浙江大学

题　　目：黑白棋小游戏——以 Javascript 实现

专业与班级：理科试验班（生命与环境）1203 班

姓　　名：俞杰超

学　　号：3120103005

指导　教师：沈睿

时　　间：2012 年 12 月 31 日

目录

摘要及简介：

本文通过分析一个黑白棋小游戏的算法，来展示 javascript 的使用以及它是如何在网页中产生作用的。

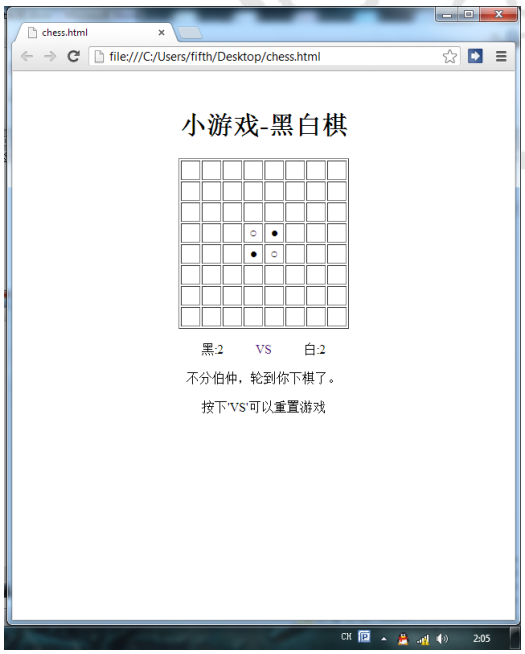
程序所使用的语言涉及 javascript、html 和 css 等，其中主体是由 javascript 编写，html 用做搭建网页框架，css 则用来对一些 html 元素做一些简要修饰。

程序采取的算法是贪心算法，即局部最优解。算法的简要概述及具体描述将在正文中做出说明。

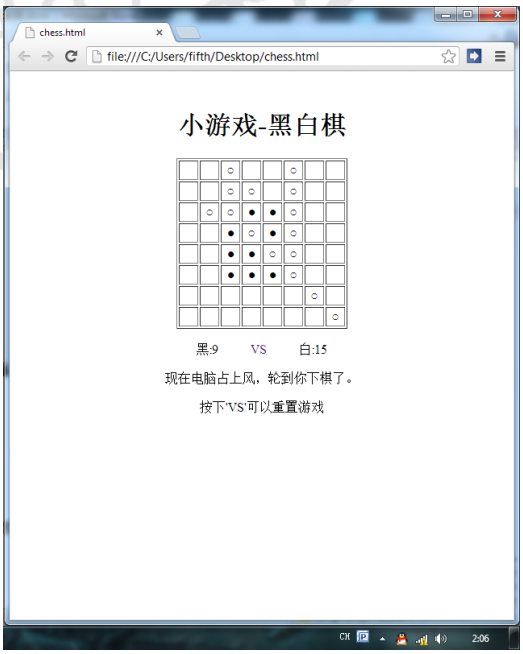
关键字：

黑白棋　Javascript　贪心法　优先级　检索

网页预览：



游戏开始



游戏进行过程中

注：

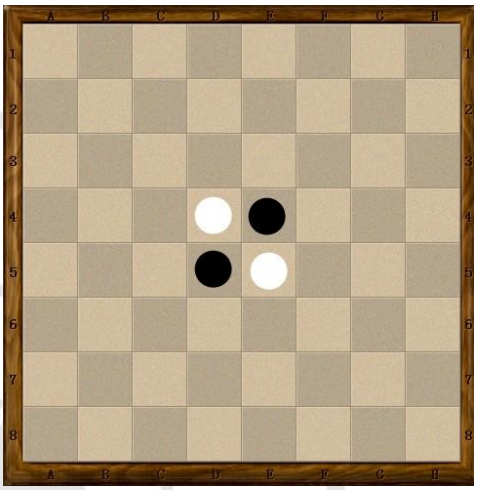
可点击 http://test.myqsc.com/fifth/chess/chess.html 浏览（需有浙大内网）

也可打开压缩包中所附的 html 文件本地使用

正文：

【1】黑白棋介绍：

【1.1】游戏概述：



黑白棋又称反棋（Reversi）、苹果棋、翻转棋、奥赛罗棋（Othello）等，是一种在西方和日本非常流行的棋类游戏。在日本，黑白棋爱好者的数目约为六千万人。

游戏规则很简单，通过下棋来依次翻转对方的棋子，最后依据棋子的数目来判断输赢。

有一种说法是这样的，只需要几分钟学会它，却需要一生的时间去精通它。

【1.2】游戏规则及胜负判定：

黑白棋的游戏棋盘有 8 行 8 列，共分成 64 格。开局时，棋盘正中央的 4 格先预放置两黑两百间隔的四个棋子。

黑白棋通行的惯例是由黑方先行，然后双方轮流落子。只要落子和棋盘上已经存在的任意一枚己方的棋子在一条线上（横，竖，斜线均可）夹着对方的棋子，就能将对方的棋子变为己方棋子。如果在棋盘上的任意位置都不能按上述规则下子时，就跳过这轮下子，让对手下子。如果场上有符合规则的位置可以下，则必须下子，不得弃权。当两个人都不能下子（包括棋盘被填满或者一方场上没有棋子）时，游戏结束，清算场上的棋子数，棋子数多的一方获胜。

【1.3】游戏策略

出于黑白棋特殊的游戏规则，很容易出现场上比分的剧烈变化，很可能在最后的几回合中双方的比分被数次逆转，所以在比赛中过多地着眼于比分是没有意义的，位置比比分更重要。

中间位置的棋子会被各个位置夹击，而边缘的棋子则只会被同行的棋子夹击，四个角上的棋子则完全不会被夹击，这就是所谓的“金角银边草肚皮”。

一些玩家在游戏中喜欢占据中间的位置，而使对手占据一组棋子的外缘。然后在游戏的后期就可以通过将棋子放在棋盘边缘的位置上来在几回合中翻转对方大量的棋子。

最终你希望占据棋盘的边缘，这样对手就不能更外部放置棋子了。基于此，你要尽量避开在与边缘紧邻的位置下棋。

四个角上是必须争取占据的位置，因为这些位置上的棋子无法被翻转。

【2】Javascript 介绍

【2.1】Javascript 简介

JavaScript 是一种基于对象和事件驱动并具有相对安全性的客户端脚本语言。同时也是一种广泛用于客户端 Web 开发的脚本语言，通常被用来给 HTML 网页添加动态功能。Javascript 最初由网景公司的 Brendan Eich 设计，是一种动态、弱类型、基于原型的语言，内置支持类。JavaScript 是 Sun 公司的注册商标。Ecma 国际以 JavaScript 为基础制定了 ECMAScript 标准。JavaScript 也可以用于其他场合，如服务器端编程。完整的 JavaScript 实现包含三个部分：ECMAScript，文档对象模型，字节顺序记号。

Netscape 公司在最初将其脚本语言命名为 LiveScript。在 Netscape 在与 Sun 合作之后将其改名为 JavaScript。JavaScript 最初受 Java 启发而开始设计的，目的之一就是“看上去像 Java”，因此语法上有类似之处，一些名称和命名规范也借自 Java。但 JavaScript 的主要设计原则源自 Self 和 Scheme。尽管 JavaScript 作为给非程序人员的脚本语言，而非作为给程序人员的编程语言来推广和宣传，但是 JavaScript 具有非常丰富的特性。

【2.2】程序中使用的 Javascript 语法

【2.2.1】分支

```
if   (condition)
     {
          code to be executed if condition is true
     }
else
     {
          code to be executed if condition is not true
     }
```

【2.2.2】循环

【2.2.2.1】for 循环

```
for   (statement 1; statement 2; statement 3)
     {
          the code block to be executed
     }
statement 1 sets a variable before the loop starts (var i=0)
statement 2 defines the condition for the loop to run (i must be less than 5)
```

statement 3 increases a value (i++) each time the code block in the loop has been executed.

【2.2.2.2】while 循环

```
while (condition)
{
    code block to be executed
}
```

【2.2.3】数组

数组创建：

```
var  arrayName = new Array()
```

数组赋值：

```
var  arrayName = new Array()
arrayName[0]=“variables1”
arrayName[1]=“variables2”
arrayName[0]=“variables3”
```

或

```
var  arrayName = new Array(“variables1”, “variables2”, “variables3”)
```

Javascript 不支持多维数组，需要先定义一个一维数组，再将一维数组的成员定义成数组（之所以可以进行再定义，因为 JavaScript 是弱类型），实现的代码如下：

```
var  arrayName=new Array();
for (var i=0;i<=arrayNumber;i++)
    {
        arrayName[i]=new Array();
    }
```

【2.2.4】函数

函数声明：

```
function functionName(var1,var2,…,varN)
    {
        code
    }
```

函数调用时只需直接调用函数名即可。

【2.2.5】其它 Javascript 语句

alert()：弹出一个警告窗口，括号里为要提示的信息，可以是字符串，变量，表达式，函数等。

document.write()：在网页上显示数据，括号里为要显示的信息，可以是字符串，变量，表达式，函数等。

document.getElementById().innerHTML：获取当前页面指定 id 的元素，括号里为需要获取的元素的 id，innerHTML 属性获取该 id 的标签内部的 HTML 代码。

window.location.reload()：重新载入当前页面。括号里为需要重载的页面，为空则重载当前页面。

【3】用 Javascript 实现黑白棋的算法简述

本程序使用贪心法实现人机对战。

【3.1】变量及函数定义

【3.1.1】变量

occupy：这是一个二维数组，用来储存每个格子被占领的状态。1 表示被黑子占领，2 表示被白子占领，0 表示没有被任何子占领。这是一个全局变量。

order：这是一个二维数组，用来储存每个格子的优先级。分 1-5 共 5 个优先级，当棋盘的格子被占领之后，优先级被标记为 0。这是一个全局变量。

search：这是一个二维数组，用来储存每个格子能将多少对方的棋子翻转。这是一个全局变量，但在每次电脑检索时都会重新刷新一遍这个数组里面储存的数据。

max：这是一个一维数组，用来储存翻转对方棋子的最大数及该位置的坐标。max[0]储存能翻转的最多棋子数，max[1]储存能翻转最多棋子时的棋盘横坐标，max[2] 储存能翻转最多棋子时的棋盘纵坐标。这是一个全局变量，但在每次电脑检索的时候会初始化。

action：这是一个变量，用来标记一方是否有下子动作。当一方有下子动作的时候，action 会被赋值为 1。这是一个全局变量，action 的初始化值为 0，且每次玩家下子时或电脑开始检索时会被重置为 0。

end_or_not：这是一个变量，用来标记对战是否结束。若程序判断对战已经结束，则 end_or_not 会被标记为 0，否则 end_or_not 会保持为 1。这是一个全局变量，end_or_not 的初始化值为 1，在判断对战结束的时候会被赋值为 0。

【3.1.2】函数

lose_white()：判断白色一方是否还能下子，可以下子返回 0，不可以下子返回 1。

lose_black()：判断黑色一方是否还能下子，可以下子返回 0，不可以下子返回 1。

battle()：检查全盘的状态，统计白子和黑子的数量。

end(b,w)：结束对战，返回输赢信息并重载页面。b 表示黑子的数量，w 表示白子的数量。

computer()：AI 的操作。

message(b,w)：返回展示对战动态所需的 HTML 代码。b 表示黑子的数量，w 表示白子的数量。

box()：返回打印棋盘所需的 HTML 代码。

check(x,y,z)：检索棋盘并翻转对方的棋子。x 表示该点的横坐标，y 表示该点的纵坐标，z 表示下的子是黑子还是白子，1 表示黑子，2 表示白子。

check_up(x,y,z)， check_down(x,y,z)， check_left(x,y,z)， check_right(x,y,z)，check_leftup(x,y,z)，check_rightup(x,y,z)，check_leftdown(x,y,z)，check_rightdown(x,y,z)分别向上、下、左、右、左上、右上、左下、右下检索。

search_a(x,y,z)：检索棋盘并计算能翻转多少对方的棋子。x 表示该点的横坐标，y 表示该点的纵坐标，z 表示下的子是黑子还是白子，1 表示黑子，2 表示白子。

search_up(x,y,z)， search_down(x,y,z)， search_left(x,y,z)， search_right(x,y,z)，search_leftup(x,y,z)，search_rightup(x,y,z)，search_leftdown(x,y,z)，search_rightdown(x,y,z)分别向上、下、左、右、左上、右上、左下、右下检索。

【3.2】棋盘优先级初始状态

5 表示最高优先级，1 表示最低优先级，由高到低优先级依次递减，0 表示此格已被占领，不参与优先级排序。见左表。

棋盘优先级初始状态

5	2	4	4	4	4	2	5
2	1	3	3	3	3	1	2
4	3	4	4	4	4	3	4
4	3	4	0	0	4	3	4
4	3	4	0	0	4	3	4
4	3	4	4	4	4	3	4
2	1	3	3	3	3	1	2
5	2	4	4	4	4	2	5

【3.3】棋盘占据状态初始状态

1 表示黑子，2 表示白子。见右表。

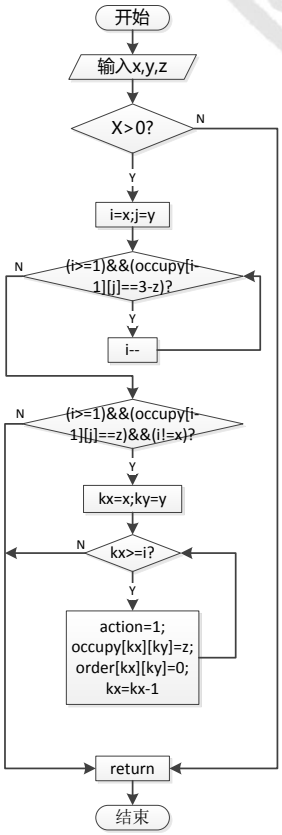
棋盘占据状态初始状态

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	2	1	0	0	0
0	0	0	1	2	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

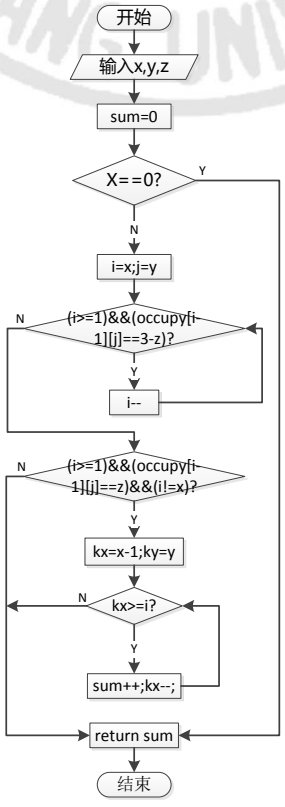
【3.4】落子之后翻转对方的棋子

以向上检索为例，具体逻辑结构见下面的流程图。

算法流程图——check_up(x,y,z)



算法流程图——search_up(x,y,z)



【3.5】计算机检索最优的落子点

以向上检索为例，具体逻辑结构见左边的流程图。

【3.6】判断对战结束

【3.6.1】双方都无子可下

调用 lose_white()和 lose_black()函数，若 lose_white()&&lose_black()返回为真，则表示双方都没有子可下，这时调用 end(b,w)，其中 b 参数输入黑子的个数，w 参数输入白子的个数。

【3.6.2】一方棋盘上剩余的子数目为零

调用 battle()函数，若(black==0)||(white==0)返回为真，则表示一方棋盘上剩余的棋子数为零，这时调用 end(b,w)，其中 b 参数输入黑子的个数，w 参数输入白子的个数。

【3.6.3】棋盘下满

调用 battle()函数，若 total>=64 返回为真，则表示棋盘上的格子已经被下满了，这时调用 end(b,w)，其中 b 参数输入黑子的个数，w 参数输入白子的个数。

【3.6.4】end(b,w)函数

获得参数 b 和 w，其中 b 表示黑子的数量，w 表示白子的数量。在函数内部比较 b 与 w 的大小：若 b>w 返回为真，则提示“你赢了”，将 end_or_not 赋值为 1，点击确定之后重载页面；若 b<w 返回为真，则提示“电脑赢了”，将 end_or_not 赋值为 1，点击确定之后重载页面；若 b==w 返回为真，则提示“平局”，将 end_or_not 赋值为 1，点击确定之后重载页面。

【4】Javascript 代码

见附件

【5】其它相关内容：

【5. 1】超文本标记语言

超文本标记语言（HyperText Markup Language，HTML）是为「网页创建和其它可在网页浏览器中看到的信息」设计的一种标记语言。HTML 被用来结构化信息——例如标题、段落和列表等等，也可用来在一定程度上描述文档的外观和语义。1982 年由蒂姆·伯纳斯-李创建，由 IETF 用简化的 SGML（标准通用标记语言）语法进行进一步发展的 HTML，后来成为国际标准，由万维网联盟（W3C）维护。

HTML 档案最常用的扩展名（扩展名）是.html，但是像 DOS 这样的旧操作系统限制扩展名为最多 3 个字符，所以.htm 扩展名也允许使用。现在.htm 扩展名使用的比较少一些了，但是仍旧受到支持。编者可以用任何文本编辑器或所见即所得的 HTML 编辑器来编辑 HTML 文

件。

早期的 HTML 语法规则定义较为松散，这有助于不熟悉网络出版的人采用。网页浏览器接受了这个事实，使之可以显示语法不严格的网页。随着时间的流逝，官方标准渐渐趋于严格的语法，但是浏览器继续显示一些远称不上合乎标准的 HTML。使用 XML 的严格规则的 XHTML（可扩展超文本标记语言）是 W3C 计划中的 HTML 的接替者。虽然很多人认为它已经成为当前的 HTML 标准，但是它实际上是一个独立的、和 HTML 平行发展的标准。W3C 目前建议使用 XHTML 1.1、XHTML 1.0 或者 HTML 4.01 标准编写网页，但已有不少网页转用较新的 HTML5 编码撰写（如 Google）。

【5.2】层叠样式表

CSS 是英语 Cascading Style Sheets（层叠样式表单）的缩写，它是一种用来表现 HTML 或 XML 等文件样式的计算机语言。

CSS 目前最新版本为 CSS3，能够真正做到网页表现与内容分离的一种样式设计语言。相对于传统 HTML 的表现而言，CSS 能够对网页中的对象的位置排版进行像素级的精确控制，支持几乎所有的字体字号样式，拥有对网页对象盒模型的能力，并能够进行初步交互设计，是目前基于文本展示最优秀的表现设计语言。CSS 能够根据不同使用者的理解能力，简化或者优化写法，针对各类人群，有较强的易读性。

【6】相关链接：

【6.1】黑白棋算法分析(一)——扛枪的博客:http://jcs130.iteye.com/blog/1136964

【6.2】百度百科——黑白棋：http://baike.baidu.com/view/24242.htm

【6.3】维基百科——黑白棋：
http://zh.wikipedia.org/wiki/%E9%BB%91%E7%99%BD%E6%A3%8B

【6.4】百度百科——Javascript：http://baike.baidu.com/view/16168.htm

【6.5】维基百科——javascript：http://zh.wikipedia.org/wiki/Javascript

【6.6】w3school 在线教程——Javascript 教程：
http://www.w3school.com.cn/js/index.asp

【6.7】W3schools Online Web Tutorials——Javascript Tutorial：
http://www.w3schools.com/js/default.asp

【6.8】百度百科——超文本标记语言：http://baike.baidu.com/view/692.htm

【6.9】维基百科——HTML：http://zh.wikipedia.org/wiki/Html

【6.10】w3school 在线教程——HTML 教程：
http://www.w3school.com.cn/html/index.asp

【6.11】W3schools Online Web Tutorials——HTML Tutorial：
http://www.w3schools.com/html/default.asp

【6.12】百度百科——css：http://baike.baidu.com/view/15916.htm

【6.13】维基百科——层叠样式表：http://zh.wikipedia.org/wiki/Css

【6.14】w3school 在线教程——CSS 教程：
http://www.w3school.com.cn/css/index.asp

【6.15】W3schools Online Web Tutorials——CSS Tutorial：

http://www.w3schools.com/css/default.asp

【7】附件：

全代码如下，读者可将其复制到 txt 文本编辑器中，保存并修改文件后缀为 html，使用 ie、chrome 等浏览器打开即可，如出现显示乱码的情况，请手动将网页编码设置成自动获取或者 utf8。

```
<html><head>
    <script type="text/javascript">
        var   occupy=new Array();
              for (i=0;i<=7;i++)
              {
                    occupy[i]=new Array();
              }
        var order=new Array();
              for (i=0;i<=7;i++)
              {
                    order[i]=new Array();
              }
        var search=new Array();
              for (i=0;i<=7;i++)
              {
                    search[i]=new Array();
              }
        var max=new Array();
        var action=0;
        var end_or_not=1;
        for (i=0;i<=7;i++)
        {
        for (j=0;j<=7;j++)
        {
              occupy[i][j]=0;
        }
        }
        occupy[3][3]=2;
        occupy[3][4]=1;
        occupy[4][3]=1;
        occupy[4][4]=2;
        order[0][0]=5;
        order[0][7]=5;
        order[7][0]=5;
        order[7][7]=5;
        for (i=2;i<=5;i++)
```

```
{
	order[0][i]=4;
	order[2][i]=4;
	order[5][i]=4;
	order[7][i]=4;
	order[i][0]=4;
	order[i][2]=4;
	order[i][5]=4;
	order[i][7]=4;
}
for (i=2;i<=5;i++)
{
	order[1][i]=3;
	order[6][i]=3;
	order[i][1]=3;
	order[i][6]=3;
}
order[0][1]=2;
order[0][6]=2;
order[1][0]=2;
order[1][7]=2;
order[6][0]=2;
order[6][7]=2;
order[7][1]=2;
order[7][6]=2;
order[1][1]=1;
order[1][6]=1;
order[6][1]=1;
order[6][6]=1;
order[3][3]=0;
order[3][4]=0;
order[4][3]=0;
order[4][4]=0;
function lose_white()
{
	for (first=5;first>=1;first--)
	{
		for (ki=0;ki<=7;ki++)
		{
			for (kj=0;kj<=7;kj++)
			{
				if (order[ki][kj]==first)
				{
					search[ki][kj]=search_a(ki,kj,2);
```

```
                        if (search[ki][kj]>0) {return 0;}
                    }
                }
            }
        }
        return 1;
}
function lose_black()
{
        for (first=5;first>=1;first--)
        {
            for (ki=0;ki<=7;ki++)
            {
                for (kj=0;kj<=7;kj++)
                {
                    if (order[ki][kj]==first)
                    {
                        search[ki][kj]=search_a(ki,kj,1);
                        if (search[ki][kj]>0) {return 0;}
                    }
                }
            }
        }
        return 1;
}
function battle()
{
        black=0;
        white=0;
        for (i=0;i<=7;i++)
        {
            for (j=0;j<=7;j++)
            {
                if (occupy[i][j]==1) {black++;}
                else if (occupy[i][j]==2) {white++;}
            }
        }
        both=black+white;
        message(black,white);
        if (lose_black()&&lose_white()) {end(black,white);return;}
        if ((black==0)||(white==0)||(both>=64)) {end(black,white);return;}
        return;
}
function end(b,w)
```

```
{
	end_or_not=0;
	if (b>w)
	{
		alert("你赢了");
		window.location.reload();
		return;
	}
	if (b<w)
	{
		alert("电脑赢了");
		window.location.reload();
		return;
	}
	if (b==w)
	{
		alert("平局");
		window.location.reload();
		return;
	}
	return;
}
function computer()
{
	max[0]=0;
	max[1]=0;
	max[2]=0;
	for (first=5;first>=1;first--)
	{
		for (ki=0;ki<=7;ki++)
		{
			for (kj=0;kj<=7;kj++)
			{
				if (order[ki][kj]==first)
				{
					search[ki][kj]=search_a(ki,kj,2);
					if (search[ki][kj]>max[0])
					{
						max[0]=search[ki][kj];
						max[1]=ki;
						max[2]=kj;
					}
				}
			}
```

```
                }
                if (max[0]!=0)
                {
                    check(max[1],max[2],2);
                    return;
                }
            }
            return;
        }
        function message(b,w)
        {
            var msg='';
            msg="<p>黑:";
            msg+=b;
            msg+="          ";
            msg+="<a href='' onclick='window.location.reload();return false'
style='text-decoration:none'>VS</a>";
            msg+="          ";
            msg+="白:";
            msg+=w;
            msg+="</p>";
            if (b>w) {msg+="<p>现在你正占上风，";}
            else if (b<w) {msg+="<p>现在电脑占上风，";}
            else if (b==w) {msg+="<p>不分伯仲，";}
            msg+="轮到你下棋了。</p>";
            msg+="按下'VS'可以重置游戏";
            document.getElementById('message').innerHTML=msg;
            return '';
        }
        function box()
        {
            txt="<table border='1'>";
            for (i=0;i<=7;i++)
            {
                txt=txt+"<tr>";
                for (j=0;j<=7;j++)
                {
                    if (occupy[i][j]==1) {txt=txt+"<td align='center'><div
style='height:20px;width:20px;cursor:default'>●</div></td>";}
                    else if (occupy[i][j]==2) {txt=txt+"<td align='center'><div
style='height:20px;width:20px;cursor:default'>○</div></td>";}
                    else if (occupy[i][j]==0) {txt=txt+"<td onclick='check("+i+","+j+",1)'><div
style='height:20px;width:20px;cursor:pointer'></div></td>";}
                }
```

```
            txt=txt+"</tr>";
        }
        txt=txt+"</table>";
        return txt;
}
function check(x,y,z)
{
        action=0;
        check_up(x,y,z);
        check_down(x,y,z);
        check_left(x,y,z);
        check_right(x,y,z);
        check_leftup(x,y,z);
        check_rightup(x,y,z);
        check_leftdown(x,y,z);
        check_rightdown(x,y,z);
        table=box();
        document.getElementById('box').innerHTML=table;
        battle();
        if ((z==1)&&(end_or_not))
        {
            if (lose_white()) {alert("computer can't eat you,your turn.");return;}
            if (action==1) {computer();}
        }
        if ((z==2)&&(end_or_not))
        {
            if (action==0) {alert("computer can't eat you,your turn.");}
            if (lose_black()) {computer();return;}
        }
        return;
}
function check_up(x,y,z)
{
        if (x==0) {return;}
        i=x;
        j=y;
        while ((i>=1)&&(occupy[i-1][j]==3-z))
        {
            i=i-1;
        }
        if ((i>=1)&&(occupy[i-1][j]==z)&&(i!=x))
        {
            kx=x;
            ky=y;
```

```
            while (kx>=i)
            {
                  action=1;
                  occupy[kx][ky]=z;
                  order[kx][ky]=0;
                  kx=kx-1;
            }
      }
      return;
}
function check_down(x,y,z)
{
      if (x==7) {return;}
      i=x;
      j=y;
      while ((i<=6)&&(occupy[i+1][j]==3-z))
      {
            i=i+1;
      }
      if ((i<=6)&&(occupy[i+1][j]==z)&&(i!=x))
      {
            kx=x;
            ky=y;
            while (kx<=i)
            {
                  action=1;
                  occupy[kx][ky]=z;
                  order[kx][ky]=0;
                  kx=kx+1;
            }
      }
      return;
}
function check_left(x,y,z)
{
      if (y==0) {return;}
      i=x;
      j=y;
      while ((j>=1)&&(occupy[i][j-1]==3-z))
      {
            j=j-1;
      }
      if ((j>=1)&&(occupy[i][j-1]==z)&&(j!=y))
      {
```

```
            kx=x;
            ky=y;
            while (ky>=j)
            {
                  action=1;
                  occupy[kx][ky]=z;
                  order[kx][ky]=0;
                  ky=ky-1;
            }
      }
      return;
}
function check_right(x,y,z)
{
      if (y==7) {return;}
      i=x;
      j=y;
      while ((j<=6)&&(occupy[i][j+1]==3-z))
      {
            j=j+1;
      }
      if ((j<=6)&&(occupy[i][j+1]==z)&&(j!=y))
      {
            kx=x;
            ky=y;
            while (ky<=j)
            {
                  action=1;
                  occupy[kx][ky]=z;
                  order[kx][ky]=0;
                  ky=ky+1;
            }
      }
      return;
}
function check_leftup(x,y,z)
{
      if ((x==0)||(y==0)) {return;}
      i=x;
      j=y;
      while ((i>=1)&&(j>=1)&&(occupy[i-1][j-1]==3-z))
      {
            i=i-1;
            j=j-1;
```

```
	}
	if ((i>=1)&&(j>=1)&&(occupy[i-1][j-1]==z)&&(i!=x)&&(j!=y))
	{
		kx=x;
		ky=y;
		while ((kx>=i)&&(ky>=j))
		{
			action=1;
			occupy[kx][ky]=z;
			order[kx][ky]=0;
			kx=kx-1;
			ky=ky-1;
		}
	}
	return;
}
function check_rightup(x,y,z)
{
	if ((x==0)||(y==7)) {return;}
	i=x;
	j=y;
	while ((i>=1)&&(j<=6)&&(occupy[i-1][j+1]==3-z))
	{
		i=i-1;
		j=j+1;
	}
	if ((i>=1)&&(j<=6)&&(occupy[i-1][j+1]==z)&&(i!=x)&&(j!=y))
	{
		kx=x;
		ky=y;
		while ((kx>=i)&&(ky<=j))
		{
			action=1;
			occupy[kx][ky]=z;
			order[kx][ky]=0;
			kx=kx-1;
			ky=ky+1;
		}
	}
	return;
}
function check_leftdown(x,y,z)
{
	if ((x==7)||(y==0)) {return;}
```

```
	i=x;
	j=y;
	while ((i<=6)&&(j>=1)&&(occupy[i+1][j-1]==3-z))
	{
		i=i+1;
		j=j-1;
	}
	if ((i<=6)&&(j>=1)&&(occupy[i+1][j-1]==z)&&(i!=x)&&(j!=y))
	{
		kx=x;
		ky=y;
		while ((kx<=i)&&(ky>=j))
		{
			action=1;
			occupy[kx][ky]=z;
			order[kx][ky]=0;
			kx=kx+1;
			ky=ky-1;
		}
	}
	return;
}
function check_rightdown(x,y,z)
{
	if ((x==7)||(y==7)) {return;}
	i=x;
	j=y;
	while ((i<=6)&&(j<=6)&&(occupy[i+1][j+1]==3-z))
	{
		i=i+1;
		j=j+1;
	}
	if ((i<=6)&&(j<=6)&&(occupy[i+1][j+1]==z)&&(i!=x)&&(j!=y))
	{
		kx=x;
		ky=y;
		while ((kx<=i)&&(ky<=j))
		{
			action=1;
			occupy[kx][ky]=z;
			order[kx][ky]=0;
			kx=kx+1;
			ky=ky+1;
		}
```

```
            }
            return;
        }
        function search_a(x,y,z)
        {
            total=0;
            total_up=search_up(x,y,z);
            total_down=search_down(x,y,z);
            total_left=search_left(x,y,z);
            total_right=search_right(x,y,z);
            total_leftup=search_leftup(x,y,z);
            total_rightup=search_rightup(x,y,z);
            total_leftdown=search_leftdown(x,y,z);
            total_rightdown=search_rightdown(x,y,z);

    total=total_up+total_down+total_left+total_right+total_leftup+total_rightup+total_leftdown+total_rightdow
n;
            return total;
        }
        function search_up(x,y,z)
        {
            sum=0;
            if (x==0) {return sum;}
            i=x;
            j=y;
            while ((i>=1)&&(occupy[i-1][j]==3-z))
            {
                i=i-1;
            }
            if ((i>=1)&&(occupy[i-1][j]==z)&&(i!=x))
            {
                kx=x-1;
                ky=y;
                while (kx>=i)
                {
                    sum=sum+1;
                    kx=kx-1;
                }
                return sum;
            }
            return sum;
        }
        function search_down(x,y,z)
        {
```

```
	sum=0;
	if (x==7) {return sum;}
	i=x;
	j=y;
	while ((i<=6)&&(occupy[i+1][j]==3-z))
	{
		i=i+1;
	}
	if ((i<=6)&&(occupy[i+1][j]==z)&&(i!=x))
	{
		kx=x+1;
		ky=y;
		while (kx<=i)
		{
			sum=sum+1;
			kx=kx+1;
		}
		return sum;
	}
	return sum;
}
function search_left(x,y,z)
{
	sum=0;
	if (y==0) {return sum;}
	i=x;
	j=y;
	while ((j>=1)&&(occupy[i][j-1]==3-z))
	{
		j=j-1;
	}
	if ((j>=1)&&(occupy[i][j-1]==z)&&(j!=y))
	{
		kx=x;
		ky=y-1;
		while (ky>=j)
		{
			sum=sum+1;
			ky=ky-1;
		}
		return sum;
	}
	return sum;
}
```

```
function search_right(x,y,z)
{
	sum=0;
	if (y==7) {return sum;}
	i=x;
	j=y;
	while ((j<=6)&&(occupy[i][j+1]==3-z))
	{
		j=j+1;
	}
	if ((j<=6)&&(occupy[i][j+1]==z)&&(j!=y))
	{
		kx=x;
		ky=y+1;
		while (ky<=j)
		{
			sum=sum+1;
			ky=ky+1;
		}
		return sum;
	}
	return sum;
}
function search_leftup(x,y,z)
{
	sum=0;
	if ((x==0)||(y==0)) {return sum;}
	i=x;
	j=y;
	while ((i>=1)&&(j>=1)&&(occupy[i-1][j-1]==3-z))
	{
		i=i-1;
		j=j-1;
	}
	if ((i>=1)&&(j>=1)&&(occupy[i-1][j-1]==z)&&(i!=x)&&(j!=y))
	{
		kx=x-1;
		ky=y-1;
		while ((kx>=i)&&(ky>=j))
		{
			sum=sum+1;
			kx=kx-1;
			ky=ky-1;
		}
```

```
            return sum;
        }
        return sum;
}
function search_rightup(x,y,z)
{
        sum=0;
        if ((x==0)||(y==7)) {return sum;}
        i=x;
        j=y;
        while ((i>=1)&&(j<=6)&&(occupy[i-1][j+1]==3-z))
        {
            i=i-1;
            j=j+1;
        }
        if ((i>=1)&&(j<=6)&&(occupy[i-1][j+1]==z)&&(i!=x)&&(j!=y))
        {
            kx=x-1;
            ky=y+1;
            while ((kx>=i)&&(ky<=j))
            {
                sum=sum+1;
                kx=kx-1;
                ky=ky+1;
            }
            return sum;
        }
        return sum;
}
function search_leftdown(x,y,z)
{
        sum=0;
        if ((x==7)||(y==0)) {return sum;}
        i=x;
        j=y;
        while ((i<=6)&&(j>=1)&&(occupy[i+1][j-1]==3-z))
        {
            i=i+1;
            j=j-1;
        }
        if ((i<=6)&&(j>=1)&&(occupy[i+1][j-1]==z)&&(i!=x)&&(j!=y))
        {
            kx=x+1;
            ky=y-1;
```

```
                while ((kx<=i)&&(ky>=j))
                {
                    sum=sum+1;
                    kx=kx+1;
                    ky=ky-1;
                }
                return sum;
            }
            return sum;
        }
        function search_rightdown(x,y,z)
        {
            sum=0;
            if ((x==7)||(y==7)) {return sum;}
            i=x;
            j=y;
            while ((i<=6)&&(j<=6)&&(occupy[i+1][j+1]==3-z))
            {
                i=i+1;
                j=j+1;
            }
            if ((i<=6)&&(j<=6)&&(occupy[i+1][j+1]==z)&&(i!=x)&&(j!=y))
            {
                kx=x+1;
                ky=y+1;
                while ((kx<=i)&&(ky<=j))
                {
                    sum=sum+1;
                    kx=kx+1;
                    ky=ky+1;
                }
                return sum;
            }
            return sum;
        }
    </script>
</head>
<body>
    <br>
    <div id="headline" align="center"><h1>小游戏-黑白棋</h1></div>
    <div id="box" align="center"><script type="text/javascript">document.write(box())</script></div>
    <div id="message" align="center"><script type="text/javascript">document.write(message(2,2))</script></div>
</body></html>
```