moveSnake()

第一次尝试

出发点就错了

if (FORWARD\_DIRECTION == L) {

//gotoxy(100, 3);

while (head != NULL) {

head->x = head->x - 2; //这里出现了严重错误

//printf("%d %d", head->x, head->y);

head = head->next;

}

head = p;

gotoxy(head->x, head->y);

printf("■");

while (head != NULL) {

head = head->next;

if (head->next == NULL) break;

}

gotoxy(head->x + 2, head->y);

printf(" ");

}

if (FORWARD\_DIRECTION == U) {

//gotoxy(100, 3);

while (head != NULL) {

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

//printf("%d %d", head->x, head->y);

head = head->next;

}

head = p;

gotoxy(head->x, head->y);

printf("■");

while (head != NULL) {

head = head->next;

if (head->next == NULL) break;

}

gotoxy(head->x + 2, head->y);

printf(" ");

}

第二次尝试

head = p; //初始化头指针的位置

if (FORWARD\_DIRECTION == L) {

//gotoxy(100, 3);

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

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

nexthead->y = head->y;

nexthead->next = head;

head = nexthead;

p = head;

gotoxy(head->x, head->y);

printf("■");

while (head != NULL) { //这里出现了错误

head = head->next;

if (head->next == NULL) break;

}

gotoxy(head->x, head->y);

printf(" ");

free(head); //当free了head后，指向head的指针的next乱了

head = NULL; //这里赋headNULL没有意义

}

Bug1

修复前：

如果瞬时按下两个键

比如当前是左键，按右键会被屏蔽掉

例，当前方向为向左行走，当瞬时按下上右两个键时，方向瞬时被修改为上，然后被修改为右，当调用moveSnake()函数时，会出现蛇头撞蛇头后面方块的情况，会造成死亡。

修复方案为：

改用else if语句，只进行判断一次

void keyboardControl()

{

FORWARD\_DIRECTION = L;

while (1)

{

gradeBoard();

if (GetAsyncKeyState(VK\_UP) && FORWARD\_DIRECTION != D)

{

FORWARD\_DIRECTION = U;

}

if (GetAsyncKeyState(VK\_DOWN) && FORWARD\_DIRECTION != U)

{

FORWARD\_DIRECTION = D;

}

if (GetAsyncKeyState(VK\_LEFT) && FORWARD\_DIRECTION != R)

{

FORWARD\_DIRECTION = L;

}

if (GetAsyncKeyState(VK\_RIGHT) && FORWARD\_DIRECTION != L)

{

FORWARD\_DIRECTION = R;

}

if (GetAsyncKeyState(VK\_F1)) //按F1加快速度

{

speedUp();

}

if (GetAsyncKeyState(VK\_F2)) //按F2减慢速度

{

speedDown();

}

if (GetAsyncKeyState(VK\_SPACE)) //空格键暂停

{

while (1)

{

if (GetAsyncKeyState(VK\_ESCAPE)) //在暂停时也能响应ESC键退出游戏

{

endGame();

}

gotoxy(100, 8);

color(4);

printf("Pause.");

Sleep(300);

if (GetAsyncKeyState(VK\_SPACE))

{

gotoxy(100, 8);

printf("Start.");

break;

}

}

}

if (GetAsyncKeyState(VK\_ESCAPE))

{

endGame();

}

moveSnake();

Sleep(sleepTime); //通过改变sleepTime大小更改贪吃蛇速度

}

}

修改后：

void keyboardControl()

{

FORWARD\_DIRECTION = L;

while (1)

{

gradeBoard();

if (GetAsyncKeyState(VK\_UP) && FORWARD\_DIRECTION != D)

{

FORWARD\_DIRECTION = U;

}

else if (GetAsyncKeyState(VK\_DOWN) && FORWARD\_DIRECTION != U)

{

FORWARD\_DIRECTION = D;

}

else if (GetAsyncKeyState(VK\_LEFT) && FORWARD\_DIRECTION != R)

{

FORWARD\_DIRECTION = L;

}

else if (GetAsyncKeyState(VK\_RIGHT) && FORWARD\_DIRECTION != L)

{

FORWARD\_DIRECTION = R;

}

else if (GetAsyncKeyState(VK\_F1)) //按F1加快速度

{

speedUp();

}

else if (GetAsyncKeyState(VK\_F2)) //按F2减慢速度

{

speedDown();

}

else if (GetAsyncKeyState(VK\_SPACE)) //空格键暂停

{

while (1)

{

if (GetAsyncKeyState(VK\_ESCAPE)) //在暂停时也能响应ESC键退出游戏

{

endGame();

}

gotoxy(100, 8);

color(4);

printf("Pause.");

Sleep(300);

if (GetAsyncKeyState(VK\_SPACE))

{

gotoxy(100, 8);

printf("Start.");

break;

}

}

}

else if (GetAsyncKeyState(VK\_ESCAPE))

{

endGame();

}

moveSnake();

Sleep(sleepTime); //通过改变sleepTime大小更改贪吃蛇速度

}

}