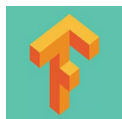


# C# 系统应用之无标题窗体移动的两种方法

原创 Eastmount 最后发布于2014-03-07 17:56:10 阅读数 6092 ☆ 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...



Eastmount

¥9.90

去订阅

在做项目界面设计中,常常为了美观需要设置窗体属性"FormBorderStyle"(窗体边框和标题栏外观)为None无标题窗口.此时隐藏标题的窗口怎样实现移动呢?我根据自己的项目从自己完成的两种方法进行讲解.

## 一.MouseDown\MouseMove事件方法实现

第一种方法我采用常用的根据MouseDown、MouseMove、MouseUp、MouseLeave事件中鼠标移动的距离,窗体坐标也跟着移动的方法.具体代码如下:

```
bool formMove = false;           // 是否开始移动
int currentXPosition = 0;         // 当前鼠标X坐标
int currentYPosition = 0;         // 当前鼠标Y坐标

// 函数: 鼠标按下
private void WelcomeWindow_MouseDown(object sender, MouseEventArgs e)
{
    formMove = true;               // 鼠标按下开始移动
    currentXPosition = MousePosition.X; // 鼠标的X坐标为当前窗体左上角X坐标
    currentYPosition = MousePosition.Y; // 鼠标的Y坐标为当前窗体的左上角Y坐标
}

// 函数: 鼠标移动
private void WelcomeWindow_MouseMove(object sender, MouseEventArgs e)
{
    if (formMove)
    {
        // 鼠标xy坐标确定窗体XY坐标 鼠标移动XY距离
        this.Left += MousePosition.X - currentXPosition;
        this.Top += MousePosition.Y - currentYPosition;
        // 鼠标当前位置赋值
        currentXPosition = MousePosition.X;
        currentYPosition = MousePosition.Y;
    }
}

// 函数: 鼠标松开
private void WelcomeWindow_MouseUp(object sender, MouseEventArgs e)
{
    formMove = false;             // 停止移动
}

// 函数: 鼠标离开
private void WelcomeWindow_MouseLeave(object sender, EventArgs e)
{
    // 初始状态
    currentXPosition = 0;
    currentYPosition = 0;
    formMove = false;
}
```

同时有的设置变量Point formPoint通过Offset和Location变换位置亦类似.但是第一种方法可能由于每次获取鼠标坐标,移动时如果鼠标不在窗口位置会产生

窗体总是向屏幕外移动的现象,因此推荐第二种调用API函数的方法.

## 二.调用API函数实现

第二种方法是采用API函数ReleaseCapture和SendMessage来实现的,ReleaseCapture函数从当前线程中窗口释放鼠标捕获,SendMessage函数的作用是调用一个窗口的窗口函数,将一条消息命令发给那个窗口.

```
SendMessage(  
IntPtr hwnd, //要接收消息的那个窗口的句柄  
int wMsg,    //消息的标识符  
int wParam, //具体取决于发送的消息  
int lParam  //具体取决于发送的消息  
)
```

其中调用API函数需要引用命名空间System.Runtime.InteropServices.代码中WM\_SYSCOMMAND常量代表要向窗口发送消息,SC\_MOVE常代表要向窗口发送移动的消息.代码如下:

```
#region 鼠标移动操作  
// 该函数从当前线程中窗口释放鼠标捕获  
[DllImport("user32.dll")]  
public static extern bool ReleaseCapture();  
  
// 发送消息移动窗体  
[DllImport("user32.dll")]  
public static extern bool SendMessage(IntPtr hwnd, int wMsg, int wParam, int lParam);  
public const int WM_SYSCOMMAND = 0x0112;    // 向窗口发送消息  
public const int SC_MOVE = 0xF010;          // 向窗口发送移动的消息  
public const int HTCAPTION = 0x0002;  
  
// 鼠标位于窗体(底部位置)按下移动操作  
private void EMSecure_MouseDown(object sender, MouseEventArgs e)  
{  
    ReleaseCapture();  
    SendMessage(this.Handle, WM_SYSCOMMAND, SC_MOVE + HTCAPTION, 0);  
}  
#endregion
```

其中,EMSecure是对话框Form,当鼠标点击到图片或控件Panel时不会响应移动,所以对需要移动的控制也响应该API函数即可.如我的程序界面Panel1位于窗体顶部,故也响应该移动事件,即panel1\_MouseDown(object sender, MouseEventArgs e).运行结果如下图所示,红色部分可以实现移动(顶部为Panel控件):



<http://blog.csdn.net/Eastmount>

提示如果是WPF可以响应Window\_MouseLeftButtonDown(object sender, MouseButtonEventArgs e){this.DragMove();}实现窗体移动.最后希望该文章对大家有所帮助,如果有错误或不足之处,请海涵.

(By:Eastmount 2014-3-7 夜2点 原创:<http://blog.csdn.net/eastmount>)

👍 点赞 1    ☆ 收藏    ➦ 分享    ...



Eastmount    博客专家

发布了450 篇原创文章 · 获赞 6227 · 访问量 496万+

他的留言板

关注