信息科学技术学院实验报告

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **课程名称** | | **高级网页设计** | | | | **班级** | | **17网络工程1班** |
| **实验名称** | | **高级网页设计大作业** | | | | **教导教师** | | **曾少宁** |
| **姓名** | **黄俊杰** | | | **学号** | **1714080903135** | | **日期** | **2019.05.09** |
|  |  | |  | | |  | |  | |

**引言**

题目是贪吃蛇小游戏。大一时用C++语言写过贪吃蛇小游戏。当时便对游戏的开发产生了浓厚的兴趣，但是自己在UI方面的知识却有限，所以只实现了基本的功能。这次学习微信小程序开发，发现可以用CSS弥补UI编程的不足，于是决定用JS语言写贪吃蛇小游戏。主要原理基本相同。对食物的设置采用随机数生成的方法、同时初始化墙壁和蛇的位置。

首页可以查看游戏所获得的历史最高分，同时增加了一个按钮可以跳转到游戏界面。而在snake页面，主要是实现一个贪吃蛇小游戏，通过鼠标点击控制蛇的移动，同时可以随着游戏的进行记录游戏所获得的分数，并记录游戏的历史最高分。

**一、代码实现**

1. **首页：**

**// index.wxml…**

<view class="container">

<view bindtap="bindViewTap" class="userinfo">

<image class="userinfo-avatar" src="{{userInfo.avatarUrl}}" background-size="cover"></image>

<text class="userinfo-nickname">{{userInfo.nickName}}</text>

</view>

<button bindtap='gotoPage1'>开始游戏</button>

<view class="usermotto">

<view class="user-motto">历史最高分</view>

<view class="user-motto">100</view>

</view>

</view>

**页面介绍：首页wxml主要获取用户头像和一个跳转到游戏页面的按钮，同时显示游戏的最高历史分数。**

/**/ index.js …**

//获取应用实例

var app = getApp()

Page({

data: {

motto: 'Hello World',

userInfo: {}

},

//事件处理函数

bindViewTap: function() {

wx.navigateTo({

url: '../logs/logs'

})

},

onLoad: function () {

console.log('onLoad')

var that = this

//调用应用实例的方法获取全局数据

app.getUserInfo(function(userInfo){

//更新数据

that.setData({

userInfo:userInfo

})

})

},

gotoPage1: function () { wx.navigateTo({ url: '/pages/snake/snake'}) }

})

**页面介绍：增加了一个gotoPage1函数，通过点击首页按钮执行函数，可以跳转到snake游戏页面，同时，通过参数值传递显示游戏的**历史最高分

1. **游戏页面snake：**

**<!--snake.wxml-->**

<view class="control" bindtouchstart="tapStart" bindtouchmove="tapMove" bindtouchend="tapEnd">

<view class="score">

<view class="title">snake</view>

<view class="scoredetail">

<view class="scoredesc">得分</view>

<view class="scorenumber">{{score}}</view>

</view>

<view class="scoredetail">

<view class="scoredesc">历史最高</view>

<view class="scorenumber">{{maxscore}}</view>

</view>

</view>

<view class="ground">

<view wx:for="{{ground}}" class="rows" wx:for-item="cols">

<view wx:for="{{cols}}" class="block block\_{{item}}" >

</view>

</view>

</view>

<modal class="modal" hidden="{{modalHidden}}" no-cancel bindconfirm="modalChange">

<view> 游戏结束，重新开始吗？ </view>

</modal>

</view>

**页面介绍：此页面主要是四个模块，即四个view。分别用来记录游戏分数、游戏最高分数、游戏背景墙的范围和在游戏结束时跳转出来的询问是否重新开始游戏的视图**

**<!—snake.js-->**

var app = getApp();

Page({//需要的数据初始化

data:{

score: 0,//比分

maxscore: 0,//最高分

startx: 0,

starty: 0,

endx:0,

endy:0,//以上四个做方向判断来用

ground:[],//存储操场每个方块

rows:28,

cols:22,//操场大小

snake:[],//存蛇

food:[],//存食物

direction:'',//方向

modalHidden: true,

timer:''

} ,

//页面加载完成时便进行游戏的初始化

onLoad:function(){

var maxscore = wx.getStorageSync('maxscore');

if(!maxscore) maxscore = 0

this.setData({

maxscore:maxscore

});

this.initGround(this.data.rows,this.data.cols);//初始化操场

this.initSnake(3);//初始化蛇

this.creatFood();//初始化食物

this.move();//蛇移动

},

//计分器，记录游戏分数，同时将最高游戏分数传给maxscore并在历史最高游戏分数模块显示

storeScore:function(){

if(this.data.maxscore < this.data.score){

this.setData({

maxscore:this.data.score

})

wx.setStorageSync('maxscore', this.data.maxscore)

}

},

//墙壁的初始化，设置蛇可移动的范围

initGround:function(rows,cols){

for(var i=0;i<rows;i++){

var arr=[];

this.data.ground.push(arr);

for(var j=0;j<cols;j++){

this.data.ground[i].push(0);

}

}

},

//初始化蛇

initSnake:function(len){

for(var i=0;i<len;i++){

this.data.ground[0][i]=1;

this.data.snake.push([0,i]);

}

},

//移动函数

move:function(){

var that=this;

this.data.timer=setInterval(function(){

that.changeDirection(that.data.direction);

that.setData({

ground:that.data.ground

});

},400);

},

/\*事件绑定，用过鼠标拖动蛇来使蛇开始移动，并监听蛇移动方向是否改变，蛇的方向不能向后改变\*/

tapStart: function(event){

this.setData({

startx: event.touches[0].pageX,

starty: event.touches[0].pageY

})

},

tapMove: function(event){

this.setData({

endx: event.touches[0].pageX,

endy: event.touches[0].pageY

})

},

tapEnd: function(event){

var heng = (this.data.endx) ? (this.data.endx - this.data.startx) : 0;

var shu = (this.data.endy) ? (this.data.endy - this.data.starty) : 0;

if(Math.abs(heng) > 5 || Math.abs(shu) > 5){

var direction = (Math.abs(heng) > Math.abs(shu)) ? this.computeDir(1, heng):this.computeDir(0, shu);

switch(direction){

case 'left':

if(this.data.direction=='right')return;

break;

case 'right':

if(this.data.direction=='left')return;

break;

case 'top':

if(this.data.direction=='bottom')return;

break;

case 'bottom':

if(this.data.direction=='top')return;

break;

default:

}

this.setData({

startx:0,

starty:0,

endx:0,

endy:0,

direction:direction

})

}

},

computeDir: function(heng, num){

if(heng) return (num > 0) ? 'right' : 'left';

return (num > 0) ? 'bottom' : 'top';

},

//食物的设置，要注意每次食物的出现要随机同时不能出现在蛇的身上，还有同时不能出现在墙壁外面或墙壁上

creatFood:function(){

var x=Math.floor(Math.random()\*this.data.rows);

var y=Math.floor(Math.random()\*this.data.cols);

var ground= this.data.ground;

ground[x][y]=2;

this.setData({

ground:ground,

food:[x,y]

});

},

//改变方向函数，分为向左、向右、向上、向下或笔直前进，但是需要限制蛇不能向后移动

changeDirection:function(dir){

switch(dir){

case 'left':

return this.changeLeft();

break;

case 'right':

return this.changeRight();

break;

case 'top':

return this.changeTop();

break;

case 'bottom':

return this.changeBottom();

break;

default:

}

},

changeLeft:function(){

var arr=this.data.snake;

var len=this.data.snake.length;

var snakeHEAD=arr[len-1][1];

var snakeTAIL=arr[0];

var ground=this.data.ground;

ground[snakeTAIL[0]][snakeTAIL[1]]=0;

for(var i=0;i<len-1;i++){

arr[i]=arr[i+1];

};

var x=arr[len-1][0];

var y=arr[len-1][1]-1;

arr[len-1]=[x,y];

this.checkGame(snakeTAIL);

for(var i=1;i<len;i++){

ground[arr[i][0]][arr[i][1]]=1;

}

this.setData({

ground:ground,

snake:arr

});

return true;

},

changeRight:function(){

var arr=this.data.snake;

var len=this.data.snake.length;

var snakeHEAD=arr[len-1][1];

var snakeTAIL=arr[0];

var ground=this.data.ground;

ground[snakeTAIL[0]][snakeTAIL[1]]=0;

for(var i=0;i<len-1;i++){

arr[i]=arr[i+1];

};

var x=arr[len-1][0];

var y=arr[len-1][1]+1;

arr[len-1]=[x,y];

this.checkGame(snakeTAIL);

for(var i=1;i<len;i++){

ground[arr[i][0]][arr[i][1]]=1;

}

this.setData({

ground:ground,

snake:arr

});

return true;

},

changeTop:function(){

var arr=this.data.snake;

var len=this.data.snake.length;

var snakeHEAD=arr[len-1][1];

var snakeTAIL=arr[0];

var ground=this.data.ground;

ground[snakeTAIL[0]][snakeTAIL[1]]=0;

for(var i=0;i<len-1;i++){

arr[i]=arr[i+1];

};

var x=arr[len-1][0]-1;

var y=arr[len-1][1];

arr[len-1]=[x,y];

this.checkGame(snakeTAIL);

for(var i=1;i<len;i++){

ground[arr[i][0]][arr[i][1]]=1;

}

this.setData({

ground:ground,

snake:arr

});

return true;

},

changeBottom:function(){

var arr=this.data.snake;

var len=this.data.snake.length;

var snakeHEAD=arr[len-1];

var snakeTAIL=arr[0];

var ground=this.data.ground;

ground[snakeTAIL[0]][snakeTAIL[1]]=0;

for(var i=0;i<len-1;i++){

arr[i]=arr[i+1];

};

var x=arr[len-1][0]+1;

var y=arr[len-1][1];

arr[len-1]=[x,y];

this.checkGame(snakeTAIL);

for(var i=1;i<len;i++){

ground[arr[i][0]][arr[i][1]]=1;

}

this.setData({

ground:ground,

snake:arr

});

return true;

},

/\*判断游戏结束函数，如果蛇头与蛇身重合，或者与墙壁重合，那么游戏结束，计算总分数，如果大于历史分数则记录进最高历史分数，跳出重新开始模块\*/

checkGame:function(snakeTAIL){

var arr=this.data.snake;

var len=this.data.snake.length;

var snakeHEAD=arr[len-1];

//如果蛇头与蛇身重合，或者与墙壁重合，那么游戏结束。跳出重新开始界面

if(snakeHEAD[0]<0||snakeHEAD[0]>=this.data.rows||snakeHEAD[1]>=this.data.cols||snakeHEAD[1]<0){

clearInterval(this.data.timer);

this.setData({

modalHidden: false,

})

}

for(var i=0;i<len-1;i++){

if(arr[i][0]==snakeHEAD[0]&&arr[i][1]==snakeHEAD[1]){

clearInterval(this.data.timer);

this.setData({

modalHidden: false,

})

}

}

//如果蛇成功吃到食物，那么蛇的长度+1，重新随机设置食物，分数+10，并重新绘制墙壁、蛇和食物的位置

if(snakeHEAD[0]==this.data.food[0]&&snakeHEAD[1]==this.data.food[1]){

arr.unshift(snakeTAIL);

this.setData({

score:this.data.score+10

});

this.storeScore();

this.creatFood();

}

},

modalChange:function(){

this.setData({

score: 0,

ground:[],

snake:[],

food:[],

modalHidden: true,

direction:''

})

this.onLoad();

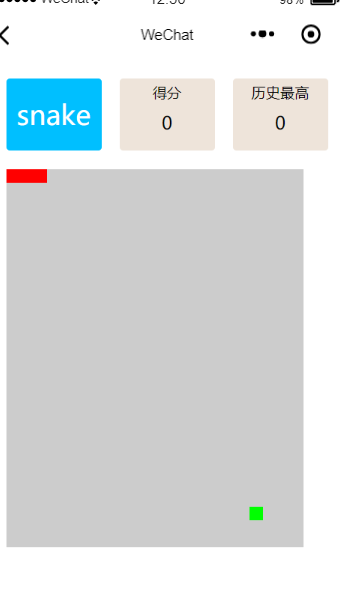
}

});

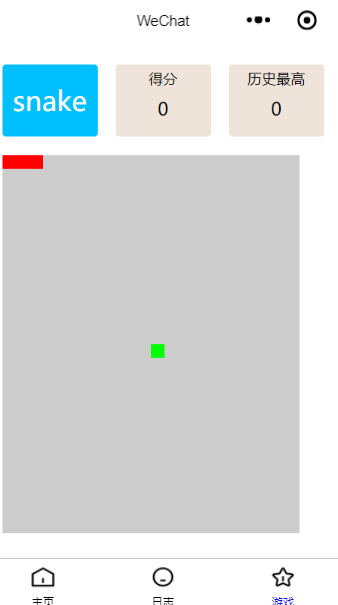
**页面介绍：这是最重要的JS代码，有许多函数实现蛇的移动、食物设置、墙壁和蛇的初始化等功能。主要有监听蛇移动方向是否改变的函数，游戏背景和蛇和食物的初始化，通过鼠标拖动蛇的移动来改变方向。代码的解释穿插注释在代码中**

**二、运行结果截图**

1. **页面一：**

** ****点击开始游戏按钮可以跳转到游戏界面**

1. 首页截图
2. **页面二：**

** 图2.**游戏截图

**鼠标长按左键移动鼠标可以改变蛇的方向**

**三、实验完成情况**

表1：实验完成情况汇总表

|  |  |  |  |
| --- | --- | --- | --- |
| 实验次数 | 实验名称 | 是否通过 | 备注 |
| 1 | Web服务器与调试工具 | Merged+ Late | 迟交 |
| 2 | HTML5语义标签 | Merged + Late | 迟交 |
| 3 | CSS3样式设计 | Merged + Late | 迟交 |
| 4 | JavaScript编程 | Merged | 按时完成 |
| 5 | HTML5移动端页面 |  | 未完成 |
| 6 | HTML5小程序页面 | Merged | 按时完成 |
| 7 | 微信小程序API用法 |  | 未完成 |

**总结**

难。主要是自己什么都不懂，瞎操作，误操作，然后要撤回之前的操作，学习GitHub的使用，学习CSS、JS、HTML，学得不牢固，然后又要学习小程序开发，什么都不会，什么都要去查，然后浪费了许多时间，做的不好而且还迟交。

