資源管理分配與生命週期

Android 使用 Activity 生命週期 (Lifecycle) 的機制來管理資源分配，當記憶體資源不足時，系統會依照優先等級進行回收。但是行動裝置不似一般的電腦，若能考量程式運作時資源使用的狀況，對於系統的穩定與執行效能是相當有幫助的。

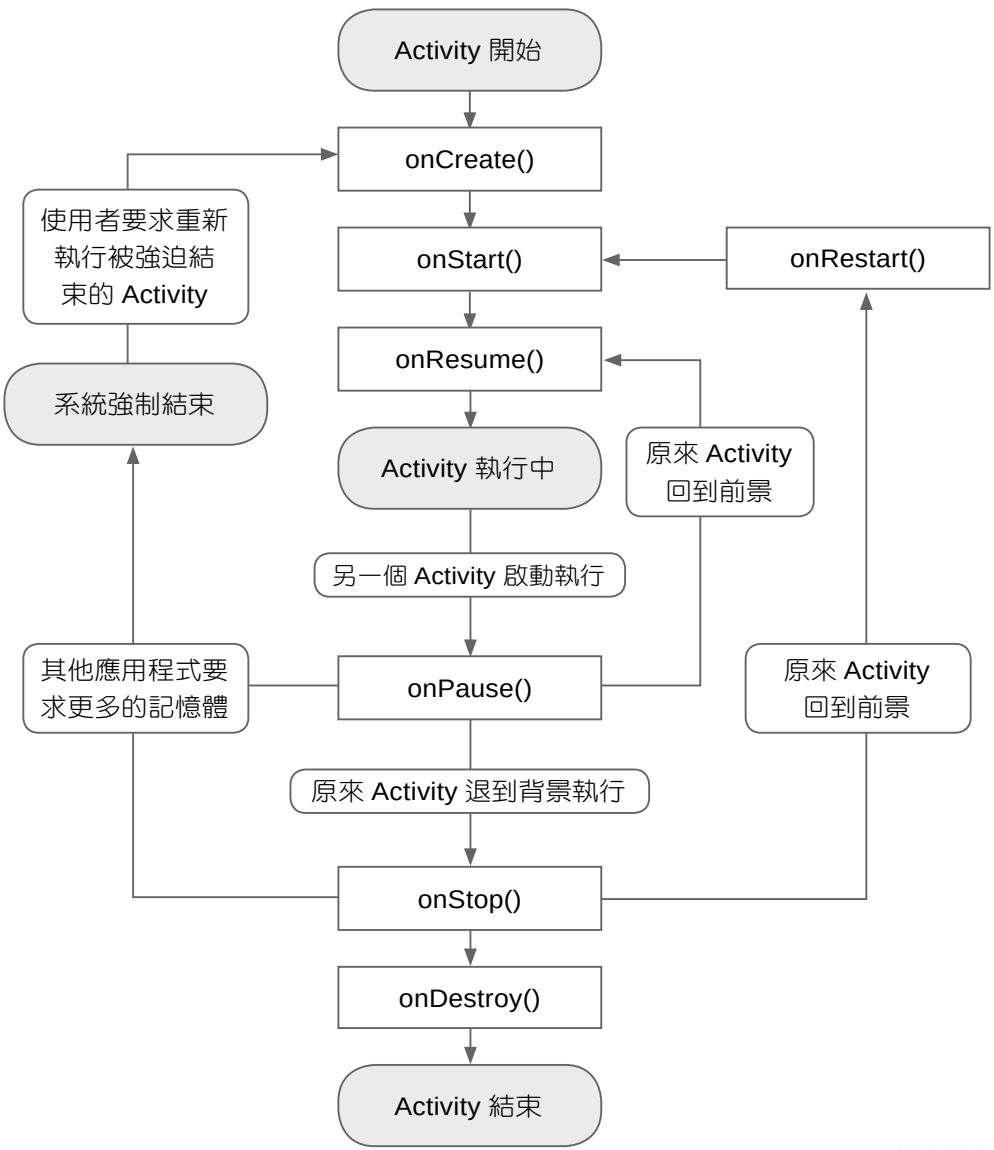
學習重點

- Activity 的生命週期
- 呼叫內建的 Activity
- 呼叫自訂的 Activity
- 由系統強制回收後再啟動
- 系統記憶體不足時的處理
- 觀察 Activity 生命週期

10

10.2 Activity 運作流程

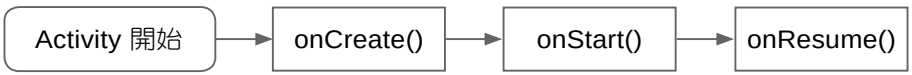
下圖是 Activity 運作時各種狀態的流程圖。其中矩形代表各種可以呼叫調用的方法，也就是當 Activity 狀態改變時可以執行的動作。灰色的橢圓形代表 Activity 會執行的重要狀態。



生命週期包含許多種不同的狀態和變化，在實際應用上，Activity 較常使用的運作流程有以下幾種。

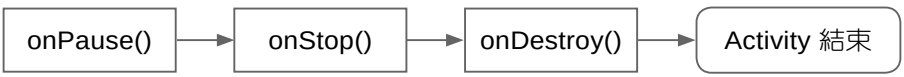
10.2.1 啟動 Activity

當一個 Activity 執行後會先執行 onCreate()，系統也在此時配置資源，接著執行 onStart() 和 onResume()，此時就可以在螢幕上看到這個 Activity。



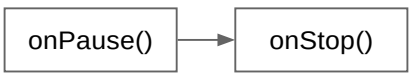
10.2.2 結束一個 Activity

當按下手機的 鍵或以程式的 finish() 結束 Activity，執行過程為 onPause()，接著執行 onStop() 和 onDestroy()，通常會在 onDestroy() 時釋放資源。



10.2.3 呼叫內建的 Activity

在執行應用程式時，使用者執行 Toast、AlertDiaog 或按下手機的 返回桌面後再按 鈕執行撥打電話應用程式，就會使得目前執行的 Activity 進入暫停狀態，目前執行的 Activity 會退到背景執行，但是 Activity 並未結束。此時執行過程為先執行 onPause() 接著執行 onStop()。然後將執行權交給另一個內建的 Activity。

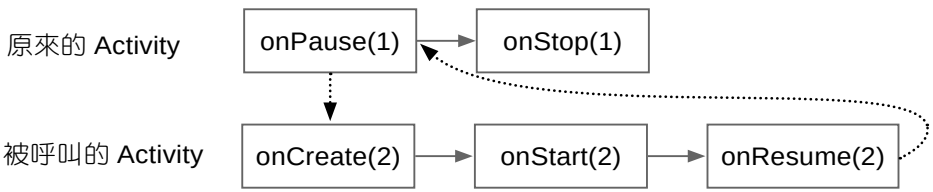


10.2.4 由內建的 Activity 返回原來的 Activity

等接完電話回來時，原來進入暫停的 Activity 又會回復執行。這個時候執行過程為先執行 onRestart() 接著執行 onStart() 和 onResume()，原來的 Activity 就會取回螢幕控制權。

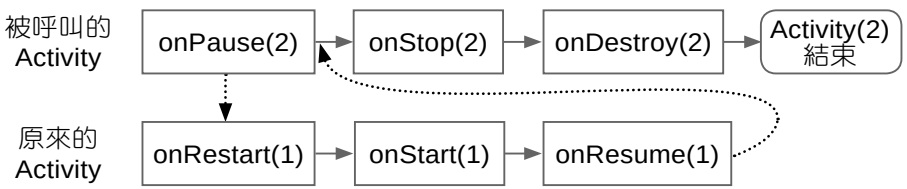
10.2.5 呼叫自訂的 Activity

從目前執行的 Activity 呼叫另一個自訂的 Activity，目前執行的 Activity 會進入暫停的狀態，也就是退到背景執行，但是 Activity 並未結束；同時另一個 Activity 被啟動執行。執行過程為 onPause(1) → onCreate(2) → onStart(2) → onResume(2) → onStop(1)。



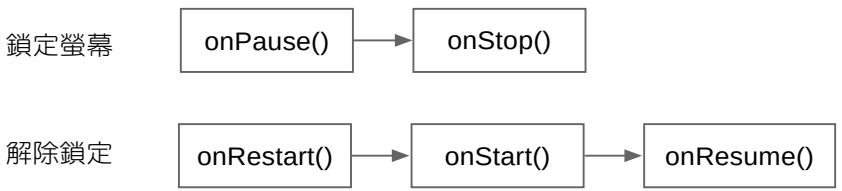
10.2.6 結束自訂的 Activity 返回原來的 Activity

接著如果由自訂的 Activity 結束返回原來呼叫的 Activity，自訂的 Activity 將會結束，而原來的 Activity 又會再回復執行。執行過程為 onPause(2) → onRestrat(1) → onStart(1) → onResume(1) → onStop(2) → onDestroy(2)。



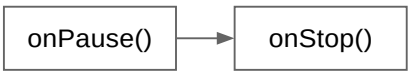
10.2.7 按 POWER 鍵鎖住螢幕 / 解除鎖定

在執行 Activity 時按下 **Power** 鍵鎖住螢幕時，將執行 onPause() → onStop()。再按 **Power** 並開啟滑蓋解除鎖定時，將執行 onRestart() → onStart() → onResume()，並取回原來 Activity 的螢幕控制權。



10.2.8 按 HOME 鍵

程式啟動會執行 onCreate() → onStart() → onResume() 並進入主畫面，這時在主畫面按 將返回桌面，執行過程為：



10.2.9 重新執行原來的程式

返回桌面後，在桌面選取原來程式的圖示，將會取回原來 Activity 的螢幕控制權，執行過程為：

10.2.10 由系統強制回收後再啟動

當 Android 系統記憶體不足時，就必須將較不重要的應用程式移除，移除的決定是由系統依據 Activity 生命週期來判斷。當 Activity 被系統強制回收後，如果要再啟動執行，可以依一般 Activity 啟動的方式啟動執行。執行過程為：

Android 2.X POWER 鎖住螢幕 / 解除鎖定方式

Android 6.X 和 Android 2.X 處理 POWER 鎖住螢幕 / 解除鎖定方式稍有不同。在 Android 2.X 目前執行的 Activity 按下 **Power** 鍵鎖住螢幕時，將執行 onPause()，再按 **Power** 並開啟滑蓋解除鎖定時，將執行 onResume()，並取回原來 Activity 的螢幕控制權。

10.3 觀察 Activity 生命週期

以下將用實際的範例來說明 Activity 生命週期的應用。

10.3.1 範例：Activity 的生命週期觀察

在以下的範例中，將利用幾個按鈕來測試 Activity 生命週期的觀察：

1. 按下 **呼叫撥號按鈕** 鈕會執行撥打電話的應用程式，在撥打電話的應用程式按下 按鈕可以結束返回 ExActivity01 的應用程式中。也可以按下 返回桌面後再按 按鈕執行撥打電話的應用程式，然後再按下 和 按鈕從應用程式中選擇 ExActivity01 應用程式返回原來執行的 ExActivity01 應用程式中。

2. 按下 **開啟另一個自訂的 Activity** 鈕，原來 ExActivity01 進入暫停的狀態，然後開啟自訂的另一個 Activity：「Second」。在「Second」類別中按下 **返回主程式** 鈕或手機面板上的 按鈕結束「Second」，這時原來進入暫停的狀態的 ExActivity01 將被復原。

3. 按下 **FINISH() 結束** 鈕相當於按下 按鈕，會結束應用程式。

在這些應用程式的執行過程中，我們將使用 Toast 訊息來觀察生命週期的變化。

新增專案並完成主版面配置

請建立 <ExActivity01> 專案，開啟 <content_main.xml> 版面配置檔後佈建三個 Button，命名為 btnDial、btnPage2 和 btnFinish，分別開啟對應的 Intent。

<ExActivity01/app/res/layout/content_main.xml>

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/
  res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    …略
    tools:context=".MainActivity" >

    <Button android:id="@+id/btnDial"
       android:layout_width="fill_parent"
       android:layout_height="wrap_content"
       android:text="呼叫撥號按鈕" />

   <Button android:id="@+id/btnPage2"
      android:layout_width="fill_parent"
      android:layout_height="wrap_content"
      android:layout_below="@id/btnDial"
      android:text="開啟另一個自訂的 Activity" />
```

多媒體應用

MediaPlayer 元件可以播放音訊及視訊，並且進行控制。VideoView 元件可以播放視訊。MediaRecorder 元件可以進行媒體採樣，要製作手機錄音功能軟體就不是件困難的任務了！

學習重點

- MediaPlayer 元件
- VideoView 視訊播放器
- SurfaceView 元件語法
- MediaPlayer 與 SurfaceView 結合
- 錄製音訊

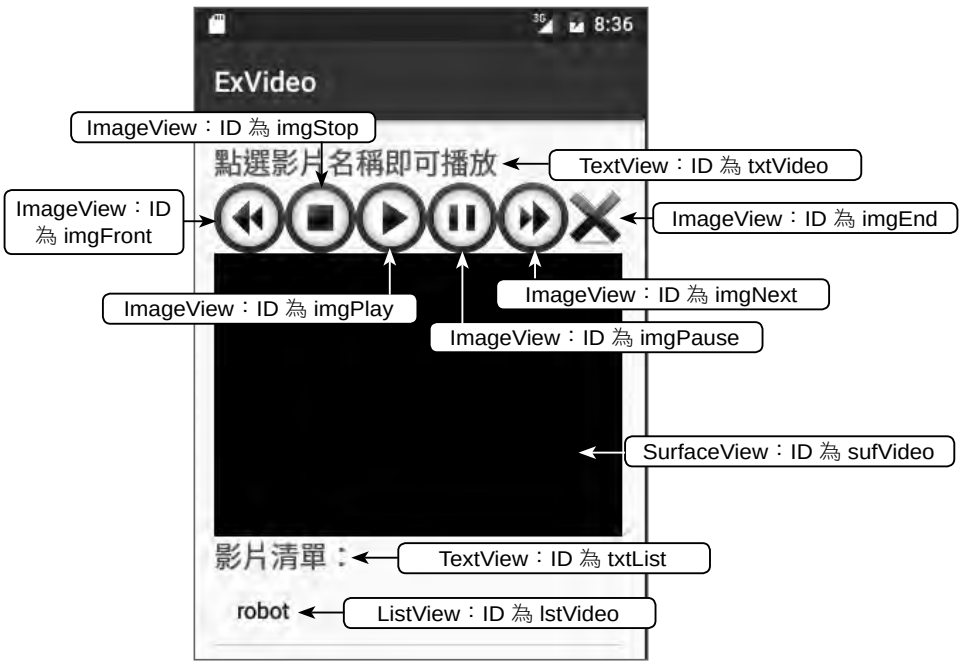
13

13.2.5 範例：自訂格式視訊播放器

本範例視訊播放器功能、介面與操作都與 13.1.4 節的播放音訊範例相同。

新增專案並完成版面配置

新增 <ExVideo> 專案，<content_main.xml> 版面配置檔完成如下：



加入執行的程式碼

整體變數宣告及 onCreate() 啟動程式碼：

<ExVideo/app/java/com.ehappy.exvideo/MainActivity.java>

```
…略
31 private SurfaceView sufVideo;
…略
39 private MediaPlayer mediaplayer;
40 private SurfaceHolder sufHolder;
41 @Override
42 protected void onCreate(Bundle savedInstanceState) {
…略
66    mediaplayer=new MediaPlayer();
67    //建立 Surface 相關物件
68    sufHolder=sufVideo.getHolder();
69    if (ActivityCompat.checkSelfPermission(MainActivity.this,
         Manifest.permission.READ_EXTERNAL_STORAGE) !=
         PackageManager.PERMISSION_GRANTED) {
70       ActivityCompat.requestPermissions(MainActivity.this, new
            String[]{Manifest.permission.READ_EXTERNAL_STORAGE}, 1);
71    }
72 }
```

- 第 40 列，宣告 SurfaceHolder 物件整體變數，於第 68 列做相關設定後，此 SurfaceHolder 物件即可做為視訊顯示用。
- 第 69-71 列，檢查是否取得執行時授權。

播放視訊的 playVideo 方法程式碼：

續：<ExVideo/app/java/com.ehappy.exvideo/MainActivity.java>

```
140 private void playVideo(String path) {
141   mediaplayer.reset();
142   mediaplayer.setAudioStreamType(AudioManager.STREAM_MUSIC);
143   mediaplayer.setDisplay(sufHolder);
144   try
145   {
146      mediaplayer.setDataSource(path); //播放影片路徑
147      mediaplayer.prepare();
148      mediaplayer.start(); //開始播放
149      txtVideo.setText("影片名稱：" + videoname[cListItem]); //更新名稱
```

```
150     mediaplayer.setOnCompletionListener(new OnCompletionListener() {
151        public void onCompletion(MediaPlayer arg0) {
152           nextVideo(); //播放完後播下一片
153        }
154     });
155   } catch (IOException e) {}
156   falgPause=false;
157 }
```

- 第 142-143 列，設定 MediaPlayer 元件的資料流格式及顯示的 SurfaceHolder 物件後就可播放視訊。
- 第 146 列，因為本範例的視訊檔位於 SD 卡，所以要使用 setDataSource 方法設定視訊來源檔案。

其餘處理按鈕程式 (listener)、選取 ListView 項目處理程式 (lstListener)、播放上一片 (frontVideo) 及播放下一片 (nextVideo) 等的程式碼皆與 MusicPlayer 專案相同，不再贅述。

儲存專案後執行專案。

Google Maps 應用程式

Google 地圖自 Android 4.X 開始就建議使用最新的 Google Maps Android API 。除了必須建立 **Google Maps Activity** 型別專案外，還要申請 Google Maps 的 API Key。

Google 地圖程式可以應用的範圍很廣，除了單純的顯示地圖位置、地標之外還能切換不同的顯示模式。搭配上不同的應用資訊，對於日常生活、甚至商務應用都能有所發揮。

學習重點

- Google Maps 應用程式準備工作
- 建立 Google Maps 應用程式
- 加入 Google Maps 控制功能
- 取得現在位置的相關資訊
- 在 Google Maps 加上標記

15

15.3 加入 Googl Maps 控制功能

Google Maps 應用程式預設會顯示指南針和比例縮放圖示，也可以控制不顯示之。此外也可以設定地圖的顯示樣式、可否使用手勢控制地圖縮放，同時也可以建立觀看地圖的視點位置。

15.3.1 地圖的顯示樣式

GoogleMap 的 setMapType() 方法可以改變地圖顯示樣式。共有 5 種地圖樣式：

樣式名稱	顯示樣式
Normal	一般地圖。可顯示道路、河流等自然景觀。
Hybrid	混合衛星地圖及道路地圖。
Satellite	衛星地圖。
Terrain	地形圖。可顯示道路、河流等自然景觀、地形數據和輪廓線。
None	不顯示地圖，只顯示空的網格。

例如：設定 mMap 地圖顯示的樣式為一般地圖。

```
mMap.setMapType(GoogleMap.MAP_TYPE_NORMAL); // 一般地圖
```

15.3.2 設定是否顯示指南針或比例縮放圖示

以 GoogleMap 的 getUiSettings() 方法可設定或取得現在地圖的狀態，最重要的是以 setCompassEnabled()、setZoomControlsEnabled() 方法設定指南針和比例縮放圖示是否顯示，預設為 true，若設為 false，將不顯示。

例如：顯示比例縮放圖示，不顯示指南針。

```
mMap.getUiSettings().setZoomControlsEnabled(true);
mMap.getUiSettings().setCompassEnabled(false);
```

15.3.3 設定是否可用手勢控制

getUiSettings() 方法也可設定多種手勢。getUiSettings() 手勢常用的方法如下：

方法	顯示樣式
setAllGesturesEnabled (boolean enabled)	開啟或關閉所有的手勢。
setRotateGesturesEnabled(boolean enabled)	是否可用手勢控制地圖旋轉。
setScrollGesturesEnabled(boolean enabled)	是否可用手勢控制地圖左右移動。
setTiltGesturesEnabled(boolean enabled)	是否可用手勢調整俯視的角度。
setZoomGesturesEnabled(boolean enabled)	是否可用手勢控制地圖縮放。

例如：設定可使用手勢控制地圖縮放。

```
mMap.getUiSettings().setZoomGesturesEnabled(true);
```

15.3.4 設定視點的位置

以 CameraUpdateFactory 物件的 newCameraPosition() 方法可以建立觀看地圖的視點位置，再以 animateCamera 方法移到該視點位置。

語法：

```
GoogleMap 變數 .animateCamera(CameraUpdateFactory.
newCameraPosition( 視點位置 ));
```

參數 **視點位置** 是一個 CameraPosition 型別物件，包含 target(位置)、zoom(縮放)、bearing(指南針旋轉角度) 及 tilt(俯視的角度)。

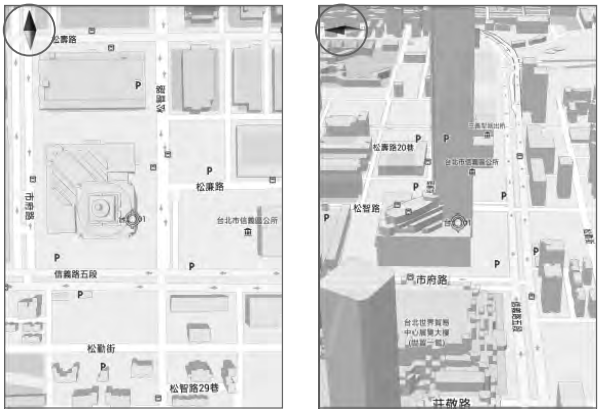
例如：設定地圖的視點位置為台北 101、放大倍率為 17、指南針旋轉 90 度、俯視的角度 90 度。

```
CameraPosition cameraPosition = new CameraPosition.Builder()
  .target(Taipei101)  // 台北 101
  .zoom(17)           // 放大倍率
  .bearing(90)        // 指南針旋轉 90 度
  .tilt(90)           // 俯視的角度
  .build();           // 建立 CameraPosition 物件

mMap.animateCamera(CameraUpdateFactory.newCameraPosition(
  cameraPosition));
```

bearing 為指南針 (或稱指北針) 的角度，角度為 0^0~360^0，預設為 0^0 表示指向北方，即螢幕的上方為地圖的北方；若設定 bearing=90 則指南針旋轉 90^0 朝向左邊，如此螢幕的左方為地圖的北方，螢幕的上方則為地圖的東方。tilt 為由空中俯視的角度，角度為 0^0~90^0，左下圖為 0^0 表示由空中俯視，右下圖 tilt=90 表示由地面平視。

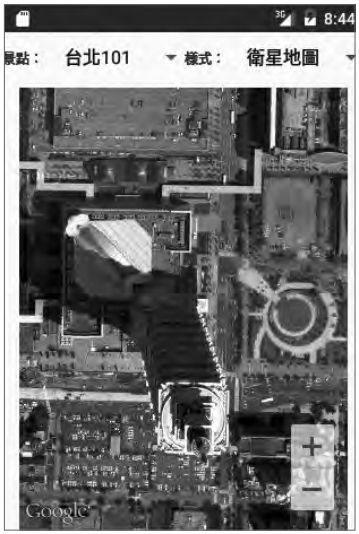
要特別注意的是有時指南針圖示不會出現，要顯示指南針圖示必須在地圖上用二隻手指點觸再左右旋轉才會出現。



15.3.5 範例：設定 Google Maps 的顯示樣式、視點位置

建立 Google Maps 應用程式專案，預設以一般地圖模式顯示，縮放比例為 17，顯示中心點為「台北 101」，顯示地圖縮放圖示、指南針。也可以從 **樣式** 下拉式選單中選擇以一般地圖、混合地圖、衛星地圖或地形圖顯示。

景點 下拉式清單中可選擇顯示的地圖中心點，包括「台北 101」、「日月潭」和「高雄六合夜市」。



新增 Google Maps 專案

新增 **Google Maps Activity** 型別專案 <ExGoogleMap02>。

更改應用程式名稱

將 **Title** 欄位由預設「Map」更改為應用程式名稱「ExGoogleMap02」。