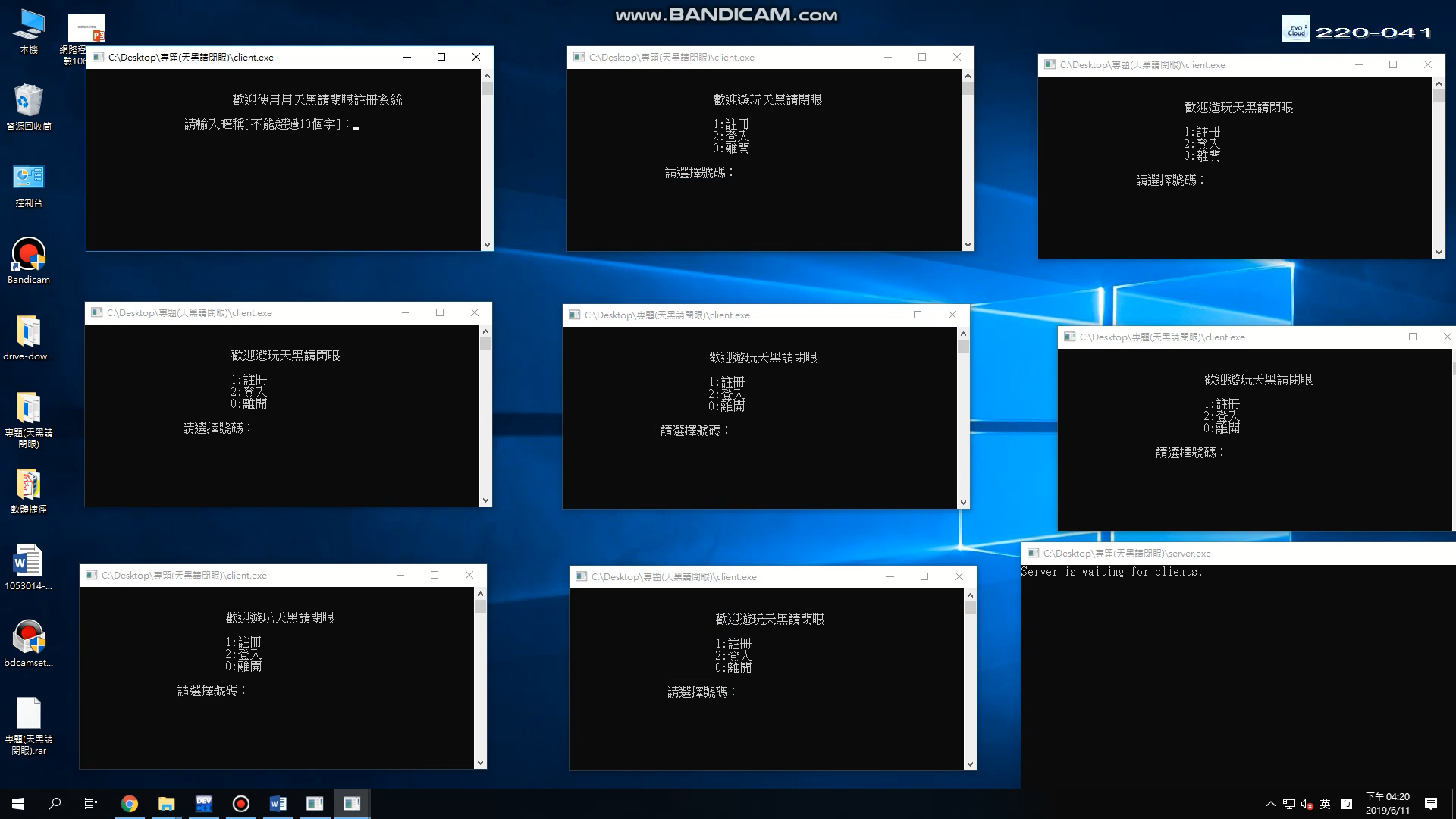
1. 題目:天黑請閉眼

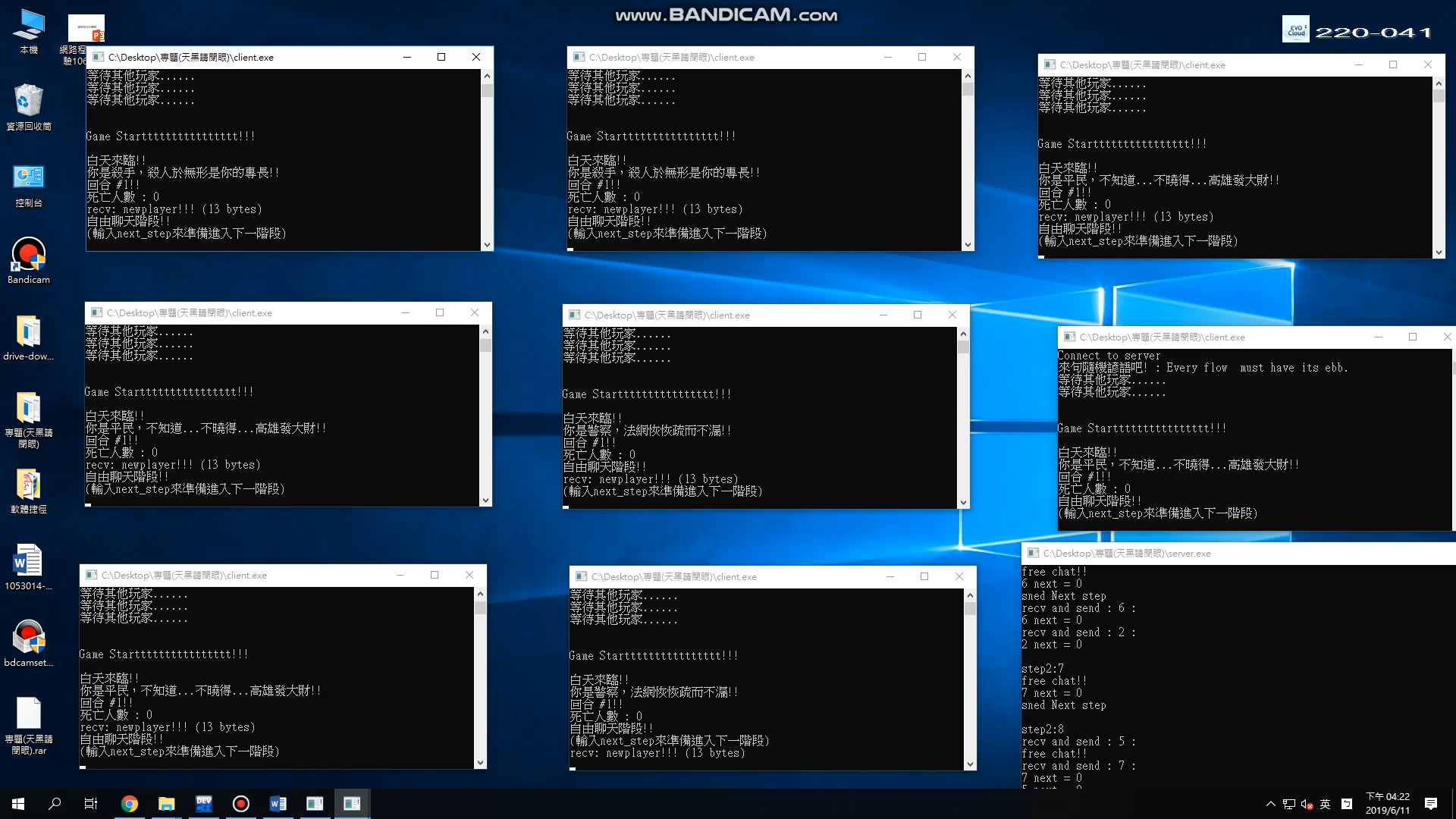
組員:1053014 林煥挺 1063007 葉哲睿

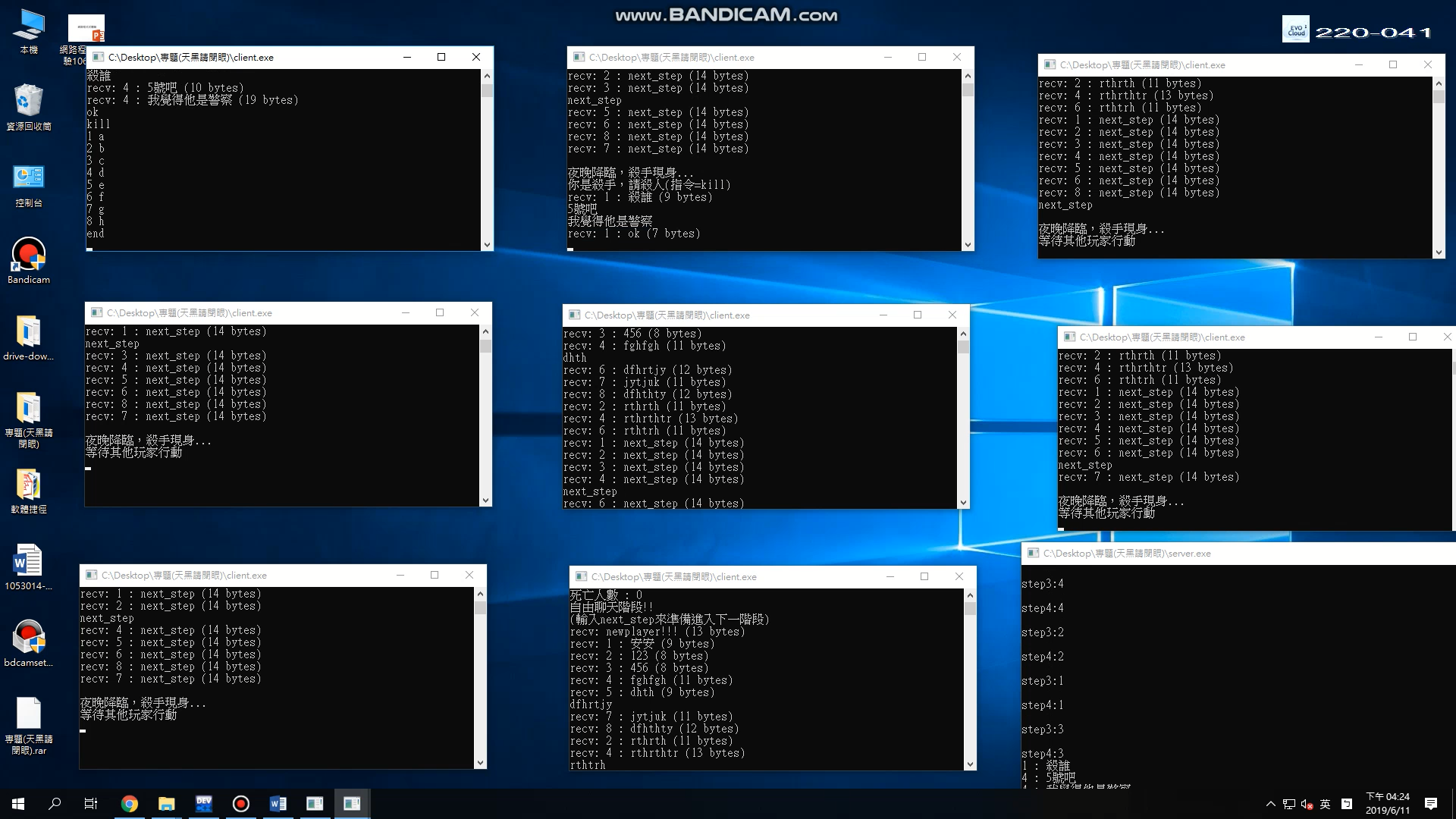
1. 完成作品描述

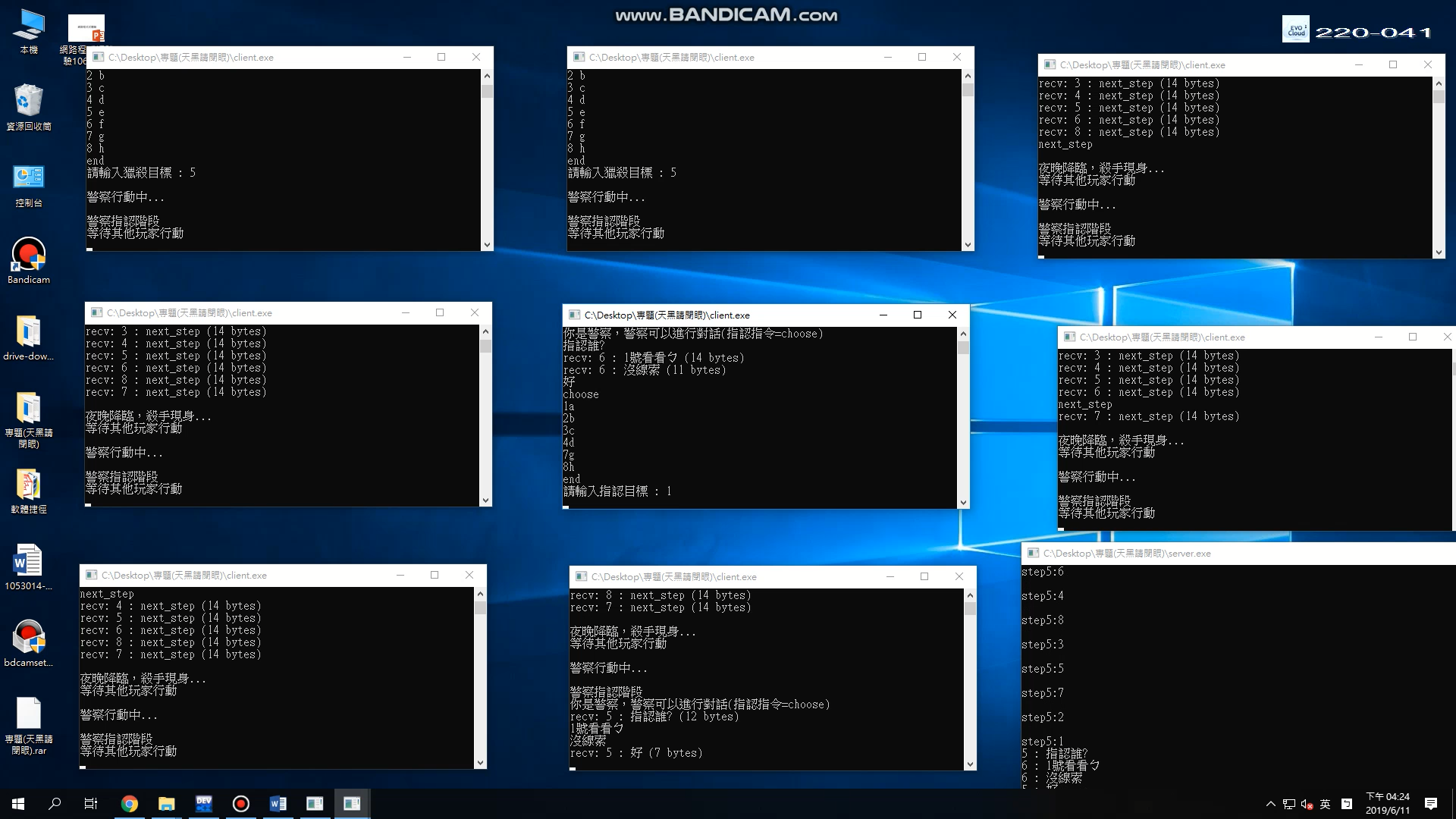
是一款有登入介面的真正八人版天黑請閉眼連線遊戲，在遊戲大廳也有能夠觀看遊戲規則的選項，遊戲中職業分成殺手、警察和平民，以挖掘線索來完成遊戲，若警察死光，則殺手獲勝，若殺手死光，則警察和平民獲勝。

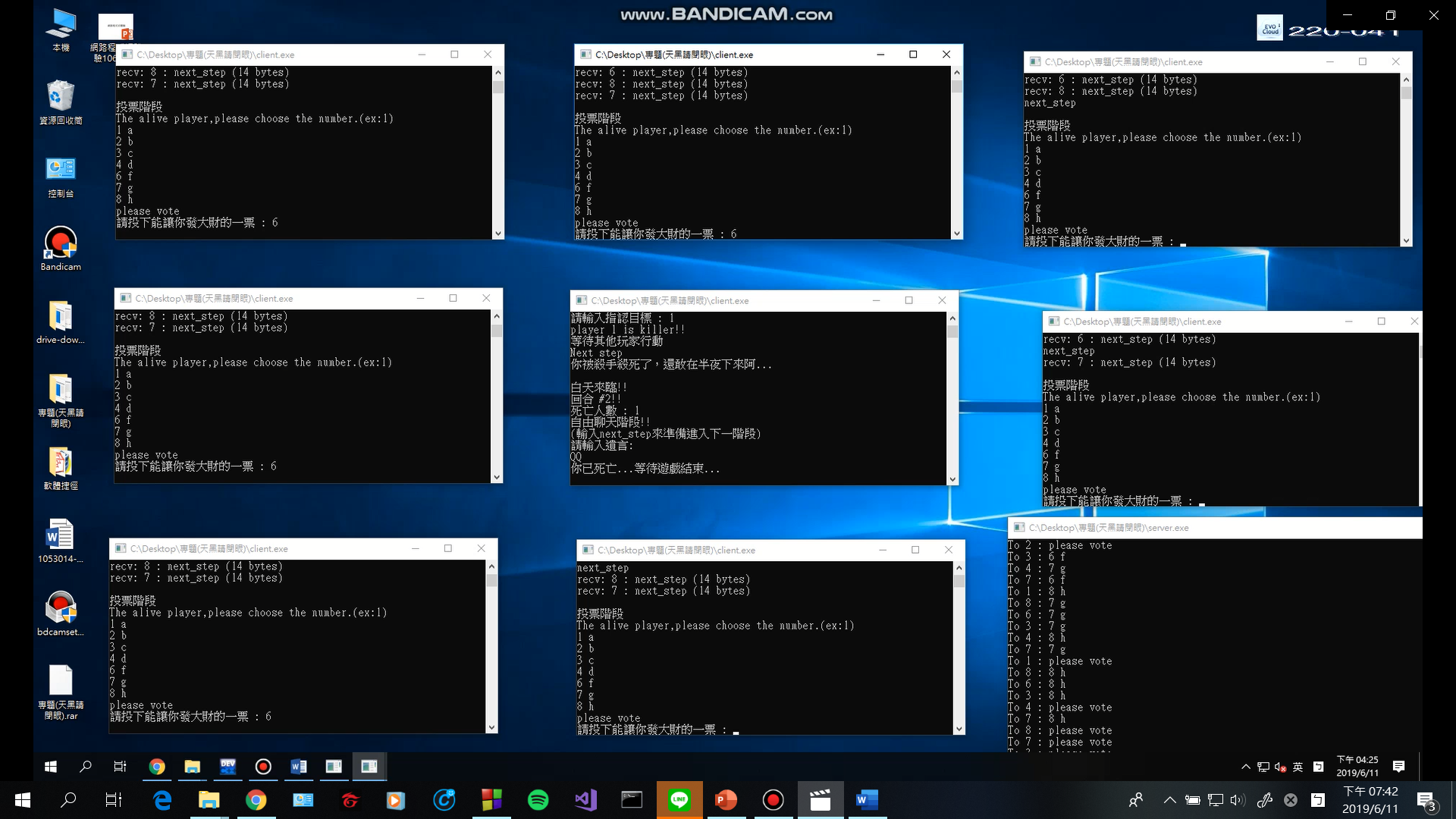
3.執行畫面

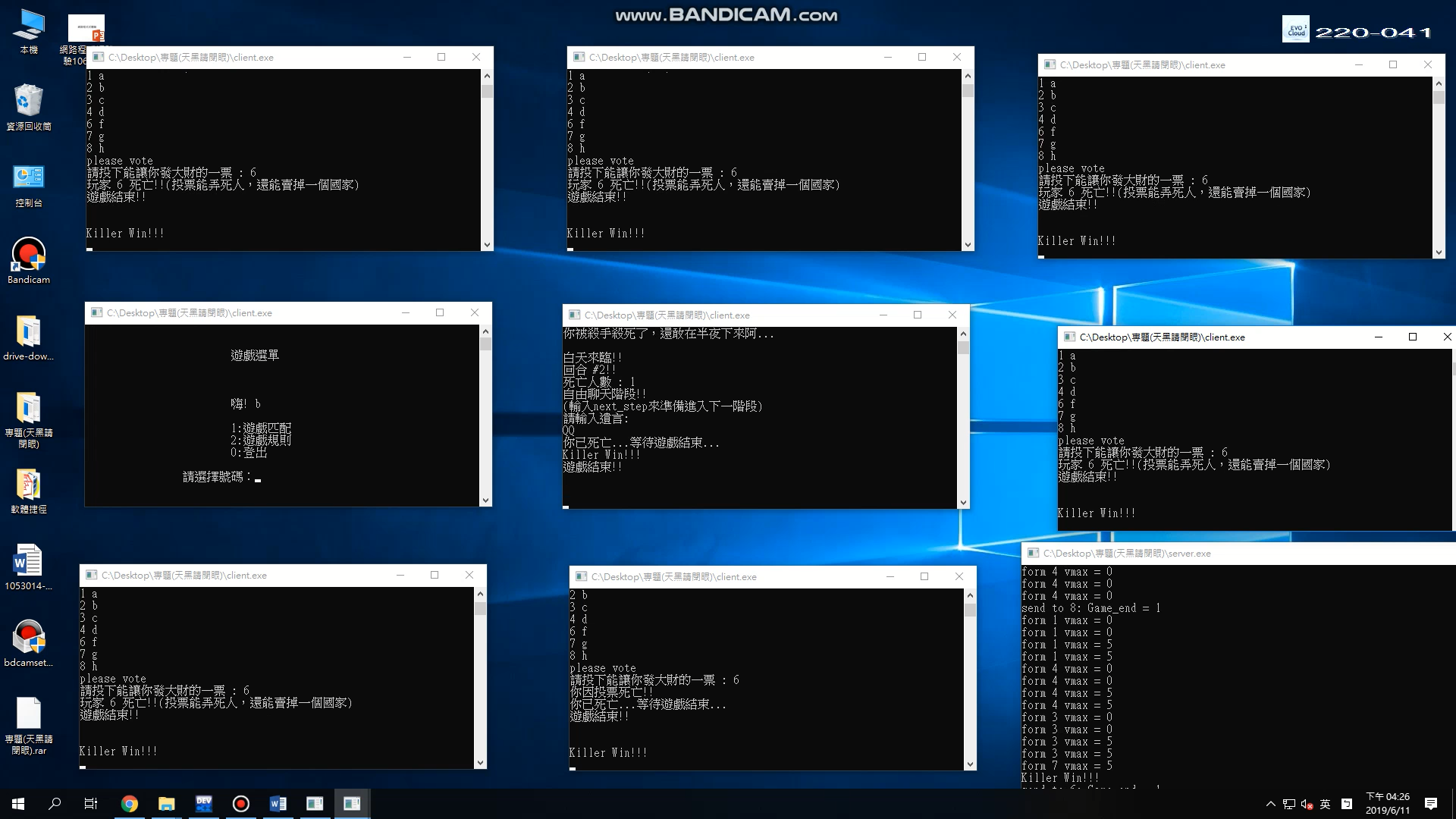
登入畫面 

第一天

殺手階段

警察階段

投票階段

遊戲結束

4. 本作品架構與具備功能

主要是運用TCP進行遊玩，在每個client設置一個thread跟server端進行接收，而在main裡運行輸入的動作，server端方面，我們為每個連進來的client都設置一個thread用以進行接收跟傳輸。

5. 本作品的核心

* 基本上連線是以TCP為主，畢竟要確保有收到相對應的指令，例如:下一步驟的指令，或是勝敗結果的指令，作品以小黑窗的形式展現，本遊戲的重點著重在於玩家之間的辯論，所以我在玩家辯論階段不是至時間限制，讓玩家自己掌控。

6. 使用非課程所教的技術

無

7. 人員分工

林煥挺 : 登入註冊介面、遊戲選單、遊戲規則介面、執行隨機諺語

葉哲睿 : 遊戲介面、遊戲規則、遊戲主體

8.

第10週:找組員

第11週:找組員

第12週:討論專題方向

第13週:討論專題方向

第14週:開始實做

第15週:開始實做

第16週:Debug

第17週:完成專題並繳交

9.心得 :

* 林煥挺 : 很快地來到了學期末，這學期學到很多有關網路程式設計的東西，也利用這次期末專題的機會，把整學期所學到的都運用上了，期望在未來的課程，也能將這學期所學的都派上用場。
* 葉哲睿 : 這學期學期來，我認為老師腳的東西都很淺顯易懂，讓我在專題運用上很有效率，原本在規畫這個專題的時候，我們是想做聊天室，配合一些小遊戲的，但是後來想到聊天室的延伸，我們可以做以聊天為主的遊戲，所以就改成天黑請閉眼了，一開始再決定的時候我還覺得滿簡單的，但是到後來我開始規劃架構之後，才發現有好多好多的問題和困難，所幸後來有一一找到解決辦法。

10.

程式碼

Client:

#include "stdio.h"

#include "string.h"

#include "windows.h"

#include <stdlib.h>

#include <pthread.h>

#include <ctype.h>

#define sleep(x) Sleep(1000 \* (x))

#define MAXLINE 1024

char reg\_name[30]="",reg\_pwd[10]="",input[1024]="";

char on\_name[30],on\_pwd[10],Die[10],c\_ans[100];

int test=0,playernum=1,Round=1,step=1,kill=0,n,career,killer\_state=0,target=0;//round

int police\_state=0;

int game\_end=0;///0=遊戲未結束 1=殺手勝利 2=警察勝利

int gamestart=0,speakable=1,my\_num,output=0,alive=1;

int vote\_num=0,vote\_state=0;

struct Player

{

char name[10];

int num;

int career;

};

struct Player user;

SOCKET sd;

struct sockaddr\_in serv;

char str[1024];

char str\_r[1024];

WSADATA wsadata;

int timeout = 50,n;

pthread\_t thread1;

void ten\_c()

{

SOCKET sd;

struct sockaddr\_in serv,cli;

char str[1024];

WSADATA wsadata;

int n,serv\_len;

char str\_r[1024];

WSAStartup(0x101,(LPWSADATA) &wsadata); // 呼叫 WSAStartup() 註冊 WinSock DLL 的使用

sd=socket(AF\_INET, SOCK\_DGRAM, 0);

char broadcast = 'a';

if( setsockopt(sd, SOL\_SOCKET, SO\_BROADCAST, &broadcast, sizeof(broadcast))<0)

printf("setsockopt() error!\n");

cli.sin\_family = AF\_INET;

cli.sin\_addr.s\_addr = 0;

cli.sin\_port = htons(5678);

if( bind(sd, (LPSOCKADDR) &cli, sizeof(cli)) <0 )

printf("bind error!\n");

serv\_len=sizeof(serv);

int count=0;

//while(count<5){

n=recvfrom(sd, str, MAXLINE, 0,(LPSOCKADDR) &serv,&serv\_len ); //由echo server接收

str[n]='\0';

printf("來句隨機諺語吧! :%s\n",str);

count++;

//}

closesocket(sd); //關閉 socket

WSACleanup(); // 結束 WinSock DLL 的使用

//return 0;

}

///用戶註冊系統

void regist()

{

system("cls");

printf("\n\n\t\t\t歡迎使用用天黑請閉眼註冊系統\n\n");

while(1)

{

//輸入用戶名

printf("\t\t請輸入暱稱[不能超過10個字]：");

scanf("%s",reg\_name);

//判斷暱稱

if(strlen(reg\_name)<=10)

{

while(1)

{

//輸入入密碼

printf("\n\t\t請輸入密碼[密碼長度為五位數]：");

scanf("%s",reg\_pwd);

//判斷密碼

if(strlen(reg\_pwd)==5)

{

printf("\n\n\t\t已註冊完成!!! 您的暱稱是%s,密碼是%s\n\n",reg\_name,reg\_pwd);

system("pause");

system("cls");

break;

}

else

{

printf("\n\t\t密碼的長度為%d，請重新輸入\n",strlen(reg\_pwd));

//system("pause");

sleep(0);

}

}

break;

}

else

{

printf("\n\t\t暱稱的長度為%d，請重新輸入\n\n",strlen(reg\_name));

sleep(0);

}

}

}

void \*recvMess(void \*argu) ///接收

{

//printf("等待配對中......(%d/8)\n",playernum);

while (1)

{

//setsockopt(sd,SOL\_SOCKET,SO\_RCVTIMEO,&timeout,sizeof(timeout));

n=recv(sd, str\_r, 1024, 0); //接收來自其他client的訊息

if(n>1)

{//strcpy

//printf("step2 recv: %s (%d bytes)\n",str\_r,strlen(str\_r)+1);

if(strcmp(str\_r,"newplayer!!!")==0)

{

playernum++;

printf("有新玩家加入大廳\n");

}

else if(strcmp(str\_r,"gamestart")==0)

{

gamestart=1;

//printf("gamestart=%d\n",gamestart);

strcpy(str\_r,"");

while(1)///1.白天誰死了->2.自由說話->3.投票處死->4.黑夜殺手殺人->5.警察指認->...

{

killer\_state=0;

police\_state=0;

if(step==1)///白天------------------------------------------

{

if(kill!=0)

{

if(atoi(Die)==user.num)

{

printf("你被殺手殺死了，還敢在半夜下來阿...\n");

alive=0;

}

else

printf("%s 被殺手殘忍殺害...\n",Die);

}

n=recv(sd, str\_r, 1024, 0);

if(strcmp(str\_r,"1")==0)

{

game\_end=1;

break;

}

else if(strcmp(str\_r,"2")==0)

{

game\_end=2;

break;

}

n=recv(sd, str\_r, 1024, 0);

if(strcmp(str\_r,"Next step")==0)

{

//printf("%s to step 2\n",str\_r);

step++;

}

}

if(alive==0)

break;

if(step==2)///自由對話-------------------------------------

{

strcpy(str\_r,"");

while(1)

{

n=recv(sd, str\_r, 1024, 0); //接收來自其他client的訊息

if(strcmp(str\_r,"Next step")==0)

{

//printf("%s to step 3\n",str\_r);

step++;

break;

}

if(n>5)

printf("recv: %s (%d bytes)\n",str\_r,strlen(str\_r)+1);

}

}

if(Round!=1)///投票-----------------------------------------

{

if(step==3)

{

while(1)///接收存活名單

{

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

if(strcmp(str\_r,"please vote")==0)

{

vote\_state=1;

break;

}

}

n=recv(sd, str\_r, 10, 0);///接收投票結果

if(n>1)

{

kill+=1;

vote\_num=atoi(str\_r);

if(user.num==vote\_num)

{

printf("你因投票死亡!!\n");

alive=0;

}

else

{

printf("玩家 %d 死亡!!(投票能弄死人，還能賣掉一個國家)\n",vote\_num);

}

}

n=recv(sd, str\_r, 1024, 0);

if(strcmp(str\_r,"1")==0)

{

game\_end=1;

break;

}

else if(strcmp(str\_r,"2")==0)

{

game\_end=2;

break;

}

while(1)

{

n=recv(sd, str\_r, 100, 0);///接收next step指令

if(strcmp(str\_r,"Next step")==0)

{

//printf("%s\n",str\_r);

step++;

break;

}

}

}

}

else

{

sleep(1);

step++;

}

if(alive==0)

break;

if(step==4)///殺手回合----------------------------------------

{

target=0;

printf("\n夜晚降臨，殺手現身...\n");

strcpy(str\_r,"");

while(1)

{

if(user.career!=1)///非殺手

{

printf("等待其他玩家行動\n");

n=recv(sd, Die, 10, 0);///收到死者代碼(killer\_state==2)

//printf("Die : %s\n",Die);

if(n>1)

{

kill++;

//printf("kill = %d\n",kill);

int x=atoi(Die);

if(x==user.num)

{

alive=0;

}

}

n=recv(sd, str\_r, 1000, 0);

if(strcmp(str\_r,"Next step")==0)///收到next step指令

{

step++;

break;

}

}

else///殺手

{

strcpy(str\_r,"");

n=recv(sd, str\_r, 1024, 0);

//printf("%s\n",str\_r);

if(n>1)

{

if(killer\_state==0)///殺手聊天階段

{

printf("recv: %s (%d bytes)\n",str\_r,strlen(str\_r)+1);

}

else if(killer\_state==1)///收到存活名單

{

while(1)

{

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

if(strcmp(str\_r,"end")==0)

{

target=1;

break;

}

}

}

else if(killer\_state==2)///收到死者代碼

{

strcpy(Die,str\_r);

//printf("Die : %s\n",Die);

kill+=1;

//printf("kill = %d\n",kill);

killer\_state=3;

int x=atoi(Die);

if(x==user.num)

{

alive=0;

}

}

if(strcmp(str\_r,"Next step")==0)

{

//printf("Next step to 5\n");

step++;

break;

}

}

}

}

}

if(step==5)///警察回合----------------------------------------------

{

target=0;

printf("\n警察行動中...\n");

while(1)

{

if(user.career!=2)

{

sleep(1);

printf("等待其他玩家行動\n");

n=recv(sd, str\_r, 1024, 0);

if(strcmp(str\_r,"Next step")==0)

{

step=1;

Round+=1;

break;

}

}

else

{

n=recv(sd, str\_r, 1024, 0);

if(n>1)

{

if(police\_state==0)

printf("recv: %s (%d bytes)\n",str\_r,strlen(str\_r)+1);

else if(police\_state==1)

{

while(1)

{

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

if(strcmp(str\_r,"end")==0)

{

target=1;

break;

}

}

}

else if(police\_state==2)

{

strcpy(c\_ans,str\_r);

printf("%s\n",c\_ans);

printf("等待其他玩家行動\n");

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

if(strcmp(str\_r,"Next step")==0)

{

step=1;

Round+=1;

break;

}

}

}

}

}

}

}

if(alive==1)

while(1)

{

n=recv(sd, str\_r, 1024, 0);

if(strcmp(str\_r,"1")==0)

{

game\_end=1;

break;

}

else if(strcmp(str\_r,"2")==0)

{

game\_end=2;

break;

}

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

break;

}

else

while(1)

{

n=recv(sd, str\_r, 1024, 0);

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

if(strcmp(str\_r,"Killer Win!!!")==0)

{

game\_end=1;

break;

}

else if(strcmp(str\_r,"Police And People Win!!")==0)

{

game\_end=2;

break;

}

n=recv(sd, str\_r, 1024, 0);

printf("%s\n",str\_r);

break;

}

if(game\_end!=0)

break;

}

}

if(game\_end!=0)

{

//sleep(1);

return (NULL);

break;

}

}

}

void Gameplay()///輸入

{

int k=0,w=0,z=0;

while(1)

{

//printf("waiting for playerww......\n");

if(gamestart==0)

{

if(z==0)

ten\_c();

sleep(1);

printf("等待其他玩家......\n");

}

if(gamestart!=0)

{

//system("cls");

printf("\n\nGame Startttttttttttttttt!!!\n");

break;

}

z++;

continue;

}

int d=0;

while(1)///1.白天誰死了->2.自由說話->3.投票處死->4.黑夜殺手殺人->5.警察指認->...

{

//if(alive==0)

//break;

if(step==1)

{

//printf("\nstep:1\n");

printf("\n白天來臨!!\n");

if(Round==1)

{

if(user.career==1)

{

printf("你是殺手，殺人於無形是你的專長!!\n");

}

else if(user.career==2)

{

printf("你是警察，法網恢恢疏而不漏!!\n");

}

else if(user.career==3)

{

printf("你是平民，不知道...不曉得...高雄發大財!!\n");

}

while(1)

{

if(step==2)

break;

}

}

else

{

while(1)

{

if(game\_end==1||game\_end==2)

break;

if(step==2)

break;

}

//continue;

}

}

if(game\_end==1||game\_end==2)

break;

if(step==2)

{

//printf("\nstep:2\n");

strcpy(input,"");

printf("回合 #%d!!\n",Round);

printf("死亡人數 : %d\n",kill);

printf("自由聊天階段!!\n");

printf("(輸入next\_step來準備進入下一階段)\n");

if(Round==1)

send(sd, input, strlen(input)+1, 0);

while(1)

{

if(alive==0)

{

printf("請輸入遺言:\n");

gets(input);

strcat(input,"(遺言)");

send(sd, input, strlen(input)+1, 0);

break;

}

gets(input);

send(sd, input, strlen(input)+1, 0);

if(strcmp(input,"next\_step")==0)

{

break;

}

}

if(alive==0)

break;

while(1)

{

if(step==3)

{

//printf("next step to 3\n");

break;

}

else

continue;

}

}

if(Round!=1)////////////////////////////

{

printf("\n投票階段\n");

if(step==3)

{

while(1)

{

if(vote\_state==1)

break;

}

char vo[10];

//sleep(1);

printf("請投下能讓你發大財的一票 : ");

gets(vo);

send(sd, vo, strlen(vo)+1, 0);

}

while(1)

{

if(game\_end==1||game\_end==2)

break;

if(step==4)

break;

}

}

else

{

//printf("\nstep:3\n");

while(1)

{

if(step==4)

{

break;

}

continue;

}

}

if(game\_end==1||game\_end==2)

break;

if(alive==0)

break;

if(step==4)

{

//printf("\nstep:4\n");

if(user.career==1)///殺手

{

printf("你是殺手，請殺人(指令=kill)\n");

while(1)

{

gets(input);

send(sd, input, strlen(input)+1, 0);///殺手對話

if(strcmp(input,"kill")==0)///kill指令

{

killer\_state=1;

strcpy(input,"");

}

if(killer\_state==1)///輸入要殺的人

{

while(1)

{

if(target==1)

break;

}

printf("請輸入獵殺目標 : ");

gets(input);

send(sd, input, strlen(input)+1, 0);

killer\_state=2;

break;

}

}

while(1)

{

if(step==5)

{

break;

}

}

}

else

{

while(1)

{

if(step==5)

{

break;

}

}

}

}

if(step==5)

{

printf("\n警察指認階段\n");

if(user.career==2)///警察

{

printf("你是警察，警察可以進行對話(指認指令=choose)\n");

while(1)

{

gets(input);

send(sd, input, strlen(input)+1, 0);///警察對話

if(strcmp(input,"choose")==0)///choose指令

{

police\_state=1;

}

if(police\_state==1)///輸入要指認的人

{

while(1)

{

if(target==1)

break;

}

printf("請輸入指認目標 : ");

gets(input);

send(sd, input, strlen(input)+1, 0);

police\_state=2;

break;

}

}

while(1)

{

if(step==1)

{

sleep(1);

break;

}

}

}

else

{

while(1)

{

if(step==1)

{

sleep(1);

break;

}

}

}

}

}

if(alive==0)

printf("你已死亡...等待遊戲結束...\n");

while(1)

{

if(game\_end!=0)

{

//printf("%s\n",str\_r);

printf("遊戲結束!!\n\n\n");

sleep(8);

closesocket(sd);

break;

}

}

}

void Connect\_server()

{

WSAStartup(0x101,(LPWSADATA) &wsadata); // 呼叫 WSAStartup() 註冊 WinSock DLL 的使用

sd=socket(AF\_INET, SOCK\_STREAM, 0); //開啟一個 TCP socket.

serv.sin\_family = AF\_INET;

serv.sin\_addr.s\_addr = inet\_addr("127.0.0.1");

serv.sin\_port = htons(5678);

connect(sd, (LPSOCKADDR) &serv, sizeof(serv)); // 連接至server

send(sd, reg\_name, strlen(reg\_name)+1, 0);

n=recv(sd, str\_r, 1024, 0); //接收來自其他client的訊息

if(n>1)

{

my\_num=atoi(str\_r);

//printf("my number is : %d\n",my\_num);

}

n=recv(sd, str\_r, 1024, 0); //接收來自其他client的訊息

if(n>1)

{

career=atoi(str\_r);

//printf("my Career is : %d\n",career);

}

user.career=career;

strcpy(user.name,on\_name);

user.num=my\_num;

}

//判斷是否註冊

int judge()

{

if(strcmp(reg\_name,"")==0&&strcmp(reg\_pwd,"")==0)

{

printf("\n\n\t\t您尚未註冊，請先註冊！\n\n");

//system("pause");

printf("\t\t兩秒後自動跳轉...");

sleep(2);

system("cls");

return 0;

}

else

{

return 1;

}

}

//用戶登入

void dl()

{

int i;

system("cls");

printf("\n\n\t\t\t歡迎使用天黑請閉眼登入系統\n\n");

//三次登入機會

for(i=1; i<=3; i++)

{

printf("\t\t請輸入暱稱:");

scanf("%s",on\_name);

printf("\n\t\t請輸入密碼:");

scanf("%s",on\_pwd);

if(strcmp(reg\_name,on\_name)==0&&strcmp(reg\_pwd,on\_pwd)==0)

{

printf("\n\n\t\t歡迎遊玩天黑請閉眼!!\n\n");

test++;

//system("pause");

printf("\n\n\t\t導向遊戲選單頁面.....\n\n");

sleep(1);

system("cls");

break;

}

else

{

printf("\n\n\t\t登入失敗，請重新登入，您還有%d次機會\n\n",3-i);

//system("pause");

if(3-i==0)

{

printf("\t\t兩秒後自動跳轉...");

sleep(2);

}

}

}

}

//遊戲規則

void game\_rule()

{

int y;

system("cls");

printf("\n\n\t\t遊戲規則\n\n");

printf("\n\t\t八人版");

printf("\n\n\t\t職業分為");

printf("\n\n\t\t1 殺手(Killer)");

printf("\n\t\t2 警察(Police)");

printf("\n\t\t3 平民(People)");

printf("\n\t\t會在第一天的白天告知你的職業");

printf("\n\n\t\t遊戲分為五個階段");

printf("\n\n\t\t白天>自由對話>投票>黑夜，殺手現身>警察指認>>>>>>白天...");

printf("\n\t\t自由對話 : 可和其他活著的人進行溝通，所有人輸入next\_step進入下一階段。");

printf("\n\t\t投票 : 在第一回合以外的時候進行投票，投出你認為的嫌疑人，票數最高的玩家死亡，若有同票，以玩家代碼大者優先死亡。");

printf("\n\t\t黑夜殺手現身 : 除了殺手之外的人無法溝通，殺手選擇殺掉一位玩家，若兩位殺手不同調，則後手選擇的殺手的目標為優先，殺手的指令為kill輸入後獲得存活玩家代碼，再輸入你的目標代碼。");

printf("\n\t\t警察指認 : 除了警察之外的人無法溝通，警察選擇指認一位玩家，若兩位警察不同調，則後手選擇的警察的目標為優先，警察的指令為choose輸入後獲得存活玩家代碼，再輸入你的目標代碼。");

printf("\n\n\t\t勝敗條件");

printf("\n\n\t\t1.若警察死光，則殺手獲勝");

printf("\n\t\t2.若殺手死光，則警察和平民獲勝");

printf("\n\n\t\t先死亡的玩家請勿跳出遊戲，耐心等待整場遊戲結束，你不一定會是輸家。\n\n");

system("pause");

}

int Main\_Menu()

{

int id;

system("cls");

//輸出界面

printf("\n\n\t\t\t歡迎遊玩天黑請閉眼\n\n");

printf("\t\t\t1:註冊\n");

printf("\t\t\t2:登入\n");

printf("\t\t\t0:離開\n\n");

//輸入功能號碼

printf("\t\t請選擇號碼：");

scanf("%d",&id);

return id;

}

int Game\_Menu()

{

int id2;

system("cls");

printf("\n\n\t\t\t遊戲選單\n\n");

printf("\n\n\t\t\t嗨! %s\n\n",on\_name);

printf("\t\t\t1:遊戲匹配\n");

printf("\t\t\t2:遊戲規則\n");

printf("\t\t\t0:登出\n\n");

printf("\t\t請選擇號碼：");

scanf("%d",&id2);

return id2;

}

int main()

{

int id,id2,out;

int y\_or\_n;

while(1)///登入畫面

{

int test2=0;

// /\*

while(1)///主選單

{

id=Main\_Menu();

switch(id)

{

case 1:

regist();///註冊

break;

case 2:

if(judge()>0)

dl();///登入

break;

case 0:

printf("\n\t\t\t確定真的要離開遊戲嗎TAT\n\n");

printf("\t\t\t1:是\n");

printf("\t\t\t2:否\n");

printf("\n\t\t請選擇號碼：");

scanf("%d",&y\_or\_n);

switch(y\_or\_n)

{

case 1:

exit(1);

break;

case 2:

continue;

break;

default:

printf("\n\t\t輸入的號碼有誤，請回到選單再次輸入\n");

//system("pause");

printf("\t\t兩秒後自動跳轉...");

sleep(2);

system("cls");

}

break;

default:

printf("\n\t\t輸入的號碼有誤，請再次輸入\n");

printf("\t\t兩秒後自動跳轉...");

sleep(2);

//system("pause");

system("cls");

}

if(test==1)

break;

}

//\*/

while(1)///遊戲選單

{

system("cls");

printf("\n\n\t\t\t遊戲選單\n\n");

printf("\n\n\t\t\t嗨! %s\n\n",on\_name);

printf("\t\t\t1:遊戲匹配\n");

printf("\t\t\t2:遊戲規則\n");

printf("\t\t\t0:登出\n\n");

printf("\t\t請選擇號碼：");

scanf("%d",&id2);

switch(id2)

{

case 1:

system("cls");

printf("\t\t\t\nConnect to server\n");

Connect\_server();///連結至server

pthread\_create(&thread1, NULL, &recvMess, NULL);///開啟多緒

Gameplay();

break;

case 2:

game\_rule();

break;

case 0:

printf("\n\t\t\t確定真的要登出且關閉遊戲嗎嗎TAT\n\n");

printf("\t\t\t1:是\n");

printf("\t\t\t2:否\n");

printf("\n\t\t請選擇號碼：");

scanf("%d",&out);

if(out==1)

{

printf("\t\t登出中...");

sleep(1);

test2++;

exit(0);

}

else if(out==2)

continue;

else

{

printf("\n\t\t輸入的號碼有誤，請回到選單再次輸入\n");

printf("\t\t兩秒後自動跳轉...");

sleep(2);

system("cls");

}

break;

}

if(test2!=0)

break;

}

}

}

Server:

#include <stdio.h> /\* for printf(), fprintf() \*/

#include <winsock.h> /\* for socket(),... \*/

#include <stdlib.h> /\* for exit() \*/

#include <time.h>

#include <ctype.h>

#define sleep(x) Sleep(1000 \* (x))

#define MAXLINE 1024

void \*ThreadMain(void \*arg); /\* Main program of a thread \*/

struct ThreadArgs /\* Structure of arguments to pass to client thread \*/

{

int clntSock; /\* Socket descriptor for client \*/

int num;

char name[10];

int career;

};

int cli[10]= {0},cnum=0,gamestart=0,killer=0,police=0,people=0,next=0,alive[8]= {1,1,1,1,1,1,1,1},ToNextRound=0,vote\_done=0,vote[8]= {0,0,0,0,0,0,0,0};

int killtemp=0,kdone=0;

int cdone=0,ctemp=0;

int career[8];

int game\_end=0;

char player[8][10];

time\_t t1,t2;

int ten()

{

SOCKET serv\_sd; /\* socket 描述子 \*/

int cli\_len, n,i=1,RanNum,l;

char str[MAXLINE];

char QuoteA[MAXLINE]=" Man proposes, God disposes.";

char QuoteB[MAXLINE]=" Every flow must have its ebb.";

char QuoteC[MAXLINE]=" Experience is the best teacher.";

char QuoteD[MAXLINE]=" Great minds think alike.";

char QuoteE[MAXLINE]=" It is never too late to mend.";

//char QuoteA[MAXLINE]="人若軟弱就是自己最大的敵人;人若勇敢就是自己最好的朋友。";

//char QuoteB[MAXLINE]="腳下的路還很長，活著就要堅決走下去。";

//char QuoteC[MAXLINE]="生命不是要超越別人，而是要超越自己。";

//char QuoteD[MAXLINE]="羨慕別人得到的，不如珍惜自己擁有的。";

//char QuoteE[MAXLINE]="黑暗總會過去，黎明終會到來。";

struct sockaddr\_in serv, cli;

WSADATA wsadata;

WSAStartup(0x101, &wsadata); //呼叫 WSAStartup() 註冊 WinSock DLL 的使用

serv\_sd=socket(AF\_INET, SOCK\_DGRAM, 0);// 開啟 UDP socket

char broadcast = 'a';

setsockopt(serv\_sd, SOL\_SOCKET, SO\_BROADCAST, &broadcast, sizeof(broadcast));

cli\_len = sizeof(cli);

printf("server will broadcast.\n");

cli.sin\_family = AF\_INET;

cli.sin\_addr.s\_addr = inet\_addr("255.255.255.255");

cli.sin\_port = htons(5678);

for(l=0;l<5;l++){

RanNum=(rand()%5)+1;

if(RanNum==1)

strcpy(str,QuoteA);

else if(RanNum==2)

strcpy(str,QuoteB);

else if(RanNum==3)

strcpy(str,QuoteC);

else if(RanNum==4)

strcpy(str,QuoteD);

else if(RanNum==5)

strcpy(str,QuoteE);

//memset(str, i%10 +'0', sizeof(str));

sendto(serv\_sd, str, strlen(str), 0,(LPSOCKADDR)&cli,cli\_len);

printf("server broadcast: %s\n",str); // 顯示送去client 的字串

sleep(1);

//i++;

}

//結束 WinSock DLL 的使用

closesocket(serv\_sd);

//closesocket(cli\_sd);

WSACleanup();

}

int main(int argc, char \*argv[])

{

srand( time(NULL) );

struct ThreadArgs \*threadArgs; /\* Pointer to argument structure for thread \*/

DWORD threadID; /\* Thread ID from CreateThread() \*/

char str[1024]="";

int servSock; /\* Socket descriptor for server \*/

int clntSock,j=0,i,new; /\* Socket descriptor for client \*/

WSADATA wsaData; /\* Structure for WinSock setup communication \*/

struct sockaddr\_in echoServAddr; /\* Local address \*/

struct sockaddr\_in echoClntAddr; /\* Client address \*/

unsigned int clntLen; /\* Length of client address data structure \*/

if (WSAStartup(0x101, &wsaData) != 0) /\* Load Winsock 2.0 DLL \*/

{

printf("WSAStartup() failed");

exit(1);

}

/\* Create socket for incoming connections \*/

if ((servSock = socket(PF\_INET, SOCK\_STREAM, IPPROTO\_TCP)) < 0)

printf("socket() failed");

/\* Construct local address structure \*/

memset(&echoServAddr, 0, sizeof(echoServAddr)); /\* Zero out structure \*/

echoServAddr.sin\_family = AF\_INET; /\* Internet address family \*/

echoServAddr.sin\_addr.s\_addr = htonl(INADDR\_ANY); /\* Any incoming interface \*/

echoServAddr.sin\_port = htons(5678); /\* Local port \*/

/\* Bind to the local address \*/

if (bind(servSock, (struct sockaddr \*) &echoServAddr, sizeof(echoServAddr)) < 0)

printf("bind() failed");

/\* Mark the socket so it will listen for incoming connections \*/

if (listen(servSock, 3) < 0)

printf("listen() failed");

printf("Server is waiting for clients.\n");

for (;;) /\* Run forever \*/

{

clntLen = sizeof(echoClntAddr);

clntSock = accept(servSock, (struct sockaddr \*) &echoClntAddr, &clntLen);

if(cnum>=8)///?湔鈭箸?寥?-

{

strcpy(str,"The number is full");

send(clntSock,str,strlen(str)+1,0);

closesocket(clntSock);

}

else

{

strcpy(str,"");

recv(clntSock, str, 1000, 0);///?交?拙振?梁迂

for(i=0; i<8; i++)

{

if(cli[i]==0)

{

int c;

cli[i]=clntSock;

new=i;

cnum++;

printf("The %d client socket is in cli\_sd[%d]\n",cnum,i);

threadArgs = (struct ThreadArgs \*) malloc(sizeof(struct ThreadArgs));

threadArgs -> clntSock = clntSock;

threadArgs -> num = new+1;

strcpy(threadArgs -> name,str);

strcpy(player[i],str);

itoa(new+1,str,10);

send(cli[new],str,strlen(str)+1,0);///??拙振?Ⅳnum

while(1)///???瑟平

{

int x=rand();

x%=3;

if(x==0&&killer<2)

{

killer++;

c=1;

threadArgs ->career=c;

career[i]=c;

break;

}

else if(x==1&&police<2)

{

police++;

c=2;

threadArgs ->career=c;

career[i]=c;

break;

}

else if(x==2&&people<4)

{

people++;

c=3;

threadArgs ->career=c;

career[i]=c;

break;

}

}

printf("\n\tplayer num :%d\n\tname :%s\n\tcareer :%d",threadArgs->num,threadArgs->name,threadArgs->career);

strcpy(str,"");

itoa(c,str,10);

send(cli[new],str,strlen(str)+1,0);///??拙振?瑟平

if (CreateThread(NULL, 0, (LPTHREAD\_START\_ROUTINE) ThreadMain, threadArgs, 0, (LPDWORD) &threadID) == NULL)

printf("thread\_create() failed");

ten();

printf("New client with thread ID: %ld and socket:%d\n", threadID,clntSock);

printf("New client cnum:%d and cli\_sd:%d\n", cnum,cli[cnum-1]);

break;

}

}

strcpy(str,"newplayer!!!");

for(j=0; j<10; j++)

{

if(cli[j]>0)

{

if(j!=new)

{

send(cli[j],str,strlen(str)+1,0);

}

}

}

}

/\* Create separate memory for client argument \*/

}

}// end main

void \*ThreadMain(void \*threadArgs)

{

int Round=1;///round

int killer\_alive=2,police\_alive=2;

int all\_alive=8;

int vote\_player=0,vote\_num=0;

int next\_ok[8]={0};

int timeout = 50;

int clntSock,j,i,num,step=1,c;

char who[10]; /\* Socket descriptor for client connection \*/

int m\_alive=1;

/\* Extract socket file descriptor from argument \*/

clntSock = ((struct ThreadArgs \*) threadArgs) -> clntSock;

num = ((struct ThreadArgs \*) threadArgs) -> num;

strcpy(who,((struct ThreadArgs \*) threadArgs) -> name);

c =((struct ThreadArgs \*) threadArgs) ->career;

char echoBuffer[1000]; /\* Buffer for echo string \*/

int recvMsgSize; /\* Size of received message \*/

/\* Send received string and receive again until end of transmission \*/

while(1)

{

if(cnum>=8&&gamestart==0)///?湔鈭箸?寥?-

{

sleep(1);

strcpy(who,"game are going to start...");

send(clntSock, who, strlen(who)+1, 0);

gamestart=1;

strcpy(who,"gamestart");

sleep(1);

send(clntSock, who, strlen(who)+1, 0);

}

if(gamestart==1)///1.?賢予隤唳香鈭?>2.?芰隤芾店->3.?巨?香->4.暺?畾箸?畾箔犖->5.霅血???->...

{

while(1)

{

for(j=0;j<8;j++)

{

vote[j]=0;

}

next=0,vote\_done=0;

if(step==1)///?賢予---------------------------------------------------------------------------------------------------------------

{

printf("\nstep1:%d\n",num);

printf("Day coming!!\n");

if(Round==1)

sleep(2);

if(all\_alive<=8)///???斗

{

sleep(1);

if(police\_alive==0)

game\_end=1;

if(killer\_alive==0)

game\_end=2;

char e[10];

strcpy(e,"");

itoa(game\_end,e,10);

send(clntSock, e, strlen(e)+1, 0);

printf("send to %d: Game\_end = %d\n",num,game\_end);

if(game\_end!=0)

break;

}

strcpy(who,"Next step");

send(clntSock, who, strlen(who)+1, 0);

printf("sned Next step\n");

step++;

}

if(step==2)///?芰撠店------------------------------------------------------------------------------------------------------------

{

kdone=0;

cdone=0; ctemp=0;

strcpy(who,"");

strcpy(echoBuffer,"");

printf("\nstep2:%d\n",num);

printf("free chat!!\n");

while (1) // zero indicates end of transmission

{

printf("%d next = %d\n",num,next);

// Receive message from client

//setsockopt(clntSock,SOL\_SOCKET,SO\_RCVTIMEO,&timeout,sizeof(timeout));

recvMsgSize = recv(clntSock, echoBuffer, 1000, 0);

if(recvMsgSize>1)

{

for(j = 0 ; j < 8 ; j++)

{

if(cli[j]!=clntSock&&alive[j]==1)

{

sprintf(who, "%d", num);

//printf("str = %s\n", str);

strcat(who," : ");

strcat(who,echoBuffer);

send(cli[j], who, strlen(who)+1, 0);

}

}

printf("recv and send : %s\n",who);

if(strcmp(echoBuffer,"next\_step")==0)

{

next++;

printf("next = %d\n",next);

while(1)

{

if(next>=all\_alive)

{

printf("next>=all\_alive : %d\n",num);

if(alive[num-1]==1)

{

strcpy(who,"Next step");

send(clntSock, who, strlen(who)+1, 0);

}

break;

}

}

}

if(next>=all\_alive)

break;

}

if(m\_alive==0)

{

break;

}

}

if(m\_alive!=0)

step++;

}

if(m\_alive==0)

{

break;

}

if(Round!=1)///?巨------------------------------------------------------------------------------------------------------

{

printf("\nstep3:%d\n",num);

sleep(1);

if(step==3)

{

char pp[1024]="The alive player,please choose the number.(ex:1)";

send(clntSock, pp, strlen(pp)+1, 0);

if(m\_alive==1)

{

for(j=0; j<8; j++)

{

if(alive[j]==1)

{

char temp[10]="";

itoa(j+1,temp,10);

strcpy(pp,temp);

strcat(pp," ");

strcat(pp,player[j]);

send(clntSock, pp, strlen(pp)+1, 0);

printf("To %d : %s\n",num,pp);

}

}

strcpy(pp,"please vote");

send(clntSock, pp, strlen(pp)+1, 0);

printf("To %d : %s\n",num,pp);

}

if ((recvMsgSize = recv(clntSock, echoBuffer, 1000, 0)) < 0)///?交隞?Ⅳ

{

int error\_code=WSAGetLastError();

if(error\_code==WSAECONNRESET)

{

printf("Host disconnected unexpectedly\n");

closesocket(clntSock); // Close client socket

cli[num-1]=0;

cnum--;

break;

}

else

printf("recv failed with erroe code : %d\n",error\_code);

}

else

{

printf("player %d vote to : %s\n",num,echoBuffer);

int v=atoi(echoBuffer);

vote[v-1]+=1;

printf("vote[%d-1] = %d\n",v,vote[v-1]);

vote\_done+=1;

printf("vote\_done = %d\n",vote\_done);

while(1)

{

if(vote\_done>=all\_alive)

{

int vmax=0;

for(j=0; j<8; j++)///?曄巨?豢?憭?

{

printf("form %d vmax = %d\n",num,vmax);

if(vote[j]>vote[vmax])

vmax=j;

}

strcpy(pp,"");

itoa(vmax+1,pp,10);

send(clntSock, pp, strlen(pp)+1, 0);

all\_alive-=1;

alive[vmax]=0;

vmax+=1;

alive[vmax-1]=0;

if(career[vmax-1]==1)

killer\_alive-=1;

else if(career[vmax-1]==2)

police\_alive-=1;

if(num==vmax)

m\_alive=0;

if(num==vmax)

{

m\_alive=0;

}

step++;

break;

}

}

}

}

}

else

{

printf("\nstep3:%d\n",num);

step++;

}

if(Round>1)

{

if(all\_alive<=8)///???斗

{

sleep(1);

if(police\_alive==0)

game\_end=1;

if(killer\_alive==0)

game\_end=2;

char e[10];

strcpy(e,"");

itoa(game\_end,e,10);

send(clntSock, e, strlen(e)+1, 0);

printf("send to %d: Game\_end = %d\n",num,game\_end);

if(game\_end!=0)

break;

sleep(1);

strcpy(who,"Next step");

send(clntSock, who, strlen(who)+1, 0);///next step?誘

}

}

if(m\_alive==0)

break;

if(step==4)///畾箸???---------------------------------------------------------------------------------------------------------

{

killtemp=0;

printf("\nstep4:%d\n",num);

sleep(1);

while(1)

{

if(c==1)///畾箸?

{

if ((recvMsgSize = recv(clntSock, echoBuffer, 1000, 0)) < 0)

{

int error\_code=WSAGetLastError();

if(error\_code==WSAECONNRESET)

{

printf("Host disconnected unexpectedly\n");

closesocket(clntSock); // Close client socket

cli[num-1]=0;

cnum--;

break;

}

else

printf("recv failed with erroe code : %d\n",error\_code);

}

else if(strcmp(echoBuffer,"kill")==0)///killer\_state==1

{

char pp[1024]="The alive player,please choose the number.(ex:1)";

printf("%s\n",pp);

send(clntSock, pp, strlen(pp)+1, 0);

for(j=0; j<8; j++)

{

if(alive[j]==1)///?捏??瘣餉?

{

char temp[100]="";

int t=j+1;

itoa(t,temp,10);

strcpy(pp,temp);

strcat(pp," ");

strcat(pp,player[j]);

send(clntSock, pp, strlen(pp)+1, 0);

printf("%s\n",pp);

}

}

strcpy(pp,"end");

send(clntSock, pp, strlen(pp)+1, 0);

recvMsgSize = recv(clntSock, echoBuffer, 1000, 0);

int x=atoi(echoBuffer);

killtemp=x;

printf("vote : %d\n",killtemp);

kdone++;

if(kdone<2)

{

for(j=0; j<8; j++)///?喟策?虫?畾箸?雿?蝯西狐

{

if(cli[j]!=clntSock&&career[j]==1)

{

if(alive[j]==1)

{

//printf("str = %s\n", str);

strcpy(who,"the other killer voted : ");

strcat(who,echoBuffer);

send(cli[j], who, strlen(who)+1, 0);

printf("%s\n",who);

printf("kdone : %d\n",kdone);

}

}

}

}

break;

}

else

{

for(j=0; j<8; j++)///?喟策?虫?畾箸?雿?閮

{

if(cli[j]!=clntSock&&career[j]==1)

{

if(alive[j]==1)

{

sprintf(who, "%d", num);

//printf("str = %s\n", str);

strcat(who," : ");

strcat(who,echoBuffer);

send(cli[j], who, strlen(who)+1, 0);

printf("%s\n",who);

}

}

}

}

}

else///?捏??

{

break;

}

}

while(1)

{

if(kdone>=killer\_alive)

{

char deadplayer[10]="";

itoa(killtemp,deadplayer,10);

send(clntSock, deadplayer, strlen(deadplayer)+1, 0);///?單捏鈭狐(killer\_state==2)

printf("send to %d : %s\n",num,deadplayer);

strcpy(who,"Next step");

send(clntSock, who, strlen(who)+1, 0);///next step?誘

step++;

}

if(step==5)

{

sleep(1);

break;

}

}

}

if(step==5)///霅血???---------------------------------------------------------------------------------------------

{

printf("\nstep5:%d\n",num);

sleep(1);

while(1)

{

if(c==2)///霅血?

{

if ((recvMsgSize = recv(clntSock, echoBuffer, 1000, 0)) < 0)

{

int error\_code=WSAGetLastError();

if(error\_code==WSAECONNRESET)

{

printf("Host disconnected unexpectedly\n");

closesocket(clntSock); // Close client socket

cli[num-1]=0;

cnum--;

break;

}

else

printf("recv failed with erroe code : %d\n",error\_code);

}

if(strcmp(echoBuffer,"choose")==0)///police\_state==1

{

char pp[1024]="The alive player,please choose the number.(ex:1)";

printf("%s\n",pp);

send(clntSock, pp, strlen(pp)+1, 0);

for(j=0; j<8; j++)

{

if(alive[j]==1&&career[j]!=2)///?郎撖?瘣餉?

{

char temp[100]="";

itoa(j+1,temp,10);

strcpy(pp,temp);

strcat(pp,player[j]);

send(clntSock, pp, strlen(pp)+1, 0);

printf("%s\n",pp);

}

}

strcpy(pp,"end");

send(clntSock, pp, strlen(pp)+1, 0);

recvMsgSize = recv(clntSock, echoBuffer, 1000, 0);

int x=atoi(echoBuffer);

ctemp=x;

printf("vote : %d\n",ctemp);

cdone++;

if(cdone==1)

{

for(j=0; j<8; j++)///?喟策?虫?霅血?雿?蝯西狐

{

if(cli[j]!=clntSock&&career[j]==2)

{

if(alive[j]==1)

{

//printf("str = %s\n", str);

strcpy(who,"the other police voted : ");

strcat(who,echoBuffer);

send(cli[j], who, strlen(who)+1, 0);

printf("%s\n",who);

printf("cdone : %d\n",cdone);

}

}

}

}

break;

}

else

{

for(j=0; j<8; j++)///?喟策?虫?霅血?雿?閮

{

if(cli[j]!=clntSock&&career[j]==2)

{

if(alive[j]==1)

{

sprintf(who, "%d", num);

//printf("str = %s\n", str);

strcat(who," : ");

strcat(who,echoBuffer);

send(cli[j], who, strlen(who)+1, 0);

printf("%s\n",who);

}

}

}

}

}

else

break;

}

while(1)

{

if(cdone>=police\_alive)

{

if(c==2)

{

char knowplayer[100]="";

char temp[100];

itoa(ctemp,temp,10);

if(career[ctemp-1]==1)

{

strcpy(knowplayer,"player ");

strcat(knowplayer,temp);

strcat(knowplayer," is killer!!");

}

else if(career[ctemp-1]==2)

{

strcpy(knowplayer,"player ");

strcat(knowplayer,temp);

strcat(knowplayer," is police!!");

}

else if(career[ctemp-1]==3)

{

strcpy(knowplayer,"player ");

strcat(knowplayer,temp);

strcat(knowplayer," is people!!");

}

send(clntSock, knowplayer, strlen(knowplayer)+1, 0);///?單?隤狐(police\_state==2)

printf("%s\n",knowplayer);

}

break;

}

}

}

all\_alive-=1;

alive[killtemp-1]=0;

if(career[killtemp-1]==1)

killer\_alive-=1;

else if(career[killtemp-1]==2)

police\_alive-=1;

if(num==killtemp)

m\_alive=0;

strcpy(who,"Next step");

send(clntSock, who, strlen(who)+1, 0);///next step?誘

step=1;

Round+=1;

}

while(1)

{

sleep(1);

if(game\_end==1)

{

printf("Killer Win!!!\n");

strcpy(who,"Killer Win!!!");

send(clntSock, who, strlen(who)+1, 0);

break;

}

else if(game\_end==2)

{

printf("Police And People Win!!\n");

strcpy(who,"Police And People Win!!");

send(clntSock, who, strlen(who)+1, 0);

break;

}

}

if(game\_end!=0)

break;

}

}

sleep(3);

closesocket(clntSock); // Close client socket

cli[num-1]=0;

cnum--;

free(threadArgs);

return (NULL);

}

/\*

while(1)

{

printf("%d ready\n",num);

sleep(30);

}

\*/