#include <stdio.h>#include <stdlib.h>#include <graphics.h>#include <conio.h>

#include <time.h>#include <pthread.h>#pragma comment(lib, "pthreadVC2.lib")

#include <Windows.h>#pragma comment(lib,"winmm.lib")#include "struct.h"

#include "isin.h"#include "initmap.h"

IMAGE bk, f1, f2, m1, m2, xx1, xx2, boom1, boom2, win;

int fooddata[42][28] = { { 0 } };

int poisondata[42][28] = { { 0 } };

int poisonstyle[42][28] = { { 0 } };

int score = 0;

char name[20];

int mushroom = 0;

int mushNum = 10;

int main() {

initgraph(630, 460);

initmapC();//地图初始化

loadimage(&bk, "snakefile\\map-3.jpg", 630, 420);//初始化背景图片

putimage(0, 0, &bk);

loadimage(&f1, "snakefile\\jl.jpg", 21, 21);

loadimage(&f2, "snakefile\\ym.jpg", 21, 21);

loadimage(&m1, "snakefile\\mushroom1.jpg", 15, 15);

loadimage(&m2, "snakefile\\mushroom2.jpg", 15, 15);

loadimage(&xx1, "snakefile\\xx1.jpg", 15, 15);

loadimage(&xx2, "snakefile\\xx2.jpg", 15, 15);

loadimage(&boom1, "snakefile\\potato1.jpg", 15, 15);

loadimage(&boom2, "snakefile\\potato2.jpg", 15, 15);

loadimage(&win, "snakefile\\win.jpg", 630, 420);

Snake \*snake = createSnake();

printSnake(snake);//产生并且打印一条最初的蛇

char c;//保存键盘读取的字符

int prelength = snake->length;//用来保证在蛇的长度没有发生改变的情况下，不会重复产生毒草

int createMoreFood = 0;

settextstyle(15, 0, \_T("宋体"));

outtextxy(0, 445, \_T("当前吃到食物数量："));

char s[50];

int maxscore = getScore();

sprintf\_s(s, "%d", maxscore);

outtextxy(180, 445, \_T("通关最少食物数量："));

outtextxy(320, 445, s);

PlaySound(TEXT("snakefile\\dragon.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

while (1) {

if (createMoreFood % 150 == 0) {

mushNum++;

}

score = snake->score;

if (snake->length == 31) {

victory();

}

sprintf\_s(s, "%d ", snake->score);

outtextxy(140, 445, s);

createMoreFood++;

if (!snake->food || createMoreFood == 1 || createMoreFood % 120 == 0) {

createFood(snake);//保存产生的食物坐标

}

if (!snake->poison || createMoreFood == 1 || ((snake->length) % 2 == 0 && snake->length != prelength) || createMoreFood % 140 == 0) {

createPoison(snake);

}

if (mushroom && createMoreFood % 3 == 0) {

hidePoison();

}

else if (mushroom) {

showPoison();

}

if (!mushroom && createMoreFood % 79 == 0) {

mushroom = 1;

createMushroom(snake, mushNum);

}

if (mushroom && createMoreFood % 129 == 0) {

mushroom = 0;

clearPoison();

}

prelength = snake->length;//操作完成后，保存当前长度

if (\_kbhit()) {//如果键盘输入，则读取

if (c = \_getch()) {

if (c == 27) {

exit(0);

closegraph();

}

c = \_getch();

if (c == 72 && snake->direction != 2) {

snake->direction = 1;

snake->dx = 0;

snake->dy = -15;

}

if (c == 80 && snake->direction != 1) {

snake->direction = 2;

snake->dx = 0;

snake->dy = 15;

}

if (c == 75 && snake->direction != 4) {

snake->direction = 3;

snake->dx = -15;

snake->dy = 0;

}

if (c == 77 && snake->direction != 3) {

snake->direction = 4;

snake->dx = 15;

snake->dy = 0;

}

}

}

slowMove(snake);//移动蛇

isEat(snake);//判断是否吃到了食物

isPoison(snake);//判断是否吃到了毒草

isDead(snake);//判断是否撞到自己的身体

isHit(snake);//判断是否撞到边界

isHitWall(snake);//判断是否撞墙

}

\_getch();

closegraph();

return 0;

}

//=========================================================

void createMushroom(Snake \*p, int n) {

int k = 0;

while (k < n) {

srand((unsigned)time(NULL));

Poison \*poison = (Poison\*)malloc(sizeof(Poison));

int x = 0;

int y = 0;

x = rand() % 42;

y = rand() % 28;

poison->x = x \* 15 + 7;

poison->y = y \* 15 + 7;

while (map[x][y] || fooddata[x][y] || poisondata[x][y] || isInSnakeBody(p, poison)) {

x = rand() % 42;

y = rand() % 28;

poison->x = x \* 15 + 7;

poison->y = y \* 15 + 7;

}

poisondata[x][y] = 1;

poisonstyle[x][y] = 2;

putimage((poison->x) - 7, (poison->y) - 7, &m2, NOTSRCERASE);//掩码

putimage((poison->x) - 7, (poison->y) - 7, &m1, SRCINVERT);//精灵

free(poison);

k++;

}

}

void clearPoison() {

IMAGE temp;

for (int i = 0; i < 42; i++) {

for (int j = 0; j < 28; j++) {

if (poisonstyle[i][j] != 1 && poisonstyle[i][j] != 0) {

poisondata[i][j] = 0;

poisonstyle[i][j] = 0;

SetWorkingImage(&bk);

getimage(&temp, i \* 15, j \* 15, 15, 15);

SetWorkingImage();

putimage(i \* 15, j \* 15, &temp);

}

}

}

}

void hidePoison() {

IMAGE temp;

for (int i = 0; i < 42; i++) {

for (int j = 0; j < 28; j++) {

if (poisonstyle[i][j] != 1 && poisonstyle[i][j] != 0) {

SetWorkingImage(&bk);

getimage(&temp, i \* 15, j \* 15, 15, 15);

SetWorkingImage();

putimage(i \* 15, j \* 15, &temp);

}

}

}

}

void showPoison() {

for (int i = 0; i < 42; i++) {

for (int j = 0; j < 28; j++) {

if (poisonstyle[i][j] != 1 && poisonstyle[i][j] != 0) {

putimage(i \* 15, j \* 15, &m2, NOTSRCERASE);//掩码

putimage(i \* 15, j \* 15, &m1, SRCINVERT);//精灵

}

}

}

}

void rankingSave() {

FILE \*fp = NULL;

fopen\_s(&fp, "snakefile\\resource\\Ranking\\player.txt", "r+");

fscanf\_s(fp, "%s", name, 20);

fclose(fp);

fopen\_s(&fp, "snakefile\\resource\\Ranking\\map3\\ranking.txt", "a+");

fprintf(fp, "%s", "\n");

fprintf(fp, "%s", name);

fprintf(fp, "%s", "\n");

fprintf(fp, "%d", score);

fclose(fp);

int i = 0;

if (fopen\_s(&fp, "snakefile\\resource\\Ranking\\map3\\num.txt", "r") == 0) {

fscanf\_s(fp, "%d", &i);

fclose(fp);

i++;

fopen\_s(&fp, "snakefile\\resource\\Ranking\\map3\\num.txt", "w+");

fprintf\_s(fp, "%d", i);

fclose(fp);

}

else {

fopen\_s(&fp, "snakefile\\resource\\Ranking\\map3\\num.txt", "w+");

fprintf\_s(fp, "%d", 1);

fclose(fp);

}

}

int getScore() {

FILE \*fp = NULL;

int score = 0;

if (fopen\_s(&fp, "snakefile\\resource\\map3score.txt", "r+") == 0) {

fscanf\_s(fp, "%d", &score);

fclose(fp);

}

return score;

}

void saveScore(int score) {

FILE \*fp = NULL;

fopen\_s(&fp, "snakefile\\resource\\map3score.txt", "w+");

fprintf\_s(fp, "%d", score);

fclose(fp);

}

void isHitWall(Snake \*p) {

if (map[(p->head->x - 7) / 15][(p->head->y - 7) / 15]) {

dead(p);

}

}

void\* updateProgressBar(void \*data) {

//EnterCriticalSection(&cs);

Data \*d = (Data\*)data;

if (d->prelength < d->nowlength) {

for (int i = 0; i < (d->nowlength) - (d->prelength); i++) {

for (int k = 0; k < 7; k++) {

IMAGE green;

IMAGE m;

loadimage(&green, "snakefile\\progressbar1.jpg", 21, 21);

//======

putimage((d->prelength + i - 2) \* 21 + (k + 1) \* 3, 420, &green);

Sleep(30);

}

Sleep((15 \* i - i \* i) / 10 \* 2);

}

}

if (d->prelength > d->nowlength) {

for (int i = 0; i < (d->prelength) - (d->nowlength); i++) {

for (int k = 0; k < 7; k++) {

IMAGE black;

loadimage(&black, "snakefile\\black.jpg", 21, 21);

putimage((d->prelength - i - 1) \* 21 - (k + 1) \* 3, 420, &black);

Sleep(30);

}

Sleep((15 \* i - i \* i) / 10 \* 2);

}

}

pthread\_exit((void \*)0);

return 0;

}

void dead(Snake \*p) {

PlaySound(NULL, NULL, SND\_FILENAME | SND\_ASYNC);

PlaySound(TEXT("snakefile\\gameover.wav"), NULL, SND\_FILENAME | SND\_ASYNC);

IMAGE lose;

loadimage(&lose, "snakefile\\lose.jpg", 630, 444);

putimage(0, 0, &lose);

\_getch();

closegraph();

exit(0);

}

void victory() {

rankingSave();

if (getScore() != 0 && score < getScore()) {

saveScore(score);

}

else if (getScore() == 0) {

saveScore(score);

}

PlaySound(NULL, NULL, SND\_FILENAME | SND\_ASYNC);

putimage(0, 0, &win);

\_getch();

\_getch();

closegraph();

exit(0);

}

void cutTail(Snake \*p, int k) {

p->length -= k;

SnakeBody \*pb = p->tail;

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

pb = pb->pre;

}

while (pb != NULL) {

IMAGE img;

SetWorkingImage(&bk);

getimage(&img, (pb->x) - 7, (pb->y) - 7, 15, 15);

SetWorkingImage();

putimage((pb->x) - 7, (pb->y) - 7, &img);

pb = pb->next;

}

for (int i = 0; i < k; i++) {//删除蛇的链表中的节点

p->tail = p->tail->pre;

SnakeBody \*temp = p->tail->next;

free(temp);

p->tail->next = NULL;

}

Data \*data = (Data\*)malloc(sizeof(Data));

data->prelength = (p->length) + k;

data->nowlength = p->length;

pthread\_t thread;

pthread\_create(&thread, NULL, updateProgressBar, (void\*)data);

}

void isHit(Snake \*p) {//判断是否撞到边界

if (p->head->x < 0 || p->head->x > 640 || p->head->y < 0 || p->head->y > 420) {

dead(p);

}

}

void isDead(Snake \*p) {//判断是否撞到自己

SnakeBody \*pb = p->head->next;

while (pb != NULL) {

if (p->head->x == pb->x && p->head->y == pb->y) {

dead(p);

}

pb = pb->next;

}

}

void isEat(Snake \*p) {

if (fooddata[(p->head->x - 7) / 15][(p->head->y - 7) / 15]) {

p->score++;

fooddata[(p->head->x - 7) / 15][(p->head->y - 7) / 15] = 0;

if (p->speed < 22) {

p->speed++;

}

addHead(p);

Data \*data = (Data\*)malloc(sizeof(Data));

data->prelength = p->length;

data->nowlength = p->length + 1;

p->length++;

p->food = 0;

pthread\_t thread;

pthread\_create(&thread, NULL, updateProgressBar, (void\*)data);

}

}

void isPoison(Snake \*p) {

if (poisondata[(p->head->x - 7) / 15][(p->head->y - 7) / 15]) {

poisondata[(p->head->x - 7) / 15][(p->head->y - 7) / 15] = 0;

if (p->length == 1) {

dead(p);

}

p->poison = 0;

if (poisonstyle[(p->head->x - 7) / 15][(p->head->y - 7) / 15] != 1) {

if (p->speed > 14) {

p->speed--;

}

cutTail(p, 1);

}

else {

cutTail(p, (p->length) / 2);

}

poisonstyle[(p->head->x - 7) / 15][(p->head->y - 7) / 15] = 0;

}

}

void addHead(Snake \*p) {//当吃到食物时，头部增长一节

SnakeBody \*newhead = (SnakeBody\*)malloc(sizeof(SnakeBody));

newhead->x = p->head->x;

newhead->y = p->head->y;

p->head->pre = newhead;

newhead->next = p->head;

newhead->pre = NULL;

p->head = newhead;

if (p->direction == 3 || p->direction == 4) {

for (int k = 0; k < 5; k++) {

int i = (p->dx) / 5;

p->head->x += i;

setfillcolor(RGB(135, 206, 235));

bar((p->head->x) - 7, (p->head->y) - 7, (p->head->x) - 7 + 14, (p->head->y) - 7 + 14);

Sleep(20);

}

}

if (p->direction == 1 || p->direction == 2) {

for (int k = 0; k < 5; k++) {

int i = (p->dy) / 5;

p->head->y += i;

setfillcolor(RGB(135, 206, 235));

bar((p->head->x) - 7, (p->head->y) - 7, (p->head->x) - 7 + 14, (p->head->y) - 7 + 14);

Sleep(20);

}

}

}

void slowMove(Snake \*p) {//正常移动，头伸尾减

SnakeBody \*newhead = (SnakeBody\*)malloc(sizeof(SnakeBody));

newhead->x = p->head->x;

newhead->y = p->head->y;

p->head->pre = newhead;

newhead->next = p->head;

newhead->pre = NULL;

p->head = newhead;

if (p->direction == 3 || p->direction == 4) {

for (int k = 0; k < 5; k++) {

int i = (p->dx) / 5;

p->head->x += i;

setfillcolor(RGB(135, 206, 235));

bar((p->head->x) - 7, (p->head->y) - 7, (p->head->x) - 7 + 14, (p->head->y) - 7 + 14);

printTail(p, k);

Sleep(p->speed);

}

}

if (p->direction == 1 || p->direction == 2) {

for (int k = 0; k < 5; k++) {

int i = (p->dy) / 5;

p->head->y += i;

setfillcolor(RGB(135, 206, 235));

bar((p->head->x) - 7, (p->head->y) - 7, (p->head->x) - 7 + 14, (p->head->y) - 7 + 14);

printTail(p, k);

Sleep(p->speed);

}

}

SnakeBody \*temp = p->tail;

p->tail = p->tail->pre;

free(temp);

p->tail->next = NULL;

}

void printTail(Snake \*p, int k) {//打印尾巴，分五次打印，以保证流畅

int dx = (p->tail->x) - (p->tail->pre->x);

int dy = (p->tail->y) - (p->tail->pre->y);

dx = -dx;

dy = -dy;

if (dx != 0) {

IMAGE img;

int i = (dx / 5)\*(k + 1);

SetWorkingImage(&bk);

getimage(&img, (p->tail->x) - 7, (p->tail->y) - 7, 15, 15);

SetWorkingImage(&img);

setfillcolor(RGB(135, 206, 235));

bar(i, 0, i + 14, 14);

SetWorkingImage();

putimage((p->tail->x) - 7, (p->tail->y) - 7, &img);

}

if (dy != 0) {

IMAGE img;

int i = (dy / 5)\*(k + 1);

SetWorkingImage(&bk);

getimage(&img, (p->tail->x) - 7, (p->tail->y) - 7, 15, 15);

SetWorkingImage(&img);

setfillcolor(RGB(135, 206, 235));

bar(0, i, 14, i + 14);

SetWorkingImage();

putimage((p->tail->x) - 7, (p->tail->y) - 7, &img);

}

}

void printSnake(Snake \*p) {//打印整条蛇的函数

SnakeBody \*pb = p->head;

setfillcolor(RGB(135, 206, 235));

while (pb->next != NULL) {

bar((pb->x) - 7, (pb->y) - 7, (pb->x) - 7 + 14, (pb->y) - 7 + 14);

pb = pb->next;

}

bar((pb->x) - 7, (pb->y) - 7, (pb->x) - 7 + 14, (pb->y) - 7 + 14);

}

void createFood(Snake \*p) {//创造食物

p->food = 1;

srand((unsigned)time(NULL));

int x = 0;

int y = 0;

x = rand() % 42;

y = rand() % 28;

Food \*food = (Food\*)malloc(sizeof(Food));

food->x = x \* 15 + 7;

food->y = y \* 15 + 7;

while (map[x][y] || fooddata[x][y] || poisondata[x][y] || isInSnakeBody(p, food)) {

x = rand() % 42;

y = rand() % 28;

food->x = x \* 15 + 7;

food->y = y \* 15 + 7;

}

fooddata[x][y] = 1;

putimage((food->x) - 10, (food->y) - 10, &f2, NOTSRCERASE);//掩码

putimage((food->x) - 10, (food->y) - 10, &f1, SRCINVERT);//精灵

free(food);

}

void createPoison(Snake \*p) {//创造毒草

p->poison = 1;

srand((unsigned)time(NULL));

Poison \*poison = (Poison\*)malloc(sizeof(Poison));

int x = 0;

int y = 0;

x = rand() % 42;

y = rand() % 28;

poison->x = x \* 15 + 7;

poison->y = y \* 15 + 7;

while (map[x][y] || fooddata[x][y] || poisondata[x][y] || isInSnakeBody(p, poison)) {

x = rand() % 42;

y = rand() % 28;

poison->x = x \* 15 + 7;

poison->y = y \* 15 + 7;

}

poisondata[x][y] = 1;

poisonstyle[x][y] = 1;

putimage((poison->x) - 7, (poison->y) - 7, &boom2, NOTSRCERASE);//掩码

putimage((poison->x) - 7, (poison->y) - 7, &boom1, SRCINVERT);//精灵

free(poison);

}

Snake\* createSnake() {//创造蛇

Snake \*p;

SnakeBody \*head;

SnakeBody \*tail;

p = (Snake\*)malloc(sizeof(Snake));

head = (SnakeBody\*)malloc(sizeof(SnakeBody));

tail = (SnakeBody\*)malloc(sizeof(SnakeBody));

head->x = 1 \* 15 + 7;

head->y = 0 \* 15 + 7;

tail->x = 0 \* 15 + 7;

tail->y = 0 \* 15 + 7;

head->next = tail;

head->pre = NULL;

tail->next = NULL;

tail->pre = head;

p->head = head;

p->tail = tail;

p->food = 0;

p->poison = 0;

p->star = 0;

p->length = 1;

p->direction = 4;

p->speed = 22;

p->dx = 15;

p->dy = 0;

p->pfood = NULL;

p->ppoison = NULL;

p->score = 0;

return p;

}

//============结构体===========================

typedef struct Star {

int x;

int y;

}Star;

typedef struct Food {//保存食物坐标

int x;

int y;

Food \*next;

Food \*pre;

}Food;

typedef struct Poison {//保存毒草的坐标以及产生的所有毒草

int x;

int y;

int style;

Poison \*next;

Poison \*pre;

}Poison;

typedef struct SnakeBody {//蛇的身体，保存节点

int x;

int y;

struct SnakeBody \*next;

struct SnakeBody \*pre;

}SnakeBody;

typedef struct Snake {//管理蛇的头和尾，还有各种参数

SnakeBody \*head;

SnakeBody \*tail;

int dx; int dy;

int food; int poison; int star;

Food \*pfood;//用来保存产生的食物和毒草的链表头

Poison \*ppoison; Star \*pstar;

int length; int direction;

int speed; int score;

int poisonNum; int foodNum;

int foodRate; int poisonRate;

int starRate; int mushroom;

}Snake;

typedef struct Data {

int prelength;

int nowlength;

}Data;

//===================保存进度===========================

void saveProgress(Snake \*p) {

IMAGE save;

getimage(&save, 0, 0, 630, 420);

saveimage("d:\\snakefile\\resource\\InfiniteResource\\save.jpg", &save);

//=============================

FILE \*fp = NULL;

fopen\_s(&fp, "d:\\snakefile\\resource\\InfiniteResource\\snake.txt", "w+");

SnakeBody \*sp = p->head;

while (sp != NULL) {

fprintf\_s(fp, "%d", sp->x);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", sp->y);

fprintf\_s(fp, "%s", "\n");

sp = sp->next;

}

fclose(fp);

//==============================

fopen\_s(&fp, "d:\\snakefile\\resource\\InfiniteResource\\poison.txt", "w+");

Poison \*poison = p->ppoison;

while (poison != NULL) {

fprintf\_s(fp, "%d", poison->x);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", poison->y);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", poison->style);//1是土豆雷

fprintf\_s(fp, "%s", "\n");

poison = poison->next;

}

fclose(fp);

//==============================

fopen\_s(&fp, "d:\\snakefile\\resource\\InfiniteResource\\food.txt", "w+");

Food \*food = p->pfood;

while (food != NULL) {

fprintf\_s(fp, "%d", food->x);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", food->y);

fprintf\_s(fp, "%s", "\n");

food = food->next;

}

fclose(fp);

//==============================

fopen\_s(&fp, "d:\\snakefile\\resource\\InfiniteResource\\snakedata.txt", "w+");

fprintf\_s(fp, "%d", p->direction);//方向

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->length);//长度

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->score);//分数

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->speed);//速度

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->food);//是否有食物

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->poison);//是否有毒草

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->star);//是否有星星

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->foodRate);//食物产生速率

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->poisonRate);//毒草产生速率

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->starRate);//星星产生速率

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->foodNum);//食物数量

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->poisonNum);//毒草数量

fprintf\_s(fp, "%s", "\n");

fclose(fp);

if (p->star) {

fopen\_s(&fp, "d:\\snakefile\\resource\\InfiniteResource\\star.txt", "w+");

fprintf\_s(fp, "%d", p->pstar->x);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%d", p->pstar->y);

fprintf\_s(fp, "%s", "\n");

fclose(fp);

}

}

//===========读取保存的进度======================

Snake\* readSave() {

IMAGE save;

loadimage(&save, "snakefile\\resource\\InfiniteResource\\save.jpg", 630, 420);

putimage(0, 0, &save);

//==========================================

Snake \*p = (Snake\*)malloc(sizeof(Snake));

p->head = NULL;

p->pfood = NULL;

p->ppoison = NULL;

p->pstar = NULL;

//==========================================

FILE \*fp;

fopen\_s(&fp, "snakefile\\resource\\InfiniteResource\\snakedata.txt", "r");

fscanf\_s(fp, "%d", &(p->direction));

fscanf\_s(fp, "%d", &(p->length));

fscanf\_s(fp, "%d", &(p->score));

fscanf\_s(fp, "%d", &(p->speed));

fscanf\_s(fp, "%d", &(p->food));

fscanf\_s(fp, "%d", &(p->poison));

fscanf\_s(fp, "%d", &(p->star));

fscanf\_s(fp, "%d", &(p->foodRate));

fscanf\_s(fp, "%d", &(p->poisonRate));

fscanf\_s(fp, "%d", &(p->starRate));

fscanf\_s(fp, "%d", &(p->foodNum));

fscanf\_s(fp, "%d", &(p->poisonNum));

fclose(fp);

//===========================================

if (p->direction == 1) {

p->dx = 0;

p->dy = -15;

}

if (p->direction == 2) {

p->dx = 0;

p->dy = 15;

}

if (p->direction == 3) {

p->dx = -15;

p->dy = 0;

}

if (p->direction == 4) {

p->dx = 15;

p->dy = 0;

}

//==========================================

fopen\_s(&fp, "snakefile\\resource\\InfiniteResource\\snake.txt", "r");

int k = 0;

SnakeBody \*sp = NULL;

while (k <= p->length) {

SnakeBody \*temp = (SnakeBody\*)malloc(sizeof(SnakeBody));

temp->pre = NULL;

temp->next = NULL;

fscanf\_s(fp, "%d", &(temp->x));

fscanf\_s(fp, "%d", &(temp->y));

if (p->head == NULL) {

p->head = temp;

}

if (sp != NULL) {

sp->next = temp;

temp->pre = sp;

}

sp = temp;

if (k == p->length) {

p->tail = sp;

}

k++;

}

fclose(fp);

//==========================================

fopen\_s(&fp, "snakefile\\resource\\InfiniteResource\\poison.txt", "r");

k = 0;

Poison \*poison = NULL;

while (k < p->poisonNum) {

Poison \*ps = (Poison\*)malloc(sizeof(Poison));

ps->next = NULL;

ps->pre = NULL;

fscanf\_s(fp, "%d", &(ps->x));

fscanf\_s(fp, "%d", &(ps->y));

fscanf\_s(fp, "%d", &(ps->style));

if (p->ppoison == NULL) {

p->ppoison = ps;

}

if (poison != NULL) {

poison->next = ps;

ps->pre = poison;

}

poison = ps;

k++;

}

fclose(fp);

//==========================================

fopen\_s(&fp, "snakefile\\resource\\InfiniteResource\\food.txt", "r");

k = 0;

Food \*food = NULL;

while (k < p->foodNum) {

Food \*f = (Food\*)malloc(sizeof(Food));

f->next = NULL;

f->pre = NULL;

fscanf\_s(fp, "%d", &(f->x));

fscanf\_s(fp, "%d", &(f->y));

if (p->pfood == NULL) {

p->pfood = f;

}

if (food != NULL) {

food->next = f;

f->pre = food;

}

food = f;

k++;

}

fclose(fp);

//===========================================

if (p->star) {

fopen\_s(&fp, "snakefile\\resource\\InfiniteResource\\star.txt", "r");

Star \*s = (Star\*)malloc(sizeof(Star));

fscanf\_s(fp, "%d", &(s->x));

fscanf\_s(fp, "%d", &(s->y));

p->pstar = s;

fclose(fp);

}

return p;

}

//===========判断产生位置是否重复==================

#pragma once

#include "struct.h"

#include <stdio.h>

int isInFood(Snake \*p, Poison \*poison) {//Food \*newfood/ Star \*star

Food \*food = NULL;

if (p->pfood != NULL) {

food = p->pfood;

}

else {

return 0;

}

while (food->next != NULL) {

if (poison->x == food->x && poison->y == food->y) {

return 1;

}

food = food->next;

}

if (poison->x == food->x && poison->y == food->y) {

return 1;

}

return 0;

}

int isInPoison(Snake \*p, Poison \*newpoison) { //Star \*star/ Food \*food

Poison \*poison = NULL;

if (p->ppoison != NULL) {

poison = p->ppoison;

}

else {

return 0;

}

while (poison->next != NULL) {

if (newpoison->x == poison->x && newpoison->y == poison->y) {

return 1;

}

poison = poison->next;

}

if (newpoison->x == poison->x && newpoison->y == poison->y) {

return 1;

}

return 0;

}

int isInSnakeBody(Snake \*p, Food \*food) {// Poison \*poison/Star \*star/

SnakeBody \*pb = p->head;

while (pb->next != NULL) {

if (food->x == pb->x && food->y == pb->y) {

return 1;

}

pb = pb->next;

}

if (food->x == pb->x && food->y == pb->y) {

return 1;

}

return 0;

}

//==============菜单===============================

#include <stdio.h> #include <conio.h> #include <Windows.h>

#include <graphics.h> #include <string.h> #include "ImageMove.h"

#include "getRankRelease.h" //enter 13 空格 32 ESC 27

//UP -32 72 DOWN -32 80 LEFT -32 75 RIGHT -32 77

int main() {

Loadin();

return 0;

}

void Loadin() {

char name[20];

char c;

int n = 0;

char password[50] = "";

system("mode con cols=40 lines=10");

while (1) {

n = 0;

system("cls");

printf("1.登陆\n");

printf("2.试玩\n");

printf("3.注册账号\n");

printf("Esc(退出)\n");

c = \_getch();

if (c == 27) {

exit(0);

}

else if (c == '2') {

FILE \*t = NULL;

fopen\_s(&t, "snakefile\\resource\\Ranking\\player.txt", "w+");

fprintf\_s(t, "%s", "游客");

fclose(t);

system("map1.exe");

}

else if (c == '1') {

system("cls");

printf("请输入用户名：");

scanf\_s("%s", name, 20);

printf("请输入密码：");

char ch;

char str[20];

int n = 1;

strcpy\_s(password,sizeof(password),"");

while ((ch = \_getch()) != 13) {

if (ch == 8 && n != 1) {

int i = 0;

while (password[i + 1] != '\0') {

i++;

}

password[i] = '\0';

printf("\b \b");

n--;

}

else if(ch != 8){

printf("\*");

n++;

sprintf\_s(str, "%c", ch);

strcat\_s(password, n, str);

}

}

//scanf\_s("%s", password, 20);

FILE \*fp = NULL;

FILE \*f = NULL;

if (fopen\_s(&fp, "snakefile\\resource\\UserData\\data.txt", "r") != 0) {

printf("error\n");

Sleep(500);

}

else {

fopen\_s(&f, "snakefile\\resource\\UserData\\num.txt", "r");

int num = 0;

int k = 0;

char pw[50];

char str[20];

fscanf\_s(f, "%d", &num);

while (k < num) {

fscanf\_s(fp, "%s", str, 20);

fscanf\_s(fp, "%s", pw, 50);

if (strcmp(name, str) == 0 && strcmp(password, pw) == 0) {

FILE \*t = NULL;

fopen\_s(&t, "snakefile\\resource\\Ranking\\player.txt", "w+");

fprintf\_s(t, "%s", name);

fclose(t);

SnakeBegin();

n = 1;

}

k++;

}

fclose(fp);

fclose(f);

if (n == 0) {

printf("error\n");

Sleep(500);

}

}

}

else if (c == '3') {

system("cls");

printf("请输入用户名：");

scanf\_s("%s", name, 20);

printf("请输入密码：");

char ch;

char str[20];

int n = 1;

strcpy\_s(password, sizeof(password), "");

while ((ch = \_getch()) != 13) {

if (ch == 8 && n != 1) {

int i = 0;

while (password[i + 1] != '\0') {

i++;

}

password[i] = '\0';

printf("\b \b");

n--;

}

else if(ch != 8){

printf("\*");

n++;

sprintf\_s(str, "%c", ch);

strcat\_s(password, n, str);

}

}

//scanf\_s("%s", password, 20);

FILE \*fp = NULL;

if (fopen\_s(&fp, "snakefile\\resource\\UserData\\data.txt", "a+") == 0) {

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%s", name);

fprintf\_s(fp, "%s", "\n");

fprintf\_s(fp, "%s", password);

fclose(fp);

}

int i = 0;

if (fopen\_s(&fp, "snakefile\\resource\\UserData\\num.txt", "r") == 0) {

fscanf\_s(fp, "%d", &i);

fclose(fp);

i++;

fopen\_s(&fp, "snakefile\\resource\\UserData\\num.txt", "w+");

fprintf\_s(fp, "%d", i);

fclose(fp);

}

else {

fopen\_s(&fp, "snakefile\\resource\\UserData\\num.txt", "w+");

fprintf\_s(fp, "%d", 1);

fclose(fp);

}

system("cls");

printf("注册成功");

Sleep(500);

}

}

}

void SnakeBegin() {

initgraph(630, 420);

IMAGE begin, newgame, choose, ranking, con, rankingbk;

IMAGE map1, map2, map3, map4;

IMAGE map1rank, map2rank, map3rank, map4rank;

int style = 0;

char c;

loadimage(&begin, "snakefile\\begin.jpg", 630, 420);

loadimage(&newgame, "snakefile\\newgame.jpg", 630, 420);

loadimage(&choose, "snakefile\\choose.jpg", 630, 420);

loadimage(&ranking, "snakefile\\ranking.jpg", 630, 420);

loadimage(&con, "snakefile\\continue.jpg", 630, 420);

loadimage(&rankingbk, "d:\\snakefile\\rankingbk.jpg", 630, 420);

loadimage(&map1, "snakefile\\map-1.jpg", 630, 420);

loadimage(&map2, "snakefile\\map-2.jpg", 630, 420);

loadimage(&map3, "snakefile\\map-3.jpg", 630, 420);

loadimage(&map4, "snakefile\\map-4.jpg", 630, 420);

putimage(0, 0, &begin);

while (1) {

if (\_kbhit()) {

c = \_getch();

if (c == 27) {

closegraph();

exit(0);

}

if (style == 11 && c == 13) {

system("infinitesave.exe");

style = 1;

putimage(0, 0, &newgame);

}

else if (style == 1 && c == 13) {

system("infinite.exe");

}

else if (style == 21 && c == 13) {

system("map1.exe");

style = 1;

putimage(0, 0, &newgame);

}

else if (style == 22 && c == 13) {

system("map2.exe");

style = 1;

putimage(0, 0, &newgame);

}

else if (style == 23 && c == 13) {

system("map3.exe");

style = 1;

putimage(0, 0, &newgame);

}

else if (style == 24 && c == 13) {

system("map4.exe");

style = 1;

putimage(0, 0, &newgame);

}

if (style == 0) {

imageUp(begin, newgame);

style = 1;

}

else if (c == -32) {

c = \_getch();

if (style == 34 && c == 80) {

loadimage(&map4rank, "snakefile\\rankingbk.jpg", 630, 420);

SetWorkingImage(&map4rank);

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

outtextxy(230, 60, \_T("地图四排行榜"));

getMapDRank();

SetWorkingImage();

imageUp(map3rank, map4rank);

style = 35;

}

else if (style == 35 && c == 72) {

imageDown(map4rank, map3rank);

style = 34;

}

else if (style == 33 && c == 80) {

loadimage(&map3rank, "snakefile\\rankingbk.jpg", 630, 420);

SetWorkingImage(&map3rank);

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

outtextxy(230, 60, \_T("地图三排行榜"));

getMapCRank();

SetWorkingImage();

imageUp(map2rank, map3rank);

style = 34;

}

else if (style == 34 && c == 72) {

imageDown(map3rank, map2rank);

style = 33;

}

else if (style == 32 && c == 80) {

loadimage(&map2rank, "snakefile\\rankingbk.jpg", 630, 420);

SetWorkingImage(&map2rank);

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

outtextxy(230, 60, \_T("地图二排行榜"));

getMapBRank();

SetWorkingImage();

imageUp(map1rank, map2rank);

style = 33;

}

else if (style == 33 && c == 72) {

imageDown(map2rank, map1rank);

style = 32;

}

else if (style == 31 && c == 80) {

loadimage(&map1rank, "snakefile\\rankingbk.jpg", 630, 420);

SetWorkingImage(&map1rank);

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

outtextxy(230, 60, \_T("地图一排行榜"));

getMapARank();

SetWorkingImage();

imageUp(rankingbk, map1rank);

style = 32;

}

else if (style == 32 && c == 72) {

imageDown(map1rank, rankingbk);

style = 31;

}

else if (style == 3 && c == 80) {

loadimage(&rankingbk, "snakefile\\rankingbk.jpg", 630, 420);

SetWorkingImage(&rankingbk);

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

outtextxy(230, 60, \_T("无尽模式排行榜"));

getInfiniteRank();

SetWorkingImage();

imageUp(ranking, rankingbk);

style = 31;

}

else if (style == 31 && c == 72) {

imageDown(rankingbk, ranking);

style = 3;

}

else if (style == 1 && c == 80) {

imageUp(newgame, con);

style = 11;

}

else if (style == 11 && c == 72) {

imageDown(con, newgame);

style = 1;

}

else if (style == 1 && c == 77) {

//imageLeft(newgame, choose);

putimage(0, 0, &choose);

style = 2;

}

else if (style == 2 && c == 75) {

//imageRight(choose,newgame);

putimage(0, 0, &newgame);

style = 1;

}

else if (style == 2 && c == 77) {

//imageLeft(choose, ranking);

putimage(0, 0, &ranking);

style = 3;

}

else if (style == 3 && c == 75) {

//imageRight(ranking,choose);

putimage(0, 0, &choose);

style = 2;

}

else if (style == 2 && c == 80) {//从此以下为具体地图

imageUp(choose, map1);

style = 21;

}

else if (style == 21 && c == 80) {

imageUp(map1, map2);

style = 22;

}

else if (style == 22 && c == 80) {

imageUp(map2, map3);

style = 23;

}

else if (style == 23 && c == 80) {

imageUp(map3, map4);

style = 24;

}

else if (style == 21 && c == 72) {

imageDown(map1, choose);

style = 2;

}

else if (style == 22 && c == 72) {

imageDown(map2, map1);

style = 21;

}

else if (style == 23 && c == 72) {

imageDown(map3, map2);

style = 22;

}

else if (style == 24 && c == 72) {

imageDown(map4, map3);

style = 23;

}

}

}

}

\_getch();

closegraph();

}

//===============读取排行榜================

void getMapDRank() {

setbkmode(TRANSPARENT);

settextstyle(25, 0, \_T("宋体"));

FILE \*fp = NULL;

if (fopen\_s(&fp, "snakefile\\resource\\Ranking\\map4\\num.txt", "r") != 0) {

outtextxy(230, 150, \_T("No Information"));

}

else {

int k = 0;

int n = 0;

char name[20];

int sc = 0;

char first[20] = "NULL"; char second[20] = "NULL"; char third[20] = "NULL";

int score[3] = { 1000,1000,1000 };

fscanf\_s(fp, "%d", &n);

fclose(fp);

fopen\_s(&fp, "snakefile\\resource\\Ranking\\map4\\ranking.txt", "r");

while (k < n) {

fscanf\_s(fp, "%s", name, 20);

fscanf\_s(fp, "%d", &sc);

if (sc < score[0]) {

strcpy\_s(third, second);

strcpy\_s(second, first);

strcpy\_s(first, name);

score[2] = score[1];

score[1] = score[0];

score[0] = sc;

}

else if (sc < score[1]) {

strcpy\_s(third, second);

strcpy\_s(second, name);

score[2] = score[1];

score[1] = sc;

}

else if (sc < score[2]) {

strcpy\_s(third, name);

score[2] = sc;

}

k++;

}

fclose(fp);

for (int i = 0; i < 3; i++) {

if (score[i] == 1000) {

score[i] = 0;

}

}

char temp[20];

settextstyle(20, 0, \_T("宋体"));

sprintf\_s(temp, "%d", score[0]);

outtextxy(240, 120, first);

outtextxy(340, 120, temp);

sprintf\_s(temp, "%d", score[1]);

outtextxy(240, 180, second);

outtextxy(340, 180, temp);

sprintf\_s(temp, "%d", score[2]);

outtextxy(240, 240, third);

outtextxy(340, 240, temp);

}

}

//===============上下移动图片============

void imageDown(IMAGE a, IMAGE b) {

IMAGE temp1;

IMAGE temp2;

putimage(0, 0, &a);

for (int k = 2; k <= 420; k += 2) {//下移

SetWorkingImage(&a);

getimage(&temp2, 0, 0, 630, 420 - k);

SetWorkingImage(&b);

getimage(&temp1, 0, 420 - k, 630, k);

SetWorkingImage();

putimage(0, 0, &temp1);

putimage(0, k, &temp2);

Sleep(2);

}

}

void imageUp(IMAGE a, IMAGE b) {

IMAGE temp1;

IMAGE temp2;

putimage(0, 0, &a);

for (int k = 2; k <= 420; k += 2) {//上移

SetWorkingImage(&a);

getimage(&temp1, 0, k, 630, 420 - k);

SetWorkingImage(&b);

getimage(&temp2, 0, 0, 630, k);

SetWorkingImage();

putimage(0, 0, &temp1);

putimage(0, 420 - k, &temp2);

Sleep(2);

}

}