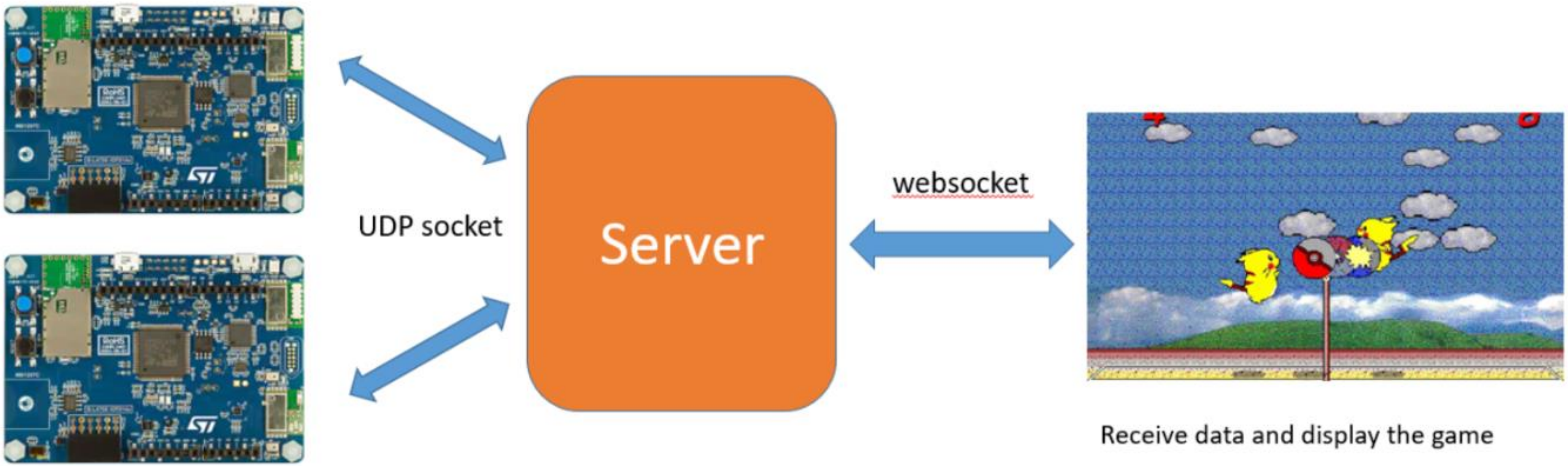
Pikachu-Volleyball
進度報告

陳志偉 電機四 B06901126



UDP socket

Server

websocket

Receive data and display the game

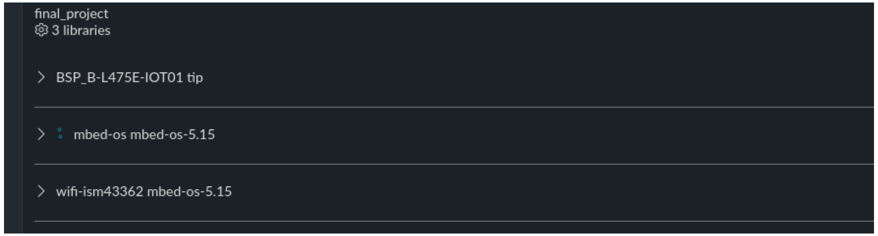
Detect and send player movement

環境設定

- Completely in linux
- WEB & Socket server in VM or WSL
  https://github.com/microsoft/WSL/issues/4150(Setting bridge network)

SMT32 & Mbed-os

- Mbed-os & 相關libraries版本確定



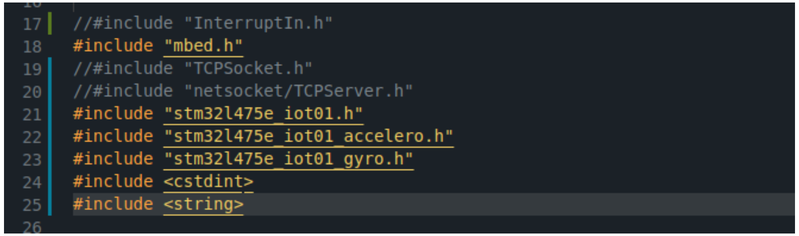
final_project
3 libraries

BSP_B-L475E-IOT01 tip

mbed-os mbed-os-5.15

wifi-ism43362 mbed-os-5.15

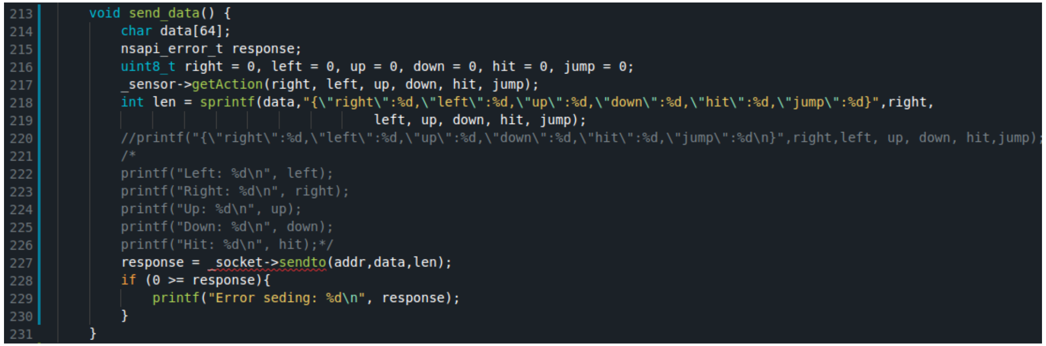
- 去除重複的library



```
//#include "InterruptIn.h"
#include "mbed.h"
//#include "TCPSocket.h"
//#include "netsocket/TCPServer.h"
#include "stm32l475e_iot01.h"
#include "stm32l475e_iot01_accelero.h"
#include "stm32l475e_iot01_gyro.h"
#include <cstdint>
#include <string>
```

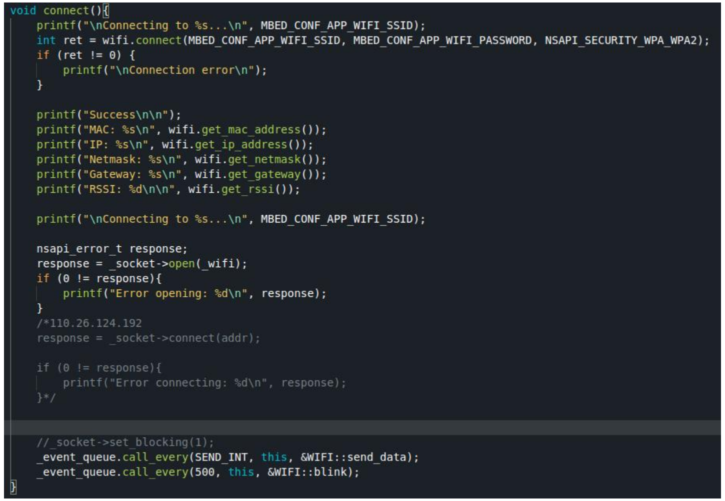
遊玩體驗 & 遇到的問題

- 控制有明顯的延遲感
  - ➔ 解決方法：註解SMT32用以檢查data的print()，print()所用的時間超過sampling的時間。



```
void send_data() {
    char data[64];
    nsapi_error_t response;
    uint8_t right = 0, left = 0, up = 0, down = 0, hit = 0, jump = 0;
    _sensor->getAction(right, left, up, down, hit, jump);
    int len = sprintf(data,"{\"right\":%d,\"left\":%d,\"up\":%d,\"down\":%d,\"hit\":%d,\"jump\":%d}",right,
                                left, up, down, hit, jump);
    //printf("{\"right\":%d,\"left\":%d,\"up\":%d,\"down\":%d,\"hit\":%d,\"jump\":%d\n}",right,left, up, down, hit,jump);
    /*
    printf("Left: %d\n", left);
    printf("Right: %d\n", right);
    printf("Up: %d\n", up);
    printf("Down: %d\n", down);
    printf("Hit: %d\n", hit);*/
    response = _socket->sendto(addr,data,len);
    if (0 >= response){
        printf("Error seding: %d\n", response);
    }
}
```

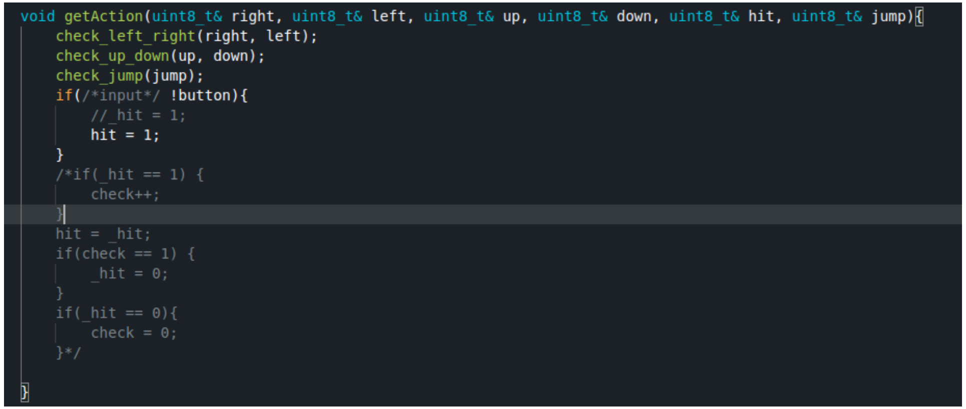
- workflow沒有規劃較明確的規劃
  - ➔ Wifi連接情況...
  - ➔ UDPSocket發送失敗...
  - ➔ 遊戲內連接不到Socket Server...



```
void connect(){
    printf("\nConnecting to %s...\n", MBED_CONF_APP_WIFI_SSID);
    int ret = wifi.connect(MBED_CONF_APP_WIFI_SSID, MBED_CONF_APP_WIFI_PASSWORD, NSAPI_SECURITY_WPA_WPA2);
    if (ret != 0) {
        printf("\nConnection error\n");
    }

    printf("Success\n\n");
    printf("MAC: %s\n", wifi.get_mac_address());
    printf("IP: %s\n", wifi.get_ip_address());
    printf("Netmask: %s\n", wifi.get_netmask());
    printf("Gateway: %s\n", wifi.get_gateway());
    printf("RSSI: %d\n\n", wifi.get_rssi());

    printf("\nConnecting to %s...\n", MBED_CONF_APP_WIFI_SSID);

    nsapi_error_t response;
    response = _socket->open(_wifi);
    if (0 != response){
        printf("Error opening: %d\n", response);
    }
    /*110.26.124.192
    response = _socket->connect(addr);

    if (0 != response){
        printf("Error connecting: %d\n", response);
    }*/


    //_socket->set_blocking(1);
    _event_queue.call_every(SEND_INT, this, &WIFI::send_data);
    _event_queue.call_every(500, this, &WIFI::blink);
}
```

遊玩體驗 & 遇到的問題

- 操控的流暢及
  - 用（USER_BUTTON）取代外接按鍵，原本的（USER_BUTTON）為Reset功能
  - 修改SMT32上對殺球（USER_BUTTON）按鍵的判斷



```
void getAction(uint8_t& right, uint8_t& left, uint8_t& up, uint8_t& down, uint8_t& hit, uint8_t& jump){
    check_left_right(right, left);
    check_up_down(up, down);
    check_jump(jump);
    if(/*input*/ !button){
        //_hit = 1;
        hit = 1;
    }
    /*if(_hit == 1) {
        check++;
    }
    hit = _hit;
    if(check == 1) {
        _hit = 0;
    }
    if(_hit == 0){
        check = 0;
    }*/

}
```

  - 操作 殺球+方向 實現極困難
  - 上(UP)和跳(JUMP)在操作上幾乎沒有區別
    *上下左右(加速度計)
    * 跳（陀螺儀）

待完成

- 明確的workflow
- 區分移動及殺球時的方向
- 方向+殺球 的判斷方式
- 把server與web server結合
- …