

# 使用DirectX截屏

转载

缘梦逍遥 于 2011-08-31 10:05:20 发布 1516 收藏 2

分类专栏: [DirectX](#) 文章标签: [null](#) [image](#) [date](#) [file](#) [扩展](#)



[DirectX 专栏收录该内容](#)

4 篇文章 0 订阅

订阅专栏

网上有很多关于DirectX截屏的文章，但大都是屏幕截图，很少有窗口截图，本文则两者都涉及到，先讲如何截取整个屏幕，再讲如何截取某个窗口，其实二者的区别不大，只是某个参数的设置不同而已，最后我们还将扩展到任意区域的截图。

首先看一下截屏用到的函数，最核心的当然是D3DXSaveSurfaceToFile，先看下函数原型

```
HRESULT D3DXSaveSurfaceToFile(  
    LPCTSTR pDestFile,  
    D3DXIMAGE_FILEFORMAT DestFormat,  
    LPDIRECT3DSURFACE9 pSrcSurface,  
    CONST PALETTEENTRY * pSrcPalette,  
    CONST RECT * pSrcRect  
);
```

第一个参数是指向设备的指针，不多说啦

第二个参数是截图文件的类型，支持的类型还不少，主要有下面这些

BMP, JPG, TGA, PNG, DDS, PPM, DIB, HDR, PFM

这里我们使用BMP-即位图格式

第三个参数是指向Surface的指针，也就是保存了截图数据的表面

第四个参数是Surface的调色板，这里不使用，设置为NULL

最后一个参数是Surface的矩形区域，也就是我们可以只截取Surface上某一矩形区域的数据，其实截取全屏和截取窗口的差别也就在这个参数的设置上

其他的函数在下面会逐一讲解

现在来定义我们的截屏函数，首先我们需要一个设备指针，因为在DX中，任何操作都与设备密切相关，所以设备指针几乎是每个DX函数都要用到的参数，我们这个函数也不例外，其次需要一个窗口句柄，当我们截取窗口时，把窗口句柄传入，当我们截取整个屏幕时，直接传入NULL。最后我们需要一个字符串参数来指定截图对应的文件名，如下

```
1 BOOL ScreenShot(LPDIRECT3DDEVICE9 lpDevice, HWND hWnd, TCHAR* fileName)
```

详细步骤:

首先我们需要获取显示模式，注意这里获取的是显卡的显示模式，而不是设备的显示模式，因为设备的显示模式既有窗口模式，也有全屏模式，所以它的分辨率是不确定的，而显卡的显示模式返回的始终是最大分辨率，我们需要创建整个屏幕区域对应的Surface，当截取真个屏幕时，直接保存即可，当截取窗口时，我们将窗口所对应的区域保存即可

获取显卡显示模式的代码如下

```
HRESULT hr;

// Get adapter display mode
D3DDISPLAYMODE mode;
if (FAILED(hr = lpDevice->GetDisplayMode(0, &mode)))
    return hr;
```

下面开始创建表面，这个表面是对应整个屏幕的

```
// Create the surface to hold the screen image data
LPDIRECT3DSURFACE9 surf;
if (FAILED(hr = lpDevice->CreateOffscreenPlainSurface(mode.Width,
    mode.Height, D3DFMT_A8R8G8B8, D3DPOOL_SYSTEMMEM, &surf, NULL))) //注意第四个参数不能是D3DPOOL_DEFAULT
{
    return hr;
}
```

接下来获取屏幕对应的数据，这个函数实际上是将显存中的数据拷贝到系统内存中

```
// Get the screen data
if (FAILED(hr = lpDevice->GetFrontBufferData(0, surf)))
{
    surf->Release();
    return hr;
}
```

接下来我们判断是截取窗口还是截取屏幕，很简单，只需判断hWnd是否为NULL即可，如果是截取窗口则设置窗口对应的矩形区域即可

```
// area to capture
RECT *rect = NULL;

WINDOWINFO windowInfo;
windowInfo.cbSize = sizeof(WINDOWINFO);

if(hWnd) // capture window
{
    GetWindowInfo(hWnd, &windowInfo);
    rect = &windowInfo.rcWindow;
}
```

最后一部，保存截图！

```
// Save the screen data to file  
hr = D3DXSaveSurfaceToFile(fileName, D3DXIFF_BMP, surf, NULL, rect);  
  
surf->Release() ;  
  
return hr ;
```

大功告成！

完整代码

```

BOOL ScreenShot(LPDIRECT3DDEVICE9 lpDevice, HWND hWnd, TCHAR* fileName)
{
    HRESULT hr;

    // Get adapter display mode
    D3DDISPLAYMODE mode;
    if (FAILED(hr = lpDevice->GetDisplayMode(0, &mode)))
        return hr;

    // Create the surface to hold the screen image data
    LPDIRECT3DSURFACE9 surf;
    if (FAILED(hr = lpDevice->CreateOffscreenPlainSurface(mode.Width,
        mode.Height, D3DFMT_A8R8G8B8, D3DPOOL_SYSTEMMEM, &surf, NULL))) //注意第四个参数不能是D3DPOOL_DEFAULT
    {
        return hr;
    }

    // Get the screen data
    if (FAILED(hr = lpDevice->GetFrontBufferData(0, surf)))
    {
        surf->Release() ;
        return hr ;
    }

    // area to capture
    RECT *rect = NULL ;

    WINDOWINFO windowInfo ;
    windowInfo.cbSize = sizeof(WINDOWINFO) ;

    if(hWnd) // capture window
    {
        GetWindowInfo(hWnd, &windowInfo) ;
        rect = &windowInfo.rcWindow ;
    }

    // Save the screen data to file
    hr = D3DXSaveSurfaceToFile(fileName, D3DXIFF_BMP, surf, NULL, rect);

    surf->Release() ;

    return hr ;
}

```

那么如何实现任意区域截屏呢，我想大家已经想到了，假设使用鼠标拖拽的方法截图，记下鼠标按下和抬起时的坐标，构造一个RECT，然后传递给 D3DXSaveSurfaceToFile函数就可以了，需要注意到是，由于鼠标拖拽到方向是任意的，所以在构造RECT的时候要注意right < left或者bottom < top 的情况，用下面的方法可以处理

```
int left = 0 ;
int right = 0 ;
int top = 0 ;
int bottom = 0 ;
RECT rect ;

case WM_LBUTTONDOWN:
    left = ( short )LOWORD( lParam );
    top = ( short )HIWORD( lParam );
    break ;

case WM_LBUTTONUP:
    right = ( short )LOWORD( lParam );
    bottom = ( short )HIWORD( lParam );

    rect.left = min(left, right) ;
    rect.right = max(left, right) ;
    rect.top = min(top, bottom) ;
    rect.bottom = max(top, bottom) ;
    // 调用截图函数
```

(文/AutumnWinter)

来自: <http://www.pin5i.com/showtopic-26129.html>