Mobile Phone Programming

Lecture 8 - Android Data Storage

陳彥仰
NTU CSIE
http://mikechen.com

Announcements

- Fill out team project form today
- HW3 due Thur, Nov 19
  - TA hours available if you have questions
- Project UI presentation on Dec 3
  - define user tasks
  - initial UI sketches + 3 users' feedback
  - present tasks, original + revised UI, related apps

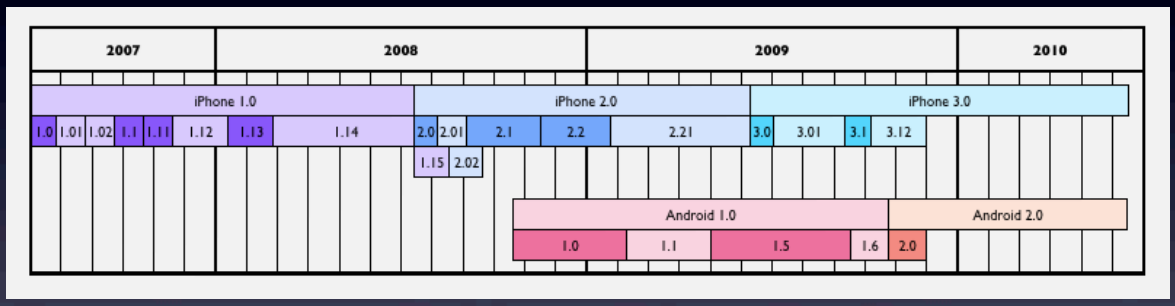
98學年度大學校院 網路通訊軟體與創意應用競賽

- http://ncsiac2009.csie.ncu.edu.tw/ ：手機應用組
- 報名日期：2009年11月15日止
- 初賽：2010年2月9日前繳交初賽書面資料
- 決賽：2010年4月24日進行決賽
  - 第一名：每隊獎金NT100,000元
  - 第二名：每隊獎金NT80,000元
  - 第三名：每隊獎金NT50,000元
  - 佳作五組：每隊獎金NT30,000元

News This Week (Nov 13)

- iPhone
  - 5000 sold in China the first 3 days
  - Now @ 100000 apps
- Samsung announced Bada ("ocean") platform
  - By 2011, mostly Bada, followed by Android, then WinMo. No Symbian.
- Nokia ships first Maemo mobile "computer" N900
  - based on Debian Linux

iPhone / Android OS Timeline



News This Week (Nov 13)

- Google acquires AdMob ad network for $750 million
  - AdMob shows relevant ads on mobile apps and mobile web pages
- Google Analytics for Mobile Apps
  - track "page views" and "events" in your mobile apps

Overview

- Industry news
- Cool app concept
  - http://www.youtube.com/watch?v=tb0pMeg1UN0
- Android data storage

Data Storage

Types of Persistent State

- Settings / User preference
  - e.g. username, password, font size
- Application data
  - e.g. train schedule, weather, stock prices
- Cache
  - temporary storage
  - e.g. Google maps for offline use, KKBox songs in offline mode

Types of Data Storage

- Android
  - Preferences
  - Files
  - Databases (SQLite)
  - Network
- Not accessible from other apps except through Content Providers

Preferences

- android.content.SharedPreferences
- Lightweight mechanism to store and retrieve key-value pairs of primitive data types.
  - boolean, float, int, and string
- Can use as a persistent hashtable to store whatever you like
- Typically used to store application preferences
  - Load in onCreate(), save in onStop()

Preferences Example

```
import android.content.SharedPreferences;
public class Calc extends Activity {
public static final String PREFS_NAME = "MyPrefsFile";
    …
    @Override
    protected void onCreate(Bundle state){
      super.onCreate(state);
    …
      // Restore preferences
      SharedPreferences settings = getSharedPreferences(PREFS_NAME, 0);
      boolean silent = settings.getBoolean("silentMode", false);
    …
    }
```

Preferences Example

```
    @Override
    protected void onStop(){
       super.onStop();

       // Save user preferences.
       SharedPreferences settings = getSharedPreferences
(PREFS_NAME, 0);
       SharedPreferences.Editor editor = settings.edit();
       editor.putBoolean("silentMode", mSilentMode);
       editor.commit();
    }
}
```


android.preference package

- Classes that manage application preferences and implement the preferences UI.
  - Consistent UI with system and other applications

android.preference package

- PreferenceScreen
  - Root of a Preference hierarchy (in a layout xml file).
- Preference UI
  - CheckBoxPreference
  - DialogPreference
  - EditTextPreference
  - ListPreference
  - RingtonePreference

Preferences UI Example

```
<?xml version="1.0" encoding="utf-8"?>
<PreferenceScreen xmlns:android="http://
schemas.android.com/apk/res/android">
    <CheckBoxPreference
        android:title="Checkbox Preference"
        android:defaultValue="false"
        android:summary="This can be true or false"
        android:key="checkboxPref" />
```

Preferences UI Code Example

- extends PreferenceActivity
- `addPreferencesFromResource (R.layout.MyPref)` loads the UI definition from xml
- use findPreference("myPref") to get a reference to a specific Preference

Live Demo: Preference UI

10min break

Files

- `java.io.File`
- Store files directly on device or on removable storage medium
  - By default, not accessible across apps (MODE_PRIVATE)
  - `Environment.getExternalStorageDirectory()`
- To package a static file at compile time, place the file in `res/raw` then open it with `Resources.openRawResource ( R.raw.myFileName )`
  - `android.content.ContextWrapper.getResources()`

Live Demo: Loading Static Files

Files

- To read data from a file:
  - `Context.openFileInput()` and pass it the local name and path of the file. It returns a standard Java `FileInputStream` object.
- To write to a file:
  - `Context.openFileOutput()` with the name and path. It returns a `FileOutputStream` object.
- Tip: type `sdk/tools/adb shell` and `cd` into `/data/data/` to view the files

Live Demo: Writing Files

Database

Example Scenarios

- I want to see ratings of 10 restaurants near me (out of 100000 restaurants)
  - How to efficiently find the10 nearest restaurants?
  - Sort by rating?
- Battery died while I'm saving my own rating and review
  - Is it possible that only part of it got saved?

Database ACID Properties

- **A**tomicity: guarantee that either all of the tasks of a transaction are performed or none of them are.
- **C**onsistency: remains in a consistent state before the start of the transaction and after the transaction is over (whether successful or not).
- **I**solation: other operations cannot access or see the data in an intermediate state during a transaction.
- **D**urability: once the user has been notified of success, the transaction will persist, and not be undone.

android.database.sqlite

- SQLite
  - light-weight, cross-platform database
  - support (complex) SQL queries
  - limitations:
    - partial support for triggers
    - limited ALTER TABLE support (can't modify/delete columns)

android.database.sqlite

- SQLite
  - light-weight, cross-platform database
  - support (complex) SQL queries
  - limitations:
    - partial support for triggers
    - limited ALTER TABLE support (can't modify/delete columns)
- Android sqlite3 command-line program
  - sdk/tools/adb shell
  - # sqlite3 /data/data/tw.edu.ntu/databases/info.db

android.database.sqlite

- insert: inserting a row into database
- update: updating rows in the database
- delete: deleting rows in the database
- query: returns a Cursor over result set
  - allowing lazy retrieval of actual data (instead of putting a collection of rows straight into memory)

Database Example

- One table in the database
  - DATABASE_NAME = “data”
  - DATABASE_TABLE = “notes”
- 3 columns:
  - _id: integer, primary key, autoincrement
  - title and body: text, not null
- Extend SQLiteOpenHelper to help with creating and upgrading database

Database Example

```
static class DatabaseHelper extends SQLiteOpenHelper {
    DatabaseHelper(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);
    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL("CREATE TABLE notes (_id integer primary key autoincrement,
                                        title text not null, body text not null);");
    }
    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS notes");
        onCreate(db);
    }
}
```


Database Example

```
DbHelper = new DatabaseHelper(mCtx);
mDb = mDbHelper.getWritableDatabase();

// create
ContentValues initialValues = new ContentValues();
initialValues.put(KEY_TITLE, title);
initialValues.put(KEY_BODY, body);
mDb.insert(DATABASE_TABLE, null, initialValues);
```

Database Example

```
// delete

mDb.delete(DATABASE_TABLE, KEY_ROWID + "=" + rowId,
null);


// update

ContentValues args = new ContentValues();

args.put(KEY_TITLE, title);

args.put(KEY_BODY, body);

mDb.update(DATABASE_TABLE, args, KEY_ROWID + "=" +
rowId, null)
```


Database Query Example

```
Cursor

query(boolean distinct, String table,
String[] columns,
String selection,
String[] selectionArgs, String groupBy, String having,
String orderBy, String limit


// query

mDb.query(true, DATABASE_TABLE,
new String[] {KEY_ROWID,KEY_TITLE, KEY_BODY},
KEY_ROWID + "=" + rowId,
null,null, null, null, null);
```

Android Data Storage

- Preferences
  - light-weight hashtable for primitive types
- Files
- Databases (SQLite)
  - complex query and transaction support
- Network
  - on demand
  - caching for performance and offline use

Next 2 Weeks

- Fill out team project form ASAP
- HW3 due Thur, Nov 19
- Project UI presentation on Dec 3
- Questions?