Tkinter (continuation)
Tasks

- Implement text editor (ala Windows notepad)
- Implement URL downloader (like a web browser but without rendering html)
  - Entry for URL
  - ScrolledText for result
- Implement Minesweeper
Help for tasks

- Manipulating text in ScrolledText
  - clean – text.delete(1.0, END)
  - get – text.get(1.0, END)
  - insert
    text.insert(INsert, text)
    text.insert(END, text)
Alarm callbacks

- `widget.after(delay_ms, callback, *args)`
  - registers an alarm callback that is called after a given time
- `widget.cancel(id)`
  - cancels an alarm callback

See `e01_text_tick_after.py`
Alarm callbacks

- Can be used for animations

See e02_button_animation.py
Canvas

- Widget for displaying drawings

```python
master = Tk()

w = Canvas(master, width=200, height=100)
w.pack()

w.create_line(0, 0, 200, 100)
w.create_line(0, 100, 200, 0, fill="red", dash=(4, 4))

w.create_rectangle(50, 25, 150, 75, fill="blue")
```
Canvas

- drawing
  - arc
  - bitmap
  - image
  - line
  - oval (a circle or an ellipse)
  - polygon
  - rectangle
  - text
Canvas

- Added items (drawings) are kept until removed
  - can be modified

```python
i = w.create_line(xy, fill="red")

w.coords(i, new_xy)
w.itemconfig(i, fill="blue")

w.delete(i)

w.delete(ALL)
```

See e03_canvas_move.py
Canvas

• Items can have tags
  ▪ tag ~ symbolic names

```
item = canvas.create_line(0, 0, 100, 100, tags="uno")
canvas.itemconfig(item, tags=("one", "two"))
canvas.addtag_withtag("three", "one")
items = canvas.find_withtag("one")
```
Misc

- Fullscreen app
  - `root.attributes("-fullscreen", True)`

- Mouse cursors
  - `widget.configure(cursor="cross")`
    - "arrow" "circle" "clock" "cross" "dotbox" "exchange" "fleur" "heart" "heart" "man" "mouse" "pirate" "plus" "shuttle" "sizing" "spider" "spraycan" "star" "target" "tcross" "trek" "watch"
Task

- Create a simple “shooting” game
  - targets shot from the corner at random direction
  - destroyed by mouse clicks