**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 | 计算机科学与技术 | 班 级 | 1802 |
| 姓 名 | 邱晓鹏 | 学 号 | 20184552 |

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)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)，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）。

一、项目简介……………………………………………………………………………………4

1、选题目的…………………………………………………………………………………4

2、选题意义…………………………………………………………………………………4

3、任务概况…………………………………………………………………………………4

4、分工………………………………………………………………………………………4

二、项目内容……………………………………………………………………………………4

1、整体结构…………………………………………………………………………………4

2、系统框架图………………………………………………………………………………5

3、UML类图…………………………………………………………………………………5

4、程序流程图………………………………………………………………………………6

5、重点难点分析及解决方案………………………………………………………………8

6、调试难点及解决方法……………………………………………………………………9

7、系统交互界面及结果展示………………………………………………………………9

三、总结与体会…………………………………………………………………………………11

1、综合评价…………………………………………………………………………………11

2、收获与体会………………………………………………………………………………11

3、不足………………………………………………………………………………………12

四、附录…………………………………………………………………………………………13

# 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

3、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）

4、分工：

我负责蛇的状态记录和改变的实现，部分图案设计，图案读取存储。

# 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：

主选单

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：

蛇的控制（wasd，方向键）

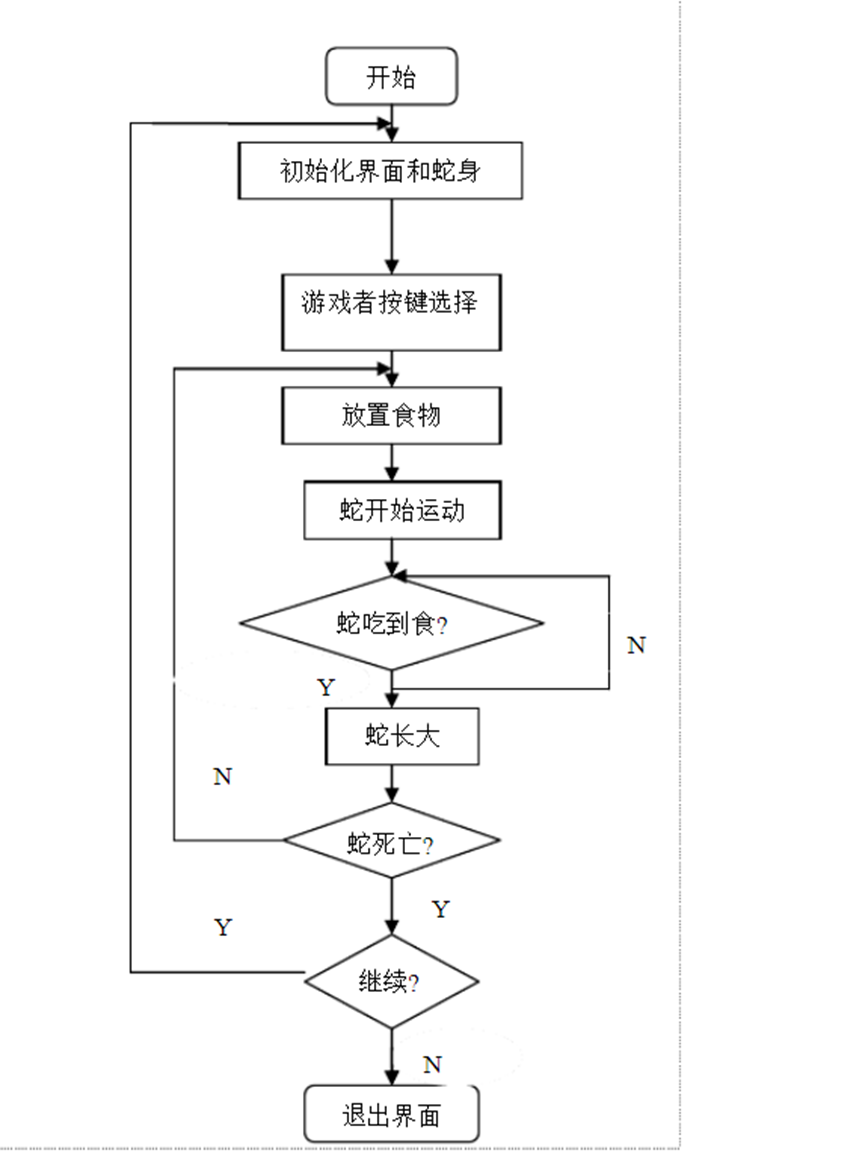
模式设置（经典模式、普通模式、街机模式）

难度设置（低中高）

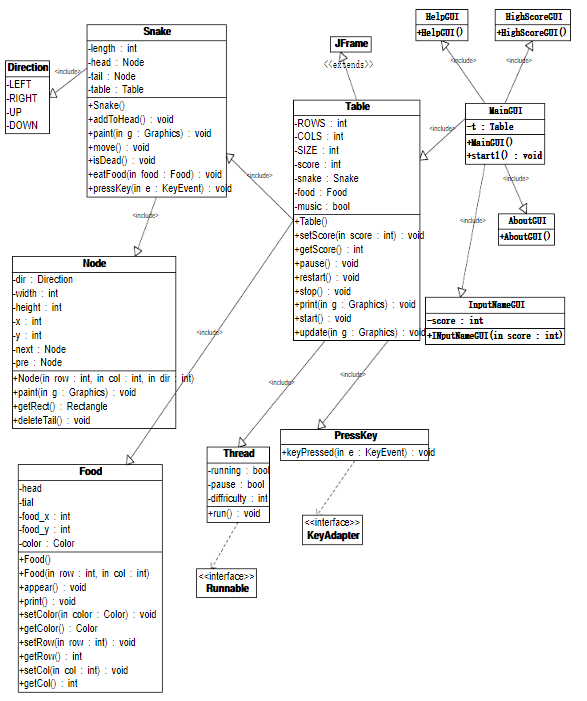
3）积分（每次吃掉一个食物增加5分，并显示出来）

使用easyX来完成图案的绘制，使用QT制作界面。

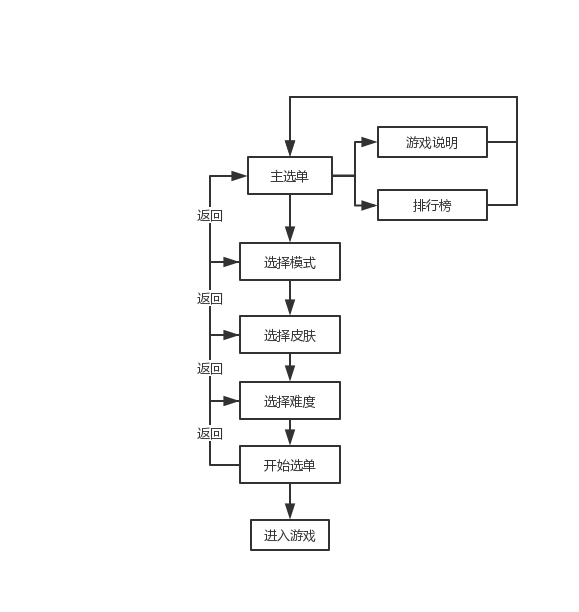
2、系统框架图：

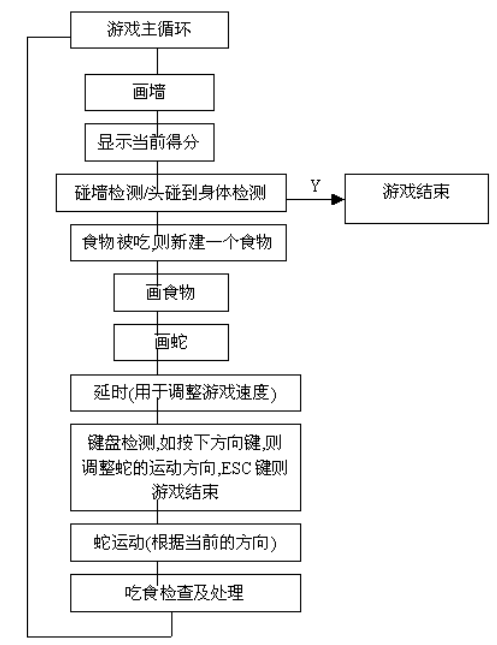


3、UML类图：



4、程序流程图：





5、重点难点分析及解决方案

重点：接受按键的过程。

难点：接受按键的过程

因为游戏是实时进行的，无论是否有按键都要进行游戏，不能出现停顿，显示按键等情况。按键的输入也是无法预测的。同时，一个按键、组合键可能接受到两个返回值，导致出现不可预知的错误。

解决方案：

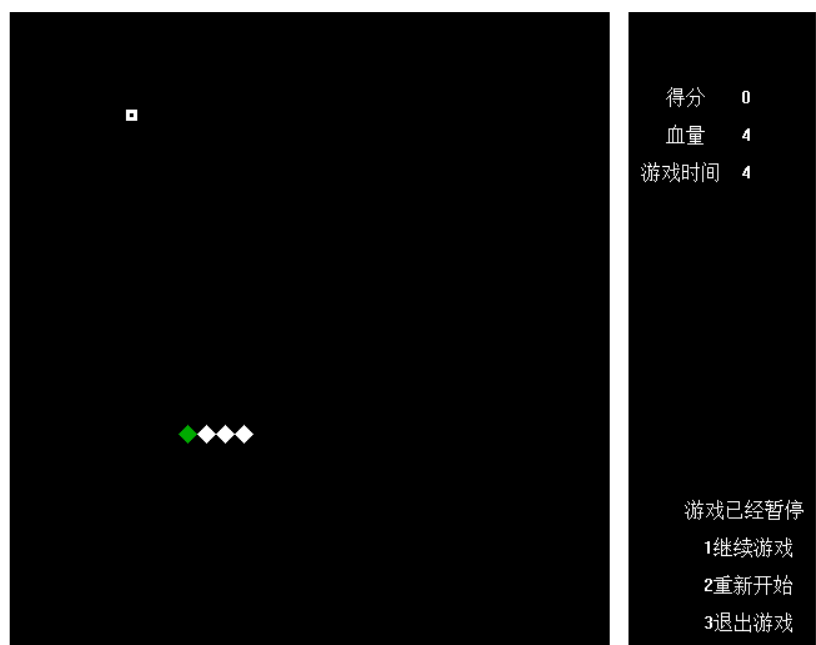
检测是否有键盘按键来决定是否接受，从而避免游戏停顿。对于按键返回值进行查询，针对两个返回值的情况进行第二次接受返回值处理，从而获取正确的按键信息。

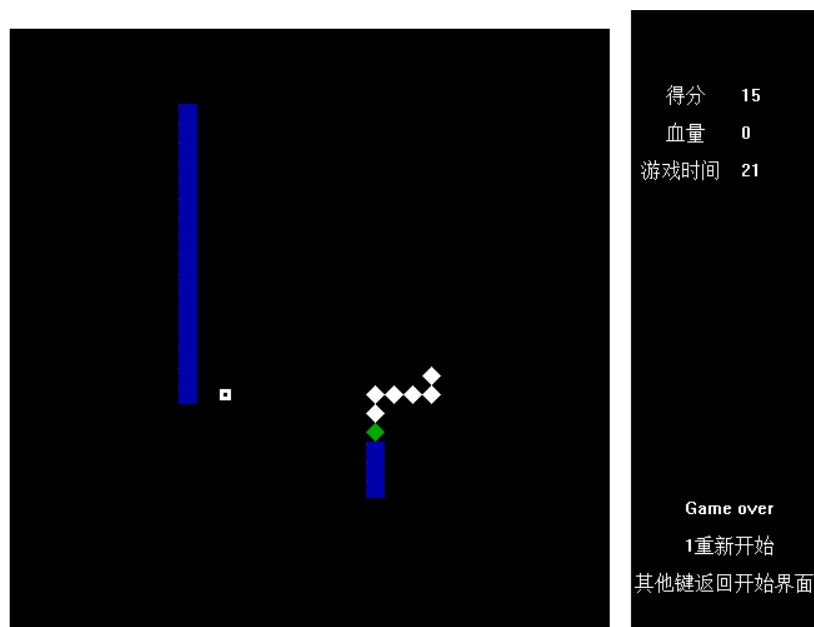
6、调试难点及解决方法

难点：游戏是动态过程，无法将调试过程中的变量信息及时输出，断点调试无法正常游戏，因此很难查看在游戏中的变量情况。

解决方法：对不同的信息采取不同的调试方法。对于实时性要求不高的部分，采取输出到文件的方式，在结束后利用运行过程中产生的信息进行分析，判断程序的运行情况。对于实时性要求高的部分，尽量直接创建对应环境，将变量内容转化为游戏反映的画面上，观察在游戏中的情况推测出变量的实际变化情况。对于无法反映在画面上的内容，设计在想要调试点附近的函数调用等以便设置断点，使其在之前能够正常运行。

7、系统交互及结果界面：









# 三、总结与体会

1、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

合理的头文件管理能更加方便修改。课设期间一共进行了两次大的头文件修改，一次将函数定义全部放到cpp文件中，只在头文件中保留声明，从而使头文件的信息更加简单明了，能迅速找到某个功能实现的位置，方便管理。并且因为头文件中包含所有声明，又避免了调用的函数类等位于下方而导致无法编译的错误，加快调试进度。第二次是将所有头文件统一到一个头文件中，从而解决各个文件之间相互引用的过程中需要不停添加引用头文件的问题，使头文件引用简洁明了，简化无用的引用编码过程，提高效率。

辅助程序有必要尽早创建。因为是游戏，免不了需要提供一些图案等来美化，免不了一些必要的游戏测试过程，更简洁的功能添加甚至自制模式。这些的功能在最后程序中很可能不具备，但有必要在开始时提前创建好，辅助开发过程中的快速调试和快速添加所需要的素材功能。

3、不足

缺少代码版本管理能力。为了基本功能外更多功能的实现，提前编码了很多函数以供备用。在随着开发中的进行，不少地方改变了实现方法。备用函数的选用，不同实现方法的效果测试，不同人手中的代码版本不同，都会导致编码可能产生重复、缺少、难以测试等问题，使开发效率低下。同时因为没有版本管理，出现了问题，难以回退，进一步拖累了开发效率。今后要加强这方面的学习，提高开发的效率。

# 四、附录

Snake的类定义：：

class Snake

{

private:

int len;

list<pos> body;

pos head;

int speed;

direction dir;

int blood;

int maxblood;

int past;

bool ifnext;//下一步是否有效,0无效,1有效

pos next;//下一步位置

int speedpast;

public:

bool havebean;

Snake(pos \_head, int \_blood, int \_maxblood, list<pos> &\_body, int \_speed, int \_len = 3, direction \_dir = d);

bool ifmove();//是否移动,移动时修改ifnext并记录下一格

//以下函数需要在ifmove（）使用后使用

void correctmove();//修正下一步超出地图的现象，修正返回1，否则返回0

void move();//将蛇在地图矩阵上进行修改

bool died(int key = 0);//判断是否死亡，返回1死亡，返回0存活

//bool eatfood();//判断是否吃到食物，吃到返回1，没吃到返回0

//void addlenth();//增加长度，在身体链表中增加一个头部

void eatfood();//吃豆子后改变蛇的数学

direction change\_direction();//从键盘上接受按键，改变方向

direction getdirection();//返回蛇现在的方向

void changespeed(int ss = 0);//改变速度，未做完

bool ifblood();//判断是否撞到可摧毁墙，并扣除血量。如果撞到返回1，否则返回0

int getblood();//返回血量

int addblood(int i = 1);//增加血量

pos gethead();

pos gettail();

pos getfirstbody();

list<pos> getbody();

pos getnext();

void changeblood(int bb);//改变血量

void changemaxblood(int bb);//改变最大血量

};

接受按键的函数：

bool keyboardhit(char &c)//如果接受到按键，返回1，否则返回0。将按键的值存在c中，如果接受到的是两位字节，将进行特殊处理

{

char d;

if (\_kbhit())

{

c = \_getch();

if ((c == 0) || (c == 0xE0) || (c == -32))//如果接受到两位字节的按键

{

d = \_getch();

if (c == -32)

switch (d)

{

case 72:

c = 'w';

break;

case 80:

c = 's';

break;

case 75:

c = 'a';

break;

case 77:

c = 'd';

break;

default:

c = 0;

break;

}

}

//将方向键转化为小写的wasd

if ((c >= 'A') && (c <= 'Z'))

c = c - ('A' - 'a');

return 1;

}

else return 0;

}

改变方向：

direction Snake::change\_direction()

{

char c;

pos p = body.front();

if (keyboardhit(c))

{

switch (c)

{

case 'w':

if (!((head.x == p.x) && ((head.y - 1) == p.y)))

dir = w;

break;

case 'a':

if (!(((head.x - 1) == p.x) && (head.y == p.y)))

dir = a;

break;

case 's':

if (!((head.x == p.x) && ((head.y + 1) == p.y)))

dir = s;

break;

case 'd':

if (!(((head.x + 1) == p.x) && (head.y == p.y)))

dir = d;

break;

case (char)27:

pause();

break;

default:

break;

}

}

return dir;

}

提前准备的函数：

void print(COLORREF photo[][\_proportion], pos p)//在p点打印图案

{

COORD coord = postocoord(p);

int i, k;

for (i = coord.X; i < coord.X + 15; i++)

{

for (k = coord.Y; k < coord.Y + 15; k++)

{

putpixel(i, k, photo[i - coord.X][k - coord.Y]);

}

}

}