

Organ-Field GUI Help

0.1 概述

Chapter 1 Organ-Field Button(OFB)

1.0 引用与类

目前 Organ-Field GUI 没有开发完成,因此在使用前需手动在你的项目中添加"OFButton.cpp"与"OFButton.h"文件.

使用 Organ-Field Button 前需要引用头文件: `#include "OFButton.h"`

使用 Organ-Field Button 前需要引用类名: `OFButton OFB;`

1.1 宏

- 在创建按钮时,需要指定按钮的状态,按钮共有 5 种状态,以下是按钮状态宏.

宏 Macro	值 Value	含义 Meaning
<code>normal</code>	0	正常状态: 表示按钮是可交互的, 并且可用的
<code>disable</code>	1	禁用状态: 表示当前组件处于非交互状态, 但是之后可以被启用
<code>hover</code>	2	悬停状态: 当用户使用光标或者其他的元素, 置于其上方的时候, 显示这样的状态(鼠标划过)
<code>click</code>	3	激活状态: 表示用户已经按下按钮, 且还未结束按按钮的动作(鼠标点击)
<code>loading</code>	4	加载状态: 表示操作正在加载中, 组件正在反映, 但是操作还未完成

1.2 函数

1.2.1 CreateButton

- 这个函数用于创建一个按钮

```
void CreateButton(
    string ButtonName,
    int x, int y,
    int width, int height,
    short SetState,
    void ClickEvent()
);
```

- 参数

< ButtonName > 设置按钮名称
 < x,y > 指定按钮坐标
 < width > 设置按钮宽度
 < height > 设置按钮高度
 < SetSate > 设置按钮状态(此处填写按钮状态宏)
 < ClickEvent() > 指定要执行的事件(此处填写函数名)

1.2.2 ButtonDefaultStyle

- 按钮样式快速设置函数,此函数令所有按钮使用同一套已经预设好的样式,不再需要自己进行复杂的按钮样式设置.

```
void ButtonDefaultStyle();
```

1.2.3 SetButtonStyle Series

- 这一系列的函数用于自定义设置各按钮状态的样式(不能与 1.2.2 ButtonDefaultStyle 同时使用)

```
//设置正常状态按钮的样式
void SetNormalStyle(
    short buttonrim,
    COLORREF rimcolor,
    short fillstyle,
    short fillhatch,
    COLORREF fillcolor,
    COLORREF textcolor
);
```

```
//设置禁用状态按钮的样式
void SetDisableStyle (
    short buttonrim,
    COLORREF rimcolor,
    short fillstyle,
    short fillhatch,
    COLORREF fillcolor,
    COLORREF textcolor
);
```

```
//设置悬停状态按钮的样式
void SetHoverStyle (
    short buttonrim,
    COLORREF rimcolor,
    short fillstyle,
    short fillhatch,
    COLORREF fillcolor,
    COLORREF textcolor
);
```

```
//设置激活状态按钮的样式
void SetClickStyle (
    short buttonrim,
    COLORREF rimcolor,
    short fillstyle,
    short fillhatch,
    COLORREF fillcolor,
    COLORREF textcolor
);
```

//设置加载状态按钮的样式

```
void SetLoadingStyle (
    short buttonrim,
    COLORREF rimcolor,
    short fillstyle,
    short fillhatch,
    COLORREF fillcolor,
    COLORREF textcolor
);
```

● 填充样式

宏 Macro	含义 Meaning
BS_SOLID	固实填充
BS_NULL	不填充
BS_HATCHED	图案填充
BS_PATTERN	自定义图案填充
BS_DIBPATTERN	自定义图像填充



● 参数

- < buttonrim > 设置按钮边框线粗细
- < rimcolor > 设置按钮边框线颜色
- < fillstyle > 设置按钮填充样式
- < fillhatch > 指定填充图案
- < fillcolor > 指定填充颜色
- < textcolor > 指定按钮字体颜色

● 颜色

```
COLORREF RGB(
    BYTE byRed,    //颜色的红色部分
    BYTE byGreen,  //颜色的绿色部分
    BYTE byBlue    //颜色的蓝色部分
);
//颜色值范围: 0~255
```

● 填充图案

宏 Macro	含义 Meaning
HS_HORIZONTAL	 填充横线
HS_VERTICAL	 填充竖线
HS_FDIAGONAL	 填充斜线
HS_BDIAGONAL	 填充反斜线
HS_CROSS`	 填充网格线
HS_DIAGCROSS	 填充斜网格线

● 示例代码

```
#include "Window.h"
#include "OFButton.h"

Window win;
OFButton OFB;

int main() {
    //初始化绘图窗口(窗口宽度,窗口高度,显示控制台);
    win.Initialize_Window(1200, 700, EW_SHOWCONSOLE);

    //设置各按钮状态的样式(按钮边框线粗细,边框线颜色,指定填充样式,指定填充图案,指定填充颜色,指定按钮字体颜色);
    OFB.SetNormalStyle(1, RGB(180, 180, 180), BS_SOLID, NULL, RGB(225, 225, 225), RGB(30, 30, 30));
    OFB.SetDisableStyle(1, RGB(191, 191, 191), BS_SOLID, NULL, RGB(204, 204, 204), RGB(130, 130, 130));
    OFB.SetHoverStyle(1, RGB(180, 180, 180), BS_SOLID, NULL, RGB(229, 243, 255), RGB(30, 30, 30));
    OFB.SetClickStyle(1, RGB(10, 89, 247), BS_SOLID, NULL, RGB(204, 232, 255), RGB(30, 30, 30));
    OFB.SetLoadingStyle(1, RGB(10, 89, 247), BS_SOLID, NULL, RGB(204, 204, 204), RGB(130, 130, 130));

    system("pause");
};

//注: 在不需要参数时必须填 NULL
```