

FLTK绘制三角函数

骆克云

2015.12.15

FLTK绘制三角函数

程序环境

功能介绍与程序思想

运行截图

核心代码

程序环境

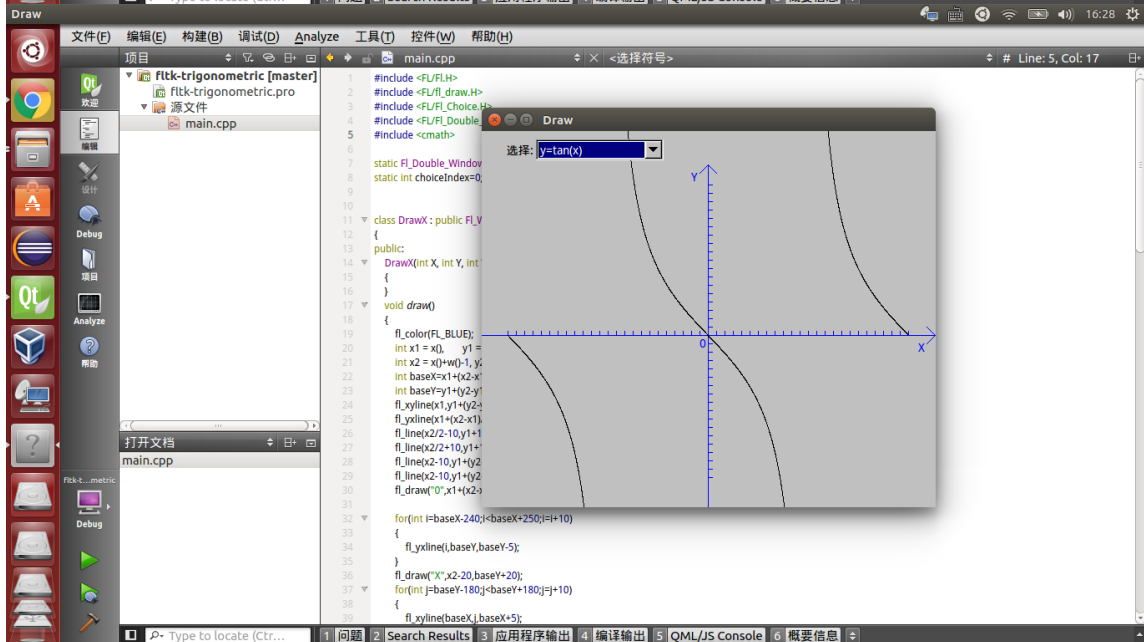
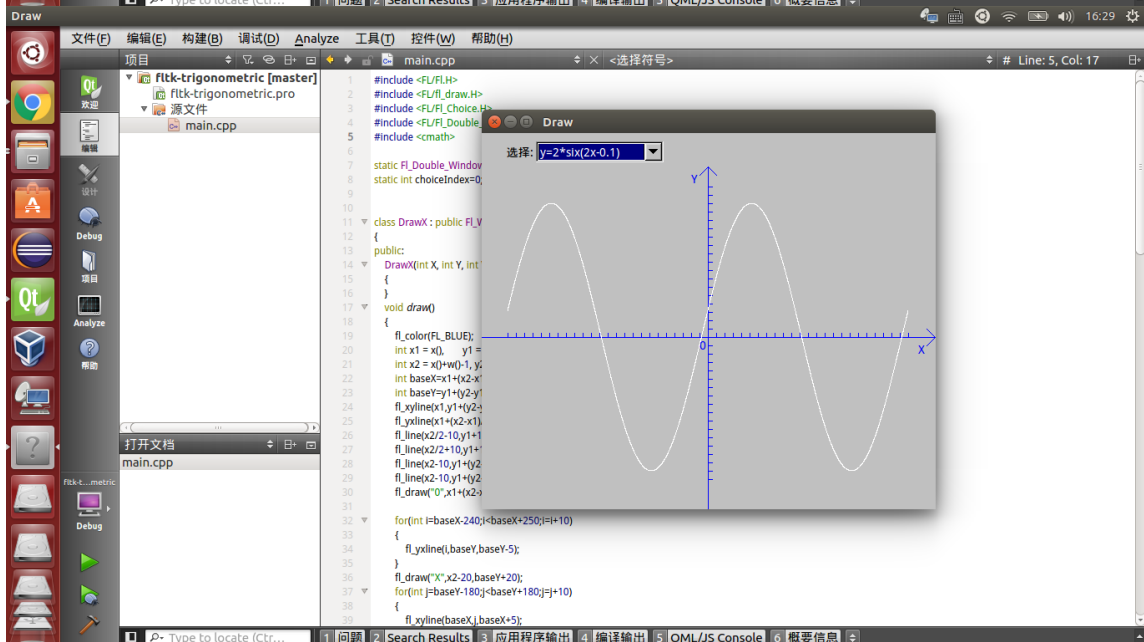
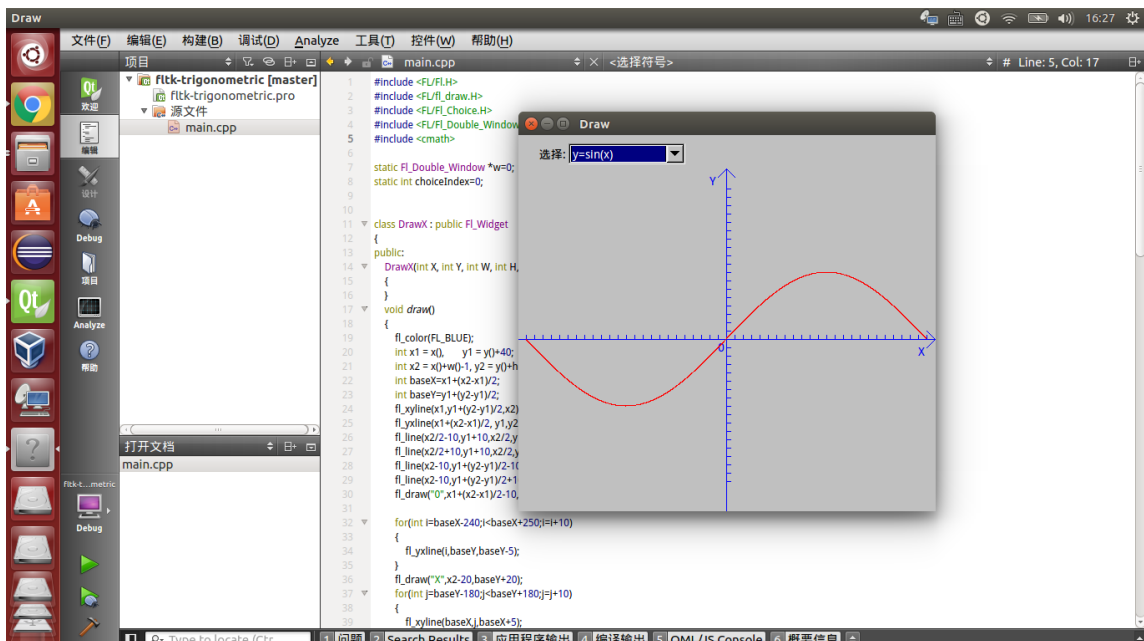
- FLTK版本：1.3.3
- 系统: Ubuntu14.04.3 LTS
- IDE: Qt 5.5.1
- C/C++版本: GCC/G++ 4.8.4

功能介绍与程序思想

程序运行时，可以选择下拉框来决定所画的三角函数的种类，包括sin/cos/tan等，主要涉及到坐标轴的绘制,函数图形的显示，回调函数的设计，坐标单位的拾取等。

- **坐标轴的绘制:**
在窗口中选取窗口的横纵终点画直线
- **函数图形的显示:**
绘制三角函数时，以连续的点代替曲线，即函数 $y = f(x)$ 的值为纵坐标，x的值为横坐标
- **回调函数的设计:**
当点击选择下拉框时，回调函数响应，根据所选的函数绘制不同的图形
- **坐标单位的拾取:**
为方便计算，以整个坐标横轴为一个周期
- **实现的函数**
 - $y = \sin(x)$
 - $y = \cos(x)$
 - $y = \tan(x)$
 - $y = 2 * \sin(2x - 0.1)$

运行截图




```

void draw()
{
    fl_color(FL_BLUE);
    int x1 = x(), y1 = y()+40;
    int x2 = x()+w()-1, y2 = y()+h()-1;
    int baseX=x1+(x2-x1)/2;
    int baseY=y1+(y2-y1)/2;
    fl_xyline(x1,y1+(y2-y1)/2,x2);
    fl_yxline(x1+(x2-x1)/2, y1,y2);
    fl_line(x2/2-10,y1+10,x2/2,y1);
    fl_line(x2/2+10,y1+10,x2/2,y1);
    fl_line(x2-10,y1+(y2-y1)/2-10,x2,y1+(y2-y1)/2);
    fl_line(x2-10,y1+(y2-y1)/2+10,x2,y1+(y2-y1)/2);
    fl_draw("0",x1+(x2-x1)/2-10,y1+(y2-y1)/2+15);

    for(int i=baseX-240;i<baseX+250;i=i+10)
    {
        fl_yxline(i,baseY,baseY-5);
    }
    fl_draw("X",x2-20,baseY+20);
    for(int j=baseY-180;j<baseY+180;j=j+10)
    {
        fl_xyline(baseX,j,baseX+5);
    }
    fl_draw("Y",baseX-20,y1+20);
    switch (choiceIndex) {
    case 0:
        fl_color(FL_RED);
        for (double i=-240+baseX;i<240+baseX;i=i+0.1)
        {
            float xPoint=(i-baseX)/-240.0*3.1415926;
            float yPoint=80*sin(xPoint)+baseY;
            fl_point(i,yPoint);
        }
        break;
    case 1:
        fl_color(FL_YELLOW);
        for (double i=-240+baseX;i<240+baseX;i=i+0.1)
        {
            float xPoint=(i-baseX)/240.0*3.1415926;
            float yPoint=-80*cos(xPoint)+baseY;
            fl_point(i,yPoint);
        }
        break;
    case 2:
        fl_color(FL_BLACK);
        for (double i=-240+baseX;i<240+baseX;i=i+0.1)
        {
            float xPoint=(i-baseX)/240.0*3.1415926;
            float yPoint=80*tan(xPoint)+baseY;

```

```

        fl_point(i,yPoint);
    }
    break;
case 3:
    fl_color(FL_WHITE);
    for (double i=-240+baseX;i<240+baseX;i=i+0.1)
    {
        float xPoint=(i-baseX)/-240.0*3.1415926;
        float yPoint=2*80*sin(2*xPoint-0.2)+baseY;
        fl_point(i,yPoint);
    }
    break;
default:
    fl_color(FL_GREEN);
    fl_line(x1,y1,x2/2,y2);
    break;
}
}

```

2.回调函数

```

static void Choice_CB(Fl_Widget*widget)
{
    Fl_Choice *ch = (Fl_Choice*)widget;
    choiceIndex=ch->value();
    DrawX *draw_x=new DrawX(0, 0, w->w(), w->h(),0);
    w->resizable(draw_x);
    w->redraw();
}

```