

C语言API编写窗体界面和按钮

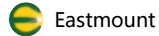
原创 Eastmount 最后发布于2014-05-25 02:43:50 阅读数 11082 ☆ 收藏

展开



Python+TensorFlow人工智能

该专栏为人工智能入门专栏，采用Python3和TensorFlow实现人工智能相关算法。前期介绍安装流程、基础语法...



¥9.90

去订阅

最近有个同学的程序需要用对话框的方式实现,但前面都是通过黑框形式完成的,老师突然让增加一个界面,本来准备采用MFC完成的,但后来一想,该程序核心东西是体现在它的算法上,控制台的程序并非不好.转念一想,如果使用MFC这样的方法好像需要再倒腾一遍,想到了大一在"C语言能干大事"中的东西,所以就准备采用这种形式直接完成,投机取巧的方法.

主要是通过调用Windows API界面的CreateWindows函数完成,同时也能创建按钮,把参数设置为button,如果想响应该按钮,只需在回调函数中增加消息WM_COMMAND判断即可.代码如下:

```
#include<windows.h>
#include<stdio.h>

// 声明回调函数
LRESULT CALLBACK WndProc (HWND, UINT, WPARAM, LPARAM) ;

// 主函数 程序入口
int WINAPI WinMain (HINSTANCE hInstance, HINSTANCE hPrevInstance,
                    PSTR szCmdLine, int iCmdShow)
{
    static TCHAR szAppName[] = TEXT ("HelloWin") ;
    HWND        hwnd ;           // 用来保存成功创建窗口后返回的句柄
    MSG          msg ;           // 定义消息结构体变量
    WNDCLASS      wndclass ;      // 窗体类

    wndclass.style          = CS_HREDRAW | CS_VREDRAW ; // 指定窗口风格
    wndclass.lpfnWndProc    = WndProc ; // 函数指针, 指向处理窗口消息的函数入口
    wndclass.cbClsExtra     = 0 ;       // 结构体后附加的字节数, 一般总为0
    wndclass.cbWndExtra     = 0 ;       // 窗体实例附加的字节数, 一般总为0
    wndclass.hInstance      = hInstance ; // 模块句柄
    wndclass.hIcon          = LoadIcon (NULL, IDI_APPLICATION) ; // 图标句柄 任务栏显示的图标
    wndclass.hCursor        = LoadCursor (NULL, IDC_ARROW) ; // 光标句柄
    wndclass.hbrBackground  = (HBRUSH)GetStockObject(WHITE_BRUSH); // 背景颜色COLOR_BACKGROUND
    wndclass.lpszMenuName   = NULL ;    // 菜单名的字符串
    wndclass.lpszClassName = szAppName ; // 自定义类名, 不要与其他类名重复

    if (!RegisterClass (&wndclass))
    {
        MessageBox (NULL, TEXT ("注册类失败!"), szAppName, MB_ICONERROR) ;
        return 0 ;
    }

    int x = ((GetSystemMetrics(SM_CXSCREEN)/2)-200); //x居中
    int y = ((GetSystemMetrics(SM_CYSCREEN)/2)-200); //y居中

    // 创建窗体API
    hwnd = CreateWindow(szAppName,TEXT("毕业设计"),WS_CAPTION|WS_SYSMENU|WS_MINIMIZEBOX,x,y,400,400,NULL,NULL,hInstance,NULL);
    // 显示窗体的API 传入需要显示的窗体句柄和显示方式 ShowWindow(hwnd,iCmdShow);
    // 刷新窗体的API
    UpdateWindow(hwnd);

    // 从系统的应用程序线程消息队列中取得一个消息
    while(GetMessage(&msg,NULL,0,0) > 0)
```

```

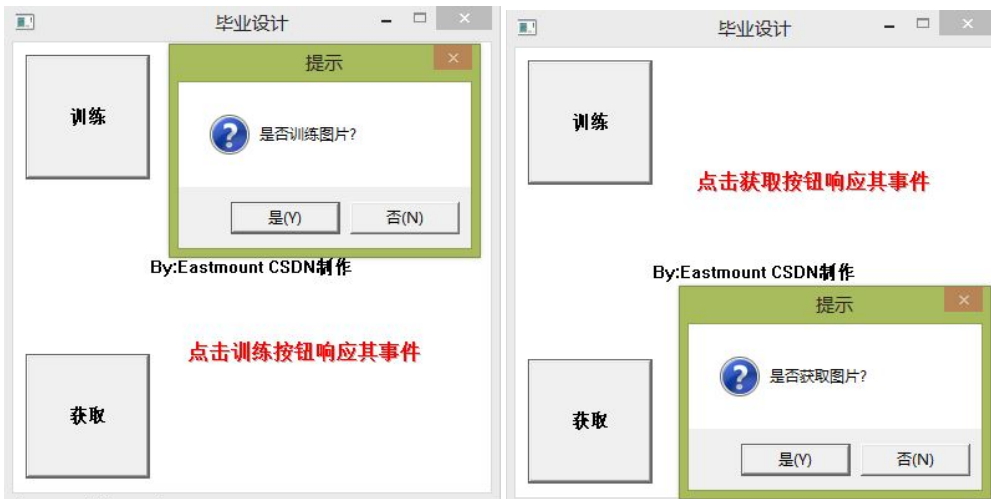
        {
            DispatchMessage(&msg);
        }
        return msg.wParam;
    }
}

LRESULT CALLBACK WndProc (HWND hwnd, UINT message, WPARAM wParam, LPARAM lParam)
{
    HDC                hdc;                // 句柄
    PAINTSTRUCT        ps;
    RECT               rect;               // 矩形
    HINSTANCE          hInstance;         // 窗口实例
    static HWND        hwndButton[2];     // 按钮句柄

    switch (message)
    {
        case WM_CREATE: // 创建按钮
        {
            hInstance = ((LPCREATESTRUCT)lParam)->hInstance;
            // 按钮1
            hwndButton[0] = CreateWindow("BUTTON", "训练",
                WS_TABSTOP | WS_VISIBLE | WS_CHILD | BS_DEFPUSHBUTTON,
                10, 10, 100, 100, hwnd, NULL,
                (HINSTANCE)GetWindowLong(hwnd, GWL_HINSTANCE), NULL);
            // 按钮2
            hwndButton[1] = CreateWindow("BUTTON", "获取",
                WS_TABSTOP | WS_VISIBLE | WS_CHILD | BS_DEFPUSHBUTTON,
                10, 250, 100, 100, hwnd, NULL,
                (HINSTANCE)GetWindowLong(hwnd, GWL_HINSTANCE), NULL);
            return 0;
        }
        case WM_PAINT: // 绘制文字
        {
            hdc = BeginPaint(hwnd, &ps);
            GetClientRect(hwnd, &rect);
            DrawText(hdc, TEXT("By:Eastmount CSDN制作"), -1, &rect, DT_SINGLELINE | DT_CENTER | DT_VCENTER);
            EndPaint(hwnd, &ps);
            return 0;
        }
        case WM_COMMAND: // 响应按钮消息
        {
            if((HWND)lParam == hwndButton[0])
            {
                MessageBox(NULL, TEXT("是否训练图片?"), TEXT("提示"), MB_YESNO | MB_ICONQUESTION);
            }
            if((HWND)lParam == hwndButton[1])
            {
                MessageBox(NULL, TEXT("是否获取图片?"), TEXT("提示"), MB_YESNO | MB_ICONQUESTION);
            }
            return 0;
        }
        case WM_CLOSE: // 关闭
        {
            if(IDYES==MessageBox(hwnd, "是否关闭程序?", "提示", MB_YESNO | MB_ICONQUESTION))
            {
                DestroyWindow (hwnd);
            }
            return 0;
        }
        case WM_DESTROY: // 退出程序
        {
            PostQuitMessage (0) ;
            return 0 ;
        }
    }
    return DefWindowProc (hwnd, message, wParam, lParam) ;
}

```

程序运行结果如下图所示:



其中主要涉及到的函数原型如下所示:

```
// 创建窗体
hwnd = CreateWindow (
    szClassName,          /* Classname */
    "Windows App",        /* Title Text */
    WS_OVERLAPPEDWINDOW,  /* default window */
    CW_USEDEFAULT,        /* Windows decides the position */
    CW_USEDEFAULT,        /* where the window ends up on the screen */
    544,                  /* The programs width */
    375,                  /* and height in pixels */
    HWND_DESKTOP,         /* The window is a child-window to desktop */
    NULL,                 /* No menu */
    hThisInstance,        /* Program Instance handler */
    NULL                  /* No Window Creation data */
);

// 创建按钮
hwndButton = CreateWindow(
    "BUTTON",             // predefined class
    "OK",                 // button text
    WS_VISIBLE | WS_CHILD | BS_DEFPUSHBUTTON, // styles
    10,                   // starting x position
    10,                   // starting y position
    100,                  // button width
    100,                  // button height
    hwnd,                 // parent window
    NULL,                 // No menu
    (HINSTANCE) GetWindowLong(hwnd, GWL_HINSTANCE),
    NULL                  // pointer not needed
);
```

最后,写这篇文章主要是怀念自己大一时的生活,从一个什么都不知道的孩子,通过学习C语言,C语言能干大事开始接触编程.同时,我认为这个程序也是非常还的入门程序,希望刚接触程序的同学也可以看看,编编自己感兴趣的程序、写写博客、AC题目、编写游戏、聊天软件、移动开发,能从程序和生活中找到一些让自己心灵美妙的东西.

(By:Eastmount 2014-5-25 夜2点半 原创CSDN<http://blog.csdn.net/eastmount/>)

👍 点赞 10 ☆ 收藏 🔄 分享 ...



Eastmount 博客专家

发布了450 篇原创文章 · 获赞 6333 · 访问量 502万+

他的留言板

关注