**数据结构课程设计**

**班级：1615403**

**学号：161540205**

**姓名：张佳钰**

**指导老师：孙涵**

9.贪吃蛇小游戏

采用的数据结构：

静态数组 队列

算法思想：

通过js的setInterval函数 每隔一段时间执行一次函数 动态生成食物

food还有移动蛇身

移动蛇：

一个数组记录蛇身 不断地执行入队出队操作

源程序：

函数主要写在 snake.js中

/// <reference path="jquery.min.js" />

var width=0;

var height=0;

var hard=0;

var flag=parseInt(0);

var a;

var snake\_head;

var snake\_tail;

var snake\_body;

var snake\_direction;

var on\_going=0;

var start;

var direction;

var visit;

var all\_blank;

var food\_num;

var all\_food\_num=10;//设置总的食物的数量

var food=new Array(all\_food\_num);

food.fill(-1);

var show;

var key=new Array();

console.log(a);

function init\_visit(visit,flag)

{

visit=new Array();

for(var i=0;i<=(height+1);i++)

{

visit[i]=new Array();

for(var j=0;j<=(width+1);j++)

{

visit[i][j]=flag;

}

}

return visit;

}

function pick\_start\_point()

{

var flag=1;

var i=0;

flag=Math.floor(Math.random()\*4);

var j;

var x,y;

y=Math.floor(Math.random()\*(width-4))+4;

x=Math.floor(Math.random()\*(height-4))+4;

start\_point={x:x,y:y,direction:flag};

cur=start\_point;

return start\_point;

}

function point\_to\_count(i,j)

{

return (i-1)\*(width)+j;

}

function get\_food()

{

var y=Math.floor(Math.random()\*(width))+1;

var x=Math.floor(Math.random()\*(height))+1;

//console.log(width,height,x,y);

while((snake\_body.findIndex(n=>n==point\_to\_count(x,y))!=-1)&&on\_going==1)

{

y=Math.floor(Math.random()\*(width))+1;

x=Math.floor(Math.random()\*(height))+1;

}

return {x:x,y:y};

}

function show\_place(num,type="normal")

{

if(type=="food")

{

$(`input[id=${num}]`).addClass("food");

$(`input[id=${num}]`).removeClass("default");

}

else if(type=="get\_food")

{

$(`input[id=${num}]`).removeClass("food");

$(`input[id=${num}]`).addClass("walk");

}

else

{

$(`input[id=${num}]`).addClass("walk");

$(`input[id=${num}]`).removeClass("default");

}

}

function back\_place(num,type="normal")

{

if(type=="food")

{

$(`input[id=${num}]`).removeClass("food");

$(`input[id=${num}]`).addClass("default");

}

else

{

$(`input[id=${num}]`).removeClass("walk");

$(`input[id=${num}]`).addClass("default");

}

}

function key\_event()

{

var p;

var x=[-1,0,1,0];

var y=[0,1,0,-1];

console.log(snake\_body);

show = setInterval(function(){

console.log('pp');

if(food\_num<all\_food\_num)

{

var flag=-1;

for(var i=0;i<food\_num;i++)

{

if(food[i]==-1)

{

flag=i;

break;

}

}

var new\_food=get\_food();

food.splice(flag,1,new\_food);

console.log("ne\_food");

console.log(new\_food.x,new\_food.y,point\_to\_count(new\_food.x,new\_food.y),visit[new\_food.x][new\_food.y]);

console.log(visit[new\_food.x]);

for(var j=1;j<13;j++)

console.log(visit[j]);

visit[new\_food.x][new\_food.y]=1;

show\_place(point\_to\_count(new\_food.x,new\_food.y),"food");

food\_num++;

}

var new\_x,new\_y;

new\_x=snake\_head.x+x[snake\_direction];

new\_y=snake\_head.y+y[snake\_direction];

if(new\_x<1||new\_x>height||new\_y<1||new\_y>width)

{

alert("you lose!!");

clearInterval(show);

re\_start();

}

else

{

console.log("visit: ",visit[new\_x][new\_y]);

if(snake\_body.findIndex(x=>x==point\_to\_count(new\_x,new\_y))!=-1)

{

alert("lose");

clearInterval(show);

re\_start();

}

else

{

console.log("vist : new direction: ",new\_x,new\_y,visit[new\_x][new\_y]);

var flag=-1;

for(var i=0;i<all\_food\_num;i++)

{

if(food[i]!=-1)

{

console.log("food",food[i]);

if(food[i].x==new\_x&&food[i].y==new\_y)

{

flag=i;

break;

}

}

}

if(flag!=-1)

{

food.splice(flag,1,-1);

console.log("all\_blank: ",all\_blank);

snake\_body.push(point\_to\_count(new\_x,new\_y));

snake\_head={x:new\_x,y:new\_y};

show\_place(point\_to\_count(snake\_head.x,snake\_head.y),"get\_food");

visit[snake\_head.x][snake\_head.y]=2;

console.log("get\_food");

console.log("visit: ",new\_x,new\_y,visit[new\_x][new\_y]);

food\_num--;

all\_blank--;

}

else

{

snake\_body.push(point\_to\_count(new\_x,new\_y));

snake\_head={x:new\_x,y:new\_y};

var xx,yy;

console.log("back\_place");

visit[new\_x][new\_y]=2;

//onsole.log("00000",snake\_tail.x,snake\_tail.y,visit[snake\_tail.x][snake\_tail.y]);

snake\_tail=snake\_body.shift();

console.log(snake\_tail);

xx=Math.floor((snake\_tail-1)/width)+1;

yy=snake\_tail-width\*(xx-1);

console.log("xx,yy",xx,yy);

//console.log("visit: ",new\_x,new\_y,visit[new\_x][new\_y]);

show\_place(point\_to\_count(snake\_head.x,snake\_head.y));

back\_place(snake\_tail);

//console.log("00000",snake\_tail.x,snake\_tail.y,visit[snake\_tail.x][snake\_tail.y]);

visit[xx][yy]=0;

//console.log("00000",snake\_tail.x,snake\_tail.y,visit[snake\_tail.x][snake\_tail.y]);

}

for(var j=1;j<13;j++)

console.log(visit[j]);

}

}

}, 400\*(all\_blank/(height\*width)));

document.onkeydown=function(event){

var e = event || window.event || arguments.callee.caller.arguments[0];

if(e)

{

visit=init\_visit(visit,0);

visit[start.x][start.y]=1;

//var key=[37,39];

var weight=[-1,1];

console.log(e.keyCode);

for(var i=0;i<key.length;i++)

{

if(e.keyCode==key[i])

{

if(hard==1)

{

snake\_direction=(weight[i]+snake\_direction+4)%4;

}

else

{

snake\_direction=i;

}

console.log("snake\_direction: ",snake\_direction);

return false;

break;

}

}

console.log(e.keyCode);

}

};

}

function make\_matrix(w,h)

{

var text='';

var class\_type;

var count=0;

height=h;

width=w;

visit=init\_visit(visit,0);

console.log(visit);

console.log(height+1,width+2);

all\_blank=height\*width-1;

start=pick\_start\_point();

direction=start.direction;

snake\_direction=start.direction;

snake\_head={x:start.x,y:start.y};

snake\_tail={x:start.x,y:start.y};

snake\_body=new Array();

snake\_body.push(point\_to\_count(snake\_head.x,snake\_head.y));

visit[start.x][start.y]=2;

console.log("start",visit,start.x,start.y);

console.log(start);

class\_type="default";

for(var i=1;i<=height;i++)

{

for(var j=1;j<=width;j++)

{

//class\_type= d[i][j]==0?"default":"block";

count++;

text+=`<input type="button" class="${class\_type}" value="" name="${count}" id="${count}" onmousedown="check\_bomb(this.id)">`;

}

text+=`</br>`;

}

//console.log(text);

$('#bg').append(text);

$(`input[id=${point\_to\_count(start.x,start.y)}]`).removeClass("default");

$(`input[id=${point\_to\_count(start.x,start.y)}]`).addClass("walk");

$('#re\_start').show();

// $('#help').show();

$('#ask\_height\_width').hide();

}

function start\_game()

{

if(on\_going==0)

{

on\_going=1;

food\_num=0;

console.log(visit);

var h=$('#height').val();

var w=$('#width').val();

hard=parseInt($('#hard').val());

if(hard==0)

{

key=[38,39,40,37];

}

else

{

key=[37,39];

}

console.log(h,w);

w=parseInt(w);

h=parseInt(h);

console.log(w+1,h+2);

make\_matrix(w,h);

key\_event();

}

}

function re\_start()

{

width=0;

height=0;

hard=0;

flag=parseInt(0);

snake\_head='';

snake\_tail='';

snake\_body=new Array();

snake\_direction=0;

on\_going=0;

start=0;

direction=0;

visit=new Array();

all\_blank=0;

food\_num=0;

food=new Array(all\_food\_num);

food.fill(-1);

flag=0;

on\_going=0;

$('#bg').empty();

clearInterval(show);

$('#re\_start').hide();

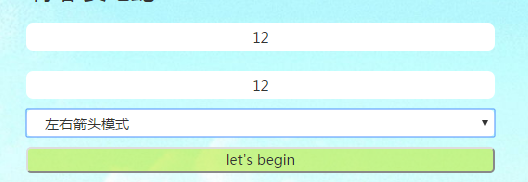
$('#ask\_height\_width').show();

}

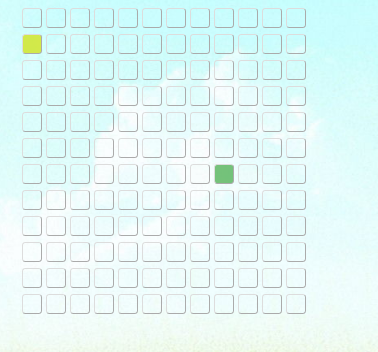
//打开方式 双击index.html(推荐 用 chrome 打开)



输入width height 选择模式 四键模式或者两键模式



点击 let’s begin



就开始了

Js代码行数

337

感想：

在写贪吃蛇的时候 发现自己好像就小时候玩过 决定打开7k7k小游戏玩一会 没一会就败了 但也差不多知道怎么写 然后就开始了 咦 好像还不难 效果也还行