省時省力！這大概是用Python寫GUI最快的方式了！

AI碼科技 2022-01-16 20:20

以下文章來源於Python技術，作者派森醬

Python技術

Python 技術由一群熱愛Python 的技術人組建，專業輸出高質量原創的Python 系列文...

AI碼科技

編程吧，少年！這裡有等你修復的Bug！

14篇原創內容

公眾號

我們在之前的文章 **丟棄Tkinter！幾行代碼快速生成漂亮GUI！** 中，給大家介紹了一款python 的GUI 神器—— PySimpleGUI，並且給大家演示了一些基本的用法。這篇文章收到好多讀者的反饋，說這個確實比較簡單，除了界面稍微有點“原始”，沒毛病。

其實像PySimpleGUI 這類GUI 界面，跟Web 頁面是不具備可比性的，後者想做得美觀簡直太容易了。而GUI 界面本來就是為了生成可執行的軟件而生的，在美觀上先天性不足。

PySimpleGUI 是python GUI 框架中的佼佼者，適用於快速生成簡潔大方的GUI。使用它來寫GUI 已經比較快了，那麼還有沒有更快的方法嗎？

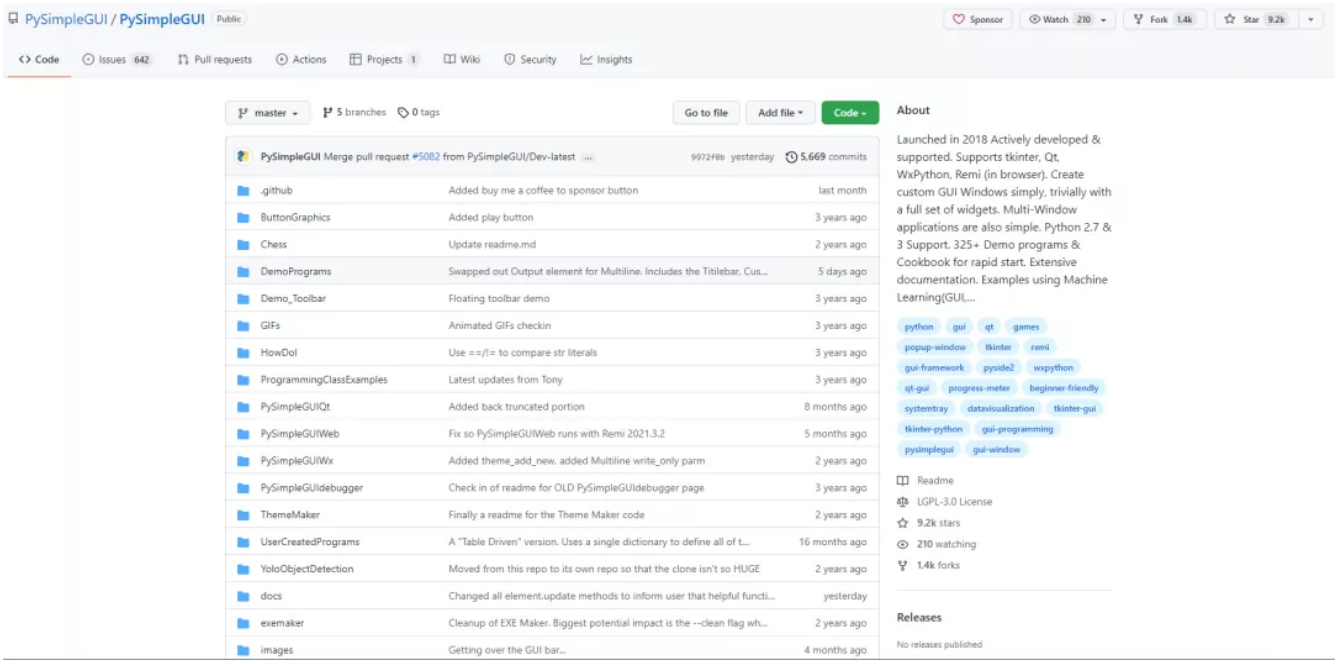
答案是肯定的，本文就為你揭曉！

GUI 實例

PySimpleGUI 在GitHub上的地址是：

> https://github.com/PySimpleGUI/PySimpleGUI

大家可以訪問看看，其首頁是這樣的：



有很多內容是不是？

這裡面有一個重要的內容，在DemoPrograms 文件夾下，這個文件夾是作者寫的一些demo 實例。作者真的是深諳我們這些懶蟲的心理，即使有了這麼簡單好用的GUI 框架，到了要寫實例的時候，我們也可能會去網絡上搜索實例，然後採用 CV大法 。框架作者可能料想到這一點，所以他自己寫了很多不同的實例，讓你真正的拿來即用。

這個文件夾下大概有300多個實例，基本上可以囊括我們平時使用python 寫GUI 所能遇到的各種組件和佈局了。

CV 幾個看看

有了這個神器，我們只需要把這個GitHub 上的項目給複製到本地，然後將這些實例運行一遍，大致知道每個實例u哪些內容。後續當我們自己要寫GUI 時，我們只需要找到對應的實例，然後復制代碼就可以了。是不是很簡單？

下面我來運行幾個demo ，給大家展示一下這裡面的實例都是什麼樣子的。

聊天界面

我們先複製一下源碼：

```
#!/usr/bin/env python
```

```
import PySimpleGUI as sg

'''
A chatbot with history
Scroll up and down through prior commands using the arrow keys
Special keyboard keys:
    Up arrow - scroll up in commands
    Down arrow - scroll down in commands
    Escape - clear current command
    Control C - exit form
'''


def ChatBotWithHistory():
    # -------  Make a new Window  ------- #
    # give our form a spiffy set of colors
    sg.theme('GreenTan')

    layout = [[sg.Text('Your output will go here', size=(40, 1))],
              [sg.Output(size=(127, 30), font=('Helvetica 10'))],
              [sg.Text('Command History'),
               sg.Text('', size=(20, 3), key='history')],
              [sg.ML(size=(85, 5), enter_submits=True, key='query', do_not_clear=False),
               sg.Button('SEND', button_color=(sg.YELLOWS[0], sg.BLUES[0]), bind_return_
               sg.Button('EXIT', button_color=(sg.YELLOWS[0], sg.GREENS[0]))]]

    window = sg.Window('Chat window with history', layout,
                       default_element_size=(30, 2),
                       font=('Helvetica', ' 13'),
                       default_button_element_size=(8, 2),
                       return_keyboard_events=True)

    # --------- Loop taking in user input and using it  --- #
    command_history = []
    history_offset = 0

    while True:
        event, value = window.read()

        if event == 'SEND':
            query = value['query'].rstrip()
            # EXECUTE YOUR COMMAND HERE
```

```
            print('The command you entered was {}'.format(query))
            command_history.append(query)
            history_offset = len(command_history)-1
            # manually clear input because keyboard events blocks clear
            window['query'].update('')
            window['history'].update('\n'.join(command_history[-3:]))

        elif event in (sg.WIN_CLOSED, 'EXIT'):            # quit if exit event or X
            break

        elif 'Up' in event and len(command_history):
            command = command_history[history_offset]
            # decrement is not zero
            history_offset -= 1 * (history_offset > 0)
            window['query'].update(command)

        elif 'Down' in event and len(command_history):
            # increment up to end of list
            history_offset += 1 * (history_offset < len(command_history)-1)
            command = command_history[history_offset]
            window['query'].update(command)

        elif 'Escape' in event:
            window['query'].update('')


ChatBotWithHistory()
```

運行一下，看看效果：

這是一個帶歷史記錄的聊天軟件，如果你需要做一個類似的軟件的話，可以直接複製代碼，然後稍微改動一下。

組件大全

我們再來看一個例子：

```
#!/usr/bin/env python
"""
    Example of (almost) all Elements, that you can use in PySimpleGUI.
    Shows you the basics including:
        Naming convention for keys
        Menubar format
        Right click menu format
        Table format
        Running an async event loop
        Theming your application (requires a window restart)
        Displays the values dictionary entry for each element
        And more!

    Copyright 2021 PySimpleGUI
"""

import PySimpleGUI as sg

def make_window(theme):
    sg.theme(theme)
    menu_def = [['&Application', ['E&xit']],
                ['&Help', ['&About']] ]
```

```
right_click_menu_def = [[], ['Nothing','More Nothing','Exit']]

# Table Data
data = [["John", 10], ["Jen", 5]]
headings = ["Name", "Score"]

input_layout =  [[sg.Menu(menu_def, key='-MENU-')],
            [sg.Text('Anything that requires user-input is in this tab!')],
            [sg.Input(key='-INPUT-')],
            [sg.Slider(orientation='h', key='-SKIDER-'),
             sg.Image(data=sg.DEFAULT_BASE64_LOADING_GIF, enable_events=True, key='-
            [sg.Checkbox('Checkbox', default=True, k='-CB-')],
            [sg.Radio('Radio1', "RadioDemo", default=True, size=(10,1), k='-R1-'), sg
            [sg.Combo(values=('Combo 1', 'Combo 2', 'Combo 3'), default_value='Combo
             sg.OptionMenu(values=('Option 1', 'Option 2', 'Option 3'),  k='-OPTION M
            [sg.Spin([i for i in range(1,11)], initial_value=10, k='-SPIN-'), sg.Text
            [sg.Multiline('Demo of a Multi-Line Text Element!\nLine 2\nLine 3\nLine 4
            [sg.Button('Button'), sg.Button('Popup'), sg.Button(image_data=sg.DEFAULT

asthetic_layout = [[sg.T('Anything that you would use for asthetics is in this tab!')
           [sg.Image(data=sg.DEFAULT_BASE64_ICON,  k='-IMAGE-')],
           [sg.ProgressBar(1000, orientation='h', size=(20, 20), key='-PROGRESS BAR-

logging_layout = [[sg.Text("Anything printed will display here!")], [sg.Output(size=(

graphing_layout = [[sg.Text("Anything you would use to graph will display here!")],
                   [sg.Graph((200,200), (0,0),(200,200),background_color="black", key=
                   [sg.T('Click anywhere on graph to draw a circle')],
                   [sg.Table(values=data, headings=headings, max_col_width=25,
                             background_color='black',
                             auto_size_columns=True,
                             display_row_numbers=True,
                             justification='right',
                             num_rows=2,
                             alternating_row_color='black',
                             key='-TABLE-',
                             row_height=25)]]

specalty_layout = [[sg.Text("Any \"special\" elements will display here!")],
                   [sg.Button("Open Folder")],
                   [sg.Button("Open File")]]
```

```
    theme_layout = [[sg.Text("See how elements look under different themes by choosing a
                    [sg.Listbox(values = sg.theme_list(),
                      size =(20, 12),
                      key ='-THEME LISTBOX-',
                      enable_events = True)],
                      [sg.Button("Set Theme")]]

    layout = [[sg.Text('Demo Of (Almost) All Elements', size=(38, 1), justification='cent
    layout +=[[sg.TabGroup([[  sg.Tab('Input Elements', input_layout),
                               sg.Tab('Asthetic Elements', asthetic_layout),
                               sg.Tab('Graphing', graphing_layout),
                               sg.Tab('Specialty', specalty_layout),
                               sg.Tab('Theming', theme_layout),
                               sg.Tab('Output', logging_layout)]], key='-TAB GROUP-')]]

    return sg.Window('All Elements Demo', layout, right_click_menu=right_click_menu_def)


def main():
    window = make_window(sg.theme())

    # This is an Event Loop
    while True:
        event, values = window.read(timeout=100)
        # keep an animation running so show things are happening
        window['-GIF-IMAGE-'].update_animation(sg.DEFAULT_BASE64_LOADING_GIF, time_betwe
        if event not in (sg.TIMEOUT_EVENT, sg.WIN_CLOSED):
            print('============ Event = ', event, ' ==============')
            print('-------- Values Dictionary (key=value) --------')
            for key in values:
                print(key, ' = ',values[key])
        if event in (None, 'Exit'):
            print("[LOG] Clicked Exit!")
            break
        elif event == 'About':
            print("[LOG] Clicked About!")
            sg.popup('PySimpleGUI Demo All Elements',
                     'Right click anywhere to see right click menu',
                     'Visit each of the tabs to see available elements',
                     'Output of event and values can be see in Output tab',
                     'The event and values dictionary is printed after every event')
        elif event == 'Popup':
```

```
            print("[LOG] Clicked Popup Button!")
            sg.popup("You pressed a button!")
            print("[LOG] Dismissing Popup!")
        elif event == 'Test Progress bar':
            print("[LOG] Clicked Test Progress Bar!")
            progress_bar = window['-PROGRESS BAR-']
            for i in range(1000):
                print("[LOG] Updating progress bar by 1 step ("+str(i)+")")
                progress_bar.UpdateBar(i + 1)
            print("[LOG] Progress bar complete!")
        elif event == "-GRAPH-":
            graph = window['-GRAPH-']       # type: sg.Graph
            graph.draw_circle(values['-GRAPH-'], fill_color='yellow', radius=20)
            print("[LOG] Circle drawn at: " + str(values['-GRAPH-']))
        elif event == "Open Folder":
            print("[LOG] Clicked Open Folder!")
            folder_or_file = sg.popup_get_folder('Choose your folder')
            sg.popup("You chose: " + str(folder_or_file))
            print("[LOG] User chose folder: " + str(folder_or_file))
        elif event == "Open File":
            print("[LOG] Clicked Open File!")
            folder_or_file = sg.popup_get_file('Choose your file')
            sg.popup("You chose: " + str(folder_or_file))
            print("[LOG] User chose file: " + str(folder_or_file))
        elif event == "Set Theme":
            print("[LOG] Clicked Set Theme!")
            theme_chosen = values['-THEME LISTBOX-'][0]
            print("[LOG] User Chose Theme: " + str(theme_chosen))
            window.close()
            window = make_window(theme_chosen)

    window.close()
    exit(0)

if __name__ == '__main__':
    main()
```

我們來看看運行之後的效果：

這個demo 是PySimpleGUI 所有組件的集合，每一個tab 都是一個分類。這裡麵包括進度條、畫布、主題、滾動條等等。如果你想要找界面組件，到這個demo 的源碼裡面找就對了。

總結

這裡面還有更多實例，大家就自己去探索吧！這裡主要是給大家介紹一個快速開發GUI 的方法，俗稱 cv大法 。不過這只是一種快速開發方式，大家有時間還是去看看源碼，了解一下原理比較好！

大家有什麼需要探討的，可以在評論區留言！

<end>

- **Python如何實現'2021年12月18日'轉換為'2021/12/18'格式?**
- **用Python給代碼安個進度條，太香了吧**