

基于 VC 平台的黑白棋开发技术

中国矿业大学(徐州)计算机科学与技术学院 蔡世玉 郭 姝

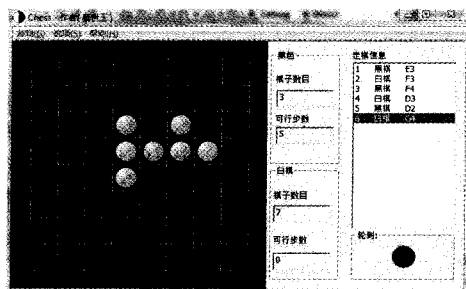
[摘 要]黑白棋作为一款经典的棋类博弈游戏,经久不衰,具有很强的趣味性,深受广大游戏爱好者和编程人员的喜爱。本文对其主要技术的实现方法进行了探讨。

[关键词]黑白棋 棋类游戏 VC

1. 引言

黑白棋,又叫 Othello,奥赛罗,苹果棋,翻转棋。游戏通过相互翻转对方的棋子,最后以棋盘上谁的棋子多来判断胜负。它的游戏规则简单,因此上手很容易,但是它的变化又非常复杂。有一种说法是,只需要几分钟学会它,却需要一生的时间去精通它。可以说,它有很强的可玩性。本文以 VC 为开发平台,论述黑白棋游戏的主要实现技术。

2. 界面的显示(见下图)



黑白棋的界面显示图

2.1 棋盘的实现

棋盘是通过在基于 CFormView 的 CChessView 上加载位图实现的。位图资源存放在 Chess\Chess\res\ 里面。在 CChessView 类里面的实现函数是 Draw_Othello()。在界面的右下方还有一个小面板,名称是“轮到:”,这个是用来提示当前轮到哪一方下棋了,这个是直接在面板上画出来的,函数是 Draw_Color(),没有加载位图。同时,为了使主对话框在最小化之后或是移出了系统桌面能够恢复到原来的图像,在 CChessView 里面还增加了 OnPaint()函数,可以响应如是最小化之类的信息,重画整个界面。

2.2 主界面各控件的实现

三个 Group Box,用来更好地区分各控件的位置。

四个 Edit Control,属性为只读,用来显示双方的棋子数和可行步数。

四个 Static Text,用来说明各个 Edit Control 的功能。

一个 List Box,属性为只读,用来显示双方已走的棋的位置信息,类似棋谱。

用来显示信息的是四个 Edit Control 和 List Box。分别为它们添加 DDX 控件变量,分别是 b_discs, b_moves, w_discs, b_moves, lbx。在游戏过程中通过这些变量来控制这些控件的行为。

2.3 “游戏开始选项”对话框的实现

这个对话框就是点击游戏(G)按钮下的新棋局(N)后出现的选择游戏类型和棋子颜色的对话框。主要有六个 Radio Button 控件,通过点击它来选择要玩的游戏和棋子颜色(先后手),选择人人对战时没有不用选择先后手,程序默认黑色为先手。并且为这个对话框架生成一个类, CInitializeDlg。

// 响应“新棋局”按键函数

```
void CChessView::OnFileNew()
```

```
{  
    // TODO: Add your command handler code here
```

```
    CInitializeDlg dlg;
```

```
    dlg.DoModal(); // 调用“游戏开始选项”对话框
```

```
    if (OkButton == 1) // 当“确认”按钮按下时
```

```
    {  
        OkButton = 0;
```

```
        Initialize(); // 初始化游戏,画出初始棋盘,输出双方初始
```

```
信息
```

```
    }
```

```
}
```

2.4 菜单命令的实现

主要有三个主菜单和各自的子菜单:

游戏(G):新棋局(N),载入棋局(L),保存棋局(S),退出(X)。

选项(S):悔棋(U),帮助(H),关于 Chess(A)。

每个子菜单都是通过 CChessView 下添加事件处理来实现其功能的。

3. 棋的实现类(Othello_rj)

为了跟主界面交互增加了一些函数。每个棋都有一个接口函数: int Main(),用来接收鼠标响应处理过后传来的坐标。接收到坐标后再给每个棋的成员变量赋值。如:当前的步数: Bushu;棋盘数组 board[i][j]的每个值,来保存棋盘的信息。在人机对战类中还有相应的算法来决定要下的棋子的坐标值,然后再传送到界面上。

4. 两个对话框和棋的实现类之间的信息交互

程序开始时将初始化整个主界面,各个控件和棋盘区都为原始状态,棋盘区为空。点击游戏(G)后将执行 CChessView::OnFileNew()函数,调用“游戏开始选项”对话框,当确定“确定”按钮被按下后,在 CInitializeDlg 类中将初始化三个全局变量: ChessType(棋的种类), human, computer (Human == 1 时为先手),之后调用 CChessView::Initialize(),初始化各个变量,并在主界面上画出初始棋盘,等待玩家的鼠标响应。

响应鼠标左击信息后,在 CChessView::OnLButtonDown 函数里面,通过获得棋盘位图的大小和位置,来确定鼠标点击的是棋盘上的哪个格子,当坐标合法时,将调用 CChessView::Othello_rj_Do(),这个是在 CChessView 下相对应的棋处理函数,然后调用 Othello_rj::int Main()把得到的坐标值传到棋的实现类里面处理,之后将能得到处理后的棋盘信息,并通过 CChessView::Draw_Othello_rj()重画棋盘,调用 CChessView::Othello_rj_Message()改变各控件的值和状态,以实现输出信息。

5. 游戏设计中主要算法

5.1 悔棋

当下了第一个棋后,“悔棋”按钮变可见,可以悔棋。对四个不同的棋,主要有堆栈来保存每一步棋的信息,同样以黑白棋(人机对战)为例:堆栈为: std::stack<Othello_rj> st;堆栈存放的是整个类,在每一步合法的下棋后,都会把当前的整个类压入栈中: st.push(othello)。当点击“悔棋”按钮是将调用 CChessView::OnEditUndo(),对于人机对战,这时要连续弹出两个类: st.pop(),并且调用 Othello_rj::Prev()来把这个栈的顶元素,即一个完整的 Othello_rj 类的各个变量值赋给当前的类,并且重画棋盘,重新输出信息,并删除 List Box 中的最后两个数值。

5.2 保存

保存一个盘棋的实质就是保存这个棋的堆栈。当要点击“保存棋局”按钮时,调用 CChessView::OnFileSave(),此时会打开另一个窗口,让你输入一个要保存文件的文件名,后缀名程序自动设定为.CHESS。当点击“保存”按钮后,将调用 CChessView::Serialize()。这里用的是序列化保存。先保存三个全局变量: ChessType, human, computer。之后将会调用 Othello_rj::Serialize()。这时再保存一个 int 变量 StackSize,为要保存的栈的元素个数,这在下面的“载入”功能时有用。之后是最关键的一步,保存整个当前栈 st。增加一个辅助栈 std::stack<Othello_rj> st;因为要保存整个栈需要遍历整个栈,所以需要有一个栈来保存弹出来的元素,当序列化保存完整个栈后,再把 st 里面保存的元素重新压入到当前栈 st,之后即可以继续当前游戏。而为了操作方便,还要把当前的状态再序列化保存,并且保存了另外一个数据: record[i][j],这个数据保存的是当前这盘棋所走的棋谱信息,这个数据为的是在载入时可以把载入的棋所走的信息直接输出在 List Box 上,所以只用保存一次。

5.3 载入

“载入”是在任意时刻都允许的。点击“载入”按钮后,调用 CChessView::OnFileOpen(),这时会出现一个窗口,让你选择要载入的文件,文件的后缀名必须是.CHESS,点击“打开”按钮后,同样调用的是 CChessView::Serialize(),以黑白棋(人机对战)为例,先载入三个全局变量和 StackSize,之后清空当前栈,现在要做的就是把序列化保存的数据拿出来再生成一个当前栈,调用 Othello_rj::Serialize2(),这时将会按顺序把保存进去的数据再拿出来赋给当前类 othello_rj, (下转第 199 页)

```

B=[1 0 0;-1 1 0;0 1 0;0 1 -1;0 0 1]; %误差方程系数阵
L=[5.835 3.782 9.640 7.384 2.270]; %观测高差值
l=[0 -23 0 14 0]; %常数阵
p=[2.9 3.7 2.5 3.3 4.0];
P=diag(p) %权阵
x0=[243.318 247.123 239.753]; %参数初始值
Nbb=B'*P*B; %法方程系数
w=B'*P*l;
disp('解算法方程参数改正数')
x=inv(Nbb)*w
disp('求观测值改正数')
v=B*x-l
disp('求观测值平差值')
Lv=L'+v/1000
disp('单位权估值')
d=(v'*P*v)/2;
d0=d*0.5 %单位权中误差
disp('end of adjustment')

```

在教学中鼓励学生自己独立开发设计程序,编写专业相关程序的过程是一个深化测量平差基础知识的过程。用程序来解决一个平差问题时,例如上面的间接平差,首先要弄清楚间接平差的步骤,然后才能编写程序,对数据进行求解。

(3)实践教学

从传统意义上说,测量平差实践环节较少,大量的习题虽然可以加深学生的应用能力,但在某种程度上来说,还不足以让学生更透彻地理解测量平差的实质。为了让学生更透彻地理解测量平差的实质、巩固学习知识并与其它课程融合,在教学过程中,将测量平差的课程设计作为学生动手能力培养的重要环节。在平差教学课程设计中,教师给出设计目的、设计任务和要求,所选例题和习题的类型尽量选接近现实的工程实例,要求学生对于一个具体的控制网,采用不同的平差法在计算机上完成权阵的建立,误差方程和法方程的建立过程等。学生通过动手编写计算机程序,并对实例进行编程计算,不仅将枯燥的理论很好地理解,而且极大地提高了学生学习的兴趣,增强了学生的解决实际问题的能力。

(上接第197页) 然后先把这个类压入到辅助栈st中。因为是堆栈,所以保存进去时的数据跟现在拿出来数据的顺序刚好相反,所以当把所有数据都拿出来保存在st里面时,再把st里面的数据压入到st,即当前栈中,这样,就把以前保存的棋的栈完整地复原到当前栈中了,这时再调用一次 othello_rj::Serialize2(),把剩下的数据赋给当前类,回到以前保存时那个棋的状态,重画棋盘,输出信息到控件,再调用 CChessView::LoadOthello_rj_Lbx(),把棋谱信息输出到 List Box 上。

5.4 人机对战

◆棋子数贪心策略

电脑搜索棋盘每一个可以下的位置,在合法的格子上下子后计算一下双方的棋子数。在所有合法的下法下寻找一个使得自己(这里指电脑方)比对方棋子多的位置。当然这个“多”有可能是负数,因为在一些情况可能不管怎么下一步自己的棋子是无论如何也不可能比对方多,不过可以把差距拉到最小。

```

void Othello_rj::ZhiNeng_h0() { /* 寻找能够吃子最多的一步 */
int i,j,sum=-100,t;
Save(); /* 先把当前的棋盘保存起来 */
Bubian=1;
for(i=1;i<=8;i++) {
for(j=1;j<=8;j++) {
Load();
if (board[i][j]==0) {
x=i;
y=j;
Bubian=1; /* 判断这个棋子是否有效 */
Youxiao=0;
Run();
if (Youxiao==1) /* 有效时改变棋盘 */
board[i][j]=ch;
Bubian=0;
Run();
Count(); /* 统计这招棋下完之后电脑跟用户棋子数的差数 */
if (Shunxu==1) t=Secondsum-Firstsum;

```

设计结束后,对学生进行考核,为了培养学生的创造性思维和综合素质,改变以前,即大部分要求学生死记硬背,只要会做题就能合格考试方式指导。教师应根据学生在学习中所表现的意识和品质,完成实际任务的数量和质量,设计内容的先进性与合理性,独立工作能力的大小、计算编程能力、软件应用能力、理论知识掌握等内容多方面综合考核学生,对学生的技术设计进行考评记分。考评成绩分为优、良、中、及格和不及格五个档次。2007、2008 级测绘工程专业测量平差课程教学效果如表 1 所示。

表 1 2007、2008 级测量平差课程教学效果对比

年级	评定等级				
	优秀	良好	中等	及格	不及格
2007	3	20	12	20	5
2008	8	23	20	8	1

3. 结语

通过上述例子可以看出,在具体的教学过程,通过将 MATLAB 软件和测量平差课程结合,不仅改变了传统的教学过程中偏重理论鄙视实践,学生实际能力普遍偏低现状,更主要的是应用 MATLAB 进行平差计算,可以非常清晰地体现平差计算的基本原理,计算思路清晰,非常适合对测量平差的学习和掌握,极大地提高了学生学习的综合素质和教学的质量。

参考文献

- [1]张智星.MATLAB 程序设计与应用[M].北京:清华大学出版社,1993
- [2]高宁,高彩云.MATLAB 在测绘领域中的应用[J].平顶山学院学报,2008(1):28-31
- [3]武汉测绘科技大学测量平差教研室.测量平差基础.北京:测绘出版社,1996
- [4]赵宝锋.误差理论与测量平差基础课程教学研究与实践[J].矿山测量,2006(4):73-75

```

if (Shunxu==0) t=Firstsum-Secondsum;
if (t>sum) { /* 比原来的大的话就保存这时的坐标值 */
x1=i;
y1=j;
sum=t;
}
}
}
Load(); /* 恢复原来的棋盘 */
}

```

◆限制行动力策略

限制行动力策略就是要让对方这些可以下的格子最少,这样子对方就有可能被迫下到一些很不好的格子。具体实现是,首先保存当前棋盘信息,再依次搜索每一个可以下的位置,在当前可以下的格子下子后,检测对方可以下的格子数,搜索结束后,选择一个使对方可以下的格子数最少的下法。

6. 总结

博弈游戏是人工智能的重要体现之一,棋类游戏是一种完全知识博弈类游戏,参与双方在任何时候都完全清楚每一个棋子是否存在,位于何处,二者完全处于平等的角色。因此对于计算机,如何在棋类游戏中利用搜索算法,提高效率,形成不同的走法,评估局面,成为其重要的因素,也是体现其人工智能技术的重要因素之一。

参考文献

- [1]钦科技.Visual C++ 游戏设计.北京:科海电子出版社,2003.1
- [2]何健辉.游戏软件设计与开发大揭秘.北京:人民邮电出版社,2000
- [3]坂本千寻.Visual C++ 专业游戏程序设计.北京:中国铁道出版社,2004
- [4]肖永亮等.计算机游戏程序设计.北京:电子工业出版社,2005
- [5]Tom Meigs.顶级游戏设计.北京:电子工业出版社,2004