****

**实**

**验**

**报**

**告**

**课 程 Web开发**

**题 目 拼图游戏**

**分 数**

**学院名称 智能与计算学部**

**专 业 软件工程**

**学生姓名 刘兴宇**

**学 号 3017218063**

**年 级 2017**

**班 级 1**

目录

[一、实验目的 1](#_Toc25846672)

[二、实验内容 1](#_Toc25846673)

[三、实验步骤 1](#_Toc25846674)

[四、实验结果 6](#_Toc25846675)

[五、实验结论 8](#_Toc25846676)

[六、源代码 8](#_Toc25846677)

## 实验目的

本次实验目为练习JavaScript的程序编写，以实现Web页面的交互功能

## 二、实验内容

制作一个拼图游戏（3\*3）

## 三、实验步骤

1. 确定需求

* 一个拼图游戏要实现一个3\*3的网格布局，其中一块为空白，剩余8块可以自由移动，当8块位置正确后，再添加剩余的一块，就可以完成拼图
* 要有完整图片的提示信息
* 记录游戏时间、步数
* 可以随机打乱拼图

2. 设计页面原型

根据需求，设计界面原型。

* 上方居中显示标题和
* 主题部分分为两大块：游戏区和提示区。游戏区为一个3\*3网格，每个格中填充一张图片块，其中一个块为空白。点击上、下、左、右四个方向有空白块的图片可以向空白块移动该图片。提示区上方为完整图片，下面是计时和步数。在最下面有重新开始的按钮和添加最后一块（检查胜利）的按钮，如果在所有图片位置正确时点击，则会自动填充最后一块，结束计时，显示成功的对勾提示。否则会显示错误提示。
* 底端显示作者信息
* 设计响应式页面以适应所有宽度的设备

3. 实现布局

编写index.html、index.css

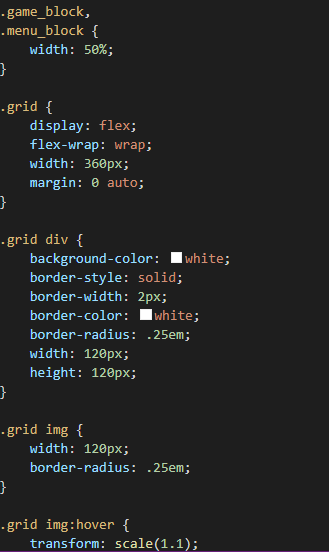
关键技术：

* 游戏区



游戏区game\_block中存在网格块grid，其中包含九个子块p0-p8，每个子块中有一张图片。

样式使用flex布局，设置固定宽度使得每行显示三个块，且中间不保留空隙

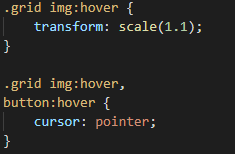


* 菜单区

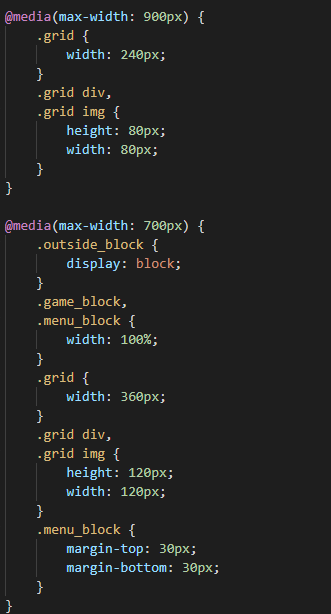


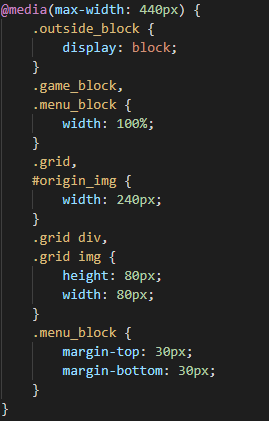
样式不做过多赘述

* 伪类动作



* + 对图片、按钮，使用:hover {cursor: pointer}使的当鼠标悬停时，变为指针
  + 对拼图图片，使用:hover{transform.scale(1.1)}设置鼠标悬停在图片上方时，图片略微放大以指示该图片
* 响应式设计





使用了媒体查询，确保在页面宽度改变时拼图显示正常

同时在<head>中设置viewport视窗，限制移动端的视窗长度

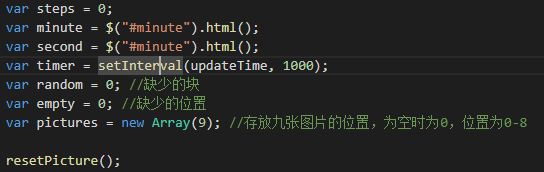
<meta name="viewport" cotent="width=device-width">

4. 实现交互

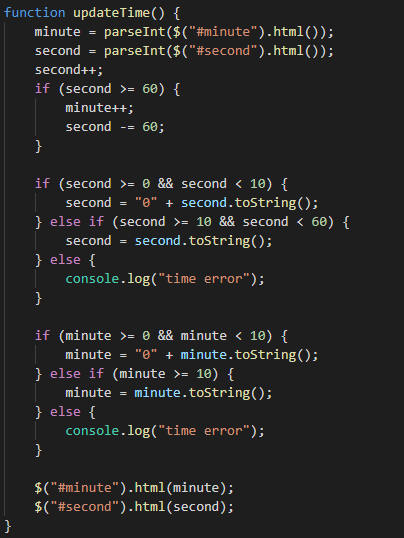
编写index.js，使用jQuery编写

核心交互功能

* 初始化页面：声明变量，启动定时器，初始化拼图

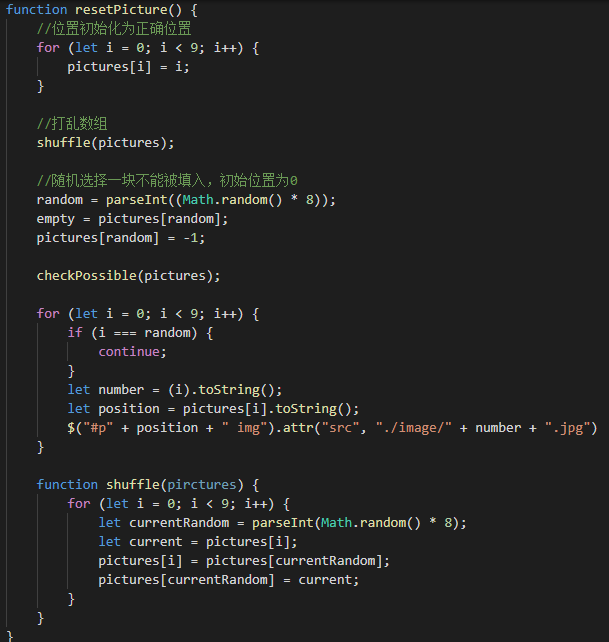


Steps记录步数，minute、second为HTML页面上显示的时间，timer是计时器，每1秒执行updateTime函数，更新页面上的时间：（判断秒数进位分钟，个位数的分、秒显示为两位）



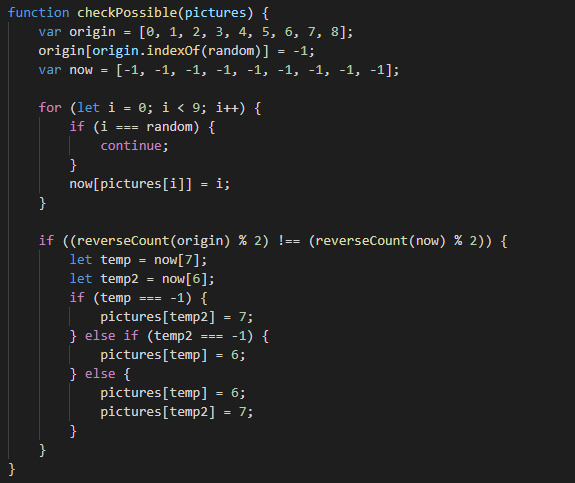
Random为游戏中缺少的图片块（0-8）编号，empty为游戏中网格当前为空白的块（0-8）编号，pictures是一个数组，下标为图片块编号，值为图片块当前的位置编号，如果图片块缺少，值为-1

resetPicture函数对拼图重新进行排序



首先先将pictures数组初始化为正确顺序，然后再调用shuffle函数按照随机交换方法打乱。之后，生成一个0-8的随机数，作为缺少的图片块的编号，并且更新empty、pictures的值

之后，要检查当前生成的随机打乱是否有解，调用checkPossible函数



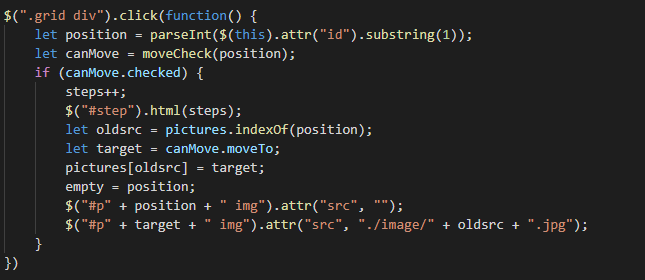
原理为，首先建立棋盘数组，存储网格中每个块放置的图片块编号。初始数组为正确顺序（缺少的块编号为-1），打乱数组为刚才打乱后的顺序。如果一个随机打乱有解，那么对于初始数组和打乱数组的逆序数，其奇偶性应该相同。如果不同，则将打乱数组倒数第2、3个元素对调，即可变为可行解。

最后，根据打乱顺序设置图片

* 操作拼图

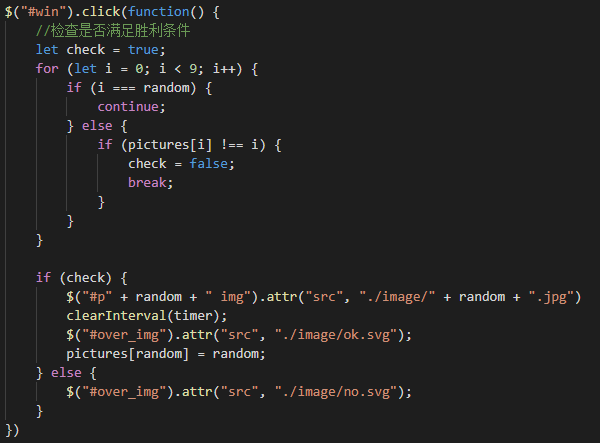
首先判断点击块是否可以移动

如果可以，获取目标位置，交换之，并记录步数



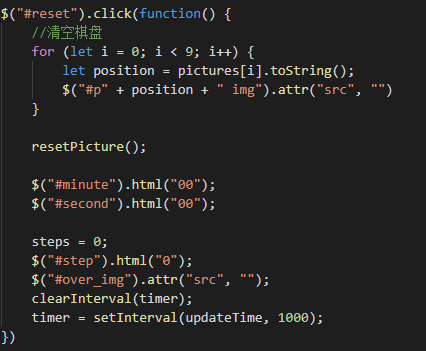
* 获胜

判断除去缺少块以外的其它块是否在正确位置上，如果在，添加缺少图片，显示对勾，停止计时器，补全pictures数组。否则，显示x



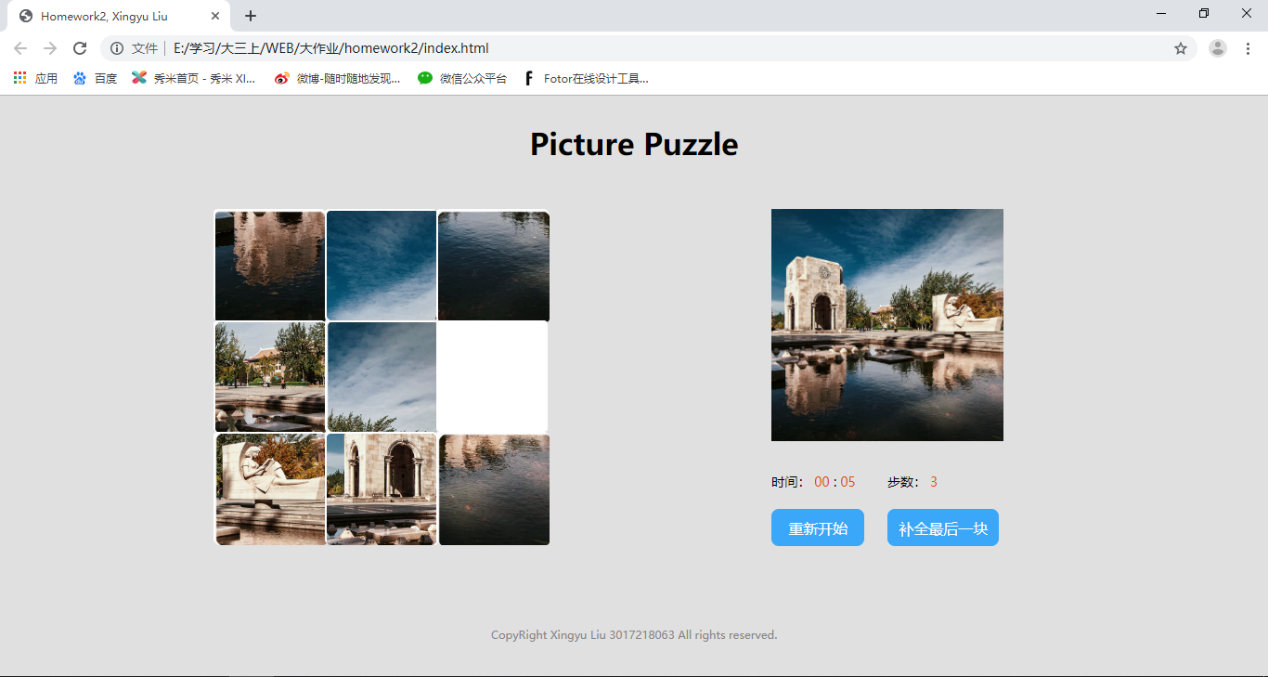
* 重置页面

刷新布局，重设定时器和计步器

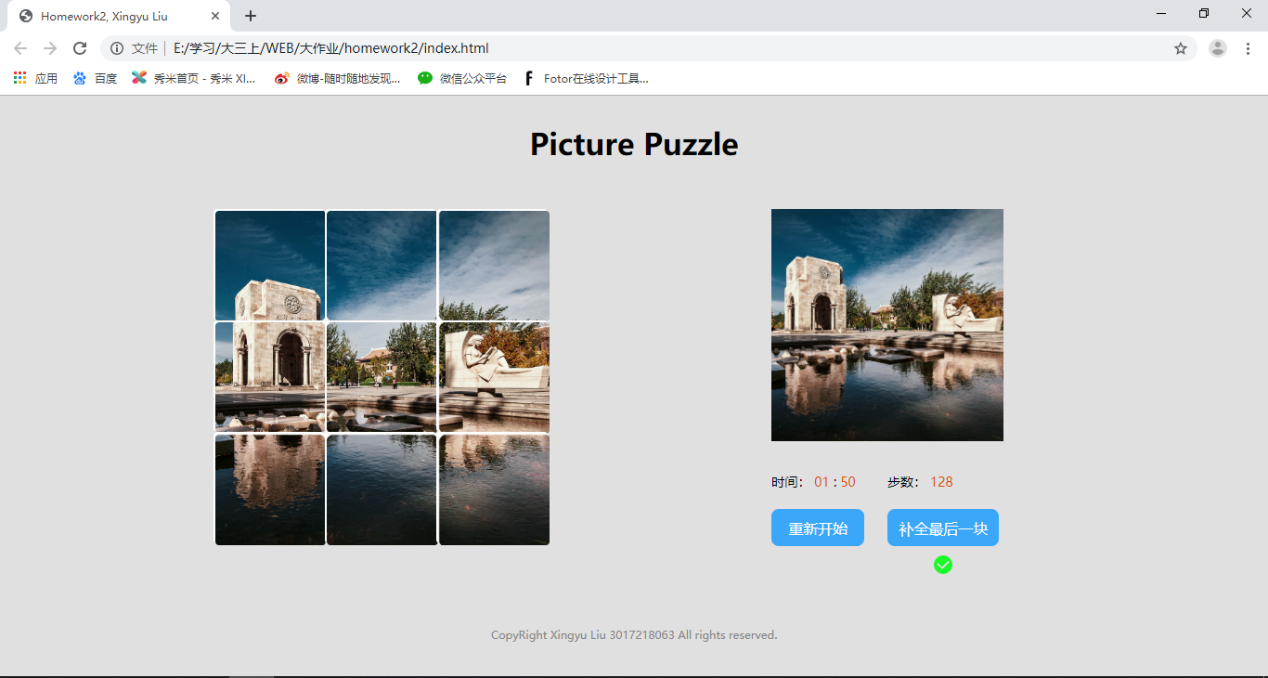


## 四、实验结果

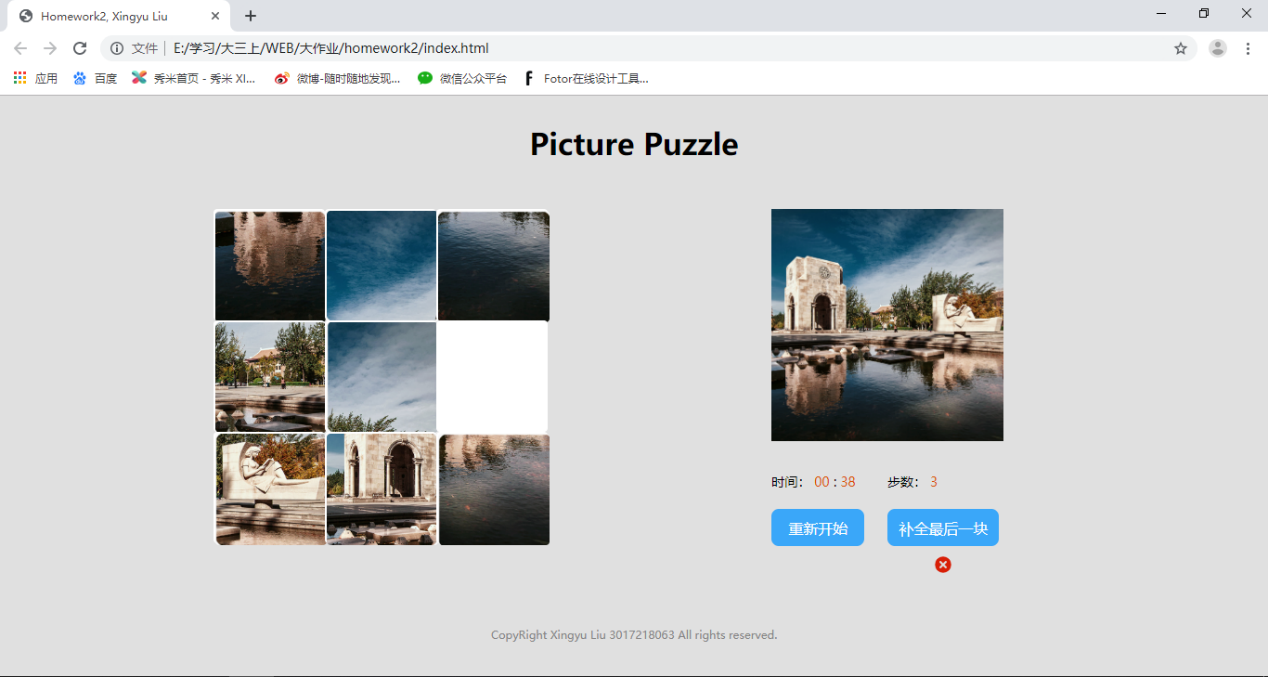
1. 界面效果



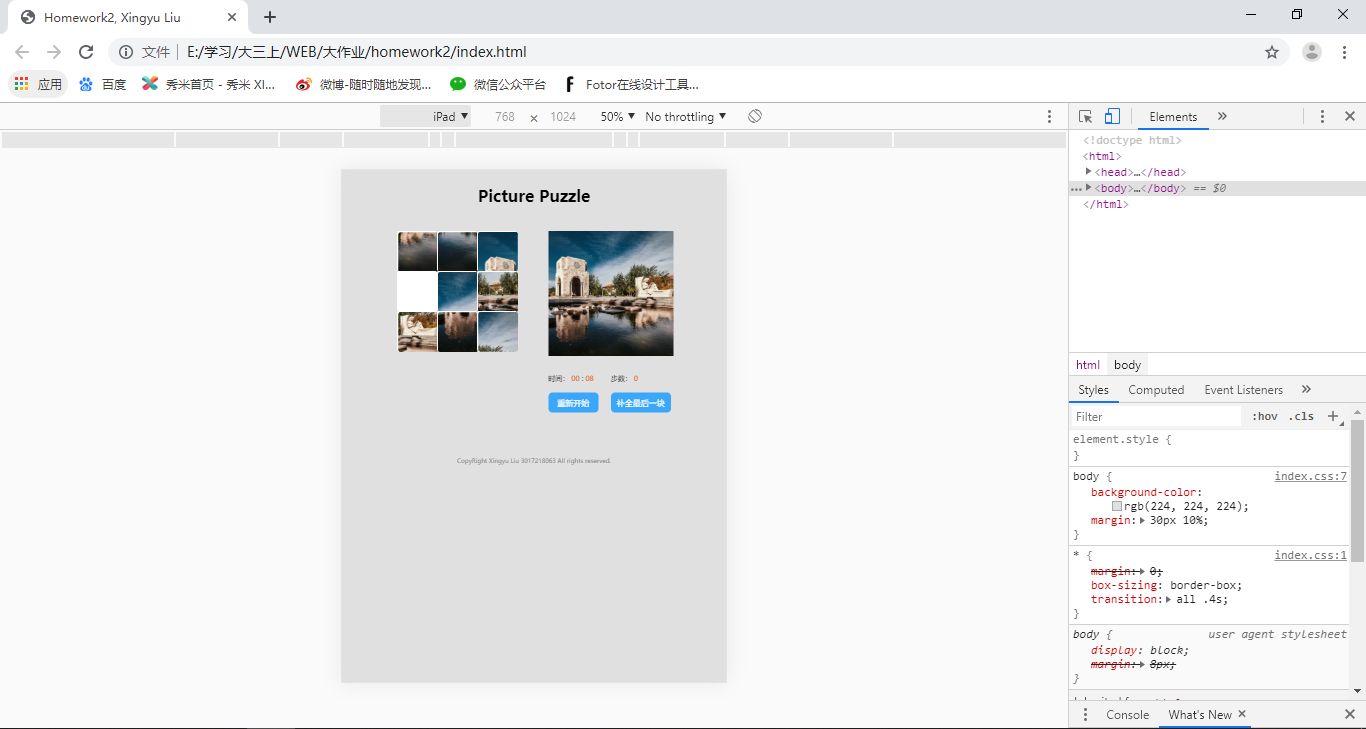
2. 获胜



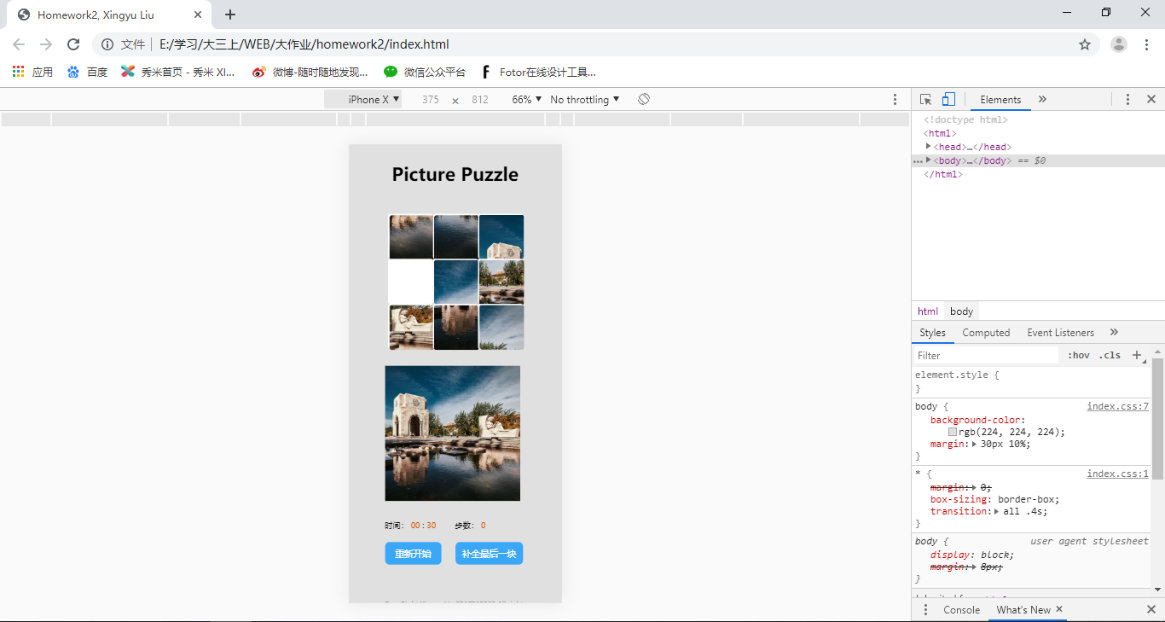
3. 不满足胜利条件



4. 平板电脑效果：



5. 手机效果



## 五、实验结论

熟练掌握了JQuery对DOM文档的操作

## 六、源代码

1. index.html

<!DOCTYPE html>

<html>

<head>

    <title>Homework2, Xingyu Liu</title>

    <meta charset=" tf-8">

    <meta name="viewport" content="width=device-width">

    <link rel="stylesheet" type="text/css" href="./index.css">

    <script type="text/javascript" src="./jquery-3.4.0.min.js"></script>

</head>

<body>

    <div class="title\_block">

        <h1>Picture Puzzle</h1>

    </div>

    <div class="outside\_block">

        <div class="game\_block">

            <div class="grid">

                <div id="p0"><img></div>

                <div id="p1"><img></div>

                <div id="p2"><img></div>

                <div id="p3"><img></div>

                <div id="p4"><img></div>

                <div id="p5"><img></div>

                <div id="p6"><img></div>

                <div id="p7"><img></div>

                <div id="p8"><img></div>

            </div>

        </div>

        <div class="menu\_block">

            <div class="menu\_content">

                <div class="origin\_img\_block">

                    <img id="origin\_img" src="./image/all.jpg">

                </div>

                <div class="score\_block">

                    <span id="time\_tip" class="tip">时间：</span>

                    <span id="minute" class="score time">00</span>

                    <span>:</span>

                    <span id="second" class="score time">00</span>

                    <span id="step\_tip" class="tip">步数：</span>

                    <span id="step" class="score">0</span>

                </div>

                <button id="reset">重新开始</button>

                <button id="win">补全最后一块</button>

                <div class="game\_over">

                    <img id="over\_img" src="">

                </div>

            </div>

        </div>

    </div>

    <div class="bottom\_block">

        <span class="message">CopyRight Xingyu Liu 3017218063 All rights reserved.</span>

    </div>

    <script type="text/javascript" src="./index.js"></script>

</body>

</html>

2. index.css

\* {

    margin: 0;

    box-sizing: border-box;

    transition: all .4s;

}

body {

    background-color: rgb(224, 224, 224);

    margin: 30px 10%;

}

h1 {

    text-align: center;

}

.title\_block {

    margin-bottom: 50px;

}

.outside\_block {

    width: 100%;

    min-height: 450px;

    display: flex;

    justify-content: space-around;

}

.game\_block,

.menu\_block {

    width: 50%;

}

.grid {

    display: flex;

    flex-wrap: wrap;

    width: 360px;

    margin: 0 auto;

}

.grid div {

    background-color: white;

    border-style: solid;

    border-width: 2px;

    border-color: white;

    border-radius: .25em;

    width: 120px;

    height: 120px;

}

.grid img {

    width: 120px;

    border-radius: .25em;

}

.grid img:hover {

    transform: scale(1.1);

}

.grid img:hover,

button:hover {

    cursor: pointer;

}

.menu\_content {

    margin: 0 auto;

}

#origin\_img,

.menu\_content {

    width: 250px;

}

.score\_block {

    margin-top: 30px;

    font-size: 14px;

}

.score {

    color: #db5000;

}

#second {

    margin-right: 30px;

}

#reset,

#win {

    border: none;

    background-color: #3aa7f9;

    color: white;

    border-radius: .5em;

    font-size: 16px;

    margin-top: 20px;

}

#reset {

    width: 100px;

    height: 40px;

    margin-right: 25px;

    float: left;

}

#win {

    width: 120px;

    height: 40px;

}

.game\_over {

    margin-top: 10px;

    padding-left: 175px;

}

#over\_img {

    width: 20px;

}

.bottom\_block {

    width: 100%;

    font-size: 12px;

    color: rgb(134, 134, 134);

    text-align: center;

}

button {

    outline: none;

}

img[src=""],

img:not([src]) {

    opacity: 0;

}

@media(max-width: 900px) {

    .grid {

        width: 240px;

    }

    .grid div,

    .grid img {

        height: 80px;

        width: 80px;

    }

}

@media(max-width: 700px) {

    .outside\_block {

        display: block;

    }

    .game\_block,

    .menu\_block {

        width: 100%;

    }

    .grid {

        width: 360px;

    }

    .grid div,

    .grid img {

        height: 120px;

        width: 120px;

    }

    .menu\_block {

        margin-top: 30px;

        margin-bottom: 30px;

    }

}

@media(max-width: 440px) {

    .outside\_block {

        display: block;

    }

    .game\_block,

    .menu\_block {

        width: 100%;

    }

    .grid,

    #origin\_img {

        width: 240px;

    }

    .grid div,

    .grid img {

        height: 80px;

        width: 80px;

    }

    .menu\_block {

        margin-top: 30px;

        margin-bottom: 30px;

    }

}

3. index.js

$(function() {

    var steps = 0;

    var minute = $("#minute").html();

    var second = $("#minute").html();

    var timer = setInterval(updateTime, 1000);

    var random = 0; //缺少的块

    var empty = 0; //缺少的位置

    var pictures = new Array(9); //存放九张图片的位置，为空时为0，位置为0-8

    resetPicture();

    //移动块操作

    $(".grid div").click(function() {

        let position = parseInt($(this).attr("id").substring(1));

        let canMove = moveCheck(position); //是否可以移动

        if (canMove.checked) {

            steps++; //计步

            $("#step").html(steps);

            //交换src，更新enpty和pictures

            let oldsrc = pictures.indexOf(position);

            let target = canMove.moveTo;

            pictures[oldsrc] = target;

            empty = position;

            $("#p" + position + " img").attr("src", "");

            $("#p" + target + " img").attr("src", "./image/" + oldsrc + ".jpg");

        }

    })

    //重新开始

    $("#reset").click(function() {

        //清空棋盘中图片src

        for (let i = 0; i < 9; i++) {

            let position = pictures[i].toString();

            $("#p" + position + " img").attr("src", "")

        }

        //打乱顺序

        resetPicture();

        //重置时间、计步、重启定时器

        $("#minute").html("00");

        $("#second").html("00");

        steps = 0;

        $("#step").html("0");

        $("#over\_img").attr("src", "");

        clearInterval(timer);

        timer = setInterval(updateTime, 1000);

    })

    //添加最后一张图片

    $("#win").click(function() {

        //检查是否满足胜利条件

        let check = true;

        for (let i = 0; i < 9; i++) {

            if (i === random) {

                continue;

            } else {

                if (pictures[i] !== i) {

                    check = false;

                    break;

                }

            }

        }

        //胜利，添加图片src，中断计时器，更新pictures（使得不能在移动），显示对勾

        if (check) {

            $("#p" + random + " img").attr("src", "./image/" + random + ".jpg")

            clearInterval(timer);

            $("#over\_img").attr("src", "./image/ok.svg");

            pictures[random] = random;

        } else {

            $("#over\_img").attr("src", "./image/no.svg");

        }

    })

    //检查是否能够移动，返回checked为是否能移动，moveTo为目标方块编号

    function moveCheck(position) {

        var result = {

            "checked": false,

            "moveTo": -1

        }

        if (position === empty) {

            return result;

        } else {

            let temp = [];

            let up = position - 3;

            if (up < 0) {

                up = -1;

            } else {

                temp.push(up);

            }

            let down = position + 3;

            if (down >= 9) {

                down = -1;

            } else {

                temp.push(down);

            }

            let left = position - 1;

            if (left === 2 || left === 5 || left === -1) {

                left = -1;

            } else {

                temp.push(left);

            }

            let right = position + 1;

            if (right === 9 || right === 6 || right === 3) {

                right = -1;

            } else {

                temp.push(right);

            }

            for (let i = 0; i < temp.length; i++) {

                if (pictures.indexOf(temp[i]) === -1) {

                    result.moveTo = temp[i];

                    result.checked = true;

                    return result;

                }

            }

            return result;

        }

    }

    //打乱

    function resetPicture() {

        //位置初始化为正确位置

        for (let i = 0; i < 9; i++) {

            pictures[i] = i;

        }

        //打乱数组

        shuffle(pictures);

        //随机选择一块不能被填入，初始位置为0

        random = parseInt((Math.random() \* 8));

        empty = pictures[random];

        pictures[random] = -1;

        //检查是否有解

        checkPossible(pictures);

        //设置图片src

        for (let i = 0; i < 9; i++) {

            if (i === random) {

                continue;

            }

            let number = (i).toString();

            let position = pictures[i].toString();

            $("#p" + position + " img").attr("src", "./image/" + number + ".jpg")

        }

        //随机交换法打乱数组

        function shuffle(pirctures) {

            for (let i = 0; i < 9; i++) {

                let currentRandom = parseInt(Math.random() \* 8);

                let current = pictures[i];

                pictures[i] = pictures[currentRandom];

                pictures[currentRandom] = current;

            }

        }

    }

    //使用 下标【图片编号】数组代表方块中存放的图片（-1为空），原数组和打乱数组逆序数奇偶相同则有解

    function checkPossible(pictures) {

        var origin = [0, 1, 2, 3, 4, 5, 6, 7, 8];

        origin[origin.indexOf(random)] = -1;

        var now = [-1, -1, -1, -1, -1, -1, -1, -1, -1];

        for (let i = 0; i < 9; i++) {

            if (i === random) {

                continue;

            }

            now[pictures[i]] = i;

        }

        //不相同，通过交换打乱数组倒数2、3位顺序，变为相同（注意-1时的特殊情况）

        if ((reverseCount(origin) % 2) !== (reverseCount(now) % 2)) {

            let temp = now[7];

            let temp2 = now[6];

            if (temp === -1) {

                pictures[temp2] = 7;

            } else if (temp2 === -1) {

                pictures[temp] = 6;

            } else {

                pictures[temp] = 6;

                pictures[temp2] = 7;

            }

        }

    }

    //求逆序数

    function reverseCount(arr) {

        var result = 0;

        for (let i = 0; i < arr.length; i++) {

            let temp = arr[i];

            for (let j = i + 1; j < 9; j++) {

                let compared = arr[j];

                if (compared < temp) {

                    result++;

                }

            }

        }

        return result;

    }

    //更新计时器时间，进位和补0

    function updateTime() {

        minute = parseInt($("#minute").html());

        second = parseInt($("#second").html());

        second++;

        if (second >= 60) {

            minute++;

            second -= 60;

        }

        if (second >= 0 && second < 10) {

            second = "0" + second.toString();

        } else if (second >= 10 && second < 60) {

            second = second.toString();

        } else {

            console.log("time error");

        }

        if (minute >= 0 && minute < 10) {

            minute = "0" + minute.toString();

        } else if (minute >= 10) {

            minute = minute.toString();

        } else {

            console.log("time error");

        }

        $("#minute").html(minute);

        $("#second").html(second);

    }

})