

doi: 10.7690/bgzd.2016.12.011

# 基于 MFC 软件界面设计的视觉优化

张弦弦<sup>1</sup>, 朱全松<sup>2</sup>

(1. 中国兵器装备集团自动化研究所武器装备信息与控制研发中心, 四川 绵阳 621000;

2. 中国兵器装备集团自动化研究所装药中心, 四川 绵阳 621000)

**摘要:** 针对基于 MFC 的软件界面设计存在色彩单一、控件呆板等问题, 采用加载皮肤、个性化自绘和嵌入 Flash 3 种方法进行视觉优化。介绍软件界面设计和微软基础类库, 详细分析软件界面的 3 种设计视觉优化方法的实现形式, 对比了各自的优缺点。实践证明: 视觉优化能够增强软件界面的美观性、友好性、多样性; 3 种方法的灵活运用, 使得软件界面设计的视觉优化方法条理化、系统化, 是有效可行的。

**关键词:** MFC; 界面设计; 视觉优化**中图分类号:** TP311.51 **文献标志码:** A

## Visual Optimization Based on MFC Software Interface Design

Zhang Xianxian<sup>1</sup>, Zhu Quansong<sup>2</sup>(1. *Research & Development Center of Weapon Equipment Information & Control, Automation Research Institute of China South Industries Group Corporation, Mianyang 621000, China;*2. *Center of Ammunition Charging, Automation Research Institute of China South Industries Group Corporation, Mianyang 621000, China*)

**Abstract:** In view of the problems of the MFC based software interface design, such as the humdrum color, the inflexible control machine and so on. Use loaded skin, personalized self painting, and embedded Flash three methods for visual optimization. This paper introduces the software interface design and the Microsoft foundation classes library, and analyzes the realization forms of the three visual optimization methods in detail, and compares their advantages and disadvantages. Practice has proves that: visual optimization can enhance the beauty, friendliness and diversity of the software interface. Flexibly use of three methods to make the visual optimization method of the software interface design is organized and systematically. It is feasible and effective.

**Keywords:** MFC; interface design; visual optimization

## 0 引言

熟悉 VC、MFC 的程序员经常会遇到这样的问题: 设计的软件功能完全实现, 但是界面极其难看。这是由于界面上的控件、工具栏、菜单、窗体等使用的都是 Windows 默认的元素<sup>[1]</sup>。软件设计可分为编码设计和用户界面设计 2 个部分。一个成功的软件, 不仅能实现用户所需的功能, 界面的美观性和友好性也是必不可少的重要因素。

## 1 软件界面设计

软件界面设计是对软件的用户界面进行美化、优化、规范化的设计。包括软件启动界面设计、框架设计、按钮设计、菜单设计、面板设计等。要求设计人员从用户的使用角度出发, 规范设计、合理布局软件界面, 确保人机交互界面具有风格一致、美观协调、操作便捷等特性<sup>[2]</sup>。

## 2 MFC 简介

微软基础类库 (microsoft foundation classes,

MFC), 以 C++ 类的形式封装了 Windows API(application programming interface), 即应用程序接口的大部分函数, 并提供一个面向对象的应用程序框架。程序人员可利用该框架创建 Windows 应用程序。MFC 的优点是效率高, 其包含的大量 Windows 句柄和内建控件封装类能有效地减少应用程序开发人员的工作量<sup>[3]</sup>。

## 3 软件界面设计的视觉优化

每个软件界面都有不同的视觉表现, 形、色、质相辅相成。用户视觉上的体验决定了一个软件的接受程度; 因此, 软件界面的视觉化设计成为软件界面设计的重中之重。使用 MFC 框架开发的应用程序界面在美观上不尽如人意。笔者在软件开发过程中总结以下 3 种优化软件界面视觉效果的方法。

### 3.1 整体优化

软件界面美化最简单的方式是加载皮肤, 以实现整个软件界面风格的改变。常使用 SkinMagic、

收稿日期: 2016-10-16; 修回日期: 2016-11-19

作者简介: 张弦弦(1980—), 女, 重庆人, 工学学士, 高工, 从事军工数字化技术研究。

Skin++、USkin 等皮肤库对软件整体风格进行调整。以 SkinMagic 为例，SkinMagic 是一个基于图片，为 C++ 设计的皮肤函数库。通过该库的 API 可快速添加绚丽的软件界面视觉效果。实现流程如图 1。

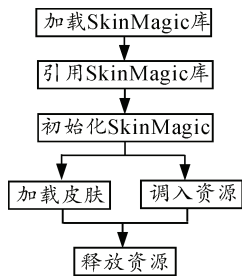
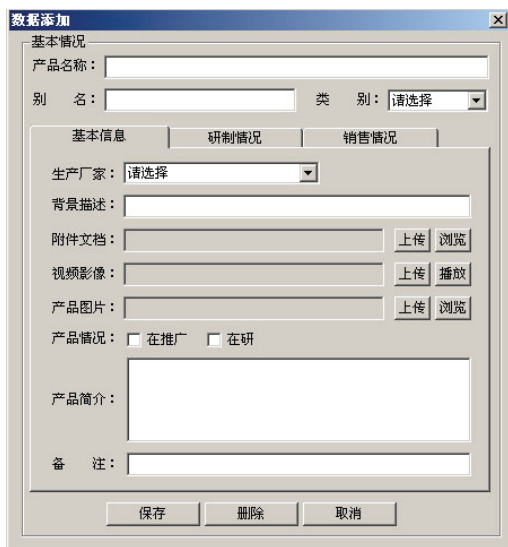


图 1 MFC 加载皮肤流程



(a) Windows 默认界面



(b) 加载 SkinMagic 后

图 2 加载皮肤前后对比图

1) 加载 SkinMagic 库。

把库中的头文件 SkinMagicLib.h、静态库文件 SkinMagic.lib、动态库文件 SkinMagic.dll 以及皮肤

文件 Skin.swf 添加到工程中。

2) 在 stdafx.h 中引用头文件和库。

```
#include "SkinMagicLib.h"
#pragma comment(lib, "SkinMagic.lib")
```

3) 初始化 SkinMagic 库。

```
BOOL CMainApp::InitInstance()[4]
{
    VERIFY( 1 == InitSkinMagicLib
(AfxGetInstanceHandle(), NULL, NULL, NULL)); //
初始化库
```

```
VERIFY( 1 == LoadSkinFile("Skin.smf")); //
载入皮肤文件
```

4) 加载皮肤。

```
BOOL CMainDlg::OnInitDialog()
{
    SetWindowSkin(m_hWnd, "Dialog"); // 窗口
加载皮肤
    SetDialogSkin("Dialog"); // 对话框加载皮肤
}
```

5) 释放资源。

```
BOOL CMainApp::ExitInstance()
{
    ExitSkinMagicLib();
}
```

加载皮肤前后的界面对比如图 2 所示。

3.2 个性化界面自绘

虽然软件界面美化不是 MFC 的强项，但仍可通过一些途径来实现<sup>[5]</sup>，如通过 picture 控件加载图片，自绘控件等。常用方法有：

1) 使用 MFC 的既有函数来设置界面，如 SetDialogBkColor 调整对话框的背景色；SetTextColor 设置文本颜色；SetBkImage 设定 list 控件的背景图片等。

2) 通过 MFC 的消息机制，在 Windows 绘制界面上的元素时截获消息，修改元素的状态和行为。可用 WM\_PAINT 消息进行窗口的重绘；WM\_ERASEBKGDND 更改窗口、控件的背景；WM\_CTLCOLOR 改变控件的颜色；WM\_DRAWITEM 改变元素的界面框架；WM\_CUSTOMDRAW 进行自定义绘图等<sup>[6-8]</sup>。

以下代码实现对话框中通过 WM\_PAINT 消息进行图片调用的功能<sup>[9-10]</sup>。

```
BOOL CPicDlg::OnInitDialog()
{
    if (m_bmp.m_hObject != NULL)
```

```

m_bmp.DeleteObject();
    HBITMAP hbmp = (HBITMAP)::
LoadImage(AfxGetInstanceHandle(),m_picture_filena
me,IMAGE_BITMAP,500,400,LR_CREATEDIBSEC
TION|LR_LOADFROMFILE); //加载图片
    m_bmp.Attach(hbmp);
    //显示图片
    CRect rect;
    GetDlgItem(IDC_STATIC)->GetWindowRect(&r
ect);//得到窗口的坐标
    ScreenToClient(&rect);
    GetDlgItem(IDC_STATIC)->MoveWindow(r
ect.left,rect.top,500,400,TRUE);//调整大小
}
void CPicDlg::OnPaint()
{
    CPaintDC dc(GetDlgItem(IDC_STATIC));//
定义需要重绘的控件
    CRect rcclient;
    CBrush brush;//定义画刷
    brush.CreatePatternBrush(&m_bmp);// 创建
画刷
    dc.FillRect(rcclient,&brush);
    dc.BitBlt(rcclient.left,rcclient.top,rcclient.Wi
dth(),rcclient.Height(),&memdc,rcclient.left,rcclient.t
op,SRCPAINT);//调用图片
}

```

3) 派生界面元素。创建继承 MFC 标准元素的新元素，对该元素进行自定义和重绘。

通过 MFC 个性化界面自绘方法得到的界面如图 3 所示。



图 3 个性化界面自绘

### 3.3 嵌入 Flash

Flash 动画在许多领域中有着举足轻重的作用，

其强大的视觉冲击力能在最短时间内吸引人的眼球。应用程序中嵌入 Flash，使软件变得生动活泼，增强人机界面的友好性，给用户留下深刻的印象<sup>[11]</sup>。可在 MFC 中嵌入 Shockwave Flash Object 控件实现软件中 Flash 动画的播放。MFC 嵌入 Flash 流程如图 4 所示。

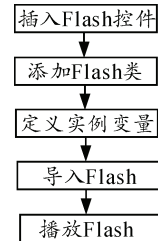


图 4 MFC 嵌入 Flash 流程

- 1) 工程中加入 Shockwave Flash Object 控件。
- 2) 在对话框中添加 CShockwaveFlash 类。
- 3) 定义 CShockwaveFlash 类的实例，如：CShockwaveFlash m\_flash。

4) 使用成员函数 LoadMovie、Play 来导入、播放 Flash。其部分代码如下：

```

CFlashDlg::OnInitDialog()
{
    m_flash.Create(NULL,WS_CHILD|WS_VISIBL
E,CRect(0,0,500,400),this,"");//创建实例
    m_flash.LoadMovie(0," e://flash.swf" );//导入
Flash
    m_flash.Play();//播放 Flash
}

```

### 4 结束语

3 种界面设计的视觉优化方法各有优缺点：整体优化加载皮肤简单快捷，但是千篇一律，并且修改皮肤比较困难；个性化界面自绘功能强大，能随心所欲地设计软件界面，但需要程序员具有较强的审美能力和界面设计水平，要了解 MFC 框架的基本运作原理和熟悉 Windows 消息机制，具有较强的 VC 编程能力；嵌入 Flash 动画能实现界面的动态性和绚丽性，但需要程序员具备一定的美工基础和 Flash 设计水平。

在实际应用中，3 种方法的灵活运用以及结合运用，有效地解决了应用软件界面的色彩单一、控件呆板等问题，大幅度地美化了用户界面，展示了软件的多样性和个性设计。