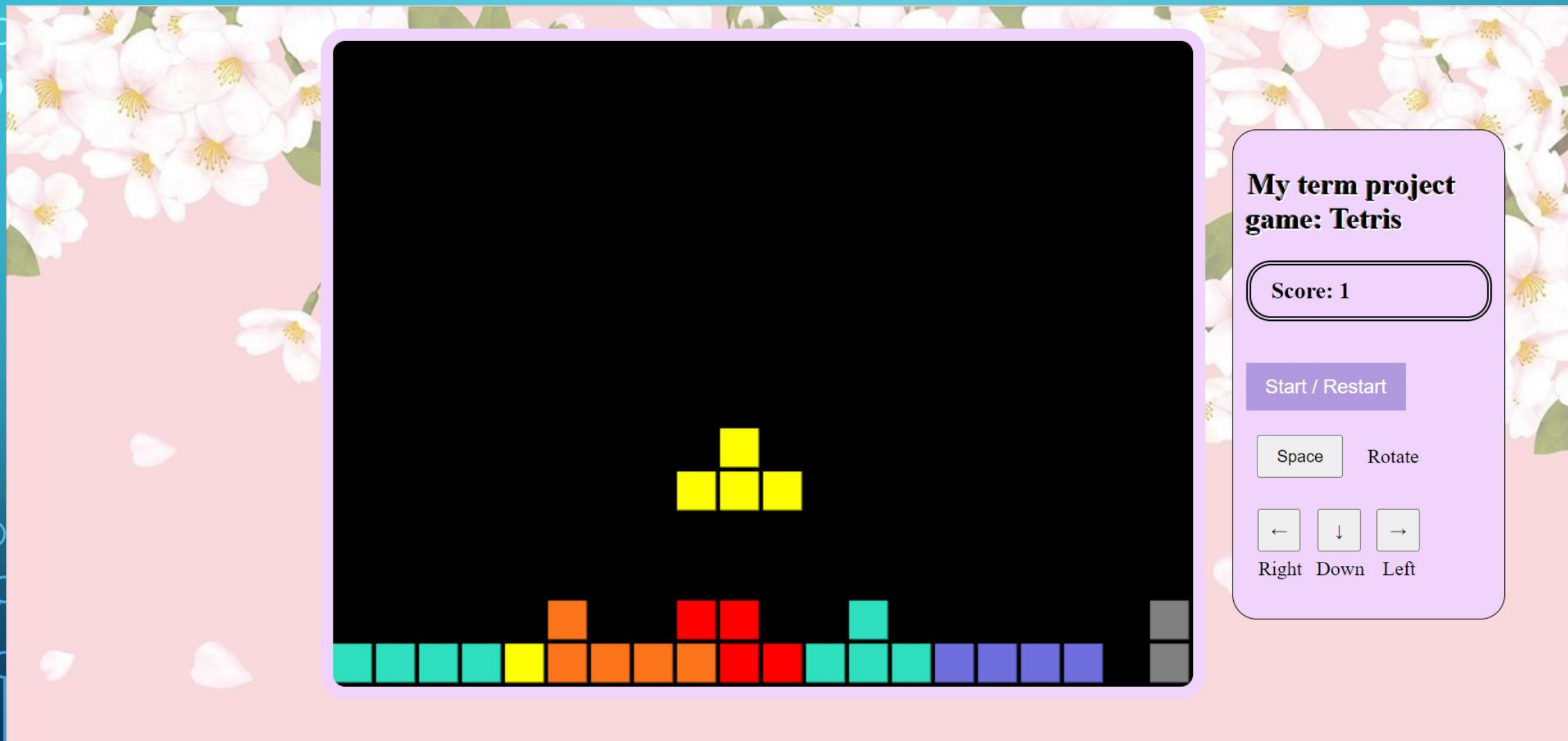




# JAVASCRIPT TERM PROJECT: TETRIS

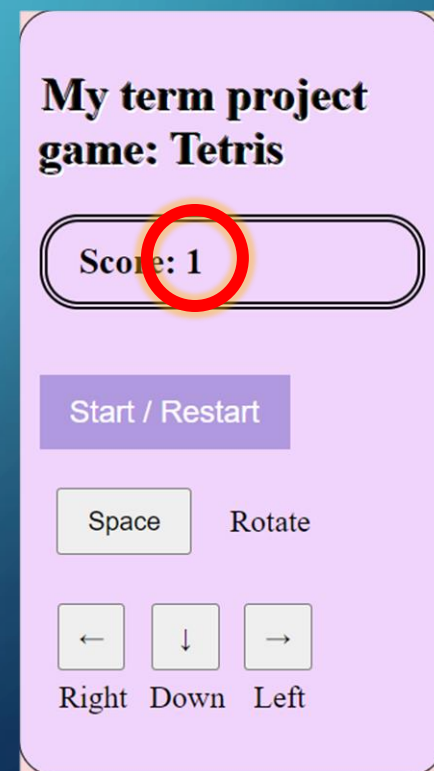
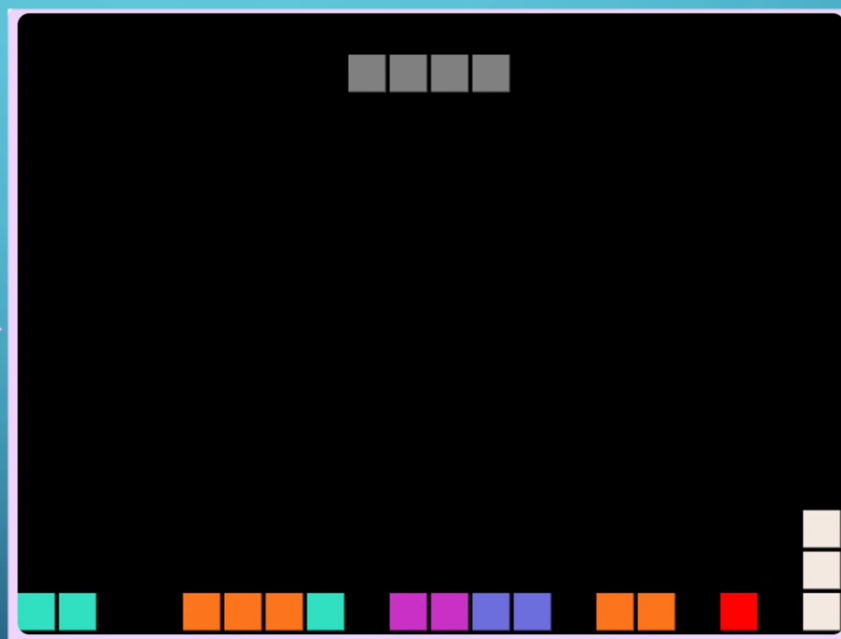
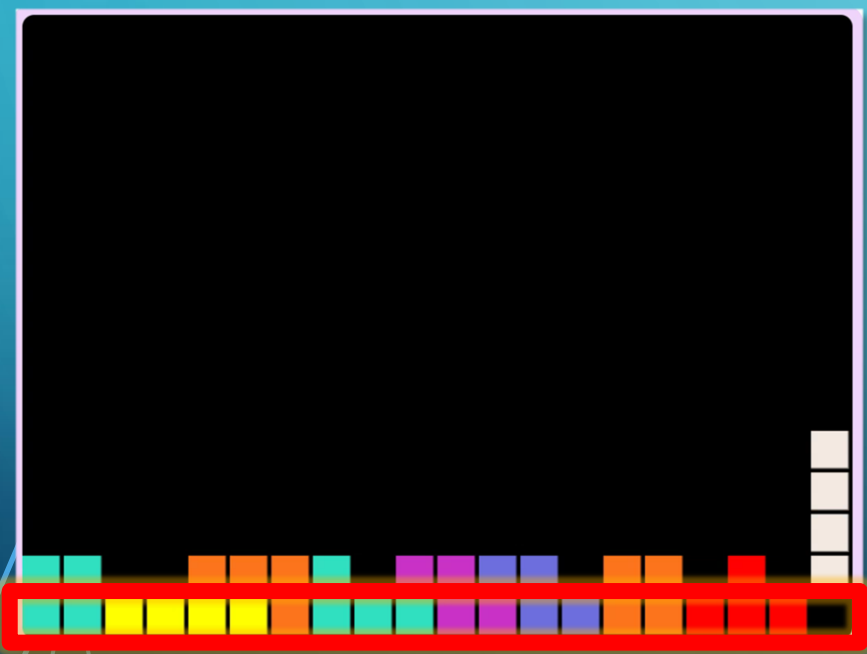
110704039 許甄芸

# 遊戲介面



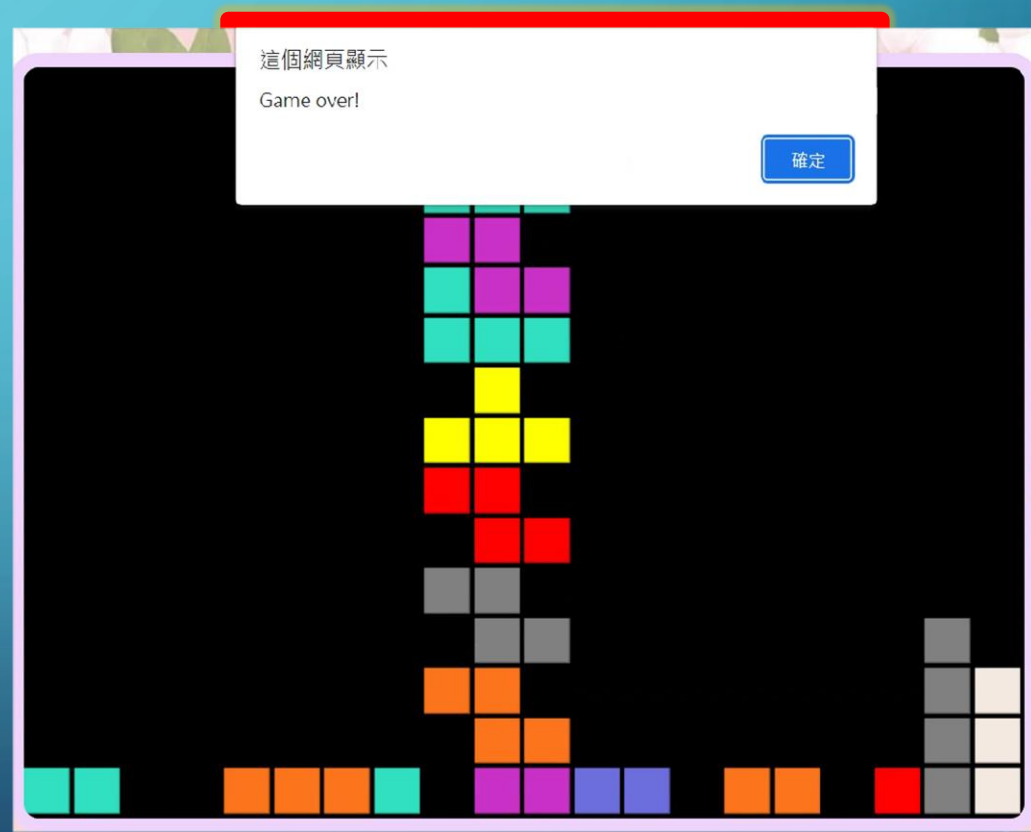
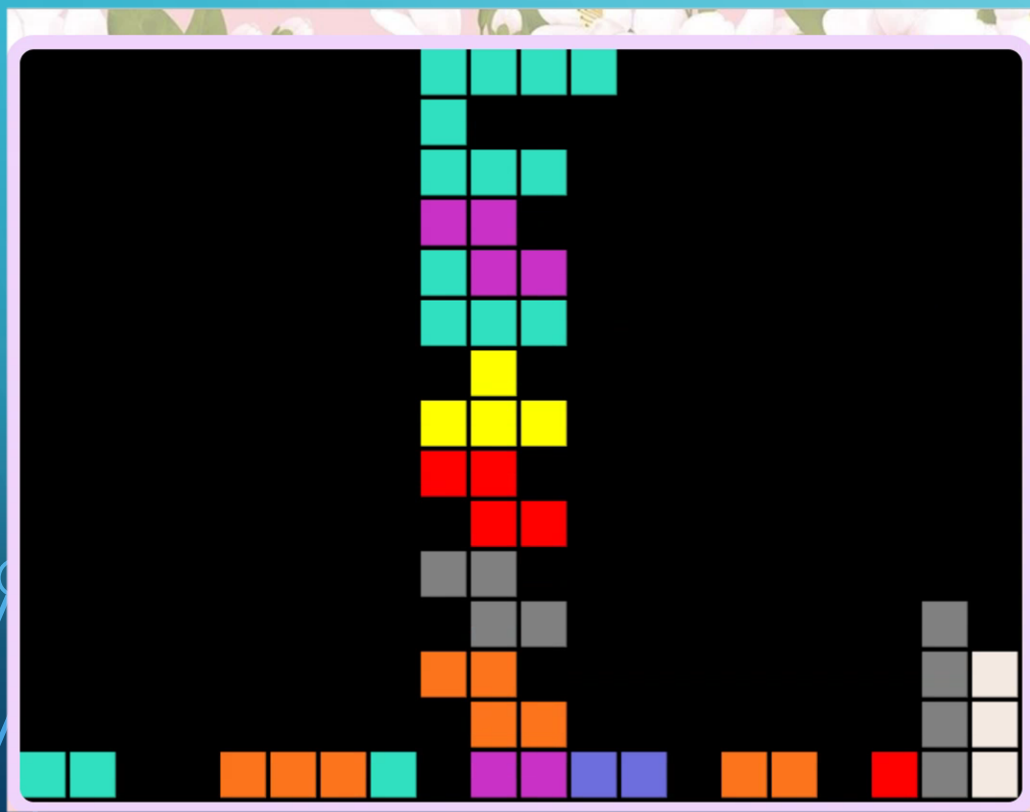
# 遊戲規則

- 空白鍵旋轉方塊、鍵盤←↓→分別為左移向下和右移
- 當方塊排滿一行就會消失，並且記分板score顯示加1



# 遊戲規則

- 當方塊觸碰到邊界沒有空間落下時，網頁跳出“Game over!”視窗



# 開發環境&包含檔案

i

裝置規格

裝置名稱

ZenBook

處理器

AMD Ryzen 7 4700U with Radeon Graphics

2.00 GHz

已安裝記憶體(RAM)

16.0 GB (15.4 GB 可用)

裝置識別碼

47164421-D124-4306-A936-510210351433

產品識別碼

00326-10000-00000-AA103

系統類型

64 位元作業系統，x64 型處理器

手寫筆與觸控

此顯示器不提供手寫筆或觸控式輸入功能

相關連結

網域或工作群組

系統保護

進階系統設定

Windows 規格

版本

Windows 11 家用版

版本

21H2

安裝於

2022/3/9

OS 組建

22000.675

體驗

Windows 功能體驗套件 1000.22000.675.0

Microsoft 服務合約

Microsoft 軟體授權條款

- 開發平台:Notepad++
- 瀏覽器:Google Chrome. Ink
- Term\_110704039.html
- Term\_110704039.js
- Term\_110704039.css
- 圖片 \*1(jpg檔)

# 使用到的函數

- gameStart()
- init()
- run()
- initBackground()
- initBlock()
- drawBlock()
- drawStaticBlock()
- createRandom()
- judgeCollision\_down()
- judgeCollision\_other()
- down\_speed\_up()
- move()
- up\_change\_direction()
- clearUnderBlock()
- gameover()

# BACKGROUND

- 使用canvas(畫布)

```
3 <head>
4   <meta charset="UTF-8">
5   <title>俄羅斯方塊</title>
6   <link rel = "stylesheet" type = "text/css" href = "Term_110704039.css">
7 </head>
8 <body>
9   <canvas width="700" height="525" style="border:10px solid #F0D4FC;border-radius: 20px;" id="tetris"></canvas>
10   <div class="info">
11     <h2>My term project game: Tetris</h2>
```

- 初始化地圖-fillstyle

```
44 // 初始化地圖
45 initBackground() {
```



# 方塊的生成

- 隨機數生成方塊類型和顏色-switch case

```
89 // 生成隨機數返回方塊類型或顏色類型
90 createRandom(type){
91     let temp = this.width/2-1
92
93     if (type == 'rBlock'){ //如果是方塊類型
94         this.num_block = Math.round(Math.random()*4+1)
```

```
118     if (type == 'rColor'){ //如果是顏色類型
119         let num_color = Math.round(Math.random()*8+1)
```



# 方塊碰撞的判斷

- 判斷方向為下的碰撞-for、if

```
153 // 判斷方塊之間是否碰撞(下)，以及變形時是否越過下邊界  
154 ❏judgeCollision_down() {
```

- 判斷方向為左右的碰撞-for、if

```
173 //判斷方塊之間是否碰撞(左右)，以及變形時是否越過左右邊界  
174 ❏judgeCollision_other(num) {
```

# 排滿一行的變化

- 清除排滿的那層方塊-for、if else

```
371 //一行滿了清空方塊，上面方塊Y坐標+1
372 clearUnderBlock() {
373     //刪除低層方塊
```

- 並且將分數加1

```
392 if (arr_row.length == this.width) {
393     //計算成績grade
394     this.grade++
```

- 將未排滿的方塊往下移

```
403 //讓上面的方塊往下掉一格
404 for(let i = 0; i < this.arr_store_color.length; i++){
405     if (this.arr_store_Y[i] < line_num) {
406         this.arr_store_Y[i] = this.arr_store_Y[i]+1
407     }
408 }
409 }
410 }
```

# 方向鍵功能函數

- 上下左右鍵控制及空白鍵旋轉-switch

```
427 //方向鍵功能函數  
428 document.onkeydown = (e) => {
```

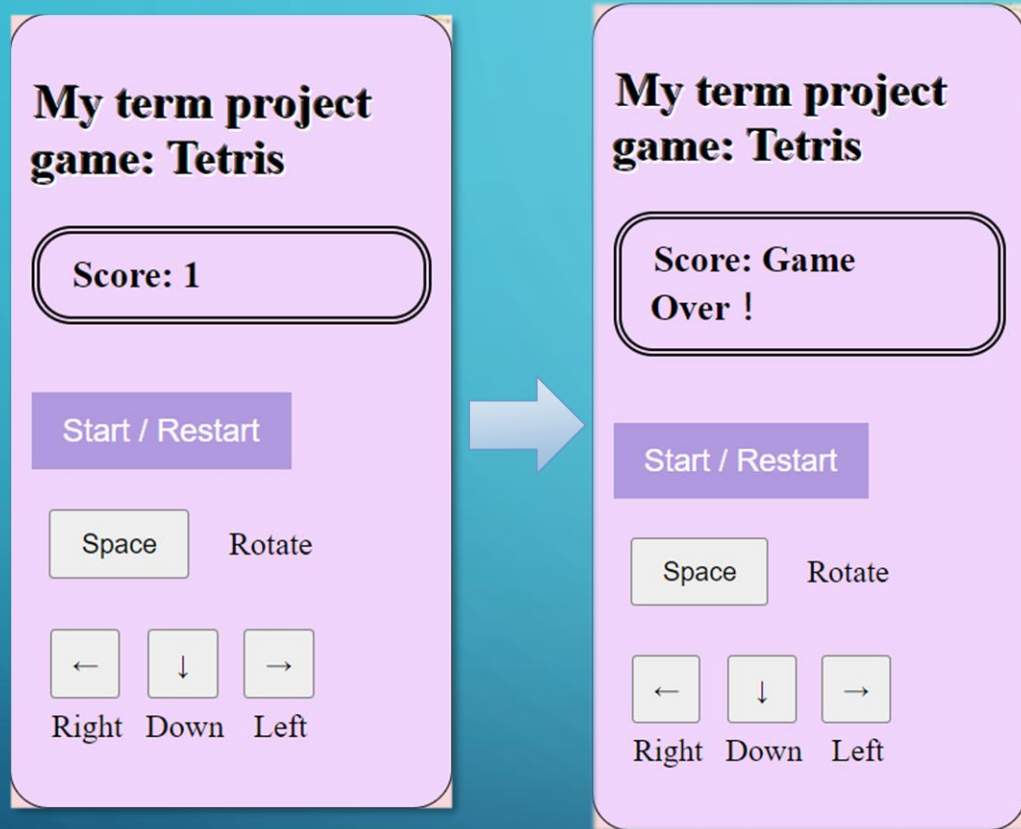
```
432     switch(e.keyCode) {
```

# 判斷遊戲結束

- 利用for、if判斷方塊是否觸碰上邊界

```
412 //判斷遊戲結束
413 gameover() {
414     for(let i=0; i < this.arr_store_X.length; i++){
415         if (this.arr_store_Y[i] == 0) {
416             clearInterval(this.ident);
417             this.over = true;
418             alert("Game over!");
419         }
420     }
421 }
422 }
423 }
```

# 美編排版



- 當遊戲結束計分顯示Game Over!
- H2字體陰影
- 按鍵特效
- Background-image



# 結論與心得

我從小時候就很喜歡玩俄羅斯方塊這個遊戲  
沒想到這次有機會自己用程式寫出這個遊戲  
但寫的過程充分地發現了自己的不足  
還有很多很多需要學習並且熟練的地方  
debug的過程真的很難很辛苦  
也請教了很多學長姊 真的很感謝他們  
最後終於讓程式完整並成功地跑出來後  
真的很有成就感非常的興奮  
而幫作品美編排版其實是最喜歡的部份  
許多版面設計都有加上我的小巧思  
總之真的是非常有趣的經驗~  
謝謝老師當初願意讓我加簽這門課  
讓我有這些實作經驗、學到了這麼多的技巧。



The background is a blue gradient. In the corners, there are white line-art illustrations of circuit boards or neural networks, with lines and small circles representing nodes.

THANKS FOR WATCHING