

# FINAL PROJECT : DADISHU.....

---

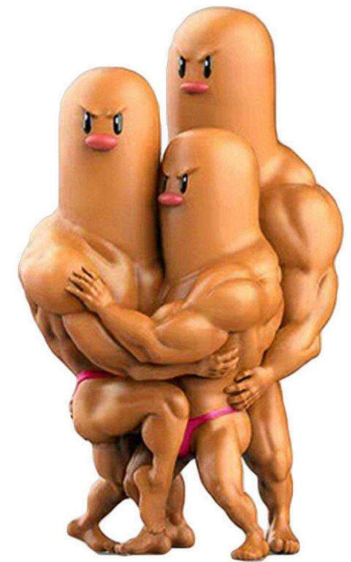
組員:蕭捷晨，黃楷軒，陳學境，陳柏亘，陳奕嘉，蔡子謙，張育誠

# 動機

---

- 這學期在數位系統實驗課中學到了seven display，keypad和dot matrix的verilog程式。因此我們整合串聯這幾種的程式，實作一個簡易的打地鼠遊戲。

地鼠三兄弟->



- 
- **Seven display:** 顯示秒數倒數，分數累計
  - **Dot matrix:** 顯示地鼠位置，擊中地鼠時的圖案(笑臉)
  - **Keypad:** 按鈕對應dot matrix的位子

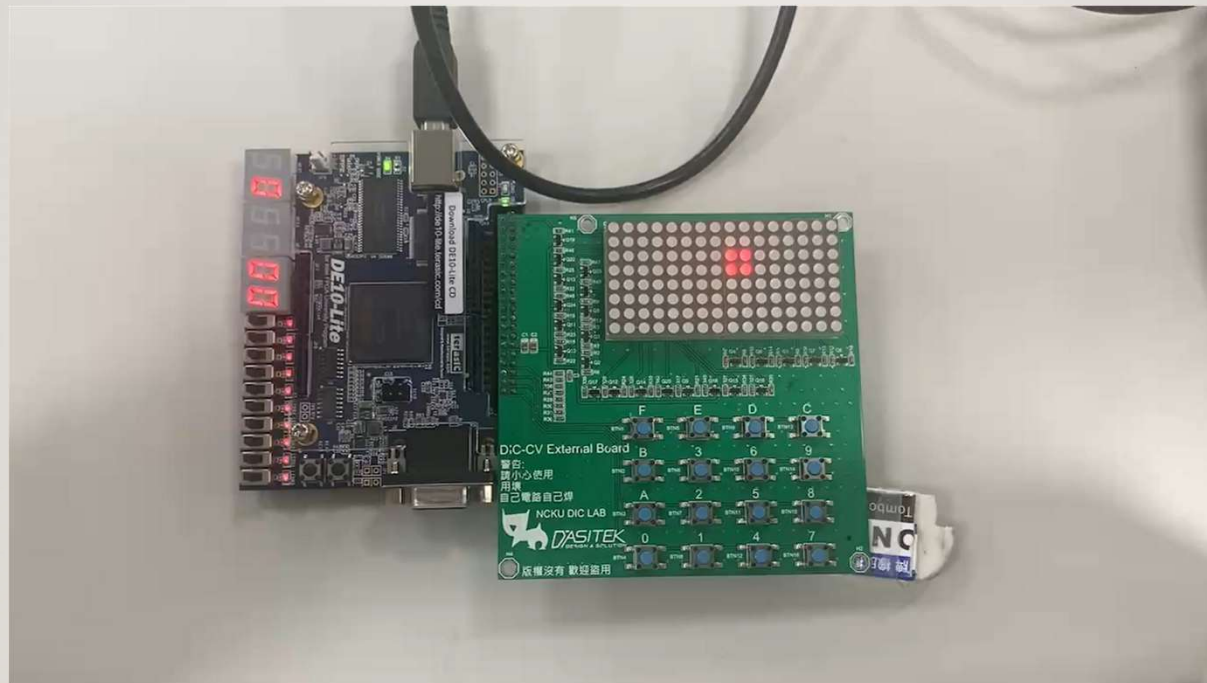
# 遊戲流程

---

1. 遊戲開始，倒數60秒
2. Dot matrix會顯示地鼠(4\*4的紅色方塊)，一秒一隻，若打中將馬上再顯示一隻
3. 按下keypad bottom擊中地鼠得一分
4. 遊戲最後20秒分數翻倍
5. 倒數計時歸0，遊戲結束

# Demo影片

---



# 各個.V檔的介紹

---

- dadishu.v : matrix的地鼠生成
- whack\_points.v : 處理分數累計和時間倒數
- keypadScanner.v : 掃描按鍵是否按下
- LFSR.v : 生成隨機數用於地鼠出現位置
- TopModule.v : 串聯上述四個程式的.V檔



## 遇到的問題I

---

再隨機生成地鼠位置的部分原本想用random(方法:隨機生成數字對16取餘數，對應dot matrix的16個區塊)，但後來發現不支援random，所以採用LFSR。但生成的數字很快就重複且然後看起來不夠亂，所以再修正feedback值，將其複雜化(assign feedback =  $\text{lfsr}[7] \wedge \text{lfsr}[5] \wedge \text{lfsr}[4] \wedge \text{lfsr}[3] \wedge \text{lfsr}[1] \wedge \text{lfsr}[0]$ )，然後增加LFSR的輸出數量

## 遇到的問題2

---

- 我們分成二組撰寫兩部分程式，分別是:地鼠的隨機生成，倒數計時的顯示。但兩邊的偽亂數(LFSR)和按鈕偵測的方法不一樣，所以在整合時修改了這兩個程式以及 input output 的流向(keypadscanner.v檔)



## 遇到的問題3

- 偽隨機的程式因為一秒只刷新一次地鼠出現位置過於規則，再考慮reset還會重設，每次種子都是一樣的，所以多玩幾次可能會重複，我們將LFSR改成16bit，每個clock 運算一次，reset 不重設。讓地鼠生成的位置不再這麼好預測增加遊戲的難度和趣味性!

```
always @(posedge clk or negedge rst) begin
    if (!rst) begin
        lfsr ≤ 8'b11111111; // 初始化 LFSR
        counter ≤ 32'd0; // 初始化计数器
    end else begin
        if (counter == 32'd50000000) begin
            // 计数器到达 50,000,000 时更新 LFSR
            lfsr ≤ {lfsr[6:0], feedback}; // LFSR 更新，反馈位加入到最低位
            counter ≤ 32'd0; // 重置计数器
        end else begin
            // 否则继续计数
            counter ≤ counter + 1;
        end
    end
end
```

## 日後延伸

---

原本想增加遊戲體驗，讓不同地鼠有不同分數，但由於**dot matrix**實作有限，想不到如何有效讓玩家辨識不同地鼠代表不同分數。之後有時間可以加入**vga**的實作把不同地鼠用不一樣的色塊顯示，讓遊戲更加完善

# 分工表

---

- 地鼠生成和keypad:蕭捷晨，陳柏亘
- 分數計算及遊戲倒數:陳學境，陳奕嘉，張育誠
- 程式整合:蔡子謙
- 簡報和報告:黃楷軒