# 1、系统概述

**1.1、系统简介**

windows扫雷游戏的网页版复刻

**1.2、系统目标**

1. 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
2. 扫雷游戏的功能和规则和Windows经典扫雷游戏相同
3. 具有难度调节（行列数）和统计信息功能

**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**

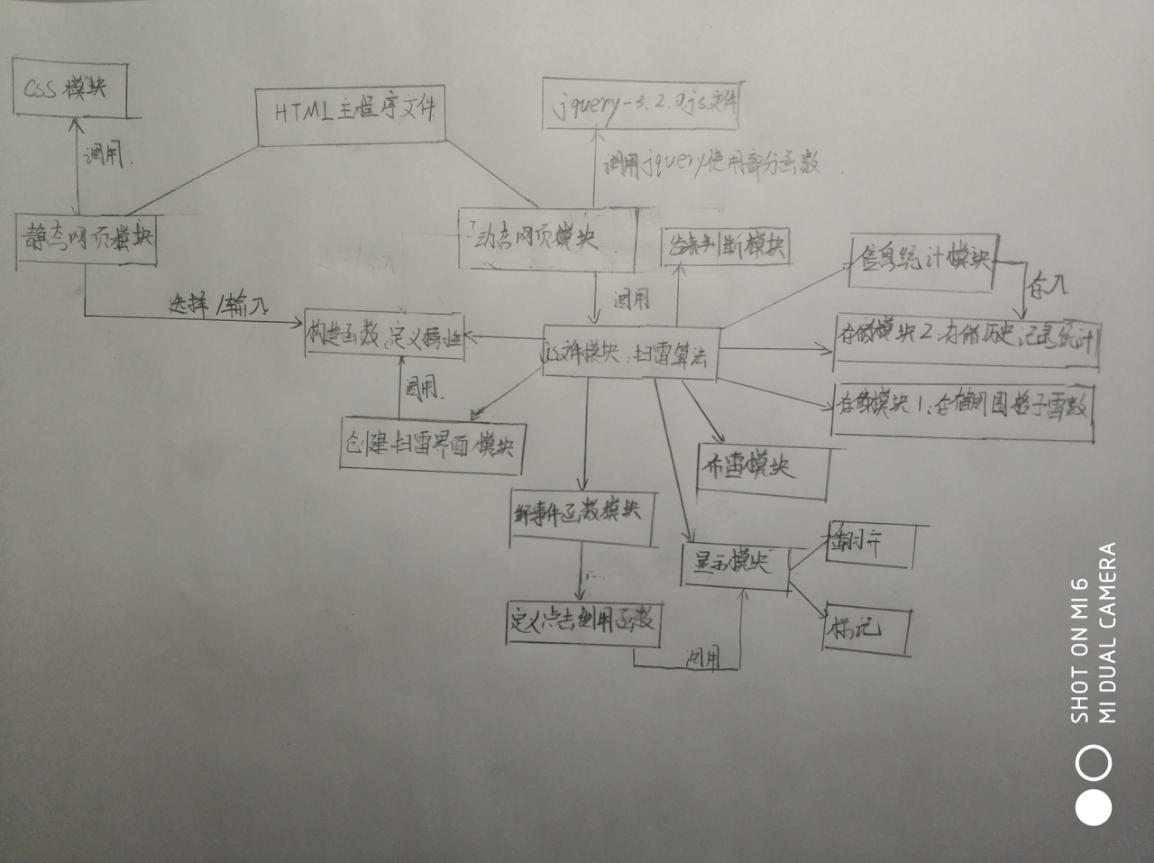
编程软件：visual studio code 2017

开发语言：HTML，CSS，JavaScript

试运行软件：google chrome浏览器最新版

# 2、总体结构设计

**2.1、软件结构**



****

**2.2、设计思想**

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

# 3、模块设计

在此以列表形式说明各模块的名称、功能、接口等信息。

**3.1、模块1：静态网页模块** ：saolei\_cg.html, saolei\_cg.css

3.1.1、功能描述

用静态的HTML和CSS创建网页

3.1.2、接口描述

输入：玩家的选择信息，包括难度设置选择（初级/中级/高级/自定义），行列设置（行数width,列数height，雷数mine\_num）并输入到JS模块。

输出：从JS模块获取统计信息

已玩游戏：

已胜游戏：

游戏胜率：

最多连胜：

最多连败：

当前连局：

平均耗时：

3.1.3、实现思路

创建基本网页，提供选择和输入功能

**3.2、模块2：构造函数模块：MineCraft**

3.2.1、功能描述

定义一个构造函数,里面是一系列的属性

3.2.2、接口描述

输入：

Width //整个扫雷的行数

Height //整个扫雷的列数

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

调用：

创建扫雷界面模块 BuildTiles()

3.2.3、数据结构描述

this.num1 = width;//行数

this.num2 = height;//列数

this.mine\_num = mine\_num;//雷的个数

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

this.obj = obj;//扫雷放置的格子对象

this.flag = true;//判断是否为第一次点击，默认为ture

this.arr = new Array();//数组，存储周围八个格子的索引

this.arr\_2 = new Array();//数组，存储周围八个格子的索引

this.time\_dsq = null;//计时数-定时器

this.time\_dc = '';//所用时间

this.arr\_time = new Array();//时间统计信息

this.details = new Array();//游戏统计详情

this.type = type;//游戏类型：初级/中级/高级/自定义

this.BuildTiles();//创建游戏函数

**3.3、模块3：创建扫雷界面函数模块：BuildTiles**

3.3.1、功能描述

在网页上创建扫雷的行列界面

3.3.2、接口描述

被构造函数模块调用

调用绑事件函数模块

3.3.3、数据结构描述

索引，类

num1 //整个扫雷的行数

num2 //整个扫雷的列数

mine\_num //雷的个数

tiles = [] //数组里面存放的是每个小格子

obj //扫雷放置的对象

type //游戏类型：初级/中级/高级/自定义

3.3.4、实现思路

在传进来的对象上画整体格子，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，和格子图片插入到整个扫雷界面中

**3.4、模块4：绑事件函数模块** ：Event

3.4.1、功能描述

定义各类鼠标操作事件

3.4.2、接口描述

输入：鼠标悬停，鼠标移出，鼠标按下（左键，中键，右键），鼠标弹起

输出：左键0/中键1/右键2

接口：调用点击函数模块ChangeStyle

3.1.3、数据结构描述

obj //扫雷放置的对象

3.4.4、实现思路

将鼠标的各个状态分类并定义操作事件

**3.5、模块5：点击调用函数模块：ChangeStyle**

3.5.1、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

3.5.2、接口描述

输入：鼠标操作事件，扫雷放置的对象和索引

调用：存储周围格子的函数模块：Store

布雷函数模块：SetMineCraft

结束判断模块：Over

统计信息模块:DetailStatistics

显示周围函数模块：ShowAll

3.5.3、数据结构描述

obj //扫雷放置的对象

num1: // 鼠标判断

num\_index: //格子索引

3.5.4、实现思路

如果是第一次点击，即调用布雷函数，如果不是雷，改为翻开状态并显示周围雷数，如果是雷则调用结束模块和统计信息模块

**3.6、模块6：布雷函数模块：SetMineCraft**

3.6.1、功能描述

在第一次点击之后根据雷的个数随机布雷

3.6.2、接口描述

输入信息：num,arr\_first,num\_first//雷的个数、最开始被点击的格子周围的八个、被点击的那个格子

调用绑事件函数模块Event

被点击调用函数模块：ChangeStyle调用

3.6.3、数据结构描述

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

tiles = [] //数组里面存放的是每个小格子

3.6.4、实现思路

在点击之后根据雷的个数使用随机函数布雷

**3.7、模块7：存储周围格子的函数模块：Store**

3.7.1、功能描述

在数组中遍历并存储周围雷的个数

3.7.2、接口描述

输入: num 通过点击调用函数输入格子的索引

输出：indexs //周围雷的个数

obj //扫雷放置的对象

3.7.3、数据结构描述

num //传入格子索引

indexs //周围雷的个数

3.7.4、实现思路

传入格子的索引，遍历其上下左右的格子并存入indexs统计个数

**3.8、模块8：显示周围函数模块：ShowAll**

3.8.1、功能描述

如果该格子周围没有雷，自动翻开周围8个格子，然后再判断周围八个格子的周围8隔格子是否有雷

3.8.2、接口描述

输入：num 通过点击调用函数输入格子的索引

3.8.3、数据结构描述

num //传入格子索引

3.8.4、实现思路

递归实现

**3.9、模块9：结束判断模块：Over**

3.9.1、功能描述

判断结束并统计信息

3.9.2、接口描述

输入： Num //雷的个数

obj //扫雷放置的对象

输出：字符串：'恭喜你获得成功'/'被炸死！'

true/false //赢或输

调用：统计信息模块DetailStatistics

3.9.3、数据结构描述

obj //扫雷放置的对象

Num //雷的个数

3.9.4、实现思路

如果被排出来的格子数等于总格子数-雷数，这游戏成功结束

如果被点击的是雷，则炸死

然后统计结果

**3.10、模块10：统计信息模块:DetailStatistics**

3.10.1、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

3.10.2、接口描述

输入：time\_arr = [[],[],[]] //时间统计信息

type //游戏类型：初级/中级/高级/自定义

Play //bool 输赢

输出：details = [[],[],[]] // 游戏统计详情

3.10.3、数据结构描述

数组

3.10.4、实现思路

用数组统计和储存各项数据

# 4、数据库与数据结构设计

**4.1、 数据库及数据表**

**无**

**4.2、 数据结构设计**

给出本系统内所使用的每个数据结构的名称、标识符以及它们之中每个数据项、记录、文卷和系的标识、定义、长度及它们之间的层次的或表格的相互关系。

构造函数：

var MineCraft = function(width, height, mine\_num, obj, type)

{

this.num1 = width;//行数

this.num2 = height;//列数

this.mine\_num = mine\_num;//雷的个数

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

this.obj = obj;//扫雷放置的格子对象

this.flag = true;//判断是否为第一次点击，默认为ture

this.arr = new Array();//数组，存储周围八个格子的索引

this.arr\_2 = new Array();//数组，存储周围八个格子的索引

this.time\_dsq = null;//计时数-定时器

this.time\_dc = '';//所用时间

this.arr\_time = new Array();//时间统计信息

this.details = new Array();//游戏统计详情

this.type = type;//游戏类型：初级/中级/高级/自定义

this.BuildTiles();//创建游戏函数

};

**4.3、 数据存储设计**

统计信息存入本地的localStorege中

# 5、接口设计

**5.1、外部接口**

鼠标控制，键盘输入

**5.2、内部接口**

调用jQuery文件使用函数，内部定义函数的调用

# 6、其他设计

无