基于游戏交互的广告宣传系统设计与实现

摘要：这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。这是摘要。。。。。。。这是摘要。。。。。。。这是摘要。。。。。。。这是摘要。。。。。。。这

关键词：这是摘要。。。。。。。这是摘要。。。。。。。这是摘要。。。。。。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。这是摘要。。。。。。。

Based on the game design and implementation of interactive advertising system

Abstract**:** this this this this this this this this this this this this this this this this this this this this this this this this this this this

Keywords**:** this this this this this this this this this this this this this this this this this this this this this this this this this this

1 引言

1.1 研究背景

随着互联网的普及，广告宣传的方式日益增多。利用网站上的广告横幅、文本链接、多媒体的方法，在互联网刊登或发布广告，通过网络传递到互联网用户的一种高科技广告运作方式。与传统的四大传播媒体（报纸、杂志、电视、广播）广告及备受垂青的户外广告相比，互联网广告具有得天独厚的优势，是实施现代营销媒体战略的重要一部分。

互联网广告已经进入了崭新的时代，新推出来的H5动画，可以拥有和幻灯片展示差不多的效果，也可以用H5制作小动画。再加上对用户交互的控制，H5小游戏也应运而生。之前火爆一时的“围住神经猫”也就是利用H5技术开发的。

目前智能手机的普及，使手机端的广告推广更加便利。可能只需要一个链接，一个分享，或者一个二维码，就可以参与到公司的营销活动中去。伴随着简单好玩的游戏更能吸引用户的眼球。

1.2 系统概述

目前做游戏开发方面的主要以原生APP开发为主，其周期比较长，而且系统也比较复杂。随着H5技术的推广，利用前端技术在网页上也可以制作出简单易玩的小游戏了。

这个系统需要掌握的知识包含HTML、CSS、JS、JQ等。

2 技术基础

2.1 HTML

HTML又称为超级文本标记语言，是一种规范，它通过标记符号来标记要显示的网页中的各个部分。

超级文本标记语言文档制作不是很复杂，但功能强大，支持不同数据格式的文件嵌入，这也是万维网盛行的原因之一，其主要特点如下：

一、简易性：超级文本标记语言版本升级采用超集方式，从而更加灵活方便。

二、可扩展性：超级文本标记语言的广泛应用带来了加强功能，增加标识符等要求，超级文本标记语言采取子类元素的方式，为系统扩展带来保证。

三、平台无关性：虽然个人计算机大行其道，但使用MAC等其他机器的大有人在，超级文本标记语言可以使用在广泛的平台上，这也是万维网（WWW）盛行的另一个原因。

四、通用性：另外，HTML是网络的通用语言,一种简单、通用的全置标记语言。它允许网页制作人建立文本与图片相结合的复杂页面，这些页面可以被网上任何其他人浏览到，无论使用的是什么类型的电脑或浏览器。

超文本标记语言定义了多种数据类型的元素内容，如脚本数据和样式表的数据，和众多类型的属性值，包括ID、名称、URI、数字、长度单位、语言、媒体描述符、颜色、字符编码、日期和时间等。所有这些数据类型都是专业的字符数据。

网页的本质就是超级文本标记语言，通过结合使用其他的Web技术（如：脚本语言、公共网关接口、组件等），可以创造出功能强大的网页。因而，超级文本标记语言是万维网（Web）编程的基础，也就是说万维网是建立在超文本基础之上的。

2.2 CSS

层叠样式表(英文全称：Cascading Style Sheets)是一种用来表现HTML（标准通用标记语言的一个应用）或XML（标准通用标记语言的一个子集）等文件样式的计算机语言。CSS不仅可以静态地修饰网页，还可以配合各种脚本语言动态地对网页各元素进行格式化。

CSS 能够对网页中元素位置的排版进行像素级精确控制，支持几乎所有的字体字号样式，拥有对网页对象和模型样式编辑的能力。

CSS为HTML标记语言提供了一种样式描述，定义了其中元素的显示方式。CSS在Web设计领域是一个突破。利用它可以实现修改一个小的样式更新与之相关的所有页面元素。总体来说，CSS具有以下特点：

CSS提供了丰富的文档样式外观，以及设置文本和背景属性的能力；允许为任何元素创建边框，以及元素边框与其他元素间的距离，以及元素边框与元素内容间的距离；允许随意改变文本的大小写方式、修饰方式以及其他页面效果。

CSS可以将样式定义在HTML元素的style属性中，也可以将其定义在HTML文档的header部分，也可以将样式声明在一个专门的CSS文件中，以供HTML页面引用。总之，CSS样式表可以将所有的样式声明统一存放，进行统一管理。

CSS样式表可以单独存放在一个CSS文件中，这样我们就可以在多个页面中使用同一个CSS样式表。CSS样式表理论上不属于任何页面文件，在任何页面文件中都可以将其引用。这样就可以实现多个页面风格的统一。

在使用HTML定义页面效果的网站中，往往需要大量或重复的表格和font元素形成各种规格的文字样式，这样做的后果就是会产生大量的HTML标签，从而使页面文件的大小增加。而将样式的声明单独放到CSS样式表中，可以大大的减小页面的体积，这样在加载页面时使用的时间也会大大的减少。另外，CSS样式表的复用更大程序的缩减了页面的体积，减少下载的时间。

2.3 JavaScript

JavaScript一种直译式脚本语言，是一种动态类型、弱类型、基于原型的语言，内置支持类型。它的解释器被称为JavaScript引擎，为浏览器的一部分，广泛用于客户端的脚本语言，最早是在HTML（标准通用标记语言下的一个应用）网页上使用，用来给HTML网页增加动态功能。

JavaScript脚本语言具有以下特点:

(1)脚本语言。JavaScript是一种解释型的脚本语言,C、C++等语言先编译后执行,而JavaScript是在程序的运行过程中逐行进行解释。

(2)基于对象。JavaScript是一种基于对象的脚本语言,它不仅可以创建对象,也能使用现有的对象。

(3)简单。JavaScript语言中采用的是弱类型的变量类型,对使用的数据类型未做出严格的要求,是基于Java基本语句和控制的脚本语言,其设计简单紧凑。

(4)动态性。JavaScript是一种采用事件驱动的脚本语言,它不需要经过Web服务器就可以对用户的输入做出响应。在访问一个网页时,鼠标在网页中进行鼠标点击或上下移、窗口移动等操作JavaScript都可直接对这些事件给出相应的响应。

(5)跨平台性。JavaScript脚本语言不依赖于操作系统,仅需要浏览器的支持。因此一个JavaScript脚本在编写后可以带到任意机器上使用,前提上机器上的浏览器支 持JavaScript脚本语言,目前JavaScript已被大多数的浏览器所支持。

不同于服务器端脚本语言，例如PHP与ASP，JavaScript主要被作为客户端脚本语言在用户的浏览器上运行，不需要服务器的支持。所以在早期程序员比较青睐于JavaScript以减少对服务器的负担，而与此同时也带来另一个问题：安全性。

而随着服务器的强壮，虽然程序员更喜欢运行于服务端的脚本以保证安全，但JavaScript仍然以其跨平台、容易上手等优势大行其道。同时，有些特殊功能（如AJAX）必须依赖Javascript在客户端进行支持。

2.4 jQuery

jQuery是一个快速、简洁的JavaScript框架，是继Prototype之后又一个优秀的JavaScript代码库（或JavaScript框架）。jQuery设计的宗旨是“write Less，Do More”，即倡导写更少的代码，做更多的事情。它封装JavaScript常用的功能代码，提供一种简便的JavaScript设计模式，优化HTML文档操作、事件处理、动画设计和Ajax交互。

jQuery 是继prototype 之后又一个优秀的轻量级JavaScript 框架。其宗旨是———“Write Less, Do More”,写更少的代码,做更多的事情。 它是一个快速和简洁的JavaScript 库，可以简化HTML 文档元素的遍历，事件处理，动画和Ajax 交互以实现快速Web 开发，它被设计用来改变编写JavaScript 脚本的方式。

jQuery 的文档非常丰富，因为其轻量级的特性，文档并不复杂，随着新版本的发布，可以很快被翻译成多种语言，这也为jQuery 的流行提供了条件。jQuery 被包在语法上，jQuery 支持CSS1-3 的选择器， 兼容IE 6.0+, FF 2+,Safari 3.0+, Opera 9.0+, Chrome 等浏览器。同时，jQuery 有约几千种丰富多彩的插件，大量有趣的扩展和出色的社区支持，这弥补了jQuery功能较少的不足并为jQuery 提供了众多非常有用的功能扩展。加之其简单易学，jQuery 很快成为当今最为流行的JavaScript 库，成为开发网站等复杂度较低的Web 应用程序的首选JavaScript 库，并得到了大公司如微软，Google 的支持。

jQuery 最有特色的语法特点就是与CSS 语法相似的选择器，并且它支持CSS1 到CSS3 的几乎所有选择器，并兼容所有主流浏览器，这为快速访问DOM 提供了方便。

2.5 开发环境

项目的开发环境比较简单，只需要一个文本编辑器、浏览器、以及PS即可。  
编辑器使用的是Sublime Text，浏览器使用的是chrome模拟手机模式。

3 记忆翻牌

3.1 游戏介绍

这是一个H5翻牌游戏，首先通过关注微信公众号回复关键字“游戏”打开游戏的首页，然后点击参与游戏进入游戏页面。页面中一共有16张卡片，所有卡片都是显示背面的，其中每两张卡片的正面图片是一样的。通过点击卡片，让卡片进行反转，如果连续点击的两张卡片的正面是一样的，那么就隐藏消失；如果两张牌正面不一样，那么这两张牌就还原回去。在相应时间内如果所有卡片都正确翻转之后为通过游戏，后台会随机给用户发放红包；否则提示用户游戏闯关失败。

3.2 游戏设计

1、游戏首页

首页包括一个展示banner、一个参与游戏按钮以及一个游戏规则按钮。背景使用canvas做成流星效果，点击游戏规则弹出游戏规则的弹框，点击参与游戏进入游戏页面，如图3.1、3.2。

图3.1 游戏首页

图3.2 游戏规则

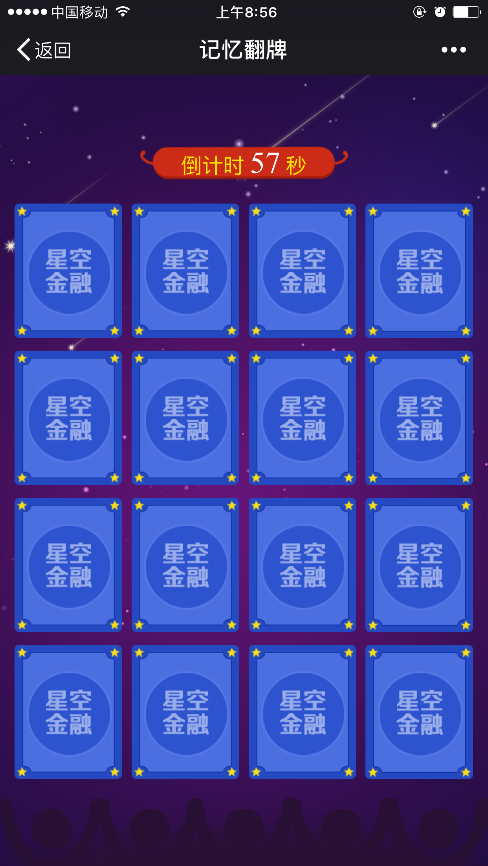
2、游戏页面

图3.3 游戏页面

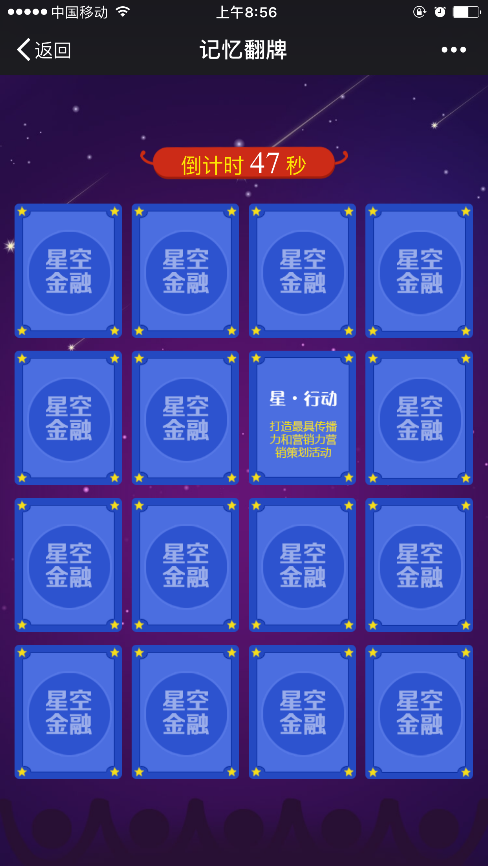
游戏页面包括16张小卡片以及倒计时提示框，小卡片可以点击，每次点击则会翻到正面，若两张卡片正面一样则消失，若不一样，则还原为背面，单次最多可以有两张卡片同时被翻为正面，如图3.3、3.4。

图3.4 卡片翻转

3、游戏结束

当所有卡片正确反转之后，弹出闯关成功对话框；当时间用完，弹出闯关失败，如图3.5、3.6。

图3.5 闯关成功



图3.7 闯关失败

3.3 代码实现

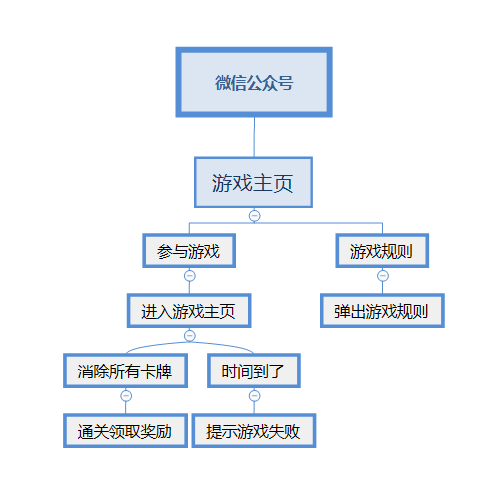
经过设计之后得出流程图3.8。

图3.8 流程图

3.3.1 页面HTML编写

在游戏开始制作之前我们需要先搭个页面的框架，如下：  
<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>记忆翻牌</title>  
 <meta name="viewport" content="width=device-width,initial-scale=1,maximum-scale=1,user-scalable=0">  
 <link rel="stylesheet" href="src/index.css">  
</head>  
<body>  
 <div class="content"></div>  
 <!-- 遮罩层 -->  
 <div class="shade"></div>  
 <!-- 游戏规则 -->  
 <div class="dialog dialog-guize"></div>  
 <!-- 成功 -->  
 <div class="dialog dialog-success">  
 <p></p>  
 <div class="btn-know"></div>  
 </div>  
 <!-- 失败 -->  
 <div class="dialog dialog-fail">  
 <div class="btn-know"></div>  
 </div>  
<script src="src/jquery.min.js"></script>  
<script src="src/modernizr.js"></script>  
<script src="src/index.js"></script>  
</body>  
</html>

其中meta标签定义了浏览器的宽度与设备的宽度一致，且不可手动缩放页面，保证了页面的正确展示。引入的样式文件index.css是页面的样式部分。类名为content的是游戏的内容部分，之后的页面展示，交互等都在这里进行操作。类名为shade的是页面遮罩层，颜色为黑色半透明。其余的以dialog开头的类名，分别为游戏规则、游戏成功、游戏失败的弹出框。底部分别引入需要使用的JS类库，分别为DOM操作库jquery.min.js,canvas检测脚本modernizr.js,以及业务的主要逻辑代码index.js。

3.3.2 页面逻辑部分搭建

在这里新建一个为Game的类，把游戏的步骤，方法等写在原型链prototype上面，函数需要的参数写在this上面，由指针调用，如表3.1。

表3.1 Game参数

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 功能 | 名称 | 功能 |
| content | 操作的DOM对象 | \_showElement | 已经翻转的卡片对象 |
| width | 页面的宽度 | \_showList | 翻转卡片标记值数组 |
| height | 页面的高度 | \_fail | 游戏失败（布尔值） |
| timer | 定时器对象 | \_success | 游戏成功（布尔值） |
| \_time | 倒计时时长 | \_score | 成功翻转的卡片数量 |
| cardCount | 显示卡片的个数 |  |  |
| cardList | 卡片的数组 |  |  |

上表中对需要用到的参数做了汇总，其中content为$('.content')，width和height取的是$(window)的宽和高，timer默认为null，\_time初始化为60秒，cardCount初始化为16，cardList、\_showElement、\_showList初始为空数组，\_fail，\_success初始化为fasle，\_score初始化为0。

游戏中需要用到的方法如表3.2。

表3.2 游戏方法

|  |  |  |  |
| --- | --- | --- | --- |
| 名称 | 功能 | 名称 | 功能 |
| shuffle | 洗牌方法 | time | Game 计数器 |
| drawStar | 绘制流星 | fail/success | Game 失败/成功 |
| step\_1 | Game 第一步 | play | Game 翻牌逻辑 |
| step\_2 | Game 第二步 | event | Game 事件监听 |

首先我们定义游戏方法Game 并初始参数

var Game = function(){  
 this.content = $('.content');  
 this.width = $(window).width()>540?540:$(window).width();  
 this.height = $(window).height();  
 this.timer = null;  
 this.\_time = 60;  
 this.cardCount = 16;  
 this.cardList = [];  
 this.\_showElement = [];  
 this.\_showList = [];  
 this.\_fail = false;  
 this.\_success = false;  
 this.\_score = 0 ;  
 this.step\_1();  
}

然后编写数组洗牌方法shuffle，这个方法主要在初始化卡片的时候对卡片的顺序进行随机排列，保证游戏的随机性。

function shuffle(arr){  
 var len = arr.length;  
 for(var i = 0; i < len - 1; i++){  
 var idx = Math.floor(Math.random() \* (len - i));  
 var temp = arr[idx];  
 arr[idx] = arr[len - i - 1];  
 arr[len - i -1] = temp;  
 }  
 return arr;  
}

为了使首页的展示更加好看些，加入了H5的新元素canvas，调用2d绘图方法模拟流星的动画。

在游戏初始化的时候，默认执行了step\_1方法，代码如下：

Game.prototype.step\_1 = function(){  
 var \_this = this;  
 \_this.content.css({  
 width:\_this.width,  
 height:\_this.height  
 })  
 \_this.event();  
 \_this.content.addClass('step-1');  
 var html ='';  
 html += '<img src="images/banner.png" alt="标题" class="banner">'  
 + '<img src="images/canyuyouxi.png" alt="参与游戏" class="btn btn-play">'  
 + '<canvas id="canvas" width="750" height="900"></canvas>'  
 + '<img src="images/youxiguize.png" alt="规则" class="btn btn-guize">';  
 \_this.content.html(html);  
 drawStar();  
}

上段代码，首先给content容器定义宽高为视窗的大小。然后执行event事件方法，该方法把所有事件都委托在body上面，我们放在最后面说。其次就是给content容器添加step-1的类名，表示现在执行到游戏的第一步。然后给content内容中添加一个标题，一个参与游戏的按钮，一个规则按钮，以及canvas背景动画。所有的样式都写在CSS 文件中，这里就不阐述了。主要是编写了页面的背景，元素的定位等。

点击参与游戏按钮，会执行step\_2方法，代码如下：

Game.prototype.step\_2 = function(){  
 var \_this = this;  
 \_this.content.removeClass('step-1');  
 \_this.content.addClass('step-2');  
 //变量初始化  
 // \_this.\_time = 60;  
 \_this.cardList = [];  
 var html='';  
 html += '<div class="time">倒计时 <span>'+\_this.\_time+'</span> 秒</div>';  
 html += '<div class="row" style="padding-left:1%;">'  
 for(var i =0;i<\_this.cardCount;i++){  
 html+= '<div class="card" style="width:'+this.width\*0.22+'px;height:'+this.width\*0.22\*1.25+'px;"></div>'  
 \_this.cardList.push(i%(\_this.cardCount/2)+1);  
 }  
 html += '</div>';  
 //卡片随机分配  
 shuffle(\_this.cardList);   
 \_this.content.html(html);  
 \_this.time();  
}

这段代码首先移除了content容器上的类名step-1，添加了step-2。容器内部添加了倒计时模块和卡片模块。顺便初始化了cardList，利用我们之前写好的洗牌方法对数组进行洗牌，然后将倒计时和卡片插入到content容器中。

这里有个翻牌的动画，卡片的两面分别使用了伪类before和after，before放置卡片的默认样式，after放置卡片对应的游戏图片

.card:before{  
 background: url(../images/i00.png) center no-repeat;  
 background-size: cover;  
 backface-visibility:hidden;  
 transition: all 500ms;  
}  
.card.hide{  
 opacity: 0;  
 visibility: hidden;  
}  
.card.show:before{  
 transform:translate3d(0,0,0) rotateY(180deg);  
}  
.card.show:after{  
 transform:translate3d(0,0,0) rotateY(360deg);  
}  
.card:after{  
 background-size: cover;  
 transform:translate3d(0,0,0) rotateY(180deg);  
 backface-visibility:hidden;  
 transition: all 500ms;  
}  
.card-1:after{  
 background-image: url(../images/i01.png);  
}

上面的样式主要描述了卡片的三种状态，默认样式，翻转样式和隐藏样式，利用transition来做动画变换，达到翻转和渐隐的效果，其中翻转的话backface-visibility定义了翻转的时候背面不可见。

在第一步的时候我们把event方法已经直接调用了，在这里来分析下event描述了哪些事件的委托。

Game.prototype.event = function(){  
 var \_this = this;  
 document.body.addEventListener('touchstart',function(ev){  
 ev.stopPropagation();  
 ev.preventDefault();  
 $el = $(ev.target);  
 //打开规则   
 if($el.hasClass('btn-guize')){  
 $('.shade').show();  
 $('.dialog-guize').slideDown();  
 return;  
 }  
 //隐藏规则  
 if($el.hasClass('dialog-guize')){  
 $('.shade').hide();  
 $('.dialog-guize').slideUp();  
 return;  
 }  
 //进入游戏  
 if($el.hasClass('btn-play')){  
 \_this.step\_2();  
 return;  
 }  
 //点击卡片  
 if($el.hasClass('card')){  
 \_this.play($el);  
 }  
 //我知道了  
 if($el.hasClass('btn-know')){  
 $('.shade').hide();  
 $('.dialog').hide();  
 }  
 });  
 document.body.addEventListener('touchmove',function(ev){  
 ev.stopPropagation();  
 ev.preventDefault();  
 });  
 document.body.addEventListener('touchend',function(ev){  
 ev.stopPropagation();  
 ev.preventDefault();  
 });  
}

我们对document.body 进行事件委托，监听body的touchstart、touchmove、touchend。并对body清除默认事件preventDefault和阻止冒泡stopPropagation。在这个游戏里面，并没有对touchmove和touchend的具体交互，所以主要操作在touchstart上。主要分为以下几个事件：

1. 点击规则按钮，弹出游戏规则弹框。
2. 点击游戏规则，弹框隐藏该弹框。
3. 点击参与游戏，进入step\_2()。
4. 点击卡片，执行play方法。
5. 点击弹框中的“我知道了”，关闭弹框。

方法中的ev.target 为事件的触发DOM元素，对齐进行class判断，可以得出点击的是哪个DOM元素。

Game.prototype.play = function(el){  
 var \_this = this;  
 if(\_this.\_fail) return;  
 if(el.hasClass('show')) return;  
 if(\_this.\_showList.length>=2) return;  
 var index = el.index();  
 el.addClass('show card-'+\_this.cardList[index]);  
 \_this.\_showList.push(\_this.cardList[index]);  
 \_this.\_showElement.push(el);  
 if(\_this.\_showList.length==2){  
 setTimeout(function(){  
 if(\_this.\_showList[0]==\_this.\_showList[1]){  
 \_this.\_showElement[0].removeClass('show').addClass('hide');  
 \_this.\_showElement[1].removeClass('show').addClass('hide');  
 \_this.\_score+=2;  
 }else{  
 \_this.\_showElement[0].removeClass('show');  
 \_this.\_showElement[1].removeClass('show');  
 }  
 \_this.\_showList=[];  
 \_this.\_showElement=[];  
 if(\_this.\_score==\_this.cardCount){  
 \_this.\_success=true;  
 \_this.success();  
 }  
 },500)  
 }  
}

以上代码为点击卡片的逻辑代码，主要逻辑为点击卡片给卡片添加类名“show”，根据当前的卡片索引index，添加对应的类名“card-”加索引，以显示卡片。当卡片显示的数量为二时，对比两个卡片索引对应的数组值是否一样，如果一样给卡片添加”hide”类名，使卡片隐藏，且给\_score加上两分。如果两张卡片不一样，则移除类名“show”使卡片翻转回去。如果卡片翻开的数量大于两张，则会退出改方法。

4 拼字游戏

3.1 游戏介绍

项目的开发环境比较简单，只需要一个文本编辑器、浏览器、以及PS即可。  
编辑器使用的是Sublime Text，浏览器使用的是chrome模拟手机模式。

3.2 游戏设计

项目的开发环境比较简单，只需要一个文本编辑器、浏览器、以及PS即可。  
编辑器使用的是Sublime Text，浏览器使用的是chrome模拟手机模式。

3.3 代码实现

项目的开发环境比较简单，只需要一个文本编辑器、浏览器、以及PS即可。  
编辑器使用的是Sublime Text，浏览器使用的是chrome模拟手机模式。