

说明

including graphics 文件夹中的库已经编译好了 graphics 库的内容，**不需要再把 graphics 库中的众多.c 源文件添加到项目中，但仍需要将 graphics 库的众多.h 头文件添加到项目中!** not including graphics 文件夹中的库需要自行将 graphics 库所有文件添加到 vs 项目中

DevC++ 与 VS

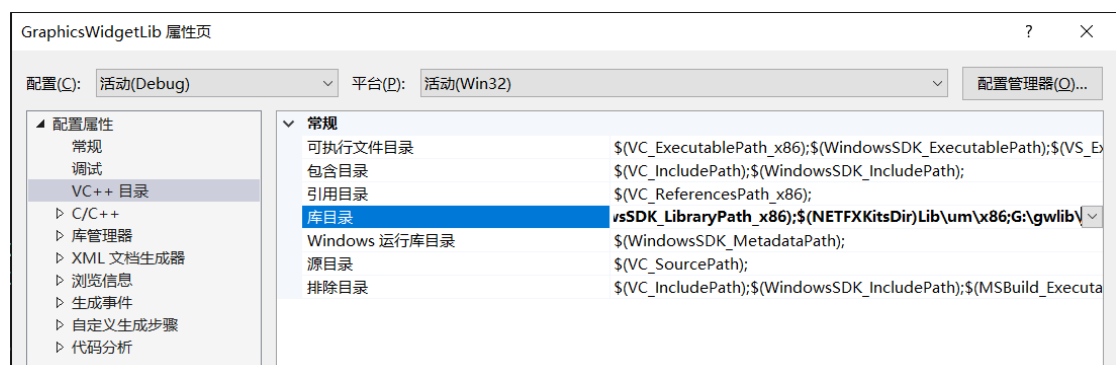
两个版本的头文件相同，都在 include 文件夹下，而 vs 的库文件在 lib 文件夹，devc 的库文件在 lib-devc，仅支持 including graphics!

devc 的库文件配置参考 SGL 图形库的配置视频

<https://www.bilibili.com/video/av12895517>，操作大同小异。

配置项目文件 (vs2017)

1. 新建一个空项目，将相关文件复制到项目目录下 (.lib 库文件不必)
2. 项目 - 属性 - VC++ 目录 - 库目录中，增加 .lib 库文件所存放的路径



3. 开始编写代码吧!

用法说明

Widget 结构成员简介

```
struct WStruct
//warning: should not directly change the private values
{
//public:
    //common members of widgets
    double x;//左上角坐标
    double y;//左上角坐标
    double width;
    double height;
    int value;//不同控件值的含义不同
    void(*valueChangedEvent)(Widget w);//slider的事件处理
    void(*setRange)(Widget w, int a, int b);//设置slider的范围
    string content;//内容，一般为显示在控件上面的文本
    Design design;//控件颜色设计，默认设置好，需要的话可以自行修改
    enum WidgetStatus status;//状态
    MouseClickCall mouseClicked;//鼠标事件
    void(*show)(Widget w);
    void(*hide)(Widget w);
    void(*activateScroll)(Widget w);//output控件专属（以后可能会更新给其他控件）不需要自己调用
    void(*update)(Widget w);//刷新控件绘图

    //particular for input widget
    bool cursorVisible;
    int cursorPos;
    CharCall charInput;
    KeyCall keyPress;
    InputCall inputCall;//input的输入事件

    //particular for label and output
    int size;//字体大小，默认15

    //particular for output
    void(*append)(Widget w, string s);//给content追加一行内容，用于output和list

    //particular for slider
    int range[2];//slider的范围，建议通过调用setRange修改而不是直接修改
```

```

//private:
    //common members of widgets
    string name; //除了list以外，只用于查找控件的标识，不会显示在屏幕上
    enum WidgetType type;
    bool visible;
    MouseMoveCall mouseMove;
    Scroll scroll;

    //particular for input widget
    TimeCall timer;

    //particular for output widget
    int lines;
    int displayLines;
    int curLine;
};

```

控件的创建

```

enum WidgetType
{
    GW_BUTTON,
    GW_INPUT,
    GW_LABEL,
    GW_OUTPUT,
    GW_SLIDER,
    GW_LIST
};

Widget newWidget(string name, enum WidgetType type);
void setWidgetPos(Widget w, double x, double y, double width, double height);
void registerWidget(Widget w);
Widget findWidgetByName(string name);

```

newWidget 用于创建新的控件，创建后调用 setWidgetPos 设置位置和大小，然后 registerWidget 注册（注意注册的是创建的拷贝，所以注册完可以把创建的 free 掉），需要查找控件时用 findWidgetByName。注意：新建控件默认不显示，需要调用一次 show 成员函数

事件

控件本身有着默认的鼠标、键盘等事件，用于控制颜色、控件状态等，因此，在创建自定义事件以后，需要在自定义事件中调用一次默认函数。以 button 为例：

```
void myEvent(Widget w, int x, int y, int button, int event)
{
    buttonMouseClicked(w, x, y, button, event);
    if (w->status == GW_SELECTED)
    {
        //Custom operations
    }
}
```

如果你想自定义全局操作（用 graphics 库的 registerMouseEvent 等），别忘了在你的自定义事件函数中调用控件的全局控制函数，否则所有控件都会失效！

默认全局控制函数如下：

```
void widgetMouseEvent(int x, int y, int button, int event);
void widgetCharEvent(char c);
void widgetKeyEvent(int key, int event);
void widgetTimeEvent(int timerID);
```

最后，祝编程愉快！(*^▽^*)