//打印蛇与覆盖蛇

void DrawSnake(int flag)

{

if (flag == 1) //打印蛇

{

color(10); //颜色设置为绿色

CursorJump(2 \* snake.x, snake.y);

printf("■"); //打印蛇头

for (int i = 0; i < snake.len; i++)

{

CursorJump(2 \* body[i].x, body[i].y);

printf("□"); //打印蛇身

}

}

else //覆盖蛇

{

if (body[snake.len - 1].x != 0) //防止len++后将(0, 0)位置的墙覆盖

{

//将蛇尾覆盖为空格即可

CursorJump(2 \* body[snake.len - 1].x, body[snake.len - 1].y);

printf(" ");

}

}

}

//移动蛇

void MoveSnake(int x, int y)

{

DrawSnake(0); //先覆盖当前所显示的蛇

face[body[snake.len - 1].y][body[snake.len - 1].x] = KONG; //蛇移动后蛇尾重新标记为空

face[snake.y][snake.x] = BODY; //蛇移动后蛇头的位置变为蛇身

//蛇移动后各个蛇身位置坐标需要更新

for (int i = snake.len - 1; i > 0; i--)

{

body[i].x = body[i - 1].x;

body[i].y = body[i - 1].y;

}

//蛇移动后蛇头位置信息变为第0个蛇身的位置信息

body[0].x = snake.x;

body[0].y = snake.y;

//蛇头的位置更改

snake.x = snake.x + x;

snake.y = snake.y + y;

DrawSnake(1); //打印移动后的蛇

}

//执行按键

void run(int x, int y)

{

int t = 0;

while (1)

{

if (t == 0)

t = 3000; //这里t越小，蛇移动速度越快（可以根据次设置游戏难度）

while (--t)

{

if (kbhit() != 0) //若键盘被敲击，则退出循环

break;

}

if (t == 0) //键盘未被敲击

{

JudgeFunc(x, y); //判断到达该位置后，是否得分与游戏结束

MoveSnake(x, y); //移动蛇

}

else //键盘被敲击

{

break; //返回Game函数读取键值

}

}

}

//游戏主体逻辑函数

void Game()

{

int n = RIGHT; //开始游戏时，默认向后移动

int tmp = 0; //记录蛇的移动方向

goto first; //第一次进入循环先向默认方向前进

while (1)

{

n = getch(); //读取键值

//在执行前，需要对所读取的按键进行调整

switch (n)

{

case UP:

case DOWN: //如果敲击的是“上”或“下”

if (tmp != LEFT&&tmp != RIGHT) //并且上一次蛇的移动方向不是“左”或“右”

{

n = tmp; //那么下一次蛇的移动方向设置为上一次蛇的移动方向

}

break;

case LEFT:

case RIGHT: //如果敲击的是“左”或“右”

if (tmp != UP&&tmp != DOWN) //并且上一次蛇的移动方向不是“上”或“下”

{

n = tmp; //那么下一次蛇的移动方向设置为上一次蛇的移动方向

}

case SPACE:

case ESC:

case 'r':

case 'R':

break; //这四个无需调整

default:

n = tmp; //其他键无效，默认为上一次蛇移动的方向

break;

}

first: //第一次进入循环先向默认方向前进

switch (n)

{

case UP: //方向键：上

run(0, -1); //向上移动（横坐标偏移为0，纵坐标偏移为-1）

tmp = UP; //记录当前蛇的移动方向

break;

case DOWN: //方向键：下

run(0, 1); //向下移动（横坐标偏移为0，纵坐标偏移为1）

tmp = DOWN; //记录当前蛇的移动方向

break;

case LEFT: //方向键：左

run(-1, 0); //向左移动（横坐标偏移为-1，纵坐标偏移为0）

tmp = LEFT; //记录当前蛇的移动方向

break;

case RIGHT: //方向键：右

run(1, 0); //向右移动（横坐标偏移为1，纵坐标偏移为0）

tmp = RIGHT; //记录当前蛇的移动方向

break;

case SPACE: //暂停

system("pause>nul"); //暂停后按任意键继续

break;

case ESC: //退出

system("cls"); //清空屏幕

color(7); //颜色设置为白色

CursorJump(COL - 8, ROW / 2);

printf(" 游戏结束 ");

CursorJump(COL - 8, ROW / 2 + 2);

exit(0);

case 'r':

case 'R': //重新开始

system("cls"); //清空屏幕

main(); //重新执行主函数

}

}

}