

GeorgioLa's Note

Note like code, code as note!

[C#]

[C#] WinForm 中如何實現「拖曳元件」

□ 四月 28, 2014 □ GEORGIOWAN □ 發表留言

作論文轉播字卡系統時，「字卡編輯器」的功能中需要用到拖曳元件的效果，

關於拖曳元件的概念，可分為三部份 **MouseDown**、**MouseMove** 與 **MouseUp**，

對於該元件或控制項實作這三樣事件，即可達到拖曳元件的效果！

=====

在作拖曳元件時，可先定義一些全域變數幫助拖曳時計算元件位置。

```
#region 移動字卡元件相關變數
private int _ControlPositionX, _ControlPositionY;    // 元件初始位置
private int _CursorInitialX = 0, _CursorInitialY = 0; // 滑鼠拖曳初始位置
private bool _isMoveState = false; // 移動狀態
#endregion
```

□

.(<https://georgiosky2000.files.wordpress.com/2014/04/111.png>).

[MouseDown]

```
#region MouseDown 紀錄滑鼠與元件位置
private void pictureBox_MouseDown(object sender, MouseEventArgs e)
{
    if (!_isMoveState) // 假如非移動狀態
    {
        _ControlPositionX = ((PictureBox)sender).Left; // 紀錄視窗初始 X
        _ControlPositionY = ((PictureBox)sender).Top; // 紀錄視窗初始 Y
        _CursorInitialX = Cursor.Position.X; // 紀錄滑鼠初始 X
        _CursorInitialY = Cursor.Position.Y; // 紀錄滑鼠初始 Y
        _isMoveState = true; // 設定為移動狀態
    }
}
#endregion
```

[_https://georgiosky2000.files.wordpress.com/2014/04/1111.png](https://georgiosky2000.files.wordpress.com/2014/04/1111.png)

[MouseMove]

```
#region MouseMove 移動元件運算
private void pictureBox_MouseMove(object sender, MouseEventArgs e)
{
    if (_isMoveState) // 若為移動狀態
    {
        int tX = Cursor.Position.X; // 暫存目前滑鼠位置 X
        int tY = Cursor.Position.Y; // 暫存目前滑鼠位置 Y
        // 舊視窗位置 + 滑鼠移動距離 = 新位置
        ((PictureBox)sender).Left = _ControlPositionX + (tX - _CursorInitialX); // 設定視窗新位置 X
        ((PictureBox)sender).Top = _ControlPositionY + (tY - _CursorInitialY); // 設定視窗新位置 Y
    }
}
#endregion
```

[_https://georgiosky2000.files.wordpress.com/2014/04/1112.png](https://georgiosky2000.files.wordpress.com/2014/04/1112.png)



[MouseUp]

```
#region MouseUp 結束移動元件
private void pictureBox_MouseUp(object sender, MouseEventArgs e)
{
    // 設定非拖曳狀態
    _isMoveState = false;
}
#endregion

#endregion
```

_(<https://georgiosky2000.files.wordpress.com/2014/04/1113.png>).

以上例子是以 PictureBox 為例的程式碼截圖，當然也可換成 Control 來符合更多控制項種類，另外，在實作拖曳實若控制項或元件有閃爍的情況，可藉由 DoubleBuffer 來減緩此種現象，DoubleBuffer 為雙重緩衝繪製的機制，最簡單的方法就是在該 Form 的屬性中，將DoubleBuffered 改選為 True 即可。

DoubleBuffered	True
Enabled	True

_(<https://georgiosky2000.files.wordpress.com/2014/04/1114.png>).

Done!

附上該方法參考網址(<http://blog.kerash.tw/2013/05/csharp-%E8%A6%96%E7%AA%97%E6%8B%96%E6%9B%B3-drag-window-form/>).

ADVERTISEMENT

廣告

AIRFRANCE 

PARIS TWD FROM **21,894**

LONDON TWD FROM **22,999**

ROME TWD FROM **23,223**

RTN ALL INC.

DEPARTING FROM TAIPEI

[BOOK NOW](#)

[SEE CONDITIONS](#)

REPORT THIS AD

AIRFRANCE 

PARIS TWD FROM **21,894**

LONDON TWD FROM **22,999**

ROME TWD FROM **23,223**

RTN ALL INC.

DEPARTING FROM TAIPEI

[BOOK NOW](#)

[SEE CONDITIONS](#)

REPORT THIS AD



□ C#、CONTROLS、DRAG、WINFORM、拖曳、拖曳元件、拖曳控制項

在WORDPRESS.COM寫網誌。佈景主題：NUCLEARE，發表者：CRESTAPROJECT。