



JavaScript Lab

RPG極簡雛形製作

Ryan Chung

A large yellow square containing the letters 'JS' in a bold, black, sans-serif font, representing the JavaScript logo.



需求

- 做一個九宮格可以移動的遊戲雛形
- 可以移動主角，但不會走出範圍
- 可以判斷遇到人、障礙物
- 可以判斷走到終點
- 主角的圖案跟方向一致

我要去右下角



抵達終點！



檔案

- index.html：主要顯示頁面
- main.js：主要程式運作
- style.css：網頁樣式檔
- images：存放圖片的資料夾
(<https://ppt.cc/fN2Eox>)



index.html

- 標題、canvas、div、p

```
1  <!DOCTYPE html>
2  <html>
3      <head>
4          <meta charset="utf-8">
5          <title>Simple RPG</title>
6          <link rel="stylesheet" href="style.css">
7          <script src="https://code.jquery.com/jquery-3.4.1.min.js"></script>
8          <script src="main.js"></script>
9      </head>
10     <body>
11         <h1>我要去右下角</h1>
12         <div>
13             <canvas id="myCanvas" width="600" height="600"></canvas>
14             <div id="talkBox"></p>
15         </div>
16     </body>
17 </html>
```



style.css

```
1 ▼ body{
2     text-align: center;
3 }
4
5 ▼ canvas{
6     border: 2px solid black;
7     background-image: url(images/grasses68.gif);
8 }
9
10 ▼ #talkBox{
11     height: 20px;
12     color: brown;
13     font-size: 26px;
14     font-weight: bold;
15 }
```



main.js

- 全域變數宣告、事件觸發

JS main.js > ...

```
1  //mapArray : 決定地圖中每個格子的元素
2  //ctx : HTML5 Canvas使用
3  //currentImgMainX, currentImgMainY : 決定主角所在座標
4  //imgMountain, imgMain, imgEnemy : 障礙物、主角、敵人的圖片物件
5
6  let mapArray, ctx, currentImgMainX, currentImgMainY;
7  let imgMountain, imgMain, imgEnemy;
8
9  //當網頁元件載入完成要做的事情
10 > $(document).ready(function(){ ...
39   });
40  //當有人按下按鍵後要做的事情
41 > $(document).keydown(function(event){ ...
102  });
```



main.js

- 設定地形、擺上主角

```
8
9 //當網頁元件載入完成要做的事情
10 $(document).ready(function(){
11     //遊戲地圖
12     //0:可走、1:障礙、2:終點、3:敵人
13     mapArray = [0,1,1,0,0,0,3,1,2];
14     ctx = $("#myCanvas")[0].getContext("2d");
15
16     //擺主角
17     imgMain = new Image();
18     imgMain.src = "images/spriteSheet.png";
19     currentImgMainX = 0;
20     currentImgMainY = 0;
21     imgMain.onload = function(){
22         ctx.drawImage(imgMain,0,0,80,130,currentImgMainX, currentImgMainY, 200, 200);
23     };
```



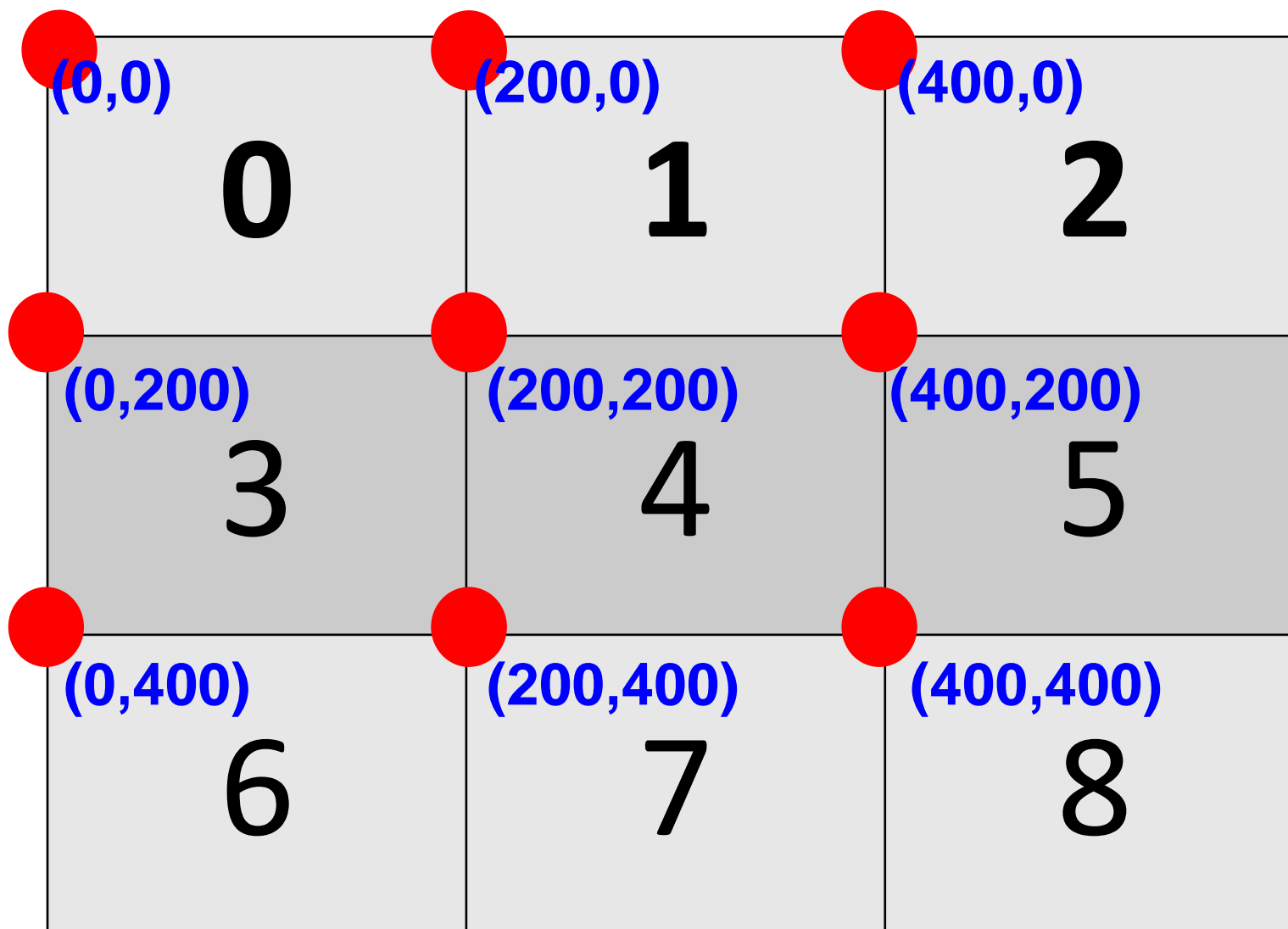
main.js

- 擺上障礙物、敵人

```
24 //擺障礙物與敵人
25 imgMountain = new Image();
26 imgMountain.src = "images/material.png";
27 imgEnemy = new Image();
28 imgEnemy.src = "images/Enemy.png";
29 imgMountain.onload = function(){
30     imgEnemy.onload = function(){
31         for(let x in mapArray){
32             if(mapArray[x]==1){
33                 ctx.drawImage(imgMountain,32,65,32,32,x%3*200, Math.floor(x/3)*200, 200, 200);
34             }else if(mapArray[x]==3){
35                 ctx.drawImage(imgEnemy,7,40,104,135,x%3*200, Math.floor(x/3)*200, 200, 200);
36             }
37         }
38     }
39 };
40
41 });
```




編號與座標的對應





main.js

```
42 //當有人按下按鍵後要做的事情
43 $(document).keydown(function(event){
44     let targetImgMainX,targetImgMainY, targetBlock, cutImagePositionX;
45     // 主角即將要移動過去的目標位置 主角即將要移動過去的那一格編號 依據主角朝向什麼方向而決定的圖片
46     event.preventDefault();
47     //避免點擊鍵盤出現瀏覽器的其他行為，例如捲動、放大、換頁...
48     //根據使用者按鍵指示，對應計算目標位置、主角新的方向圖片
49     switch(event.code){
50         case "ArrowLeft": //向左
51             targetImgMainX = currentImgMainX-200;
52             targetImgMainY = currentImgMainY;
53             cutImagePositionX = 175;
54             break;
55         case "ArrowUp": //向上
56             targetImgMainX = currentImgMainX;
57             targetImgMainY = currentImgMainY-200;
58             cutImagePositionX = 355;
59             break;
60         case "ArrowRight": //向右
61             targetImgMainX = currentImgMainX+200;
62             targetImgMainY = currentImgMainY;
63             cutImagePositionX = 540;
64             break;
65         case "ArrowDown": //向下
66             targetImgMainX = currentImgMainX;
67             targetImgMainY = currentImgMainY+200;
68             cutImagePositionX = 0;
69             break;
70         default: //其他按鍵不回應
71             return;
72     }
```

- 按鍵判斷
- 座標設定
- 排除其他狀況



main.js

- 判斷目標位置是否在邊界內
- 判斷目標位置是否有障礙物/敵人

```
73 //在邊界內
74 if(targetImgMainX<=400 && targetImgMainX>=0 &&
75     targetImgMainY<=400 && targetImgMainY>=0){
76     targetBlock = targetImgMainX/200+targetImgMainY/200*3;
77 }else{ //超出邊界
78     targetBlock = -1;
79 }
80 //清除主角原本所在位置
81 ctx.clearRect(currentImgMainX, currentImgMainY, 200, 200);
82 if(targetBlock == -1 || mapArray[targetBlock]==1 || mapArray[targetBlock]==3){
83     //所有異常(出界、遇到敵人、遇到障礙物都不動)
84 }else{ //正常情況就設定新的位置
85     $("#talkBox").empty();
86     currentImgMainX = targetImgMainX;
87     currentImgMainY = targetImgMainY;
88 }
89 //在新的位置上畫上主角
90 ctx.drawImage(imgMain,cutImagePositionX,0,80,130,currentImgMainX, currentImgMainY, 200, 200);
```



main.js

- 依據目標位置內容，顯示提示訊息於下方

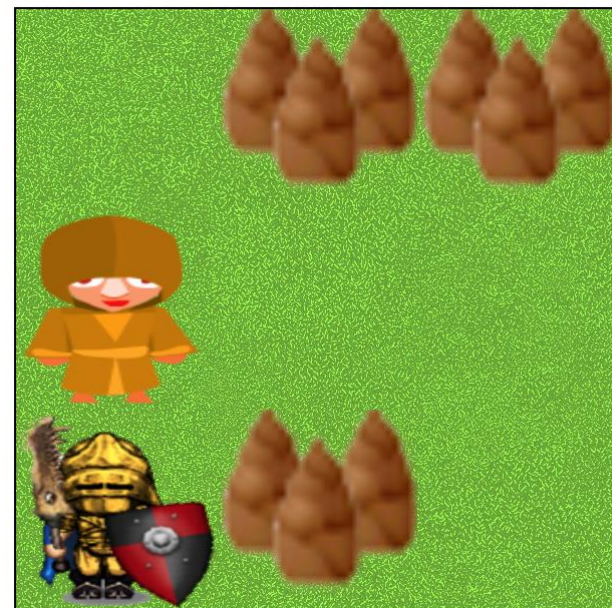
```
91      //對應用文字顯示狀態
92      switch(mapArray[targetBlock]){
93          case undefined:
94              $("#talkBox").text("邊界");
95              break;
96          case 1:
97              $("#talkBox").text("有山");
98              break;
99          case 2:
100              $("#talkBox").text("抵達終點");
101              break;
102          case 3:
103              $("#talkBox").text("哈摟");
104              break;
105      }
106
107  });
```



總算完成！進行測試

- 主角是否會走出邊界？
- 是否遇到障礙物、敵人會無法走，並出現訊息
- 是否有判斷抵達終點？
- 是否有依方向轉頭？

我要去右下角



嗨~



Recap

- HTML
 - canvas 、 div 、 id
- JavaScript
 - image load 、 canvas draw image 、 cut image
 - for ... in 、 array 、 switch
 - keydown 、 preventDefault
- 資料檔案
 - 圖片



Lab

- 可以16 x 16嗎？
- 一個物品可以跨越多格嗎？
- 如何寫出小精靈？
- 如何寫出OOXX遊戲？
- 加一顆按鈕攻擊敵人

我要去右下角



哈搜

揍他