佛山科学技术学院

电子信息工程学院

《2018秋学期教学实践周》报告

题 目：设计并实现贪吃蛇

学 院：电子信息工程学院

专 业：计算机科学与技术

班 级：18计算机科学与技术1

学 号：20180310115

姓 名：肖逍遥

指导教师：刘宪国

起止日期： 2018年 12月 29日 至 2019年 1月 12 日

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 递交的电子材料清单：   1. 《C语言程序设计课程设计》代码一份 2. 《C语言程序设计课程设计》报告一份  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | | 评价项目 | 评分标准 |  | 各项评分 | | |  | | 课程设计态度 | 态度积极，能按时到指定设计地点进行课程设计，不旷课，不迟到，不早退，学习态度认真，遵守课程设计阶段的纪律，作风严谨，按时完成课程设计规定的任务，按时上交课程设计有关资料。 | 优秀  | 良好  | 中等  | 及格  | 不及格   | | 课程设计作品 | 能熟练使用C语言开发工具顺利完成系统开发与调试，能按课程设计任务要求完成所有的工作，程序设计风格良好。作品设计原理科学、合理，结构清晰，测试实验数据可靠、详实。 | 优秀  | 良好  | 中等  | 及格  | 不及格   | | 课程设计报告 | 论文结构严谨，逻辑性强；行文流畅、专业技术用语准确；论文撰写符合规范，图表完备、整洁，符号统一、编号齐全；中英文摘要准确、恰当。 | 优秀  | 良好  | 中等  | 及格  | 不及格   |   课程设计最终成绩：  优秀 良好 中等 及格 不及格  指导老师签章：  日 期： 2019 年 月 日 |

目录

[1. 课程设计目的 - 1 -](#_Toc534747894)

[2. 课程设计的内容 - 1 -](#_Toc534747895)

[3. 课程设计的功能要求 - 1 -](#_Toc534747896)

[3.1. 页面菜单模块： - 1 -](#_Toc534747897)

[3.2. 初始化地图模块： - 1 -](#_Toc534747898)

[3.3. 食物更新模块： - 1 -](#_Toc534747899)

[3.4. 移动贪吃蛇模块： - 2 -](#_Toc534747900)

[3.5. 贪吃蛇生命模块： - 2 -](#_Toc534747901)

[3.6. 获取贪吃蛇速度模块： - 2 -](#_Toc534747902)

[3.7. 移动光标模块： - 2 -](#_Toc534747903)

[3.8. 帮助系统模块： - 2 -](#_Toc534747904)

[3.9. 退出系统模块： - 2 -](#_Toc534747905)

[4. 概要设计 - 3 -](#_Toc534747906)

[4.1. 系统功能框架图 - 3 -](#_Toc534747907)

[4.2. 各功能模块描述: - 4 -](#_Toc534747908)

[4.3. 页面菜单模块(menu)： - 4 -](#_Toc534747909)

[4.4. 初始化地图模块（drawmap）： - 4 -](#_Toc534747910)

[4.5. 食物更新模块(createfood)： - 5 -](#_Toc534747911)

[4.6. 移动贪吃蛇模块(input)： - 5 -](#_Toc534747912)

[4.7. 贪吃蛇生命模块(death)： - 6 -](#_Toc534747913)

[4.8. 获取贪吃蛇速度模块(speed)： - 6 -](#_Toc534747914)

[4.9. 移动光标模块(gotoxy)： - 7 -](#_Toc534747915)

[4.10. 帮助系统模块(help)： - 7 -](#_Toc534747916)

[4.11. 退出系统模块(exit)： - 8 -](#_Toc534747917)

[5. 详细设计 - 8 -](#_Toc534747918)

[5.1. 各功能模块的具体实现算法——流程图 - 8 -](#_Toc534747919)

[5.1.1. 主控 main 函数流程图 - 8 -](#_Toc534747920)

[5.1.2. 菜单模块menu函数流程图 - 9 -](#_Toc534747921)

[5.1.3. 初始化地图模块drawmap函数流程图 - 9 -](#_Toc534747922)

[5.1.4. 创造食物模块createfood函数流程图 - 10 -](#_Toc534747923)

[5.1.5. 移动贪吃蛇模块input函数流程图 - 11 -](#_Toc534747924)

[5.1.6. 贪吃蛇生命模块death函数流程图 - 12 -](#_Toc534747925)

[5.1.7. 贪吃蛇速度模块speed函数流程图 - 12 -](#_Toc534747926)

[5.1.8. 贪吃蛇获取光标模块gotoxy函数流程图 - 13 -](#_Toc534747927)

[5.2. 一部分重点设计具体实现的代码 - 13 -](#_Toc534747928)

[5.2.1. gotoxy函数移动光标模块设计 - 13 -](#_Toc534747929)

[5.2.2. input函数移动贪吃蛇模块设计 - 14 -](#_Toc534747930)

[5.2.3. createfood食物生成模块设计 - 15 -](#_Toc534747931)

[6. 系统测试与调试 - 16 -](#_Toc534747932)

[7. 课程设计心得体会 - 17 -](#_Toc534747933)

# 课程设计目的

进一步加深、巩固所学专业课程《C语言程序设计》的基本理论知识，理论联系实际，进一步培养自己综合分析问题和解决问题的能力。掌握运用C语言独立地编写、调试应用程序和进行其它相关设计的技能。熟悉使用结构体与时间函数和windows窗口控制函数。

# 课程设计的内容

设计并实现一个贪吃蛇小游戏，要求具有以下功能模块：

1. 页面菜单模块。
2. 初始化地图模块。
3. 食物更新模块。
4. 移动贪吃蛇模块。
5. 判断贪吃蛇是否存活模块。
6. 获取贪吃蛇速度模块。
7. 移动窗口光标模块。
8. 帮助系统模块
9. 退出系统模块

# 课程设计的功能要求

## 页面菜单模块：

* 开始游戏
* 退出游戏

## 初始化地图模块：

* 实现贪吃蛇存在范围（这里用围墙来限定区域）
* 初始化贪吃蛇（3单位身体长度）
* 初始化食物

## 食物更新模块：

* 在区域内随机刷新食物
* 食物不与蛇身体重合

## 移动贪吃蛇模块：

* 根据玩家按键移动贪吃蛇（w,s,a,d）
* Space 空格可暂停或继续游戏
* 当贪吃蛇前进时按相反方向移动无效
* 除了程序控制键之外的其他按键不会影响贪吃蛇
* 移动贪吃蛇时不会造成屏幕闪烁BUG

## 贪吃蛇生命模块：

当蛇吃到自己身体或者触碰围墙时蛇死亡

## 获取贪吃蛇速度模块：

* 根据蛇的身体长度来设置速度等级
* 当身体长度足够长时则通关游戏

## 移动光标模块：

* 获取输出流窗口句柄
* 定义使用存储坐标结构体
* 将坐标与窗口句柄传入函数实现移动光标

## 帮助系统模块：

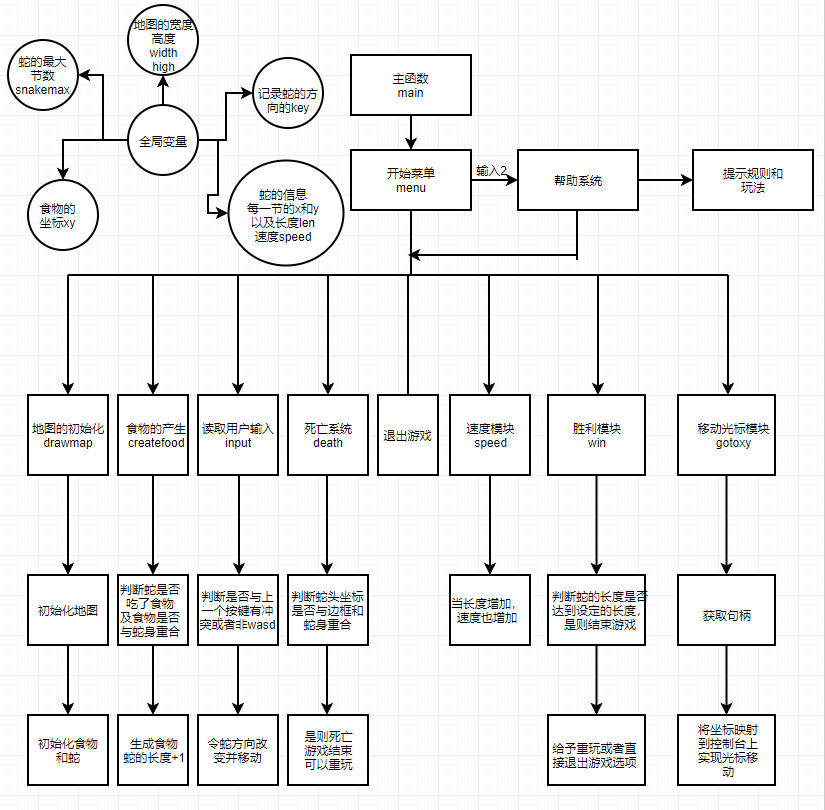
提示玩家一些按键控制和游戏规则

## 退出系统模块：

游戏通关或游戏结束时给予退出或重玩选择

# 概要设计

## 系统功能框架图



*系统的大概功能框架（图1）*

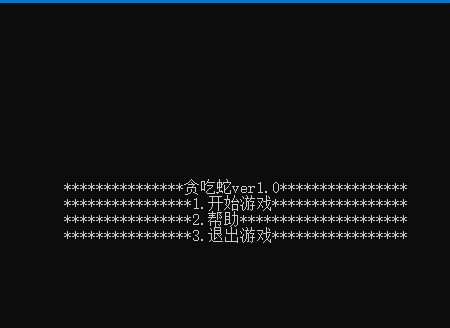
## 各功能模块描述**:**

## 页面菜单模块(menu)：

页面的主菜单，命名为menu，该函数初定设计为：

1. 按下1按键可以直接进入游戏。
2. 按下2可以显示帮助系统.
3. 按下3直接结束游戏.

我在此函数中利用kbhit来判断用户是否有输入按键，以key2来记录用户输入的值，以此判断并返回不同的返回值，若返回1，可以在主函数判断程序结束，游戏结束。若返回0，则在主函数中继续。



*菜单运行界面（图2）*

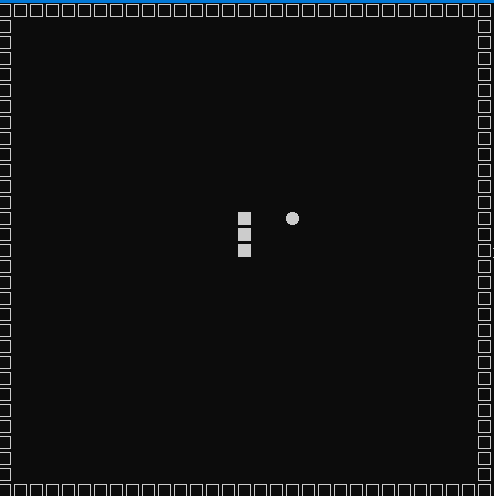
## 初始化地图模块（drawmap）：

对地图的初始化，首先先要设定食物边框和蛇神的符号。

●代表食物 □代表边框 ■代表蛇身。

地图要初始化，就需要对地图进行大小的确定和划分，这里我预处理定义了width和high来代表地图的宽和高，以此来打印边框和最初的食物还有蛇的身体。

蛇的身体也初步设定为默认刷在地图的中间(width/2,high/2)。

其次，由于符号□所占字节为2，所以在地图初始化for循环的时候，打印行数是需要+2来打印的。且食物的横坐标也应该是偶数。

*初始化的游戏界面（图三）*

## 食物更新模块(createfood)：

在每次蛇吃到食物后，食物应该更新在地图的随机一块区域，因此该模块主要功能应该是：

1.判断蛇是否有吃到食物。

2.利用随机数更新食物。

2.1限定随机数的范围以避免食物更新在地图外或地图边缘。

2.2判断更新的食物是否在蛇身上。

3.令蛇的长度加一 同时打印出食物本身。

## 移动贪吃蛇模块(input)：

移动贪吃蛇应该不断读取用户输入的按键，为此需要利用到kbhit函数。而该函数因为需要不断读取按键，应该在该函数里增加fflush(stdin)来及时清空缓存区。

因为在输入按键之后需要判断是否和上一个按键相反或者是wasd以外的按键，所以需要设置一个临时变量记录用户输入的值。然后判断用户输入的值是否与全局变量key值冲突，若冲突则判定为输入无效。若不是wasd也令其无效。若是空格则可以跳转到暂停函数。

贪吃蛇的移动则使用以下方法进行：

1. 判断方向是否改变
2. 将光标移到蛇尾处(len-1)
3. 打印“ ”擦除蛇尾的方块
4. 将当前蛇身全部+1，以做到移位
5. 蛇头++（若横向移动则+2），蛇头处重新打印方块

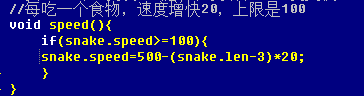
因为该方法使用的是光标移动的方法，且只会打印头部以及擦除尾部，因此初步判断是不会出现屏幕闪烁。

## 贪吃蛇生命模块(death)：

判断蛇是否死亡，需要获取蛇头的坐标。而当蛇头的坐标不会与蛇身的坐标重合，或者不会与墙的坐标重合，可以判断蛇依旧存活。如果重合则判定为死亡，则自动跳出循环。

## 获取贪吃蛇速度模块(speed)：

判断该模块依赖于sleep函数，因为该函数可以让蛇的每次移动中途有间隔。因此利用该函数可以将间隔的时间随着len的长度主机变短。另外设定该速度上限为100。



*初始速度设置在500(图4)*

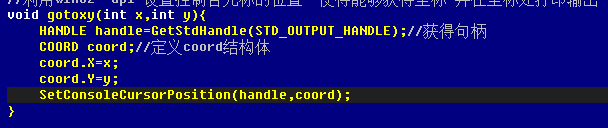


*游戏中的速度显示效果（图5）*

## 移动光标模块(gotoxy)：

该模块为该贪吃蛇游戏设计的核心。该模块利用win32获取句柄，并且将坐标映射到控制台，则光标可以通过该坐标来移动，做到打印地图，食物等事情。具体设计是定义coord结构体，将所想要的坐标赋值于coord。利用SetConsoleCursorPosition来将coord的坐标定位在句柄上。

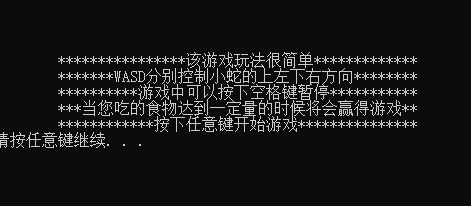
该函数贯穿整个程序，因此是最重要的设计部分。



*具体的函数设计（图6）*

## 帮助系统模块(help)：

该部分设计较为简单，主要是提示用户一些注意点。主要是printf打印出游戏的规则。

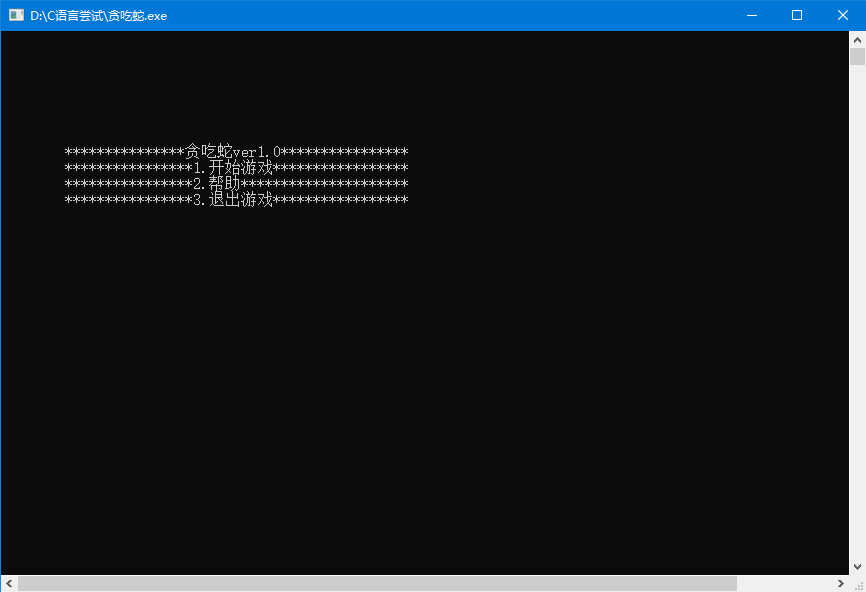


*帮助界面（图7）*

## 退出系统模块(exit)：

该部分应该设计退出函数的选项，通过exit直接结束以达到直接退出系统。

## 软件运行的界面：



*软件的开始界面（图8）*

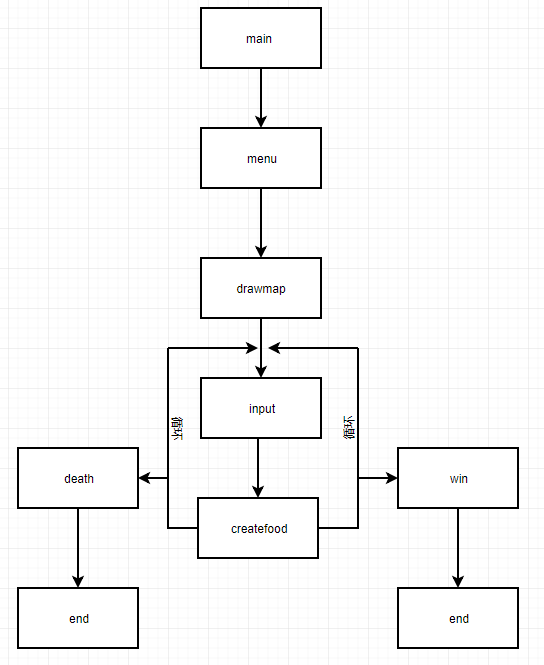


*游戏运行的具体截图（图9）*

# 详细设计

## 各功能模块的具体实现算法——流程图

### 主控 main 函数流程图

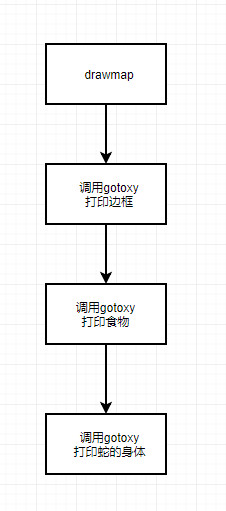


*main函数流程图(图10)*

### 菜单模块menu函数流程图

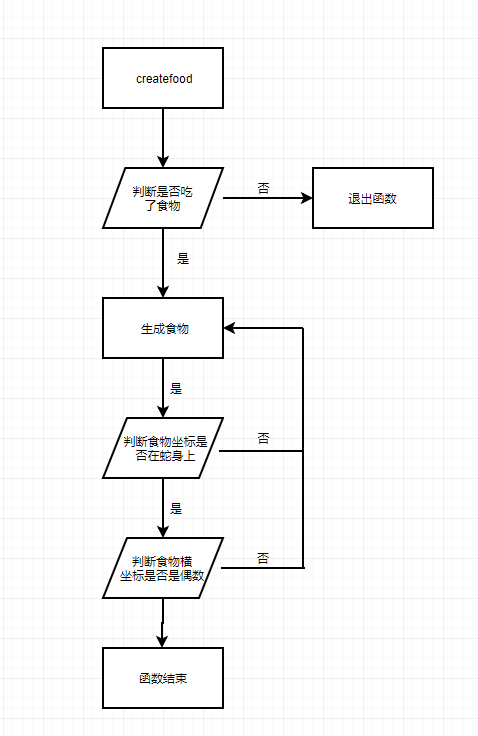
*menu函数流程图（图11）*

### 初始化地图模块drawmap函数流程图



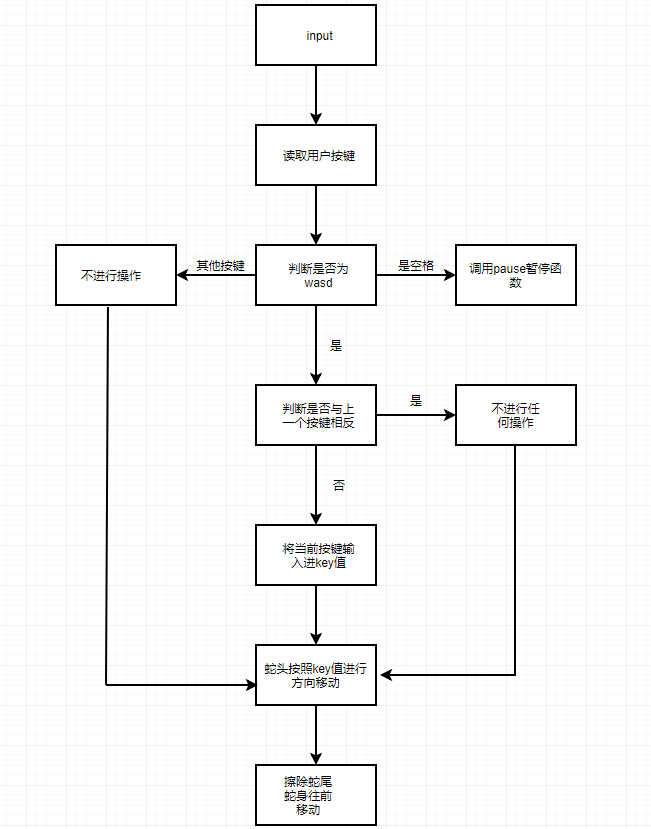
*drawmap函数流程图（图12）*

### 创造食物模块createfood函数流程图



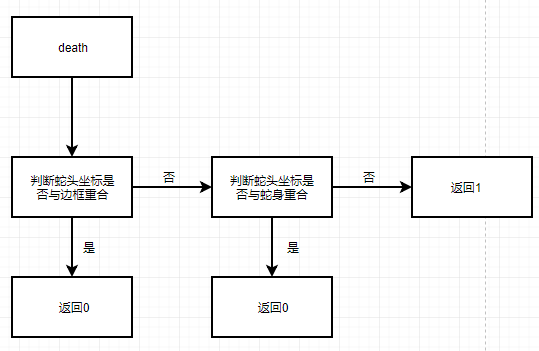
*食物生成模块createfood流程图（图13）*

### 移动贪吃蛇模块input函数流程图



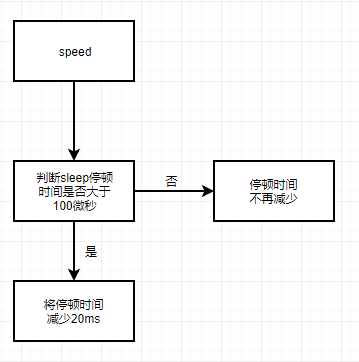
*贪吃蛇移动模块input函数(图14)*

### 贪吃蛇生命模块death函数流程图



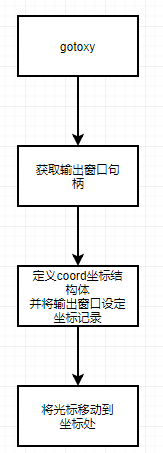
*生命模块death函数流程图(图15)*

### 贪吃蛇速度模块speed函数流程图



*speed函数流程图（图16）*

### 贪吃蛇获取光标模块gotoxy函数流程图



*gotoxy函数流程图(图17)*

## 一部分重点设计具体实现的代码

### 该贪吃蛇的重点设计核心无疑是gotoxy这个函数，通过这个函数，能够令光标不断在控制台上移动，从而做到打印食物，打印边框，打印蛇等举措。而关于该函数我的理解不多。基本算是百度查阅到的代码并且参考摘抄使用。而他的具体实现代码很简单，是通过handle获取句柄，然后就能够映射到控制台上输出。具体代码如下：

1. void gotoxy(int x,int y){
2. HANDLE handle=GetStdHandle(STD\_OUTPUT\_HANDLE);//获得句柄
3. COORD coord;//定义coord结构体
4. coord.X=x;
5. coord.Y=y;
6. SetConsoleCursorPosition(handle,coord);
7. }

### 而第二个的重点核心代码便是蛇的移动部分，这里我将蛇的移动和打印一起做，因此将他们合并命名为input。而一个重点就是如何做按键的过滤和读取。开始按键的过滤烦扰了我好久，后来才想到做一个暂存变量暂时存入用户输入的按键，然后做判断。若判断为wasd且不是相反冲突键，便将其的值直接赋予全局变量中的key值。具体设计如下：

1. void input(){
2. int i;int t;
3. if(kbhit())//利用kbhit来读取键盘按键的数据
4. {
5. fflush(stdin);//及时清空缓存区 以免妨碍下一个的按键接收（快速按键）
6. t=getch();
7. //判断是否是相反的键
8. if((t=='w'||t=='W')&&(key!='s'&&key!='S')){
9. key=t;
10. }else
11. if((t=='s'||t=='S')&&(key!='w'&&key!='W')){
12. key=t;
13. }else
14. if((t=='a'||t=='A')&&(key!='d'&&key!='D')){
15. key=t;
16. }else
17. if((t=='d'||t=='D')&&(key!='a'&&key!='A')){
18. key=t;
19. }else
20. if(t==' '){
21. pause();//如果输入的是空格
22. }
23. //令key =t输入进去的数据（设定为WASD分别是上左下右方向控制）
24. }
25. // 若不输入 或者输入无效 自动以key的初始值来执行
26. gotoxy(snake.x[snake.len-1],snake.y[snake.len-1]);
27. printf("  ");//最后的一格擦除
28. //剩下的往前挪动
29. for (i=snake.len-1;i>0;i--){
30. snake.x[i]=snake.x[i-1];
31. snake.y[i]=snake.y[i-1];
32. }
33. //移动方向的处理
34. switch(key){
35. case 'w':
36. case 'W':
37. snake.y[0]--;
38. break;
39. case 's':
40. case 'S':
41. snake.y[0]++;
42. break;
43. case 'a':
44. case 'A':
45. snake.x[0]-=2;
46. break;
47. case 'd':
48. case 'D':
49. snake.x[0]+=2;
50. break;
51. }
52. gotoxy(snake.x[0],snake.y[0]);
53. printf("■");
54. }

### 而另外一个重点设计的内容也就是食物的生成了，食物的生成第一需要一个随机数来随机坐标所在的位置，第二需要判断食物的坐标是否与蛇的身体重合，第三还需要判断食物的横坐标是否是偶数。这样就需要做一个循环，食物与蛇身重合的概率小，但食物横坐标是奇数的概率大，而且如果只是因为是奇数就再次生成坐标则太过于大费周章，因此我直接将横坐标+1以获得偶数。这样也解决了贪吃蛇运行卡顿的问题，具体实现的代码如下：

1. void createfood(){
2. //首先判断蛇是否吃了食物
3. //因为蛇头所代表的 符号占两个字节 所以坐标必须是偶数。
4. //且不能刷在蛇身上
5. if(food.x==snake.x[0]&&food.y==snake.y[0]){
6. while(1){
7. int i;
8. int flag=1;//标志，满足不在蛇身上的标志
9. srand(time(NULL));//随机函数种子
10. food.x=rand()%(width-4)+2;
11. food.y=rand()%(high-2)+1;
12. for(i=0;i<snake.len;i++){
13. if(snake.x[i]==food.x&&snake.y[i]==food.y){
14. flag=0;
15. break;//跳出for循环 重新生成坐标or直接进入if判断跳出死循环
16. }
17. }
18. if(flag=1&&food.x%2!=0){
19. food.x++;
20. break;
21. }
22. if(flag=1&&food.x%2==0){
23. break;//跳出死循环
24. }
25. }
26. gotoxy(food.x,food.y);
27. printf("●");
28. snake.len++;
29. }
30. }

# 系统测试与调试

在调试贪吃蛇的过程中主要出现以下几个 BUG

Bug1：运行中按下wasd以外的键程序直接结束。

分析原因：没有增加按键的过滤，直接将输入的值赋予key全局变量。

解决方案：在增加按键的过滤判断后bug解决。

Bug2：运行中贪吃蛇每次吃下食物后卡顿。

分析原因：在createfood的时候因为食物横坐标会大概率刷出奇数，导致循环不断重跑造成的卡顿。

解决方案：在刷出奇数的情况直接将横坐标+1，以达到算法的优化。

Bug3：在按下相反的方向键的情况下，贪吃蛇直接死亡。

分析原因：也是没有增加按键的过滤。

解决方案：直接增加按键的相反判断后解决。

Bug4：吃了食物之后没反应

分析原因：因为食物的横坐标是奇数

解决方案：将食物的横坐标控制为偶数，否则重新生成

Bug5：打印横边框出现重叠

分析原因：和Bug4原因相同

解决方案：在打印横边框时+2处理。

# 课程设计心得体会

在设计了这次实践周的课程项目《贪吃蛇》之后，我对C语言的熟练度稍微又更加上了一层，之前的我学了C语言仅仅只停留在做一些小算数题和一些作业，而这次才是第一次真正意义上去设计一个比较独立的程序。在这次设计中，我学会了如何去系统的执行实现一个项目的流程，学会了如何去封装函数，也学习了如何不断的调试程序以消除bug。如果只停留在做计算题的程度的话，我们是永远无法去成为一个好的程序员的，而这次的项目使我第一次有了实现程序的流程和思想，虽然实现的途中有着许多困扰的问题，但是在思考的过程也是作为一个程序员所必备的素质。

贪吃蛇这个程序或许不大，但是他的启发作用却实实在在的作用在了我身上，从贪吃蛇的功能开始分析，具体到每一个模块，再到模块的具体功能的设计，最后用代码将功能实现出来。然后再把这些功能拼装起来，最后组成一个完整的贪吃蛇的小程序。或许很简单，但其实一开始的时候我是完全没有任何头绪的，然后我一点点的将他们分开来去实现，去不断解决问题，再不断去实现功能，查阅资料……在这个过程中我感受到了编程的魅力，编程在以前的我看来，就是不断的在电脑前打着代码，枯燥而又乏味。现在看来，编程其实是一步一步的去实现每一个功能，完善每一个模块，最后将他们以程序的方式运行出来，这是一种创造，是一种从无到有的事情。可以说在电脑的世界里我们充当着一个创世主的角色。虽然这话说的有点大，但是意思是差不多的。

在以后，我也会努力的学习代码知识，创造是需要更多的知识的。没有知识将无法支撑我们的创造。这每一种语言其实就相当于我们的工具箱，我们不断用这些工具去实现更多的事情，做出更多的功能，最后将他们拼装，我想这就是程序员吧。