高等学校数字媒体技术系列教材

游戏工具开发

Game Tools Development

王方石　吴　炜　编著

高等教育出版社

高等学校数字媒体技术系列教材

游戏工具开发

Youxi Gongju Kaifa

王方石　吴　炜　编著

高等教育出版社·北京

内容提要

本书系统地介绍了开发游戏工具所需的各种基础知识，并结合实例详细介绍了具体游戏工具的实现过程。全书共8章。其中，第1～5章分别介绍了最基础的Windows窗口编程和Windows API游戏编程，用MFC、CEGUI、Qt进行游戏工具基本界面编程，以及高级控件集成、各种关卡数据的存储模式和整个文件管理器的实现，使读者具备进行游戏工具开发所需的各种基础知识。第6、7章结合粒子系统工具和3D地图编辑器游戏工具的实现，较为详细地介绍了具体游戏工具的实现过程。第8章介绍了游戏工具的优化。通过本书的学习，使读者真正具备开发商业游戏工具的能力。

本书可作为高等学校本科计算机科学与技术、数字媒体技术等专业相应课程教材，也可供相关技术人员参考使用。

图书在版编目（CIP）数据

游戏工具开发 / 王方石，吴炜编著. --北京：高等教育出版社，2014.7

ISBN 978-7-04-039875-5

Ⅰ. ①游… Ⅱ. ①王… ②吴… Ⅲ. ①游戏程序－程序设计－高等学校－教材 Ⅳ. ①TP311.5

中国版本图书馆CIP数据核字（2014）第111482号

策划编辑	倪文慧	责任编辑	倪文慧	封面设计	王 琰	版式设计	杜微言
插图绘制	杜晓丹	责任校对	刘丽娴	责任印制	张泽业		

出版发行	高等教育出版社	咨询电话	400-810-0598
社 址	北京市西城区德外大街4号	网 址	http://www.hep.edu.cn
邮政编码	100120		http://www.hep.com.cn
印 刷	北京丰源印刷厂	网上订购	http://www.landraco.com
开 本	787mm×1092mm 1/16		http://www.landraco.com.cn
印 张	16.25	版 次	2014年7月第1版
字 数	370千字	印 次	2014年7月第1次印刷
购书热线	010-58581118	定 价	26.00元

物 料 号 39875-00

高等学校数字媒体技术系列教材编委会

序

数字媒体包括文字、图形、图像、音频、视频和动画等各种媒体形式，互联网、移动、IPTV、数字出版等各种传播形式，以及媒体的采集、存取、加工和分发的数字化过程。近20年来，随着互联网应用的普及，数字媒体技术发展迅速，已成为推动我国文化创意产业和电子信息产业发展的重要力量。为此，迫切需要培养一支适应时代要求和产业需求、具有创新能力、复合交叉型的数字媒体技术专业人才队伍。

数字媒体技术是信息技术和媒体艺术相结合的一门交叉学科，美国麻省理工学院的媒体艺术与科学实验室是该学科的探索者和领跑者。我国早期的数字媒体技术专业人才培养起源于计算机辅助设计与图形学专业方向和数字图像处理专业方向。2004年教育部批准设置数字媒体技术专业后，许多高校纷纷开设数字媒体技术专业，面向动画、游戏、新媒体、虚拟现实等领域培养本科生或研究生，受到了社会和企业的欢迎。

为了进一步推动数字媒体技术专业人才培养，教育部示范性软件学院建设工作办公室于2010年3月成立了数字媒体技术专业规范研制专家组，旨在进一步加强高等学校数字媒体技术专业建设和人才培养，组织高水平教材的编写工作。2011年10月，专家组编制完成《高等学校数字媒体技术专业规范》；同时依据专业规范，委托有丰富教学实践经验的教师编写一批专业基础课程和专业核心课程教材。这套教材正是过去两年来的重要工作成果之一。

数字媒体技术的发展日新月异，新概念、新方法、新技术层出不穷，这一切需要我们与时俱进地反映到大学课程的教学计划和教学内容中。相信这套教材的出版能够起到积极推动各高校数字媒体技术专业建设、提高教学水平和人才培养质量的作用。

希望广大教师在教学过程中对教材提出宝贵的意见和建议，使其在广泛使用过程中不断完善。

教育部示范性软件学院建设工作办公室

2013年7月

前　言

我国网络游戏市场经历了2010年震荡又重新崛起的过程之后，该领域的价值被重新挖掘，游戏的多元化探索也在不断向纵深发展。同时，硬件带宽升级也带动了游戏业的发展。据统计，2013年Q2中国网络游戏市场超过200亿元，环比增长8.1%，同比增长26.6%。移动游戏成为2013年行业热点。网页游戏以及手机游戏将成为发展重点，其3D化趋势越来越明显。在这样的大背景下，一方面国内游戏的高速发展让人们对游戏产业的价值刮目相看，另一方面真正符合游戏公司要求的人才却非常缺乏。由于利益和师资的问题，社会培训机构培养的人才只能满足公司的低端需求，而对于游戏编程开发这种对计算机编程基础、算法、数据结构、计算机图形学和数学都有很高要求的岗位，社会培训机构则很难培养出满足需求的人才。这就对我国高校相应专业的教育提出了更高的要求。

本书是作者在北京交通大学软件学院从事游戏开发教学和游戏公司从事游戏开发工作积累经验的基础上编写的。书中较为全面地介绍了游戏编程中的一个重要分支——游戏工具编辑器开发所需要的专业领域知识和基本方法；系统地介绍了游戏中主流编辑器的原理及其实现步骤，并且给出了详细的实现代码。学习本书需要读者具有一定的C++基础和基本的编程思想。

本书适用于48学时的教学，学时分配如下：第1章Windows编程基础4学时，第2章界面编程基础6学时，第3章界面集成6学时，第4章关卡数据文件4学时，第5章文件管理器实现6学时，第6章粒子系统工具开发6学时，第7章3D地图编辑器实现10学时，第8章游戏工具优化6学时。

本书第1~3章由王方石撰写，第4~8章由吴炜撰写，王方石对全书做了统稿与校对工作。北京工业大学软件学院副院长朱青教授仔细审阅了全书，提出了很多有益的意见和建议，在此表示衷心的感谢。

感谢北京交通大学软件学院数字媒体实验室为编者提供的优良科研条件和各种便利，使得本书的编写得以顺利完成。感谢实验室各位同学的支持和帮助，特别是温阳、吴琼、师磊、王震军、朱雪阳、郑帅、吴非成几位同学为本书所做的贡献。

特别感谢北京交通大学软件学院的卢苇院长为编者提供的极大帮助和支持。

游戏编程涉及的专业领域知识十分广泛，与之相关的学科也很多，由于作者的学识及水平有限，书中难免存在不足之处，敬请读者批评指正。

作　者

2014年4月

目　录

第1章 Windows 编程基础

游戏开发是一项系统的工程，特别是大型网络游戏的开发。所以要想学好游戏编程，必须系统地学习进行游戏开发所需的各种知识。在游戏开发过程中离不开各种游戏工具，方便好用的游戏工具不仅可以提高策划人员和美工人员的效率，同时还可为程序开发人员节省大量的时间。因此在游戏开发过程中，各种游戏工具的开发就显得非常重要。目前游戏工具开发的主流平台是 Windows 平台，本章主要介绍在 Windows 平台下进行游戏工具开发所需要掌握的 Windows 基础知识。

1.1 Windows 概述

Windows 是由美国 Microsoft 公司开发的视窗操作系统，它是目前世界上使用最广泛的操作系统。Windows 的发展始于 Windows 1.0 版本，是 Microsoft 公司对商业视窗操作系统的第一次尝试，也是一个非常失败的产品。Windows 1.0 完全建立在 DOS 基础上，不能执行多任务，运行速度慢且交互性很差，这可能是导致其失败的最主要原因。此外，当时的计算机硬件配置也很低，不能满足 Windows 1.1 的运行要求。

此后，Microsoft 很快推出了 Windows 2.0，但是 Windows 2.0 依然是建立在 DOS 基础上的视窗管理器，没有实质性的改变。

1. Windows 3.x

1990 年 5 月 22 日，Microsoft 正式发布具备图形用户界面、支持 VGA 标准及配置，且具有与当前 Windows 系统的 3D 功能相似的 Windows 3.0。该操作系统还有非常出色的文件管理和内存管理功能，因此获得了用户的普遍认同。

此后 Microsoft 推出的 Windows 3.1 支持多媒体扩展，具有音频和视频功能，是一个出色的、全面的操作系统，用户能以统一的方式工作。在其基础上，Microsoft 还推出了其他版本，如可支持网络功能的 Windows 3.11（适用于工作站的 Windows）。唯一的问题在于 Windows 3.1 仍是一个 DOS 应用程序，运行在 DOS 扩展器上。

2. Windows 95

1995 年 8 月，Microsoft 推出了 Windows 95，它彻底取代了前面的版本。今天的 Windows 系统依然保留了 Windows 95 的桌面、任务栏和“开始”菜单。

Windows 95 是一个混合的 16 位/32 位系统，它是 MS-DOS 和视窗产品的直接后续版本。该系统第一次抛弃了对前一代 16 位 x86 的支持，因此它要求使用 Intel 公司的 80386 处理器，或者在保护模式下运行于一个兼容的速度更快的处理器。它以对图形用户接口（GUI）进行的重要改进和底层工作(Underlying Workings)为特征，同时也是第一个捆绑了 DOS 版本(Microsoft DOS 7.0）的视窗版本。这样，Microsoft 就可以保持由视窗 3.x 建立起来的 GUI 市场的统治地位，使得其他公司的产品无法提供针对系统的底层操作服务。Windows 95 具有双重角色，它带来了更强大、更稳定、更实用的桌面图形用户界面，也结束了桌面操作系统间的竞争。

3．Windows NT

1996 年 7 月，微软推出了 Windows NT 4.0，它有 4 个版本：工作站版（Workstation）、终端服务器版和另外两个服务器版。Windows NT 4.0 首次加入了 Internet Explorer 浏览器，并与通信服务紧密集成，提供文件和打印服务，能运行客户机/服务器应用程序，内置了 Internet/Intranet 功能。

4．Windows 98

1998 年中期，Microsoft 推出了 Windows 98，它不像 Windows 95 那样是一个换代产品，至多只能算是一个改进版本。但毫无疑问它也占有很重要的地位。Windows 98 是全 32 位的，很好地集成了 DirectX、3D 图形、网络及 Internet。

5．Windows 2000

Windows 2000 采用 NT 技术，并在其上做了大量改进，使该平台比此前的 Windows 操作系统平台更可靠，更易扩展、部署、管理和使用。值得一提的是，Windows 2000 提供了对多国语言的支持。

Windows 2000 还提供了增强的网络管理功能，利用“网上邻居”的“添加网络组件”命令，可以为网络增加网络管理和监视工具，监视网络设备的活动，并向网络控制台工作站汇报情况。

Windows 2000 家族有两大类平台（共 4 种操作系统）：第一类是工作站平台 Windows 2000 Professional；第二类是服务器平台。服务器平台有如下 3 种。

（1）Windows 2000 Server：除了包含 Windows 2000 Professional 的所有特性外，还提供了简单的网络管理服务。

（2）Windows 2000 Advanced Server：除了包含 Windows 2000 Server 的所有特性外，还提供了更好的可扩展性和有效性，并且支持更多的内存、处理器以及群集。

（3）Windows 2000 Data Center Server：除了包含 Windows 2000 Advanced Server 的所有特性外，还提供了更多的内存和对处理器的支持，适用于大型数据仓库和在线事务处理等重要应用。

6. Windows XP

Windows XP 于 2001 年 10 月 25 日正式发布，XP 即 eXPerience（体验）。与 Windows 2000 相同，Windows XP 基于 NT 技术，是纯 32 位的操作系统，其功能更健壮、更稳定。

Windows XP 在网络特性上的增强有不少值得称道的方面。很多用户都知道 Windows 2000 中的脱机文件功能，Windows XP 对其做了进一步优化，可将脱机文件加密，使其更为安全。在兼容性方面，Windows 2000 的一个弊病是与不少应用软件（特别是游戏软件）不兼容。但是，在 Windows XP 中不存在这个问题，在其他 Windows 系统上可运行的软件几乎都可在 Windows XP 上运行。Windows XP 的核心代码是基于 Windows 2000 的，所以在 Windows NT 4.0/2000 上进行升级安装是十分容易的。Windows XP 提供了 4 个适用于不同用途的版本：个人版（支持 1 个 CPU）、专业版（支持 2 个 CPU）、服务器版（支持 4 个 CPU）和高级服务器版（支持 8 个 CPU）。

7. Windows Vista

全新的 Windows Vista 于 2006 年 11 月 30 日发布。人们可以在 Windows Vista 上对下一代应用程序（如 WinFX、Avalon、Indigo 和 Aero）进行开发创新。Windows Vista 更加安全可信，其版本可分为 Home 和 Business 两大类，分别对应 Windows XP 的各个版本。Windows Vista 内置 DirectX 10，使用了更多的动态链接库，不向下兼容，显卡的画质和速度得到了革命性的提升。

8. Windows 7

2009 年 10 月 22 日上午 11 时（UTC-4）Microsoft 公司首席运营官史蒂夫·巴尔默正式在纽约召开 Windows 7 的发布会。Windows 7（开发代号为 Blackcomb 和 Vienna，后更改为 7）是微软公司目前最新的 Windows 操作系统版本，供个人计算机使用，适用于家庭和商业工作环境、笔记本电脑、平板电脑及多媒体中心等。

Windows 7 集成了 DirectX 11 和 Internet Explorer 8。Windows 7 RC 中文版界面以 DirectX 11 作为 3D 图形接口，不仅支持未来的 DX11 硬件，还向下兼容当前的 DirectX 10 和 10.1 硬件。DirectX 11 增加了新的计算 shader 的技术，允许 GPU 执行更多的通用计算工作，而不仅仅是 3D 运算，这可以鼓励开发人员更好地将 GPU 作为并行处理器使用。

1.2 Windows 基础编程

1.2.1 匈牙利命名法则

Windows 操作系统项目非常庞大，几千名程序员可能需要几年的时间才能完成，若没有一套标准的代码编写规则，代码的可阅读性将会很差，因此 Microsoft 公司的总设计师查尔斯·西蒙尼创立了一套 Microsoft 代码的编写规范。这套规范实用性较强，已成为编写代码的指导说明

书。Microsoft 公司所有的 API、界面、技术文件等都遵循此规范。由于查尔斯·西蒙尼是匈牙利人，因此将该规范称为匈牙利命名法。

匈牙利命名法是计算机程序设计中的一种命名规则，使用此方法命名的变量显示了其数据类型。根据匈牙利命名法的规则，一个变量名由一个或多个小写字母开始，这些字母有助于记忆变量的类型和含义，其后可以是程序员给出的任何名称，后半部分的首字母可以是大写字母，用以区别前面的类型指示字母。

表 1-1　匈牙利命名法的前缀代码指导说明书

前　缀	数据类型(基本类型)
c	字符
by	字节(无符号字符)
n	短整数和整数(表示一个数)
i	整数
x, y	短整数(通常用于 x 坐标和 y 坐标)
cx, cy	短整数(通常用于表示 x 和 y 的长度，c 用于计数)
b	布尔型(整数)
w	UINT(无符号整数)和 WORD(无符号字)
l	LONG(长整数)
dw	DWORD(无符号长整数)
fn	函数指针
s	字符串
sz, str	以 0 字节终止的字符串
lp	32 位的长整数指针
h	编号(常用于表示 Windows 对象)
msg	消息

1．变量的命名

应用匈牙利命名法时，变量可用表 1-1 中的前缀代码来表示。另外，当一个变量由一个或几个子名构成时，每一个子名都要以大写字母开头。例如：

```
char   *szGameName;          //以 0 字节终止的字符串
int    *lpiGameObject;       //指向 32 位类型的 int 指针
BOOL   bGameStart;           //一个布尔变量
WORD   dwHeroCount;          //一个 32 位的无符号整型
```

在匈牙利命名法中对局部变量不进行说明，以 g_或 g 表示全局变量：

```
int g_iZPos;                    //一个全局 z 坐标的变量
int g_iTimer;                   //一个全局的时间变量
char *g_szString;               // 一个全局的以 0 字节终止的字符串
```

2. 函数的命名

采用匈牙利命名法命名函数的方式与命名变量的方式相同，但无前缀。要求函数名称中每个单词的第一个字母大写，例如：

```
int DrawSphere(int ix,int iy,int iz);
void *LoadTexture(char *szString);
```

函数命名中的下划线是非法的，例如下列函数名为无效的匈牙利命名：

```
int Draw_Circle(int ix, int iy);
```

3. 类型和常量的命名

采用匈牙利命名法命名类型和常量，要求其名字均为大写字母，在名字中可以使用下划线。例如：

```
const   LONG   NUM_SECTORS = 100;
#define   MAX_HEIGHT 100;
typedef unsigned char   TBYTE;     //用户自定义类型
```

在实际开发过程中，大多数程序员喜欢用下划线，因为这样可以使类型名和常量名更具可读性。

4. 类的命名

类命名的约定稍显麻烦，但仍有很多人在遵守这个约定，并独立地进行补充。所有 C++的类必须以大写的 C 为前缀，组成类名的每个单词的第一个字母都必须大写。例如：

```
class CVector                   //向量类
{
public:
    CVector();{ix=iy=yz=imagnitude=0;}
    CVector(int x,int y,int z)
    {
        ix = x;
        iy = y;
        iz = z;
    }
    private:
    int ix,iy,iz;
    int imagnitude;
};
```

5. 参数的命名

函数的参数命名和标准变量命名的约定相同，但也不总是如此。例如下面的函数定义：

```
void   SetPos(int x,int y);
```

若严格按照匈牙利命名法命名函数，应该是：

```
void   SetPos (int ix,int iy);
```

但这两种命名无太大区别，只是书写方式不同而已。在匈牙利原型中甚至可以不写函数的参数名，而仅写类型名：

```
void   SetPos (int, int);
```

当然，这仅在原型中使用，真正的函数声明必须带有可赋值的变量名。

匈牙利命名法的命名规则是 Microsoft 代码的规范，在实际开发过程中可以参照匈牙利命名法的命名规则制定适合某一项目的代码规范，这些规范可以根据项目的不同而有所不同。

1.2.2　Windows 类

Windows 操作系统是一个多任务、面向对象的图形操作系统。在 Windows 编程中类的概念非常重要，Windows 编程中的窗口、对话框、列表框、文本框等实际上都是窗口，它们之间的区别就在于定义它们的类不同。Windows 类的本质就是对其对应窗口类型的描述。

在 Windows 编程中有许多预定义的 Windows 类，如按钮、对话框、列表框、文本框等。用户也可根据实际需要创建新的 Windows 类。在实际开发中，程序员可以为自己编写的每一个应用程序创建至少一个 Windows 类。在创建一个窗口时，应考虑以一个 Windows 类作为 Windows 的一个模板，以便在其中处理信息。

描述 Windows 类信息的数据结构有两个：WNDCLASS 和 WNDCLASSEX。在早期版本中使用 WNDCLASS，目前常用新的扩展版 WNDCLASSEX。两者结构非常相似，以下为 Windows 头文件中定义的扩展版 WNDCLASSEX。

```
typedef struct _WNDCLASSEX
    {
        UINT cbSize;                //结构的字节数
        UINT style;                 //类型的标识
        WNDPROC lpfnWndProc;        //相应的消息处理函数
        int cbClsExtra;             //附加内存空间
        int cbWndExtra;             //附加内存空间
        HANDLE hInstance;           //窗口的事例句柄
        HICON hIcon;                //窗口的小图标
        HCURSOR hCursor;            //窗口的鼠标形状
        HBRUSH hbrBackground;       //背景颜色
        LPCTSTR lpszMenuName;       //窗口菜单
        LPCTSTR lpszClassName;      //窗口名称
```

```
    HICON hIconSm;          //句柄的小图标
} WNDCLASSEX
```

1. 字段 cbSize

字段 cbSize 非常重要，它是 WNDCLASSEX 结构本身的大小：

```
winclass.cbSize = sizeof (WNDCLASSEX);
```

2. 字段 style

字段 style 是描述该窗口一般属性信息的标识。该标识的取值较多，在此不能一一列出，仅列出常用的标识（见表 1-2 所示）。用户可以对这些取值任意进行逻辑“或”和“与”运算，定义所希望的窗口类型。

表 1-2 Windows 的类型标识

标 志	说 明
CS_HREDRAW	若移动或改变了窗口宽度，则刷新整个窗口
CS_VREDRAW	若移动或改变了窗口高度，则刷新整个窗口
CS_OWNDC	为该类中的每个窗口分配一个单值的设备描述表(DC，将在本章后面的内容中详细描述)
CS_DBLCLKS	当用户双击时向窗口程序发送一个双击的信息，同时光标位于属于该类的窗口中
CS_PARENTDC	在母窗口中设定一个子窗口的剪贴区，以便能够将子窗口添加到母窗口中
CS_SAVEBITS	在一个窗口中保存用户图像，以便在该窗口被遮住、移动时不必每次刷新屏幕。但是这样会占用更多的内存，并且比人工进行同样的操作要慢得多
CS_NOCLOSE	禁用系统菜单中的关闭命令

注意：用黑体显示的部分为最常用的标识。

3. 字段 lpfnWndProc

字段 lpfnWndProc 是一个指向事件句柄的函数指针。在此所设定的基本上都是该类的回调函数。在 Windows 编程中经常使用回调函数，其工作原理为：当有事件发生时，Windows 通过调用一个已建立的回调函数来通知开发者，避免开发者盲目地进行查询，随后在回调函数中再进行所需的操作。

此过程即 Windows 事件循环和事件句柄的基本操作过程。需先向 Windows 类申请一个回调函数(当然需要使用特定的原型)，当一个事件发生时就调用它。

```
wc.lpfnWndProc=WindowProc;//相应的消息处理函数
```

4. 字段 cbClsExtra 和字段 cbWndExtra

字段 cbClsExtra 和字段 cbWndExtra 原设计为：指示 Windows 将附加的运行时间信息保存

到 Windows 类的某些单元中。但在实际编程中，通常会将这两个值设为 0。

```
wc.cbClsExtra=0; //附加内存空间
wc.cbWndExtra=0;//附加内存空间
```

5. 字段 hInstance

字段 hInstance 是一个简单的，在启动时传递给 WinMain()函数的句柄实例，因此只需从 WinMain()函数参数中复制即可：

```
wc.hInstance=hInstance;//窗口的事例句柄
```

其余的字段与 Windows 类的图像处理有关，在介绍它们之前，先回顾一下句柄的概念。在 Windows 程序和类型中经常会见到位图句柄、光标句柄、任意事件的句柄等。句柄是一个基于内部 Windows 类型的标识符，均用整数表示。以 h 为前缀的类型通常都是一个句柄。

6. 字段 hIcon

字段 hIcon 用于设定表示应用程序类型的图标。开发者可以装载自己定制的图标，但若使用系统提供的图标，则需要为它设置一个句柄。使用 LoadIcon()函数可得到一个常用系统图标的句柄，例如，下面的代码用于装载一个标准应用程序的图标：

```
winclass.hIcon = LoadIcon(NULL, IDI_APPLICATION);
```

以下为 LoadIcon()函数的原型：

```
HICON LoadIcon(HINSTANCE hInstance, LPCTSTR lpIconName);
```

其中，hInstance 是一个从应用程序装载图标资源的实例，将其设置为 NULL，表示装载一个标准的图标。lpIconName 是图标标识符，表示被装载图标资源的名称，是以 NULL 为终止符的字符串。当 hInstance 为 NULL 时，lpIconName 的取值如表 1-3 所示。

表 1-3 LoadIcon()的图标标识符

值	说 明
IDI_APPLICATION	默认应用程序图标
DI_ASTERISK	星号
IDI_EXCLAMATION	惊叹号
IDI_HAND	手形图标
IDI_QUESTION	问号
IDI_WINLOGO	Windows 徽标

7. 字段 hCursor

字段 hCursor 与 hIcon 相似，是一个图像对象句柄。不同的是，hCursor 是一个指针进入到