2048小游戏

院系：数学科学学院

专业：信息与计算科学

班级：信息二班

小组成员：季博文 王梓明 孔令辉 赵化龙 苏蓉春

团队分工：

|  |  |
| --- | --- |
| 产品经理 | 季博文 |
| 项目经理 | 王梓明 |
| 设计者 | 赵化龙 |
| 开发者 | 季博文 孔令辉 王梓明 苏蓉春 |
| 测试者 | 赵化龙 季博文 |

摘要

在生活节奏不断加快的现代社会，轻松的益智游戏对缓解生活的压力，调节情绪具有重要的意义。《2048》是一款比较流行的数字游戏。游戏的设计初衷是一款益智类游戏，其特点在于轻松，简单，因此开发要求做到各项功能完备，操作简单，易学易用。我们团队通过javascrip编写命令，通过html和css等编写网页，通过windows控制台实现2048的基本运行。 我们设计的游戏，拥有计时功能，同是可以看到最高分，总分。本着简单，易于操作和缓解生活压力，降低生活节奏的初衷,我们在游戏界面中指明了控制方法，让游戏的用户界面更加人性化。

一、学习调研

1.背景知识…………………………………………………4

2.开发环境…………………………………………………5

3.建库………………………………………………………5

二、需求分析

1.可行性分析………………………………………………5

2.基础需求…………………………………………………6

3.特殊需求…………………………………………………6

4.技术选型…………………………………………………7

三、定理规范

1.管理规范…………………………………………………7

2.流程规范…………………………………………………7

3.开发规范…………………………………………………7

四、概要设计

设计图 ……………………………………………………10

五、详细设计

1.游戏主页面 ……………………………………………11

2该项目的主要结构………………………………………11

3.游戏结束页面 …………………………………………11

4.简单的流程图 …………………………………………12

5.相关代码设计 …………………………………………13

六、测试及测试报告

1.测试过程 ………………………………………………15

2.测试用例 ………………………………………………15

3.测试报告 ………………………………………………17

七、软件实现

可运行代码 ………………………………………………21

八、产品发布

1.反馈 ……………………………………………………27

2.产品迭代/演进…………………………………………27

一.学习调研

**1.背景知识：**

2014年Gabriele Cirulli利用周末的时间写2048这个游戏的程序，仅仅只是好玩而已。他想用一种不同的视觉展现效果和更快速的动画来创造属于我自己的游戏版本，AsherVollmer和GregWohlwend作为Threes的开发者，他们在开发这款游戏的时候投入了大量的时间和精力。在制作2048的过程中，他决定把项目托管到Github上去，并且通过Github完成项目。完成了这个游戏之后，通过GithubPages发布，并发表在DesignerNews上。

（1）开发工具：

HTML超文本标记语言（英文：HyperText Markup Language）

是为“网页创建和其它可在网页浏览器中看到的信息”设计的一种置标语言。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。由蒂姆•伯纳斯-李给出原始定义，由IETF用简化的SGML（标准通用置标语言）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。

Javascript是一种脚本语言，是不需要进行编译的，也是浏览器中的一部分，经常用在web客户端脚本语言，js主要是用web的开发，可以给网站添加各种各样的动态效果，让网页更加美观。

2**开发环境**

开发环境：JavaScript

应用软件**：**Hbuilder

应用环境：火狐浏览器

**3、建库**：Github

地址：https://github.com/Ji-BoWen/task.git

二.需求分析

1 .**可行性分析**

（1）技术可行：根据游戏设置的要求，可以通过js进行源代码的编辑，通过windows界面进行用户界面的编辑和优化，结合时间和目前学习水平等各项因素，项目所要求功能和性能完全可以实现。

（2）运行可行性：游戏基本要求是.net framework4.0及以上在大部分的用户设备上可以实现，并且游戏运行对硬件几乎无要求，项目运行可以得到基本保证。

2.**基础需求**：

（1）游戏操作需求：

可使用↑(W)/↓(S)/←(A)/→(D)进行操作；

（2）2048游戏算法：基本的算法是，如果2^k是最大的数字，那么我们努力将2^(k-1)放在它的旁边，然后再把2^(k-2)放在2^(k-1)的旁边，以此类推。

为了达成这一点，我们努力将最大的数字放在角落，然后将第二大的数字放在它旁边，以此类推。将最大的数字放在角落，这样就可以留出足够的空间来组合出更大的数字。

最终我们期望形成这样的图形：

x x x x

4 2 x x

8 16 32 64

1024 512 256 128

（3）特殊需求：可记算时间，增添用户的体验感

（4）技术选型：

CSS框架：简单地说，就是一些事先写好的css，你只需要给你的html元素加上一些特定的类，就可以快速的得到一些想要的效果。

三、定理规范

**1.管理规范**

组长制作项目目录以及各项要求，组员自主领取分工任务，除此之外由组长负责。

**2.流程规范**

组员完成的工作通过文件或GitHub汇总给组长，组长确认后交还，各工作成果由本人上传GitHub，不能上传也可组内帮忙上传。

最后由组长整理完整的2048项目。

**3.开发规范**

（1）变量名推荐使用驼峰法来命名(camelCase):

（2）空格与运算符通常运算符 ( = + - \* / ) 前后需要添加空格:

（3）复杂语句的通用规则:

•将左花括号放在第一行的结尾。

•左花括号前添加一空格。

•将右花括号独立放在一行。

•不要以分号结束一个复杂的声明。

（4）对象定义的规则:

•将左花括号与类名放在同一行。

•冒号与属性值间有个空格。

•字符串使用双引号，数字不需要。

•最后一个属性-值对后面不要添加逗号。

•将右花括号独立放在一行，并以分号作为结束符号。

（5）命名规则

•变量和函数为小驼峰法标识, 即除第一个单词之外，其他单词首字母大写（ lowerCamelCase）

•全局变量为大写 (UPPERCASE )

•常量 (如 PI) 为大写 (UPPERCASE )

（6） 后缀

•HTML 文件后缀可以是 .html (或 .htm)

•CSS 文件后缀是 .css 。

•JavaScript 文件后缀是 .js 。、

（7）属性值

•HTML5 属性值可以不用引号。

•属性值我们推荐使用引号:

•如果属性值含有空格需要使用引号。

•混合风格不推荐的，建议统一风格。

•属性值使用引号易于阅读。

（8）HTML 注释

•注释可以写在 <!-- 和 --> 中:

（9）样式表

样式表使用简洁的语法格式 ( type 属性不是必须的):

<link rel="stylesheet" href="styles.css">

在 HTML 中载入 JavaScript

使简洁的语法来载入外部的脚本文件 ( type 属性不是必须的 ):

<script src="myscript.js">

四、概要设计

**箭头显示**

**键盘移动**

**时间**

**最大数**

**总分**

**注意了哦**

**填满16格**

**退出界面**

**确定**

**取消**

**成绩**

**游戏结束**

**游戏启动**

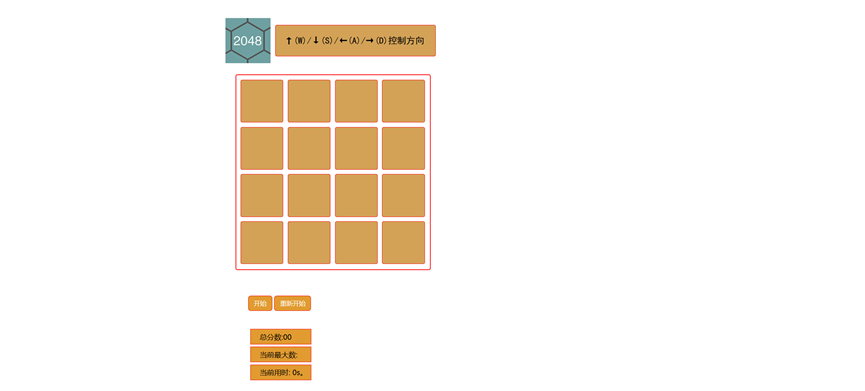
**开始**

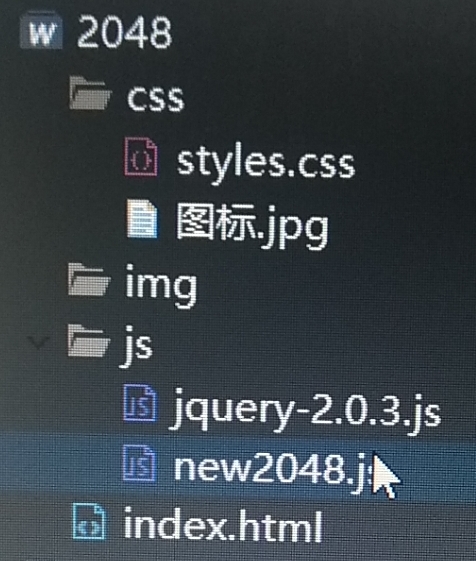
**游戏运行**

五、详细设计

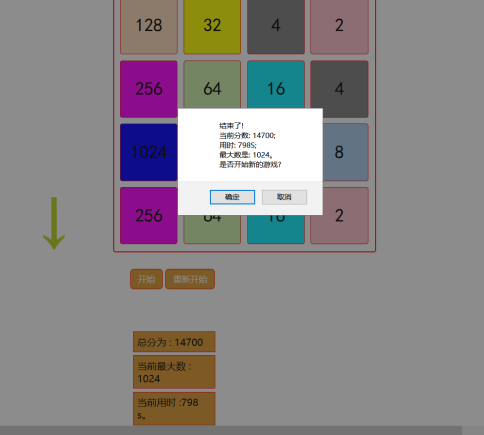
**1.游戏主页面**

截图如下:

**2，该项目的主要结构如下：**



**3.游戏结束页面**



**4.简单的流程图**

是

移动获得分数

两个数相加，暂时加和

并保存最大的数

是

游戏结束

否

注意了哦

否

是

是否有空格？

同一方向上是否相加

↑(W)/↓(S)/←(A)/→(D)控制方向移动

4\*4方格随机产生两个数字

游戏开始

**5.相关代码设计：**

HTML通过创建盒子模型存放相关文本

CSS对于创建的标签赋予属性，如内边距 外边距 边框 背景 相关的文本属性 浮动以及相对定位等

JS对要实现的功能进行设计：

（1）初始化好定义的变量 keys colors score max time t

（2）用window.onkeydown = function(e) 来获得键盘按下的事件

（3）用function isEnd()判定结束时的成绩

function isEnd() {

var f = false;

if (locations.indexOf(0) == -1) {

$("#danger").text("注意了哦~");

{if (isEndX() && isEndY()) {

clearTimeout(t);

if (window.confirm("结束了!\n当前分数: " + score + ";\n用时: " + time

+ "S;\n最大数是: " + max + "。\n是否开始新的游戏?")) {

init();

} else {window.close();}

}

} else {$("#danger").text("");

}return f;}

（4）function isEndX() / function isEndY()判断横向/纵向的数组，如果相邻位置的数不相同,就结束

unction isEndX() {

var f = false;

var w = new Array();

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

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

w[i] = locations[4 \* j + i];}

f = (w[0] != w[1] && w[1] != w[2] && w[2] != w[3]);

if (!f) {

break;

}

}

return f;

}

（5）创建向四个方向移动相加的方法

Eg：（向下）

function toDown() {

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

var row = [ locations[i + 12], locations[i + 8], locations[i + 4],

locations[i] ];

configurationD(i, row); }

locations[createLocation()] = createFixedNum();

paint();}

（6）用function paint()来计算分数和最大数

function paint() {

for (var i = 0; i < 16; i++) {

$("#box" + keys[i]).text((locations[i] == 0) ? "" : locations[i]);

var index = (locations[i] == 0) ? 0

: (locations[i].toString(2).length - 1);

$("#box" + keys[i]).css("background", colors[index]);

if (locations[i] > max) {

max = locations[i];}

}

$("#score").text("总分为 : " + score);

$("#max").text("当前最大数 : " + max);

}

（7）在空位置中随机生成2和4 ，生成2的概率为0.8。

在空位置中随机生成，如果该位置有值,就返回重新生成

function createFixedNum() {

return Math.random() < 0.8 ? 2 : 4;

}

function createLocation() {

var num = Math.floor(Math.random() \* 16);

while (locations[num] != 0) {

num = Math.floor(Math.random() \* 16);

}

return num;

}

（8）用function showtime()来记时

function showtime() {

$("#time").text("当前用时 :" + (++time) + " s。");

}

六、测试及测试报告

**1.测试过程**

（1）拟定测试计划，根据游戏的具体情况，仔细分析游戏各个步骤运行可能出现的状况，对各种情况进行汇总，然后拟定出一份具体测试步骤。

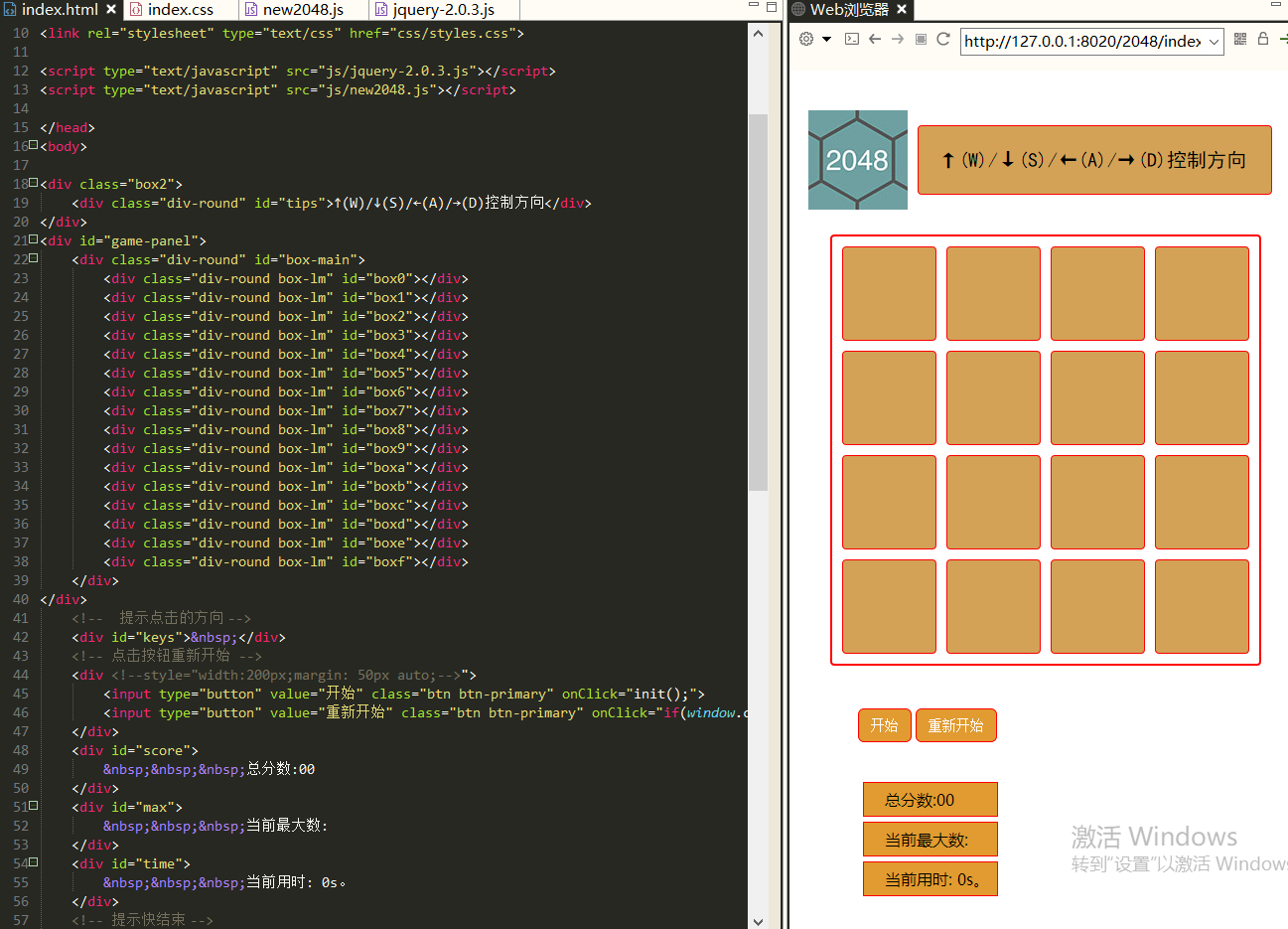
（2）初步测试，根据拟定好的测试计划对游戏进行调试，出现意外情况时应该及时记录，测试完后对出现的意外情况进行分析，然后提出改进方案再进行修改。

（3）深度测试，对游戏进行最后的修改，确认没有漏洞后再进行调试，从各个方面进行一次整体排查，直到不再出现意外情况为止。

**2.测试用例**

（1）初始界面

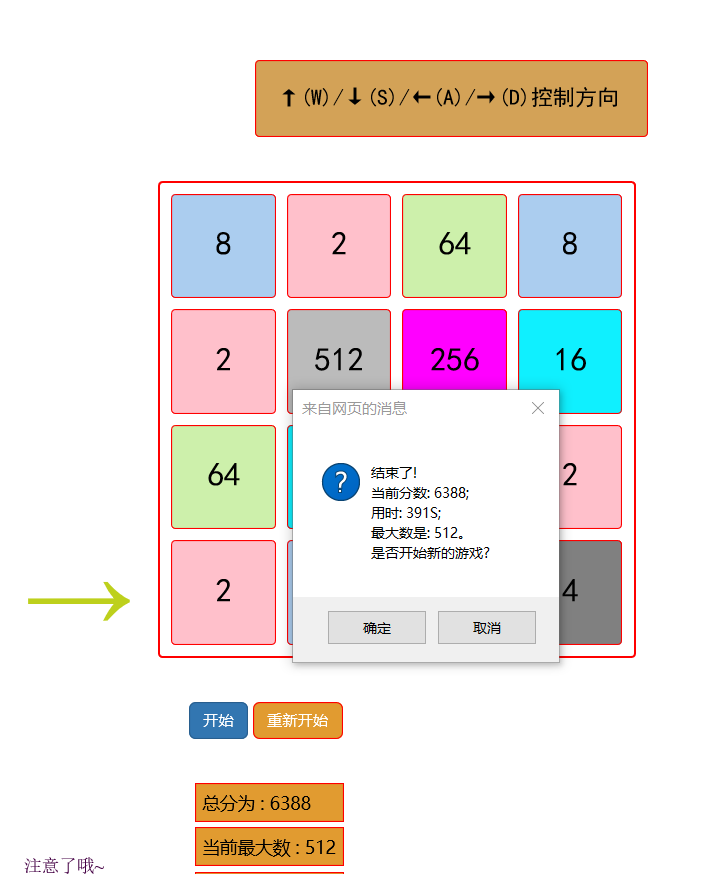
我们做的游戏相对简便，打开游戏后，会弹出初始界面，初始界面的内容有游戏的操作方法，开始按钮，重新开始按钮，总分等。



**（2）正式开始测试**

每次可以选择上下左右其中一个方向去滑动，每滑动一次，所有的数字方块都会向滑动的方向靠拢，系统也会在空白的地方出现一个数字方块，相同的数字方块在靠拢，相撞时会相加。不断叠加最终拼凑出2048这个数字就算成功。但游戏不会就此结束，会继续直到无路可走，当无路可走时，游戏会提示结束，并显示最高分，总分，总用时。

**（3）例子如下：**



**3.测试报告**

测试方法在主要是黑盒测试，由测试人员通过自己亲自去玩，然后观察游戏在运行过程中，所需要实现的功能有没有实现，以及在测试过程中有没有发现新的问题记录在案。然后根据测试后所产生的一系列的问题，有本组负责编码人员对游戏进行改进，然后在进行测试，就这样重复的进行这样的工作，直到游戏的运行与预想时的状况基本一致时，才算通过测试。

**测试1**

名称:测试在计算机不同浏览器中能否正常打开和运行

目的:使游戏能够在不同的浏览器中正常运行

测试结果:经过游戏在不同的浏览器中运行发现，游戏能够在浏览器中正常运行，并且操作流畅。

**测试2**

名称:数字出现的位置是否是随机的

目的:使游戏打开后能够正常运行，数字能成功的随机出现。

测试结果:经过测试打开游戏后能够正常运行，数字能成功的随机出现。

**测试3**

名称:相同数字的背景颜色是否相同

目的:使游戏游戏界面美观，易于辨认数字，方便玩家使用 测试结果:经过测试相同数字的背景颜色相同。

**测试4**

名称:数字移动是否能跟随按键控制方向而移动到相应的方向

目的:使数字能够跟随按键控制方向而向相应的方向移动

测试结果:经过测试数字能够跟随按键控制方向而向相应的方向移动。

**测试5**

名称:当出现相同数字能不能相加且合并为一个数字，如2,2合并为4.

目的:使程序能够正常运行

测试结果:经过测试当出现相同数字能相加且合并为一个数字

**测试6**

名称:随着数字合并相加，分数框的数字是否也是叠加向上的。

目的:当数字合并次数增加，分数就会增加

测试结果:通过测试当数字合并次数增加，分数框的数字叠加向上。

**测试7**

名称:当游戏界面中出现2048，游戏是否成功并出现游戏结束选择界面。

目的:游戏能否成功

测试结果:当游戏界面中出现2048，游戏成功并出现游戏结束选择界面。

**测试8**

名称:在游戏结束的界面的时候，是否可以重新开始游戏。

目的:测试能否从新开始游戏

测试结果:通过测试当游戏结束时，玩家通过按界面的按钮“确定”可以重新开始游戏。

**测试9**

名称**:**在游戏16个格中都有数字时，是否下方提示“要注意了哦”。

目的：增添用户体验的紧张感

测试结果：通过测试当16个格中都有数字时，下方提示“要注意了哦”。

**测试10**

名称：在游戏结束时，是否弹出成绩的相关数据。

**目的：**给用户更直观的游戏数据，更好的游戏体验。

测试结果：在游戏结束时，在网页中间弹出成绩的相关数据。

**测试11**

名称：数字移动时，是否出现带有数字移动方向的箭头

目的：增添用户体验

测试结果：在游戏结束时，数字移动时，在4\*4方格左下角出现带有数字移动方向的箭头

**测试12**

名称：用户能否点击“联系我们”给予反馈。

目的：更好的与用户沟通，提升版本更新

测试结果：界面下方点击黑色字体“联系我们”即可进入到季博文的码云，可以给予反馈，访问过链接同时变成蓝色

七、软件实现

HTML代码

<!DOCTYPE html>

<html>

<head>

<meta http-equiv="Content-Type" content="text/html; charset=utf-8" />

<title>2048小游戏</title>

<style type="text/css">

\*{margin:0;padding:0;list-style-type:none;}

</style>

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

<script type="text/javascript" src="js/jquery-2.0.3.js"></script>

<script type="text/javascript" src="js/new2048.js"></script>

</head>

<body>

<div class="box2">

<div class="div-round" id="tips">↑(W)/↓(S)/←(A)/→(D)控制方向</div>

</div>

<div id="game-panel">

<div class="div-round" id="box-main">

<div class="div-round box-lm" id="box0"></div>

<div class="div-round box-lm" id="box1"></div>

<div class="div-round box-lm" id="box2"></div>

<div class="div-round box-lm" id="box3"></div>

<div class="div-round box-lm" id="box4"></div>

<div class="div-round box-lm" id="box5"></div>

<div class="div-round box-lm" id="box6"></div>

<div class="div-round box-lm" id="box7"></div>

<div class="div-round box-lm" id="box8"></div>

<div class="div-round box-lm" id="box9"></div>

<div class="div-round box-lm" id="boxa"></div>

<div class="div-round box-lm" id="boxb"></div>

<div class="div-round box-lm" id="boxc"></div>

<div class="div-round box-lm" id="boxd"></div>

<div class="div-round box-lm" id="boxe"></div>

<div class="div-round box-lm" id="boxf"></div>

</div>

</div>

<!-- 提示点击的方向 -->

<div id="keys">&nbsp;</div>

<!-- 点击按钮重新开始 -->

<div <!--style="width:200px;margin: 50px auto;-->">

<input type="button" value="开始" class="btn btn-primary" onClick="init();">

<input type="button" value="重新开始" class="btn btn-primary" onClick="if(window.confirm('您确定重新开始吗?')){init();}">

</div>

<div id="score">

&nbsp;&nbsp;&nbsp;总分数:00

</div>

<div id="max">

&nbsp;&nbsp;&nbsp;当前最大数:

</div>

<div id="time">

&nbsp;&nbsp;&nbsp;当前用时: 0s。

</div>

<!-- 提示快结束 -->

<div id="danger">

<div style="text-align:center;margin:50px 0; font:normal 14px/24px 'MicroSoft YaHei';">

<p><b><a class="one" href="https://gitee.com/ji\_bowen" target="\_blank">联系我们</a></b></p>

</div>

</body>

</html>

CSS代码

.div-round {

display: inline-block;

/\*margin-bottom: 0;\*/

font-size: 20px;

font-weight: normal;

line-height: 1.428571429;

font-family: "黑体";

text-align: left;

white-space: nowrap;

vertical-align: middle;

background-image: none;

border: 1px solid transparent;

border-radius: 4px;

margin:auto 0;

}

#box-main {

width: 420px;

height: 420px;

padding: 5px;

border: 2px solid red;

margin: 0 auto;

position: relative;

top: 40px;

right: 28px;

}

#box0,#box1,#box2,#box3,#box4,#box5,#box6,#box7,#box8,#box9,#boxa,#boxb,#boxc,#boxd,#boxe,#boxf

{

height: 68px;

width: 93px;

border: 1px solid red;

float: left;

margin: 5px;

padding-top: 25px;

font-size: 30px;

text-align: center;

background: #d3a257;

font-size: 30px;

}

#game-panel,#score,#max,#time{

width: 630px;

position: relative;

left: 550px;

top: 25px;

}

#body{

background-color: darkmagenta;

width: 100%;

height: 100%;

background-image: url(../img/背景0.jpg);

}

.box2{

width: 100px;

height: 100px;

/\*background-color: darkgrey;\*/

position: relative;

left: 500px;

top: 40px;

background-image:url(图标.jpg)

/\*background-repeat:repeat-x;\*/

}

/\*.box1{

width: 100px;

height: 100px;

background-color: fuchsia;

position: relative;

left: 200px;

top:10px;}\*/

#score,#max,#time{

width: 122px;

border: 1px solid red;

font-family: "微软雅黑";

padding: 6px;

margin: 5px;

background-color:#e19b30;

/\*position: relative;

right: 30px;\*/

/\*background-image: url(../img/背景0.jpg);\*/

}

#tips,#danger {

width: 250px;

/\*margin: 20px auto;\*/

}

#tips{

width:315px;

/\*float:left;\*/

padding:20px;

border:1px solid red;

background: #d3a257;

position: relative;

left: 110px;

top: 15px;}

#keys {

font-size: 100px;

color: #bdd01c;

position: relative;

bottom: 50px;

left: 400px;

/\*background-color: #285E8E;\*/

}

#danger {

color: #440044;

position: relative;

bottom: 50px;

left: 400px;

}

.btn {

display: inline-block;

padding: 6px 12px;

margin-bottom: 0;

font-size: 14px;

font-weight: normal;

line-height: 1.428571429;

text-align: center;

white-space: nowrap;

vertical-align: middle;

cursor: pointer;

background-image: none;

border: 1px solid transparent;

border-radius: 6px;

}

.btn:hover,.btn:focus {

color: #333333;

text-decoration: none;

}

.btn-primary {

color: #ffffff;

background-color: #e19b30;

border-color: red;

z-index: 99;

position: relative;

bottom: 10px;

left:550px ;}

.btn-primary:hover,.btn-primary:focus,.btn-primary:active,.btn-primary.active,.open .dropdown-toggle.btn-primary

{

color: #ffffff;

background-color: #3276b1;

border-color: #285e8e;

}

.btn-primary:active,.btn-primary.active,.open .dropdown-toggle.btn-primary

{

background-image: none;

}

a.one:link {color:black;}

a.one:visited {color:blue;}

a.one:hover {color:red;}

a.one{

position: relative;

left: 210px;

top: 65px;

border: 1px solid red;

font-family: "微软雅黑";

padding: 6px 20px;

background-color:#e19b30;

}

**js可运行代码**：详见GitHub

仓库地址：https://github.com/Ji-BoWen/task.git

八、产品发布

**1.用户反馈**

可直接点击下方联系我们的链接即可进入我们的码云，给与反馈

**2.产品迭代/演进**

下一版本将会更新双人2048竞赛模式