**面向对象程序设计（C++）**

**课程设计报告**

|  |  |  |  |
| --- | --- | --- | --- |
| 题 目 | 贪吃蛇游戏 | | |
| 日 期 | 2019年8月1日 至 2019年 9月 7 日 | | |
| 专 业 | 计算机科学与工程 | 班 级 | 计1802 |
| 姓 名 | 赫洱锋 | 学 号 | 20184443 |

报告成绩

|  |
| --- |
|  |

**报告**撰写的内容与要求

1. 项目简介：介绍本次课程设计选题的目的、意义、任务概况、本人在项目中的分工等内容。
2. 项目内容：系统的设计与实现的全面描述，介绍系统整体结构、系统框架图、UML类图、程序流程图、重点难点分析及解决方案、调试难点及解决方法、系统交互界面及结果展示等。本部分内容应以记叙或[白描手法](http://www.so.com/s?q=%E7%99%BD%E6%8F%8F%E6%89%8B%E6%B3%95&ie=utf-8&src=wenda_link" \t "_blank)为基调，在完整叙述的基础上，对自己认为有重要意义或需要研究解决的问题进行重点叙述，其它内容则可简述。
3. 总结或体会：对课设效果进行综合评价，着重介绍自身的收获与体会，内容较多时可列出[小标题](http://www.so.com/s?q=%E5%B0%8F%E6%A0%87%E9%A2%98&ie=utf-8&src=wenda_link" \t "_blank)，逐一列举。总结或体会的最后部分，应针对实习中发现的自身不足，简要地提出今后学习，努力的方向。
4. 报告正文一律采用计算机排版、A4纸**双面**打印，正文字体为**小四号**宋体，**1.35倍**行距，正文页数不低于10页，不超过30页（其中代码总计不超过3页，**中文字数**不少于4000字）。要求语句通顺、论述严谨、规范、正确。
5. 请注意封面页、扉页、评语页等的打印及装订顺序。分别为①封面页（其背面为扉页“实习报告的内容与要求”）、②目录、③正文、④其他附件（如有）。

# 项目简介--------------------------------------------------------------2页

1. 选题目的：----------------------------------------------------------------------------2页
2. 选题意义：----------------------------------------------------------------------------2页
3. 分工：-----------------------------------------------------------------------------------2页

# 项目内容------------------------------------------------------------2~8页

1. 系统的设计与实现的全面描述：---------------------------------------------------2页
2. 系统框架图：----------------------------------------------------------------------------3页
3. UML类图：-------------------------------------------------------------------------------4页
4. 程序流程图：----------------------------------------------------------------------------5页
5. 重点难点分析：----------------------------------------------------------------------5~6页
6. 解决方案：-------------------------------------------------------------------------------7页
7. 调试难点：-------------------------------------------------------------------------------7页
8. 解决方法：-------------------------------------------------------------------------------7页
9. 系统交互及结果界面：------------------------------------------------------------8~10页

# 总结与体会

1. 课设效果综合评价：------------------------------------------------------------------10页
2. 收获与体会：----------------------------------------------------------------------10~11页
3. 自身不足及今后努力方向：-------------------------------------------------------------11页

# 一、项目简介

1、选题目的：

贪吃蛇是家喻户晓的益智类小游戏，选择这个题目一是为了将自己的所学知识加以运用，二是一直以来贪吃蛇这个游戏就深深地吸引着我，它的制作方法对于以前的我而言都是很神秘的。我希望通过自己的所学知识把它剖析开来，真正的了解它的本质和精髓。虽然我的编程能力不是很强，但是我有信心，在这次学习中我将从实践和实际的项目中提高自己的编程能力。因此在小组讨论后共同决定选择了这个题目。

2、选题意义：

提高自身编程能力，通过开发游戏提高自己对编程的兴趣

1. 任务概况：

设计并实现贪吃蛇游戏系统；有欢迎与登录界面、说明界面和游戏界面。欢迎界面显示游戏的名称和游戏菜单，选择不同的菜单可以实现开始游戏、阅读游戏说明、按键说明及退出游戏功能。设计游戏地图、绘制蛇身及随机出现食物。实现贪吃蛇功能，即控制蛇的方向、设置蛇撞墙的情况。显示游戏积分、执行步数及所用时间。实现在失败界面进行选择（重玩一局或退出游戏）

5、分工：负责可破坏类墙的生成，模式编写，难度编写，地图编写，蛇体调节以及调试完善工作。

# 二、项目内容

1、系统的设计与实现的全面描述：

本游戏包括启动器部分（QT），游戏部分（控制台）

启动器界面：

1）主选单

2）模式、皮肤、难度选单

游戏界面：

1）程序运行界面（显示界面）

2）操作界面（右侧）（包括信息统计与游戏选单）

①继续游戏

②重新开始

③结束游戏

游戏设置：

1）蛇的控制（wasd，方向键）

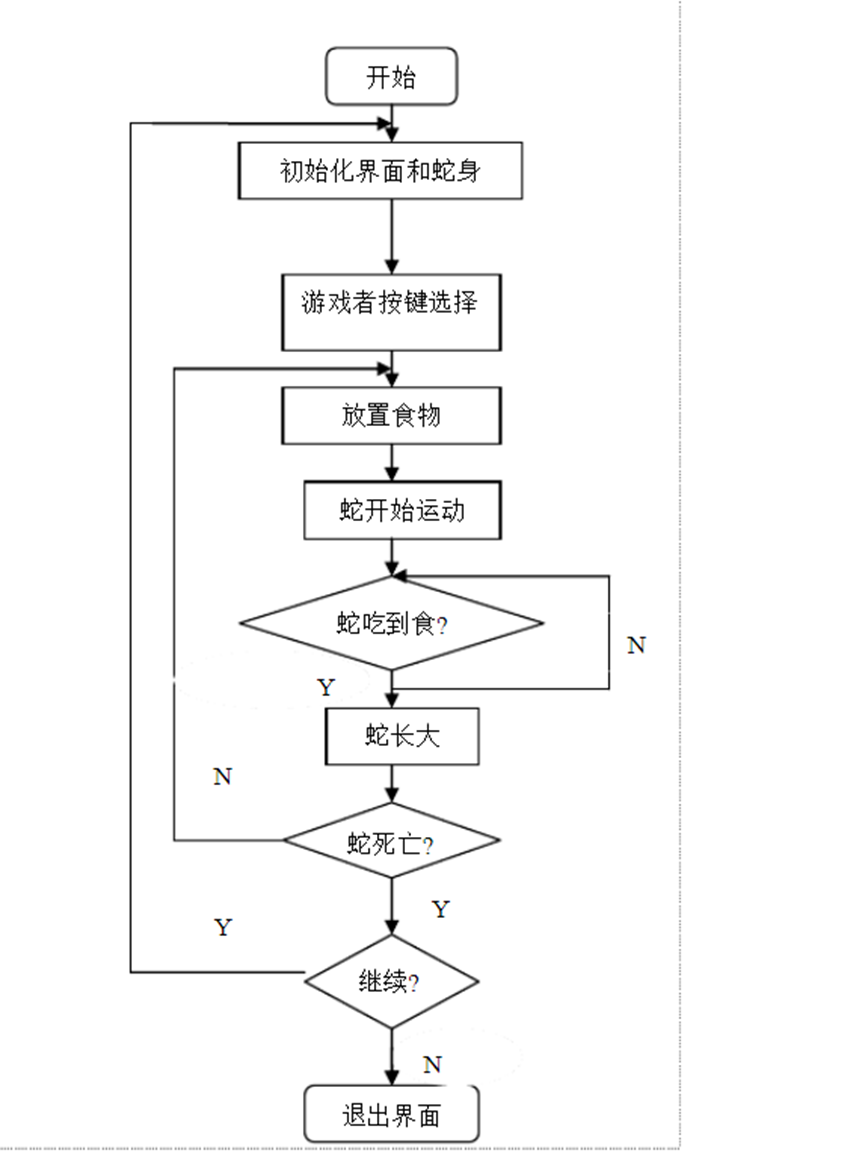
2）模式设置（经典模式、普通模式、街机模式）

3）难度设置（低中高）

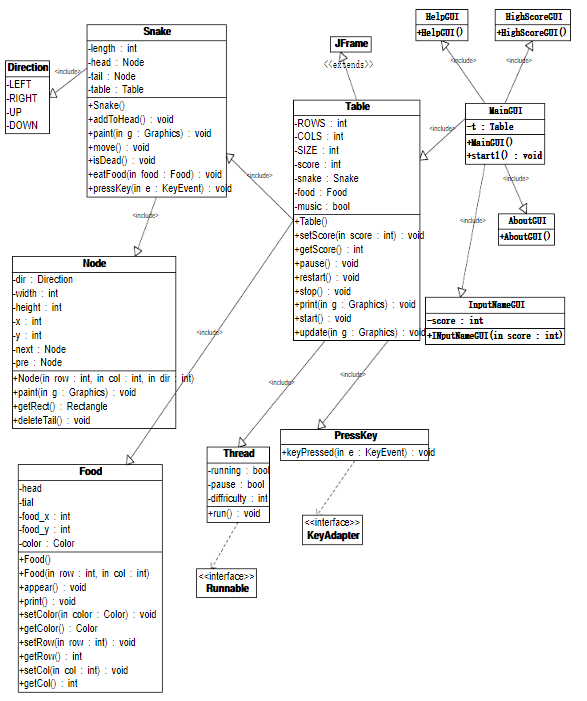
4）积分（每次吃掉一个食物增加5分，并显示出来）

使用easyX来完成图案的绘制，使用QT制作界面。

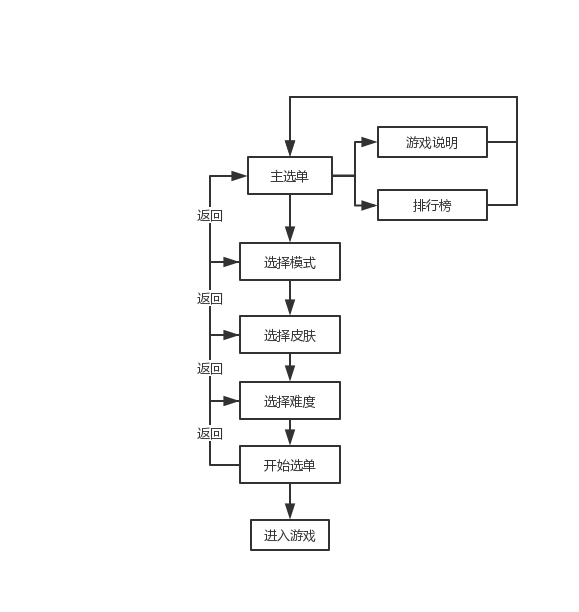
2、系统框架图：

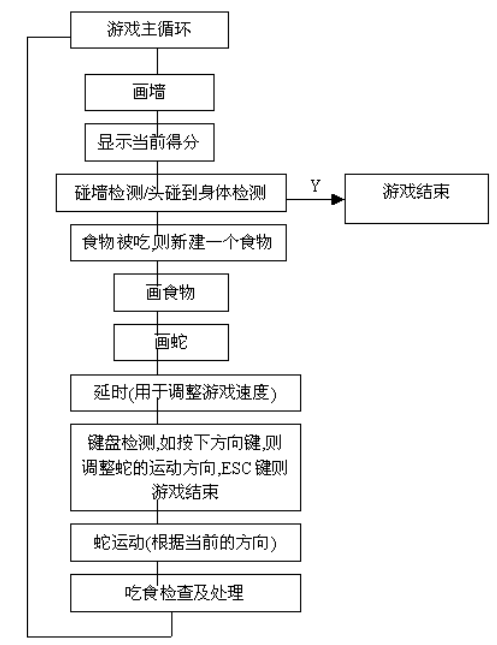


3、UML类图：



4、程序流程图：





5、重点难点分析：

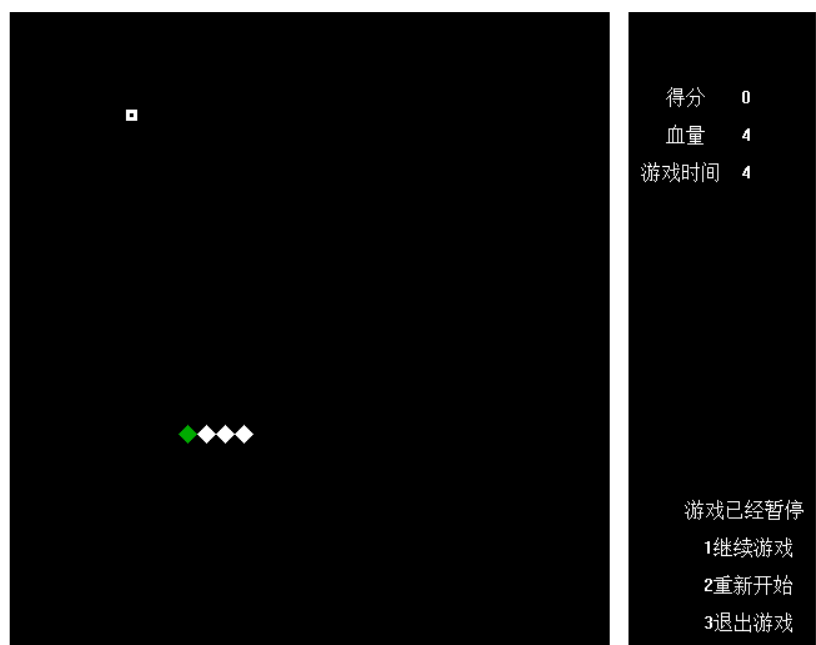
游戏界面，贪吃蛇的游戏界面包括墙的绘制，蛇身体的绘制，蛇移动范围的限制，食物的控制遗迹以及积分注释界面的绘制，其中难点是蛇身体的绘制、死亡后刷新像素点避免蛇的尸体残留、在蛇撞到墙上的时候避免蛇的身体覆盖墙体导致墙体消失等。这个程序的关键是表示蛇的图形以及蛇的移动。用一个小矩形表示蛇的一节身体，身体每长一节，增加一个矩形块。移动时必须从蛇头开始，所以蛇不能向相反方向移动，也就是蛇尾不能改作蛇头。如果不按任何键，蛇自行在当前方向上前移，当游戏者按了有效的方向键后，蛇头朝着指定的方向移动，一步移动一节身体，所以当按了有效的方向键后，先确定蛇头的位置，然后蛇身体随着蛇头移动，图形的实现是从蛇头的新位置开始画出蛇，这时原来蛇的位置和新蛇的位置差一个单位，所以看起来蛇会多一节身体，所以将蛇的最后一节用背景色覆盖。食物的出现和消失也是画矩形块和覆盖矩形块。

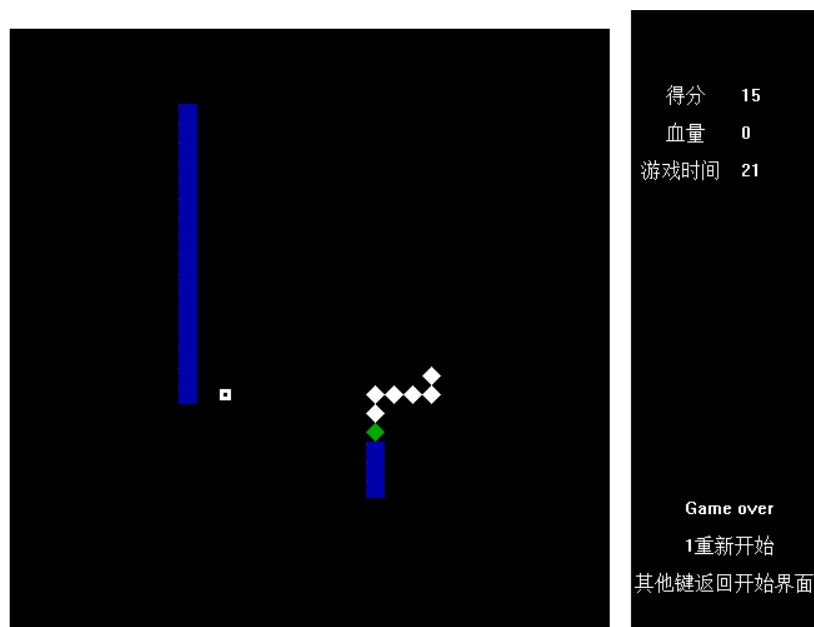
6、解决方案：用easyX绘制地图、蛇身、蛇头、以及食物，在蛇死亡后刷新界面确保不会出现bug，最终选择给地图的每个15\*15的像素点设置属性并依照属性填充颜色、形状，这样只要弄清楚填充颜色和清屏的先后顺序就避免了大部分覆盖问题。蛇的移动最后决定通过蛇头移动并判定是否吃到豆子，如果吃到则蛇尾不动，吃不到则蛇尾消失来体现移动，并依此设置了难度选择、地图选择、血量选择等。

7、调试难点：在优化游戏可玩性时选择为蛇增加血量并增加可以摧毁但是扣血的墙体，并设计了加血道具完善游戏性，但实践途中由于改变蛇的类的成员产生了问题。调试时链接动态符号库失败也困扰了我们很久

8、解决方法：采用友元的知识改变一部分私有成员使得蛇的更多属性可以被改变，同时通过改变编译器设置也解决了链接动态符库失败的问题

9、系统交互及结果界面：









# 三、总结与体会

1、课设效果综合评价：

整体上达到了课设要求 ，游戏的完成度很高，但仍有可以优化的地方，界面不够美观，游戏可玩性不足，可以添加更多模式以及成就系统增加游戏的可玩性。

2、收获与体会：

在设计这个程序中我主要学会了如何运用以下有关C++的知识：1)函数定义是要做到顾名思义是很重要的，它对读程序的人正确认识程序十分重要，在修改这个程序的过程中也能很快找到程序各模块的作用，大大增加了程序的可读性。2)分析函数先从main()函数入手。Main()函数是源程序编译时的开始，从main()函数开始读函数可将其他函数的功能理解得更透彻。3)在做程序的时候先列框架，将这个程序所要达到的目的（功能）分析出来，选择正确的数据结构然后在将程序模块化，按照模块编写函数更加简单合理。4)我还了解了很多的库函数的作用，如字符串函数中有很多对字符串进行处理的函数，其功能我都有所了解。5）学会了有关头文件的使用方法，及系统调配问题的解决方法等。6）了解到了注释的重要性。

通过这次试验，让我学到很多这个学期没有接触过的很多知识，完成一个贪吃蛇的项目；第一步就是先理清思路，设计主界面，主界面是以一块封闭的画布，并在画布内设计蛇体和食物。蛇头如果碰到墙或者与蛇身的某一部分碰到就会死亡；如果蛇头碰到食物，就会自动增加一节蛇身，并更改得分。想要写出好的程序，还是要有扎实的基础，这样遇到问题就不会一筹莫展来。在编程时我们要有想象力，不要拘泥于现有的贪吃蛇游戏，而是要把自己想到的再加上所学的知识，遇到问题不能怕，而是要专心找出问题的原因，再回来看看书本的知识，不懂的还可以问问同学。编程过程还要养成良好的编程习惯，+这样不仅自己能够很好看懂自己的代码，也让人家容易看懂自己的代码，而不至于代码知识自己会看，出来问题只有自己会解决而其他人都不会解决；遇到问题时要考虑最简单的解决方案，只有简单的方案不能满足要求时再考虑复杂的方案。最主要编程还是要多与人沟通，不会的不能立即问别人，但是一段时间后解决不了时还是应该向别人请教请教，“三人行必有我师”正是这个意思。通过这次实验，我深刻地体会到了面向对象与面向过程的区别。程序在最初设计时由于面向对象的思想不够成熟，导致在最后想要增加一些新的功能时，出现了一些麻烦。总之，我们还需要进一步提升自己的能力。

3、自身不足及今后努力方向：

自身编程能力有限，理论大过实际导致实际编程时出现很多意想不到的小问题影响了编程进度，对程序的阅读能力不足，编程习惯不好导致阅读程序时遇到困难，影响编程，此后应该更加积极的进行编程练习，多使用注释以及采用合适的函数名和变量名保证程序可读性。

# 附录：本人编写代码（部分）

void mapinit(int a)

{

if (a == 3)

{

for (int i = 0; i < 34; i = i ++)

{

for (int j = 0; j < 34; j++)

{

map[1].mapp[i][j] = 0;

}

}

for (int i = 0; i < 34; i = i + 33)

{

for (int j = 0; j < 34; j++)

{

map[1].mapp[i][j] = 5;

}

}

for (int j = 0; j < 34; j = j + 33)

{

for (int i = 0; i < 34; i++)

{

map[1].mapp[i][j] = 5;

}

}

for (int i = 10; i < 34; i = i + 33)

{

for (int j = 5; j <= 20; j++)

{

map[1].mapp[i][j] = 4;

}

}

for (int i = 20; i < 34; i = i + 33)

{

for (int j = 20; j <= 25; j++)

{

map[1].mapp[i][j] = 4;

}

}

void SignInRank()

{

Rank rankFile[6];

TCHAR name[16];

InputBox(name, 16, \_T("请输入用户名(英文字符15个以内)"));

//outtext(name);

//closegraph();

string user\_name;

user\_name = TCHARTurnString(name);

rankFile[5].name = user\_name;

rankFile[5].score = total.score;

rankFile[5].time = total.alltime;

ifstream infile("../txt/Rank.txt", ios::in);

char title[6];

infile >> title;

//cout << title;

for (int i = 1; i <= 5; i++)

{

int n;

char No[5];

char useless[3][10];

string name;

int score, time;

infile >> No >> n >> useless[0] >> name >> useless[1] >> score >> useless[2] >> time;

//cout << No << n << useless[0] << name << useless[1] << score << useless[2] << time;

rankFile[n-1].name = name;

rankFile[n - 1].score = score;

rankFile[n - 1].time = time;

}

infile.close();

}

for (int i = 3; i < 34; i = i + 33)

{

for (int j = 20; j < 34; j++)

{

map[1].mapp[i][j] = 5;

}

}

for (int j = 0; j < 34; j = j + 33)

{

for (int i = 0; i < 34; i++)

{

map[1].mapp[i][j] = 5;

}

}

for (int i = 10; i < 34; i = i + 33)

{

for (int j = 5; j <= 20; j++)

{

map[1].mapp[i][j] = 4;

}

}

for (int i = 20; i < 34; i = i + 33)

{

for (int j = 20; j <= 25; j++)