

Python DearPyGui 进阶



何小有 [关注](#)

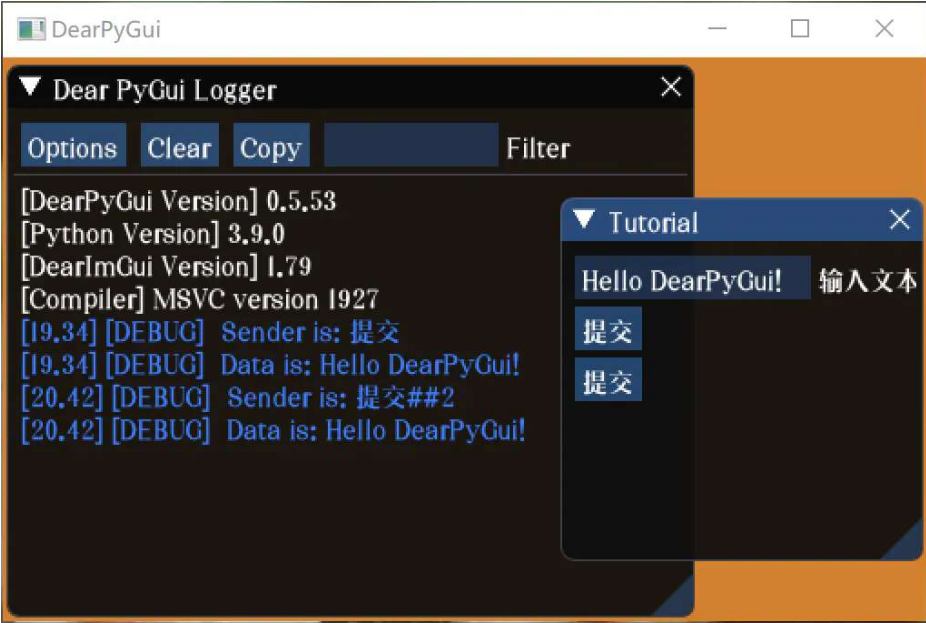
2020.11.02 20:09:13 字数 848 阅读 38

控件和窗口回调

每个输入控件都有一个回调，该回调在控件交互时运行，回调用于为控件提供功能实现。回调可以在创建时或在创建后使用 `set_item_callback` 分配给窗口控件。

在 DearPyGui 中，应用于控件的每个回调方法都必须包含一个 `sender` 和 `data` 参数。DearPyGui 使用 `sender` 参数来通知回调——哪个控件通过发送 `name` 来触发回调。控件再通过指定 `callback_data` 参数来发送数据到回调方法的 `data` 参数。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6 def button_callback(sender, data):
7     log_debug(f"Sender is: {sender}")
8     log_debug(f"Data is: {data}")
9
10 show_logger() # 使用logger窗口显示结果
11
12 with window("Tutorial"):
13     add_input_text("输入文本", default_value="Hello DearPyGui!")
14     add_button("提交", callback=button_callback, callback_data=get_value("输入文本"))
15     add_button("提交##2", tip="创建控件后设置了回调")
16     set_item_callback("提交##2", callback=button_callback, callback_data=get_value("输入文本"))
17
18 start_dearpygui()
```



dearpygui_callbackdata.png

每次与控件交互时，都可以使用回调来更新变量的值。



何小有

[关注](#)

总资产49 (约3.39元)

Python DearPyGui 进阶

阅读 38

Python DearPyGui 常用控件二

阅读 7

Python DearPyGui 常用控件一

阅读 22

推荐阅读

Android TV开发之使用Leanback传输控件

阅读 41

Vue 前端可视化活动编辑器

阅读 607

Android 11 新特性 (含Android Studio 4.0 新特性)

阅读 6,787

Vue中computed和watch的区别

阅读 586

含泪狂刷Android面试100题，面试时吊打面试官

阅读 459

```
7     my_var = get_value("输入复选框")
8     log_debug(my_var)
9
10    show_logger() # 使用logger窗口显示结果
11
12    with window("Tutorial"):
13        add_checkbox("输入复选框", callback=update_var)
14
15    start_dearpygui()
```

<

>

dearpygui_senderdata.png

但是，上面的方法缺乏灵活性，无法实现通用的回调方法，下面是通过使用 `sender` 参数实现的一种更聪明的方法。

```
1  from dearpygui.core import *
2  from dearpygui.simple import *
3
4  add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6  def update_var(sender, data):
7      my_var = get_value(sender)
8      log_debug(my_var)
9
10     show_logger() # 使用logger窗口显示结果
11
12     with window("Tutorial"):
13         add_checkbox("输入复选框", callback=update_var)
14         add_input_text("输入文本", callback=update_var)
15         add_input_int("输入整数", callback=update_var)
16
17     start_dearpygui()
```

<

>

写下你的评论...

评论0

赞

...


dearpygui_senderdatasender.png

窗口以及窗口类型的控件具有特殊的回调，这些回调在特殊的事件（例如窗口大小调整、窗口关闭等）上触发。`on_close` 将在窗口关闭时运行分配给该参数的回调，而 `set_resize_callback` 将在窗口被调整大小时运行，并且可以使用 `handler` 参数设置成任意窗口，默认值为 主窗口。

另外，如果你希望每帧都执行一次回调，则可以使用 `set_render_callback` 方法。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6 def close_me(sender, data):
7     log_debug(f"{sender} 窗口已经关闭")
8
9 def render_me(sender, data):
10    log_debug(f"窗口 {sender} 运行了 set_render 回调")
11
12 def resize_me(sender, data):
13    log_debug(f"窗口 {sender} 运行了 set_resize 回调")
14
15 show_logger() # 使用logger窗口显示结果
16
17 with window("Tester", on_close=close_me):
18     add_text('调整此窗口的大小将触发 set_resize 回调')
19     add_text('关闭此窗口将触发 close 回调')
20
21 # logger窗口狂刷: [x.xx] [DEBUG] 窗口 Main Application 运行了 set_render 回调
22 # set_render_callback(render_me)
23
24 set_resize_callback(resize_me, handler="Tester")
25
26 start_dearpygui()
```


dearpygui_setrendercallback.png

运行时添加和删除控件

Dear PyGui 支持在运行时动态添加和删除任何控件或窗口。通过使用同运行任何控件的

写下你的评论...

评论0

赞

...

而通过在添加控件时使用 `before` 参数，可以设置将新控件放在哪个控件之前，默认情况下，会将新控件放在最后。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 def add_buttons(sender, data):
5     add_button("New Button 2", parent="Secondary Window")
6     add_button("New Button", before="New Button 2")
7
8 def delete_buttons(sender, data):
9     delete_item("New Button")
10    delete_item("New Button 2")
11
12 show_debug()
13
14 with window("Tutorial"):
15     add_button("Add Buttons", callback=add_buttons)
16     add_button("Delete Buttons", callback=delete_buttons)
17
18 with window("Secondary Window"):
19     pass
20
21 start_dearpygui()
```

dearpygui_deleteitem.png

删除窗口时，默认情况下，会删除窗口及其子控件，如果只想删除子控件，可以将 `children_only` 参数设置为 `True` 值。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6 def add_widgets(sender, data):
7     with window("Secondary Window"):
8         add_button("New Button 2")
9         add_button("New Button")
10        add_button("New Button 3", parent="Secondary Window")
11
12 def delete_widgets(sender, data):
13     delete_item("Secondary Window")
14
15 def delete_children(sender, data):
```

写下你的评论...

评论0

赞

...

```
21 add_button("添加窗口以及控件", callback=add_widgets)
22 add_button("删除窗口以及子控件", callback=delete_widgets)
23 add_button("删除窗口的子控件", callback=delete_children)
24
25 start_dearpygui()
```

dearpygui_childrenonly.png

值与数据存储

添加新的窗口控件时，会将 `value` 添加到 Value**存储系统** 中，默认情况下，此 `value` 的标识符是控件的 `name`。我们可以使用 `source` 参数覆盖标识符，这样做有一个好处，就是让多个控件控制同一个 `value`。

- 使用 `get_value` 从 Value**存储系统** 中检索 `value`
- 使用 `set_value` 可以手动更改 `value`

为了使不同 `value` 类型的控件在 Value**存储系统** 中使用相同的值，必须先创建较大的 `value`。例如，当创建一个 `input_float3` 控件时，存储的 `value` 类型为 `[float, float, float]`，后来创建的 `input_float2` 可以使用与 `input_float3` 相同的 `source` 值。

但是，如果先创建了 `input_float2`，然后尝试与 `input_float3` 共享其 `source` 值，则不起作用。为了使 `input_float3` 和 `input_float2` 共享相同的 `source` 值，我们首先需要创建 `input_float3`，这可以通过 `add_value()` 方法完成。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6 def print_me(sender, data):
7     log_debug(f"复选框-value: {get_value('value_1')}")
8     log_debug(f"文本-value: {get_value('value_2')}")
9     log_debug(f"颜色-value: {get_value('color4')}")
10
11 def reset(sender, data):
12     set_value("value_1", False)
13     set_value("value_2", "Hello DearPyGui!")
14
```

写下你的评论...

评论0

赞

...

```
20 add_input_text("文本输入 2", source="value 2", password=True, tip="此文本框已隐藏密码")
21 add_button("打印 source value", callback=print_me)
22 add_button("重置 source value", callback=reset)
23 # 将较小的输入类型的控件链接到较大的控件时的特殊情况, 首先创建较大的值
24 add_value("color4", (0.0, 0.0, 0.0, 0.0))
25 add_color_edit3("颜色选择 3", source="color4")
26 add_color_edit4("颜色选择 4", source="color4")
27
28 start_dearpygui()
29
```

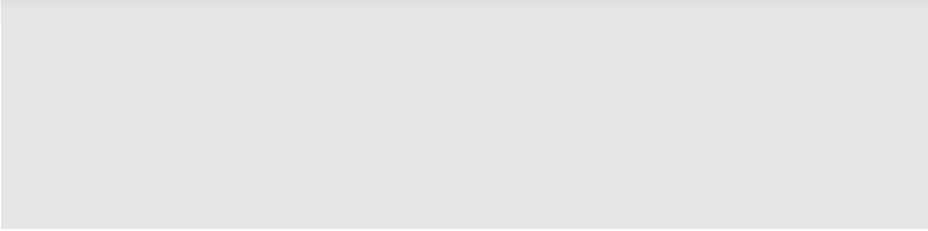
< >

dearpygui_sourceaddvalue.png

Dear PyGui 还支持传入任意 Python 数据对象类型（甚至可以自定义数据类型）用于数据存储。使用 `add_data`，我们可以传入任意数据类型，并通过 `get_data("name")` 进行访问。

```
1 from dearpygui.core import *
2 from dearpygui.simple import *
3
4 add_additional_font('三极中柔宋.ttf', 18, glyph_ranges='chinese_simplified_common')
5
6 def store_data(sender, data):
7     custom_data = {
8         "Radio Button": get_value("Radio Button"),
9         "Checkbox": get_value("Checkbox"),
10        "Text Input": get_value("Text Input"),
11    }
12    add_data("stored_data", custom_data)
13
14 def print_data(sender, data):
15     log_debug(get_data("stored_data"))
16
17 show_logger()
18
19 with window("Tutorial"):
20     add_radio_button("Radio Button", items=["选项1", "选项2"])
21     add_checkbox("Checkbox", label="复选框")
22     add_input_text("Text Input", label="文本输入")
23     add_button("存储数据", callback=store_data)
24     add_button("打印数据", callback=print_data)
25
26 start_dearpygui()
```

< >



dearpygui_adddata.png

0人点赞 >



 Python相关知识



"小礼物走一走，来简书关注我"

赞赏支持

还没有人赞赏，支持一下



何小有
总资产49 (约3.39元) 共写了8.8W字 获得370个赞 共323个粉丝

关注

写下你的评论...

 评论0  赞 ...