# 1、系统概述

**1.1、系统简介**

开发一个具有扫雷游戏功能的网站，玩家可在电脑的浏览器上打开并使用，具有难度调节（行列数）和统计信息功能

扫雷规则：

1.左键点击显示当前格子是否为雷，如果为雷的话，GameOver，如果不是雷的话，这个格子会显示周围八个格子内的雷数量。

2.鼠标右键标记，标记可能的雷，标记了之后取消需要再次右键点击该格子，左键无效果。

3.鼠标中键（滚轮）按下后，快捷扫雷（如果周围雷数和未被标记且未被翻开的格子相等，会将这些格子一并翻开）

主要功能基本完全复刻了windows7扫雷的功能

**1.2、术语表**

无

**1.3、系统运行环境**

操作系统：Windows系统

编程平台：visual studio code/记事本

**1.4、开发环境**

编程软件：visual studio code 2017

开发语言：HTML，CSS，JavaScript

试运行软件：google chrome浏览器最新版

# 数据结构说明

在HTML的JS中直接使用的变量数据结构：

width 整数 扫雷表格宽度

high 整数 扫雷表格长度

num 整数 雷数

val 整数 难度 1-初级，2-中级，3-高级，4-自定义

arr\_num 数组 设置的难度对应雷数 [0,10 40,120]

arr\_type 数组 设置的难度对应行列数 [0,10 16,24]

arr\_details 数组 存储的统计信息

len 整数 信息记录长度

扫雷函数程序构造函数使用的数据结构：

this.num1 = num1; 整数//行数

this.num2 = num2; 整数//列数

this.mine\_num = mine\_num; 整数//雷的个数

this.tiles = new Array(); 数组//数组里面存放的是每个小格子

this.obj = obj; 对象//扫雷放置的对象

this.flag = true; 布尔//判断是否为第一次点击

this.arr = new Array(); 数组//格子位置-行

this.arr\_2 = new Array(); 数组//格子位置-列

this.time\_dsq = null; 整数//所用时间-秒

this.time\_dc = ' '; 整数//时间显示-秒

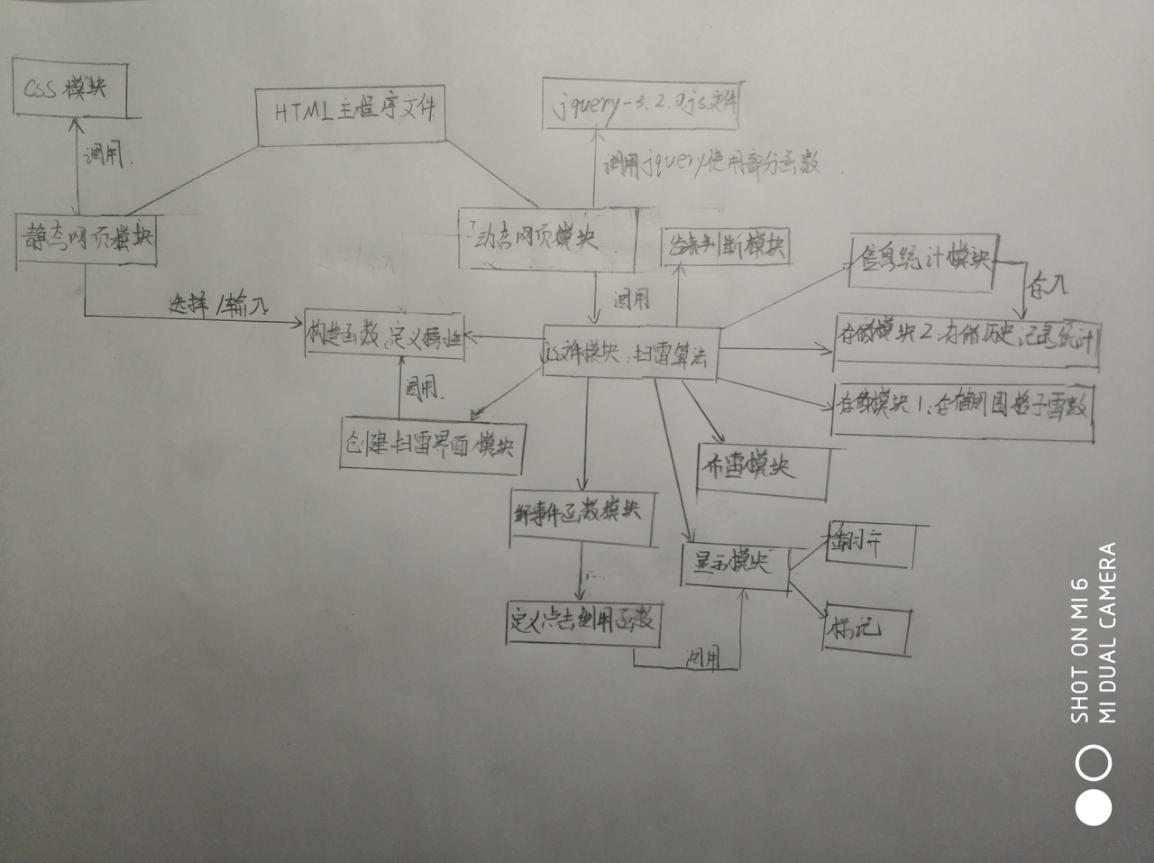
this.arr\_time =new Array(); 数组//时间统计信息

this.details =new Array(); 数组//游戏统计详情

this.type = type; 整数//游戏类型：初级/中级/高级/自定义--1/2/3/4

# 3、模块设计

**3.1、软件结构**



****

**3.2、功能设计说明**

以HTML为主文件调用CSS文件，JQUERY函数文件和JS扫雷算法程序文件，将扫雷游戏的各项功能分解并用函数实现。

先使用HTML和CSS搭建一个简单的静态网页，然后用JS实现创建扫雷的游戏界面和信息统计，然后调用扫雷的函数进行游戏。

**3.3、模块1：JS静态与动态网页构造模块**

3.3.1、设计图

无

3.3.2、功能描述

输入数据，调用扫雷游戏函数，创建扫雷游戏表格，进行游戏，存取与显示时间和数据记录

3.3.3、输入数据

width 整数 扫雷表格宽度

high 整数 扫雷表格长度

num 整数 雷数

val 整数 难度 1-初级，2-中级，3-高级，4-自定义

3.3.4、输出数据

扫雷游戏界面

arr\_details 数组 已玩游戏，已胜游戏，胜率，最多连胜，最多连败，当前连局；

arr\_grade 数组 所用时间

3.3.5、数据设计

arr\_details 数组 已玩游戏，已胜游戏，胜率，最多连胜，最多连败，当前连局；

arr\_grade 数组 所用时间

width 整数 扫雷表格宽度

high 整数 扫雷表格长度

num 整数 雷数

val 整数 难度 1-初级，2-中级，3-高级，4-自定义

3.3.6、算法和流程

根据输入的数据调用扫雷程序构造函数，通过JS实现网页的动态点击显示

3.3.7、函数说明

调用构造函数模块

3.3.8 全局数据结构与该模块的关系

顶层数据输入模块，向构造函数模块中输入数据并输出最终数据

**3.4、模块2：顶层构造函数模块MineCraft**

3.4.1、设计图

无

3.4.2、功能描述

创建构造函数，定义创建扫雷表格

3.4.3、输入数据

num1;//行数

num2;//列数

mine\_num;//雷的个数

obj;//扫雷放置的对象

type;//游戏类型：初级/中级/高级/自定义

从HTLM的主JS程序中输入调用

3.4.4、输出数据

扫雷游戏界面

3.4.5、数据设计

this.num1 = num1; 整数//行数

this.num2 = num2; 整数//列数

this.mine\_num = mine\_num; 整数//雷的个数

this.tiles = new Array(); 数组//数组里面存放的是每个小格子

this.obj = obj; 对象//扫雷放置的对象

this.flag = true; 布尔//判断是否为第一次点击

this.arr = new Array(); 数组//格子位置-行

this.arr\_2 = new Array(); 数组//格子位置-列

this.time\_dsq = null; 整数//所用时间-秒

this.time\_dc = ' '; 整数//时间显示-秒

this.arr\_time = [[],[],[]]; 数组//时间统计信息

this.details = [[],[],[]]; 数组//游戏统计详情

this.type = type; 整数//游戏类型：初级/中级/高级/自定义--1/2/3/4

3.4.6、算法和流程

无

3.4.7、函数说明

构造函数

3.4.8 全局数据结构与该模块的关系

无

**3.5、模块3：创建扫雷界面函数模块BuildTiles**

3.5.1、设计图

无

3.5.2、功能描述

在网页上创建扫雷的行列界面

3.5.3、输入数据

this.num1 整数//行数

this.num2 整数//列数

3.5.4、输出数据

obj.style.width 整数 表格宽度

obj.style.height 整数 表格高度

3.5.5、数据设计

this.num1 整数//行数

this.num2 整数//列数

obj.style.width 整数 表格宽度

obj.style.height 整数 表格高度

indexOfdiv 整数 格子号

3.5.6、算法和流程

在传进来的对象上画整体格子，每个小格子51px大小，总大小就为个数\*单个大小，定义小格子class，为每个小格子添加索引，将小格子存入数组中，加入对象，绘制扫雷行列

3.5.7、函数说明

被构造函数调用，调用点击的绑事件函数

3.5.8 全局数据结构与该模块的关系

使用构造函数中数据

**3.6、模块4：绑事件函数模块Event**

3.6.1、设计图

无

3.6.2、功能描述

定义各类鼠标操作事件

3.6.3、输入数据

鼠标悬停，鼠标移出，鼠标按下（左键，中键，右键），鼠标弹起

3.6.4、输出数据

常数：左键0

中键1

右键2

obj 对象 //扫雷放置的对象

3.6.5、数据设计

obj 对象 //扫雷放置的对象

Index 整数 索引，鼠标的格子位置

3.6.6、算法和流程

分别定义与判断鼠标各个状态

3.6.7、函数说明

被创建页面函数和布雷函数调用

3.6.8 全局数据结构与该模块的关系

无

**3.6、模块5：点击调用函数模块ChangeStyle**

3.6.1、设计图

无

3.6.2、功能描述

鼠标点击之后进行判断，是否有雷，有雷结束游戏并统计信息，无雷翻开并显示周围雷数，并自动判断周围雷数翻开

3.6.3、输入数据

0/1/2 常数 鼠标操作事件

Obj 对象 扫雷放置的对象

Index 整数 索引

3.6.4、输出数据

网页动态变化

3.6.5、数据设计

3.6.6、算法和流程

是左键的话，this.flag 是之前定义的用于判断是否为第一次点击，store函数，存放被点击的格子周围的8个格子，如果是第一次点击 即调用布雷函数 更改flag状态，如果不是第一次点击，被点击的格子不是未点击状态，无效。如果不是雷。改为翻开状态，递归函数判断周围格子的情况，就是扫雷游戏上一点开会出现一片的那种，最后判断游戏是否结束。

是右键进行标记。

是中键调用快速扫雷函数模块

3.6.7、函数说明

调用：存储周围格子的函数

布雷函数

结束判断函数

统计信息显示周围函数

被绑事件函数调用

3.6.8 全局数据结构与该模块的关系

无

**3.7、模块6：布雷函数模块 SetMineCraft**

3.7.1、设计图

无

3.7.2、功能描述

在第一次点击之后根据雷的个数随机布雷

3.7.3、输入数据

Num //雷的个数

arr\_first //最开始被点击的格子周围的八个

num\_first //被点击的那个格子的索引

3.7.4、输出数据

无

3.7.5、数据设计

Num 整数 雷的个数

arr\_first 整数 最开始被点击的格子周围的八个

num\_first 整数 被点击的那个格子的索引

tiles 数组 里面存放的是每个小格子

arr\_index 数组 周围八个格子的索引

3.7.6、算法和流程

在第一次点击之后根据雷的个数使用随机函数布雷，即更改某些格子的值变为有雷的状态

3.7.7、函数说明

调用绑事件函数，点击调用函数，在第一次点击之后进行布雷

3.7.8 全局数据结构与该模块的关系

使用构造函数中雷的个数和鼠标位置的索引

**3.8、模块7：存储周围格子的函数模块：Store**

3.8.1、设计图

无

3.8.2、功能描述

储存周围的位置,传入格子的index

3.8.3、输入数据

num 整数 通过点击调用函数输入格子的索引

3.8.4、输出数据

indexs 整数//周围雷的个数

obj 对象//扫雷放置的对象

3.8.5、数据设计

num 整数 通过点击调用函数输入格子的索引

indexs 整数//周围雷的个数

obj 对象//扫雷放置的对象

tiles\_2d[ ][ ] 数组//存储格子的行列数

3.8.6、算法和流程

数组存储行列标号，将行列存入数组，在数组中遍历并存储周围雷的个数

3.8.7、函数说明

被点击函数和显示函数调用

3.8.8 全局数据结构与该模块的关系

对传入的索引存储其周围格子的雷数

**3.9、模块8：显示周围函数模块 ShowAll**

3.9.1、设计图

无

3.9.2、功能描述

点击后自动翻开周围无雷格子并显示雷数

3.9.3、输入数据

num 整数 通过点击调用函数输入格子的索引

3.9.4、输出数据

动态翻开显示

3.9.5、数据设计

num 整数 通过点击调用函数输入格子的索引

Arr 数组 格子的行，列

3.9.6、算法和流程

如果该格子周围没有雷，自动翻开周围8个格子，除开标记，然后再判断周围八个格子的周围8隔格子是否有雷，使用递归方法。

3.9.7、函数说明

被点击调用函数调用

3.9.8 全局数据结构与该模块的关系

传入格子的索引

**3.10、模块9：结束判断模块：Over**

3.10.1、设计图

无

3.10.2、功能描述

判断结束并统计信息

3.10.3、输入数据

Num 整数//雷的个数

obj 对象//扫雷放置的对象

3.10.4、输出数据

字符串：'恭喜你获得成功'/'被炸死！'

Flag true/false //赢或输

3.10.5、数据设计

Num 整数//雷的个数

obj 对象//扫雷放置的对象

Flag true/false //赢或输

3.10.6、算法和流程

获取剩余的雷,如果被排出来的格子数等于总格子数-雷数，游戏成功结束,如果被点击的是雷，则炸死，最后调用统计函数进行游戏数据统计

3.10.7、函数说明

被点击事件函数调用判断点击后是否结束游戏

3.10.8 全局数据结构与该模块的关系

使用构造函数中雷的个数和存储的标记数，将输赢结果传入统计函数

**3.11、模块10：统计信息模块DetailStatistics**

3.11.1、设计图

无

3.11.2、功能描述

每次结束游戏后将数据存入localStorage中，统计已玩游戏，已胜游戏，胜率，最多连胜，最多连败，当前连局

3.11.3、输入数据

(play, state)//传入0开始/1结束和胜负（ture-胜，false-败）

this.time\_dsq = null; 整数//所用时间-秒

this.time\_dc = ' '; 整数//时间显示-秒

this.tiles = new Array(); 数组//数组里面存放的是每个小格子

this.obj = obj; 对象//扫雷放置的对象

this.type = type; 整数//游戏类型：初级/中级/高级/自定义--1/2/3/4

3.11.4、输出数据

arr\_details 数组 已玩游戏，已胜游戏，胜率，最多连胜，最多连败，当前连局；

arr\_grade 数组 所用时间

3.11.5、数据设计

this.time\_dsq = null; 整数//所用时间-秒

this.time\_dc = ' '; 整数//时间显示-秒

this.tiles = new Array(); 数组//数组里面存放的是每个小格子

this.obj = obj; 对象//扫雷放置的对象

this.type = type; 整数//游戏类型：初级/中级/高级/自定义--1/2/3/4

arr\_details 数组 已玩游戏，已胜游戏，胜率，最多连胜，最多连败，当前连局；

arr\_grade 数组 所用时间

3.11.6、算法和流程

对比传入数据计算结果并存入数组

3.11.7、函数说明

被点击事件函数和结束判断函数调用

3.11.8、 全局数据结构与该模块的关系

计算数据信息，传入HTML模块动态显示

# 4、 接口设计

**4.1、 用户接口**

鼠标移动，点击，键盘输入，浏览器显示

**4.2、 外部接口**

鼠标控制，键盘输入

**4.3、 内部接口**

4.3.1、 接口说明

见上部分软件结构图和函数说明，在HTML网页中调用构造函数，构造函数通过鼠标的点击调用次一级的各个函数

4.3.2、 调用方式

均为函数调用