
中華大學
資訊工程學系

期末專題報告
智慧型數位生活記事本

Android 手機軟體程式開發

指導教授：張欽智 老師

學生：黃世笙(B09702221)

呂維源(B09702003)

尤冠几(B09702021)

林 勤(B09702303)

專題題號：PRJ2011-CSIE-10014

中華民國 101 年 1 月 18 日

目錄

1. 摘要	1
2. 研究背景與目的.....	2
2.1. 背景.....	2
2.2. 目的.....	3
3. 開發環境	4
3.1. 軟體.....	4
3.2. 硬體.....	5
4. 專題分工	6
5. 專題進度表	7
6. 系統設計與架構.....	8
6.1. 使用案例圖	8
6.2. 資料庫結構	9
6.2.1. 實體關聯圖.....	9
6.2.2. 關聯網要圖.....	9
6.3. 功能分析	10
7. 專題內容	11
7.1. 軟體 LOGO 設計	11
7.2. 主程式畫面配置設計	11
7.3. 筆記本功能設計與實作	14
7.3.1. 選擇編輯或瀏覽.....	15
7.3.2. 插入圖片.....	16
7.3.3. 拍照功能.....	18
7.3.4. 將訊息發佈上 Facebook.....	20
7.4. 行事曆功能設計與實作	24
7.4.1. 今日.....	25
7.4.2. 指定日期.....	26
7.4.3. 記錄與提醒.....	27
7.5. 趣味遊戲	34
7.5.1. 打地鼠遊戲.....	34
7.5.2. 翻牌遊戲.....	36
7.5.3. 貪食蛇遊戲.....	38
7.6. 作者	41
資料參考	44

1. 摘要

現在的社會各方面的發展都日行千里，相對的人們的生活也更為繁忙，大量的事務也隨之而來。若是沒有是適當的方式來做記錄，可能因此而 MISS 一些事情。但是傳統的筆記本在這個資訊時代已顯得不夠方便，我們迫切希望可以藉由更方便與快速的方式提醒與記下人們生活中事務。而手機又是隨時會攜帶在身上的東西。所以我們開始思考，如果手機可以紀錄我們所有的事情呢？

提到手機，這項電子產品從發明到現在已有十數年的歷史，近年來智慧型手機更在市場蓬勃發展，不但高度提升了手機的硬體功能，新的作業系統與豐富的應用程式，也帶給人們更佳的使用經驗。為了跟上這種時代的改變，我們不得不在此加緊腳步，研究相關的課題。

在目前智慧型手機的市場中，又以搭載 ANDROID 的產品為最大宗。ANDROID 系統開放原始碼，支援的硬體周邊也相當多元。所以我們的開發環境，也將以 ANDROID 為出發點。希望在這個功能強大的系統上面，能做出一個好用的 - 生活智慧記事本！

2. 研究背景與目的

2.1. 背景

智慧型手機是一種運算能力及功能比傳統手機更強的手機。通常使用的作業系統有：[Symbian](#)、[Windows Mobile](#)、[iOS](#)、[Linux](#)(含 [Android](#)、[Maemo](#) 和 [WebOS](#))、[Palm OS](#) 和 [BlackBerry OS](#)，他們之間的應用軟體互不相容。因為可以安裝第三方軟體，所以智慧型手機有比一般手機更多豐富的功能。硬體方面因為手機的尺寸比較大，可以搭載更強大的運算裝置以及高畫質螢幕，提供高容量儲存等等。規格可以說也勝出傳統 3G 手機許多。

[Android](#)(出現在早期的科幻電影中，指的是機器人的意思)是 [Google](#) 公司一個基於 [Linux](#) 為核心的軟體平台和作業系統，以 [JAVA](#) 為主要編輯器。[Android](#) 的特點是開源，他的 [SDK](#) 是開放給任何開發商，所有開發商都可以隨意更改介面。例如 [HTC](#) 的 [HTC Sense](#)，[Samsung](#) 的 [Touchwiz](#) 等等。2008 年開始，谷歌就不斷更新 [Android](#) 的版本，分別推出 1.5 [Cupcake](#)，1.6 [Donut](#)，2.0~2.1 [Eclair](#)，2.2 [Froyo](#)，2.3~2.4 [Gingerbread](#)。

2.2.目的

我們希望藉由 ANDROID 的開發環境，設計出一個可以記錄生活大大小小的軟體，並在 ANDROID 手機上面使用，我們稱它為生活智慧記事本。這個軟體將繼承一些傳統筆記本的記錄格式，並且可以藉由智慧型手機提供的觸控功能來使用，也可以與使用者的個人電腦同步。我們設計生活智慧記事本最大的目的在於，它可以主動的提醒使用者，未來有哪些事情將會發生！

3. 開發環境

3.1. 軟體



Eclipse IDE for Java Developers

Eclipse 是一款由 Java 所編寫成的整合開發環境，最早是由 Object Technologies International 所開發，於 1996 年被 IBM 收購後便改成 Common Public License 的授權方式，使 Eclipse 成為開放原始碼的軟體。雖然最初主要是用於 Java 語言的開發，但透過活躍的社群和許多的外掛、插件的支持下，使 Eclipse 成為一個多功能、高品質且支援多種程式語言的軟體開發環境，同時也是 Google 官方所推薦的 Android 開發平台。

Android SDK

Android 的軟體開發工具包。其中包括 Android 作業系統、Android 模擬器、Android 開發工具、函式庫、應用程式 API 與 Android 範例程式。它提供了

Windows/ Linux/ Mac 各種平台的開發套件，使我們能夠透過各種平台來開發 Android 軟體，所有的 Android 應用程式都是使用 Java 語言來撰寫的，再利用 Android 模擬器來做測試與除錯。

3.2. 硬體

Notebook x4

由於我們四位成員各有一台筆記型電腦，使我們在討論、團隊作業、資料蒐集上都相當的便利，不論是教室、圖書館、便利商店都能成我們聚會與討論的場所。

HTC Desire

HTC Desire 是台灣宏達電所推出以 Google Android 為作業系統的智慧型手機，於 2010 年 5 月 4 日正式發布，使用了高效能 1GHz 的 Snapdrgon 處理器、3.7 吋 AMOLED 螢幕、HTC sense 服務，雖然原先是搭載了 Android 2.1 作業系統，但能夠過官方升級至目前主流的 Android 2.3.3，是一款相當合適的測試平台。

4. 專題分工

組長: 黃世筌

相機功能、多媒體開檔功能、介面美化、海報製作、

LOGO 繪製、影片後製、報告整合、小品遊戲

組員: 呂維源

行事曆功能、遊戲選單、程式整合、音效設定、

Layout 版面配置、報告整合、小品遊戲

組員: 林勤

筆記本功能、主選單功能、測試與解決、功能結合、

報告 PPT 檔製作、報告整合、小品遊戲

組員: 尤冠几

Facebook 連結功能、功能構思、影片拍攝及處理

5. 專題進度表

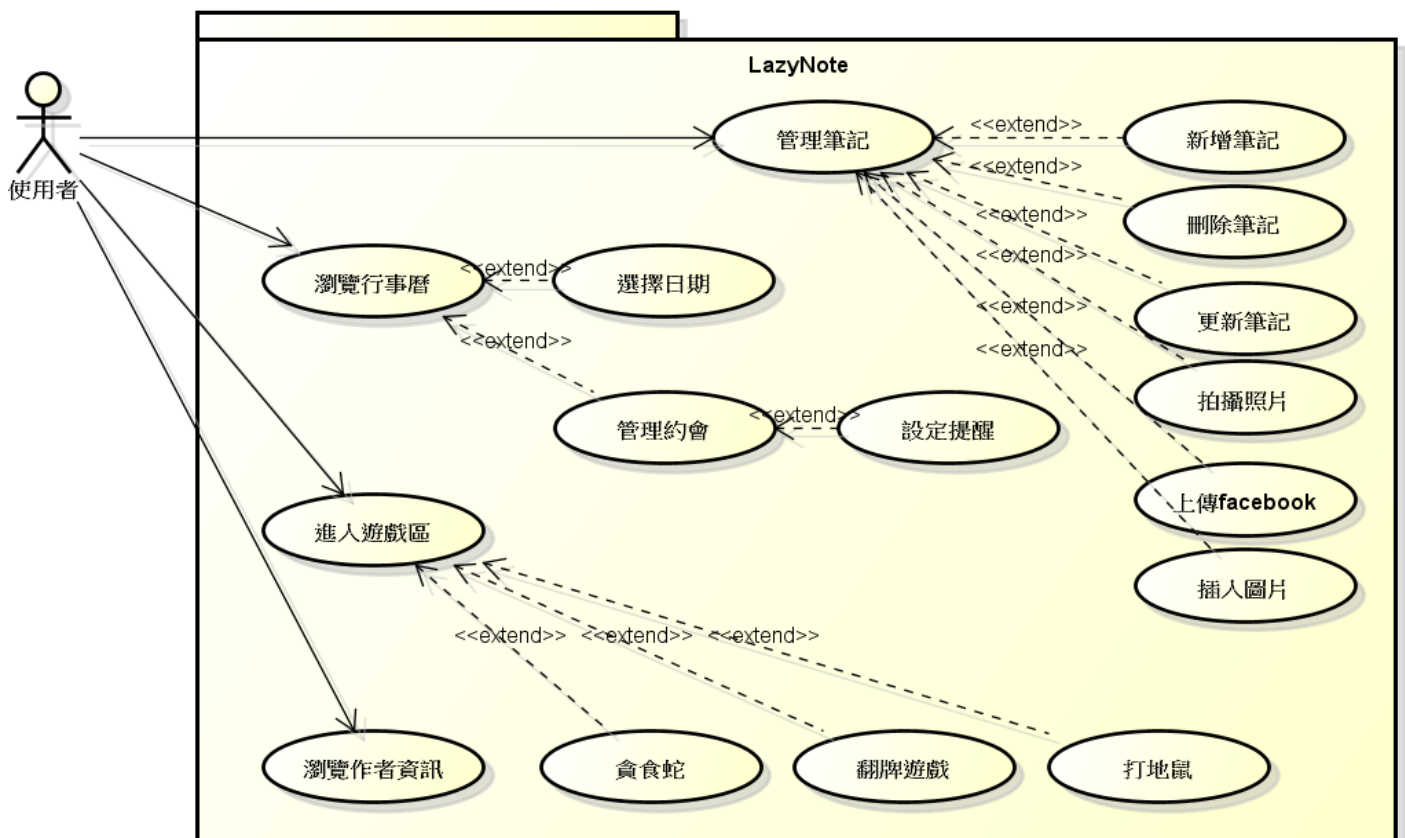
工作項目	二月	三月	四月	五月	六月	七月	八月	九月	十月	十一月	十二月	一月
記事本雛形	■	■	■									
筆記本功能			■	■	■	■	■					
行事曆功能				■	■	■	■					
多媒體應用				■	■	■	■					
專題 LOGO							■					
版面配置			■	■				■	■			
介面美化								■	■	■		
測試與回報										■	■	■

6. 系統設計與架構

6.1. 使用案例圖

基本上在設計一個程式的初期，要先設想可能會有那些功能要提供給使用者

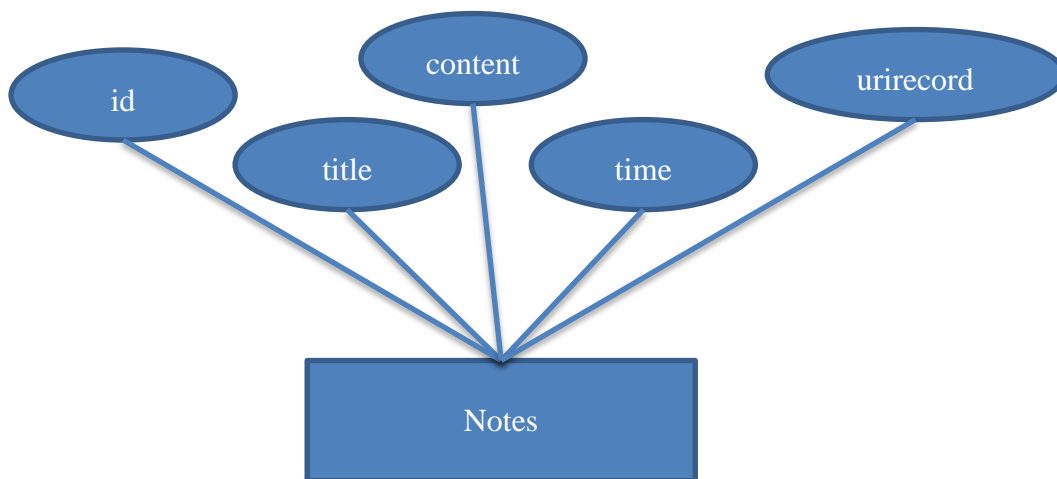
。為了方便釐清我們的系統架構，我們設計了一張 Use Case Diagram 如下：



6.2. 資料庫結構

使用資料庫來儲存所有筆記本內的資料，使用 `id` 來分別每一筆資料、`title` 來記錄每個筆記的標題、使用 `content` 來記錄每個筆記的內容、使用 `time` 來記錄編輯筆記的時間、使用 `urirecord` 來記錄圖片的 `uri` 位置。

6.2.1. 實體關聯圖



6.2.2. 關聯網要圖

Notes				
id	title	content	time	urirecord

6.3.功能分析

此系統提供四大功能:

A. 管理筆記

使用者進入管理筆記的子程式之後，有六項功能可以選擇。

可以新增筆記、刪除筆記、更新筆記、拍攝照片、將筆記上傳至 FACEBOOK 塗鴉牆，或是插入一張圖片，當然包括純粹的瀏覽。

B. 瀏覽行事曆

使用者選擇行事曆功能，將會進入一個月曆的畫面。可以在此畫面選擇日期，今天、或是特定的日期，在選定日期之後可以管理約會，新增、刪除，或是修改。在完成約會的建立之後使用者可以決定是否設定提醒的時間。如果有設定時間，時間到了之後就會發出音效，達到提醒的目的。

C. 進行遊戲

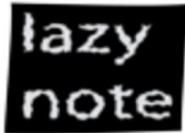
遊戲區的部分，使用者有三個遊戲可以選擇，打地鼠、翻牌遊戲，或是貪食蛇。

D. 瀏覽作者資訊

點選瀏覽作者資訊則可以觀看作者的基本資料。

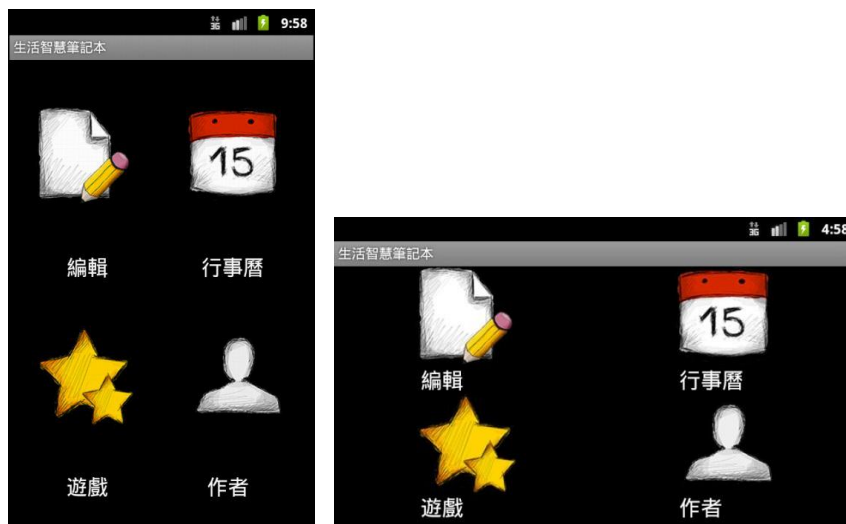
7. 專題內容

7.1. 軟體 LOGO 設計



由於我們這次所設計出的記事本，是希望能夠提供一個最簡易與直覺的操縱，所以將名稱取為 Lazy note，代表此軟體不需要繁瑣的設定與操作，就算是懶人也能用得很愉快。抖動的字底是為了體現慵懶的感覺，而使用黑色的背景與白色的字體則是為了凸顯軟體的名稱。

7.2. 主程式畫面配置設計



這是主程式畫面，設計為 2x2 的 Table 型 Layout。左上角是編輯按鈕，使用者點擊按鈕之後可以進入筆記本主畫面。右上角為行事曆按鈕，點擊之後可以進入

行事曆畫面。左下角為遊戲區，裡面有作者群自製的小遊戲提供娛樂。最後是右下角的作者資訊，有作者的姓名與學號。

主要實作方式是在layout的部分先設置四個Image Button以及四個Label，分別放入我們設計的圖片以及所需的文字如下：

```
<LinearLayout android:layout_width="match_parent" android:id="@+id/linearLayout2"
android:layout_height="wrap_content" android:weightSum="1">
    <TextView
        android:id="@+id/textView3"
        android:layout_width="160dp"
        android:layout_height="wrap_content"
        android:text="          編輯"
        android:textAppearance="?android:attr/textAppearanceLarge" />
    <TextView android:layout_height="wrap_content" android:id="@+id/textView2"
android:textAppearance="?android:attr/textAppearanceLarge" android:layout_width="160dp"
android:text="          行事曆"></TextView>
</LinearLayout>
<LinearLayout
    android:id="@+id/linearLayout5"
    android:layout_width="match_parent"
    android:layout_height="wrap_content" >
    <ImageButton
        android:id="@+id/ibt_menu_game"
        android:layout_width="150dp"
        android:layout_height="196dp"
        android:layout_weight="0.68"
        android:background="@drawable/background_black"
        android:src="@drawable/menu_game" >
    </ImageButton>
```

```

<ImageButton
    android:id="@+id/ibt_menu_syn"
    android:layout_width="150dp"
    android:layout_height="match_parent"
    android:layout_weight="0.46"
    android:background="@drawable/background_black"
    android:src="@drawable/menu_synchronize" >
</ImageButton>
</LinearLayout>

```

接著在SmartNoteActivity.java裡面宣告這些按鈕，並給予按鈕設定監聽器。

最後在這四個Image Button的監聽器裡面，利用Intent的方式來呼叫我們的子

程式如下：

```

private OnClickListener edit = new OnClickListener()
{
    public void onClick(View v) {
        SmartNoteMenuActivity.mediaPlayer.start();
        // TODO Auto-generated method stub
        Intent intent =new Intent();
        intent.setClass(SmartNoteMenuActivity.this,Note.class);
        startActivity(intent);
    }
};

```

所以當使用者點擊按鈕之後，就會觸發按鈕監聽器，就達到透過主選單呼叫子程

式的目的。而為了增加功能性，我們也實作了點擊按鈕會出現音效的效果，首先

import 這個可以處理多媒體撥放的 API，

```
import android.media.MediaPlayer;
```

然後在 SmartNoteActivity.java 中宣告這個變數

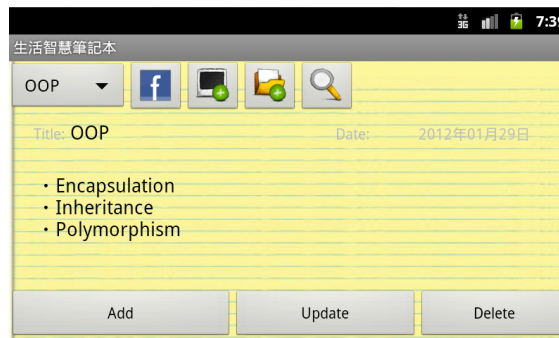
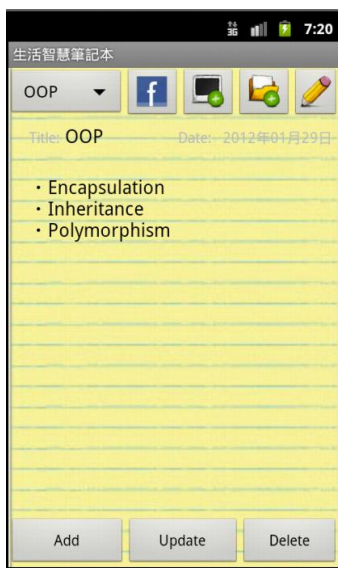
```
SmartNoteMenuActivity.mediaPlayer = MediaPlayer.create(this, R.raw.click);
```

最後在按鈕監聽器的函示裡加上這一行程式碼，

```
SmartNoteMenuActivity.mediaPlayer.start();
```

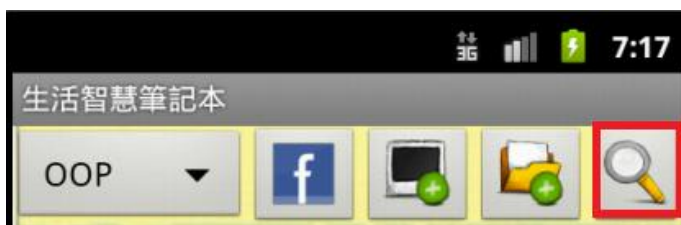
以上我們可以呼叫子程式並有按鈕音效的主選單就完成了。

7.3. 筆記本功能設計與實作



現在看到的是筆記本的主要頁面，裡面有七個按鈕，兩個 Text Box，三個 Label 以及一個 Spinner (下拉式選單)。

7.3.1. 選擇編輯或瀏覽



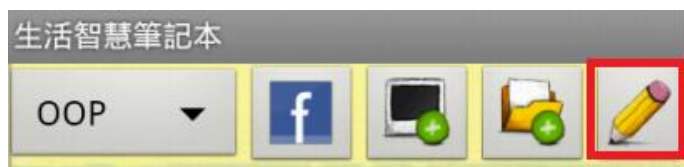
首先介紹的是最右邊的圖式按鈕，它的圖片是一張放大鏡，而主要控制的功能是可否編輯筆記的內容。因為 Lazy note 可以使用觸控式翻頁，所以為了要讓此功

能不與編輯模式牴觸，故設計了這個按鈕來切換編輯模式或是瀏覽模式。

```
listener_editmode = new OnClickListener() {  
    public void onClick(View v) {  
        Note.mediaPlayer.start();  
        imageButton_mode.setImageResource(R.drawable.icon_edit);  
        mEditTextContent.setOnTouchListener(null);
```

```
        imageButton_mode.setOnClickListener(listener_touchmode);  
    }  
};
```

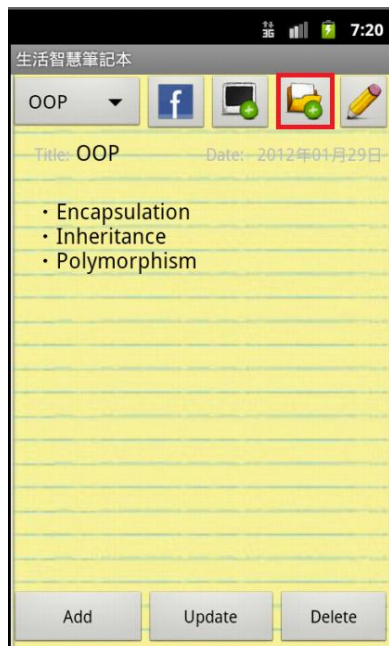
當使用者點擊這個按鈕的時候，它會切換到編輯模式，關閉觸控翻頁的功能，並且修改圖示成為一隻鉛筆。



而如果使用者再次點擊按鈕，則會切換到瀏覽模式，開啟觸控的功能，並將圖示修改成放大鏡。

```
listener_touchmode = new OnClickListener() {  
    public void onClick(View v) {  
        Note.mediaPlayer.start();  
        imageButton_mode.setImageResource(R.drawable.icon_see);  
        mEditTextContent.clearFocus();  
        mEditTextContent.setOnTouchListener(listen1);  
        imageButton_mode.setOnClickListener(listener_editmode );  
    }  
};
```

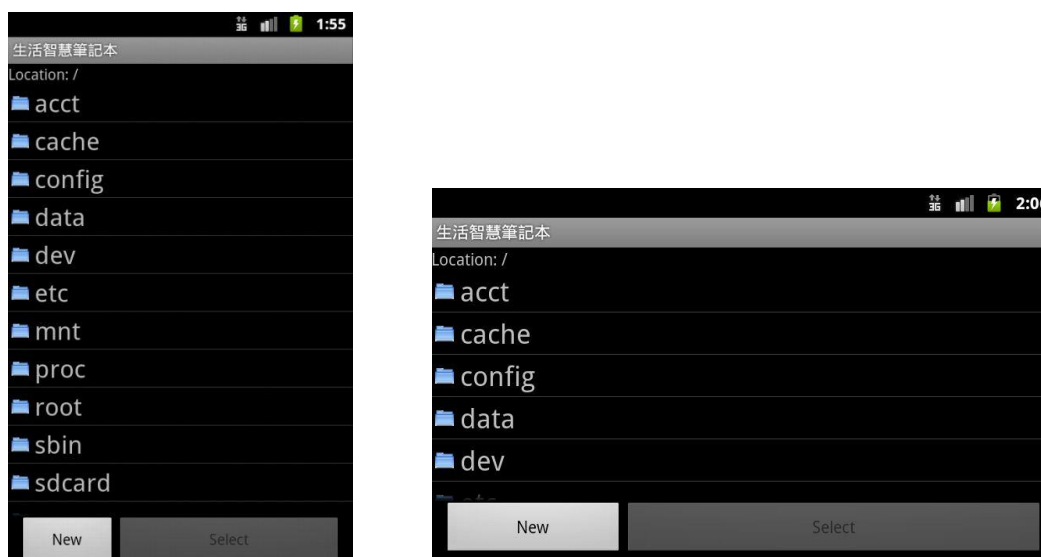
7.3.2. 插入圖片



因為有時候記事需要圖片的輔助，所以我們有插入圖片的功能。在該按鈕的監聽器之中，我們引用了一個開檔函式庫 `File Dialog`，在呼叫這個函式的同時，並指定路徑是 SD 卡如下：

```
listener_openfile = new OnClickListener() {  
    public void onClick(View v) {  
        Note.mediaPlayer.start();  
        Intent intent = new Intent(getApplicationContext(),FileDialog.class);  
        intent.putExtra(FileDialog.START_PATH, "/sdcard");  
        startActivityForResult(intent, 1);  
    }  
};
```

所以在點擊開啟圖片的按鈕之後，使用者會進入以下的畫面



使用者可以在這裡尋找想要插入的圖片，當選定好了之後，這個 File Dialog 函式庫可以幫我們傳回該圖片的 URI，再利用 Spannable 從 URI 輸入圖片。

```
public synchronized void onActivityResult(final int requestCode,int resultCode, final Intent data)
{
    if (resultCode == Activity.RESULT_OK)
        if (requestCode == 0) {System.out.println("Saving..."); }
        else if (requestCode == 1) {System.out.println("Loading..."); }
    String filePath = data.getStringExtra(FileDialog.RESULT_PATH);
    Uri uri = Uri.parse(filePath);
    mEditTextContent.append(getSpannableString(uri));
    recordNote(uri);//記錄行為
} else if (resultCode == Activity.RESULT_CANCELED) {
    System.out.println("Error...");
}
}
```

7.3.3. 拍照功能



有些時候現場拍照並製作為筆記相當方便，所以這邊加入了拍照的功能。

透過 Android 所提供的 Camera Api 來控制手機的相機。

首先開啟使用相機的權限以及啟用自動定焦功能。

```
<uses-permission android:name="android.permission.CAMERA"></uses-permission>
<uses-feature android:name="android.hardware.camera"></uses-feature>
<uses-feature android:name="android.hardware.camera.autofocus"></uses-feature>
```

在 surfaceView 啟動的時候，透過 Camera.open()取得使用的 camera 物件，並旋轉 90 度來解決相機預覽畫面是橫向的問題。

```
public void surfaceCreated(SurfaceHolder holder) {
    m_camera = Camera.open();
    try {
        m_camera.setPreviewDisplay(holder);
        m_camera.setDisplayOrientation(90);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

當按下拍照按鈕時，執行對焦後的回應函式。

```
public void onClick(View arg0) {  
    Button clickedButton = (Button) arg0;  
    if (clickedButton.getId() == R.id.autoFocusButton) {  
        m_cameraPreviewSurface.getCamera().autoFocus(myAutoFocusCallback);}  
    }  
}
```

對焦成功時就將照片儲存起來。

```
public void onAutoFocus(boolean success, Camera camera) {  
    if (success) {  
        camera.takePicture(myShutterCallback, null, myJpegCallback);  
    } else {  
        Toast.makeText(Cameras.this, "Can not focus", Toast.LENGTH_LONG)  
            .show();}  
    }  
};
```

透過 bitmap 物件將圖片做矩陣旋轉，使圖片儲存後是正的。之後再透過 outputStream 物件將圖片儲存在 SD 卡內。

```
Bitmap bitmap = BitmapFactory.decodeByteArray(data, 0, data.length);  
int w = bitmap.getWidth();  
int h = bitmap.getHeight();  
Matrix mtx = new Matrix();  
mtx.postRotate(90);  
Bitmap rotatedBMP = Bitmap.createBitmap(bitmap, 0, 0, w, h, mtx, true);  
String path= String.format("/sdcard/%d.jpg",System.currentTimeMillis());  
outStream = new FileOutputStream(path);  
rotatedBMP.compress(CompressFormat.JPEG, 100, outStream);  
outStream.write(data);  
outStream.close();  
Toast.makeText(Cameras.this, "相片大小:" + data.length,Toast.LENGTH_LONG).show();
```



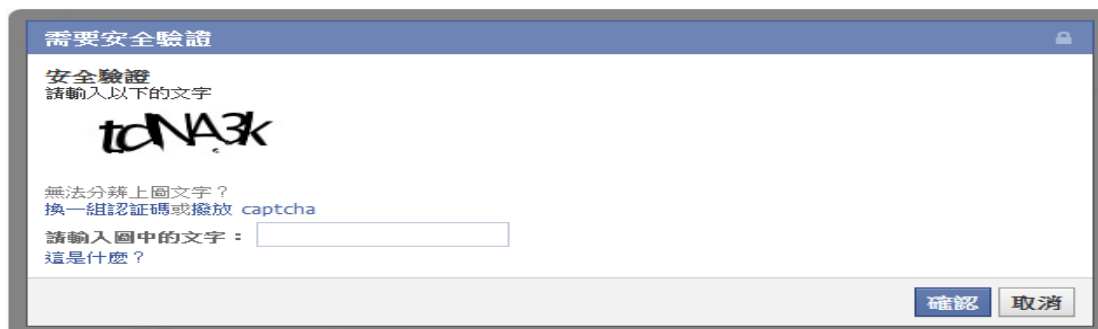
7.3.4. 將訊息發佈上 Facebook

除了將事情記錄在手機裡，我們也有提供將訊息發佈在 facebook 個人塗鴉牆上的功能。但是要將應用程式與 Facebook 連結，還需要先進行以下三個步驟：

A. 申請應用程式

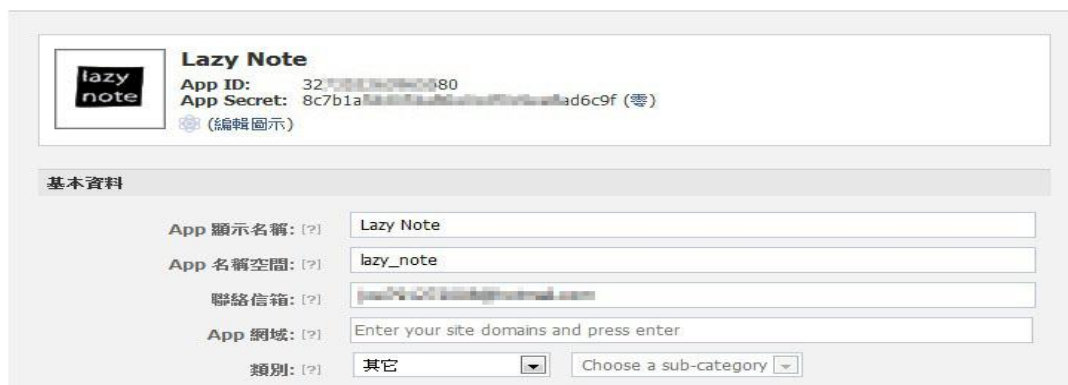
登入自己的 Facebook 後至最底下選擇“開發人員”→“應用程式”→“建立新的應用程式”

輸入自己想要的“程式名稱”以及“名稱空間” → “繼續”



輸入顯示的“安全驗證”(隨機產生) → “繼續” → 建立完成。

應用程式 ▶ Lazy Note ▶ 基本



B. 製作 Hash Key

OpenSSL: 至 [OpenSSL for Windows](#) 下載 Binaries 檔案，裡頭有

openssl.exe。

keytool -安裝 JDK，Java\jre6\bin 裡頭有 keytool.exe。

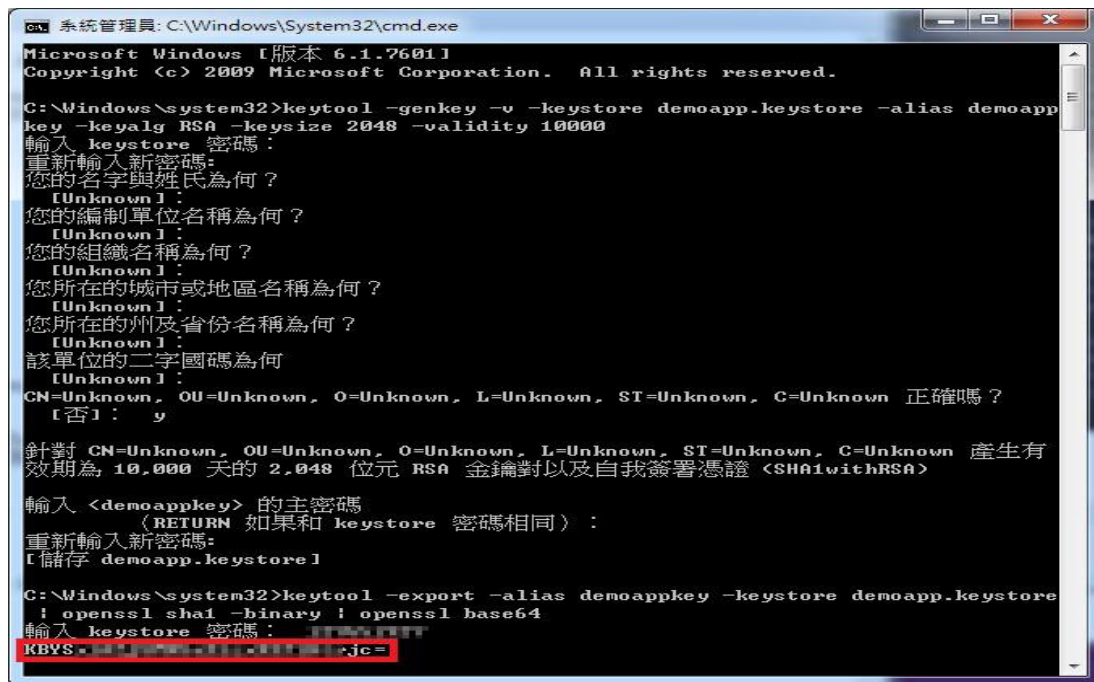
取得上述兩樣工具後，按下述方法操作

- i. 將 openssl.exe 存到 C:\Program Files\Java\jre6\bin 內。
- ii. 在 dos 裡利用 keytool 製作 keystore。

語法 : keytool -genkey -v -keystore ola2048.keystore -alias ola2048
-keyalg RSA -keysize 2048 -validity 10000 。

- iii. 在 dos 裡利用 keytool 以及 openssl 取得剛剛 keystore 的 Hash Key 。

語法 : keytool -exportcert -alias ola2048 -keystore ola2048.keystore |
openssl sha1 -binary | openssl base64 。



```
Microsoft Windows [版本 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Windows\system32>keytool -genkey -v -keystore demoapp.keystore -alias demoapp
key -keyalg RSA -keysize 2048 -validity 10000
輸入 keystore 密碼:
重新輸入新密碼:
您的名字與姓氏為何?
[Unknown]:
您的編制單位名稱為何?
[Unknown]:
您的組織名稱為何?
[Unknown]:
您所在的城市或地區名稱為何?
[Unknown]:
您所在的州及省份名稱為何?
[Unknown]:
該單位的二字國碼為何
[Unknown]:
CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown 正確嗎?
[否]: y
針對 CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown 產生有
效期為 10,000 天的 2,048 位元 RSA 金鑰對以及自我簽署憑證 (SHA1withRSA)
輸入 <demoappkey> 的主密碼
(RETURN 如果和 keystore 密碼相同):
重新輸入新密碼:
[儲存 demoapp.keystore]

C:\Windows\system32>keytool -export -alias demoappkey -keystore demoapp.keystore
! openssl sha1 -binary ! openssl base64
輸入 keystore 密碼:
KBYS=.....JC=
```

C. Android 與 FB 連結憑證設定

輸入取得的 Hash Key → “儲存”



完成三大步驟之後，在按鈕監聽器裡面宣告 facebook 的相關程式碼。如下

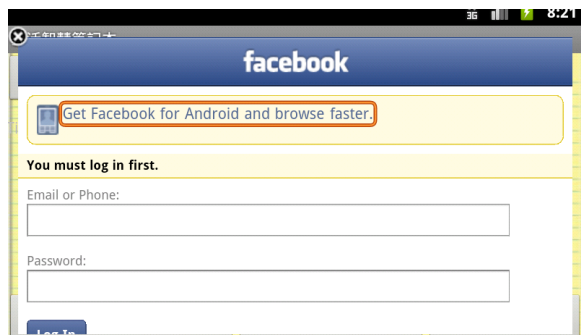
```

public void postToFacebook(){
facebook.authorize(this, new String[] { "email", "publish_stream",
"read_stream", "user_photos" },
    new DialogListener() {
        public void onComplete(Bundle values) {}
        public void onFacebookError(FacebookError error) {}
        public void onError(DialogError e) {}
        public void onCancel() {}
    });
}

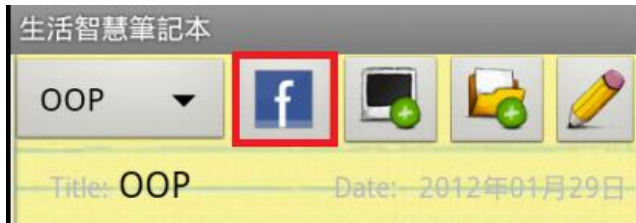
```



所以在點擊圖示按鈕之後，如果使用者尚未登入，則會出現一個登入畫面。



在登入成功之後，就可以將訊息發送到塗鴉牆了。之後每次要上傳訊息只要在點擊臉書圖示按鈕即可。



```

postToWall(mEditTextContent.getText().toString());
}

```

7.4. 行事曆功能設計與實作



這是行事曆畫面，主要用來將所存的事件詳記在日期上提醒自己；

白色字體則為平日、紅色字體則為假日；*18則為選擇日期，選到日期則會變為藍色方格及紅色字體，一開始起始日期為今日日期(指定日期)，而最上列文字的部分為敘述選此天的絕對位置；有*代表此天有事件紀錄(記錄/提醒)。



當使用者按下 menu 會顯示今日、指定日期及記錄提醒這三項選擇。

7.4.1. 今日

今日功能主要是你可能選擇時，選擇到離今日很遙遠的日期，一個個切月份跳回

是很麻煩的，設立此功能可以立即的跳回今日日期。



當使用者使用今日功能，會將記錄的日期存為目前的日期後並初始為今日日期，

最後將回傳做完成的動作。

```
public boolean onOptionsItemSelected(MenuItem item){
    Main.Main_mediaPlayer.start();
    Calendar calendar = Calendar.getInstance();
    calendarView.ce.grid.currentYear=calendar.get(Calendar.YEAR);
    calendarView.ce.grid.currentMonth=calendar.get(Calendar.MONTH);
    calendarView.ce.grid.currentDay=calendar.get(Calendar.DATE);
    calendarView.invalidate();
    return true;}

```

7.4.2. 指定日期



指定日期的功能主要為依據選擇的日期可以做增減調整，以使用於查看或者建立

此天的事件，如下圖：



當使用者選擇指定日期，指定到的日期在做確定及取消。

```
public boolean onOptionsItemSelected(MenuItem item){
    Main.Main_mediaPlayer.start();
    builder = new AlertDialog.Builder(activity);
    builder.setTitle("指定日期");
    myDateLayout = (LinearLayout) getLayoutInflater().inflate(R.layout.mydate, null);
    dpSelectDate = (DatePicker)myDateLayout.findViewById(R.id.dpSelectDate);
    tvDate = (TextView) myDateLayout.findViewById(R.id.tvDate);
    tvLunarDate = (TextView) myDateLayout.findViewById(R.id.tvLunarDate);
    dpSelectDate.init(calendarView.ce.grid.currentYear,calendarView.ce.grid.currentMonth,
calendarView.ce.grid.currentDay, this);
    builder.setView(myDateLayout);
    builder.setPositiveButton("確定", this);
    builder.setNegativeButton("取消", null);
    builder.setIcon(R.drawable.calendar_small);
}
```

```

adMyDate = builder.create();
onDateChanged(dpSelectDate, dpSelectDate.getYear(), dpSelectDate
.getMonth(), dpSelectDate.getDayOfMonth());
adMyDate.show();
return true;}

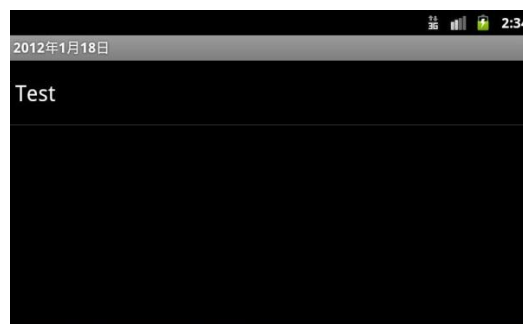
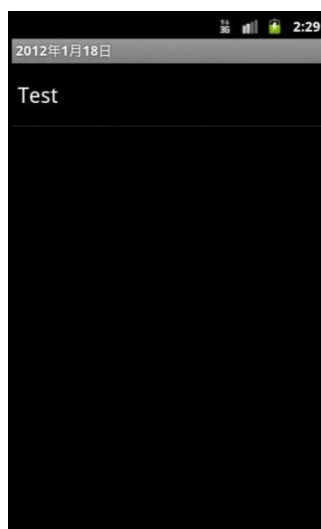
```



7.4.3. 記錄與提醒



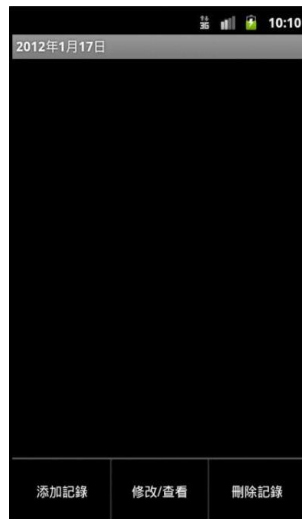
記錄與提醒主要是在指定日期上查看此日的事件，而使用者點入時，會列出此日期的事件。



使用者選擇記錄與提醒，會將目前指定的日期傳值給日期選單的類別。

```
public boolean onOptionsItemSelected(MenuItem item){
    Main.Main_mediaPlayer.start();
    Intent intent = new Intent(activity, AllRecord.class);
    intent.putExtra("year", calendarView.ce.grid.currentYear);
    intent.putExtra("month", calendarView.ce.grid.currentMonth);
    intent.putExtra("day", calendarView.ce.grid.currentDay1);
    activity.startActivity(intent);
    return true;
}
```

當使用者要新增、編輯及刪除事件，按下 menu 會出現添加記錄、修改/查看及刪除記錄的選項。



A. 添加記錄



添加記錄主要是為了在此日記錄事件，已便於之後能夠提醒及查看此日的事件。

當使用者點選後，將日期傳去 Record.class 做新增事件。

```
class OnAddRecordMenuItemClick extends MenuItemClickParent implements
    OnMenuItemClickListener{
    public boolean onMenuItemClick(MenuItem item){
        Intent intent = new Intent(activity, Record.class);
        activity.startActivity(intent);
        return true;
    }
    public OnAddRecordMenuItemClick(Activity activity){
        super(activity);
    }
}
```

B. 修改/查看



修改/查看主要是將事件內容做查看及修改設定；當使用者使用此功能時，如果日期選單內無事件資料，則已 false 表示不能查看；如果日期選單有事件資料，則表示 true，並將事件 id 傳至 Record.class。

```
class OnEditRecordMenuItemClick extends MenuItemClickParent implements
```

```
    OnMenuItemClickListener{
```

```
        public boolean onMenuItemClick(MenuItem item){
```

```
            AllRecord allRecord = (AllRecord) activity;
```

```
            int index = allRecord.getSelectedItemPosition();
```

```
            if (index < 0)
```

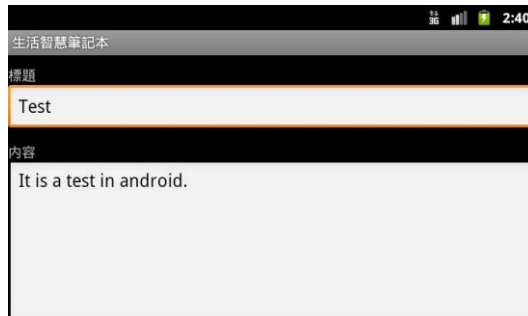
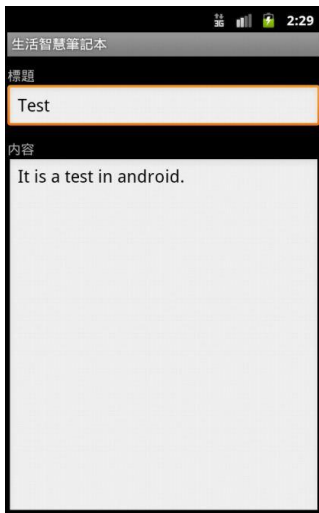
```
                return false;
```

```
            allRecord.startEditRecordActivity(index);
```

```
            return true;} 
```

```
        public OnEditRecordMenuItemClick(Activity activity){
```

```
            super(activity);} }
```



而在新增記錄及修改/查看皆會至 Record.class 做查看及編輯，如果是選擇修改/查看，則就更改事件的内容。

```
if (edit){  
    Grid.dbService.updateRecord(id, etTitle.getText().toString(),  
                                etContent.getText().toString(),remindTime, shake,ring);  
    AllRecord.recordArray.set(index, etTitle.getText().toString());  
    AllRecord.myListActivity.setListAdapter(AllRecord.arrayAdapter);  
}
```

如果是新增記錄，則在資料庫中加入一筆事件所有記錄；


```

else{
    Grid.dbService.insertRecord(etTitle.getText().toString(),
    etContent.getText().toString(), AllRecord.year + "-" +
    AllRecord.month + "-" + AllRecord.day, remindTime, shake, ring);
    AllRecord.arrayAdapter.insert(etTitle.getText().toString(), 0);
    AllRecord.idList.add(0, Grid.dbService.getMaxId(AllRecord.year
    + "-" + AllRecord.month + "-" + AllRecord.day));
}

```

事件須要有時間的提醒，則可以做鬧鐘的設定，在當日時間到便會告知使用者。



將事件設定鬧鐘時間以便此日可以提醒自己；而remindMsg為事件的內容、shake為震動布林值，調整震動則布林值為 1 及ring為響鈴的布林值，調整響鈴則布林值為 1，設定好後透過bundleRet存在remind結構裡。

```

public void onReceive(Context context, Intent intent){
    DBService dbService = new DBService(context);
    Remind remind = dbService.getRemindMsg();
    if (remind != null){
        Intent myIntent = new Intent(context, AlarmAlert.class);
        Bundle bundleRet = new Bundle();
        bundleRet.putString("remindMsg", remind.msg);
        bundleRet.putBoolean("shake", remind.shake);
        bundleRet.putBoolean("ring", remind.ring);
        myIntent.putExtras(bundleRet);
        myIntent.addFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        context.startActivity(myIntent);
    }
}

```

C. 刪除記錄



刪除記錄主要將此日取消的事件或者是錯誤的事件做刪除；當使用者選擇指定好事件做刪除，根據事件的id移除資料庫內的此事件。

```
class OnDeleteRecordMenuItemClick extends MenuItemClickParent implements
```

```
    OnMenuItemClickListener{
```

```
    public OnDeleteRecordMenuItemClick(Activity activity){
```

```
        super(activity);} 
```

```
    public boolean onMenuItemClick(MenuItem item){
```

```
        AllRecord allRecord = (AllRecord) activity;
```

```
        int index = allRecord.getSelectedItemPosition();
```

```
        if (index < 0)
```

```
            return false;
```

```
        recordArray.remove(index);
```

```
        int id = idList.get(index);
```

```
        idList.remove(index);
```

```
        allRecord.setListAdapter(arrayAdapter);
```

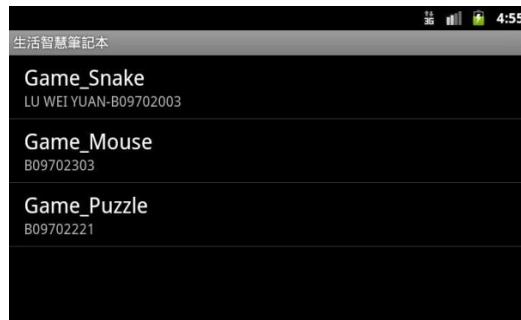
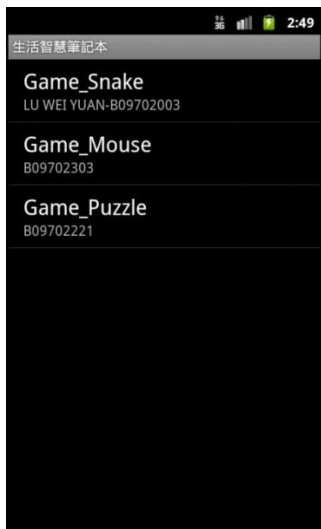
```
        Grid.dbService.deleteRecord(id);
```

```
        return true
```

```
    }
```

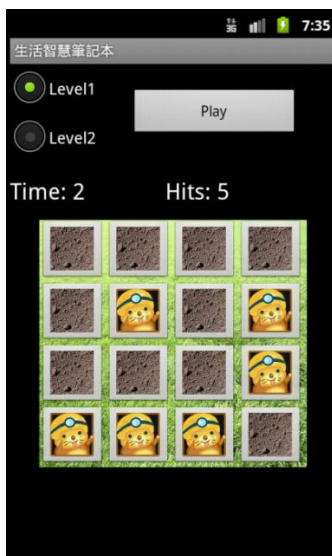
```
}
```

7.5. 趣味遊戲

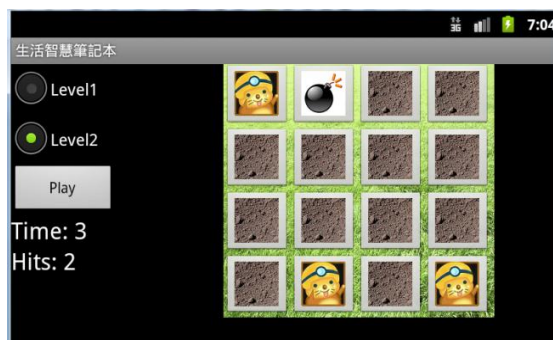


從選單點選遊戲後，會出現三則遊戲選單及遊戲的製作者，而 Game_Snake 為貪食蛇遊戲與製作者為 B09702003(呂維源)、Game_Mouse 為打地鼠遊戲與製作者為 B09702303(林勤)及 Game_Puzzle 為翻牌遊戲與製作者 B09702221(黃世筌)。

7.5.1. 打地鼠遊戲



Level1



Level2

這個遊戲名為打地鼠。顧名思義就是會在 16 個不同的地洞會隨機出現地鼠，如果玩家在消失之前成功點擊它，則獲得分數，遊戲時間為十秒。而這次的遊戲有設計兩個 Level。Level1 只有隨機出現地鼠，而 level2 就會出現炸彈，如果不將炸彈點擊移除，則扣玩家分數，時間遊戲一樣十秒。

實作的方式就是在 layout 的地方設置 16 個 image Button，還有兩個 ratio Button 跟一個 play 按鈕。接著在程式碼中宣告一個 image Button 陣列，然後在初始化這些按鈕的時候先設定背景圖片為地洞。接著利用亂數 image Button 可以隨機決定換不換圖片，而每個 image Button 都會作判斷。如果是五成的出現率，

```
private void randomButton () {
    ImageButton b;
    for(int i=0;i<16;i++)
        {b=(ImageButton)findViewById(R.id.imageButton1+myRandom.nextInt(4)*5+myRandom.n
extInt(4));
        if(randomShow(i)>1){b.setImageResource(R.drawable.mole1);b.setOnClickListener(hit);}
        else{b.setImageResource(R.drawable.hole);b.setOnClickListener(null);}}
    if(level2.isChecked()){
        if(randomShow(17)>1)level2randomButton(myRandom.nextInt(4)*5+myRandom.nextInt(4));};
```

則平均每秒會有八個地鼠出現在螢幕上讓玩家點擊。

```

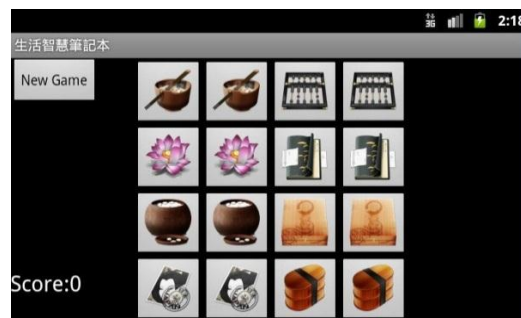
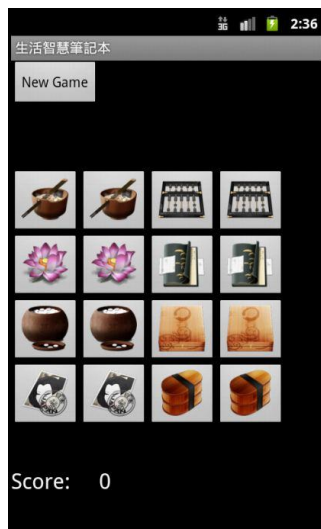
private int randomShow(int btnNumber){
//決定出現機率·預設為8分之3
if(randomShow[btnNumber]==-range && myRandom.nextInt(16)>10) {
    randomShow[btnNumber]=range;}
    else if (randomShow[btnNumber]>-range)randomShow[btnNumber]--;

return randomShow[btnNumber];};

```

7.5.2. 翻牌遊戲

這是一個考驗頭腦記憶能力的遊戲，開始遊戲後將 16 張卡片覆蓋在桌面，當掀開兩張相同的卡片後，就會加分並使卡片維持在掀開的狀態，反之則會將卡片蓋回去並扣除些微的分數，當所有的卡片都被掀開後就結束遊戲。



按下 New Game 以後隨機把代表卡片的數字放入陣列，並將卡片設置為蓋起。

```

public void newGame() {
    Random generator = new Random();

    int iTemp=0, iRandom=0;

    iScore=0;

    ((TextView)findViewById(R.id.txt2)).setText(Integer.toString(iScore));

    for(int i=0;i<16;i++){
        ((ImageButton)findViewById(ibuttons[i])).setImageResource(R.drawable.cover);

        bFaceUp[i]=false;

        iRandom=generator.nextInt(16);

        iTemp=iValues[i];

        iValues[i]= iValues[iRandom];

        iValues[iRandom]=iTemp;}}

```

當點擊卡片時，如果卡片是覆蓋而且是抽的第一張卡，顯示卡片圖案。

```

if(!bFaceUp[iNum]&& iFirstCard==-1)
{
    ((ImageButton)findViewById(ibuttons[iNum])).setImageResource(images[
iValues[iNum]]);

    iFirstCard=iNum;

    bFaceUp[iNum]=true;
}

```

當點擊卡片時，如果卡片是覆蓋但不是抽的第一張卡。

```

else if(!bFaceUp[iNum]&& iFirstCard!=-1)

```

如果第一張卡與第二張卡內容相同，顯示圖片內容並且增加分數。

```

if(iValues[iFirstCard]==iValues[iNum])
{
((ImageButton)findViewById(ibuttons[iNum])).setImageResource(images[iValues[iNum]]);
    iFirstCard=-1;
    bFaceUp[iNum]=true;
    iScore+=200;

    ((TextView)findViewById(R.id.txt2)).setText(Integer.toString(iScore));
};}

```

第一張卡與第二張卡不同，把卡片設定回覆蓋，並且扣取分數。

```

else
{
    Display display= new Display(iNum,iFirstCard);
    ((ImageButton)findViewById(ibuttons[iNum])).setImageResource(images[iValues[iNum]]);
    iScore-=20;
    ((TextView)findViewById(R.id.txt2)).setText(Integer.toString(iScore));
};
handler.postDelayed(display, 500);
bFaceUp[iFirstCard]=false;
iFirstCard=-1;}

```

7.5.3. 貪食蛇遊戲



這是貪食蛇遊戲，遊戲開始前先讀六秒，之後給予 60 秒的遊戲時間，控制蛇去吃老鼠；每吃到一隻老鼠會多一節維字身體，BodyNumber 會計維字增加的節數，且再加遊戲時間 9 秒，如果遊戲期間碰壁或者是碰到身體則會縮短二節；直到遊戲時間結束後看能增加多少維字身體節數。

首先開啟使用喚起的權限，取消螢幕的索屏，如下：

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

當老鼠被吃掉的時候，根據限制範圍會隨機下一隻老鼠的位置，

```
public MouseObj(Drawable drawable,Rect limitRect) {
    super(drawable);
    this.actRect=limitRect;}
public void random(Rect limitRect){
    this.actRect=limitRect;
    this.moveTo(actRect.left+r.nextInt(actRect.width()-this.getWidth()),actRect.top+r.nextInt(ac
tRect.height()-this.getHeight()));
}
public void random(){
    this.random(this.actRect);}
```

蛇會根據頭的座標移動，並控制座標吃老鼠，

```
public void move() {
    move(this.dstVectorX, this.dstVectorY);}
```

而每吃掉一隻老鼠，則會增加時間，

```

public void addTime(int addMicroseconds){
    overTime+=addMicroseconds;}

public boolean isTimeOver(){
    if(isTimePause)
        return pauseTime>overTime;
    else
        return System.currentTimeMillis()>overTime;}

```

也會增加身體節數。

```

public void add() {
    exGameObj newBody = new exGameObj(tail.rs.getDrawable(R.drawable.body));
    newBody.nextMove.set(tail.nextMove.x, tail.nextMove.y);
    for (int i = 0; i < tail.logPath.length; i++) {
        newBody.logPath[i].set(tail.logPath[i].x, tail.logPath[i].y);
        tail.logPath[i].set(tail.nextMove.x, tail.nextMove.y);}
    bodys.add(newBody);}

```

當蛇碰壁或者碰到自己，則縮短蛇的節數。

```

public void cut(int bodyIndex) {
    if (bodyIndex>0&&bodyIndex < bodys.size()) {
        exGameObj lastBody = bodys.get(bodyIndex-1);
        tail.setRect(lastBody.getRect());
        tail.nextMove.set(lastBody.nextMove.x, lastBody.nextMove.y);
        for (int i = 0; i < tail.logPath.length; i++) {
            tail.logPath[i].set(lastBody.logPath[i].x,
                lastBody.logPath[i].y);}
        for (int i = bodyIndex; i < bodys.size(); i++) {
            bodys.remove(bodys.size() - 1);}}

```

7.6. 作者



作者介紹這一次專題開發成員，有黃世筌(B09702221)、呂維源(B09702003)、尤冠几(B09702021)及林 勤(B09702303)，當從 Menu 啟動 Maker.class 則呼叫 layout 裡的 maker.xml 做成員介紹。

```
public class Maker extends Activity{  
    @Override  
    protected void onCreate(Bundle savedInstanceState){  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.maker);  
    }  
}
```

ImageView02 為設定圖片id，設定為 *student1* [黃世筌] 的圖檔照片

```
<ImageView  
    android:id="@+id/ImageView02"  
    android:layout_width="153dp"  
    android:layout_height="match_parent"  
    android:src="@drawable/student1" />
```

*ImageView03*為設定圖片id，設定為*student2* [呂維源] 的圖檔照片

```
<ImageView
    android:id="@+id/ImageView03"
    android:layout_width="77dp"
    android:layout_height="118dp"
    android:layout_weight="1.18"
    android:src="@drawable/student2" />
</LinearLayout>
```

*TextView04*為設定文字id，設定文字內容為組員:黃世荃 學號:B09702221

```
<TextView
    android:id="@+id/TextView04"
    android:layout_width="109dp"
    android:layout_height="100dp"
    android:text="組員:黃世荃
                學號:B09702221 " />
```

*TextView02*為設定文字id，設定文字內容為組員:呂維源 學號:B09702003

```
<TextView
    android:id="@+id/TextView02"
    android:layout_width="120dp"
    android:layout_height="101dp"
    android:layout_weight="0.19"
    android:text="組員:呂維源
                學號:B09702003 " />
</LinearLayout>
```

*ImageView01*為設定圖片id，設定為*student3* [尤冠几] 的圖檔照片

```
<ImageView
    android:id="@+id/ImageView01"
    android:layout_width="77dp"
    android:layout_height="118dp"
    android:layout_weight="1.13"
    android:src="@drawable/student3" />
```

*ImageView1*為設定圖片id，設定為*student4* [林 勤] 的圖檔照片

```
<ImageView
    android:id="@+id/imageView1"
    android:layout_width="77dp"
    android:layout_height="118dp"
    android:layout_weight="1.18"
    android:src="@drawable/student4" />
</LinearLayout>
```

*Student3*為設定文字id，設定文字內容為組員:尤冠几 學號:B09702021

```
<TextView
    android:id="@+id/Student3"
    android:layout_width="109dp"
    android:layout_height="match_parent"
    android:layout_weight="0.08"
    android:text="組員:尤冠几
                學號:B09702021 " />
```

*Student4*為設定文字id，設定文字內容為組員:林 勤 學號:B09702303

```
<TextView
    android:id="@+id/Student4"
    android:layout_width="120dp"
    android:layout_height="101dp"
    android:layout_weight="0.20"
    android:text="組員:林 勤
                學號:B09702303 " />
```

資料參考

- [1] 林城(民 100) · **Google Android 2.X 應用程式開發實戰** · 碁峯股份有限公司 · 民國 100 年 3 月。
- [2] 李寧(民 100) · **Android 案例開發完全講義** · 碁峯股份有限公司。
- [3] 余志龍等人(民 100) · **Google Android SDK 開發範例大全** · 悅知文化股份有限公司。
- [4] *Stark, Jonathan* (2010) , Building Android Apps With HTML, CSS, and JavaScript, Oreilly & Associates Inc, Sep. 2010
- [5] *Jeff, F* (2011) Hashimi, Sayed Y./ Komatineni, Satya/MacLean, Dave ° Springer-Verlag New York Inc °
- [6] Rogers, Rick/ Lombardo, John/ Mednieks, Zigurd/ Meike, Blake(2010) Android Application Development ° Oreilly & Associates Inc °
- [7] Ableson, W. Frank/ Collins, Charlie/ Sen, Robi Unlocking Android: A Developer' s Guide ° Oreilly & Associates Inc °
- [8] Meier, Reto(2008) Professional Android Application Development ° John Wiley & Sons Inc ° <http://developer.android.com/index.html>
- [9] ANDROID 中文資源網站 · <http://blog.chinatimes.com/tomsun/archive/2011/01/12/539502.html#596727>
- [10] 招佑 · 天翔 · 手機軟體開發教學 · <http://tomkuo139.blogspot.com/search/label/Android>
- [11] ANDROID 官方網站 · <http://developer.android.com/index.html>