4.0最终版本（2021.7.11）
index.html

```
<!DOCTYPE html>
<html>
    <head>
        <title>2048小游戏    ——by:orall</title>
        <!-- 调用外部css（建议这个调用方法） -->
        <link rel="stylesheet" href="2048.css" type="text/css" />
        <script type="text/javascript" src="2048.js"></script>
    </head>
    <body onload = "main()">
    <!-- <div style="font-size:25px;"margin:0 auto; width:250px;
height:400px;border:1px solid #dbd4b5 "> -->
    <br><br><br><br><br><br>
    <div style="margin:0 auto; width:300px; height:400px;">
    <div style="text-align:left;color:#756d5f;font-size:40px;float:
left;">2048</div>

    <div id="score" style="line-height:55px;font-size:20px;text-
align:center;color:#6d675b;background-color: ;">分数:0</div>

    <div style="float: left;">合并相同方块，得到2048的方块!</div>
    <button onclick="main()" >New</button>


    <!-- 以下都是数字格子 -->

    <div id="11" class="gezi"></div>
    <div id="12" class="gezi"></div>
    <div id="13" class="gezi"></div>
    <div id="14" class="gezi"></div>


    <div style="clear:both;">
    <div id="21" class="gezi"></div>
    </div>
```

```
    <div id="22" class="gezi"></div>
    <div id="23" class="gezi"></div>
    <div id="24" class="gezi"></div>

    <div style="clear:both;">
    <div id="31" class="gezi"></div>
    </div>
    <div id="32" class="gezi"></div>
    <div id="33" class="gezi"></div>
    <div id="34" class="gezi"></div>

    <div style="clear:both;">
    <div id="41" class="gezi"></div>
    </div>
    <div id="42" class="gezi"></div>
    <div id="43" class="gezi"></div>
    <div id="44" class="gezi"></div>
    </div>
    </body>


    </html>
```

2048.css

```
body{
    background-color: #e2e2c4;
    height: 100%;
}

.gezi0{
    line-height:60px;           /* 垂直居中*/
    font-size:30px;
    background-color:#ccc0b2;
    text-align:center;          /* 水平居中*/
    color:#29291b;width:60px;
    height: 60px; float: left;
```

```
    border:3px solid rgb(160, 145, 124);

}

.gezi2{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#ede3d8;
    text-align:center;
    color:#786d62;
    width:60px;
    height: 60px;
    float: left;
    border:3px solid rgb(160, 145, 124);
}


.gezi4{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#eadfc8;
    text-align:center;
    color:#7c6f62;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}


.gezi8{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#f7b26f;
    text-align:center;
    color:#f6f4ec;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
```

```
}


.gezi16{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#f79251;
    text-align:center;
    color:#fffcf3;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi32{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#ff855c;
    text-align:center;
    color:#fffbf6;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi64{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#f86434;
    text-align:center;
    color:#fff3e4;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi128{
    font-weight: 700;
    line-height:60px;
```

```
    font-size:30px;
    background-color:#f7d367;
    text-align:center;
    color:#fffef7;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi256{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#f6c949;
    text-align:center;
    color:#fefefc;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi512{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#fbdb52;
    text-align:center;
    color:#fffee6;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}

.gezi1024{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#fed83e;
    text-align:center;
    color:#fffec7;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
```

```
}

.gezi2048{
    font-weight: 700;
    line-height:60px;
    font-size:30px;
    background-color:#fed52d;
    text-align:center;
    color:#ffffbe;width:60px;
    height: 60px; float: left;
    border:3px solid rgb(160, 145, 124);
}
```

2048.js

```
        var score = 0;
        //创建二维数组
        var map =new Array(4);
        for(var i=0;i<4;i++){
            map[i]=new Array(4);
        }
        //初始化
        for(var i = 0 ; i < 4 ; i++ ){
            for(var j = 0 ; j < 4 ; j++ ){
                map[i][j]= 0;
            }
        }

        var isnew = 0;              //是否需要创造

        function show() {
            document.getElementById("score").innerText = "分数:"+score;

            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {
                    if( map[i][j] == 0 ){

document.getElementById(String(i+1)+String(j+1)).className = "gezi0";
```

```
document.getElementById(String(i+1)+String(j+1)).innerText = " ";
                    }else{

document.getElementById(String(i+1)+String(j+1)).className =
"gezi"+String(map[i][j]);


document.getElementById(String(i+1)+String(j+1)).innerText = String(map[i]
[j]);
                    }


                }

            }

        }

        function judge() {
            var isblack = 0;
            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {
                    if (map[i][j] == 2048) {            //出现2048
                        confirm("游戏胜利"); //在页面上弹出确认对话框
                    }
                    if (map[i][j] != 0) {
                        isblack++;
                    }
                }
            }

            //看是否还可以继续进行有效移动
            var isfail = 1;
            if (isblack == 16) {
                //分析一列
                for (var j = 0; j < 4 && isfail != 0;  j++) {
                    var stk = new Array(4);                     //用数组表示
栈
```

```
                    var top = 0;
                    for (var i = 0; i < 4 && isfail != 0; i++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;             //说明还可以继续进行有效移
动
                            break;
                        }
                    }
                }

                //分析一行
                for (var i = 0; i < 4 && isfail != 0; i++) {
                    var stk = new Array(4);                     //用数组表示
栈
                    var top = 0;
                    for (var j = 0; j < 4 && isfail != 0; j++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;
                            break;
                        }
                    }
                }
                if (isfail == 1) {
                    confirm("游戏失败"); //在页面上弹出确认对话框
                }
            }
        }

        function move(model) {              //移动
```

```
            var isvalid = 0;
            if (model == 1) {       //up
                for (var j = 0; j < 4; j++) {       //对每一列进行分析
                    var stk = new Array(4);         //用数组表示栈
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }

                    var top = 0;                    //top表示还没东西的位置（top = size）
                    for (var i = 0; i < 4; i++) {

                        //将不是0的插入
                        if ( map[i][j] != 0) {       //利用栈来排列（栈原理）
                            //top = size
                            if ( top  == 0 ||  stk[top-1] != map[i][j]) {  //top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top-1] = map[i][j] * 2;
                            }


                        }
                    }
                    if ( top != 4) {          //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
```

```
                        stk[top] = 0;
                        top++;
                    }*/

                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[k][j] = stk[k];
                    }
                }
            }else if( model == 2 ){          //down
                for (var j = 0; j < 4; j++) {        //对每一列进行分析
                    var stk = new Array(4);
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }
                    var top = 0;

                    for (var i = 3; i >= 0 ; i--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {            //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[3 - k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
```

```
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }
                    */
                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[3-k][j] = stk[k];
                    }
                }
            }
            else if ( model == 3) {          //left
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0 ;
                    for (var j = 0; j < 4; j++) {
                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][k]  != stk[k]) {
```

```
                                isvalid = 1;
                            }
                        }

                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }*/

                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[i][k] = stk[k];
                    }
                }
            }
            else if ( model == 4) {                //right
                for (var i = 0; i < 4; i++) {            //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0;
                    for (var j = 3; j >= 0; j--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {            //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;

                            }
```

```
                    }
                }
                if (top != 4) {           //如果不等于4，说明是有效移动
                     //跟原来比较一下
                    for (var k = 0; k < top; k++) {
                        if (map[i][3 - k] != stk[k]) {
                            isvalid = 1;
                        }
                    }
                }
                /*
                for (var k = top; k < 4; k++)
                {
                    stk[top-1] = 0;
                    top++;
                }*/

                //将结果放入map中
                for (var k = 0; k < 4; k++) {
                    map[i][3-k] = stk[k];
                }

            }
        }
        return isvalid;

    }

    function up() {
        if (move(1)) {            //如果能进行有效移动，则创造新的数字
            new_num();
        }
    }

    function down() {
        if (move(2)) {
            new_num();
        }
    }
```

```
function left() {
    if (move(3)) {
        new_num();
    }
}

function right() {
    if (move(4)) {
        new_num();
    }
}

function new_num() {
    //开始时棋盘内随机出现两个数字，出现的数字仅可能为2或4。
    //每有效移动一步，棋盘的空位(无数字处)随机出现一个数字(依然可能为2或4)。

    var isok = 0;
    while (isok != 1) {
        //Math.floor(Math.random()*(3-0+1)+0);
        var pos_x = Math.floor(Math.random()*(3-0+1)+0);
        var pos_y = Math.floor(Math.random()*(3-0+1)+0);
        if (map[pos_x][pos_y] == 0 ) {
            if( Math.floor(Math.random()*(1-0+1)+0) == 0 ){
                 map[pos_x][pos_y] = 2;
            }else{
                map[pos_x][pos_y] = 4;
            }
            isok = 1;
        }
    }


    return;
}


document.onkeydown=function(event){
            var e = event || window.event ||
```

```
arguments.callee.caller.arguments[0];
                    if(e && e.keyCode==65){ // 按a
                        //要做的事情
                        left();
                        judge();              //判断游戏是否成功或者失败
                        show();
                      }
                    if(e && e.keyCode==68){ // 按d
                         //要做的事情
                         right();
                         judge();              //判断游戏是否成功或者失败
                         show();
                    }
                    if(e && e.keyCode==87){ //w
                         //要做的事情
                         up();
                         judge();              //判断游戏是否成功或者失败
                         show();
                    }
                    if(e && e.keyCode==83){ //s
                         //要做的事情
                         down();
                         judge();              //判断游戏是否成功或者失败
                         show();
                    }

                };


        function main() {
            for(var i = 0 ; i < 4 ; i++ ){
                for(var j = 0 ; j < 4 ; j++ ){
                    map[i][j]= 0;
                }
            }
            new_num();
            show();
        }
```

3.0(2021.7.10)

```
<!-- 版本：2048小游戏3.0 -->
<!-- 时间：2021.7.10    -->
<!--         ——by:orall -->
<!-- 版本区别：1.完善了背景颜色；2.字体垂直居中-->
<!-- 待完善：数字颜色      -->



<html>
    <head>
        <title>2048小游戏    ——by:orall</title>
    </head>


    <body onload = "main()" style="background-color: #e2e2c4;">
    <!-- <div style="font-size:25px;"margin:0 auto; width:250px;
height:400px;border:1px solid #dbd4b5 "> -->
    <div style="margin:0 auto; width:300px; height:400px;">
    <h2 style="text-align:center;">2048小游戏</h2>
    <button onclick="main()">New</button>
    <div id="score" style="font-size:25px;text-align:center;">得分:0</div>
    <!-- 以下都是数字格子 -->

    <div id="11" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="12" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="13" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="14" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
```

```
float: left;border:3px solid rgb(160, 145, 124);"></div>

    <!-- 注意第一个的区别  -->
    <div id="21" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;clear:both;text-
align:center;color:#29291b;width:60px;height: 60px; float: left;border:3px
solid rgb(160, 145, 124);"></div>
    <div id="22" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="23" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="24" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>

    <div id="31" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;clear:both;text-
align:center;color:#29291b;width:60px;height: 60px; float: left;border:3px
solid rgb(160, 145, 124);"></div>
    <div id="32" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="33" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="34" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>

    <div id="41" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;clear:both;text-
align:center;color:#29291b;width:60px;height: 60px; float: left;border:3px
solid rgb(160, 145, 124);"></div>
    <div id="42" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="43" style="line-height:60px;font-size:25px;background-
```

```
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    <div id="44" style="line-height:60px;font-size:25px;background-
color:#dbd4b5;text-align:center;color:#29291b;width:60px;height: 60px;
float: left;border:3px solid rgb(160, 145, 124);"></div>
    </div>
    </body>


    </html>

    <script type="text/javascript">


        var score = 0;
        //创建二维数组
        var map =new Array(4);
        for(var i=0;i<4;i++){
           map[i]=new Array(4);
        }
        //初始化
        for(var i = 0 ; i < 4 ; i++ ){
            for(var j = 0 ; j < 4 ; j++ ){
                map[i][j]= 0;
            }
        }

        var isnew = 0;              //是否需要创造

        function show() {
            document.getElementById("score").innerText = "得分:"+score;

            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {

                    if( map[i][j] == 0 ){

document.getElementById(String(i+1)+String(j+1)).innerText = " ";
                    }else{
```

```
document.getElementById(String(i+1)+String(j+1)).innerText = String(map[i][j]);
                    }

                }

            }

        }

        function judge() {
            var isblack = 0;
            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {
                    if (map[i][j] == 2048) {            //出现2048
                        confirm("游戏胜利"); //在页面上弹出确认对话框
                    }
                    if (map[i][j] != 0) {
                        isblack++;
                    }
                }
            }

            //看是否还可以继续进行有效移动
            var isfail = 1;
            if (isblack == 16) {
                //分析一列
                for (var j = 0; j < 4 && isfail != 0;  j++) {
                    var stk = new Array(4);                     //用数组表示栈
                    var top = 0;
                    for (var i = 0; i < 4 && isfail != 0; i++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
```

```
                            isfail = 0;                 //说明还可以继续进行有效移
动
                            break;
                        }
                    }
                }

                //分析一行
                for (var i = 0; i < 4 && isfail != 0; i++) {
                    var stk = new Array(4);                         //用数组表示
栈
                    var top = 0;
                    for (var j = 0; j < 4 && isfail != 0; j++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;
                            break;
                        }
                    }
                }
                if (isfail == 1) {
                    confirm("游戏失败"); //在页面上弹出确认对话框
                }
            }
        }

        function move(model) {                  //移动
            var isvalid = 0;
            if (model == 1) {       //up
                for (var j = 0; j < 4; j++) {       //对每一列进行分析
                    var stk = new Array(4);         //用数组表示栈
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }
```

```
                    var top = 0;                    //top表示还没东西的位置
（top = size）
                    for (var i = 0; i < 4; i++) {

                        //将不是0的插入
                        if ( map[i][j] != 0) {        //利用栈来排列（栈原理）
                            //top = size
                            if ( top  == 0 ||  stk[top-1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top-1] = map[i][j] * 2;
                            }


                        }
                    }
                    if ( top != 4) {           //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top] = 0;
                        top++;
                    }*/

                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[k][j] = stk[k];
                    }
```

```
                }
            }else if( model == 2 ){          //down
                for (var j = 0; j < 4; j++) {        //对每一列进行分析
                    var stk = new Array(4);
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }
                    var top = 0;

                    for (var i = 3; i >= 0 ; i--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {            //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {           //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[3 - k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }
                    */
```

```
                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[3-k][j] = stk[k];
                    }
                }
            }
            else if ( model == 3) {          //left
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0 ;
                    for (var j = 0; j < 4; j++) {
                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][k]  != stk[k]) {
                                isvalid = 1;
                            }
                        }

                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
```

```
                        stk[top-1] = 0;
                        top++;
                    }*/

                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[i][k] = stk[k];
                    }
                }
            }
            else if ( model == 4) {              //right
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0;
                    for (var j = 3; j >= 0; j--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;

                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][3 - k] != stk[k]) {
                                isvalid = 1;
                            }
```

```
                    }
                }
                /*
                for (var k = top; k < 4; k++)
                {
                    stk[top-1] = 0;
                    top++;
                }*/

                //将结果放入map中
                for (var k = 0; k < 4; k++) {
                    map[i][3-k] = stk[k];
                }

            }
        }
        return isvalid;

    }

    function up() {
        if (move(1)) {            //如果能进行有效移动，则创造新的数字
            new_num();
        }
    }

    function down() {
        if (move(2)) {
            new_num();
        }
    }

    function left() {
        if (move(3)) {
            new_num();
        }
    }

    function right() {
```

```
            if (move(4)) {
                new_num();
            }
        }

        function new_num() {
            //开始时棋盘内随机出现两个数字，出现的数字仅可能为2或4。
            //每有效移动一步，棋盘的空位(无数字处)随机出现一个数字(依然可能为2或4)。

            var isok = 0;
            while (isok != 1) {
                //Math.floor(Math.random()*(3-0+1)+0);
                var pos_x = Math.floor(Math.random()*(3-0+1)+0);
                var pos_y = Math.floor(Math.random()*(3-0+1)+0);
                if (map[pos_x][pos_y] == 0 ) {
                    if( Math.floor(Math.random()*(1-0+1)+0) == 0 ){
                         map[pos_x][pos_y] = 2;
                    }else{
                        map[pos_x][pos_y] = 4;
                    }
                    isok = 1;
                }
            }


            return;
        }


        document.onkeydown=function(event){
                    var e = event || window.event ||
arguments.callee.caller.arguments[0];
                    if(e && e.keyCode==65){ // 按a
                        //要做的事情
                        left();
                        judge();              //判断游戏是否成功或者失败
                        show();
                      }
                    if(e && e.keyCode==68){ // 按d
```

```
                    //要做的事情
                    right();
                    judge();              //判断游戏是否成功或者失败
                    show();
                }
                if(e && e.keyCode==87){ //w
                    //要做的事情
                    up();
                    judge();              //判断游戏是否成功或者失败
                    show();
                }
                if(e && e.keyCode==83){ //s
                    //要做的事情
                    down();
                    judge();              //判断游戏是否成功或者失败
                    show();
                }

            };


    function main() {
        for(var i = 0 ; i < 4 ; i++ ){
            for(var j = 0 ; j < 4 ; j++ ){
                map[i][j]= 0;
            }
        }
        new_num();
        show();
    }



</script>
```

2048小游戏2.0(2021.7.6)

```
<!-- 版本：2048小游戏2.0 -->
<!-- 时间：2021.7.6      -->
<!--         ——by:orall -->
<!-- 版本区别：1.增加了重新开始游戏和分数；2.添加了背景颜色和字体大小-->


<html>
    <head>
        <title>2048小游戏    ——by:orall</title>
    </head>


    <body onload = "main()" style="background-color: #5d6d7e;">
    <!-- <div style="font-size:25px;"margin:0 auto; width:250px;
height:400px;border:1px solid #ffcc00 "> -->
    <div style="margin:0 auto; width:250px; height:400px;">
    <h2 style="text-align:center;">2048小游戏</h2>
    <button onclick="main()">New</button>
    <!-- 以下都是数字格子 -->
    <div id="score" style="font-size:25px;text-
align:center;">score:0</div>
    <div id="11" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="12" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="13" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="14" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>

    <!-- 注意第一个的区别  -->
    <div id="21" style="font-size:25px;background-
color:#ffcc00;clear:both;text-
```

```
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="22" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="23" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="24" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>

    <div id="31" style="font-size:25px;background-
color:#ffcc00;clear:both;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="32" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="33" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="34" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>

    <div id="41" style="font-size:25px;background-
color:#ffcc00;clear:both;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="42" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="43" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
    <div id="44" style="font-size:25px;background-color:#ffcc00;text-
align:center;color:#3399ff;width:60px;height: 60px; float: left;border:1px
solid #000;"></div>
```

```
    </div>
    </body>


    </html>

    <script type="text/javascript">


        var score = 0;
        //创建二维数组
        var map =new Array(4);
        for(var i=0;i<4;i++){
            map[i]=new Array(4);
        }
        //初始化
        for(var i = 0 ; i < 4 ; i++ ){
            for(var j = 0 ; j < 4 ; j++ ){
                map[i][j]= 0;
            }
        }

        var isnew = 0;               //是否需要创造

        function show() {
            document.getElementById("score").innerText = "score:"+score;

            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {

                    if( map[i][j] == 0 ){

document.getElementById(String(i+1)+String(j+1)).innerText = " ";
                    }else{

document.getElementById(String(i+1)+String(j+1)).innerText = String(map[i]
[j]);
                    }
```

```
                }

            }

        }

        function judge() {
            var isblack = 0;
            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {
                    if (map[i][j] == 2048) {            //出现2048
                        confirm("游戏胜利"); //在页面上弹出确认对话框
                    }
                    if (map[i][j] != 0) {
                        isblack++;
                    }
                }
            }

            //看是否还可以继续进行有效移动
            var isfail = 1;
            if (isblack == 16) {
                //分析一列
                for (var j = 0; j < 4 && isfail != 0;  j++) {
                    var stk = new Array(4);                     //用数组表示
栈
                    var top = 0;
                    for (var i = 0; i < 4 && isfail != 0; i++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;             //说明还可以继续进行有效移
动
                            break;
                        }
                    }
```

```
                }

                //分析一行
                for (var i = 0; i < 4 && isfail != 0; i++) {
                    var stk = new Array(4);                         //用数组表示
栈
                    var top = 0;
                    for (var j = 0; j < 4 && isfail != 0; j++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;
                            break;
                        }
                    }
                }
                if (isfail == 1) {
                    confirm("游戏失败"); //在页面上弹出确认对话框
                }
            }
        }

        function move(model) {                //移动
            var isvalid = 0;
            if (model == 1) {       //up
                for (var j = 0; j < 4; j++) {        //对每一列进行分析
                    var stk = new Array(4);          //用数组表示栈
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }

                    var top = 0;                     //top表示还没东西的位置
（top = size）
                    for (var i = 0; i < 4; i++) {

                        //将不是0的插入
```

```
                    if ( map[i][j] != 0) {        //利用栈来排列（栈原理）
                        //top = size
                        if ( top  == 0 ||  stk[top-1] != map[i][j]) {
//top != map[i][j]

                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            score += map[i][j] * 2;
                            stk[top-1] = map[i][j] * 2;
                        }


                    }
                }
                if ( top != 4) {           //如果不等于4，说明是有效移动
                    //跟原来比较一下
                    for (var k = 0; k < top; k++) {
                        if (map[k][j] != stk[k]) {
                            isvalid = 1;
                        }
                    }
                }
                /*
                for (var k = top; k < 4; k++)
                {
                    stk[top] = 0;
                    top++;
                }*/

                //将结果放入map中
                for (var k = 0; k < 4; k++) {
                    map[k][j] = stk[k];
                }
            }
        }else if( model == 2 ){          //down
            for (var j = 0; j < 4; j++) {        //对每一列进行分析
                var stk = new Array(4);
                for(var i = 0 ; i < 4 ; i++ ){
```

```
                        stk[i] = 0;
                    }
                    var top = 0;

                    for (var i = 3; i >= 0 ; i--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {            //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[3 - k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }
                    */
                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[3-k][j] = stk[k];
                    }
                }
```

```
            }
            else if ( model == 3) {          //left
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0 ;
                    for (var j = 0; j < 4; j++) {
                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][k]  != stk[k]) {
                                isvalid = 1;
                            }
                        }

                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }*/

                    //将结果放入map中
```

```
                    for (var k = 0; k < 4; k++) {
                        map[i][k] = stk[k];
                    }
                }
            }
            else if ( model == 4) {              //right
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0;
                    for (var j = 3; j >= 0; j--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                score += map[i][j] * 2;
                                stk[top - 1] = map[i][j] * 2;

                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][3 - k] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
```

```
                    stk[top-1] = 0;
                    top++;
                }*/

                //将结果放入map中
                for (var k = 0; k < 4; k++) {
                    map[i][3-k] = stk[k];
                }

            }
        }
        return isvalid;

    }

    function up() {
        if (move(1)) {            //如果能进行有效移动，则创造新的数字
            new_num();
        }
    }

    function down() {
        if (move(2)) {
            new_num();
        }
    }

    function left() {
        if (move(3)) {
            new_num();
        }
    }

    function right() {
        if (move(4)) {
            new_num();
        }
    }
```

```
        function new_num() {
            //开始时棋盘内随机出现两个数字，出现的数字仅可能为2或4。
            //每有效移动一步，棋盘的空位(无数字处)随机出现一个数字(依然可能为2或4)。

            var isok = 0;
            while (isok != 1) {
                Math.floor(Math.random()*(3-0+1)+0);
                var pos_x = Math.floor(Math.random()*(3-0+1)+0);
                var pos_y = Math.floor(Math.random()*(3-0+1)+0);
                if (map[pos_x][pos_y] == 0 ) {
                    if( Math.floor(Math.random()*(1-0+1)+0) == 0 ){
                         map[pos_x][pos_y] = 2;
                    }else{
                        map[pos_x][pos_y] = 4;
                    }
                    isok = 1;
                }
            }


            return;
        }


        document.onkeydown=function(event){
                    var e = event || window.event ||
arguments.callee.caller.arguments[0];
                    if(e && e.keyCode==65){ // 按a
                        //要做的事情
                        left();
                        judge();              //判断游戏是否成功或者失败
                        show();
                      }
                    if(e && e.keyCode==68){ // 按d
                         //要做的事情
                         right();
                         judge();              //判断游戏是否成功或者失败
                         show();
                    }
```

```
                    if(e && e.keyCode==87){ //w
                        //要做的事情
                        up();
                        judge();              //判断游戏是否成功或者失败
                        show();
                    }
                    if(e && e.keyCode==83){ //s
                        //要做的事情
                        down();
                        judge();              //判断游戏是否成功或者失败
                        show();
                    }

                };


        function main() {
            for(var i = 0 ; i < 4 ; i++ ){
                for(var j = 0 ; j < 4 ; j++ ){
                    map[i][j]= 0;
                }
            }
            new_num();
            show();
        }



    </script>
```

(2021.7.6)
直接根据C++做的2048雏形进行修改

```
<!-- 版本：2048小游戏1.0 -->
<!-- 时间：2021.7.6     -->
<!--         ——by:orall -->

<html>
```

```
    <head>
        <title>2048小游戏    ——by:orall</title>
    </head>


    <body onload = "main()">
    <h2>2048小游戏</h2>
    <div>
    <div id="11" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="12" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="13" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="14" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>

    <!-- 注意第一个的区别  -->
    <div id="21" style="clear:both;text-
align:center;color:#00FF00;width:30px;height: 30px; float: left;border:1px
solid #000;"></div>
    <div id="22" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="23" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="24" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>

    <div id="31" style="clear:both;text-
align:center;color:#00FF00;width:30px;height: 30px; float: left;border:1px
solid #000;"></div>
    <div id="32" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="33" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="34" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>

    <div id="41" style="clear:both;text-
```

```
align:center;color:#00FF00;width:30px;height: 30px; float: left;border:1px
solid #000;"></div>
    <div id="42" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="43" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    <div id="44" style="text-align:center;color:#00FF00;width:30px;height:
30px; float: left;border:1px solid #000;"></div>
    </div>
    </body>


    </html>

    <script type="text/javascript">

        //创建二维数组
        var map =new Array(4);
        for(var i=0;i<4;i++){
            map[i]=new Array(4);
        }
        //初始化
        for(var i = 0 ; i < 4 ; i++ ){
            for(var j = 0 ; j < 4 ; j++ ){
                map[i][j]= 0;
            }
        }


        var isnew = 0;              //是否需要创造

        function show() {
            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {

                    if( map[i][j] == 0 ){

document.getElementById(String(i+1)+String(j+1)).innerText = " ";
                    }else{
```

```
document.getElementById(String(i+1)+String(j+1)).innerText = String(map[i][j]);
                    }

                }

            }

        }

        function judge() {
            var isblack = 0;
            for (var i = 0; i < 4; i++) {
                for (var j = 0; j < 4; j++) {
                    if (map[i][j] == 2048) {            //出现2048
                        confirm("游戏胜利"); //在页面上弹出确认对话框
                    }
                    if (map[i][j] != 0) {
                        isblack++;
                    }
                }
            }

            //看是否还可以继续进行有效移动
            var isfail = 1;
            if (isblack == 16) {
                //分析一列
                for (var j = 0; j < 4 && isfail != 0;  j++) {
                    var stk = new Array(4);                     //用数组表示栈
                    var top = 0;
                    for (var i = 0; i < 4 && isfail != 0; i++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
```

```
                            isfail = 0;                 //说明还可以继续进行有效移
动
                            break;
                        }
                    }
                }

                //分析一行
                for (var i = 0; i < 4 && isfail != 0; i++) {
                    var stk = new Array(4);                         //用数组表示
栈
                    var top = 0;
                    for (var j = 0; j < 4 && isfail != 0; j++) {
                        if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                            stk[top] = map[i][j];
                            top++;
                        }
                        else {
                            isfail = 0;
                            break;
                        }
                    }
                }
                if (isfail == 1) {
                    confirm("游戏失败"); //在页面上弹出确认对话框
                }
            }
        }

        function move(model) {                  //移动
            var isvalid = 0;
            if (model == 1) {       //up
                for (var j = 0; j < 4; j++) {       //对每一列进行分析
                    var stk = new Array(4);         //用数组表示栈
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }
```

```
                    var top = 0;                    //top表示还没东西的位置（top = size）
                    for (var i = 0; i < 4; i++) {

                        //将不是0的插入
                        if ( map[i][j] != 0) {        //利用栈来排列（栈原理）
                            //top = size
                            if ( top  == 0 ||  stk[top-1] != map[i][j]) {  //top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                stk[top-1] = map[i][j] * 2;
                            }


                        }
                    }
                    if ( top != 4) {           //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top] = 0;
                        top++;
                    }*/

                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[k][j] = stk[k];
                    }
                }
```

```
            }else if( model == 2 ){          //down
                for (var j = 0; j < 4; j++) {        //对每一列进行分析
                    var stk = new Array(4);
                    for(var i = 0 ; i < 4 ; i++ ){
                        stk[i] = 0;
                    }
                    var top = 0;

                    for (var i = 3; i >= 0 ; i--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {            //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {           //如果不等于4，说明是有效移动
                        //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[3 - k][j] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }
                    */
                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
```

```
                        map[3-k][j] = stk[k];
                    }
                }
            }
            else if ( model == 3) {          //left
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0 ;
                    for (var j = 0; j < 4; j++) {
                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]
                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][k]  != stk[k]) {
                                isvalid = 1;
                            }
                        }

                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
                        stk[top-1] = 0;
                        top++;
                    }*/
```

```
                    //将结果放入map中
                    for (var k = 0; k < 4; k++) {
                        map[i][k] = stk[k];
                    }
                }
            }
            else if ( model == 4) {               //right
                for (var i = 0; i < 4; i++) {           //对每一行进行分析
                    var stk = new Array(4);
                    for(var k = 0 ; k < 4 ; k++ ){
                        stk[k] = 0;
                    }
                    var top = 0;
                    for (var j = 3; j >= 0; j--) {

                        //将不是0的插入
                        if (map[i][j] != 0) {           //利用栈来排列
                            if (top == 0 || stk[top - 1] != map[i][j]) {
//top != map[i][j]

                                stk[top] = map[i][j];
                                top++;
                            }
                            else {
                                stk[top - 1] = map[i][j] * 2;
                            }
                        }
                    }
                    if (top != 4) {          //如果不等于4，说明是有效移动
                         //跟原来比较一下
                        for (var k = 0; k < top; k++) {
                            if (map[i][3 - k] != stk[k]) {
                                isvalid = 1;
                            }
                        }
                    }
                    /*
                    for (var k = top; k < 4; k++)
                    {
```

```
                    stk[top-1] = 0;
                    top++;
                }*/

                //将结果放入map中
                for (var k = 0; k < 4; k++) {
                    map[i][3-k] = stk[k];
                }

            }
        }
        return isvalid;

    }

    function up() {
        if (move(1)) {            //如果能进行有效移动，则创造新的数字
            new_num();
        }
    }

    function down() {
        if (move(2)) {
            new_num();
        }
    }

    function left() {
        if (move(3)) {
            new_num();
        }
    }

    function right() {
        if (move(4)) {
            new_num();
        }
    }
```

```
        function new_num() {
            //开始时棋盘内随机出现两个数字，出现的数字仅可能为2或4。
            //每有效移动一步，棋盘的空位(无数字处)随机出现一个数字(依然可能为2或4)。

            var isok = 0;
            while (isok != 1) {
                Math.floor(Math.random()*(3-0+1)+0);
                var pos_x = Math.floor(Math.random()*(3-0+1)+0);
                var pos_y = Math.floor(Math.random()*(3-0+1)+0);
                if (map[pos_x][pos_y] == 0 ) {
                    if( Math.floor(Math.random()*(1-0+1)+0) == 0 ){
                         map[pos_x][pos_y] = 2;
                    }else{
                        map[pos_x][pos_y] = 4;
                    }
                    isok = 1;
                }
            }


            return;
        }


        document.onkeydown=function(event){
                    var e = event || window.event ||
arguments.callee.caller.arguments[0];
                    if(e && e.keyCode==65){ // 按a
                        //要做的事情
                        left();
                        judge();              //判断游戏是否成功或者失败
                        show();
                      }
                    if(e && e.keyCode==68){ // 按d
                         //要做的事情
                         right();
                         judge();              //判断游戏是否成功或者失败
                         show();
                    }
```

```
                if(e && e.keyCode==87){ //w
                    //要做的事情
                    up();
                    judge();            //判断游戏是否成功或者失败
                    show();
                }
                if(e && e.keyCode==83){ //s
                    //要做的事情
                    down();
                    judge();            //判断游戏是否成功或者失败
                    show();
                }

            };


    function main() {
        new_num();
        show();
    }



</script>
```