

# 模仿贪吃蛇模式的英语造句游戏的实现<sup>①</sup>

李家<sup>1</sup>, 郑磊<sup>1</sup>, 杜怀英<sup>2</sup>, 王雪<sup>3</sup>

<sup>1</sup>(辽宁师范大学 城市与环境学院, 大连 116029)

<sup>2</sup>(大连英语岛培训学校, 大连 116033)

<sup>3</sup>(国家海洋环境监测中心, 大连 116023)

**摘要:** 在少儿英语教学中, 连词造句是一种重要的教学形式。为提高学生的学习兴趣, 模仿贪吃蛇游戏, 采用 JavaScript 脚本语言, 开发了基于 web 的吃单词造句的教学游戏, 取得了较好的教学效果。

**关键词:** 贪吃蛇游戏; 英语造句; JavaScript

## Implementation of Snake Game for Making English Sentences

LI Jia<sup>1</sup>, ZHENG Lei<sup>1</sup>, DU Huai-Ying<sup>2</sup>, WANG Xue<sup>3</sup>

<sup>1</sup>(City and Environment College, Liaoning Normal University, Dalian 116029, China)

<sup>2</sup>(Dalian English Island Training School, Dalian 116033, China)

<sup>3</sup>(National Maritime Environment Monitoring Center, Dalian 116023, China)

**Abstract:** Making sentence using given words is an important approach in juvenile English teaching. To increase student interest in learning, this paper develops an English teaching game based web using JavaScript. It imitates the snake game and it results in better teaching effect.

**Key words:** snake game; make English sentence; JavaScript

传统的英语教学大多是“粉笔加黑板”式的教学, 教学方式呆板, 教学内容枯燥。尤其对于小学生, 他们的思维方式大多是感性思维, 对于抽象的语法往往产生畏惧感。针对小学生好动、好奇、富有好胜心的特点, 在小学英语课堂中开展游戏教学, 在“玩中学”, 有助于激发学生的学习兴趣<sup>[1]</sup>。

在少儿英语教学中, 连词造句是一种重要的教学形式。为增加连词造句的趣味性, 模仿常见的贪吃蛇游戏, 设计并实现了连词造句游戏。为了将游戏运行在基于 web 的网络教学环境中网站上, 采用了 JavaScript 脚本语言编码实现。

## 1 确定游戏规则

### 1.1 传统贪吃蛇规则

传统贪吃蛇游戏是基于网格的。在游戏的起始画面上只有一节蛇和一个食物, 随机定位在游戏网格中。按下方向键后, 蛇沿着该方向移动, 一次前进一个网

格。移动时遇到食物可以吃掉, 将其变为蛇身上的一节, 系统同时再生成一个食物, 随机定位在网格中。玩家继续选择方向键去吃食物, 在此过程中蛇身逐渐变长。在移动中如果蛇触到网格边界或撞到自身, 则游戏结束, 游戏得分就是所吃的食物数量。

### 1.2 连词造句游戏规则

英语连词造句游戏模仿传统贪吃蛇游戏, 将一个英语句子的所有单词和蛇头同时随机定位到游戏网格中, 单词的一个字母占一个网格, 单词在游戏网格中水平放置, 单词之间不相连。按下方向键后, 蛇头沿该方向移动; 遇到单词则吃掉, 将其作为蛇节显示在蛇身上; 先吃的单词挨着蛇头, 后吃的单词靠近蛇尾。吃单词时要按照单词在句子中顺序来吃, 吃错单词则游戏结束, 得分就是所吃单词的数量。在移动中如果蛇触到网格边界或撞到自身, 则游戏也结束。

将传统贪吃蛇游戏改造成为连词造句游戏时, 在规则上主要有以下改动:

① 收稿时间:2011-01-29;收到修改稿时间:2011-03-14

1) 一次性创建食物, 将一个句子的多个单词同时随机分布到网格中, 玩家要按单词在句子中的顺序选择单词吃掉;

2) 一个单词在网格的一行中, 蛇沿垂直方向移动时碰到单词的任一字母都可以吃掉这个单词;

3) 吃掉的单词要附加到蛇身的后面, 而不是像传统贪吃蛇游戏那样, 吃食时将食物变成蛇身最前面一节即可。

## 2 算法设计及实现

### 2.1 游戏初始化

在页面 body 的 onload 事件中初始化游戏, 包括创建游戏网格, 创建蛇头并为蛇头随机定位、创建食物等 (图 1)。

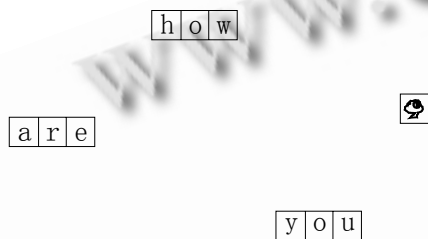


图 1 英语造句游戏初始画面

#### 2.1.1 创建游戏网格

游戏网格采用二维数组实现, 采用类似游程编码的方式, 利用行下标和列下标隐含每个网格的位置, 并不是在页面上显示网格。游戏中用数组元素的值标识每个单元格, 可取的值为‘F’、‘S’或空字符串, F 表示食物, S 表示蛇身。创建游戏网格的代码为:

```
MainMap = [];
for(var i = 0; i < Rows; i++)
{
    MainMap[i] = [];
    for(var j = 0; j < Cols; j++)
    {
        MainMap[i][j] = "";
    }
}
```

#### 2.1.2 创建蛇身数组并为蛇头随机定位

页面中除蛇身外无其它 span 标签, 所以用 span 数组存储蛇身。利用 span 的 background-image 属性, 用不同的图片表示蛇身, 包括蛇头、英文字母、空格、

标点符号等图片。用以下代码建立蛇身数组:

```
AllSnakes = gameFrame.all.tags('span');
```

增加一节蛇身的函数为:

```
function CreateSnake(charName)
{
    //判断是否出界
    if(SnakeX >= Cols || SnakeX < 0 || SnakeY >= Rows
    || SnakeY < 0)
    {
        GameOver = true;
        return;
    }
    //在蛇身数组中加入一个带图片的 span
    gameFrame.insertAdjacentHTML("beforeEnd" ,
    "<span x="+SnakeX+" y="+SnakeY
    +"style='position:absolute;left:"+SnakeX*CellW+";t
    op:"+SnakeY*CellH+";width:"+CellW+";height:"+CellH
    +"; background-image: url("+charName+");' ></span>");
    //在网格上标记该位置为蛇
    MainMap[SnakeY][SnakeX] = 'S';
}
```

代码中 charName 为新建的图片名称, SnakeX、SnakeY 为新建的这节蛇身的位置, 即游戏网格二维数组的列下标和行下标, span 用 style 属性的 left 和 top 定位, 同时用自定义属性 x、y 记录其坐标值, 用于后续处理。初始化时要随机定位蛇头, 给出 SnakeX、SnakeY 的值, 然后调用该函数显示蛇头。

#### 2.1.3 创建食物单词

页面中除食物外无其它 div 标签, 所以将食物用 div 数组存储。利用 div 的 background-image 属性, 用图片表示食物单词, 单词的每个字母用一张图片表示, 占用食物数组中的一个 div。用以下代码建立食物数组:

```
AllFoods = gameFrame.all.tags('div');
创建食物数组函数的主要代码为:
for (var i=0;i<words.length;i++)
{
    //为单词随机定位
    var xy=randLocation(words[i]);
    var x0=xy[0];
    var y0=xy[1];
```

```

// 一个单词的在纵向上的影响范围是从
//y0-words[i].length 到 y0+words[i].length
for(var y=y0-words[i].length;y<y0+words[i].length;y
++)
{
for (var j=0;j<words[i].length;j++)
{
//确定单词中第 j 个字母的列下标
var x=x0+j;
if (y==y0) //仅在 y0 这一行显示单词
{
//该单词中第 j 个字母
var c=words[i].charAt(j);
//将带有第 j 个字母的 div 插入到页面中的由
//x*CellW 和 y*CellH 指定的位置上
gameFrame.insertAdjacentHTML("beforeEnd","<div
x="+x+"y="+y+"word="+words[i]+"wordSequence="+
i+"No="+j+" style='display:block;position:absolute;left:"
+x*CellW+";top:"+y*CellH+";width:"+CellW+";height:"
+CellH+";background-image:url(images/"+c+".jpg);'></d
iv>");
//将该网格标记为食物
MainMap[y][x] = "F";
}
Else
//y0 以外的行中的网格标记为范围
MainMap[y][x] = "E";
} } }

```

该函数对句子中的每一个单词在网格中随机定位。定位时，一个单词的影响范围如图 2 所示。图中‘E’表示单词 hello 的影响范围，在此范围内不能有其它单词。这是因为蛇在吃食物时，一次增加一个单词的长度。沿纵向移动吃食物时，在此范围内如有其它单词，会出现蛇穿单词而过的现象。创建食物时将单词所在单元格标为‘F’，将单词的影响范围标记为‘E’，据此标记对单词随机定位。代码

中为 div 增加了 word、wordSequence、No 等自定义属性，用于后续处理。word 属性表示该字母所属的单词，No 属性表示该字母是所属单词的第几个字母，wordSequence 属性表示该字母所属单词是句子中的第几个单词。

E	E	E	E	E
E	E	E	E	E
E	E	E	E	E
E	E	E	E	E
h	e	l	l	o
E	E	E	E	E
E	E	E	E	E
E	E	E	E	E
E	E	E	E	E

图 2 一个单词的影响范围

## 2.2 蛇的移动和吃食物

游戏在按下光标键（上，下，左，右）后开始，系统根据按键确定前进方向，同时启动定时器：

```
setInterval(Move, sInterval);
```

这样每经过一个时间间隔 sInterval，系统就调用 move 函数一次，蛇身前进一个单元格。蛇身前进的代码如下：

```

if(MainMap[SnakeY][SnakeX] == ")
{
//将网格中原蛇尾处单元格的值清空
MainMap[AllSnakes[0].y][AllSnakes[0].x] = ";
//标记蛇头前的网格为蛇，同时在 AllSnakes 数组
中增加一个 span，临时放置一个空格
CreateSnake("images/space.jpg");
//蛇身数组中后面的图象向前串
for(var i=AllSnakes.length-1;i>0;i--)AllSnakes[i].sty
le.backgroundImage=AllSnakes[i-1].style.backgroundIm
age;
//删除原来的蛇尾
AllSnakes[0].removeNode(true);
}

```

游戏中将蛇身数组中单词和字母反序存放（图 3），即将蛇头和先吃的单词放在数组的后面，将后吃的单词放在数组的前面。所以蛇的前进是这样实现的：

1) 利用蛇身 span 上的自定义属性 x、y，将原蛇尾处的单元格的标记清空；

2) 调用 CreateSnake 函数，将蛇头前的单元格的标记为蛇，同时在该位置创建一个 span，加入一个空格图片，自动附加到蛇身数组的最后；

3) 将蛇身数组中每个 span 的 backgroundImage 的值向后串，则蛇头的图片串到了最后增加的那个 span 上，每个字母都前进了一个单元格；

## 4) 删除原来的蛇尾。

图片	u	o	y		e	r	a		w	o	h	蛇头
下标	0	1	2	3	4	5	6	7	8	9	10	11

图 3 句子 how are you 在蛇身数组中的存储

移动时吃食物的代码如下:

```

if(MainMap[SnakeY][SnakeX] == 'F')
{
    for (var i=0; i<AllFoods.length; i++)
    {
        //根据自定义属性 x、y，找到该字母
        if(AllFoods[i].x==SnakeX&&AllFoods[i].y
==SnakeY)
        { //确定所吃单词范围
            for(vark=i-AllFoods[i].No;k<i-AllFoods[i].No+AllFoods[i].word.length;k++){MainMap[AllFoods[k].y][AllFoods[k].x] = "";
                //应用正则表达式，取得食物图片（即字母）的名称
                varimage=AllFoods[k].style.backgroundImage.replace(url\\(\\)/g, "");
                //将该图片加入到蛇身数组的最后的 span 上
                CreateSnake(image);
                //changeImage();
                vartemp=AllSnakes[AllSnakes.length-1].style.backgroundImage; //保留新增加的 span 的 background Image
                for (var j=AllSnakes.length-1; j>0;j--)
                //前面的图像向后串
                AllSnakes[j].style.backgroundImage=AllSnakes[j-1].style.backgroundImage;
                //新增加的图片放在数组的最前面，保证数组是倒序的
                AllSnakes[0].style.backgroundImage= temp;
                //根据前进方向确定蛇头的下一个位置
                SnakeX = SnakeX + GoX;
                SnakeY = SnakeY + GoY;
                if(k==i-AllFoods[i].No+ AllFoods[i]. word.length-1)
                {

```

```

CreateSnake("images/space.jpg");
changeImage();
break;
} } break; } } }

```

代码中主要步骤如下:

1) 对食物数组做循环，利用 div 的自定义属性 x、y，根据坐标找到蛇头前的那个字母所在的 div;

2) 利用 div 的自定义属性 No (所属单词的第几个字母)，确定所吃单词在食物数组中的范围。无论蛇头前是该单词的第几个字母，i-AllFoods[i].No 为该单词的第 0 个字母在食物数组中的下标，i-AllFoods[i].No+AllFoods[i].word.length-1 为该单词的最后一个字母在食物数组中的下标。

3) 对该单词的每一个字母，调用 CreateSnake 函数，将蛇头前的单元格的标记为蛇，同时在该位置创建一个 span，加入该字母的图片，自动附加到蛇身数组中的最后;

4) 把蛇身数组最后一个 span 上的 background Image 保存到 temp 中; 将蛇身数组中每个 span 的 backgroundImage 的值向后串，则蛇头的图片串到了最后增加的那个 span 上，每个字母都前进了一个单元格; 将保存在 temp 中的新增加的图片赋给数组中第 0 个 span，保证数组是倒序的;

5) 根据前进方向确定蛇头的下一个位置，回到第 3)步，循环处理下一个字母;

6) 在最后一个字母之后加上空格。

### 3 结语

以上介绍了基于贪吃蛇模式实现英语造句游戏的主要算法和代码。游戏创意新颖，比较巧妙地将英语连词造句和贪吃蛇游戏结合起来; 算法设计和实现比较合理，充分利用了 JavaScript 语言的特性和 div、span 等页面标签的属性特性; 为相关软件的开发提供了借鉴。

### 参考文献

- 张翠云.科学有效地开展小学英语游戏教学.山东师范大学外国语学院学报(基础英语教育),2006,(4):61-63.
- 蒋民.JavaScript 实例、技巧与应用.北京:清华大学出版社,2003.