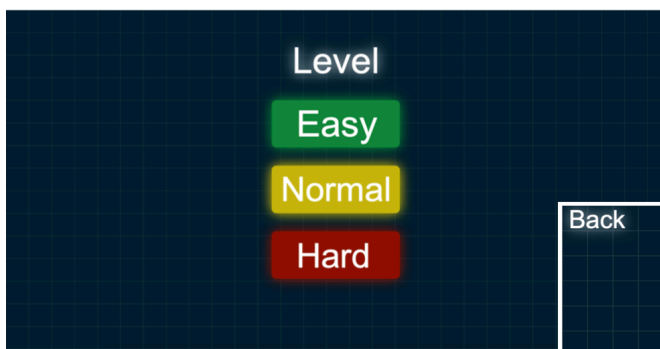


# 簡易戰機遊戲

## 系統功能/摘要

遊戲畫面：

### 介面



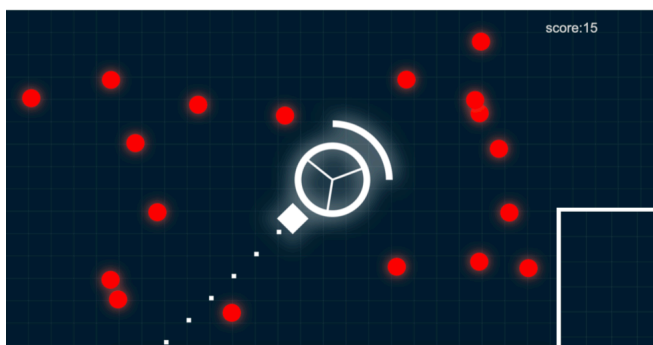
困難度選單

準備開始！

Back

Start

### 介面



遊戲畫面

Game Over !

Game Over

score:555

Restart

遊戲介面：

選擇難度 → 準備開始 → 遊戲 → 結算

## 遊戲規則：

畫面中央有一個固定位置的飛船，在固定的時間間隔下發射出子彈，可用滑鼠相對於飛船的方向來控制砲口的方向。畫面的外圍會生成許多紅色的敵人，敵人會朝著飛船飛去，若擊中船體則遊戲結束。擊毀每個敵人可以根據難度獲得5/10/15分。

## 道具：

- 綠色(Buff)：若用子彈擊中，則可以使船體升級，加快填彈速度。船體等級共有五級，每個等級的船體顏色皆不同！
- 藍色(Slow)：若用子彈擊中，則可以使所有敵人暫時緩速。
- 黃色(Bomb)：若用子彈擊中，則可以清除畫面中所有敵人。

## 系統開發平台

Visual Studio Code

## 程式說明

共三個檔案：

main.html(主網頁)、canvas.css、canvas.js(主程式)

## 程式說明：

物件：船體、子彈(物件陣列)、敵人(物件陣列)、三個道具

## 重要function：

chooseLevel(): 先觸發這個function，開始選擇難度畫面。

draw(): 每次執行完會遞迴觸發draw，一直畫出新的畫面狀態。

update(): 將time interval設成(1000/60)秒，更新所有物件的位置等狀態。

\*程式有許多細節(例如物件的移動、生成、判斷子彈、敵人擊中等)，就不一一解釋。

\*所有按鈕的設計都是判斷click時的滑鼠位置是否在按鈕框框內

## 程式碼：

```
J5 canvas.js X
Users > wei4r > Desktop > canvas > J5 canvas.js > ...
1  var canvas = document.getElementById("mycanvas")
2  var ctx = canvas.getContext("2d")
3  var ww = window.innerWidth,
4    wh = window.innerHeight
5  var time = 0;
6  var enemy_level=0
7  var gaming = 0
8  var score = 0
9  var pos_per_side = 3
10 var enemy_speed
11 var enemy_spawn_speed=[150,140,130,120,110,100,90,80,70]
12 var bullet_spawn_speed=[7,6,5,4,3]
13 var ship_color=["white","#F5E191","#F5AD57","#F57A57","#7E0802"]
14 var move_speed
15 var end = 0;
16 var bullets=[], spawn_pos=[], enemies=[]
17 var ship, level
18 var time_tag,time_tag_slow,time_tag_bomb
19 var ori_time_bomb,ori_time_slow,ori_time_buff
20 var state, game_int
21 > for(let i=0;i<=pos_per_side+1;i+=1){...
43 }//生成「會生成敵人的點」的位置陣列
44 class Ship{
45 >   constructor(args){...
52   }
53 >   draw(){...
98   }
99 >   update(){...
104  }
105 }
106 class Bullet{
107 >   constructor(args){...
118   }
119 >   draw(){...
125   }
126 >   update(){...
133   }
134 >   hit(enemy){...
141   }
142 >   level_up(){...
155   }
156 >   hit_slow(){...
168   }
169 >   hit_bomb(){...
183   }
184   }
185 class Enemy{
186 >   constructor(args){...
198   }
199 >   draw(){...
211   }
212 >   move(){...
229   }
230 >   collision(){...
240   }
241   }
242 class Buff{
243 >   constructor(args){...
255   }
256 >   draw(){...
277   }
278 >   move(){...
295   }
296 >   collision(){...
304   }
305   }
306 class Slow{
307 >   constructor(args){...
319   }
320 >   draw(){...
358   }
359 >   move(){...
374   }
375 >   collision(){...
383   }
384   }
385 class Bomb{
386 >   constructor(args){...
398   }
399 >   draw(){...
427   }
428 >   move(){...
443   }
444 >   collision(){...
452   }
453 }
```

```

454 > function RandomInt(start,end){~
457 }//工具程式：隨機生成start~end中的一個整數
458 function start(level){
459     state="start"
460 >     switch(level){~
473 }//設定各難度參數
474     gameStart()
475 }
476 function gameStart(){
477     end=0
478     drawBG()
479     drawStart()
480     canvas.addEventListener('click', function(evt) {
481         if(!gaming){
482             if(!end){ //start
483                 if ((MousePos.x > -100) && (MousePos.x < 100) && (MousePos.y < 50) && (MousePos.y > -50)){
484                     setup()
485                     game_int = setInterval(update,1000/60);
486                     requestAnimationFrame(draw)
487                 }
488                 else if ((MousePos.x > -(ww/2))) && (MousePos.x < -(ww/2)+150) && (MousePos.y < -(wh/2)+60) && (MousePos.y > -(wh/2))){
489                     chooseLevel()
490                 }
491             }
492         }
493     }, false);
494 }
495 }
496 > function GameOver(){~
540 }//畫出、執行GameOver畫面
541 > function drawStart(){~
564 }//畫出「Start」的圖案
565 > function drawBG(){~
581 }//畫出背景
582 > function drawScore(){~
589 }//畫出分數
590 > function setup(){~
611 }//初始化各個參數
612 > function drawLevel(){~
667 }//畫出選擇難度畫面
668 function update(){
669     if(!end){
670 >         if((time+1)%500==0){~
674 }         //每過固定時間升級敵人
675 >         if(time%bullet_spawn_speed[ship.level-1]==0){~
685 }         //bullet         //生成子彈
686 >         if((time+1)%enemy_spawn_speed[enemy_level]==0){~
705 }         //respawn enemy //重新生成敵人
706 >         if((time+1)%500==0){~
717 }         //spawn buff         //生成、移動藍色道具
718 >         if((time+1)%700==0){~
726 }         //slow down         //生成、移動藍色道具
727 >         if((time+1)%900==0){~
736 }         //bomb         //生成、移動黃色道具
737         slow.move()
738         slow.collison()
739         bomb.move()
740         bomb.collison()
741         buff.move()
742         buff.collison()
743         ship.update()
744 >         bullets.forEach(bullet => {~
752 }         //check hit 確認子彈是否擊中任何物體
753 >         enemies.forEach(enemy=>{~
758 }
759         time++
760     }
761     else{
762         clearInterval(game_int)
763     }
764 }
765 function draw(){
766     if(end==0){
767         drawBG()
768         drawScore()
769         ship.draw()
770         bullets.forEach(bullet =>{
771             bullet.draw()
772         });
773         enemies.forEach(enemy=>{
774             enemy.draw()
775         })
776         buff.draw()
777         if((slow.alive==1||time<=time_tag_slow)&&time>10)
778             slow.draw()
779         bomb.draw()

```

```

780     requestAnimationFrame(draw)
781   }
782   else if(end==2){
783     drawPass()
784     window.cancelAnimationFrame(draw)
785   }
786 }
787 function chooseLevel(){
788   state="level"
789   drawLevel()
790   canvas.addEventListener('click', function(evt) {
791 >     if(state=="level"){...
792   }
793   }, false);
794 }
795
796 var buff = new Buff
797 var slow = new Slow
798 var bomb = new Bomb
799 > ship=new Ship({-
800 })//建立船體
801
802 spawn_pos.forEach(pos => {
803   var enemy = new Enemy
804   enemies.push(enemy)
805 })//建立敵人
806 > var MousePos={-
807 }
808
809 canvas.addEventListener("mousemove",function(evt){
810   MousePos.x=evt.x-ww/2
811   MousePos.y=evt.y-wh/2
812 })
813
814 chooseLevel()      //觸發遊戲

```

## 結論與心得：

整個程式開發了約兩週，寫遊戲的過程中我算是非常享受的，雖然進度非常緩慢，常一個功能要寫好幾天，但是開發完成之後非常連我自己都會忍不住連續玩好幾十分鐘。這次的project把重心放在遊戲機制的平衡以及遊玩的體驗，便決定把所有物件都用幾何圖形來完成，沒有另外花費心力設計、引用客製化的icon，成為了一個簡約典雅的小遊戲。

我覺得最困難的部分就是讓擊中道具後產生一個漸漸消失的文字(ex. Level up) 後來慢慢摸索之後是另外用一個參數來記錄預計要在何時讓文字完全消失的時間點，在這個時間內讓透明度從1漸漸降為0。

可改進的部分與可增加的功能：整體icon可更精緻化、遊戲得分線上排行榜。