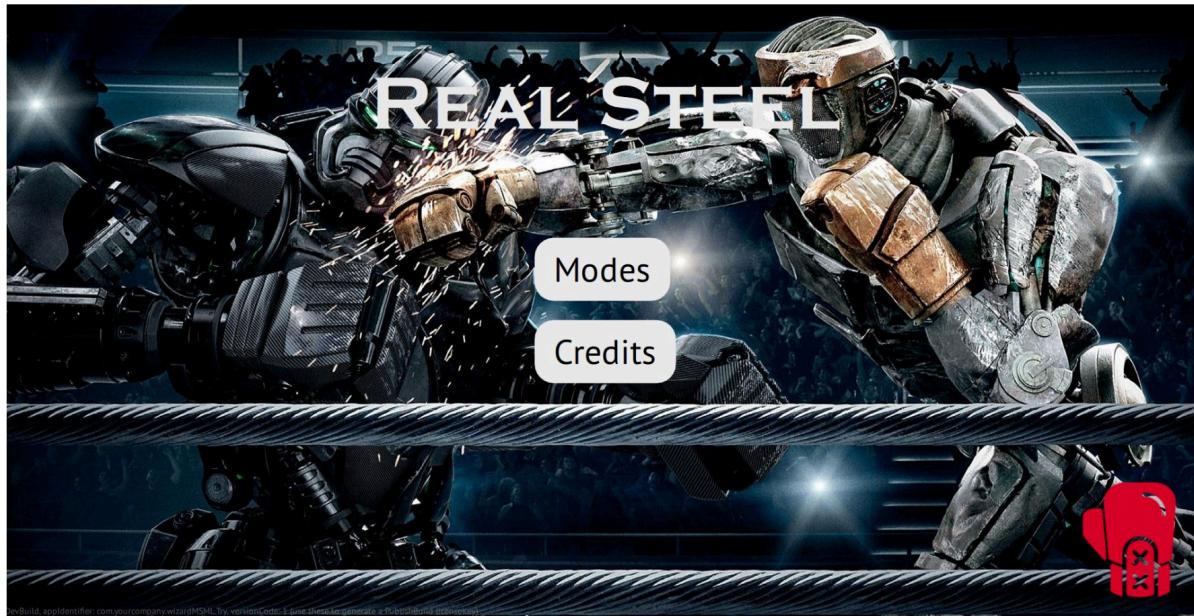


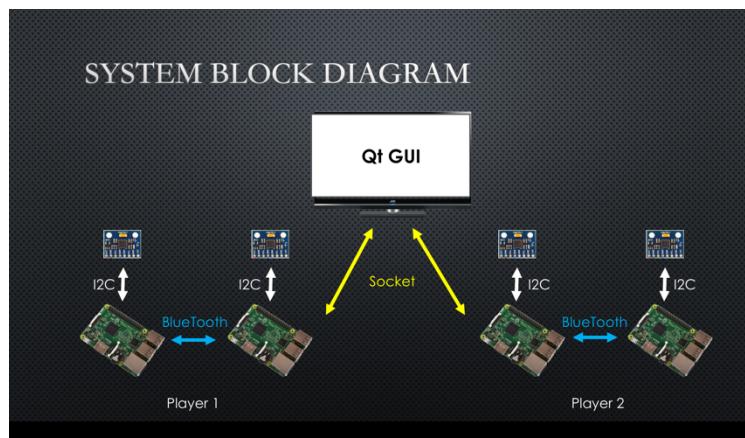
# REAL STEEL 鋼鐵擂台

吳建昇 b01901045、謝沅廷 b01901046



## 摘要

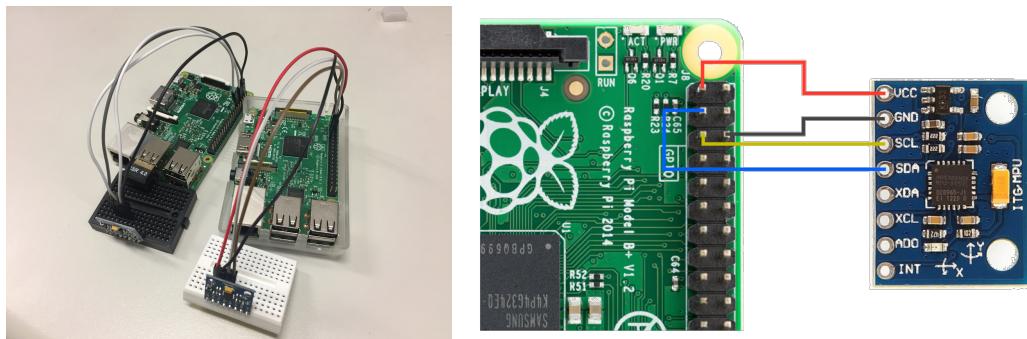
本組利用Raspberry Pi設計一款拳擊遊戲，讓使用者能透過現實世界的肢體動作來體驗揮拳、防禦以及傾斜等操作，利用四個Raspberry Pi 3、四個 MPU6050 六軸陀螺儀，並結合 Qt GUI Programming來完成介面設計，其中MPU6050和Pi之間利用I2C溝通、左右手間用Bluetooth SPP連接、Pi和Qt間使用TCP Socket串接，而螢幕端則是QML處理事件和製作畫面動作，系統圖如下：



# 作法

## I. Raspberry Pi & MPU6050

1. 使用 PiBits library: <https://github.com/richardghirst/PiBits>
2. 連結兩個裝置如下圖，並設定 I2C address: 0x68



3. 利用 Yaw-Pitch-Roll 角度以及真實加速度 (扣除重力的影響) 偵測使用者的動作
4. 設定 pitch 的角度界線在 -15 度 和 +15 度. 設定 roll 角度界線在 -45 度. 設定加速的感測界線為 5g ( $m/s^2$ ).
5. De-Bounce 設定：避免感測到多次的加速度，加入了類似 de-bounce 的設定，必須要有兩次的加速度變換才會偵測到一次的出拳（揮拳的瞬間跟停下來的瞬間會有兩次的加速度變化），另外揮拳會有出拳和收拳兩個動作，因此為了解決這個狀況，我們限制了出拳攻擊偵測的速度，也就是說我們假設一個人無法在 1 秒之內出兩拳（利用 sleep 和 counter 來完成）

## II. Raspberry Pi & Raspberry Pi

1. 左手和右手利用 Serial Port Profile (SPP) 去彼此溝通，左手設定為 SPP-client 主動的去連接右手的 SPP-Server
2. 配對兩者的藍芽模組，利用以下指令配對
  - A. sudo bluetoothctl 進入設定介面
  - B. 輸入設定 agent on, default-agent, scan on, discoverable on 並尋找配對的藍芽位址

- C. trust XX-XX-XX-XX-XX-XX 然後再 pair XX-XX-XX-XX-XX-XX，兩端同時同意配對即可，之後不用再重複此動作。
- 3. Client 端建立 rfcomm socket 然後指定去連接 remote device 的 port 10. Server 端設定藍芽的 BT-Dongle Class 為 0x1f00 並設定相關的 SPP\_SPEC (SDP Interoperability Requirements).

### III. Raspberry Pi & Qt GUI

- 1. Qt GUI acts as a TCP server and Raspberry Pi is a TCP client, which one user connect to port 8888 and the other to 7777.
- 2. Qt supports some TCP socket library such as <QTcpSocket>, <QtNetwork>, etc. We define two main function, acceptConnection() and startRead() to wait and read the data from Pi.

```
Server::Server(QObject* parent): QObject(parent)
{
    connect(&server, SIGNAL(newConnection()),
            this, SLOT(acceptConnection()));
    server.listen(QHostAddress::Any, 8888);
}

void Server::acceptConnection()
{
    client = server.nextPendingConnection();
    connect(client, SIGNAL(readyRead()),
            this, SLOT(startRead()));
}

void Server::startRead()
{
    char buffer[1024] = {0};
    client->read(buffer, client->bytesAvailable());
    cout << buffer << endl;
    //client->close();
}
```

- 3. TCP client on Pi:

- A. 設定 server.sin\_addr.s\_addr = inet\_addr( IP );
- B. 設定 server.sin\_family = AF\_INET;
- C. 設定 server.sin\_port = htons( PORT );

### IV. Deployment Setting

- 1. 修改 /etc/rc.local 的檔案，加上我們要系統開機自動執行的指令，注意必須使用絕對路徑，以及必須要加&讓程式在背景執行。舉例來說，執行我們 client 端的程式碼如下：

/home/pi/ESL/Pi/MPU6050-right-user2/demo\_dmp &

### V. Qt GUI Programming

簡單來說，就是將 Qt 的信號和 QML 的信號連接再一起(用 Qt 的 Connect)，藉由手套上的 Rpi 傳送過來的訊號，來控制畫面中各種物體的狀態(用各種 QML 的 State 以及

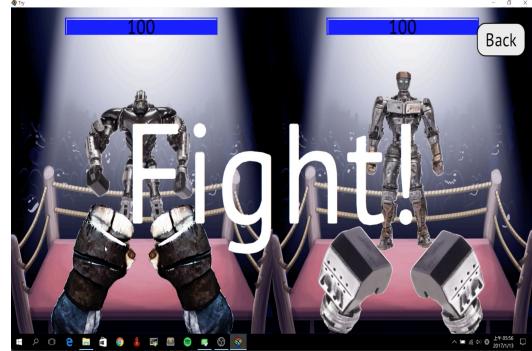
Transition)，並撥放相對應的影音(QML Audio or Video)。以下針對各部分稍微說明，QML 這邊的程式設計示意圖如下：

```
Main.qml: states for different scenes
- MenuScene
- SelectLevelScene
  | Signals: wait for connections
- CreditsScene
- GameScene: handle main game logics
  > OneArena
  > Countdown text
  > Loader: to load the characters file "Duo.qml"
    | Signals: Atom punch or Zeus punch or side chandes
    - Simple Blood Bar
    - Atom: severral states for different movements
    - Zeus
    - Atom_punches
    - Zeus_punches
- WinningScene: play winning video and sound
```

這個是改寫自 V-Play 官方的 MultipleSceneMultipleLevel 的例子，我用 main 去置換背景 (Scene) 該要是哪一個，大致上如同 demo 所看到的，開頭的菜單(menu)可以選擇要準備開始遊戲，或是看 credits，再來 SelectLevelScene 實際是等待玩家一二連線的背景，當它收到玩家連線的信號，就會進去 GameScene。GameScene 裡面有所有遊戲的邏輯，以及遊戲時的畫面，包含決鬥場、亞當、宙斯、血條，以及他們各自對應的拳頭，裡面的 Loader 會去抓 Duo.qml，然後打擊的信號會跟這個連接，所以當 MPU 感測到動作，從 Rpi 傳過來後，信號就會一路往下傳到 Atom 以及它的拳頭，去改變成相對應的狀態，然後在 gamescene 裡面判斷得分、扣血以及最後的勝利。

## 成果

本組完成了整套遊戲的 deployment 版本，使用者戴上手套並開機後即可開始遊戲，遊戲裝備以及介面設計如圖：



YouTube demo video: <https://www.youtube.com/watch?v=jziXI6g9NkU&feature=youtu.be>

github: <https://github.com/YuanTingHsieh/ESL>

成功實作”連接”、“斷線”的信號傳遞，以及兩個玩家的”左拳”、“右拳”、“防禦”、“左傾”、“右傾”，加上各種相對應的遊戲邏輯，還有使用者頁面的呈現，如同 demo 所顯示的那些，也成功的佈署程式以及手套。

## 討論

### I. C 與 C++的 Library linking:

本次硬體架設中，很大的一部份是整合 MPU 的程式以及各種通訊方式，我們使用的 SPP 程式是預先編譯好的，因此有一個 library “spp-sdp-register” 是沒有原碼的只有執行檔，但在當初在編譯的時候似乎是用 C++去編譯，因此我們必須用 extern 的方式來告訴 makefile 再執行這個 library 相關時要用不一樣的方式，如下：

```
26  extern "C" {  
27      #include "spp-sdp-register.h"  
28  }
```

### II. 藍芽：

之所以 pi 和 Qt 間不使用藍芽連接的原因是，Qt 5.7 版並不支援 mac 和 windows 的藍芽，然後一般的桌機也沒有配備藍芽，因此選擇用網路的方式。我們在實驗中有觀察到藍芽似乎又被障礙物屏蔽的問題（可能是被散射），因為我們一旦把 pi 裝入盒子中然後塞滿衛生紙，pi 之間藍芽的溝通似乎就會變得斷斷續續。

### III. Network Setting:

網路設定方面，基本上利用 pi 內建的網路 config 檔案，像是/etc/wpa 或是 etc/network 下面的設定檔，要建立自動的網路連線不是問題，也可以給無線網路 priority 的參數設定，幾個特別要注意的點是，這樣的設定檔格式都有一定限制，像是不能多空白等。另外，如果要加速 pi 的開機速度和省電，有幾種方法可以關掉 dhcp 的連線，可以透過 systemctl stop dhcpc.service 或是用 raspi-config 去設定。最後，很重要的是如果要讓 pi 自動執行的程式需要用到網路，那必須要確定 pi 已經拿到 network IP address 之後再執行，我們是利用 sleep 的方式等待。

#### IV. Qt:

其實 Qt、QML 這邊的文件還算是充裕，當然比起 C++, C 各種 function 的實作，官方文件相對上沒有這麼完善，就需要更多 trial and error。但總的來說，還是一個跨平台很方便的 GUI 程式設計語言，在 linux 或 windows 上，只要 git clone 下來，用 Qtcreator 打開，就可以編譯並執行能跑在自己機器上面的遊戲程式。現在程式的狀況，只有考慮兩位玩家只打一回合的情況，如果把狀態都弄好，應該可以延伸成可以好幾回合的對戰。另外，我們也有想到可以讓遊戲變得更好玩的地方，像是加上左右鉤拳，以及防守左右邊仍會被鉤拳攻擊等等。

## 參考資料

- Raspberry pi MPU6050:  
<http://blogger.gtwang.org/2014/12/raspberry-pi-mpu6050-six-axis-gyro-accelerometer-1.html>
- Bluetooth communication SPP server and client  
client: <http://ruten-proteus.blogspot.tw/2014/08/Bluetooth-Kit-tutorial-04-Linux-BlueZ-01-Client.html>  
server: <http://ruten-proteus.blogspot.tw/2014/08/Bluetooth-Kit-tutorial-04-Linux-BlueZ-02-Server.html>
- wpa\_config: <http://raspberrypi.stackexchange.com/questions/11631/how-to-setup-multiple-wifi-networks>
- V-play 官方文件: <https://v-play.net/doc/>
- Qt官方文件: <http://doc.qt.io/qt-5/>
- Windows deployment: <http://doc.qt.io/qt-5/windows-deployment.html>

## 附錄

參考的 open source 應該就是 Qt, QML ,Vplay，這邊寫上之後要再使用的注意事項好了。

1. 網路的設定需要調整: 之前 demo 是都連我們自己手機的基地台，要記得兩隻右手，跟你要跑 qml 程式的機器，需要連到同樣的無線網路。
2. 要先開右手，右手連接上程式之後，在開啟左手的 Rpi，左右連接好之後，畫面 wait for connection 上面相對應的 user 就會變亮。
3. 程式所需要的畫面以及音樂檔案，因為有版權，並沒有上傳到 github，要再私下拿。