

EECS 2070 02 Digital Design Labs 2020  
Final Project Report  
Group54

學號：108062233 108062227 姓名：彭淵 鄭守維

## 1. 實作過程

這次設計的這個投籃機我們主要將它分成五個 state  
分別是 ready、chse\_second、cnt\_down、play、endgame

```
parameter [2:0] ready = 3'b000; //state declaration
parameter [2:0] chse_second = 3'b001;
parameter [2:0] cnt_down = 3'b010;
parameter [2:0] play = 3'b011;
parameter [2:0] endgame = 3'b100;

reg [5:0] num1, num2, num3, num4, num, num1_next, num2_next, num3_next, num4_next, keyin_num; // other declaration
reg [3:0] cnt, endgame_cnt, cntdown;
wire [2:0] div, rem; //divider reminder
reg [2:0] flag_clk;
reg [2:0] state, state_next;
reg standard, std;

/*////////////////////////////////////*/
```

Ready：在遊戲開始前播放閒置時間的動畫，按下 enter 鍵進入下一個 state

Chse\_second：使用鍵盤的 123 來選擇要玩的秒數

Cnt\_down：一個投籃開始前有 321 倒數的動畫

Play：玩投籃且有計分，計分的方法是將超聲波傳感器 HC-SR04 安裝在籃框下方，只要球進籃框通過傳感器，就可以藉由距離的變化判斷進球，以此來累加分數，而在倒數期間也可以使用 switch 來暫停遊戲

Endgame：最後計分結束的結束畫面

在後面寫了一個 FSM 來控制 state 的改變以及每個 state 時 7-segment 的表現  
像是在 ready state 按下 enter 鍵就會進入下一個 state

```
//state control
always @(posedge new_clk_26 or posedge new_rst) begin
    if(new_rst) begin
        state <= ready;
        num1 <= 6'b000000;
        num2 <= 6'b000000;
        num3 <= 6'b000000;
        num4 <= 6'b000000;
    end
    else begin
        state <= state_next;
        num1 <= num1_next;
        num2 <= num2_next;
        num3 <= num3_next;
        num4 <= num4_next;
    end
end

always @(*) begin
    case(state)
        ready: begin
            if(new_enter) begin
                state_next = chse_second; //if press the enter bottom, change to the next state
            end
            else begin
                state_next = ready;
            end
            num1_next = 6'b000001;
            num2_next = 6'b000000;
            num3_next = 6'b000011;
            num4_next = 6'b000000;
        end
        chse_second: begin
            if(start_play) begin // if press the start_play bottom, change to the next state
                state_next = cnt_down;
                num1_next = 6'b000000;
                num2_next = 6'b000011;
            end
        end
    endcase
end
```

這邊則是設計了各個動畫的計時器 clk

```
assign div = cnt / 4;
assign rem = cnt % 4;

always @(posedge new_clk_13) begin
    flag_clk <= flag_clk + 1;
end

always @(posedge new_clk_24) begin // for state ready anime
    if(cnt >= 4'd15) begin // cnt = 0~3 => down arrow from digit 1 to digit 4
        cnt <= 4'd0; // cnt = 4~7 => up arrow from digit 4 to digit 1
    end // cnt = 8 | 9 | 12 | 13 => digit 1~4 all light
    else begin // cnt = 10 | 11 | 14 | 15 => digit 1~4 all dark
        cnt <= cnt + 1;
    end
end

always @(posedge new_clk_26) begin // for state endgame anime
    if(state==endgame) begin // endgame_cnt = 0 | 2 | 4 | 6 => digit 1~4 all dark
        if(endgame_cnt>=4'd9) begin // endgame_cnt = 1 | 3 | 5 | 7 => digit 1~4 all light
            endgame_cnt <= 4'd9;
        end
        else begin
            endgame_cnt <= endgame_cnt + 1;
        end
    end
    else begin
        endgame_cnt <= 4'd0;
    end
end
```

在後面控制 7-segment 的地方用來控制每個時刻顯示不同的動畫

```
//7-segment
always @(*) begin
    case(state)
        ready: begin
            case(flag_clk)
                2'b01: begin
                    if(cnt<4'd8) begin
                        if(div==0) begin
                            num = 6'b010110;
                        end
                        else begin
                            num = 6'b010111;
                        end

                        if(rem==0) begin
                            if(div==0) digit = 4'b0111;
                            else digit = 4'b1110;
                        end
                        else begin
                            digit = 4'b1111;
                        end
                    end
                    else begin
                        num = 6'b001000;
                        if(cnt==4'd8 || cnt==4'd9 || cnt==4'd12 || cnt==4'd13) begin
                            digit = 4'b0111;
                        end
                        else begin
                            digit = 4'b1111;
                        end
                    end
                end
                2'b10: begin
                    if(cnt<4'd8) begin
                        if(div==0) begin
                            num = 6'b010110;
                        end
                    end
                end
            end
        end
    end
end
```

這個 always 是在控制 chse\_second state 時，使用鍵盤的 1、2、3 來選則秒數

```
always @(posedge clk or posedge new_rst) begin // for state chse_second
    if(new_rst) begin // In chse_second state, if 1 or 2 or 3 key in ke
        keyin_num <= 6'b000001; // have same value
    end
    else begin
        if(state==chse_second) begin
            if (been_ready && key_down[last_change] == 1'b1) begin
                if(key_num!=4'b1111) begin
                    if(key_num==4'b0001) begin
                        keyin_num <= 6'b000001;
                    end
                    else if(key_num==4'b0010) begin
                        keyin_num <= 6'b000010;
                    end
                    else begin
                        keyin_num <= 6'b000011;
                    end
                end
            end
        end
        else begin
            keyin_num <= keyin_num;
        end
    end
end
end
end
```

同時我們也設計了在選擇遊戲的 30、60、90 秒時，會有不同的分數標準。

分別要得到比 5、10、20 分高才會在 endgame state 得到 good 的動畫

否則則會得到 suck

這個 always 就是在做這個判斷

```
always @(*) begin
    if(state==play) begin
        if(keyin_num==6'd1) begin // keyin_num = 1, means 30 seconds, score has to be higher than
            if(sc_cnt1>6'd0 || sc_cnt2>6'd5) begin
                standard = 1;
            end
            else begin
                standard = 0;
            end
        end
        else if(keyin_num==6'd2) begin // keyin_num = 2, means 60 seconds, score has to be higher than 1
            if(sc_cnt1>=6'd1 && sc_cnt2>=6'd0) begin
                standard = 1;
            end
            else begin
                standard = 0;
            end
        end
        else begin // keyin_num = 3, means 90 seconds, score has to be higher than 2
            if(sc_cnt1>=6'd2 && sc_cnt2>=6'd0) begin
                standard = 1;
            end
            else begin
                standard = 0;
            end
        end
    end
    else begin
        standard = std;
    end
end
end
```

最後這個 always 是在計算 play state 時分數的計算

```
always @(posedge new_clk_23 or posedge new_rst) begin // counting score in state play
    if(new_rst) begin
        sc_cnt1 <= 6'b000000;
        sc_cnt2 <= 6'b000000;
    end
    else begin
        if(state==play) begin
            if(dis < 20'd1000 && !pause) begin
                if(sc_cnt1==6'b001001 && sc_cnt2==6'b001001) begin
                    sc_cnt1 <= sc_cnt1;
                    sc_cnt2 <= sc_cnt2;
                end
                else begin
                    if(sc_cnt2==6'b001001) begin
                        if(sc_cnt1==6'b001001) begin
                            sc_cnt1 <= 6'b001001;
                            sc_cnt2 <= 6'b001001;
                        end
                        else begin
                            sc_cnt1 <= sc_cnt1 + 1;
                            sc_cnt2 <= 6'b000000;
                        end
                    end
                    else begin
                        sc_cnt2 <= sc_cnt2 + 1;
                    end
                end
            end
            else begin
                sc_cnt1 <= sc_cnt1;
                sc_cnt2 <= sc_cnt2;
            end
        end
        else begin
            sc_cnt1 <= 6'b000000;
            sc_cnt2 <= 6'b000000;
        end
    end
end
```

## 2. 學到的東西與遇到的困難

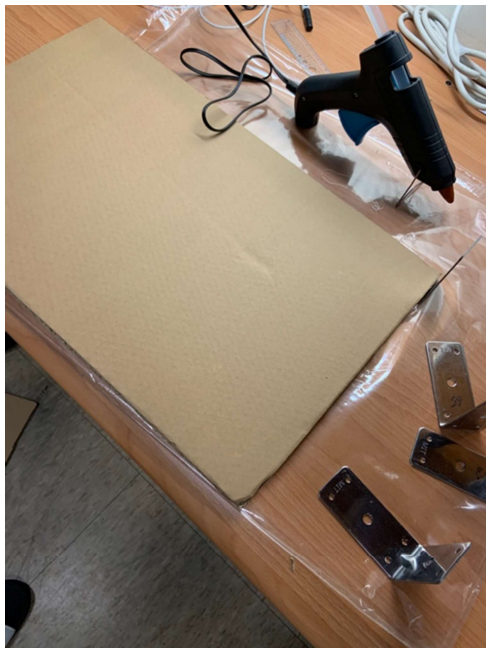
這次的 final project 對我們來說也是一次重大的挑戰，從自己構思要設計出什麼樣的不同結果，到自己去採買各種物品及工具，然後面對各種硬體以及實作辦法。

第一個遇到的問題就是硬體的問題，一開始我們想要將這個投籃機設計成籃筐加上軌道鏈條，讓他是可以做一個左右移動的籃筐增加難度，但是其中實在遇到太多的困難無法解決再加上時間的問題，所以只好放棄最初的這一個構想。

第二個遇到的也是硬體上的困難，一開始在測試時接上買來的感測器但什麼東西都沒放分數還是會一直的增加，換上買來的另外一顆後結果還是一樣，所以想說應該是 code 裡面還有 bug 一直在 code 中找，但是最後在再換上一顆新的感測器後就可以正常的感應以及使用了。

最後則是實體把投籃機做出來的困難。一開始我們本來使用的是紙箱的材質加上膠水粘貼，但是發先太軟了所以會不穩固且容易壞。接著我們在找了好幾種材料後選定使用一種較厚較堅硬但卻也容易裁切的硬紙板。然後發現他在連接的 L 型轉角處不容易牢固，所以去五金行找了 L 形狀的鐵板以及熱熔

膠槍來加強他的穩定性。



### 3. 想對老師或助教說的話

這次的 project 不管是在比較不熟悉的各種硬體又或是練習了一整個學期的 verilog 上都遇到了不少困難，從頭自己構思想像中的成品，到買了材料實作後有了一段相當大的落差。在面對各種硬體材料時，也碰到了不少機器不良了問題而多花了很多的時間，可能這就是實際面對硬體實作可理想中的差距吧。不管結果如何，這次的邏輯設計確實讓我們獲得了不少全新的體驗，面對問題與解決問題，苦惱的同時卻也樂在其中！謝謝老師與助教這一學期辛苦的教導，讓這堂課學習到的不只是邏輯設計的知識，更是學到如何用正確的態度解決問題的意義非凡的一堂課！