

基于混合编程的外军飞机性能三维可视化软件^①

朱 敏, 高 鹰, 石 宇, 刘 扬

(空军航空大学 信息对抗系, 长春 130022)

摘 要: 针对外军新型飞机及装备逐渐增加和更新换代, 而传统管理方式难以对其进行高效管理和性能分析的问题, 设计制作了针对外军飞机性能的可视化软件. 通过自定义控件的设计, 使 XNA 的属性和方法运用于 .NET 框架之下, 实现了飞机三维模型的动态加载并进行了适当改进, 完成了碰撞检测及参考线和文本的绘制. 将飞机三维模型的名称和飞机 ID 存入对应的数据表中, 减少了数据库的大小. 测试结果表明, 人机交互效果良好, 二次加载速度提升约 98%.

关键词: NET; 混合编程; 可视化软件; XNA; SQL SERVER; 自定义控件

3D Visualization Software of Foreign Military Aircraft Performance Based on Hybrid Programming

ZHU Min, GAO Ying, SHI Yu, LIU Yang

(Department of Information Countermeasure, Aviation University of Air Force, Changchun 130022, China)

Abstract: Under the increase and upgrade of new foreign military aircrafts and equipment, the traditional management is difficult to manage it and analyze its performance efficiently. To solve this problem, we design and manufacture the 3D visualization software of foreign military aircraft performance. By designing custom controls, the properties and methods of XNA are used under NET Framework. The dynamic loading of 3D aircraft model and appropriate improvements are realized. At the same time, the reference lines and texts are drew and the collision detection is completed. It reduces the size of the database and enhances the performance through storing the path of the 3D model in the corresponding data table. The result shows that human-computer interaction is good and the second load is 98% faster.

Key words: NET; hybrid programming; visualization software; XNA; SQL SERVER; custom controls

随着时代的不断发展, 新技术新装备层出不穷, 而军用飞机种类和数量也在日渐增加之中. 传统的手记脑记等记录的方式已很难满足复杂军事斗争的要求. 如何对外军军用飞机进行科学化、人性化、三维可视化的高效管理与性能分析, 已经越来越引起人们的重视. 改变传统的军用飞机管理方式, 对于提高工作效率、减轻人们的工作负担具有重要的意义.

针对这个问题, 本文在 WinForm 应用程序框架下, 利用自定义控件结合 XNA 和数据库技术, 快速实现了三维模型的动态加载技术的改进及线条、文本的绘制工作, 通过 DevExpress 第三方控件的合理运用, 使界面更加友好. 完成了飞机性能分析三维可视化软件的

制作, 达到了快速查看三维模型并进行性能动画分析的目的.

1 系统整体构成及模块功能设计

本文的设计目标是制作一个基于虚拟现实的、运行于 .NET 环境的、具有三维交互功能的外军飞机性能分析系统软件. 该软件由桌面系统和飞机性能信息及模型数据库组成, 其中桌面系统由数据库管理、基础查询和三维展示、重点性能的动画分析和系统管理等模块构成. 主要实现了飞机性能数据的存储、管理和检索查询、三维模型的可视化展示、数据对比图形的动画展示等功能. 图 1 为桌面系统的模块框图.

^① 收稿时间:2014-10-06;收到修改稿时间:2014-11-13

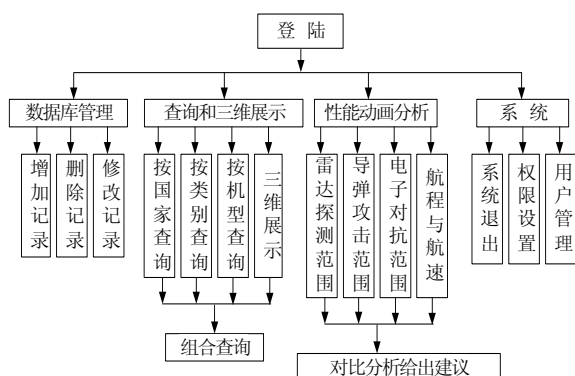


图1 桌面系统模块图

软件中飞机信息数据库管理和基础查询三维显示模块共同实现了对飞机信息的存储管理,前者提供了飞机基本信息的添加、修改、删除功能,后者则根据前者已经入库的参数信息以及选择的对比项目,实现了具体飞机的三维显示以及根据所选性能参数所做的静态图形化对比。飞机三维静态展示界面如图2所示(以选择美国军机为例)。当某一飞机信息在数据库管理模块中被修改或删除后,飞机基础查询和三维展示模块以及性能动画分析模块中将会做出相应的改变。在飞机性能动画分析模块中,可以根据需要在数据库中提取所选机型的性能参数信息,将其作为三维模型的属性值,进行动画立体展示,并对相关结果给出汉字提示。系统管理模块提供了增加用户信息和管理用户权限等功能。



图2 飞机三维静态展示界面

2 关键技术研究

2.1 XNA 技术简介

XNA 是微软开发的主要用于中小型三维游戏设

计的软件,并且支持多种平台。自2006年第一个XNA版本发布以来,XNA2.0和XNA3.0版本相继问世,如今最新的XNA版本更已是XNA4.0。XNA的基本框架包括图像引擎(Graphic Engine)、游戏应用程序模型(Game Application Model)和内容管道(Content Pipeline)3个部分,所有这些DLL文件用C#编写,并且是完全托管的^[1]。

与DirectX和OpenGL相比,XNA有着相当的优越性^[2,3]。首先,它集成了软件开发中常用的多种功能,包括音效、绘图、物理运算、I/O输入输出控制等,大大提高了程序的开发效率,凸显了人性化的优势。其次,可编程流水线的运用使XNA实现了对每个像素点的任意操作,提高了图形编程的灵活性。最后,XNA使用素材管道^[4](Content Pipeline)加载资源,降低了加载资源的难度,提高了资源加载的速度,这是XNA最大的特点。

2.2 定义控件的设计

控件作为一种可重用的组件或对象,除了自己独特且美观的外表外,它还应有自己的属性和方法,并且能够响应必要的系统或用户事件,以此来完成与用户人性化的交互。Visual Studio C#提供了自己创建个性化控件的功能,称之为自定义控件。自定义控件的创建方式大体有以下两种:一是将现有控件进行合理组合,从而改进控件的外观并实现一些新的功能;二是自己设计用户界面和代码来建立新控件。本系统中将采用第二种方式。

2.2.1 用自定义控件的必要性

用三维场景表示的桌面环境来替代传统的二维界面,这无疑更贴近人们的自然习惯^[5]。因此,在任何三维图形软件中,一个具有沉浸感的三维场景界面是整个界面的重要组成部分,实现二维控制面板与三维操作场景交互便成为了必须解决的问题。本软件使用XNA和C#语言^[6]进行开发。令人遗憾的是,XNA本身不具备与三维对象的交互功能^[7]。它会自动创建一个Game窗口,在这个窗口里三维程序可以初始化硬件并提供更新和绘制,然而在其中要实现文字显示、系统的设置甚至右键菜单等操作都远不如在WinForm中灵活^[8],有人提出将之隐藏而将绘制输出到另一个WinForm窗体的控件(如Panel)里的解决方案,但这显然会降低程序的执行效率,且往往取不到理想的效果。本软件通过自定义控件的方式将XNA的属性和方法运用于

WinForm 框架之下,从而很好地解决了这个问题。

2.2.2 自定义控件的继承关系设计

在本软件中重用了微软公布的示例代码 WinForms ContentLoadingSample 的 GraphicsDeviceControl.cs、GraphicsDeviceService.cs 和 ServiceContainer.cs 文件,使用面向对象的办法^[9],设计了自定义控件 DisplayControl 和 CartoonControl,且都继承自 GraphicsDeviceControl,分别用来进行静态旋转模型展示和模型动画性能分析展示。根据用途的不同,创造了一个模型的基类 BasicModel,以及他的两个子类: DisplayModel 和 CartoonModel,分别运用于 DisplayControl 和 CartoonControl 中,其中后者拥有私有的方法 GetWorld、GetView 和 CollidesWith,用来完成模型动画性能分析展示中的碰撞检测等功能。另外,还封装了用于汉字提示的模块 Font,继承框图如图 3 所示。

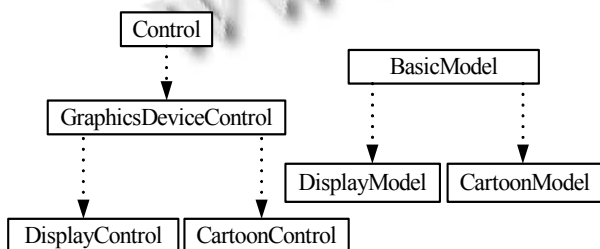


图 3 自定义控件和相关类继承框图

2.3 三维模型的动态加载技术

在 3DSMAX 中创建好飞机的三维模型后,运用 PandaDirectXMaxExporter 插件将其导出为 .X 文件,同时它的贴图会自动生成一个位图文件(注意这里的位图文件长和宽的大小必须是 4 的倍数,否则程序加载模型时会报错),它是模型的纹理文件,模型的 .X 文件会引用这些纹理文件^[10],因此必须确保该纹理文件与 .X 文件保存在同一个路径之下(本系统中将之放在程序的相对路径里)。这里, .X 是一种三维模型格式, XNA 技术并不能直接识别该格式的文件,必须先将之编译为 .xnb 模型文件,再进行加载。

2.3.1 xnb 文件的生成及加载

ContentBuilder 类和 ContentManager 类负责实现三维模型动态加载: ContentBuilder 实现动态导入 .X 模型文件并编译成 XNA 专用的 .xnb 文件, ContentManager 用来加载 .xnb 模型。ContentBuilder 类声明了一个用于生成 .xnb 文件的内容编译器的列表(本系统

使用了内建的 ModelProcessor 处理器^[11]),然后声明了三个用于生成操作的辅助类: 一个 MSBuild Engine 对象, 一个 MSBuild Project 对象和一个自定义的 ErrorLogger 类型用于实现 MSBuild ILogger 接口。常规的模型生成及加载的主要步骤如下:

- 调用 ContentManager.Unload 方法卸载前一个模型的数据。
- 调用 ContentBuilder.Add 方法进行内容编辑器和 3 个辅助对象的声明。
- 调用 ContentBuilder.Build 生成选择的内容并检查是否有生成错误。
- 调用 ContentManager.Load 读取刚生成的模型(.xnb 文件), 将其显示到界面上。程序运行结束后释放存储了生成内容的临时文件夹。

2.3.2 对加载技术的改进

在实际操作中发现,程序每次调用 ContentBuilder.Build 时需要耗费 6 秒左右的时间,这样便大大减慢了模型加载的速度,造成程序“假死”的现象。为了避免这个问题,张静将调用 ContentBuilder.Build 生成 .xnb 模型文件的耗时操作调整到模型入库程序中,避免在模型加载时再进行这项工作,加快了模型加载速度^[12],但是却以大大增大数据库大小为代价(采用下面数据库设计的存储方式 2),并且会带来存储效率低下的问题。另外,模型的入库或者修改操作显然也会因此而耗费较多时间。事实上,因加载模型而带来的耗时操作是难以彻底避免的。为此,在初次加载时,系统引入 BackgroundWorker 组件来开辟一个单独的线程进行初始加载操作,同时主界面上以进度条的形式来大概地显示加载的进度。另外软件将内容编辑器生成的 .xnb 文件(包括图片属性信息不完整时生成的“文件名_0.xnb”文件,最多两个文件)移入程序的相对路径之内,并且不再将之删除(这样会牺牲一点磁盘空间,但与改善性能相比,这是值得的),当程序试图加载曾经生成过的内容文件时,就无需进行实际的处理工作了。这样就极大地加快了模型二次加载速度。图 4 是三维模型的加载流程。

基于上面的设计思路,对 ContentBuilder 类的修改及初次改进的关键代码如下:

```

public string Build()
{
    .....
}
  
```

```

string[] files = Directory.GetFiles(buildDirectory +
"\\\" + "bin\\Content");
foreach (string file in files)
{
    File.Copy(file, Path.Combine(Application.StartupPath,
"bin\\Content\\" + Path.GetFileName(file), true);
}
return null;
}

```

```

private void gridView1_RowClick(object sender,
DevExpress.XtraGrid.Views.Grid.RowClickEventArgs
e){
    .....
    backgroundWorker1.RunWorkerAsync(Application.StartupPath + "\\Content\\" + Program.modelStr); //初次加载
    .....
}

```

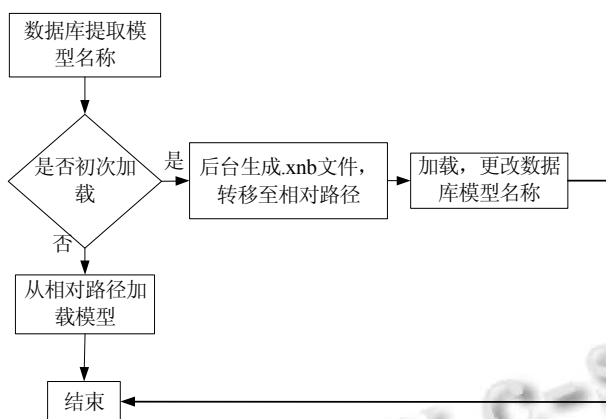


图4 三维模型加载流程图

2.4 数据库的设计

数据库设计^[13]是针对给定的应用环境, 设计最优的数据库模式, 使之能够有效地存储数据, 尽可能满足各种数据的应用要求. 合理的数据库设计对提高数据存储的效率, 保证数据的完整性和一致性具有重要意义. 系统采用 SQL SERVER 数据库技术来实现大批数据的存储与管理. 目前, 存储方式主要有以下两种: ①模型文件存储在指定的本地目录中, 将存储路径存入数据库; ②模型以二进制数据流方式存入数据库中, 作为 BLOB 字段值^[14]. 考虑到系统三维模型相对复杂,

如果直接将模型以二进制数据方式放入数据库, 容易导致数据库越来越庞大, 存储效率低, 对数据库性能造成潜在威胁. 因此, 系统采用第一种方式, 但是仅仅将模型名称记录到数据库中(初次加载过后, 模型名称将会以名称加 ID 的形式重新入库, 以此作为唯一标识), 而不是直接存入完整路径, 既减小了字段大小, 又提高了程序的可移植性, 调用时再获取模型相对路径.

3 模块运行及结果分析

为检验三维表达效果、用户交互情况及三维模型加载速度, 软件先在数据库中添加少量记录, 并对程序进行了测试. 首先看到左侧导航栏根据数据库中的国家名称动态加载了 NavBarControl 控件, 逐级展开任一选项, 数据表中按选项逐级做出了筛选, 实现了信息的分类管理.



图5 性能动画分析场景

在三维静态展示界面中, 点击一行记录, 数据对比图形做出了更改, 三维模型开始加载, 同时状态栏进度条展示进度(仅在初次加载时出现), 加载后的界面如图 2 所示, 图中以飞机的雷达探测距离与飞机航速的图表对比为例(具体对比项目我们可以自主选择). 由于程序直接将生成的 .xnb 文件移动到程序的相对路径之中, 而不是随着临时文件夹的消失而删除, 模型二次加载时直接从相对路径中加载文件, 速度明显加快, 耗时小于 0.1 秒(初次加载时间大约是 5.9 秒, 根据模型复杂程度不同, 该值会有所变化), 加载时间减少了 98%, 即便关闭程序, 再重新运行, 加载时间也依然不到 0.1 秒, 达到了预期目的.

图 5 展示了在三维动画分析时, 在蓝天白云的场景中两机按照数据库中相应的飞行速度进行动画, 最终相撞时的场景, 通过调用 CollidesWith 函数和模块 Font 完成了碰撞检测和汉字模块的绘制, 其中的格子型参考线是飞机距离的标尺. 在这个界面中, 可以通过左下角弹出的工具箱进行飞机初始属性以及距离的

设置,调整界面动画效果.工具箱初始设置界面如图6所示.同时在最下方随着飞机行进位置的变化,一些性能参数的变化情况也将以图表的形式动态展示出来,运行效果良好.



图 6 工具箱初始设置界面

4 结语

本系统通过自定义控件的方式,完成了在.NET 框架环境下应用XNA技术,实现了外军飞机性能分析三维可视化软件的构建和开发.本系统还对三维模型的动态加载技术进行了改进,并与其他已有的加载改进措施进行了对比.用户交互体验良好,达到了快速查看三维模型及并进行性能动画分析的目的.

参考文献

- 1 Benjamin N. Professional XNA Game Programming: For Xbox 360 and Windows. Indianapolis: Wrox Press, 2007.
- 2 李柯.高速列车虚拟仿真中三维特效的设计与实现.成都:西南交通大学,2013.
- 3 Grootjans R. XNA 2.0 game programming recipes: a problem-solution approach. Berkeley, California, USA. Apress Press. 2006. 1-25.
- 4 Microsoft Corporation Content pipeline.[2009-06-11]. <http://creators.xna.com>.
- 5 何中辰,刘金刚.基于鼠标跟踪的三维交互机制的研究.西北大学学报(自然科学版),2012,42(2):181-182.
- 6 Ferguson E, Rockhold B, Heck B. Video game development using XNA game studio and C# NET. Journal of Computing Science in Colleges, 2008, 23(4): 186-188.
- 7 乔卉,龚庆武,江传文.面向电力培训的三维交互仿真平台的设计与实现.电力自动化设备,2013,33(6):157-158.
- 8 黄子俊,向虎,汤锦棉,等.装备维修仿真训练设计与实现.计算机工程与设计,2012,33(12): 4677-4680.
- 9 Belyaev SY, Plotnikov M. Object-oriented high-performance particle system. Proc. of the 6th the International Workshop on Nondestructive Testing and Computer Simulation in Science and Engineering Bellingham, WA: SPIE, 2003. 272-275.
- 10 Reed A. Learning XNA 4.0.O'Reilly Media, 2010.12.
- 11 Nitschke B. Professional XNA Game Programming: For Xbox 360 and Windows. Indianapolis: WroxPress,2007.
- 12 张静,季民,崔先国,等.油样存储三维表达与标签动态加载方法研究.科学技术与工程,2012,12(6): 1422-1423.
- 13 杨国有.合同管理信息系统数据库设计.电子设计工程,2014,22(5): 1-2.
- 14 单春燕.应用 BLOB 技术实现图像资料的存储.中国管理信息化,2013,16(7): 88-89.