**JavaScript课程设计**

**贪吃蛇小游戏**

湖北工业大学工程技术学院

**2019年12月20日**

目录

[摘要 课程设计内容与要求](#_Toc1991664579_WPSOffice_Level1) [1](#_Toc1991664579_WPSOffice_Level1)

[项目分析](#_Toc1169818692_WPSOffice_Level1) [2](#_Toc1169818692_WPSOffice_Level1)

[1. 可行性分析](#_Toc929968159_WPSOffice_Level1) [2](#_Toc929968159_WPSOffice_Level1)

[2. 系统需求分析](#_Toc588865447_WPSOffice_Level1) [2](#_Toc588865447_WPSOffice_Level1)

[a) 功能性需求](#_Toc1169818692_WPSOffice_Level2) [2](#_Toc1169818692_WPSOffice_Level2)

[b) 性能需求](#_Toc929968159_WPSOffice_Level2) [2](#_Toc929968159_WPSOffice_Level2)

[d) 开发环境](#_Toc588865447_WPSOffice_Level2) [3](#_Toc588865447_WPSOffice_Level2)

[e) 1](#_Toc1456922353_WPSOffice_Level2) [3](#_Toc1456922353_WPSOffice_Level2)

[3. 步骤分析](#_Toc1456922353_WPSOffice_Level1) [3](#_Toc1456922353_WPSOffice_Level1)

[4. 技术分析](#_Toc885443777_WPSOffice_Level1) [3](#_Toc885443777_WPSOffice_Level1)

[5. 原型图](#_Toc1739369976_WPSOffice_Level1) [3](#_Toc1739369976_WPSOffice_Level1)

[编码](#_Toc2043783668_WPSOffice_Level1) [5](#_Toc2043783668_WPSOffice_Level1)

[全部代码](#_Toc871174311_WPSOffice_Level1) [6](#_Toc871174311_WPSOffice_Level1)

[总结](#_Toc283139731_WPSOffice_Level1) [17](#_Toc283139731_WPSOffice_Level1)

# 摘要 课程设计内容与要求

本次JavaScript课程设计提供的项目是贪吃蛇小游戏，这是一款可以为大家提供的一种当前比较流行的休闲小游戏。贪吃蛇是家喻户晓的益智类小游戏，选择这个题目一是为了将自己的所学知识加以运用，二是一直以来贪吃蛇这个游戏就深深地吸引着很多人，它的制作方法对于很多同学而言都是很神秘的。所以我们希望通过这学期所学知识把它剖析开来，真真正正的了解它的本质和精髓。在这次学习中我们将从实践和实际的项目中提高自己的编程能力。

任务要求：

1. 游戏的启动：由用户触击键盘任一按键或页面提供的操纵区域任一按钮触发，蛇从区域中随机一点出发，运动限制在游戏区域内；
2. 提供符合贪吃蛇基本玩法的操纵功能：即可以用上、下、左、右键游戏区蛇的运动方向，使之向着有食物的方向运动，并吞吃食物使身体增长，如果蛇在移动过程中，撞到墙壁或身体交叉蛇头撞到自己的身体，即游戏结束；
3. 调节蛇的运动速度：即用户可以通过键盘按键时间长短调节蛇的速度来选择不同的难度；
4. 游戏版图：提供25\*25块可操纵区域；
5. 元素标志：
   1. 蛇：蛇头由与蛇身不相同的颜色进行区分，外观为一串圆圈，身体长度由固定数值5开始，每吃掉一个事物增加一个长度，运动方向为直线运动，只走横和竖的方向，不走斜线；
   2. 食物：与蛇不相同的矩形进行区分，食物出现按随即分布原则，蛇吃掉一份后随即在游戏区域内出现一份新的食物；
6. 得分：按蛇每吃掉一个食物得 100 分计算；
7. 界面友好，图形界面，方便玩家使用，具有较好的容错能力，玩家在游戏过程中，除了规定的按键外，其他按键均忽略。

# 项目分析

1. 可行性分析

贪吃蛇游戏是一种简单的大众的游戏，自从计算机实现以来，深受广大电脑玩家的喜爱，做一个简单的贪吃蛇小游戏。

贪吃蛇的核心算法时如何实现移动和吃掉食物，没有碰到食物的时候，把当前运动方向上的下个节点入队，并以蛇节点的颜色绘制这个节点，然后把头指针所指的节点出队，并以游戏框架内部背景色重绘出队的节点，这样就可以达到移动的效果。而在吃到食物的时候，则只需把食物入队即可。实现贪吃蛇的基本的蛇身移动、吃到食物身体增长、和碰到墙壁或自身死亡，能够暂停和开始。

本次课程设计采用HTML + CSS + JavaScript语言完成。JavaScript是一种属于网络的脚本语言，已经被广泛用于Web应用开发，通常JavaScript脚本是通过嵌入在HTML中来实现自身的功能的。是一种解释性脚本语言（代码不进行预编译）。主要用来向HTML（标准通用标记语言下的一个应用）页面添加交互行为。其跨平台特性支持绝大多数浏览器，可以在多种平台下运行。

JavaScript脚本语言同其他语言一样，有它自身的基本数据类型，表达式和算术运算符及程序的基本程序框架。JavaScript提供了四种基本的数据类型和两种特殊数据类型用来处理数据和文字。而变量提供存放信息的地方，表达式则可以完成较复杂的信息处理。

1. 系统需求分析
   1. 功能性需求

本课程设计主要是完成贪吃蛇小游戏的基本操作。用户可以自己练习和娱乐。系统需求如下：

* + 1. 利用方向键来改变蛇的运行方向。
    2. 空格键暂停或继续游戏，并在随机的地方产生食物。
    3. 吃到食物就变成新的蛇体，碰到墙或者自身则游戏结束，否则正常运行。
  1. 性能需求

在设计方面本着方便、实用及娱乐性高的宗旨，在对界面进行设计的过程中，始终坚持清晰明了，在性能方面能够实现效率高，不易出错等优点。

* 1. 界面需求

游戏主界面模块：主要包括游戏图形区域界面、游戏开始按钮、暂停游戏按钮、关闭按钮。游戏的主界面力求美观，爽心悦目。

游戏控制模块：主要完成控制游戏的开始、暂停、退出等功能，这部分做到易懂、易操作而且准确率高，不易出错。

游戏的主界面是进入游戏后，能够给玩家第一感官的部分，主要包括游戏图形区域界面、游戏的速度选择更新界面、游戏分数的显示更新界面、新生方块的预览、游戏开始按钮、暂停游戏按钮、退出游戏按钮以及游戏排行榜按钮。从很大程度上决定了玩家对游戏的兴趣问题，因此，游戏的主界面力求美观，爽心悦目。

游戏控制模块是游戏的中心环节，主要完成控制游戏的开始、暂停、退出等功能。为了能够给玩家一个很好的游戏环境，这部分做到易懂、易操作。

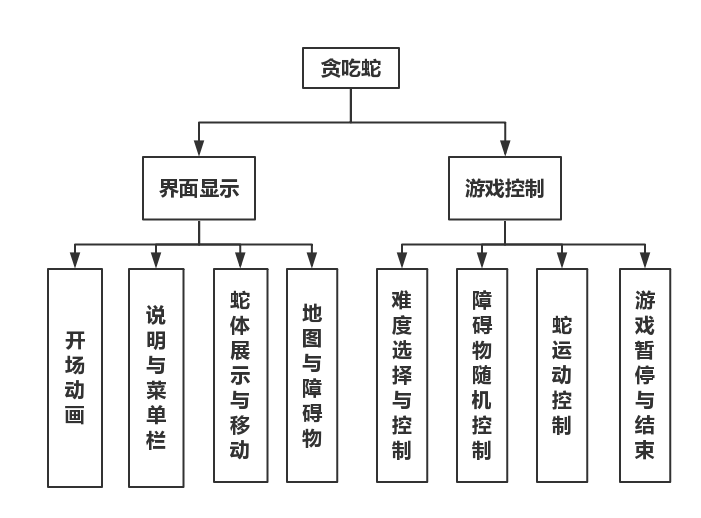
* 1. 开发环境

操作系统及版本：macOS 10.15.2

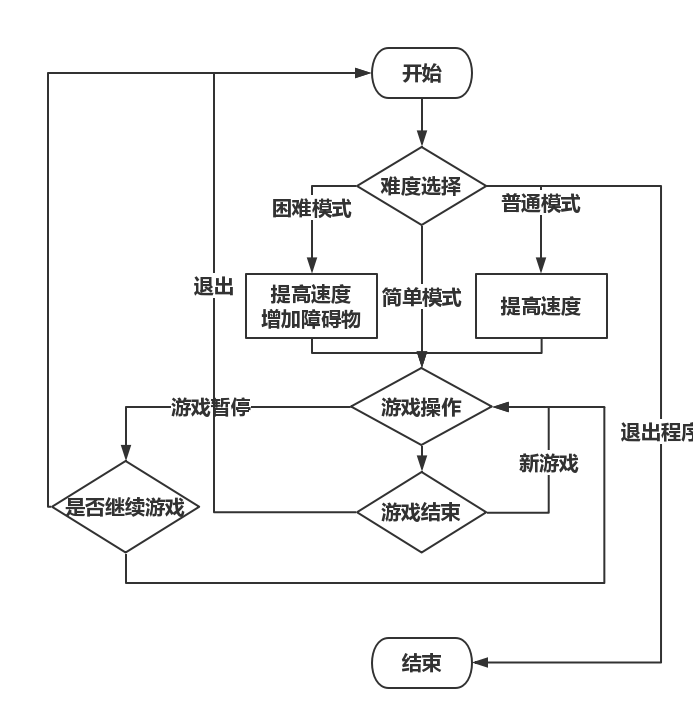
开发工具：JetBrains WebStorm 2019.3

开发语言：HTML + CSS + JavaScript

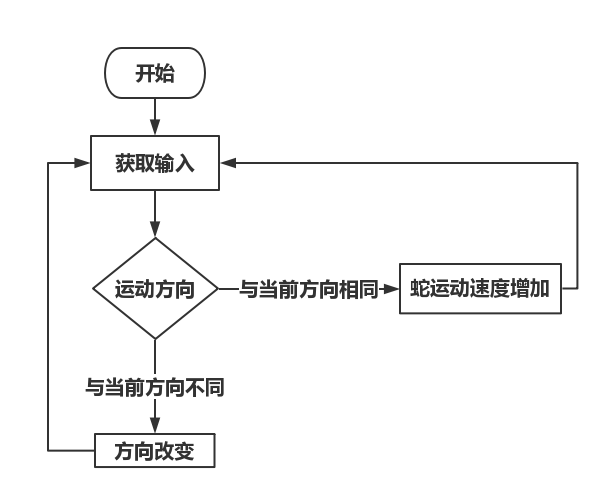
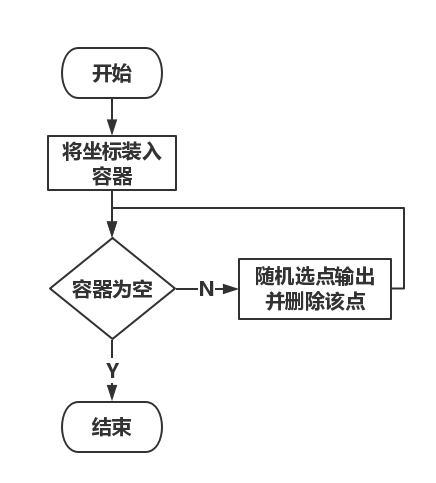
1. 系统概要设计
   1. 功能设计



* 1. 程序流程

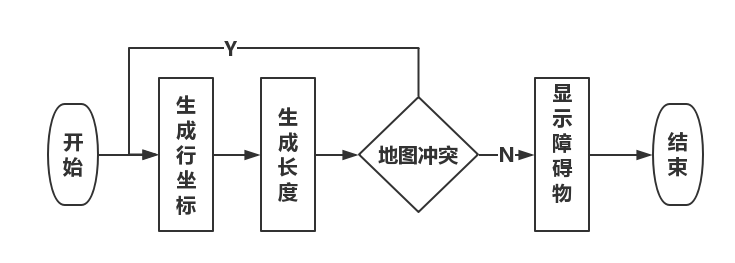


游戏处理流程图



载入动画流程图 控制方向流程图

* 1. 算法设计
     1. 如何控制蛇的移动，仔细分析蛇的移动可以看出，在蛇的每一次移动中，蛇头向前拓展一个单位，而后所有的蛇体都向前拓展一个单位，再把蛇尾去掉，倘若吃掉一个食物，那么就不用再把蛇尾去掉，蛇体的长度就又长了一个单位，为了描述当前蛇头的运动方向，还要用一个变量来保存，用其值表示当前蛇头的运动方向。
     2. 用循环来实现蛇的移动，假设当前不对蛇体进行转向，那么在界面上蛇就会按照当前运动方向进行移动，如果通过键盘输入了方向，也将其指令分为三种类别，第一种是与当前蛇的运动相同，那么就加速，使蛇更快的朝着当前方向运动，第二种就是与当前蛇的运动相反，此时应该忽略掉这种指令，不能进行实现，第三种就是剩下的方向，那么就按照输入的方向，对现在蛇的方向进行修改。
     3. 游戏结束的判断，贪吃蛇游戏结束存在两种判断，第一种是蛇头撞到障碍物比如地图边界，那么游戏结束，第二种是蛇头撞到自己的蛇体，游戏也会结束，那么就在蛇移动后判断，蛇头的坐标与障碍物和自己蛇体的所有点是否相同，如果存在一个相同的点，那么游戏结束，反之，游戏继续。
     4. 食物和障碍物的控制，食物的出现必须在游戏开始界面，或者当蛇吃掉前一个食物的时候，才会生成下一个食物，也就是说整个地图只会有一个食物，这里引入随机，让下一个食物随机出现在地图的每一个地方，而障碍物的出现，便是在难度选择之后进行出现，且在这一局游戏中改变。但是食物和障碍物都会出现冲突，比如下一个食物的出现，用随机生成了一个坐标，但是这个坐标却是蛇体的一部分或者是某个障碍物，那么这就不合理，于是在随机的前提下，要确保生成数据的合理性，同时为了使障碍物更加规范，障碍物也不一定是一个点，可能是一堵墙，或者两堵墙挨在一起，这些都需要通过细节控制。

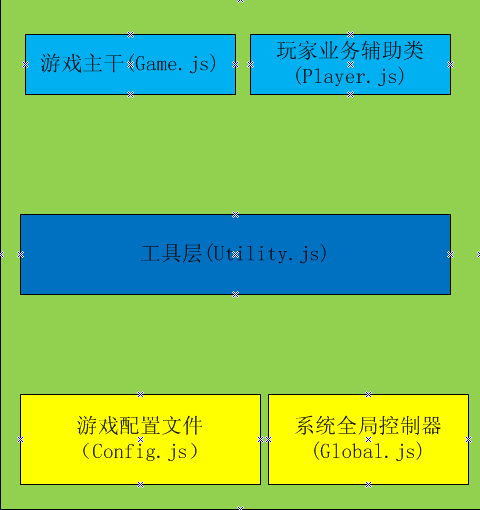


障碍物生成流程图

1. 步骤分析
2. 技术分析
3. 原型图

游戏架构：

此游戏架构大概分为三层：



蛇碰到边界或者碰到自己

游戏中途欲中断，稍后继续游戏

单击“是”再来一局

单击“否”退出游戏

单击“暂停游戏”

游戏暂停

单击“开始游戏”按钮，

游戏开始，蛇开始随机。

进入游戏画面

再次单击开始游戏，游戏恢复

**图1** **游戏处理流程**图

键盘操作：

向上

右移 F11加速

左移 F12减速

向下 Enter开始/暂停

<https://blog.csdn.net/weixin_44389268/article/details/86529870>

<https://blog.csdn.net/u012054869/article/details/51734537>

<https://blog.csdn.net/iteye_1819/article/details/82525830>

<https://www.jianshu.com/p/0e526102c6fe>

<https://blog.csdn.net/weixin_42839080/article/details/81416452>

<https://blog.csdn.net/weixin_42839080/article/details/81458736>

# 编码

XX实现

页面美化

# 全部代码

<!DOCTYPE html>  
<html lang="en">  
<head>  
 <meta charset="UTF-8">  
 <title>贪吃蛇</title>  
</head>  
<style type="text/css">  
 .face {  
 height: 400px;  
 width: 600px;  
 margin-left: auto;  
 margin-right: auto;  
 position: relative;  
 background-color: pink;  
 }  
  
 #playground {  
 height: 400px;  
 width: 450px;  
 float: left;  
 position: relative;  
 }  
  
 .menu {  
 height: 400px;  
 width: 150px;  
 float: left;  
 background-color: skyblue;  
 }  
  
 #snack {  
 height: 20px;  
 width: 20px;  
 background-color: red;  
 position: absolute;  
 left: 0px;  
 top: 0px;  
 }  
  
 #food {  
 height: 20px;  
 width: 20px;  
 background: blue;  
 position: absolute;  
 }  
  
 .body {  
 height: 20px;  
 width: 20px;  
 background: green;  
 position: absolute;  
 left: 0px;  
 top: 0px;  
 }  
  
 #score {  
 font-size: 30px;  
 font-weight: bold;  
 color: red;  
 }  
  
 .menu div {  
 font-size: 20px;  
 font-weight: bold;  
 }  
</style>  
<body>  
<div class="face">  
 <!-- 小蛇移动的操场 -->  
 <div id="playground">  
 <!-- 食物 界面中的蓝色小方块-->  
 <div id="food"></div>  
 <!-- 小蛇 界面中的红色小方块-->  
 <div id="snack"></div>  
 </div>  
 <!-- 计算得分 -->  
 <div class="menu">  
 <div>得分<span id="score"></span></div>  
 </div>  
</div>  
<script type="text/javascript">  
 //获取元素节点  
 var ***jsDiv*** = ***document***.getElementById("playground");  
 var ***jsSnack*** = ***document***.getElementById("snack");  
 var ***jsFood*** = ***document***.getElementById("food");  
 var ***jsBody*** = ***document***.getElementById("playground");  
 var ***jsScore*** = ***document***.getElementById("score");  
 //设置全局变量  
 var ***timer***;//创建定时器为全局变量  
 var ***timer1*** = setInterval(eat, 10);//检测位置碰撞，并且吃掉食物；  
 var ***srr*** = [];//记录蛇行动的位置  
 var ***num*** = 0;//记录数组的长度  
 var ***jsSnackBody***;//么米吃掉一个食物，蛇的身体  
  
 //开始游戏  
 ***document***.onkeydown = function (e) {  
 var evt = e || ***window***.event;  
 switch (evt.keyCode) {  
 //向左移动  
 case 37:  
 clearInterval(***timer***);  
 ***timer*** = ***window***.setInterval(runLeft, 10)  
  
 function runLeft() {  
 if (***jsSnack***.offsetLeft > 0) {  
 ***jsSnack***.style.left = ***jsSnack***.offsetLeft - 1 + "px";  
 ***jsSnack***.style.top = ***jsSnack***.offsetTop + "px";  
 //记录小蛇的位置  
 ***srr***.push([***jsSnack***.offsetLeft, ***jsSnack***.offsetTop]);  
 ***num***++;//记录数组的长度  
 }  
 }  
  
 break;  
 //向上移动  
 case 38:  
 clearInterval(***timer***);  
 ***timer*** = ***window***.setInterval(runTop, 10);  
  
 function runTop() {  
 if (***jsSnack***.offsetTop > 0) {  
 ***jsSnack***.style.top = ***jsSnack***.offsetTop - 1 + "px";  
 ***jsSnack***.style.left = ***jsSnack***.offsetLeft + "px";  
 //记录小蛇的位置  
 ***srr***.push([***jsSnack***.offsetLeft, ***jsSnack***.offsetTop]);  
 ***num***++;//记录数组的长度  
 }  
 }  
  
 break;  
 //向右移动  
 case 39:  
 clearInterval(***timer***);  
 ***timer*** = ***window***.setInterval(runRight, 10);  
  
 function runRight() {  
 if (***jsSnack***.offsetLeft + ***jsSnack***.offsetWidth <= 450) {  
 ***jsSnack***.style.left = ***jsSnack***.offsetLeft + 1 + "px";  
 ***jsSnack***.style.top = ***jsSnack***.offsetTop + "px";  
 //记录小蛇的位置  
 ***srr***.push([***jsSnack***.offsetLeft, ***jsSnack***.offsetTop]);  
 ***num***++;//记录数组的长度  
 }  
 }  
  
 break;  
 //向下移动   
 case 40:  
 clearInterval(***timer***);  
 ***timer*** = ***window***.setInterval(runBottom, 10);  
  
 function runBottom() {  
 if (***jsSnack***.offsetTop + ***jsSnack***.offsetHeight <= 400) {  
 ***jsSnack***.style.top = ***jsSnack***.offsetTop + 1 + "px";  
 ***jsSnack***.style.left = ***jsSnack***.offsetLeft + "px";  
 //记录小蛇的位置  
 ***srr***.push([***jsSnack***.offsetLeft, ***jsSnack***.offsetTop]);  
 ***num***++;//记录数组的长度  
 }  
 }  
  
 break;  
 }  
 }  
  
 //食物随机出现  
 function Pos() {  
 ***jsFood***.style.left = parseInt(***Math***.random() \* (430 - 20 + 1) + 20) + "px";  
 ***jsFood***.style.top = parseInt(***Math***.random() \* (380 - 20 + 1) + 20) + "px";  
 }  
  
 Pos();  
  
 function eat() {  
  
 rectangleCrashExamine(***jsSnack***, ***jsFood***);  
  
 function rectangleCrashExamine(obj1, obj2) {  
 var obj1Left = obj1.offsetLeft;  
 var obj1Width = obj1.offsetLeft + obj1.offsetWidth;  
 var obj1Top = obj1.offsetTop;  
 var obj1Height = obj1.offsetTop + obj1.offsetHeight;  
 var obj2Left = obj2.offsetLeft;  
 var obj2Width = obj2.offsetLeft + obj2.offsetWidth;  
 var obj2Top = obj2.offsetTop;  
 var obj2Height = obj2.offsetTop + obj2.offsetHeight;  
 //检测碰撞  
 //碰撞检测原理：  
 //蛇在实物的左边、右边、上边、下边的时候，说明没有发生碰撞，那么我们取反，就说明发生碰撞  
 if (!(obj1Left > obj2Width || obj1Width < obj2Left || obj1Top > obj2Height || obj1Height < obj2Top)) {  
 //碰撞后身体  
 ***jsSnackBody*** = ***document***.createElement("div");  
 ***jsSnackBody***.setAttribute("class", "body");  
 ***jsBody***.appendChild(***jsSnackBody***);  
 Pos();//怪物的位置随机变换  
 setInterval(follow, 10);//身体跟随的定时器  
 }  
 }  
 }  
  
 function follow() {  
 //检查一共添加了多少身体  
 var bodyNum = ***document***.getElementsByClassName("body");  
 //记录得分  
 ***jsScore***.innerHTML = bodyNum.length;  
 //蛇每次移动1个像素，那么新的身体应该跟随在当前数组的倒数第20个数组的位置;依次加等;  
 var place = 0;  
 for (var i = 0; i < bodyNum.length; i++) {  
 place += 20;  
 bodyNum[i].style.left = ***srr***[***num*** - place][0] + 'px';  
 bodyNum[i].style.top = ***srr***[***num*** - place][1] + 'px';  
 }  
 }  
</script>  
</body>  
</html>

# 总结