Scanner with Tcl/Tk

A user interface without any hassle wiithin 5min

Onkobu Tanaake

2018-09-20T10:00:00

I decided to go with pure Tcl/Tk because there are only linux computers in our household. The program has a simple structure and may offer a way into programming. The basic structure is taken from a documentation[1].

Figure 1. Spaltenweise von links nach rechts, dann Scannen

Figure 2. Columns from left to right, finally scan

```
# Window title
wm title . "Aufgaben Scannen"

# Frame definition
grid [ttk::frame .c -padding "3 3 12 12"] -column 0 -row 0 -sticky
grid columnconfigure . 0 -weight 1; grid rowconfigure . 0 -weight

# Descriptive text
grid [ttk::label .c.lblDesc -text "Wähle von links nach rechts und

# First column
grid [ttk::label .c.lblFach -text "1. Fach"] -column 1 -row 2 -sti
grid [ttk::label .c.lblArt -text "2. Art"] -column 2 -row 2 -stick
grid [ttk::label .c.lblEingabe -text "3. Seite/ Titel"] -column 3

# ... more ui elements ...

# The button to scan
grid [ttk::button .c.scan -text "Scannen" -command {scan $fach $ar
```

```
proc scan {fach art extra} {
  # pre-conditions omitted
  try {
    set results [exec $command -d $scanner --format jpeg --resolut
    set status 0
  } trap CHILDSTATUS {results options} {
    set status [lindex [dict get $options -errorcode] 2]
  }
}
```

These few lines are the program's basic structure. Only the other columns are missing as well as a few lines of fear-driven code to catch errors and validate input. The button starts the sub command to commandeer the scanner. Its output goes into a folder with the week's number. Grid layout looks like this:

Figure 3. Ganze UI mit eingezeichnetem Grid-Layout

Figure 4. Entire UI with outlined grid layout

My target audience was quite fast understanding the principle of operation: choose from left to right. Text input receives a task's title or page number. Date and number of week are derived automatically. A starter on the desktop decorated with a scanner image are easy to invoke. It only requires a few minutes of work for my daughter to scan a week's work.

Tkinter is not necessary to use themes. And it requires only 4 lines of code. I prefer the classic *ttkthemes*[2]. The Python code is not necessary. Place a symbolic link to the folder `ttkthemes` into `.local` and append to lookup path:

```
lappend auto_path "~/.local/ttkthemes/themes/"
package require Tk
package require ttk::theme::plastik

ttk::style theme use plastik
```