**贪吃蛇小程序**

**课程名称：面向对象程序设计**

**姓名：张宇**

**院系：电子与信息工程学院计算机科学与技术专业**

**班级：计算机科学与技术1班**

**学号：2251745**

**完成日期：2023年12月27日**

**1、设计思路与功能描述**

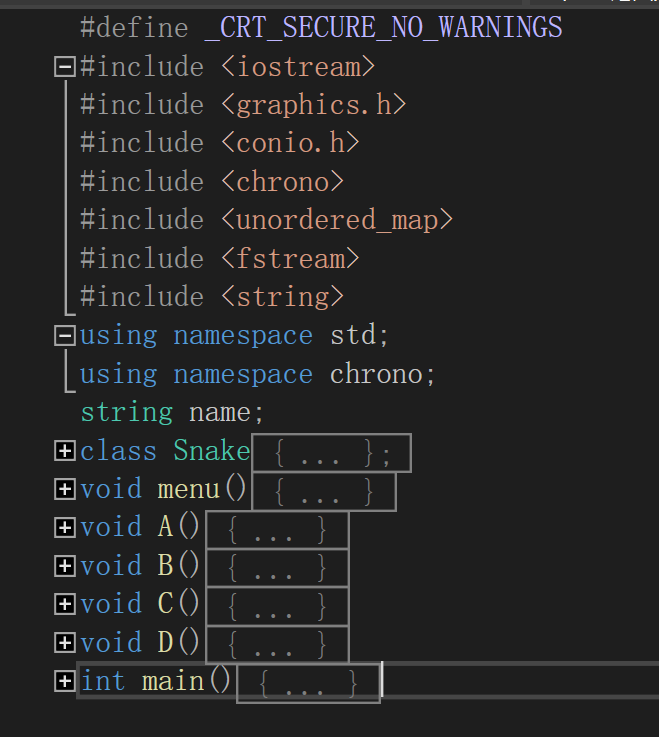
在刚拿到这个题目的时候还是处于一个比较懵的状态，毕竟是需要做一个相对完善的小游戏啦，虽然说大一高程课也做个小游戏，但是高程课是把一个小游戏给拆分成了很多个功能，然后我们按照他的步骤完成一个个的功能后，就自然而然地完成了一个小游戏的设计。但是这次的作业基本没有给什么思路，是需要我们自己把一个完整的小游戏自行拆分成各个子部分，然后再逐步完成，这其实并不是一个轻松的工作。

现在回想一下其实也感觉并没有什么设计的思路，也就是按部就班地一点点写，只能说走一步看一步吧。一开始先使用EasyX库设计页面的初始化，将基本的地图显示出来，然后再随机出现一些果实，接着是蛇身的显示，在显示了蛇身后就需要考虑蛇的移动啦，然后就是需要实现蛇在移动的过程中如果接触到例如果实、炸弹、墙壁、蛇身等物体所触发的事件。在完成这些最基本的东西以后，其实一个简单的贪吃蛇就已经完成了，后面要做的就是根据题目的要求添加一些计时、计分的东西啦，就是关于页面的设计，还有就是关于历史记录的存储，同时还要考虑有关历史记录的增添改查的操作等等。

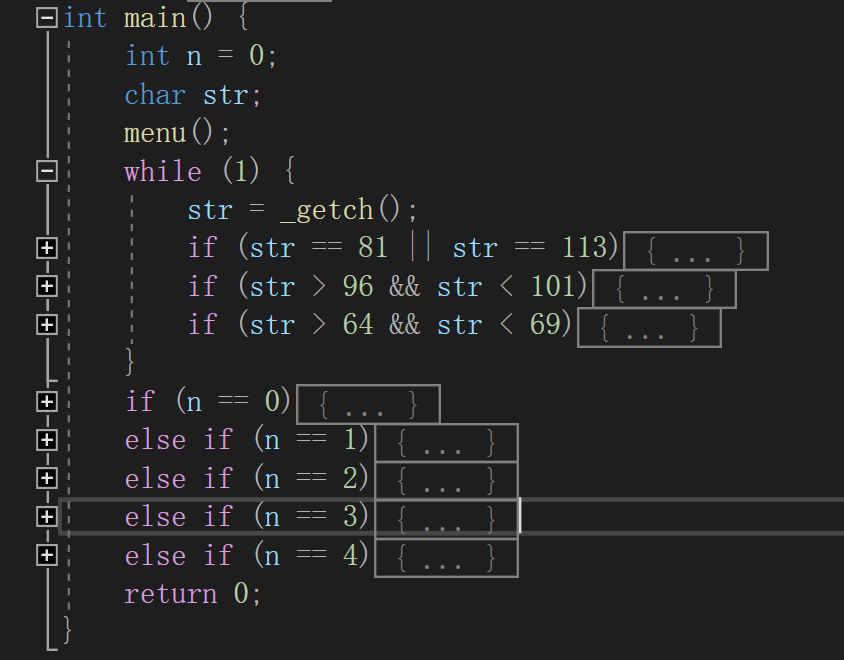
以上就是最基本的入门版贪吃蛇的实现了，紧接着是根据高级版、进阶版等版本的更高要求对代码进行补充调整，完成了三个版本以后就是对于加分项的尽力而为啦，前面几个加分项的内容还是比较实现的，也只需要对于代码做出简单的修改也就完成了，至于后续的联机对抗等的实现，考虑到临近期末，各种学习压力还是蛮大的，也就暂时搁置了。

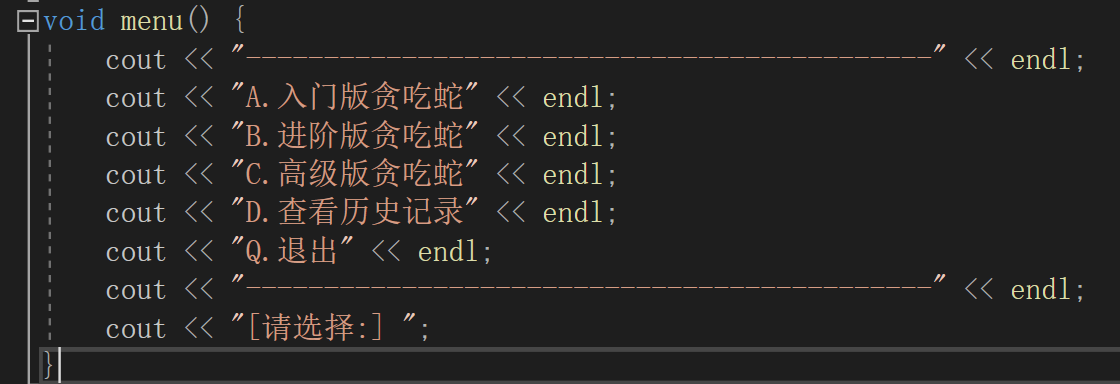
基本完成了目标的作业后，我就邀请了室友、同学等帮我进行代码的调试工作，就是不断地玩这个游戏，来找一些可能会出现的bug或者游戏不完善的地方，在这个过程中我也收获了很多，比如我增添了空格键暂停游戏的功能，以及在第二个版本中因为长时间无法结束游戏，就添加了一个Q键结束游戏的功能。此外也修复了很多代码中的小bug，我也明白了确实只有在玩的过程中才会发现代码中存在的各种各样的小问题。

这是我本题用到的函数总览。

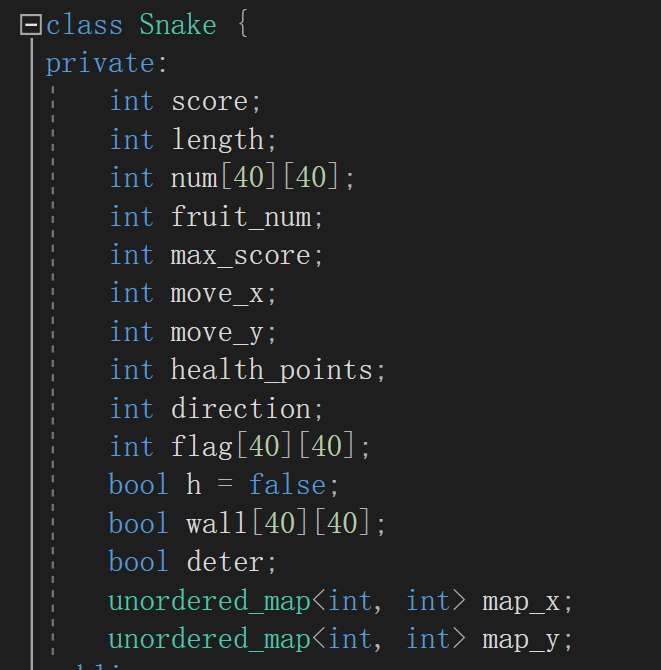


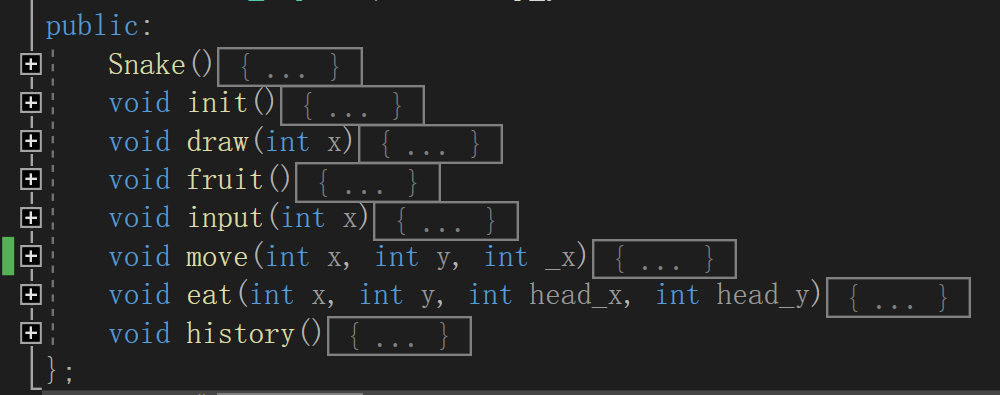
这是我本题中的main函数和菜单函数的内容， 里面主要是菜单的打印，以及供用户选择游戏模式，用户名的输入，或者查看游戏的历史记录等等功能。





接下来是我本次程序中最重要的Snake类了，这里展示了该类中包含的相关变量以及用到的各种函数，为了节省篇幅，这里就不对这些函数进行一一展示，在本报告的第4部分的源代码展示中可以展开每个函数的功能以及具体实现方法。 其中init()函数是初始化函数，用于对类中的变量进行初始化赋值的操作；draw()函数是画图函数，用于页面中各种细节的显示；fruit()函数是随机显示果实、能量果和炸弹的函数；input()函数是读入键盘输入的函数，同时还用于将游戏记录写入历史存到；move()函数是用来控制蛇移动的函数，同时也会对当前位置是否发生事件以及发生哪种事件进行判断；eat()函数是用来吃果子的函数，同时会根据当前页面中果子剩余量判断是否需要再生成果实；history()函数是用来读取历史记录，并找到历史最高分的函数。





**2、在实验过程中遇到的问题与解决方法**

（1）问题：在写代码的过程中我意识到如果要完成一个很完善的游戏其实是一件很麻烦的事情，就拿对于历史记录的增删改查来说吧，这里要求有较多的输入以及输入提示来确保程序的可用性，保证用户体验，这里就要考虑到很多非法输入等等问题，就让事情愈发复杂起来。

解决方法：对于上述问题，我只能说尽量地保证用户的体验感，但是其中肯定还存在这一些问题，就拿对历史记录的增删改查来说吧，对于每次操作后，都应该对于用户进行询问是否要继续改，但是考虑到这只是本次游戏中很小的一方面，于是就对问题进行了简化，本次操作后程序都会结束，只能保证对历史记录的一次操作，若想对于其反复操作也只能不断的重新运行代码。

（2）问题：对于人机对战以及后续的多人对战问题毫无思路。

解决方法：由于临近期末，学习压力比较大，再加上对于人机对战以及后续的多人对战问题没有涉及，所以也就搁置了，如果后面周末有时间可能会再加以完善吧。

**3、心得体会**

本次作业实现的贪吃蛇小游戏并没有很难，所需要用到的东西我们基本也都使用过，个人感觉只是比较复杂，需要实现的东西比较多，同时也没有给出设计思路，这个只能自己来构思完成，可能算是一个难点吧。不能不说，写的时候还是很有感觉的，也并没有像之前那样那么多在写作业的感觉，毕竟是在做个小游戏，做完以后我还额外进行了用户试玩以及体验反馈，同时再根据反馈对我的代码和游戏功能进行调整，提高用户的体验感。最终写完代码也是很有成就感的，虽然只是贪吃蛇这样一个简单的小游戏，同时也体会到了制作游戏的不易，想要做一个完善的游戏就需要各个方面都考虑到，就很繁琐，但因为自己独立完成了一个小游戏，也并不会觉得自己不行。因为临近期末等各种各样的原因，致使我后续的人机对战以及多人对抗等模式没有得以实现，还是很遗憾的，以后有时间我还是很考虑继续完善这次作业的代码，可以真正称得上是自己做的一个有可玩性的小游戏啦。

写了很多次大作业后感觉到其实每次大作业看着都很唬人，尤其是没有循序渐进的过程时，总觉得自己好像做不出来，掌握的知识不够解决问题，但其实只有自己动起手来，一点一点地完成代码，才能知道自己的不足，也才有学习的方向和动力，一个个小功能的实现，一个个函数的编码，最终叠加起来，原来一个看似不可能的大作业也就一步步的解决了。

**4、源代码**

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

#include <graphics.h>

#include <conio.h>

#include <chrono>

#include <unordered\_map>

#include <fstream>

#include <string>

using namespace std;

using namespace chrono;

string name;

class Snake {

private:

int score;

int length;

int num[40][40];

int fruit\_num;

int max\_score;

int move\_x;

int move\_y;

int health\_points;

int direction;

int flag[40][40];

bool h = false;

bool wall[40][40];

bool deter;

unordered\_map<int, int> map\_x;

unordered\_map<int, int> map\_y;

public:

Snake() {

history();

init();

}

void init() {

for (int i = 0; i < 40; i++) {

for (int j = 0; j < 40; j++) {

num[i][j] = 0;

flag[i][j] = -1;

wall[i][j] = false;

if (j == 39 || j == 0 || i == 0 || i == 39) {

wall[i][j] = true;

}

}

}

num[19][19] = 2;

num[20][19] = 1;

map\_x[1] = 20;

map\_y[1] = 19;

map\_x[2] = 19;

map\_y[2] = 19;

score = 0;

length = 2;

fruit\_num = 0;

move\_x = 0;

move\_y = 0;

health\_points = -1;

direction = -1;

deter = true;

}

void draw(int x) {

for (int i = 0; i < 40; i++) {

fillrectangle(10, 10 + 12 \* i, 20, 20 + 12 \* i);

fillrectangle(10 + 12 \* i, 10, 20 + 12 \* i, 20);

fillrectangle(478, 10 + 12 \* i, 488, 20 + 12 \* i);

fillrectangle(10 + 12 \* i, 478, 20 + 12 \* i, 488);

}

setfillcolor(CYAN);

fillcircle(15 + 12 \* 20, 15 + 12 \* 19, 5);

fillcircle(15 + 12 \* 19, 15 + 12 \* 19, 3);

fruit();

setfillcolor(BROWN);

solidrectangle(500, 10, 654, 15);

solidrectangle(500, 10, 505, 210);

solidrectangle(649, 10, 654, 215);

solidrectangle(500, 210, 654, 215);

setfillcolor(WHITE);

solidrectangle(505, 15, 512, 210);

solidrectangle(505, 15, 649, 22);

solidrectangle(642, 15, 649, 210);

solidrectangle(505, 203, 649, 210);

solidrectangle(505, 109, 649, 116);

LOGFONT f;

gettextstyle(&f); // 获取当前字体设置

f.lfHeight = 28; // 设置字体高度为 28

\_tcscpy(f.lfFaceName, \_T("黑体")); // 设置字体为“黑体”(高版本 VC 推荐使用 \_tcscpy\_s 函数)

f.lfQuality = ANTIALIASED\_QUALITY; // 设置输出效果为抗锯齿

settextstyle(&f); // 设置字体样式

settextcolor(YELLOW);

outtextxy(516, 120, \_T("本局得分:"));

outtextxy(516, 26, \_T("历史最高:"));

outtextxy(90, 510, \_T("加速区"));

outtextxy(330, 510, \_T("减速区"));

if (x == 3) {

outtextxy(510, 250, \_T("Length:"));

outtextxy(610, 250, \_T("2"));

outtextxy(500, 300, \_T("HP:"));

IMAGE img;

loadimage(&img, \_T("HP.png"), 30, 30);

putimage(535, 300, &img);

putimage(560, 300, &img);

putimage(585, 300, &img);

putimage(610, 300, &img);

putimage(635, 300, &img);

}

else {

outtextxy(510, 270, \_T("Length:"));

outtextxy(610, 270, \_T("2"));

}

outtextxy(510, 350, \_T("Time:00:00"));

f.lfHeight = 20;

settextstyle(&f);

settextcolor(RED);

outtextxy(510, 392, \_T("按Q键结束游戏"));

outtextxy(500, 430, \_T("按空格键暂停游戏"));

f.lfHeight = 48;

settextstyle(&f);

settextcolor(YELLOW);

outtextxy(526, 152, \_T("0"));

outtextxy(553, 152, \_T("0"));

outtextxy(580, 152, \_T("0"));

outtextxy(607, 152, \_T("0"));

outtextxy(526, 58, \_T("0"));

outtextxy(553, 58, \_T("0"));

outtextxy(580, 58, \_T("0"));

outtextxy(607, 58, \_T("0"));

if (max\_score < 100) {

TCHAR s[5];

\_stprintf(s, \_T("%d"), max\_score / 10);

outtextxy(580, 58, s);

}

else if (max\_score < 1000) {

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), max\_score % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), max\_score / 100);

outtextxy(553, 58, s);

}

else {

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), max\_score % 1000 % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), max\_score % 1000 / 100);

outtextxy(553, 58, s);

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), max\_score / 1000);

outtextxy(526, 58, \_s);

}

}

void fruit() {

IMAGE img;

loadimage(&img, \_T("apple.jpg"), 10, 10);

srand(static\_cast<unsigned int>(time(nullptr)));

int new\_num= rand() % 5 + 1;

fruit\_num += new\_num;

for (int i = 0; i < new\_num; i++) {

while (1) {

int x = rand() % 38 + 1;

int y = rand() % 38 + 1;

if (flag[x][y] == -1 && !num[x][y]) {

flag[x][y] = 0;

putimage(10 + 12 \* x, 10 + 12 \* y, &img);

break;

}

}

}

int bomb = rand() % 3;

if (bomb == 0) {

IMAGE bomb\_img;

loadimage(&bomb\_img, \_T("bomb.png"), 10, 10);

while (1) {

int bomb\_x = rand() % 38 + 1;

int bomb\_y = rand() % 38 + 1;

if (flag[bomb\_x][bomb\_y] == -1 && !num[bomb\_x][bomb\_y]) {

flag[bomb\_x][bomb\_y] = 2;

putimage(10 + 12 \* bomb\_x, 10 + 12 \* bomb\_y, &bomb\_img);

break;

}

}

}

int orange = rand() % 2;

if (orange == 0) {

fruit\_num++;

IMAGE orange\_img;

loadimage(&orange\_img, \_T("orange.png"), 10, 10);

while (1) {

int orange\_x = rand() % 38 + 1;

int orange\_y = rand() % 38 + 1;

if (flag[orange\_x][orange\_y] == -1 && !num[orange\_x][orange\_y]) {

flag[orange\_x][orange\_y] = 1;

putimage(10 + 12 \* orange\_x, 10 + 12 \* orange\_y, &orange\_img);

break;

}

}

}

}

void input(int x) {

if (x == 3) {

health\_points = 5;

}

auto start = high\_resolution\_clock::now();

int second = 0;

while (deter) {

auto current = high\_resolution\_clock::now();

auto duration\_m = duration\_cast<minutes>(current - start);

auto duration = duration\_cast<seconds>(current - start);

if (duration.count() < 60) {

second = (int)duration.count();

int du\_s = (int)duration.count();

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

TCHAR s[5];

\_stprintf(s, \_T("%d"), du\_s);

if (du\_s < 10) {

outtextxy(636, 350, s);

}

else {

outtextxy(622, 350, s);

}

}

else {

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

second = (int)duration.count();

int d\_s = (int)duration.count() % 60;

int d\_m = (int)duration.count() / 60;

TCHAR s[5];

\_stprintf(s, \_T("%d"), d\_s);

if (d\_s < 10) {

outtextxy(622, 350, \_T("0"));

outtextxy(636, 350, s);

}

else {

outtextxy(622, 350, s);

}

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), d\_m);

if (d\_m < 10) {

outtextxy(593, 350, \_s);

}

else {

outtextxy(580, 350, \_s);

}

}

bool fl = false;

if (\_kbhit()) {

fl = true;

char key = \_getch();

if ((((move\_x == 0 && move\_y == -1) || direction == 0) && (key == 's' || key == 'S')) ||

(((move\_x == 0 && move\_y == 1) || direction == 1) && (key == 'w' || key == 'W')) ||

(((move\_x == -1 && move\_y == 0) || direction == 2) && (key == 'd' || key == 'D')) ||

(((move\_x == 1 && move\_y == 0) || direction == 3) && (key == 'a' || key == 'A')) ||

(!h && (key == 'a' || key == 'A'))) {

}

else {

switch (key) {

case 'w':

move\_x = 0;

move\_y = -1;

break;

case 's':

move\_x = 0;

move\_y = 1;

break;

case 'a':

move\_x = -1;

move\_y = 0;

break;

case 'd':

move\_x = 1;

move\_y = 0;

break;

case 'q':

deter = false;

break;

case 'W':

move\_x = 0;

move\_y = -1;

break;

case 'S':

move\_x = 0;

move\_y = 1;

break;

case 'A':

move\_x = -1;

move\_y = 0;

break;

case 'D':

move\_x = 1;

move\_y = 0;

break;

case 'Q':

deter = false;

break;

case 32:

move\_x = 0;

move\_y = 0;

break;

}

}

}

if (move\_x || move\_y) {

move(move\_x, move\_y, x);

int head\_x = map\_x.find(1)->second + move\_x;

if (head\_x < 20) {

Sleep(50);

}

else {

Sleep(100);

}

}

}

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(RED);

outtextxy(515, 460, \_T("Game Over!"));

ofstream file("history.txt", ios::app);

if (file.is\_open()) {

if (x == 1) {

file << "版本：入门版 用户名：" << name << " 得分：" << score << " 用时：" << second << "s" << endl;

}

else if (x == 2) {

file << "版本：进阶版 用户名：" << name << " 得分：" << score << " 用时：" << second << "s" << endl;

}

else if (x == 3) {

file << "版本：高级版 用户名：" << name << " 得分：" << score << " 用时：" << second << "s" << endl;

}

file.close();

}

}

void move(int x, int y, int \_x) {

h = true;

direction = -1;

auto it\_x = map\_x.find(length), it\_y = map\_y.find(length);

num[it\_x->second][it\_y->second] = 0;

setfillcolor(BLACK);

solidrectangle(10 + 12 \* it\_x->second, 10 + 12 \* it\_y->second, 20 + 12 \* it\_x->second, 20 + 12 \* it\_y->second);

int tail\_x = map\_x.find(length - 1)->second, tail\_y = map\_y.find(length - 1)->second;

solidrectangle(10 + 12 \* tail\_x, 10 + 12 \* tail\_y, 20 + 12 \* tail\_x, 20 + 12 \* tail\_y);

setfillcolor(CYAN);

fillcircle(15 + 12 \* tail\_x, 15 + 12 \* tail\_y, 3);

map\_x[length] = tail\_x;

map\_y[length] = tail\_y;

int head\_x = map\_x.find(1)->second + x, head\_y = map\_y.find(1)->second + y;

setfillcolor(BLACK);

solidrectangle(10 + 12 \* map\_x.find(1)->second, 10 + 12 \* map\_y.find(1)->second, 20 + 12 \* map\_x.find(1)->second, 20 + 12 \* map\_y.find(1)->second);

setfillcolor(CYAN);

solidcircle(15 + 12 \* map\_x.find(1)->second, 15 + 12 \* map\_y.find(1)->second, 5);

map\_x[1] = head\_x;

map\_y[1] = head\_y;

num[head\_x][head\_y] = 1;

fillcircle(15 + 12 \* head\_x, 15 + 12 \* head\_y, 5);

for (int i = 1; i < length - 1; i++) {

num[map\_x.find(length - i)->second][map\_y.find(length - i)->second]++;

if (i == length - 2) {

map\_x[length - i] = map\_x.find(length - i - 1)->second - x;

map\_y[length - i] = map\_y.find(length - i - 1)->second - y;

}

else {

map\_x[length - i] = map\_x.find(length - i - 1)->second;

map\_y[length - i] = map\_y.find(length - i - 1)->second;

}

}

if (flag[head\_x][head\_y] == 0) {

flag[head\_x][head\_y] = -1;

eat(x, y, head\_x, head\_y);

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

TCHAR s[5];

\_stprintf(s, \_T("%d"), length);

if (\_x == 3) {

outtextxy(610, 250, s);

}

else {

outtextxy(610, 270, s);

}

score += 10;

if (score > max\_score) {

if (score < 100) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 10);

outtextxy(580, 58, s);

}

else if (score < 1000) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 100);

outtextxy(553, 58, s);

}

else {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 1000 % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score % 1000 / 100);

outtextxy(553, 58, s);

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), score / 1000);

outtextxy(526, 58, \_s);

}

}

if (score < 100) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 10);

outtextxy(580, 152, s);

}

else if (score < 1000) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 100 / 10);

outtextxy(580, 152, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 100);

outtextxy(553, 152, s);

}

else {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 1000 % 100 / 10);

outtextxy(580, 152, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score % 1000 / 100);

outtextxy(553, 152, s);

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), score / 1000);

outtextxy(526, 152, \_s);

}

}

else if (flag[head\_x][head\_y] == 1) {

flag[head\_x][head\_y] = -1;

eat(x, y, head\_x, head\_y);

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

TCHAR s[5];

\_stprintf(s, \_T("%d"), length);

if (\_x == 3) {

outtextxy(610, 250, s);

}

else {

outtextxy(610, 270, s);

}

score += 100;

if (score > max\_score) {

if (score < 100) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 10);

outtextxy(580, 58, s);

}

else if (score < 1000) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 100);

outtextxy(553, 58, s);

}

else {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 1000 % 100 / 10);

outtextxy(580, 58, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score % 1000 / 100);

outtextxy(553, 58, s);

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), score / 1000);

outtextxy(526, 58, \_s);

}

}

if (score < 100) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 10);

outtextxy(580, 152, s);

}

else if (score < 1000) {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 100 / 10);

outtextxy(580, 152, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score / 100);

outtextxy(553, 152, s);

}

else {

f.lfHeight = 48;

settextstyle(&f);

TCHAR s\_[5];

\_stprintf(s\_, \_T("%d"), score % 1000 % 100 / 10);

outtextxy(580, 152, s\_);

TCHAR s[5];

\_stprintf(s, \_T("%d"), score % 1000 / 100);

outtextxy(553, 152, s);

TCHAR \_s[5];

\_stprintf(\_s, \_T("%d"), score / 1000);

outtextxy(526, 152, \_s);

}

if (\_x == 3 && health\_points<= 5) {

IMAGE HP\_img;

loadimage(&HP\_img, \_T("HP.png"), 30, 30);

putimage(535 + 25 \* health\_points, 300, &HP\_img);

health\_points++;

}

}

else if (flag[head\_x][head\_y] == 2) {

deter = false;

}

if (num[head\_x][head\_y] > 1 || wall[head\_x][head\_y]) {

if (\_x == 1) {

deter = false;

}

else if (\_x == 2) {

for (int i = 0; i < 40; i++) {

for (int j = 0; j < 40; j++) {

if (num[i][j]) {

setfillcolor(BLACK);

solidrectangle(10 + 12 \* i, 10 + 12 \* j, 20 + 12 \* i, 20 + 12 \* j);

setfillcolor(WHITE);

fillrectangle(10 + 12 \* i, 10 + 12 \* j, 20 + 12 \* i, 20 + 12 \* j);

wall[i][j] = true;

num[i][j] = 0;

}

}

}

for (int i = 1; i <= length; i++) {

map\_x.erase(i);

map\_y.erase(i);

}

length = 2;

move\_x = 0;

move\_y = 0;

fruit();

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

outtextxy(610, 270, \_T(" "));

TCHAR s[5];

\_stprintf(s, \_T("%d"), length);

outtextxy(610, 270, s);

int cou = 0;

while (1) {

cou++;

head\_x = rand() % 38 + 1;

head\_y = rand() % 38 + 1;

int a = rand() % 4;

switch (a) {

case 0:

tail\_x = head\_x;

tail\_y = head\_y + 1;

break;

case 1:

tail\_x = head\_x;

tail\_y = head\_y - 1;

break;

case 2:

tail\_x = head\_x + 1;

tail\_y = head\_y;

break;

case 3:

tail\_x = head\_x - 1;

tail\_y = head\_y;

break;

}

if (!wall[head\_x][head\_y] && !wall[tail\_x][tail\_y]) {

num[head\_x][head\_y] = 1;

num[tail\_x][tail\_y] = 2;

map\_x[1] = head\_x;

map\_y[1] = head\_y;

map\_x[2] = tail\_x;

map\_y[2] = tail\_y;

setfillcolor(CYAN);

fillcircle(15 + 12 \* head\_x, 15 + 12 \* head\_y, 5);

fillcircle(15 + 12 \* tail\_x, 15 + 12 \* tail\_y, 3);

break;

}

if (cou > 100) {

deter = false;

break;

}

}

}

else if (\_x == 3) {

health\_points--;

setfillcolor(BLACK);

solidrectangle(535 + 25 \* health\_points, 300, 560 + 25 \* health\_points, 350);

if (!health\_points) {

deter = false;

}

else {

for (int i = 0; i < 40; i++) {

for (int j = 0; j < 40; j++) {

if (num[i][j]) {

setfillcolor(BLACK);

solidrectangle(10 + 12 \* i, 10 + 12 \* j, 20 + 12 \* i, 20 + 12 \* j);

if (!wall[i][j]) {

IMAGE img;

loadimage(&img, \_T("apple.jpg"), 10, 10);

fruit\_num++;

putimage(10 + 12 \* i, 10 + 12 \* j, &img);

flag[i][j] = 0;

}

else {

setfillcolor(WHITE);

fillrectangle(10 + 12 \* i, 10 + 12 \* j, 20 + 12 \* i, 20 + 12 \* j);

}

num[i][j] = 0;

}

}

}

for (int i = 1; i <= length; i++) {

map\_x.erase(i);

map\_y.erase(i);

}

length = 2;

move\_x = 0;

move\_y = 0;

fruit();

LOGFONT f;

gettextstyle(&f);

f.lfHeight = 28;

settextstyle(&f);

settextcolor(YELLOW);

outtextxy(610, 250, \_T(" "));

TCHAR s[5];

\_stprintf(s, \_T("%d"), length);

outtextxy(610, 250, s);

int cou = 0;

while (1) {

cou++;

head\_x = rand() % 38 + 1;

head\_y = rand() % 38 + 1;

direction = rand() % 4;

switch (direction) {

case 0:

tail\_x = head\_x;

tail\_y = head\_y + 1;

break;

case 1:

tail\_x = head\_x;

tail\_y = head\_y - 1;

break;

case 2:

tail\_x = head\_x + 1;

tail\_y = head\_y;

break;

case 3:

tail\_x = head\_x - 1;

tail\_y = head\_y;

break;

}

if (!wall[head\_x][head\_y] && !wall[tail\_x][tail\_y]) {

num[head\_x][head\_y] = 1;

num[tail\_x][tail\_y] = 2;

map\_x[1] = head\_x;

map\_y[1] = head\_y;

map\_x[2] = tail\_x;

map\_y[2] = tail\_y;

setfillcolor(CYAN);

fillcircle(15 + 12 \* head\_x, 15 + 12 \* head\_y, 5);

fillcircle(15 + 12 \* tail\_x, 15 + 12 \* tail\_y, 3);

break;

}

if (cou > 100) {

deter = false;

break;

}

}

}

}

}

}

void eat(int x, int y, int head\_x, int head\_y) {

auto it\_x = map\_x.find(length), it\_y = map\_y.find(length);

setfillcolor(CYAN);

solidcircle(15 + 12 \* it\_x->second, 15 + 12 \* it\_y->second, 5);

auto i\_x = map\_x.find(length - 1), i\_y = map\_y.find(length - 1);

if (it\_x->second == i\_x->second) {

int tail\_x = it\_x->second;

int tail\_y = 2 \* it\_y->second - i\_y->second;

fillcircle(15 + 12 \* tail\_x, 15 + 12 \* tail\_y, 3);

map\_x[length + 1] = tail\_x;

map\_y[length + 1] = tail\_y;

num[tail\_x][tail\_y] = length + 1;

}

else {

int tail\_x = 2 \* it\_x->second - i\_x->second;

int tail\_y = it\_y->second;

fillcircle(15 + 12 \* tail\_x, 15 + 12 \* tail\_y, 3);

map\_x[length + 1] = tail\_x;

map\_y[length + 1] = tail\_y;

num[tail\_x][tail\_y] = length + 1;

}

fruit\_num--;

length++;

if (!fruit\_num) {

fruit();

}

}

void history() {

ifstream file("history.txt");

if (file.is\_open()) {

string line;

while (getline(file, line)) {

if (line.find("得分：") != string::npos) {

size\_t pos = line.find("得分：");

int current = stoi(line.substr(pos + 6));

max\_score = max(max\_score, current);

}

}

file.close();

}

else {

max\_score = 0;

}

}

};

void menu() {

cout << "--------------------------------------------" << endl;

cout << "A.入门版贪吃蛇" << endl;

cout << "B.进阶版贪吃蛇" << endl;

cout << "C.高级版贪吃蛇" << endl;

cout << "D.查看历史记录" << endl;

cout << "Q.退出" << endl;

cout << "--------------------------------------------" << endl;

cout << "[请选择:] ";

}

void A() {

initgraph(666, 550);

Snake snake\_A;

snake\_A.draw(1);

snake\_A.input(1);

Sleep(2000);

}

void B() {

initgraph(666, 550);

Snake snake\_B;

snake\_B.draw(2);

snake\_B.input(2);

Sleep(2000);

}

void C() {

initgraph(666, 550);

Snake snake\_C;

snake\_C.draw(3);

snake\_C.input(3);

Sleep(2000);

}

void D() {

ifstream file("history.txt");

if (!file.is\_open()) {

cout << "暂无历史记录" << endl;

exit(1);

}

string line;

while (getline(file, line)) {

cout << line << endl;

}

file.close();

cout << "是否要对历史记录进行删改(1表示是，0表示否)：";

int n;

cin >> n;

if (n == 1) {

cout << "删除记录请按1，修改用户名请按2，查找用户名请按3，退出请按0：";

int nu;

cin >> nu;

if (nu == 1) {

cout << "请输入要删除的记录所在行：";

int l;

cin >> l;

ifstream file("history.txt");

vector<string> lines;

string line;

while (getline(file, line)) {

lines.push\_back(line);

}

file.close();

int count = 1;

ofstream out("history.txt");

for (const string& save : lines) {

if (count != l) {

out << save << '\n';

}

count++;

}

out.close();

cout << "删除完成，程序结束" << endl;

}

else if (nu == 2) {

string x;

getline(cin, x);

cout << "请输入要修改的用户名：";

string fin;

getline(cin, fin);

cout << "请输入修改后的用户名：";

string \_fin;

getline(cin, \_fin);

bool fi = false;

ifstream file("history.txt");

vector<string> lines;

string line;

while (getline(file, line)) {

if (line.find(fin) != string::npos) {

fi = true;

size\_t pos = line.find(fin);

line.replace(pos, fin.length(), \_fin);

}

lines.push\_back(line);

}

file.close();

ofstream out("history.txt");

for (const string& save : lines) {

out << save << '\n';

}

out.close();

if (!fi) {

cout << "未查询到该用户名" << endl;

}

else {

cout << "用户名修改成功" << endl;

}

}

else if (nu == 3) {

string x;

getline(cin, x);

cout << "请输入要查找的用户名：";

string fin;

getline(cin, fin);

ifstream file("history.txt");

string line;

bool fi = false;

while (getline(file, line)) {

if (line.find(fin) != string::npos) {

fi = true;

cout << line << endl;

}

}

file.close();

if (!fi) {

cout << "未查询到该用户名" << endl;

}

}

}

}

int main() {

int n = 0;

char str;

menu();

while (1) {

str = \_getch();

if (str == 81 || str == 113) {

break;

}

if (str > 96 && str < 101) {

n = str - 96;

cout << str << endl;

break;

}

if (str > 64 && str < 69) {

n = str - 64;

cout << str << endl;

break;

}

}

if (n == 0) {

if (str == 113) {

cout << "q";

}

else {

cout << "Q";

}

}

else if (n == 1) {

cout << "请输入你的用户名：";

getline(cin, name);

A();

}

else if (n == 2) {

cout << "请输入你的用户名：";

getline(cin, name);

B();

}

else if (n == 3) {

cout << "请输入你的用户名：";

getline(cin, name);

C();

}

else if (n == 4) {

D();

}

return 0;

}