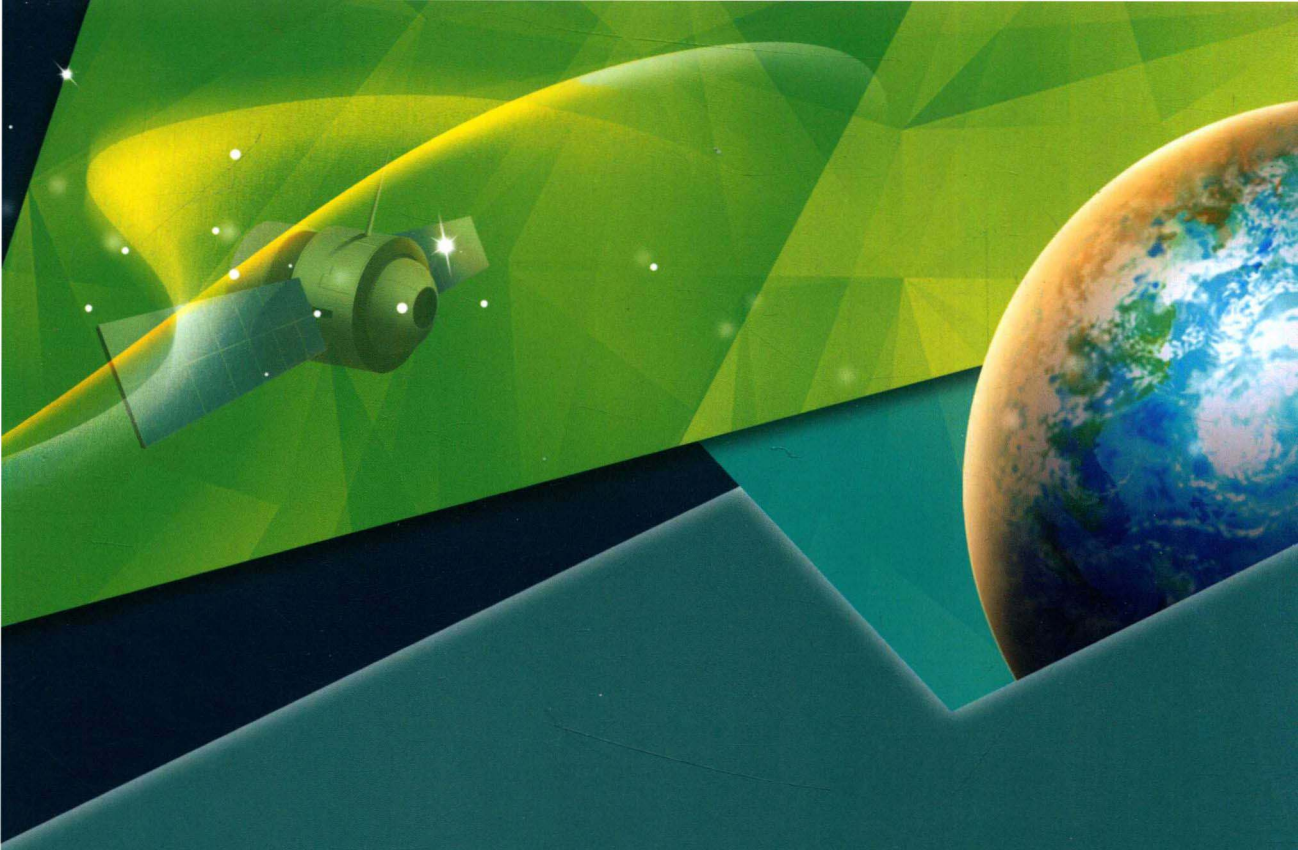




高等学校遥感信息工程实践与创新系列教材

计算机图形学实习教程 ——基于C#语言

于子凡 编著



WUHAN UNIVERSITY PRESS

武汉大学出版社

高等学校遥感信息工程实践与创新系列教材编审委员会

顾问 李德仁 张祖勋 龚健雅 郑肇葆

主任委员 秦 昆

副主任委员 胡庆武

委员 (按姓氏笔画排序)

马吉平 王 玥 王树根 付仲良 刘亚文

张 熠 李建松 李 欣 周军其 巫兆聪

胡庆武 胡翔云 秦 昆 袁修孝 贾永红

贾 涛 高卫松 崔卫红 潘 励

高等学校遥感信息工程实践与创新系列教材



计算机图形学实习教程

——基于C#语言

于子凡 编著



WUHAN UNIVERSITY PRESS

武汉大学出版社

图书在版编目(CIP)数据

计算机图形学实习教程:基于C#语言/于子凡编著. —武汉:武汉大学出版社,2017.9

高等学校遥感信息工程实践与创新系列教材

ISBN 978-7-307-19655-1

I. 计… II. 于… III. ①计算机图形学—高等学校—教材 ②C语言—程序设计—高等学校—教材 IV. ①TP391.411 ②TP312.8

中国版本图书馆CIP数据核字(2017)第215374号

责任编辑:王金龙

责任校对:李孟潇

整体设计:马佳

出版发行:武汉大学出版社 (430072 武昌 珞珈山)

(电子邮件:cbs22@whu.edu.cn 网址:www.wdp.com.cn)

印刷:湖北民政印刷厂

开本:787×1092 1/16 印张:8 字数:188千字 插页:1

版次:2017年9月第1版 2017年9月第1次印刷

ISBN 978-7-307-19655-1

定价:24.00元

版权所有,不得翻印;凡购买我社的图书,如有质量问题,请与当地图书销售部门联系调换。

序

实践教学是理论与专业技能学习的重要环节，是开展理论和技术创新的源泉。实践与创新教学是践行“创造、创新、创业”教育的新理念，是实现“厚基础、宽口径、高素质、创新型”复合型人才培养目标的关键。武汉大学遥感信息工程类(遥感、摄影测量、地理国情监测与地理信息工程)专业人才培养一贯重视实践与创新教学环节，“以培养学生的创新意识为主，以提高学生的动手能力为本”，构建了反映现代遥感学科特点的“分阶段、多层次、广关联、全方位”的实践与创新教学课程体系，夯实学生的实践技能。

从“卓越工程师计划”到“国家级实验教学示范中心”建设，武汉大学遥感信息工程学院十分重视学生的实验教学和创新训练环节，形成了一套针对遥感信息工程类不同专业和专业方向的实践和创新教学体系，形成了具有武大特色以及遥感学科特点的实践与创新教学体系、教学方法和实验室管理模式，对国内高等院校遥感信息工程类专业的实验教学起到了引领和示范作用。

在系统梳理武汉大学遥感信息工程类专业多年实践与创新教学体系和方法基础上，整合相关学科课间实习、集中实习和大学生创新实践训练资源，出版遥感信息工程实践与创新系列教材，服务于武汉大学遥感信息工程类在校本科生、研究生实践教学和创新训练，并可为其他高校相关专业学生的实践与创新教学以及遥感行业相关单位和机构的人才技能实训提供实践教材资料。

攀登科学的高峰需要我们沉下去动手实践，科学研究需要像“工匠”般细致入微实验，希望由我们组织的一批具有丰富实践与创新教学经验的教师编写的实践与创新教材，能够在培养遥感信息工程领域拔尖创新人才和专门人才方面发挥积极作用。



2017年1月

前 言

当前,计算机已经成为测绘行业的重要工具,测绘行业的从业人员必须熟练掌握计算机编程技能。而计算机图形学技术是制作电子地图的基本工具,因此计算机图形学成为测绘学科相关专业的专业基础课。计算机图形学中包含了大量的算法,充满各种理论、模型、技巧,仅依靠阅读课本,难以全面、准确地体会和理解各种算法的思路和精妙之处。动手编程实践,对于学习计算机图形学的各种算法很有好处,便于学生及时发现自己的理解偏差。

作为一门基础课,计算机图形学课程往往安排在大学低年级阶段。学生刚刚学完或同步学习面向对象程序设计课程,学到的一些基本编程技能和方法需要通过实践加以巩固和提高。但低年级学生对本专业的基本理论和方法缺乏了解,练习编程缺乏思路。计算机图形学正好提供了大量的算法,这些算法相对独立,编程难度也不大,可以为编程初学者提供大量的编程学习素材。

本书将计算机图形学的课程学习和提高编程技能紧密地结合起来。在本书中,按照章节组织计算机图形学的基本内容,各章节中选择有代表性的典型算法,首先分析算法思路,研究算法模型,然后用目前应用最广泛的 Visual C#语言对算法进行编程实现。第1章介绍了一个软件平台的编制方法,在随后的第2~8章分别介绍了计算机图形学中的各种基本内容,包括基本图形生成、图形填充和裁剪、图形的变换、图形投影、投影图形的消隐方法和曲线生成算法。每一章内容既相对独立,又是作为软件平台的组成部分,当所有章节的练习内容都做完以后,一个完整的图形学实习软件就形成了。各章的程序都有详细的注释,以帮助读者更好地理解编程思路。在程序实现过程中,尽可能使用常用的编程技巧,以帮助读者积累编程经验。为了帮助初学者降低学习难度和上升坡度,所有程序都经过检验,证明无误,读者只要加以模仿并认真体会教材内容,就能完成教材设定的基本任务。

本书在各章附有作业,为希望进一步提升编程能力的读者提供素材。

本书可作为与信息技术相关的专业本科生的计算机图形学辅助教材,也可以为需要学习和提高 Visual C#编程的初学者提供参考。

由于编者水平有限,书中难免存在不足和缺憾之处,敬请读者在阅读过程中,及时加以批评指正!

作者

2017年7月

目 录

第 1 章 实验平台建立.....	1
1.1 创建新项目	1
1.2 建立菜单	4
1.3 建立菜单响应函数	6
1.4 技巧：“橡皮筋”技术	8
本章作业	11
第 2 章 基本图形生成	12
2.1 生成直线的 DDA 算法	12
2.1.1 理论分析.....	12
2.1.2 程序实现.....	15
2.2 生成直线的中点算法.....	18
2.2.1 理论分析.....	18
2.2.2 编程实现.....	18
2.3 生成圆的 Bresenham 算法	22
2.3.1 理论分析.....	22
2.3.2 编程实现.....	23
本章作业	26
第 3 章 图形填充	27
3.1 扫描线填充算法.....	27
3.1.1 操作说明.....	27
3.1.2 编程实现.....	27
本章作业	33
第 4 章 二维裁剪	34
4.1 Cohen-Sutherland 算法	34
4.1.1 理论分析.....	34
4.1.2 编程实现.....	34
4.2 中点裁剪算法.....	37
4.2.1 理论分析.....	37

4.2.2 编程实现	38
4.3 梁友栋-Barsky 裁剪算法	41
4.3.1 理论分析	41
4.3.2 编程实现	41
4.4 窗口对多边形裁剪	43
4.4.1 理论分析	43
4.4.2 编程实现	44
本章作业	50
第5章 图形变换	51
5.1 平移	51
5.1.1 理论分析	51
5.1.2 编程实现	51
5.2 旋转	52
5.2.1 理论分析	52
5.2.2 编程实现	53
5.3 缩放	55
5.3.1 理论分析	55
5.3.2 编程实现	55
5.4 对称变换	59
5.4.1 理论分析	59
5.4.2 编程实现	60
5.5 错切变换	62
5.5.1 理论分析	62
5.5.2 编程实现	63
本章作业	65
第6章 平面投影	66
6.1 平行投影	66
6.1.1 理论分析	66
6.1.2 编程实现	67
6.2 透视投影	72
6.2.1 理论分析	72
6.2.2 编程实现	72
6.3 简单投影	75
6.3.1 理论分析	75
6.3.2 编程实现	76
6.4 场景漫游	77

6.4.1 理论分析	77
6.4.2 编程实现	79
本章作业	86
第7章 消隐	87
7.1 地形显示 1	87
7.1.1 理论分析	87
7.1.2 编程实现	88
7.2 地形显示 2	91
7.2.1 理论分析	91
7.2.2 编程实现	92
7.3 Z 缓冲区算法	95
7.3.1 理论分析	95
7.3.2 编程实现	95
本章作业	101
第8章 曲线	102
8.1 Bezier 曲线	102
8.1.1 理论分析	102
8.1.2 编程实现	104
8.2 B 样条曲线	111
8.2.1 理论分析	111
8.2.2 编程实现	112
本章作业	117
参考文献	118

第 1 章 实验平台建立

Microsoft .NET 是微软公司开发的一种面向网络、支持各种用户终端的新一代开发平台环境，其核心目标之一就是搭建第三代因特网平台，解决网络之间的协同合作问题，最大限度地获取信息，提供尽可能全面的服务。

C#语言是微软公司专门为 .NET 平台设计的开发语言之一。C#是从 C 和 C++派生出来的一种简单、现代、面向对象和类型安全的编程语言。微软宣称：C#是开发 .NET 框架应用程序的最好语言。C#运行于 .NET 之上，其特性与 .NET 紧密相关，它本身没有运行库，其强大的功能有赖于 .NET 平台的支持。

Visual Studio 是微软提供的集成开发环境，用于创建、运行和调试各种 .NET 编程语言编写的程序。Visual Studio 提供了若干种模版，帮助用户使用 .NET 编程语言(包括 C#、VB.NET、Java 语言)开发 Windows 窗体程序、控制台程序、WPF 程序等多种类型的应用程序，建立网站等。其中，Windows 窗体程序是在 Windows 操作系统中执行的程序，通常具有图形用户界面。

本书通过 C#语言编写 Windows 窗体程序来实现图形学的各种算法，并在图形用户界面上显示算法生成的图形。本书各种例子是在 Windows10 操作系统环境中，使用 Visual Studio 2015 集成开发环境实现的。

1.1 创建新项目

C#以项目为单位组织与管理软件的编制，以窗口来显示各种图形。因此，我们首先需要创建一个具有单一窗口的项目。打开 Visual Studio 2015，可以看到如图 1-1 所示的界面。

按照如图 1-2 所示的操作步骤，依次点击菜单项“文件”→“新建”→“项目”。

在出现的画面(见图 1-3)中，选择“Visual C#”→“Windows 窗体应用程序”，并在“名称(N)”栏输入工程名(本书中采用“学号加姓名”)，在“位置(L)”栏中确定将工程放在哪个文件夹中。

按“确认”键后，系统建立一个空的 Windows 窗体程序框架，其中包含一个名为 Form1 的窗体，用鼠标调整 Form1 窗体的大小，如图 1-4 所示。

在 Form1 窗体被选中的情况下(如果没有被选中，用鼠标点击一下 Form1 窗体即可)，系统窗口右下角的属性窗口列出了窗体 Form1 的属性值，找到 Text 属性，键入属性值“计算机图形学练习平台”(见图 1-5)。由于 Form1 窗体是应用程序的主窗口，该属性值在程序执行时作为程序的标题显示出来。

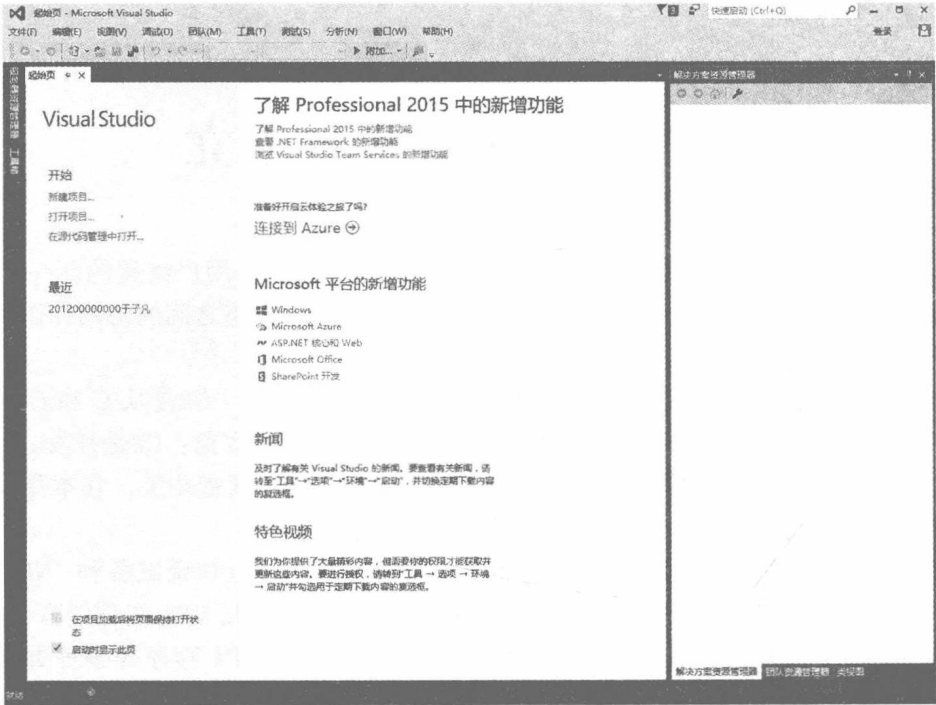


图 1-1 Visual Studio 2015 起始界面

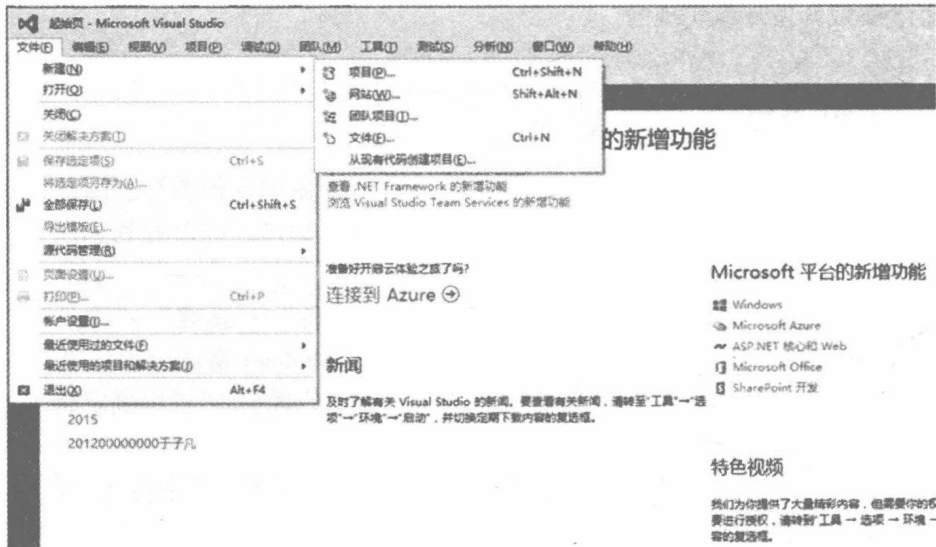


图 1-2 建立新项目步骤



图 1-3 项目命名步骤

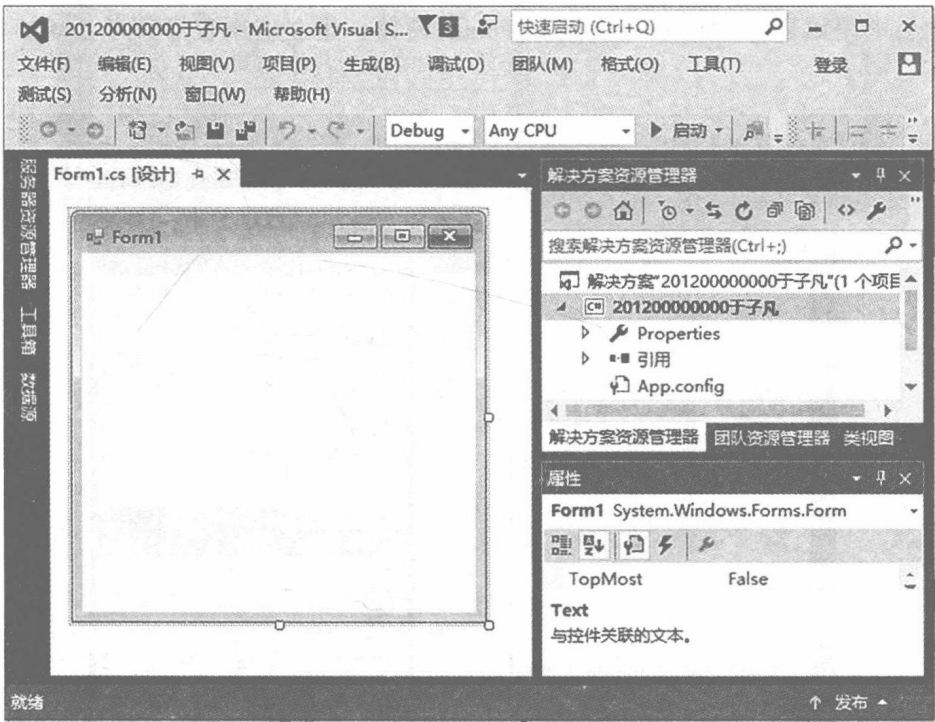


图 1-4 系统自动建立的 Windows 窗体应用空程序



图 1-5 改变窗体项目显示名的方法

1.2 建立菜单

一个软件中，菜单是必不可少的，它引导使用者正确地操作软件。一个完整的菜单由若干个菜单项组成，菜单项可以分成主菜单项、子菜单项，在子菜单项下仍然可以建立下一级的子菜单。整个菜单就是这样构成了一个多级层次。

用鼠标指向 VS2015 平台左边框架的“工具箱”，工具箱内容显示出来，在“菜单和工具栏”项目中用鼠标将“MenuStrip”拖到 Form1 窗体，建立应用程序菜单系统，如图 1-6 所示。

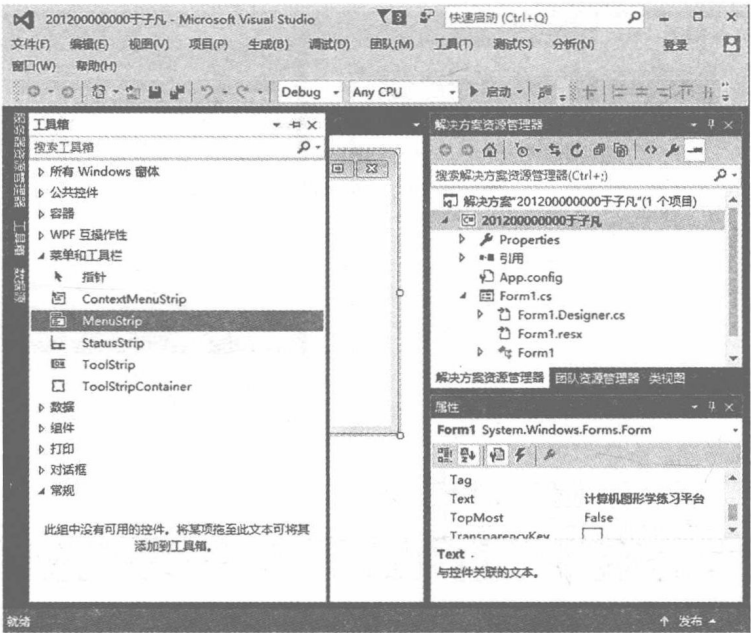


图 1-6 用控件生成程序菜单的方法

在 Form1 窗体下面，系统建立了名为“MenuStrip1”的控件，用鼠标选中该控件，Form1 窗体出现菜单第一项，显示“请在此处键入”，如图 1-7 所示。输入“基本图形生成”，可以看到该菜单项右边和下面分别出现两个空菜单项，右边是与该菜单项同级并列的另一个菜单项，下面则是该菜单项的子菜单。现在建立子菜单，在下面的子菜单项中输入“DDA 直线”，如图 1-8 所示。

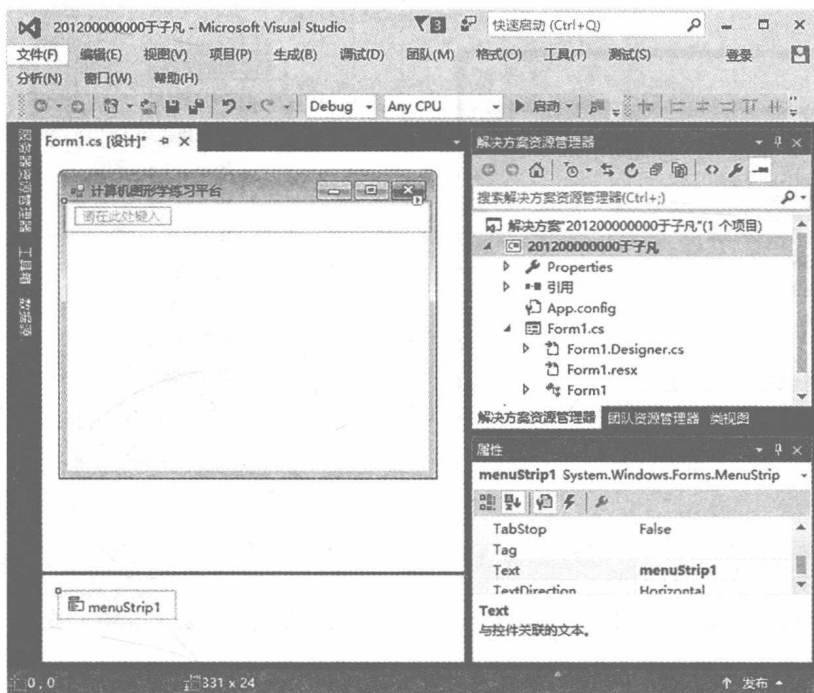


图 1-7 建立菜单项方法

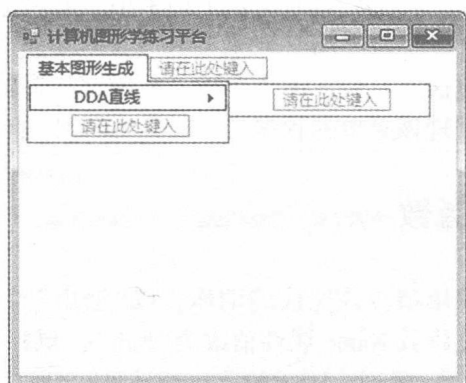


图 1-8 建立子菜单项方法

在 DDA 直线菜单项选中的情况下，图 1-7 所示的系统窗口右下角的属性窗口列出了 DDA 直线菜单项的全部属性。在该属性窗口中找到 Name 属性，鼠标点击属性值，改成

DDALine，如图 1-9 所示。

在图 1-5 中，我们修改了窗体项目的 Text 属性，在这里我们又修改了菜单项目的 Name 属性。Text 属性仅仅是在项目显示时展现出来的信息，Name 属性则是项目的名字，系统内部其他项目都是根据该名字查找、使用该项目。

用同样的方法依次建立各项子菜单，如图 1-10 所示，菜单项“基本图形生成”建立完毕。用同样的方法依次建立其他菜单项，建立的菜单如图 1-11 所示。

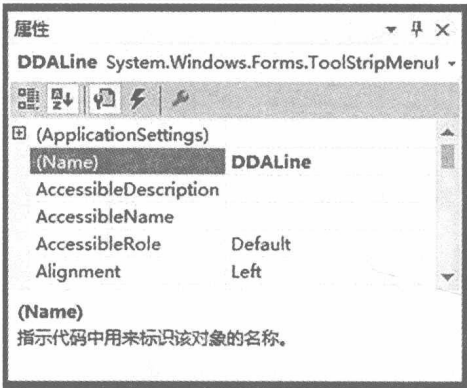


图 1-9 改变菜单项目名字的方法



图 1-10 菜单项“基本图形生成”的子菜单内容

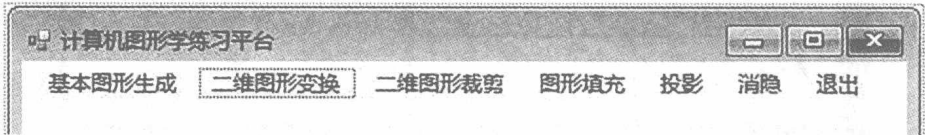


图 1-11 完成的菜单内容

至此，程序菜单系统建立完毕。含有子菜单的菜单项为父菜单项，图 1-11 中除了“退出”菜单外都含有子菜单项。父菜单的功能是导出子菜单，不需要对其编写菜单响应函数。菜单响应函数的编写方法是直接双击菜单项，系统建立菜单响应程序框架。在建立菜单响应程序框架中，增加处理该菜单的程序。下面以“退出”菜单响应函数为例作说明。

1.3 建立菜单响应函数

菜单响应函数定义了菜单项应该完成的动作，以“退出”菜单项为例说明如何建立一个菜单项的响应函数。首先将其 Name 属性值改为“Exit”，鼠标双击“退出”菜单项，系统出现如图 1-12 所示画面。可以看到，刚才的菜单设计是在“Form1.cs[设计]”页面中进行，现在打开的是“Form1.cs”页面，它是专门针对窗体 Form1 进行编程的环境，针对 Form1 编写的程序都存储在文件 Form1.cs 中。

在鼠标双击“退出”菜单项后，系统在 Form1.cs 中自动添加了一个空函数 Exit_Click。从函数名就可以看出，它是鼠标点击名为“Exit”的某个单元后的响应函数。Exit 就是刚才



图 1-12 系统自动建立的菜单响应空函数

在建立“退出”菜单项时为菜单项 Name 属性设置的属性值。所以该函数就是“退出”菜单响应函数，功能是结束应用程序。因此，添加内容如下面阴影部分所示(本书中，新添加的程序语句均用加阴影的方式表示出来)：

```
private void Exit_Click(object sender, EventArgs e)
{
    this.Close();
}
```

在添加程序语句的过程中，可以看到 VS 为编程者提供了很多帮助，如在跳出的下拉菜单中列出了当前所有可以输入的语句，我们既可以自己手工继续键入，也可以用鼠标在所列出的语句中选择一个。VS 平台具有自动查错功能，未输完或者错误的语句都会被红色波浪线提示。初学者应该尽量利用这些帮助，既可以加快输入速度，又可以避免错误。

按启动调试键 F5 键，如果没有任何语法、逻辑上的错误，系统编译并执行程序，得到如图 1-13 所示画面。

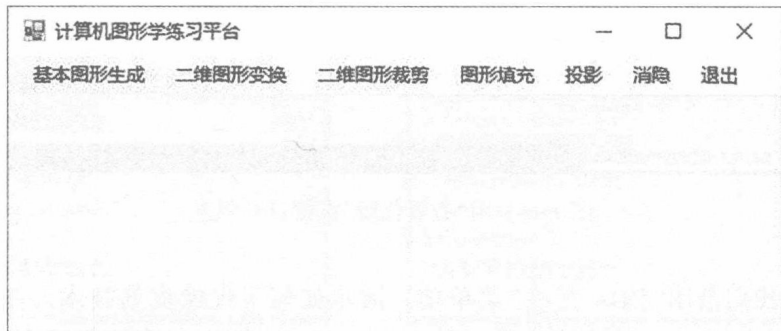


图 1-13 程序执行效果图

用鼠标滑过各个父菜单可以看到其所含的子菜单组。由于还没有实现菜单响应，点击它们，没有任何反应。只有一个例外，点击“退出”菜单，立即结束程序。

1.4 技巧：“橡皮筋”技术

用鼠标画直线需要确定两个端点。当我们确定直线的第二点时，很难记住第一点位置，这样对于将要画出的直线位置不好把握。“橡皮筋”技术能帮助我们画线时进行定位，它就像一根橡皮筋，在第一点和鼠标移动点之间始终保持一条直线，实际作用是将要画图形进行预览，如果用户对图形满意，进行确定，要画的图形就由算法完成了。

“橡皮筋”技术的关键是能够消除刚画的图形，将现场恢复成作图前的场景。在实现方面有两种技术途径：一是画图形前保留作图背景图像，需要恢复场景时将保留的背景图像拷贝到原处；二是利用“异或”方式画图形的技巧。“异或”方式具有这样的特点：将图形画第一次，有图形显现，在原地将图形画第二次，原来显现的图形消失，就好像被擦除了一样。在图形背景单纯的情况下，还可以进一步简化：第一次用前景色画图形，第二次用背景色画图形。这样，第二次的背景色图形正好覆盖第一次用前景色画的图形，从而将显现的图形擦除。

下面用实例介绍在画直线过程中，如何用前景色和背景色增加一根“橡皮筋”。

用 Visual Studio 2015 打开刚刚编制的图形程序，可以看到上次编制的 Form1 窗体和菜单系统，它们都存在于 Form1.cs[设计]页面中。Form1.cs 页面用来保存设计页面的后台程序，如果不能看到 Form1.cs 页面，在系统窗口右边的“解决方案资源管理器”中找到并点击“查看代码”按钮或者直接点击 Form1 来打开 Form1.cs 页面，如图 1-14 实线圈和虚线圈所示。



图 1-14 用“查看代码”按钮打开程序

在这里，我们借用“DDA 直线”菜单项，演示如何实现橡皮筋技术。在 Form1.cs[设计]页面中，双击“DDA 直线”菜单项，系统在 Form1.cs 页面自动生成空函数：private void DDALine_Click(object sender, EventArgs e)。在程序中添加如下内容：

```
namespace _201200000000 于子凡
```

```
{
    public partial class Form1 : Form
```

```
{
    public Form1()
```

```
{
        InitializeComponent();
```

```
        Color BackColor1 = Color.White;
```

```
        Color ForeColor1 = Color.Black;
```

```
        public int MenuID, PressNum, FirstX, FirstY, OldX, OldY;
```

```
        private void DDALine_Click(object sender, EventArgs e)
```

```
{
            MenuID = 1; PressNum = 0;
```

```
            Graphics g = CreateGraphics(); //创建图形设备
```

```
            g.Clear(BackColor1); //设置背景色
```

其中，MenuID 变量是菜单项的标志，PressNum 变量记录鼠标按键次数，FirstX、FirstY 用来记录线段第一端点坐标。实际的选点操作由 Form1 窗体中的鼠标操作完成，但因为窗体中的鼠标操作还要为众多的其他图形操作服务，并非为 DDA 直线所专有，所以要设置 MenuID 变量使窗体能够明确鼠标操作的服务对象。

在 Form1.cs[设计]页面点击 Form1 窗体空白处选中窗体，在系统窗口右下角的属性窗口中点击“事件”按钮(如图 1-15 圆圈所示)，在点击以后列出的所有窗体事件中，分别双击“MouseClicked”事件(如图 1-16 所示)和“MouseMove”，系统自动建立空的响应函数 Form1_MouseClick 和 Form1_MouseMove。

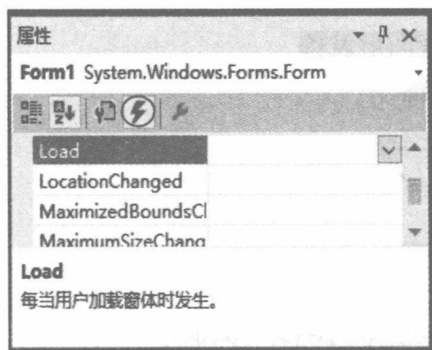


图 1-15 在属性框中选择所有事件

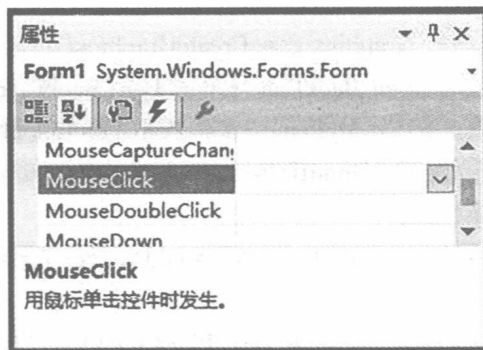


图 1-16 双击选中具体事件