

# JavaScript Lab

## RPG 極簡雛形製作

Ryan Chung



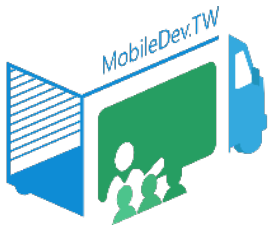
# 需求

- 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



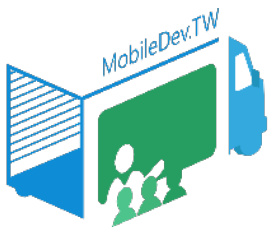
抵達終點！



# 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

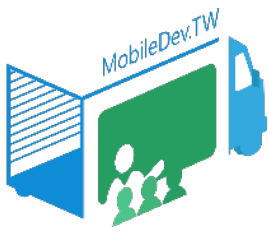
<https://github.com/ryanchung403/images/tree/main/simple-rpg>



# index.html

- 標題、canvas、div、p

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Simple RPG</title>
    <link rel="stylesheet" href="style.css">
  </head>
  <body>
    <h1>我要去右下角</h1>
    <div>
      <canvas id="myCanvas" width="600" height="600"></canvas>
      <p id="talkBox"></p>
    </div>
    <script src="https://unpkg.com/jquery"></script>
    <script src="main.js"></script>
  </body>
</html>
```

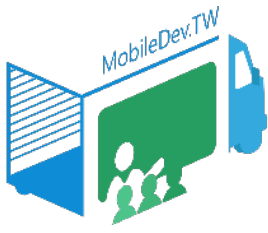


# style.css

```
body{
    text-align: center;
}

canvas{
    border: 2px solid black;
    background-image: url(images/grasses68.gif);
}

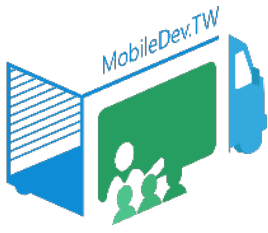
#talkBox{
    height: 20px;
    color: brown;
    font-size: 26px;
    font-weight: bold;
}
```



# 新增 jsconfig.json

- 幫助產生jQuery的語法提示

```
{  
  "typeAcquisition": {  
    "include": [  
      "jquery"  
    ]  
  }  
}
```

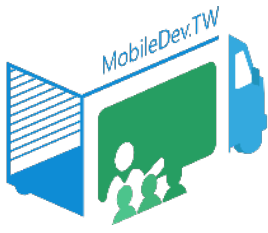


# main.js

- 全域變數宣告、事件觸發

```
let mapArray, ctx, currentImgMain;  
let imgMountain, imgMain, imgEnemy;  
//mapArray - 決定地圖中每個格子的元素  
//ctx - HTML5 Canvas用  
//currentImgMainX, currentImgMainY - 決定主角所在座標  
//imgMountain, imgMain, imgEnemy - 障礙物, 主角, 敵人的圖片物件  
const gridLength = 200;  
//網頁載入完成後初始化動作  
$(function(){  
  
});  
  
//處理使用者按下按鍵  
$(document).on("keydown",function(event){  
  
});
```





# main.js

- 設定地形、擺上主角

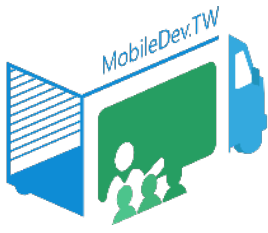
//網頁載入完成後初始化動作

```
$(function(){
  mapArray = [ //0-可走,1-障礙,2-終點,3-敵人
    [0,1,1],
    [0,0,0],
    [3,1,2]
  ];
  ctx = $("#myCanvas")[0].getContext("2d");

  imgMain = new Image();
  imgMain.src = "images/spriteSheet.png";
  currentImgMain = {
    "x":0,
    "y":0
  };

  imgMain.onload = function(){
    ctx.drawImage(imgMain, 0,0,80,130,currentImgMain.x,currentImgMain.y,gridLength,gridLength);
  };
});
```

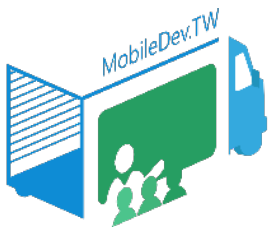




# main.js

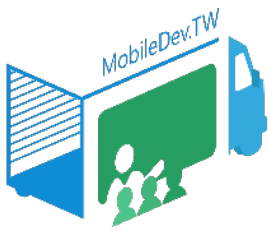
- 擺上障礙物、敵人

```
imgMountain = new Image();
imgMountain.src = "images/material.png";
imgEnemy = new Image();
imgEnemy.src = "images/Enemy.png";
imgMountain.onload = function(){
    imgEnemy.onload = function(){
        for(var x in mapArray){
            for(var y in mapArray[x]){
                if(mapArray[x][y]==1){
                    ctx.drawImage(imgMountain, 32,65,32,32,y*gridLength,x*gridLength,gridLength,gridLength);
                }else if(mapArray[x][y]==3){
                    ctx.drawImage(imgEnemy, 7,40,104,135,y*gridLength,x*gridLength,gridLength,gridLength);
                }
            }
        }
    }
}
});
```



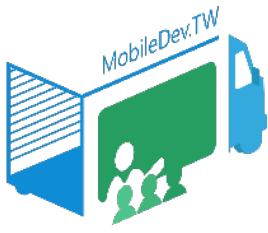
# 2維陣列與座標的對應

<b>(0,0)</b> [0][0]	<b>(200,0)</b> [0][1]	<b>(400,0)</b> [0][2]
<b>(0,200)</b> [1][0]	<b>(200,200)</b> [1][1]	<b>(400,200)</b> [1][2]
<b>(0,400)</b> [2][0]	<b>(200,400)</b> [2][1]	<b>(400,400)</b> [2][2]



# main.js

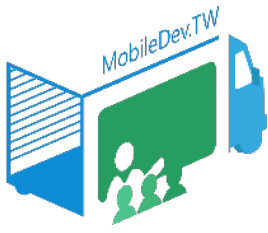
- 按鍵判斷
- 座標設定
- 排除其他狀況



# main.js

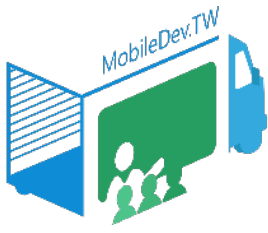
//處理使用者按下按鍵

```
$(document).on("keydown",function(event){  
    let targetImg, targetBlock, cutImagePositionX;  
    //cutImagePositionX - 決定主角臉朝向哪個方向  
    targetImg = { //主角的目標座標  
        "x":-1,  
        "y":-1  
    };  
    targetBlock = { //主角的目標(對應2維陣列)  
        "x":-1,  
        "y":-1  
    }  
  
    event.preventDefault();  
    //避免鍵盤預設行為發生，如捲動/放大/換頁...  
    //判斷使用者按下什麼並推算目標座標
```



# main.js

```
switch(event.code){
  case "ArrowLeft":
    targetImg.x = currentImgMain.x - gridLength;
    targetImg.y = currentImgMain.y;
    cutImagePositionX = 175;//臉朝左
    break;
  case "ArrowUp":
    targetImg.x = currentImgMain.x;
    targetImg.y = currentImgMain.y - gridLength;
    cutImagePositionX = 355;//臉朝上
    break;
  case "ArrowRight":
    targetImg.x = currentImgMain.x + gridLength;
    targetImg.y = currentImgMain.y;
    cutImagePositionX = 540;//臉朝右
    break;
  case "ArrowDown":
    targetImg.x = currentImgMain.x;
    targetImg.y = currentImgMain.y + gridLength;
    cutImagePositionX = 0;//臉朝下
    break;
  default://其他按鍵不處理
    return;
}
```



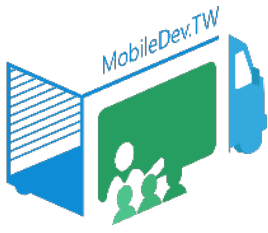
# main.js

//確認目標位置不會超過地圖

```
if(targetImg.x<=400 && targetImg.x>=0 && targetImg.y<=400 && targetImg.y>=0){  
    targetBlock.x = targetImg.y / gridLength;  
    targetBlock.y = targetImg.x / gridLength;  
}else{  
    targetBlock.x = -1;  
    targetBlock.y = -1;  
}
```

//清空主角原本所在的位置

```
ctx.clearRect(currentImgMain.x, currentImgMain.y, gridLength, gridLength);
```



# main.js

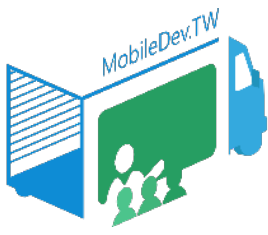
```
if(targetBlock.x!=-1 && targetBlock.y!=-1){

    switch(mapArray[targetBlock.x][targetBlock.y]){
        case 0: // 一般道路(可移動)
            $("#talkBox").text("");
            currentImgMain.x = targetImg.x;
            currentImgMain.y = targetImg.y;
            break;
        case 1: // 有障礙物(不可移動)
            $("#talkBox").text("有山");
            break;
        case 2: // 終點(可移動)
            $("#talkBox").text("抵達終點");
            currentImgMain.x = targetImg.x;
            currentImgMain.y = targetImg.y;
            break;
        case 3: // 敵人(不可移動)
            $("#talkBox").text("哈摟");
            break;
    }
}else{
    $("#talkBox").text("邊界");
}

//重新繪製主角
ctx.drawImage(imgMain, cutImagePositionX,0,80,130,currentImgMain.x,currentImgMain.y,gridLength,gridLength);
});
```

<https://gist.github.com/ryanchung403/4f1b9fb39d2be48db6607bdb1d3683>





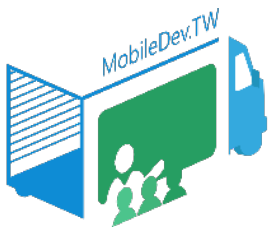
# 總算完成！進行測試

- 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



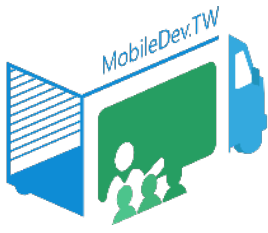
嗨~



# 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 1

- 參考下列程式碼，優化圖片載入流程

<https://gist.github.com/ryanchung403/09bacebfca6c0b4e9f87023d3a5da419>

Source : html5canvastutorials

## Lab 2

- 修改地圖從3x3變成8x6，每格大小從200改為100
- 在地圖中增加一個番茄，經過之後可以吃掉番茄
- 注意地圖的邊界判斷，終點維持在右下角

我要去右下角

