 소프트웨어를 전달하는 방법이라 믿는 구글과 OHA의 생각대로, 안드로이드는 개발자를 주 대상으로 한다.

개발 플랫폼으로서의 안드로이드는 강력하고 직관적이며, 모바일 장치를 위해 프로그래밍해본 경험이 없는 개발자라도 유용한 어플리케이션을 빠르고 쉽게 만들도록 해준다. 안드로이드 어플리케이션이 이를 실행하기 위한 장치의 수요를 얼마나 혁신적으로 만들어낼 수 있는지는, 특히 개발자가 다른 플랫폼을 위한 어플리케이션을 작성할 수 없어서 안드로이드용 어플리케이션을 작성하는 경우 쉽게 확인해볼 수 있다.

소프트웨어 개발과 플랫폼 도입을 주도하는 것은 항상 하부에 놓인 시스템의 실제적인 부분에 대한 열린 접근이다. 인터넷의 타고난 개방성과 중립성은 인터넷이 그의 첫 10년 동안 수십억 달러의 기술 산업을 위한 플랫폼이 되는 것을 지켜봤다. 그 이전에는 개인용 컴퓨터의 폭발적인 증가와 컴퓨터 프로그래밍을 불가해한 것에서 메인스트림으로 옮겨놓는 것을 가능케 한 윈도우 운영체제의 강력한 API와 리눅스 같은 오픈 시스템이 그 역할을 감당했었다. 이러한 개방성과 힘은 마음만 있다면 누구든 최소한의 비용으로 상상의 나래를 펼칠 수 있도록 보장한다. 이는 휴대폰의 경우 지금껏 해당되지 않는 이야기였으며, 좋은 휴대폰 어플리케이션이 드문 이유이자 무료로 사용 가능한 것이 여전히 적은 이유이기도 하다.

회사 역시 안드로이드가 제공하는 제어 수준으로 인해 매료될 것이다. 안드로이드는 인기 있는 자바의 엔터프라이즈 프로그래밍 언어를 사용하고, 라이선스 비용은 무료인데다가, 사용자 요구에 대한 접근 및 제어 수준을 제공하여 탁월한 엔터프라이즈 플랫폼을 선사한다.

안드로이드에만 있는 것

앞서 나열한 3D 그래픽스와 네이티브 데이터베이스 지원 같은 많은 기능은 다른 모바일 SDK에서도 제공한다. 안드로이드를 차별화하는 독특한 기능 몇 가지를 살펴보면 다음과 같다.

- ❑ **구글 맵 어플리케이션** 모바일용 구글 맵은 매우 인기 있는데, 안드로이드는 구글 맵을 여러분의 어플리케이션에서 사용할 수 있도록 재사용 가능한 컨트롤 형태로 제공한다. MapView 위젯은 익숙한 구글 맵 인터페이스를 사용하는 맵기반 어플리케이션 구축을 위해, 여러분의 액티비티에 구글 맵을 표시하고, 조작하고, 주석을 달 수 있도록 해준다.

- ❑ **백그라운드 서비스와 어플리케이션** 백그라운드 서비스는 다른 어플리케이션이 사용되고 있는 동안 혹은 휴대폰이 여러분에게 무언가를 알리고자 벨을 울리거나, 불빛을 깜박이거나, 진동을 발생시키기 전까지의 기간인 유휴상태에 있는 동안 조용히 동작하는, 이벤트중심 모델event-driven model을 사용하는 어플리케이션을 만들도록 해준다. 주식 시장을 추적하는 어플리케이션이나, 여러분의 자산 구성에 생긴 주요한 변화를 알려주는 어플리케이션, 혹은 여러분의 현재 위치와 시간 그리고 발신자의 신원에 따라 벨 소리나 볼륨을 바꾸는 서비스 등이 있을 수 있다.

- ❑ **공유 데이터와 프로세스간 통신** 안드로이드는 인텐트와 콘텐트 공급자를 통해 여러분의 어플리케이션이 메시지를 주고받고, 처리를 수행하고, 데이터를 공유하도록 해준다. 또한 여러분은 이들 메커니즘을 사용해 네이티브 안드로이드 어플리케이션이 제공하는 데이터와 기능을 적극 활용할 수도 있다. 이러한 오픈 전략의 위험을 완화하기 위해, 각 어플리케이션의 프로세스와 데이터 저장소 그리고 파일은 11장에 설명된 완전한 권한기반 보안 메커니즘을 통해 다른 어플리케이션과 명시적으로 공유되지 않는 한 공개되지 않는다.

- ❑ **모든 어플리케이션은 동등하게 만들어진다** 안드로이드는 네이티브 어플리케이션과 서드파티에서 개발한 어플리케이션을 차별하지 않는다. 이는 소비자로 하여금 모든 네이티브 어플리케이션을 동일한 하부 데이터와 하드웨어에 접근하는 서드파티 대체물로 전부 교체할 수 있도록 함으로써, 장치가 가진 룩앤필look and feel을 바꾸는 전례 없는 능력을 제공한다. SDK 초기 릴리즈에서는 "잠금 해제unlock" 화면과 "통화 경험in-call experience" 화면을 바꿀 수 없다.

- ❑ **P2P 장치간 어플리케이션 메시징** 안드로이드는 현재 상태 알림, 인스턴트 메시징, 그리고 장치간/어플리케이션간 통신을 지원하는 피어-투-피어 메시징을 제공한다.

모바일 개발 판도의 변화

기존에 존재하고 있는 모바일 개발 플랫폼은 모바일 개발을 둘러싼 사방에 독점이라는 아우라를 만들었다. 설계상의 문제 때문이건 아니면 네이티브 어플리케이션 개발에 수반되는 비용이나 복잡도의 부수효과 때문이건 간에, 대부분의 휴대폰은 처음 상자를 개봉했을 때와 거의 동일한 상태로 남아있을 것이다.

반면, 안드로이드는 근본적인 변경을 허용할 뿐만 아니라 심지어 장려하기까지 한다. 소비자 장치로서 안드로이드 폰은 소비자가 새로운 폰에 대해 요구하는 표준 어플리케이션의 핵심 집합을 탑재하고 있지만, 진짜 힘은 사용자가 장치의 모양, 분위기, 기능을 완전히 바꿀 수 있는데 있다.

안드로이드는 개발자에게 엄청난 기회를 선사한다. 모든 안드로이드 어플리케이션은 그저 폰의 최상위에 있는 샌드박스에서 동작하는 소프트웨어가 아니라, 폰 자체의 고유한 한 부분이다. 이제 여러분은 속도가 느린 장치에서 동작할 수 있는 작은 화면 버전의 소프트웨어가 아닌, 사람들이 폰을 사용하는 방식을 바꾸는 모바일 어플리케이션을 작성할 수 있다.

오픈 소스 개발 프레임워크로서 안드로이드는 이미 존재하고 있는 모바일 개발 플랫폼뿐만 아니라 미래의 것과도 여전히 경쟁해야 하지만, 개발 환경의 강력함에 있어서는 안드로이드가 훨씬 유리하다. 폰이 가진 리소스에 대한 완전한 접근과 더불어, 모바일 어플리케이션 개발에 대한 안드로이드의 자유로우면서도 열린 접근방식은 확실히 올바른 방향으로 내디딘 위대한 걸음이다.

개발 프레임워크 소개

홍보는 이쯤 해서 마무리하기로 하고, 이제 안드로이드를 위한 어플리케이션 개발 방법에 대해 살펴보자. 안드로이드 어플리케이션은 자바를 프로그래밍 언어로 사용해 작성되지만, 전통적인 자바 VM이 아닌 *Dalvik*이라는 커스텀 가상 머신을 통해 실행된다.

이번 장 후반부에서는 안드로이드 소프트웨어 스택의 기술적인 설명으로 시작해, SDK에 포함된 내용을 살펴보고, 안드로이드 라이브러리를 소개하며, Dalvik 가상 머신을 살펴봄으로써 프레임워크에 대해 알아볼 것이다.

각각의 안드로이드 어플리케이션은 각자 자신만의 Dalvik 인스턴스 내에서 별도의 프로세스로 동작하며, 메모리와 프로세스 관리에 대한 모든 책임을 안드로이드 런타임에 양도한다. 이 런타임은 리소스 관리를 위해 필요 시 프로세스를 멈추고 종료시킨다.

Dalvik과 안드로이드 런타임은 드라이버와 메모리 관리를 포함한 하위 수준의 하드웨어 상호작용을 처리하는 리눅스 커널 위에 얹혀있으며, API 집합은 하부에 놓인 모든 서비스와 기능 그리고 하드웨어에 대한 접근을 제공한다.

SDK에는 무엇이 들어있을까?

안드로이드 소프트웨어 개발 킷(SDK)은 여러분이 안드로이드 어플리케이션을 개발, 테스트, 디버그 하는데 필요한 모든 것을 가지고 있다. 다운로드 받은 SDK에는 아래와 같은 것들이 포함되어 있다.

- **안드로이드 API** SDK의 핵심은 개발자에게 안드로이드 스택에 대한 접근을 제공하는 안드로이드 API 라이브러리다. 이는 구글이 네이티브 안드로이드 어플리케이션을 만들기 위해 사용한 것

과 동일한 라이브러리다.

- **개발 도구** 안드로이드 소스 코드를 실행 가능한 안드로이드 어플리케이션으로 바꾸기 위해, SDK는 여러분의 어플리케이션을 컴파일하고 디버그 하도록 해주는 몇 가지 개발 도구를 포함하고 있다. 개발자 도구에 대해서는 2장에서 보다 자세히 배울 것이다.

- **안드로이드 에뮬레이터** 안드로이드 에뮬레이터는 변경 가능한 여러 스킨을 가지고 있는 완전히 인터랙티브한 안드로이드 장치 에뮬레이터다. 이 에뮬레이터를 사용하면 여러분의 어플리케이션이 실제 안드로이드 장치에서 어떻게 보여지고 또 행동하게 되는지를 살펴볼 수 있다. 모든 안드로이드 어플리케이션은 Dalvik 가상 머신에서 실행되기 때문에 이 소프트웨어 에뮬레이터는 하나의 훌륭한 환경이 된다. 사실, 이 에뮬레이터는 하드웨어 중립적이기 때문에, 그 어떤 단일한 하드웨어 구현보다도 더 나은 독립적인 테스트 환경을 제공한다.

- **풍부한 문서** SDK는 각 패키지와 클래스에 무엇이 포함되어 있는지 그리고 이들의 사용법은 어떻게 되는지를 자세하고 정확하게 설명하고 있는 광범위한 코드 수준의 레퍼런스 정보를 가지고 있다. 이 코드 문서 외에도 안드로이드 레퍼런스 문서는 안드로이드 개발을 시작하는 방법과 안드로이드 개발 이면의 원리를 자세히 설명한다.

- **샘플 코드** 안드로이드 SDK는 각 API 기능의 사용법을 조명하는 간단한 프로그램뿐만 아니라, 안드로이드로 할 수 있는 몇 가지 것들을 설명하는 엄선된 샘플 어플리케이션을 포함하고 있다.

- **온라인 지원** 상대적으로 늦게 뛰어들었음에도 불구하고, 안드로이드는 활발한 개발자 커뮤니티를 형성했다. http://code.google.com/android/groups에 있는 구글 그룹Google Groups은 구글의 안드로이드 개발 팀이 정기적으로 개시하는 정보와 더불어 현재 활동중인 안드로이드 개발자 포럼이다.

안드로이드는 널리 사용되고 있는 이클립스Eclipse IDE 사용자를 위해, 프로젝트 생성을 단순화하고 안드로이드 에뮬레이터와 디버깅 도구를 이클립스에 긴밀히 통합하는 전용 플러그 인을 발표했다. 이 ADT 플러그 인의 기능은 2장에서 보다 자세히 다룬다.

안드로이드 소프트웨어 스택의 이해

안드로이드 소프트웨어 스택은 그림 1-1에 보이는 요소들로 구성되는데, 이들에 대한 자세한 내용은 그림 아래에 설명되어 있다. 간단히 말해, 리눅스 커널과 C/C++ 라이브러리 모음은, 런타임과 어플리케이션을 위한 서비스 및 이들의 관리를 제공하는 어플리케이션 프레임워크를 통해 노출된다.

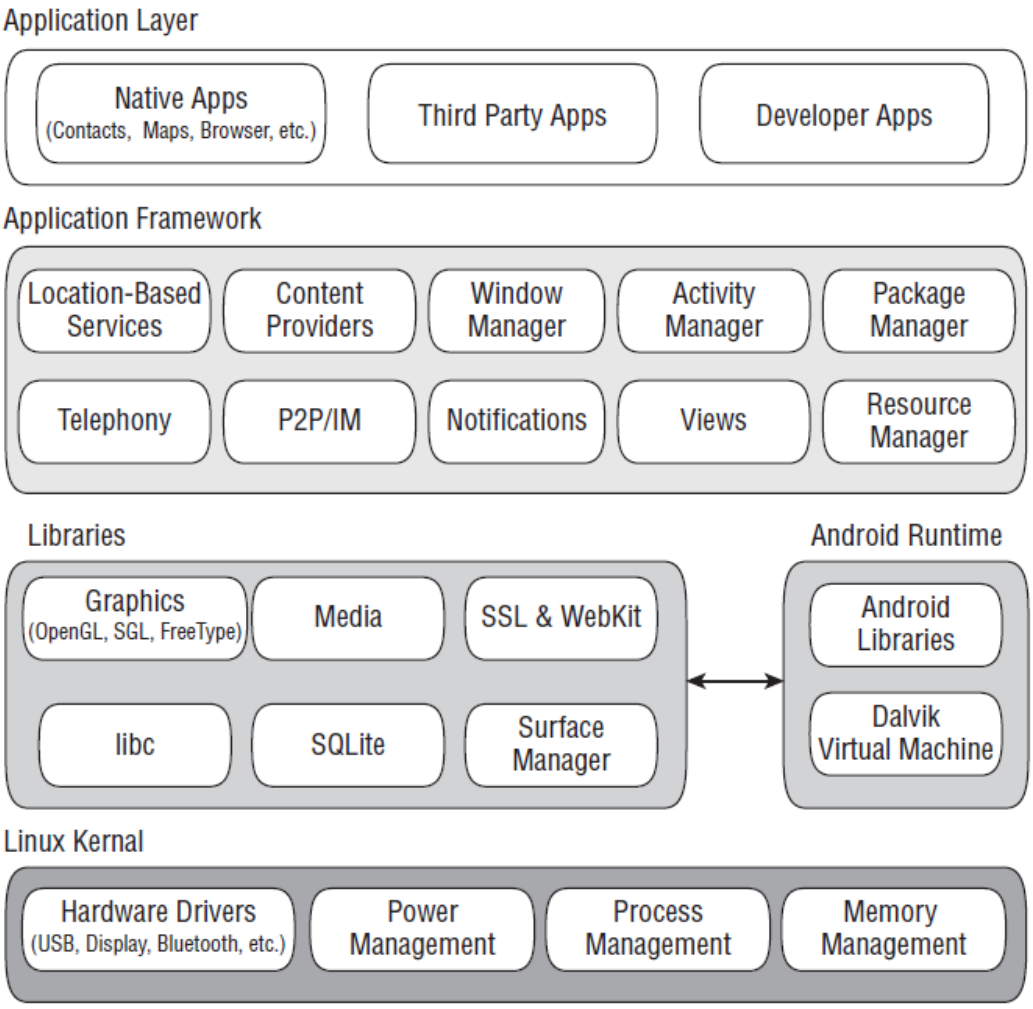

그림 1-1

- **리눅스 커널** 하드웨어 드라이버, 프로세스와 메모리 관리, 보안, 네트워크, 전력 관리 등의 핵심 서비스는 리눅스 2.6 커널이 담당한다. 또한 커널은 하드웨어와 스택의 나머지부분 사이에 추상 계층을 제공한다.

- **라이브러리** 라이브러리는 커널 바로 위에서 동작한다. 안드로이드는 libc와 SSL 같은 다양한 C/C++ 코어 라이브러리뿐만 아니라 아래와 같은 것들도 포함하고 있다.

  - 오디오 및 비디오 미디어 재생을 위한 미디어 라이브러리
  - 디스플레이 관리를 제공하는 서피스 관리자Surface manager
  - 2D 및 3D 그래픽스를 위한 SGL, OpenGL 등의 그래픽스 라이브러리
  - 네이티브 데이터베이스 지원을 위한 SQLite
  - 통합 웹 브라우저와 인터넷 보안을 위한 SSL과 WebKit

- **안드로이드 런타임** 안드로이드 폰을 모바일 리눅스 구현이 아닌 안드로이드 폰이게끔 만드는 것은 다름아닌 안드로이드 런타임이다. 코어 라이브러리와 Dalvik 가상 머신을 포함하고 있는 안드로이드 런타임은 라이브러리와 함께 여러분의 어플리케이션을 움직이게 하는 엔진으로서, 어플리케이션 프레임워크의 기본을 형성한다.

  - **코어 라이브러리** 안드로이드 개발은 자바로 이뤄지지만 Dalvik은 자바 VM이 아니다. 코어 안드로이드 라이브러리는 코어 자바 라이브러리에서 사용 가능한 대부분의 기능뿐만 아니라 안드로이드 전용 라이브러리도 제공한다.

  - **Dalvik 가상 머신** Dalvik은 장치가 복수개의 인스턴스를 효율적으로 실행할 수 있도록 보장하기 위해 최적화된 레지스터기반의 가상 머신이다. Dalvik은 스레딩threading과 하위 수준의 메모리 관리를 위해 리눅스 커널을 의존한다.

- **어플리케이션 프레임워크** 어플리케이션 프레임워크는 안드로이드 어플리케이션을 만드는데 사용되는 클래스를 제공하며, 하드웨어 접근 및 사용자 인터페이스와 어플리케이션 리소스 관리를 위한 일반적 추상을 제공한다.

- **어플리케이션 계층** 네이티브 어플리케이션과 서드파티 어플리케이션은 모두 동일한 API 라이브러리를 사용하는 어플리케이션 계층 위에 구축된다. 어플리케이션 계층은 어플리케이션 프레임워크를 통해 사용 가능한 클래스와 서비스를 사용함으로써 안드로이드 런타임 내에서 실행된다.

Dalvik 가상 머신

안드로이드의 핵심 구성요소 가운데 하나가 바로 Dalvik 가상 머신이다. 안드로이드는 자바 MEJava Mobile Edition 같은 전통적인 자바 가상 머신(VM)을 사용하지 않고, 단일 장치상에서 복수개의 인스턴스가 효율적으로 실행되게끔 보장하도록 설계된 자신만의 고유한 커스텀 VM을 사용한다.

Dalvik VM은 장치 하부에 놓인 리눅스 커널을 사용해 보안, 스레딩, 그리고 프로세스와 메모리 관리 등의 하위 수준 기능을 처리한다. 하부에 있는 리눅스 OS상에서 직접 실행되는 C/C++ 어플리케이션을 작성하는 것 역시 가능하지만, 대부분의 경우에는 이렇게 해야 할 이유가 없다.

> *이 책은 Dalvik 내에서 실행되는 어플리케이션 작성에만 초점을 맞춘다. 만일 여러분이 하고자 하는 바가 안드로이드의 리눅스 커널 및 C/C++의 취약점을 찾아내거나, Dalvik의 일부를 수정하거나, 혹은 내부를 뜯어보고자 하는 것이라면, http://groups.google.com/group.android-internals에 있는 안드로이드 인터널 구글 그룹Android Internals Google Group을 살펴보자.*

안드로이드 하드웨어와 시스템 서비스에 대한 모든 접근은 Dalvik을 중간 단으로 사용해 관리된다. VM을 사용해 어플리케이션 실행을 호스팅 함으로써, 개발자는 특정 하드웨어 구현에 대해 전혀 걱정할 필요가 없도록 보장하는 추상 계층을 갖는다.

Dalvik VM은 메모리 사용량을 최소로 하도록 보장하기 위해 최적화된 포맷인 Dalvik 실행 파일을 실행한다. .dex 실행이미지는 클래스로 컴파일 된 자바 언어를 SDK가 제공하는 도구를 사용해 변환 함으로써 만들어진다. Dalvik 실행이미지를 만드는 방법은 다음 장에서 보다 자세히 배울 것이다.

안드로이드 어플리케이션 아키텍처

안드로이드 아키텍처는 컴포넌트 재사용 개념을 장려하며, 여러분이 액티비티와 서비스 그리고 데이터를 발행해, 여러분이 설정한 보안 제한에 의해 관리되는 접근을 통해 다른 어플리케이션과 공유할 수 있도록 해준다.

여러분으로 하여금 연락처 관리자나 폰 다이얼러 대체물을 만들도록 해주는 메커니즘은, 여러분의 어플리케이션 컴포넌트를 노출하여, 다른 개발자가 새로운 UI 및 기능 확장을 만들거나 그 위에 구축하도록 할 수 있다.

아래의 어플리케이션 서비스는 모든 안드로이드 어플리케이션의 구조적인 주춧돌로서, 여러분의 소프트웨어를 위해 사용될 프레임워크를 제공한다.

- **액티비티 관리자** 3장에 설명된 액티비티 스택 관리를 포함하여, 액티비티의 수명 주기를 제어한다.
- **뷰** 액티비티의 사용자 인터페이스를 구성하는데 사용되며, 4장에서 살펴본다.
- **알림 관리자** 사용자에게 신호를 보내기 위한 일관되면서도 간섭적이지 않은 메커니즘을 제공하며, 8장에서 살펴본다.
- **콘텐트 공급자** 여러분의 어플리케이션이 다른 어플리케이션과 데이터를 공유할 수 있도록 해주며, 6장에서 살펴본다.
- **리소스 관리자** 문자열과 그래픽스 같은 외부화externalize되는 비 코드 리소스non-code resources를 지원하며, 3장에서 살펴본다.

안드로이드 라이브러리

안드로이드는 어플리케이션 개발을 위한 많은 API를 제공한다. 아래에 있는 코어 API 목록을 살펴보면 과연 어떤 일을 할 수 있는지 감이 올 것이다. 모든 안드로이드 장치는 적어도 이들 API에 대한 지원 만큼은 제공할 것이다.

- **android.util** 코어 유틸리티 패키지는 특수 컨테이너, 문자열 포맷터, XML 파싱 유틸리티 같은 하위 수준의 클래스를 포함하고 있다.

- **android.os** 운영체제 패키지는 메시지 전달, 프로세스간 통신, 시계 기능, 디버깅 같은 기본적인 운영체제 서비스에 대한 접근을 제공한다.

- **android.graphics** 그래픽스 API는 캔버스, 색상, 그리기 요소drawing primitives를 지원하는 하위 수준의 그래픽스 클래스를 제공하여 캔버스 위에 그릴 수 있도록 해준다.

- **android.text** 텍스트 출력 및 파싱을 위한 텍스트 처리 도구.

- **android.database** 데이터베이스 작업 시 커서cursors 처리에 필요한 하위 수준의 클래스를 제공한다.

- **android.content** 콘텐트 API는 리소스, 콘텐트 공급자, 패키지를 다루는 서비스 제공함으로써 데이터의 접근과 발행을 관리하는데 사용된다.

- **android.view** 뷰는 핵심 사용자 인터페이스 클래스다. 모든 사용자 인터페이스 요소는 사용자와 상호작용하는 컴포넌트를 제공하기 위한 일련의 뷰로 구성된다.

- **android.widget** 위젯 클래스는 어플리케이션에서 사용하기 위해 "미리 만들어둔" 사용자 인터페이스 요소로서, 뷰 패키지 위에 구축된다. 리스트, 버튼, 레이아웃이 여기에 포함된다.

- **com.google.android.maps** 여러분의 어플리케이션 내에서 사용할 수 있는 네이티브 맵 컨트롤에 대한 접근을 제공하는 상위 수준의 API다. MapView 컨트롤과 더불어 내포된 맵을 제어하고 주석을 다는데 사용되는 Overlay와 MapController 클래스를 포함한다.

- **android.app** 어플리케이션 모델에 대한 접근을 제공하는 상위 수준의 패키지. 어플리케이션 패키지는 모든 안드로이드 어플리케이션을 위한 토대를 형성하는 액티비티와 서비스 API를 가지고 있다.

- **android.provider** 공급자 패키지는 개발자가 표준 콘텐트 공급자에 쉽게 접근할 수 있도록, 모든 안드로이드 배포distributions에 포함된 표준 데이터베이스에 대한 접근을 제공하는 클래스를 제공한다.

- **android.telephony** 전화 API는 장치의 전화 스택phone stack과 직접 상호작용할 수 있는 능력을 제공하며, 전화, 전화 상태, SMS 메시지를 보내고, 받고, 모니터 하도록 해준다.

- **android.webkit** WebKit 패키지는 여러분의 액티비티와 쿠키 관리자cookie manager에 브라우저를 내포하기 위한 WebView 컨트롤을 포함하여, 웹기반 콘텐트를 다루기 위한 API를 가지고 있다.

안드로이드 스택은 안드로이드 API 외에도 어플리케이션 프레임워크를 통해 노출되는 C/C++ 라이브러리 셋을 포함하고 있다. 이들 라이브러리는 다음과 같다.

- **OpenGL** OpenGL ES 1.0 API에 기반을 둔 3D 그래픽스 지원에 사용되는 라이브러리
- **FreeType** 비트맵 및 벡터 폰트 렌더링 지원
- **SGL** 2D 그래픽스 엔진 제공을 위해 사용되는 코어 라이브러리
- **libc** 리눅스기반 임베디드 장치에 최적화된 표준 C 라이브러리
- **SQLite** 어플리케이션 데이터 저장에 사용되는 작고 가벼운 관계형 데이터베이스 엔진
- **SSL** 안전한 인터넷 통신을 위한 Secure Sockets Layer 암호 프로토콜 사용 지원

고급 안드로이드 라이브러리

코어 라이브러리는 여러분이 안드로이드 어플리케이션을 만들기 시작하는데 필요한 모든 기능을 제공하지만, 오래지 않아 정말로 흥미로운 기능을 제공하는 고급 API를 찾으려 할 것이다.

안드로이드는 폭넓은 범위의 다양한 모바일 하드웨어를 대상으로 하길 기대하므로, 아래에 나와있는 API의 적합성과 구현은 이들이 구현된 장치에 따라 다양할 것이라는 사실을 염두에 둬야 한다.

- **android.location** 위치기반 서비스 API는 여러분의 어플리케이션이 장치의 현재 물리적인 위치에 접근할 수 있도록 해준다. 위치기반 서비스는 장치에서 사용할 수 있는 위치고정position-fixing 하드웨어나 기술이 무엇이든 관계 없이 사용해 위치 정보에 대한 일반적 접근을 제공한다.
- **android.media** 미디어 API는 스트리밍 된 미디어를 포함해, 오디오 및 비디오 미디어 파일의 재생과 기록에 대한 지원을 제공한다.
- **android.opengl** 안드로이드는 여러분의 어플리케이션을 위한 역동적인 3D 사용자 인터페이스를 만드는데 사용할 수 있는, OpenGL ES API를 사용하는 강력한 3D 렌더링 엔진을 제공한다.
- **android.hardware** 사용 가능한 경우, 하드웨어 API는 카메라, 가속도 센서, 나침반 센서를 포함한 센서 하드웨어를 노출하는데, 이는 10장에서 살펴본다.
- **android.bluetooth, android.net.wifi, android.telephony** 안드로이드는 또한 블루투스, Wi-Fi, 전화 하드웨어를 포함한 하드웨어 플랫폼에 대한 하위 수준의 접근을 제공하는데, 이는 10장에서 살펴본다.

요약

이번 장은 최신 휴대폰에서 사용 가능한 하드웨어 기능의 상당한 발전에도 불구하고, 이를 위해 사용 가능한 소프트웨어는 뒤쳐져있음을 설명했다. 개방성의 부재와 사용하기 힘든 개발 킷 그리고 하드웨어 종속적인 API는 모바일 소프트웨어의 혁신을 억눌러왔다.

안드로이드는 보통 기존에 존재하는 전용 모바일 개발 프레임워크와 관련된 제약사항 없이, 개발자에게 모바일 장치를 위한 혁신적인 소프트웨어 어플리케이션을 만들 수 있도록 기회를 준다.

여러분은 어플리케이션 계층 및 개발 툴 킷과 더불어, Dalvik VM, 커스텀 런타임, 코어 라이브러리, 리눅스 커널을 포함하고 있는 완전한 안드로이드 소프트웨어 스택을 살펴봤다. 이들은 모두 오픈 소스로 사용 가능해질 것이다.

소비자가 안드로이드 폰을 가지고 싶어하도록 만들 어플리케이션을 만들어야만 하는 개발자(안드로이드의 주요 고객으로서)의 의무와 함께 오픈 핸드셋 얼라이언스를 소개했다.

또한 아래와 같은 것들도 배웠다.

- ❑ 하드웨어 기능의 확장 범위와 함께 휴대폰이 개발자에게 더 나은 접근을 제공하는 도구의 수요를 어떻게 만들었는지.
- ❑ 피어-투-피어 메시징, 네이티브 맵 지원, 하드웨어 접근, 백그라운드 서비스, 프로세스간 메시징과 장치간 메시징, 공유 데이터베이스, 2D와 3D 그래픽스 등 안드로이드를 통해 개발자가 사용할 수 있는 몇 가지 기능에 관해.
- ❑ 모든 안드로이드 어플리케이션은 동등하게 만들어지며, 코어 네이티브 어플리케이션의 대체를 포함해 한 어플리케이션을 다른 것으로 완전히 대체할 수 있다는 사실.
- ❑ 안드로이드 SDK는 개발자 도구와 API 그리고 광범위한 문서를 포함하고 있다는 사실.

다음 장에서는, 안드로이드 SDK를 다운로드 받아 설치하고, 이클립스에 안드로이드 개발 환경을 구축해 본다.

또한 여러분의 첫 번째 안드로이드 어플리케이션을 만들기에 앞서, 능률적인 개발, 테스팅, 디버깅을 위한 안드로이드 개발자 도구 플러그 인의 사용법을 배울 것이다.

안드로이드 어플리케이션의 빌딩 블록에 대해 배우고 난 뒤에는, 여러분이 만들 수 있는 서로 다른 어플리케이션 타입을 살펴보며, 모바일 장치용 어플리케이션 개발 시 따라야 할 몇 가지 설계 고려사항을 이해하기 시작할 것이다.