



# Memory matching game 匯報(第三組)

組長:黃溢懿

組員:李皓瀧, 林震東, 黃顥錡, 陳尚樂

# 組員



**Autumn White**  
AutumnWhiteQiuBai

5A21 Ricky Wong



**Kyle**  
kylec1114

5B02 Kyle Chan



**Wong Yat Yi, Jerry**  
Jerrywongyy

5C22 Jerry Wong



**Ghost**  
ghostsky911

5D09 Sky Lee



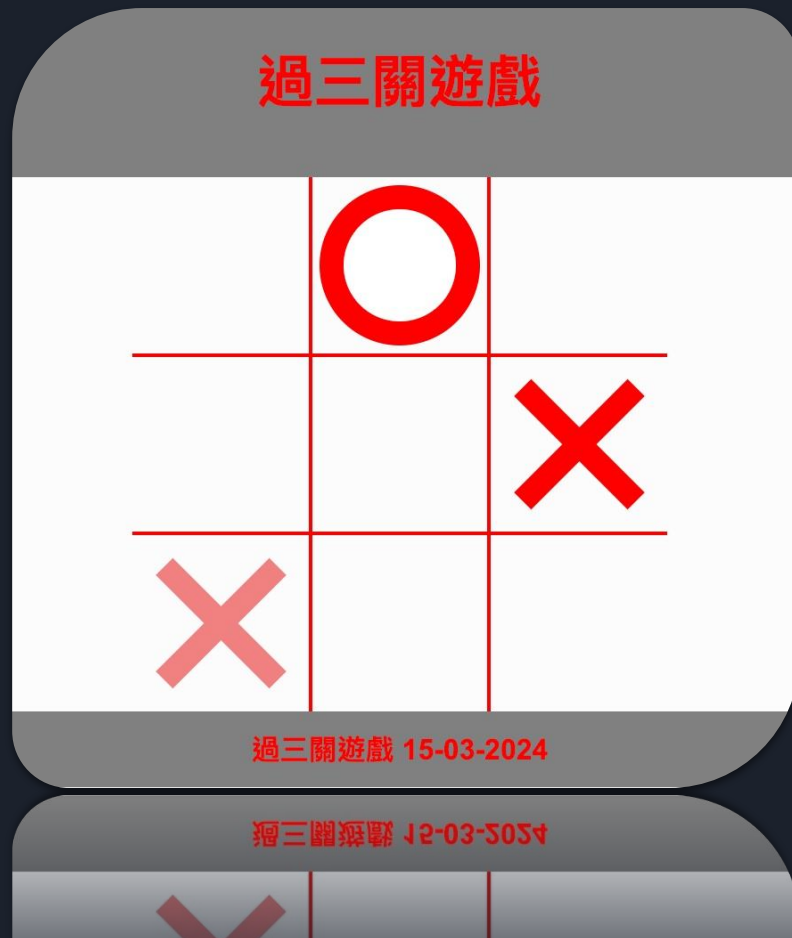
**pasu**  
pasu1215

5A11 Pasu Lam



## 團隊過往經驗

這是我們過往的製作的遊戲——  
過三關



Cards : 0

Time  
17.65

記憶配對

Best  
17.65

Restart !



# 遊戲背景和資料：

記憶配對遊戲讓玩家翻開卡片，記住卡片上的圖案或數字，尋找配對。遊戲訓練記憶力，鍛煉邏輯思維。簡單有趣，培養專注力。

記憶碎片



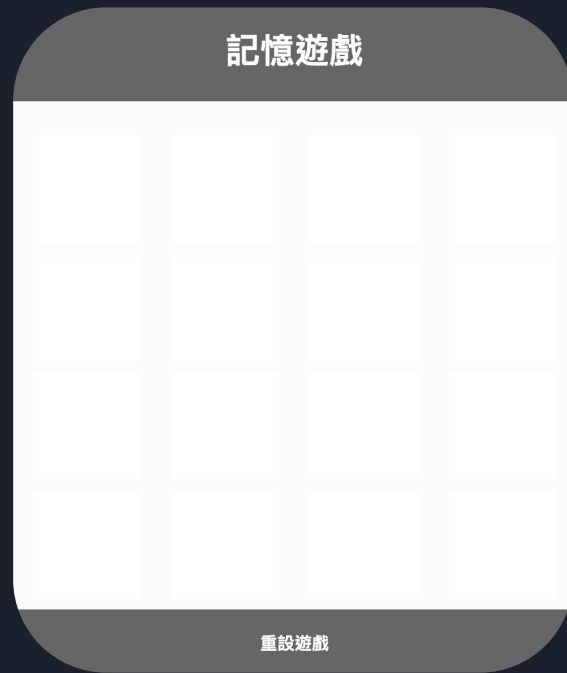
重設遊戲

繼續遊戲

# 製作遊戲介面的結構

當2張卡抽出相同的  
圖案時就是成功的第一步

而把所有卡抽出後才是  
成功結束整個遊戲



# 如何製作memory matching game？

背景：

- 當2張卡抽出相同的圖案時就是成功的第一步
- 而把所有卡抽出後才是成功結束整個遊戲

## 1.HTML架構：

在創建網頁標題為memory-matching-game  
在網頁中添加主要區塊，要有4x4的可點擊網格

記住加入header(標題)和footer(頁腳)

```
</style>
</head>
<body>
  <center>
    <header>
      <h1>記憶遊戲</h1>
    </header>

    <div id="game" class="grid">
      <!-- JavaScript動態生成卡片-->
    </div>

    <footer>
      <b> 重設遊戲</b>
    </footer>
  </center>
```



## 2.CSS樣式：

```
.grid {  
  margin: auto;  
  display: grid;  
  grid-template-columns: repeat(4, 1fr);  
  gap: 10px;  
  padding: 10px;  
  justify-content: center;  
  text-align: center;  
}  
  
.card {  
  margin: auto;  
  text-align: center;  
  width: 100px;  
  height: 100px;  
  background-color: #FFF;  
  display: flex;  
  justify-content: center;  
  align-items: center;  
  font-size: 2em;  
  cursor: pointer;  
}
```

為所有元素設定指定字體

為網格設定紅色背景和點擊指示符

網格區塊居中顯示，每個網格項目平均分佈

設定區塊裏的圖案/emo 😊

遊戲版面要置中對齊

## 3.Javascript邏輯：

### 3.1初始化：將變數matchCount預設為0

一開始，玩家會看見 memory matching game 的版面。玩家要按下區塊 = (抽卡)

- 不同區塊的圖案要隨機放置
- 當玩家隨機點擊第一個跟第二個區塊時，區塊會顯示圖案

### 3.2檢測兩個區塊的圖案的條件：


當對的時候：

- 如果區塊的圖案不相同，第一個跟第二個區塊要反轉。並且繼續遊戲。
- 變數matchCount要減2

當配對錯的時候：

- 如果區塊的圖案相同，第一個跟第二個區塊不用反轉。並且繼續遊戲。
- 變數matchCount要繼續加上去
- 同時設一個變數為 match count去記錄玩家按下不同區塊的次數

### 3.3當遊戲結束時的條件：

- 當玩家把所有卡全部揭開後，必須要顯示遊戲結束，然後生成一個重玩遊戲的按鈕，還原一開始 memory matching game 的版面。
- 當變數clickCount大於或等於 16，必須要中止遊戲進行 **!!**

另外的event listener幫助

- 加入event listener 監聽器來處理玩家的點擊動作。

```
// Emoji 列表
const emojis = [ '😄', '😄', '😄', '😄', '😄', '😄', '😄', '😄', '😄'];
let selectedEmojis = [...emojis, ...emojis]; // 需要兩份emoji來匹配
let selected = []; // 存儲選擇的卡片
let matched = []; // 存儲匹配成功的卡片

// 初始化遊戲
function initGame() {
  selectedEmojis.sort(() => 0.5 - Math.random()); // 打亂emoji順序*****
  const game = document.getElementById('game');
  game.innerHTML = ''; // 清空遊戲面板
  selectedEmojis.forEach((emoji, index) => {
    const card = document.createElement('div');
    card.classList.add('card');
    card.dataset.index = index;
    card.innerHTML = emoji;
    game.appendChild(card);
  });
  setTimeout(hideEmojis);
}
```

**Q&A 時間**



多謝

