基于 C#的黑白棋游戏的设计与实现

学　　院　　软件学院

专　　业　　软件工程

年　　级　　2007 级

姓　　名　　杨阳

指导教师　　张坤龙

2011 年　　6 月　　15 日

摘　　要

黑白棋游戏软件是通过对实际黑白棋游戏规则的分析和理解，利用代码进行制作出的可供人和人之间进行黑白棋娱乐对战的游戏软件，对于黑白棋的普及和推广都有重要的作用。黑白棋游戏软件将黑白棋由现实世界拓展到了虚拟世界，让世界上更多的人能享受这种游戏的乐趣。

论文在认真分析了黑白棋的起源，发展和规则的基础上，设计了黑白棋游戏的界面，功能模块和数据文件。该黑白棋可以实现网络上的联网对战，并且可以实时统计黑白棋双方棋盘上的子数，并及时对双方的胜负做出判断。在界面方面对游戏模式界面和游戏操作界面进行了统一规划。

黑白棋游戏软件的实现主要采用 C#进行编写，实现了单机双人对战以及非单机网络对战两种模式，提供了双方子数的即时显示以方便玩家。

关键词：黑白棋；联网对战；C#；网络；界面

ABSTRACT

Othello software is based on the actual Othello's rules of the game analysis and understanding，n accordance with the rules made by code available between people from the dark white entertainment play software，For Othello 's popularity and promotion has important role。Othello software will Othello’s world of expanding to virtual world，Let the world more and more people can enjoy the fun of the game.

Paper carefully analyzed the Othello's origin, development and rules based on，Design a black and white game interface, function module and data files。This Othello can realize the networking on the network, and can real-time statistics against black white board of both parties, and promptly son to both number in the outcome of the judgement. In data aspects of game mode interface and the game was unified planning operation interface.

Othello software for the realization of written mainly by c #, this application implements a single machine double against and the standalone netplay two kinds of modes, and also provides both the number of child with convenient instantly shows players.

Keywords: Othello; Networking against; C#

目　　录

中文译文

致谢

第一章　　绪论

1.1　开发背景

黑白棋软件（Othello）是一种游戏双方通过交替落子，设法使自己的子出现在对方的子两边从而将对方的子转化为自己的子的棋类。它的整个实现都依据现存的黑白棋规则以及界面来设计实现，将实际的桌上游戏转化为网络上的虚拟游戏。从整个游戏架构来讲，这个游戏软件是把黑白棋转化为代码并使其能够在网络上运行。从整个游戏设计理念上来讲，这个游戏的目的是为了进一步普及这个风靡全球的黑白棋游戏，能让更多的人学会并享受这个游戏的乐趣。

1.1.1　黑白棋的起源与规则

黑白棋起源于 19 世纪的英国。这个游戏从出现到风靡经历了漫漫的历史长河。其名字为黑白棋的原因是沿用了莎士比亚四大悲剧之一的奥赛罗来为此棋命名。所以黑白棋又称奥赛罗棋。图 1-1 即为一个典型的黑白棋实际对战时的场景。此图取材于网络。

图 1-1　　黑白棋对战场景

黑白棋在娱乐时需要双人对弈。棋盘为 8×8 方格，黑白棋总共使用 64 个棋子，每个棋子分正反两面，分别是黑色和白色。轮到一方下棋时，必须把棋下在与对方棋子相邻的空位上，要求所下的棋子和原有的己方棋子夹住对方的至少 1 个棋子，然后把被夹住的子变成己方的颜色。下棋过程中，任何棋子既不会从棋盘上拿走，也不会从 1 个格子移到另 1 个格子,吃子时，不会发生连锁反应，吃进的棋子不能再夹吃其他的子。当双方都无棋可下，或者方格全部占满后，棋局结束，子多的一方为胜方。

1.1.2 黑白棋在国内的发展情况

黑白棋这种游戏由于自身的一些特点，比如棋子特殊（双面），翻转频繁等原因，实体棋类并未在中国迅速普及。而与此相反的是，中国的黑白棋发展情况主要是靠网络以及虚拟游戏。黑白棋真正发展起来还是在网络普及以后，黑白棋作为一种经典的策略性游戏，受到了广大网友的喜爱。

1.2 相关概念

下面简单介绍一下黑白棋游戏以及规则中所涉及的一些相关概念。这有助于即使未接触过黑白棋的读者也能对黑白棋能建立一个初步的印象。

黑白棋的棋盘是一个有 8*8 方格的棋盘。下棋时将棋下在空格中间，而不是像围棋一样下在交叉点上。开始时在棋盘正中有两白两黑四个棋子交叉放置，黑棋总是先下子。

游戏时把自己颜色的棋子放在棋盘的空格上，而当自己放下的棋子在横、竖、斜八个方向内有一个自己的棋子，则被夹在中间的全部翻转会成为自己的棋子。并且，只有在可以翻转棋子的地方才可以下子。如果玩家在棋盘上没有地方可以下子，则该玩家对手可以连下。双方都没有棋子可以下时棋局结束，以棋子数目来计算胜负，棋子多的一方获胜。

在棋盘还没有下满时，如果一方的棋子已经被对方吃光，则棋局也结束。将对手棋子吃光的一方获胜。现行的计分方法规定是：双方分先下偶数局数的棋(如 4 局)，胜 1 分，负 0 分，和 0.5 分，分数多的取胜。假如分数一样，就以棋子数目来计算胜负。那么棋子数目怎么判断呢？如果双方将棋盘下满，当然好判断。如果棋局结束，黑棋有 34 个子，白棋有 30 个子，那么黑棋胜 34-30=4 个子。但是如果棋盘没下满呢，如黑棋有 34 个子，白棋有 27 个子。此时有两种计分方法：日本的计分法是，剩余的空格全部给胜方，那么黑棋胜 34+3-27=10 个子。欧洲的计分法是，剩余的空格双方各得一半，那么黑棋胜(34+1.5)-(27+1.5)=7 个子。

1.3 黑白棋游戏程序的现状与发展

1.3.1 黑白棋游戏程序的现状

在 90 年代中期互联网开始普及，冒起了一些大型黑白棋游戏网站。

原先黑白棋在国内集中在 3 个游戏对战平台——中游、联众和边锋。黑白棋的发展实际上是伴随着中国网络的发展。三大平台鼎盛时期也是中国黑白棋的鼎盛时期。期间高手辈出，新人不断。现在国内最强的一批高手都是那个时期成长起来的。

最近，某聊天软件游戏推出了黑白棋游戏。基于该聊天软件用户的基数庞大，

在线下黑白棋的玩家数量达到了前所未有的地步。原先消失的高手们也纷纷出现。中国黑白棋出现了欣欣向荣的景象。

1.3.2 黑白棋游戏程序的发展

随着计算机科学技术的不断发展，黑白棋游戏程序的发展出现了两个不同的分支：

以追求更强棋力为目标的黑白棋游戏程序，这类程序不断在计算机棋力方面追求更新更强的突破，通过不断的设计，大量实战对局的存储和精益求精的算法，一个又一个棋力超群的黑白棋游戏程序应运而生。最早的文曲星上的苹果棋棋力十分低下，初学者在几局过后便可轻松获胜。而现在的很多黑白棋游戏软件的计算机棋力已经可以与专业棋手一较高下，甚至有的游戏软件可以横扫大多数人类棋手。这是黑白棋其中一个发展方向，即在计算机棋力方面寻求突破。

以追求更普及的游戏推广为目标的黑白棋游戏程序，这类程序大多数不加入计算机棋手，而是以推广游戏为目的，这类游戏程序一般来说以单机人人对战和网络人人对战为主，方便广大黑白棋爱好者在网上进行进一步的操作与娱乐。这类黑白棋游戏程序虽然多数没有计算机棋手，但大部分都自带网络功能来方便运行和联机。目前此类程序由于应用面广，上手难度低，更符合受众需求而在整个黑白棋发展中起了不可估量的作用。

本论文所针对的黑白棋游戏软件即为第二类可联网的黑白棋游戏软件。

1.4 论文的组织结构

本文从黑白棋游戏的背景和发展呢出发，逐步讨论黑白棋游戏软件的设计与实现。

第二章叙述了对黑白棋游戏的需求，从不同方面阐述了用户需要的是怎样的一个游戏软件。第三章论述了黑白棋游戏软件的设计方案。第四章讨论了黑白棋游戏软件的具体实现，和运行效果。第五章是对已完成工作的总结和对未来工作的展望。

第二章　　黑白棋游戏软件的需求分析

本章阐述了黑白棋软件（Othello）的用户需求，分别从用例分析、功能需求、使用场景、运行环境等方面描述系统应满足的要求。通过本章，读者可以对黑白棋游戏有一个初步的认识，并可以清楚黑白棋游戏软件是一个什么样的系统。

2.1　用例需求

用例是程序提供的功能块，换句话来说用例演示了人们如何使用程序。通过用例观察程序，能够将程序实现与程序目标分开，有助于了解最重要的部分——满足用户要求和期望，而不会沉浸于实现细节。通过用例，用户可以看到程序提供的功能，从而深入开展项目工作。

所以根据用户的需求，黑白棋游戏软件(Othello）的用户分为两大类：

1.单机用户，即只在同一台计算机上进行人人对战的使用者。

2.网络用户，即需要用 ip 进行连接，在网络上进行对战的使用者。

其具体用例分析如图 2-1 所示。

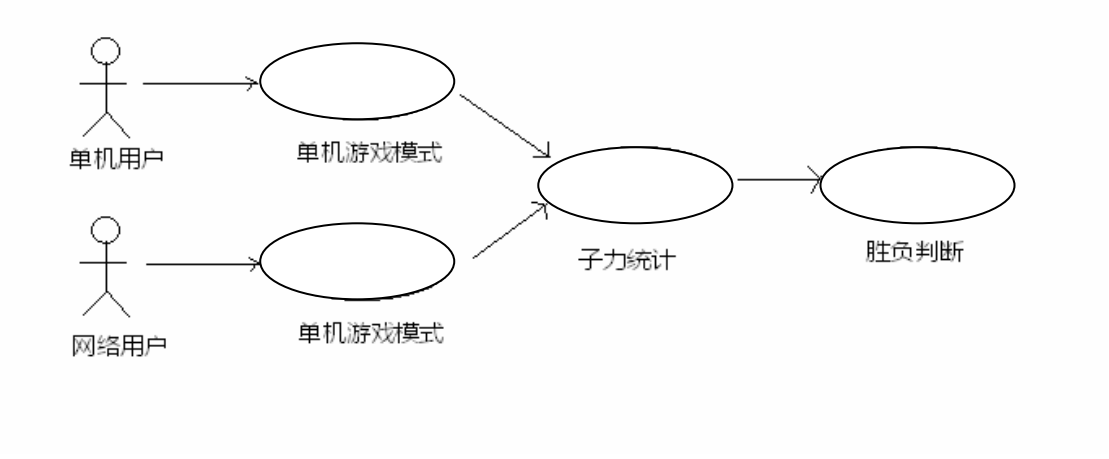

图 2-1　黑白棋游戏软件用例

2.2　功能需求

整个黑白棋游戏软件需要实现的主要功能描述如下：

1) 单机对战：游戏双方可以在同一台计算机上进行双人对战游戏。
2) 网络对战： 游戏双方可以通过查找 IP 的方式在不同计算机上通过联网来进行网络对战。
3) 胜负判定：系统会根据游戏规则在游戏结束时对游戏的获胜方以及失败方进行判定。

4) 子力统计：无论是单机对战还是网络对战，系统会对双方即时的子力进行统计并显示在软件的特定单元内。

2.3 使用场景需求

整个黑白棋游戏软件完成后所满足的使用场景应如图 2-2 所示

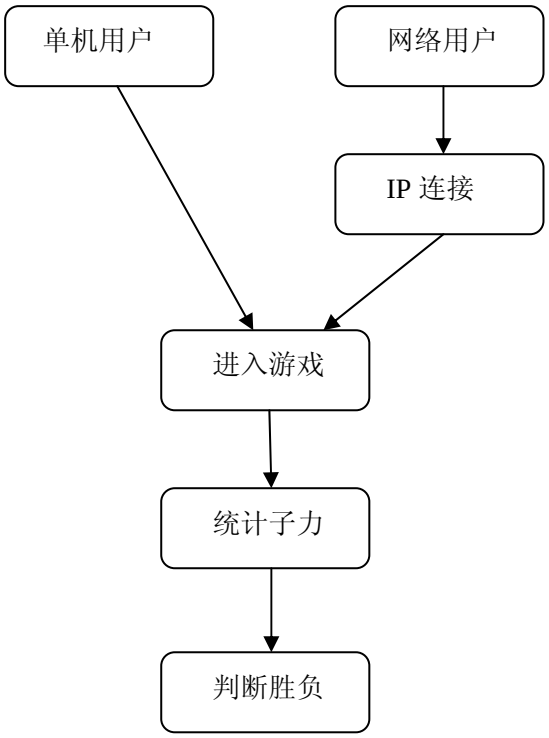

图 2-2 黑白棋游戏软件使用场景

使用场景应满足两类用户的需求，这两类分别为单机对战人员，网络对战人员。

1) 单机对战人员：

单机对战人员是本软件使用的一种特定的目标用户人群，他们需要在本地运行这个游戏并在本地上进行单机对战，他们的主要用途是不联网的游戏运行，因此单机对战人员的使用步骤按照需求应如下：

a) 首先启动游戏。
b) 在游戏模式中选择单机并进入游戏。
c) 享受游戏。
d) 退出游戏。

2) 网络对战人员：

网络对战人员是本软件使用的一种特定的目标用户人群，他们需要在网络上

联机这个游戏并进行网络对战，他们的主要用途是联网的游戏运行，因此网络对战人员的使用步骤按照需求应如下：

a) 首先启动游戏。
b) 主机在游戏模式中选择联网并创建游戏，加入者在游戏模式中选择联网并输入主机的 IP 地址加入游戏。
c) 选择要调用的模型，为下一步的数据处理和计算做准备。（可选操作）。
d) 退出游戏。

2.4 软件硬件需求

根据需求分析，黑白棋游戏软件采用的软硬件设施如下。

运行系统: 安装了.Net Framework3.5（.Net Framework 是微软的几个开发团队一起努力发展的成果，是一个可以用来快速开发、部署网站服务及应用程序的开发平台。这个架构是两个项目的结果，第一个项目的目的是用来改善 windows 作业平台上的程序开发；第二个项目则是制作一个以发展服务软件为目标的开发平台[6]。）的 WINDOWS 系统。

运行硬件环境：普通 PC 机

开发系统：WINDOWS 7

开发环境：Microsoft Visual Studio 2010

开发语言：C#

第三章　　黑白棋游戏软件的设计实现

3.1 游戏的整体架构

整个程序的设计理念基于简洁明快，力争用最简单的代码，最清晰的界面，最朴实的操作手法来让整个程序得到最多数黑白棋爱好者的认可与使用。此黑白棋的制作并不是面向小众，而是能让尽可能多的玩家使用本棋去娱乐。首先从游戏的直观感觉上来说一下设计方案。整体上程序是一个左侧有二维图右侧为各种信息的界面，这样设计直观大气。棋盘设置二维数组来表示自身，数组双下标表示棋盘上不同的下棋点，数组元素的值代表棋格上的状态，共有三种情况，分别是 0 未下棋，1 代表黑方，2 代表白方。程序需要接收棋手操作，有位置单击左键下棋，无位置自动轮到对手下棋。当单击左键时也就是棋手落子时，先判断能否放子，也就是说已经有子位置不能下棋，然后再判断该位置能否形成有效赤子，如果条件满足则在该位置落子，落子时按如下步骤进行，先调用定位函数，将棋盘的相应位置画上圆形代表函数，再调用染色函数，将对手的棋子变成自己颜色的棋子，然后根据吃掉对手棋子的个数，将一旁的子力对比图改变，再将数组中的相应元素赋值，标志该位置已经落子，最后将下棋的权限交给另一名玩家。

3.2 单机游戏部分

黑白棋游戏使用的棋盘为八乘八，在游戏过程中这共计六十四个棋格会不断的在无棋子到有棋子的情况下变化，我们通过使用一个具有八乘八共六十四个元素的二维数组 Chess[8][8]以此来表示棋盘，更通过此数组来记录棋盘的状态。我们知道数组元素共有 3 种情况，所以我们设置了 3 个不同的值：1 表示黑色棋子，2 表示白色棋子，0 表示此格无棋子。先检查二维数组这样才能开始绘制棋盘上的棋子，如果检验的结果元素是 0，那么我们不进行棋子的绘制，如果检验的结果元素是 1，那么一个黑色棋子将被绘制，如果检验的结果元素是 1，那么一个白色棋子将被绘制。之后我们对整个数组中的 1，2 和 0 进行统计，这样我们可以把结果传送给记比分的部分，将 1 和 2 表现在比分统计上，即为具体的双方子力对比。为了方便棋子坐标绘制和鼠标定位，两个一维数组 x[8]和 y[8]被创建以表示每个棋格的左上角坐标[1]。

我们之后在整个界面的左边按照设定好的距离横竖各画九条线，这样便形成了一个八乘八的棋盘界面。但是游戏窗口大小的变化时棋盘也应该能随之扩大和缩小，这里就有一个很明显的问题我们不能以绝对坐标来绘制棋盘，我们需要根据窗口的不同状态来动态决定棋盘的大小，绘制棋盘时不能直接指定绝对坐标，计算。棋盘动态坐标的算法如下：

```
private void Pict_Paint(object sender, System.Windows.Forms.PaintEventArgs e)
{
    this.DrawIt(e.Graphics);
}

private void DrawIt(Graphics Graph)
{
    this.DrawGrid(Graph);
    this.DrawPoint(Graph);
}

private void DrawPoint(Graphics Graph)
{
    for (int i=0;i<GridCnt;i++)
    {
        for (int j=0;j<GridCnt;j++)
        {
            if (Pos[i,j]==1)       // Draw Black One
                Graph.DrawImage(Img1,i*Wid+(Wid-PictA.Width)/2+
                    Space,j*Hei+(Hei-PictA.Height)/2+Space);
            else if (Pos[i,j]==2)// Draw White One
                Graph.DrawImage(Img2,i*Wid+(Wid-PictA.Width)/2+
                    Space,j*Hei+(Hei-PictA.Height)/2+Space);
         }
    }
}

private void DrawGrid(Graphics dc)
{
    Pen BPen = new Pen(Color.Black, 1);
    for (int i=0;i<=GridCnt;i++)
    {
        // Draw Vertical Line.
        dc.DrawLine(BPen,i* Wid+Space,Space,i* Wid+Space,PictHeight+Space);
    }
    for (int i=0;i<=GridCnt;i++)
    {
        // Draw Horizontal Line.
        dc.DrawLine(BPen,Space,i* Hei+Space,PictWidth+Space,i* Hei+Space);
    }
}

private void ClearArrow()
{
```

```
        for(int i=0;i<=7;i++)
        {
                Arrow[i]=false;
        }
    }

    private bool CheckDownHere(int PosX,int PosY,bool IsBlack)
        {
                bool Rtn=false;
                if ((PosX<0)|(PosX>=GridCnt)|(PosY<0)|(PosY>=GridCnt))
                  {
                        return Rtn;
                  }
    int unSelf=0;
    int Self=0;
    if (IsBlack)
        {
                Self=1;
                unSelf=2;
        }
    else
        {
                Self=2;
                unSelf=1;
        }
```

所以当玩家在自己想下棋的地方点了一下鼠标之后，按照游戏玩法，这个棋格如果是空的，我们就在棋格中画一个棋子[2]。在程序中，画一个黑色的实心圆表示黑色棋子，画一个白色的实心圆表示白色棋子。通过调用 mousePressEvent()函数，可以把鼠标点击位置的坐标保存下来，然后和棋盘上的坐标进行比较，也就是和 x[8],y[8]比较就可以得知并记录哪个棋格被鼠标点击了。然后就可以以该棋格中心点为中心，沿着该棋格绘制一个黑色的圆或绘制一个白色的圆。但是注意的是，绘制这个棋子不是在相应的棋盘上绘制，因为整个棋盘是动态的，先根据记录的棋格位置，鼠标所点击的，修改之前创立的二维数组的值为 1 或 2，以此来标记棋格，然后调用绘图函数 paintEvent()[3]，由这个函数 paintEvent()以二维数组 Chess 的值为基础，将全部棋盘重新进行绘制，这时，在之前棋盘状态的情况下，鼠标的点击位置就会放置相应绘制的圆形，这就达到了目的，也就是通过鼠标点击下棋。

黑白棋的核心算法是由其游戏规则所直接决定的，最重要的就是对当前棋格周围 8 个格的情况进行判断。在棋盘的某个棋格被鼠标所点击时，要在鼠标点击位置周围 8 个方向分别进行判定，这 8 个方向的判断过程基本一致，仅以向上方

判断为例[4]，并假设现在该黑方下棋，有以下几种情况：

1、鼠标点击位置有棋子

2、鼠标点击位置上方是黑棋；

3、鼠标点击位置上方是白棋，之后无黑棋

4、如果鼠标点击位置上方是白棋，之后有黑棋。

判断完成之后，按照同样的游戏规则，依次判断四周方向，判断出可以翻转的棋子，就将相应格子里的棋子颜色翻转。当然周围 8 格都没子的情况下不能设置棋子。所谓颜色翻转，在程序中要做的是把棋格相应的二维数组元素值由 2 改为 1，然后按照重新绘制棋子，这样就可以把棋子的颜色变化。向上方判断的黑白棋算法如下所示[5]，其它方向的判断是类似的：

```
if (PosX-1>=0)
                {
                    if (Pos[PosX-1,PosY]==unSelf)
                    {
                        for (int i=PosX-2;(i<GridCnt)&(i>=0);i--)
                        {
                            if (Pos[i,PosY]==Self)
                            {
                                Arrow[0]=true;
                                Rtn=true;
                                break;
                            }
                            else if(Pos[i,PosY]==0)
                                break;
                        }
                    }
                }
                if (PosX+1<GridCnt)
                {
                    if (Pos[PosX+1,PosY]==unSelf)
                    {
                        for (int i=PosX+2;(i<GridCnt)&(i>=0);i++)
                        {
                            if (Pos[i,PosY]==Self)
                            {
                                Arrow[1]=true;
                                Rtn=true;
                                break;
                            }
                            else if(Pos[i,PosY]==0)
                                break;
```

```
            }
        }
    }
```

比分模块需要提供传递的槽，该槽的作用是把下棋双方的棋子数显示在两个模拟部件上。而每次点完鼠标之后，需要向下棋方模块传递，以实时监控比赛双方子力进程。

整个黑白棋游戏的核心是棋盘，绘制出了棋盘，做出鼠标点击操作和黑白棋核心算法，由该模块触发比分模块的变化，即棋盘模块传递给比分模块[6]，该信号可携带表示双方棋子个数的参数。

3.3 网络游戏部分

根据需求分析中提出的要求,本项目设计开发的网络部分为客户端整合的需要实现以下功能：

（1）、设计实现黑白棋游戏服务器端。所有客户端程序通过 IP 连接[7]来和其它客户端进行通信和游戏。

（2）、实现客户端程序。客户端程序能和服务器进行通信[8]，实现基本的下棋算法，棋子计数功能。

要具体实现中，程序采用 C#语言编程方式[9]实现，这样可以增加服务器的通用性和可移植性。客户端部分采用图形库实现，客户端实现了二个界面，一是登陆界面，二是下棋界面。整个游戏的整体架构基于以上两个界面实现。

此部分主要目的是实现连接功能[10]，之后跳转到 4.2 中介绍的单机界面来实现网络对战。

```
private void StartListen(){
    //Please Waitting...
            this.PlsWait=true;
            Skt=SvrLsn.AcceptTcpClient();
            Strm = Skt.GetStream();

    // tell client user color.
            if (MeIsBlack)
                this.SendMsg("isb:false");
            else
                this.SendMsg("isb:true");

            Thread.Sleep(100);
            this.SendMsg("iam:"+MyName);
    // play game is ready.
```

```
            this.PlsWait=false;
            this.Start.Enabled=true;
            this.Setup.Enabled=true;
            this.GetData();
        }
```

3.4 游戏模式选择的界面

在界面设计的过程中，注重了直观与美观，使得用户在使用过程中能迅速找寻到自己需要的功能。在图 3-1 中，有两个选项可以选择。当用户单击“在本机上交互游戏”的单选框后，便进入了单机双人游戏模式[11]，执黑棋的玩家和执白棋的玩家可以在双方棋子对应的姓名输入框之后输入双方姓名，以在之后的游戏界面中来注明双方身份，双方姓名不可一致。当双方姓名输入完毕后，单击开始键便可以进入游戏运行界面。

图 3-1　此为单机模式选项框

在图 3-2 中，当用户单击“通过网络联网游戏”的单选框后，便进入了网络双人游戏模式，默认选项卡是主机选项卡。主机玩家需要在姓名输入框中输入自己的姓名，并在其后的阵营选项中选择，单击黑棋图标则执黑先行，单击白棋图标则执白后手。完成以上操作后单击创建键，程序会默认创建一个以主机 IP 地址为网络地址[12]，以主机选择阵营为主机操作阵营，以加入者阵营为剩余操作阵营的黑白棋游戏，并等待加入者输入信息连接。此时加入者单击加入选项卡，在连接输入框内输入需要连接的主机 IP[13]，确认无误后单击连接键，双方便会连接至以创建好的游戏并参加游戏。

图 3-2　此为联机加入模式选项框

3.5 游戏界面

在图 4-3 中为双方的游戏界面，在此界面中，双方可在界面左侧的红棕色棋盘上进行游戏，游戏方法为当轮到己方走棋时，移动鼠标单击目标棋盘格，程序会自动在目标棋盘格生成一枚与当前下棋方颜色一致的棋子，并对棋盘形式进行判定，是否有一方胜出，若未有获胜方，则将棋盘上处于翻转状态的棋子进行翻转，全部完成后则进入下位棋手走棋状态。界面右方从上到下分别是该哪个棋手走棋，界面上棋子总个数，总空格，黑棋玩家名字，黑棋个数，白棋玩家名字和白棋个数，底下的三个功能按钮，开始，是重新开始游戏，设置，是跳回角色选择界面，退出，是退出程序。

图 3-3　此为游戏过程界面

当棋盘上布满棋子的时候游戏结束，系统会对胜负进行判定并出现提示框来通知双方谁是获胜者。

图 3-4　此为游戏结果界面

第四章　　总结与展望

黑白棋游戏软件(Othello)对于整个黑白棋在国内外的发展是非常重要的。一种棋类如果能在网络上被接受，认可并被推广，无疑是对于这种棋类的影响力很有帮助。黑白棋游戏软件能否被大多数人所接受很大程度上决定了黑白棋在未来的发展理念。

本论文主要讨论了黑白棋游戏软件的制作方式方法以及黑白棋的很多相关内容，目的是制作一款应用面较广，受众较多的黑白棋游戏软件。

在编写论文的最初阶段确定了一些本黑白棋游戏软件必须提供并且需求明确的功能。例如，可以进行人人单机对战（即双人可以在同一台电脑上进行交互式游戏）,可以进行网络上的联线对战[14]（通过 IP 查询并定位对方位置从而在两台不同的电脑上进行联网对战）。界面采用传统直观的黑白棋界面，通过简洁的界面来让初学者更进一步上手此游戏。初步根据个人水平决定制作人人对战的单机版黑白棋，之后又加入了互联网对战模式以提高本棋的需求和应用层面。界面设计采用棋盘在左，即时信息在右的简洁风格，符合大多数人的审美需求以及使用习惯。

此后，在几次思考以及更多的需求分析又确定了整个游戏软件的工作细节，包括：模式设定的界面的布局；单机对战与联网对战的区别；联网游戏的设计方式等。最后决定通过 C#进行编译，通过 C#编写程序使得源代码易于读懂，制作简洁且能够被需要的人员进行进一步拓展和改编，大大提高了本黑白棋游戏程序的应用范围。同时，随着程序相关细节的敲定，也确定了整个游戏的最终界面布局，以及每个按钮的位置，以达到最合适，最方便的使用效果。

在程序几次试运行后，整个游戏也改进并添加了一些功能。比如说在创立主机的过程中可以选择是先手还是后手，这样就避免了之前总是主机执黑先行，加入者总是执白后手的局面，使游戏设计更为人性化。同时游戏界面也提供了单独的模式选择界面和游戏界面，能够更加直观的为用户提供进一步的使用帮助。模式选择界面也分为上下两部分，将单机部分与网络部分分开布局，明确了不同模式的区别，使用户能够更加便利的选择自己需要的模式。

目前整个黑白棋游戏程序已经完成了黑白棋的基本操作并且提供了双重模式选择，能够很好的为广大黑白棋爱好者提供服务。但是目前部分原计划中的拓展功能未能搭建起来，例如复盘功能，保存功能等等。这些功能需要对 C#和 Visual Studio 进行进一步的分析和研究[15]。经过不断的改进与完善，我相信这个黑白棋游戏软件一定会成为一个高性能，易于使用的软件，为黑白棋游戏在网络上的推广，为更多的人使用到黑白棋软件，为其他黑白棋游戏爱好者相互切磋沟通提供了一个良好的平台。

整个黑白棋游戏程序的设计理念本着简洁，实用的原则，从界面到程序都尽

量做到精简和方便，希望以后能够有黑白棋爱好者切实使用这款软件来进行对弈。

参考文献

[1] 丹尼索利斯.2008 C# 图解教程[M]. 北京：人民邮电出版社, 2009. 1-20.

[2] Brian Rose. Othello:A Minute to Learn...A Life time to Master[M].Anjar Co., 2004 23-25

[3] Allis.Searching for Solutions in Games and Artificial Intelligence[J]. PhD thesis, University of Limburg, Maastricht, 1994. 45-57

[4] 古田邦彦. 奥赛乐技巧速成[M].天津：天津科技出版社，1989.1-1.67-69

[5] 蔡世玉. 基于 VC 平台的黑白棋开发技术[J]. 中国矿业大学计算机科学与技术学院.1-2

[6] 荣钦科技. Visual C++ 游戏设计[M].北京：科海电子出版社，2003.1.45-47

[7] 何健辉.游戏软件设计与开发大揭密[M].北京:人民邮电出版社,2000.20-24

[8] 杜秀全, 程家兴. 博弈算法黑白棋中的应用[J]. 计算机技术与发展. 2007.67-70

[9] 蔡自兴, 徐光佑. 人工智能及其应用(第三版)[M]. 清华大学出版社, 2004.16-22

[10] Rose Bron. 黑白棋指南[M]. 2005.34-37

[11] 迈克尔阿比实, 图形程序开发人员指南（前导工作室译）[M].机械工业出版社，1998.70-72

[12] 安德鲁特纳贝姆, 计算机网络（第三版，熊桂喜等译）[M].清华大学出版社，1998.123-156

[13] 杨文龙，姚淑珍，吴云，软件工程[M].电子工业出版社，1997.234-235

[14] 袁鹏飞. 中文版 SQL Server2000 数据库系统管理[M].北京：人民邮电出版社, 2001.120-123

[15] 杨芙清.软件复用及相关技术.计算机科学[J].1999,26(5):1-4

致　　谢

本论文的完成最需要感谢的人就是我的导师——张坤龙老师，从毕业设计开始后张老师就每周定时定期的对我进行论文进度和细节方面的指导，因为我在编程水平方面的技术不是很好，张老师一直悉心帮助我解决在编译过程中的问题，在我完成毕业设计的过程中给予了非常大的帮助。感谢张老师在整个过程中的耐心指导与细致引导，才能使本次毕业设计的最终完善成为可能。