http://www.google.co.kr/search?hl=ko&lr=lang\_ko&newwindow=1&rlz=1B3MOZA\_koKR384KR384&tbs=lr%3Alang\_1ko&q=%EA%B5%AC%EA%B8%80+%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C+%EA%B0%9C%EB%B0%9C%ED%99%98%EA%B2%BD+%EA%B5%AC%EC%B6%95&aq=f&aqi=&aql=&oq=&gs\_rfai=

http://wiz.pe.kr/640

http://developer.android.com/sdk/eclipse-adt.html

http://developer.android.com/guide/developing/eclipse-adt.html

http://jpub.tistory.com/32

http://www.androidpub.com/89895

http://www.androidpub.com/?mid=android\_dev\_info&category=127161&page=2&document\_srl=588

http://developer.android.com/sdk/installing.html

1. JDK (Java Development Kit) Install

다 운로드 사이트에서 "JDK6"를 다운로드 후 설치합니다. (JRE만 설치해도 실행은 되지만 JDK를 설치 하는 것이 좋을 것 같네요.)

java.sun.com

2. EClipse Install

url : http://www.eclipse.org/downloads/

1. download : http://www.eclipse.org/downloads/download.php?file=/eclipse/downloads/drops/R-3.5.2-201002111343/eclipse-SDK-3.5.2-win32.zip

2. uncompress eclipse-SDK-3.5.2-win32.zip

location : C:\WEB\_APP\eclipse

3. Android SDK Install

1. download : http://developer.android.com/sdk/index.html

http://developer.android.com/sdk/installing.html

2. uncompress android-sdk\_r06-windows.zip

location : C:\WEB\_APP\ADT

3. ADD PATH

%PATH% = %PATH%;C:\WEB\_APP\ADT\android-sdk

3. Installing the ADT Plugin ( http://developer.android.com/sdk/eclipse-adt.html )

1. Start Eclipse, then select Help > Install New Software.

\* first time : workspace setting : C:\WEB\_APP\eclipse\_workspace

2. In the Available Software dialog, click Add....

3. In the Add Site dialog that appears, enter a name for the remote site (for example, "Android Plugin") in the "Name" field.

In the "Location" field, enter this URL:

https://dl-ssl.google.com/android/eclipse

Note: If you have trouble acquiring the plugin, you can try using "http" in the URL,

instead of "https" (https is preferred for security reasons).

Click OK.

4. Back in the Available Software view,

you should now see "Developer Tools" added to the list.

Select the checkbox next to Developer Tools,

which will automatically select the nested tools Android DDMS

and Android Development Tools. Click Next.

5. In the resulting Install Details dialog,

the Android DDMS and Android Development Tools features are listed.

Click Next to read and accept the license agreement and install any dependencies,

then click Finish.

6. Restart Eclipse.

4. Configuring the ADT Plugin

Once you've successfully downnloaded ADT as described above,

the next step is to modify your ADT preferences in Eclipse to point to the Android SDK directory:

1. Select Window > Preferences... to open the Preferences panel (Mac OS X: Eclipse > Preferences).

2. Select Android from the left panel.

3. For the SDK Location in the main panel, click Browse... and locate your downloaded SDK directory.

4. Click Apply, then OK.

-------- -------- -------- -------- -------- -------- -------- -------- -------- -------- -------- --------

# Creating an Android Project

The ADT plugin provides a New Project Wizard that you can use to quickly create a new Android project (or a project from existing code). To create a new project:

1. Select File > New > Project.

2. Select Android > Android Project, and click Next.

3. Select the contents for the project:

\* Enter a Project Name. This will be the name of the folder where your project is created.

\* Under Contents, select Create new project in workspace. Select your project workspace location.

\* Under Target, select an Android target to be used as the project's Build Target. The Build Target specifies which Android platform you'd like your application built against.

Unless you know that you'll be using new APIs introduced in the latest SDK, you should select a target with the lowest platform version possible.

Note: You can change your the Build Target for your project at any time: Right-click the project in the Package Explorer, select Properties, select Android and then check the desired Project Target.

\* Under Properties, fill in all necessary fields.

o Enter an Application name. This is the human-readable title for your application — the name that will appear on the Android device.

o Enter a Package name. This is the package namespace (following the same rules as for packages in the Java programming language) where all your source code will reside.

o Select Create Activity (optional, of course, but common) and enter a name for your main Activity class.

o Enter a Min SDK Version. This is an integer that indicates the minimum API Level required to properly run your application. Entering this here automatically sets the minSdkVersion attribute in the <uses-sdk> of your Android Manifest file. If you're unsure of the appropriate API Level to use, copy the API Level listed for the Build Target you selected in the Target tab.

4. Click Finish.

Tip: You can also start the New Project Wizard from the New icon in the toolbar.

Once you complete the New Project Wizard, ADT creates the following folders and files in your new project:

src/

Includes your stub Activity Java file. All other Java files for your application go here.

<Android Version>/ (e.g., Android 1.1/)

Includes the android.jar file that your application will build against. This is determined by the build target that you have chosen in the New Project Wizard.

gen/

This contains the Java files generated by ADT, such as your R.java file and interfaces created from AIDL files.

assets/

This is empty. You can use it to store raw asset files.

res/

A folder for your application resources, such as drawable files, layout files, string values, etc. See Application Resources.

AndroidManifest.xml

The Android Manifest for your project. See The AndroidManifest.xml File.

default.properties

This file contains project settings, such as the build target. This files is integral to the project, as such, it should be maintained in a Source Revision Control system. It should never be edited manually — to edit project properties, right-click the project folder and select "Properties".

# Creating an AVD

An Android Virtual Device (AVD) is a device configuration for the emulator that allows you to model real world devices.

In order to run an instance of the emulator, you must create an AVD.

To create an AVD from Eclipse:

1. Select Window > Android SDK and AVD Manager, or click the Android SDK and AVD Manager icon in the Eclipse toolbar.

2. In the Virtual Devices panel, you'll see a list of existing AVDs. Click New to create a new AVD.

3. Fill in the details for the AVD.

Give it a name, a platform target, an SD card size, and a skin (HVGA is default).

Note: Be sure to define a target for your AVD that satisfies your application's Build Target (the AVD platform target must have an API Level equal to or greater than the API Level that your application compiles against).

4. Click Create AVD.

Your AVD is now ready and you can either close the SDK and AVD Manager, create more AVDs, or launch an emulator with the AVD by selecting a device and clicking Start.

For more information about AVDs, read the Android Virtual Devices documentation.

----------- ----------- ----------- ----------- ----------- ----------- -----------

서울 위도 경도

위도(Latitude): 북위 37도

경도(Longitude): 동경 126도

Latitude (위도) : 37°30'56.76"N

Longitude (경도) : 126°54'27.71"E

http://graynote.tistory.com/entry/%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9CAndroid-SDK-%EA%B0%9C%EB%B0%9C-%EA%B8%B0%EB%B3%B8-%EA%B0%95%EC%A2%8C

# 하나의 안드로이드 응용프로그램은 과제(task)들의 집합이다.

응용프로그램이 수행하는 과제를 안드로이드에서는 활동(Activity)라고 부른다.

하나의 활동은 응용프로그램이 수행하는 고유한 하나의 과제 또는 작업을 뜻한다.

# Method

onCreate -> onStart -> onResume

onPause -> onStop -> onDestroy

onRestart -> onStart

# 활동의 필수적이지 않은 상태를 onSaveInstanceState에서 Bundle에 저장

메모리가 부족해서 운영체제가 활동을 죽이는 경우가 많다면, onSaveInstanceState()메소드 콜백시

Bundle 객체에 상태 정보를 저장하는 것도 한 가지 방법이 된다. 그러나 onSaveInstanceState()가

모든 상황에서 반드시 호출된다는 보장은 없으므로, 응용프로그램의 실행에 필수적인 자료들은

onPause() 메서드에서 저장하는것이 안전하다.

- onSaveInstanceState() 메서드는 아직 제출되지 않은 필드 자료라던가 사용자에 대한

응용프로그램의 반응성을 높이기 위한 상태 정보등 필수적이지 않은 정보를 저장하는 곳으로 적합하다.

이후 사용자가 해당 활동을 다시 활성화하면 그 Bundle이 onCreate() 메서드를 전달된다.

이를 이용해서 활동은 일시정지 전의 상태를 정확히 복원할 수 있다. 또한 onStart() 콜백 메서드 이후에

onRestoreInstanceState() 콜백을 이용해서 이 Bundle의 정보를 읽는 것도 가능하다.

- 안드로이드 운영체제가 활동을 죽인 경우 해당 Activity의 isFinishing()메서드가 false를 돌려준다.

이 메서드를 onPause() 메서드에서 사용할 수 있으나, 그 시점에서 이 메서드가 true를 돌려준다고 해도

이후에 onStop()에서 활동이 죽을 수 있음을 주의해야 한다.

# 전이

한 할동에서 다른 활동으로 넘어가는 것을 활동의 전이(transition)라고 부른다.

startActivity() , finish() 메서드를 이용

- 일시적인 활동 전이

한 자식 활동이 대화상자를 표시한 후 원래의 활동으로 돌아가는 것

무모활동은 "자식" 활동을 띄운 후 결과를 기다린다.

-> startActivityForResult(), onActivityResult()

# AndroidManifest.xml : 응용프로그램의 신원(identity)에 대한 중요한 정보를 담는다.

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

package="net.softm.sample"

android:versionCode="1"

android:versionName="1.0">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

<activity android:name=".MainMenuActivity"

android:label="@string/app\_name">

<intent-filter>

<action android:name="android.intent.action.MAIN" />

<category android:name="android.intent.category.LAUNCHER" />

</intent-filter>

</activity>

<activity android:name="AudioActivity"/>

<activity android:name="MovingActivity"/>

<activity android:name="StillActivity"/>

</application>

<uses-permission android:name="android.permission.RECORD\_AUDIO" />

<uses-permission android:name="android.permission.SET\_WALLPAPER" />

<uses-permission android:name="android.permission.CAMERA" />

<uses-permission android:name="android.permission.WRITE\_SETTINGS" />

<uses-sdk android:minSdkVersion="8" />

</manifest>

- @string/app\_name : string.xml에 선언된 app\_name값을 지칭한다.

- @drawable/icon : /res/drawable/iconf

- 응용프로그램은 네개의 활동들로 구성된다

1. MainMenuActivity

2. AudioActivity

3. MovingActivity

4. StillActivity

- 응용프로그램의 주 진입점 : MainMenuActivity

- 권한

- 오디오 녹음(RECORD\_AUDIO)

- 기기의 바탕화면 설정(SET\_WALLPAPER)

- 내장카메라 접근(CAMERA)

- 설정 쓰기(변경)(WRITE\_SETTINGS)

.

.

.

권한들은 다음과 같은 여러 지점들에서 시행(점검 및 적용)된다.

- 활동이나 서비스를 시작할 때

- 콘텐트 제공자가 제공하는 자료에 접근할 때

- 함수 호출 수준에서

- Intent 겍체로 방송을 보내거나 받을때

권한의 기보적인 보호 수준으로는

- 보통 ( normal )

- 위험 ( dangerous )

- 서명 ( signature )

# 그 외의 의도 필터 설정

Intent 객체로 대표되는 "의도"들중에는 특정 활동이나 구성요소를 지정하지 않는 것들도 있다.

운영체제는 그런 "암묵적" 의도들의 처리를 의도 필터에 기초해서 결정한다.

의도 필터는 활동과 서비스들, 방송 수신자들에 적용된다. 의도 필터는 안드로이드 운영체제가 보낸의도들 중 해당

구성요서가 받아들이는 것이 어떤것인지를 정의한다.

<intent-filter>

<action android:name="android.intent.action.VIEW" />

<action android:name="android.intent.category.BROWSABLE"/>

<action android:name="android.intent.category.DEFAULT"/>

<data android:scheme="geoname"/>

</intent-filter>

이 의도 필터는 VIEW라는 미리 정의된 작용을 사용한다.

이 작용은 특정 콘텐트의 내용을 표시하는 기능을 한다.

이 필터는 BROWSABLE이라는 범주에 속하며 "geoname"이라는 자료 스킴을 사용한다.

따라서 만일 geoname://로 시작하는 Uri를 가진 의도가 전달되면 이 의도 필터를 가진 활동이 시동된다.

# Uri 수준에서 콘텐트 제공자 권한을 시행

<grant-uri-permissions>요소를 이용하면 Uri 수준에서 권한들을 세밍랗게 시행할 수 있다.

# 입력에 관한 하드웨어/소프트웨어 요구사항 지정

<uses-configuration>이용

<uses-configuration android:reqTouchScreen-"finger"/>

--> 손가락 터치만으로 입력을 받는다

5방향 네비게이션, 하드웨어 키보드, 키보드 종류, 네비게이션 수단( 방향패트, 트랙볼, 휠등) 터치스크린 설정등 당양한 구성특성들있다.

android:reqFiveWayNaN : 5방향 네비게이션 컨트롤을 요구하는지 : true, false

android:reqHardKeyboard : 하드웨어 키보드 요구하는지 : true, false

android:reqKeyboardType : 키보드의 종류 : "undefined" ( 기본값 )

"nokeys" ( 키보드 필요 없음 )

"qwerty" ( 표준 QWERTY 키보드 필수 )

"twelvekey" ( 12키패드(숫자키패드) 필수 )

android:reqNavigation : 네비게이션 장치 종류 : "undefined" ( 기본값 )

"nonav" 네비게이션 장치 필요없음

"dpad" 방향 패트 필수

"trackball" 트랙볼 필수

"wheel" 네비게이션 휠 필수

android:reqTouchScreen : 필요한 터치스크린의 종류 : "undefined" ( 기본값 )

"notouch" 터치스크린 필요 없음

"stylus" 감압식(스타일러스) 터치스크린 필수

"finger" 정정식(손가락) 터치스크린 필수

하나의 특성에서 여러 가지 값들을 "또는"으로 결합할 수 없다.

응용프로그램이 하드웨어 키보드를 요구하며 감압식 터치스크린 "또는" 정전식 터치스키린"을 요구한다면

다음과 같이 <uses-configuration>요소를 두개 만들어야한다.

<use-configuration android:reqHardKeyboard="true" android:reqTouchScreen="finger" />

<use-configuration android:reqHardKeyboard="true" android:reqTouchScreen="stylus" />

# 최소 안드로이드 SDK 버전 지정

기본적으로 안드로이이드 응용프로그램은 안드로이드 SDK의 풀 버전들 모두(1.0부터 최신까지, 베타 버전 제외)와 호환된다고 가정된다.

특정 버전의 안드로이드 SDK가 꼭 필요한 응용프루로그램정의 ( API 버전 을 적용합니다 )

<use-sdk android:minSdkVersion="2" />

# 다른 패키지 링크

모든 안드로이드 응용프로그램은 기본적으로 표준 안드로이드 패키지들(android.app 등)에 링크되며,자신의 패키지를 인식한다.

응용프로그램이 다른 추가적인 패키지들과 링크된다면 각 패키지를 안드로이드 매니페스트 파일의

<application> 요소안에 <uses-library>요소로 등록해야 한다.

다음은 그러한 <use-library>요소의 예이다.

<use-library android:name="com.mylibrary.stuff" />

# 그 외의 응용프로그램 구성 설정들

- <application>요소의 여러 특성들을 통해서 응용프로그램 전역 테마들을 설정할 수 있다.

- <instrumentation>요소를 통해서 응용프로그램의 계장 속성들을 설정할 수 있다.

# 자원 디렉터리 계통 구조

- /res/drawable/ 그래픽 자원들

- /res/layout/ 사용자 인터페이스 자원들, 위젯들

- /res/values/ 문자열, 색상값 같은 단순 자료들

# 안드로이드 자산 패키징 도구 aapt

aapt (Android Asset Packaging Tool )

R.java를 생성한다.

# 자원의 형식

텍스트 문자열, 그래픽 이미지, 사용자 인터페이스 디자인을 위한 색상 스킴 등 등, 안드로이드 응용프로그램이 사용할 수 있는 자원의 형식은 다양하다.

/res 하위 디렉터리에 저장됨

자원들이 저장되는 디렉터리 이름과 파일 이름에는 엄격한 규칙이 적용된다

예를 들어 모든 자원 파일은 이름이 영문 소문자와 숫자, 밑줄로만 이루어져야한다.

문자열 /res/values/ strings.xml(권장됨) <string>

문자열 배열 /res/values/ arrays.xml(권장됨) <string-array>

색상값 /res/values/ colors.xml(권장됨) <color>

크기(dimension) /res/values/ dimens.xml(권장됨) <dimen>

단순표시물(drawable) /res/values/ drawables.xml(권장됨) <drawable>

비트그래픽 /res/values/ 예) img.png, img.jpg, red.xml 지원되는 그래픽 파일 또는 도형 등의 표시물을 정의하는 XML 파일들

애니메이션시퀀스(트위닝) /res/anim/ fancy\_anim1.xml <set>,<alpha>,<scale>,<translate>,<rotate>

메뉴파일 /res/menu/ my\_menu1.xml, more\_options.xml <menu>

XML 파일 /res/xml/ some.sml, more.xml 개발자가 정의

원본(raw)파일 /res/raw/ some\_audio.mp3, some\_text.txt

레이아웃 파일 /res/layout/ start\_screen.xml, main\_screen.xml 다양함, 반드시 레이아웃 요소이어야함

스타일 및 테마 /res/values/ styles.xml, themes.xml(권장됨) <style>

# 자원 형식에 따른 자원 저장방식

안드로이드 자산 패키징 도구(aapt)는 /res 디렉토리 계통구조에 있는 적절한 형식의 자원 파일들을 모두 파악해서,

그 자원들에 접근하는 데 필요한 변수 정의를 담은 클래스 파일 R.java를 만들어 낸다.

여러 형식의 자원들을 실제로 저장하고 사용하는 예들이 잠시 후에 나온다. 일단 지금은 자원의 형식에 따른 여러 자원 저장 방식을 살펴보자.

- 문자열 색삭 등의 단순 자원

<?xml version="1.0" encoding="utf-8"?>

이 선언 다음에는 루트 노드인 <resources>가 오고, 그 다음에는 자원의 형식에 따라 <string>이나 <color>등이 온다.

# 코드에서 자원에 접근

String myString = getResource().getString(R.string.hello);

--> getString()메소드는 Resource 클래스(android.content.res.Resource)에 정의됨

# Eclipse에서 단순 자원값을 설정

- strings.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<string name="hello">Hello World, ResouceRoundup!</string>

<string name="app\_name">ResouceRoundup</string>

<color name="prettyTextColor">#FF0000</color>

<dimen name="textPointSize">14pt</dimen>

<drawable name="redDrawable">#F00</drawable>

<string-array name="flavors">

<item>바닐라</item>

<item>초코</item>

<item>딸기</item>

</string-array>

</resources>

- ResouceRoundup.java

package net.softm.sample;

import android.app.Activity;

import android.graphics.drawable.ColorDrawable;

import android.os.Bundle;

public class ResouceRoundup extends Activity {

String myResourceString = getResources().getString(R.string.hello);

int myColor = getResources().getColor(R.color.prettyTextColor);

float myDimen = getResources().getDimension(R.dimen.textPointSize);

ColorDrawable myDraw = (ColorDrawable)getResources().getDrawable(R.drawable.redDrawable);

String[] aFlvors = getResources().getStringArray(R.array.flavors);

/\*\* Called when the activity is first created. \*/

@Override

public void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.main);

}

}

# 자원 다루기

- 문자열 자원 다루기

java의 String형식에 해당하는 문자열 자원은 아마도 가장 간단한 자원일 것이다.

문자열 자원의 용도로는 폼 뷰의 텍스트 이름표나 도움말 텍스트 등을 들 수 있따.

응용 프로그램 이름 역시 기본적으로는 하나의 문자열 자원으로 자장된다.

문자열 자원은 /res/values 디렉터리의 XML 파일에 저장해야 한다.

그 XML 파일은 빌드 시점에서 컴파일되어서 응용프로그램 패키지에 포함된다.

문자열 안에 작은 따옴표가 있는 경우에는 그것을 역슬래시로 탈출(escape)시키거나,

아니면 문자열 전체를 큰따옴표로 감싸야 한다.

- 문자열에 볼드, 이탤릭, 밑줄 적용

문자열 자원은 문자 스타일 지정을 위한 HTML 요소 세개를 지원한다.

볼드, 이탤릭, 밑줄을 위한 <b>,<i>,<u>

<string name="txt"><b>볼드</b><i>이탤릭</i><u>밑줄</u></string>

# 문자열 자원을 서식 문자열로 사용

문자열의 일부분이 다른 값들로 치환되는 서식 문자열(format string)을 문자열 자원으로 만들 수 도 있다.

예를 들어 다음은 점수와 "승리" 또는 "패배"를 표시하기 위한 서식 문자열이다.

<string name="winLose">점수: %1$d점!(총 %2$d점 중) 당신의 %3$s.</string>

그런데 이런 서식 문자열에 볼드나 이탤릭,밑줄을 적용하려면 해당 HTML요소의 태그를 HTML개체(entity) 형태로 탈출 시켜야한다.

<string name="winLose">점수: %1$d점!(총 %2$d점 중) 당신의 &lt;i&gt;%3$s&lt;/i&gt;.</string>

# 코드에서 문자열 자원 사용하기

String mySimpleWinString;

mySimpleWinString = getResources().getString(R.string.winLose);

String escapedWin = TextUtils.htmlEncode("승리");

String resultText = String.format(mySimpleWinString, 5,5,escapedWin);

import android.text.HTML;

import android.text.TextUtils;

...

String myStryledWinString;

myStyledWinString = getResources().getString(R.string.winLoseStyled);

String escapedWin = TextUtils.htmlEncode("승리");

String resultText = String.format(mystyledWinString, 5,5,escapedWin);

CharSequence styledResults = Html.fromHtml(resultText);

# 문자열 배열 다루기

String[] aFlavors = getResources().getStringArray(R.array.flavors);

# 색상 다루기

- #RGB ( 예:#F00, 12비트 빨강)

- #ARGB ( 예:#8F00, 12비트 알파 50% 빨강 )

- #RRGGBB ( 예: #FF00FF, 24비트 자홍색 )

- #AARRGGBB ( 예:#80FF00FF, 24비트 알파 50% 자홍색 )

자원 XML파일에서 색상 값을 정의하는 요소는 <color>이다.

문자열 자원에서처럼 name특성으로 자원이름을 요소의 내용으로 색상의 값을 지정한다.

다음은 간단한 색상 자원 파일(/res/values/colors.xml)의 예이다.

<?xml version="1.0" encoding="utf-8"?>

<resources>

<color name="background\_color">#006400</color>

<color name="text\_color">#FFE4C4</color>

</resources>

color 자원은 int형으로 값을 반환한다.

int myResourceColor = getResources().getColor(R.color.prettyTextColor);

# 크기 다루기

텍스트 컨트롤아니 버튼 등의 사용자 인터페이스 위젯들은 특정한 크기로 그려진다.

그러한 크기(dimension)들도 자원으로 저장할 수 있다. 크기 값에는 항상 측정 단위를 뜻하는 접미사가 붙는다.

크기 자원들 역시 프로젝트의 /res/values 디렉토리의 XML파일에 저장해야한다.

그 XML파일은 빌드 시점에서 컴파일되어서 응용프로그램 패키지에 포함된다.

자원 XML파일에서 크기 자원을 정의하는 요소는 <dime>이다

- 안드로이드 SDK가 지원하는 자원의 단위

픽셀 실제 화면 픽셀 px 20px

인치 물리적 길이 in 1in

밀리미터 물리적 길이 mm 1mm

포인트 흔히 쓰이는 글자 크기 단위 pt 14pt

밀도 독립적 픽셀 160dpi 화면을 기준으로한 픽셀단위 dp 1dp

축적 독립적 픽셀 가변 글꼭 표시에 최적임 sp 14sp

- /res/values/dimens.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<dimen name="FoureenPt">14pt</dimen>

<dimen name="OneTnch">1in</dimen>

<dimen name="TenMillimeters">10mm</dimen>

<dimen name="TenPixels">10px</dimen>

</resources>

- float myDimension = getResources().getDimension(R.dimen.textPointSize);

# 단순 표시물 다루기

표시물(drawable) 자원 형식을 이용해서 간단한 직사각형을 정의하고, 그것을 다른 화면 요소들에

적용할 수 있다. 이런 단순한 표시물 자원은 직사각형 내부를 채우는 데 사용할 색상 값으로 정의된다는 점에서

색상 자원과 상당히 비슷하다.

단순 표시물 자원들은 프로젝트의 /res/values 디렉토리의 XML파일에 자장해야 한다.

- /res/values/drawables.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<drawable name="red\_rect">#F00</drawable>

</resources>

import android.graphics.drawable.ColorDrawable;

...

ColorDrawable myDraw = (ColorDrawable)getResources().getDrawAble(R.drawable.redDrawable);

# 이미지 다루기

아이콘이나 그래픽 같은 시각적 요소가 흔히 쓰인다.

안드로이드는 여러가지 이미지 형식들을 지원하는데, 그런 형식의 파일들은 응용프로그램용 자원으로서 직접 포함시킬 수 있다.

PNG(Portable Network Graphics) 권장됨(무손실) .png

아홈 조각 확장성 이미지 권장됨(무손실) .9.png

JPG(Joint Photographic Experts Group) 권장되지는 않음(유손실) .jpg, .jpeg

GIF(Graphics Interchange Format) 사용하지 않는 것이 종흠 .gif

# 아홈조각 확장성 이미지 다루기

전화기 화면 크기는 기기에 따라 다양하다.

만약 화면크기마다 개별적인 그래픽 자원을 사용해야 한다면 그래픽 디자이너가 많은 시간을 낭비해야 합니다.

─────┼─────────────┼──────────────┼─────────────┼───

│비례되지않음 │수평만 비례 │비례되지않음 │

│ │ │ │

─────┼─────────────┼──────────────┼─────────────┼───

│ 수직만비례 │수평,수직 비례│ 수직만비례 │

│ │ │ │

─────┼─────────────┼──────────────┼─────────────┼───

│비례되지않음 │ 수평만 비례 │비례되지않음 │

│ │ │ │

─────┼─────────────┼──────────────┼─────────────┼───

draw9patch 도구를 이용해 생성할 수 있음.

# 코드에서 이미지 자원 사용하기

- /res/drawable 디렉토리에 .png 파일을 추가함

import android.widget.ImageView;

...

ImageView flagImageView = (ImageView)findViewById(R.id.ImageView01);

flagImageView.setImageResource(R.drawable.flag);

import android.graphics.drawable.BitmapDrawable;

...

BitmapDrawable bitmapFlag = ( BitmapDrawable ) getResources().getDrawable(R.drawable.flag);

int iBitmapHeightInPixels = bitmapFlag.getIntrinsicHeight();

int iBitmapWidthInPixels = bitmapFlag.getIntrinsicWidth();

import android.graphics.drawable.NinePatchDrawable;

...

NinePatchDrawable stretchy = ( NinePatchDrawable )getResources().getDrawable(R.drawable.pyramid);

int iStretchyHeightInPixels = stretchy.getIntrinsicHeight();

int iStretchyWidthInPixels = stretchy.getIntrinsicWidth();

# 애니메이션 다루기

안드로이드는 애니메이션과 "트위닝(tweening)"을 지원함

일부 애니메이션은 비례(scaling), 페이딩, 회전 등도 사용한다.

그러한 동작들을 동시에 적용하거나 순차적으로 적용할 수 있으며, 각자 다른 보간자(interpolator)들을 적용 가능합니다.

애니메이션 시퀀스는 특정 그래픽 파일에 국한되는 것이 아니므로, 시퀀스를 하나 작성했다면 그것을

여러 그래픽 파일들에 적용할 수 있다.

예를 들어 달, 별, 다이아몬드 그래픽에 하나의 비례 애니메이션 시퀀스를 적용해서 모두 똑같이 고동치게 만들거나,

하나의 회전 시퀀스를 적용해서 동일하게 회전하도록 만들 수 있다.

그래픽 애니메이션 시퀀스 자원은 /res/anim 디렉터리에 특정한 형식의 XML 파일로 저장해야 한다.

그 XML 파일은 빌드 지점에서 컴파일되어 응용프로그램 이진파일에 포함한다.

다음은 간단한 애니메이션 자원 파일(/res/anim/spin.xml)의 예이다.

이 애니메이션은 대상 그래픽을 총 10초 동안 그 자리에서 반시계 방향으로 네 번 회전한다.

<?xml version="1.0" encoding="utf-8" ?>

<set xmlns:android="http://schemas.android.com/apk/res/android"

android:shareInterpolator="false"?

<set>

<rotate

android:fromDegrees="0"

android:toDegrees="-1440"

android:pivotX="50%"

android:pivotY="50%"

android:duration="10000"

/>

</set>

</set>

import android.view.animation.Animation;

import android.view.animation.AnimationUtils;

import android.widget.ImageView;

...

ImageView flagImageView = (ImageView)findViewById(R.id.ImageView01);

flagImageView.setImageResource(R.drawable.flag);

...

Animation an = AnimationUtils.loadAnimation(this, R.anim.spin);

flagImageView.startAnimation(an);

# 메뉴 다루기

메뉴자원도 프로젝트에 포함시킬 수 있다.

애니메이션 자원과 마찬가지로, 메뉴 역시 특정 위젯이나 컨트롤에 국한되지 않으므로 한 메뉴 자원을 여러 UI요소들에

사용하는 것이 가능하다.

각 메뉴 자원은 /res/menu 디렉토리의 XML 파일에 저장된다.

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/start"

android:title="출발!"

android:orderInCategory="1"></item>

<item

android:id="@+id/stop"

android:title="정지!"

android:orderInCategory="4"></item>

<item

android:id="@+id/accel"

android:title="부릉! 가속!"

android:orderInCategory="2"></item>

<item

android:id="@+id/decel"

android:title="제동!"

android:orderInCategory="3"></item>

</menu>

메뉴는 Eclipse용 안드로이드 플러그인의 메뉴 편집기를 이용해서 만들 수 있다.

메뉴 편집기에서도 각 메뉴 항목마다 다양한 설정이 가능하다.

위의 예에서는 각 항목마다 항목의 식별자, 제목(이름표), 표시 순서만 정의했는데,항목의 제목을 문자열로

직접 지정하는 대신 문자열 자원의 식별자를 지정할 수도 있따.

<menu xmlns:android="http://schemas.android.com/apk/res/android">

<item

android:id="@+id/start"

android:title="@string:start"

android:orderInCategory="1"></item>

<item

android:id="@+id/stop"

android:title="@string:stop"

android:orderInCategory="4"></item>

</menu>

public boolean onCreateOptionsMenu(Menu menu) {

getMenuInflater().inflate(R.menu.speed,menu);

return true;

}

# XML파일 다루기

/res/my\_pets.xml

<?xml version="1.0" encoding="utf-8"?>

<pets>

<pet name="Bit" type="Bunny" />

<pet name="Niddle" type="Bunny" />

<pet name="Stack" type="Bunny" />

<pet name="Queue" type="Bunny" />

<pet name="Heap" type="Bunny" />

<pet name="Null" type="Bunny" />

<pet name="Nigiri" type="Fish" />

<pet name="Sashimi" type="Fish" />

<pet name="Kiwi" type="Fish" />

</pets>

import java.io.IOException;

import org.xmlpull.v1.XmlPullParserException;

import android.content.res.XmlResourceParser;

import android.util.Log;

.........

XmlResourceParser myPets = getResources().getXml(R.xml.my\_pets);

int eventType = -1;

while(eventType != XmlResourceParser.END\_DOCUMENT) {

if(eventType == XmlResourceParser.START\_DOCUMENT) {

Log.d(DEBUG\_TAG,"Document Start");

} else if ( eventType == XmlResourceParser.START\_TAG) {

String strName = myPets.getName();

if ( strName.equals("pet") ) {

Log.d(DEBUG\_TAG,"Found a PET");

Log.d(DEBUG\_TAG,"Name : " + myPets.getAttributeValue(null,"name"));

Log.d(DEBUG\_TAG,"Species : " + myPets.getAttributeValue(null,"type"));

}

try {

eventType = myPets.next();

} catch (XmlPullParserException e) {

// TODO Auto-generated catch block

e.printStackTrace();

} catch (IOException e) {

// TODO Auto-generated catch block

e.printStackTrace();

}

}

}

# 원본 파일 다루기

특정한 형식으로 가공되지 않은 원본 파일(raw file)들 역시 프로젝트의 자원으로 사용할 수 있다.

오디오 파일이나 동영상 파일 등, 응용프로그램이 사용해야 하되 안드로이드 자원 패키징 도구(appt)가

직접적으로 지원하지 않는 형식의 파일이라면 모두 지금 말하는 원본 파일에 해당한다.

경로 : /res/raw

# 자원에대한 참조

@[추가적인\_패키지\_이름:]자원형식/변수\_이름

# 레이아웃 다루기

# 스타일 다루기

- /res/value/styles.xml

<?xml version="1.0" encoding="utf-8"?>

<resources>

<style name="mandatory\_text\_field\_style">

<item name="android:textColor">#000000</item>

<item name="android:textSize">14pt</item>

<item name="android:textStyle">bold</item>

</style>

<style name="optional\_text\_field\_style">

<item name="android:textColor">#0F0F0F</item>

<item name="android:textSize">12pt</item>

<item name="android:textStyle">italic</item>

</style>

</resources>

# 테마 다루기

스타일을 이용하는것은 동일하며

AndroidManifest.xml 파일 활동요소(<activity)의 android:theme 를 에 styles.xml을 지정한다.

# 시스템 자원 참조하기 ( platforms/android-1.5/data/res/ )

android.R의 하위 클래스들

1. 페이드인/아웃을 위한 애니메이션 시퀀스들

2. 이메일/전화번호 종류들(집,회사, 휴대전화 등등)의 배열

3. 표준 시스템 색상들

4. 응용프로그램 섬네일 이미지와 아이콘들

5. 공통으로 쓰이는 여러 표시물들과 레이아웃들

6. 오류 문자열들과 표준 버튼 텍스트들

7. 시스템 스타일들과 테마들

# 다중 응용프로그램 구성 관리

1. 홈

2. Menu

3. Settings

4. Locale & text

5. Select locale

6. 원하는 로켈 선택

# 현지화 기기 구성을 고료한 자원 조직화

안드로이드 시스템은 자원 디렉터리의 이름을 보고 특정 로켈과 특정 기기 구성에 가장 적합한 디렉터리를 선택해서

그 디렉터리 안의 자원들을 으용프로그램에 제공한다.

시스템이 요구하는 자원 디렉터리 명명 규칙을 살펴보자

기본자원 : /res/values, /res/drawable/ 같이 자원 형식 이름으로만 된 디렉터리에 저장된다.

현지화나 기기 구성 조건에 따른 자원들을 담을 디렉터리에는 기본 이름 뒤에 특정한 한정사를 붙인다.

표 5.7에 여러 한정사들이 나열되어 있다.

한 디렉터리에 여러 개의 한정사들을 적용할 수 있는데, 그런 경우에는 대시(-) 한정사들을 연결한다.

# 코드에서 특정 구성의 자원을 선택하려면

# 뷰, 위젯, 레이아웃

- 안드로이드 뷰 : android.view 패키지

android.view.View 클래스

이 View 클래스는 안드로이드 UI의 기본적인 구축 요소에 해당한다.

- 안드로이드 위젯 : android.widget, android.appwidget

ImageView,FrameLayout,EditText, Button..

- 안드로이드 레이아웃 : 뷰의 일종이지만 스스로 무언가를 그리지 않는다.

LinearLayout : 수평,수직 줄을 표현

AbsoluteLayout : 위젯의 구체적인 위치를 개발자가 지정d

# 사용자에게 텍스트를 표시하는 TextView

화면에 고정된 텍스트 문자열 또는 이름표를 표시함

TextView 위젯은 다른 화면 요소와 함께 쓰이는 경우가 많다.

TextView 위젯에서 좀더 특화된 텍스트 표시 위젯을 파생하기도 한다.

다른 위젯들과 마찬가지로, TextView 자체는 View 의 파생클래스이며

android.widget패키지 안에 들어있다.

TextView는 View의 일종이므로 Viewe가제공하는 표준속성들(너비, 높이, 여백, 가시성 등등)을 지원한다.

물론 TextView 자신만의 속성들도 제공한다.

- setText()

- getText()

# 레이아웃 크기 설정

<TextView

android:id="@+id/TextView04"

andorid:layout\_width="wrap\_content"

andorid:layout\_height="wrap\_content"

android:lines="2"

android:ems="12"

android:text="@string/autolink\_test" />

- autoLink

● web : 텍스트 안의 URL을 링크로 만든다. 클릭하면 웹 브라우저가 떠서 해당 페이지가 표시된다.

● email : 텍스트 안의 이메일 주소를 링크로 만듬, 메일 클라이언트가 뜬다.

● phone : 텍스트 안의 전화번호를 링크로 만듬, 클릭하면 전화 걸기 응용프로그램이 뜬다.

● map : 텍스타안의 지리적 주소를 링크로 만든다. 클릭하면 Google 지도 응용프로그램이 뜬다.

<TextView

android:id="@+id/TextView04"

andorid:layout\_width="wrap\_content"

andorid:layout\_height="wrap\_content"

android:lines="2"

android:ems="12"

android:text="@string/autolink\_test"

android:autoLink="web|email"

/>

한특성에 |를 이용해서 두가지 값을 지정함 --> all/none 사용가능

linksClickable : false 하면 링크 클릭이 먹지 않게됨

# EditText와 Spinner를 이용한 사용자 입력

<EditTExt

android:id="@+id/EditText01"

android:layout\_height="wrap\_content"

android:hint="type here"

android:lines="4"

android:layout\_width="fill\_parent" />

http://www.androidpub.com/keyboard : 한글 입력기

setSelection()

selectAll()

getText()

setText()

# 자동 완성기능

auto-complete : AutoCompleteTextView

MultiAutoCompleteTextView

final String[] COLORS = {"red","green","orange","blue","purple", "black","yellow","cyan","mageta" };

ArrayAdapter<String> adapter = new ArrayAdapter<String>(this,android.R.layout.simple\_dropdown\_item\_1line,COLORS);

AutoCompleteTextView text = (AutoCompleteTextView) findViewById(R.id.AutoCompleteTextView01);

text.setAdapter(adapter);

사용자가가 입력 상자에서 한 글자를 입력했을 때, 그 글자가 COLORS배열에 있는 색상 이름의 첫글자와 일치하면 해당 색상 이름이 사용자에게 제시된다.

<AutoCompelteTextView

android:id="@+id/AutoCompleteTextView01"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

andorid:completionHint="Pick acolor or type your own"

android:completionThreshold="1" />

- completionThreshold : 자동완성 드롭다운이 언제 표시될 것인지 결정

- completionHint : 드롬다운 목록 아래 표시될 도움말

- MultiAutoCompleteTextView의 사용법은 AutoCompleteTextView와 같되 여러 항목을 구분하기 위한 Tokenizer객체를 지정해야 한다는 점이 다르다.

MultiAutoCompleteTextView mtext = (MultiAutoCompleteTextView) findViewById(R.id.MultiAutoCompleteTextView01);

mtext.setAdapter(adapter);

mtext.setTokenizer(new MultiAutoCompleteTextView.CommaTokenizer());

# 입력 필터를 이용한 사용자 입력 제한

InputFilter ( android.text.InputFilter )

final EditText text\_filtered = (EditText) findViewById(R.id.input\_filtered);

text\_filtered.setFilters(new InputFilter[] {

new InputFilter.AllCaps(),

new InputFilter.LengthFilter(2)

});

# Spinner 위젯을 이용한 제한적인 텍스트 선택

<Spinner

android:id="@+id/Spinner01"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:entries="@array/colors"

android:prompt="@string/spin\_prompt" />

final Spinner spin = (Spinner) findViewByIDd(R.id.Spinner01);

TextView text\_sel = (TextView)spin.getSelectedView();

String selected\_text = text\_sel.getText();

# 버튼, 체크상자, 라디오 버튼

Button, ToggleButton, CheckBox, RadioButton

- android.widget.Button

- 안드로이드 시스템 자원 문자열들 android.R.String을 통해서 접근가능하다.

# 화면 방향의 변화에 대한 반응

현재 출시된 안드로이드 기긱들은 화면의 가로 모드와 세로 모드의 매끄러운 전환을 지원한다.

응용프로그램이 화면 방향의 변화에 반응하려면 OrientationEventListener(android.view.OrienationEventListener)를 통해

SensorManager 사건을 감지해야한다.

- 주의 안드로이드 SDK 1.5 R1이전에는 OrientationListener라는 클래스로 화면 방향 전환을 처리하는 경우가 많았다.

그러나 그 이후 버전들에서는 이 클래스가 폐끼 대상이 되었으므로 사용해서는 안되다.

또한 한드로이드 SDK 1.5에서는 화면이 새 방향에 맞게 자동으로 조정된다.

이러한 행동은 방향 변화를 야기할만한 물리적 사건(이름테면 T-Moblie G1 단말기에서 키보드를 꺼내는 등)과

비슷한 방식으로 일어난다. 기존 응용프로그램이 시스템이 감시하는 사건과 비슷한 사건을 감시했다면, 이러한

SDK변경 사항들에 주의를 기울여야 할 것이다.

방향감지기 치트시트 : http://www.novoda.com/blog/?p=77

# 스타일 다루기

공통적인 뷰 위젯 특성 값들의 집합

# 테마 다루기

개별 위젯이 아니라 활동의 모든 View 객체들에 적용되는 특성 집합

setTheme(R.style.right);

setTheme(R.style.green\_glow);

setContentView(R.layout.style\_samples);

# ViewGroup을 이용한 사용자 인터페이스 조직화

ViewGroup을 이용 조직화된 방식으로 화면에 표현할 수 있다.

addView() - ViewGroup 객체를 정의하는 요소 안에 자식 뷰 요소들을 포함시키면 된다.

LinearLayout 요소 안에 <TextView>요소들을 포함시킨 경우가 바로 그러한 예에 해당한다.

ViewGroup에서 파생된 하위 클래들은 크게 다음 두 범주로 나뉜다.

- 레이아웃

- 뷰 컨테이너 위젯

안드로이드 SDK는 개발자가 설계한 레이아웃을 시각적으로 표시해주는 계통구조 표시기

Gallery, GridView, ImageSwitcher, ScrollView, TabHost, ListView등등..

자식 뷰들을 다양한 방식으로 표시하는 브라우저 같은 것으로 생각한다면 보다 이해가 쉬울 것이다.

예를 들어 ListView는 항목들을 수직 방향의 목록 형태로 표시한다.

사용자는 목록을 상하로 스크롤하면서 개별 자식 뷰들에 접근한다.

반대로 Gallery는 항목들을 수평으로 나열하는데, 그중 "현재" 항목이 중앙에 좀 더 크게 표시된다.

사용자는 화면을 좌우로 스크롤하면서 개별 항목을 볼 수 있다.

좀 더 복잡한 커네이너인 TabHost는 뷰 객체들을 여러 개의 탭들로 조직화할 수 있게 한다.

하용자가 특정 탭을 선택하면 그 탭에 속한 위젯들이 표시된다.

# 계통구조 표시기

Eclipse 안드로이드 플러그인의 레이아웃 자원 편집기 외에,

안드로이드 SDK는 계통구조 표시기(Hierarchy Viewer)라는 사용자 인터페이스 도구를 제공한다.

명령 프롬프트에서 안드로이드 SDK의 /tools 디렉토리에서

hierarchyviewer 실행

# 내장 레이아웃 클래스

- AbsoluteLayout

- FrameLayout

- LinearLayout

- RelativeLayout

- TableLayout

android:layout\_특성\_이름="값"

- ViewGroup 특성들

특성이름 적용대상 설명 값

android:layout\_height 부모뷰,자식뷰 뷰의 높이,자식 뷰의 경우 필수임 크기값 또는 fill\_parent또는 wrap\_content

android:layout\_width 부모뷰,자식뷰 뷰의 너비,자식 뷰의 경우 필수임 크기값 또는 fill\_parent또는 wrap\_content

android:layout\_margin 자식뷰 뷰의 상하좌우의 여분 공간 크기값

android:layout\_marginTop 자식뷰 뷰의 위쪽 여분 공간 크기값

android:layout\_marginBottom 자식뷰 뷰의 아래쪽 여분 공간 크기값

android:layout\_marginRight 자식뷰 뷰의 오른쪽 여분 공간 크기값

android:layout\_marginLeft 자식뷰 뷰의 왼쪽 여분 공간 크기값

# AdapterViews 계통의 자료주도적 뷰 컨테이너들

ListView : 뷰 객체들을 수직 방향의 목록 형태로 보여준다.

수직 스크롤을 지원한다.

일반적으로 목록의 각 항목은 자료를 담는 뷰이다.

사용자는 목록의 한 항목을 선택해서 일정한 작업을 수행할 수 있다.

GridView : 뷰 객체들을 고정된 개수의 열들로 이루어진 격자(grid)형태로 배치한다.

격자의 각 칸에 이미지 아이콘이 배치되는 경우가 많다.

사용자는 한 항목을 선택해서 일정한 작업을 수행할 수 있다.

GalleryView : 뷰 객체들을 수평 방향의 목록 형태로 표시한다.

수평 스크롤을 지원한다. 역시 이미지들을 표시하는 데 흔히 쓰인다.

사용자는 한 항목을 선택해서 일정한 작업을 수행할 수 있다.

- ArrayAdapter

- CursorAdapter

# 그외의 뷰 컨테이너들

- 전환 위젯 (switcher): Viewswitcher에서 파생된 전환 위젯들은 자식 뷰가 단 둘이며,

한 시점에서 둘 중 하나만 표시할 수 있다. 이런 종류의 위젯은 두 자식 뷰를 애니메이션과 함께 전환해 준다.

기본적인 전환 위젯으로는 ImageSwitcher, TextSwitcher가 있다.

- 대화상자(dialog): 사실 대화상자는 View에서 파생된 위젯이 아니다.

오히려 활동과 비슷한데, 다른점이라면 팝업 창 형태라는 것이다.

대화상자도 전체화면 활동과 동일한 방식으로 레이아웃을 담을 수 있으며,

제공하는 사건들도 같은 것들이 많다. 물론 대화상자만의 사건들도 있다.

대화상자는 Dialog클래스에서 파생된 클래스의 객체이다.

- 스코롤 위젯: 뷰에 스크롤 기능을 제공하고자 한다면, View 파생 위젯들에 항상 존재하는 스코롤바 관련 특성들을 사용하는 대신

ScrollView 위젯(수직 스크롤)이나 HorizentalScrollView(수평 스크롤)를 사용하는 게 더 쉬울 수 있다.

이 위젯들에 다른 뷰를 담으면 그 뷰 가장자리에 스크롤바가 생긴다.

뷰에 스크롤 기능이 생기면 크기 제약이 느슨해지므로 레이아웃이 변할 수 있다.

- SlidingDrawer:안드로이드 1.5에 새로 도임된 컨테이너로 "서랍"을 흉내내는 SlidingDrawer가 있다.

이것은 서랍 손잡이에 해당하는 컨트롤과 위젯들을 담는 컨테이너 뷰로 구성된 것이다.

사용자는 손잡이 부분을 클릭하거나 끌어서 컨테이너 뷰를 열거나 닫을 수 있다.

사랍의 방향은 수평, 수직 모두 가능하다. 이 컨테이너는 항상 더 큰 화면을 표시하는 다른 레이아웃 안에 쓰인다.

그런 만큼, 응용프로그램 설정 화면을 제공하고자 할 때 특히나 유용하다.

예를 들어 게임 응용프로그램의 경우 사용자가 서랍을 열면 게임이 멈추고, 서랍 안에서 게임을 컨트롤 방식을 변경하고,

다시 서랍을 닫으면 게임이 재개되게 하는등..

# AppWidget을 통한 응용프로그램 뷰 접근

응용프로그램을 이전에 불가능했던 방식으로 안드로이드 운영체제와 좀 더 밀접하게 통합할 수 있다.

AppWidtget은 한 응용프로그램이 다른 안드로이드 응용프로그램을 하나의 위젯처럼 사용할 수 있게 만드는 수단이다.

AppWidtget 자체는 하나의 뷰 객체이며, 흔히 홈 화면(일반 컴퓨터의 "데스크톱" 똔는 "바탕화면"에 해당)같은

컨테이너 객체들에 담긴다. 이런 측면에서 AppWidget은 일종의 "데스크톱 플러그인" 메커니즘이라고 할 수 있다.

응용프로그램에 AppWdiget을 적용했을 때의 장점을 보여주는 예 몇가지를 들자면

여행 응용 프로그램이 간단한 AppWidget을 통해서 현재의 항공 한전 수준 정보를 외부에 제공한다.

사진 갤러리 응용 프로그램이 간단한 AppWidget을 통해서 현재의 항송 안전 수준 정볼를 외부에(안드로이드 홈 화면 등)제공한다.

사진 갤러리 응용프로그램이 간단한 AppWidget을 통해서 "오늘의 사진"을 외부에 제공한다.

"할 일"(to-do) 응용프로그램이 간단한 AppWidget을 통해서 오늘의 고순위 과제 몇 가지를 제공한다.

또는 사용자의 다음 일정이 달력에 나타나게 할 수도 있다.

AppWidget을 통해서 자신의 정보나 기능을 노출하는 응용프로그램을 AppWidget 제공자라고 부른다.

(콘텐츠를 제공하는 응용프로그램을 콘텐츠 제공자라고 부른느 것과 마찬가지이다.)

그리고 응용프로그램이 제공하는 AppWidget들을 담을 수 있는 화면 구성요소를 AppWidget호스트라고 부른다.

# 응용프로그램을 AppWidget 제공자로 만들려면

1. 응용프로그램의 안드로이드 매니페스트파일에 <receiver>요소(BrodacastReceiver 에 해당)를 추가하고,

그 안에서 <intent-filter> 요소를 이용해서 의도 필터를 AppWidgetManager.ACTION\_APPWIDGET\_UPDATE를 설정한다.

또한 <receiver> 요소 안에서 <meta-data> 요소를 이용해서 AppWidget위쳊으로 노출할 뷰를 정의한 XML레이아웃을 설정한다.

2. 다음으로, AppWidget으로 제공할 뷰를 정의하는 XML자원 파일을 만든다. 이 자원 파일에는 하나의 <appwidget-provider>요소가 있어야 한다.

3. Intent 방송, 갱신등을 처리하도록 AppWidgetProvider클래스를 적절히 구현한다.

# 응용프로그램을 AppWidget호스트로 만들기

흔히 하는 일은 아니지만, 응용프로그램을 AppWidget호스트로 만들 수 도 있따.

AppWidget호스트(anddorid.appwidget.AppWidgetHost)는 AppWidgte들을 담고 표시할 수 있는 컨테이너이다.

호스트에 어떤 AppWidget들을 어떤식으로 담아서 표시할 것인지는 개발자의 몫이다.

그러나, 그 어떤 AppWidget호스트이든 AppWidget 서비스(사용가능한 AppWidget들을 제공하는)와의 작절한 상호작용을 통해서 작동하므로,

개발자는 그 부분을 잘 파악해서 규칙에 맞게 구현해야 한다. 응용프로그램을 AppWidget호스트로 만드는 좀 더 자세한 방법은 SDK참고

# 그래피 애니메이션

커스텀 View 클래스를 만들고 Canvas와 Paint를 이용해서 도형, 텍스트, 애니메이션을 화면에 그리는 방법

PNG, JPG같은 그래픽 이미지와 텍스트, 그리고 기본 도형들을 홤녀에 그리는 수단을 제공한다.

그런 표시물들에는 다양한 색상과 스타일, 그래디언트를 적용할 수 있으며, 표준적인 이미지 변환 기능들을 적용해서 표시물의 형태와 모습을

변환하는 것도 가능하다. 더 나아가서, 표시물들을 시간에 따라 움직이고 변형함으로써 애니메이션을 구현할 수 있다.

# Canvas와 Pait다루기

화면에 뭐가를 그리려면 유효한 Canvas가 있어야한다.

보통은 응용프로그램을 위한 커스텀 뷰(View 파생 클래스)를 만드록 그 뷰는 onDraw()메서드를 구현함으로써

유효한 Canvas를 얻는다.

private static class ViewWithRedDot extents View {

public ViewWithRedDot(Context context) {

super(context);

}

@Override

projected void onDraw(Canvas canvas){

canvas.drawColor(Color.BLACK);

Paint circlePaint = new Paint();

circlePaint.setColor(Color.RED);

canvas.drawCicle(canvas.getWidth()/2,

canvas.getHeight()/2,

canvas.getWidth()/3, circlePaint);

}

}

onCreate()에서 이 뷰를 활동의 화면으로 설정하는 예

setContentView(new ViewWithRedDot(this));

# "화폭"에 해당하는 Canvas

Canvas(android.graphics.Canvas)는 화면의 직사각형 영역에 뭔가를 그리는 수단들을 제공한다.

이 메서드들은 영역 절단(clipping, 주어진 범위를 벗어난 부분은 화면에 나타나지 않게 하는것)도 잘지원한다.

Canvas가 나타내는 화면 공간의 크기는 Canvas를 담은 컨테이너 뷰에 의해 결정된다.

Canvas의 구체적인 크기는 getHeight()메서드와 getWidth()메서드로 알아낼 수 있다.

# "물감"에 해당하는 Paint

Paint(android.graphics.Paint)가 단지 색상 갑만 담고 있는것은 아니다.

스타일과 복합 색 및 렌덜이 정보를 캡슐화한다.

그러한 정보는 그래픽이나 도형, 특정한 글꼴(Typeface)이 지정된 텍스트에 적용할 수 있따.

- 색상 설정

android.graphics.Color

Paint redPaint = new Paint();

redPaint.setColor(Color.RED);

- 안티앨리이싱 설정

anti-aliasing은 그래픽이 화면에 좀 더 매끄럽게 나타나도록 하는 기술

Paint aliasedPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

- 스티알 설정

Paint의 스타일은 표시물 내부를 색으로 채우는 방식을 결정한다.

예를 들어 다음은 Paint의 스타일을 STROKE로 설정하는 코드인데,

이러면 표시물의 외곽선만 난타나고 색이 채워지지는 않는다.(이것이 기본 스타일이다.)

Paint linePaint - new Paint();

linePaint.setStyle(Paint.Style.STROKE);

- 그래디언트 설정

gradeint는 여러색들이 배끄럽게 변화되는 모습을 보여준다.

android.graphics.Shader에서 파생된 LinearGradient, RadialFradient, SweepGradent

모든 그래디언트는 적어도 두개의 색(시작색과 끝색)을 요구한다.

배열에 담긴 셋 이상의 색상들을 사용하는 것도 있다.

서로 다른 그래디언트들은 색상의 변화가 "흘러가는" 방향에서 차이를 보인다.

그래디언트가 반사 또는 반복되게 설정하는 것도 가능하다.

Paint 객체의 그래디언트는 setShader() 메서드로 설정한다.

- 선형 그래디언트를 위한 LinearGradient

색이 직선을 따라 변하는 것을 선형 그래디언트라고 한다.

표시물에 선형 그래디언트를 적용하려면 LinearGradient 객체를 인수로 해서

Paint의 setShader()메서드를 호출하고 그 Paint로 Canvas에 표시물을 그리면 된다.

다음이 그렇나 예이다

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.LinearGradient;

import android.graphics.Paint;

import android.graphics.Shader;

...

Paint circlePaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

LinearGradient linGrad = new LinearGradient(0,0,25,25,Color.RED,Color.BLACK,Shader.TileMode.MIRROR);

circlePaint.setShader(linGrad);

canvas.drawCircle(100,100,100,criclePaint);

- 방사상 그래디언트를 위한 RadialGradinet

색이 한 점에서 시작해서 사방으로 변해 가는 것을 방사상(radial) 그래디언트라고 한다.

표시물에 방사상 그래디언트를 적용하려면 RadialGradient 객체를 인수호 해서 Paint의 setShader() 메서드를 호출하고

그 Paint로 Canvas에 표시물을 그리면 된다.

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.RadialGradient;

import android.graphics.Paint;

import android.graphics.Shader;

....

Paint ciclePaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

RadialGradient radGrad = new RadialGradient(250,175,50,Color.RED,Color.BLACK,Shader.TileMode.MIRROR);

circlePaint.setShader(radGrad);

canvas.drawCircle(250,175,50,circlePaint);

- 쓸기 그래디언트

한 점에 고정된 막대를 원형으로 쓸듯이 움직이면서 색상을 변하게 하는것을 쓸기(sweep) 그래디언트라고 한다.

SweepGradient 객체를 인수로 해서 Paint의 setShader()메서드를 호출하고 그 Paint로 Canvas에 표시물을 그리면된다.

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.SweepGradient;

import android.graphics.Paint;

import android.graphics.Shader;

....

Paint ciclePaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

SweepGradient sweepGrad = new SweepGradient(canvas.getWidth()-175,canvas.getHieght()-175,new int[]{Color.RED,Color.YELLOW,Color.GREEN,Color.BLUE,Color.MAGENTA}, null);

circlePaint.setShader(sweepGrad);

canvas.drawCircle(canvas.getWidth()-175,canvas.getHieght()-175,100,circlePaint);

# 텍스트 그리기

기본 서체(typefase)와 스타일을 제공한다.

그 외의 글꼴들을 사용하는 것도 가능하다.

원하는 글꼴 파일을 응용프로그램 패키지에 포함하고, 실행 시점에서 AssetMananger로 그 글꼴 파일을 적재하면 된다.

# 기본 서체와 스타일 사용하기

안드로이드는 기본적으로 산스세리프(sans serif)서체를 사용하나, 고정폭(monospace)서체와 세리프서체도 제공한다.

다음은 기본 서체(산스세리프)와 안티앨리어싱을 적용해서 Canvas에 텍스트를 그리는 예이다.

import android.graphics.Canvas;

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Typeface;

...

Paint mPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

Typeface mTpye;

mPaint.setTextSize(16);

mPaint.setTypeface(null);

canvas.drawText("Default Typeface", 20, 20, mPaint);

다른 서체를 사용하고 싶다면 명시적으로

Typeface mTpye = Typeface.create(Typeface.MONOSPACE, Typeface.NORMAL);

Typeface mTpye = Typeface.create(Typeface.SERIF, Typeface.ITALIC);

\* 안드로이드의 모든 기본 글꼴이 각각 모든 스타일을 지원하는 것은 아니다.

예를 들어 기본 세리프 글꼴의 경우 볼드(굵게)는 지원하나 이탤릭은 지원하지 않는다.

기기마다 다를 수 도 있으므로, 원하는 글꼴과 스타일이 실제로 지원되는지를 하드웨어에서

점검해 봐야 할것이다.

안티앨리어싱, 밑줄, 취소선(strike-through) 여부 등의 기타 속성들은 Paint 객체의 setFlags()메서들로 설정할 수 있다.

mPaint.setFlags(Paint.UNDERLINE\_TEXT\_FLAG);

# 커스텀 글꼴 사용하기 ( http://www.chessvariants.com/d.font/utrecht.html )

기본 글꼴 이외의글꼴을 사용하고 싶다면, 원하는 글꼴 파일(.ttf등)을 응용프로그램의 자산으로 퐇마시키고 실행 시점에서 그것을 불러와서 텍스트에 적용하면된다.

1. /assets/fonts/에 폰트를 넣는다.

2. Typeface의 createFromAsset()메서들 이용해서 Chess Utrecht글꼴이 적용도니 객체를 생성한다.

import android.graphics.Color;

import android.graphics.Paint;

import android.graphics.Typeface;

...

Paint mPaint = new Paint(Paint.ANTI\_ALIAS\_FLAG);

Typeface mTpye = Typeface.create(getContext().getAssets(), "fonts/chess1.ttf");

# 텍스트가 화면에서 차지하는 크기 알아내기

Paint의 measureText(), getTextBounds() 메서드를 사용하면 된다.

# 비트맵 이미지 다루기

android.graphics.Bitmap

# Canvas에 비트맵 이미지 그리기

import android.graphics.Bitmap;

import android.graphics.BitmapFactory;

...

Bitmap pic = BitmapFactory.decodeResource(getResources(), R.drawable.bluejay);

canvas.drawBitmap(pic,0,0,null);

# 비티맵 이미지의 확대, 축소

Bitmap sm = Bitmap.createScaledBitmap(pic,50,75,false);

getWidth(), getHeight()

# 변호나 행렬을 이용한 비트맵 변환

createBitmap() 메서드에 적절한 반사(mirror) 변호나 행렬을 적용해서, 원본 비트맵(앞에서 생성한 pic)의 좌우가 뒤집힌 Bitmap객체를 생성하는 예이다.

import android.graphics.Bitmap;

import android.graphics.Matrix;

...

Matrix mirrorMatrix = new Matrix();

mirrorMatrix.preScale(-1,1);

Bitmap mirrorPic = Bitmap.createBitmap(pic,0,0,pic.getWidth(),pic.getHeight(), mirrorMatrix, false);

하나의 Matrix객테로 여러개의 변환을 적용하는 것도 가능하다.

Matrix mirrorAndTilt30 = new Matrix();

mirrorAndTilt30.preRotate(30);

mirrorAndTilt30.preScale(-1,1);

recycle()메서드를 호출해서 비트맵이 사용하던 메모리를 해제해 주는것이 좋다.

pic.recycle();

# 도형 다루기 - 건너뜀

# 에니메이션 다루기 - 건너뜀

------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- ------- -------

안드로이드 자료 저장 및 관리 API

- 파일시스템

- SQLite데이터베이스

- 콘텐츠 제공자 인터페이스를 통해서 안드로이드 시스템의 다른 응용프로그램이 가진 자료에 접근

- 자신이 콘텐츠 제공자가 되어서 다른 응용프로그램들이 자신의 자료에 접근

# 응용프로그램 선호설정 다루기

키/값 쌍들로 저장됨

저장 가능한 값

부울값

부동소수점 값

정수값

긴정수값

문자열 값

- 선호설정

android.content 패키지의 SharedPrefrence인터페이스가 제공한다.

응용프로그램에서 선호설정을 사용하기 위햇 알아야 할 기본적인 사항들은 다음과 같다.

- 목적에 따라 전용 또는 공유 SharedPreferences 인스턴스를 얻는 방법

- SharedPreferences 인스턴스의 메서드들을 이용해서 원하는 항목의 값을 조회하는 방법

- SahredPreferences.Edit 객체를 이용한 항목 변경 및 적용(commit)방법

- 전용, 공유 SharedPreferences 인스턴스 얻기

import android.content.SharedPreferences;

...

SharedPreferences settingsActivity = getPreferences(MODE\_PRIVATE);

한 응용프로그램의 모든 활동이 공유하는 공유 선호설정에 접근하려면 이와는 다른 메서드를 이용해서

SharedPreferences인스턴스를 얻어야 한다.

import android.content.SharedPreferences;

...

SharedPreferences setttings = getSharedPreferences("MyCustomSharedPreferences",0);

메서드가 다를 뿐만 아니라, 원하는 공유 선호설정의 이름을 지정해야한다는 점도 다름

public static final String PREFERENCE\_FILENAME = "AppPrefs";

- 설정항목 조회

메서드 용도

SharedPreferences.contains() 주어진 이름의 항목이 존재하는지 파악한다.

SharedPreferences.edit() 항목들을 변경하기 위한 편집기 객체를 얻는다.

SharedPreferences.getAll() 모든 항목(키/값 쌍)을 담은 Map객체를 얻는다.

SharedPreferences.getBoolean() 주어진 이름의 부울 항목의 값을 얻는다.

SharedPreferences.getFloat() 주어진 이름의 부동소수점 항목의 값을 얻는다.

SharedPreferences.getInt() 주어진 이름의 정수 항목의 값을 얻는다.

SharedPreferences.getLong() 주어진 이름의 긴정수 항목의 값을 얻는다.

SharedPreferences.getString() 주어진 이름의 문자열 항목의 값을 얻는다.

- 선호설정 항목 추가, 수정, 삭제

선호설정을 변경하려면 선호설정 편집기, 즉 SharedPreferences.Editor객체를 이용해야 한다.

메서드 용도

SharedPreferences.Editor.clear() 선호설정의 모든 항목을 제거한다. 변경적용(commit)시 이 연산이 먼저 수행된다.(현재 편집 세션에서 이 메서드가 언제 호출되었느냐와는 무관하게). 그런 다음에 다른 변경들이 적용된다.

SharedPreferences.Editor.remove() 주어진 이름의 항목을 제거한다. 변경적용(commit)시 이 연산이 먼저 수행된다.(현재 편집 세션에서 이 메서드가 언제 호출되었느냐와는 무관하게). 그런 다음에 다른 변경들이 적용된다.

SharedPreferences.Editor.pubBoolean() 주어진 이름의 부울 항목의 값을 새로 설정한다.

SharedPreferences.Editor.putFloat() 주어진 이름의 부동소수점 항목의 값을 새로 설정한다.

SharedPreferences.Editor.putInt() 주어진 이름의 정수 항목의 값을 새로 설정한다.

SharedPreferences.Editor.putLong() 주어진 이름의 긴정수 항목의 값을 새로 설정한다.

SharedPreferences.Editor.putString() 주어진 이름의 문자열 항목의 값을 새로 설정한다.

SharedPreferences.Editor.commit() 현재 편집 세션의 모든 변경을 적용한다.

import android.content.SharedPreferences;

...

SharedPreferences settingsActivity = getPreference(MODE\_PREIVATE);

SharedPreferences.Editor prefEditor = setttingsActivity.edit();

prefEditor.putLong("SomeLong",java.lang.Long.MIN\_VALUE);

prefEditor.commit();

Sample : SimplePrefs 참조

- 안드로이드 파일 시스템에 있는 선호설정 파일

- 선호설정

/data/data/<응용프로그램 패키지 이름>/shared\_prefs/<선호설정파일이름>.xml

<선호설정파일이름>부분은 전용 선호설정이면 활동 클래스 이름, 공유 선호설정이면 생성시 주어진 이름이다.

<?xml version="1.0" encoding="utf-8" standalone="yes"?>

<map>

<string name="String\_Pref">시험용 문자열</string>

<int name="Int\_Pref" value="-2147483648" />

<float name="Float\_Pref" value="-Infinity" />

<long name="Long\_Pref" value="9223372036854775807" />

<boolean name="Boolean\_Pref" value="false" />

</map>

# 파일일기 - 건너뜀

# XML 파일 읽기

안드로이드 SDK에는 XML 파일을 다루기 위한 여러 편의 수단들이 있다.

이들은 SAX, XmlPull v1, DOM Level 2 Core(제한적임)등을 지원한다.

- XML 관련 주요 패키지들

android.sax.\* 표준 SAX 처리부 작성을 위한 프레임워크

android.util.Xml.\* XMLPullParser를 포함한 XML 유틸리티

org.xml.sax.\* 핵심 SAX기능성을 제공한다.

프로젝트 사이트 : http://www.saxproject.org/

javax.xml.\* SAX와 제한된 DOM Level 2 Core지원.

org.w3c.docm DOM Level 2 Core 인터페이스

org.xmlpull.\* XmlPullParser와 XMLSerializer 인터페이스.

프로젝트 사이트 : http://www.xmlpull.org/

\* XmlResourceParser라는 XmlPullParser 구현 클래스를 이용해서 응용프로그램의 자원으로 포함된 정적 XML파일을 파싱하는 예

ResourceRoundup 참조 ( 5장 디렉터리에 있다)

# SQLite 데이터베이스를 이용한 구조적 자료 저장 - 건너뜀

# 네트워킹 API

AndroidMainfest.xml에 <uses-permission>요소를 추가해야한다.

<use-permission

android:name="android.permission.INTERNET"/>

- HttpURLConnection 클래스

- 네트워크에서 XML파일 가져와 파싱하기 (341페이지 참고)

-----------------------------------------------------------------------------------

#Dialog

AlertDialog

ProgressDialog

DatePickerDialog

TimePickerDialog

# 핸드폰에서 할일 ( 안드로드폰 디버깅 모드 설정 )

홈 - 메뉴 - 설정 - 응용프로그램 -

알수없는 소스(체크)

개발 -

USB 디버깅(체크)

켜진 상태로 유지(체크)

모의 위치 허용(체크)

# PC에서 할일

HTC Desire : 일련번호 : 12B9WE630017, 전화번호 : 010 7132 3209

0. 필수 설치 프로그램

- Eclipse + Android plug-in (DDMS) :

http://www.eclipse.org/

http://dl-ssl.google.com/android/eclipse/

1. 컴퓨터 안드로이드폰을 USB를 통해 연결합니다.

-> micro SD카드가 핸드폰연결되었는지 확인합니다.

2. 연결시 핸드폰에서 선택옵션

HTC 동기화

디스크 드라이브(선택)

3. 탐색기에서 HTCSync2.0.34.exe 실행

- USB 드라이버를 포함하고 있는것 같습니다.

4. 선택적으로 설치할 프로그램

- Droid Explorer : http://de.codeplex.com/

- Android Screen Cast : http://code.google.com/p/androidscreencast/

5. AndroidManifest.xml 수정

- android:debuggable="true"

- android:minSdkVersion="7" (2.1)

<application android:icon="@drawable/icon" android:label="@string/app\_name" android:debuggable="true">

<uses-library android:name="android.test.runner" />

</application>

<uses-sdk android:minSdkVersion="7" />

# 핸드폰에서 할일 ( 안드로드폰 디버깅 모드 설정 )

1. 연결시 핸드폰에서 선택옵션

HTC 동기화(선택)

디스크 드라이브

참고

http://drkein.tistory.com/154

http://www.kandroid.org/board/board.php?board=sourcecode&command=body&no=39

http://onjo.tistory.com/2037

---------------------------------------------------------------------------

eclipse plugin install

http://subclipse.tigris.org/

1. Install

Eclipse update site URL: http://subclipse.tigris.org/update\_1.6.x