

## Android Keyword Dirven Test Tool

專題編號：108-CSIE-S008

執行期限：107 年第 1 學期至 108 年第 1 學期

指導教授：劉建宏 教授

專題參與人員：105820059 吳彥陞

### 一、摘要

在 Android 應用程式的數量越來越多的情況下，其品質也日益受到重視，多數的 Android 自動化測試工具，都需要具有程式背景的人員去撰寫或是維護低階的腳本，在感測器事件的支援並非十分充足，大部分的情況都要手動進行測試。為了使感測器相關事件的測試更有效率，本專題將關鍵字驅動測試 (Keyword Driven Test) 的概念延伸運用，研發一套 Android 自動化測試工具和驅動測試 APP，並支援部分感測器測試，但對於實體裝置進行有效率的自動化測試感測器相關事件仍相當困難。

關鍵詞：關鍵字驅動測試、感測器

### 二、緣由與目的

搭載 Android 系統的行動裝置，是目前市場上主流之一，其應用程式的數量也越來越多，品質也越來越受到重視，自動化測試工具的需求也日益增加。多數的自動化測試工具，對於感測器的測試並非十分完整，加上難以覆寫、阻斷 Android 實體裝置感測器的資訊，使得難以針對感測器事件進行自動化測試。

為此將研發一套將關鍵字驅動測試適用於 Android 的自動化測試的工具和驅動測試 APP，支援 Emulator 進行感測器相關事件測試和蒐集實體感測器資料給開發者進行判斷。

### 三、實作方法

驅動測試 APP 方面，除了設計 Android UI 元件相關的關鍵字之外，也

會設計和感測器有關的關鍵字。在測試工具方面，開發一個簡單的命令行工具負責解析測試腳本，並傳送測試命令給 Android 裝置中的驅動測試 APP 執行測試。此工具同時支援實體裝置和 Android SDK(軟體開發套件)中的 Emulator 來進行測試。

#### ● 測試感測器事件

當系統接收到改寫 Emulator 感測器資料的關鍵字後，會先透過 Socket 將 Emulator 感測器中的資料改寫，再將結果紀錄在測試報告中。

#### ● 蒐集實體感測器資料

由於在 UI Automator 2 中，所有的測試都必須綁定一個 Target APP，因此本專題設計一個沒有任何畫面的 APP 作為載體，透過此 APP 能夠從外部執行定義於 manifests 中的 service，藉由此 service 來蒐集實體裝置感測器的資料。

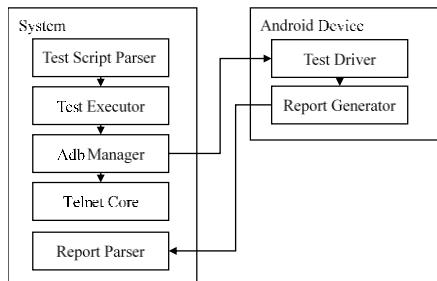
### 四、系統架構與流程

圖一為本專題的系統架構圖，其架構分為 7 個子系統，其架構如圖一所示：

- Test Script Parser : 將接收到測試腳本轉換成測試案例。
- Test Executor : 負責執行測試案例。
- Adb Manager : 將命令傳送給 SDK 中的 adb.exe，以命令行的方式驅動測試執行。
- Telnet Core : 使用 Socket 的方式改寫 Emulator 的資料。

# 實務專題計畫成果摘要報告

- Report Parser : 負責將 xml 的測試報告解析並輸出在命令行中。



圖一、系統架構圖

- Report Generator : 負責蒐集測試產生的資料和分析這些資料，最後產生報表。
- Test Driver: 安裝在 Android 裝置中的測試 APP，負責驅動 Android 測試和蒐集資料。執行測試前，系統會將此 APP 傳送到 Android 裝置上

系統使用流程大致如下：

- (一) 使用編輯器撰寫測試腳本，圖二為測試腳本的範例

```
<?xml version="1.0" encoding="UTF-8"?>
<testScript>

    <testCase>
        <name>deviceClick</name>
        <category>Device</category>
        <method>click</method>
        <parameter>
            <pos_x>250</pos_x>
            <pos_y>100</pos_y>
            <wait_time>2000</wait_time>
        </parameter>
    </testCase>

    <testCase>
        <name>deviceIdle</name>
        <category>Device</category>
        <method>idle</method>
        <wait_time>2000</wait_time>
    </testCase>
</testScript>
```

圖二、測試腳本範例圖

- (二) 在命令行中使用測試工具，如圖三所示，使用者在命令行中輸入 java -jar KwTestTool.jar testCalculator.xml，執行 testCalculator.xml 測試腳本

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.1069]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
D:\大學\專題\KwTestTool>java -jar KwTestTool.jar testCalculator.xml
```

圖三、執行測試工具範例圖

- (三) 等待系統安裝驅動測試 APP，如圖四所示。

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.1069]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
D:\大學\專題\KwTestTool>java -jar KwTestTool.jar testCalculator.xml
Install test core apk to device
Success
```

圖四、安裝 APP 範例圖

- (四) 執行測試過程，如圖五所示。

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.17134.1069]
(c) 2018 Microsoft Corporation. 著作權所有，並保留一切權利。
D:\大學\專題\KwTestTool>java -jar KwTestTool.jar testCalculator.xml
Install test core apk to device
Success
current execute: clickCalculator
current execute: clickButton_3
current execute: clickButton_5
current execute: checkEqual
```

圖五、執行測試過程範例圖

- (五) 產生測試報告，如圖六所示。

```
<?xml version="1.0" encoding="UTF-8"?>
<report>
    <testCase>
        <name>clickCalculator</name>
        <category>Device</category>
        <method>click</method>
        <result>PASS</result>
        <message />
        <time>0.127 s</time>
    </testCase>
    <testCase>
        <name>clickButton_3</name>
        <category>UiComponent</category>
        <method>click</method>
        <result>PASS</result>
        <message />
        <time>0.859 s</time>
    </testCase>
    <testCase>
        <name>clickButton_5</name>
        <category>UiComponent</category>
        <method>clickSeveralTimes</method>
        <result>PASS</result>
        <message />
        <time>2.747 s</time>
    </testCase>
    <testCase>
        <name>checkEqual</name>
        <category>UiComponent</category>
        <method>assertText</method>
        <result>PASS</result>
        <message />
        <time>0.748 s</time>
    </testCase>
</report>
```

圖六、測試報告範例圖

## 五、結語

本專題將實作一個 Android 關鍵字驅動測試的測試工具和驅動測試 APP，並且提供感測器相關事件的測試。透過撰寫 xml 的測試腳本使測試腳本更容易閱讀，降低撰寫 Android 測試腳本的成本，並且透過 Android SDK(軟體開發套件)中的 emulator 工具，透過 socket 去改

寫部分感測器、電池、GPS 資料，和蒐集實體感測器資料，使得自動化相關事件的難度下降。儘管如此，目前仍然難以對實體裝置進行有效率的感測器事件測試。

這個專題讓我明白自己能力的不足，從一開始的選題、找資料、設計到程式的撰寫，都遇到相當多的問題，雖然不斷嘗試，但最後的成品和最初想做的落差非常大，自己一人製作的過程中，常常陷入設計的盲點和沒注意到的問題，導致花費了大量的時間去修改寫不好的地方，到最後時間不足急忙趕出個成品。

## 六、參考文獻

- [1] 郭柏廷，Android 應用程式之 GUI 自動化測試方法，碩士論文，國立臺北科技大學資訊工程研究所，臺北，2013 年。
- [2] 陳皇各，Android 關鍵字驅動測試函式庫與工具之設計與開發，碩士論文，國立台北科技大學資訊工程研究所，臺北 2014 年
- [3] 邱威霖，利用 Arduino 機器手測試 Android 應用程式非靜止行為研究，碩士論文，國立臺北科技大學資訊工程研究所，臺北，2015 年