



第八組組長：11036006 王佑強 組員：11036023陳友亮

老師：張隆君

《行動應用開發》



拼車猜猜

經典小遊戲合集

大綱

專案簡述

動機、目的、架構

使用者介面與流程

介面設計原則與使用者流程

技術簡述

系統主要功能

分工

分工人員、工作分配

關鍵程式碼說明

程式碼邏輯、主要功能程式碼



拼圖遊戲



賽車遊戲

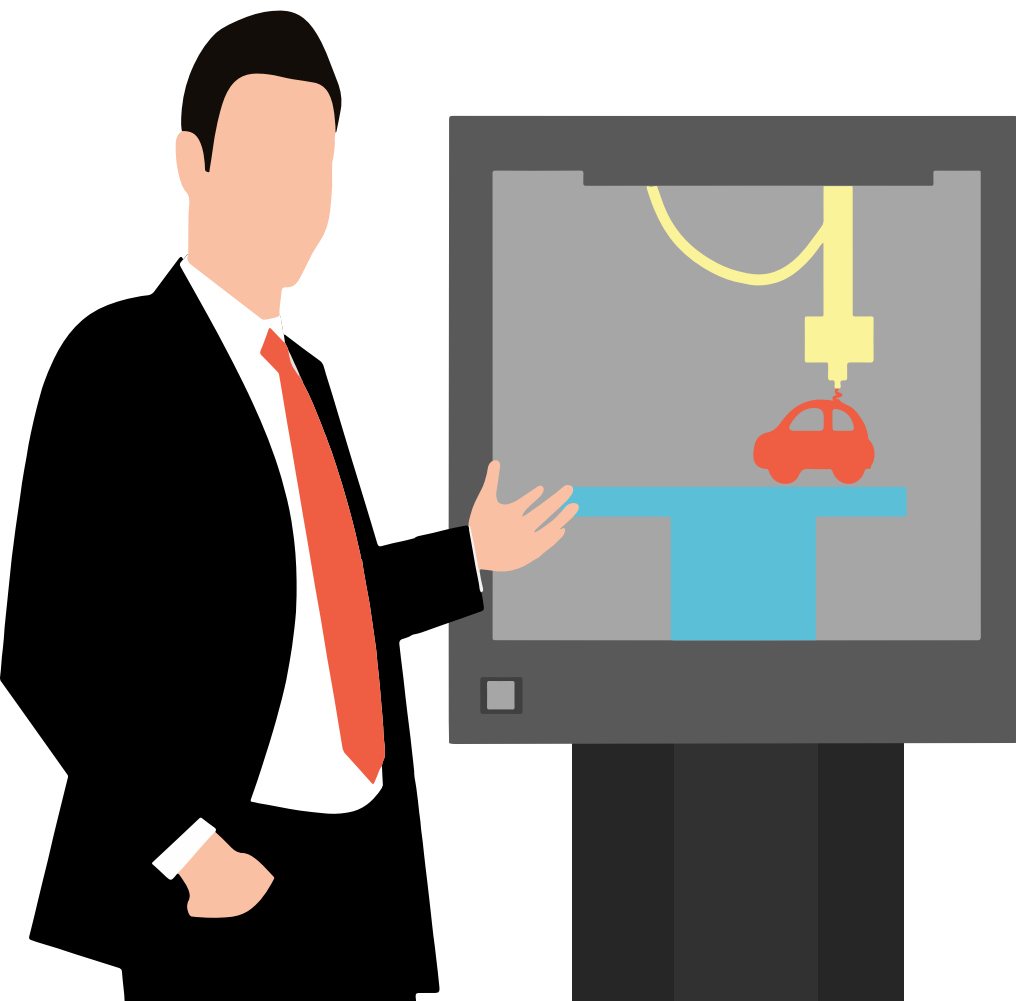


猜數字



剪刀石頭布

專案動機

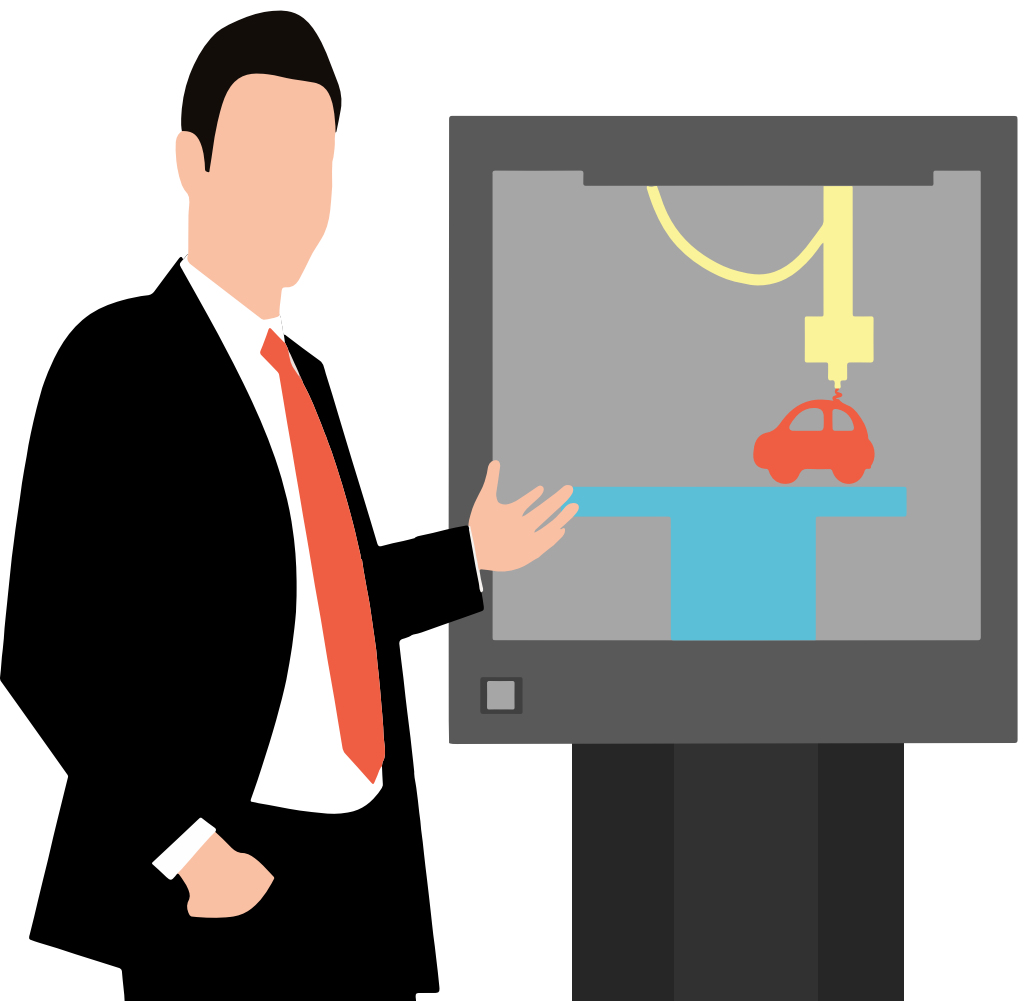


單一遊戲不僅玩的快，也散的快
不停下載新遊戲使得硬體可用空間減少



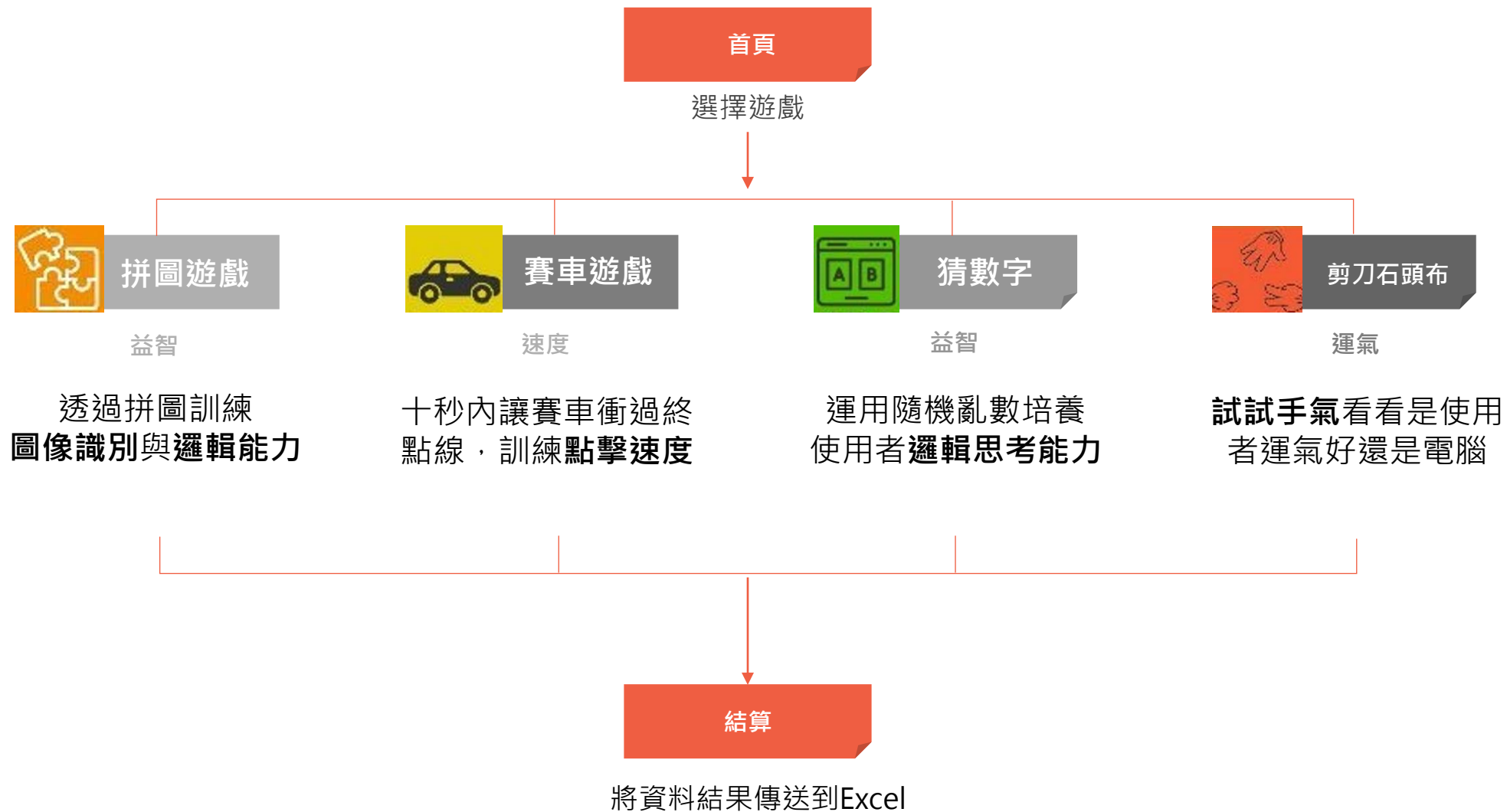
經過兩學期的學習，整合上課所學
打造出理想型遊戲供大家遊玩

專案目的



- 提供一站式多遊戲平台
- 易於學習，培養腦力、反應能力
- 實作並重新複習過去所學，培養邏輯概念

系統架構



介面設計與流程



首頁



簡潔有力、一目瞭然



多種顏色變化



多使用圖片

介面設計與流程



首頁

介面設計與流程

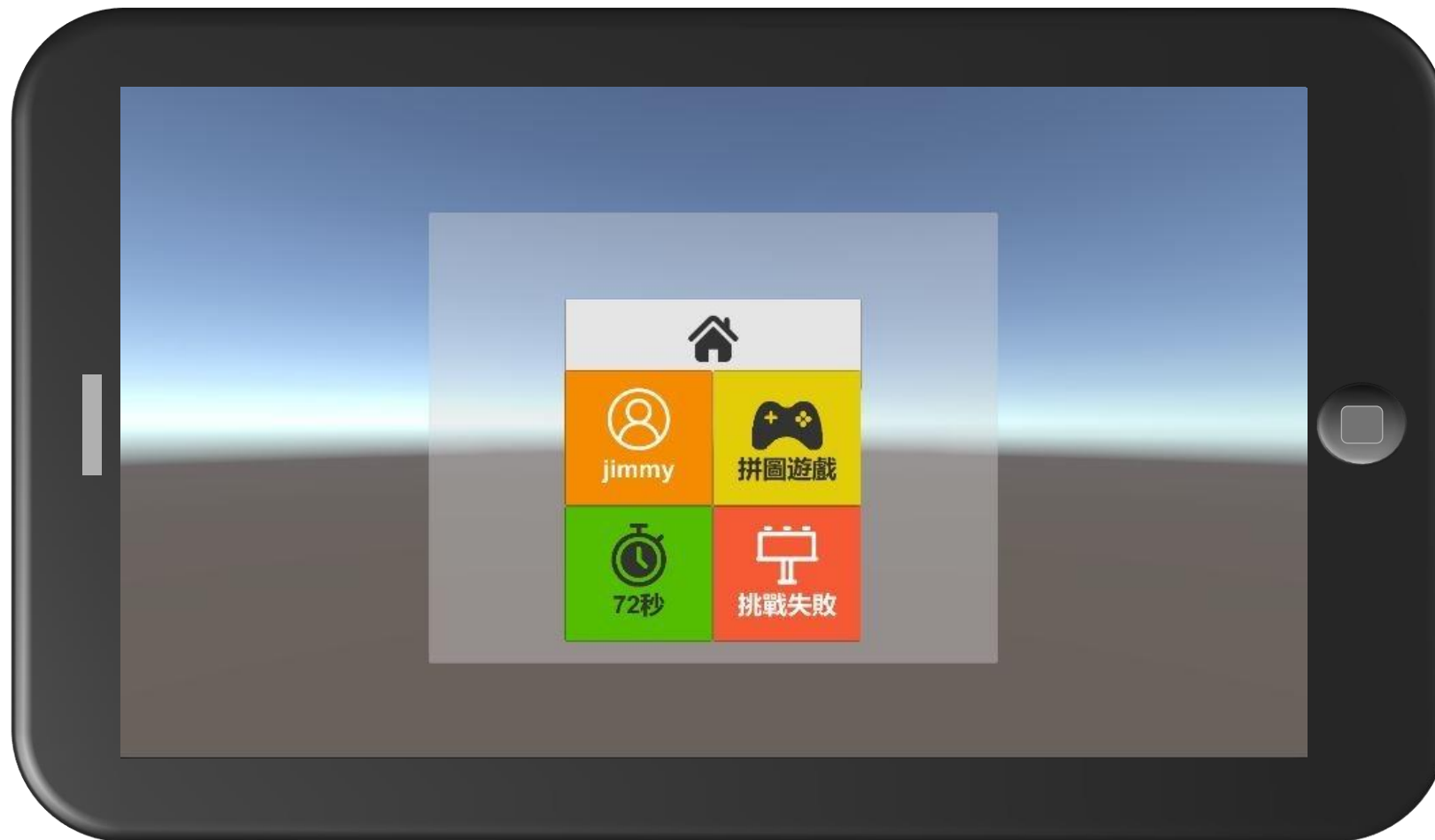
拼圖介面



操作簡單



經典加以創新



結算畫面

將結果資料儲存至Excel
(玩家名稱、遊戲名稱、時長、挑戰狀況)

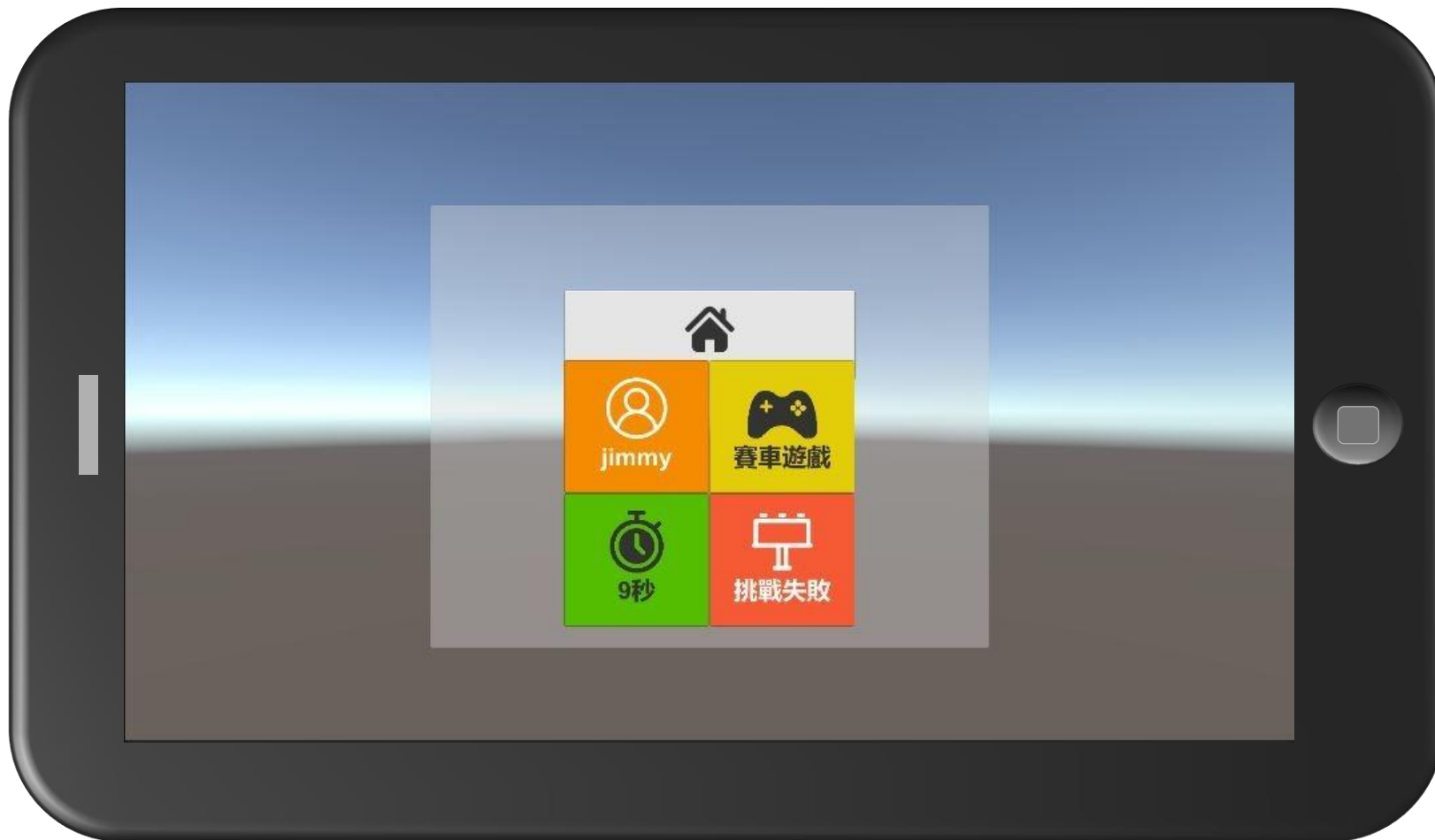
介面設計與流程



首頁

介面設計與流程

賽車遊戲



結算畫面

將結果資料儲存至Excel
(玩家名稱、遊戲名稱、時長、挑戰狀況)



限時規則

限時機制，鍛鍊手指肌肉的
點擊速度。



音效刺激

點擊時出來的音效，產生心
理作用，讓使用者有刺激感。

介面設計與流程



首頁

介面設計與流程

猜數字遊戲

次數限制

限制可猜測數字的機會

遊戲說明 按鈕

1A2B

答案

遊戲成功

時間

01:25

次數

5

遊戲結果

6780

再試一次

離開遊戲

歷史紀錄

記錄每一次輸入的數字
以及猜測之結果

提示

受不了的時候可以點

結算畫面

將結果資料儲存至Excel
(玩家名稱、遊戲名稱、時長、挑戰狀況)

介面設計與流程



首頁

介面設計與流程

剪刀石頭布

勝利條件

勝負計算



經典中的經典

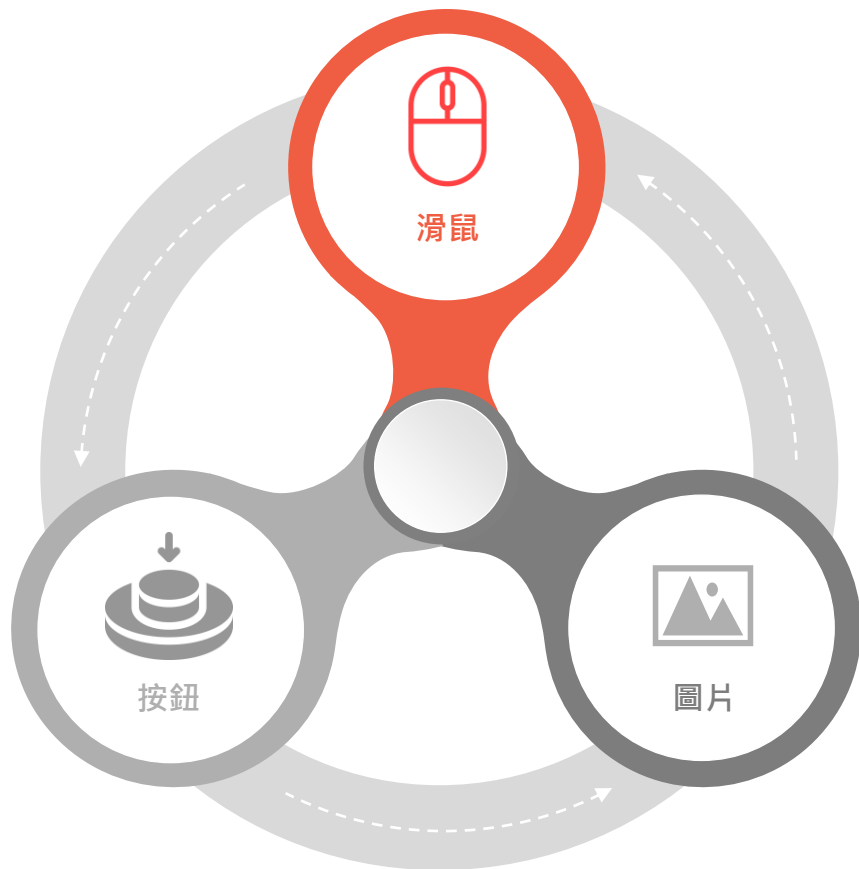
結算畫面

將結果資料儲存至Excel
(玩家名稱、遊戲名稱、時長、挑戰狀況)

每一局顯示結果

技術簡述

觸發事件



- 1 圖片事件：
Image物件新增Event Trigger事件
Type選取 Pointer Click，並執行對應的函數。
- 2 滑鼠事件：
Input.GetMouseButtonDown(0)
當滑鼠在場景任意處按下左鍵按鈕時觸發
- 3 按鈕事件：
Button物件新增OnClick事件並套用函式
當點擊Button按鈕時觸發函式

技術簡述



碼表

運用協程Coroutines

允許將碼表分散到多個畫面(frame)



隨機值

隨機 int，不包含最大值

隨機產生一個整數，範圍：最小值 ~ 最大值(不包含)。



拼圖交換

檢查拼圖可移動的方向

If判斷物件周圍的值是否為空，條件達成時並執行兩值交換。



多Button處理

創建按鈕列表

for迴圈建立每一個按鈕觸發事件，通過判斷點擊按鈕名稱調用相應的方法。

分工

11036006 王佑強

拼圖、賽車遊戲、整合
首頁、遊戲畫面、結束畫面
Excel 寫入、文件製作



猜數字、猜拳遊戲
遊戲畫面、猜數字結束畫面
簡報製作

11036023 陳友亮

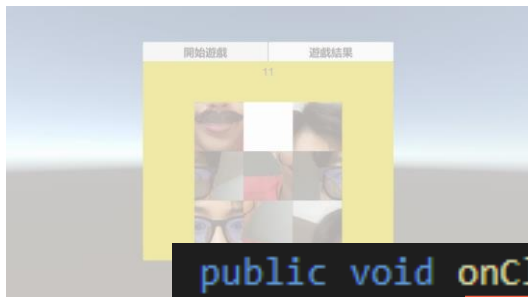
關鍵程式碼說明

```
//使用ExcelPackage打開Excel
using (ExcelPackage package = new ExcelPackage(newFile))
{
    ExcelWorksheet worksheet;
    int row_count = 0; // 初始化列數為0
    if (flag == 0)
    {
        //加入新的頁面，名稱為sheetName
        worksheet = package.Workbook.Worksheets.Add(sheetName);
        //加入列
        worksheet.Cells[row_count + 1, 1].Value = "玩家名稱";
        worksheet.Cells[row_count + 1, 2].Value = "遊戲名稱";
        worksheet.Cells[row_count + 1, 3].Value = "遊戲時間";
        worksheet.Cells[row_count + 1, 4].Value = "過關結果";
        row_count = 1;
    }
    else
    {
        worksheet = package.Workbook.Worksheets[sheetName]; //選擇頁面
        row_count = worksheet.Dimension.Rows; //計算目前頁面的列數
        //開始寫入資料
        worksheet.Cells[row_count + 1, 1].Value = global.playerName;
        worksheet.Cells[row_count + 1, 2].Value = global.gameName;
        worksheet.Cells[row_count + 1, 3].Value = global.useTime + "秒";
        if (global.gameName == "拼圖遊戲")
        {
            worksheet.Cells[row_count + 1, 4].Value = global.puzzleResult;
            gameResult.GetComponent<Text>().text = global.puzzleResult;
        }
        else if (global.gameName == "賽車遊戲")
        {
            worksheet.Cells[row_count + 1, 4].Value = global.racingResult;
            gameResult.GetComponent<Text>().text = global.racingResult;
        }
        else
        {
            worksheet.Cells[row_count + 1, 4].Value = global.racingResult;
            gameResult.GetComponent<Text>().text = global.racingResult;
        }
    }

    // 在畫面顯示使用後資訊與結果
    playerName.GetComponent<Text>().text = global.playerName;
    gameName.GetComponent<Text>().text = global.gameName;
    useTime.GetComponent<Text>().text = global.useTime.ToString() + "秒";
    Debug.Log("成功");
}

//儲存
package.Save();
}
```

1.碼表 -- 運用協程Coroutines



```
public void onClick_btn_inPuzzleGame(){ //進入 拼圖遊戲 Button
    global.arr = global.arr.OrderBy(x => ran.Next()).ToArray(); //將array重新用亂數排序
    global.playerName = InputText.GetComponent<InputField>().text; //取得玩家名稱
    SceneManager.LoadScene("Game"); //切換scene->Game
    global.count = 0; //初始化拼圖交換次數
    global.puzzleResult = "失敗";
    global.gameName = "拼圖遊戲";
    global.stopTime = false;
}
```

```
19 void Update()
20 {
21     if(timer_Start){
22         StartCoroutine("settimer");
23         timer_Start = false;
24         Debug.Log(timer_i);
25     }
26     if(global.stopTime == true){
27         global.useTime = timer_i;
28         StopCoroutine("settimer");
29     }
30 }
```

(2.遊戲按鈕 -- 進入每個遊戲時，取得玩家名稱、初始化遊戲結果、**啟動碼表**)

碼表 -- 運用協程Coroutines



```
19 void Update()
20 {
21     if(timer_Start){
22         StartCoroutine("settimer");
23         timer_Start = false;
24         Debug.Log(timer_i);
25     }
26     if(global.stopTime == true){
27         global.useTime = timer_i;
28         StopCoroutine("settimer");
29     }
30 }
31 }
```

```
public void onClick_btn_InpuzzleGame() { //進入 遊戲開始 Button
    global.arr = global.arr.OrderBy(x => ran.Next()).ToArray(); //將array重新排列
    global.playerTime = InputText.GetComponent<InputField>().text; //取得玩家名稱
    SceneManager.LoadScene("Game"); //切換scene-Game
    global.count = 0; //初始值為0
    global.puzzleResult = "失敗";
    global.gameName = "拼圖遊戲";
    global.stopTime = false;
}
```

1.碼表 -- 運用協程Coroutines



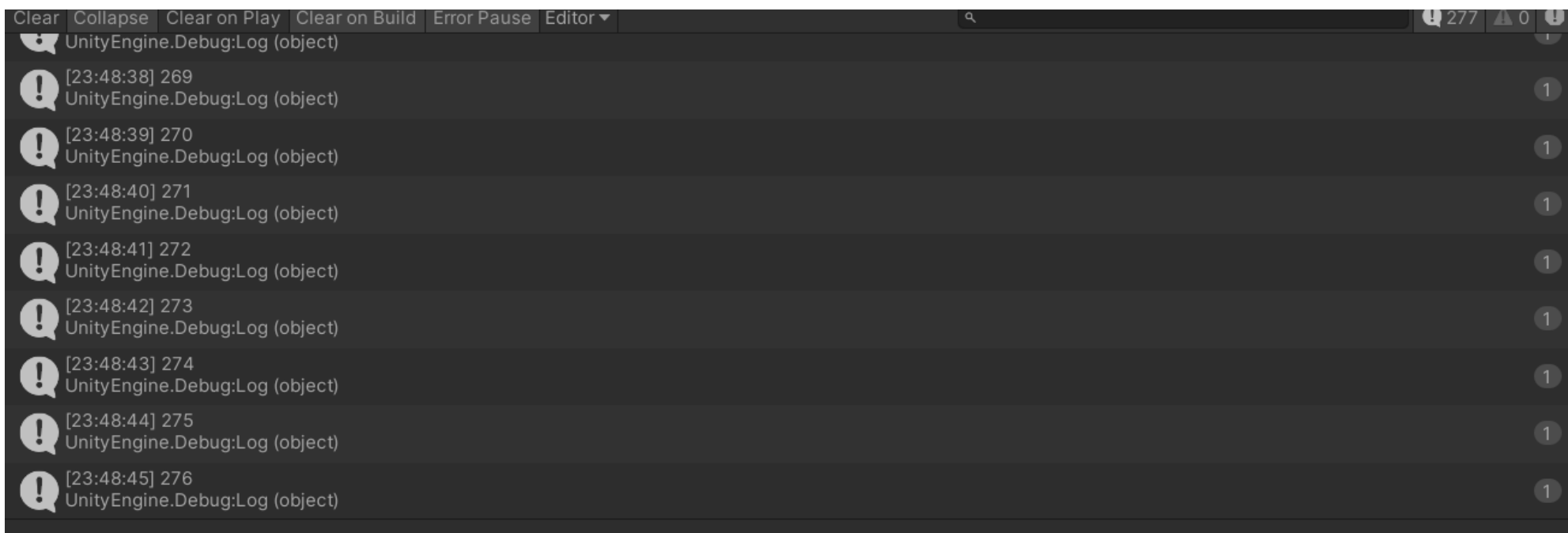
```
19  void Update()  
20  {  
21      if(timer_Start){  
22          StartCoroutine("settimer");  
23          timer_Start = false;  
24          Debug.Log(timer_i);  
25      }  
26      if(global.stopTime == true){  
27          global.useTime = timer_i;  
28          StopCoroutine("settimer");  
29      }  
30  }  
31 }
```

```
public void onClick_btn_InPuzzleTime() { // 進入 計算時間 方法  
    global.arr = global.arr.OrderBy(x => ran.Next()).ToArray(); // 用Array重新排列陣列  
    global.playerTime = InputText.GetComponent<InputField>().text; // 取得玩家名稱  
    SceneManager.LoadScene("Game"); // 切換scene-Game  
    global.count = 0; // 初始化陣列長度  
    global.puzzleResult = "失敗";  
    global.gameName = "計算時間";  
    global.stopTime = false;  
}
```

1.碼表 -- 運用協程Coroutines



```
19 void Update()  
20 {  
21     if(timer_Start){  
22         StartCoroutine("settimer");  
23         timer_Start = false;  
24         Debug.Log(timer_i);  
25     }  
26     if(global.stopTime == true){  
27         global.useTime = timer_i;  
28         StopCoroutine("settimer");  
29     }  
30 }  
31 }
```



```
public void onClick_btn_InpuzzleTime() { //進入 遊戲開始 Button  
    global.arr = global.arr.OrderBy(x => ran.Next()).ToArray(); //用array重新排列數字  
    global.playerTime = InputText.GetComponent<InputField>().text; //取得玩家名稱  
    SceneManager.LoadScene("Game"); //切換scene-Game  
    global.count = 0; //初始化遊戲次數  
    global.puzzleResult = "失敗";  
    global.gameName = "拼圖遊戲";  
    global.stopTime = false;  
}
```

1.碼表 -- 運用協程Coroutines

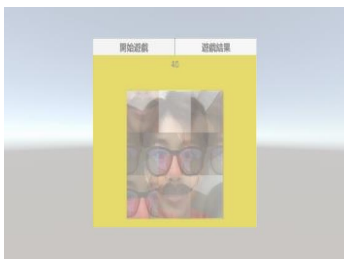


3.拼圖遊戲



```
// Start is called before the first frame update
void Start()
{
    // 將本遊戲的圖庫資源加載到 Unity 中
    img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("img1", typeof(Sprite));
    img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("img2", typeof(Sprite));
    img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("img3", typeof(Sprite));
    img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("img4", typeof(Sprite));
    img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("img5", typeof(Sprite));
    img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("img6", typeof(Sprite));
    img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("img7", typeof(Sprite));
    img8.GetComponent<Image>().sprite = Resources.Load<Sprite>("img8", typeof(Sprite));
}
```

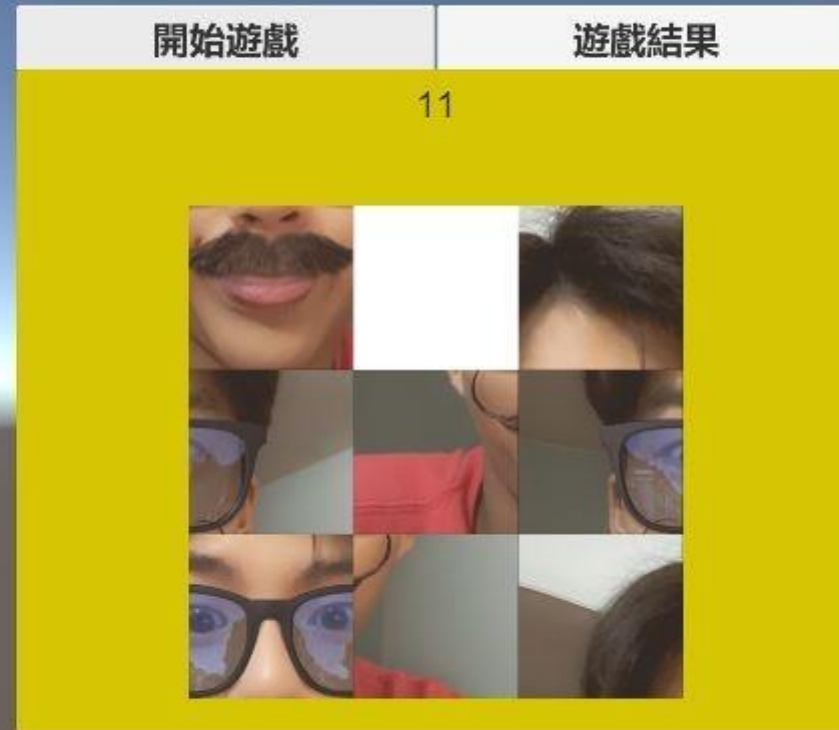
3.拼圖遊戲



將八塊拼圖以產生的亂數放入拼板，達成每次隨機

```
// Start is called before the first frame update
void Start()
{
    // 將各圖檔依照亂術後的順序，放入各img物件
    img1.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[0]);
    img2.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[1]);
    img3.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[2]);
    img4.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[3]);
    img5.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[4]);
    img6.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[5]);
    img7.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[6]);
    img8.GetComponent<Image>().sprite = Resources.Load <Sprite>("圖_0"+global.arr[7]);
}
```

3.拼圖遊戲



```
// Start is called before the first frame update  
void Start()  
{  
    // 將本遊戲的各個圖片的圖片，存到img陣列  
    img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("img1", typeof(Sprite));  
    img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("img2", typeof(Sprite));  
    img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("img3", typeof(Sprite));  
    img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("img4", typeof(Sprite));  
    img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("img5", typeof(Sprite));  
    img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("img6", typeof(Sprite));  
    img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("img7", typeof(Sprite));  
}
```

4.拼圖遊戲

檢查每片拼圖可移動的方向，並且檢查受否成功

開始遊戲

遊戲結果

```
public void onclick_img2(){
    //檢查左邊是否為空
    if(img1.GetComponent<Image>().sprite == null){
        //移動拼圖
        img1.GetComponent<Image>().sprite = img2.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        //累加拼圖次數
        global.count += 1;
        txt_count.GetComponent<Text>().text = global.count.ToString();
    }else if(img3.GetComponent<Image>().sprite == null){
        img3.GetComponent<Image>().sprite = img2.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        global.count += 1;
        txt_count.GetComponent<Text>().text = global.count.ToString();
    }else if(img5.GetComponent<Image>().sprite == null){
        img5.GetComponent<Image>().sprite = img2.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        global.count += 1;
        txt_count.GetComponent<Text>().text = global.count.ToString();
    }
    //檢查拼圖是否成功
    check_success();
}
```

```
// Start is called before the first frame update
void Start()
{
    // 將本遊戲的圖庫與物件的圖庫，都加入img0中
    img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[0]");
    img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[1]");
    img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[2]");
    img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[3]");
    img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[4]");
    img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[5]");
    img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[6]");
    img8.GetComponent<Image>().sprite = Resources.Load<Sprite>("img0/global.arr[7]");
}
```

關鍵程式碼說明

5.拼圖遊戲

check_siccess(); 函式

檢查拼圖是否又在正確位置上

```
public void check_success(){ //比對拼對的數量
    int success=0;

    if(img1.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_01"))
        success +=1;
    if(img2.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_02"))
        success +=1;
    if(img3.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_03"))
        success +=1;
    if(img4.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_04"))
        success +=1;
    if(img5.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_05"))
        success +=1;
    if(img6.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_06"))
        success +=1;
    if(img7.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_07"))
        success +=1;
    if(img8.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_08"))
        success +=1;
    if(img9.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_09"))
        success +=1;
    if(success == 8){
        Debug.Log("success");
        global.puzzleResult = "挑戰成功";
        fill_missingPiece();
    }else{
        global.puzzleResult = "挑戰失敗";
    }
}
```

```
// Start is called before the first frame update
void Start()
{
    // 將本遊戲中所有需要的圖片，都load到global.arr[]中
    img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_01", global.arr[0]);
    img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_02", global.arr[1]);
    img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_03", global.arr[2]);
    img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_04", global.arr[3]);
    img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_05", global.arr[4]);
    img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_06", global.arr[5]);
    img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_07", global.arr[6]);
    img8.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_08", global.arr[7]);
    img9.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_09", global.arr[8]);
}
```

```
void fill_missing_Piece()
{
    // 檢查所有拼圖是否都正確
    if(img1.GetComponent<Image>().sprite == null){
        // 拼圖01
        img1.GetComponent<Image>().sprite = img1.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        // 拼圖02
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    }else if(img1.GetComponent<Image>().sprite == null){
        // 拼圖03
        img1.GetComponent<Image>().sprite = img1.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    }else if(img1.GetComponent<Image>().sprite == null){
        // 拼圖04
        img1.GetComponent<Image>().sprite = img1.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    }else if(img1.GetComponent<Image>().sprite == null){
        // 拼圖05
        img1.GetComponent<Image>().sprite = img1.GetComponent<Image>().sprite;
        img2.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    }
    // 檢查所有拼圖是否都正確
    check_success();
}
```

關鍵程式碼說明

6.拼圖遊戲

fill_missingPiece(); 函式

填滿最後一片拼圖

```
public void fill_missingPiece(){ //填滿缺少的拼圖
    if(img1.GetComponent<Image>().sprite == null)
        img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_01");
    else if(img2.GetComponent<Image>().sprite == null)
        img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_02");
    else if(img3.GetComponent<Image>().sprite == null)
        img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_03");
    else if(img4.GetComponent<Image>().sprite == null)
        img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_04");
    else if(img5.GetComponent<Image>().sprite == null)
        img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_05");
    else if(img6.GetComponent<Image>().sprite == null)
        img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_06");
    else if(img7.GetComponent<Image>().sprite == null)
        img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_07");
    else if(img8.GetComponent<Image>().sprite == null)
        img8.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_08");
    else if(img9.GetComponent<Image>().sprite == null)
        img9.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_09");
}
```

```
// Start is called before the first frame update
void Start()
{
    // 將本遊戲中所有需要的圖片，都load到記憶體中
    img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_01");
    img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_02");
    img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_03");
    img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_04");
    img5.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_05");
    img6.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_06");
    img7.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_07");
}
```

```
void Update()
{
    // 檢查是否所有圖片都load到記憶體中
    if(img1.GetComponent<Image>().sprite == null){
        // 圖片未load
        img1.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_01");
        img2.GetComponent<Image>().sprite = null;
        // 圖片未load
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    } else if(img2.GetComponent<Image>().sprite == null){
        img2.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_02");
        img3.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    } else if(img3.GetComponent<Image>().sprite == null){
        img3.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_03");
        img4.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    } else if(img4.GetComponent<Image>().sprite == null){
        img4.GetComponent<Image>().sprite = Resources.Load<Sprite>("圖_04");
        img5.GetComponent<Image>().sprite = null;
        global.count += 1;
        text_count.GetComponent<Text>().text = global.count.ToString();
    }
    // 檢查是否所有圖片都load到記憶體中
    check_success();
}
```

```
void check_success()
{
    // 檢查是否所有圖片都load到記憶體中
    if(img1.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_01"))
        success += 1;
    if(img2.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_02"))
        success += 1;
    if(img3.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_03"))
        success += 1;
    if(img4.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_04"))
        success += 1;
    if(img5.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_05"))
        success += 1;
    if(img6.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_06"))
        success += 1;
    if(img7.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_07"))
        success += 1;
    if(img8.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_08"))
        success += 1;
    if(img9.GetComponent<Image>().sprite == Resources.Load<Sprite>("圖_09"))
        success += 1;
    if(success == 9){
        Debug.Log("success");
        global.pieceResult = "拼圖完成";
        // 1. 顯示勝利畫面
        // 2. 顯示勝利畫面
        global.pieceResult = "拼圖完成";
    }
}
```

關鍵程式碼說明

7.賽車遊戲

每次點擊畫面賽車前進速度



7.賽車遊戲

每次點擊畫面賽車前進速度

```
void Update()
{
    if(Input.GetMouseButtonDown(0)){
        speed=0.025F;           // 賽車前進速度
        startPo= Input.mousePosition;
        Audio.GetComponent().Play();
    }

    car.transform.Translate(speed,0,0);
    speed*=0.98F;    // 賽車緩速
}
```



8.賽車遊戲

判斷賽車是否在時間內抵達終點，決定成功與否

```
// Update is called once per frame
void Update()
{
    float length = flag.transform.position.x - car.transform.position.x;
    if(length>0){
        if (timeRemaining > 0)
        {
            timeRemaining -= Time.deltaTime;
            distance.GetComponent<Text>().text = "剩下" + string.Format("{0:f2}", timeRemaining)+
        }
        if (timeRemaining < 0){
            distance.GetComponent<Text>().text = "挑戰失敗" ;
            global.racingResult = "挑戰失敗" ;
            global.stopTime = true;
        }
        // distance.GetComponent<Text>().text = "距離終點還有" + length.ToString("F2") + "m";
    }else{
        distance.GetComponent<Text>().text = "抵達終點";
        global.racingResult = "挑戰成功" ;
        global.stopTime = true;
    }
}
```

```
void Update()
{
    if(Input.GetMouseButton(0)){
        speed+=0.025f; // 賽車加速速度
        startPos = Input.mousePosition;
        Audio.GetComponent<AudioSource>().Play();
    }

    car.transform.Translate(speed,0,0);
    speed+=0.005f; // 賽車速度
}
```

關鍵程式碼說明

9.寫入Excel

結算畫面



9.寫入Excel

將遊戲結果與玩家資訊寫入Excel、並且呈現在result畫面上

```
//使用ExcelPackage打開Excel
using (ExcelPackage package = new ExcelPackage(newFile))
{
    ExcelWorksheet worksheet;
    int row_count = 0; // 初始化列數為0
    if (flag == 0)
    {
        //加入新的頁面，名稱為sheetName
        worksheet = package.Workbook.Worksheets.Add(sheetName);
        //加入列
        worksheet.Cells[row_count + 1, 1].Value = "玩家名稱";
        worksheet.Cells[row_count + 1, 2].Value = "遊戲名稱";
        worksheet.Cells[row_count + 1, 3].Value = "遊玩時間";
        worksheet.Cells[row_count + 1, 4].Value = "通關結果";
        row_count = 1;
    }
    else
    {
        worksheet = package.Workbook.Worksheets[sheetName]; //選擇頁面
        row_count = worksheet.Dimension.Rows; //計算目前頁面的列數
        //開始寫入資料
        worksheet.Cells[row_count + 1, 1].Value = global.playerName;
        worksheet.Cells[row_count + 1, 2].Value = global.gameName;
        worksheet.Cells[row_count + 1, 3].Value = global.useTime + "秒";
        if(global.gameName == "拼圖遊戲"){
            worksheet.Cells[row_count + 1, 4].Value = global.puzzleResult;
            gameResult.GetComponent<Text>().text = global.puzzleResult;
        }
        else if(global.gameName == "賽車遊戲"){
```



關鍵程式碼說明

9.寫入Excel

將遊戲結果與玩家資訊寫入Excel、並且呈現在result畫面上



```
if(global.gameName == "拼圖遊戲"){
    worksheet.Cells[row_count + 1, 4].Value = global.puzzleResult;
    gameResult.GetComponent<Text>().text = global.puzzleResult;
}
else if(global.gameName == "賽車遊戲"){
    worksheet.Cells[row_count + 1, 4].Value = global.racingResult;
    gameResult.GetComponent<Text>().text = global.racingResult;
}
else{
    worksheet.Cells[row_count + 1, 4].Value = global.racingResult;
    gameResult.GetComponent<Text>().text = global.racingResult;
}
```

// 在畫面顯示使用者資訊與結果

```
playerName.GetComponent<Text>().text = global.playerName;
gameName.GetComponent<Text>().text = global.gameName;
useTime.GetComponent<Text>().text = global.useTime.ToString() + "秒";
Debug.Log("成功");
```

//儲存

```
package.Save();
```

```
}
```



9.寫入Excel

將遊戲結果與玩家資訊寫入Excel、並且呈現在result畫面上



玩家名稱	遊戲名稱	遊玩時間	通關結果
jimmy	拼圖遊戲	72秒	挑戰失敗
jimmy	賽車遊戲	9秒	挑戰失敗
jimmy	1A2B遊戲	84秒	挑戰成功
jimmy	剪刀石頭	73秒	挑戰成功
jimmy	剪刀石頭	16秒	挑戰失敗
jimmy	剪刀石頭	15秒	挑戰失敗



10.猜數字遊戲

時間計時器

時間
00:38

剩餘次數
6

歷史紀錄	結果
5496	0A1B
2187	1A1B
2149	0A0B
9783	2A0B

1

2

3

4

5

6

7

8

9

0

⬅️✕

💡

OK

結束

10.猜數字遊戲

時間計時器

時間

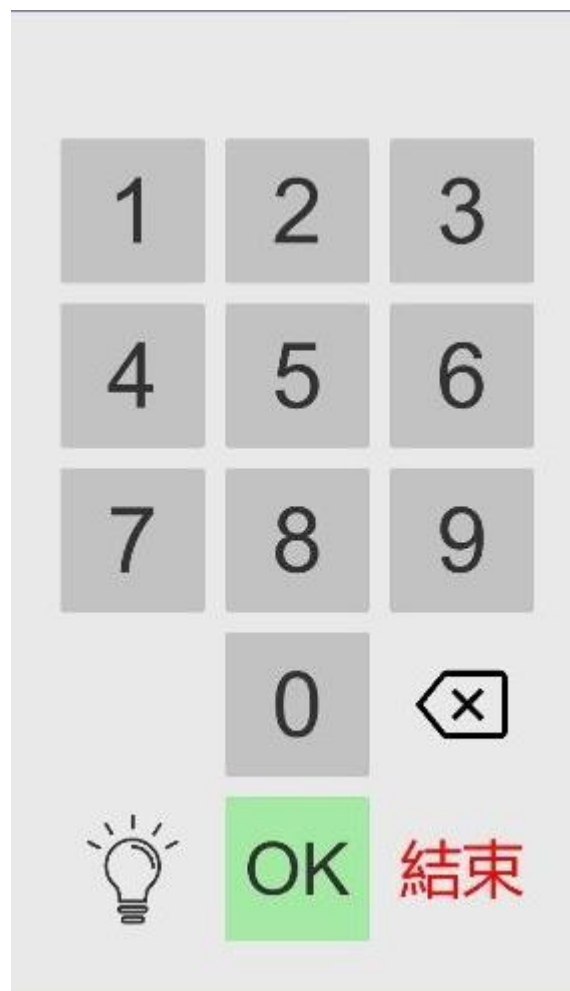
00:38

```
public float timer_f = 0;  
public int timer_i = 0;
```

```
timer_f = Time.timeSinceLevelLoad;  
timer_i = Mathf.FloorToInt(timer_f);  
int minute = timer_i / 60;  
int second = timer_i % 60;  
time_data = minute.ToString("00") + ":" + second.ToString("00");  
txt_time.GetComponent<Text>().text = time_data;
```

11.猜數字遊戲

Buttons陣列處理(for迴圈)



11.猜數字遊戲

Buttons陣列處理(for迴圈)

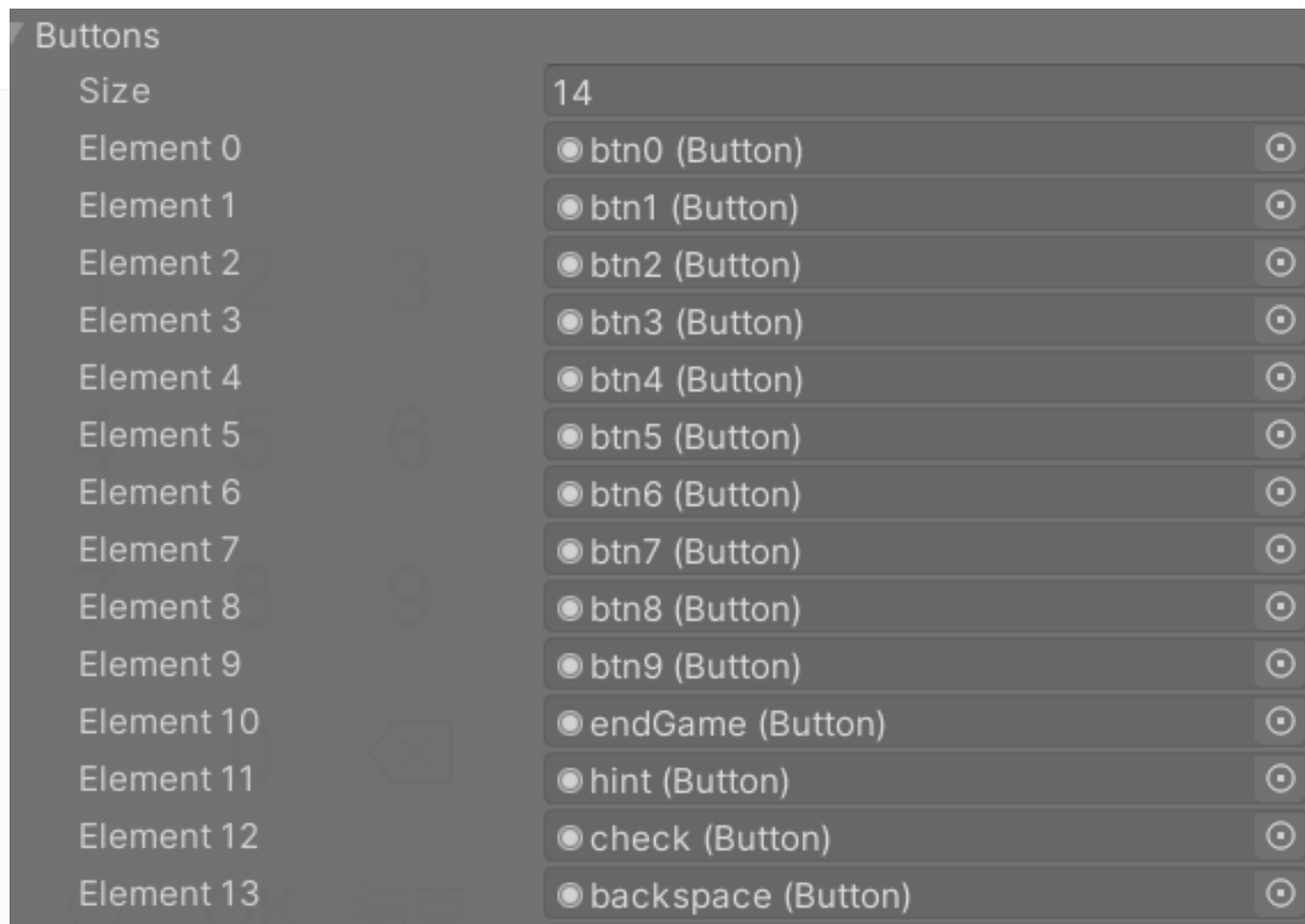
```
public List<Button> buttons = new List<Button>();
```

Buttons		
Size	14	
Element 0	● btn0 (Button)	⊙
Element 1	● btn1 (Button)	⊙
Element 2	● btn2 (Button)	⊙
Element 3	● btn3 (Button)	⊙
Element 4	● btn4 (Button)	⊙
Element 5	● btn5 (Button)	⊙
Element 6	● btn6 (Button)	⊙
Element 7	● btn7 (Button)	⊙
Element 8	● btn8 (Button)	⊙
Element 9	● btn9 (Button)	⊙
Element 10	● endGame (Button)	⊙
Element 11	● hint (Button)	⊙
Element 12	● check (Button)	⊙
Element 13	● backspace (Button)	⊙

11.猜數字遊戲

Buttons陣列處理(for迴圈)

```
public List<Button> buttons = new List<Button>();  
  
void Start()  
{  
    AddClickEvents();  
}
```



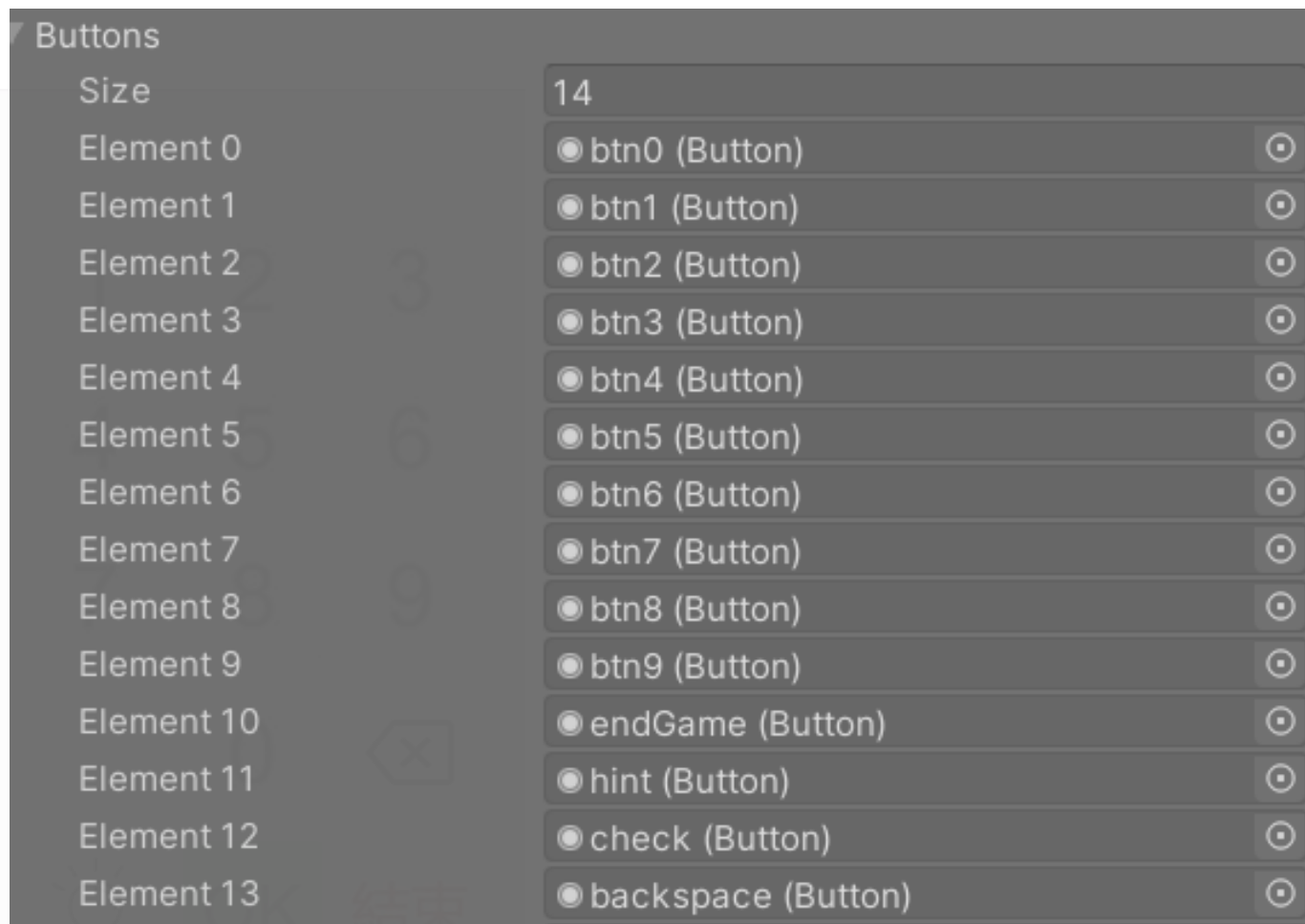
11.猜數字遊戲

Buttons陣列處理(for迴圈)

```
public List<Button> buttons = new List<Button>();

void Start()
{
    AddClickEvents();
}

void AddClickEvents()
{
    for(int i = 0; i < buttons.Count; i++){
        int j = i;
        buttons[i].onClick.AddListener(() =>
ClickEvent(j));
    }
}
```



11.猜數字遊戲

Buttons陣列處理(for迴圈)

```
void ClickEvent(int a)
{
    switch (buttons[a].name)
    {
        case "btn0":
            ClickNumberButton(0);
            break;
        case "btn1":
            ClickNumberButton(1);
            break;
        ...
        case "endGame":
            endGame();
            break;
        case "hint":
            hint();
            break;
        case "check":
            check();
            break;
        case "backspace":
            backspace();
            break;
        default:
            break;
    }
}
```

12.猜數字遊戲

退格鍵



```
string numstring = inputField.text;
if (numstring.Length > 0) {
    inputField.text = numstring.Remove (numstring.Length - 1);
} else {
    inputField.text = "";
}
```



OK

結束

13.猜數字遊戲

亂數產生(正確解答) 儲存到陣列, Length = 4

```
int[] a = new int[4];

bool repeat = false;

do {
    int pwd = 0;
    pwd = Random.Range(0123,9877);
    int x = 1000;

    for(int k = 0;k < a.Length;k++) {
        a[k] = pwd / (int)x;
        pwd = pwd % (int)x;
        x = x / 10;
    }

    repeat = false;
}while(repeat == true);

for(int k = 0;k < a.Length;k++) {
    global.CompleteLpcation[k] = (a[k]).ToString();
}
```



13.猜數字遊戲

亂數產生(正確解答) 儲存到陣列, Length = 4

```
do {
    int pwd = 0;
    pwd = Random.Range(0123,9877);
    int x = 1000;

    for(int k = 0;k < a.Length;k++) {
        a[k] = pwd / (int)x;
        pwd = pwd % (int)x;
        x = x / 10;
    }

    repeat = false;

    for(int i = 0;i < a.Length && repeat == false;i++) {
        for(int j = i+1;j < a.Length && repeat == false;j++) {
            if(a[i]==a[j]) {
                repeat = true;
                break;
            }else {
                repeat = false;
            }
        }
    }
}while(repeat == true);

for(int k = 0;k < a.Length;k++) {
    global.Completelocation[k] = (a[k]).ToString();
}
```

14.猜數字遊戲

檢查輸入數字是否為四位數 4



1234

您每次需要輸入4個數字.

```
if(input_data.Length != 4){  
    inputError.GetComponent<Text>().text = "您每次需要輸入4個數字.";
```


14.猜數字遊戲

檢查是否猜對

```
}else{
    int A=0,B=0;
    for(int i = 0;i < global.CompleteLpcation.Length;i++) {
        for(int j = 0;j < input_data.Length ;j++) {

            if(global.CompleteLpcation[i] == Char.ToString(input_data[j]) && i == j) {
                A++;
            }else if(global.CompleteLpcation[i] == Char.ToString(input_data[j])) {
                B++;
            }else {
                //沒事發生
            }
        }
    }
}
```

15.猜數字遊戲

達到4A代表遊戲成功 (資料儲存至全域變數)

```
if(A == 4) {
    Debug.Log("4A");
    count +=1;
    result_data = true;
    if(result_data){
        global.IAtoBResult = true;
    }else{
        global.IAtoBResult = false;
    }
    global.IAtoBName = "1A2B";
    global.IAtoBCount = Convert.ToString(count);

    for(int k = 0;k < global.CompleteLpcation.Length;k++) {
        data_answer += global.CompleteLpcation[k];
    }

    global.IAtoBAnswer = data_answer;
    global.IAtoBTime = time_data;
    global.stopTime = true;

    SceneManager.LoadScene("Result_1A2B");
}else {

    count +=1;
```

15.猜數字遊戲

未達到4A,紀錄顯示在歷史紀錄

歷史紀錄	結果
5496	0A1B
2187	1A1B
2149	0A0B
9783	2A0B

```
SceneManager.LoadScene("Result_1A2B");  
}else {  
  
    count +=1;
```

15.猜數字遊戲

未達到4A,紀錄顯示在歷史紀錄
產生一個物件

```
}else {  
  
    count +=1;  
    GameObject obj = Instantiate(CellPrefab);  
    obj.transform.SetParent(record.transform);  
    GameObject obj2 = obj.transform.GetChild(0).gameObject;  
    GameObject obj3 = obj.transform.GetChild(1).gameObject;  
    obj2.GetComponent<Text>().text = input_data;  
    obj3.GetComponent<Text>().text = A+"A"+B+"B";  
    obj.transform.localScale = new Vector3(1,1,1);  
    post_processing();  
}
```

15.猜數字遊戲

如果總數達到0 遊戲結束(資料傳輸)

```
public void post_processing()
{
    inputField.text = "";
    txt_count.GetComponent<Text>().text = Convert.ToString(total-count);
    if( total - count <= 0 )
    {
        global.IAtoBName = "1A2B";
        global.IAtoBCount = Convert.ToString(count);
        global.IAtoBResult = false;
        for(int k = 0;k < global.CompleteLpcation.Length;k++) {
            data_answer += global.CompleteLpcation[k];
        }
        global.IAtoBAnswer = data_answer;
        global.IAtoBTime = time_data;
        SceneManager.LoadScene("Result_1A2B");
    }
}
```

16.猜數字遊戲

以秒計算到最後的關卡已經載入完的時間並無條件捨去

```
public void post_processing()
{
    inputField.text = "";
    txt_count.GetComponent<Text>().text = Convert.ToString(total-count);
    if( total - count <= 0 )
    {
        global.IAtoBName = "1A2B";
        global.IAtoBCount = Convert.ToString(count);
        global.IAtoBResult = false;
        for(int k = 0;k < global.CompleteLpcation.Length;k++) {
            data_answer += global.CompleteLpcation[k];
        }
        global.IAtoBAnswer = data_answer;
        global.IAtoBTime = time_data;
        SceneManager.LoadScene("Result_1A2B");
    }
}
```



unity

THANK YOU