

JavaScript 课程设计

贪吃蛇小游戏

2019-2020 学年第一学期

湖北工业大学工程技术学院

201710253110 唐可寅

2019 年 12 月 21 日

提交日期: 2019 年 12 月 31 日

目录

摘要 课程设计内容与要求.....1

项目分析.....2

1. 可行性分析..... 2

2. 系统需求分析..... 2

 a) 功能性需求..... 2

 b) 性能需求..... 2

 c) 界面需求..... 3

 d) 开发环境..... 3

3. 系统概要设计..... 3

 a) 系统架构设计..... 3

 b) 系统功能设计..... 4

 c) 系统流程设计..... 4

 游戏处理流程图..... 4

 d) 系统算法设计..... 5

 障碍物生成流程图..... 6

编码..... 7

1. 基本功能实现（20 行代码） 7

2. 创建食物对象实现..... 7

3. 蛇移动的函数..... 8

系统代码清单..... 9

总结..... 18

摘要 课程设计与要求

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、右键改变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 块可操纵单元；
5. 元素标志：
 - a) 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 开始递增，每吃掉一个食物增加一个长度，运动方向为直线运动，只走横和竖的方向，不走斜线；
 - b) 食物：由与蛇不相同的矩形进行区分，食物出现按随机分布原则，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：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。

贪吃蛇的核心算法是如何实现移动和吃掉食物,没有碰到食物时,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。而在吃到食物时,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。

2. 系统需求分析

a) 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:

- i. 利用方向键来改变蛇的运行方向。
- ii. 空格键暂停或继续游戏,并在随机的地方产生食物。
- iii. 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。

b) 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。

c) 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，这部分要做到易懂、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。

d) 开发环境

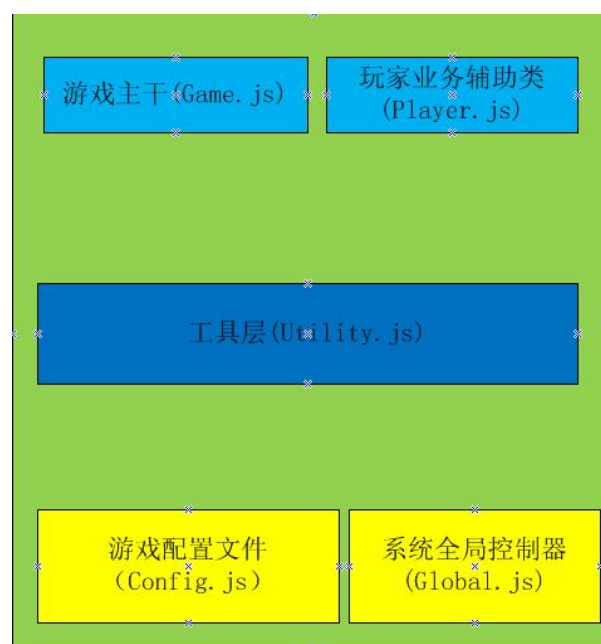
操作系统及版本：macOS 10.15.2 & Chrome 79.0.3945.79

开发工具：JetBrains WebStorm 2019.3

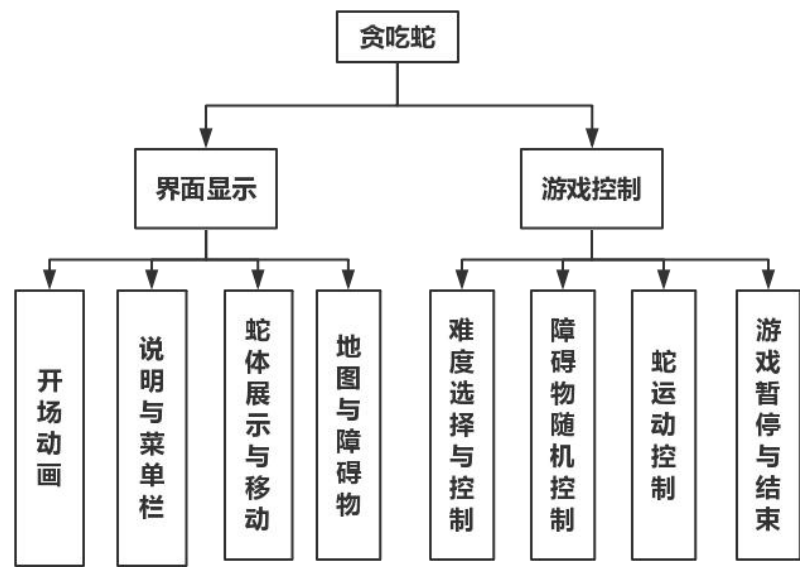
开发语言：HTML + CSS + JavaScript

3. 系统概要设计

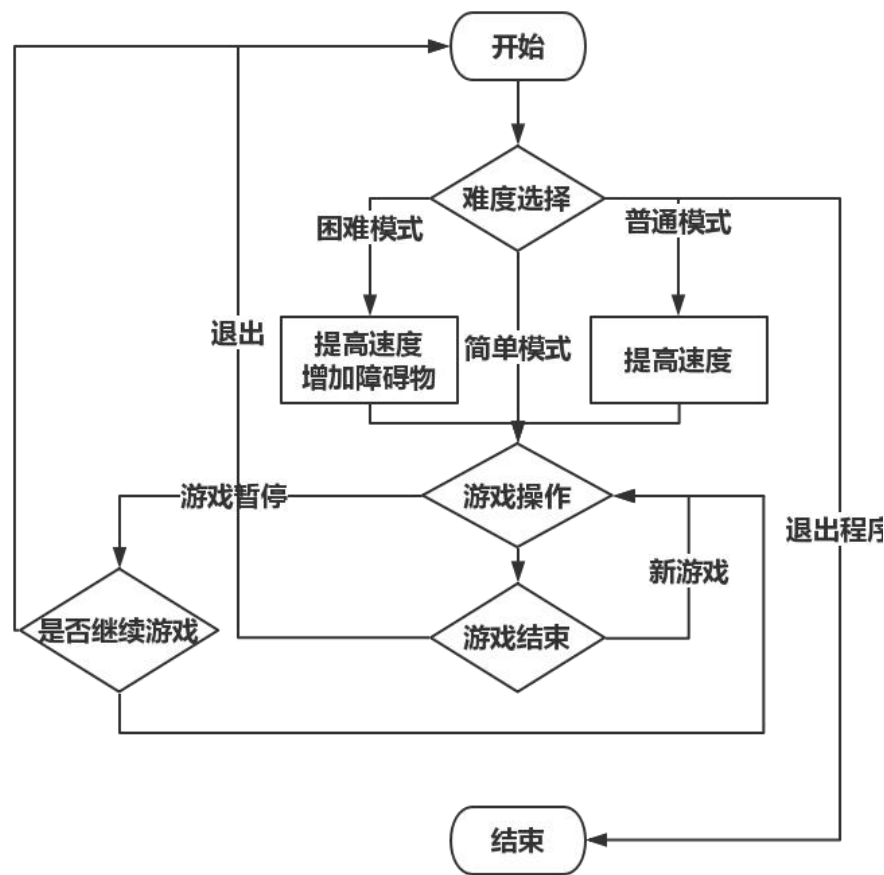
a) 系统架构设计



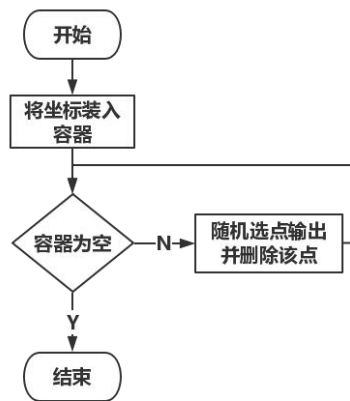
b) 系统功能设计



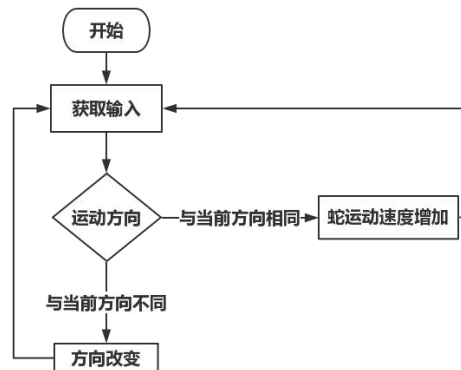
c) 系统流程设计



游戏处理流程图



载入动画流程图

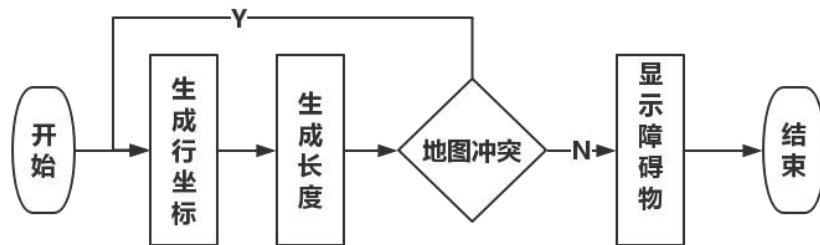


控制方向流程图

d) 系统算法设计

- i. 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。
- ii. 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。
- iii. 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。
- iv. 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

编码

1. 基本功能实现 (20 行代码)

```
<html lang="zh">
<head></head>
<body>
<canvas id="can" width="400" height="400" style="background: Black"></canvas>
<script>
    var sn = [42, 41], dz = 43, fx = 1, n, ctx =
document.getElementById("can").getContext("2d");

    function draw(t, c) {
        ctx.fillStyle = c;
        ctx.fillRect(t % 20 * 20 + 1, ~(t / 20) * 20 + 1, 18, 18);
    }

    document.onkeydown = function (e) {
        fx = sn[1] - sn[0] == (n = [-1, -20, 1, 20][(e || event).keyCode - 37] ||
fx) ? fx : n
    };
    !function () {
        sn.unshift(n = sn[0] + fx);
        if (sn.indexOf(n, 1) > 0 || n < 0 || n > 399 || fx == 1 && n % 20 == 0 ||
fx == -1 && n % 20 == 19)
            return alert("GAME OVER");
        draw(n, "Lime");
        if (n == dz) {
            while (sn.indexOf(dz = ~(Math.random() * 400)) >= 0) ;
            draw(dz, "Yellow");
        } else
            draw(sn.pop(), "Black");
        setTimeout(arguments.callee, 130);
    }();
</script>
</body>
</html>
```

2. 创建食物对象实现

```
function Food(width, height, bgColor) {
    this.width = width || 20;
```

```
this.height = height || 20;  
this.bgColor = bgColor || 'green';  
}
```

3. 蛇移动的函数

```
Snake.prototype.snakeMove = function (food, map) {  
  //蛇身体  
  for (var i = this.body.length - 1; i > 0; i--) {  
    this.body[i].x = this.body[i - 1].x;  
    this.body[i].y = this.body[i - 1].y;  
  }  
  //蛇头的移动  
  switch (this.direction) {  
    case 'right' : this.body[0].x++;break;  
    case 'left' : this.body[0].x--;break;  
    case 'top' : this.body[0].y--;break;  
    case 'bottom' : this.body[0].y++;break;  
  }  
  
  //-----吃食物长身体-----  
  var snakeHeaderX = this.body[0].x * this.width;  
  var snakeHeaderY = this.body[0].y * this.height;  
  var foodX = food.x;  
  var foodY = food.y;  
  if (snakeHeaderX == food.x && snakeHeaderY == food.y) {  
    this.body.push({  
      x: this.body[this.body.length - 1].x,  
      y: this.body[this.body.length - 1].y,  
      bgColor: getColorForRandom()  
    })  
    food.render(map);  
  }  
}
```

系统代码清单

```
<html lang="zh">
<head>
    <meta charset="UTF-8">
    <title>贪吃蛇</title>

    <!-- CSS 样式-->
    <style type="text/css">
        body {
            margin: 0;
            padding: 0;
            -webkit-user-select: none;
            -moz-user-select: none;
            -ms-user-select: none;
            user-select: none;
            -webkit-user-select: none;
            -moz-user-select: none;
            -ms-user-select: none;
            user-select: none;
        }

        table {
            border-collapse: collapse;
            width: 500px
        }

        td {
            border: 1px solid white;
            width: 20px;
            height: 20px;
        }

        #wrap {
            width: 500px;
            height: 500px;
            background-color: #8B8386;
            margin: 0 auto;
            position: absolute;
            border: 1px solid #8B2500;
        }
    </style>

```

```
#tar {
  width: 20px;
  height: 20px;
  position: absolute;
  background-color: #FFEE26;
}

.num {
  width: 20px;
  height: 20px;
  background-color: #FDFF68;
  border: 1px solid blue;
  border-radius: 50%;
  position: absolute;
}

input {
  color: red;
  font-size: 20px;
  font-weight: bold;
  position: absolute;
  display: block;
  width: 60px;
  height: 35px;
}

#btn2 {
  top: 45px;
  left: 0px;
}

#btn4 {
  top: 90px;
  left: 70px;
}

#btn1 {
  top: 45px;
  left: 140px;
}

#btn3 {
  top: 0px;
  left: 70px;
```

```
    }

    #btn {
        position: absolute;
        background-color: black;
        width: 200px;
        height: 125px;
    }

    #txt {
        width: 80px;
        height: 55px;
        background-color: black;
        position: absolute;
        left: 60px;
        top: 35px;
        color: white;
        font-size: 16px;
        font-weight: bold;
        line-height: 55px;
        text-align: center;
        cursor: move;
    }
</style>
</head>
<body>
<div id="btn" style="left:520px;top:20px">
    <input type="button" value="→" id="btn1"/>
    <input type="button" value="←" id="btn2"/>
    <input type="button" value="↑" id="btn3"/>
    <input type="button" value="↓" id="btn4"/>
    <div id="txt">点此拖动</div>
</div>
<div id="wrap" style="left:20px;top:20px">
    <div id="tar"></div>
    <table>
        <tbody id="tb">
            </tbody>
        </table>
    <div class="num"></div>
    <div class="num"></div>
    <div class="num"></div>
    <div class="num"></div>
    <div class="num"></div>
```

```
</div>

<!--JavaScript 代码-->
<script type="text/javascript">
    var tb = document.getElementById('tb');
    var wrap = document.getElementById('wrap');
    var tr = new Array();
    var td = new Array();
    for (var i = 0; i < 25; i++)          //绘制地图
    {
        tr[i] = document.createElement('tr');
        tb.appendChild(tr[i]);
        for (var j = 0; j < 25; j++) {
            td[j] = document.createElement('td');
            tr[i].appendChild(td[j]);
        }
    }
    var tar = document.getElementById('tar'); //食物定位
    function food() {
        var a, b;
        a = Math.random() * 460;
        a = Math.floor(a / 20) * 20 + 20;
        b = Math.random() * 460;
        b = Math.floor(b / 20) * 20 + 20;
        tar.style.left = a + "px";
        tar.style.top = b + "px";
    }

    food();
    var s = document.getElementsByClassName('num'); //定位蛇的位置
    s[0].style.backgroundColor = "#3EFF1A";
    var x = 300, y = 400, f, d, p, Fraction = 0, t = 600, tt = 60;
    for (var k = 0; k < s.length; k++) {
        s[k].style.left = x + "px";
        s[k].style.top = y + 20 * k + "px";
    }

    function move(obj, direction)          //控制移动函数开始
    {
        clearTimeout(d);
        s = document.getElementsByClassName('num');
        var l = new Array();
        l[0] = s[0].style.left;
        var w = new Array();
```

```
w[0] = s[0].style.top;
obj;          //移动方向
s[0].style.left = x + "px";
s[0].style.top = y + "px";
for (var k1 = 1; k1 < s.length; k1++) {
    l[k1] = s[k1].style.left;
    w[k1] = s[k1].style.top;
    s[k1].style.left = l[k1 - 1];
    s[k1].style.top = w[k1 - 1];
}
if (parseInt(s[0].style.left) == parseInt(tar.style.left) &&
parseInt(s[0].style.top) == parseInt(tar.style.top)) {
    t = t - tt;
    tt -= 5;
    if (tt == 0) tt = 5;
    var new_div = document.createElement('div');
    new_div.className = "num";
    Fraction += 100;
    if (p == 1) {
        new_div.style.top = s[0].style.top;
        new_div.style.left = parseInt(s[0].style.left) + 20 + "px";
    }
    if (p == 2) {
        new_div.style.top = s[0].style.top;
        new_div.style.left = parseInt(s[0].style.left) - 20 + "px";
    }
    if (p == 3) {
        new_div.style.left = s[0].style.left;
        new_div.style.top = parseInt(s[0].style.top) - 20 + "px";
    }
    if (p == 4) {
        new_div.style.left = s[0].style.left;
        new_div.style.top = parseInt(s[0].style.top) + 20 + "px";
    }
    s[0].parentNode.insertBefore(new_div, s[0]);
    food();
    s[0].style.backgroundColor = "#3EFF1A";
    for (var k = 1; k < s.length; k++) {
        s[k].style.backgroundColor = "#FDFF68";
    }
}
if (parseInt(s[0].style.left) < 0 || parseInt(s[0].style.left) >= 500 ||
parseInt(s[0].style.top) < 0 || parseInt(s[0].style.top) >= 500) {
    alert("GAME OVER!" + "你的分数为: " + Fraction + "分");
}
```

```
        food();
        x = 300;
        y = 400;
        for (var k = 0; k < 5; k++) {
            s[k].style.left = x + "px";
            s[k].style.top = y + 20 * k + "px";
        }
        for (var kk = s.length - 1; kk > 4; kk--) {
            s[kk].parentNode.removeChild(s[kk]);
        }
        t = 700;
        tt = 60;
        return false;
    }
    for (k = 1; k < s.length; k++) {
        if (parseInt(s[0].style.left) == parseInt(s[k].style.left) &&
        parseInt(s[0].style.top) == parseInt(s[k].style.top)) {
            alert("GAME OVER!" + "你的分数为: " + Fraction + "分");
            food();
            x = 300;
            y = 400;
            for (var k = 0; k < 5; k++) {
                s[k].style.left = x + "px";
                s[k].style.top = y + 20 * k + "px";
            }
            for (var kk = s.length - 1; kk > 4; kk--) {
                s[kk].parentNode.removeChild(s[kk]);
            }
            t = 700;
            tt = 60;
            return false;
        }
    }
    d = setTimeout(direction, t);
}

document.getElementById('btn1').onclick = function R() //绑定鼠标点击事件
{
    x = parseInt(s[0].style.left);
    y = parseInt(s[0].style.top);
    if (parseInt(s[0].style.left) < parseInt(s[1].style.left)) return false;
    p = 1;
    move(x = x + 20, R);
}
```



```
document.getElementById('btn2').onclick = function L() {
    x = parseInt(s[0].style.left);
    y = parseInt(s[0].style.top);
    if (parseInt(s[0].style.left) > parseInt(s[1].style.left)) return false;
    p = 2;
    move(x = x - 20, L);
}

document.getElementById('btn3').onclick = function T() {
    x = parseInt(s[0].style.left);
    y = parseInt(s[0].style.top);
    if (parseInt(s[0].style.top) > parseInt(s[1].style.top)) return false;
    p = 3;
    move(y = y - 20, T);
}

document.getElementById('btn4').onclick = function B() {
    x = parseInt(s[0].style.left);
    y = parseInt(s[0].style.top);
    if (parseInt(s[0].style.top) < parseInt(s[1].style.top)) return false;
    p = 4;
    move(y = y + 20, B);
}

document.onkeydown = function (event) //绑定键盘事件
{
    var e = event || window.event;
    if (e && e.keyCode == 39) {
        function R() {
            x = parseInt(s[0].style.left);
            y = parseInt(s[0].style.top);
            if (parseInt(s[0].style.left) < parseInt(s[1].style.left)) return
false;

            p = 1;
            move(x = x + 20, R);
        }

        R();
        return false;
    }
    if (e && e.keyCode == 37) {
        function L() {
            x = parseInt(s[0].style.left);
            y = parseInt(s[0].style.top);
            if (parseInt(s[0].style.left) > parseInt(s[1].style.left)) return
false;

            p = 2;
```

```
        move(x = x - 20, L);
    }

    L();
    return false;
}

if (e && e.keyCode == 38) {
    function T() {
        x = parseInt(s[0].style.left);
        y = parseInt(s[0].style.top);
        if (parseInt(s[0].style.top) > parseInt(s[1].style.top)) return
false;

        p = 3;
        move(y = y - 20, T);
    }

    T();
    return false;
}

if (e && e.keyCode == 40) {
    function B() {
        x = parseInt(s[0].style.left);
        y = parseInt(s[0].style.top);
        if (parseInt(s[0].style.top) < parseInt(s[1].style.top)) return
false;

        p = 4;
        move(y = y + 20, B);
    }

    B();
    return false;
}

}

var btn = document.getElementById('btn');
var btn_l, btn_t;

function btn_move() {
    btn_l = clientX - parseInt(wrap.style.left)
}

confirm("可用键盘的上下左右四个按键进行游戏操作!")
var btn = document.getElementById('btn'); //开始制作操作区拖动效果
var txt = document.getElementById('txt');
var new_x, new_y, bool = false;
```

```
txt.onmousedown = function () {  
    var eve = event || window.event;  
    bool = true;  
    new_x = eve.clientX - parseInt(btn.style.left);  
    new_y = eve.clientY - parseInt(btn.style.top);  
}  
btn.onmousemove = function btn_move(event) {  
    if (bool) {  
        var e = event || window.event;  
        btn.style.left = e.clientX - new_x;  
        btn.style.top = e.clientY - new_y;  
        if (parseInt(btn.style.left) < 520) btn.style.left = "520px";  
        if (parseInt(btn.style.top) < 0) btn.style.top = "0px";  
        if (parseInt(btn.style.top) > 395) btn.style.top = "395px";  
    }  
}  
document.body.onmouseup = function () {  
    bool = false;  
}  
</script>  
</body>  
</html>
```

总结

通过 2 周的课程设计，在导师的指导和开发小组共同努力下，终于完成了贪吃蛇小游戏的设计与编写。该游戏程序实现了用方向键控制蛇在围墙内爬行、随机产生食物、控制蛇吃掉食物并加分、控制游戏中闯关的数目和蛇的运行速度、显示得分和关数、结束处理等功能。该游戏程序具有界面友好、操作方便、控制准确和容错能力强等特点。

然而这个贪吃蛇游戏还存在一些缺陷，应该在如下几个方面加以改进。

1. 进入游戏后，到蛇死亡，只能玩一次就会退出程序。应加入多次游戏的控制能力。
2. 蛇的美观不足，需将蛇头、蛇尾及蛇身做进一步美化，使其更像真实的蛇。

贪吃蛇小游戏的设计与编写对我们有很多帮助，学习编写贪吃蛇的游戏对掌握前端语言知识有很大的帮助。通过编程实践，还能拓展思路，让我们去寻找需要调用哪些函数，怎样提高程序的质量等。

要写出好的程序，需要我们有扎实的基础，这样遇到一些基本算法的时候就会游刃有余了。在编程时我们要有丰富的想象力，不要拘泥于固定的思维方式，遇到问题的时候要多想几种解决问题的方案。丰富的想象力是建立在丰富的知识基础上，所以我们要通过多个途径来帮助自己建立较丰富的知识结构。