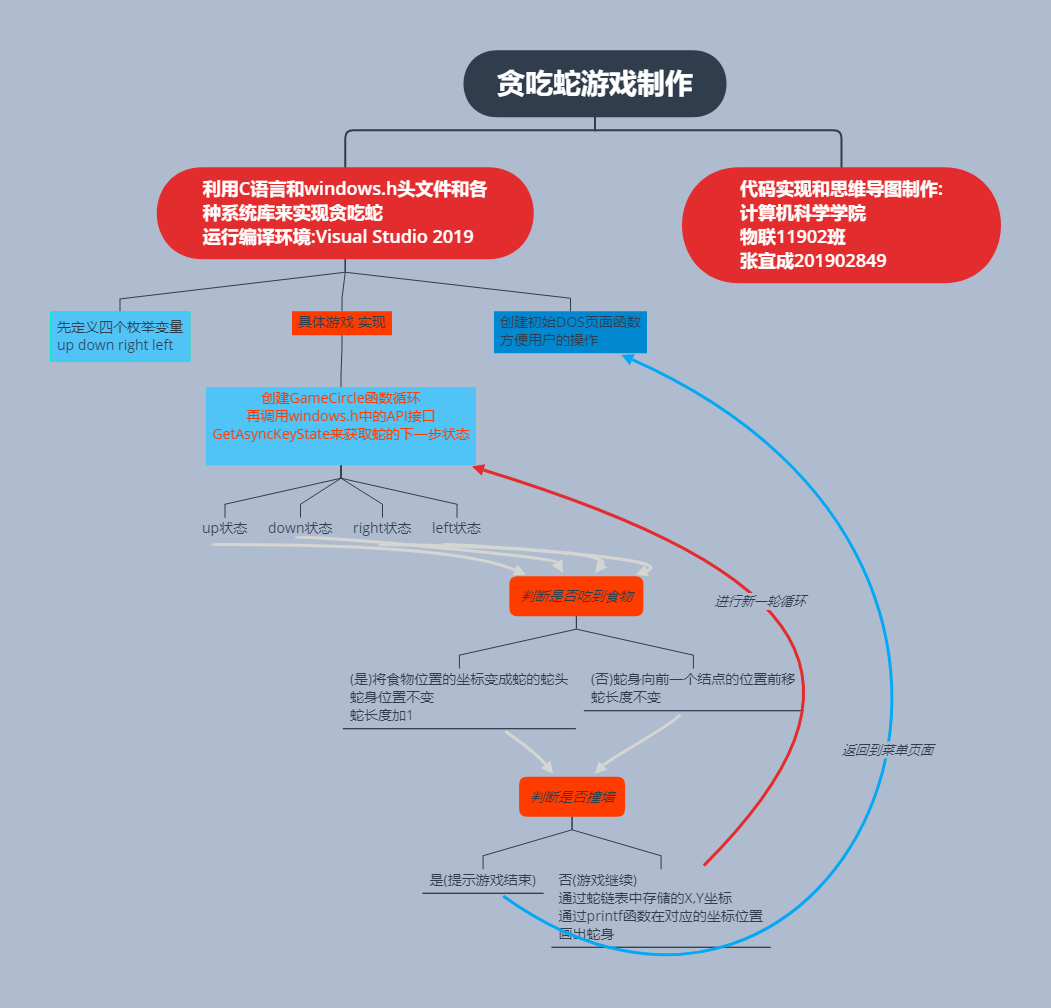
* + 1. 小组成员:

计算机科学学院 物联网工程11902班 张xx

负责代码实现

负责游戏方案设计

* + 1. 设计方案:



* + 1. 实现过程:

第一步: 先实现用户UI界面方便用户的操作

void welcometogame()//开始界面

{

Pos(40, 12);

printf("欢迎来到贪食蛇游戏！");

Pos(40, 16);

printf("飞！！芜湖！！起飞！！");

Pos(40, 25);

system("pause");

//清屏操作

system("cls");

Pos(13, 12);

printf("用↑(UP).↓(DOWN).←(LEFT).→(RIGHT)分别控制蛇的移动， F1 为加速，2 为减速\n");

Pos(13, 13);

printf("加速将能得到更高的分数。\n");

system("pause");

//清屏操作

system("cls");

}

第二步:创建对应的枚举变量enum { U = 1, D, L, R };

创建对应的贪吃蛇和食物的结构体

typedef struct SNAKE //蛇身的一个节点

{

int x;

int y;

struct SNAKE\* next; //结构体指针

}snake;

第三步定义相关全局变量

//全局变量//

int score = 0, add = 10;//总得分与每次吃食物得分。

int status;//status是状态

snake\* head, \* food;//蛇头指针，食物指针

snake\* q;//遍历蛇的时候用到的指针

int endgamestatus = 0; //游戏结束的情况，1：撞到墙；2：咬到自己；3：主动退出游戏。

int S\_Speed = 150;

第四步利用windows.h中的API接口实时获取键盘信息

while (1)

{

//利用API获取键盘输入

if (GetAsyncKeyState(VK\_UP) && status != D) //如果向上则无法向下直接掉头

{

status = U;

}

else if (GetAsyncKeyState(VK\_DOWN) && status != U) //如果向下则无法向上直接掉头

{

status = D;

}

else if (GetAsyncKeyState(VK\_LEFT) && status != R) //如果向左则无法向右直接掉头

{

status = L;

}

else if (GetAsyncKeyState(VK\_RIGHT) && status != L) //如果向右则无法向左直接掉头

{

status = R;

}

else if (GetAsyncKeyState(VK\_SPACE)) //如果Windows API读取到SPACE 则执行暂停程序函数

{

pause();

}

else if (GetAsyncKeyState(VK\_ESCAPE)) //如果Windows API读取到ESC 则执行暂停程序函数

{

//第三种情况(直接用ESC弹出游戏)

endgamestatus = 3;

break; //直接弹出while循环

}

第五步利用前面API接口获取到的status信息来分成多种情况来处理,每次对蛇身和食物处理完后再将蛇和食物重新画出来

void snakemove()//蛇前进,上U,下D,左L,右R

{

//蛇的下一个蛇头

snake\* nexthead;

nexthead = (snake\*)malloc(sizeof(snake));

//如果向上移动

if (status == U)

{

//横坐标x不变

nexthead->x = head->x;

//总左边y-1

nexthead->y = head->y - 1;

//将原先的头指针赋值给 新的头指针的下一个元素

nexthead->next = head;

//将新的头指针赋值给(全局变量头指针)

//(更新头指针)

head = nexthead;

if (nexthead->x == food->x && nexthead->y == food->y)//如果下一个有食物//

{

//得分

score = score + add;

//生成新的食物

createfood();

}

else //如果没有食物//

{

//找到蛇尾

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

////将原先尾部残留的蛇尾用空白替换掉

//Pos(q->next->x, q->next->y);

//printf(" ");

//删除蛇尾

free(q->next);

q->next = NULL;

}

}

if (status == D)

{

nexthead->x = head->x;

nexthead->y = head->y + 1;

//将原先的头指针赋值给 新的头指针的下一个元素

nexthead->next = head;

//将新的头指针赋值给(全局变量头指针)

//(更新头指针)

head = nexthead;

if (nexthead->x == food->x && nexthead->y == food->y) //有食物

{

//得分

score = score + add;

//生成新的食物

createfood();

}

else //没有食物

{

//找到蛇尾

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

//在结尾使用过一次system("cls"); 所以不用着急清除蛇尾，在结尾部分直接画蛇

////将原先尾部残留的蛇尾用空白替换掉

//Pos(q->next->x, q->next->y);

//printf(" ");

//删除蛇尾

free(q->next);

q->next = NULL;

}

}

if (status == L)

{

nexthead->x = head->x - 2;

nexthead->y = head->y;

//将原先的头指针赋值给 新的头指针的下一个元素

nexthead->next = head;

//将新的头指针赋值给(全局变量头指针)

//(更新头指针)

head = nexthead;

if (nexthead->x == food->x && nexthead->y == food->y)//有食物

{

//得分

score = score + add;

//生成新的食物

createfood();

}

else //没有食物

{

//找到蛇尾

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

//在结尾使用过一次system("cls"); 所以不用着急清除蛇尾，在结尾部分直接画蛇

////将原先尾部残留的蛇尾用空白替换掉

//Pos(q->next->x, q->next->y);

//printf(" ");

//删除蛇尾

free(q->next);

q->next = NULL;

}

}

if (status == R)

{

nexthead->x = head->x + 2;

nexthead->y = head->y;

//将原先的头指针赋值给 新的头指针的下一个元素

nexthead->next = head;

//将新的头指针赋值给(全局变量头指针)

//(更新头指针)

head = nexthead;

if (nexthead->x == food->x && nexthead->y == food->y)//有食物

{

//得分

score = score + add;

//生成新的食物

createfood();

}

else //没有食物

{

//找到蛇尾

q = head;

while (q->next->next != NULL)

{

q = q->next;

}

//在结尾使用过一次system("cls"); 所以不用着急清除蛇尾，在结尾部分直接画蛇

////将原先尾部残留的蛇尾用空白替换掉

//Pos(q->next->x, q->next->y);

//printf(" ");

//删除蛇尾(释放掉内存空间)

free(q->next);

q->next = NULL;

}

}

//判断撞墙函数(第一种游戏结束情况)

cantcrosswall();

//判断蛇是否咬到自己的函数判断

biteself(); //判断是否会咬到自己

//清屏

system("cls");

//HANDLE hOut = GetStdHandle(STD\_OUTPUT\_HANDLE);

//CONSOLE\_CURSOR\_INFO cci;

//GetConsoleCursorInfo(hOut, &cci);

//cci.bVisible = FALSE;

//SetConsoleCursorInfo(hOut, &cci);

//1.重置边框(重置图形边框)

creatMap();

//2.画蛇

q = head;

while (q->next != NULL)

{

Pos(q->x, q->y);

printf("●");

q = q->next;

}

Pos(q->x, q->y);

printf("●");

//4.画食物

Pos(food->x, food->y);

printf("■");

}

第六步实现游戏的结束(判断撞墙或者直接结束游戏则endgame)

void endgame()//结束游戏

{

system("cls");

Pos(24, 12);

//撞墙

if (endgamestatus == 1)

{

printf("对不起，您撞到墙了。游戏结束.");

}

//咬自己

else if (endgamestatus == 2)

{

printf("对不起，您咬到自己了。游戏结束.");

}

//按ESC

else if (endgamestatus == 3)

{

printf("您的已经结束了游戏。");

}

Pos(24, 13);

printf("您的得分是%d\n", score);

system("pause");

//直接弹出程序

exit(0);

}

4.开发心得:  
通过本次合作开发项目,相对于独立开发项目更加容易一些,可以明确分工谁进行游戏设计与代码实现,在分工后,每个人又可以发挥自己的长处,对于我张xx来说我对于代码实现比较好做,我比较喜欢敲代码,对于张xx来说,对游戏的设计更加熟悉,在我们两者的努力下,终于完成了贪吃蛇的制作,来来回回贪吃蛇代码总共实现了620多行,我觉得自己认真做了就受益匪浅,我们都为这次能自己做出游戏而引以为嗷,在人生中我也认为这不仅是一个项目,而是一次历练,可以让我们更加提升我们的能力.