游戏架构：

此游戏架构大概分为三层：



Game.html：

说明：包含了界面的展示，以及一些事件的入口。

完整代码：

<html>

<head>

<title>html5 snake game</title>

<script src ="game.js" ></script>

</head>

<body>

<table>

<tr>

<td>

<!--

游戏面板

-->

<canvas id="cvsPnl" style="top:0px;border:5px solid;color:#FF9900" width="10" height="10" >你的浏览器不支持这个游戏！</canvas>

</td>

<td valign="top">

<!--

配置面板

-->

<div id="divSpeed">

<canvas id="cvsSpeed" width="120px" height="55px" ></canvas>

<select id="selSpeed">

<option value="1000">1000</option>

<option value="800">800</option>

<option value="500">500</option>

<option value="300">300</option>

<option value="200">200</option>

</select>

</div>

<div id="divGameCtl">

<input id="btnStartGame" type="button" value="开始"></input>&nbsp;&nbsp;

<input id="btnPauseGame" type="button" value="暂停" ></input>

<br />

</div>

<br />

<div id="divPlayerName">

<canvas id="cvsName" width="50px" height="52px"></canvas><input type="text" id="txtName" style="width:120px;"></input>

<input id="btnRememberMe" value="记住我" type="button"></input>

</div>

<div>

<canvas id="cvsScore" ></canvas>

</div>

</td>

</tr>

<table>

<div id="debug"></div>

<script type="text/javascript">

////游戏画板

var canvasGamePnl = document.getElementById("cvsPnl");

var context = canvasGamePnl.getContext("2d");

////分数

var canvasScore = document.getElementById("cvsScore");

var contxtScore = canvasScore.getContext("2d");

////速度

var canvasSpeed = document.getElementById("cvsSpeed");

var contxtSpeed = canvasSpeed.getContext("2d");

////姓名

var canvasName = document.getElementById("cvsName");

var contxtName = canvasName.getContext("2d");

////开始

$("btnStartGame").onclick=function(){

GameStart(context,contxtScore);

}

////暂停/继续

$("btnPauseGame").onclick = function(){

if($("btnPauseGame").value == "继续"){

RunGame(context,contxtScore);

$("btnPauseGame").value = "暂停";

gameStatus = 1;

}

else{

PauseGame();

$("btnPauseGame").value = "继续";

gameStatus = 2;

}

}

////加载

window.onload=function(){

$("cvsPnl").width =screenWidth;

$("cvsPnl").height=screenHeight;

$("selSpeed").selectedIndex = 0;

DrawFont(contxtSpeed,"选择游戏速度",120);

DrawFont(contxtName,"姓名",50);

$("txtName").value = GetPlayerName();

}

////键盘事件handler

document.onkeydown = function (){

var key = document.all ? event.keyCode : arguments[0].keyCode;

////left

if(key == 37){

if(direction != "right"){

direction = "left";

}

}

////up

else if(key == 38){

if(direction != "down"){

direction ="up";

}

}

////right

else if(key == 39){

if(direction != "left"){

direction = "right";

}

}

////down

else if(key == 40){

if(direction != "up"){

direction = "down";

}

}

}

////设置速度

$("selSpeed").onchange=function(){

if(gameStatus != 3){

sleepTime = parseInt(GetSelectObj("selSpeed").value);

}

}

/////记住我

$("btnRememberMe").onclick = function(){

ScorePlayerName($("txtName").value);

alert("已保存");

}

////绘出文字

function DrawFont(context,txt,size){

context.font='30px impact';

context.fillStyle=fontColor;

context.textAlign='left';

// context.shadowColor="#00ff00";

// context.shadowOffsetX = 15;

// context.shadowOffsetY=-10;

context.fillText(txt,0,50,size);

}

</script>

</body>

</html>

Game.js:

说明：包含了游戏的主干逻辑，业务逻辑层的实现。

完整代码：

document.write("<script language='javascript' src='config.js'></script>");

document.write("<script language='javascript' src='utility.js'></script>");

document.write("<script language='javascript' src='player.js'></script>");

document.write("<script language='javascript' src='global.js'></script>");

////////////////////////

////游戏入口////////////

////////////////////////

function GameStart(context,contxtScore){

InitGame(context,contxtScore);

RunGame(context,contxtScore);

}

////////////////////////

////初始化游戏////////////

////////////////////////

function InitGame(context,contxtScore){

////贪吃蛇

for(var i = initSize ;i > 0;i --){

snakeArr[i - 1] = new Object();

snakeArr[i - 1].x = (initSize - i + 1) \* unitSize;

snakeArr[i - 1].y = 0;

}

////方向

direction = "right";

////食物

food = new Object();

////绘制屏幕方格

DrawScreen(context);

////随即食物

RandomFood(context);

////分数

DrawScore(contxtScore,score);

}

////////////////////////

////运行游戏////////////

////////////////////////

function RunGame(context,contxtScore){

if(timer){

clearInterval(timer);

}

timer = setInterval(function(){

if(IsGameOver()){

alert("game over!");

clearInterval(timer);

return;

}

EatFoodHandler(context,contxtScore);

Refresh(context);

SetPosition();

DrawSnake(context);

},sleepTime);

}

////////////////////////

////刷新////////////////

////////////////////////

function Refresh(context){

FillRect(context,snakeArr[snakeArr.length - 1].x,snakeArr[snakeArr.length - 1].y,unitSize,unitSize,screenColor);

DrawRect(context,snakeArr[snakeArr.length - 1].x,snakeArr[snakeArr.length - 1].y,unitSize,unitSize,lineColor);

}

////////////////////////

////画蛇身//////////////

////////////////////////

function DrawSnake(context){

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

FillRect(context,snakeArr[i].x,snakeArr[i].y,unitSize,unitSize,snakeColor);

DrawRect(context,snakeArr[i].x,snakeArr[i].y,unitSize,unitSize,lineColor);

}

}

////////////////////////

////画屏幕//////////////

////////////////////////

function DrawScreen(context){

for(var i = screenLeft;i < screenLeft + screenWidth / unitSize;i ++){

for(var j = screenTop;j < screenTop + screenHeight / unitSize;j ++){

FillRect(context,i \* unitSize,j \* unitSize,unitSize,unitSize,screenColor);

DrawRect(context,i \* unitSize,j \* unitSize,unitSize,unitSize,lineColor);

}

}

}

////////////////////////

////设置坐标////////////

////////////////////////

function SetPosition(){

for(var i = snakeArr.length - 2;i >= 0 ;i --){

snakeArr[i + 1].x = snakeArr[i].x;

snakeArr[i + 1].y = snakeArr[i].y;

}

if(direction == "left"){

snakeArr[0].x -= unitSize;

}

else if(direction == "right"){

snakeArr[0].x += unitSize;

}

else if(direction == "up"){

snakeArr[0].y -= unitSize;

}

else if(direction == "down"){

snakeArr[0].y += unitSize;

}

}

////////////////////////

////判断是否结束游戏///

////////////////////////

function IsGameOver(){

if(snakeArr[0].x < 0 ||snakeArr[0].x > screenWidth){

return true;

}

if(snakeArr[0].y < 0 ||snakeArr[0].y > screenHeight){

return true;

}

for(var i = 1;i < snakeArr.length;i ++){

if(snakeArr[0].x == snakeArr[i].x && snakeArr[0].y == snakeArr[i].y){

gameStatus = 3;

return true;

}

}

return false;

}

////////////////////////

////随即食物////////////

////////////////////////

function RandomFood(context){

food.x = GetRandom((screenWidth / unitSize) - 1) \* unitSize;

food.y = GetRandom((screenHeight / unitSize) - 1) \* unitSize;

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

if(food.x == snakeArr[i].x && food.y == snakeArr[i].y){

RandomFood(context);

}

}

FillRect(context,food.x ,food.y ,unitSize,unitSize,foodColor);////utility.js

DrawRect(context,food.x ,food.y ,unitSize,unitSize,lineColor);////utility.js

}

////////////////////////

///食物处理/////////////

////////////////////////

function EatFoodHandler(context,contxtScore){

if(direction == "left"){

if((snakeArr[0].x - unitSize == food.x) && snakeArr[0].y == food.y){

IncreaseLen(contxtScore);

ClearFood(context);

RandomFood(context);

}

}

else if(direction == "right"){

if(snakeArr[0].x + unitSize == food.x && snakeArr[0].y == food.y){

IncreaseLen(contxtScore);

ClearFood(context);

RandomFood(context);

}

}

else if(direction == "up"){

if(snakeArr[0].x == food.x && (snakeArr[0].y - unitSize == food.y)){

IncreaseLen(contxtScore);

ClearFood(context);

RandomFood(context);

}

}

else if(direction == "down"){

if(snakeArr[0].x == food.x && snakeArr[0].y + unitSize == food.y){

IncreaseLen(contxtScore);

ClearFood(context);

RandomFood(context);

}

}

}

////////////////////////

////清除食物////////////

////////////////////////

function ClearFood(context){

FillRect(context,food.x,food.y,unitSize,unitSize,screenColor);

DrawRect(context,food.x,food.y,unitSize,unitSize,lineColor);

}

////////////////////////

////增加长度////////////

////////////////////////

function IncreaseLen(contxtScore){

var newObj = new Object();

newObj.x = food.x;

newObj.y = food.y;

snakeArr.unshift(newObj);

////分数增加

IncreaseScore(contxtScore);////player.js

}

////暂停

function PauseGame(){

clearInterval(timer);

}

Utility.js：

说明：提供了一些工具方法，工具层。

完整代码：

////返回客户端对象

function $(clientId){

return document.getElementById(clientId);

}

////获得SELECT选中项

function GetSelectObj(clientId){

var obj = $(clientId);

var index = obj.selectedIndex; // 选中索引

return obj.options[index];

}

////调试使用

function DebugVar(param){

$("debug").innerHTML=param;

}

function DebugTxt(text){

$("debug").innerHTML = text;

}

////画方块

function DrawRect(context,left,right,width,height,color){

//设置填充样式

context.strokeStyle = color;

context.strokeRect(left,right, width, height);

}

////填充方块

function FillRect(context,left,right,width,height,color){

//设置填充样式

context.fillStyle = color;

context.fillRect(left,right,width,height);

}

////绘画文字

function DrawScore(context,txt){

context.moveTo(0,0);

context.clearRect(0,0,500,500);

context.font='60px impact';

context.fillStyle=fontColor;

context.textAlign='center';

// context.shadowColor="#00ff00";

// context.shadowOffsetX = 15;

// context.shadowOffsetY=-10;

txt = "分数:" + txt;

context.fillText(txt,100,100,fontSize);

}

////绘制渐变

function ScreenGradient(context){

var grd = context.createLinearGradient(0,0,screenWidth,screenHeight);

grd.addColorStop(0,"#FFCC00");

grd.addColorStop(1,"#99FFFF");

context.fillStyle = grd;

context.fillRect(0,0,screenWidth,screenHeight);

}

////生产随机数

function GetRandom(n){return Math.floor(Math.random()\*n+1)}

////存储键值对

function addKV(k,v){

localStorage.setItem(k,v);

}

////取得键值对的值

function getV(k){

return localStorage.getItem(k);

}

////获得本地存储的所有值并转化为字符串

function getAllValueToStr(){

var content = "";

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

//key(i)获得相应的键，再用getItem()方法获得对应的值

content += localStorage.key(i)+ " : " + localStorage.getItem(localStorage.key(i)) + "<br />";

}

}

Player.js：

说明：包含了玩家的一些操作，game.js下面一层，属于业务逻辑辅助层。

完整实现：

function GetPlayerName(){

var name = getV("userName");

return name != null ? name : "new player" ;

}

function IncreaseScore(context){

score += 10;

DrawScore(context,score);////utility.js

}

function ScorePlayerName(name){

addKV("userName",name);

}

Config.js：

说明：游戏相关配置以及变量，系统配置层。

完整代码：

////////////////////////

////变量、配置参数//////

////////////////////////

var screenWidth = 800;////屏幕宽度

var screenHeight = 500;////屏幕高度

var unitSize = 20;////单元格大小

var initSize = 3;////初始长度

var screenLeft = 0;////屏幕横起始坐标

var screenTop = 0;////屏幕纵起始坐标

var snakeArr = new Array();////贪吃蛇数组

var food;////食物

var snakeColor = "#009999";////蛇身颜色

var direction;////蛇的方向

var screenColor = "#99CCFF";////屏幕颜色

var lineColor = "#ffffff";////线条颜色（方格）

var lineWidth = 3;////线条宽度

var foodColor = "#FFCC00"////食物颜色

var fontColor = "#996600";////分数颜色

var fontSize = 300;////分数字体大小

var timer;////定时器

var sleepTime = 200;////休眠时间

var level=0;////级别

var score=0;////分数

var currentPlayer;////当前玩家

var gameStatus = 0;////0:未开始 1:运行 2:暂停 3:已结束

global.js：（暂时没有用到，用于扩展），系统全局控制，例如场景绘制