

Android藍芽聊天室

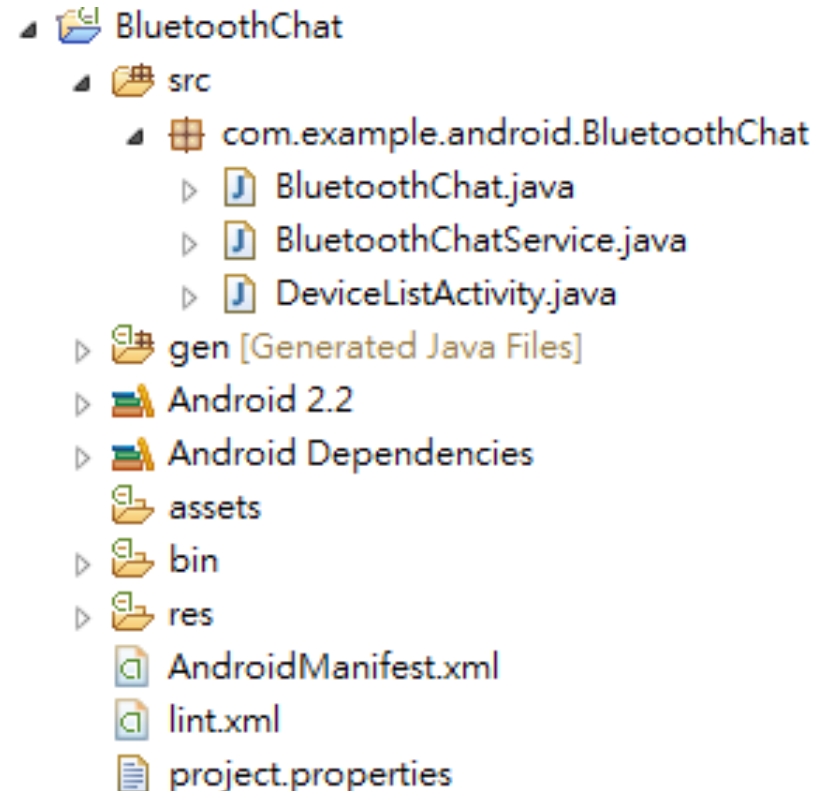
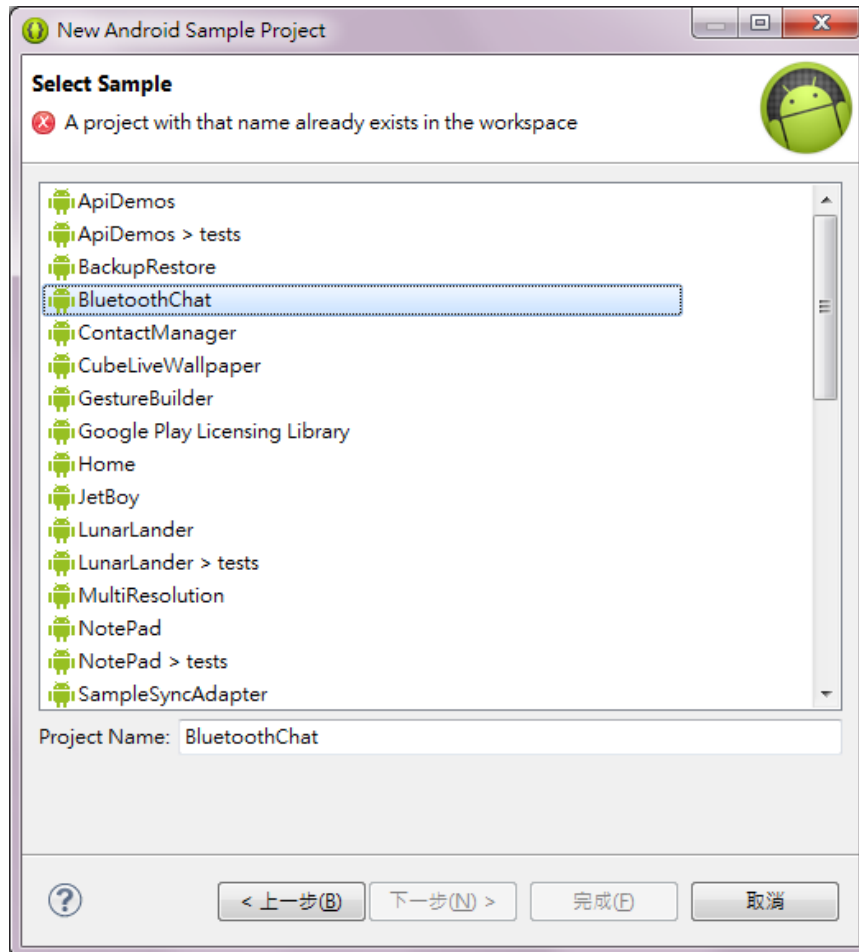
SDK內的範例程式

建國科技大學 資管系

饒瑞佺

2013/7 V1

Android 2.2 SDK



Android藍芽聊天室

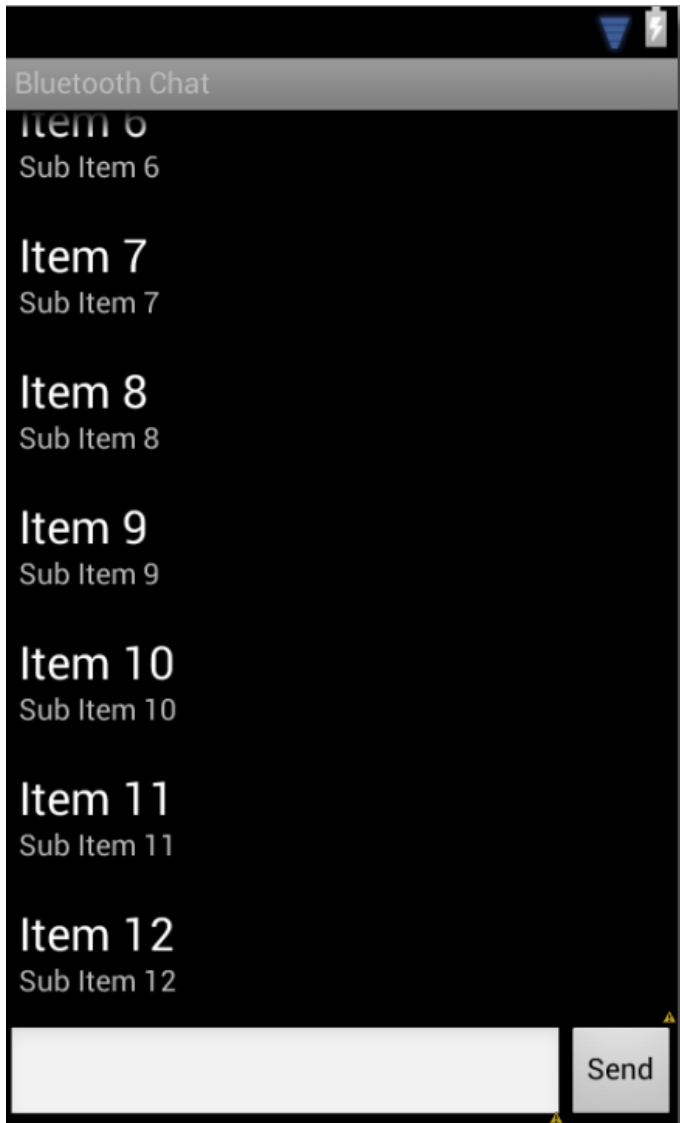
- **BluetoothChat.java**：主畫面，顯示聊天訊息與設定藍芽
- **BluetoothChatService.java**：藍芽傳輸service，包含三個主要thread：
 - AcceptThread：負責監聽藍芽socket
 - ConnectThread：負責藍芽socket連線
 - ConnectedThread：藍芽連線後通訊程序
- **DeviceListActivity.java**：搜尋藍芽設備介面

AndroidManifest.xml

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.android.BluetoothChat"
    android:versionCode="1"
    android:versionName="1.0">
    <uses-sdk minSdkVersion="6" />
    <uses-permission android:name="android.permission.BLUETOOTH_ADMIN" />
    <uses-permission android:name="android.permission.BLUETOOTH" />

    <application android:label="@string/app_name"
        android:icon="@drawable/app_icon" >
        <activity android:name=".BluetoothChat"
            android:label="@string/app_name"
            android:configChanges="orientation|keyboardHidden">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
        <activity android:name=".DeviceListActivity"
            android:label="@string/select_device"
            android:theme="@android:style/Theme.Dialog"
            android:configChanges="orientation|keyboardHidden" />
    </application>
</manifest>
```

layout



ListView

從下往上顯示

android:stackFromBottom="*true*"

TextView

Button

BluetoothChat.java

- onCreate :
 - 判斷設備是否支援藍芽
 - 取得BluetoothAdapter : `mBluetoothAdapter`
- onStart :
 - 透過BluetoothAdapter判斷藍芽是否開啟 + `onActivityResult()`
 - 呼叫 `setupChat()`
- onResume : 恢復 service
- onDestroy : 停止 service

BluetoothChat.java

- **setupChat()** :
 - 設定arrayadapter用以顯示對話資訊(對應到listview)
 - 監聽輸入框enter事件(按軟體鍵盤enter也可以送出)
 - mOutEditText.setOnEditorActionListener(mWriteListener);
 - 設定送出按鈕
 - 取得mChatService = new BluetoothChatService(this, **mHandler**);

取得service
對應狀態

BluetoothChatService.java

BluetoothChat.java

- **sendMessage:**

- 使用 mChatService 判斷是否連線

- 送出資訊

- `byte[] send = message.getBytes();`

- `mChatService.write(send);`

BluetoothChat.java

- onCreateOptionsMenu : 建立menu選項
- onOptionsItemSelected : 設定menu對應事件
 - 掃描 : DeviceListActivity.class + onActivityResult()
 - 可被發現 : ensureDiscoverable()
- Handler mHandler : 處理對話訊息的進出

BluetoothChatService.java

```
/**  
 * This class does all the work for setting up and managing Bluetooth  
 * connections with other devices. It has a thread that listens for  
 * incoming connections, a thread for connecting with a device, and a  
 * thread for performing data transmissions when connected.  
 */
```

全部共3個thread

- AcceptThread：負責監聽藍芽socket
- ConnectThread：負責藍芽socket連線
- ConnectedThread：藍芽連線後通訊程序

BluetoothChatService.java

- 建構子：
 - 取得BluetoothAdapter：`mAdapter`
- start()：
 - 對應AcceptThread
 - 從BluetoothChat.java的onResume()呼叫
- connect(BluetoothDevice device)：
 - 對應ConnectThread
 - 從BluetoothChat.java的onActivityResult內的case
REQUEST_CONNECT_DEVICE 呼叫
- connected(BluetoothSocket socket, BluetoothDevice device)：
 - 從BluetoothChatService.java的AcceptThread呼叫

DeviceListActivity.java

- 在AndroidManifest.xml中設定android:theme

android:theme="@android:style/Theme.Dialog"
將一個Activity顯示為對話框模式

```
<activity android:name=".DeviceListActivity"  
    android:label="@string/select_device"  
    android:theme="@android:style/Theme.Dialog"  
    android:configChanges="orientation|keyboardHidden" />
```

android:theme

<code>android:theme="Theme.Light"</code>	背景為白色
<code>android:theme="Theme.Light.NoTitleBar"</code>	白色背景並無標題欄
<code>android:theme="Theme.Light.NoTitleBar.Fullscreen"</code>	白色背景，無標題欄，全螢幕
<code>android:theme="Theme.Black"</code>	背景黑色
<code>android:theme="Theme.Black.NoTitleBar"</code>	黑色背景並無標題欄
<code>android:theme="Theme.Black.NoTitleBar.Fullscreen"</code>	黑色背景，無標題欄，全螢幕
<code>android:theme="Theme.Wallpaper"</code>	用系統桌面為應用程式背景
<code>android:theme="Theme.Wallpaper.NoTitleBar"</code>	用系統桌面為應用程式背景，且無標題欄
<code>android:theme="Theme.Wallpaper.NoTitleBar.Fullscreen"</code>	用系統桌面為應用程式背景，無標題欄，全螢幕
<code>android:theme="Translucent"</code>	
<code>android:theme="Theme.Translucent.NoTitleBar"</code>	
<code>android:theme="Theme.Translucent.NoTitleBar.Fullscreen"</code>	
<code>android:theme="Theme.Panel"</code>	
<code>android:theme="Theme.Light.Panel"</code>	
<code>android:theme="@android:style/Theme.Dialog"</code>	將一個Activity顯示為對話框模式
<code>android:theme="@android:style/Theme.NoTitleBar"</code>	不顯示應用程式標題欄
<code>android:theme="@android:style/Theme.NoTitleBar.Fullscreen"</code>	不顯示應用程式標題欄，並全螢幕

DeviceListActivity.java

```
// Setup the window  
requestWindowFeature(Window.FEATURE_INDETERMINATE_PROGRESS);
```

- <http://www.cnblogs.com/salam/archive/2010/11/30/1892143.html>

DeviceListActivity.java

- onCreate :
 - 設定arrayadapter用以顯示配對
(mPairedDevicesArrayAdapter)與未配對的設備
(mNewDevicesArrayAdapter)
 - 設定BroadcastReceiver(mReceiver)，用以接收設備搜尋完畢
 - 取得BluetoothAdapter (mBtAdapter)
 - 設定按鈕進行doDiscovery()
- onDestroy()
 - 取消BluetoothAdapter (mBtAdapter)

DeviceListActivity.java

- OnItemClickListener :
 - 取得所選設備MAC
 - `intent.putExtra(EXTRA_DEVICE_ADDRESS, address);`
 - 回到BluetoothChat.java 中的onActivityResult , 並取得設備MAC
 - `String address = data.getExtras().getString(DeviceListActivity.EXTRA_DEVICE_ADDRESS);`