

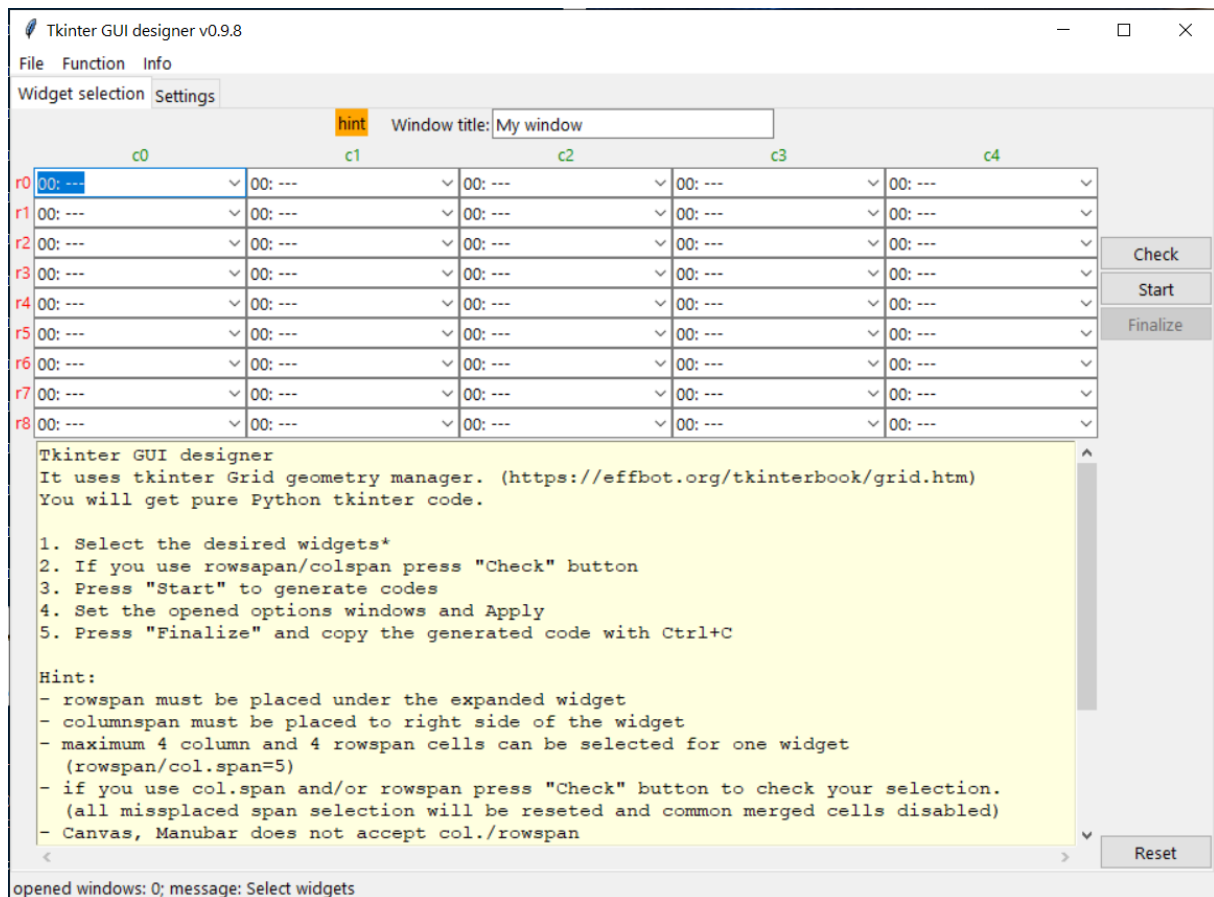
v1.2

Tkinter GUI designer

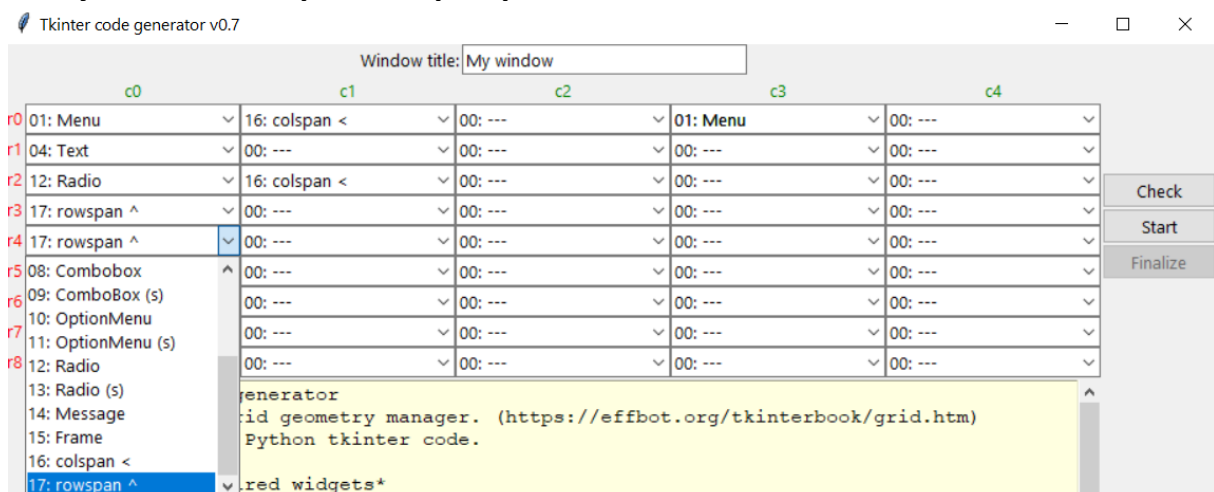
It is written in Python and you will get pure Python tkinter code.

It uses tkinter Grid geometry manager. (<https://effbot.org/tkinterbook/grid.htm>)

1. Select the desired widgets*



2. If you use rowspan/colspan press "Check" button



v1.2

	c0	c1	c2	c3	c4
r0	01: Menu	16: colspan <	00: ---	01: Menu	00: ---
r1	04: Text	00: ---	00: ---	00: ---	00: ---
r2	12: Radio	16: colspan <	00: ---	00: ---	00: ---
r3	17: rowspan ^	00: ---	00: ---	00: ---	00: ---
r4	17: rowspan ^	00: ---	00: ---	00: ---	00: ---
r5	00: ---	00: ---	00: ---	00: ---	00: ---
r6	00: ---	00: ---	00: ---	00: ---	00: ---
r7	00: ---	00: ---	00: ---	00: ---	00: ---
r8	00: ---	00: ---	00: ---	00: ---	00: ---

- All common merged cells will be disabled (for example c1-r3, c1-r4 on screenshot)
- Rowspan must be placed under the expanded widget
- Columnspan must be placed to right side of the widget
- Maximum 4 column and 4 rowspan cells can be selected for one widget (rowspan/colspan=5)
- If you use colspan and/or rowspan press "Check" button to check your selection.
- All misplaced span selection will be reseted
- Canvas and Menubar does not accept col./rowspan
- You can save and load widget selection and default widget parameters
- Generated code can be Function or Object type
- Default paramertes can be modified for simplified (s) widgets

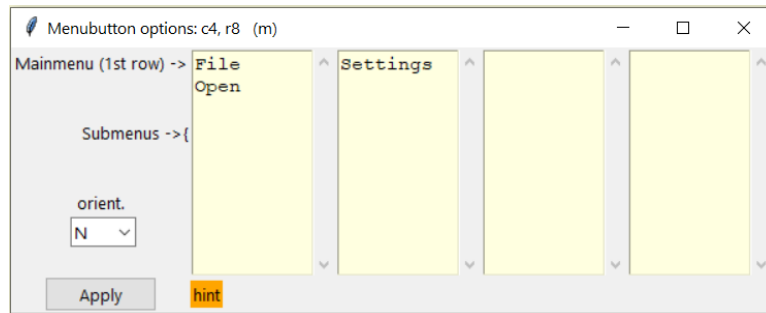
3. Press "Start" to generate codes

Related option widows open

Menubar

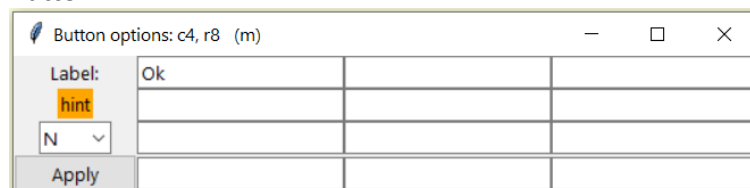
You can put 4 main menu (top blue line) and any submenu
First row: main menu (x4), other rows: submenu (any)
Available settings: separate: '-' ; checkbox: '#' (eg: #yes)

MenuButton



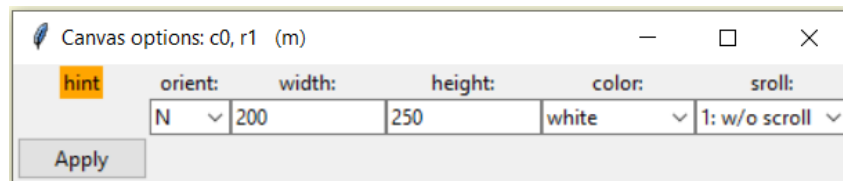
You can put 4 main menu (top row) and any submenu
Available settings: orientation.

Button



You can put 3x4 button in a frame. (min 1 button max 12button)
Available settings: orientation. N- North, E-East, S-South, W-West, NE,SE,SW,NW, N+S
(expand between max N and max S), E+W (expand between max W and max E)

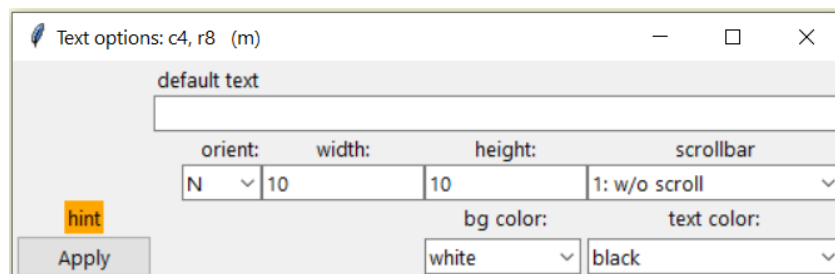
Canvas



Available settings: orientation, width, height and color, scrollbar (without, with horizontal, with vertical or both)

You can select some color, select 'pick a color' to open color selector or write any standard python color code here.

Text



Available settings: default text, orientation, width, height, scrollbar (without, with horizontal, with vertical or both) and background and text color

You can select some color, select 'pick a color' to open color selector or write any standard python color code here.

Label

Available settings: default text and orientation

Combobox

You can place up to 4 comboboxes with unlimited options

Available settings:

Orientation, Cell width, Label (for each combobox), unlimited options (for each combobox)

Option menu

You can place up to 4 option menus with unlimited options

Available settings:

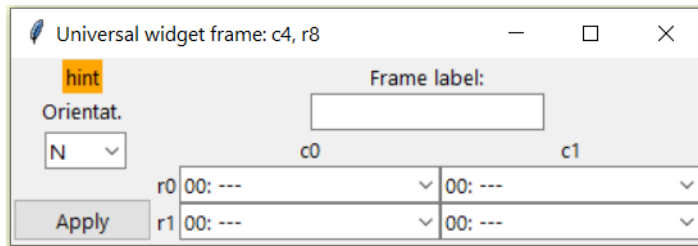
Orientation, Label (for each combobox), unlimited options (for each combobox)

Radio

You can place up to 4 radio with 3 options

Available settings: Orientation, 3 options (for each combobox)

Universal Frame



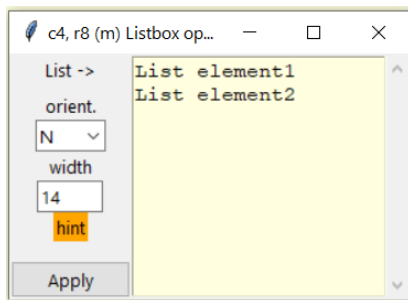
You can place any widget* (up to 4)

*menu (3x3), button (4x3), canvas, text with scroll, entry, label, combobox (4x3), optionmenu (4x3),radio (4x3)

Available settings:

Orientation, 4 widget selection, label text (with text: LabelFrame / without: Frame)

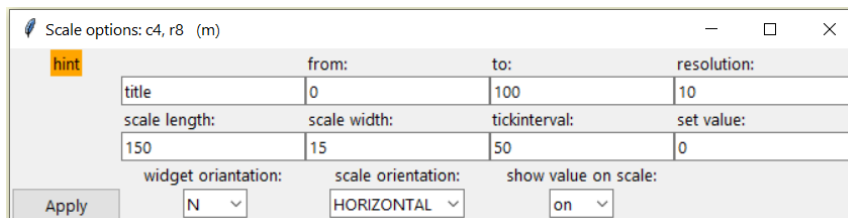
Listbox



You can place any line into the list

Available settings: Orientation, width

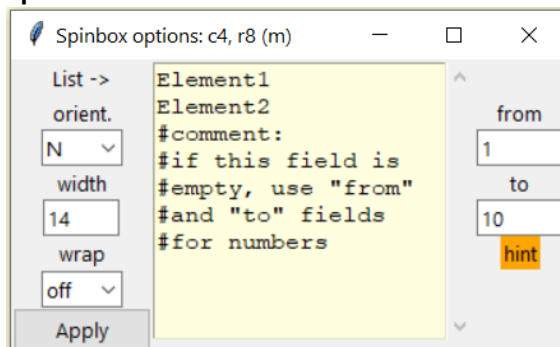
Scale



Available settings:

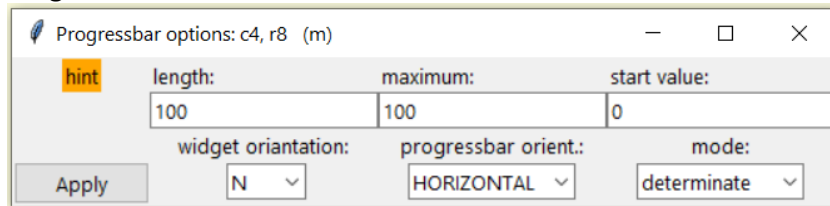
scale label, from, to, resolution, scale length, scale width, thickinterval, set value, widget orientation, scale orientation, show value

Spinbox



Available settings: from, to, wrap, list items, width, orientation

Progressbar



Progressbar options: c4, r8 (m)

hint

length: 100 maximum: 100 start value: 0

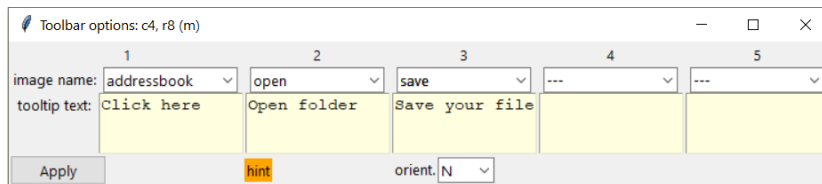
widget orientation: N progressbar orient.: HORIZONTAL mode: determinate

Apply

Available settings:

length, maximum, start value, widget orientation, progressbar orientation, mode

Toolbar



Toolbar options: c4, r8 (m)

	1	2	3	4	5
image name:	addressbook	open	save	---	---
tooltip text:	Click here	Open folder	Save your file		

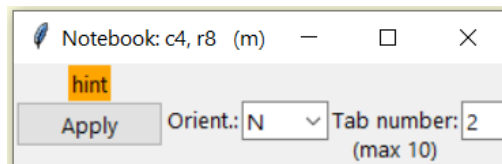
Apply hint orient: N

Available settings:

PNG image select, Tooltip text, orientation, hint



Notebook



Notebook: c4, r8 (m)

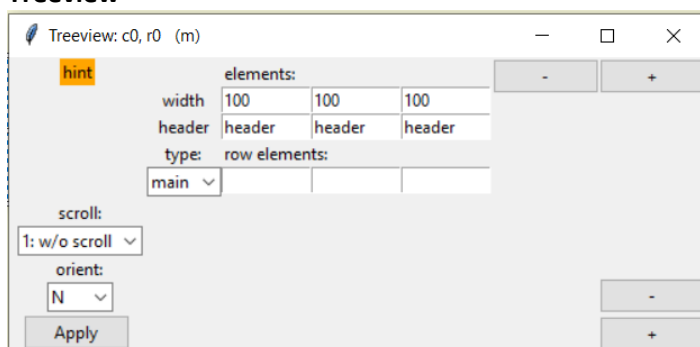
hint

Apply Orient.: N Tab number: 2 (max 10)

Available settings:

Orientation, Tab number -> it will open as many universal frame option windows as tab numbers you set

Treeview



Treeview: c0, r0 (m)

hint

elements:

	width	header	type
1	100	header	row elements:
2	100	header	
3	100	header	

scroll: 1: w/o scroll

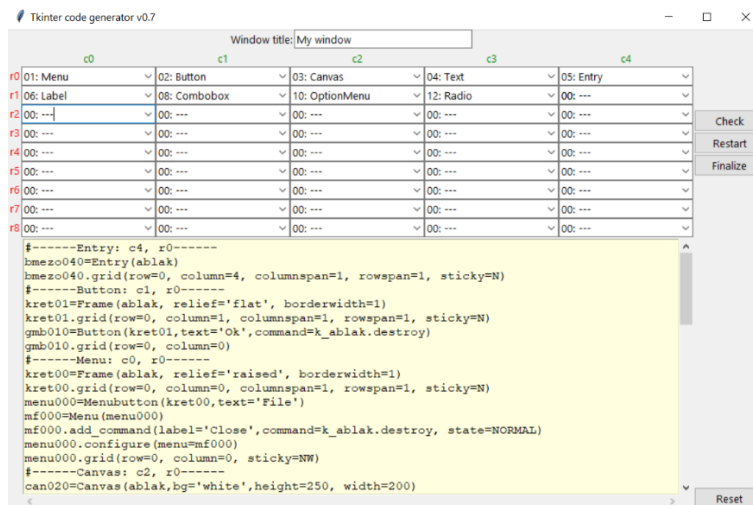
orient: N

Apply

Available settings:

Orientation, scrollbar (without, with horizontal, with vertical or both), unlimited columns, width for each columns, header for each columns, unlimited row, that can be main and sub. (You can leave it empty)

4. Set the opened options windows and Apply



Your code is not done yet, you have to finalize it.

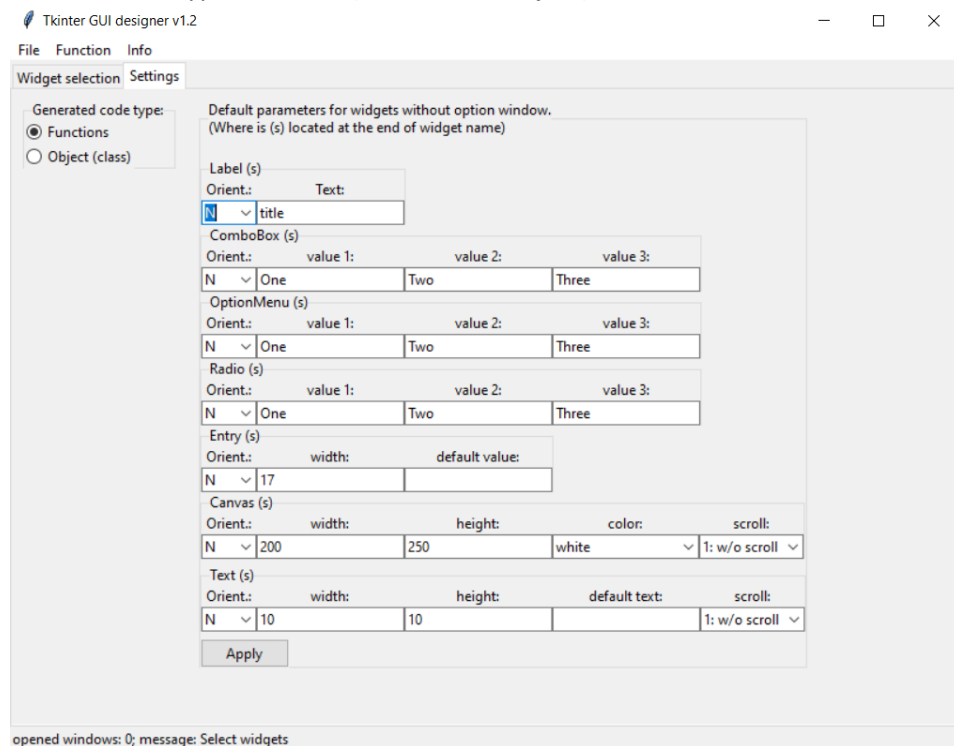
5. Press "Finalize" and copy the generated code with Ctrl+C

Hint:

- rowspan must be placed under the expanded widget
- colspan must be placed to right side of the widget
- maximum 4 column and 4 rowspan cells can be selected for one widget (rowspan/colspan=5)
- if you use colspan and/or rowspan press "Check" button to check your selection.
 - all misplaced span selection will be reseted and
 - common merged cells disabled
- Canvas and Menubar does not accept column/rowspan
- Open second session of Python IDLE and use it for checking the generated code without closing the running GUI generator
- widget name with (s) like 'Label (s)', 'Combobox (s)', 'Optionmenu (s)' and 'Radio (s)' means simple. It uses default parameters and does not open option window.
- Toolbar PNG files must be in ico folder. For example: If path of your file is c:\myprog\myprog.py, PNG files location is c:\myprog\ico. Sample PNG files in the package.
- You can save and load widget selection
- Generated code can be Function or Object type
- Default paramertes can be modified for simplified (s) widgets

Settings tab:

- In case of widgets without option window (where is (s) located at the end of widget name), possible to modify the default parameters
- Generated code type selection (Function or Object)

**Supported widgets:**

manubar (4x any), menubutton (4x any), button (4x3), canvas with scroll, text with scroll, entry, label, combobox (4x3), optionmenu (4x3), radio (4x3), message, Universal Frame, Listbox, Scale, Spinbox, Toolbar with Tooltip text, Notebook, Treeview with scroll

Menu:

- **File:**
 - Open widget selection
 - Save widget selection
- **Function:**
 - Check: validate widget selection
 - Start: Start code generating
 - Finalize: Finalize generated code
 - Reset
- **Info:**
 - Github
 - Email
 - Donation

***Generated code type**

```

File  Function  Info
Widget selection  Settings
Generated code type:
● Functions
○ Object (class)

from tkinter import *
from tkinter import ttk

k_ablak=Tk()
k_ablak.title('My window')
ablak=Frame(k_ablak, relief='flat', borderwidth=1)
ablak.grid(row=0, column=0)
#-m---Menubar: c0, r0---
menubar00aa= Menu(ablak)
submenu00aa0=Menu(menubar00aa, tearoff=0)
menubar00aa.add_cascade(label='File', menu=submenu00aa0)
submenu00aa0.add_command(label='Open', command=
submenu00aa1=Menu(menubar00aa, tearoff=0)
menubar00aa.add_cascade(label='Settings', menu=
k_ablak.config(menu=menubar00aa)
#-m---Text: c0, r1-----
frm10aa=Frame(ablak, relief='flat', borderwidth=1)
frm10aa.grid(row=1, column=0, columnspan=1, row

```

default/unselected: Function

```

File  Function  Info
Widget selection  Settings
Generated code type:
○ Functions
● Object (class)

from tkinter import *
from tkinter import ttk

class Mywindow:
    def __init__(self):
        self.k_ablak=Tk()
        self.k_ablak.title('My window')
        self.ablak=Frame(self.k_ablak, relief='flat', borderwidth=1)
        self.ablak.grid(row=0, column=0)
#-m---Text: c0, r1-----
        self.frm10aa=Frame(self.ablak, relief='flat', borderwidth=1)
        self.frm10aa.grid(row=1, column=0, columnspan=1, row=1)
        self.text10aa0=Text(self.frm10aa,height=1)
        self.text10aa0.grid(row=0, column=0, sticky=N)
#-m---Button: c0, r2-----
        self.btn20aa0=ttk.Button(self.ablak,text='Open')
        self.btn20aa0.grid(row=2, column=0, columnspan=1, row=2)
#-m---Menubar: c0, r0---

```

selected: Object

Note: I am not professional Python programmer, but I hope you can use it and it can help you to create Tkinter GUI much more easier.

*If you have any comment, suggestion write me: epromirok@gmail.com
[https://github.com/horrorfodrasz/Tkinter GUI designer](https://github.com/horrorfodrasz/Tkinter_GUI_designer)*

If you like it you can donate me

Paypal



Thank you!