图形学课程设计

Growth of life

成员:王倩倩 2019122135020 13
郑佳丽 2019122135020 27

目录

CONTENTS

1.详细设计

1.详细设计

1.1 系统完整功能描述

1.1.1 描述

在一块固定大小的区域内，游戏玩家通过控制贪吃蛇的移动去吃食物，吃到食物后蛇身体长度加1，分数加1。食物被蛇吃到后立马消失，并再次随机产生。蛇撞到四周墙壁或者自己身体时死亡。

1.1.2 实现目标

1. 蛇的移动、蛇的死亡；
2. 吃食物、产生食物。

1.1.3 按键约定

为提升用户的体验感，我们对操作按键做以下约定：

1. 空格键/enter键控制游戏开始；
2. 上下左右按键分别控制贪吃蛇的四个运动方向。

1.1.4 实现原理

1. 利用canvas画布完成运动场地、食物、贪吃蛇的展示；
2. 利用数组存储贪吃蛇的坐标位置；
3. 利用上、下、左、右键改变贪吃蛇的蛇头坐标；
4. 不断重新绘制页面，给人造成贪吃蛇运动的错觉。

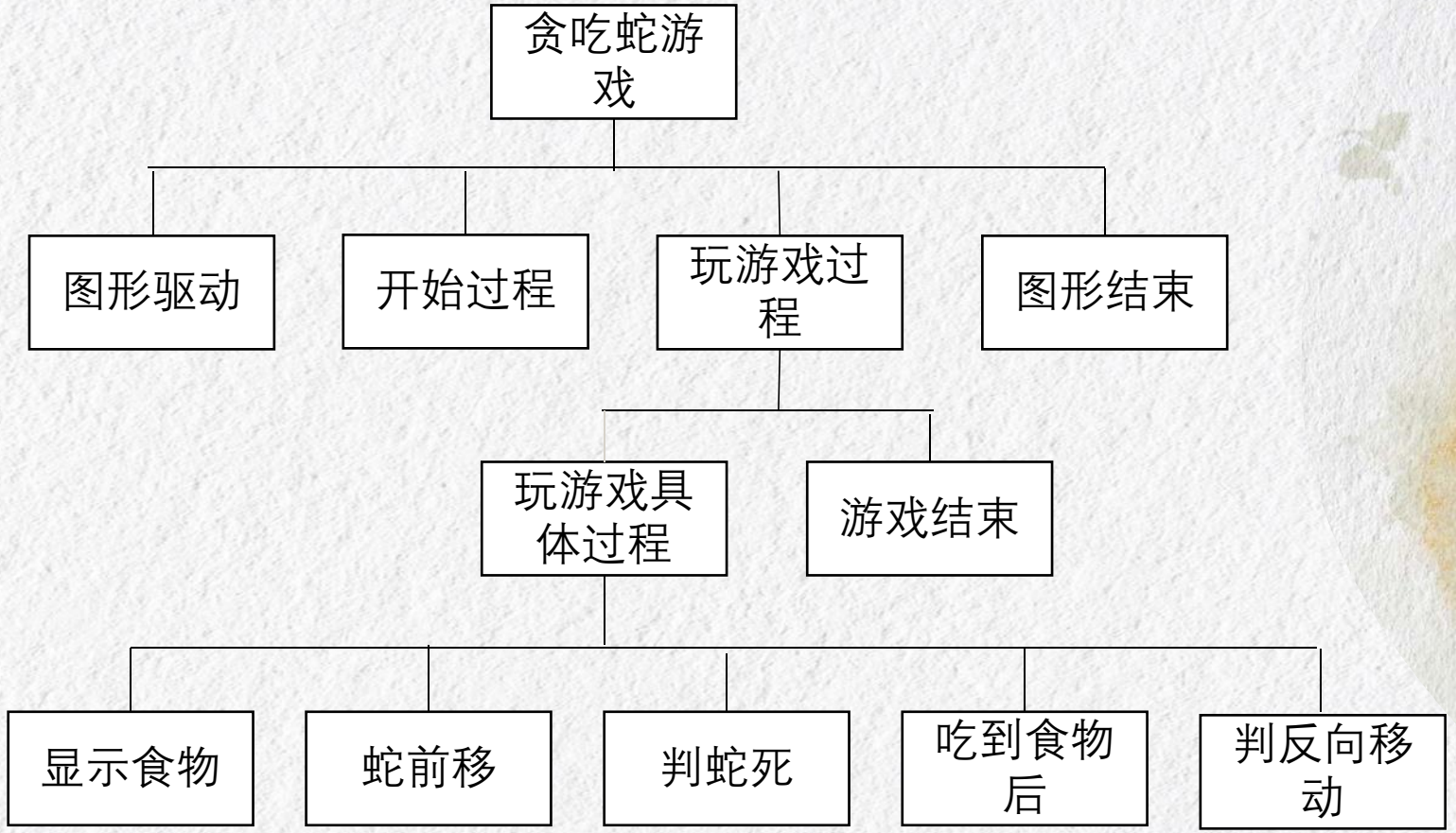
1.详细设计

1.2 各模块流程等说明

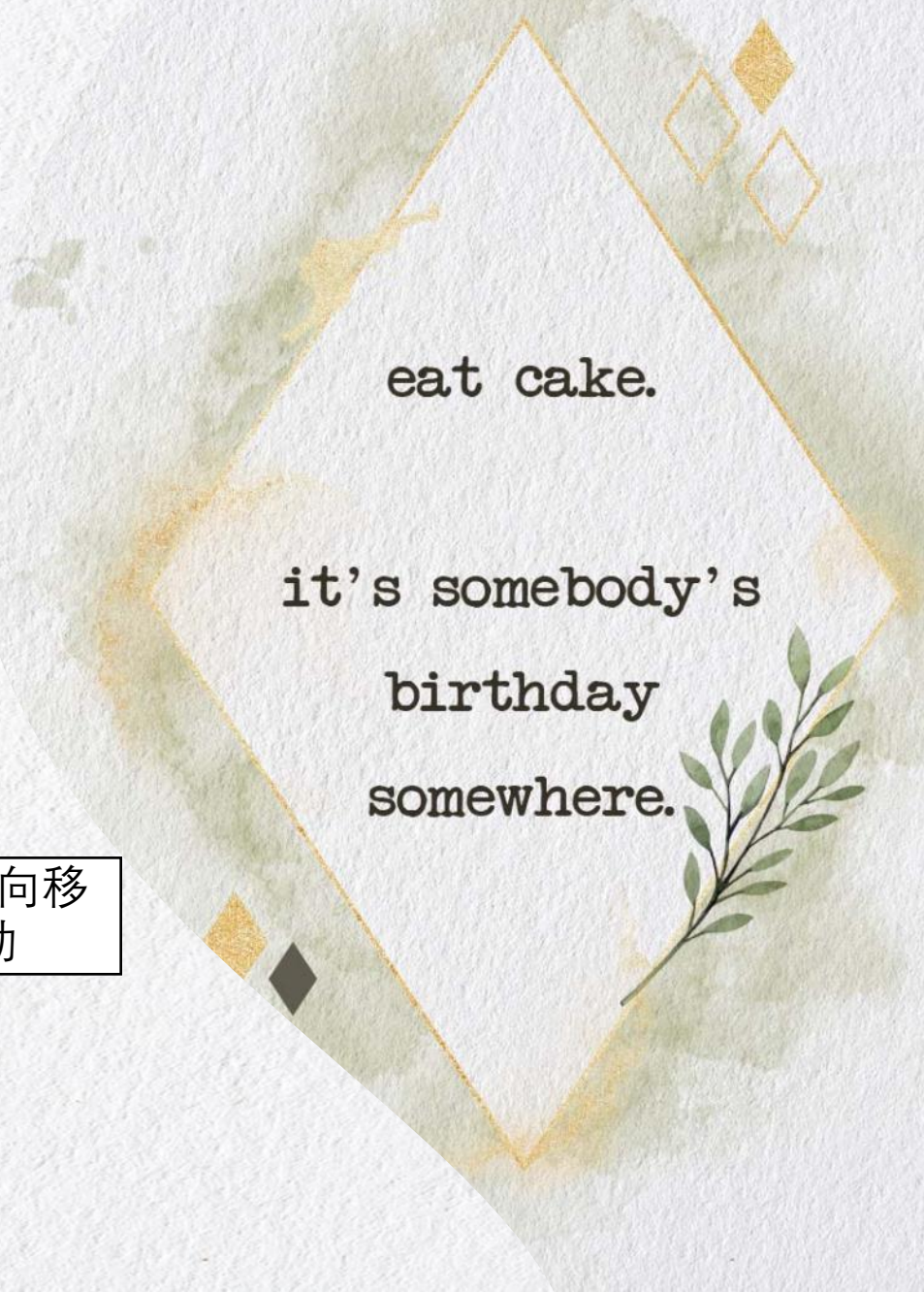
模块应具有高内聚性、低耦合性。这样程序运行才稳定，效率更高。

根据功能将贪吃蛇游戏划分为四个大模块，七个小模块。每个模块均有自己的功能，而且，比较容易画出各模块的流程图。

1.详细设计


贪吃蛇游戏
图形驱动
开始过程
玩游戏过程
图形结束
玩游戏具体过程
游戏结束
显示食物
蛇前移
判蛇死
吃到食物后
判反向移动

模块枝干图


eat cake.
it's somebody's
birthday
somewhere.

1.详细设计

简要说明：

图形驱动：完成初始化图形系统功能；

开始画面模块：根据设置绘制出玩游戏区域四周的墙壁；

玩游戏的过程是此游戏程序的核心，又可细分为好多小的功能模块；

其中，显示食物模块：仅完成显示食物功能；

蛇前移模块：通过算法实现蛇的某方向移动；

判蛇死模块：通过判断蛇是否碰到墙壁或自己来判断蛇是否死亡。

吃到食物后模块：只有蛇吃到食物后才起作用。即当蛇吃到食物后，蛇自身延长一节并绘出蛇。

通过键盘操控来玩游戏主要由判反向移动模块完成，即当蛇不是反向移动时，按照键盘上方向键指示移动。

游戏结束模块：仅完成提示语功能与输出分数。

最后图形结束模块：完成关闭图形功能。

1.详细设计

1.2.1模块划分

1. 图形驱动模块

```
/* 定义事件执行监测 */
function onKeyDown(event) {
        if (status == -1) {
                status = 0;
                food();
                run();
        }
        if (window.event) // IE
        {
                keynum = event.keyCode;
        } else if (event.which) // Netscape/Firefox/Opera
        {
                keynum = event.which;
        }
        if (keynum == 38 && head_for != 0)
                head_for = 3;
        if (keynum == 40 && head_for != 3)
                head_for = 0;
        if (keynum == 37 && head_for != 2)
                head_for = 1;
        if (keynum == 39 && head_for != 1)
                head_for = 2;
        if (keynum == 80)
                pause_flag = !pause_flag;
        if (keynum != 80)
                pause_flag = false;
}
```

1.详细设计

1.2.1模块划分

2. 开始画面模块

```
/* 定义事件执行监测 */
function onKeyDown(event) {
        if (status == -1) {
                status = 0;
                food();
                run();
        }
        if (window.event) // IE
        {
                keynum = event.keyCode;
        } else if (event.which) // Netscape/Firefox/Opera
        {
                keynum = event.which;
        }
        if (keynum == 38 && head_for != 0)
                head_for = 3;
        if (keynum == 40 && head_for != 3)
                head_for = 0;
        if (keynum == 37 && head_for != 2)
                head_for = 1;
        if (keynum == 39 && head_for != 1)
                head_for = 2;
        if (keynum == 80)
                pause_flag = !pause_flag;
        if (keynum != 80)
                pause_flag = false;
}
```

```
scene = new THREE.Scene(); /** * 创建场景对象Scene*/
        light = new THREE.DirectionalLight('white', 1.0, 0);
        light.position.set(-600, -600, -600);
        scene.add(light);
        plane = CreatePlane(400);
        plane.position.set(-5, -5, -5);
        scene.add(plane);
        //start_scene
        start_scene = new THREE.Scene();
        word = CreateText('snake 3D', 80, 40);
        word2 = CreateText('press any key to start', 30, 20);
        start_scene.add(word);
        start_scene.add(word2);
        start_scene.add(light);
        renderer.render(start_scene, camera);
        for (i = 0; i < nx; i++) {
                board[i] = new Array()
                for (k = 0; k < ny; k++) {
                        board[i][k] = 0;
                }
        } //0 = none, 1 = snake body, 2 = food
        fo = CreateCube(10, 10, 50);
        scene.add(fo);
        document.addEventListener('keydown', onKeyDown, false);
        for (i = 0; i < len; i++) {
                snake[i] = new Object();
                snake[i].x = head_pos_x + i * dir_x[3 - head_for];
                snake[i].y = head_pos_y + i * dir_y[3 - head_for];
                cube[i] = CreateCube(10, 10, 10);
                cube[i].position.x = snake[i].x * 10 - 200;
                cube[i].position.y = -snake[i].y * 10 + 190;
                scene.add(cube[i]);
                board[snake[i].x][snake[i].y] = 1;
        }
```

1.详细设计

1.2.1模块划分

3. 显示食物模块

```
/* 定义食物的坐标 */
function food() {
        var tx, ty;
        do {
                tx = Math.ceil(Math.random() * 1000) % nx;
                ty = Math.ceil(Math.random() * 1000) % ny;
        } while (board[tx][ty]);
        board[tx][ty] = 2;
        fo.position.x = tx * 10 - 200;
        fo.position.y = -ty * 10 + 190;
        fo.position.z = 20;
}
```

1.详细设计

1.2.1模块划分

4. 蛇向前移模块

```
/* 定义蛇的上下左右移动 */
function move() {
    var tx = snake[0].x + dir_x[head_for];
    var ty = snake[0].y + dir_y[head_for];
    //tx = (tx + nx) % nx;
    //ty = (ty + ny) % ny;
    if (tx >= 0 && tx < nx && ty >= 0 && ty < ny) {
        if (board[tx][ty] != 1) {
                    the_last_head = head_for;
                        if (board[tx][ty] == 2) {
                                    snake[len] = new Object();
                                    snake[len].x = snake[len - 1].x;
                                    snake[len].y = snake[len - 1].y;
                                    cube[len] = CreateCube(10, 10, 10);
                                    cube[len].position.x = snake[len].x * 10 - 200;
                                    cube[len].position.y = -snake[len].y * 10 + 190;
                                    scene.add(cube[len]);
                                    board[tx][ty] = 1;
                                    len++;
                                    food();
                        }
```

```
                                    for (i = len - 1; i > 0; i--) {
```

1.详细设计

1.2.1模块划分

5. 判蛇死模块

```
/* 定义蛇的上下左右移动 */
function move() {
        var tx = snake[0].x + dir_x[head_for];
        var ty = snake[0].y + dir_y[head_for];
        //tx = (tx + nx) % nx;
        //ty = (ty + ny) % ny;
        if (tx >= 0 && tx < nx && ty >= 0 && ty < ny) {
                if (board[tx][ty] != 1) {
                    the_last_head = head_for;
                    if (board[tx][ty] == 2) {
                        snake[len] = new Object();
                        snake[len].x = snake[len - 1].x;
                        snake[len].y = snake[len - 1].y;
                        cube[len] = CreateCube(10, 10, 10);
                        cube[len].position.x = snake[len].x * 10 - 200;
                        cube[len].position.y = -snake[len].y * 10 + 190;
                        scene.add(cube[len]);
                        board[tx][ty] = 1;
                        len++;
                        food();
                    }
                    for (i = len - 1; i > 0; i--) {
                        snake[i].x = snake[i - 1].x;
                        snake[i].y = snake[i - 1].y;
                    }
                                snake[0].x = tx;
                                snake[0].y = ty;
                } else {
```

1.详细设计

1.2.1模块划分

6. 吃到食物后处理模块

```
/* 定义蛇的上下左右移动 */
function move() {
    var tx = snake[0].x + dir_x[head_for];
    var ty = snake[0].y + dir_y[head_for];
    //tx = (tx + nx) % nx;
    //ty = (ty + ny) % ny;
    if (tx >= 0 && tx < nx && ty >= 0 && ty < ny) {
        if (board[tx][ty] != 1) {
                    the_last_head = head_for;
                        if (board[tx][ty] == 2) {
                                        snake[len] = new Object();
                                        snake[len].x = snake[len - 1].x;
                                        snake[len].y = snake[len - 1].y;
                                        cube[len] = CreateCube(10, 10, 10);
                                        cube[len].position.x = snake[len].x * 10 - 200;
                                        cube[len].position.y = -snake[len].y * 10 + 190;
                                        scene.add(cube[len]);
                                        board[tx][ty] = 1;
                                        len++;
                                        food();
                    }
```

```
                                        for (i = len - 1; i > 0; i--) {
```

1.详细设计

1.2.1模块划分

7. 判蛇反向移动模块

```
/* 定义蛇的前移或后移 */
function run() {
        if (status == -1)
                return;
        if (!pause_flag)
                move();
        render();
        setTimeout("run()", 100); //you can change speed here
}
```

1.详细设计

1.2.1模块划分

8.游戏结束模块

```
/* 定义蛇的上下左右移动 */
function move() {
    var tx = snake[0].x + dir_x[head_for];
    var ty = snake[0].y + dir_y[head_for];
    //tx = (tx + nx) % nx;
    //ty = (ty + ny) % ny;
    if (tx >= 0 && tx < nx && ty >= 0 && ty < ny) {
        if (board[tx][ty] != 1) {
                    the_last_head = head_for;
                        if (board[tx][ty] == 2) {
                                        snake[len] = new Object();
                                        snake[len].x = snake[len - 1].x;
                                        snake[len].y = snake[len - 1].y;
                                        cube[len] = CreateCube(10, 10, 10);
                                        cube[len].position.x = snake[len].x * 10 - 200;
                                        cube[len].position.y = -snake[len].y * 10 + 190;
                                        scene.add(cube[len]);
                                        board[tx][ty] = 1;
                                        len++;
                                        food();
                    }
```

```
                                        for (i = len - 1; i > 0; i--) {
```

1.详细设计

1.2.1模块划分

9. 图形结束模块

```
/* 定义蛇的上下左右移动 */
function move() {
    var tx = snake[0].x + dir_x[head_for];
    var ty = snake[0].y + dir_y[head_for];
    //tx = (tx + nx) % nx;
    //ty = (ty + ny) % ny;
    if (tx >= 0 && tx < nx && ty >= 0 && ty < ny) {
        if (board[tx][ty] != 1) {
                    the_last_head = head_for;
                        if (board[tx][ty] == 2) {
                                        snake[len] = new Object();
                                        snake[len].x = snake[len - 1].x;
                                        snake[len].y = snake[len - 1].y;
                                        cube[len] = CreateCube(10, 10, 10);
                                        cube[len].position.x = snake[len].x * 10 - 200;
                                        cube[len].position.y = -snake[len].y * 10 + 190;
                                        scene.add(cube[len]);
                                        board[tx][ty] = 1;
                                        len++;
                                        food();
                    }
```

```
                                        for (i = len - 1; i > 0; i--) {
```

1.详细设计

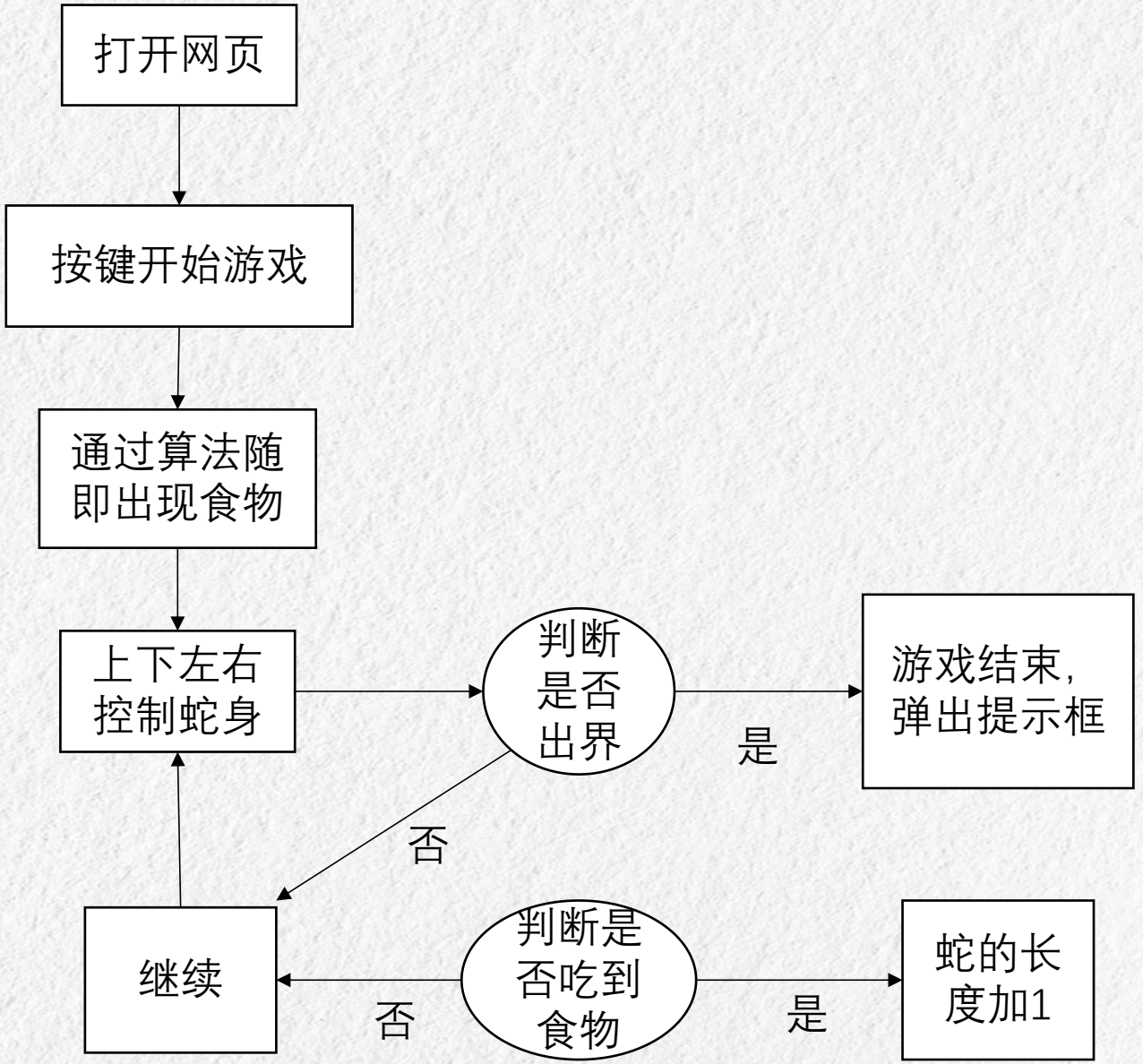
1.2.2算法流程

eat cake.

it's somebody's

birthday

somewhere.



1.详细设计

1.3实现工具&开发环境&工具库

1、必备技能：
1.HTML（主要是div盒子模型，canvas画布）
2.CSS（为你好看的游戏界面做准备）
3.JavaScript（让小蛇动起来，逻辑代码实现）

2、开发工具&开发环境：
1.HBuilder X，轻巧方便。
2.Google浏览器

3、工具库：
Jquery、Three.js、
optimer_bold.typeface.js。

1.详细设计

1. 4进度

详细设计	33天	10.21~11.17	系统完整功能描述；各模块流程、结构具体实现，关键函数、变量等说明
	4天	10.21~10.24	所采用实现工具，开发环境，主要工具库等
最终设计	24天	11.18~12.11	系统实现功能详细描述
	1天	12.11	实现功能展示(截图或视频)；小组成员实际完成工作情况及自评
	3天	12.12~12.14	课程设计展示汇报材料，包括报告用幻灯片，展示用视频或图片

2.总体设计

详细功能及代码说明

游戏基本功能

开始游戏功能 01

当用户进入游戏主界面时，可在界面中下方显眼的位置找到“开始游戏”按钮，点击后用户可进行新游戏，即单词版贪吃蛇游戏。

运动功能 02

用户可通过使用键盘上的上下左右方位键控制蛇的移动方向，蛇在控制的方向上进行直线前进。

吃食物功能 03

当界面任意位置出现食物，用户使用方位键控制蛇移动到食物周围，当蛇头碰到食物时则表示贪吃蛇吃到此食物，界面上会在任意位置出现下一个食物，用户再次控制蛇去吃这一食物。

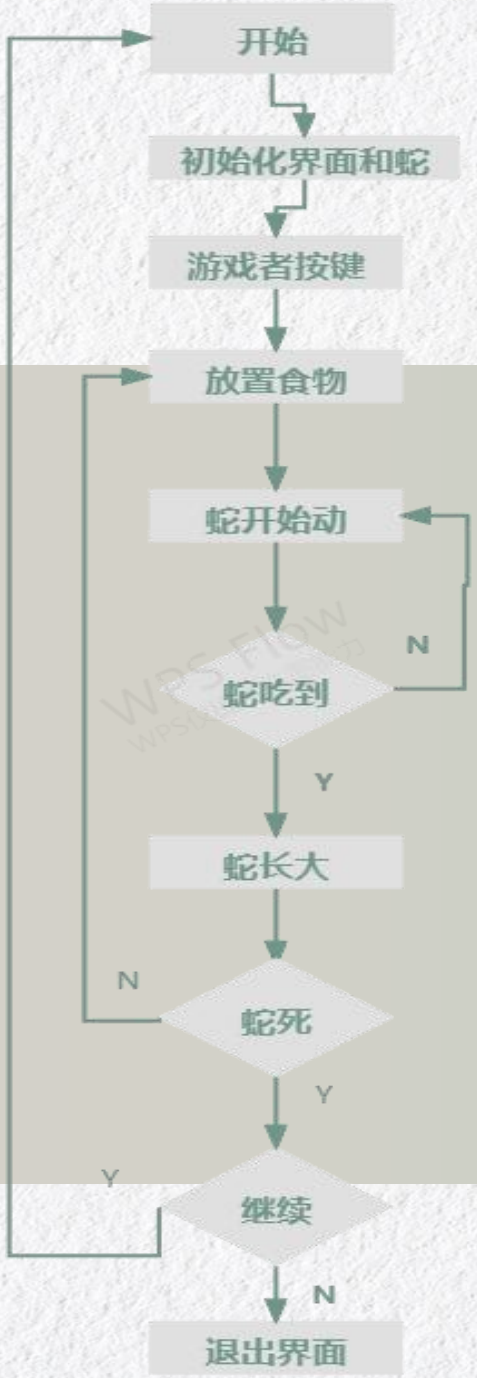
判定死亡功能 04

当蛇头在前进方向上撞到墙壁或蛇头撞到蛇身，给出死亡判定，并给出用户本次游戏得分。

2.总体设计

代 码 说 明

流程图设计



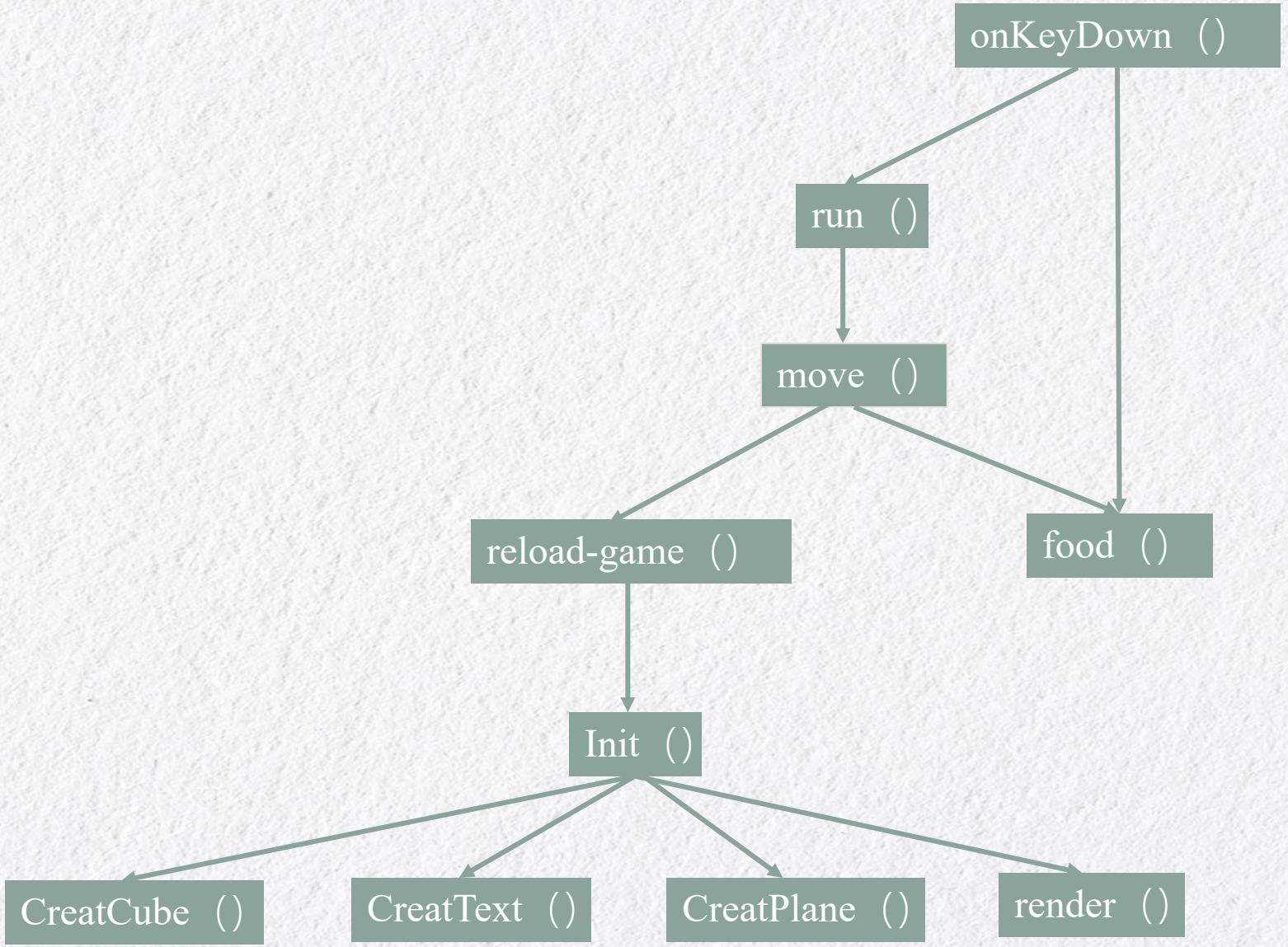
程序函数及其作用

Init()：初始化函数，对场景，相机进行初始化
CreateText()：用于显示初始界面的艺术字
CreateCube()：创建立方体函数，蛇身，实物均由立方体代表
CreatePlane()：创建蛇运动的平面区域
render()：渲染函数，对场景进行渲染
reload_game()：重新加载游戏
move()：移动函数，蛇在移动过程中会有食物的出现，游戏的结束等事件
food()：随机出现食物
run()：蛇的运动，在这里可以修改蛇的运动速度
onKeyDown()：键盘点击响应，任意键开始游戏，上，下，左，右四键控制蛇的移动方向

2.总体设计

代 码 说 明

函数调用关系



3.小组分工

3.小组分工

王倩倩	全程参与、视频录制	95
郑佳丽	全程参与	95

感谢老师的聆听！