#include"Message.h"

extern int board[20][20];

extern int player\_pick;

/\*

\*return -1 means error

\* This is server address

\*/

int bind\_server\_socket(char \*s1,char \*s2) //创建主机socket

{

unsigned long inaddr; //用来储存ip地址

struct sockaddr\_in saddr; //地址结构

memset(&saddr,0,sizeof(struct sockaddr\_in)); //开始将这一块内存清空

int sock\_id\_s;

sock\_id\_s=socket(AF\_INET,SOCK\_STREAM,0);

if(sock\_id\_s==-1)

return -1;

saddr.sin\_family=AF\_INET;

inaddr=inet\_addr(s1); //将点分地址转化为无符号长整型

memcpy(&saddr.sin\_addr,&inaddr,sizeof(inaddr));

saddr.sin\_port=htons(atoi(s2));//将端口号转化为网络字节序

if(bind(sock\_id\_s,(struct sockaddr \*)&saddr,sizeof(saddr))!=0)

return -1;

return sock\_id\_s;

}

int client\_socket\_connect(char \*s1,char \*s2) //创建客户端socket

{

unsigned long inaddr; //用来储存ip地址

struct sockaddr\_in saddr; //地址结构

memset(&saddr,0,sizeof(struct sockaddr\_in));//开始将这一块内存清空

int sock\_id\_c;

sock\_id\_c=socket(AF\_INET,SOCK\_STREAM,0);

if(sock\_id\_c==-1)

return -1;

saddr.sin\_family=AF\_INET;

inaddr=inet\_addr(s1); //将点分地址转化为无符号长整型

memcpy(&saddr.sin\_addr,&inaddr,sizeof(inaddr));

saddr.sin\_port=htons(atoi(s2)); //将端口号转化为网络字节序

connect(sock\_id\_c,(struct sockaddr\*)&saddr,sizeof(saddr));

return sock\_id\_c;

}

int decompre(MSGSTRUCT msg)

{

switch(msg.msgType)

{

case 1:

printf("get chess position\n");

if (player\_pick==0)

//player\_pick==0 means that the machine is server

//so the sever machine only recive the message from client.

//the value of client for board array is 2

board[msg.x][msg.y]=2;

else

//player\_pick!=0 means that the machine is client

//so the client machine only recive the message from sever.

//the value of sever for board array is 1

board[msg.x][msg.y]=1;

break;

case 2:

printf("对方向您提出和棋请求,您是否同意？\n");

break;

case 3:

printf("对方同意了你的的和棋请求\n");

break;

case 4:

printf("对方拒绝了您的和棋请求，所以您得继续完成本次博弈\n");

break;

case 5:

printf("%s",msg.byMsg);

break;

default:

return -1; //如果获取的消息不正确返回-1

break;

}

}