

创新实践报告

题目 贪吃蛇小游戏

创新实践小组成员

姓名 邵子奇 学号 201326811416

姓名 谢云昊 学号 201326811019

姓名 周涤心 学号 201326810129

班级 计算机实验班1301班

目录

一．关于贪吃蛇——————————P3

二．功能设定———————————P3

三．借鉴对象———————————P4

四．成员分工———————————P4

五．时间进程———————————P5

六．运行结果———————————P6

七．主要代码框架—————————P7

八．遇到的问题——————————P8

九．缺陷与不足——————————P9

十．心得体会———————————P10

附录：具体代码及注释———————P11

一．关于贪吃蛇

贪吃蛇又名贪食蛇，是一款经典的小游戏。玩家使用方向键操控一条长长的蛇不断吞下食物，同时蛇身随着吞下的食物不断变长，当蛇头撞到蛇身或障壁时游戏结束。贪吃蛇最初为人们所知的是诺基亚手机附带的一个小游戏，它伴随着诺基亚手机走向世界。现在的贪吃蛇出现了许多衍生版本，并被移植到各种平台上。

二．功能设定

贪吃蛇作为一款经典的游戏，具有许许多多的衍生版本。我们基于娱乐性和可玩性，为我们的贪吃蛇创新性的设定了如下几个功能块。

**单人游戏：**蛇速不仅随着蛇身变化，而且随着音乐节奏变化。

**多人对战：**实现多台机器联机对战。

**人机对战：**实现人机对战。

**暂停继续：**实现游戏的暂停和继续。

**游戏美化：**添加开场动画和结束动画，背景音乐，背景图片等使界面和整个过程流畅美观。

三．借鉴对象

没有源代码，主要借鉴一篇新浪上的博文《手把手教MFC贪吃蛇》



该博文内容详见文件附带word《手把手教MFC贪吃蛇》

四．成员分工

**邵子奇：**主要代码的编写和注释。

**谢云昊：**word文档的整理。

**周涤心：**游戏美化部分动画和图片的编辑，word文档和PPT的编辑和整理。

**外挂：吴建涛：**通信部分，图像导入部分。

五．时间进程

**总体时间：**

第四周----第十六周

**具体进程：**

第四周----第八周：研究基于控制台的贪吃蛇程序。

第九周——第十周：研究并学习基于MFC的贪吃蛇程序。

第十一周：实现了双人游戏。

第十二周：实现了人机对战。

第十三周：加入背景音乐，实现了速度随音乐节奏变化。

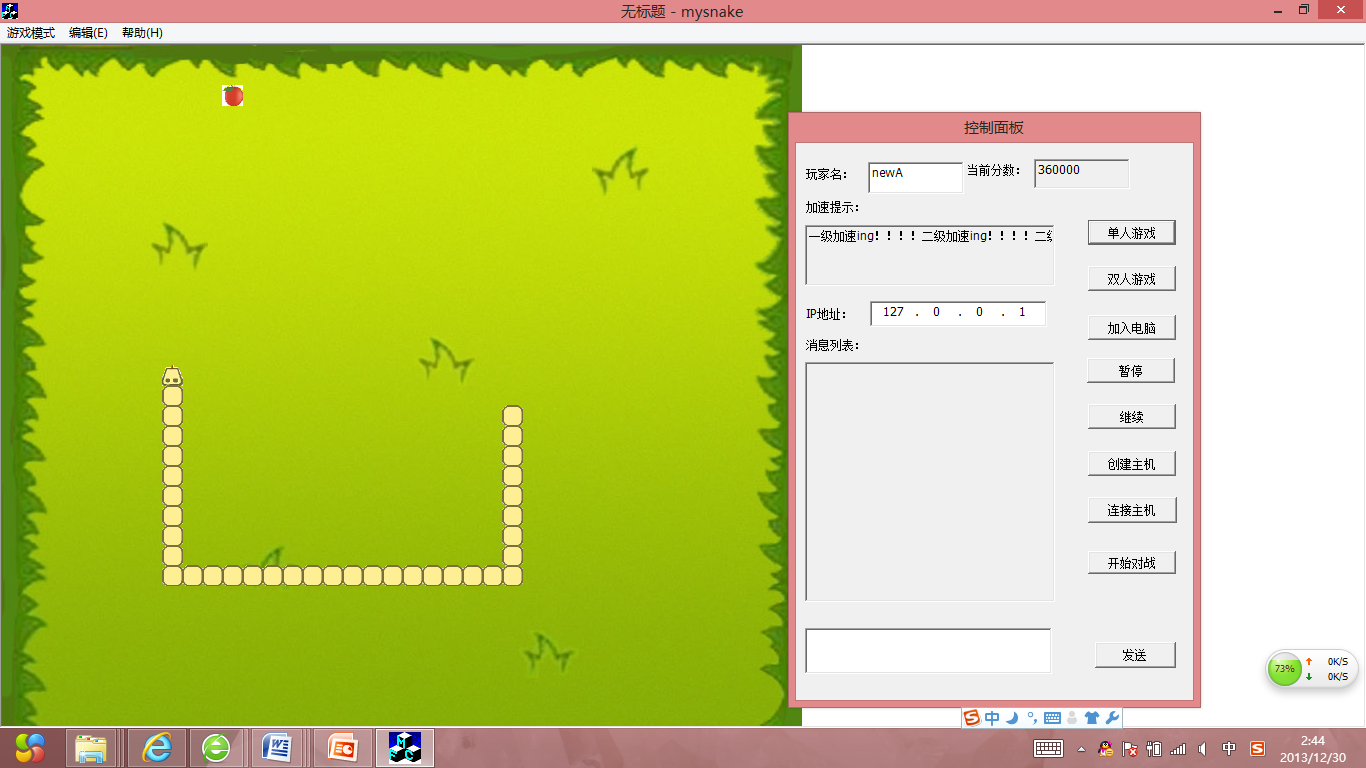
第十四周：全面更新整理代码。添加了游戏美化部分和联机部分。

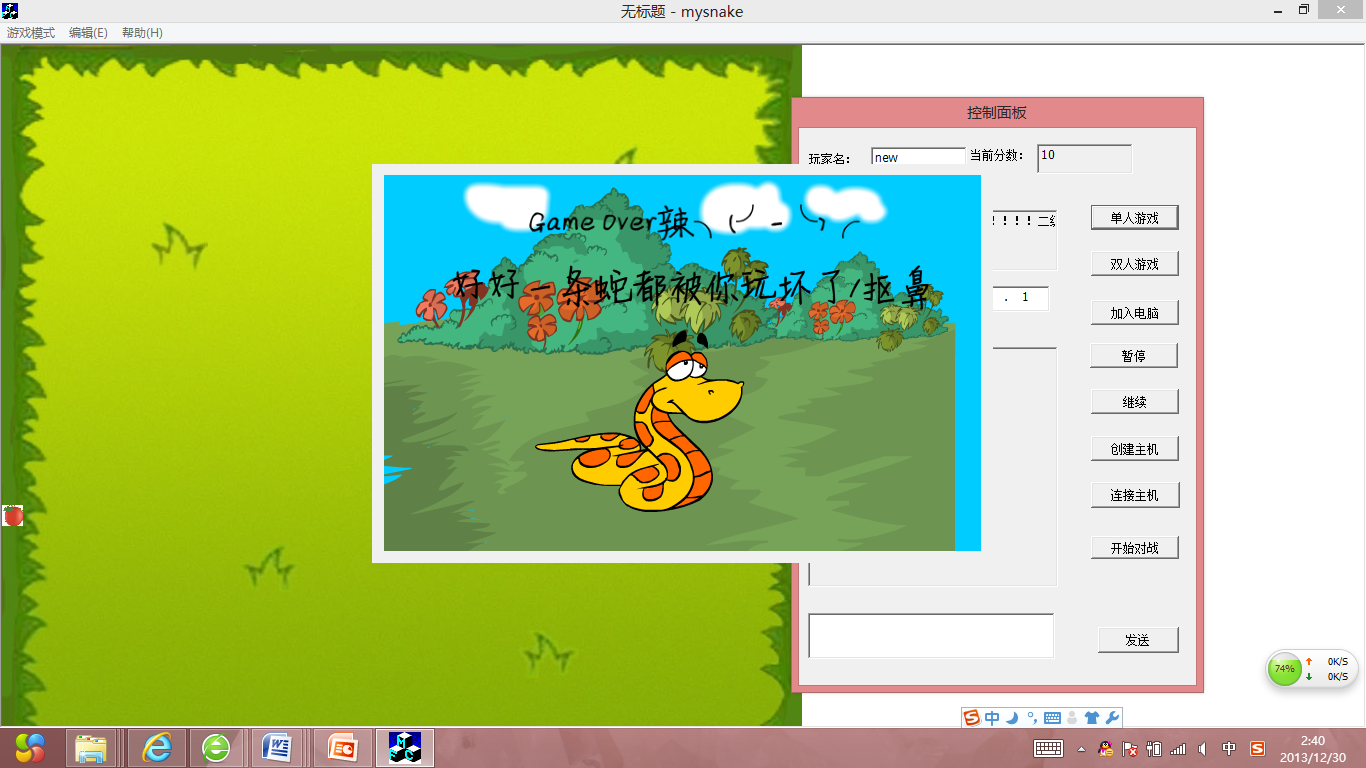
第十五周：继续攻克联机延迟问题，最后修复各种Bug。

第十六周：整合完成最终成果。上交并展示。

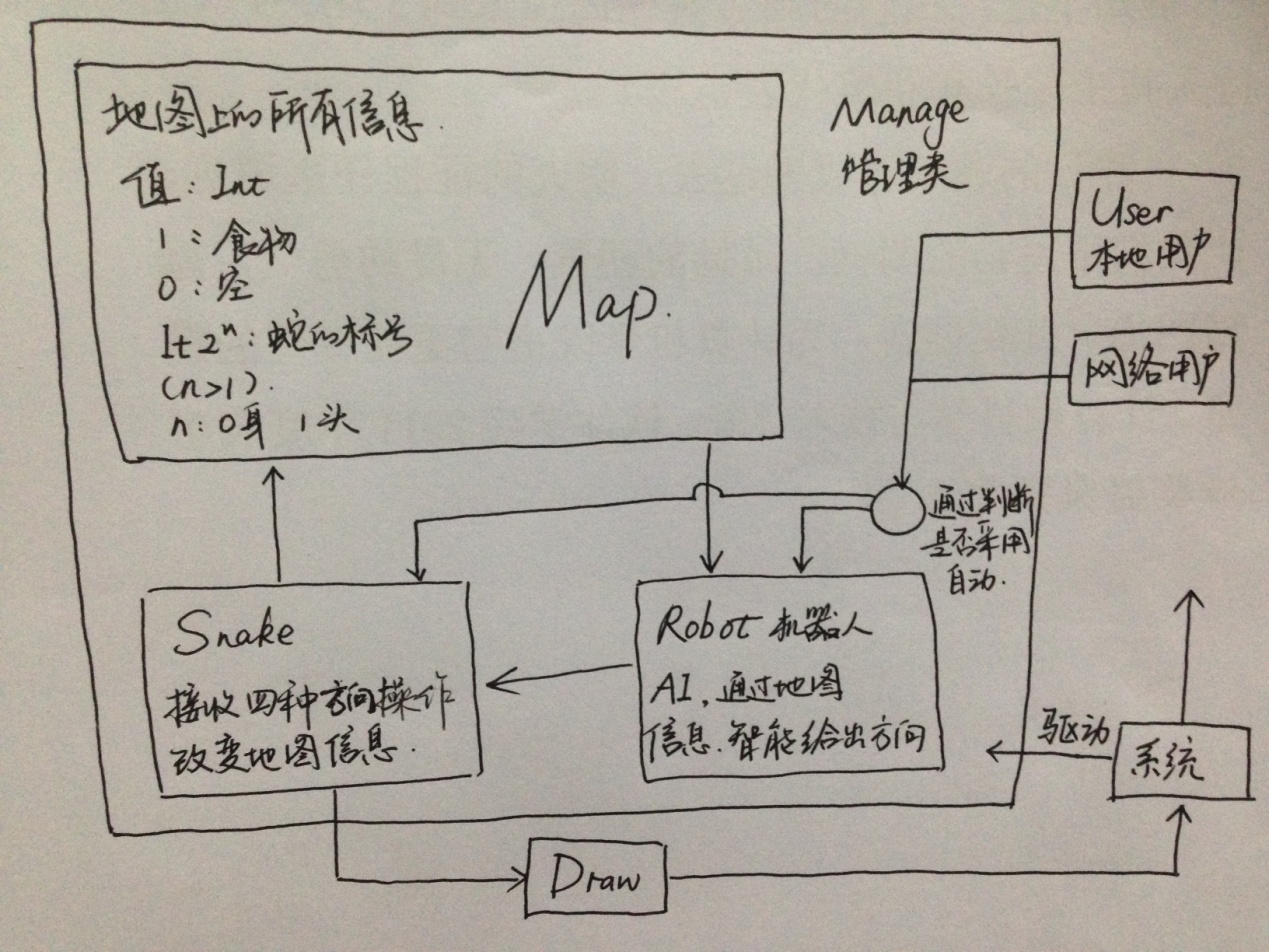
六．运行结果







七．主要代码框架



八．遇到的问题

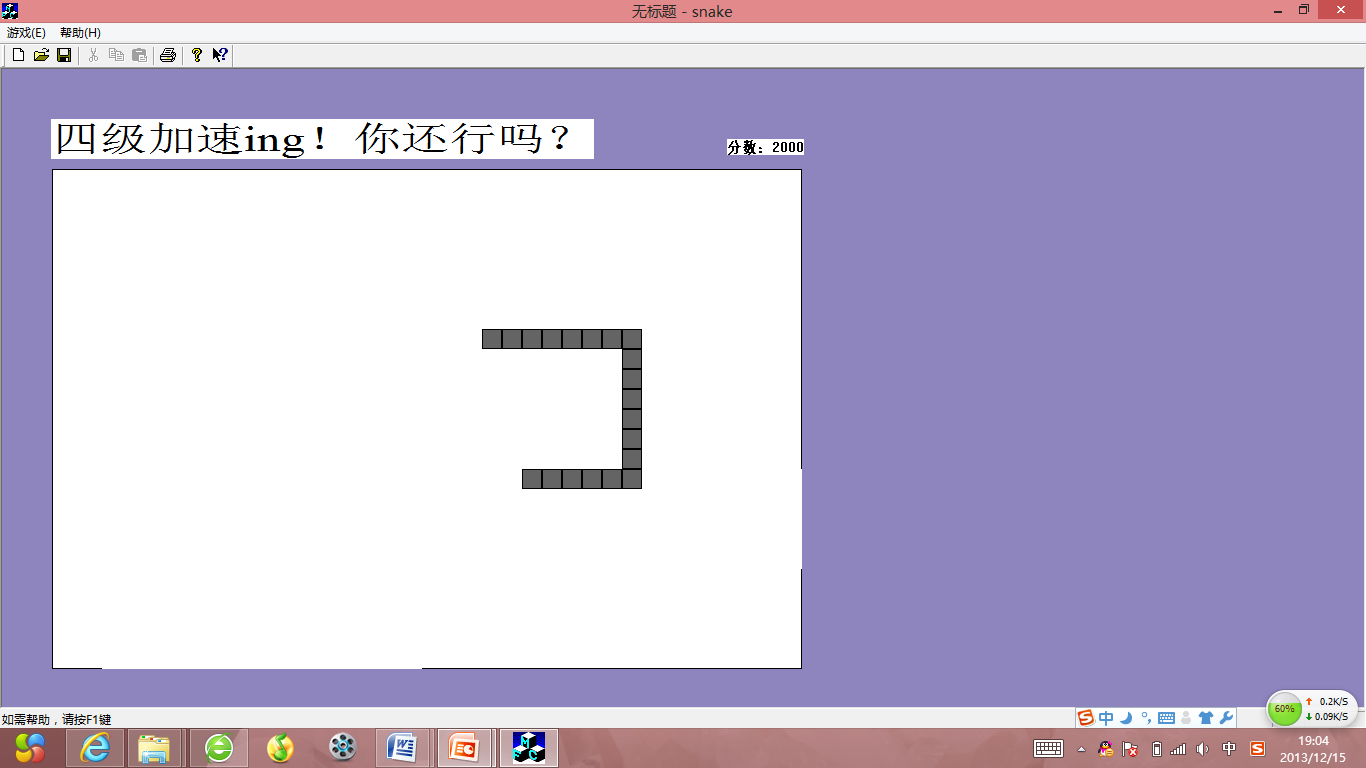
1、**遇到问题：**开始两条蛇的时候，新添加的蛇不能得到限制，如可以跑出界面，可以撞自己，两条蛇之间可以交叉，重叠等。

**解决方案：**增加对新添加的蛇的撞界判断。

2、**遇到问题：**人机对战部分机器部分不够智能，只会找最短路径去找食物但是不会躲避。

**解决方案：**重新编写机器控制部分的代码。

3、**遇到问题：**贪吃蛇会吃着吃着不出现食物。



**解决方案：**食物出现在了蛇的身上。所以在代码编辑的过程中，生成食物前，先判断是否有蛇的标记。

4、**遇到问题：**加入音乐之后不能操作了。

**解决方案：**用了MCIsendstring的函数。

**5、遇到问题：**导入自己的图片很麻烦，图片会闪。

**解决方法：**放弃原来用自带的CDC作图，避免了指针传递中出现的各种问题，去网上下载了一个Cbitmapex的类。

**6、遇到问题：**通讯问题里，用自带的SoCket无法创建

**解决方法：**发现是在创建工程时没有打上这个选项的勾，然后放弃用这个类。使用通信的基础函数。

在使用这些函数时，又发现serve端时accept 会塞死程序会卡住。Recv函数也同样会卡。因此就采用了多线程，使用afxwinthread,

通信中只能传递字符串，所以必须建立相对应的功能映射。

**7、遇到问题：**游戏同步问题

**解决方法：**尚未完美解决。现在只能放慢速度。

九．缺陷与不足

这次实践最后联机的延迟问题还未解决。整个游戏还不够精致，例如没有游戏介绍，按键说明等，用户适用性不强。整个实践过程中，小组成员的沟通不够好，分工不够明确合理。

十．心得体会

**谢云昊：**这次试验的所有代码编辑与修改我都未参与，主要是不会做，小懒惰，所以谈不上什么。我看过类似书，目前感觉很难，看了没感觉，没收获。但通过这次学习过程，让我认识到自己必须尽快学习w32 mfc 。通过学习这些代码，我越发觉得下学期要学的类很重要，因为对类知识不知道，对函数库不熟悉，没学如何建立工程，所以对我来说编程还谈不上，我现在会的不过是部分基础知识，简单的算法。

**邵子奇：**在这次的创新实践中，我感受到了编写一个工程的不易。但在整个过程中个，我获取信息的能力有较大的提升，对MFC也有了初步的了解。我希望在下次的创新实践中可以做的更好。

**周涤心：**这次的创新实践中，我体会到了，小组是一个团队，每个人会的和擅长的东西都不一样，正是每个人都贡献了自己的力量，才能完成整个项目。在这次创新实践中，我很感谢子奇，因为我们之中就他学习了MFC，所以承担起来大部分编写代码的工作。而我也利用了自己的所长，为贪吃蛇小游戏编辑了动画和背景图片以及负责整理word和PPT。能为我们的游戏做点什么，让我觉得非常高兴和充实。我也希望在下学期的创新实践里，我可以在代码部分作出多一些贡献。

附录：具体代码及注释

**Flash类：**

// flash.cpp : 实现文件

//

#include "stdafx.h"

#include "mysnake.h"

#include "flash.h"

#include "afxdialogex.h"

// flash 对话框

IMPLEMENT\_DYNAMIC(flash, CDialogEx)

flash::flash(CWnd\* pParent /\*=NULL\*/)

: CDialogEx(flash::IDD, pParent)

{

n=0;

}

flash::~flash()

{

}

void flash::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

DDX\_Control(pDX, IDC\_SHOCKWAVEFLASH1, movie);

CString str;

TCHAR szCurrentDir[MAX\_PATH];

::GetCurrentDirectory(MAX\_PATH,szCurrentDir);

str.Format(\_T("%s"),szCurrentDir);

switch(n){

case 1:str = str+\_T("\\开始动画.swf");

break;

case 2:str = str+\_T("\\Game Over.swf");

break;

}

movie.LoadMovie(0, str);

movie.Play();

}

BEGIN\_MESSAGE\_MAP(flash, CDialogEx)

END\_MESSAGE\_MAP()

// flash 消息处理程序

**MainFrm类：**

// MainFrm.cpp : implementation of the CMainFrame class

//

#include "stdafx.h"

#include "mysnake.h"

#include "MainFrm.h"

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

/////////////////////////////////////////////////////////////////////////////

// CMainFrame

IMPLEMENT\_DYNCREATE(CMainFrame, CFrameWnd)

BEGIN\_MESSAGE\_MAP(CMainFrame, CFrameWnd)

//{{AFX\_MSG\_MAP(CMainFrame)

// NOTE - the ClassWizard will add and remove mapping macros here.

// DO NOT EDIT what you see in these blocks of generated code !

//}}AFX\_MSG\_MAP

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CMainFrame construction/destruction

CMainFrame::CMainFrame()

{

// TODO: add member initialization code here

}

CMainFrame::~CMainFrame()

{

}

BOOL CMainFrame::PreCreateWindow(CREATESTRUCT& cs)

{

if( !CFrameWnd::PreCreateWindow(cs) )

return FALSE;

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

cs.y=0;cs.x=0;

cs.cx=800;cs.cy=860;

return TRUE;

}

/////////////////////////////////////////////////////////////////////////////

// CMainFrame diagnostics

#ifdef \_DEBUG

void CMainFrame::AssertValid() const

{

CFrameWnd::AssertValid();

}

void CMainFrame::Dump(CDumpContext& dc) const

{

CFrameWnd::Dump(dc);

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CMainFrame message handlers

**Manage类：**

#include "stdafx.h"

#include "manage.h"

#include"net.h"

manage::manage(void)

{

user=robot=0;

Cell::size=20;

memset(Map,0,sizeof(Map));

}

manage::~manage(void)

{

}

int manage::login(string name){

int n=2;while(n<31){

if((user&(1<<n))==0){//空缺的用户

control[n]=new Snake;

control[n]->init(1+n\*2,5,n);

control[n]->name=name;user=user|(1<<n);

return n;

}n++;

}

return 0;//失败

}

bool manage::setdir(int direct,int id){

if(direct==5)

robot=robot^(1<<id);//机器人控制转化

//else

//if(robot&(1<<id))//机器

//{

// return 0;

//}

else{//用户

return control[id]->SETDIR(direct);

}

return 1;

}

bool manage::go(){

int t=1<<2;int id=2;char u[8]={0};flash a;

while(id<31){

if(t&user)

if(only&&robot&(1<<id))//机器

{

man.control=control[id];

int dir = man.AI\_c();

//char tt[4] = { 0 };

//tt[0] = '0' + id; tt[1] = dir + '0';

//dlg.sendmessage(4, tt);

setdir(dir,id);

man.control->SETDIR(dir);

u[0]=id;

int w = man.control->run();

switch (w)

{

//case 1:eat();break;

case 0:break;

case 1:break;//食物

case WALL: leave(id);

u[0]=id;

a.n=2;

a.DoModal();

dlg.Create(IDD\_NET);

dlg.ShowWindow(SW\_SHOW);

return false; break;//撞墙死

default:

a.DoModal();

dlg.Create(IDD\_NET);

dlg.ShowWindow(SW\_SHOW);

leave(id);u[0]=id;

//dlg.sendmessage('5',u);

return false;

}

}

else//网络部分

{

u[0]=id+'0';

::Sleep(10);

if(dlg.server&&!only)

dlg.sendmessage('7',u);

::Sleep(20);

int w=control[id]->run();

switch (w)

{

//case 1:eat();break;

case 0:break;

case 1:break;//食物

case WALL:

leave(id);

u[0]=id+'0';

if(!only)

dlg.sendmessage('5',u);

return false; //撞墙死

default:

leave(id);

u[0]=id+'0';

if(!only)

dlg.sendmessage('5',u);

if (w&(1 << id)){ ; }//自杀

else if (t<0){ ; }//和别人一起死

else {

int kill = getn(w);

}//被kill杀

return false;

}

}

id++;t=t<<1;

}

return 1;

}

void manage::draw(CDC\* pDC){

int id=1<<2;int t=2;

while(t<31){

if(id&user) control[t]->draw(pDC);

t++;id=id<<1;

}

}

void manage::draw(){

food.setstyle(0);

food.draw();

int id=1<<2;int t=2;

while(t<31){

if(id&user)

control[t]->draw();

t++;id=id<<1;

}

}

void manage::leave(int id){

user=user&(~(1<<id));

robot=robot^(1<<id);

delete control[id];

}

void eat(){

//grow()

int x,y;

if(dlg.server||only)

while(1){

x=rand()%30;

y=rand()%30;//HERE

char t[3]={0};t[0]=x;t[1]=y;

if(Map[x][y]==0){

if(dlg.server)

dlg.sendmessage('8',t);

food.x=x;

food.y=y;

setfood(food.point());

break;

}

}

}

**MySnakeView类**

// mysnakeView.cpp : implementation of the CMysnakeView class

//

#include "stdafx.h"

#include "mysnake.h"

#include<afxsock.h>

#include "mysnakeDoc.h"

#include "mysnakeView.h"

#include<afxwin.h>

#include<MMSystem.h>

#pragma comment(lib, "WINMM.LIB")

#ifdef \_DEBUG

#define new DEBUG\_NEW

#undef THIS\_FILE

static char THIS\_FILE[] = \_\_FILE\_\_;

#endif

#include"Robt.h"

#include"manage.h"

#include"net.h"

#include"flash.h"

int CMysnakeView::mode;

flash a;

int pause=0;

int cont=0;

int onstart=0;

int two=0;

int addai=0;

clock\_t start;

int id,id2;

manage m;

net dlg;

int rob=1;

/////////////////////////////////////////////////////////////////////////////

// CMysnakeView

//int user[2];

IMPLEMENT\_DYNCREATE(CMysnakeView, CView)

BEGIN\_MESSAGE\_MAP(CMysnakeView, CView)

//{{AFX\_MSG\_MAP(CMysnakeView)

ON\_WM\_KEYDOWN()

//}}AFX\_MSG\_MAP

ON\_WM\_TIMER()

// ON\_WM\_PAINT()

ON\_WM\_ERASEBKGND()

ON\_WM\_CHAR()

// ON\_COMMAND(IDC\_BUTTON4, &CMysnakeView::OnButton4)

// ON\_COMMAND(IDC\_BUTTON5, &CMysnakeView::OnButton5)

// ON\_COMMAND(IDC\_BUTTON6, &CMysnakeView::OnButton6)

// ON\_COMMAND(IDC\_BUTTON7, &CMysnakeView::OnButton7)

ON\_BN\_CLICKED(IDC\_BUTTON5, &CMysnakeView::OnBnClickedButton5)

END\_MESSAGE\_MAP()

/////////////////////////////////////////////////////////////////////////////

// CMysnakeView construction/destruction

CPoint temfood;

CMysnakeView::CMysnakeView()

{

// TODO: add construction code here

//user[1]=m.login("me");

//m.setdir(5,user[1]);

POINT s;s.x=15;s.y=10;

setfood(s);temfood=s;

//backgroud.Draw(10\*size,10\*size,size,size,snake,80,120,size\*2,2\*size,80);

//backgroud.Create(1000,800);

//backgroud.Load("snake.bmp");

}

CMysnakeView::~CMysnakeView()

{

snake.~CBitmapEx();

}

BOOL CMysnakeView::PreCreateWindow(CREATESTRUCT& cs)

{

// TODO: Modify the Window class or styles here by modifying

// the CREATESTRUCT cs

return CView::PreCreateWindow(cs);

}

/////////////////////////////////////////////////////////////////////////////

// CMysnakeView drawing

void CMysnakeView::OnDraw(CDC\* pDC)

{

backgroud.Load(L"back.bmp");

CMysnakeDoc\* pDoc = GetDocument();

ASSERT\_VALID(pDoc);

m.draw();

backgroud.Draw(pDC->m\_hDC);

// TODO: add draw code for native data here

}

/////////////////////////////////////////////////////////////////////////////

// CMysnakeView diagnostics

#ifdef \_DEBUG

void CMysnakeView::AssertValid() const

{

CView::AssertValid();

}

void CMysnakeView::Dump(CDumpContext& dc) const

{

CView::Dump(dc);

}

CMysnakeDoc\* CMysnakeView::GetDocument() // non-debug version is inline

{

ASSERT(m\_pDocument->IsKindOf(RUNTIME\_CLASS(CMysnakeDoc)));

return (CMysnakeDoc\*)m\_pDocument;

}

#endif //\_DEBUG

/////////////////////////////////////////////////////////////////////////////

// CMysnakeView message handlers

void CMysnakeView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)

{

// TODO: Add your message handler code here and/or call default

/\*CBitmap bmpBrick;

bmpBrick.LoadBitmap(IDB\_SNAKE);POINT p;

p.x=p.y=10;

Drawmap(&bmpBrick,p);

\*/

if(only)

switch(nChar)

{

case VK\_UP:

if(mode==3)m.setdir(UP, dlg.n);

if(mode==2||mode==1)m.setdir(UP,id);

//dlg.sendmessage(4, "4");

break;

case VK\_DOWN:

if(mode==3)m.setdir(DOWN,dlg.n);

if(mode==2||mode==1)m.setdir(DOWN,id);

//dlg.sendmessage(4, "3");

break;

case VK\_LEFT:

if(mode==3)m.setdir(LEFT, dlg.n);

if(mode==2||mode==1)m.setdir(LEFT,id);

//dlg.sendmessage(4, "1");

break;

case VK\_RIGHT:

if(mode==3)m.setdir(RIGHT,dlg.n);

if(mode==2||mode==1)m.setdir(RIGHT,id);

//dlg.sendmessage(4, "2");

break;

//default :test.grow(1);

}

else

{

if(nChar == 87) rob=!rob;

if(!rob)

switch(nChar)

{case 'e':

break;

case VK\_UP:

dlg.require(UP, dlg.n);

//dlg.sendmessage(4, "4");

break;

case VK\_DOWN:

dlg.require(DOWN,dlg.n);

//dlg.sendmessage(4, "3");

break;

case VK\_LEFT:

dlg.require(LEFT, dlg.n);

//dlg.sendmessage(4, "1");

break;

case VK\_RIGHT:

dlg.require(RIGHT,dlg.n);

//dlg.sendmessage(4, "2");

break;

//default :m.setdir(5, dlg.n);

}

}

this->RedrawWindow();

CView::OnKeyDown(nChar, nRepCnt, nFlags);

}

void CMysnakeView::Drawmap(CBitmap\* icon,POINT p)

{

CClientDC clintDC(this);

CDC MenDC;

MenDC.CreateCompatibleDC(&clintDC);

CBitmap \*pOldDC = MenDC.SelectObject(icon);

clintDC.BitBlt((p.x+8)\*20,(p.y+1)\*20,20,20,&MenDC,0,0,SRCCOPY);

MenDC.SelectObject(pOldDC);

}

void CMysnakeView::OnTimer(UINT\_PTR nIDEvent)

{

if(only){

if(onstart==1)

Onstart();

if(two==1)

OnTwo();

if(addai==1)

AddAI();

if(pause==1)

Onpause();

if(cont==1)

OnContinue();

CPaintDC dc(this); // device context for painting

POINT p;

p.x=food.x; p.y=food.y;

int t=m.go();

this->RedrawWindow();

if(temfood!=food.point()){

//P->Rectangle(food.x\*size,food.y\*size,(food.x+1)\*size,(food.y+1)\*size);

temfood=food.point();

}

if(mode==1)

{

speed();

scores();

if(!t)

{

KillTimer(1);

mciSendString(TEXT("close song"),0,0,0);

a.n=2;

a.DoModal();

}

}

}

else{

if(rob&&dlg.n!=0)

{ m.man.control=m.control[dlg.n];

int w=m.man.AI\_c();

//m.setdir(w,dlg.n);

dlg.require(w,dlg.n);

}

if(dlg.server){

m.go();

///dlg.sendmessage('7',"a");

}

}

this->RedrawWindow();

CView::OnTimer(nIDEvent);

}

void CMysnakeView::OnInitialUpdate()

{

CView::OnInitialUpdate();

a.n=1;

a.DoModal();

dlg.Create(IDD\_NET);

dlg.ShowWindow(SW\_SHOW);

this->SetTimer(1,300,NULL);

// TODO: 在此添加专用代码和/或调用基类

}

BOOL CMysnakeView::OnEraseBkgnd(CDC\* pDC)

{

// TODO: 在此添加消息处理程序代码和/或调用默认值

return true;//CView::OnEraseBkgnd(pDC);

}

UINT newdlg(LPVOID P){

return 1;

}

void CMysnakeView::OnChar(UINT nChar, UINT nRepCnt, UINT nFlags)

{

// TODO: 在此添加消息处理程序代码和/或调用默认值

switch(nChar)

{

case 'w':

if(mode==2||mode==1)m.setdir(UP,id2);

//dlg.sendmessage(4, "4");

break;

case 's':

if(mode==2||mode==1)m.setdir(DOWN,id2);

//dlg.sendmessage(4, "3");

break;

case 'a':

if(mode==2||mode==1)m.setdir(LEFT,id2);

//dlg.sendmessage(4, "1");

break;

case 'd':

if(mode==2||mode==1)m.setdir(RIGHT,id2);

//dlg.sendmessage(4, "2");

break;

//default :test.grow(1);

}

this->RedrawWindow();

CView::OnChar(nChar, nRepCnt, nFlags);

}

void CMysnakeView::speed(void)

{

CString history;

switch((clock()-start)/1000)

{

case 13:

KillTimer(1);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"一级加速ing！！！！");

::SetTimer(this->m\_hWnd,1,125,NULL);

break;

case 33:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"二级加速ing！！！！");

::SetTimer(this->m\_hWnd,1,100,NULL);

break;

case 44:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"三级加速ing！！！！");

::SetTimer(this->m\_hWnd,1,75,NULL);

break;

case 52:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"四级加速ing！！！！");

::SetTimer(this->m\_hWnd,1,50,NULL);

break;

case 63:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"缓口气吧");

::SetTimer(this->m\_hWnd,1,75,NULL);

break;

case 81:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"五级加速ing！！！！");

::SetTimer(this->m\_hWnd,1,50,NULL);

break;

case 96:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"准备终极加速");

::SetTimer(this->m\_hWnd,1,75,NULL);

case 115:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"第一次终极加速");

::SetTimer(this->m\_hWnd,1,35,NULL);

break;

case 138:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"准备第二次终极加速");

::SetTimer(this->m\_hWnd,1,50,NULL);

break;

case 148:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"第二次终极加速");

::SetTimer(this->m\_hWnd,1,25,NULL);

break;

case 202:

KillTimer(1);

//dlg.GetDlgItemTextW(IDC\_EDIT6,history);

dlg.SetDlgItemTextW(IDC\_EDIT6,L"恭喜你，可以歇一会了");

::SetTimer(this->m\_hWnd,1,70,NULL);

break;

}

}

void CMysnakeView::OnBnClickedButton5()

{

// TODO: 在此添加控件通知处理程序代码

}

void CMysnakeView::Onstart(void)

{

mode=1;

::SetTimer(this->m\_hWnd,1,150,NULL);

id=m.login("玩家1");

mciSendString(TEXT("open Crotia.mp3 alias song"),NULL,0,NULL);

mciSendString(TEXT("play song"),0,0,0);

start=clock();

onstart=0;

}

void CMysnakeView::OnTwo(void)

{

mode=2;

id=m.login("玩家1");

id2=m.login("玩家2");

::SetTimer(this->m\_hWnd,1,200,NULL);

two=0;

}

void CMysnakeView::AddAI(void)

{

int t = m.login("rrr");

m.setdir(5, t);

addai=0;

}

int CMysnakeView::scores(void)

{

char a[1000]={0};

int len=m.control[id]->body.size()+1;

int scores;

if(len<20)

scores=len\*10;

else if(len>=20&&len<30)

scores=len\*100;

else if(len>=50)

scores=len\*1000;

else scores=len\*10000;

itoa(scores,a,10);

CString as;

as=a;

dlg.SetDlgItemInt(IDC\_EDIT9,scores);

return 0;

}

void CMysnakeView::Onpause(void)

{

KillTimer(1);

pause=0;

}

void CMysnakeView::OnContinue(void)

{

SetTimer(1,200,NULL);

cont=0;

}

**Net类：**

// net.cpp : 实现文件

//

#include "stdafx.h"

#include "mysnake.h"

#include "net.h"

#include "afxdialogex.h"

#include<afxsock.h>

#include<stdlib.h>

#include"mysnakeView.h"

#include<string>

#include"manage.h"

#include"snake.h"

#include"NetSnake.h"//直接控制的蛇，由客户机创建，被主机管理

int finish = 0;

char name[20][100];

int user[10];

int only=1;

CString severname;

// net 对话框

/\*

字符第一个表示命令：

0：忽略，无内容

1：文字消息

2：加入玩家，后面是玩家名字

3：返回的ID，和名字

4：玩家的操作，包括ID,和按键

5：退出

\*/

IMPLEMENT\_DYNAMIC(net, CDialogEx)

net::net(CWnd\* pParent /\*=NULL\*/)

: CDialogEx(net::IDD, pParent)

{

server=0;

}

net::~net()

{

}

void net::DoDataExchange(CDataExchange\* pDX)

{

CDialogEx::DoDataExchange(pDX);

this->SetDlgItemTextW(IDC\_EDIT3,L"当前分数：");

this->SetDlgItemTextW(IDC\_EDIT4,L"IP地址：");

this->SetDlgItemTextW(IDC\_EDIT5,L"消息列表：");

this->SetDlgItemTextW(IDC\_EDIT7,L"加速提示：");

}

BEGIN\_MESSAGE\_MAP(net, CDialogEx)

ON\_BN\_CLICKED(IDC\_BUTTON2, &net::OnBnClickedButton2)

ON\_BN\_CLICKED(IDC\_CONNECT, &net::OnBnClickedConnect)

ON\_BN\_CLICKED(IDC\_BUTTON3, &net::OnBnClickedButton3)

ON\_COMMAND(IDOK, &net::OnIdok)

ON\_COMMAND(IDCANCEL, &net::OnIdcancel)

ON\_BN\_CLICKED(IDC\_BUTTON4, &net::OnBnClickedButton4)

// ON\_BN\_CLICKED(IDC\_BUTTON5, &net::OnBnClickedButton5)

ON\_BN\_CLICKED(IDC\_BUTTON5, &net::OnBnClickedButton5)

ON\_BN\_CLICKED(IDC\_BUTTON6, &net::OnBnClickedButton6)

ON\_BN\_CLICKED(IDC\_BUTTON7, &net::OnBnClickedButton7)

END\_MESSAGE\_MAP()

void convtCStrToChar(char \*nstringa,CString const & strParam)

{

//char \*ncharStr = new char[20];

int i = 0;

for (i = 0; i<wcslen(strParam); i++)

{

nstringa[i] = (char)strParam[i];

}

nstringa[i] = '\0';

}

CString ip;

CString text;

CString acmsg;

SOCKET s;

vector<SOCKET> client;

SOCKET clientAccept;//创建通讯的SOCKET

// net 消息处理

void net::require(int dir,int id){

char tt[4] = { 0 };

tt[0] = dir+'0';

tt[1] = id+'0';

// m.control[id]->SETDIR(direct);

if(server){

m.setdir(dir,id);

}

else{//客户机

sendmessage('4',tt);

}

}

void net::select(char a, CString s){//接受消息

int nn ;CString str;

char dd[20]={0};

convtCStrToChar(dd, s);

switch (a-'0')

{

case 0:

case 1:

this->GetDlgItemTextW(IDC\_EDIT1, str);

this->SetDlgItemTextW(IDC\_EDIT1,str+s);

break;

case 2:

if(!server) {

this->MessageBox(L"case 2;wrong");return;}

nn=m.login(dd); //客户机发出,由主机接受创建，并把结果发送

sendmessage('3',dd);

//

break;

case 3:

if(server) this->MessageBox(L"case 3error");

nn=m.login(dd);

if(dd==name) n=nn;//客户机接受

return;

case 4://方向加id,客户机提出的方向改变请求

//if(!server) this->MessageBox(L"case 4;wrong");

//return;}

m.setdir(((char)s[0] - '0'),((char)s[1] - '0'));

break;

case 5:

if(server) this->MessageBox(L"case 5 error");

m.leave(s[0] - '0');//有人退出

if(n==s[0]-'0')

n=0;

break;

case 6:

if(server) this->MessageBox(L"case 6 error");

m.login("server");//第一个，为主机的

::convtCStrToChar(dd,name);//客户机接受

sendmessage('2',dd);//主机发送开始信息，客户机发送注册信息

return ;

case 7:

if(server) this->MessageBox(L"case 7 error");

m.control[dd[0]-'0']->grow(1);//变长

//m.go();

break;

case 8:

if(server) this->MessageBox(L"case 8error");

::setfood(CPoint(dd[0],dd[1]));

break;//食物

case 9://方向的直接操作权利,传送绝对坐标，id,x,y\0.

if(server) this->MessageBox(L"case 9error");

m.control[(char)s[0] - '0']->d\_go( (char)s[1] - '0',(char)s[2] - '0');

break;

default:

break;

}

}

UINT newclient(LPVOID p){

SOCKET tem = clientAccept;

while (!finish){

//////////////////////////////////////////////////////////////////////////

char recvBuf[256], sendBuf[256];

int BytesRecv;

memset(recvBuf, 0, 256);

memset(sendBuf, 0, 256);

BytesRecv = recv(tem, recvBuf, 256, 0);//获取信息

char command = recvBuf[0];

char use[22]={0};

memcpy(use, recvBuf + 1, 20); use[20] = '\0';

CString w=(recvBuf+1);

dlg.sendmessage('1',recvBuf);

dlg.select(recvBuf[0],w);

if (BytesRecv == SOCKET\_ERROR)

{

//cout << "recv error" << endl;

MessageBox(NULL, L"recv ERROR", L"ERROR", 0);

client.clear();

finish=1;

break;

}

char naaa[20];

/\*convtCStrToChar(naaa, dlg.name+"coming \n");

dlg.select('1', naaa);\*/

//cout << "收到客户端发来的数据:" << recvBuf << endl;

memcpy(sendBuf, recvBuf, BytesRecv);

//strcat(sendBuf, "收到");

int ret;

for (int i = 0; i < client.size(); i++)

{

ret = send(client[i], sendBuf, strlen(sendBuf) + 1, 0);//发送给每一个人

if (ret == SOCKET\_ERROR)

{

::MessageBox(NULL, L"send --ERROR", L"ERROR", 0);

client.clear();

finish=1;

break;

}

}

}

CString str;

dlg.GetDlgItemTextW(IDC\_EDIT1,str);

str+="有人退出，连接中断\n";

dlg.SetDlgItemTextW(IDC\_EDIT1,str);

closesocket(s);

WSACleanup();

return 1;

}

UINT SERVER(LPVOID P){

//加载windows sockets动态库

//MAKEWORD(2,2)表示使用WINSOCK2版本

//.wsaData用来存储系统传回的关于WINSOCK的资料.

//cout << "服务器在127.0.0.1:6666端口监听..." << endl;

finish=0;

struct sockaddr\_in clientaddr;

int len = sizeof(clientaddr);

while (1){

if (finish) break;

clientAccept = accept(s, (struct sockaddr\*)&clientaddr, &len);

if (clientAccept == INVALID\_SOCKET){

//cout << "accept error" << endl;

::MessageBox(NULL, L"ACCEPT ERROR", L"ERROR", 0);

break;

//return;

};//等待就受连接请求

int h = 1;

for (int i = 0; i < client.size(); i++){

if (client[i] == clientAccept)

h = 0;

}

if (h)

client.push\_back(clientAccept);

//cout << "客户端连接成功，现在开始收发数据..." << endl;

::AfxBeginThread(newclient, 0);

//cout << "向客户端发送数据 " << sendBuf << "成功" << endl;

//::MessageBox(NULL, (LPCWSTR)sendBuf, L"OK", 0);

}

//dlg.setitem();

return 1;

}

void net::OnBnClickedButton2()//server

{

only=0;

this->GetDlgItemTextW(IDC\_NAME, name);

server = 1;

//this->GetDlgItem(IDC\_IPADDRESS1);

this->GetDlgItemTextW(IDC\_IPADDRESS1, ip);

this->initserver();

CWinThread \*p = AfxBeginThread(SERVER,0);

// TODO: 在此添加控件通知处理程序代码

}//server

int net::initserver(){

WSADATA wsaData;

if (WSAStartup(MAKEWORD(2, 2), &wsaData) != 0){

//cout << "WSAStartup error" << endl;

::MessageBox(NULL, L"wsastartup ERROR", L"ERROR", 0);

}

s = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);//创建监听的SOCKET s

if (s == INVALID\_SOCKET){

//cout << "socket error" << endl;

WSACleanup();

//return;

}//创建一个流套接字，返回套接字号s

struct sockaddr\_in addrServ;//sockaddr\_in 是在windows/linux下面的结构

addrServ.sin\_family = AF\_INET;//指代协议族，在socket编程中只能是AF\_INET

addrServ.sin\_port = htons(6666);//存储端口号，在linux下，端口号的范围0~65535,同时0~1024范围的端口号已经被系统使用或保留。

char ipp[20];

convtCStrToChar(ipp, ip);

addrServ.sin\_addr.S\_un.S\_addr = inet\_addr(ipp);//sin\_addr存储IP地址

int runcode;

runcode = bind(s, (struct sockaddr\*)&addrServ, sizeof(struct sockaddr\_in));

if (runcode == SOCKET\_ERROR){

//cout << "bind error" << endl;

WSACleanup();

//return;

};//将服务器协议族、地址、端口号与一个定义的socket -> s绑定

runcode = listen(s, 2);

if (runcode == SOCKET\_ERROR){

//cout << "listen error" << endl;

::MessageBox(NULL, L"listen ERROR", L"ERROR", 0);

WSACleanup();

//return;

};//设置套接字为监听接入连接的状态

this->SetDlgItemTextW(IDC\_EDIT1, CString("连接建立\n"));

return 1;

}

UINT CLIENT(LPVOID P){

while (1){//主要部分

char recvBuf[256], sendBuf[256];

int BytesSend;

memset(recvBuf, 0, 256);

memset(sendBuf, 0, 256);

strcpy(sendBuf, "0");

int ret;

ret = recv(s, recvBuf, 256, 0);//接受

CString tt; tt=(recvBuf + 1);

dlg.select(recvBuf[0], tt);//分拣

if (ret == SOCKET\_ERROR)

{

MessageBox(NULL,L"SOCKET\_ERROR",L"as",0);

break;

//return 1;

}

//cout << "收到服务器发来的数据:" << recvBuf << endl;

}

closesocket(s);

WSACleanup();

return 1;

}

void net::OnBnClickedConnect()

{

// TODO: 在此添加控件通知处理程序代码

only=0;

this->GetDlgItemTextW(IDC\_NAME, name);

this->initclient();

sendmessage('1', "hello");

CWinThread \*p = AfxBeginThread(CLIENT, 0);

}

int net::initclient(){

WORD wVersionRequested;

WSADATA wsaData;

int ret;

server = 0;

wVersionRequested = MAKEWORD(2, 2);

ret = WSAStartup(wVersionRequested, &wsaData);

if (ret != 0) {

MessageBox( L"Winsockfail", L"as", 0);

return 1;

}//初始化Winsock

s = socket(AF\_INET, SOCK\_STREAM, IPPROTO\_TCP);

if (s == INVALID\_SOCKET)

{

MessageBox( L"SOCKET fail", L"as", 0);

return 1;

}//创建连接到服务器的SOCKET对象

char ipp[20];

this->GetDlgItemTextW(IDC\_IPADDRESS1,ip);

convtCStrToChar(ipp, ip);

struct sockaddr\_in addrSrv;

addrSrv.sin\_addr.S\_un.S\_addr = inet\_addr(ipp);//服务器地址

addrSrv.sin\_family = AF\_INET;//指代协议族

addrSrv.sin\_port = htons(6666);//服务器端口号

//搭建地址信息

int namelen = sizeof(struct sockaddr);

ret = connect(s, (struct sockaddr\*)&addrSrv, namelen);

if (ret == SOCKET\_ERROR)

{

MessageBox( L"connect fail", L"as", 0);

return 1;

}

char ss[200]={0};

::convtCStrToChar(ss,name+" coming!\n \0");

sendmessage('1',ss);

return 1;

}

BOOL net::OnInitDialog()

{

srand(time(NULL));

CDialogEx::OnInitDialog();

ip = "127.0.0.1";

char s[10]="new";s[3]=('0'+rand()%20);

CString ee=s;

this->SetDlgItemTextW(IDC\_IPADDRESS1, ip);

this->SetDlgItemTextW(IDC\_NAME, ee);

this->GetDlgItemTextW(IDC\_NAME, name);

// TODO: 在此添加额外的初始化

return TRUE; // return TRUE unless you set the focus to a control

// 异常: OCX 属性页应返回 FALSE

}

int net::sendmessage(char a, char \*c){

//if(a=='t') return ;

char msg[250] = { 0 };

strcpy(msg + 1, c);

msg[0] = a;

if(s)

if(!(a==4&&n==0))

if (server){

int BytesSend;

char t[3]={0};t[1]=2+'0';

for (int i = 0; i < client.size(); i++){//主机

if(a=='6'){//除了'6'其他部分主机都是直接给别人

t[0]='6';t[1]++;

BytesSend = send(client[i], t, 4, 0);}

else{

BytesSend = send(client[i], msg, strlen(msg) + 1, 0);//发送部分

}

if (BytesSend == SOCKET\_ERROR)

{

MessageBox(L"send fail-server", L"as", 0);

client.erase(client.begin(),client.begin()+1);

return 1;

}

}

}

else{

int BytesSend;

BytesSend = send(s, msg, strlen(msg) + 1, 0);//发送部分

if (BytesSend == SOCKET\_ERROR)//客户机

{

//MessageBox(L"send fail-client", L"as", 0);

closesocket(s);s=0;

WSACleanup();

return 1;

}

}

return 1;

}

void net::OnBnClickedButton3()

{

only=0;

CString text;

this->GetDlgItemTextW(IDC\_EDIT2, text);

text ;

char ff[20];

convtCStrToChar(ff, text);

sendmessage('1', ff);

// TODO: 在此添加控件通知处理程序代码

}

void net::OnIdok()

{

// TODO: 在此添加命令处理程序代码

}

void net::OnIdcancel()

{

// TODO: 在此添加命令处理程序代码

this->DestroyWindow();

}

void net::OnBnClickedButton4()//开始按钮

{

only=0;

//m.login()

char ss[200]={0};

::convtCStrToChar(ss,name);

n=m.login(ss);

::Sleep(20);

sendmessage('6', "go");

// TODO: 在此添加控件通知处理程序代码

}

/\*

字符第一个表示命令：

0：忽略，无内容

1：文字消息

2：加入玩家，后面是玩家名字

3：返回的ID，和名字

4：玩家的操作，包括按键ID,

5：退出

\*/

void net::OnBnClickedButton5()

{

// TODO: 在此添加控件通知处理程序代码

only=1;

onstart=1;

}

void net::OnBnClickedButton6()

{

// TODO: 在此添加控件通知处理程序代码

only=1;

two=1;

}

void net::OnBnClickedButton7()

{

// TODO: 在此添加控件通知处理程序代码

addai=1;

}

**Robt类：**

// Robt.cpp: implementation of the CRobt class.

//

//////////////////////////////////////////////////////////////////////

#include "stdafx.h"

#include "mysnake.h"

#include "Robt.h"

#include<math.h>

#include<algorithm>

#ifdef \_DEBUG

#undef THIS\_FILE

static char THIS\_FILE[]=\_\_FILE\_\_;

#define new DEBUG\_NEW

#endif

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

bool rout[30][30];

CPoint dir[5];

int cal(POINT cur,POINT aim,int l,int f);

int cal\_c(POINT cur,POINT aim,int l,int f);

CRobt::CRobt()

{

level=2;control=0;

}

CRobt::~CRobt()

{

}

int dis(CPoint a,CPoint b){

return abs(a.x-b.x)+abs(a.y-b.y);

}

bool comp(POINT& a,POINT& b){

return a.y>b.y;

}

int CRobt::AI\_c(){

dir[LEFT]=CPoint(-1,0);

dir[RIGHT]=CPoint(1,0);

dir[DOWN]=CPoint(0,1);

dir[UP]=CPoint(0,-1);

CPoint cur=control->head.point();

POINT sum[4];//方向与安全值绑定

float k=1;

for(int i=0;i<4;i++)

{sum[i].x=i+1;

if(abs(dis(cur,food.point()))>abs(dis(cur+dir[i+1],food.point())))//下一步更接近食物

k=10;

else

k=0.5;

sum[i].y=cal\_c(cur+dir[i+1],food.point(),level,1)\*k;//食物的影响力

k;}

std::sort(sum,sum+4,comp);//对安全值排序

int p=0;

for(int i=0;i<4;i++)

{sum[i].y\*=(4-i);//扩大差距

p+=sum[i].y;

}

int tem;

if(p==0) return 1;//无路可走

tem=rand()%p;//在0到总值之间取随机数，权值大的更有机会

int w=0;

while(1){

tem-=sum[w].y;

if(tem<0)

return sum[w].x;

w++;

}

return 1;

}

int CRobt::AI(){

dir[LEFT]=CPoint(-1,0);

dir[RIGHT]=CPoint(1,0);

dir[DOWN]=CPoint(0,1);

dir[UP]=CPoint(0,-1);

CPoint cur=control->head.point();

rout[cur.x][cur.y]=1;//初始化，起点设为1

int direct=-1;//默认方向

int value[5];

value[LEFT]=cal(cur+dir[LEFT],food.point(),level,1);

value[2]=cal(cur+dir[2],food.point(),level,1);

value[3]=cal(cur+dir[3],food.point(),level,1);

value[4]=cal(cur+dir[4],food.point(),level,1);

int min=999;

int way[4]={0};int len=0;

for(int i=1;i<5;i++){

if(value[i]>=0&&value[i]<min){min=value[i];direct=i;}

}

for(int q=1;q<=4;q++)

if(min==value[q]) way[len++]=q;

if(direct==-1)return DOWN;//找不到就为下

return way[rand()%len];

}

int AI\_di(){//蛇不会主动吃，但很难死

//dir[LEFT]=CPoint(-1,0);

//dir[RIGHT]=CPoint(1,0);

//dir[DOWN]=CPoint(0,1);

//dir[UP]=CPoint(0,-1);

//memset(rout,0,sizeof(rout));

//CPoint cur=control->head.point();

//rout[cur.x][cur.y]=1;//初始化，起点设为1

//

//int direct=-1;//默认方向

//int value[5];

//value[LEFT]=cal(cur+dir[LEFT],food.point(),level);

//value[2]=cal(cur+dir[2],food.point(),level);

//value[3]=cal(cur+dir[3],food.point(),level);

//value[4]=cal(cur+dir[4],food.point(),level);

//int min=999;

//for(int i=1;i<5;i++){

// if(value[i]>0&&value[i]<min){min=value[i];direct=i;}

//}

//if(direct==-1)direct=DOWN;//找不到就为下

return 1;

}

int AI\_D(){

int d[4]={1,2,3,4};

//int len=4;

//float distance=999.0f;

//

//if(food.x>cur.x){

//if(ifsafe(cur+dir[RIGHT])) return RIGHT;}

//else if(food.x<cur.x) {

// if(ifsafe(cur+dir[LEFT])) return LEFT;}

//else//在同一x上

//{

//if(food.y>cur.y){

//if(ifsafe(cur+dir[DOWN])) return DOWN;}

//else

// if(ifsafe(cur+dir[UP])) return UP;}

//if(direct==-1)

//{

// if(food.y>cur.y){

//if(ifsafe(cur+dir[DOWN])) return DOWN;}

//else if(food.y<cur.y) {

// if(ifsafe(cur+dir[UP])) return UP;}

//else//在同一y上

//{

//if(food.x>cur.x){

//if(ifsafe(cur+dir[RIGHT])) return RIGHT;}

//else

//if(ifsafe(cur+dir[LEFT])) return LEFT;}

//

//}//测试每个方向，遍历每个方向的可行性

//if(direct==-1) direct=1;

return 1;

}

int CRobt::go(){

if(control==0) return false;

control->SETDIR(AI\_c());

int t=control->run();

if(ifsafe(control->head.point())) {

delete control;

control=0;

}

return t;

}

void CRobt::draw(CDC\*p){

if(control) control->draw(p);

else

level;

backgroud.Draw(p->m\_hDC);

}

int Min(int a,int b){ if(a<b)return a;return b;}

int cal(POINT cur,POINT aim,int l,int f){//计算最段路径，l的次数决定计算的深度

if(f) memset(rout,0,sizeof(rout));

if(!ifsafe(cur))//死路

return 1000;

rout[cur.x][cur.y]=1;

if(l==0)

return dis(cur,aim);

POINT p;p.x=cur.x-1;p.y=cur.y;

POINT p1;p1.x=cur.x+1;p1.y=cur.y;

POINT p2;p2.x=cur.x;p2.y=cur.y+1;

POINT p3;p3.x=cur.x;p3.y=cur.y-1;

int t2=1000;

int t1=cal(p,aim,l-1,0);if(t1>0) {t2=t1;}

t1=Min(cal(p1,aim,l-1,0),t2);if(t1>=0) {t2=t1;}

t1=Min(cal(p2,aim,l-1,0),t2);if(t1>=0) {t2=t1;}

t1=Min(cal(p3,aim,l-1,0),t2);if(t1>=0){t2=t1;}

if(t2==1000) return 1000;

return t2+1;

/\*POINT p;

p.x=cur.x-1;p.y=cur.y;

if(rout[p.x][p.y]==0&&!crash(p))//该路为通

return 1+cal(p,aim,l-1);

else return 99999;//死路

p.x=cur.x+1;p.y=cur.y;

if(rout[p.x][p.y]==0&&!crash(p))//该路为通

return 1+cal(p,aim,l-1);

else return 99999;//死路

p.x=cur.x;p.y=cur.y+1;

if(rout[p.x][p.y]==0&&!crash(p))//该路为通

return 1+cal(p,aim,l-1);

else return 99999;//死路

p.x=cur.x;p.y=cur.y-1;

if(rout[p.x][p.y]==0&&!crash(p))//该路为通

return 1+cal(p,aim,l-1);

else return 99999;//死路

\*/

}

int cal\_c(POINT cur,POINT aim,int l,int f){//计算安全值（该方向下不返回的可能路径）

if(f)

memset(rout,0,sizeof(rout));

if(!ifsafe(cur)||(rout[cur.x][cur.y]))//死路

return 0;

rout[cur.x][cur.y]=1;//记录该点，不再重复

if(l==0) return 1;

int r=1;//cur该点可走

for(int i=1;i<5;i++)

r+=cal\_c((CPoint)cur+dir[i],aim,l-1,0);//遍历所有可能路径

return r;

}

**Snake类：**

// snake.cpp: implementation of the snake class.

//

//////////////////////////////////////////////////////////////////////

#include "stdafx.h"

#include "snake.h"

#ifdef \_DEBUG

#undef THIS\_FILE

static char THIS\_FILE[]=\_\_FILE\_\_;

#define new DEBUG\_NEW

#endif

//////////////////////////////////////////////////////////////////////

// Construction/Destruction

//////////////////////////////////////////////////////////////////////

Cell food;

void setfood(POINT p){

//Map[food.x][food.y]=0;

Map[p.x][p.y]=1;

food.x=p.x;food.y=p.y;

}

//蛇的运动

int Snake::run()

{

int r=0;

POINT TEM=head.point();//获取蛇头坐标

int direct =cur\_dir;

old\_dir=cur\_dir;

//如果没吃到食物

if(growth==0)

{

wipepoint(body[end].point());//擦掉蛇尾在地图上的标记

body[end].moveto(head.point());//蛇尾顶到蛇头上

end--;//新蛇尾

if(end==-1) end=body.size()-1;

}

//吃到食物后

else

{

growth--;

int tt=(end+1)%body.size();

body.insert(body.begin()+tt,head);//把新蛇尾插入向量

body[tt].setsign(getsign(n,0));

end=tt-1;

if(end==-1) end=body.size()-1;

}

//尾部处理：擦除地图点，重置end为最后的点

//方向与坐标变化

if(direct==UP)

TEM.y--;

if(direct==DOWN)

TEM.y++;

if(direct==LEFT)

TEM.x--;

if(direct==RIGHT)

TEM.x++;

//判断这样运动后是否安全

int y=Map[TEM.x][TEM.y];

if(!ifsafe(TEM))

{ if(y==0) return WALL;

return y;}

else if(y==1)

{eat();

grow(1);

r=1;

}

//改变蛇头属性

head.x=TEM.x;head.y=TEM.y;

head.setpoint();

head.setstyle(direct);

//改变蛇身属性

for(int i=0;i<body.size();i++){

body[i].setsign(getsign(n,0));

body[i].setpoint();

body[i].setstyle(5);

}

return r;

}

//设置方向

bool Snake::SETDIR(int direct){

switch(old\_dir){

case UP:if(direct==DOWN) return false;break;

case DOWN:if(direct==UP) return false;break;

case LEFT:if(direct==RIGHT) return false;break;

case RIGHT:if(direct==LEFT) return false;break;

}//防止回走

cur\_dir=direct;

return true;

}

/\*bool Snake::dead(POINT& p){

if(p.x<0||p.x>=30)

return true;

if(p.y<0||p.y>=30)

return true;

if(Map[p.x][p.y]&2)

return true;

return false;

}\*/

// 物蛇号

// 0 0 0 无

// 1 1 1 有

// 1 2 4 8 数字

//初始化

Snake::Snake(int x,int y,int s){

init(x,y,s);

}

void Snake::init(int x,int y,int s)

{

scores=0;

n=s;

head.init(x,y,getsign(n,1),DOWN);

growth=0;end=1;

cur\_dir=DOWN;old\_dir=DOWN;

Cell t;

t.init(x,y-1,getsign(n,0),5);

body.push\_back(t);

t.init(x,y-2,getsign(n,0),5);

body.push\_back(t);

head.setpoint();

body[0].setpoint();

body[1].setpoint();

}

//画蛇

//用bitmap类画

void Snake::draw(){

head.draw();

int as=body.size();

while(as--)

body[as].draw();

}

//旧版本CDC画法

void Snake::draw(CDC\* pDC){

head.draw(pDC);

int as=body.size();

while(as--)

body[as].draw(pDC);

}

Snake::~Snake(){

Map[head.x][head.y]=0;//尸体变成食物

for(int i=0;i<body.size();i++)

Map[body[i].x][body[i].y]=0;

}

//判断是否有障碍

bool ifsafe(POINT p){

if(p.x<0||p.x>=MAX\_W) return false;

if(p.y<0||p.y>=MAX\_H) return false;

if(Map[p.x][p.y]>1||Map[p.x][p.y]<0)

return false;

return true;

}

**Unit类：**

// Unit.cpp : 实现文件

//

#include "stdafx.h"

#include<algorithm>

#include "Unit.h"

using namespace std;

#define START\_X 0

#define START\_Y 0

CBitmapEx backgroud;

// Cell

CBitmapEx snake;

CPoint pos[10];//用来分块导入蛇的图像

const int size=20;//方格大小

int Map[100][100];

float Cell::k=1;

int Cell::size;

//初始化单元格

Cell::Cell():x(11),y(40),sign(0),style(7)

{

if(snake.GetWidth()<10){

//snake.Load("snake.bmp");

//backgroud.Create(1000,800);

//backgroud.Load("snake.bmp");

pos[0].x=120;pos[0].y=120;

pos[1].x=0;pos[1].y=0;

pos[2].x=40;pos[2].y=0;

pos[3].x=120;pos[3].y=0;

pos[4].x=80;pos[4].y=0;

pos[5].x=80;pos[5].y=120;

}

}

Cell::Cell(int a,int b,int col,int s){

init(a,b,col,s);

}

void Cell::init(int a,int b,int col,int s){

x=a;y=b;sign=col;style=s;

}

Cell::~Cell()

{

//this->setstyle(0);

}

// Cell 成员函数

/\*

style :

蛇头: 1 2 3 4

食物：0

\*/

void Cell::draw(CDC\* pDC){

pDC->Rectangle(x\*size,y\*size,x\*size+k\*size,y\*size+k\*size);

}

void Cell::draw(){

snake.Load(L"snake.bmp");//导入蛇的图像

CBitmapEx& tem=snake;

pos[0].x = 120; pos[0].y = 120;

pos[1].x = 0; pos[1].y = 0;

pos[2].x = 40; pos[2].y = 0;

pos[3].x = 120; pos[3].y = 0;

pos[4].x = 80; pos[4].y = 0;

pos[5].x = 80; pos[5].y = 120;

//tem.Pixelize(getn(sign));

backgroud.DrawTransparent(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[style].x,pos[style].y,size\*2,2\*size,100,RGB(255,255,255));

//switch(10){

//case 1://color

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[1].x,pos[1].y,size\*2,2\*size,100);break;

//case 2:

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[2].x,pos[2].y,size\*2,2\*size,100);break;

//case 3:

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[3].x,pos[3].y,size\*2,2\*size,100);break;

//case 4:

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[4].x,pos[4].y,size\*2,2\*size,100);break;

//case 0:

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[0].x,pos[0].y,size\*2,2\*size,80);break;

//case 5://body

// backgroud.Draw(START\_X+x\*size,START\_Y+y\*size,size,size,tem,pos[5].x,pos[5].y,size\*2,2\*size,100);break;

//}

}

//判断能否移动

bool Cell::move(int a,int b){

if(x+a<0||x+a>MAX\_W||y+b<0||y+b>MAX\_H)

return 0;

x+=a;y+=b;

return true;

}

//n 与sign的转化

int getsign(int n,bool head){

if(head)return 1+(1<<n)+(1<<31);

return 1+(1<<n);}

//n为1是食物，3是墙，后面为其他玩家 第31位标示为身体与头

int getn(int sign){

for(int i=2;i<31;i++)

if((sign>>i)&1)

return i;

return 0;

}

void drawrec(CDC\* pDC,POINT& p,int style){

switch(style)

{

default:

pDC->Rectangle(p.x\*size,p.y\*size,(p.x+1)\*size,(p.y+1)\*size);}

}