東華大學資訊工程系

100學年度大學部專題研究報告

Where am I? Where should I go?

Android手機慣性導航系統

之設計與實作

指導教授：紀新洲 教授

專題參與人員：

阮玉璋

黃朝威

張書華

翁武湘

郭哲瑋

中 華 民 國　　101　 　年　　6　　月 　 1　 日

**國立東華大學資訊工程學系**

**專題報告原創性聲明書**

本人鄭重聲明：

所呈交的專題報告是在指導老師指導下進行的研究工作及取得的研究成果。除文中已經註明引用的內容外，本報告不包含任何其他個人或集體已經發表或撰寫過的研究成果。對本文的研究做出重要貢獻的個人與集體，均已在文中已明確方式標明。若有違上述聲明，願依校規處分及承擔法律責任。

題　　目： Where am I? Where should I go?

Android手機慣性導航系統之設計與實作

學生簽章：＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿＿

日 期: 　101.06.01

**計畫摘要**

近年來由於智慧型手機越來越普及，而且智慧型手機的功能也越來越強大，導航程式也成了常見的實用手機程式之一。不過各式各樣的導航程式不外乎都是需要利用GPS來運作，而很多地方尤其是室內的GPS訊號微弱甚至根本收不到訊號，在收不到GPS訊號的狀況下這些導航程式就失去了他們導航的功能了。所以我們決定把慣性導航的概念應用到手機導航上，使導航程式可以不需要接收外界的訊號即可運作。

關鍵詞 : Android，G-sensor，慣性導航，Canvas，QR code

**目錄**

1. **前言**----------------------------------------------------------1

2. **研究目的**------------------------------------------------------2

3. **研究方法與成果**------------------------------------------------3

3.1環境背景-------------------------------------------------3

3.2程式架構-------------------------------------------------4

3.3 UI界面--------------------------------------------------5

3.4 QR Code的介紹-------------------------------------------8

3.4.1 QR code解密---------------------------------------8

3.4.2 QR code結合---------------------------------------8

3.4.3 QR code應用---------------------------------------9

3.5 畫面的顯示----------------------------------------------10

3.5.1 畫面繪製------------------------------------------10

3.5.1.1 Bitmap--------------------------------------11

3.5.1.2 Canvas--------------------------------------13

3.5.1.3 Paint---------------------------------------14

3.5.2 增加效率------------------------------------------16

3.6 移動距離與方向的數值運算--------------------------------17

3.6.1 加速度的取值與校正--------------------------------17

3.6.2 移動距離的推算------------------------------------19

3.6.2.1 最原始的測距想法----------------------------19

3.6.2.2 加速度的區間化------------------------------20

3.6.2.3 誤差值累加的解決----------------------------21

3.6.2.4 以查表代替實際運算--------------------------22

3.6.3 方向的判定----------------------------------------22

3.6.3.1 方向判定的區間------------------------------23

3.6.3.2 轉彎的容錯與校正----------------------------23

4. **結果及討論**---------------------------------------------------24

**5. 參考文獻**-----------------------------------------------------24

1. **前言**

起初對這個想法的做法頗為茫然，剛開始測試時就出現了一些問題:手機硬體是否能支援應用，大部分智慧型手機才有內建傳感器；如何正確地取得感測器的數值與運行時間，這對誤差值與正確性有很大的影響。

由於模擬器沒辦法模擬傳感器，每修正一次就必須燒入實體手機進行測試；這在撰寫程式的效率上產生一定地影響。

之後不斷的與教授跟學長進行討論，不斷的嘗試與失敗，才漸漸描繪出目前的主題。

**2.研究目的**

當初構想是在完全沒有訊號的情況下迷路，希望能夠藉由手機的G-sensor與其他感應器的偵測，在手機上繪出使用者所走過的路徑，讓使用者不會在同一個區域打轉。

在深入研究G-sensor應用時發現，由G-sensor的基礎「陀螺儀」來製作的一些慣性導航的研究，發現對於人的步行振福產生的加速度過於敏感，為了取得較為正確的頻率與加速度方向，通常會使用濾波器去校正，但還是有50%的誤差。

故而引起我們的思考，如果以手機的加速度感應儀來偵測速度與振幅，配合磁場感應儀來偵測轉彎角度，配合程式抓取感測器的時間，再取得不同步行方式所產生的加速度與振幅頻率的測量數據進行過濾，希望能減少其誤差的產生，並帶畫面上繪出我們所走過的路徑。

而在地圖方面，雖然目前Google map 推出了離線導航系統，讓人們在沒有連線時也能使用，但是地圖大小仍是以城市地圖為主，並沒有一些建築物內部的細部結構地圖，於是我們就把範圍縮小到在建築物內，結合建築物的格局規劃與G-sensor的應用，進行手機慣性導航的設計與實作，讓使用者能在不連線的狀況下，找到並定位自己在建築內的所在位置與相對應方向。

1. **研究方法與成果**

**3.1環境背景**

Android是一個由Google公司主導的手持平台。以Linux為核心並使用 Java作為程式開發語言。為了方便於程式的開發，我們使用Eclipse來實作我們的程式，而手機介面的部分是由xml語言所撰寫。在開發程式時，我們選用了Android 2.3.3 版本來實作我們的程式。

需要的程式工具 :

1. JDK (Java開發工具包)
2. Eclipse (整合開發工具平台)
3. Android ADT (基於 Eclipse 的開發工具擴充套件)
4. Android SDK (Android 軟體開發套件)

|  |  |
| --- | --- |
| 程式工具 | 版本 |
| Java Development Kit, JDK | **1.7.0\_03-b05** |
| Eclipse IDE for Java Developers | **Indigo Service Release 2** |
| Android Development Tools , ADT | **18.0.0** |
| Software Development Kit, SDK | **Revision 19** |
| Application Programming Interface, API | **10** |

使用硬體 :

1. HTC Desire S
2. HTC Sensation XL
3. HUAWEI IDEOS U8500

**3.2程式架構**

**3.3 UI界面**

開始

暫停

觸控

清除

結束

**主程式**

開啟QR Code掃描程式

**手機畫面更新**

**QR Code**

**讀取**

**重力感測器**

**方位感測器**

**解讀QR Code**

**位移運算**

**Senor監聽**

**繪圖**

**UI界面**

**程式開始**

**主選單**

**MAIN**

**說 明**

**開 始**

**結 束**

**程式簡介**

**Finish**

**樓層一**

**樓層四**

**樓層三**

**樓層二**

**QR code切換**

**導航**

**定位**

**繪圖**

**導航**

**定位**

**繪圖**

**導航**

**定位**

**繪圖**

**導航**

**定位**

**繪圖**

何謂UI（User Interface）： UI是介於使用者與硬體之間， 用來彼此互動溝通的橋梁， 其用途是讓使用者能夠方便有效率地去操作硬體以達成某項功能或工作， 使用者介面定義廣泛， 包含了人機互動與圖形使用者介面， 凡參與人類與機械的資訊交流的領域都存在著使用者介面。

首先打開程式之後是我們程式的主選單， 裡面分別有三個按鈕： 「說明」 、 「開始」 、 「結束」； 點選「說明」的按鈕會出現本程式的簡介， 你也可以點選「開始」按鈕直接進入程式， 點選「結束」按鈕則直接關閉程式。

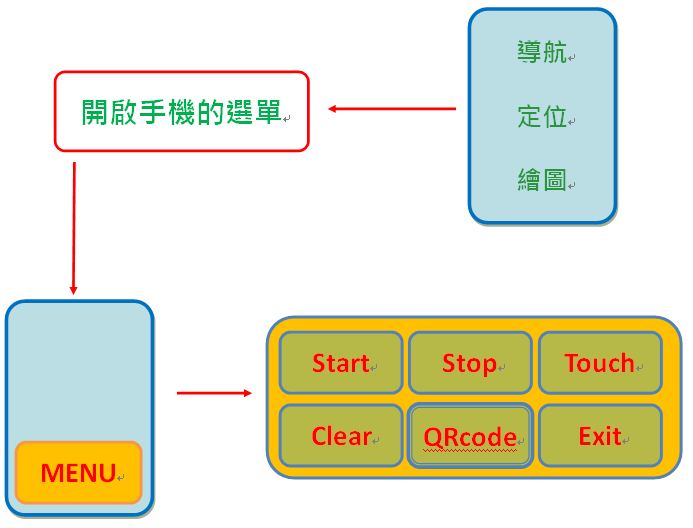
在進入程式之前先簡單介紹一下程式的元件， 一般視窗中呈現的各種元件， 例如： 按鈕、 文字區塊、 開關等， 稱作表單元件。 這些表單元件都是在Android開發環境的的XML檔裡設計，在XML檔你可以自行設計原件的佈局， 像是選擇畫面的背景、原件的排版，增加文字、按鈕、選單等等。

畫面上所看到的按鈕也是自己設計的， 但是XML檔只能設計表單原件的外表跟樣式， 真正的功能則是要寫在Android開發環境的java檔裡， 要讓一個按鈕正確的運作必須在java檔裡定義一個button並且跟XML設計的button接起來， 然後再處理特定的互動事件。

我們先將第一個畫面稱為Activity 1， 當我們按下「開始」按鈕觸發事件後會切換到第二個畫面Activity 2， 發生的這個事件是透過Intent來產生， Intent將Activity 1 和Activity 2 聯接起來， 當有觸發時就會切換過去

在「Intent」這樣的事件處理觀念裡，Android 試圖將事件解釋為「應用程式的意圖」或是「使用者的意圖」，並試著去解釋該意圖的目的，若 Android 系統本身能理解應用程式的意圖，便會「自行」去處理該意圖所應執行的工作。

接下來的第二個畫面是我們工學院四個樓層的按鈕， 分別intent到四個樓層的Activity， 四個Activity的功能基本上是一樣的， 我就挑一樓出來講解。



進入到樓層的Activity後會產生該樓層的地圖， 然後按下手機上的menu鍵會跳出一個選單的元件**，** 上面分別有六個自己定義的選單會對應到各自的觸發事件。

***Start* ：**

開始功能， 在還沒按下開始的選單之前***Stop***跟***Touch***的功能被鎖住， 當觸發開始事件後會跳出一個對話框， 說明可以***Touch***地圖上的任一位置當做你目前的起始點， 接下來地圖上的標記就會依照你行走的路線移動。

***Stop* ：**

暫停(恢復)功能， 當移動中按下暫停選單， 地圖上的標記會停止更新， 也不能執行***Touch***功能， 當再次按下***Stop*** 後則會恢復正常。

***Touch* ：**

觸摸功能， 直接使標記移動到觸摸螢幕的位置， 為了避免在使用中不小心觸摸到螢幕， 造成錯誤的標記位置， 所以限制每次使用***Touch***功能只能觸摸一次

***Clear* ：**

清除功能， 當畫面太過混亂或是不需要的標記產生時， 執行Clear功能把目前的畫面上的標記與線條全部清除掉。

***QRcode* ：**

掃描條碼功能， 呼叫外部的***QRcode***掃描器， 掃描條碼並且將數據傳回手機， 這部分的詳細內容在之後的章節再詳加介紹。

***Exit* ：**

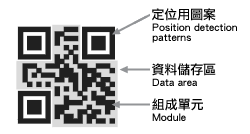
結束功能， 要離開程式的時候點選***Exit***選單， 會關閉所有地圖標記並且清空資料， 此時會回到選擇樓層的畫面。

**3.4 QR code的介紹**

**3.4.1 QR code解密**

QR 是「**Q**uick **R**esponse」的縮寫， 是快速反應的意思， 因為發明者希望QR code的內容能夠快速的被解讀，是目前最常見的一種「二維條碼」。

QR code優點比「一維條碼」儲存更多資料， 而且在掃描條碼的時候也不用像「一維條碼」一樣需要直線對準掃描器。

標準的QR code呈現一個正方形而且只有黑白兩色， 在三個角落會各有一個「回」字形狀的小正方形， 這是用來幫助掃描軟體鎖定QR code的圖片， 無論以任何角度掃描， 資料仍可正確被讀取。

**3.4.2 QR code結合**

我們程式中的QR code功能是呼叫外部的一個APP程式， 利用APP將我們設計的QR code資訊傳回我們的程式， 下面是我們呼叫所用的程式碼以及APP介紹:



**「com.google.zxing.client.android.SCAN」**這一行程式碼是google和zxing團隊一起設計出來， 當設計者使用時程式會自行將Android手機中， 屬於zxing團隊製作的QR code APP產生連結， 我們所使用的APP是**「BarcodeScanner4.0」**掃描器。

而我們用來定位的QR code， 是透過<http://www.calm9.com/labs/qrcode>這個QR code條碼產生器的網站， 設計出我們專屬的QR code資訊。

**3.4.3 QR code 應用**

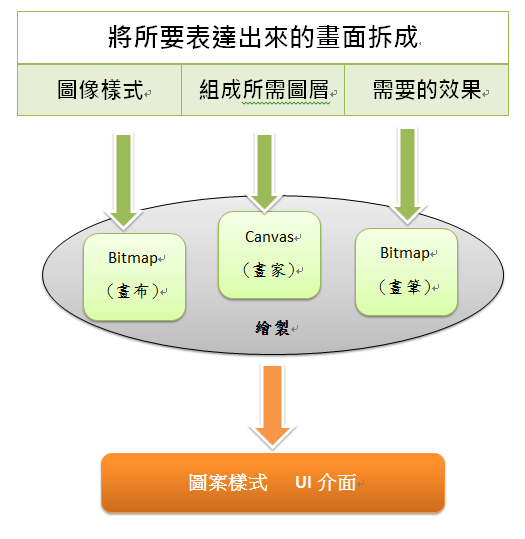
我們將字串訊息儲存進QR code， 以右邊的QR code例子，其內容為 **『OK0870903 工學院A337教室』** ， 然後我們將這串文字分割成三個部分， 接下來分別介紹各個部分。

首先我們要先宣告一個**String**將所有內容儲存起來， 這裡先稱為**temp。**

1. 取出**temp**的前2個字元， 也就是範例中的**「OK」**， 這是拿來判斷這一個QR code是不是自己設計的， 當掃到的條碼開頭是**「OK」**才會繼續存取條碼內容， 若是掃到的條碼開頭不是**「OK」**就會跳出一個錯誤的對話框， 並且不會對程式造成任何反應。
2. 取出**temp**的3到5的字元 、 6到8的字元以及第9個字元， 分別是範例中的 **「087」** 、 **「090」 、 「3」** ，這3個值是拿來定位我們在地圖上的**X、Y、Z**的座標， 其中**Z**代表的是樓層， **X**跟**Y**代表位於**Z**樓層中的座標點， 也就是QR code的所在位置。
3. 取出**temp**最後的文字內容， 範例中的**「工學院A337教室」** ， 也就是QR code所在位置的名稱。

第二部分中取出**X、Y、Z**的值會回傳到程式畫面顯示的部分， 並且將該QR code(所在位置) 顯示在畫面上。

**3.5畫面顯示**



將分別介紹:Bitmap、Canvas、Paint的基本應用方式以及如何使用這些function裡的一些使用，繪製出屬於自己的顯示畫面，不再是死板的內建的樣板模式，而這三種使用方式都會交互影響且缺一不可。

**3.5.1畫面繪製**

一開始在準備繪製地圖時，參考了許多不同的資料，例如Google Map API地圖、GPS定位、RPG地圖模式、將地圖檔直接繪出…等作法。

而最終會考慮將地圖檔直接繪出的有下列幾個原因

1. Google Map API :

考慮到我們的前提是在沒有任何訊號的情況下依然能進行導航指引，Google map、GPS 與訊號定位的選項立即被剔除在外；雖然Google map 近期新推出了離線導航功能，但由於地圖尺寸差異的關係，暫不使用。

(2) GPS定位 :

GPS分為訊號定位與衛星定位，其中衛星定位的侷限性較大，需要在空曠、無遮蔽的情況下才能準確定位；而訊號定位也是要藉由收到訊號的基地台來進行三角定位。

此兩種定位方式皆需要藉由訊號的傳遞定位，這在我們的假設沒有訊號的狀況下產生矛盾。

(3) RPG地圖模式 :

雖然能夠用程式將地圖產生成類似RPG地圖的模式，但每一個區塊的定義、判定過程繁瑣，不利於製作者使用。

(4) 地圖檔直接繪出 :

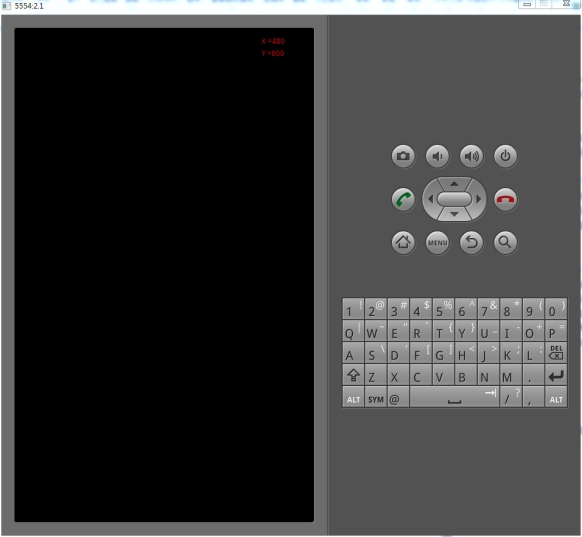
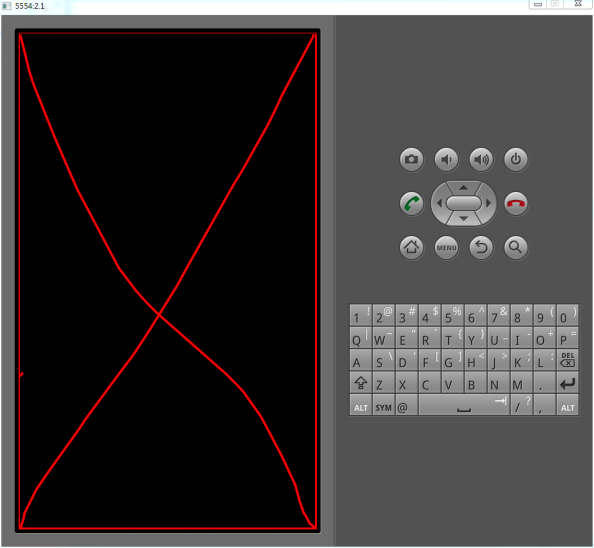
算是其中最有效率的做法，只要獲得地圖的比例尺，路線位置，規範出路徑走法位置，就能迅速的繪出使用。

**3.5.1.1 Bitmap**

不管是在程式裡或是現實中，要描繪一個有形體的物品，首先都必須要有一張畫布，這張畫布就是bitmap，這張畫布可以是自訂的，也可以是事先存放好的圖片；下列是有使用的function。

在graphics 目錄下的bitmap 裡 :

Bitmap.createBitmap();依函數值得不同，創建不同的畫布圖。

bitmap = Bitmap.createBitmap(w,h, Bitmap.Config.ARGB\_4444);

w, h為欲創建的畫布尺寸大小，此處需要的是一張與螢幕同大小的畫布，故先使用view目錄下的display class 去抓取螢幕的尺寸

winw = display.getWidth();

winh = display.getHeight();

winw，winh是取得的手機螢幕的尺寸大小，將值帶入w ,h 裡，也就是繪製一張大小尺寸與螢幕相等的空白bitmap畫布；而A(透明度)R(紅)G(綠)B(藍)分別存4bit的資料的一張透明畫布，另外有ARGB\_8888/ RGB\_565等格式，需要注意其容量大小，故依使用者需要自己調整；此處我們使用的為ARGB\_4444。

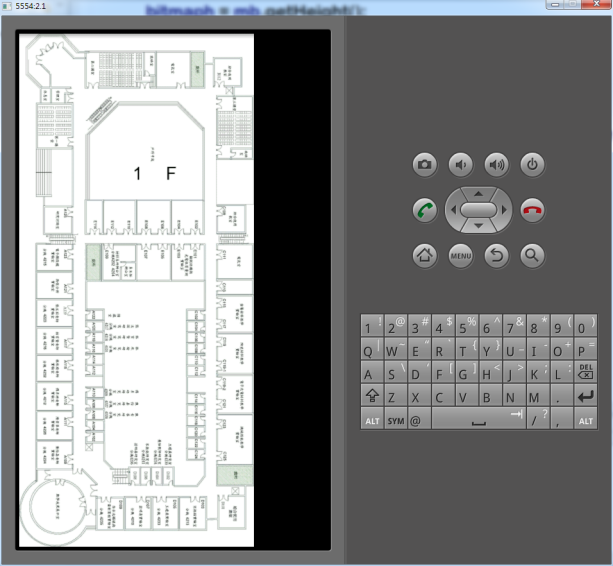
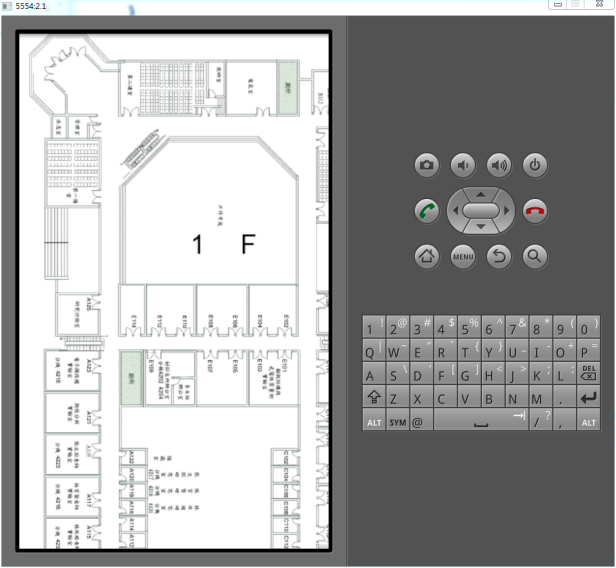
newmap = Bitmap.createBitmap(map, x, y, mapw , maph, matrix, true);

mapw = map.getWidth();

maph = map.getHeight();

在使用map圖片進行變換時，必須先用BitmapFactory.decodeResource();將map從資料夾內讀取出來，並由bitmap 下的.getWidth()取得獲得圖片資訊；mapw, maph為圖片map的長寬，再經由一個矩陣matrix進行放大縮小的變換後，由(x, y)為起始繪出一張變換後的newmap畫布。

原圖顯示 : 經矩陣縮放到貼齊螢幕



BitmapFactory.decodeResource();可以從不同的來源文件讀取圖檔

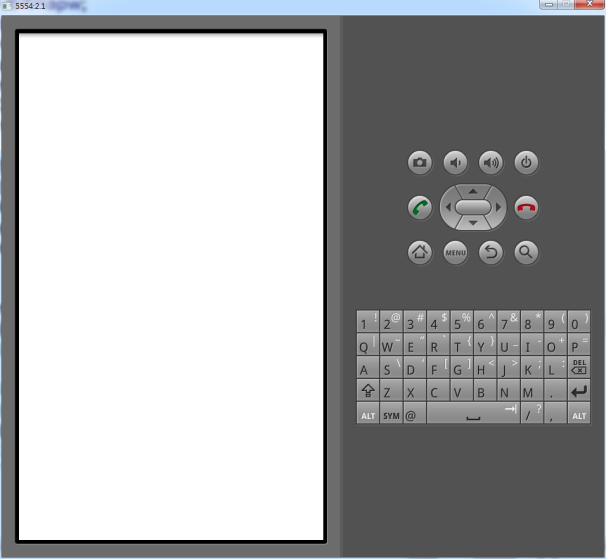
backmap = BitmapFactory.decodeResource(getResources(),R.drawable.back);

創建backmap，取得在drawable目錄下的back.png的資料，可以用來繪圖也可以用來自訂按鈕。

**3.5.1.2 Canvas** :

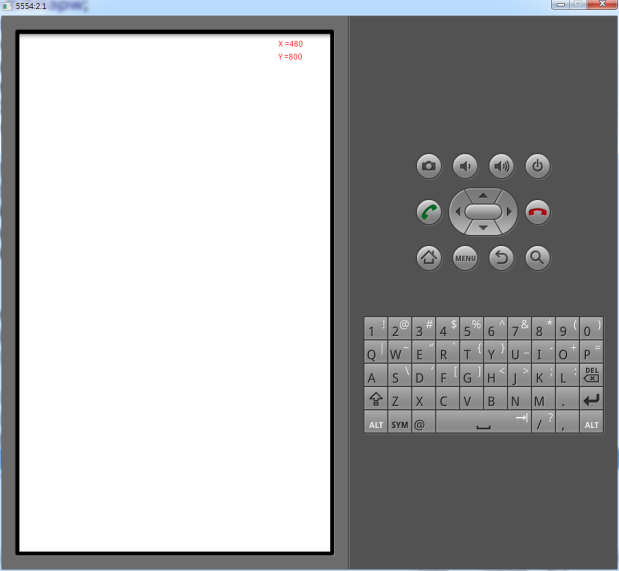
與bitmap、Paint關係密切，在canvas裡，許多繪圖的部分需要透過bitmap的使用與Paint的繪出，才能顯示在螢幕上。而我們有使用到以下幾種方式

.setBitmap(bitmap); 指定畫布



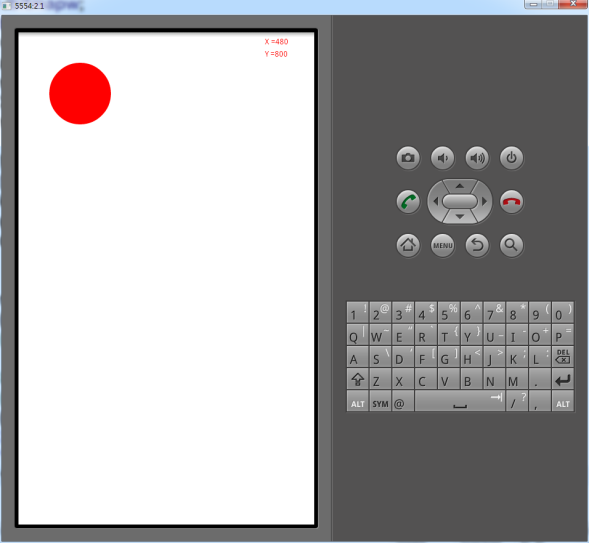
.drawBitmap(bitmap,X,Y,paint); 從座標(X,Y)為起點畫出一張bitmap的圖

建立空白背景



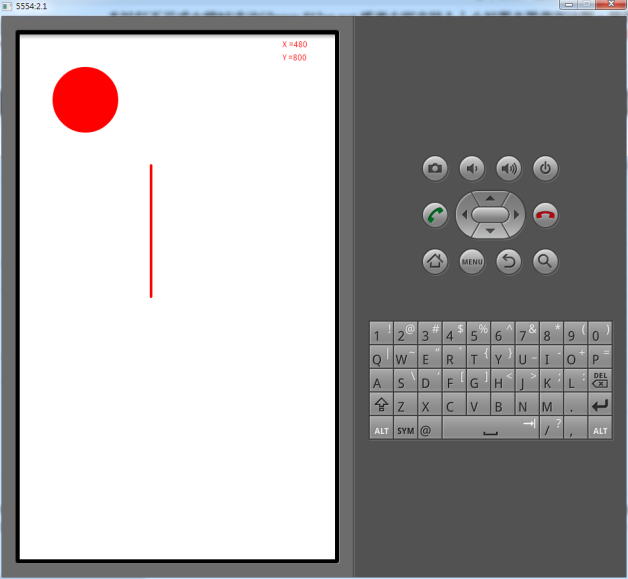
.drawText(); 可以畫出寫在裡面的文字

右上角顯示畫布尺寸



.drawCircle (X,Y,R, paint);在座標(X,Y)畫出半徑R的圓

我們用打點的方式把圓當成筆刷，進而打出路徑圖



.drawLine() 繪出線條

另外也有

.drawOval 畫出橢圓

.drawRect 畫出矩形

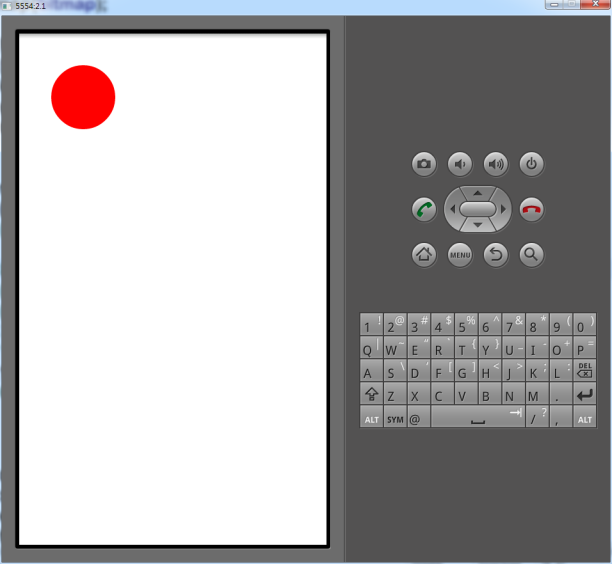
.drawArc 畫出弧形

.drawARGB/.drawRGB/.drawColor 使用單一色填充畫布

等不同的做法，配合Paint的筆刷設定，以及了解Bitmap的繪製順序，就能繪製出具有自己風格的的顯示畫面。

**3.5.1.3 Paint :**

相當於一個畫筆與調色盤，可以為用draw繪出的圖形進行渲染，修改；透過paint，可以在繪圖時控制其顏色、樣式與特殊效果。由於我們的地圖不需要太多的花樣使畫面過於雜亂，在這裡介紹幾種做法:

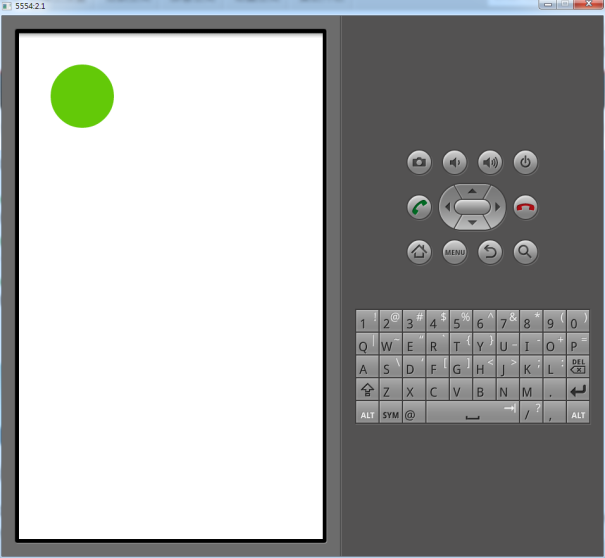
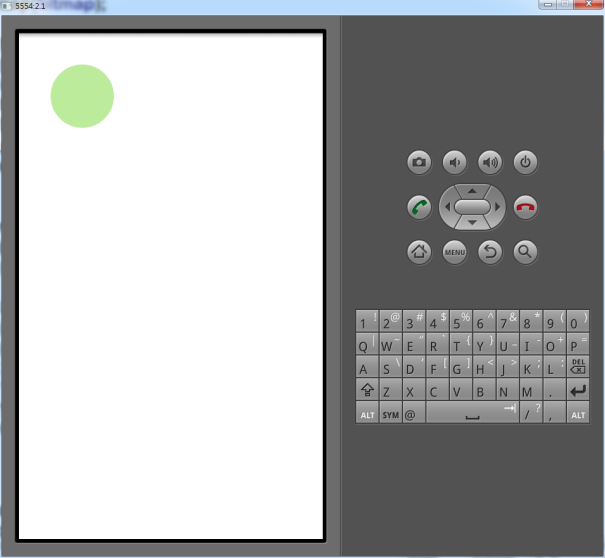


.setColor(Color.RED);

可以設定繪圖的顏色，這裡設置成紅色。

想要混色而不要單一顏色的畫也可以使用.setARGB(A,R,G,B);

或是.setAlpha(a); 調整透明度

透明度 : 200 透明度 : 100

.setAntiAlias(true);

決定線條是否平滑化，可以提高品質，但代價是效能會降低。

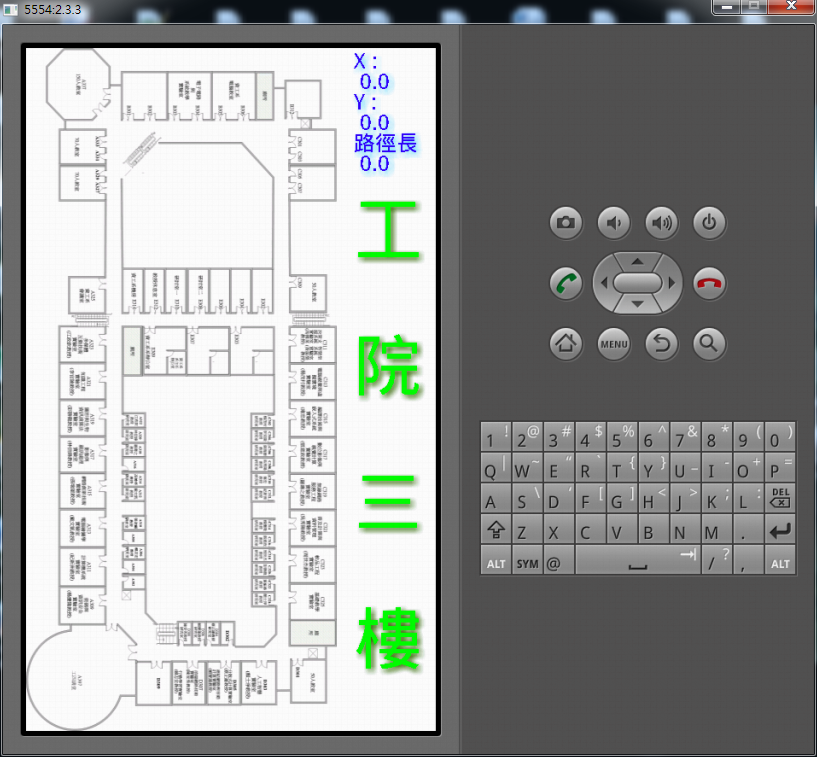
.setStrokeWidth(4);

決定筆刷大小

另外還有許多可以應用的部分來組合成特殊效果，如光影、漸層來做出不同的質感。

而以下我們綜合產生的效果 :





**3.5.2.增加效率**

設定好所有的bitmap、paint、canvas後，最後就是將他們依照你所想像的效果，依序產生在畫面上。

在撰寫程式的過程中，不管是書籍還是論壇，都提到一個重點:自繪操作是非常浪費處理器資源的一件事情，低效率的繪圖方式會影響GUI的執行緒，進而導致應用程式沒有回應甚至錯誤，需要撰寫者進行調整。

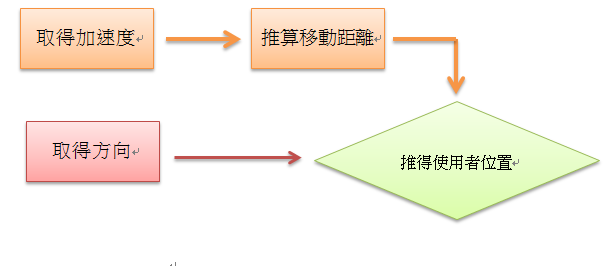
而有一種作法 : SurfaceView最常被使用在遊戲畫面繪製，因而廣泛的被拿來使用。在我們的程式中，雖然繪圖部分的使用並不像遊戲那麼的大量，但考慮到配合G-sensor的使用，抱持著能提升一點效率就是進步的想法來使用。

透過使用surfaceHolder.lockCanvas();與surfaceHolder.unlockCanvasAndPost();將步驟分成 (1)鎖定畫布(2)找到要繪圖的畫布(3)繪出畫布並解鎖畫布；也就是先把畫面定住，把要畫的部分放入另一個draw()或是Thread()理，等完成繪圖之後在匯出到畫布上繪出。

**3.6移動距離與方向的數值運算**

這個部分我們將分為3個大區塊來做詳細解說，分別是加速度取值、移動距離的計算、方向的判定，但在解說各個區塊之前，要先在此解說本程式在導航運算核心方面的程式撰寫想法。

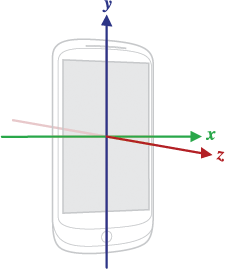
首先，在沒有GPS的情況下要導航，則我們得知當下位置的方式一定是透過偵測使用者的移動，來計算出使用者對於出發點的相對位置。藉由這種想法，我們只要讓手機可以偵測到使用者朝著哪個方向、移動了多少距離，便可以依照初始位置去推得現在的位置。



由於手機測量方位的方式為回傳手機Y軸正向所指向的方位，所以使用者必須把手機Y軸正向朝著自己移動的方向拿取，也因為如此，所以我們也是拿手機Y軸方向的加速度作為使用者移動的加速度，其內容將於後面章節詳述。

**3.6.1 加速度的取值與校正**

加速度取值的方法是利用Android的函式庫所提供的功能抓取手機X、Y、Z方向的加速度數值並存入程式的變數中進行運算。其X、Y、Z方向與手機方向之對應如下圖所示。

所以當程式抓取加速度時會有三個回傳值，

如下:

Sensor.TYPE\_ACCELEROMETER

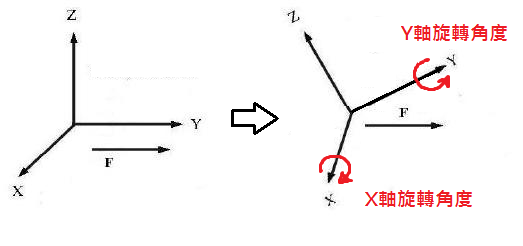
values[0] : X方向的加速度。

values[1] : Y方向的加速度。

values[2] : Z方向的加速度。

期中有一點必須特別注意，那就是手機還有一個內定的地心引力數值為-9.81(也有更精確的說法為-9.80665)，這個重力加速度的方向恆朝向地心的方向。換句話說，當手機水平靜止不動的時候values[0]=0 ，values[1]=0， values[2]=9.81。而當手機如上圖直立不動時values[0]=0 ，values[1]=9.81， values[2]=0。

由於我們程式的設計是以手機的Y軸正向作為人的前進方向，所以如果沒有適當的校正座標軸，那麼只要Y軸不為水平，則Y軸的加速度就不會是人移動的加速度了。而是會得到人移動之加速度的Y軸方向分量，甚至還會受到地心引力的Y軸分量影響。(以下為示意圖)



坐標軸的校正方面，我們是以乘上一個旋轉矩陣的方式把座標軸轉回來，以下是以旋轉X軸跟旋轉Y軸把座標軸規正的算式 :

而我們專題只需要讓Y軸維持水平即可，所以我們只旋轉X軸，把算式簡化為：

至於X軸與Y軸的旋轉角度是取至於手機的方位感測器，會於後面的章節詳細說明。

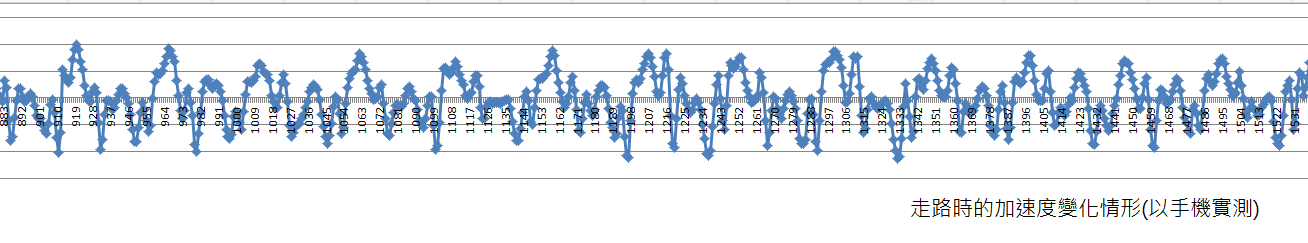
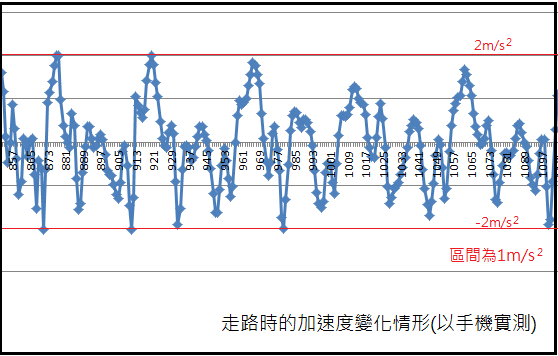
**3.6.2 移動距離的推算**

**3.6.2.1 最原始的測距想法**

我們最原始的測距想法相當單純，其步驟如下 :

1. 當開始程式時設定初速度為0，(V1 = 0)
2. 取得加速度數值 a
3. 取得時間區間 △T
4. 則可得知當下的速度V2，(V2=V1+ a\*△T )
5. 即可計算這個時間區間內移動的距離S，(S=(V1+V2)\*△T/2)
6. 之後將本次V2的值覆蓋到V1，在下一個時間區間中當初速度使用。
7. 重複上訴步驟(2)~步驟(6)。

用以上方法及可取的手機在每個時間間隔之中的移動距離，再配合方向感測器的數值即可得知手機的移動方位。

以下為走路與加速度變化的關係圖(以手機實測輸出的圖) :

由上圖可看出人走路時是由連續的加速減速相間所構成，也就是說人走路的速率，在正常情況下應該會是穩定的保持在某個區間內震盪，但是程式實際運作時卻常常發生速度不斷增加或是不斷遞減的情形，導致非常嚴重的誤差。

推斷誤差的原因如下 :

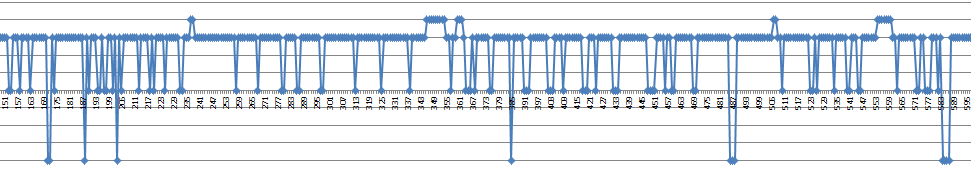
1. 重力感測器抓取加速度時的誤差。
2. 重力感測器抓取兩次加速度的時間區間內，真實的加速度仍然不斷在變化。
3. 計算速度的誤差。
4. 每次都必須依賴上一次所算出的速度當作本次時間區間的初速度，故誤差值會一直累積，而我們一個時間區間是重力感測器抓取一次加速度的時間，所以每秒鐘就有數十次的誤差累積。

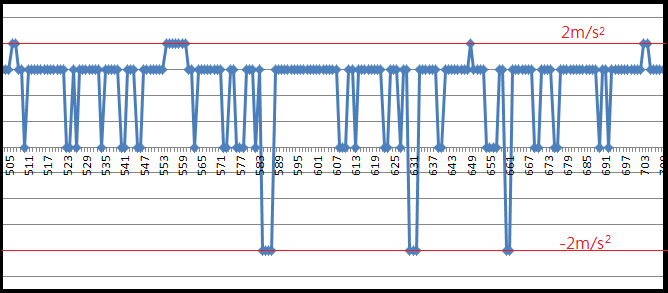
以上項目中以第4項最為嚴重，假設這一次的加速度取值導致速度運算後有些過大，則從此以後就算做到零誤差，可是基準點依然會一直是過大的狀態，也就是說速度一直都會大於真實速度。然而不可能做到零誤差，這也就是說我們的誤差會越來越大而且誤差的增加會越來越快。

為了解決這個問題所以我們提出了一個方法，就是讓加速度區間化。希望可也藉由區間化而忽略一些誤差值，防止它們持續累加。

**3.6.2.2 將加速度的值區間化**

將加速度區間化的想法為，希望藉由加速度的區間化來減少因為加速度和減速度取值的不對等所造成程式不斷加速或減速的機會。加速度區間化的方式就是抓取到加速度之後先判斷它是落在我們先制定好的加速度區間的哪一個區間，然後用這個加速度區間所對應的加速度來做運算。

以下為區間化之後，走路與加速度變化的關係圖(手機實測輸出)。



結果發現加速度區間化之後誤差不但沒有減少反而還略顯增加，我們推斷誤差的原因如下 :

1. 區間化加速度並不能真的解決誤差會隨著時間累加的問題。
2. 區間化加速度雖然可以讓數據變的單純，但是區間化這件事情本身就造成很大的誤差。

中和以上兩點可以發現誤差會更大是有道理的，這使得我們不得不想一個全新的測距想法來避免誤差在每一個時間區間之間不斷的累積。

**3.6.2.3 誤差值累加的解決**

要解決誤差值累加的問題，最根本的方式就是讓每一個時間間隔各自獨立，也就是說每一次的運算都不會對下一次的運算造成引響，所以在這裡我們參考了計步器程式的概念。

計步器程式的實作方式大約可分為兩種 :

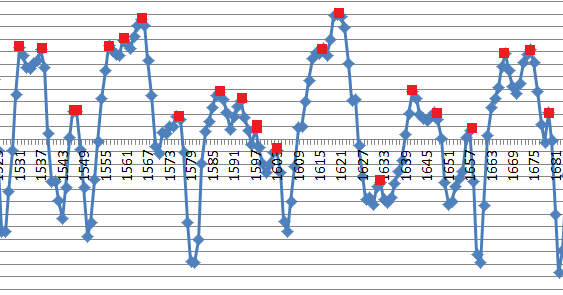
(1)預設好、或是由使用者輸入一步的距離，然後用測量加速度變化的方式來估測出使用者行走的步數，並藉此進一步地計算出使用者行走的距離。

(2)預定好、或是由使用者輸入移動速度，然後藉由測量加速度的方式判斷使用者是否有移動，一旦有移動，就用預先設定好的速度去當作使用者的移動速度，進而推出移動距離。

我們參考了第二種計步器的概念並進一步的強化。我們在程式裡預設多組的速度值，然後利用分析加速度變化的方式去選擇應該使用哪一組速度作運算。簡而言之就是程式會自行動態的去變更目前的速度設定。

**3.6.2.4 以查表代替實際運算**

以查表代替實際計算的想法是，先建立一張各種加速度變化的情形所對應的速度表，再抓取加速度時會去尋找區域性的加速度最大值，如果以圖形來說就是加速度折線圖的尖點(如下圖紅點)。



再利用抓到的區域性加速度最大值去查表，從表中找出對應的速率加以運算。

動態變更速度程式的實作方法 :

宣告三個變數 a1 , a2 , a3儲存加速度再設定一個v儲存速度(都預設為0)，當抓取加速度時a2會先將資料存入a3，然後a1 再將資料存入a2，最後在把抓取到的加速度值存入a1 ，如果a2的加速度值比a1跟a3還大則可得知a2是個一尖點，則可使用a2的值進行查表變更現有的速度。

**3.6.3 方向的判定**

由於需要判定方向，我們必須使用到手機中的方向感測器。而Android系統中判定方位使用的單位是角度。和加速度的取值方法一樣，當成是抓取方向時後會有三個回傳值，分別是：

­Sensor.TYPE\_ORIENTATION

value[0]：Y軸和北方所夾的角度(0~359)。北 = 0，東 = 90，南 = 180，

西 = 270 。

value[1]：Y軸與水平的夾角(-180~180)。且Z軸朝Y旋轉是正值。

value[2]：X軸與水平的夾角(-90~90)。Z軸朝X軸旋轉是正值。

**3.6.3.1 方向判定的區間**

當我們進入一個不能利用GPS定位的地方時，便得利用手機中的方向感測器來指出方向，方向感測器中的SensorEvent#value[0]回傳值是我們所需要的數值。而我們專題的實做，便用到方位感測器來做方向的定位。

如下表格所示：(東華大學工學院)

|  |  |
| --- | --- |
| SensorEvent#value[0] | Y軸的指向(人的前進方向) |
| 120~210 | 東華大學工學院A面 |
| 210~300 | 東華大學工學院B面 |
| 300~359 或 0~30 | 東華大學工學院C面 |
| 30~120 | 東華大學工學院D面 |

所以只要利用上面的方向資訊加上我們計算出來的移動距離，即可推出我們離出發點的相對位置。而出發的位置是在程式開始執行時由使用者手動輸入或是由QR Code讀取而來，故可藉此達成定位的目的。

**3.6.3.2 轉彎的容錯與校正**

在轉彎處，我們留有緩衝區。當進入緩衝區時，程式就會利用方向感測器來判斷是否應該轉彎，而不需要程式的運行結果剛好位於轉角處。

(緩衝區位置如右圖紅色圈圈所示)

緩衝區的計算方式如下：

1. 利用一段設定的區間，在區間內設定當使用者進入這區間時，會依照方向來判定使用者是否有轉彎的動作。
2. 如確定使用者轉向，則代表使用者目前處於轉角的位置，故地圖上的箭頭指標會校正至地圖上轉角位置上。
3. **結果與討論**

就結果而言，此概念的發展性極高；目前我們實做出東華大學工學院的慣性導航系統，在沒使用到GPS訊號的狀況下完成離線慣性導航。本程式一共利用了3種校正方法，分別是使用者觸控校正、轉彎校正與QR code使用，讓運算結果更佳的符合實際情形。在顯示地圖資訊方面也成功解決各種螢幕尺寸相容性的問題。

因為不需要接受到GPS的訊號也可進行導航，所以只要獲得地圖資訊，在任何環境、地點即可使用此方法讓自己不會迷失該地區，並配合QR code的使用，在重要的地點放入QR code，並存入該地點的資訊，讓使用者能夠立刻理解自己的所在地的訊息。

更近一步的推廣，未來搭配伺服器的建立，能夠即時更新伺服器所擁有的地圖資訊，並讓有需要的人能夠上傳地圖，進行客製化地圖，形成屬於自己的風格地圖。

1. **參考文獻**
2. Google Android 2.X應用程式開發實戰

作者:林城 (碁峯資訊)

1. 手機程式設計入門與應用Android,iphone,Windows Mobile

作者:陳彥文、彭亦暄 (旗標出版)

1. Android 2.X 應用程式開發經典

作者:鍾政欣 (碁峯資訊)

1. Google Android SDK開發範例大全

作者: 佘志龍、陳昱勛、鄭名傑、陳小鳳 (悅知文化)

1. 國立中央大學資訊工程學系碩士論文:

線性加速度感測器的研究與其應用

The Study of G-Sensor-based Systems and their

Applications

研究生: 伍星翰 指導教授: 蘇木春博士

1. Android電子書

<http://code.google.com/p/androidbmi/wiki/DiveIntoAndroid>