**Win32程序快速入门指南**

1. **程序说明**

示例程序放在Win32ShapeOrg中

* 1. \_tWinMain

\_tWinMain是程序入口。

while (GetMessage(&msg, NULL, 0, 0))开始为消息处理循环。

如果程序运行到此处将进入一个消息响应过程，即如果有消息就会进入消息响应函数

*LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)*

* 1. 全局初始化

如果要做全局初始化可以在tWinMain函数中while (GetMessage(&msg, NULL, 0, 0))前

* 1. 消息响应机制

win32程序是基于消息响应的，最核心的模块是消息响应函数

*LRESULT CALLBACK WndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam)*

消息是依附在某个窗口的。其中hWnd是窗口句柄，windows程序里，每一个窗口都有一个HWND类型的句柄用于标识这个窗口。

message是UINT类型的消息，实质上整数，消息的其它信息包含在wParam和lParam中

* 1. 绘制函数

WM\_PAINT是绘制消息，所有和绘制相关的代码都放在WM\_PAINT消息响应部分（具体在hdc = BeginPaint(hWnd, &ps);和EndPaint(hWnd, &ps);之间），win32所有绘制函数都带有一个HDC类型设备上下文句柄的参数。

*InvalidateRect(hWnd, NULL, true);*语句会发出绘制消息。如果需要更新绘制画面，就可以调用此语句。

Windows绘制机制的基础是图像设备交互([GDI](https://msdn.microsoft.com/en-us/library/dd145203%28v%3Dvs.85%29.aspx)，Graphics Device Interface)。

Brush是用来填充的刷子，绘制的东西是实心的。Pen相当于画笔，用来描述绘制直线曲线时的颜色粗细样式等等。如果不做设置，系统会提供默认的设置。

[直线和曲线函数](https://msdn.microsoft.com/en-us/library/dd145031%28v%3Dvs.85%29.aspx)在这里，这是一个[直线段的例子](https://msdn.microsoft.com/en-us/library/dd162488%28v%3Dvs.85%29.aspx)。

*//显示文字*

 *char cMessage[128];*

 *sprintf(cMessage, "%d, %d", g\_xPos, g\_yPos);*

 *SetBkMode(hdc, TRANSPARENT); //设置背景透明显示模式*

 *TextOut(hdc, g\_xPos+10, g\_yPos-10, cMessage, strlen(cMessage));*

 *//实心椭圆*

 *int r = 9;*

 *Ellipse(hdc, g\_xPos-r, g\_yPos-r, g\_xPos+r, g\_yPos+r); //当前位置坐标&位置+直径：*

 *//空心椭圆*

*MoveToEx(hdc, g\_xPos+r, g\_yPos, 0);*

 *AngleArc(hdc, g\_xPos, g\_yPos, r, 0, 360);*

 *//画线*

* 1. 一些有用的消息

[消息和消息队列](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644927%28v%3Dvs.85%29.aspx#system_defined)，Message Types里的System-Defined Messages中的WM，列出了各种类型的消息。

![C:\Users\ShaorongWang\AppData\Roaming\Tencent\Users\2101816208\QQ\WinTemp\RichOle\0WW6V$9WQZO]~()9(Y}RXDX.png]()

比较常见的消息包括:

* [鼠标](https://msdn.microsoft.com/en-us/library/windows/desktop/ff468877%28v%3Dvs.85%29.aspx)

[左键点击](https://msdn.microsoft.com/en-us/library/windows/desktop/ms645607%28v%3Dvs.85%29.aspx),WM\_LBUTTONDOWN，其它[各种消息](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644927%28v%3Dvs.85%29.aspx#system_defined)

获取鼠标位置：

 *int xPos, yPos;*

 *xPos = GET\_X\_LPARAM(lParam);*

 *yPos = GET\_Y\_LPARAM(lParam);*

为了使用GET\_X\_LPARAM和GET\_Y\_LPARAM需要在加上头文件Windowsx.h

* [键盘](https://msdn.microsoft.com/en-us/library/windows/desktop/ff468861%28v%3Dvs.85%29.aspx)

[某个键按下](https://msdn.microsoft.com/en-us/library/windows/desktop/ff468861%28v%3Dvs.85%29.aspx)WM\_KEYDOWN，按键信息在 wParam中

*switch(wParam)//看看是哪个键*

*{*

*case 'A':*

*case 'a':*

 *break;*

*case 'S':*

*case 's':*

 *break;*

*}*

* [时钟](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644902%28v%3Dvs.85%29.aspx)

 [WM\_TIMER](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644902%28v%3Dvs.85%29.aspx)，启动一个时钟的函数[SetTimer](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644906%28v%3Dvs.85%29.aspx)，这里是[例子](https://msdn.microsoft.com/en-us/library/windows/desktop/ms644901%28v%3Dvs.85%29.aspx#creating_timer)

 示例中加在程序初始化部分，消息循环之前。

*SetTimer(hWnd, 1, 30, NULL);//每隔30毫秒激活一次时钟响应*

此语句要用到窗口句柄，放在BOOL InitInstance(HINSTANCE hInstance, int nCmdShow)中。

* [WM\_SIZE](https://msdn.microsoft.com/en-us/library/windows/desktop/ms632646%28v%3Dvs.85%29.aspx)

 窗口大小改变*，*是Window Notifications系列的消息。获取窗口尺寸示意代码

 *int nWidth, nHeight;*

 *nWidth = LOWORD(lParam); // width of client area*

 *nHeight = HIWORD(lParam); // height of client area*

* 1. 关于画面的特别讨论

Win32用于动画游戏绘制时，不加特殊处理会出现闪烁情况。可以使用双缓冲技术来解决。可以[参考](http://www.cnblogs.com/wb-DarkHorse/p/3154387.html)，在hdc = BeginPaint(hWnd, &ps);和EndPaint(hWnd, &ps);之间使用以下代码

![C:\Users\ShaorongWang\AppData\Roaming\Tencent\Users\2101816208\QQ\WinTemp\RichOle\R0KFRCM}P~JE6NGAT_S_]YH.png]()

此外还要响应WM\_ERASEBKGND消息，直接return FALSE;即可。

1. **程序封装**

可以对程序进行接口封装处理。示例程序放在Win32Shape中。

*void UserInit(); //用户初始化函数*

*void UserFinish(); //用户清理函数*

*bool UserWndProc(HWND hWnd, UINT message, WPARAM wParam, LPARAM lParam);//消息响应函数*

*void UserRender(HDC hdc); //用户绘制函数*

用户添加的程序不再放在VS自动生成的主程序中。