

武汉工程大学 计算机科学与工程学院

项目报告

设计名称: 《Windows 程序设计》综合项目

设计题目: 五子棋游戏

学生学号: 1005110200

专业班级: 二〇一〇级信息技术 02 班

学生姓名: 蔡淑存

学生成绩:

指导教师 (职称):

课题工作时间: 2012 年 4 月 24 日星期二

说明:

- 1、报告中的第一、二、三项由指导教师综合设计开始前填写并发给每个学生；四、五两项（中英文摘要）由学生在完成综合设计后填写。
- 2、学生成绩由指导教师根据学生的设计情况给出各项分值及总评成绩。
- 3、指导教师评语一栏由指导教师就学生在整个设计期间的平时表现、设计完成情况、报告的质量及答辩情况，给出客观、全面的评价。
- 4、所有学生必须参加综合设计的答辩环节，凡不参加答辩者，其成绩一律按不及格处理。答辩小组成员应由 2 人及以上教师组成。
- 5、报告正文字数一般应不少于 5000 字，也可由指导教师根据本门综合设计的情况另行规定。
- 6、此表格式为武汉工程大学计算机科学与工程学院提供的基本格式（适用于学院各类综合设计），各教研室可根据本门综合设计的特点及内容做适当的调整，并上报学院批准。

成绩评定表

类别	合计 分值	各项 分值	评分标准	实际 得分	合计 得分	备注
平时 表现	10	10	按时参加综合设计,无旷课、迟到、早退、违反实验室纪律等情况。			
完成 情况	30	20	按设计任务书的要求完成了全部任务,能完整演示其设计内容,符合要求。			
		10	能对其设计内容进行详细、完整的介绍,并能就指导教师提出的问题 进行正确的回答。			
报告 质量	35	10	报告文字通顺,内容翔实,论述充分、完整,立论正确,结构严谨合理;报告字数符合相关要求,工整规范,整齐划一。			
		5	课题背景介绍清楚,综述分析充分。			
		5	设计方案合理、可行,论证严谨,逻辑性强,具有说服力。			
		5	符号统一;图表完备、符合规范要求。			
		5	能对整个设计过程进行全面的总结,得出有价值的结论或结果。			
		5	参考文献数量在3篇以上,格式符合要求,在正文中正确引用。			
答辩 情况	25	10	在规定时间内能就所设计的内容进行阐述,言简意明,重点突出,论点正确,条理清晰。			
		15	在规定时间内能准确、完整、流利地回答教师所提出的问题。			
总评成绩: _____ 分 补充说明: _____ <div style="text-align: right;"> 指导教师: _____ (签字) 日 期: _____年____月____日 </div>						

答辩记录表

答辩地点: _____ 机电大楼 411 机房 _____

答辩内容记录:

答辩成绩	合计 分值	各项 分值	评分标准	实际 得分	合计 得分	备注
	25		10	在规定时间内能就所设计的内容进行阐述, 言简意明, 重点突出, 论点正确, 条理清晰。		
15			在规定时间内能准确、完整、流利地回答教师所提出的问题。			

答辩小组成员 (签字):

_____年____月____日

一、综合设计目的、条件、任务和-content要求:

1. 设计目的

《Windows 程序设计》是计算机科学与技术专业本科生的一门学科基础课程。Windows 程序以图形用户界面 (GUI) 给用户提供各种功能, 在各行各业有着广泛的应用。基于 MFC 的 Windows 程序设计是进行 Win32 程序设计的一种主流方法。

本课程主要介绍 Windows 程序设计的思想和方法, 以及 MFC 的常见应用。本课程综合应用算法与程序设计、面向对象程序设计、数据结构、操作系统、数据库系统原理等课程的知识和方法, 面向实际应用和开发, 以培养、提高学生的程序设计能力和软件工程能力为目的。通过课程教学、上机实践、综合设计等教学环节, 使学生逐步掌握 Windows 程序设计和 MFC 的常见应用, 并能在实践中熟练应用所掌握的工具和方法解决实际问题。

2. 实施方式

本课程重在训练学生的综合设计能力和项目开发应用能力, 并强调学生的上机实践能力。按照常规考试方式, 不能体现本课程“以理论为基础, 侧重实践和应用”的特点, 也不能考查学生理解本课程知识的程度以及所培养具备的 MFC 程序设计应用能力。

因此申请课程考试改革, 拟采用项目开发和课程报告相结合的方式对课程进行考查。

实施方式: 针对某一综合性较强的 Windows 应用程序项目, 学生首先

进行上机设计, 实现系统功能, 然后根据设计内容撰写课程报告, 并在规定时间递交程序和纸质报告、答辩, 最后根据程序、报告和答辩等进行课程成绩评定。

分数比例: 项目开发的系统功能和程序代码为 60% , 课程报告为 40% 。

3. 任务和要求:

针对某一综合应用项目, 完成如下任务:

- (1) 简要的需求分析
- (2) 规划系统主要功能及其结构。
- (3) 完成各项系统功能的设计。
- (4) 形成项目文档和开发报告。

4. 工具与环境:

设计工具: Visual Studio

指导教师签字: _____ 张俊 _____

_____ 2012 年 4 月 24 日 _____

二、进度安排:

第 11 周: 系统功能总体设计。

第 12~13 周: 实现系统功能, 撰写设计报告, 并按时递交。

第 13 周: 答辩, 评定成绩。

三、应收集资料及主要参考文献:

请自行收集并填写参考文献: 5-10 项

MFC 相关的书籍;

与本应用程序相关的参考文献;

[1] 任哲等编著. MFC Windows 应用程序设计(第 2 版). 北京: 清华大学出版社, 2007. 9.

四、课程综合设计摘要 (中文):

摘 要

随着人们生活水平的日益提高,游戏已经成为了人们生活中不可缺少的一种娱乐方式之一。无论是象棋还是飞行棋,再到这儿的五子棋,都已经渗透到我们的生活中,影响着我们的休闲方式。而这些娱乐方式都可以用MFC实现,而当自己用MFC实现五子棋的功能,其中的快乐将是无穷的!

下面着重介绍用MFC实现五子棋的功能。这个程序的编译环境是VC++ 6.0,根据自己所学的有关MFC的知识加以实现,而其中的工作可以分为两块:一块是用MFC的有关知识实现棋盘和棋子的绘制,这一块是比较难的也是比较难成功的,如果能把这做成功那么就成功了一半,我们开始要想设置多大的棋盘,然后再运用函数加以实现,而在绘制棋子的时候就要想一下自己的棋子的大小和颜色;另一块就是当完成了棋盘和棋子的绘制之后就得用所学的C, C++语言加以实现怎么样判定游戏的胜负,并且能实现日常下棋时所用到的功能!这方面是最重要的也是最难的,因为这是算法的核心因此我们得设计一下它的怎么加以实现,然后再运用所学的知识来加以实现,当完成了这些功能之后用户就能进行正常的游戏。

用MFC实现五子棋很好的将古今哲理的结晶和现如今科学技术完美的结合起来,能够让我们更加理解先人的智慧然后加以继承,最后就能够更加完美的发扬和传承!

关键字: MFC ; Chess; GAME OVER; Chcek

五、课程综合设计摘要 (英文):

Abstract

As people living standard rise increasingly, the game has become the people living one of the indispensable one of the types of entertainment. Whether or FeiHangQi chess, and then to the renju here, has penetrated into our life, affecting our leisure. And the entertainment can use MFC realization, and when his realize the function of renju with MFC, among them joy will be endless!

Introduced the realization of the function with MFC renju. This program compiles environment is vc + + 6.0, according to his own knowledge about MFC knowledge realization, and one of the work can be divided into two pieces of: a with MFC knowledge about the pieces to the drawing board and realize, this was a difficult it is hard to do so you have won half success, we started to set the size of the board, then use function realization, and in rendering the pieces to think about their own pieces to the color; The other piece is when completed pieces to the drawing board and then have learn C, C + + language to realize how determine the outcome of the game, and can realize daily use to play chess function. This is the most important is the hardest, because the core of the algorithm so we have to design how it how realization, and then by using the knowledge to realization, when complete these function after users can normal game.

With MFC realize renju will be a very good the crystallization of ancient and modern philosophy and now the perfect combination of science and technology, can let us better understand their wisdom and then to inherit, finally can more perfect carry forward and inheritance.

Keywords: MFC ; Chess; GAME OVER;Chcek

摘 要	
ABSTRACT	
摘 要	
ABSTRACT	
第一章 绪论及概述	
1.1 绪论	
1.1.1 课题背景	
1.1.2 目的和意义	
1.1.3 主要问题及技术要求	
1.2 概述	
1.2.1 基本理论依据	
1.2.2 主要工作内容	
第二章 设计简介及设计方案论述	
2.1 设计简介	-3-
2.1.1 流程图简介	
2.1.2 主要功能函数简介	
2.2 设计方案	-4-
2.2.1 设计方案简述	-4-
3.1 游戏规则	-5-
3.1.1 游戏规则的说明	
3.2 游戏界面和棋子的制作	-5-
3.2.1 以MFC框架程序来设计一个五子棋的界面	
3.2.2 对话框的设计	
3.3 其它功能的实现	13-
3.4 核心算法	15-
3.4.1 核心算法之一-----鼠标左键或右键所在棋格位置	
3.4.2 核心算法之二-----棋局输赢和下平的判断	
3.4.2 核心算法之三-----游戏的退出和重置的	
第四章 设计结果及分析	
4.1 完成之后的初始化界面	-20-
4.1.1 棋盘和棋子的界面	
4.2 悔棋和反悔棋界面	22-
4.3 棋盘的大小和颜色的界面	23-
总 结	
致 谢	
参考文献	
附录	

摘 要

随着人们生活水平的日益提高，游戏已经成为了人们生活中不可缺少的一种娱乐方式之一。无论是象棋还是飞行棋，再到这儿的五子棋，都已经渗透到我们的生活中，影响着我们的休闲方式。而这些娱乐方式都可以用 MFC 实现，而当自己用 MFC 实现五子棋的功能，其中的快乐将是无穷的！

下面着重介绍用 MFC 实现五子棋的功能。这个程序的编译环境是 VC++ 6.0，根据自己所学的有关 MFC 的知识加以实现，而其中的工作可以分为两块：一块是用 MFC 的有关知识实现棋盘和棋子的绘制，这一块是比较难的也是比较难成功的，如果能把这做成功那么就成功了一半，我们开始要想设置多大的棋盘，然后再运用函数加以实现，而在绘制棋子的时候就要想一下自己的棋子的大小和颜色；另一块就是当完成了棋盘和棋子的绘制之后就得用所学的 C, C++ 语言加以实现怎么样判定游戏的胜负，并且能实现日常下棋时所用到的功能！这方面是最重要的也是最难的，因为这是算法的核心因此我们得设计一下它的怎么加以实现，然后再运用所学的知识来加以实现，当完成了这些功能之后用户就能进行正常的游戏。

用 MFC 实现五子棋很好的将古今哲理的结晶和现如今科学技术完美的结合起来，能够让我们更加理解先人的智慧然后加以继承，最后就能够更加完美的发扬和传承！

关键字：MFC ; Chess; GAME OVER; Chce

Abstract

As people living standard rise increasingly, the game has become the people living one of the indispensable one of the types of entertainment. Whether or FeiHangQi chess, then to the renju here, has penetrated into our life, affecting our leisure. And these entertainment can use MFC realization, and when his realize the function of renju with among them joy will be endless!

Introduced the realization of the function with MFC renju. This program compiles environment is vc + + 6.0, according to his own knowledge about MFC knowledge realization and one of the work can be divided into two pieces of: a with MFC knowledge about the pieces to the drawing board and realize, this was a difficult it is hard to success, if you have won half success, we started to set the size of the board, then use function realization, and in rendering the pieces to think about their own pieces to the size and The other piece is when completed pieces to the drawing board and then have to learn C, + language to realize how determine the outcome of the game, and can realize daily use play chess function. This is the most important is the hardest, because this is the core algorithm so we have to design how it how realization, and then by using their knowledge realization, when complete these function after users can normal game.

With MFC realize renju will be a very good the crystallization of ancient and modern philosophy and now the perfect combination of science and technology, can let us better understand their wisdom and then to inherit, finally can more perfect carry forward and inheritance.

Keywords: MFC ; Chess; GAME OVER;Chce

第一章 绪论及概述

1.1 绪论

1.1.1 课题背景

随着计算机技术和互联网的飞速发展，游戏软件是当今世界发展最迅速，最有影响力，最有潜力与活力的领域之一。电脑游戏的出现为人们提供了新的娱乐平台，丰富了人们的业余生活。因此与游戏有关的软件深受广大青少年的喜爱。随着近几年来软件技术的不断进步、不断发展，软件游戏的操作要求越来越简单，大多数游戏普通人学几分钟就能会玩，甚至连老年人也慢慢爱上了它。

第二次世界大战以后，电子计算机技术得到了突飞猛进的发展。在众多国家中美国的计算机技术发展最为迅速。1946年2月，世界上第一台电子数字计算机(ENIAC)在美国宾夕法尼亚大学建造完成。ENIAC机的问世，表明了电子数字计算机时代的到来，具有划时代的伟大意义，是科学技术发展史上的重大里程碑。我国也不落后，在1956年研制成功国产计算机的。1958年和1959年我国先后自主研制成功国产小型和大型电子管计算机。后来出现了集成电路和大规模集成电路，使电子计算机一代一代实现更新，性能越来越高，计算速度越来越快，硬件方面的不断发展促使了软件技术也迅速发展。在美国，集中了许多计算机软件的设计人才，他们工作之余，时常喜爱编一种能与人斗智的“游戏”，以此来锻炼编程的能力。这种“游戏”花样繁多，但其特点都是利用计算机软件事先设计好的“分析”“判断”能力反过来与人较量。由于不断修改更新，使计算机的“智力”水平与人难分高低。到了90年代，游戏软件已经为世界各国的人使用，所谓游戏软件通常是指用各种程序和动画效果相结合起来的软件产品，我们称之为“游戏软件”。目前，在网络上我们经常看到的大型3D网络游戏和WEBGAME网页游戏等都是通过用3DMAX、MAYA、FLASH、等动画软件和JAVA、C++、VB、等程序语言相结合而开发出来的，所以叫游戏软件。其中BLACKISLAND和Blizzard这两个强大的游戏制作组对游戏软件的发展起到了巨大的带动作用，还有中国的洛河工作室在国内也起到了巨大作用，他们让很多很多人喜欢上了游戏软件，打开了这个市场。如今游戏软件产业已经受到了广泛的关注，它年轻有潜力而且吸引人，让人们日益丰富的生活需求得以满足。

而五子棋游戏的诞生更是吸引着不同年龄段的人群，无论男女老少都可以玩，都喜欢玩，而当前微型计算机已经是在广大人群中流行着，用电脑来下五子棋更是一种时尚。现在网络上出现了各种各样的五子棋软件，有大师级的、新手级的等等。所有这些都满足了不同人群的需要，所以当前五子棋越来越被许多人所熟悉。目前的五子棋程序的发展也非常快，而用MFC再结合高级的基础语言（C和C++）加以实现，这就很好的满足了绝大部分人能够操作的需求！

1.1.2 目的和意义

人类之所以变得聪明，是因为我们人类一直不断地再实践中思考，在思考中实践。五子棋不仅能增强思维能力，提高记忆力，而且富含哲理，有助于修身养性。五子棋既

有现代休闲的明显特征“短、平、快”，又有古典哲学的高深学问“阴阳易理”；它既有简单易学的特性，为人民群众所喜闻乐见，又有深奥的技巧和高水平的国际性比赛；它的棋文化源远流长，具有东方的神秘和西方的直观；既有“场”的概念，亦有“点”的连接。它是中西文化的交流点，是古今哲理的结晶。

并且用上上个世纪最新兴起的软件技术加以开发，例如本文中就是以 VC++6.00 为开始背景，然后用 MFC 和 C 和 C++ 为语言基础另以开发，就很好的将古今哲理的结晶和现如今科学技术完美的结合起来，能够让我们更加理解先人的智慧然后加以继承，最后就能够更加完美的发扬和传承！

1.1.3 主要问题及技术要求

本程序的主要问题就是实现能够满下五子棋所需要的各种要求。具体说来包括以下几个方面（1）棋盘大小的控制（2）棋盘的绘制（3）棋子的绘制（4）对棋子所放位置的控制（5）判断游戏输赢而这些问题中的前四条所用的的知识是 MFC 中的在关知识，然后最后一条也是最复杂的一条不仅要用到 MFC 有关的知识，而且还要用到 C 和 C++ 语言为基础，还要以 VC6.0 为编译环境来加同实现和完善！

1.2 概述

1.2.1 基本理论依据

用户 MFC 可以设计多种样式的应用程序窗口。设计时用向导完成 MFC 的窗口的控制完成窗口大小的设置，而其中用到的函数则是 PreCreateWindow 中代码选择 SDI 窗口中加放。而能够实现用户在下棋的时候能够进行控制的时候，就得用到处理鼠标的消息函数，并且能够完全的实现左右键的控制，而其中用到的函数是 WM_LBUTTONDOWN, WM_RBUTTONDOWN, WM_MBUTTONDOWN 等函数来实现鼠标的单击和双击的控制并且不断向 ClassView 加放变量和函数来实现控制。而最后里面包含的核心则是算法刚要用到高级程序语言 C 和 C++ 来加以实现，并且还要用到 MFC 中的有关知识来加以控制。

1.2.2 主要工作内容

在基于 SDI 的情况下加以控制，然后加入对话框控制棋盘的大小的响应函数，CTicTacView() 来初始化棋盘的初始值，并且能够在进行每一局游戏的时候能开始都有相同的界面。然后不断向 ClassView 加放变量和函数来实现控制例如用 DrawBoard 和 DrawChess 函数来分别绘制棋盘和棋子。

第二章 设计简介及设计方案论述

2.1 设计简介

2.1.1 流程图简介

其主要的功能流程图简介如图 2-1 所示：

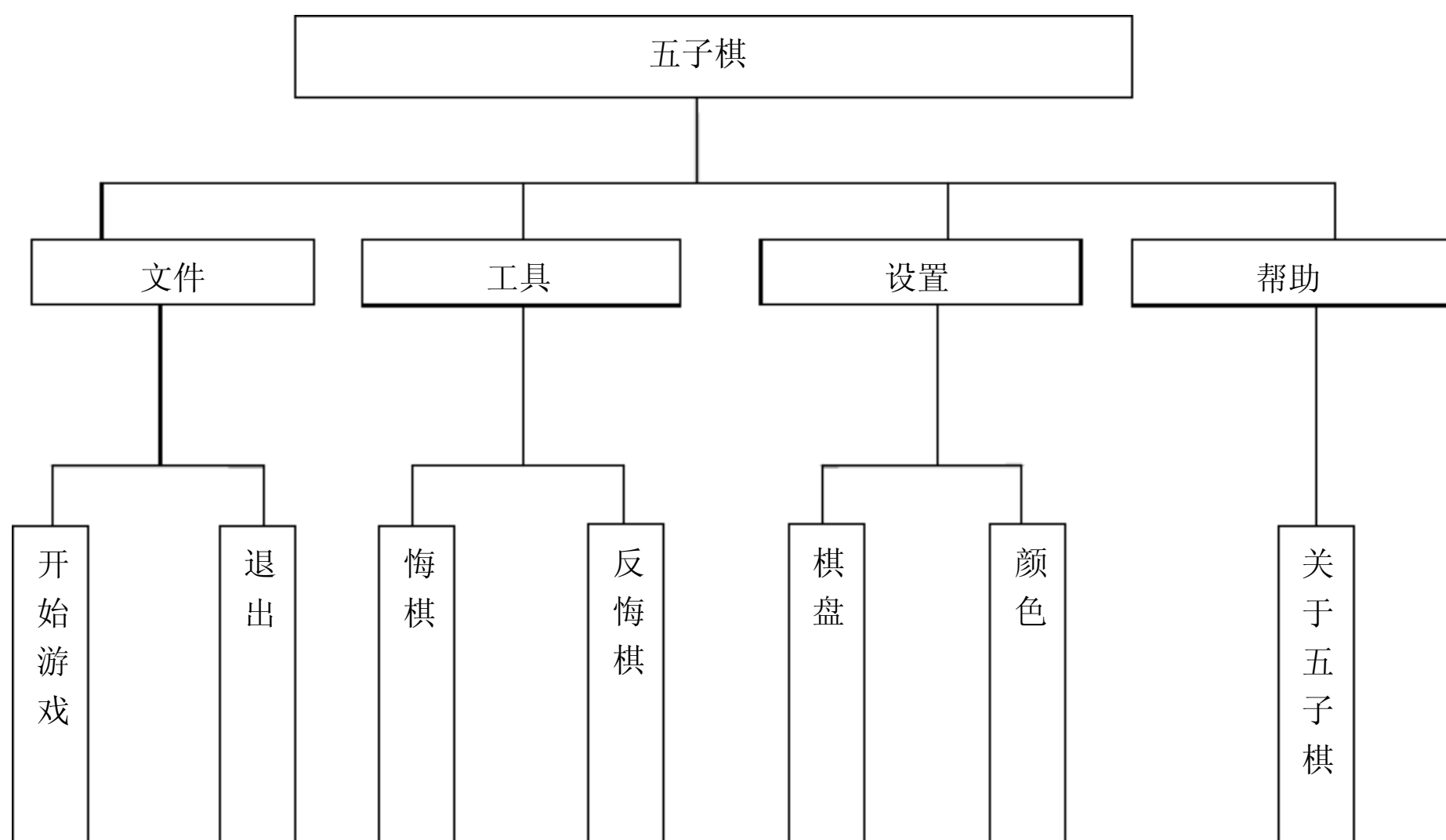


图 2-1 流程图

2.1.2 主要功能函数简介

```
BOOL PreCreateWindow(CREATESTRUCT& cs) ;
```

```
void OnDraw(CDC* pDC) ;
```

```
void CreateBoard(CDC *pDC, CRect rect, int Idx, int &HSpace, int &WSpace)
```

```

void DrawBoard(CDC *pDC, CRect rect)
BOOL OnEraseBkgnd(CDC* pDC) ;
void DrawChess(CDC *pDC, int nPos, int Chess)
Int Check(int index, int k, int chess)
Int IsWinner(void)
Int IsDraw(void)
void IsGameOver(void)
void ResetGame(void)
void OnFileExit()
void OnFileNewgame()
void OnToolRedo()
void OnToolUndo()
void OnLButtonDblClk(UINT nFlags, CPoint point)
void OnLButtonDown(UINT nFlags, CPoint point)
void OnRButtonDown(UINT nFlags, CPoint point)
void OnSetBoard()
void OnSetColor()

```

2.2 设计方案

2.2.1 设计方案简述

首先将我们所主要完成的功能列一张清单，然后根据自己所列的清单画出流程图，将每一项的子项列出来，让自己更好能够清楚自己所要完成某种任务，然后再列出自己要完成某项功能所需要的函数，最后进行源程序的写放，开始确定自己所要建的工程类型，然后再一步步的加入，最后当完成了所有功能之后再来重新补充和完善自己所写的源代码最后就是总结和运行。

第三章 详细设计

3.1 游戏规则

3.1.1 游戏规则的说明

- ①首先我们确实的是 10*10 的棋盘
- ②按黑先白后，最先在棋盘横向，纵向或斜向边成一条线的同色五个棋子的一方获胜
- ③如分不出胜负则为平局
- ④五连与禁手同时形成，判胜

3.2 游戏界面和棋子的制作

通过以前我们玩过的游戏然后再结合我们之前接触的游戏规，我们就对五子棋游戏规则与设计思路有了一个很好的理解。本节将展示如何设计简单的程序界面，并进一步美化它，以便抽象的代码化成为我们熟悉的游戏界面。

3.2.1 以 MFC 框架程序来设计一个五子棋的界面

(1)这里将使用用我们经常用到的编译工具 VC++6.0，中的 MFC 来设计游戏的主界面，用时确实为 SDI 窗口，其基本的窗口如 3-1 图所示：

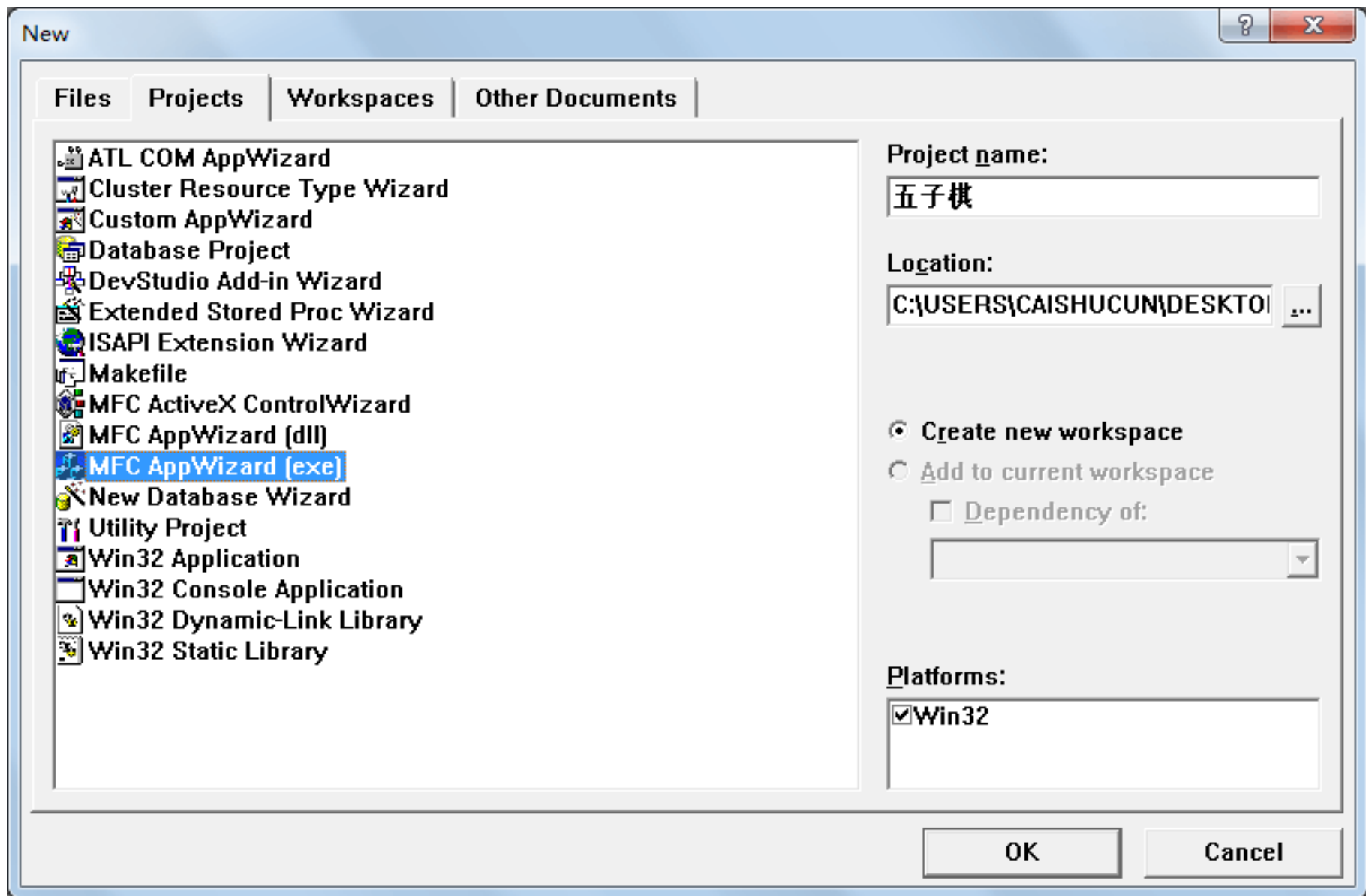


图 3-1 新建图形界面

(2) 当完成了一个工程新建的时候再选择我们需要确定的文档是单文档还是多文档还是对话框其选择的图形如图 3-2 所示:

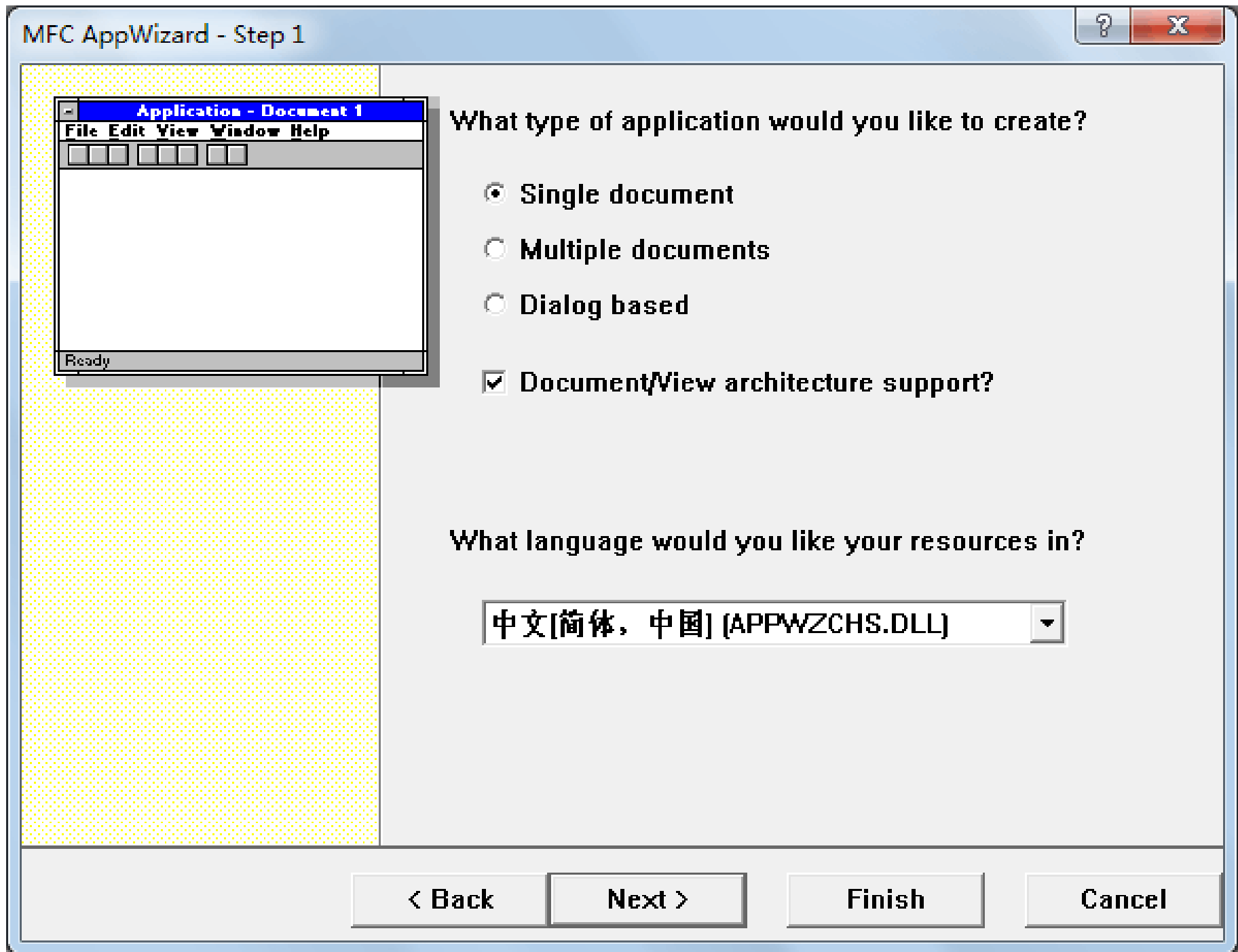


图 3-2单文档界面

(3) 然后点击完成后，就新建了一个单文档的程序然后再生成一下看自己所创建的程序是否新建正确,它的生成如图 3-3所示：

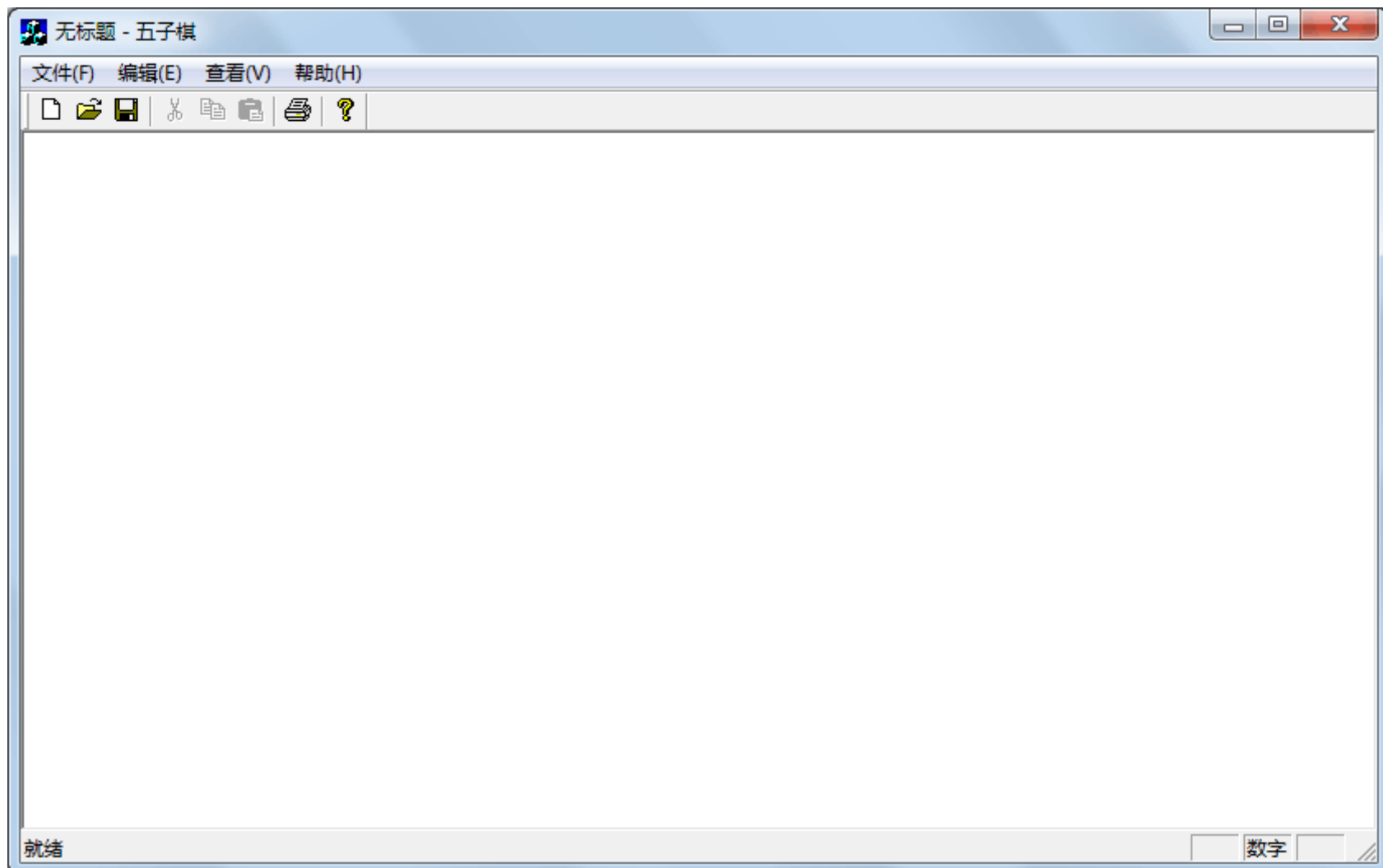


图 3-3成功创建图

3.2.2 对话框的设计

(1) 在视图资源中找到菜单的工具栏如 3-4:



图 3-4 资源视图

(2) 当找到所要找的工具之后再添加所想要的功能如图 3-5

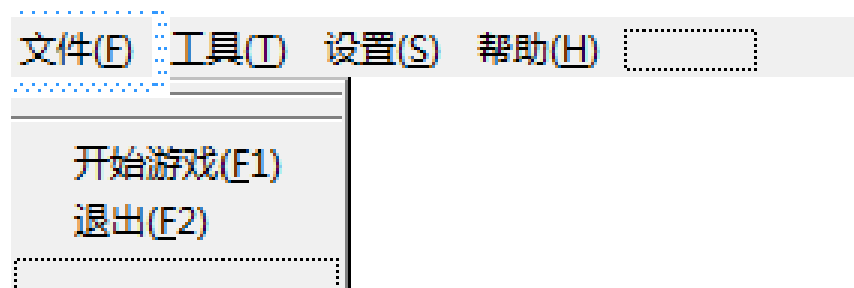


图 3-5 功能视图

然后按同样的方法向其它的菜单中加入自己所加的功

(3) 当完成了所要加的功能之后就是要修改每一项功能的 ID 如 3-6 图所示:



图 3-6 ID 显示图

然后按照同样的方法修改自己所想要实现功能的 ID 号，以便编程序的人能够清楚知道每一项功能函数。

3.2.3 五子棋中棋盘的设计意图

(1) 在 ClassView 中加放好下变量:

```

CRect *Board; //棋盘
int BoardSize; //棋盘大小
COLORREF BoardColor; //棋盘颜色
    
```

其中一个变量的加入如 3-7 图所示:



图 3-7 添加数据成员图

然后后面要加变量的时候按照同样的方法加入所要加的变量。

(2) 按照 (1) 所给的方法同样的添加成员函数 `void DrawBoard(CDC *pDC, CRect rect);` 和 `void CreateBoard(CDC *pDC, CRect rect, int Idx, int &HSpace, int &WSpace);` 来绘制棋盘，它的源代码如下：

```
void CTicTacView::DrawBoard(CDC *pDC, CRect rect画棋盘
{
    int Idx = 10000;
    int HSpace, WSpace;

    CreateBoard(pDC, rect, Idx, HSpace, WSpace);

    CPen pen(PS_SOLID, (HSpace+WSpace)/(2*Idx), RGB(0, 0, 0));
    CPen *pOldPen = pDC->SelectObject(&pen);

    for(int i = 0; i < BoardSize; i ++)
    {
        pDC->MoveTo(Board[i].right + WSpace/(2*Idx), 0);
        pDC->LineTo(Board[i].right + WSpace/(2*Idx), rect.Height());

        pDC->MoveTo(0, Board[i*BoardSize].bottom + HSpace/(2*Idx));
        pDC->LineTo(rect.Width(), Board[i*BoardSize].bottom + HSpace/(2*Idx));
    }

    pDC->SelectObject(pOldPen);
}
```

```
void CTicTacView::DrawBoard(CDC *pDC, CRect rect画棋盘
{
    int Idx = 10000;
    int HSpace, WSpace;
```

```

CreateBoard(pDC, rect, Idx, HSpace, WSpace);

CPen pen(PS_SOLID, (HSpace+WSpace)/(2*Idx), RGB(0, 0, 0));
CPen *pOldPen = pDC->SelectObject(&pen);

for(int i = 0; i < BoardSize; i ++)
{
    pDC->MoveTo(Board[i].right + WSpace/(2*Idx), 0);
    pDC->LineTo(Board[i].right + WSpace/(2*Idx), rect.Height());

    pDC->MoveTo(0, Board[i*BoardSize].bottom + HSpace/(2*Idx));
    pDC->LineTo(rect.Width(), Board[i*BoardSize].bottom + HSpace/(2*Idx));
}

pDC->SelectObject(pOldPen);
}

```

它的功能就是能够完成棋盘的大小的确定

(3) 完成了棋盘的大小的确实之后就要对它进行初始化，基其初始化的函数好下所示：

CTicTacView::CTicTacView() 初始化棋盘的初始值

```

{
    BoardColor = RGB(200, 70, 30);
    BoardSize = 10;
    UndoTimes = 5;
    Ratio = 30;
    UndoCnt = UndoTimes;
    NextChess = WHITE;

    Board = new CRect[BoardSize*BoardSize];
    Chess = new int[BoardSize*BoardSize];
    ASSERT(Board != NULL && Chess != NULL);

    for(int i = 0; i < BoardSize; i ++)
    {
        for(int j = 0; j < BoardSize; j ++)
        {
            Board[i*BoardSize+j].left = j * 120;
            Board[i*BoardSize+j].top = i * 120;
            Board[i*BoardSize+j].right = Board[i*BoardSize+j].left + 100;
            Board[i*BoardSize+j].bottom = Board[i*BoardSize+j].top + 100;

            Chess[i*BoardSize+j] = 0;
        }
    }
}

```

当完成了这之后就能得到一个棋盘开始的颜色的后面设计加上去的,刚棋盘如图 3-8 所示:

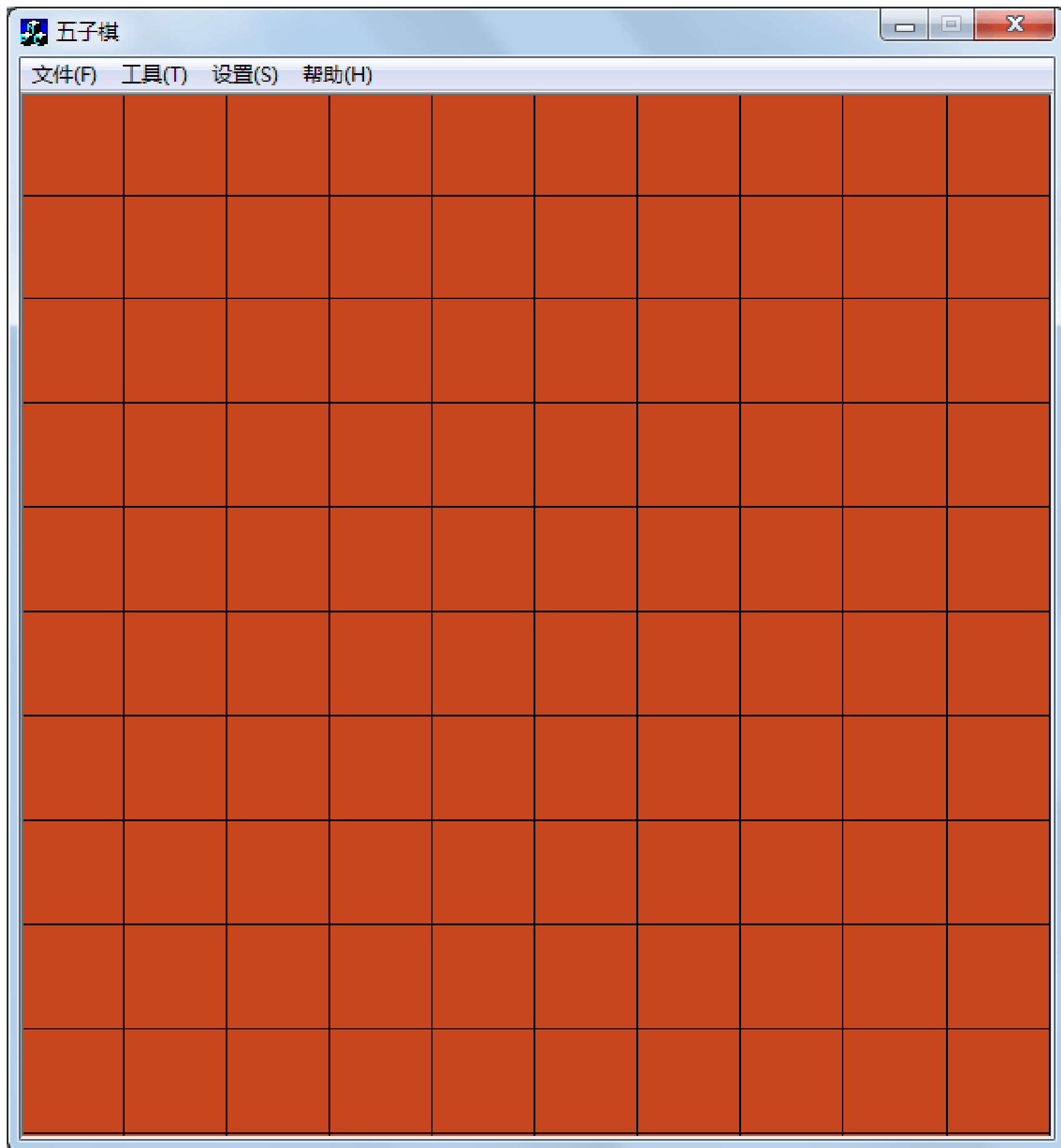


图 3-8 棋盘图

3.2.4 五子棋中棋子的设计

(1) 分别再向 ClassView 中加放如下变量, 加入变量的方法好图 3-9 所示

```
int *Chess; //每个棋格存放的棋子
int NextChess; //下一个将要下的棋子
int Ratio
```



图 3-9 添加数据成员图

(2) 按照 (1) 所给的方法同样的添加成员函数 void DrawChess(CDC *pDC, int nPos, int Chess);并在 OnDraw 中加入适当的变量来绘制棋子, 它的源代码如下所示:

```
void CTicTacView::DrawChess(CDC *pDC, int nPos, int Chess)
{
    CRect rect = Board[nPos];
    CBrush brush;
    int Width = rect.Width()/8;

    if(Chess == WHITE)
    {
        brush.CreateSolidBrush( RGB(255, 255, 255));
    }
    else
    {
        brush.CreateSolidBrush( RGB(0, 0, 0));
    }

    pDC->BeginPath();
    pDC->Ellipse(rect.left+Width, rect.top+Width, rect.right-Width, rect.bottom-Width);
    pDC->EndPath();
    pDC->SelectObject(&brush);
    pDC->FillPath();
}

void CTicTacView::OnDraw(CDC* pDC)
{
    CTicTacDoc* pDoc = GetDocument();
    ASSERT_V ALID(pDoc);

    CRect rect;
    GetClientRect(rect);

    DrawBoard(pDC, rect);
}
```



```

for(int i = 0; i < BoardSize * BoardSize; i ++)
{
    if(Chess[i])
    {
        DrawChess(pDC, i, Chess[i]);
    }
}

```

再适当的加入其它的函数来更新棋盘这样的话我们运行程序的时候就能得一个我们所想要的棋盘和棋子了它的图形如图3-10所示：

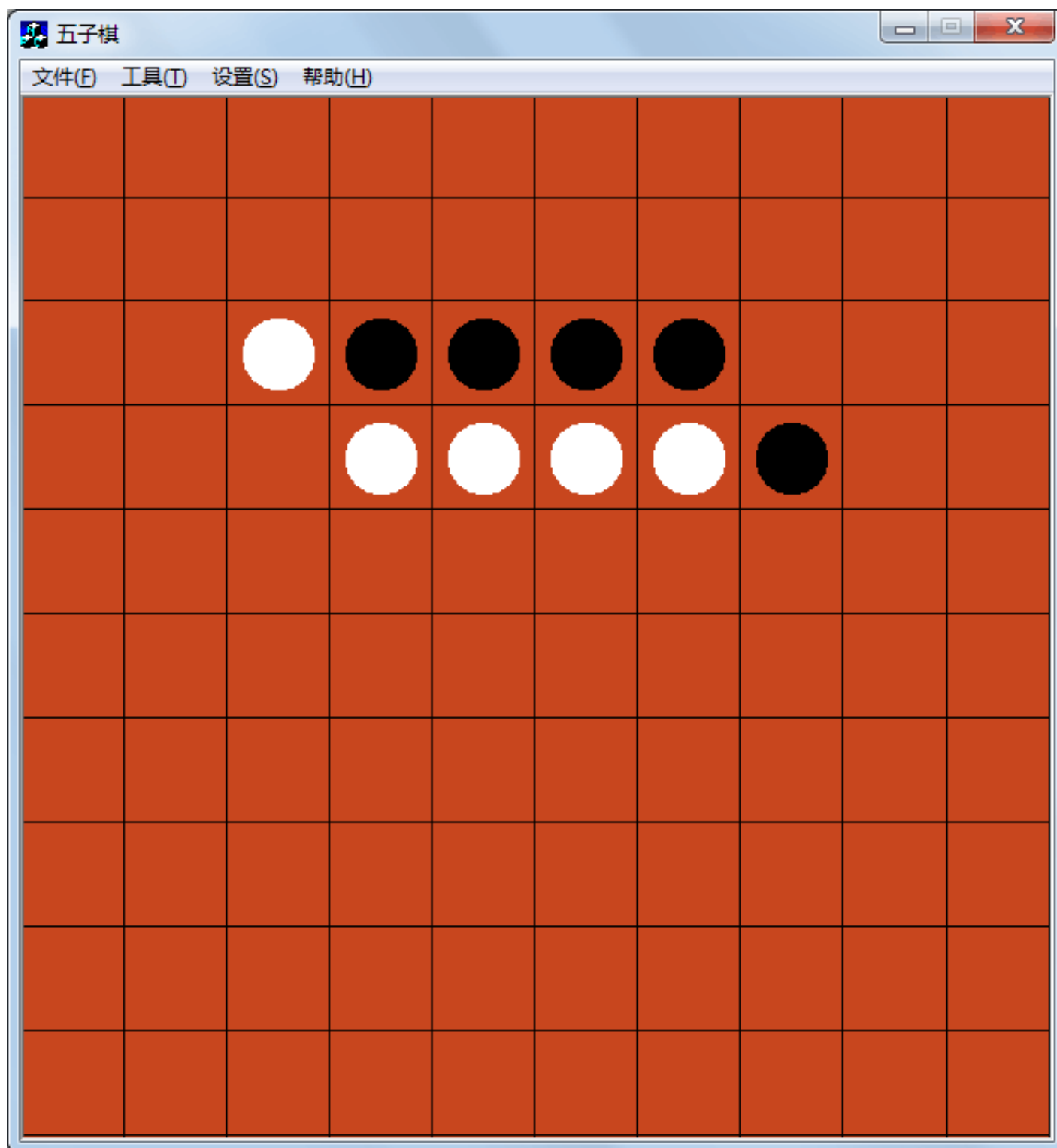


图 3-10 下棋时图

3.3 其它功能的实现

(1) 游戏开始和退出的功能的实现，它的源代码如下所示：

```

void CTicTacView::OnFileExit 退出的判断
{
    AfxGetMainWnd()->SendMessage(WM_CLOSE);
}

```

以上内容仅为本文档的试下载部分，为可阅读页数的一半内容。如要下载或阅读全文，请访问：<https://d.book118.com/766045034035011001>