

Chapter 13: 行動計算

Prof. Yuh-Shyan Chen

Department of Computer Science and
Information Engineering

National Taipei University

大綱

13-1 行動與無線

13-2 行動化

13-3 行動計算架構

13-4 行動計算程式

13-5 行動計算應用與服務

13-6 行動計算的挑戰與研究議題

13-7 結論

13-8 習題

行動與無線

- 隨著無線通訊技術的發達，使我們能夠在無線網路覆蓋範圍內，任意移動都可以連上網路取得各種服務，也就是不需要依靠實體的線路就能連上網取得服務，因此有了行動計算的概念。
- 本節將簡單介紹無線通訊的種類，接著介紹行動 (Mobility) 的意涵，並從計算機 (Computer) 的歷史演進，探討行動與計算的關係，解釋何謂行動計算。
- 「無線」為不需要實體的線路，以無線電波做為傳遞介質，在覆蓋範圍內的地方移動，都可以不用透過實體線路連上Internet，因此無線技術的發展使行動計算的服務能夠愈來愈進步。

無線網路

- 根據通訊距離概分為無線個人網路、無線區域網路、無線都會網路、無線廣域網路。以下分別介紹上述各種無線網路：
 - 無線個人網路 (WPAN , Wireless Personal Area Network) 短距離的傳輸，傳輸範圍相當一個人的移動範圍，約一公尺到十公尺，標準為IEEE 802.15，傳輸協定包含藍芽、Zigbee、NFC、UWB。
 - 無線區域網路 (WLAN , Wireless Local Area Network) 傳輸距離比WPAN長，傳輸範圍為辦公室或校園，標準為IEEE 802.11n，傳輸協定如Wifi。

無線網路

- 無線都會網路 (WMAN , Wireless Metropolitan Area Network) 傳輸範圍為一個城市，傳輸協定如WiMAX。
- 無線廣域網路 (WWAN , Wireless Wide Area Network) 傳輸範圍可橫跨國家或不同的城市，傳輸協定如3G、LTE。

行動

- 行動 (Mobility) 為移動之意，包括被動與主動的移動。
- 被動的移動，如：紙本的書被人帶著走，供讀者閱讀與查找資料；電子化的字典放在使用者的口袋被帶著走。
- 主動的移動，如：公車，捷運，火車...等交通工具搭載乘客到目的地；美國iRobot公司出產的軍用掃雷機器人及掃地機器人，具有自主的移動能力。

計算

- 「計算」即為設計硬體或軟體系統將輸入的資料轉換為資訊與知識，進而變為智慧與智能。提供我們有用的資訊，達成特定的目的。
- 具有「計算」的硬體裝置稱為計算機或電腦 (Computer)。「算盤」大概可以說是最古老的計算機，取代數手指頭算數的方式，節省大量計算的時間

計算

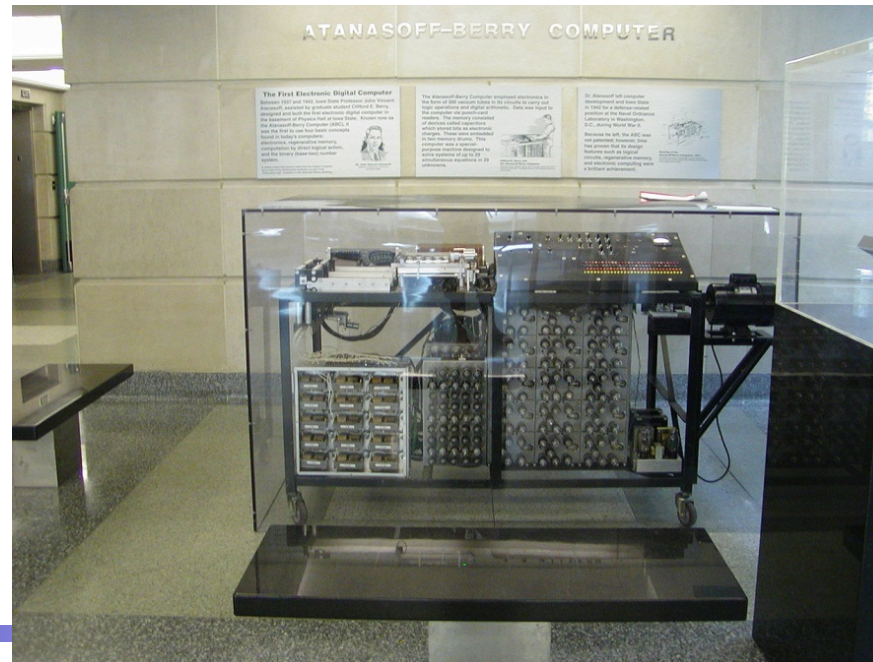
- 古希臘人在西元前一百年發明了一種名為安提基特拉機 (Antikythera Mechanism) 的天文電腦，由許多青銅齒輪組成，用來計算天體的運行週期；
- 西元1623年，德國科學家Wilhelm Schickard發明了能夠計算6位數加減的計算機；

計算

- 到了1890年代，打孔卡片計算機開始盛行，運作原理為在卡片上打孔以儲存資料，計算機讀取卡片後能夠自動計算大量的資料，當時最大的製造公司為現今的IBM公司，直到數十幾年前的程式設計師仍然必須將程式碼寫在打孔卡輸入到電腦執行。

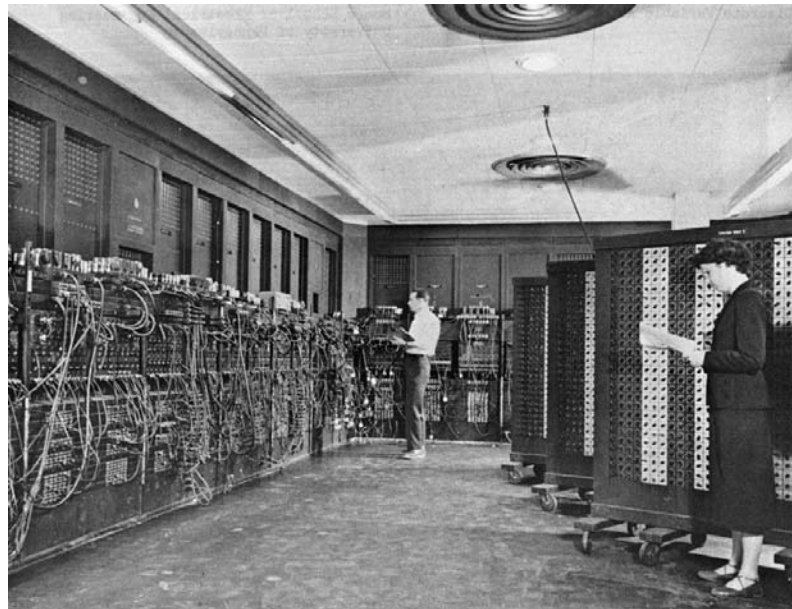
計算

- 到了1936年，美國科學家John Vincent Atanasoff及Clifford Berry發明了第一部電子電腦Atanasoff-Berry Computer，如，由真空管及繼電器的電路組成，不過並沒有可程式化 (Programmable) 的功能；



計算

- 1946年美國陸軍發展出第一部可程式化的通用型電子計算機Electronic Numerical Integrator And Computer，簡稱ENIAC，重達27,000 公斤，體積需要相當於一個大型的房間。



穿戴式計算

- 到了1960年代，Intel公司發明了由積體電路所組成的微處理器，縮小了電子計算機的體積，成本也降低許多；1980年代之後個人電腦開始盛行，接著筆記型電腦也開始發行，更有科學家如Steven Mann開始研究可穿式電腦。

Steve Mann's "wearable computer" and "reality mediator" inventions of the 1970s have evolved into what looks like ordinary eyeglasses.



行動載具與行動計算

- 直到今日，從平板電腦及智慧型手機的廣泛使用，我們可以發現電腦的體積愈來愈小，計算能力也愈來愈強，使我們可以拿在手上帶著走，需要時馬上可以使用，這是「行動計算」能夠蓬勃發展的主要原因之一。



行動計算

- 行動計算 (Mobile Computing)
- 透過行動載具將資訊運算後呈現給使用者，使用者能夠隨時隨地得到想要的資訊。例如：使用者到陌生的地方旅遊，能夠透過網路連上Internet使用位置導覽服務，得知所在位置附近的旅遊景點或加油站。
 -

行動載具與無線

- 「行動」可概分為非載具類與載具類，載具又稱為行動載具 (Mobile device)，包含行動電話 (Cellphone)、智慧型手機 (Smartphone)、掌上型電腦 (PDA)、筆記型電腦 (Laptop)、平板電腦 (Tablet) ...等等。
- 行動載具若使用無線通訊技術，便可以在無線通訊網路覆蓋範圍內，連上Internet，存取資料，或是取得網路服務，例如收發電子郵件。

行動 (Mobility) 類型

- 以下分別介紹各種行動 (Mobility) 的類型：
- 非載具類，如紙本的書、廣告宣傳單，可以放置的內容最少。
- 內容僅放在載具上，無法連上網路。如計算機、電子辭典、數位隨身聽、掌上型電玩、個人導航裝置、數位相機、數位攝影機。

行動 (Mobility) 類型

- 提供載具 (Thin Client) ，本身無儲存空間及運算能力，內容沒有放在載具上，應用程式與資料皆放在網路或後端的系統，可透過無線網路連上 Internet 進行使用或存取資料，因為無儲存空間及運算能力所以成本低，適合放置公共場所。如HP出產的精簡型電腦系列。

行動 (Mobility) 類型

- 載具、無線網路與雲端，行動載具透過無線網路使用雲端服務，為分散式運算的延伸，例如：
Dropbox雲端硬碟提供網路空間使行動載具可以上傳與存放資料。可以根據需求選擇不同的網路空間大小。

行動 (Mobility) 類型

- 資訊帶著走，資訊的移動而非載具的移動，無論在任何時間，任何地點，可以透過多元化的載具取得需要的資訊。多元化的載具表示任何地方的物體都可能可以當作載具使用，因此不需要隨身攜帶載具，直接使用附近的載具取得需要的資訊。

行動與無線

- 「無線」為不需要實體的線路，以無線電波為傳輸介質，無線技術造就了行動計算的發展
- 「行動計算」透過行動載具將資訊運算後呈現給使用者，使用者能夠隨時隨地得到想要的資訊概分為載具類與非載具類

行動與無線

- 非載具類即為一般實體的紙本資料，載具類即透過載具或稱為行動載具，能夠將大量的資料放在上面，讓人移動攜帶使用，若能連上Internet，可以得到更多服務，如資料更新或資料儲存在網路，架構可能為Client / Server及雲端網路。
- 未來的目標希望資訊能夠不透過載具移動，在任何地方都能夠以附近的物體當作載具取得服務。

行動化 (Mobilization)

- 行動化的技術
 - 定位技術 (Localization & Positioning)
 - 位置感知計算 (Location-aware Computing)
 - 情境感知計算 (Context-aware Computing)
 - ...
- 實現行動計算的過程稱為行動化 (Mobilization)

行動化 (Mobilization)

- 定位技術 (Localization) ：可分為一般定位與室內定位，
 - 一般定位為透過全球定位系統 (GPS) 來測量所在位置，但沒辦法在室內使用
 - 室內定位必須透過3G, Wifi, RFID...等定位方法才能測量在室內測量所在位置

行動化 (Mobilization)

- 位置感知計算 (Location-aware Computing)
 - 透過定位技術知道使用者的位置後，提供與該位置有關係的位置資訊。例如將位置資訊傳給Google Map網路服務，便可以看到該位置的3D街景圖。

行動化 (Mobilization)

- 情境感知計算 (Context-aware Computing)

- 結合定位與感測技術，提供與使用者更多與周遭環境有關的資訊或建議，例如在博物館內，系統可以根據使用者在室內的位置，提供適合的導覽路線。

行動計算目前有三大限制

- 第一個限制為無線網路的限制，包含了異質網路中的換手，以及行動裝置可能無法隨時連上網路造成高頻率的斷線，以及網路頻寬的限制。
- 第二個限制為行動裝置的限制，由於電池電力有限、硬碟或記憶體的容量有限，以及手機遭到竊取或遺失的不可依賴性，以及手機時常沒電關機的不可使用性所造成。
- 第三個為行動架構上的限制，目前還沒有完善的行動感知 (Mobility-Awareness) 的應用程式架構及系統架構能夠充分支援行動計算的環境。

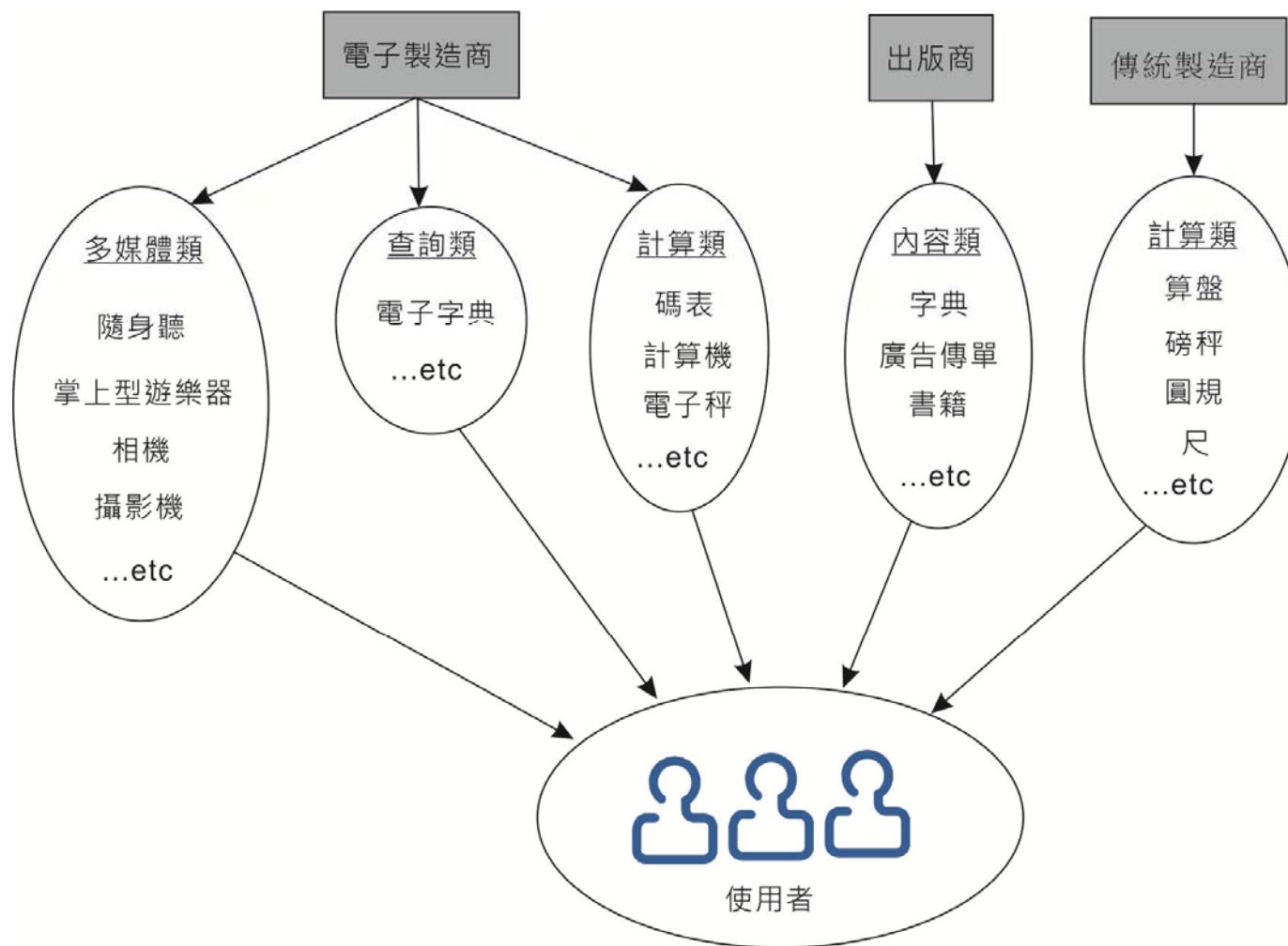
定位技術

- 定位技術是行動計算重要的基礎，他能夠知道移動中的使用者的即時位置，有了定位技術之後，才能提供位置感知計算的服務，提供即時位置相關的任何資訊，如附近景點路線查詢，隨著智慧型手機的普及，位置感知計算漸漸發展成熟，有許多方便的導航服務能夠使用，未來發展的方向為情境感知計算，資訊會跟著使用者移動，隨時隨地都能夠服務使用者。

行動計算架構

- 目前常看到的三種行動計算架構
 - 單機版
 - Client / Server行動計算的架構
 - 雲端網路行動計算的架構

圖13.1:單機版行動計算架構



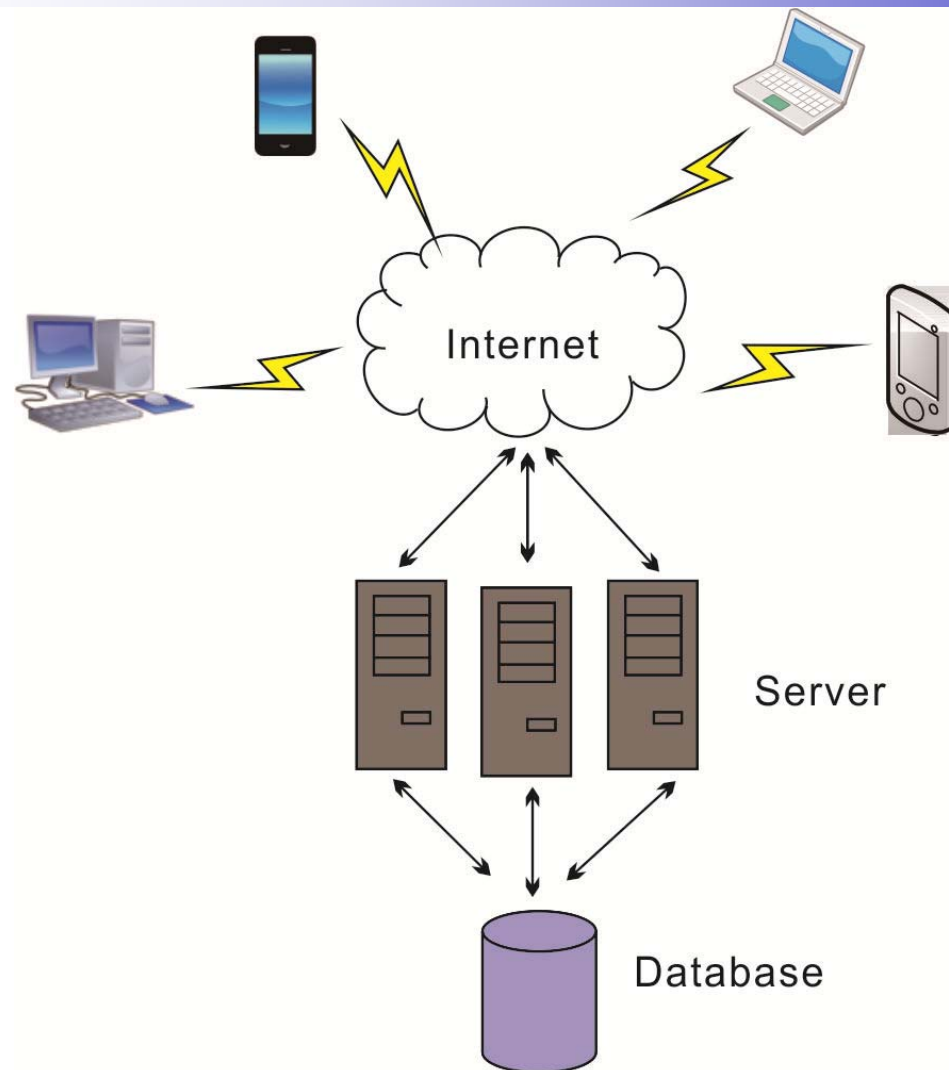
單機版行動計算架構

- 單機版行動計算架構：單機版行動計算的載具從工廠生產的時候，就已經把功能和內容寫在裡面，而且不具有連網的功能，所以使用者拿到成品時無法透過網路下載軟體改變其功能與內容。
- 大致可以分為電子產品類及非電子產品類。
- 電子產品類載具由電子製造商生產，又可分為多媒體類、查詢類、計算類，多媒體載具類例如隨身聽、掌上型電玩、相機、攝影機…等等。查詢類如早期單機版的電子字典，計算類即電子測量工具如碼表、計算機、電子秤…等等，這些都是電子類的單機載具。

單機版行動計算架構

- 非電子類載具可分為內容類及計算類，內容類即出版商印刷的紙本內容如字典、廣告傳單、書籍…等等，計算類即為傳統製造商生產的非電子計算工具如算盤、磅秤、圓規、尺…等等。使用者能夠隨身攜帶這些單機版的行動載具，進行計算與查看內容但無法連上網路更新功能與資料。

圖 13.2: Client / Server行動計算架構



Client / Server行動計算架構

- 使用者可利用平板電腦 (Tablet)、筆記型電腦 (Laptop)、智慧型手機 (Smartphone)、掌上型電腦 (PDA) ...等行動載具 (Client端) 透過無線網路基地台 (Wireless AP) 連上網路，與伺服器 (Server端) 做資料傳輸，傳送要求給Server，Server會從資料庫找出相關資料，經過計算後回傳給Client端的使用者。

圖13.3:雲端網路行動計算架構



雲端網路行動計算架構

- 行動載具如平板電腦 (Tablet)、筆記型電腦 (Laptop)、掌上型電腦 (PDA)、智慧型手機 (Smartphone)，無論使用者身處何方，只要附近有無線網路基地台，就可以連上雲端網路，我們不需要知道雲端背後的主機如何運作，便可以直接使用服務，如社群網站Facebook、微軟及亞馬遜公司的虛擬伺服器販售、或是Google的雲端應用程式Gmail及Google Doc。

行動計算架構

- 行動計算的架構逐漸由單機版架構轉變為Client / Server架構，使用者能夠連上網將更多的內容下載下來閱讀，或下載應用軟體使載具的功能能夠擴充，目前行動計算架構正漸漸往雲端行動計算架構前進，未來將能夠實現無論任何時間、任何地點都能夠使用雲端服務，達到無所不在的運算。

行動計算程式

- 行動計算程式分為兩個部分，分別是行動計算作業系統與行動計算程式
- 第一部分將介紹行動載具常使用的作業系統，並以其中的Android介紹基本程式撰寫。
 - 支援行動載具的作業系統有Apple公司開發的iOS；Microsoft公司開發的Windows Phone 7，代號為Mango；Research In Motion (RIM) 公司開發的BlackBerry OS；Nokia公司開發的Symbian；Google公司開發的Android... 等等。其中Android及Symbian為開放式平台，其原始碼開放供人下載。

行動計算程式

- 第二部分將實作三種基於Android作業系統的應用程式，分別是軌跡球、資料庫連線、Google Map的Android應用程式，將會介紹各個程式的設計原理、並解說幾段重要的程式碼如何撰寫。

行動載具作業系統

- Android：以Linux為核心的作業系統，目前由Google所擁有，以開放原始碼的方式發布，2007年推出第一台搭載Android作業系統的手機，目前的占有非常高的市占率，下圖為Android Logo。



行動載具作業系統

- BlackBerry OS：為Research In Motion公司為自家智慧型手機BlackBerry開發的作業系統，最大的特色是處理電子郵件的能力，目前最新版本為6.0。下圖為BlackBerry OS Logo。



行動載具作業系統

- iOS：由Apple為自家智慧型手機iPhone及多媒體播放器iPod Touch所開發的作業系統，重視使用者體驗，提供簡單又漂亮的使用者介面，目前也占有很高的市占率。



行動載具作業系統

- Windows Phone：Microsoft所開發的作業系統，開發代號為Mango，使用者圖形介面使用Metro介面，是由許多正方形或長方形組成，稱為動態磚。



行動載具作業系統

- Symbian：原本是Symbian為手機開發的作業系統，目前是Nokia所擁有，以開方原始碼的方式發布，智慧型手機盛行之前，Symbian的市占率具有很高的地位，雖然目前的最新版也支援智慧型手機，但市占率已經不如從前。圖13-13為Symbian OS Logo。



行動計算程式

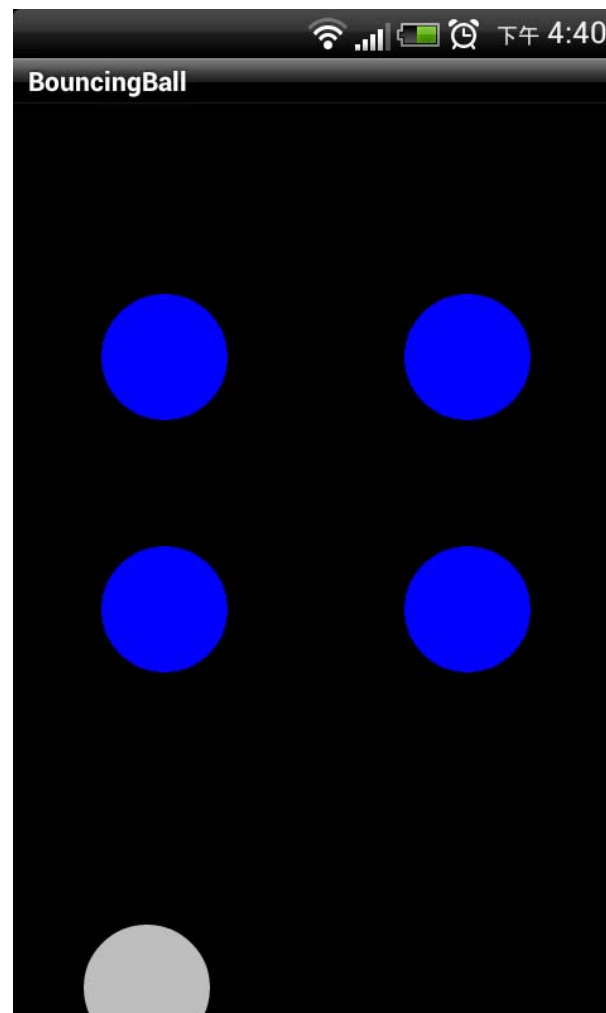
- 開發不同的行動計算作業系統應用程式，需要不同的編譯器及開發工具，例如開發Android應用程式，必須安裝JAVA的JDK開發工具包，裡面提供JAVA的編譯器，以及安裝Android提供的Android SDK開發工具包。若要開發iOS的程式通常需要一台Apple電腦或筆電，早期使用的編譯器為GCC，不過Apple近年推出新的LLVM編譯器，效能比GCC更佳。在本章節中，將以JDK及Android SDK為工具，撰寫簡單的Android應用程式。

行動計算程式

- 範例一：平衡球遊戲
- 目的：利用Android手機內建的G-Sensor（三軸加速規），根據手機的傾斜狀態，即時繪製一個圓形，模擬在真實環境中有一顆球在手機上面隨傾斜而滾動。
- 畫面上會有四個藍色圓圈，玩家必須傾斜平衡球經過所有藍色圓圈，經過的圓圈會從藍色變成紅色，當所有圓圈都變成紅色，也就是經過了所有藍色圓圈任務即達成，完成後會顯示花了多少時間

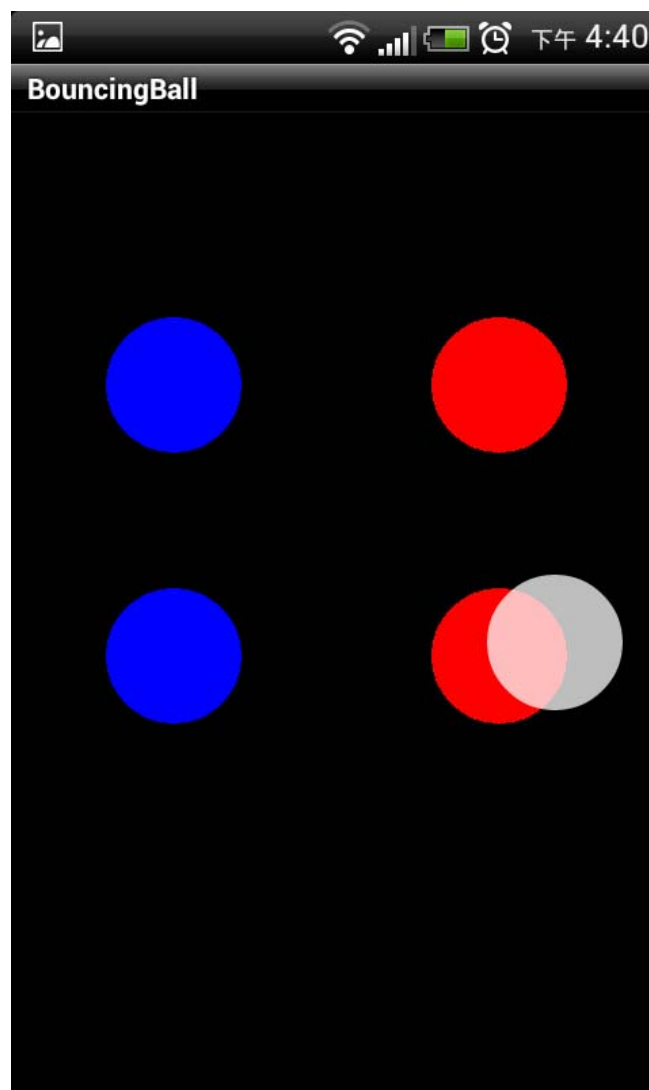
範例一：平衡球遊戲

- 遊戲一開始，可以透過傾斜手機移動白色圓形。目標是在最快時間內經過四個藍色圓形。



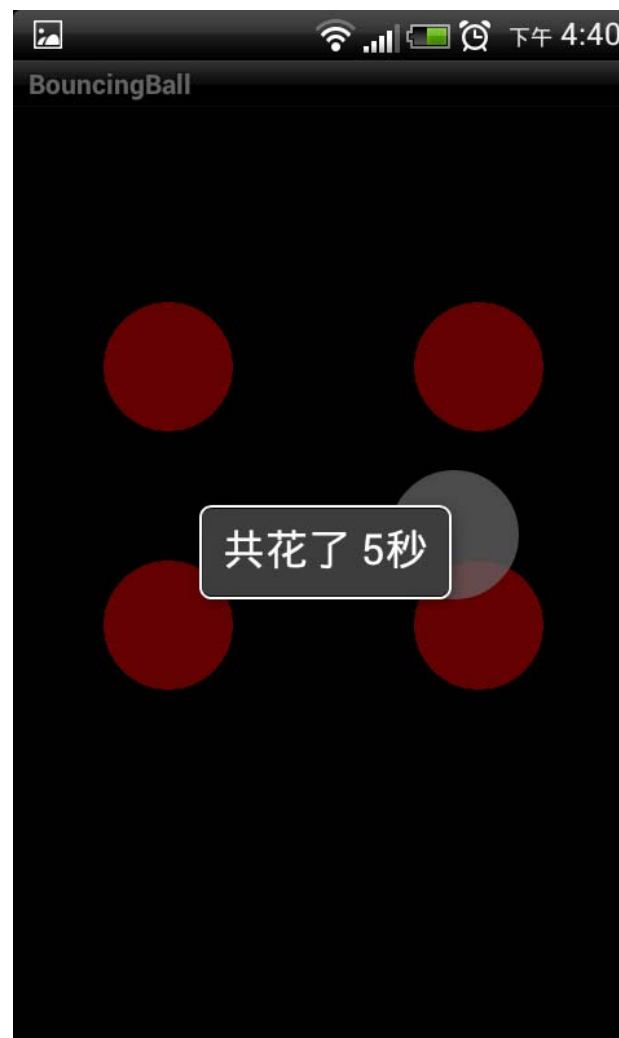
範例一：平衡球遊戲

- 經過的藍色圓形會變成紅色。



範例一：平衡球遊戲

- 當所有藍色圖形都變成紅色時，遊戲結束，程式會計算總共花了多少時間。



範例一：平衡球遊戲

- 如何實作
- 取得三軸加速規的值
- 計算加速度以及平衡球的位移
- 更新平衡球的位置
- 繪製四個藍色的洞及平衡球於螢幕
- 當平衡球經過藍色的洞，洞的顏色變成紅色
- 計算從程式開始到四個洞變紅色的經歷時間

範例一：平衡球遊戲

- 程式碼如下

```
...  
    /  / 宣告必要變數  
1. private int mXCenter;  
2. private int mYCenter;  
3. private RectF mRectF;  
4. private final Paint mPaint;  
5. private float mVx;  
6. private float mVy;  
...
```

範例一：平衡球遊戲

- 取得三軸加速規的值並計算加速度的值

```
1  ...
2  public void onSensorChanged ( SensorEvent event ) {
3    // 取得三軸加速規的值
4    mAx = event.values[0];
5    mAy = event.values[1];
6    float mAz = event.values[2];
7    // 計算加速度
8    mAx = Math.signum ( mAx ) * Math.abs ( mAx ) * ( 1 - FACTOR_FRICTION
9      * Math.abs ( mAz ) / GRAVITY );
10   mAy = Math.signum ( mAy ) * Math.abs ( mAy ) * ( 1 - FACTOR_FRICTION
11     * Math.abs ( mAz ) / GRAVITY );
12 }
13 ...
```

範例一：平衡球遊戲

● 計算平衡球的位置

```
1. ...
2. public boolean updateOvalCenter ( ) {
3.     mVx -= mAx * mDeltaT;
4.     mVy += mAy * mDeltaT;
5.     mXCenter += ( int ) ( mDeltaT * ( mVx + 0.5 * mAx * mDeltaT ) );
6.     mYCenter += ( int ) ( mDeltaT * ( mVy + 0.5 * mAy * mDeltaT ) );
7.     if ( mXCenter < RADIUS ) {
8.         mXCenter = RADIUS;
9.         mVx = -mVx * FACTOR_BOUNCEBACK;
10.    }
11.    if ( mYCenter < RADIUS ) {
12.        mYCenter = RADIUS;
13.        mVy = -mVy * FACTOR_BOUNCEBACK;
14.    }
15. }
16. ...
```

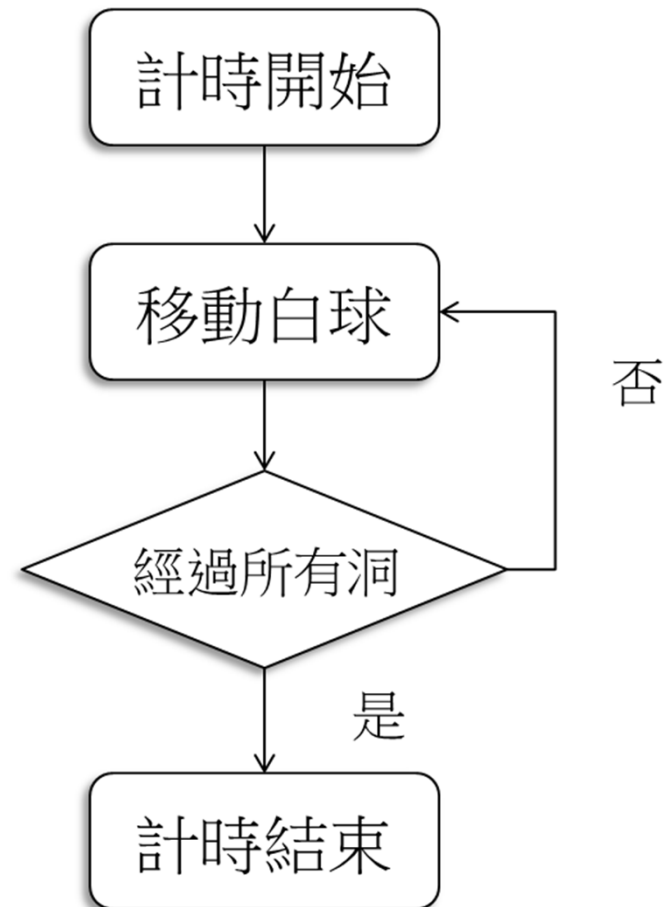
範例一：平衡球遊戲

● 繪製平衡球

```
1. ...
2. protected void onDraw ( Canvas canvas )
3. {
4.     if ( mRectF != null ) {
5.         ...
6.         canvas.drawColor ( 0xFF000000 );
7.         設定平衡球的繪圖位置
8.         mRectF.set ( mXCenter - RADIUS,
9.                     mYCenter - RADIUS,
10.                    mXCenter + RADIUS,
11.                    mYCenter + RADIUS );
12.         繪出平衡球
13.         canvas.drawOval ( mRectF, mPaint );
14.     }
15. ...
```

範例一：平衡球遊戲

- 執行流程
- 遊戲計時開始
- 移動白球到每個藍色的圓形（洞）
- 若所以藍色圓形變成紅色（經過所有洞）
- 遊戲結束，計算花費時間
- 否則繼續遊戲



範例二：備忘錄存取Mysql資料

- 目的：這個程式的主要功能是備忘錄與Mysql資料庫做連線，使用者可以在文字框輸入備忘錄，送出後會儲存於Mysql資料庫並且將資料庫的每一筆備忘錄顯示在手機的螢幕上面，每一筆備忘錄包含編號、日期、備忘事項。

範例二：備忘錄存取Mysql資料

- 右圖為程式執行實際畫面的截圖



範例二：備忘錄存取Mysql資料

- 在上方的文字欄輸入”慢跑”，按下送出鍵後，程式會到資料庫新增這筆備忘錄，並顯示於螢幕。



範例二：備忘錄存取Mysql資料

- 手機螢幕會顯示備忘錄資料庫中每一筆資料。



範例二：備忘錄存取Mysql資料

- 手機端會新增資料到備忘錄資料庫中。

Show 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

Sort by key: None Go

			log_id	update_time	memo_data
☐			1	2012-10-11 03:22:39	慢跑
☐			2	2012-10-11 03:23:52	圖書館還書
☐			3	2012-10-11 03:24:35	寄信
☐			4	2012-10-11 03:24:57	健身房

Check All / Uncheck All With selected:

Show 30 row(s) starting from record # 0
in horizontal mode and repeat headers after 100 cells

範例二：備忘錄存取Mysql資料

- 如何實作
- PHP程式儲存資料於Mysql，並從Mysql取出備忘錄的所有資料。
- 手機傳送使用者輸入的備忘錄到PHP程式並回傳資料庫裡每一筆備忘錄資料。

範例二：備忘錄存取Mysql資料

● PHP程式與Mysql資料庫連線

```
1. ...  
2. <?php  
3. $database = "資料庫名稱";  
4. $username = "帳號";  
5. $password = "密碼";  
6. // 與資料庫連線  
7. $dblink = mysql_pconnect ( "localhost", $username, $password ) or  
8. trigger_error ( mysql_error ( ), E_USER_ERROR );  
9. mysql_query ( "SET NAMES utf8", $dblink );  
10. mysql_query ( "SET CHARACTER_SET_CLIENT=utf8", $dblink );  
11. mysql_query ( "SET CHARACTER_SET_RESULTS=utf8", $dblink );  
12. mysql_select_db ( $database, $dblink );  
13. ...
```

範例二：備忘錄存取Mysql資料

● PHP程式存取Mysql資料庫

```
1. ...
2. // 若使用者輸入備忘錄資料
3. if ( strcmp ( trim ( $data ) , "" ) !=0 )
4. {
5.     // 輸入進資料庫
6.     $insertSQL = sprintf ( "INSERT INTO `memo` ( `memo_data` ) VALUES ( '%s' );",
        $data );
7.     mysql_query ( $insertSQL, $dblink ) or die ( mysql_error ( ) );
8. }
9. // 從資料庫取出所有備忘錄資料
10. $query_rs = "SELECT log_id, DATE_FORMAT ( update_time, '%Y-%m-%d' ) as
        update_time, memo_data FROM `memo` order by log_id ";
11. $rs = mysql_query ( $query_rs, $dblink ) or die ( mysql_error ( ) );
12. ...
```

範例二：備忘錄存取Mysql資料

- PHP程式回傳JSON格式的資料

```
1. ...
2. $rows = array ( );
3. $arr = array ( );
4. // 將資料庫資料存成陣列
5. while ( $r = mysql_fetch_assoc ( $rs ) ) {
6.     foreach ( $r as $key => $value )
7.     { $arr[$key] = $value; }
8.     $rows[] = $arr;
9. }
10. // 以JSON格式回傳
11. print json_encode ( $rows );
12. ...
```

範例二：備忘錄存取Mysql資料

● 手機傳資料到PHP程式

```
1. ...
2. private String sendPostDataToInternet ( String strTxt ) {
3. ...
4. // 傳送資料到PHP程式
5. String result = sendPostDataToInternet ( msg );
6. if ( httpResponse.getStatusLine ( ) .getStatusCode ( ) == 200 ) {
7.     StringBuilder builder = new StringBuilder ( );
8.     // 利用緩衝器暫存回傳資料
9.     BufferedReader bufferedReader = new BufferedReader (
10.         new InputStreamReader ( httpResponse.getEntity ( ) .getContent ( ) ) );
11.     // 讀取回傳字串並儲存
12.     for ( String s = bufferedReader.readLine ( )
13.         ;s!=null;s=bufferedReader.readLine ( ) )
14.         builder.append ( s );
15.     // 回傳JSON格式回應字串到螢幕
16.     return builder.toString ( );
17. }
```

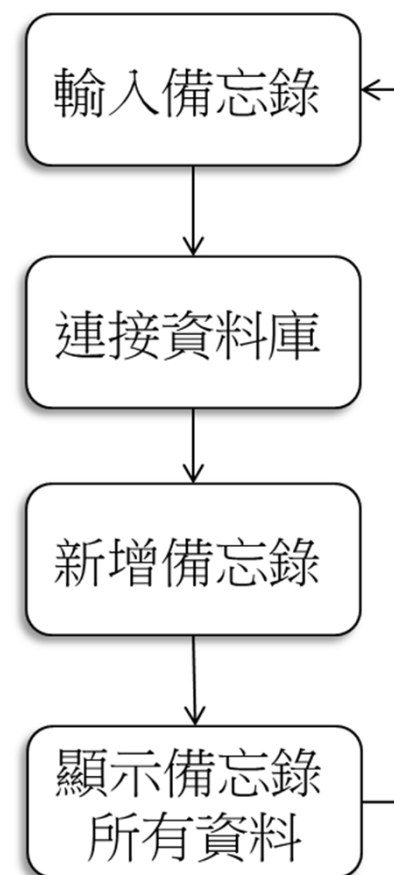
範例二：備忘錄存取Mysql資料

- 手機接收JSON格式資料，並顯示於螢幕

```
1. ...
2. if ( result != null ) {
3.     try {
4.         json_data = new JSONArray ( result ) ;
5.         for ( int i = 0; i < json_data.length ( ) ; i++ ) {
6.             // 取出JSON物件
7.             stock_data = json_data.getJSONObject ( i ) ;
8.             // 取得物件內資料
9.             result2 += ( i+1 ) +" "+stock_data.getString ( "update_time" )
10.             +" "+stock_data.getString ( "memo_data" ) +"\\n";
11.         }
12.     } catch ( JSONException e ) {
13.         e.printStackTrace ( ) ;
14.     }
15.     // 顯示於手機螢幕
16.     resultMessage.setText ( result2 ) ;
17. }
18. ...
```

範例二：備忘錄存取Mysql資料

- 執行流程
- 使用者輸入備忘錄資料
- 程式連結資料庫並新增資料
- 顯示備忘錄資料庫的資料



範例三：實作與Google Map相關的應用程式

- 目的:獲得行動載具的位置，搜尋該位置附近符合篩選條件的結果。在這個範例裡，篩選條件簡單的設定為XX店家。

範例三：實作與Google Map相關的應用程式

- 右圖為程式執行實際畫面的截圖，為程式的起始畫面



範例三：實作與Google Map相關的應用程式

- 可以拉開選單並選擇要進行哪一種類的搜尋



範例三：實作與Google Map相關的應用程式

- 我們以搜尋美食來作範例
- 可以選擇要搜尋哪一種類的餐廳



範例三：實作與Google Map相關的應用程式

- 右圖為搜尋完成後的畫面
- 其中藍色原點為行動載具(自己所在)，散落在地圖上的小箭頭為篩選出來的結果



範例三：實作與Google Map相關的應用程式

- 任意點選店家（藍色箭頭）會出現功能選單，可以進行一些動作
- 在此將店家的重要資訊遮擋起來，以免資料外流



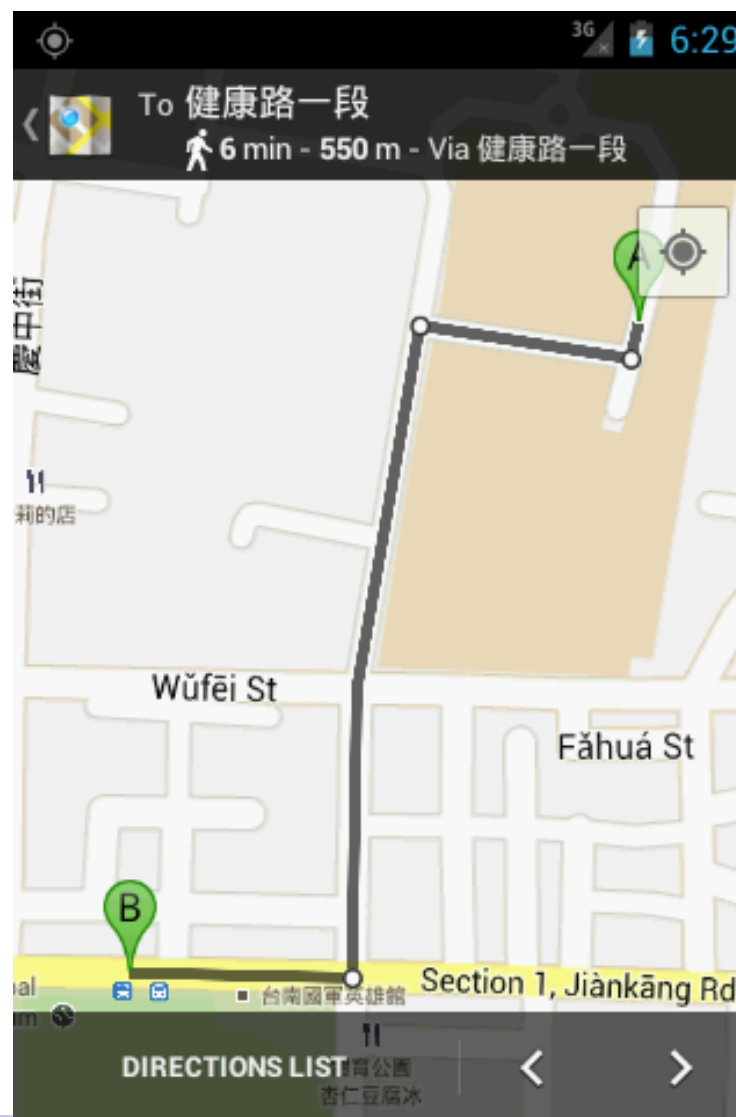
範例三：實作與Google Map相關的應用程式

- 進行路線規劃的時候會出現交通方式選項提供給使用者作參考



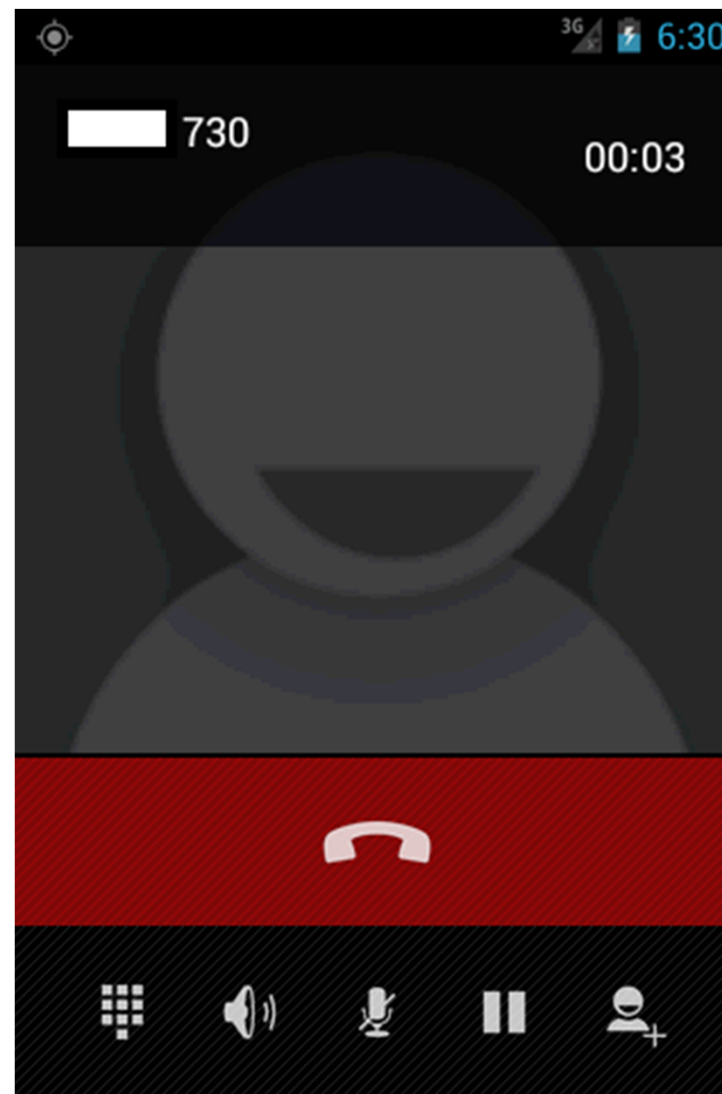
範例三：實作與Google Map相關的應用程式

- 接著進行導航的工作



範例三：實作與Google Map相關的應用程式

- 也可以直接撥打電話給該店家



範例三：實作與Google Map相關的應用程式

- 也可以加入我的最愛，可以隨時提醒自己多多光顧此店



範例三：實作與Google Map相關的應用程式

- 如何實作：
- 必須先獲得現在位置資訊。
- 利用獲得的位置資訊去執行各種動作。
- 演算法如下：

```
    / / 獲得位置資訊  
    getLocation();  
    do{  
        / / 利用位置資訊執行動作  
    }
```

範例三：實作與Google Map相關的應用程式

- 建立了存放店家資訊的簡易資料庫。
- 存放資料包含：地址、店名、電話等等，以利更多應用。
- 程式碼如下：

```
...  
  /  / 宣告必要變數  
1. MapView mapView;  
2. MapController mc;  
3. LocationManager locMan;  
4. MyLocationOverlay myLocOverlay;  
...
```

範例三：實作與Google Map相關的應用程式

```
1. ....
2. // 啟用位置偵測
3. myLocOverlay.enableMyLocation ( ) ;
4. // 若位置有改變則執行以下執行緒裡的動作
5. myLocOverlay.runOnFirstFix ( new Runnable ( ) {
6. @Override
7. public void run ( ) {
8. // TODO Auto-generated method stub
9. // 搜尋目標的方法
10. listTarget ( ) ;
11. mc.animateTo ( myLocOverlay.getMyLocation ( ) ) ;
12. mc.setZoom ( 16 ) ;
13. } } ) ;
14. ListOfOverlays.add ( myLocOverlay ) ;
15. // 繪製地圖
16. mapView.invalidate ( ) ;
```

範例三：實作與Google Map相關的應用程式

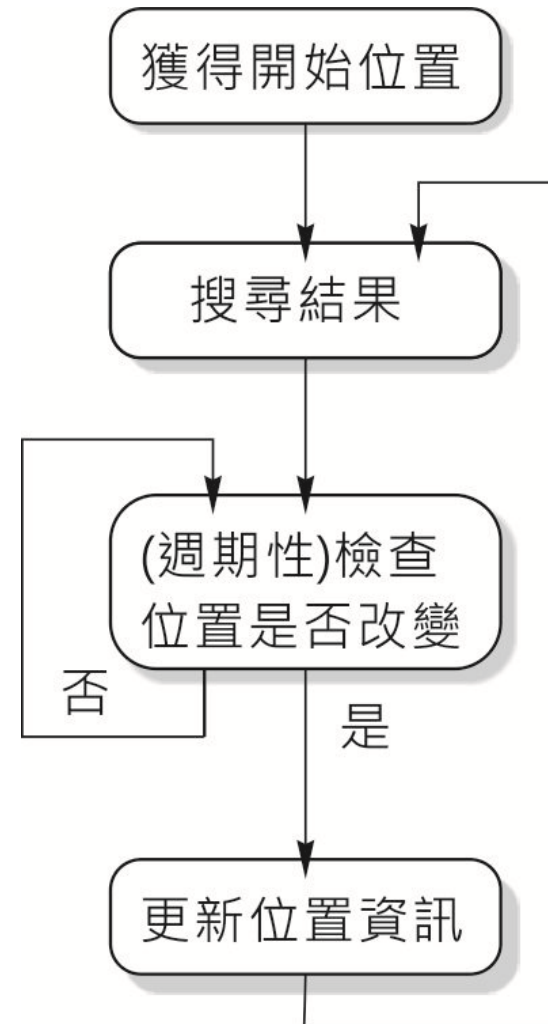
```
1. public void listTarget ( ) { // 搜尋結果的方法
2. ...
3. DriverManager.getConnection ( "jdbc:mysql: / / /
   glfinf?useUnicode=true&characterEncoding=utf8", "UserID", "PASSWORD" );
4. rs = // 獲取資料庫裡的資料，設定搜索條件 ( 搜索範圍 )
5. stat.executeQuery ( "SELECT * FROM table_name WHERE " + 搜索條件 ) );
6. while ( rs.next ( ) ) { // 當有資料則將結果標示在地圖上
7.     // 標示結果方法
8.     setTarget ( rs.getDouble ( "latitude" ) , rs.getDouble ( "longitude" ) );
9. }
10. ...
```

範例三：實作與Google Map相關的應用程式

```
1. public void setTarget ( double lat, double lon ) { // 將結果標示在地圖上的方法
2. List<Overlay> ListOfOverlays = mapView.getOverlays ( );
3. // 自定義的目標圖標，主要就是更改座標圖片，簡單複寫即可
4. TargetItemizedOverlay data = new
5. TargetItemizedOverlay ( getResources ( ).getDrawable ( R.drawable.shop ) );
6. // 利用或的的座標對應到地圖上
7. GeoPoint shopLoc = new GeoPoint ( ( int ) ( lat * 1E6 ), ( int ) ( lon * 1E6 ) );
8. OverlayItem overlayitem = new OverlayItem ( shopLoc, "", "" );
9. data.addOverlay ( overlayitem );
10. if ( ! ( data.size ( ) == 0 ) )
11.     // 將標示加入地圖
12.     ListOfOverlays.add ( data );
13. }
```

範例三：實作與Google Map相關的應用程式

- 執行流程：
- 程式一開始必須獲得使用者。
- 位置資訊才可以進下下一步動作。
- 利用位置資訊去搜尋結果。
- 需要週期性的檢查使用者位置。
- 有變動必須更新位置資訊並且重新搜尋新的結果。



行動計算程式

- 開發行動運算應用程式有許多不同的資源技巧。
- 比如想開發3D遊戲：
 - OpenGL
 - 硬體功能
 - 觸控板
 - 相機

行動計算程式

- 開發完應用程式之後試著把自己的應用程式上傳到 App Store 上面。
- (iOS : App Store, Android : Google Play, Windows Phone : Windows Phone Store) 。
- 訓練開發者的開發能力。

行動計算應用與服務

- 傳統只能撥打電話的手機使用經驗。
- 把行動計算想成傳統電話的延伸。
- 使用者不只可以隨時隨地的打電話，也可以行動載具本身的計算能力或是透過網路享受到各種服務。
- 行動定位服務 (Location-Based Service) 。
- 行動社交網路服務 (Mobile Social Network Service) 。
- 行動多媒體影音服務。
- 行動載具硬體服務整合互動應用。
- 行動雲端服務 (Mobile Cloud Service) 。

行動計算應用與服務

- 行動定位服務 (Location-Based Service)
- 行動定位服務簡單來說便是「以位置為基礎的服務 (LBS) 」。
- 利用位置資訊來提供各種服務。
- 「定位 (Location) 」不是只有位置，而是時間與位置的集合。
- Location代表著哪個時間點使用者在哪裡。
- LBS首先必須要取得定位資訊，然後再使用定位資訊提供服務。

定位機制

- 取得定位資訊的方法通常利用下列幾種：
 - GPS
 - GLONASS
 - AGPS

GPS – Global Positioning System

- 全名為全球的位系統。
- 由美國國防部研發與管理，可以滿足全球精準三維定位的需求。
- 只需要擁有GPS接收器便可以接收訊號使用該服務，不需要另外收費。
- 早期美軍方面擔心民用的GPS訊號會被敵對國家或組織利用來發動攻擊，所以加入了SA訊號 (Selective Availability) 來降低民用訊號的精準度 (大約100公尺或更差) 。
- 2000年後取消了SA訊號，因此民用的GPS現在也可以達到10公尺左右的定位精準度。

GLONASS – Global Navigation Satellite System

- 又稱全球導航衛星系統。
- 由俄羅斯開發與管理，類似美國的GPS，其定位精準度可達到15公尺。
- 預計在2012年將衛星數目增加以提高定位精準度，預計可達1公尺。
- 普及度遠遠不及GPS，可能原因是俄羅斯方面長久以來並不注重民用市場。
- 近來iPhone4S, Sony Xperia Ion等等裝置皆有搭載GPS與GLONASS結合的雙定位系統。
- 雙定位系統可讓定位更加精準快速以及有效的解決因為訊號干擾而無法定位的問題。

表13.1: GPS與GLONASS的差異

參數	GPS	GLONASS
衛星數目	24	24 (預計增加到30)
軌道面數	6	3
傾角	55度	64.08度
一圈週期	20180公里	19100公里
重複地面軌跡時間	11小時58分	11小時15分
大地基準	WGS84	SGS85 (PE-90)
時間系統改正	相對於UTC[USNO]	相對於UTC[SU]
星曆廣播	每12.5分37500bits	每2.5分7500bits
訊號區分	以碼區分	以頻率區分
L1頻帶	1575.42MHz	1.602~1.615MHz
L2頻帶	1227.60MHz	1.246~1.256MHz
時錶資料	時間偏移,頻率偏移加漂移 (Drift)	時間偏移與頻率偏移 (Offset)

AGPS – Assisted GPS

- 全名輔助全球衛星定位系統。
- 一種結合了網路訊息與GPS訊號的系統。
- 可以讓GPS在第一次定位的時間大幅縮短。

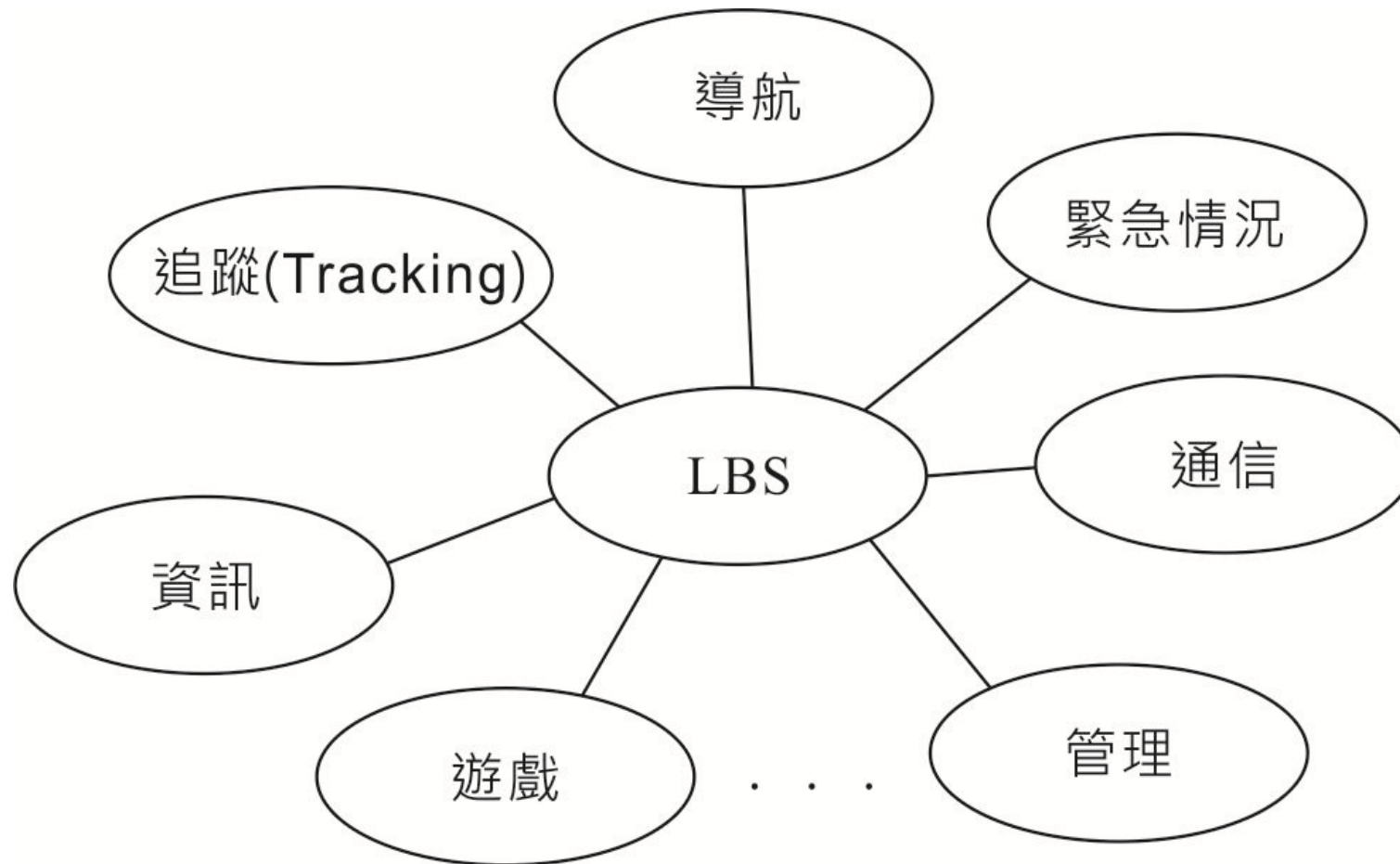
AGPS的工作原理

- ❑ AGPS裝置首先將本身訊號的基地台位置透過網路傳送到位置伺服器。
- ❑ 位置伺服器根據裝置的大概區域傳輸與該區域相關的GPS資訊到裝置上，其中包含方樣仰角等等。
- ❑ 該裝置根據AGPS資訊接收GPS原始訊號，AGPS提升GPS第一次定位的時間，也就是TTFF (Time to First Fix)。
- ❑ 該裝置在收到GPS原始訊號後解析訊號，計算裝置到衛星的偽距（定位過程中，地面接收器到衛星之間的大概距離，由於有各種影響產生誤差，故稱偽距），並將相關資訊透過網路傳輸給位置伺服器。
- ❑ 位置伺服器根據收到的偽距和其他定位設備來完成GPS資訊的處理，並估算該裝置的位置。
- ❑ 位置伺服器將該裝置的位置透過網路傳輸到應用程式上。

行動計算應用與服務

- LBS 可以透過定位資訊來尋找附近的一些附近有關生活機能的地方。
- 例如尋找店家、餐廳、ATM等等。
- 透過定位資訊來追蹤物件的去向或是提供旅遊指南等等。
- LBS也常被用在救援行動，透過GPS鎖定救援目標的位置，便可以去進行救援。

圖13.4: LBS的基礎應用



行動健康照護系統

GPS與電子羅盤



- 行動健康照護例子。
- 輪椅上放置一個行動載具，利用該行動載具的GPS與電子羅盤來監控輪椅的位置，可即時監控輪椅的位置。
- 輪椅上加裝許多感測器。

行動健康照護系統

- 舉一個例子：由於老年人口比例逐漸在升高，儼然已成為高齡化社會。
- 合作開發「行動健康照護服務」透過系統即時監控老年人的身體狀況以期能第一時間給予救護治療。
- 透過LBS，一發現老年人的身體訊息有異常或是警告，即可根據位置資訊立即到患者身邊進行治療。
- 行動健康照護服務愈來愈被重視，甚至連資策會都有開班授課關於行動健康照護的內容。

行動健康照護系統

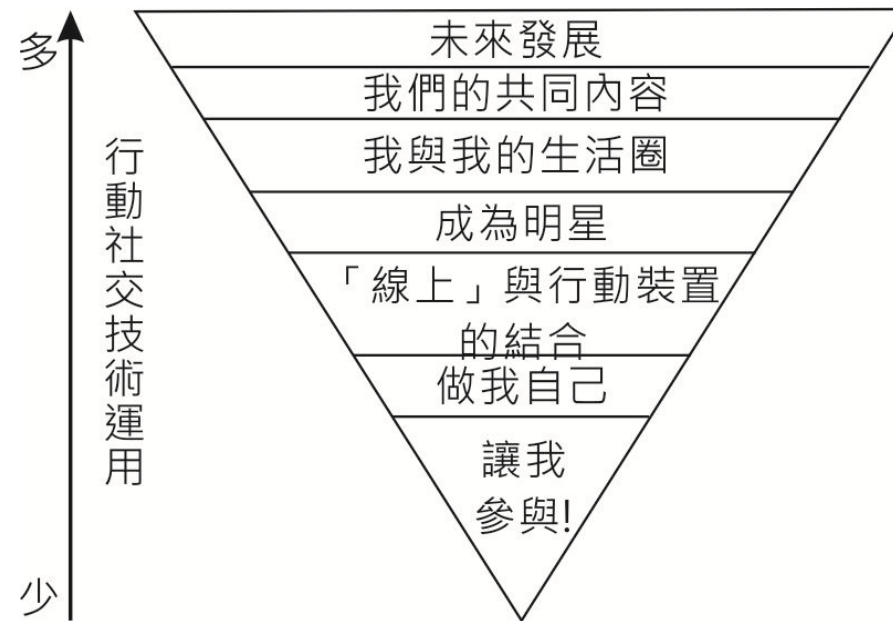


- 透過各種感測器的組合，可判斷出輪椅的動向以及使用者的坐姿狀況。

行動社交網路服務 (Mobile Social Networking Service)

- 行動社交網路簡單來說就是「可移動的社區以及內容」。
- 提供給使用者隨時隨地「進入」網路社群以及觀看內容。

圖13.5: 行動社交網路服務 (Mobile Social Networking Service)



- 行動社交網路服務大致可分為以下7類，如上圖所示。
- 上圖列出行動社交網路服務的類別，最底端用到的行動社交網路技術最少，愈往上用到的技術愈多，也愈複雜。

讓我參與！（Let Me In！）

- 意即讓使用者「進入一個社區」。
- 例如所謂的即時通訊軟體，即MSN、Yahoo Messenger等等。
- 「讓我參與！」是最簡單的服務型態。
- 最早建立存在最久的服務型態。
- 讓使用者與使用者之間進行交談，主要的交談方式是透過文字編輯傳送。
- 在「讓我參與！」中每位使用者都是匿名使用者，您只會知道對方的ID。

做我自己 (Let Me Be Me)

- 所謂的「部落格」服務。
- 讓使用者用來廣播自己來讓自己曝光。
- 就像「讓我參與！」一樣，滿足了「結交朋友」的需求。
- 需要更多的投資，簡單來說便是「經營部落格」這件事。
- 要如何「成為您自己」可以透過簡單的操作顯現出自己與別人不一樣的特點。
- 「搜尋」對於這服務來說是很重要的一項元素。

「線上」與行動載具的結合 (Merge My Online with My Mobile)

- 目的在於延伸線上社交網路可觸及的範圍而不局限於利用瀏覽器瀏覽。
- 例如透過APP瀏覽Facebook 。
- 這樣一來會有幾個挑戰：
 - 多媒體如何呈現？
 - 裝置如何管理？
 - 與瀏覽器瀏覽的使用者體驗相比會不會比較差？

成為明星 (Make Me a Star)

- 顧名思義就是「當個明星」，使用者想要展現他們的行動以及天分。
- 使用者可以根據自己自訂個性化的內容，例如影片圖片（稱為UGC-User Generated Content）。
- 「社區」必須要能吸引創造者、消費者以及批評者。
- 配合各式各樣不同的個性化內容 (UGC) 必須要投入更多的技術。
- 服務商也可以加入廣告賺取收入，或是有付費會員與免費會員的差別。

我與我的生活圈 (Me and My Circle)

- 「我與我的生活圈」是建立在「做我自己」之上。
- 主要是專注在好友圈之間的聯繫以及對話的管理。
- 提供一些必要的功能。
 - 搜尋。
 - 連結 (Linking) 。
 - 各自的關係 (Relationships) 。
 - 評分。
 - Etc 。

我與我的生活圈 (Me and My Circle)

- 「我與我的生活圈」可以將許多使用者關聯在一起，讓這些關聯在一起的使用者們可以看到一樣的內容。
- 服務供應商為了生存，也會加入一些廣告等等方式增加收入。
- 提供一些數位禮物 (Digital Gift) 的機制，讓使用者之間可以互相送禮聯絡感情，通常是用服務內部的貨幣來購買數位禮物。

我們的共同內容 (Me and My Circle and Our Content)

- 我們的共同內容」主要是將「我與我的生活圈」延伸，使得「我與我的生活圈」以「內容 (Content)」為中心元素。
- 通常是從「我與我的生活圈」與「成為明星」演變過來，使用者們可以分享與評論一些專業的內容，其內容通常是一些專業影片或是圖片 (Media)。
- 透過Youtube分享到Facebook上面，並指定群組分享給他們，甚至進行討論。

未來發展 (The Future ?)

- 強調環境感知或是位置？
- 是否要支援更多裝置？
- 與各大微網誌 (Twitter) 的結合？
- 即時分享心情經驗？

行動多媒體影音服務

- 在智慧手機以及平板電腦快速普及的帶動下，實際開通3G數據傳輸服務的用戶也愈來愈多。
- 行動電話業者常將行動多媒體服務與上網服務共同搭成專案銷售，提供行動娛樂的方便性吸引消費者使用。
- 免費Wifi（例如iTaiwan、7-Wifi、電信業者提供室內免費Wifi等等）快速的建置也讓行動影音的普及愈來愈快。

行動載具硬體服務整合互動應用

- 相機 (Camera) 。
- QR Code 。
- 感測器 (Sensors) 。
- 近場通訊 (NFC-Near Field Communication) 。

相機 (Camera)

- 相機愈來愈先進，畫質愈來愈好，因而可以做更多的應用。
- 人臉辨識。
 - 人臉辨識快速地讓使用者標記出是誰在照片裡。
 - 行動載具的數位鎖。

QR Code

- QR Code是一種二維條碼，可以儲存比傳統一維條碼更多的資訊。
- QR Code正逐漸地被廣泛運用。
- 廠商在廣告DM上放一個儲存網址的QR Code，讓拿到DM的人可以透過行動載具相機讀取條碼所儲存的網址，便可以快速的瀏覽網站。
- QR Code也可以儲存手機號碼等等個人資料，在交換號碼資本資料的時候方便快捷。

QR Code

- 下圖是一個QR Code範例，想知道內容馬上拿起行動載具讀取一下吧！



感測器 (Sensors)

- 感測器是接收信號或刺激並反映的物件，能將物理量等能量轉換成另一對應輸出的裝置。
- GPS衛星定位模組，可以幫助行動載具獲得位置資訊。
- 數位電子羅盤可以協助行動載具判斷東西南北。

近場通訊 (NFC-Near Field Communication)

- 又稱「近距離無線通訊」。
- 是一種短距離的高頻無線通訊技術，讓電子設備之間不用進行接觸即可以點對點傳輸資料。
- 距離限制大約10公分。
- NFC可當作一張IC卡，可以代替現有的大多數種類IC卡。
- NFC當作IC卡時並不耗電，就算行動載具沒電也可以進行工作。
- 其供電來自非接觸型讀卡機。

近場通訊 (NFC-Near Field Communication)

- NFC也可以進行短距離的資料傳輸。
- 與紅外線差不多。
- 傳輸距離較短，但是連結建立速度與傳輸速度快些。
- 目前內置NFC的裝置以手機居多，而且有愈來愈多的趨勢。

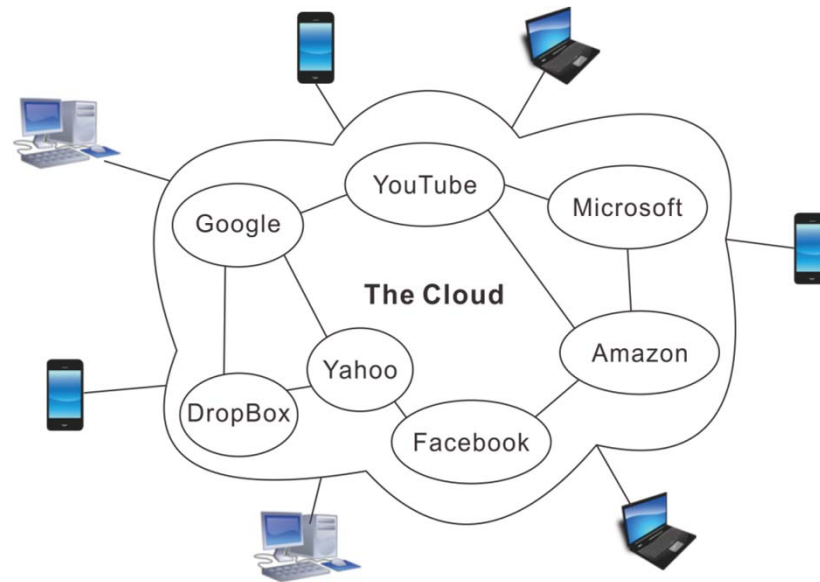
行動雲端服務 (Mobile Cloud Service)

- 所謂的「雲端服務 (Cloud Service)」即是一些網路服務商提供一些高品質的方案讓使用者可以在「雲」上面做事情。
- 瀏覽網頁即是最早出現的一種存取雲端的方式。
- 現在透過行動載具，使用者可以完全的隨時隨地與「雲」互動。
- 使用者可以隨時存取Yahoo Mail或Google Search 。

行動雲端服務 (Mobile Cloud Service)

- 與傳統的桌面應用程式或是直接透過瀏覽器瀏覽不同的是，現在使用者可以在行動載具桌面建立一個簡單的應用程式，而應用程式裡的資料全部都在「雲」上面，就這樣透過行動載具直接存取「雲」。
- 行動載具應與「雲」透過3G或Wifi等方式持續的保持連線，以方便資料的同步化以及即時性。
- 收發E-mail、檢查天氣股票等等。

行動雲端服務 (Mobile Cloud Service)

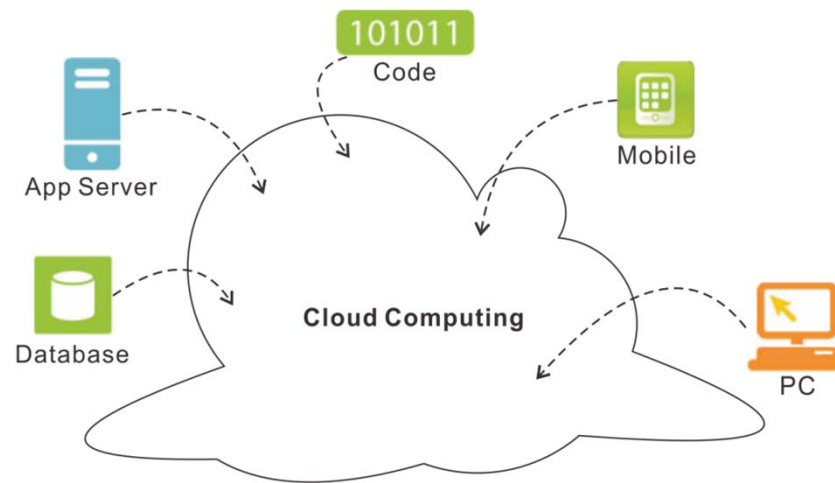


- 雲端服務示意圖
- 在雲之中有許多服務商所提供的服務，使用者（行動載具、筆記型電腦、傳統電腦）可以透過網路來輕鬆存取檢視在雲上面的資料。

行動雲端服務 (Mobile Cloud Service)

- 行動載具如何與雲端服務產生關聯。
- 如果有使用過行動載具，相信您不會想在行動載具上設計一個簡報 (PowerPoint) 或寫一份有一定分量的文件。
- 行動載具的優勢在於，您可以輕易地檢視這些文件。
- 這表示行動載具很難取代傳統電腦。
- 「雲」可以幫助我們當有需要的時候便可以即時檢視檔案。

行動雲端服務 (Mobile Cloud Service)



- 雲端計算示意圖。
- 服務商撰寫程式給使用者。
- 使用者可以透過專用程式輕鬆存取操作雲的服務。
- 「雲」要長久，使用者必須合理使用，服務商也必須要好好檢查各種細節才行。

行動計算的挑戰與研究議題

- 任何裝置可以隨時隨地存取運算環境，以上便是使用者們對於行動運算的期待。
- 實現這種期待的困難點在於：如何去開發應用程式可以持續的適應環境以及就算使用者移動或是更改裝置也可以保留工作狀態。
- 提出所有在實現這目標上會遇到的挑戰。

表13.2: 行動計算的挑戰與研究議題

議題	別名	動機 (Motive)
異質性		允許多種不同服務提供不一樣的裝置、網路與環境
可靠性 (Dependability) 與安全性	錯誤容忍度	避免錯誤發生頻率過高導致結果錯誤提供可利用性、機密性、可信賴性 (Reliability)、安全性、整理性與可維護性
隱私與信任		保護並防止個人資料的錯誤使用 定義互動物件的可靠度
移動性	錯誤容忍度	提供可以隨時隨地存取資料的應用程式能讓使用者隨身攜帶屬於自己的使用環境
環境感知	感知	感測使用者與周遭的狀態 推論環境資訊
環境管理		根據感測到的環境資訊調整系統行為使適應現在所處情況
人機互動		把使用者界面與現實世界結合 讓使用者專注在工作上，而不是在程序幕後設定

行動計算的挑戰與研究議題

- 異質性：將公共設施 (Infrastructure) 的不同之處藏起來，讓使用者與其他人互動時感覺不出差異；實際上兩者之間有喜多硬體設施設備等等差異，但程式已經在背後處理完畢。
- 可靠性 (Dependability)：將「可利用性、可信賴性 (Reliability)、安全性、整體性與可維護性」整合在一起的一個概念。
- 安全性：與可靠性緊緊相連的一個概念。如果一個系統存在著「可利用性、整體性與機密性」則可稱為具有安全性。

行動計算的挑戰與研究議題

- 移動性：讓使用者可以隨時隨地的連接到應用程式或者是存取資料。可以從兩方面來看，實體上的（與使用者或是裝置設備有關）與邏輯上的（與程式碼或者是資料有關），通常不是與實體有關就是與邏輯有關。
- 環境感知：推斷環境因素去提供資料或服務。
- 環境管理：是一種將偵測到的資料做處理的動作。利用感測到的資料去讓提供的服務適應環境。也有一些作者認為，環境管理是環境感知的其中一部份。

行動計算的挑戰與研究議題

- 人機互動：人機互動包含許多層面，這邊只簡單描述。當電腦 (Computer) 變的「更聰明」的時候，人機互動的使用以及品質也會跟著成長。隨著程式設計的進步，使用者介面也會變得更加完善。
- 要找出一個應用程式包含上面所提的各種特色依然是相當困難的。
- 如果能有一個通用性的程式設計模型相信對於應用程式的開發會有一定的幫助。

結論

- 本章一開始介紹了無線的種類、載具的定義以及計算的歷史。積體電路的發達讓計算機愈來愈小，到現在即使是手持式裝置也有一定的計算能力，加上無線的發展，讓行動計算可以用的範圍更廣，潛力更大。行動計算可以讓使用者將資訊帶著走，隨時隨地的獲得各種資訊各種資源。

結論

- 行動計算有幾個環境限制
- 無線網路環境的差異，包含換手、斷線、頻寬限制等等
- 硬體上的限制，包含電池電量、硬體儲存空間等等不足造成的硬體不可使用，以及載具遺失或儲存空間損壞的不可依賴性
- 行動計算架構上的限制，目前尚未有完善的行動感知架構可以充分支援行動計算。

結論

- 現在普及的平板以及智慧型手機上面的作業系統並非只有一家，而是百家爭鳴的情況，而各家的介面、提供的開發工具、開發環境等等皆有差異，消費者可以根據自己的需求去挑選。行動計算架構分為單機、Client / Server以及雲端幾種，本章提供了幾個在這些架構下實作的應用程式範例，供讀者參考。

結論

- 當然行動計算應用程式絕非只有範例那幾種，還有行動社交、行動定位服務等等；行動社交包含的許多層面，即時通訊軟體、部落格、社群網站等等，行動社交的發展，讓人與人之間的互動形式變得更廣。當然還有更多的應用方式並未在本章被提出，讀者們可以想想還有哪些方面的應用。

結論

- 行動計算應用程式開發並不是簡單的事情，開發過程中存在著一些挑戰以及議題，而且隨著時代演進，這些議題會被克服，也會出現新的挑戰。隨著更多的創意以及創新投入這領域，行動計算系統正一個一個被開發出來，行動計算潛力無限且商機無窮。

習題

1. 請簡述行動(Mobility)代表的意思?
2. 請簡述計算(Computing)代表的意思?
3. 請列出行動計算的種類並詳細說明?
4. 請描述Location-Aware Computing的概念?
5. 請描述Context-Aware Computing的概念?
6. 請舉例三種行動載具的作業系統?
7. 請說明行動環境有哪些限制?
8. 請畫出行動計算的兩種架構圖。
9. 請簡述行動計算的概念?
10. 請簡述無線網路技術與行動計算的關係。

習題

11. 構思一個在智慧型手機上運行的應用程式，並簡易敘述運作方式以及特色。
12. 請簡述AGPS的工作方式。
13. 除了行動健康照護服務，LBS尚可做那些應用？
14. 試討論行動健康照護服務需要怎樣的功能，並討論如何實踐。
15. 美琴自行拍攝了舞蹈影片，上傳到Youtube想要分享給網友欣賞他精湛的舞步，請問以上屬於社交網路服務的哪一環？
16. 嘗試製作屬於自己的QR Code，內容不拘。

習題

17. 試舉出目前裝備有NFC的裝置，至少3種型號。

(13-5)

17. 試舉例目前常見的行動雲端應用程式，並簡述如何運作。(13-5)

18. NFC具有以下那些特點，請打 V ?(13-5) () 短距離傳送 () 必須接觸才可傳送資料 () 可替代大部分IC卡 () 相當耗電 () 傳輸速率比紅外線慢 () 若行動載具沒電依然可當IC卡使用

19. 要設計出跨平台並且具有語音以及訊息功能的應用程式，需要考慮到怎樣的問題以及如何克服，試討論。(13-6)

參考文獻

1. http://en.wikipedia.org/wiki/Atanasoff–Berry_Computer
2. <http://en.wikipedia.org/wiki/ENIAC>
3. http://en.wikipedia.org/wiki/Steve_Mann
4. <http://www.htc.com/>
5. <http://www.apple.com/tw/ipad/>
6. <http://www.android.com/>
7. <http://www.blackberry.com.tw/>
8. <http://www.apple.com/tw/iphone/ios/>

參考文獻

9. <http://www.microsoft.com/windowsphone/zh-tw/default.aspx>
10. <http://www.symbianos.org/intro>
11. M. Satyanarayanan, “Pervasive Computing: Vision and Challenges,” IEEE Personal Communication, Aug. 2001, pp. 10-17.
13. Christine Perey , 2008 , Mobile Social Networking: Communities and Content on the Move , Informa U K Limited
14. <http://www.cise.ufl.edu/~helal/classes/f10/index.html>
Mobile Computing

參考文獻

15. http://ngis2.moi.gov.tw/Storage/MOI_NGIS/journal/26/new3.html
16. 淺談GPS/GLONASS雙星定位系統
17. <http://www.cht.com.tw/> 中華電信
18. <http://www.7wifi.com.tw/index.aspx> 7-wifi
19. <http://itaiwan.gov.tw/> iTaiwan無線上網
20. <https://play.google.com/store> Google Play 商城
21. <http://qrcode.kaywa.com/> QR Code Generator
22. <http://zh.wikipedia.org/wiki/> 近場通訊NFC

參考文獻

- 23. <https://www.dropbox.com> DropBox網站
- 24. Cristiano André da Costa, Adenauer Corrêa Yamin, Cláudio Fernando Resin Geyer, 2008,” Toward a General Software Infrastructure for Ubiquitous Computing”