

# Introduction to GUI Development with Tk and Python 3

Charles Daniels

2019-09-30

# Contents

Introduction

Background

Architecture

Main Window

”Add New” Dialog

End

# In Today's Talk...

- ▶ Rapid GUI development with TkInter and Python 3
- ▶ Some familiarity with Python would help
- ▶ Building GUIs doesn't have to be hard!
- ▶ Today's demo: a simple "sound board" program

## In Today's Talk...

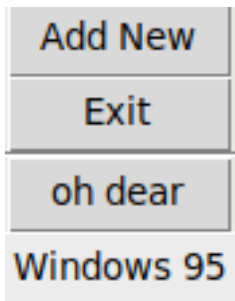


Figure: The finished application

# Background

- ▶ Tcl is one of the older embed-able scripting languages
- ▶ TkInter is a package which integrates Tcl into Python
- ▶ Tk is a *widget toolkit* (such as WPF, GTK, or qt)
- ▶ Through TkInter, we can use Tk from within Python programs

# Why TkInter?

- ▶ Python is great for rapid development
- ▶ Tk is simple and mature, and so is TkInter
- ▶ Not as fast/fancy as other toolkits, but very easy to work with

# Architecture

- ▶ Main window will have "exit" and "add new sound" buttons
- ▶ Main window will also have one button for each sound in the db
- ▶ Database will be a directory filled with .wav files
- ▶ Clicking "add new" will let us install new sounds in the db
- ▶ Clicking a sound's button will play it

# Imports

```
1 import tkinter
2 import tkinter.ttk
3 import tkinter.filedialog
4 import appdirs
5 import pathlib
6 import shutil
7 import simpleaudio
8 import sys
9
10 global_root = None
11 file_dialog_root = None
```



# Controls

```
1 root = tkinter.Tk()
2 root.title("soundboard")
3
4 # this will let us close and restart the GUI from elsewhere
5 global global_root
6 global_root = root
7
8 # "add new" button
9 add_new_btn = tkinter.ttk.Button(root, text="Add New",
10     command=show_add_new_dialog)
11 add_new_btn.grid(row=0, column=0)
12
13 # exit button
14 exit_btn = tkinter.ttk.Button(root, text="Exit", command=sys
15     .exit)
16 exit_btn.grid(row=1, column=0)
17
18 # put a separator after the "Add New" button
19 sep = tkinter.ttk.Separator(root, orient=tkinter.HORIZONTAL)
20 sep.grid(row=2, column=0, sticky="ew")
```

# Sound Buttons

```
1 # generate a button for each element
2 row = 3
3 for path in get_db().glob("*"):
4     if not path.is_file:
5         continue
6
7     # callback will be the function that get's called when
8     # the button is
9     # clicked, in this case it just calls play_sound on the
10    # file that
11    # corresponds to this button
12    callback = lambda : play_sound(path.stem)
13
14    # create the button
15    btn = tkinter.ttk.Button(root, text=path.stem, command=
16    callback)
17    btn.grid(column=0, row=row)
18
19    row += 1
```

## "Add New" Dialog Window

```
1 def show_add_new_dialog():
2     root = tkinter.Tk()
3     root.title("add new sound clip")
4
5     # allow this window to be closed
6     global file_dialog_root
7     file_dialog_root = root
8
9     # "From File" button
10    from_file_btn = tkinter.ttk.Button(root, text="Add From
11        File",
12        command = from_file_callback)
13    from_file_btn.grid(row = 0, column = 0)
14
15    root.mainloop()
```

## "From File" Callback

```
1 def from_file_callback():
2     # load the sound file
3     add_sound_from_file("./", GUI=True)
4
5     # close the file picker dialog
6     global file_dialog_root
7     file_dialog_root.destroy()
8
9     # kick over and restart the main window
10    global global_root
11    global_root.destroy()
12    show_GUI()
```

# Add New Sound to Database

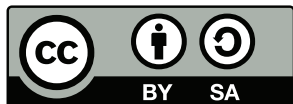
```
1 def add_sound_from_file(path, GUI=False):
2     if GUI:
3         # when loading a file via the GUI, use the path
4         # argument as the initial
5         # directory, and display a file picker dialog
6         # limited to just WAV
7         # files, since that's all we support
8         tkinter.Tk().withdraw()
9         path = tkinter.filedialog.askopenfilename(
10             initialdir=str(path),
11             filetypes=[("Audio Files", "*.wav")],
12             title="Choose a File")
13
14         if path == ():
15             # user selected "cancel"
16             return
17
18     path = pathlib.Path(path)
19
20     # guarantee the database exists
21     get_db().mkdir(exist_ok=True)
22
23     # copy file into database
24     shutil.copy(str(path), str(get_db() / path.name))
```

# Questions?

End

# License

This work is licensed under a Creative Commons “Attribution-ShareAlike 4.0 International” license.





## soundboard.py

For the sake of anyone who winds up with just this PDF, the full `soundboard.py` code is included on the next slide as a QR code. It is xz compressed then base64-encoded. The xz file should have the shasum `62cd5ffe43bfff149ce58bb383a44cdc6d42cd3`. The source code file `soundboard.py` has the shasum `666e22a03bfc28351ed13e6febd5c4dc6c3e0f78`.

An example of a pipeline extracting this information from a screenshot follows: `zbarimg --quiet --raw screenshot.png | base64 -d | xz --decompress > out.txt`

soundboard.py

