C++/项目

贪吃蛇第一版(2021.3.21)

```
/*
* 版本：贪吃蛇第一版
* 完善：1.完善了清屏
*       2.完善了交互界面
*       3.修复了 可以掉头 的bug
*       4.修复了 可以撞到自己 的bug
*
* bug：1.重新开始游戏没有进行空间的释放
*      2.一开始蛇的前进方向已经确定
*      3.按两个键会导致游戏结束
* 时间：2021.3.21
* 作者：orall
*/

//关键 GetTickCount() 和 kbhit()

#include <iostream>
#include <stdlib.h>
#include <time.h>
#include <conio.h>
#include <windows.h>
using namespace std;
int map[20][20] = { 0 };         //地图（0表示的是空，1表示的是蛇，2表示的是食物）
int snake_len = 2;               //表示蛇的长度
int h_x, h_y, t_x, t_y;          //头位置

void game();                     //游戏主体
void move(int x, int y);         //移动蛇身
void show();                     //显示地图
void show2();                    //在地图显示蛇身
void setfood();                  //放置食物
void Loading();                  //加载界面
void Setup();                    //设置界面
void gotoxy(int x, int y, int width);    //光标到指定的位置

typedef struct Snake {
    int x, y;
    Snake* next;
}Snake;

Snake* head = new Snake;
```

```
ULONGLONG ct = 200;

int main() {
    /*      隐藏光标    */
    HANDLE hc = GetStdHandle(STD_OUTPUT_HANDLE);         //获得标准输出设备句柄
    CONSOLE_CURSOR_INFO cci;                             //定义光标信息结构体

    GetConsoleCursorInfo(hc, &cci);
    cci.bVisible = 0;                                    //为0时，光标不可见
    SetConsoleCursorInfo(hc, &cci);

    Loading();                                           //进入加载界面

    system("color 07");                                  //设置背景颜色
    system("title 贪吃蛇小游戏         ——by：orall");   //设置窗口标题
    system("mode con: cols=48 lines=30");                //设置窗口大小

    game();
    return 0;
}


//游戏主体
void game() {
    show();

    //初始化
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
FOREGROUND_INTENSITY);   //设置颜色
    srand(unsigned(time(NULL)));
    //h_x,h_y就是头的位置,t_x,t_y就是尾的位置

    //从（2,2）开始

    h_x = 1 + rand() % 18;           //尽量避免一开始就在边界出现
    h_y = rand() % 18;
    gotoxy(2 + h_x, 2 + h_y, 2);
    cout << "■";

    //默认是竖的，向上走
    t_x = h_x;
    t_y = h_y + 1;
    gotoxy(2 + t_x, 2 + t_y, 2);
    cout << "■";
```

```
    head->x = h_x;
    head->y = h_y;
    head->next = new Snake;
    head->next->x = t_x;
    head->next->y = t_y;
    head->next->next = NULL;


    map[h_y][h_x] = 1;
    map[t_y][t_x] = 1;       //暂时先竖着
    int direction = 'w';     //移动的方向，一开始是向上

    setfood();
    ULONGLONG first_time = GetTickCount64(), second_time;
    while (1) {

        if (_kbhit()) {
            switch (_getch()) {
            case 'a':case 'A':
                if (direction != 'd') {          //防止出现倒退
                    direction = 'a';
                }
                break;
            case 'd':case 'D':
                if (direction != 'a') {          //防止出现倒退
                    direction = 'd';
                }
                break;
            case 'w':case 'W':
                if (direction != 's') {          //防止出现倒退
                    direction = 'w';
                }
                break;
            case 's':case 'S':
                if (direction != 'w') {          //防止出现倒退
                    direction = 's';
                }
                break;
            case 27:    //相当于【ESC】
                Setup();
                system("cls");
                show();                      //画地图
                show2();                     //画蛇身和食物
                first_time = GetTickCount64();
                break;
            default:    //按任何未控制键停止
```

```
                while (1) {
                    char tmp = _getch();
                    if (tmp == 'A' || tmp == 'a' || tmp == 'D' || tmp == 'd' || tmp == 'S'
|| tmp == 's' || tmp == 'W' || tmp == 'w') {
                        break;
                    }
                }
                first_time = GetTickCount64();  //重新获取毫秒数
            }
        }

        second_time = GetTickCount64();

        if (second_time - first_time >= ct) {
            if (direction == 'w') {
                move(h_x, --h_y);
            }
            else if (direction == 's') {
                move(h_x, ++h_y);
            }
            else if (direction == 'a') {
                move(--h_x, h_y);
            }
            else if (direction == 'd') {
                move(++h_x, h_y);
            }
            first_time = GetTickCount64();
        }
    }
}

//加载界面（其实就只是一个效果）
void Loading() {
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    system("title 游戏正在加载中");
    system("mode con cols=54 lines=5");
    int Loa, Lob;
    //颜色参考https://www.freesion.com/article/645561183/
    SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
FOREGROUND_INTENSITY);   //设置颜色
    for (Loa = 0; Loa <= 100; Loa += 4) {
        gotoxy(2, 10, 1);
        cout << Loa << "%";
        gotoxy(0, 0, 1);
        cout << " ================================================" << endl;
        for (Lob = 0; Lob < Loa / 4; Lob++) {
```

```
            gotoxy(Lob, 1, 2);      //字符宽2 高1（不是ASCLL码）
            cout << " ";
        }
        cout << endl << " ================================================" << endl;
        cout << "                正在进入游戏";
        Sleep(40);
    }
    return;
}

//光标到指定位置(这里用来解决清屏闪烁的问题）
void gotoxy(int x, int y, int width) {
    HANDLE hout;
    hout = GetStdHandle(STD_OUTPUT_HANDLE); //获得标准输入输出的句柄
    COORD pos = { x * width,y };            //表示一个字符在控制台屏幕的坐标（ASCLL码宽度为1，非ASCLL宽度为2）
    SetConsoleCursorPosition(hout, pos);    //光标移动到指定的位置
    return;
}

//前两个参数表示将要达到的头的位置
void move(int x, int y) {       //移动蛇身

    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN | FOREGROUND_INTENSITY);   //设置颜色
    int i = 1;
    if ( y == 20 || y == -1 || x == 20 || x == -1 || map[y][x] == 1 ) {             //撞到墙的情况和撞到自己的情况
        SetConsoleTextAttribute(hconsole, FOREGROUND_RED | FOREGROUND_INTENSITY);
        gotoxy(6, 5, 1);
        cout << "□□□□□□□□□□□□□□□□□□";
        gotoxy(6, 6, 1);
        cout << "□            GAME OVER!          □";
        gotoxy(7, 7, 1);
        cout << "□            按Y继续            □";
        gotoxy(8, 8, 1);

        cout << "□            按N结束            □";
        gotoxy(9, 9, 1);
        cout << "□□□□□□□□□□□□□□□□";

        char judge;

        while (1) {
            judge = _getch();
```

```
            if (judge == 'Y' || judge == 'y') {
                for ( int i = 0; i < 20; i++) {              //清空初始化
                    for (int j = 0; j < 20; j++) {
                        map[i][j] = 0;
                    }
                }
                snake_len = 2;                              //初始化蛇的长度
                head = new Snake;
                system("cls");
                game();
            }
            else if (judge == 'N' || judge == 'n') {
                exit(0);
            }
        }
    }

    if (map[y][x] == 2) {        //如果有食物，则尾部不用消失
        snake_len++;
        //头部出现
        Snake* tmp = new Snake;
        tmp->x = x;
        tmp->y = y;
        tmp->next = head;
        head = tmp;

        //地图出现头部
        map[y][x] = 1;
        //显示头部
        gotoxy(2 + x, 2 + y, 2);
        cout << "■";

        //更新得分
        SetConsoleTextAttribute(hconsole, 6);
        gotoxy(0, 0, 2);
        cout << "                      得分：";
        cout << snake_len * 5 - 10 << "分" << endl;

        setfood();
    }
    else {               //如果没有食物的话，尾部消失，头部出现

        //尾部消失
        Snake* current = NULL;
        for (current = head; i < snake_len - 1; i++) {       //找到次尾节点
            current = current->next;
```

```
        }

        //消失尾部
        gotoxy(2 + current->next->x, 2 + current->next->y, 2);
        cout << "  ";

        //地图消失尾部
        map[current->next->y][current->next->x] = 0;
        delete(current->next);
        current->next = NULL;

        //头部出现
        Snake* tmp = new Snake;
        tmp->x = x;
        tmp->y = y;
        tmp->next = head;
        head = tmp;

        //地图出现头部
        map[y][x] = 1;

        //显示头部
        gotoxy(2 + x, 2 + y, 2);
        cout << "■";
    }


}
void show() {                    //显示地图
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hconsole, 6);
    cout << "                     得分：";
    cout << snake_len * 5 - 10  << "分" <<endl;
    SetConsoleTextAttribute(hconsole, 3);

    cout << "  □□□□□□□□□□□□□□□□□□□□□□\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
    cout << "  □                                    □\n";
```

```
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □                                                □\n";
    cout << "  □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□\n";

    gotoxy(0, 23, 1);
    SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
FOREGROUND_INTENSITY);
    cout<<
        "         操作指南       |\n"
        "   【A】【D】左右移动   |\n"
        "   【W】【S】上下移动   |\n"
        "   【Esc】进入设置界面 |\n"
        "       ——by:orall     |\n"
    ;
    SetConsoleTextAttribute(hconsole, FOREGROUND_RED | FOREGROUND_INTENSITY);
    gotoxy(23, 23, 1);
    cout << "|     温馨提示:";
    gotoxy(23, 24, 1);
    cout << "|  尿急可按任";
    gotoxy(23, 25, 1);
    cout << "|  意非控制键";
    gotoxy(23, 26, 1);
    cout << "|  暂停. ";
    gotoxy(23, 27, 1);
    cout << "|";
}

void show2(){                              //画蛇身
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    for ( int i = 0; i < 20; i++) {
        for (int j = 0; j < 20; j++) {
            if (map[i][j] == 2 ) {
                SetConsoleTextAttribute(hconsole, FOREGROUND_INTENSITY | FOREGROUND_GREEN);
                gotoxy(2 + j, 2 + i, 2);
                cout << "●";
            }
            else if (map[i][j] == 1) {
                SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
```

```
FOREGROUND_INTENSITY);

                gotoxy(2 + j, 2 + i, 2);
                cout << "■";
            }

        }
    }
}

void setfood() {                        //放置食物
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hconsole, FOREGROUND_INTENSITY | FOREGROUND_GREEN );
    srand((unsigned(time(NULL))));      //设置随机种子
    int x = rand() % 20;
    int y = rand() % 20;
    if (map[y][x] == 1) {
        setfood();
    }
    else {
        //放置食物
        gotoxy(2 + x, 2 + y, 2);
        cout << "●";
        map[y][x] = 2;
    }
}

void Setup() {  //设置界面
    HANDLE hconsole = GetStdHandle(STD_OUTPUT_HANDLE);
    SetConsoleTextAttribute(hconsole, FOREGROUND_BLUE | FOREGROUND_GREEN |
FOREGROUND_INTENSITY);
    gotoxy(4, 16, 1);
    cout<<"  ----------------------------------";
    gotoxy(4, 17, 1);
    cout<<"  游戏的难度:1.简单 2.困难 3.复杂 >> ";
    switch (_getche())                                                  //有回显的键盘输
入
    {
        case '1':
            ct = 100;
            break;
        case '2':
            ct = 80;
            break;
        case '3':
            ct = 50;
```

```
            break;
        default:
            gotoxy(4, 19, 1);
            cout<<" 输入错误 请重试";
            Setup();                //递归
    }
    gotoxy(4, 19, 1);
    cout<<" 设置成功 正在保存...";
    Sleep(1000);
    return;
}
```

贪吃蛇雏形(2021.3.6)

总结:

1.移动的关键`DWORD first_time = GetTickCount64(), second_time;`

2.监控键盘是否按下`if (_kbhit())`

```
/*
* 版本：贪吃蛇雏形
* 时间：2021.3.6
* 作者：orall
*/

//关键 GetTickCount() 和 kbhit()

#include <iostream>
#include <time.h>
#include <conio.h>
#include <windows.h>
using namespace std;
int map[20][20] = { 0 };            // 地图（0表示的是空，1表示的是蛇，2表示的是食物）
int snake_len = 2;              //表示蛇的长度
int h_x, h_y, t_x, t_y;         //头位置
void move(int x, int y);            //移动蛇身
void show();                    //显示地图
void setfood();                 //放置食物、

typedef struct Snake {
    int x, y;
    Snake* next;
}Snake;

Snake* head = new Snake;

int main() {
```

```
    //初始化

    srand(unsigned(time(NULL)));
    //h_x,h_y就是头的位置,t_x,t_y就是尾的位置
    h_x = rand() % 20;
    h_y = rand() % 18;
    t_x = h_x;
    t_y = h_y + 1;

    head->x = h_x;
    head->y = h_y;
    head->next = new Snake;
    head->next->x = t_x;
    head->next->y = t_y;
    head->next->next = NULL;


    map[h_y][h_x] = 1;
    map[t_y][t_x] = 1;       //暂时先竖着
    int direction = 'w';     //移动的方向，一开始是向上
    show();

    setfood();
    DWORD first_time = GetTickCount64(), second_time;
    while (1) {

        if (_kbhit()) {
            switch (_getch()) {
            case 'a':case 'A':
                direction = 'a';
                break;
            case 'd':case 'D':
                direction = 'd';
                break;
            case 'w':case 'W':
                direction = 'w';
                break;
            case 's':case 'S':
                direction = 's';
                break;
            }
        }

        second_time = GetTickCount64();

        if (second_time - first_time >= 1000 ) {
```

```
            if (direction == 'w') {
                move(h_x, --h_y);
            }
            else if (direction == 's') {
                move(h_x, ++h_y);
            }
            else if (direction == 'a') {
                move(--h_x, h_y);
            }
            else if (direction == 'd') {
                move(++h_x, h_y);
            }
            show();
            first_time = GetTickCount64();
        }
    }


}

//前两个参数表示将要达到的头的位置
void move(int x, int y) {       //移动蛇身

    int i = 1;
    if (y == 20 || y == -1 || x == 20 || x == -1) {
        system("cls");
        cout << "游戏结束";
        Sleep(2000);
        exit(0);
    }

    if (map[y][x] == 2) {       //如果有食物，则尾部不用消失

        snake_len++;
        //头部出现
        Snake* tmp = new Snake;
        tmp->x = x;
        tmp->y = y;
        tmp->next = head;
        head = tmp;

        //地图出现头部
        map[y][x] = 1;

        setfood();
```

```
    }
    else {               //如果没有食物的话，尾部消失，头部出现

        //尾部消失
        Snake* current = NULL;
        for (current = head; i < snake_len - 1; i++) {       //找到次尾节点
            current = current->next;
        }
        //地图消失尾部
        map[current->next->y][current->next->x] = 0;
        delete(current->next);
        current->next = NULL;


        //头部出现
        Snake* tmp = new Snake;
        tmp->x = x;
        tmp->y = y;
        tmp->next = head;
        head = tmp;

        //地图出现头部
        map[y][x] = 1;
    }


}
void show() {                    //显示地图
    system("cls");
    for (int i = 0; i < 20; i++) {
        for (int j = 0; j < 20; j++) {

            if (map[i][j] == 0) {
                cout << "  ";
            }
            else if (map[i][j] == 1) {
                cout << "0";
            }
            else {
                cout << "·";
            }

            //  cout << map[i][j] << " ";
        }
        cout << endl;
    }
```

```
}

void setfood() {                        //放置食物
    srand((unsigned(time(NULL))));      //设置随机种子
    int x = rand() % 20;
    int y = rand() % 20;
    if (map[x][y] == 1) {
        setfood();
    }
    else {
        map[x][y] = 2;
    }

}
```