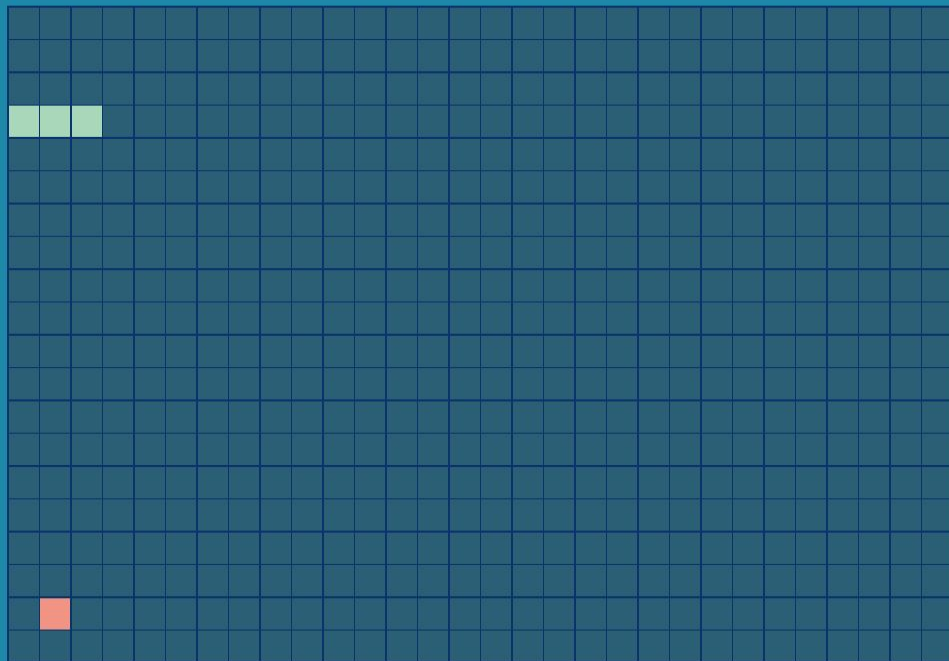


貪吃蛇小遊戲

0615336 高登泰

貪吃蛇小遊戲

0615336 高登泰



開始遊戲

重新開始

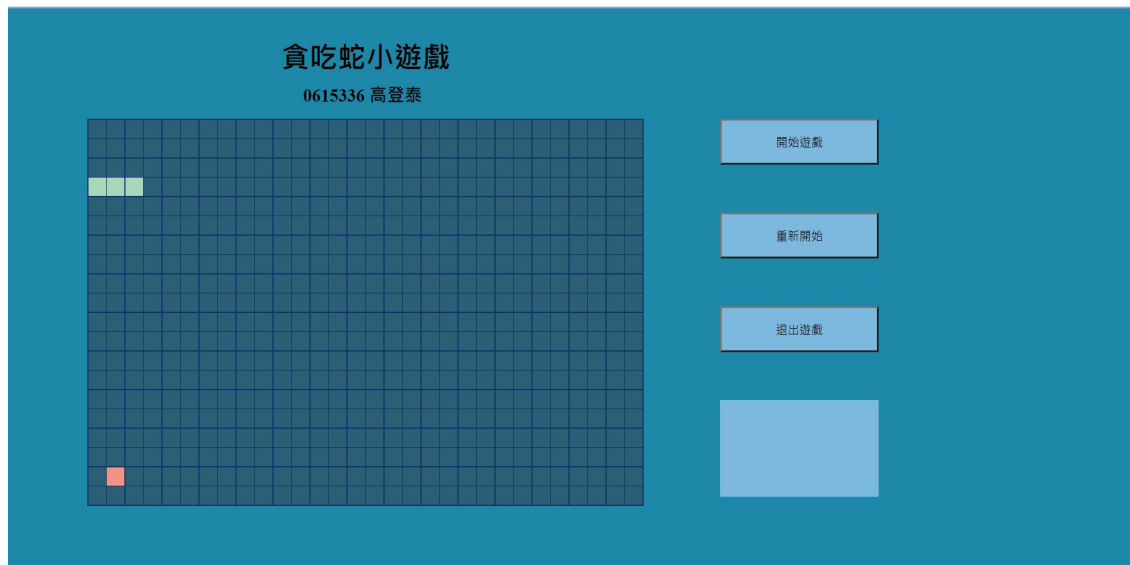
退出遊戲



預計的功能：

- 符合貪吃蛇遊戲的規則：吃食物會增加自身長度、撞到邊界會遊戲結束。
- 符合貪吃蛇遊戲的貪吃蛇運行規則：利用鍵盤上下左右鍵控制方向，並長按按鍵會讓貪吃蛇加速。
- 以開始鍵做遊戲啟動裝置。
- (未來增加)Reset按鈕。
- (未來增加)退出遊戲(頁面)按鈕。
- (未來增加)遊戲背景音樂。
- (未來增加)自訂遊戲難度：貪吃蛇的速度。
- (未來增加)分數機制：吃食物會在旁邊資訊欄顯示分數。
- (Debug)程式優化與介面美化。

- 開發環境：Visual Studio Code
- 開發語言：HTML、CSS、JavaScript。



HTML程式分為兩個部分

1. 第一部分是頁面左邊的標題欄與地圖。 Ps. 地圖方格、蛇以及食物在Js檔案產生。

```
<div id="container">
  <h1 id="h1">貪吃蛇小遊戲</h1>
  <h3 id="h3">0615336 高登泰</h3>
  <div id="map">
    <table id="mapBg"> </table>
  </div>
</div>
```

2. 第二部分是右邊的控制按鈕與顯示頁面。

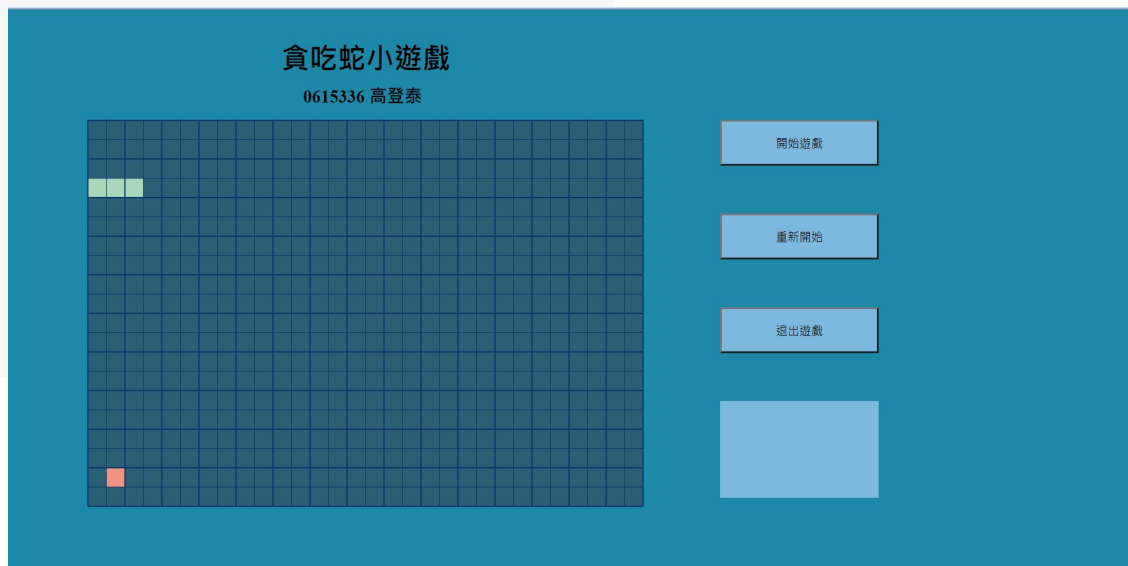
```
<div id="control">
  <button id="start">開始遊戲</button>
  <button id="reStarting">重新開始</button>
  <button id="drop">退出遊戲</button>
  <div id="gameOver"></div>
</div>
```



1. 製造地圖格子：製造600格框框(td)並加到地圖(mapBg的table)裡。

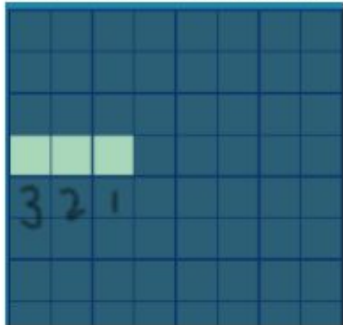
```
let mapBg = document.getElementById("mapBg");

function makeMap(){
  for (let row=0; row<20 ;row++){
    let Tr = document.createElement('tr');
    for(let col=0; col<30 ;col++){
      let Td = document.createElement('td');
      Tr.appendChild(Td);
      // 將td加到tr裡
    }
    mapBg.appendChild(Tr);
    // 將td加到mapBg的table裡
  }
}
makeMap();
```



2. 製造一開始的蛇：

- 蛇的結構是三個方塊，並放在一個名為snakeBody的Array裡。
- 蛇的位置用各個方塊的position，由於後面蛇的移動要用到各格方塊的位置，所以建snake.posi。
- 由於蛇頭、蛇身、蛇尾的排序是由右至左，所以用for迴圈時，用 `for(let i=2; i>=0 ;i--)`。



```
let snakeBody =[];

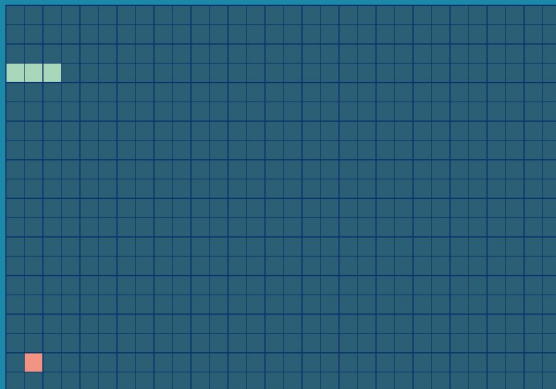
let map = document.getElementById('map');

function makeSnake(){
  for(let i=2; i>=0 ;i--){
    let snake = document.createElement('div');
    snake.className = "snake";
    //並列的小蛇
    snake.style.left = (20*i + (i+1) )+"px";
    map.appendChild(snake);
    snake.posi={           //第一條三格蛇的位置
      left:20*i + (i+1),
      top:64,
    };
    snakeBody.push(snake);
  }
}

makeSnake();
```

貪吃蛇小遊戲

0615336 高登泰



開始遊戲

重新開始

退出遊戲

3. 製造食物

- 食物的規則：不能超過四個邊界、不能跟蛇重疊。
- 利用 `parseInt(Math.random()*30)` 產生在邊界裡隨機的座標位置。
- 利用 `snakeBody[k].offsetLeft == left` 判斷是否跟蛇有重疊。

```
let bFlag = true;
let food;

function makefood(){
  do {
    //隨機產生食物座標
    //食物的左邊座標
    let left = parseInt(Math.random()*30);
    left = left*21 +1;
    //食物的上面座標
    let top = parseInt(Math.random()*20);
    top = top*21 +1;

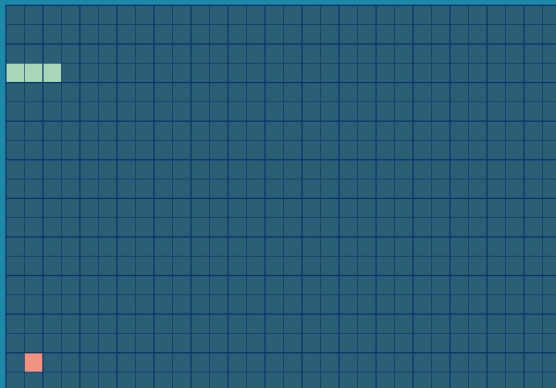
    //判斷是否跟蛇有重疊
    // .offsetLeft 左邊的長度
    for(var k=0 ;k<snakeBody.length;k++){
      if (snakeBody[k].offsetLeft == left &&
          snakeBody[k].offsetTop == top){
        bFlag = false;
      }
    }

    food = document.createElement("div");
    food.className = "food";
    food.style.left = left + 'px';
    food.style.top = top + 'px';

    food.posi={
      left:left,
      top:top
    };
    map.appendChild(food);
  }while(!bFlag);
}
makefood();
```

貪吃蛇小遊戲

0615336 高登泰



開始遊戲

重新開始

退出遊戲

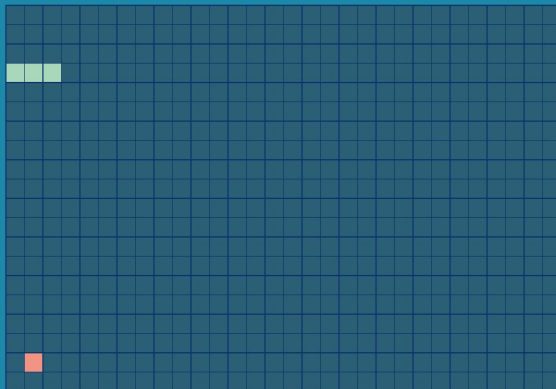
4.設計上下左右鍵

- 按鍵盤上下左右鍵->決定蛇的移動方向。
- 左上右下的key值:37、38、39、40。
- 往上走的蛇不能按下鍵，其他方向亦同。

```
let direction = "right";  
// 鍵盤按上下左右鍵 -> 決定蛇移動的方向  
document.onkeydown = function(event){  
    event =event||window.event;  
    var key = event.which || event.keyCode;  
    // 左上右下的key:37、38、39、40  
    // 往上走的蛇不能按下鍵  
    if(key==40){  
        if(direction != "up"){  
            direction="down";  
        }  
    }  
    }else if(key==39){  
        if(direction != "left"){  
            direction="right";  
        }  
    }  
    }else if(key==38){  
        if(direction != "down"){  
            direction="up";  
        }  
    }  
    }else if(key==37){  
        if(direction != "right"){  
            direction="left";  
        }  
    }  
}  
snakeMove();  
};
```

貪吃蛇小遊戲

0615336 高登泰



開始遊戲

重新開始

退出遊戲

5. 蛇的移動

- 鍵盤只會影響，蛇頭的轉向。
- 其他的蛇身會往前移動到前一格的位置。
- 因為是移動到前一格沒有移動前的位置，所以利用 `snakeBody[j].posi` 儲存位置。

```
function snakeMove(){
  let snakeHead =snakeBody[0]; //蛇頭

  //鍵盤只會影響蛇頭轉向
  if(direction == "down"){ //蛇頭轉向
    snakeHead.style.top = snakeHead.offsetTop+21+"px";
  }else if(direction == "right"){
    snakeHead.style.left = snakeHead.offsetLeft+21+"px";
  }else if(direction == "up"){
    snakeHead.style.top = snakeHead.offsetTop-21+"px";
  }else if(direction == "left"){
    snakeHead.style.left = snakeHead.offsetLeft-21+"px";
  }
}
```

//鍵盤只會影響蛇頭轉向，其餘的會跟著前一格動
//蛇身要移到蛇頭還沒移動前的位置

```
for (var j=1 ;j<snakeBody.length;j++){
  snakeBody[j].style.left =
    snakeBody[j-1].posi.left + "px";
  snakeBody[j].style.top =
    snakeBody[j-1].posi.top + "px";
}
```

- 蛇吃到食物後，食物會變成蛇的第二格方塊，並且變成蛇的property。
- 放到第二格位置，所以利用 `snakeBody.splice(1,0,food)`。
- 食物變成蛇身體後，需要再更新每隔蛇格子的位子。

```
//延續上一part的函數內容
//蛇吃食物
if (snakeHead.offsetLeft == food.offsetLeft &&
    snakeHead.offsetTop == food.offsetTop){
  food.className="snake";
  // snakeBody.push(food); //食物應該放在第二節
  snakeBody.splice(1,0,food) //在第一個索引值刪掉0個元素並插入food
  makefood();
}

//更新第一輪後蛇的位子
for(let i=0; i<snakeBody.length;i++){
  snakeBody[i].posi.left = snakeBody[i].offsetLeft;
  snakeBody[i].posi.top = snakeBody[i].offsetTop;
}
```

- 遊戲結束的兩個原因：撞到自己跟撞到邊界->結束計數器與顯示遊戲結束。
- 撞到自己：如果蛇頭與任一蛇身體的位置是一樣的，代表撞到
- 撞到邊界：蛇頭碰到邊界外的位置。

```
//延續上一part的函數內容
//蛇吃到自己；最多只會吃到第四個
for (let i=4 ;i<snakeBody.length ; i++){
  if (snakeHead.offsetTop == snakeBody[i].offsetTop &&
      snakeHead.offsetLeft == snakeBody[i].offsetLeft){
    gameOver = document.getElementById("gameOver");
  }
}
```

6. 蛇的移動

- 按開始按鈕觸動計數器
- 蛇每0.3秒會移動一次 `snakeMove()` 函數。

```
//蛇開始動
let start = document.getElementById("start");
let timer;
//定時器，每 3 秒移動
start.onclick = function(){
    timer = setInterval(function(){
        snakeMove();

    },300);
};
```



貪吃蛇小遊戲

Demo