

以 HTML、CSS 與 JavaScript 實作之

物件導向網頁象棋遊戲介面

張逸中、王士豪、何定鴻、黃信豪、陳景銘*

台灣首府大學數位娛樂與遊戲設計學系

*E-mail: ycc@tsu.edu.tw

摘要

有鑑於以外掛程式方式嵌入網頁的遊戲設計模式安全性受到質疑，且有過度消耗網路資源的問題，網頁程式回歸使用 HTML、CSS 與 JavaScript 等標準網頁技術進行設計是一個新的趨勢。然而相較於一般視窗程式語言，這種架構在物件導向機制的程式操作方面仍略顯不足，增加了設計網頁遊戲的難度。本文提出了一個基於此技術架構的程式設計解決方案，可彌補網頁元件在物件導向機制方面的不足，並用以完成一個無須外掛程式的網頁象棋遊戲介面。

關鍵字：JavaScript、CSS、HTML、網頁遊戲、物件導向、

壹、介紹

近年來由於網路品質提高、網路社群發達以及行動裝置普及等等因素，網頁遊戲的需求大增，品質與數量亦隨之成長。但由於網頁內通用的 JavaScript 腳本語言被認定功能不如一般的物件導向視窗程式語言，設計者多半選擇使用 Flash 或 Java 等程式先獨立設計遊戲，編譯成執行檔後以外掛方式嵌入網頁中。

然而這些嵌入的程式因為內容不透明，安全性備受質疑，無法獲得多數作業系統的信任，共享既有的系統資源。因此程式必須自行包裹大量的資源檔，或者在執行過程中頻繁的經過網路至母公司下載資源。這些狀況都會過度消耗網路頻寬與電力，對於依賴無線網路與電池供電的行動裝置非常不利。

有鑑於此，減少使用外掛程式，回歸以 JavaScript 配合 HTML 與 CSS 等網頁標準技術設計網頁程式成為許多專家期許的最新發展方向。本文內容就是介紹一個完全使用網頁標準技術，不借助外掛程式設計的一個網頁象棋遊戲。

程式的主要特色是經由 CSS 樣式表的協助，讓網頁上的 HTML 元件能模擬

一般物件導向視窗程式語言中的控制項物件，讓 JavaScript 可以用一般視窗程式控制物件的技巧操控這些網頁物件。譬如：能用滑鼠在網頁上拖曳棋子進行下棋的動作，棋子物件也可以互相覆蓋或被刪除等等。

貳、JavaScript 程式物件的製作

製作象棋遊戲，必須建立棋盤與棋子物件。在一般的物件導向視窗程式設計環境中，工具箱裡會有預設功能屬性已經相當完整的控制項。要製作棋子物件時，只需布置一個控制項到表單上，它就會動產生物件名稱，接著將棋子圖案設定給控制項，再拖曳至表單需要的位置即可完成靜態設計。本程式是以 HTML 元件中的 div 為物件設計的基礎，要讓 div 成為一個類似視窗程式中的控制項，後續可以使用 JavaScript 程式控制的物件必須完成以下幾項工作：

一、為物件命名

網頁元件預設沒有名稱，如需使用程式控制時呼叫非常不便。因此必須先替物件設定 id 屬性，也就是該物件在此網頁內獨一無二的名稱。方式是在 div 物件的標籤內加入 id 屬性。語法如下：

```
<div id="物件名稱">
```

二、設定物件之大小與圖案

要在 div 物件內加入影像的方式是在物件的樣式表 style 屬性中設定圖檔為其背景影像，當然同時應該將物件的高與寬度設為與圖檔相同。在本遊戲中使用的圖檔是 60*60 點的棋子圖案。此部分語法大致如下：

```
style="width: 60px; height: 60px; background-image: url('圖檔名稱');"
```

所謂的樣式表也就是所謂的 CSS(Cascading Style Sheet)技術，可以說是附加於 HTML 內用來增強網頁物件格式控制的一套功能語法。凡是 style 關鍵字之後定義的「子屬性」都稱為樣式表(CSS)定義，標準句型是「屬性名稱：屬性值」；而 HTML 的屬性寫法則是「屬性名稱="屬性值"」，兩者語法不同。

三、設定物件定位方式、位置與層次

在網頁中預設物件位置是以文字行為基礎逐行加入的，無法像一般視窗程式一樣，可用絕對座標自由移動物件位置。要讓網頁物件可在網頁中自由定位，必須修改物件 style 屬性中的定位方式為絕對定位，並設定 left 與 top 屬性，就可以讓物件定位到指定的位置了！語法如下：

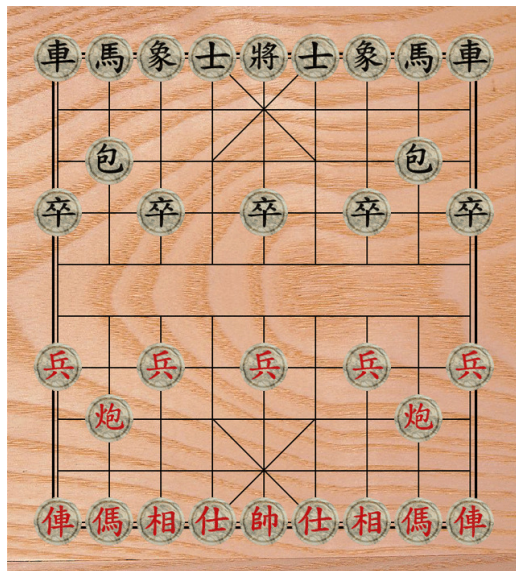
```
position: absolute; top: 610px; left: 290px; z-index: 1;
```

上述程式中最後出現的 z-index 屬性控制的是物件的「層次」。此值越大表示越上層，譬如我們設計棋盤時可以將 z-index 設定為 0，棋子則都設為 1，這樣棋子看起來就通通「放在棋盤上」了！此屬性稍後可以用 JavaScript 程式控制。

四、宣告事件副程式

本遊戲過程中棋子物件會被拖曳，所以必須能回應滑鼠事件，這必須在物件的 div 標籤中宣告，語法如：onmousedown="md()" onmousemove="mv()" onmouseup="mup()"。意義就是 MouseDown 事件發生時要執行 md() 副程式，依此類推。最終，代表一個棋子的 HTML div 標籤原始碼大致如下：

```
<div id="a1" style="position: absolute; width: 60px; height: 60px;
background-image: url('images/a1.png'); top: 610px; left: 290px; z-index: 1;"
onmousedown="md()" onmousemove="mv()" onmouseup="mup()">
</div>
```



圖一 棋盤與棋子網頁設計完成之初始畫面

參、棋盤資訊的陣列控制

下棋遊戲過程中程式必須隨時維護一個代表棋盤狀態的二維陣列，就是隨時登錄哪一個棋子在哪一個位置？但 JavaScript 語言中沒有直接宣告二維陣列的語法，因此必須使用較迂迴的方式先宣告一個一維陣列，並將陣列成員也都宣告為一維陣列。同時間，在象棋遊戲哩，各「種」棋子各有不同的移動規範，所以還必須有另一個登錄該棋子種類的二維陣列，以便程式可以檢測下棋的動作是否合法？因此本程式一開始以下列程式碼宣告這兩個二維陣列，並給予陣列成員一個初始值，登錄物件名稱的是字串陣列 Q，成員初值是空字串；登錄棋種的是數值陣列 A，成員初值是 0。

```

A = new Array(9); //宣告棋種屬性分布的陣列
Q = new Array(9); //宣告棋子 ID 分布的陣列
for (var i = 0; i < 9; i += 1) {
    A[i] = new Array(10); //宣告棋種次級陣列
    Q[i] = new Array(10); //宣告 ID 次級陣列
    for (var j = 0; j < 10; j += 1) {
        A[i][j] = 0; //棋種代碼預設值
        Q[i][j] = ""; //ID 預設值→空字串
    }
}

```

可以看到，某一個陣列成員的表示語法是：A[i][j]，與一般程式習慣寫成：A(i, j)的方式略有不同。緊接著，我們要將棋盤初始狀態，也就是每個棋子一開始的位置資訊登錄到上面兩個陣列之中，以本程式的某個棋子為例，程式如下：

```

A[0][0] = -4; //棋種代號
Q[0][0] = "b4"; //棋子物件 ID

```

本程式中以 1~7 七個數字代表「帥仕相俥馬炮兵」等七種紅色棋子，相對的負數代表黑棋的同樣棋種。字元 a 開頭的 ID 代表紅棋，b 代表黑棋。如上依序寫入各棋子的資訊就可以完成棋盤資訊陣列的初值設定。

肆、棋子拖曳移動的控制

要讓棋子可以被拖曳需要棋子可以回應 MouseDown 與 MouseMove 兩個事件，而且在玩家放開滑鼠時程式還必須檢查下棋動作的合法性，因此 MouseUp 事件也必須存在。這幾個事件副程式已經在之前的 div 標籤中宣告了，但是要執行整個拖曳動作還必須宣告幾個公用變數：C 代表被選定拖曳的物件，滑鼠壓下時以 mdx 與 mdy 記錄起始座標，drag 代表程式是否處於拖曳物件的狀態。

除此之外我們還必須控制下棋次序的輪替，每一方下完一步之後必須等另一方下。在此是以一個公用變數 T 來控制，T=true 時由紅方下，T=false 時黑方下。此部分程式碼如下：

```

function md() {
    C = event.srcElement; //取得觸發事件的物件(棋子)
    w = C.id.substr(0, 1); //取得代表黑或紅方的關鍵字
    //未輪到正確下棋方時跳出副程式
    if ((w == "a" && T == false) || (w == "b" && T == true)) return;
    drag = true; //進入拖曳狀態
    mdx = event.offsetX; //取得拖曳起點 X
    mdy = event.offsetY; //取得拖曳起點 Y
    C.style.zIndex = 2; //提升拖曳中棋子的層次
}

```

```

C.style.cursor = "pointer";//改變游標為手指按棋的圖示
x1 = p2g(C.style.posLeft);//換算網頁像素點座標為棋盤座標 X
y1 = p2g(C.style.posTop);//換算網頁像素點座標為棋盤座標 Y
}
function mv() {
    if (drag) {//如果是拖曳狀態
        C.style.posLeft += event.offsetX - mdx;//X 方向移動
        C.style.posTop += event.offsetY - mdy;//Y 方向移動
    }
}
function mup() {
    if (drag) {//如果是拖曳狀態
        drag = false;//結束拖曳狀態
        C.style.cursor = "default";//恢復正常游標
        C.style.zIndex = 1;//恢復正常層次
        chess(C);//檢視下棋動作合法性
    }
}
}

```

在 md 程式中會先取得觸發事件的物件 C，接著取得物件 id 的第一個字 w，用以判斷是哪一方的棋子？若不是該下棋的一方就停止動作。接著是記錄起點，還有，被選的物件層次必須提高，以免被其他棋子遮住；圖示也要改變，讓使用者知道該物件已被鎖定即將被拖曳，drag 值也要變成 true。

拖曳時的座標當然是以網頁上的像素點座標為準，但是棋盤座標則是以棋格為單位，因此我們的座標會以一個副程式 p2g 換算成棋格座標(x1, y1)，以便後續作下棋規則的判斷。限於篇幅，此 p2g 以及其他許多程式片段將不會在本文詳述，但原始程式碼可以在最終展示網頁之原始檔中完整取得。

接下來 mv 程式會依據 drag 變數決定是否執行拖曳動作，如果是 VB 程式，此地可寫成 If e.Button=...，就是在此直接辨識滑鼠鍵的按下狀態，不必使用 drag 變數。雖然網頁程式也有 event.button 這種語法，但不幸的，此屬性值的定義並不一，有些瀏覽器認定滑鼠左鍵值是 1，有些則是 0！因此目前必須使用 drag 變數這種寫法才能保證所有瀏覽器皆能正確執行此程式。

最後的 mup 程式就是解除拖曳狀態 drag=false，並呼叫一個 chess 副程式，這個副程式是用來檢視下棋是否合法的處理中心。如果合法！會讓棋子在指定棋格放下，並改變棋盤狀態資訊；如果不合法！棋子會跳回原來位置，就是 md 執行時記錄的(x1, y1)位置。

伍、下棋合法性的控制

如上節所述，chess 副程式控制下棋合法性，其中第一段程式會調整落子的位置，自動調整到棋格的精準位置上，並取得落子的棋格座標，程式碼如下：

```
var x = C.style.posLeft + 30; //棋子中心座標 X
var y = C.style.posTop + 30; //棋子中心座標 Y
C.style.posLeft = Math.round(x / 64) * 64 - 30; //移動至棋格 X 位置
C.style.posTop = Math.round(y / 64) * 64 - 30; //移動至棋格 Y 位置
x2 = p2g(C.style.posLeft); //取得棋子移至的棋格座標 X
y2 = p2g(C.style.posTop); //取得棋子移至的棋格座標 Y
if (outside() || occupy()) { //超出棋盤或有我方棋子
    C.style.posLeft = g2p(x1); //跳回原處 x1
    C.style.posTop = g2p(y1); //跳回原處 y1
    return;
}
```

取得目標棋格位置(x2, y2)之後首先要檢查是不是超出棋盤，以及是不是有我方的棋子在該位置，照規則棋子當然不能下在棋盤外，也不能「吃掉」自己的棋子。這些功能由 outside 與 occupy 副程式執行，限於篇幅不在此列舉詳述。接著是取得目前移動棋子的種類，然後依據棋種作不同的規則查核，程式架構如下：

```
var k = A[x1][y1]; //取得棋種代號
var legal = false; //宣告合法性變數，預設為不合法
switch (k) {
    case 1://帥
        legal = onestep() && inpalace(1);
        break;
    ...
    ...
}
```

想當然耳，每一種棋子都有自己的一個 case(案例)，每個案例也會有不同的篩選規則，每項規則都會以一個自訂副程式來檢核。譬如針對「帥」的動作是以 onestep 與 inpalace 兩個副程式來檢視合法性。這些副程式若發現任何不合法的情況會回傳 false 否則就是 true 了！譬如將與帥一次只能移動一步的限制寫在 onestep，程式碼如下：

```
function onestep() {
    if (Math.abs(x1 - x2) + Math.abs(y1 - y2) == 1) return true;
    return false;
}
```

程式的意義就是分別取得(x1-x2)與(y1-y2)之絕對值相加，如果在棋盤上只上下或左右移動一步，這個值等於 1，就回傳 true 表示合法；否則回傳 false 表示非法。這些規則副程式數量很多，無法一一詳述，請直接參考完成網頁的原

始檔。合法性檢查完後可以得到合法或不合法的兩種結果，處理程式碼如下：

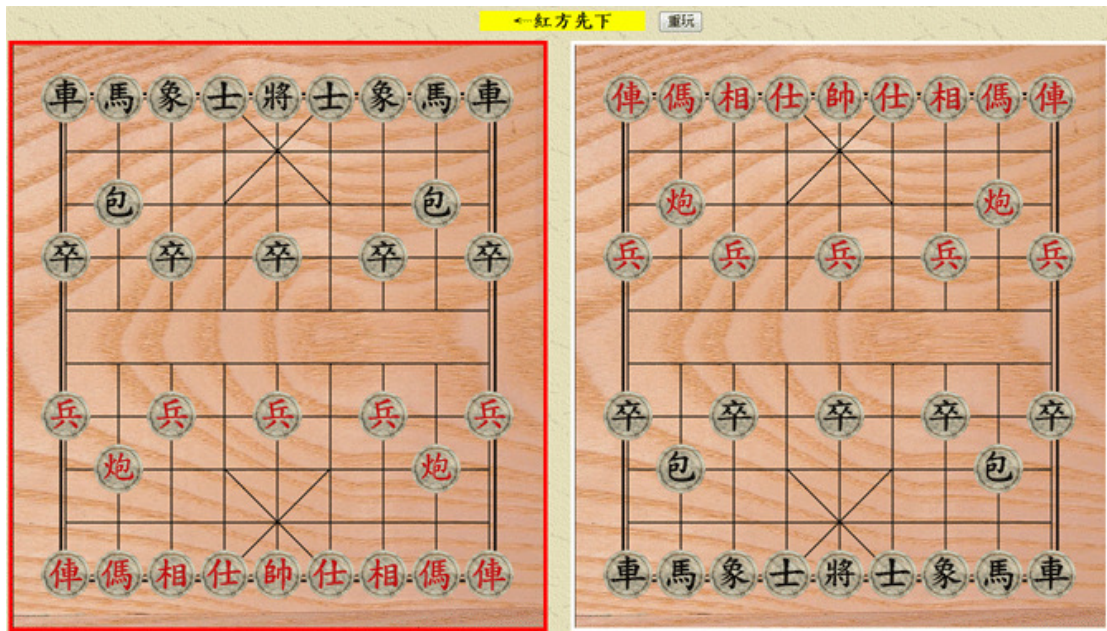
```
if (legal) { //如果合法
    if (A[x2][y2] != 0) { //目的地有對方棋子→吃棋
        var K = document.getElementById(Q[x2][y2]);
        K.style.visibility = "hidden"; //隱藏對方被吃的棋子
    }
    A[x2][y2] = A[x1][y1]; //更新棋種陣列資訊
    Q[x2][y2] = Q[x1][y1]; //更新物件名稱資訊
    A[x1][y1] = 0; //清除原位置資訊
    Q[x1][y1] = ""; //清除原位置資訊
    if (w == "a") { //紅方完成下棋
        T = false; //換黑棋下
        BG.title = "輪到黑棋下"; //顯示輪替訊息
    }
    if (w == "b") { //黑方完成下棋
        T = true; //換紅棋下
        BG.title = "輪到紅棋下"; //顯示輪替訊息
    }
} else { //如果不合法
    C.style.posLeft = g2p(x1); //跳回原位 x1
    C.style.posTop = g2p(y1); //跳回原位 y1
}
C = null; //清除選定棋子物件
```

合法的移動有時會需要「吃掉」對方的棋子，此時應該使用 JavaScript 的 `removeNode` 方法將目標位置的敵方棋子「刪除」！但不是所有主流瀏覽器都支援這個方法，所以在此改用「隱藏」物件(`visibility=hidden`)的方式讓棋子消失！

接著是看輪到誰下？在此動作合法表示已經完成下棋動作，應該切換 `T` 值，並用棋盤(BD 物件)的 `title` 屬性顯示輪到誰下？如果是不合法，就直接讓棋子跳回到開始拖曳的原來位置(`x1, y1`)。最後是清除選定的物件 `C`，就是讓玩家重新用滑鼠選棋與下棋。

陸、雙人版的設計

本研究如上所述逐一建立程式物件與程式碼，完成之網頁遊戲公布於網址：<http://ycc.tsu.edu.tw/jschess>。當然如圖一所示，這個棋盤是直式的，雖然黑紅雙方都可以執行下棋動作，但是硬要黑方以視覺上相反的方向下棋當然很不自然。目前多數網頁棋戲要讓兩人對奕時棋盤多是橫向的，也就是玩家的角度都像是站在一旁的「觀棋者」，也很不自然。因此我們繼續設計了雙人版的頁面如下：



圖二 雙人版網頁象棋畫面

上述頁面網址是：<http://ycc.tsu.edu.tw/jschess/adv.htm>，兩個棋盤能互相同步，但方向相反，方便兩個人在一個頁面中各據一方下棋。下棋切換時會有紅色框線與提示文字提醒輪到誰下？就能讓玩家以習慣的下棋方向下棋了！原因應該是設計難度的問題，要讓兩個棋盤同步當然需要較多的程式碼。

柒、結論

本研究如題完成了一個使用 HTML、CSS 與 JavaScript 標準網頁設計工具設計的網頁象棋遊戲程式。其主要意義是證明了經過 CSS 功能的協助，一般的 HTML 網頁物件也可以具備如同視窗程式控制項的物件導向操作特性；同時，JavaScript 不僅可以完成滑鼠事件副程式的設計，也可以順利完成相當複雜的陣列邏輯運算，確實具備製作較複雜的網頁互動遊戲的能力。

捌、參考文獻

- 小丰子 3C 俱樂部，Apple 與 Adobe 在吵什麼？Flash vs. HTML5&CCS3 優缺點，
<http://tw.myblog.yahoo.com/olivertu-blog/article?mid=2222>
- 耘想科技有限公司，網站上到底該不該使用 FLASH 動畫？有什麼缺點？
<http://www.pkthink.com/knowledge-info.asp?id=7>
- 張智興，JavaScript 程式設計與應用，國立清華大學出版社，2008/1
- 張逸中，中國象棋網頁，<http://ycc.tsu.edu.tw/jschess>，2013/3
- 張逸中，拖曳網頁物件，<http://ycc.tsu.edu.tw/WebPage/DragObject.htm>，2013/1
- 楊水清等編著，JavaScript 與 Ajax 網頁程式設計，博碩文化出版，2008/9