一、動機

我們組員喜愛「Temple Run」這個手機遊戲，而嵌入式系統實驗的課程中有cover到motion detection這個主題，於是認為這種類型的遊戲若能以實際的手臂揮動，而非手指於觸控螢幕上的滑動來操作，會帶來不同的體驗。在製作的過程中，也能對應用於各遊戲主機的motion detection技術，如Nintendo Switch的joy-con，有著初步的認識。

二、作法
STM32
共分成兩個 Free RTOS的Task，分別為StartTaskBLE以及StartTaskACC
StartTaskBLE：
處理BLE，執行MX_BlueNRG_MS_Process();
StartTaskACC：
1.讀取加速度計的值並通過moving average的FIR filter
FIR filter的參數設定：

```
float32_t firCoeffs[NUM_TAPS] = {
    0.2f, 0.2f, 0.2f, 0.2f, 0.2f
};
```

每個for loop的一開始，執行讀值並過filter：

```
BSP_ACCELERO_AccGetXYZ(pDataXYZ);

    float32_t xout, yout, zout;


    float32_t xin = (float32_t)pDataXYZ[0];
    float32_t yin = (float32_t)pDataXYZ[1];
    float32_t zin = (float32_t)pDataXYZ[2];


    arm_fir_f32(&Sx, &xin, &xout, BLOCK_SIZE);
    arm_fir_f32(&Sy, &yin, &yout, BLOCK_SIZE);
    arm_fir_f32(&Sz, &zin, &zout, BLOCK_SIZE);


    int16_t x = (int16_t)xout;
    int16_t y = (int16_t)yout;
    int16_t z = (int16_t)zout;
    z -= g;
```

2.上下左右的motion detection，以finite state machine實做

```
switch (x_state){
case 0: //no move
    x_max = 0;
    x_min = 0;
    if (x > x_small_threshold){
            x_max = x;
            x_state = 1;
    }
    if (x < -x_small_threshold){
            x_min = x;
            x_state = -1;
    }
    break;
case 1:
    if (x > x_max){
            x_max = x;
    }
    if (x < x_small_threshold){
            x_state = 0;
            if (x_max > x_threshold){
                    printf("right: %d\n",x_max);
                    char msgr[20];
                            sprintf(msgr, "r%d",x_max);
                            Send_Action(msgr,strlen(msgr));
                            cooldown = true;
                            cd_start_time = HAL_GetTick();
            }

            x_min = x;
    }
    break;
```

以x方向為例，x_state == 0代表靜止，一旦x軸的加速度計值達到正負small_threshold，則會把x_state切為正負1(+1代表有向右的動作，-1則是向左的動作)，直到加速度計值回到small_threshold內為止再把x_state切成0，並進入cooldown。於cooldown期間，for loop在上下左右motion detection的部分開始前就會continue掉。於x_state == 1期間，會記錄讀到的最大

（小）值，若有達到threshold，則會在x_state切回0時傳送含有方向以及數值的message給python端。z軸的實作邏輯與x軸相同。

3.shake motion的偵測，以finite state machine實作

```
switch (shake_state){
case 0:
    shake_count = 0;
    if (x > 500){
            start_time = HAL_GetTick();
            shake_state = 1;
            shake_count++;
    }
    if (x < -500){
            start_time = HAL_GetTick();
            shake_state = 2;
            shake_count++;
    }
    start_time = sHAL_GetTick();
    printf("x=%d\n", x);
    break;
case 1:
    if (x < -500){
                    start_time = HAL_GetTick();
                    shake_state = 2;
                    shake_count++;
                    if (shake_count > 10){
                            printf("shake!\n");
                            Send_Action("shake", 5);
                            shake_state = 0;
                    }
            }else if (HAL_GetTick() - start_time > 3000){
                    printf("no shake but = %d\n", shake_count);
                    shake_state = 0;
            }
            break;
case 2:
    if (x > 500){
                    start_time = HAL_GetTick();
                    shake_state = 1;
                    shake_count++;
                    if (shake_count > 10){
                            printf("shake!\n");
                            Send_Action("shake", 5);
                            shake_state = 0;
                    }
            }else if (HAL_GetTick() - start_time > 3000){
                    printf("no shake but = %d\n", shake_count);
                    shake_state = 0;
            }
            break;
}
```

以x軸的加速度計值偵測搖動，shake state == 0代表靜止狀態，一旦讀值的大小超過500，則切換state，並記錄開始偵測此次shake的時間。只要加速度計的值從+500以上跑到-500以下，就會從state 1切換的state 2，並計搖動一次，反之亦然。3秒內若shake的次數達10以上，則立即傳送shake的message給python端，否則會將state切回0並且reset紀錄到的shake次數。
2.Python
主要功能模組
Player：負責移動、跳躍、滑行、左右移動（含強度）。
Enemy、陷阱 TrapUFOManager、TSMC 加分道具：自動移動、碰撞判斷。
HUD：顯示分數、生命、校徽等資訊。
InkOverlayEffect：遮蔽畫面，增加遊戲難度。
ScreenShake：碰撞時畫面抖動效果。
BLE 監聽：負責接收來自 STM32 的動作指令。

主遊戲迴圈
每幀更新分數、計時器、速度倍率、墨魚汁（遮蔽）效果。
根據 BLE queue 內容，解析指令並控制玩家動作：

```
# BLE 控制輸入處理（新版含方向+強度）
try:
    while not ble_queue.empty():
        cmd = ble_queue.get_nowait()
        if cmd.startswith("u"):
            player.jump()
        elif cmd.startswith("d"):
            player.slide()
        elif cmd.startswith("r") or cmd.startswith("l"):
            direction = cmd[0]
            try:
                strength = int(cmd[1:])
                player.move_with_strength(direction, strength)
            except ValueError:
                pass  # 忽略無效格式
        elif cmd == "shake":
            ink_overlay.receive_shake()
except queue.Empty:
    pass
```

up：跳躍
down：滑行
r{數值}/l{數值}：根據強度左右移動
shake：觸發墨魚汁遮蔽效果
處理事件（支援鍵盤與 BLE 雙重控制）。

更新所有遊戲物件狀態(玩家、敵人、陷阱、道具等)。

```
    # 判斷小怪碰撞（需要跳或滑）
    if enemy.active and player_rect.colliderect(enemy.get_rect()):
        if enemy.height == 'down' and not player.is_jumping:
            lives -= 1
            enemy.active = False
            shake.start()
        if enemy.height == 'up' and not player.is_sliding:
            lives -= 1
            enemy.active = False
            shake.start()

    # 判斷是否踩到陷阱
    if trap_manager.get_collided(player_rect, player.is_jumping, player.i
        lives -= 1
        shake.start()
```

檢查碰撞、加分、扣命、遊戲結束等邏輯。
畫出所有遊戲元素與 HUD，並顯示特殊狀態(如無敵、雙倍分數)。

```
# Show TSMC invincible/double score timer hints (English)
if tsmc.is_invincible():
    inv_sec = tsmc.invincible_timer // 30
    inv_text = font.render(f"Invincible: {inv_sec}s", True, (255, 255, 0))
    game_surface.blit(inv_text, (10, 40))

if double_score_timer > 0:
    bonus_sec = double_score_timer // 30
    bonus_text = font.render(
        f"Double Score: {bonus_sec}s", True, (0, 255, 0))
    game_surface.blit(bonus_text, (10, 70))
```

三、成果
https://www.youtube.com/shorts/DlvLIKFMOVw
https://www.youtube.com/shorts/55i3FqFDbEo

(這裡放影片)
四、參考文獻或資料
台大二活圖片：
https://www.google.com/url?sa=i&url=https%3A%2F%2Fzh.wikipedia.org%2Fwiki%2FFile%3A%E9%9B%BB%E6%A9%9F%E4%BA%8C%E9%A4%A8.JPG&psig=AOvVaw1ysHzmvgM2ZB6P3H2sqoj5&ust=1749721782780000&source=images&cd=vfe&opi=89978449&ved=0CAMQjB1qFwoTCPjPityL6Y0DFQAAAAAdAAAAABAE
台大校徽：

https://www.ntu.edu.tw/images/about/emblem_2.png
椰林大道：
https://www.flickr.com/photos/cs_wu/38525652325

敵人資料庫：
www.flaticon.com

人物圖片：

https://www.st.com.cn/bin/ecommerce/api/image.PF270120.en.feature-description-include-personalized-no-cpn-medium.jpg

Making a 2d Version of Temple run in Pygame :
https://www.reddit.com/r/pygame/comments/b64cp1/making_a_2d_version_of_temple_run_in_pygame/