

Nhập môn phát triển ứng dụng Android

Nền vật dụng mã nguồn mở

Mức độ: Nhập môn

Frank Ableson, Software designer

22 01 2011

Android là một môi trường hệ điều hành hoàn chỉnh dựa trên hệ điều hành nhân Linux® V2.6. Lúc đầu, đích triển khai đối với Android là lĩnh vực điện thoại di động, gồm các loại điện thoại thông minh và các loại thiết bị điện thoại kiểu gập chi phí thấp. Tuy nhiên, phạm vi đầy đủ các dịch vụ điện toán của Android và sự hỗ trợ chức năng phong phú của nó có tiềm năng mở rộng vượt ra ngoài thị trường điện thoại di động. Android có thể hữu ích đối với các nền tảng và ứng dụng khác. Trong bài viết này, bạn sẽ được giới thiệu về nền Android và học cách viết mã một ứng dụng Android cơ bản.

Giới thiệu

BlackBerry và iPhone, với các nền di động dung lượng cao và hấp dẫn, đang nhắm đến hai đầu ngược chiều nhau của phổ các điện thoại di động. BlackBerry vững chắc như đá đối với những người sử dụng mức nghiệp vụ chuyên nghiệp. Để làm một thiết bị tiêu dùng, nó khó cạnh tranh được với iPhone về tính dễ sử dụng và “yếu tố tuyệt tác” (cool factor). Android, một nền tảng vẫn còn trẻ và chưa được kiểm chứng, có tiềm năng đóng vai trò ở cả hai đầu của phổ các điện thoại di động và thậm chí có thể làm cầu nối qua hố sâu ngăn cách giữa làm việc và vui chơi.

Ngày nay, có nhiều thiết bị dựa trên mạng hay có khả năng nối mạng chạy một biến thể hệ điều hành nhân Linux. Đó là một nền vững chắc: có hiệu quả về chi phí khi triển khai và hỗ trợ, và sẵn sàng được chấp nhận như là một cách tiếp cận thiết kế tốt để triển khai. Giao diện người sử dụng dành cho các thiết bị như vậy thường dựa trên HTML và có thể xem được bằng một trình duyệt cho PC hay Mac. Nhưng không phải vật dụng nào cũng cần điều khiển bằng một thiết bị tính toán chung. Hãy xem xét một vật dụng thông thường, chẳng hạn như một lò nướng, lò vi sóng hoặc để làm bánh mì. Điều gì xảy ra nếu các vật dụng gia đình của bạn được Android điều khiển và hiển thị ra một màn hình màu cảm ứng? Với một giao diện người sử dụng Android trên mặt lò, tác giả thậm chí có thể nấu nướng một món gì đó.

Trong bài này, chúng ta hãy tìm hiểu về nền Android và cách có thể sử dụng nó như thế nào cho các ứng dụng di động và không phải di động. Hãy cài đặt Bộ công cụ phát triển phần mềm Android (Android SDK) và xây dựng một ứng dụng đơn giản. Hãy [Tải về](#) mã nguồn của ứng dụng mẫu trong bài viết này.

Lược sử về Android

Nền Android là sản phẩm của [Open Handset Alliance](#) (Liên minh thiết bị cầm tay mở), một tập đoàn các tổ chức cùng hợp tác với nhau để xây dựng nên một loại điện thoại di động tốt hơn. Tập đoàn do Google đứng đầu này gồm các nhà vận hành (điện thoại) di động, các nhà sản xuất thiết bị cầm tay, sản xuất linh kiện, các nhà cung cấp nền và giải pháp phần mềm, và các công ty tiếp thị. Từ một quan điểm phát triển phần mềm, Android nằm chính giữa thế giới mã nguồn mở.

Thiết bị cầm tay có khả năng Android đầu tiên trên thị trường là thiết bị G1 do HTC sản xuất và được bán trên T-Mobile. Nó trở nên sẵn có sau gần một năm đồn đại, khi mà các công cụ phát triển phần mềm sẵn có duy nhất chỉ là một số bản phát hành Bộ công cụ phát triển phần mềm (SDK) được cải tiến dần từng bước. Khi ngày phát hành G1 gần đến, nhóm Android đã phát hành SDK V1.0 và các ứng dụng bắt đầu vượt lên nhằm tới nền tảng mới này.

Để kích thích đổi mới, Google đã tài trợ hai vòng cuộc thi "Các Thách thức Nhà Phát triển Android," nơi mà hàng triệu đô la được đổ vào các ứng dụng dự thi hàng đầu. Vài tháng sau G1, Android Market được phát hành, cho phép người sử dụng duyệt và tải về các ứng dụng trực tiếp vào điện thoại của họ. Trong khoảng 18 tháng, một nền di động mới đã bước lên vũ đài công khai.

Nền tảng Android

Với khả năng rộng rãi của Android, sẽ rất dễ dàng nhầm lẫn nó với một hệ điều hành máy tính để bàn. Android là một môi trường phân tầng, xây dựng trên nền của hệ điều hành nhân Linux, và nó bao hàm nhiều chức năng phong phú. Hệ thống con giao diện người sử dụng gồm có:

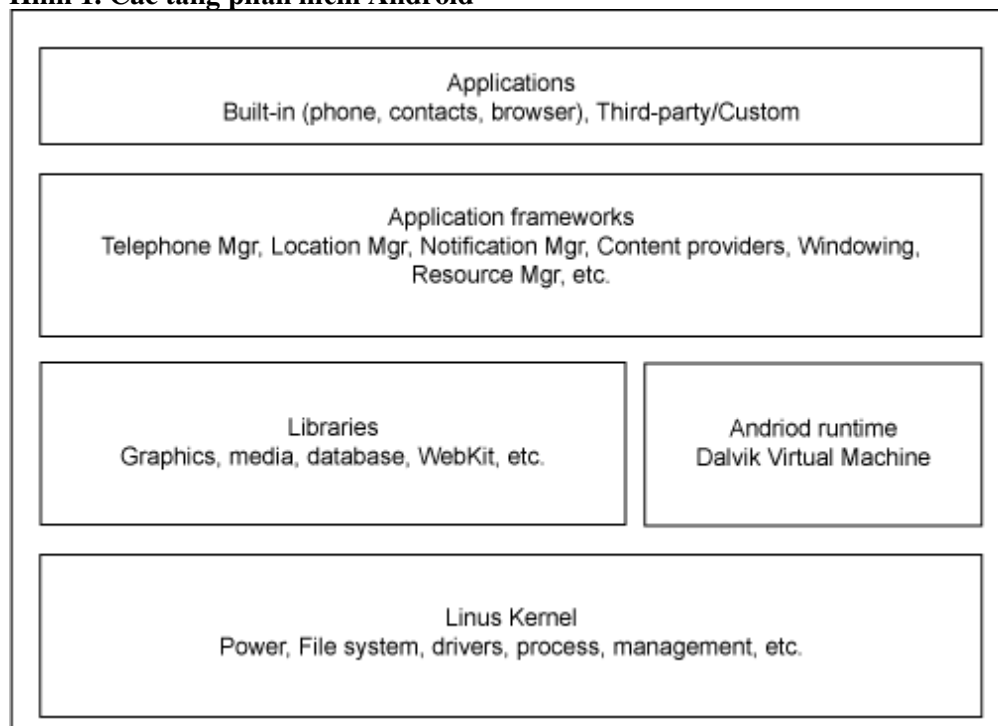
- Cửa sổ
- Khung nhìn
- Các tiểu trình để hiển thị các phần tử phổ biến như các hộp biên soạn, danh sách, danh sách thả xuống

Android gồm một trình duyệt có thể nhúng vào được, được xây dựng dựa trên WebKit, chính là máy (engine) trình duyệt mã nguồn mở cũng đang là động cơ của trình duyệt Safari di động của iPhone.

Android khoe ra một mảng dồi dào nhiều tùy chọn kết nối, gồm WiFi, Bluetooth, và dữ liệu không dây qua một kết nối di động (như GPRS, EDGE, và 3G). Một kỹ thuật phổ biến trong các ứng dụng Android là liên kết tới Google Maps để hiển thị một địa chỉ trực tiếp trong một ứng dụng. Việc hỗ trợ cho các dịch vụ dựa trên địa điểm (chẳng hạn như GPS) và các dụng cụ đo gia tốc cũng có sẵn trong chồng phần mềm Android, mặc dù không phải tất cả các thiết bị Android được trang bị với các phần cứng cần thiết. Cũng có cả hỗ trợ máy ảnh (camera) nữa.

Về mặt lịch sử, có hai lĩnh vực mà các ứng dụng di động đã gắng sức theo kịp các đối tác máy tính để bàn là đồ họa/đa phương tiện, và phương thức lưu trữ dữ liệu. Android giải quyết thách thức về đồ họa bằng sự hỗ trợ dựng sẵn cho đồ họa 2-D và 3-D, gồm cả thư viện OpenGL. Gánh nặng lưu trữ dữ liệu được giảm nhẹ vì nền Android có gồm cả cơ sở dữ liệu SQLite mã nguồn mở phổ biến. Hình 1 cho thấy một khung nhìn đơn giản hoá về các tầng phần mềm Android.

Hình 1. Các tầng phần mềm Android

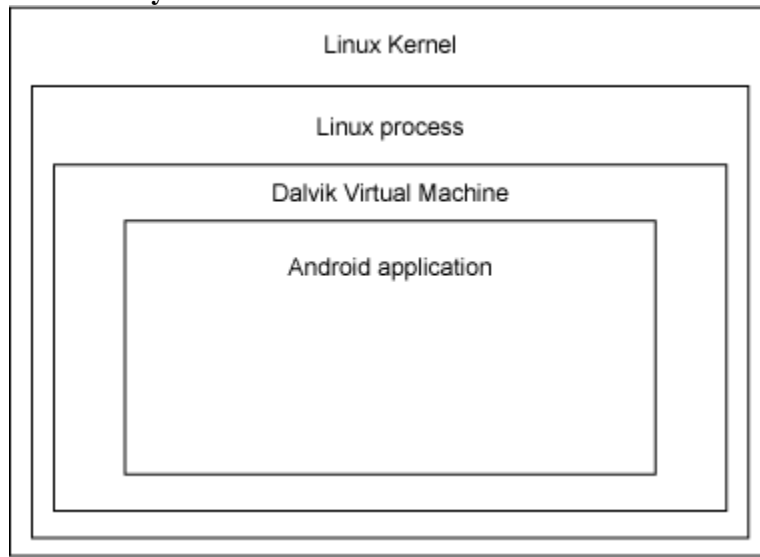


Kiến trúc ứng dụng

Như đã đề cập, Android chạy bên trên một hệ điều hành nhân Linux. Các ứng dụng Android được viết bằng ngôn ngữ lập trình Java, và chúng chạy trong một máy ảo (VM). Điều quan trọng là phải lưu ý rằng VM không phải là một máy

ảo Java (JVM) như bạn chờ đợi, mà là máy ảo Dalvik (Dalvik Virtual Machine), một công nghệ mã nguồn mở. Mỗi ứng dụng Android chạy trong một cá thể của máy ảo Dalvik, đến lượt mình, cá thể máy ảo này nằm trong một tiến trình do nhân Linux quản lý, như trình bày dưới đây.

Hình 2. Máy ảo Dalvik



Một ứng dụng Android gồm có một hoặc nhiều phân loại sau đây:

Hoạt động

Một ứng dụng có một giao diện người dùng nhìn thấy được, sẽ được thực hiện bằng một hoạt động. Khi người sử dụng chọn một ứng dụng từ màn hình chủ hoặc trình khởi chạy ứng dụng, một hoạt động được khởi động.

Dịch vụ

Một dịch vụ nên được sử dụng đối với bất kỳ ứng dụng nào cần tồn tại trong một thời gian dài, chẳng hạn như một trình giám sát mạng hoặc ứng dụng kiểm tra-cập nhật.

Trình cung cấp nội dung

Bạn có thể coi các trình cung cấp nội dung như là một máy chủ cơ sở dữ liệu. Công việc của một trình cung cấp nội dung là quản lý truy cập vào dữ liệu đang tồn tại, chẳng hạn như một cơ sở dữ liệu SQLite. Nếu ứng dụng của bạn rất đơn giản thì bạn không nhất thiết phải tạo ra một trình cung cấp nội dung. Nếu bạn đang xây dựng một ứng dụng lớn hơn, hoặc một ứng dụng để cung cấp dữ liệu sẵn sàng cho nhiều hoạt động hoặc ứng dụng khác, một trình cung cấp nội dung là phương tiện truy cập dữ liệu của bạn.

Trình thu sóng phát

Một ứng dụng Android có thể được khởi chạy để xử lý một phần tử dữ liệu hoặc đáp ứng một sự kiện, chẳng hạn như nhận một văn bản thông báo.

Một ứng dụng Android, cùng với một tập tin tên là `AndroidManifest.xml`, được triển khai lên một thiết bị. `AndroidManifest.xml` chứa các thông tin cấu hình cần thiết để cài đặt nó đúng đắn vào thiết bị. Nó gồm các tên lớp cần thiết và các kiểu sự kiện mà ứng dụng có thể xử lý, và các quyền hạn ứng dụng cần có để chạy. Thí dụ, nếu một ứng dụng đòi hỏi việc truy cập vào mạng — để tải về một tập tin chẳng hạn — thì quyền hạn này phải được tuyên bố một cách tường minh trong tập tin manifest. Nhiều ứng dụng có thể có các quyền hạn riêng này được kích hoạt. Đảm bảo an ninh bằng khai báo như vậy giúp giảm bớt khả năng xảy ra là một ứng dụng xấu có thể làm thiệt hại thiết bị của bạn.

Phần tiếp theo bàn luận về môi trường phát triển cần có để xây dựng nên một ứng dụng Android.

Các công cụ đòi hỏi phải có

Cách dễ nhất để bắt đầu phát triển các ứng dụng Android là tải về Bộ công cụ phát triển phần mềm Android và Môi

trường phát triển tích hợp Eclipse (Eclipse IDE) (xem phần [Tài nguyên](#)). Việc phát triển Android có thể diễn ra trên Microsoft® Windows®, Mac OS X, hoặc Linux.

Bài viết này giả thiết rằng bạn đang sử dụng Môi trường phát triển tích hợp Eclipse và trình cài thêm Android Developer Tools dùng cho Eclipse. Các ứng dụng Android được viết bằng ngôn ngữ Java, nhưng biên dịch và thi hành bằng máy ảo Dalvik (một máy ảo không phải Java). Viết mã bằng ngôn ngữ Java trong Eclipse là rất trực quan; Eclipse cung cấp một môi trường Java phong phú, gồm cả việc trợ giúp theo ngữ cảnh và các mách nước gợi ý về mã. Khi bộ mã Java của bạn được biên dịch sạch sẽ, Android Developer Tools sẽ đảm bảo rằng ứng dụng đó được gói đúng đắn, gồm cả tập tin AndroidManifest.xml.

Có thể phát triển các ứng dụng Android mà không cần Eclipse và trình cài thêm Android Developer Tools, nhưng bạn sẽ cần phải biết cách sử dụng Bộ công cụ phát triển phần mềm Android.

Bộ công cụ phát triển phần mềm Android được phân phối dưới dạng một tệp tin nén ZIP và giải nén vào một thư mục trên ổ đĩa của bạn. Do có một vài cập nhật Bộ công cụ phát triển phần mềm (SDK), bạn nên giữ cho môi trường phát triển của bạn có tổ chức tốt để bạn có thể dễ dàng chuyển đổi giữa các bản cài đặt SDK. SDK gồm có:

android.jar

Tệp tin lưu trữ Java chứa toàn bộ các lớp SDK Android cần thiết để xây dựng ứng dụng của bạn.

documentation.html và thư mục docs

Tài liệu SDK được cung cấp tại chỗ và trên Web. Chủ yếu nó ở dạng JavaDocs, tạo điều kiện dễ dàng cho việc dẫn hướng trong nhiều gói trong SDK. Tài liệu cũng gồm cả một Hướng dẫn Phát triển mức cao và các liên kết đến cộng đồng Android rộng rãi hơn.

Thư mục samples (các ví dụ mẫu)

Thư mục con samples chứa bộ mã nguồn đầy đủ dùng cho một loạt các ứng dụng, gồm cả ApiDemo, nó sử dụng nhiều API. Ứng dụng mẫu là một nơi khám phá tuyệt vời khi bắt đầu phát triển ứng dụng Android.

Thư mục tools (các công cụ)

Chứa toàn bộ các công cụ dòng lệnh để xây dựng nên các ứng dụng Android. Công cụ được dùng phổ biến và hữu ích nhất là tiện ích `adb` (Android Debug Bridge).

usb_driver

Thư mục chứa các trình điều khiển cần thiết để kết nối môi trường phát triển với một thiết bị có khả năng chạy Android, chẳng hạn như G1 hay điện thoại phát triển đã được mở khóa Android Dev 1. Các tệp tin này là đòi hỏi duy nhất đối với các nhà phát triển đang sử dụng nền Windows.

Các ứng dụng Android có thể được chạy trên một thiết bị thật sự hoặc trên trình mô phỏng Android (Android Emulator), gửi kèm với SDK Android. Hình 3 cho thấy màn hình chủ của trình mô phỏng Android.

Hình 3. Trình mô phỏng Android



Android Debug Bridge

Tiện ích `adb` hỗ trợ một số đối số dòng lệnh tùy chọn, mang lại các tính năng mạnh mẽ, như sao chép tập tin đến và từ thiết bị. Đối số dòng lệnh `shell` cho phép bạn kết nối với chính điện thoại và đưa ra các lệnh `shell` thô sơ. Hình 4 cho thấy lệnh trình `shell adb` đối với một thiết bị thực được kết nối với một máy tính xách tay Windows bằng một cáp USB.

Hình 4. Sử dụng lệnh `shell adb`

```

C:\WINDOWS\system32\cmd.exe
M:\tools>adb -d shell
$
$ netcfg
netcfg
lo UP 127.0.0.1 255.0.0.0 0x00000049
dummy0 DOWN 0.0.0.0 0.0.0.0 0x00000082
rnnet0 DOWN 25.1.184.133 255.255.255.252 0x00001002
rnnet1 DOWN 0.0.0.0 0.0.0.0 0x00001002
rnnet2 DOWN 0.0.0.0 0.0.0.0 0x00001002
tiwlan0 UP 192.168.2.105 255.255.255.0 0x00001043
$
$ echo $PATH
echo $PATH
/sbin:/system/sbin:/system/bin:/system/xbin
$
$ su
su
#
# cd /data/app
cd /data/app
#
# ls -l
ls -l
-rw-r--r-- system system 8615 2009-03-22 18:38 com.nsi.flashlight.apk
#
# ping google.com
ping google.com
PING google.com (74.125.45.100) 56(84) bytes of data:
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=1 ttl=241 time=99.3 ms
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=2 ttl=241 time=110 ms
64 bytes from yx-in-f100.google.com (74.125.45.100): icmp_seq=3 ttl=241 time=126 ms
^C
M:\tools>_

```

Trong môi trường shell này, bạn có thể:

- Hiện thị cấu hình mạng cho thấy nhiều kết nối mạng. Chú ý các kết nối mạng:
 - lo là kết nối cục bộ hoặc kết nối vòng trở lại.
 - tiwlan0 là kết nối WiFi với một địa chỉ cung cấp bởi một máy chủ DHCP cục bộ.
- Hiện thị nội dung của biến môi trường PATH.
- Thực hiện lệnh su để trở thành siêu người sử dụng.
- Chuyển thư mục đến /data/app, nơi lưu trữ các ứng dụng của người sử dụng.
- Thực hiện liệt kê nội dung thư mục, ở đây bạn sẽ nhìn thấy chỉ một ứng dụng đơn lẻ. Các tệp tin ứng dụng Android thực tế là các tệp tin lưu trữ có thể xem nội dung bằng WinZip hoặc công cụ tương đương. Phần mở rộng là apk.
- Phát lệnh ping để xem Google.com đã sẵn có chưa.

Từ cùng một môi trường dấu nhắc lệnh này, bạn cũng có thể tương tác với các cơ sở dữ liệu SQLite, khởi động chương trình, và nhiều nhiệm vụ mức hệ thống khác. Đây là một chức năng khá nổi bật, nếu lưu ý rằng bạn đã chỉ kết nối với một điện thoại.

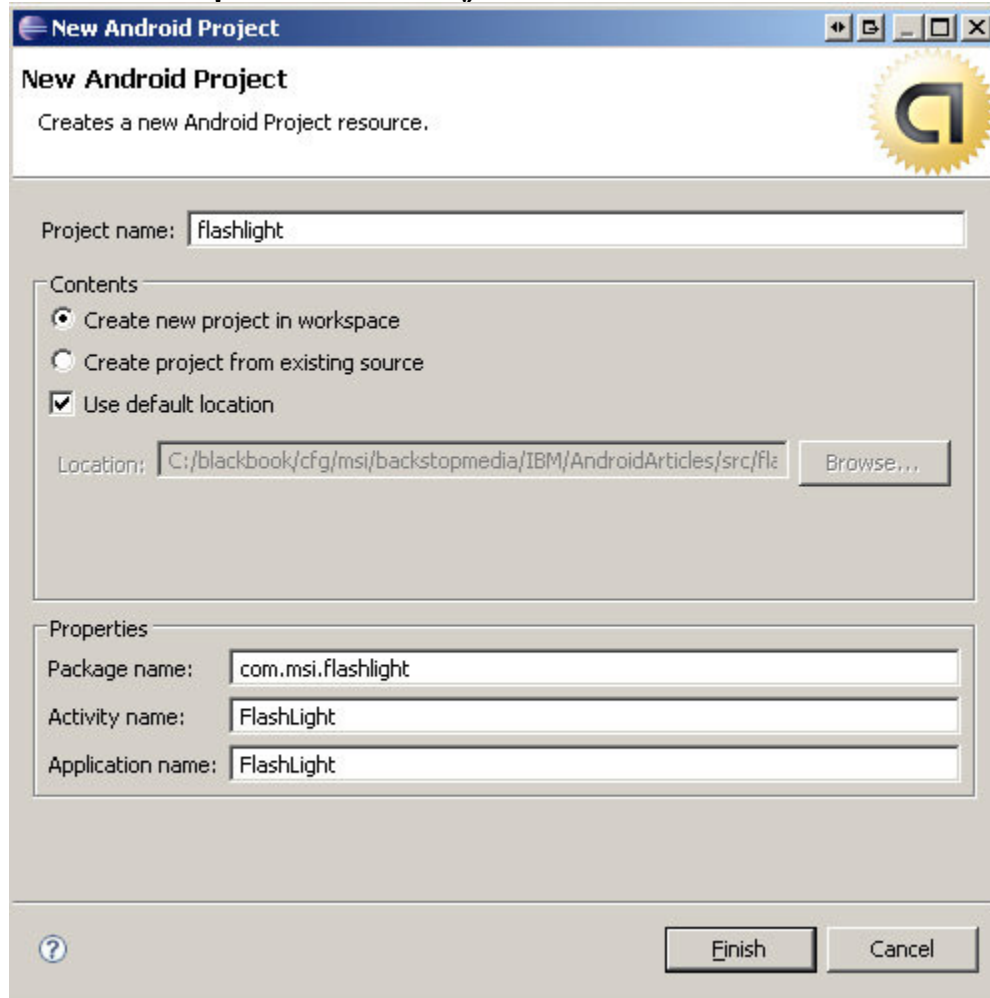
Trong phần tiếp theo, bạn sẽ tạo ra một ứng dụng đơn giản Android.

Viết mã một ứng dụng cơ bản

Phần này cung cấp một vòng tham quan nhanh như con lốc về cách xây dựng một ứng dụng Android. Ứng dụng mẫu đơn giản như bạn có thể hình dung: một ứng dụng “Hello Android” đã cải biên. Bạn sẽ bổ sung một sửa đổi nhỏ để làm cho toàn bộ màu nền của màn hình thành màu trắng để bạn có thể sử dụng điện thoại như một đèn pin. Không thật độc đáo, nhưng nó sẽ bổ ích như một ví dụ. Hãy tải về bộ mã nguồn đầy đủ.

Để tạo ra một ứng dụng trong Eclipse, chọn **File > New > Android project**, nó khởi động thủ thuật New Android Project (Dự án Android mới).

Hình 5. Thủ thuật New Android Project



Tiếp theo, bạn tạo ra một ứng dụng đơn giản chỉ có một hoạt động đơn lẻ, cùng với một bài trí giao diện người sử dụng lưu trong main.xml. Bài trí này chứa một phần tử văn bản mà bạn sắp sửa đổi thành Android FlashLight (đèn pin Android). Bài trí đơn giản được trình bày như dưới đây.

Liệt kê 1. Bài trí Flashlight

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/all_white">
    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" android:textColor="@color/all_black"
        android:gravity="center_horizontal"/>
</LinearLayout>
```

Tạo một vài nguồn màu sắc trong strings.xml.

Liệt kê 2. Màu sắc trong strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="hello">Android FlashLight</string>
    <string name="app_name">FlashLight</string>
    <color name="all_white">#FFFFFF</color>
    <color name="all_black">#000000</color>
</resources>
```

Bài trí màn hình chính có một màu nền được định nghĩa là `all_white`. Trong tệp tin `strings.xml`, bạn thấy rằng `all_white` được định nghĩa như là một bộ ba giá trị RGB là `#FFFFFF`, hay là hoàn toàn màu trắng.

Bài trí này chứa một `TextView`, (khung nhìn văn bản) đơn lẻ, đó thật ra chỉ là một đoạn văn bản tĩnh, không biên tập được. Văn bản này được đặt thành màu đen và căn chính giữa theo chiều ngang với thuộc tính `gravity`.

Ứng dụng có một tệp tin nguồn Java tên là `FlashLight.java`, như chỉ ra dưới đây.

Liệt kê 3. Flashlight.java

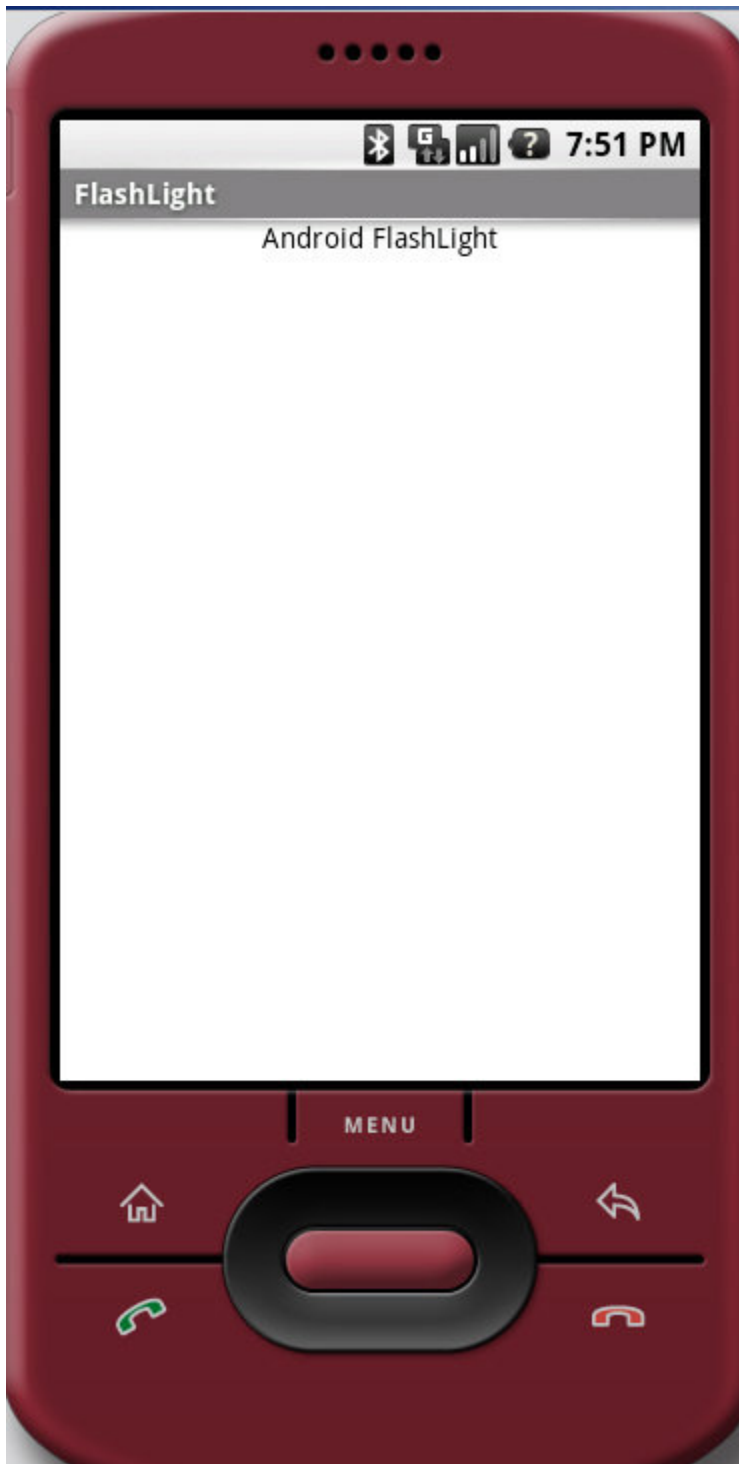
```
package com.msi.flashlight;
import android.app.Activity;
import android.os.Bundle;
public class FlashLight extends Activity {
    /** Called when the activity is first created. */
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
    }
}
```

Mã này là một văn bản thảo sẵn (boiler-plate) trực tiếp từ thủ thuật New Project:

- Nó là một phần của gói Java tên là `com.msi.flashlight`.
- Nó có hai mục nhập khẩu vào:
 - Một là của lớp `activity` (hoạt động)
 - Một là của lớp `bundle` (bó)
- Khi hoạt động này được khởi tạo, phương thức `onCreate` được gọi ra, nó chuyển giao tham đối là một (bó) `savedInstanceState`. Đừng quan tâm đến bó này đối với mục đích của chúng ta, nó được sử dụng khi một hoạt động bị treo và sau đó tiếp tục lại.
- Phương thức `onCreate` là nạm chồng đề lên phương thức của lớp `activity` có cùng tên gọi. Nó gọi phương thức `onCreate` của lớp bậc trên.
- Một lệnh gọi `setContentView()` sẽ gắn bài trí giao diện người sử dụng đã định nghĩa trong tệp tin `main.xml`. Bất cứ thứ gì trong `main.xml` và `strings.xml` đều được tự động ánh xạ tới các hằng đã định nghĩa trong tệp tin nguồn `R.java`. Đừng bao giờ chỉnh sửa tệp tin này trực tiếp, vì nó được thay đổi theo từng lần xây dựng.

Việc chạy ứng dụng sẽ trình diễn một màn hình màu trắng với văn bản màu đen.

Hình 6. Màn hình trắng của flashlight



Thiết lập tập tin AndroidManifest.xml dành cho ứng dụng FlashLight trình bày dưới đây.

Liệt kê 4. AndroidManifest.xml dùng cho FlashLight

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.msi.flashlight"
    android:versionCode="1"
    android:versionName="1.0.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        <activity android:name=".FlashLight"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>
```

```
</activity>
</application>
</manifest>
```

Tệp tin này được tạo ra một cách tự động bởi trình cắm thêm Android Developer Tools cho Eclipse. Bạn không phải làm gì hết.

Tất nhiên, ứng dụng này không cực kỳ tuyệt vời. Nhưng nó có thể tiện dụng nếu bạn muốn đọc một ít mà không phiền nhiễu giấc ngủ của vợ/chồng bạn, hoặc nếu bạn cần tìm lối đi đến hộp cầu chì trong tầng hầm trong khi mất điện.

Tóm tắt

Trong bài này, bạn đã tìm hiểu về Android ở mức rất cao và đã xây dựng nên một ứng dụng nhỏ. Hy vọng rằng ví dụ đã làm cho bạn có được sự hứng khởi đủ để khám phá thêm nền Android. Android hứa hẹn là một nền tảng mã nguồn mở làm chuyển động thị trường mà sẽ hữu ích cả ngoài phạm vi điện thoại di động.

Tải về

Mô tả	Tên	Kích thước	Phương thức tải
FlashLight source code	os-android-devel-FlashLight.zip	22KB	<u>HTTP</u>

→ [Thông tin về phương thức tải](#)

Tài nguyên

Học tập

- [Open Handset Alliance](#) (Liên minh Thiết bị cầm tay mở) là một tập đoàn 47 công ty công nghệ và điện thoại di động đã tập hợp nhau lại để tăng tốc đổi mới về điện thoại di động mang lại cho người tiêu dùng một trải nghiệm về điện thoại di động phong phú hơn, ít tốn kém hơn, và tốt hơn. Cùng với nhau, họ đã phát triển Android, nền di động hoàn chỉnh, mở và miễn phí đầu tiên.
- [Android developers site](#) (trang Web của các nhà phát triển Android) cung cấp các tài liệu, phần tải về, blog, và nhiều thứ khác nữa.
- Tìm hiểu nhiều hơn về [Máy ảo Dalvik](#).
- Xem [các bài hướng dẫn](#) có trên máy chủ YouTube bàn luận về các chi tiết bên trong của Máy ảo Dalvik.
- [Unlocking Android: A Developer's Guide \(Mở khóa Android: Một Hướng dẫn của Nhà Phát triển\)](#) cung cấp chỉ dẫn súc tích, thực tiễn về hệ điều hành và các công cụ phát triển Android.
- Để nghe các phỏng vấn và thảo luận thú vị đối với các nhà phát triển phần mềm, xem [developerWorks podcasts](#).
- Theo sát [các buổi phát tin trên mạng và sự kiện kỹ thuật](#) của developerWorks.
- Theo dõi [developerWorks trên Twitter](#).
- Xem các hội nghị, trưng bày thương mại, các buổi phát tin trên mạng, và [Events](#) các sự kiện khác trên khắp thế giới sắp diễn ra mà các nhà phát triển mã nguồn mở của IBM quan tâm đến.

- Truy cập vào vùng mã nguồn mở trên developerWorks để có các thông tin hướng dẫn cách làm, các công cụ, và các dự án cập nhật để giúp bạn phát triển với các công nghệ mã nguồn mở và sử dụng chúng với các sản phẩm của IBM.
- Theo dõi và tìm hiểu về IBM và các công nghệ mã nguồn mở và các chức năng của sản phẩm bằng các trình diễn miễn phí theo yêu cầu trên developerWorks (developerWorks On demand demos).

Lấy sản phẩm và công nghệ

- Tải về Android SDK.
- Nhận bản Eclipse IDE mới nhất.
- Đổi mới dự án phát triển mã nguồn mở tiếp theo của bạn bằng phần mềm dùng thử của IBM (IBM trial software), sẵn có để tải về hoặc trên đĩa DVD.
- Tải về các phiên bản đánh giá sản phẩm của IBM (IBM product evaluation versions) hoặc khám phá dùng thử trực tuyến trong SOA Sandbox của IBM và hãy bắt tay vào các công cụ phát triển ứng dụng và các sản phẩm phần mềm tăng giữa từ DB2®, Lotus®, Rational®, Tivoli®, và WebSphere®.

Thảo luận

- Tham gia vào các blog của developerWorks và gia nhập cộng đồng developerWorks.

Đôi nét về tác giả

Sau khi sự nghiệp trong đội bóng rổ của trường cao đẳng kết thúc mà không có một hợp đồng dài hạn nhiều năm chơi cho đội LA Lakers, Frank Ableson đã chuyển chỉ hướng của mình sang thiết kế phần mềm máy tính. Ông thích giải quyết các vấn đề phức tạp, nhất là trong các lĩnh vực truyền thông và lập giao diện phần cứng. Khi không làm việc, ông dành thời gian với người vợ Nikki và con cái. Bạn có thể gặp Frank tại địa chỉ frank@cfigsolutions.com.