### 可视化Tkinter编辑器

F1303022

5130309668

谢异

1. **起因：**

开发这样一款编辑器的原因，主要源于以前用过dreamweaver（**美国MACROMEDIA公司开发的集网页制作和管理网站于一身的所见即所得网页编辑器，它是第一套针对专业网页设计师特别发展的视觉化网页开发工具**），比较习惯可视化，而不喜欢用代码调控界面。

**二、程序功能和使用说明**

**程序功能：**

用于迅速地**制作界面**，**画图**，**并生成python代码，甚至还能用来教学。**

**使用说明：**

首先打开名为“Virtual Tkinter”的文件

图1展示的是在初始的界面。

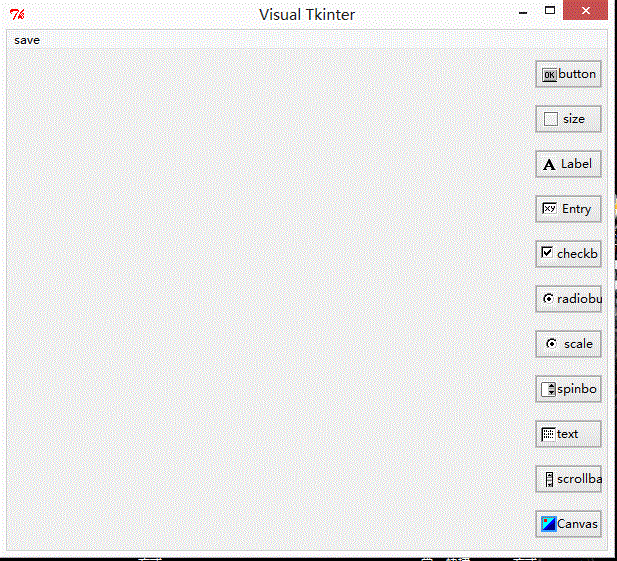


图 1 可视化编辑器的界面

可以看到右边有一排按键，总共11个。每个按键前有一个小图标暗示按键的种类。

第1个到第10个按键是用来做界面的，生成Button之类的东西。第11个按键是用来画图的。

**（1）制作界面：**

比如做一个按钮。

按钮的图标：

C:\Users\dell\Desktop\演示\3.gif

图 2 Button按钮

然后按住Button并拖动它到你想要的位置，再放开。

图三显示的是放开后弹出的窗口。

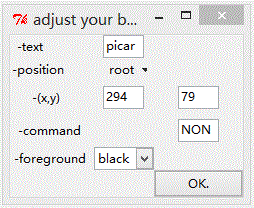
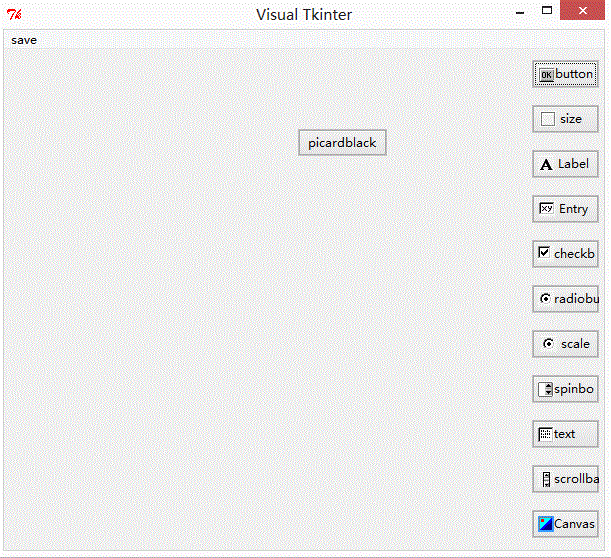


图 3 弹出窗口

弹出窗口显示一系列的设置，你可以选择改动它，如果不改动，则为默认值。然后再点击右下角的OK。



点完了之后，你就能在屏幕中间看到了一个按钮，文字是“picardblack”，我用了black来暗示它的颜色。

然后点击左上角的“save”，选择文件夹，保存。



图 5 save按钮

保存好之后，打开生成的文件，就能看到代码。

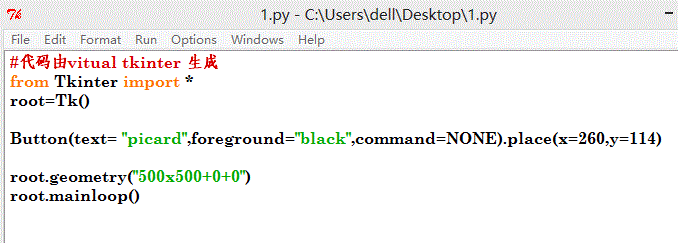
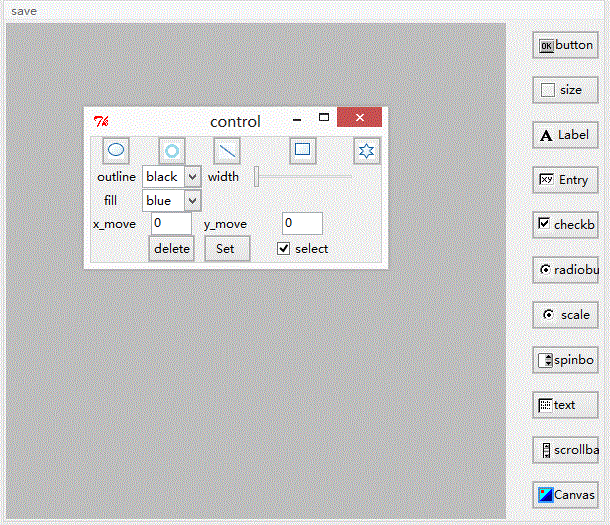


图 6 生成的文件

**（2）画图操作**

点击一个图标为C:\Users\dell\Desktop\截图02.gif的按钮。

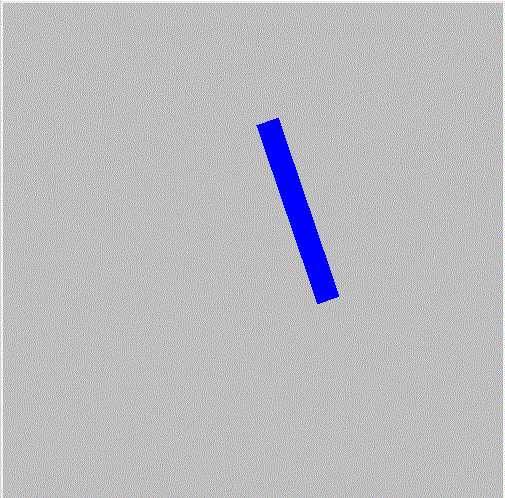


如图所示，点开后就进入编辑界面，背景颜色改变（画布），并弹出窗口。这个窗口是一个

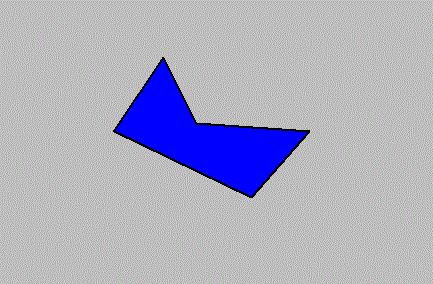
控制器，包含直线、椭圆、多边形……等操作。

比如我画一条线。首先单击直线（看图标就知道）。然后在画布上拖动。松手后就能得

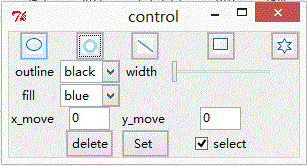
到一条直线。



现在来画多边形。首先单击多边形（最右边那个）。然后在屏幕上画点。每点一个点就会连一条线，不想画了时，点击初始的那个点就能使多边形闭合。



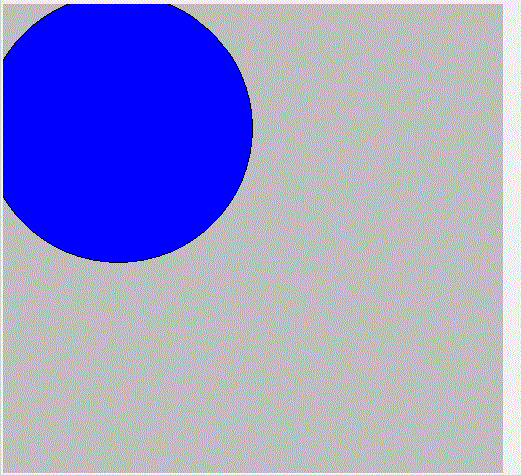
现在我来演示如何修改已经画好的。



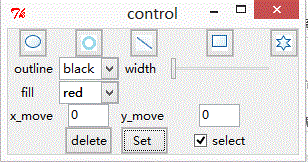
观察上图可以知道，outline 代表轮廓颜色，下拉框改变，fill代表填充颜色，下拉框改变。

Width代表线的宽度，滚动改变。X\_move代表水平方向移动，y\_move代表垂直方向移动。

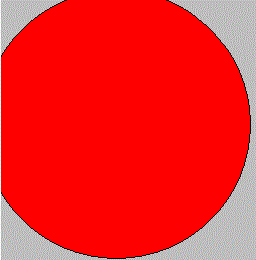
现在我们画一个圆。



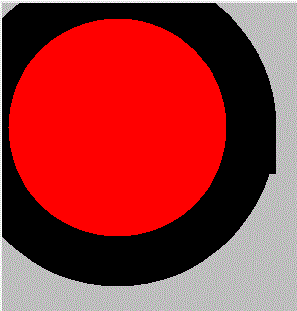
然后我们单击这个圆，把fill改为red。



**改好之后，单击set键（如果不点就不会改的哦）。**

****

**我们还可以尝试修改width**

****

**选中某一个之后，单击delete键就可以把它删掉。**

**C:\Users\dell\Desktop\截图11.gif**

**三、如何实现**

本编辑器由两部分组成，一部分画图，另一部分做界面。画图的部分我做了一个叫draw的类。每当点击Canvas按钮的时候就会创建这样的一个类。类里实现了画图，加代码的全部功能（光是画图就接近200行代码），基本独立。

另外一个做界面的，没有用类装了。本编辑器基本没用外来的模块。都是系统自带的，比如string,Tkinter,ttk,math。

**四、未来的工作**

我的大作业还有很多需要改进的地方。比如，画一个多边形，画好后，你改变不了它的轮廓的颜色与厚度。由于多边形与其它图形是有区别的，它的轮廓是直线连起来的，所以只有改变直线的颜色才能改变它的颜色。

其次，如果真要拿来做界面，也很难实现作者完全不用写代码，很多时候需要自己加上几句（比如在Command 里加上）。

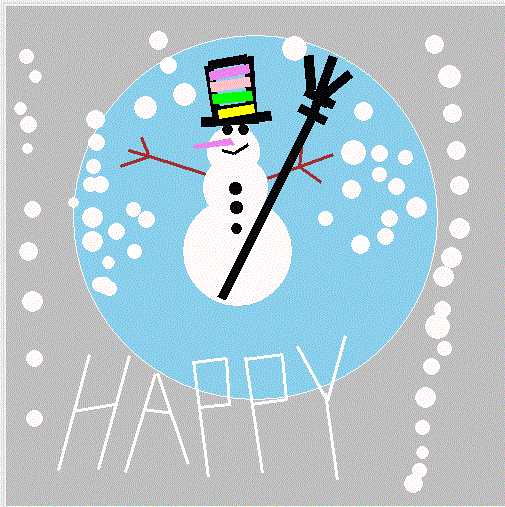
再比如，在拖动按钮时，没有拖动的效果，导致用户可能不知道自己在拖动。以及在画图时，选中一个图形，没有提示（比如虚线框围着它），导致用户不知道已经选中了。

总之，如何实现用户完全不要写代码是它的终极目标，在实现这个终极目标前有很多路要走。

**五、一些实例**

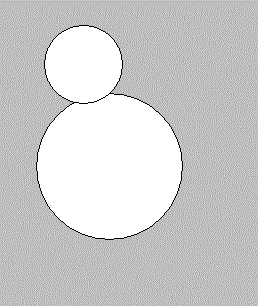
由于我的大作业是应用类的，实践才是最重要的。在“实例”文件夹里有用编辑器做出来的实例，现在我来演示一下。

**画图实例**：

****

这是第二个画图实例，可即使是用编辑器画的，也要调整位置（用x\_move，y\_move），现在我来演示画一个雪人的身体。

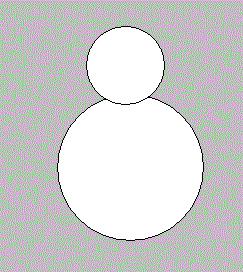
1. 单击canvasC:\Users\dell\Desktop\截图01.gif
2. 单击圆，并选择颜色和宽度。现在我们选中白色。
3. 画两个圆。



这样我们就得到了这样一幅图。发现头歪了。这时我们选中它的头部。

然后把x\_move改成一个非零的数字。C:\Users\dell\Desktop\截图03.gif

然后狂点‘set’ 键.



然后它的头就正了。

总之大致方法就是这样，如果画错了就选中某个图后按delete键。

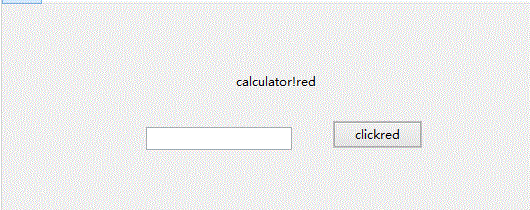
Ps：C:\Users\dell\Desktop\截图05.gif如果select前的勾没有选中，则永远也不能选中某一个图形。

**应用实例**：

现在我来演示如何迅速做一个简单的计算器。

打开一个新的编辑器。

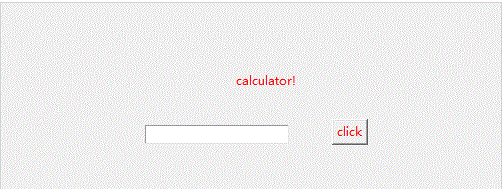
我拖动一个button，名字设置为“click”，一个entry，一个label。

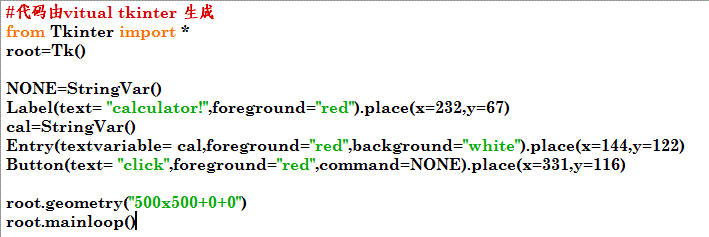


拖动好之后可以看到如图所示的界面。

我们点击“save”，选择一个文件夹。这时我们可以关闭编辑器了。

打开生成的文件，界面如图所示。此时点击click还没有功能，因为我们还没添加。





可以看到Tkinter编辑器其实已经帮我们定义了一个Stringvar（使用编辑器的人还是得有一点基础的）。

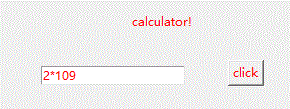
我们在顶端加入一个简单的函数（函数当然得用户写）。

C:\Users\dell\Desktop\截图09.gif

然后把C:\Users\dell\Desktop\截图10.gif，command后加上刚才定义的函数名称。、

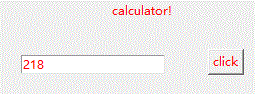
这样一个朝简易计算器就做好了。怎么样，2分钟都没过吧。来看一下效果。

在框里输入一个表达式。



然后点击click。

答案就出来了。



好了，演示就到这里。总之编辑器在开发界面的时候是很轻便的。

**六、结语**

这次开发大作业，我学到了很多。比如做程序不仅是技术活，还是体力活，把算法弄清楚之后，会有很多重复性的工作。

不过也有很多遗憾，比如自己不敢去想，以为自己什么都做不了。我做编辑器，正是因为自己已经把大体构思想好了。在确保自己能做出来后，我才去做的。可做完编辑器后我发现，自己本可以通过查资料做出更厉害的东西。尤其是当我接触到一些新奇的模块之后，我发现Python迷人的地方就在于它是开源的，于是有很多很厉害的库，借助这些库，能做出很多很棒的东西。

虽然大作业做完了，但我感觉自己在寒假还会用一些新奇的库做一些东西。