ISSN 1009-3044
Computer Knowledge and Technology 电脑知识与技术
Vol.18, No.20, July 2022

E-mail: eduf@dnzs.net.cn
http://www.dnzs.net.cn
Tel:+86-551-65690963 65690964

基于MFC的黑白棋的设计与实现

钟 闯

(湖北大学 计算机与信息工程学院,湖北 武汉 430062)

摘要:在计算机高速发展的今天,计算机图形学变得日益重要,以计算机图形学为基础的游戏开发变得越来越火热。但单纯地使用图形学知识开发设计难免烦琐,所以该文借由微软基础类库(MFC)和图形设备接口(GDI)来设计并实现了一个黑白棋游戏。通过对游戏的设计与开发,达到应用MFC和GDI的目的。

关键词:微软基础类库;图形设备接口;黑白棋;游戏开发;游戏设计

中图分类号:TP311 **文献标识码:**A

文章编号:1009-3044(2022)20-0096-03

开放科学(资源服务) 标识码(OSID):

1 概述

黑白棋,又名翻转棋。黑白棋由于规则简单,易于上手,逐渐在各个国家流行起来。游戏通过翻转对方的棋子成为自己的棋子,最后以棋盘上谁的棋子多来判断胜负。这个棋的规则虽然简单易懂,上手容易,但是它的变化又非常复杂。黑白棋的棋盘是64(8×8)个格子,初始会有2黑2白放在棋盘的正中央。黑白棋的游戏规则是:1)不管黑方或白方,下的棋子必须要造成棋子的翻转,如果两个相同的棋子相邻中间没有可以翻转的棋子则一方不可以下子。2)如果一方的下一步无法造成棋子的翻转就将下棋权移交给另一方。3)当棋盘下满或者两方都没棋子可以转换的时候,游戏结束,统计棋盘上双方的棋子,谁剩余棋子多谁胜利。由于黑白棋的特殊性,就导致下子的时候不能像围棋或五子棋那样随意下子到棋盘边缘没有棋子的位置,这就给算法的实现增加了难度。

MFC是微软公司创建的C++的基础类库,目的是减少Win32编程开发的难度。这个类库将Windows编程中的API封装成C++类,这些封装是轻量级封装,与原来的C语言的Windows编程速度相比并没损失多少,所以MFC在出来的时候,企业对其很是追捧,但过了几年后C#中的Winform、C++图形框架Qt等对MFC的冲击,又由于MFC本身的缺陷导致MFC逐渐没落,但想弄清楚Windows中的消息和绘制机制就必须学习MFC。

GDI是图形设备接口英文的缩写,它存在的目的是减少程序员绘图的工作量。原来的程序员在程序中绘图时,必须考虑底层硬件的驱动,这无疑加大了程序员的工作量和思考时间。GDI出现后让程序员的工作成本大大减少,让程序员不用再考虑底层的硬件驱动,直接使用简便的图形接口来操作复杂的底层绘制[1]。

虽然现在MFC和GDI很少人直接使用,但它们在计算机技术发展的历史上有着举足轻重的基石作用,很多桌面开发技术都是由这两者拓展而来,所以本文将用MFC和GDI来实现黑白棋游戏。

2 黑白棋游戏功能分析

游戏功能分为3个模块:

1)绘制模块,主要包括初始化绘制棋盘,棋子和游戏提示信息。

2)游戏判断模块,主要包括对双方下子位置判断,判断游戏是否结束,判断一方能否下子。

3)游戏控制模块,主要包括键盘交互,人机对战,统计双方棋子数量,重置游戏。

其功能图如图1。

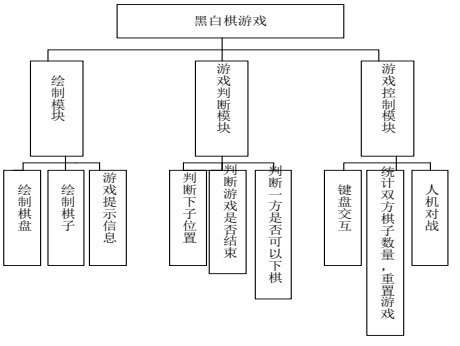

图1 黑白棋功能图

3 黑白棋游戏各个模块的总体概述

绘制模块。功能主要包括初始化棋盘大小,用GDI的绘制函数绘制棋子,游戏提示信息等。首先在800×680大小的窗口中将窗口640×640的区域分割成64个格子,并将格子的坐标信息放在一个二维数组m_rcSquares里方便使用绘制函数进行绘制,窗口底部的空位用于显示游戏的基本信息。接着用一个二维数组表示对应格子里是否下了棋子,1代表黑棋,2代表白棋,0则表示没有下棋,再用一个二维数组表示其余的空格能否下棋子,能则为1,不能则为0,初始棋盘中央是有2黑2白。在棋盘底端显示游戏的基本信息,用来显示当前棋盘黑白双方棋子的数量等信息。

收稿日期:2022-03-29

作者简介:钟闯(1995—),男,湖北武汉人,硕士在读,研究方向为计算机技术。

游戏判断模块。此模块主要是为了判断双方下子位置的合法性，众所周知，黑白棋一旦下子就必须造成棋子的转换。所以首先需要判断下子位置能不能造成棋子的转换。其次，还要判断双方是否可以下棋，一旦双方的下一步都无法造成棋子的转换，那么游戏结束。如果只是一方无法下棋而另一方可以，那么要交换下棋方。最后，棋盘的64个格子放满了棋子则游戏结束。

游戏控制模块。这个模块主要的功能是监视键盘和鼠标的输入，根据用户的输入来进行相应的反应，在MFC中使用消息映射来实现对用户键盘或者鼠标输入的响应[2]。游戏中，鼠标左键是黑棋下子，鼠标右键是白棋下子(一开始是黑棋先下)，鼠标中键是开启人机模式，在进入人机模式后，人机是白子，如果想推出人机模式就点击鼠标右键退出人机模式。F1键是重置游戏，F2键是游戏规则说明。该程序中的人机对战主要是采用的是贪心算法，所谓贪心算法，就是每一次的行动都是选择最优解[3]，在这个游戏中就是人机每次下棋的时候选择尽可能多地转换棋子的位置，而不考虑之后的结果，直到游戏结束。

4 黑白棋游戏核心内容详述

1) 程序结构

该游戏程序并没有使用MFC经典的文档视图结构，而是使用了比较简便的方法。通过继承MFC中的CWinApp类和CWnd类来简化程序。在继承CWinApp类后，需要在实现类中重写它的虚函数InitInstance[4]。虚函数是实现多态的必要条件，可以通过父类的指针来找到具体实现虚函数的子类，如果子类没有实现，则只调用父类的定义虚函数[5]。程序内，类和类中函数概述如表1。

表1　程序中核心函数和类介绍

类	概 述
CMyApp	继承于MFC中的CWinApp类，实现了InitInstance方法
CMyMainWindow	继承于MFC中的CWnd类
函 数	概 述
void DrawBoard(CDC* pDc)	用于画黑色的棋盘线
void DrawBlackChess(CDC*pDc, int iLine,int Vertical)	画指定位置的黑色棋子，画白色棋子的函数跟这个函数参数是一样的只不过名字不同
BOOL ChessRevarsal(int iLine,int iVertical)	判断下棋位置是否合法
BOOL FindRightChess(int iLine, int iVertical,int ChessType)	找到右边与当前所下棋子颜色一样最近棋子

2) 初始化和绘制功能详述

在窗口类CMyMainWindow的构造函数中，首先加入了初始化棋盘的代码，这个时候只能先构建棋盘在窗口中位置坐标并将其放到二维数组中，因为此时窗口还没有建立所以不能在此类的构造函数中直接加入绘制图形函数，同时将棋盘中间的4个格子设置为已有棋子放置，保存在二维数组中。之后还是在构造函数加入窗口的初始化代码，设置该窗口的大小为800×600。最后在类CMainWindow中的消息响应函数OnPaint中添加对棋盘线，4个初始棋子(2黑2白，位置在棋盘中央)的绘制。

除了上面初始化绘制的时候不需要判断，其余每次绘制都需要通过二维数组m_iChess来判断对应棋盘位置有没有棋子。有时需要棋盘位置上有棋子的时候才能绘制，因为这个时候要通过绘制来实现棋子颜色反转的效果，有时也需要棋盘位置上没有棋子的时候才能绘制，因为这个时候是下子，必须保证下子位置没有棋子。

该程序通过成员变量m_iBlackChess和m_iWhiteChess来统计棋盘上黑棋和白棋的数量，通过成员变量m_iChessTotal统计棋子总数。这3个成员变量通过函数void DisplayGameInfo()来绘制到窗口下方，来实时显示当前游戏信息。

3) 游戏判断功能详述

这个功能可以说是游戏核心中的核心了，有了这个功能黑白棋的规则才能付诸实现。首先当一方下子的时候，要判断他下子的8个方向(上，下，左，右，上左，下左，上右，下右)有没有和下棋方一样的棋子同时中间必须夹着另一方的棋子，在这种情况下才可以下棋，否则就无法下棋。当然不是所有的格子都需要判断8个方向，比如棋盘的4个角只用判断3个方向(以右上角为例，只用判断左，下左，下3个方向)，除了4个角，棋盘边缘的格子也只用判断3个方向(以最右一列为例，只用判断左，上左，下左3个方向)。表1中的函数BOOL FindRightChess(int iLine, int iVertical,int ChessType)函数中的代码是查找落子位置右边方向的相同棋子，其他位置的查找函数也是一样的。其次，前一棋子落完子后，需要立即判断当前棋盘空位能不能再下接下来一方的棋子(比如前一方是白子，接下来就要判断黑子有没有地方下了)，如果没有合法的位置，跳过当期下棋方，回到前一下棋方继续下棋，这个功能所对应的函数就是表1中的BOOL ChessRevarsal(int iLine,int iVertical)函数。除此之外已经下过棋子的位置不能再下子，通过类成员变量m_iChess，这个变量是个8×8的bool类型的二维数组，用于存储64个棋盘位置有没有棋子，所以每次下子前需先遍历该二维数组，然后再判断下棋位置是否合法。当两方的下一步都不能造成棋子转换或者棋盘下满了的时候游戏结束。通过成员变量m_iChessTotal统计棋子总数，如果棋子总数达到了64个，则游戏结束。

4) 游戏控制功能

这个功能主要是管理键盘交互，人机对战和游戏基本信息的显示。在MFC中键盘和鼠标所响应的函数是已经规定好的，只需要在窗口类中重写对应的方法就可以了。在该程序中，需重写键盘F1，F2按键被按下的消息事件，鼠标左键，鼠标右键，鼠标中键被按下的事件。注意，在鼠标点击的事件响应中要加入对鼠标点击时坐标的判断，有可能鼠标点击的时候刚好点击到了棋盘线上，通过MFC中CRect类中的PtInRect方法就可以进行判断。在之前程序的初始化中，已经将棋盘中每个格子的坐标放在了二维数组中，这个数组类型是CRect所以通过遍历数组然后调用该方法。

通过按下鼠标中键进入人机对战，在人机对战中，人机使用的是白色棋子，玩家使用的是黑色棋子，人机下子所采用的算法是比较经典的贪心算法，就是在玩家下完棋子后查找转换棋子最多的位置，所用的函数是将上面的BOOL FindRightChess(int iLine, int iVertical,int ChessType)等7个函数进行改造就可以了。上面的FindRightChess函数是在查找的时候同时进行棋子的翻转，而对于人机来说则是先去找能转换棋子最多的位置，这个过程不需要进行翻转，等找到了合适的位置，再调用绘制函数，除了某些细节不一样，其余代码都是和人下棋的代码是一样的。重置函数ResetGame()，这个函数在棋盘满了，双方无法下子和玩家按下F1键的时候启用。在函数内部将二维数组初始化到游戏开始的状态然后调用MFC的窗口更新函数进

行窗口图形刷新,当窗口刷新时程序会自动调用绘制函数。

5 程序测试与结果

该程序基本实现了黑白棋游戏的基本功能,运行界面如图2所示。

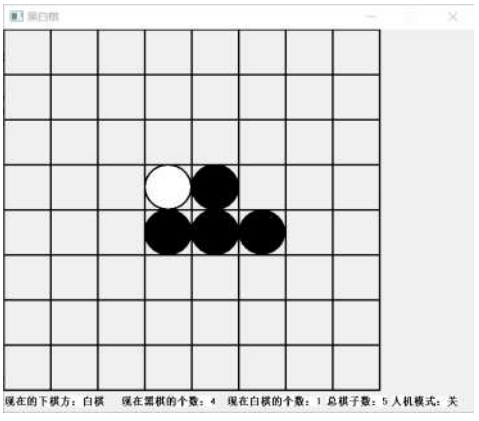

图2 黑白棋运行界面

可以看到现在是由白棋下,之前的黑棋已经造成了翻转,底部的信息提示现在棋盘上的情况和人机对战是否开启。当玩家觉得自己下得不好时,可以按下键盘上的F1键重置游戏,但前提是你要和你的对手商量好。当你找不到对手跟你一起下时可以按下鼠标中键进行人机对战。当游戏双方不清楚游戏规则时可以按下键盘上的F2键了解游戏规则。

6 结束语

该程序通过继承MFC提供的类和实现MFC消息响应函数,设计并实现了一个黑白棋游戏。程序的总体结构并不复杂,但内容却集合了数据结构、算法、图形绘制等复杂内容。选择一个好的数据结构能提升数据组织效率,选择一个好的算法能提升程序运行速度。除此之外,拥有良好的代码风格能够使程序言简意赅同时在程序出BUG的时候能快速找到位置进而提升工作时的效率。

参考文献:

[1] Petzold C.Windows程序设计:珍藏版[M].方敏,译.5版.北京:清华大学出版社,2010.
[2] 侯俊杰.深入浅出MFC:使用Visual C++5.0 & MFC 4.2[M].2版.武汉:华中科技大学出版社,2001.
[3] Drozdek A.C++数据结构与算法[M].徐丹,吴伟敏,译.北京:清华大学出版社,2014.
[4] 帕罗赛斯.MFC Windows程序设计[M].北京博彦科技发展有限责任公司,译.北京:清华大学出版社,2007.
[5] 谭浩强.C++程序设计[M].4版.北京:清华大学出版社,2021.

【通联编辑:谢媛媛】

(上接第93页)

对课堂视频中的授课教师进行目标定位,然后再进行姿态识别、生动度计算。目标定位使用计算机MTCNN+InsightFace人脸识别方法,框出视频中教师的人脸。姿态识别使用OpenPose模型,检测出视频帧中人体姿态关键点的位置。生动度计算使用相邻若干帧的教师姿态关键点的欧式距离。最后得出课堂中教师生动度变化曲线,为智慧课堂的其他部分提供教师授课状态时间流信息。

4.3 黑板知识点检测

黑板在课堂中扮演着至关重要的角色,该模块关注于课堂教学过程中知识点讲述时间流,设计并实现能够自动分析课堂教学过程中知识点时间流的模块。

在将计算机视觉识别技术应用到智慧课堂构建中可以通过对YOLOv3算法的适应性改变以应用于对课堂黑板的位置检测,继而将CTPN网络与DenseNet网络相结合,实现对于黑板中文字位置的检测与内容的识别,将识别结果与自定义知识库内容通过正则表达检索,分析课堂实时知识点内容。实时课堂知识点与学生当前疑惑度相结合进行分析,可以判断学生对当前知识点是否疑惑,并进行疑惑知识点的复习推荐。同时老师能够掌握学生的学习情况,搭建双方沟通桥梁,服务师生。

4.4 基于学情知识点推荐

为了最大限度地提升高校学生课堂学习效率,根据学情实时解决学生面对各个知识点产生的疑惑,需要对困惑学生及时推送相关知识点辅助资源。一个良好的模型不仅要执行速度快,而且要能推送最有效且个性化的辅助学习资源。

在将计算机视觉识别技术应用到智慧课堂构建中拟采用基于知识表示的(Knowledge Representation, KR)Surprise模型构建学习者之间、学习资源之间的相互关系,利用协同过滤算法构建推荐引擎对KR模型产生语义关系进行评分预测产生有效推荐条目,以消息队列的方式推荐给学生客户端。

5 结束语

智慧课堂面对学生可以分析其面部状况,面对老师可以分析其上课情况,面对黑板可以提取上课内容,获取课堂上教师姿态生动度、学生疑惑度、黑板知识点的所有信息,能根据学生上课情况,及时进行疑惑知识点的推送。互联网+教育融合创新不断发展,教育信息化推动质量提升形成内在动力,智慧教育和智慧课堂正在发生深刻变革与转型。

参考文献:

[1] Alger D.思科绿色数据中心建设与管理[M].北京:人民邮电出版社,2011.
[2] 钟景华,朱利伟,曹播.新一代绿色数据中心的规划与设计[M].北京:电子工业出版社,2010.
[3] 刘金月,刘华莹,时贵英."互联网+"背景下智慧课堂在"C程序设计"课程的教学法研究[J].微型电脑应用,2021,37(6):25-26,30.
[4] 彭红超,祝智庭.深度学习研究:发展脉络与瓶颈[J].现代远程教育研究,2020,32(1):41-50.
[5] 彭红超,祝智庭.面向智慧课堂的灵活深度学习设计框架研制[J].现代远程教育研究,2021,33(1):38-48.

【通联编辑:代影】