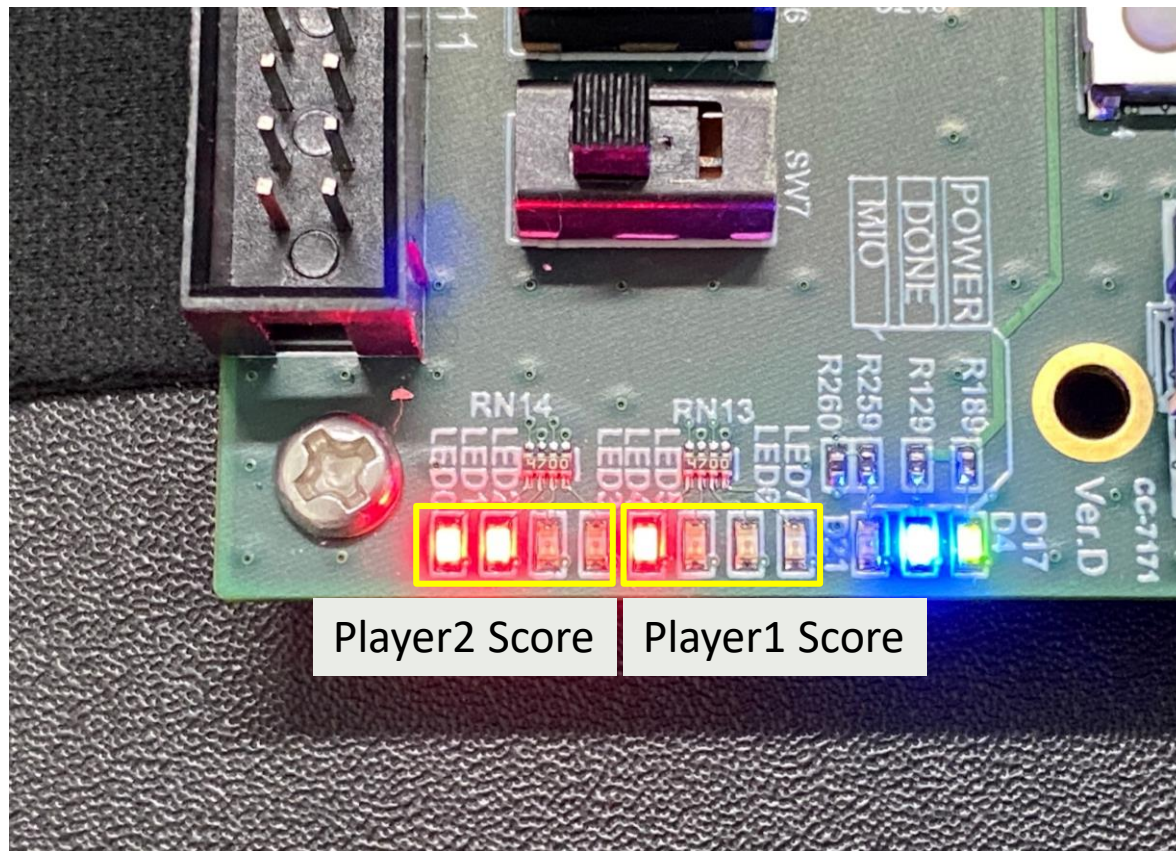
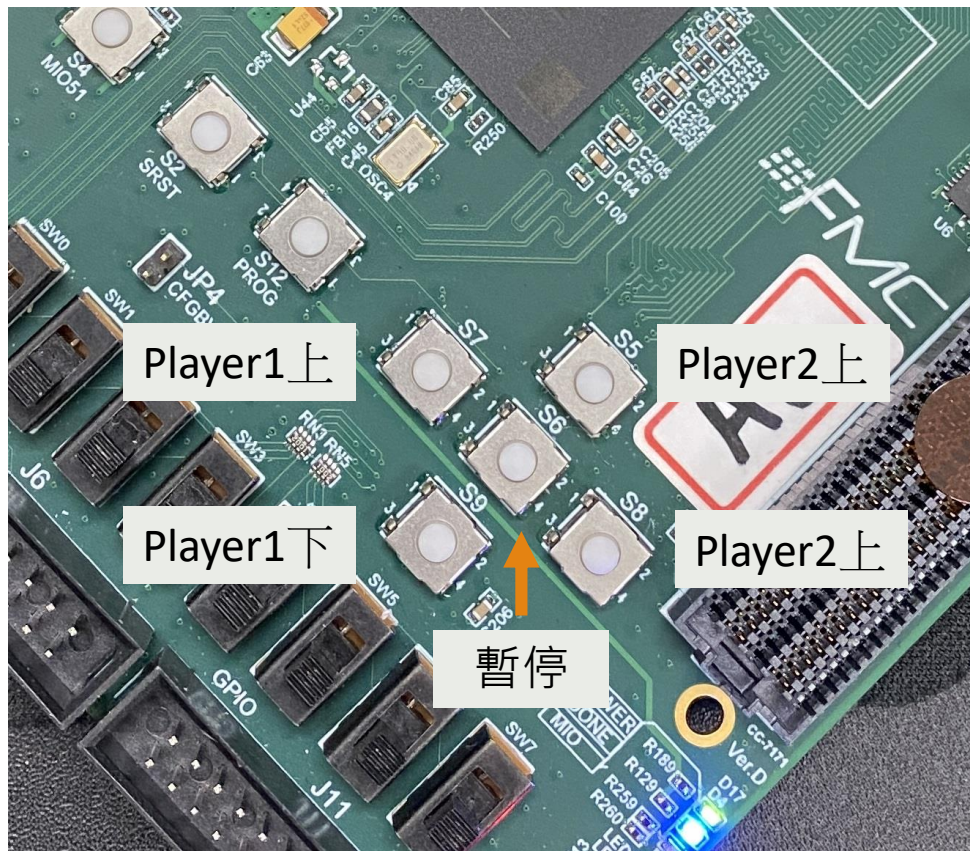


Pong Game

功能



功能



功能

Ball 依照物理現象移動(入射角等於反射角)

球的速度會隨著時間越來越快

玩家可透過按鈕移動 Bar

移動 Bar 進行擊球動作

球移動超過對方 Bar 則得分，並用 LED 顯示分數

其中一方超過 4 分則結束遊戲

遊戲暫停功能

效能

UART 傳輸速率 115200 bps

每次傳送的資料有2個Player的分數、2個Bar的Y座標、球的(X, Y)

每100毫秒傳送一次資料給電腦

介面

PL-PS：AXI

PS-PC：USB_UART

Input：GPIO按鈕

Output：遊戲畫面、LED

限制

硬體：

Zynq-7000

遊戲主程式：

在 Soc 上執行

遊戲畫面：

在 PC 上顯示，解析度 50*30

軟體：

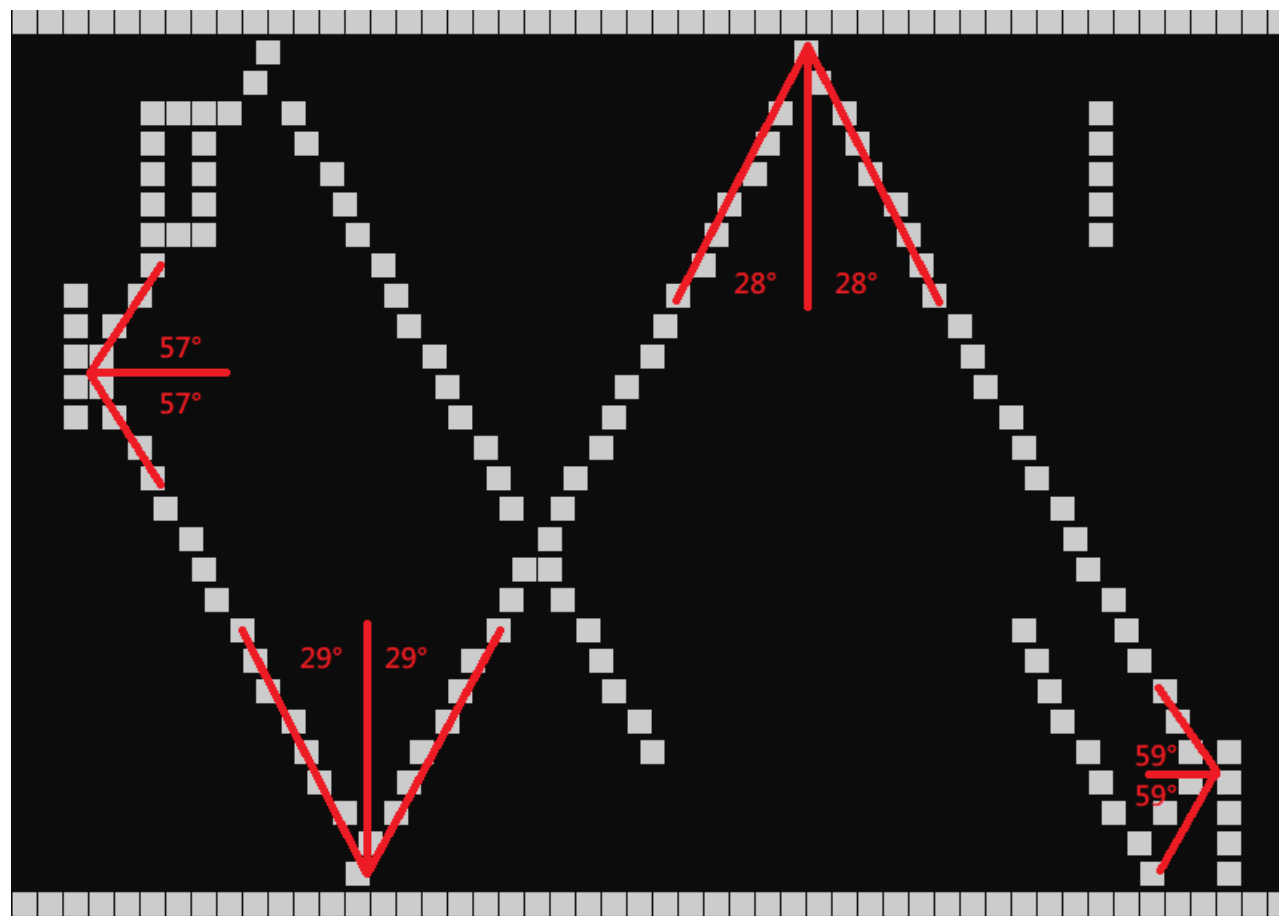
Vivado 2018.3

Vivado 2018.3 SDK

驗收

1. 球依照物理現象移動

->截取遊戲中，球碰到牆壁還有Bar前後的畫面，並用量角器測量，計算球是否依照入射角等於反射角的規律移動



驗收

2. 球的速度會隨著時間越來越快

->錄製螢幕畫面，計算球移動**10**格所花費的時間；每擊球**5**次，球移動**10**格花費的時間會越來越少

驗收結果：**2.mp4**

擊球次數**0~4**次時，球移動**10**格花費**1666**毫秒，擊球次數**5**次時，球移動**10**格花費**883**毫秒

驗收

3. 玩家可透過按鈕移動Bar
->按下按鈕觀察畫面Bar是否移動
4. 移動Bar進行擊球動作
->移動Bar觀察是否擊球有反彈

驗收結果：3-4.mov

驗收

5. 其中一方超過4分則結束遊戲

->其中一方得4分後，觀察遊戲是否重新開始

(檢查player1的分數為0分、檢查player2的分數為0分、

檢查球是否回歸原始速度(參照速度驗收)、

檢查ball的x座標為30、檢查ball的y座標為10)

驗收結果：5.mov

驗收

6. 遊戲暫停功能

->按下按鈕後觀察球是否停止移動，Player1、Player2的Bar都不能移動，球也會在停止狀態

驗收結果：6.mov

7. LED燈號

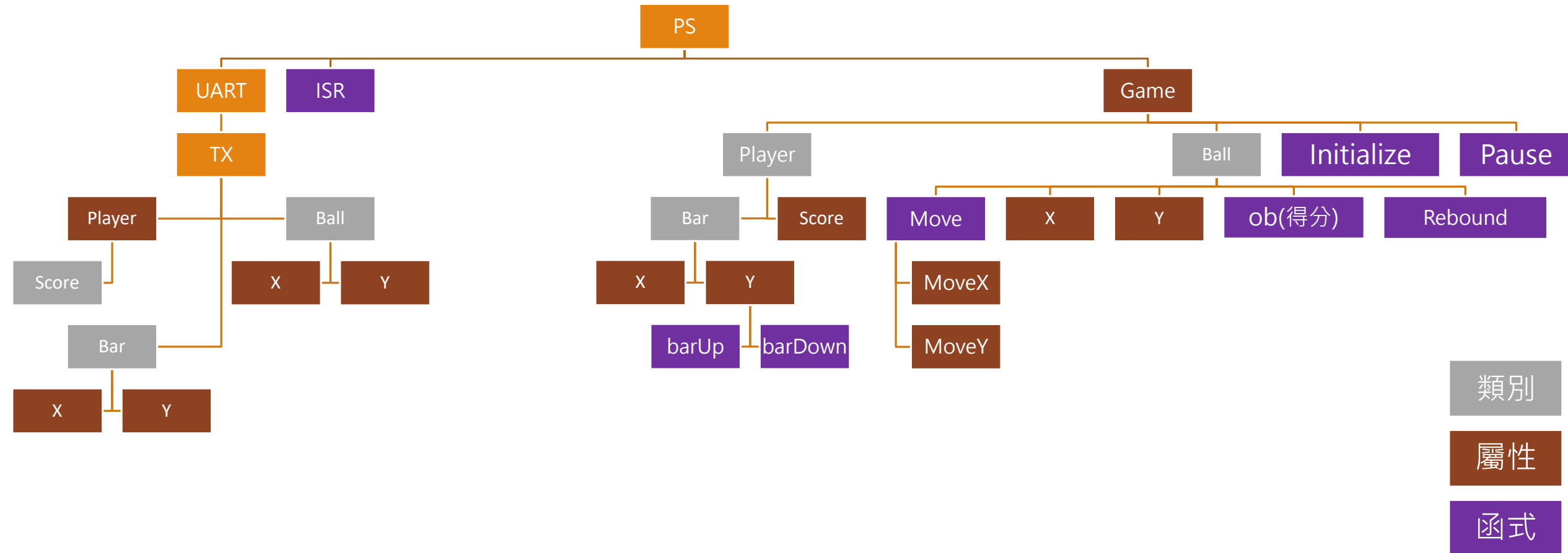
->LED顯示Player的分數

8. 球移動超過對方Bar則得分

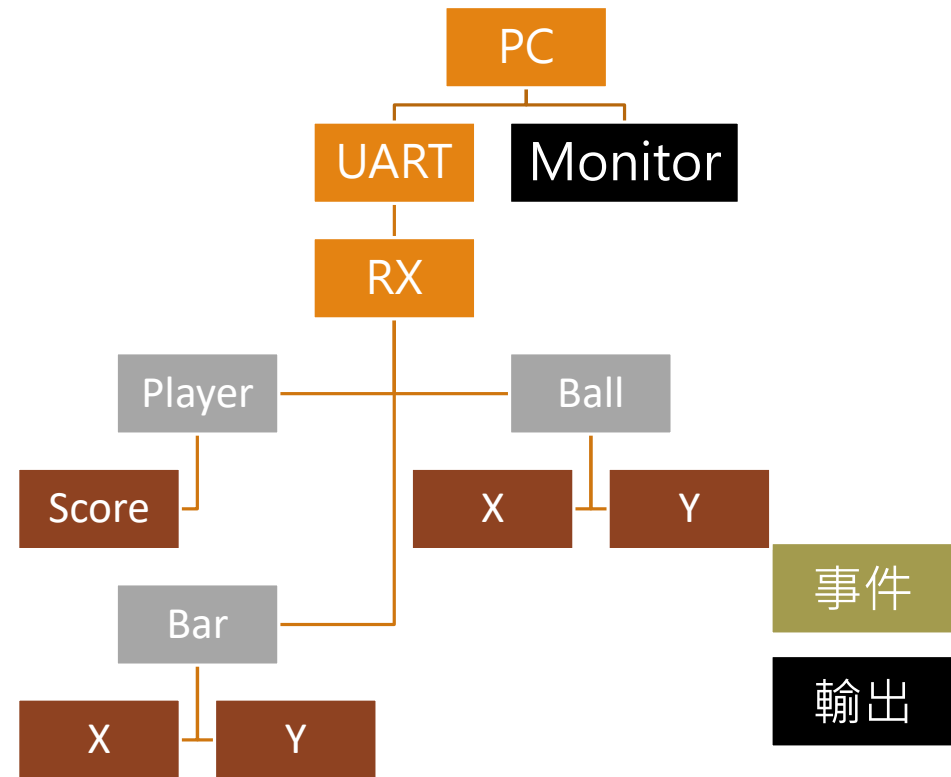
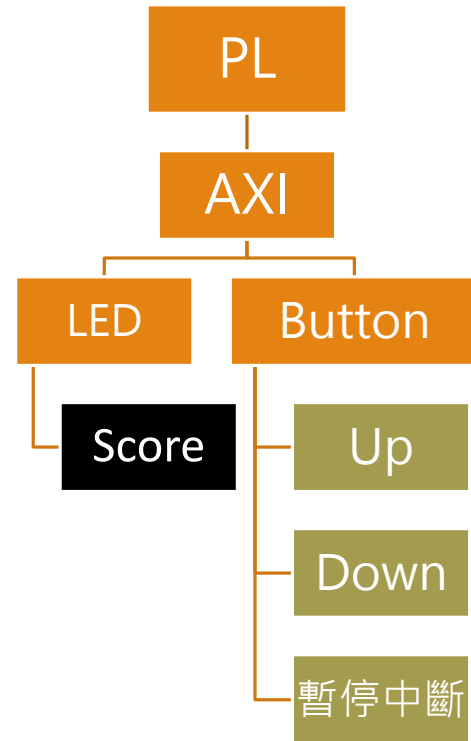
->Player1故意不接球，Player2會加一分；反之Player2故意不接球，Player1會加一分

驗收結果：7-8.mov

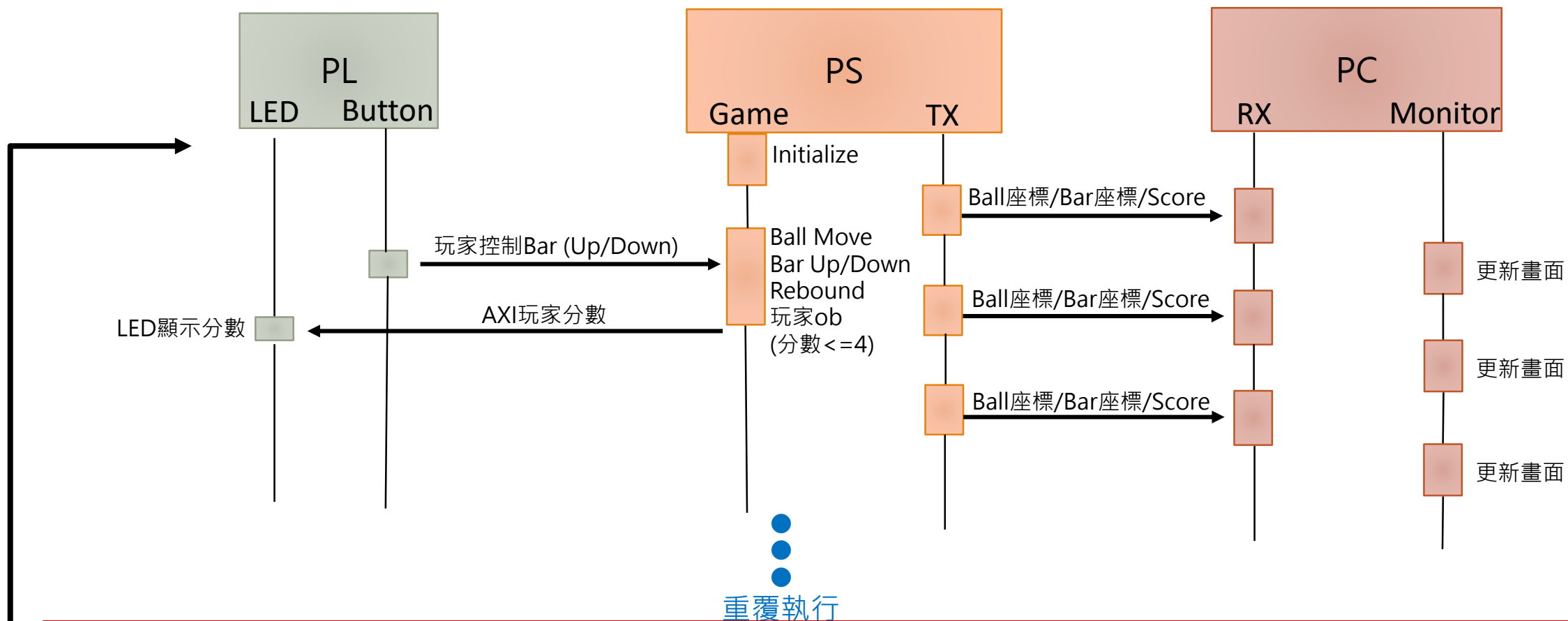
Breakdown



Breakdown

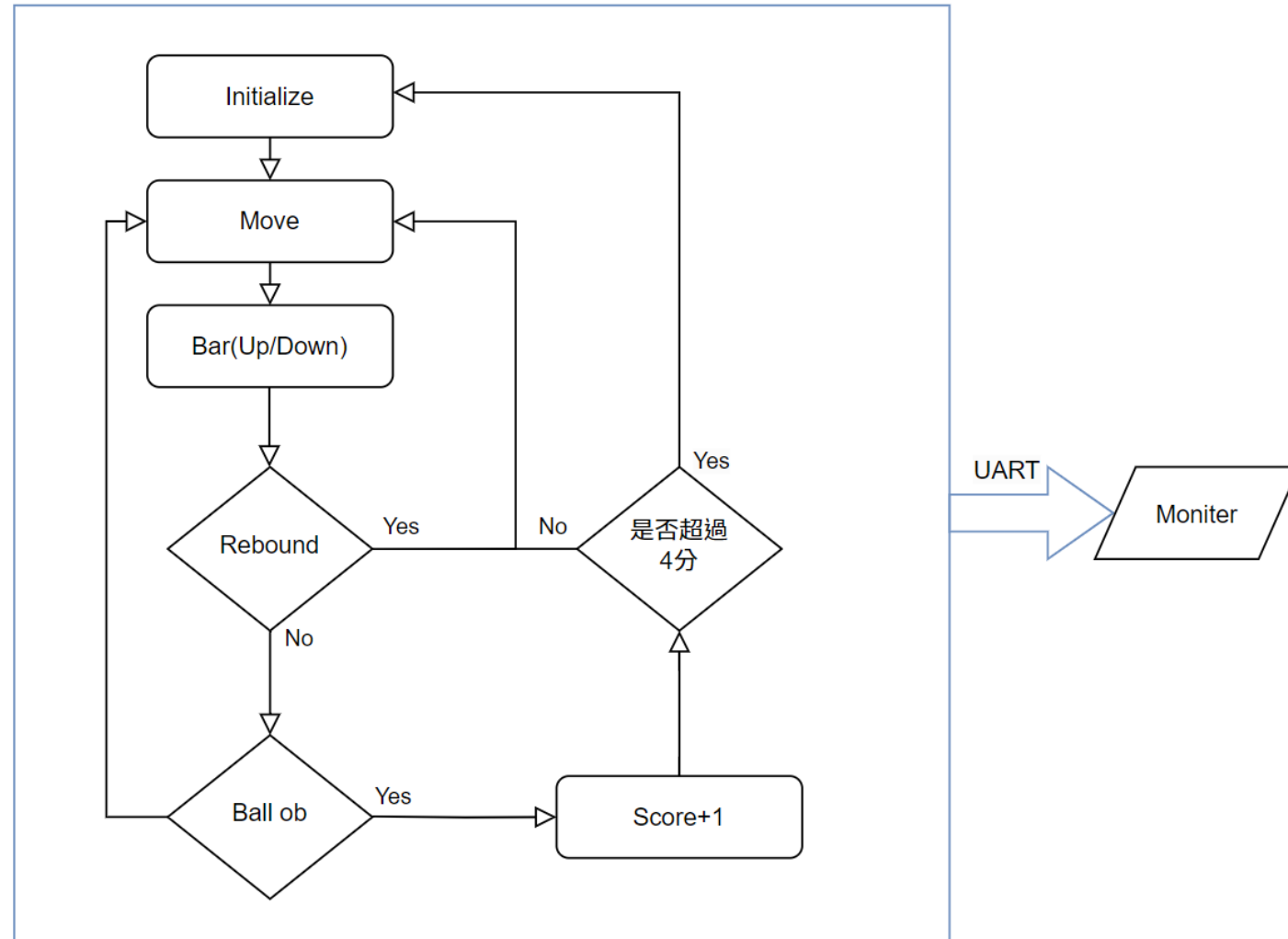


MSC

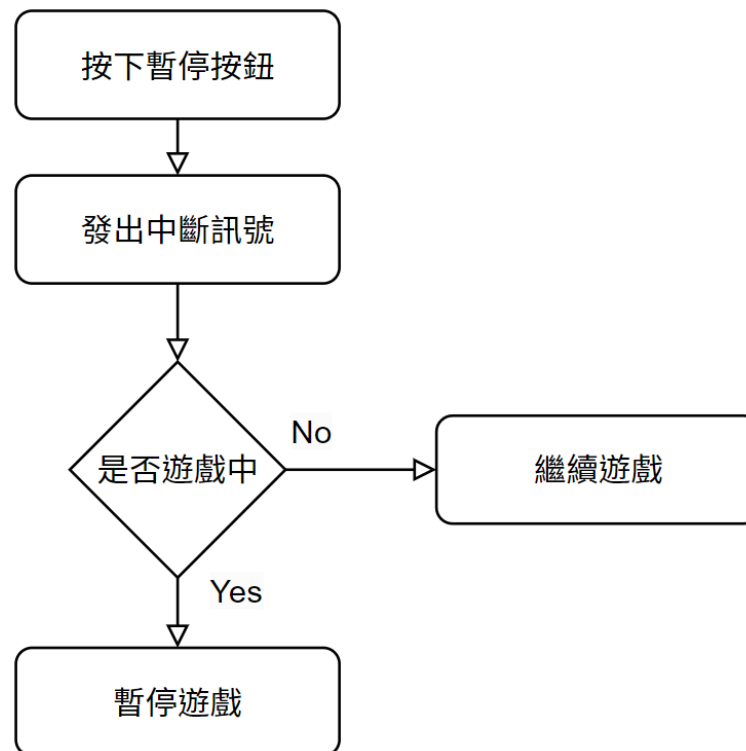


- 分數>4分
- 回到Initialize

系統流程圖



系統流程圖



API : Player

void barUp()

輸入：bar的y座標

輸出：y

方法：讓Player的Bar往上一格(y--)

參數：無

void barDown()

輸入：bar的y座標

輸出：y

方法：讓Player的Bar往下一格(y++)

參數：無

```
void barUp() {  
    if (getBarYtop() > 1) {  
        bar.up();  
    }  
}
```

```
void barDown() {  
    if (getBarYbottom() < HEIGHT - 2) {  
        bar.down();  
    }  
}
```

```
void up() {  
    y--;  
}  
void down() {  
    y++;  
}
```

API : Ball

void move()

輸入：Ball的x、y、移動方向

輸出：x, y

方法：讓Ball移動(x += moveX; y += moveY;)

參數：無

void ob()

輸入：Ball的x、WIDTH

輸出：player2.score++;

player1.score++;

x

方法：對方的分數加一分，ball的x座標回到原點

參數：無

```
void move() {
```

```
    x += moveX;
```

```
    y += moveY;
```

```
    rebound();
```

```
    ob();
```

```
}
```

```
void ob() {
```

```
    if (x > WIDTH * 2 - 3) { // right ob
```

```
        player1.goal();
```

```
        x = WIDTH;
```

```
    }
```

```
    if (x <= 2) { // left ob
```

```
        player2.goal();
```

```
        x = WIDTH;
```

```
    }
```

```
    if (player1.getScore() == 5 || player2.getScore() == 5) {
```

```
        initialize();
```

```
    }
```

```
}
```

API : Ball

void rebound()

輸入：PLAYER_1_BAR_X、PLAYER_2_BAR_X、Player的y座標、HEIGHT

輸出：moveX, moveY

方法：讓Ball反彈(ball碰到top或bottom: moveY *= -1; 碰到bar: moveX *= -1;) 並判斷擊球次數，如果達到5次就讓球加速(moveX += 0.3; moveY += 0.3;)

參數：無

```
void rebound() {
    if (y > HEIGHT - 2) { // bottom
        moveY *= -1;
    }
    if (y <= 2) { // top
        moveY *= -1;
    }
    if ((x >= PLAYER_2_BAR_X - 2) && (y > player2.getBarYtop() && y <= player2.getBarYbottom())) { // right bar
        moveX *= -1;
        count++;
    }
    if ((x <= PLAYER_1_BAR_X + 2) && (y > player1.getBarYtop() && y <= player1.getBarYbottom())) { // left bar
        moveX *= -1;
        count++;
    }
    if (count == 5 && ((moveX + 0.3) < 1) && ((moveY + 0.3) < 1)) {
        moveX += 0.3;
        moveY += 0.3;
        count = 0;
    }
}
```

Global API

`void initialize()`

輸入：無

輸出：player1; player2.score;
ball.moveX; ball.moveY;
ball.x; ball.y;
count;

方法：設定player1的分數為0分、設定player2的分數為0分
設定ball的moveX為-0.3、設定ball的moveY為0.4、
設定ball的x座標為30、設定ball的y座標為10
設定count為0

參數：無

`void pause(int Btn)`

輸入：按鈕

輸出：暫停中斷

方法：暫停/繼續遊戲

參數：Btn

```
void initialize() {  
    player1.setScore(0);  
    player2.setScore(0);  
    ball.setMoveX(-0.4);  
    ball.setMoveY(0.3);  
    ball.setX(30);  
    ball.setY(10);  
    count = 0;  
    Led_num = 0b00000000;  
    XGpio_DiscreteWrite(&LED, 1, Led_num);  
}
```

```
void pause(int Btn) {  
    if(stop == 1 && Btn == 1){  
        stop = 0;  
    } else if (Btn == 1){  
        stop++;  
    }  
}
```

中斷服務程式

void Intr_Handler()

輸入： stop、player1與2的BAR Y軸、BTN
BTN_INT XGPIO_IR_CH1_MASK

輸出： player1.bar.y、 player2.bar.y;

方法： 依照stop和輸入的Button決定
1. bar y移動
2. 遊戲暫停(pause)

參數： 無

```
void Intr_Handler()
{
    XGpio_InterruptDisable(&BTN, BTN_INT); //關閉中斷
    XGpio_InterruptClear(&BTN, BTN_INT); //中斷標誌清除
    Btn_num = XGpio_DiscreteRead(&BTN, 1);
    for (int i = 0; i < 5; ++i) {
        Btn[i] = Btn_num & (1 << i);
        //printf("Button %d is pressed. ,%d \n", Btn[i],i)
    }
    if(Btn[0] == 0 && stop == 0){
        if (Btn[2] == 4 && Btn[1] == 0) {
            if(Btn_p1d == 0){
                //print("p1 moves up.");
                player1.barUp();
                Btn_p1d = 1;
            }
        } else if (Btn[2] == 0 && Btn[1] == 2) {
            if(Btn_p1d == 0){
                //print("p1 moves down.");
                player1.barDown();
                Btn_p1d = 1;
            }
        } else {
            Btn_p1d = 0;
        }
        if (Btn[4] == 16 && Btn[3] == 0 ) {
            if(Btn_p2d == 0){
                //print("p2 moves up.");
                player2.barUp();
                Btn_p2d = 1;
            }
        } else if (Btn[4] == 0 && Btn[3] == 8) {
            if(Btn_p2d == 0){
                //print("p2 moves down.");
                player2.barDown();
                Btn_p2d = 1;
            }
        } else {
            Btn_p2d = 0;
        }
    } else {
        pause(Btn[0]);
    }
    XGpio_InterruptEnable(&BTN, BTN_INT); //開啟特定GPIO通道中斷
}
```