

高等学校计算机专业规划教材

计算机图形学

(VC++实现) (第2版)



于万波 于 硕 编著

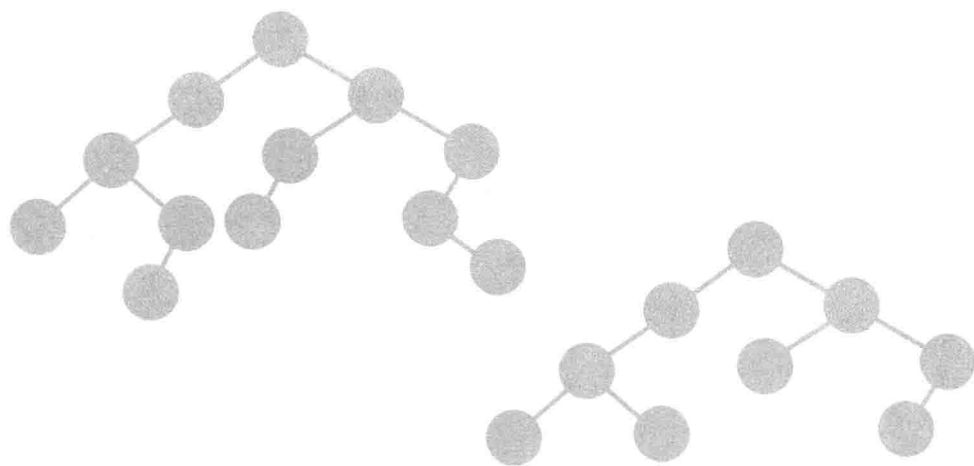
清华大学出版社

高等学校计算机专业规划教材

计算机图形学

(VC++实现) (第2版)

于万波 于 硕 编著



清华大学出版社
北京

内 容 简 介

本书是计算机图形学入门教程,以 VC++ 与 OpenGL 为工具讲解计算机图形学以及动画制作的基本知识。

第 1 章通过一些有趣的实例直观地展示了一些简单的计算机图形绘制与动画制作方法,同时让读者了解、熟悉并逐步掌握 VC++ 绘图相关类及函数的使用;第 2 章讲解二维直线与曲线绘制方法以及区域填充的基本内容;第 3 章学习三维图形投影、消隐等内容;第 4 章通过一些典型实例介绍 OpenGL;第 5、6 章讲解样条曲面、几何造型与光照模型;第 7、8、9 章研究分析建模与动画实例。

本书适合作为计算机科学与技术、软件工程、信息与计算科学、数字媒体技术、机械与建筑设计等专业的计算机图形学教材,也可供对计算机图形动画制作感兴趣的程序设计人员参考。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

计算机图形学:VC++ 实现/于万波,于硕编著. —2 版. —北京:清华大学出版社,2017
(高等学校计算机专业规划教材)
ISBN 978-7-302-46124-1

I. ①计… II. ①于… ②于… III. ①计算机图形学—高等学校—教材 IV. ①TP391.41

中国版本图书馆 CIP 数据核字(2017)第 006066 号

责任编辑:龙启铭 张爱华

封面设计:傅瑞学

责任校对:焦丽丽

责任印制:王静怡

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座 邮 编:100084

社 总 机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 装 者:北京鑫海金澳胶印有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:20.75 字 数:477 千字

版 次:2016 年 9 月第 1 版 2017 年 5 月第 2 版 印 次:2017 年 5 月第 1 次印刷

印 数:1~2000

定 价:39.00 元

产品编号:070653-01



本书第1版出版5年来,收到一些教师与读者的指正与建议,作者也在几年的使用过程中进行了总结、分析与思考。除了个别的错误外,本书主要修改了关于OpenGL的部分,增加了OpenGL的篇幅,将OpenGL与图形学原理的讲解结合到一起;加入了一些OpenGL函数解析,同时也梳理了原教材的内容。

计算机图形学的研究内容庞杂而繁多,凡是与计算机绘图相关的内容都是图形学研究的对象,讲解哪些内容,实难取舍。第2版的主导思想没有变,即讲述图形学基本原理,包括直线绘制算法、区域填充算法、三维数据的二维投影、隐藏面检测方法、光照模型等;讲解语言(VC++)、结构以及算法在图形学中的应用;讲解OpenGL是如何进行图形绘制以及动画制作的,让OpenGL的使用与对算法的理解相互促进。

各种计算机课程,应该有机联系在一起。语言软件的学习与使用既是其他课程的基础,也是一个阶段性的目的,所以在本书中,坚持强化语言的使用。

因为各校的学时不同,所以应选择相应的内容进行讲解。第1章是一些基本的(基于VC++的)绘图知识,不过,第1章并不是后面章节的基础,建议讲解6学时左右;第2章中有一些图形学二维算法,例如直线与圆的绘制、区域填充等,讲解和上机练习4~6学时;第3章讲解投影、消隐等算法,图形学基本的、重要的算法集中安排在第2、3章里,以理解为目的,可讲解8~10学时;第4章讲解OpenGL基本内容,建议讲解和上机练习6学时左右;第5章样条曲面与第6章几何造型、光照模型也是经典的图形学内容,可以讲解6~10学时;第7~9章是动画制作实例,如果想提高学生这方面的能力,可以重点讲解,讲解和上机练习18学时左右。

在目前出版物中,图形学习题不多,所以在附录中将本校近几年的图形学期末试题附上,供教师和同学们参考借鉴。

于万波参加了全书的设计,编写了第1~3章,并最后整理成稿,于硕参加编写了该书的第4、5、6、9章,周东升参加编写了第7、8章。书中个别例题、习题参考了胡欢、陈述杏、黄一彬、刘博、张艳平、杨心语、叶世利、陈妍颖等同学的作业,在此表示感谢。



最后感谢龙启铭编辑的信任、支持与指导。

期待着大家喜欢本书,也期待着得到大家的批评与建议。作者邮箱:yu_wb@sohu.com 或者 yu_wb@126.com。

作 者

2017 年 1 月



第 1 章 VC++绘图程序设计 /1

1.1 使用 CDC 类函数绘制图形	1
1.1.1 使用单文档程序绘图	1
1.1.2 绘制具有真实感的三维图形	7
1.1.3 交互绘图程序设计	10
1.1.4 绘制矩形	15
1.1.5 在指定位置输出文本	16
1.2 画笔与画刷	19
1.2.1 画笔类及其函数	19
1.2.2 画刷类	20
1.3 位图图像操作	21
1.3.1 提取位图上一点的颜色值	21
1.3.2 获取图像区域的颜色值	23
1.4 绘图与动画程序实例	24
1.4.1 小圆的弹性运动	24
1.4.2 抛物运动	27
1.4.3 小圆沿着螺旋线上升	29
1.4.4 逐帧动画制作	31
1.4.5 使用 Timer 事件函数绘制图形	32
1.4.6 移动鼠标进行书写	35
1.5 Win32 应用程序中绘图与动画制作	36
1.5.1 用多种填充形式制作动画	36
1.5.2 使用颜色渐变制作动画	37
习题 1	38

第 2 章 二维图形绘制与填充 /43

2.1 直线段绘制算法	43
2.1.1 使用直线方程计算函数值	43
2.1.2 DDA 微分绘制方法	44
2.1.3 Bresenham 算法	45

2.2	二次曲线绘制	47
2.2.1	使用方程绘制二次曲线	48
2.2.2	一般平面曲线的绘制	50
2.2.3	圆的绘制算法研究	52
2.2.4	抛物线的平移与旋转	55
2.2.5	二次贝塞尔曲线绘制算法	56
2.3	拟合曲线	59
2.3.1	最小二乘法拟合	59
2.3.2	贝塞尔曲线	61
2.3.3	B 样条曲线	65
2.4	插值曲线	66
2.4.1	简单的逐段多项式插值方法	67
2.4.2	Hermite 曲线	67
2.4.3	样条曲线	68
2.5	基于代数方程的基本规则图形填充	70
2.5.1	矩形与三角形填充	70
2.5.2	椭圆填充	72
2.5.3	抛物线围成的封闭区域填充	73
2.6	多边形填充	73
2.6.1	多边形填充的复杂性分析	74
2.6.2	扫描线填充	75
2.6.3	种子填充	80
2.7	二维分形图绘制	82
2.7.1	绘制树	82
2.7.2	绘制分形山	84
	习题 2	85

第 3 章 三维数据的二维投影 /89

3.1	三维数据投影	89
3.1.1	三维数据与二维显示	89
3.1.2	绘制空间直角坐标系	90
3.2	三维螺旋线的平行投影	91
3.2.1	参数方程及三维空间点的二维绘制	91
3.2.2	不同角度的三维螺旋线投影	93
3.3	三维数据的透视投影	95
3.3.1	平行投影与透视投影	95
3.3.2	观察坐标系下的一点透视投影	96
3.4	裁剪	98



3.4.1	二维图形裁剪	98
3.4.2	三维图形裁剪	98
3.5	视点变化下的多面体绘制	99
3.5.1	线框正方体投影绘制	99
3.5.2	视点变化下的线框正方体绘制	101
3.6	隐藏面检测	101
3.6.1	隐藏线面	101
3.6.2	一个正方体的六个面	102
3.6.3	背面检测方法	103
3.6.4	多面体的隐藏面计算	104
3.6.5	其他检测方法	104
习题 3		106

第 4 章 OpenGL /108

4.1	VC++ 中运行 OpenGL 程序	108
4.1.1	在 VC++ 中加入 glut	108
4.1.2	绘制点与线	109
4.1.3	绘制三角形与四边形	110
4.2	OpenGL 函数解析(一)	114
4.2.1	颜色设置函数 glClearColor 与 glColor	114
4.2.2	绘制函数 glBegin 与 glEnd	115
4.2.3	窗口初始化函数 glutInitWindowSize 等	116
4.2.4	OpenGL 核心函数	116
4.3	OpenGL 函数解析(二)	117
4.3.1	调用函数绘制形体	117
4.3.2	裁剪函数 glOrtho	119
4.3.3	glutSolidSphere 等	119
4.3.4	光照函数 glLight	120
4.3.5	OpenGL 实用函数	120
4.4	一个运动的正方体	121
4.4.1	三维正方体绘制与函数 gluLookAt	121
4.4.2	OpenGL 旋转函数 glRotate	123
4.4.3	使用鼠标控制旋转轴	125
4.5	具有颜色插值效果的多面体	126
4.5.1	多面体绘制	126
4.5.2	修改参数	128
4.6	OpenGL 函数解析(三)	129
4.6.1	平移函数 glTranslate 与缩放函数 glScale	129

4.6.2	面法向设置函数 glNormal3fv	129
4.6.3	双缓存函数 glutSwapBuffers	129
4.6.4	透视投影函数 glFrustum	130
4.6.5	工具函数 glut	131
4.7	OpenGL 交互操作函数	132
4.7.1	鼠标操作	132
4.7.2	键盘操作	133
4.7.3	菜单制作	134
4.8	绘制实例	135
4.8.1	绘制五角星	135
4.8.2	运动的彩色正方体	137
习题 4		140

第 5 章 样条曲面 /147

5.1	样条曲线	147
5.1.1	三维空间贝塞尔曲线	147
5.1.2	曲线的拼接	148
5.1.3	三维空间 B 样条曲线	148
5.1.4	三维空间分段插值曲线	151
5.2	贝塞尔曲面	151
5.2.1	贝塞尔曲面的定义	152
5.2.2	双一次贝塞尔曲面	152
5.2.3	双二次贝塞尔曲面	154
5.2.4	双三次贝塞尔曲面的 16 个控制点	155
5.2.5	曲面特性	157
5.3	B 样条曲面绘制	157
5.3.1	B 样条曲面定义	157
5.3.2	双二次 B 样条曲面	158
5.4	OpenGL 曲线曲面绘制	160
5.4.1	曲线绘制	160
5.4.2	曲面绘制	162
5.4.3	绘制光滑曲面	165
5.5	OpenGL 函数解析(四)	167
5.5.1	计算二维网格函数 glEvalMesh	167
5.5.2	二维求值函数 glMap2f	168
习题 5		169

第6章 几何造型与光照模型 /174

6.1 几何造型基本单元的组织	174
6.1.1 线框模型	175
6.1.2 表面模型	175
6.1.3 实体模型	176
6.2 实体模型构造方法	177
6.2.1 边界表示法	177
6.2.2 分解表示法	178
6.2.3 扫描造型法	179
6.3 场景构造与模型的重用	180
6.3.1 场景构造	180
6.3.2 模型重用	181
6.3.3 布尔运算	181
6.4 三维数据模型：地形图	182
6.4.1 绘制地形图程序	182
6.4.2 程序结构	185
6.4.3 读取数据文件	186
6.4.4 网格地形绘制	186
6.4.5 加入灯光效果	188
6.5 OpenGL 中的光照效果	189
6.5.1 按右键移动光源	189
6.5.2 安装多个光源	192
6.5.3 多个光源下的多个球体	194
6.6 OpenGL 光照函数	195
6.6.1 关于 glLight	195
6.6.2 多面体的光照效果	196
6.7 简单光照模型	198
6.7.1 镜面反射与视点位置	198
6.7.2 漫反射与环境光	200
6.7.3 Phong 光照模型	201
6.8 明暗插值与阴影生成	202
6.8.1 明暗插值方法	202
6.8.2 阴影生成	203
6.8.3 透明性	203
6.8.4 整体光照模型	204
习题 6	204

第 7 章 纹理映射: 飘动的图像与旋转的地球 /208

7.1	使用 Win32 应用程序运行 OpenGL 程序	208
7.1.1	关于 Win32 应用程序	208
7.1.2	一个飘动的图像	210
7.1.3	修改程序制作更多的飘动效果	220
7.2	构建 OpenGL 程序运行框架	223
7.2.1	函数 CreateGLWindow	223
7.2.2	函数 KillGLWindow	223
7.2.3	函数 LRESULT CALLBACK WndProc	224
7.2.4	函数 int WINAPI WinMain	224
7.2.5	OpenGL 的 glaux 辅助函数	225
7.3	网格制作与图像映射	226
7.3.1	顶点生成	226
7.3.2	网格制作	226
7.3.3	运动的网格	227
7.3.4	图像定义为纹理	228
7.3.5	图像映射到网格	228
7.4	OpenGL 函数解析(五)	229
7.4.1	OpenGL 纹理映射	229
7.4.2	OpenGL 纹理定义函数 glTexImage	229
7.4.3	OpenGL 纹理控制函数 glTexParameter	230
7.4.4	纹理与多边形颜色的融合	230
7.4.5	OpenGL 纹理坐标生成函数 glTexCoord	231
7.4.6	OpenGL 纹理映射函数应用实例	232
7.5	旋转的地球	235
7.5.1	程序实现	235
7.5.2	去掉图像的白边	238
7.5.3	球的上下左右移动	238
	习题 7	239

第 8 章 不规则图形: 粒子系统与迭代吸引子 /241

8.1	使用粒子系统制作爆炸效果	241
8.1.1	粒子系统	241
8.1.2	爆炸效果的程序实现	241
8.1.3	程序解析	246
8.1.4	修改程序实现更多效果	248
8.1.5	使用 VC++ 制作爆炸效果	250

8.1.6 使用 3ds Max 制作下雪动画	252
8.2 基于图像的图形绘制	254
8.2.1 图像动画制作	254
8.2.2 基于图像的三维图形建模	255
8.3 OpenGL 图像操作	256
8.3.1 二值图形绘制	256
8.3.2 读写像素	258
8.3.3 像素复制	261
8.4 OpenGL 函数解析(六)	263
8.4.1 OpenGL 函数 glDrawBuffer 与 glReadBuffer	263
8.4.2 OpenGL 函数 glutBitmapCharacter	264
8.4.3 OpenGL 图像操作函数	266
8.5 迭代吸引子图形绘制	266
8.5.1 正弦函数与二元二次随机多项式函数迭代	267
8.5.2 调整正弦函数观察迭代结果	268
8.5.3 离散余弦变换基函数作为辅助函数	269
习题 8	271

第 9 章 飞机动画制作与改进 /275

9.1 使用单文档运行 OpenGL 程序	275
9.1.1 单文档 OpenGL 程序	275
9.1.2 星空闪烁动画	279
9.1.3 将项目框架加入到 VC++ 选项中	281
9.2 飞机模型	282
9.2.1 运行飞机动画游戏程序	282
9.2.2 飞机数据模型分析	284
9.2.3 OpenGL 材质函数 glMaterialfv	291
9.3 动画制作	294
9.3.1 飞机的飞行	294
9.3.2 发射子弹	295
9.3.3 键盘的使用	296
9.3.4 关于动画	297
习题 9	298

附录 A 期末试题 /299

参考文献 /317

第 1 章

VC++ 绘图程序设计

计算机绘图可以使用绘图软件,如 Photoshop、3ds Max 等;可以使用专用的绘图语言与函数,如 OpenGL 等。除了专门软件以及专门语言外,很多语言(软件)本身也提供了简单的绘图功能,例如,VC++ 也有很多绘图函数。1.1 节中使用 VC++ 中 MFC 的 CDC 类封装的绘图函数绘制一些图形,除了学习 MFC 单文档中绘图功能外,还可以了解语言绘图的一些基本方式。1.2 节中通过实例深入研究一些常用的绘图函数。1.3 节介绍位图图像操作,1.4 节给出一些绘图实例,1.5 节介绍使用 Win32 应用程序绘制图形与制作动画。

1.1 使用 CDC 类函数绘制图形

VC++ 功能强大,系统庞杂,把各种功能都分散到各个类中。其中绘图功能可以使用 CDC 类中封装的函数实现。如果建立单文档(或多文档)程序,那么在其 OnDraw() 函数中调用 CDC 类函数就可以进行绘图。

1.1.1 使用单文档程序绘图

【例 1-1】 在 MFC 单文档程序的 OnDraw 函数中加入语句绘制直线。

该例题分以下几步完成(下面操作是在 VC++ 6.0 上进行的,在 VC++ 高版本上与其类似)。

(1) 建立单文档工程。

进入 VC++ 工作界面,选择 File(文件)→New(新建),在弹出的对话框中选择 Projects 选项卡,如图 1-1 所示。

在该选项卡列出的各个条目中,选择  MFC AppWizard [exe],在右边 Project name 栏写上工程名字(该例题把工程命名为 Huatui),单击右下角的 OK 按钮,出现如图 1-2 所示的界面。

在该界面上选择 Single document 单选按钮,然后单击 Finish 按钮。在随后弹出的对话框(框)窗口中单击 Finish 按钮,出现了 VC++ 的工作界面,如图 1-3 所示。

在图 1-3 中,左边有文件组织列表窗口,在这个窗口中有各个文件的组织关系。

Source Files 称为源文件;

Header Files 称为头文件;

Resource Files 称为资源文件。

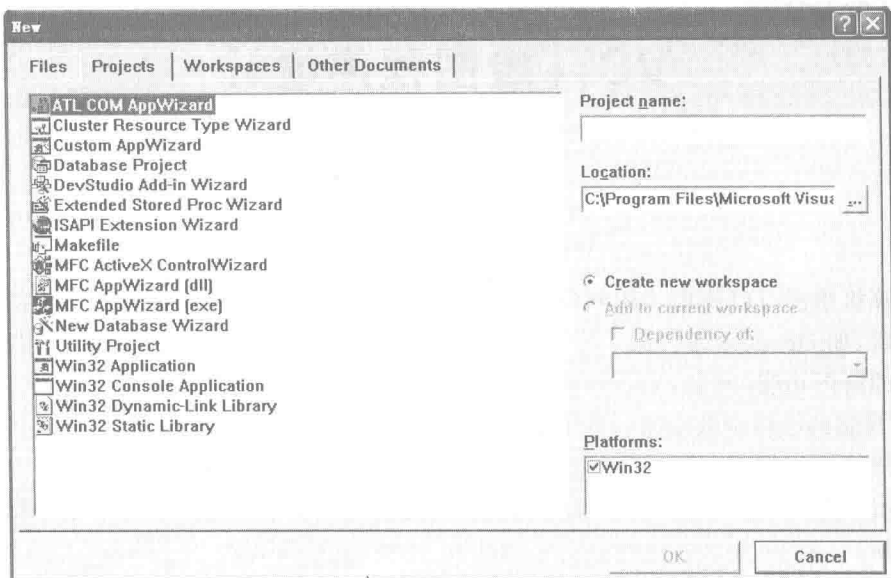


图 1-1 New 对话框 (Projects 选项卡)

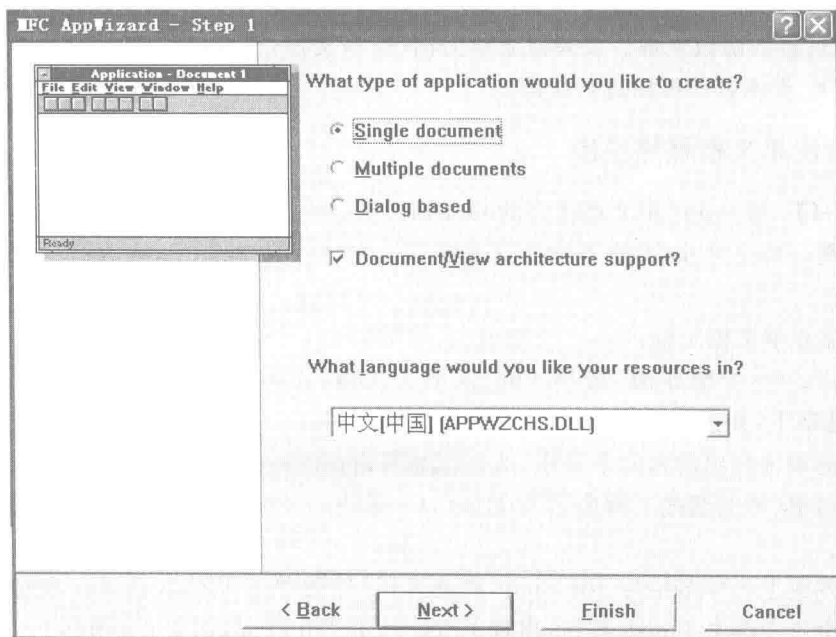


图 1-2 确定项目类型的对话框

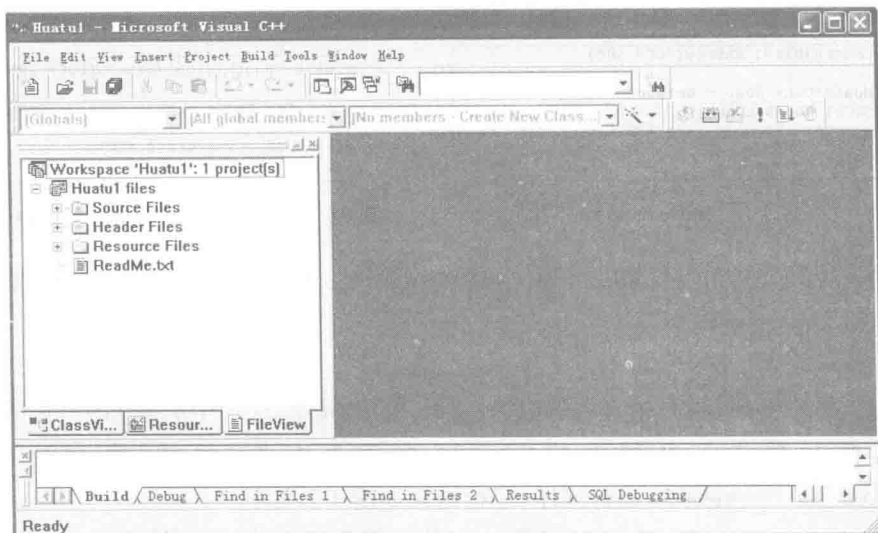


图 1-3 单文档程序设计界面 1

可以单击上面 3 个文件夹左边的 **+** 将其展开,也可以在文件夹上双击将其展开。

(2) 在 OnDraw 函数中加入语句。

展开 Source Files 文件夹,双击 Huatu1View.cpp,在右边工作区中显示出该文件的内容,拖动滚动条滑块,找到函数 void CHuat1View::OnDraw(CDC * pDC),如图 1-4 所示。

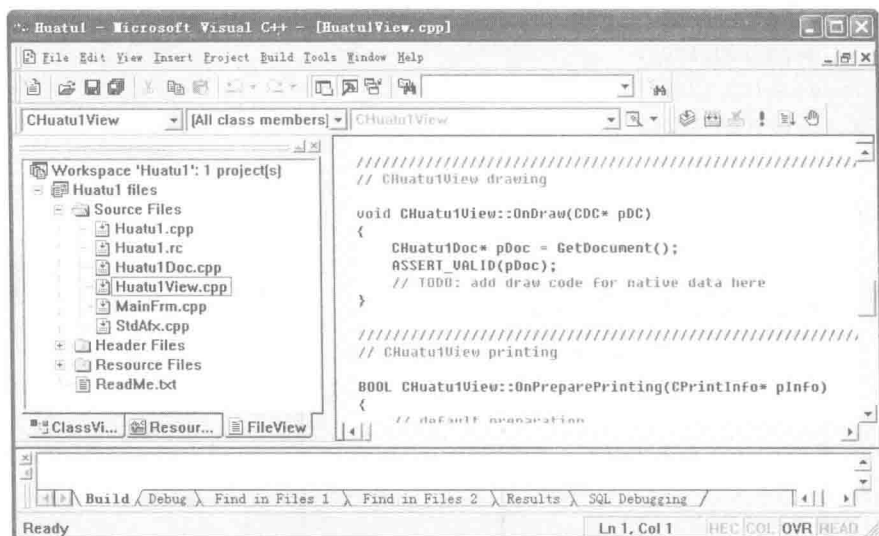


图 1-4 单文档程序设计界面 2

观察图 1-5 所示函数 OnDraw(CDC * pDC)。

在这个函数的后部加上语句:

```
pDC->LineTo(200,100);
```

如图 1-6 所示。

```
void CHuatulView::OnDraw(CDC* pDC)
{
    CHuatulDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    // TODO: add draw code for native data here
}
```

图 1-5 OnDraw 函数

```
void CHuatulView::OnDraw(CDC* pDC)
{
    CHuatulDoc* pDoc = GetDocument();
    ASSERT_VALID(pDoc);
    pDC->LineTo(200,100);
}
```

图 1-6 在 OnDraw 函数中添加绘制直线语句

【注】 语句 `// TODO: add draw code for native data here` 是注释,可以删掉,不影响语句的执行。

(3) 编译运行。

单击工具栏上的“项目编译”按钮,或者单击“编译”菜单中的 `Build Huitul.exe` `F7` 编译项目;然后单击工具栏上的“运行”按钮或者“编译”菜单中的选项 `! 执行 Huitul.exe` `Ctrl+F5`,运行程序,运行结果如图 1-7 所示。

语句 `pDC->LineTo(200,100)` 是从当前位置到绘制直线到点 `(200,100)`,该例的(默认的)当前位置为 `(0,0)`,点 `(200,100)` 在距离左边界 200、上边界 100 处。

【注】 文档中的白色窗口默认坐标是左上角为原点 `(0,0)`,从原点出发横轴正方向向右,纵轴正方向向下。

【例 1-2】 绘制多条直线。

修改例 1-1 的程序,在 `OnDraw` 函数中加入下面语句(其他都不改变)。

```
pDC->MoveTo(20,20);
pDC->LineTo(200,100);
pDC->MoveTo(50,120);
pDC->LineTo(200,120);
pDC->LineTo(200,20);
```

运行结果如图 1-8 所示。



图 1-7 在 OnDraw 函数中添加语句绘制一条直线

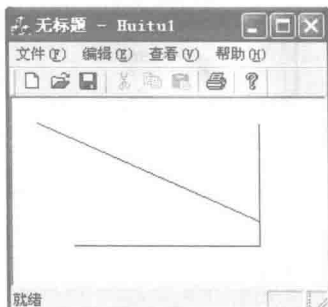


图 1-8 绘制多条直线

`MoveTo(20,20)` 是把当前焦点移到点 `(20,20)`,语句 `pDC->LineTo(200,100)` 是从点 `(20,20)` 画线到点 `(200,100)`,其他以此类推。

使用 `CDC` 类除了可以绘制直线以及移动焦点外,还可以绘制椭圆等,参考下面例题 1-3。

【例 1-3】 在 OnDraw 函数中加入语句绘制椭圆。
其他步骤不需要改变,修改 OnDraw 函数如下。

```
void CFile1View::OnDraw(CDC * pDC)
{
    CFile1Doc * pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    //TODO: add draw code for native data here
    pDC->Ellipse(30,20,300,200);
}
```

在 OnDraw 函数中加入了绘制椭圆的语句,运行结果如图 1-9 所示。

语句 `pDC->Ellipse(30,20,300,200)` 中,“30,20”为椭圆的外接矩形左上顶点坐标,“300,200”为椭圆的外接矩形右下角坐标。

【例 1-4】 使用画点函数绘制图形。

在例 1-3 的基础上,在 OnDraw 函数中加入语句,如下所示。

```
void CHuitu1View::OnDraw(CDC * pDC)
{
    CHuitu1Doc * pDoc=GetDocument();
    ASSERT_VALID(pDoc);
    for(int i=30;i<250;i++)
        for(int j=20;j<180;j++)
        {
            pDC->SetPixel(i,j,RGB(i,j,0));
        }
    pDC->Ellipse(30,20,250,180);
}
```

该程序绘制出的图形如图 1-10 所示。例 1-4 先用画点函数绘制出一个彩色矩形,然后绘制出一个椭圆,该椭圆把彩色矩形的中间覆盖了。

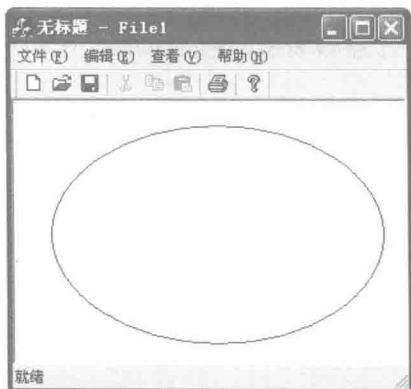


图 1-9 绘制椭圆



图 1-10 使用画点函数绘制图形