

Android App

程式設計教本 之無痛起步

引領入門

最簡單、最易懂的初學教材



第 12 章

用感測器製作水平儀與體感控制

本投影片（下稱教用資源）僅授權給採用教用資源相關之旗標書籍為教科書之授課老師（下稱老師）專用，老師為教學使用之目的，得摘錄、編輯、重製教用資源（但使用量不得超過各該教用資源內容之80%）以製作為輔助教學之教學投影片，並於授課時搭配旗標書籍公開播放，但不得為網際網路公開傳輸之遠距教學、網路教學等之使用；除此之外，老師不得再授權予任何第三人使用，並不得將依此授權所製作之教學投影片之相關著作物移作他用。

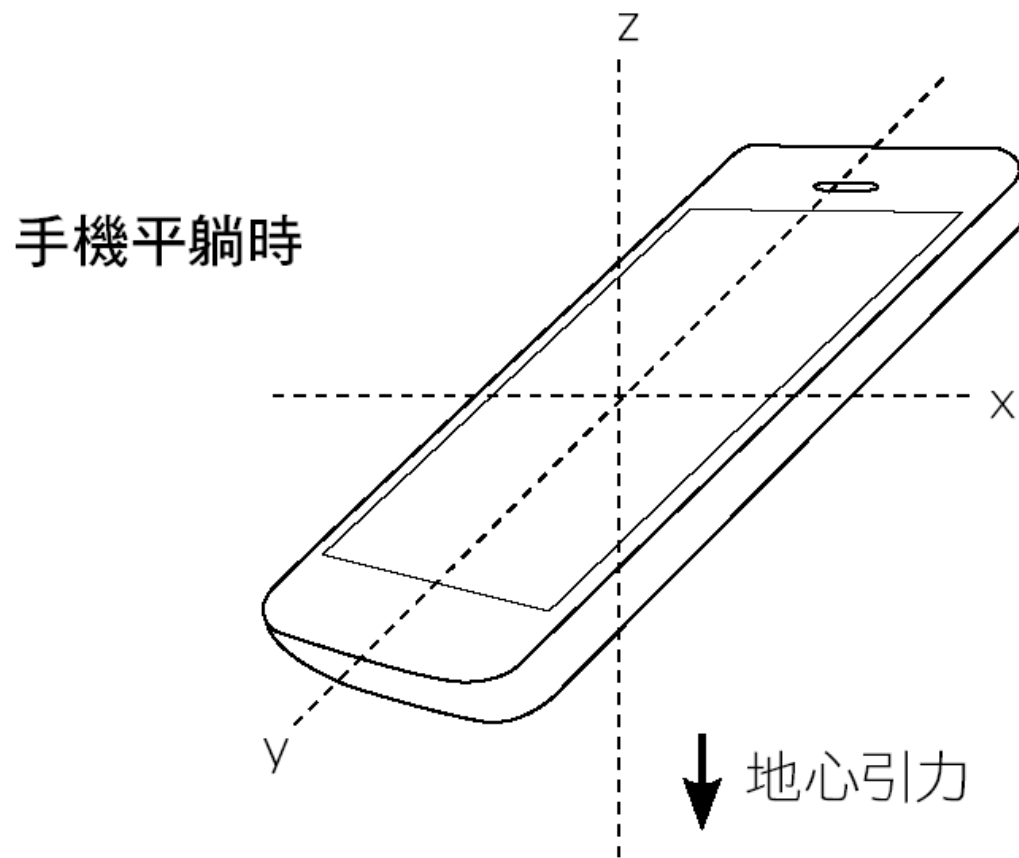
前言

- 12-1 讀取加速感測器的值
- 12-2 利用 x、y 軸的加速度值來製作水平儀
- 12-3 利用加速感測器來做體感控制

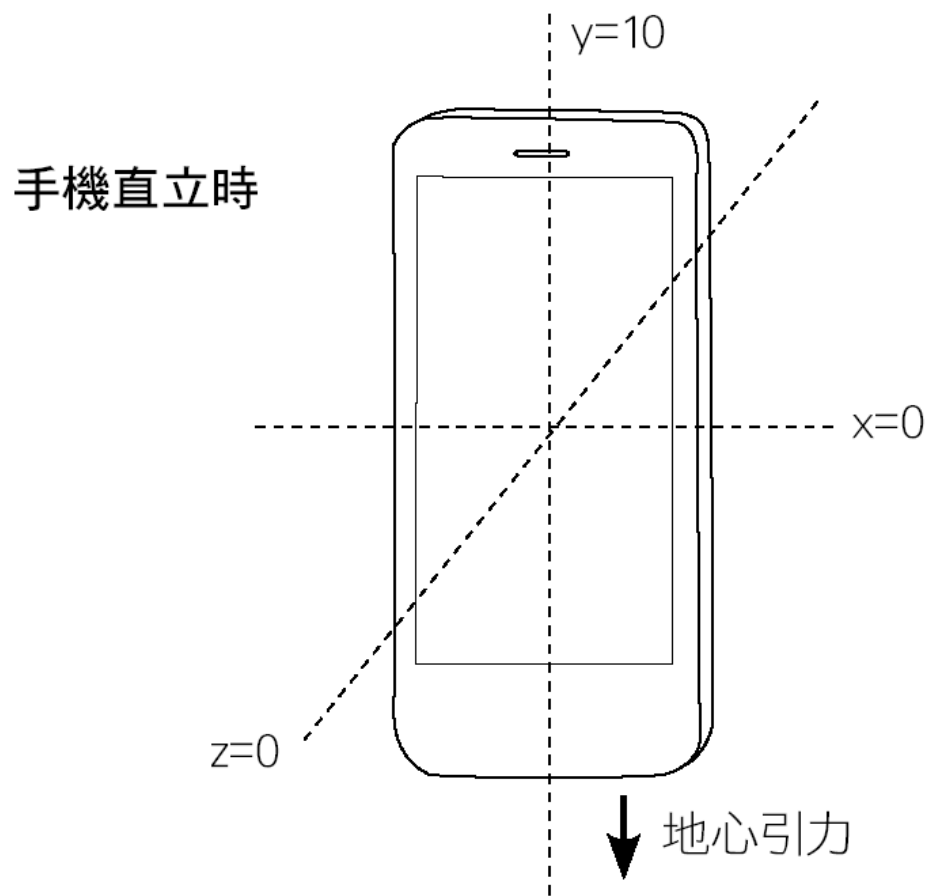
12-1 讀取加速感測器的值

- 認識加速感測器
- 取得系統的感測器物件
- 讀取感測器的值

認識加速感測器



認識加速感測器



取得系統的感測器物件

```
SensorManager sm;
```

```
sm = (SensorManager) getSystemService(SENSOR_SERVICE);
```

←
由系統服務取得感測器管理員

```
Sensor sr;
```

```
sr = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);
```

←
取得加速感測器

取得系統的感測器物件

常數	感測器種類	常數	感測器種類
TYPE_ACCELEROMETER	加速感測器	TYPE_AMBIENT_TEMPERATURE	溫度感測器
TYPE_MAGNETIC_FIELD	地磁感測器	TYPE_GYROSCOPE	陀螺儀 (可偵測旋轉)
TYPE_LIGHT	亮度感測器	TYPE_LINEAR_ACCELERATION	直線加速感測器
TYPE_PROXIMITY	距離感測器	TYPE_RELATIVE_HUMIDITY	相對濕度感測器
TYPE_PRESSURE	大氣壓力感測器	TYPE_ROTATION_VECTOR	旋轉向量感測器
TYPE_GRAVITY	重力感測器		

讀取感測器的值

```
01 public class MainActivity extends Activity
02     implements SensorEventListener { ← 實作感測器監聽介面
03     ...
04     @Override
05     public void onSensorChanged(SensorEvent event) {
06         //可由參數 event 來讀取感測器的值, 然後再做後續處理
07     }
08     @Override
09     public final void onAccuracyChanged(Sensor sensor, int accuracy) {
10         //可由參數 accuracy 來讀取改變後的精確度
11     }
12     ...
13 }
```

每當感測器的值改變時即會呼叫此方法

每當感測器的精確度改變時即會呼叫此方法

讀取感測器的值

```
01 float x = event.values[0]; ← x 軸加速度值  
02 float y = event.values[1]; ← y 軸加速度值  
03 float z = event.values[2]; ← z 軸加速度值
```

```
sm.registerListener(this, sr, SensorManager.SENSOR_DELAY_NORMAL); ←  
將 this 註冊為 sr (加速感測器物件) 的監聽器
```

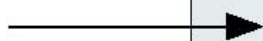
讀取感測器的值

事件頻率	適用時機
SENSOR_DELAY_NORMAL	一般狀況 (約 0.2 秒的延遲)
SENSOR_DELAY_UI	適合操作介面用 (約 0.06 秒的延遲)
SENSOR_DELAY_GAME	適合遊戲用 (約 0.02 秒的延遲)
SENSOR_DELAY_FASTEST	最快的速度 (不做延遲)

```
sm.unregisterListener(this); ← 將監聽物件解除註冊
```

範例12-1：顯示加速感測器的加速度值

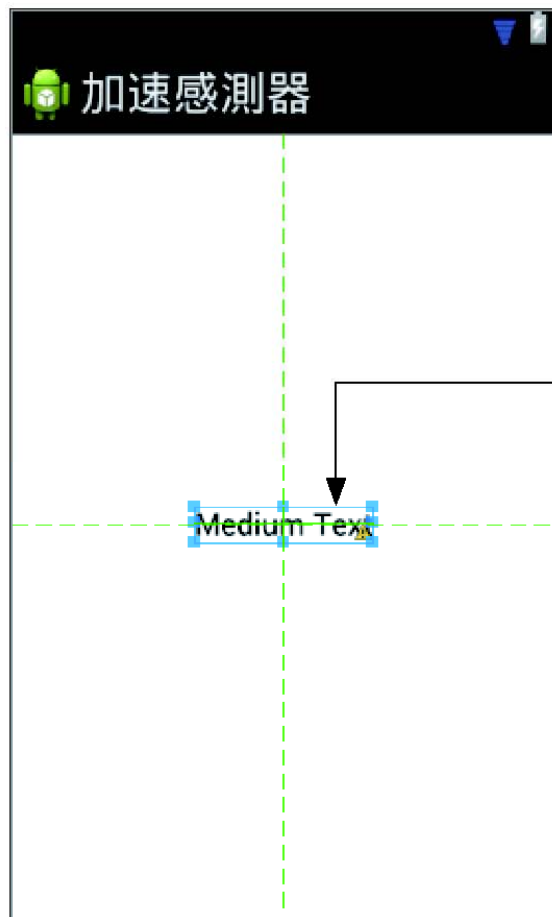
會不斷顯示最新的
三軸加速度值



顯示加速感測器的加速度值

step 1

step 2



Medium TextView 元件

Id	@+id/textView1
Center Horizontal	true
Center Vertical	true

顯示加速感測器的加速度值

step 3

```
01 public class MainActivity extends Activity
02             implements SensorEventListener {
03
04     SensorManager sm;    ← 感測器管理員
05     Sensor sr;          ← 加速感測器物件
06     TextView txv;       ← 畫面中的文字元件
07
08     @Override
09     protected void onCreate(Bundle savedInstanceState) {
10         super.onCreate(savedInstanceState);
11         setContentView(R.layout.activity_main);
12
```

顯示加速感測器的加速度值

```
13     sm = (SensorManager) getSystemService(SENSOR_SERVICE); ← 由系統服務取得感測器管理員
14     sr = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER); ← 取得加速感測器
15     txv = (TextView) findViewById(R.id.textView1); ← 取得 TextView 元件
16 }
```

顯示加速感測器的加速度值

step 4

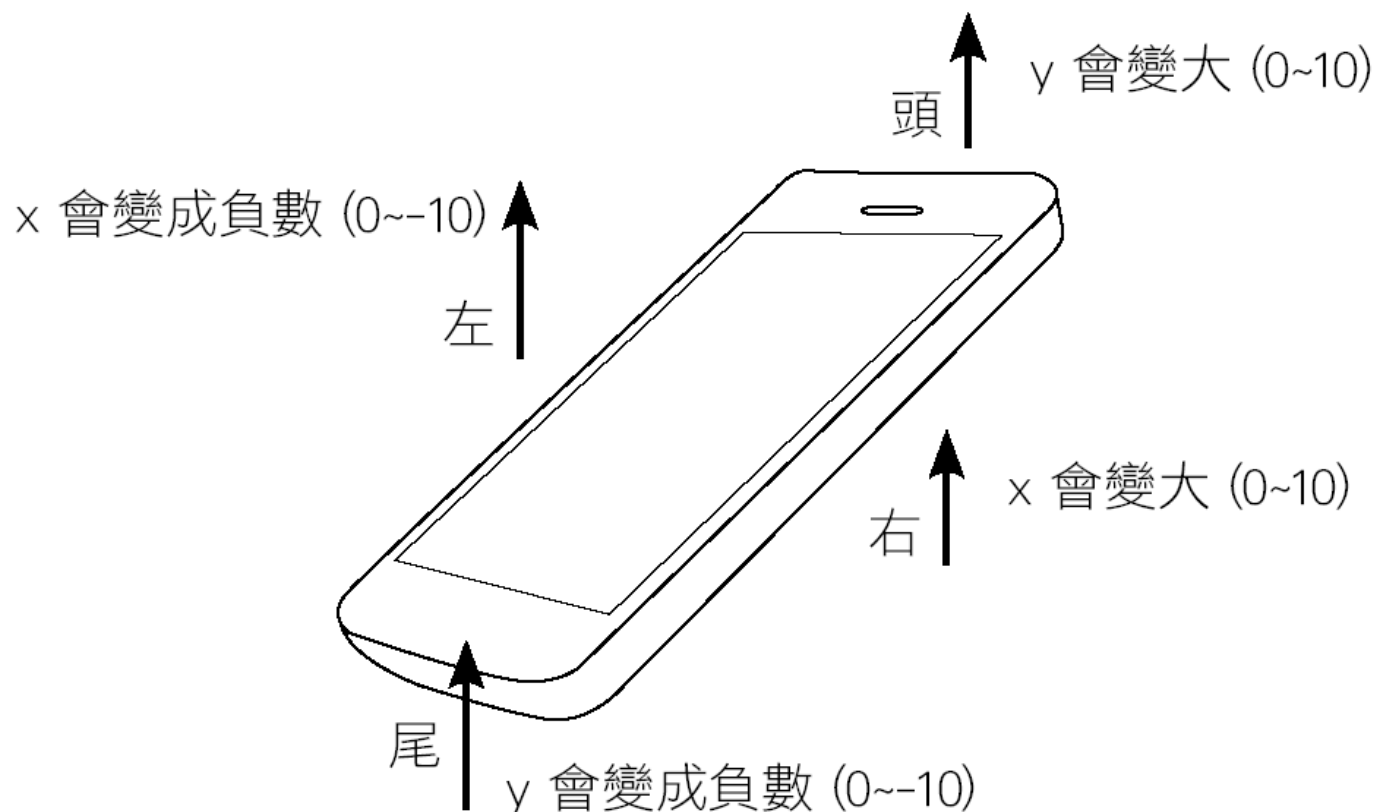
```
01 @Override
02 public void onSensorChanged(SensorEvent event) { ← 加速度值改變時
03     txv.setText(String.format("X軸: %1.2f, Y軸: %1.2f, Z軸: %1.2f",
04         event.values[0], event.values[1], event.values[2]));
05 }
06
07 @Override
08 public void onAccuracyChanged(Sensor arg0, int arg1) { } ←
                                精確度改變時不需處理
```

顯示加速感測器的加速度值

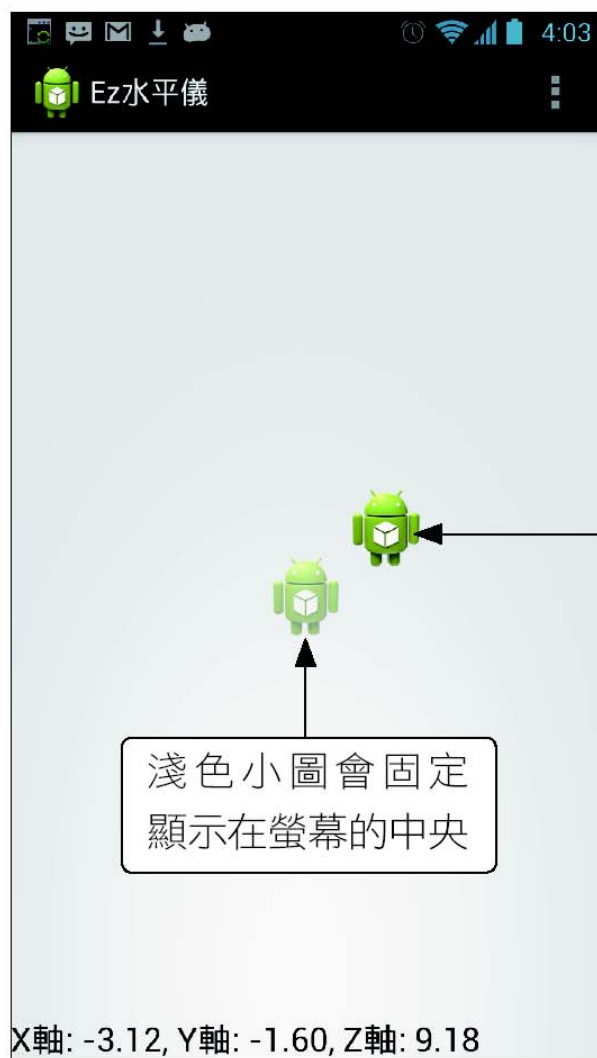
step 5

```
01 @Override
02 protected void onResume() { ← 當 Activity 畫面顯示出來時
03     super.onResume();
04     sm.registerListener(this, sr, SensorManager.
                                SENSOR_DELAY_NORMAL); ← 向加速感測器 (sr)
05 }                                     註冊監聽物件 (this)
06 @Override
07 protected void onPause() { ← 當 Activity 畫面被覆蓋時 (切換到其它程式)
08     super.onPause();
09     sm.unregisterListener(this); ← 取消監聽物件 (this) 的註冊
10 }
```


12-2 利用 x 、 y 軸的加速度值來製作水平儀



利用 x 、 y 軸的加速度值來製作水平儀



手機傾斜時，
小圖會朝傾
斜方向移動



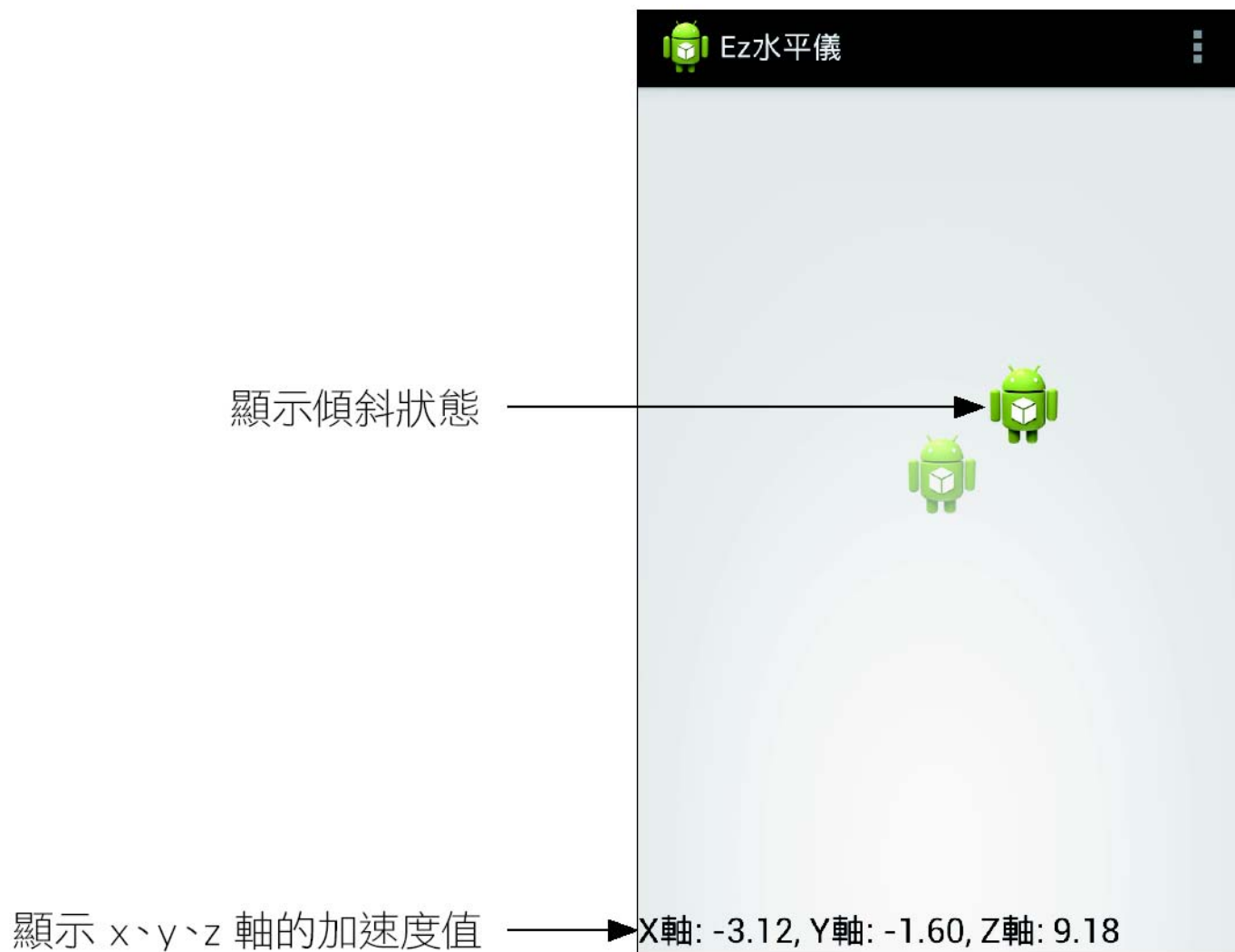
利用左邊界與上邊界來移動圖片



利用左邊界與上邊界來移動圖片

```
01 //底下假設 layout 為填滿畫面的 RelativeLayout 物件,
02 //          igv 為顯示小圖的 ImageView 物件
03 double mx = (layout.getWidth()-igv.getWidth()) /20.0; ←┐
                                                    計算 x 方向每一等份的大小
04 double my = (layout.getHeight()-igv.getHeight()) /20.0; ←┐
                                                    計算 y 方向每一等份的大小
05
06 RelativeLayout.LayoutParams parms = ←┐ 取得小圖的 LayoutParams 物件
07     (RelativeLayout.LayoutParams) igv.getLayoutParams();
08 parms.leftMargin = (int)((10-event.values[0]) * mx); ←┐
                                                    設定左邊界(加速度越大邊界越小)
09 parms.topMargin = (int)((10+event.values[1]) * my); ←┐
                                                    設定上邊界(加速度越大邊界越大)
10 igv.setLayoutParams(parms); ←┐ 將小圖套用 LayoutParams, 使邊界設定生效
```

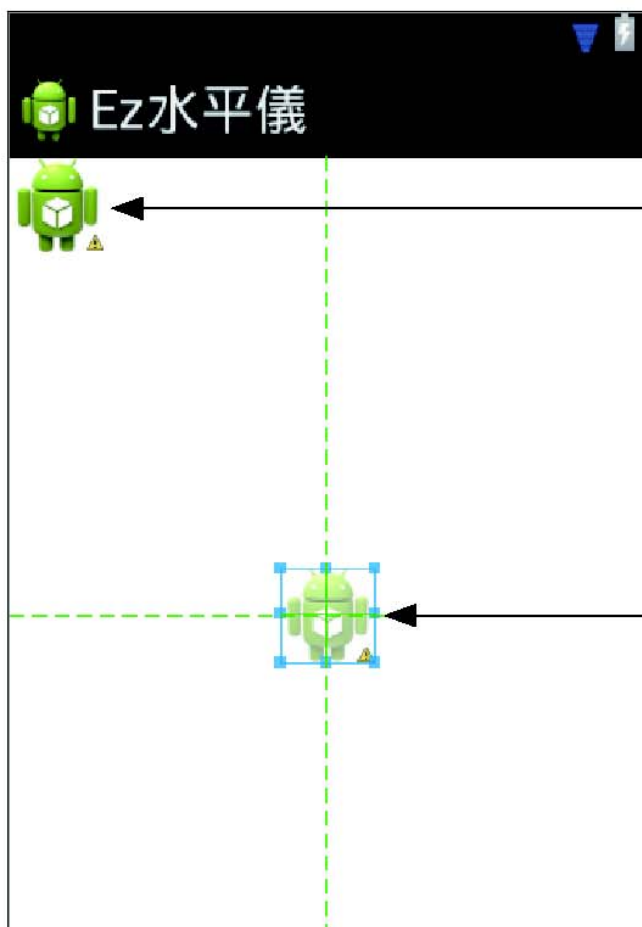
範例12-2：利用加速感測器製作水平儀



利用加速感測器製作水平儀

step 1

step 2



2 加入 ImageView 元件(設為靠左上對齊)

Id	@+id/igvMove
Src	@drawable/ic_launcher
Align Parent Left	true (靠左對齊)
Align Parent Top	true (靠上對齊)

1 加入 ImageView 元件(設為半透明並置中對齊)

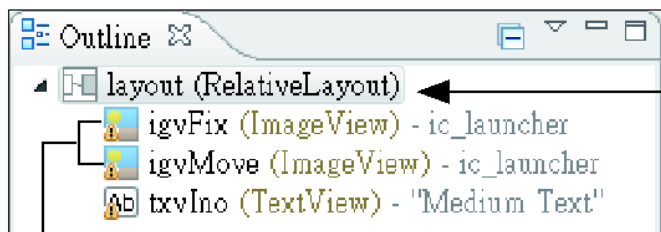
Id	@+id/igvFix
Src	@drawable/ic_launcher
Center Horizontal	true (水平置中)
Center Vertical	true (垂直置中)
Alpha	0.5 (設為半透明)

利用加速感測器製作水平儀



3 加入 Medium TextView 元件(設為靠左下對齊)

Id	@+id/txvIno
Align Parent Left	true (靠左對齊)
Align Parent Bottom	true (靠下對齊)



4 替預設的 RelativeLayout 命名為 layout

Id	@+id/layout
----	-------------

注意!igvMove 要排在 igvFix 的下面
(重疊時才會在上層)

利用加速感測器製作水平儀

step 3

```
01 public class MainActivity extends Activity
02 implements SensorEventListener { ← 實作感測器監聽介面
03     SensorManager sm;           ← 感測器管理員
04     Sensor sr;                  ← 加速感測器物件
05     TextView txv;              ← 畫面中的文字元件
06     ImageView igv;             ← 畫面中要移動的小圖
07     RelativeLayout layout;      ← 畫面的 Layout 元件
08     double mx = 0, my = 0;     ← 儲存 x,y 方向每一等份的大小 (預設值:0)
09
10     @Override
11     protected void onCreate(Bundle savedInstanceState) {
12         super.onCreate(savedInstanceState);
```


利用加速感測器製作水平儀

```
13     setContentView(R.layout.activity_main);
14
15     setRequestedOrientation(ActivityInfo.          設定螢幕不隨手機旋轉
        SCREEN_ORIENTATION_NOSENSOR); ←┐
16
17     sm = (SensorManager) getSystemService(SENSOR_SERVICE); ←┐
        由系統服務取得感測器管理員
18     sr = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER); ←┐
        取得加速感測器
19     txv = (TextView) findViewById(R.id.txvIno); ←┐
        取得 TextView 元件
20     igv = (ImageView) findViewById(R.id.igvMove); ←┐
        取得要移動的 ImageView 元件
21     layout = (RelativeLayout) findViewById(R.id.layout); ←┐
        取得 layout 元件
```

利用加速感測器製作水平儀

```
22  }
23
24  @Override
25  public void onSensorChanged(SensorEvent event) {
26      if(mx == 0) { ← 如果還沒計算過
27          mx = (layout.getWidth()-igv.getWidth()) /20.0; ←
28                                                         計算 x 方向每一等份的大小
29          my = (layout.getHeight()-igv.getHeight()) /20.0; ←
29                                                         計算 y 方向每一等份的大小
```

Next

利用加速感測器製作水平儀

```
29     }
30         取得小圖的 RelativeLayout 物件
31     RelativeLayout.LayoutParams parms = ←┐
32         (RelativeLayout.LayoutParams) igv.getLayoutParams();
33     parms.leftMargin = (int)((10-event.values[0]) * mx); ←┐
34                                     設定左邊界
35     parms.topMargin = (int)((10+event.values[1]) * my); ←┐
36                                     設定上邊界
37     igv.setLayoutParams(parms); ← 將小圖套用 RelativeLayout,
38                                     使邊界設定生效
39
40     txv.setText(String.format
41         ("X軸: %1.2f, Y軸: %1.2f, Z軸: %1.2f", ← 顯示感測器的資料
42         event.values[0], event.values[1], event.values[2]));
```

利用加速感測器製作水平儀

```
39  }
40
41  @Override
42  public void onAccuracyChanged(Sensor arg0, int arg1) { } ← 不需處理
43
44  @Override
45  protected void onResume() {
46      super.onResume();
47      sm.registerListener(this, sr, SensorManager.
48          SENSOR_DELAY_NORMAL); ← 向加速感測器 (sr) 註冊監聽物件 (this)
49  }
50  @Override
51  protected void onPause() {
52      super.onPause();
53  }
```

利用加速感測器製作水平儀

```
52     sm.unregisterListener(this); ← 取消監聽物件(this) 的註冊
53 }
54
55 @Override
56 public boolean onCreateOptionsMenu(Menu menu) {
57     // Inflate the menu;
58     getMenuInflater().inflate(R.menu.activity_main, menu);
59     return true;
60 }
61 }
```

12-3 利用加速感測器來做體感控

- 偵測手機面朝下平放的狀態
- 偵測手機搖動

偵測手機面朝下平放的狀態

```
01 if(Math.abs(x) < 1 && Math.abs(y) < 1 && z < -9) { ← 如果手機面朝下平放
02     if(mper.isPlaying()) { ← 如果正在播放, 就要暫停
03         btnPlay.setText("繼續");
04         mper.pause(); ← 暫停播放
05     }
06 }
```

偵測手機搖動

```
01 if(Math.abs(x) + Math.abs(y) + Math.abs(z) > 32) { ← 加速度總合超過 32
02     if(btnPlay.isEnabled()) ← 如果音樂已準備好(可以播放)
03         onMpPlay(null); ← 模擬按下播放鈕(切換『播放』與『暫停』狀態)
04 }
```


偵測手機搖動

```
01 if(delay > 0) { ← delay大於 0 時, 表示要略過這次的偵測
02     delay--; ← 將次數減 1, 直到 0 為止
03 }
04 else {
05     if(Math.abs(x) + Math.abs(y) + Math.abs(z) > 32) { ← 如果加速度總合超過 32
06         if(btnPlay.isEnabled()) ← 如果音樂已準備好(可以播放)
07             onMpPlay(null); ← 模擬按下播放鈕(切換『播放』與『暫停』狀態)
08         delay = 5; ← 延遲 5 次不偵測(約 1 秒)
09     }
10 }
```

範例12-3：利用加速感測器來控制音樂播放

step 1

step 2

```
01 public class MainActivity extends Activity implements
02     MediaPlayer.OnPreparedListener,
03     MediaPlayer.OnErrorListener,
04     MediaPlayer.OnCompletionListener,
05     SensorEventListener { ← 實作感測器監聽介面
06     SensorManager sm; ← 感測器管理員
07     Sensor sr; ← 加速感測器物件
08     int delay = 0; ← 用來延遲體感控制的偵測間隔
09     ...
```

實作 MediaPlayer 的 3 個的事件監聽介面

利用加速感測器來控制音樂播放

step 3

```
01 protected void onCreate(Bundle savedInstanceState) {  
02     super.onCreate(savedInstanceState);  
03     setContentView(R.layout.activity_main);  
04     ...  
05  
06     sm = (SensorManager) getSystemService(SENSOR_SERVICE);  
07     sr = sm.getDefaultSensor(Sensor.TYPE_ACCELEROMETER);  
08 }
```

由系統服務取得感測器管理員

取得加速感測器

利用加速感測器來控制音樂播放

step 4

```
01 @Override
02 protected void onResume() {
03     super.onResume();
04     sm.registerListener(this, sr,
05                        SensorManager.SENSOR_DELAY_NORMAL);
06
07 @Override
08 protected void onPause() {
09     super.onPause();
10
```

向加速感測器 (sr) 註冊監聽物件 (this)

約 0.2 秒傳一次

利用加速感測器來控制音樂播放

```
11     if (mper.isPlaying()) {           ← 如果正在播, 就暫停
12         btnPlay.setText("繼續");
13         mper.pause();                 ← 暫停播放
14     }
15     sm.unregisterListener(this);      ← 取消監聽物件(this) 的註冊
16 }
```

利用加速感測器來控制音樂播放

step 5

```
01 // 實作 SensorEventListener 介面的監聽方法
02 @Override
03 public void onAccuracyChanged(Sensor arg0, int arg1) { } ← 不需處理
04
05 @Override
06 public void onSensorChanged(SensorEvent event) {
07     float x, y, z;
08     x = event.values[0];
09     y = event.values[1]; ← x, y, z 軸的加速度值
10     z = event.values[2];
11 }
```

Next

利用加速感測器來控制音樂播放

```
12     if(Math.abs(x) < 1 && Math.abs(y) < 1 && z < -9) { ←┐
                                                    如果手機面朝下平放
13         if(mper.isPlaying()) { ←┐ 如果正在播放, 就暫停
14             btnPlay.setText("繼續");
15             mper.pause();
16         }
17     }
18     else {
19         if(delay > 0) { ←┐ delay 大於 0 時, 表示要略過這次的偵測
20             delay--; ←┐ 將次數減 1, 直到 0 為止
21         }
22     }
23         if(Math.abs(x) + Math.abs(y) + Math.abs(z) > 32) { ←┐
                                                    加速度總合超過 32
```

利用加速感測器來控制音樂播放

```
24         if (btnPlay.isEnabled()) ← 如果音樂已準備好 (可以播放)
25             onMpPlay (null); ← 模擬按下播放鈕 (切換
                                『播放』與『暫停』狀態)
26         delay = 5; ← 延遲 5 次不偵測 (約 1 秒)
27     }
28 }
29 }
30 }
```