Tkinter: Canvas

Introduction to Software Engineering

Lecture 11B

May 9, 2023

Canvas

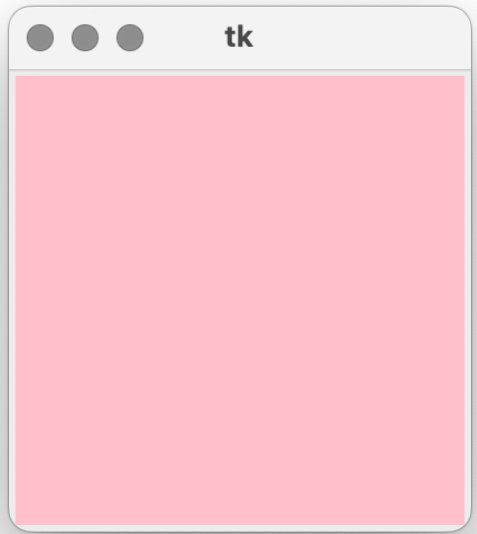
A `tkinter` *canvas* is a window in which we can draw.

The following can be drawn in canvases:

- arc bitmap,
- images,
- lines,
- polygons,
- rectangles,
- text,
- ovals,
- window.

tk.Canvas

```
1  root = tk.Tk()
2
3  canvas = tk.Canvas(
4      root,
5      width=100,
6      height=100,
7      bg="pink"
8  )
9  canvas.pack()
10
11 root.mainloop()
```

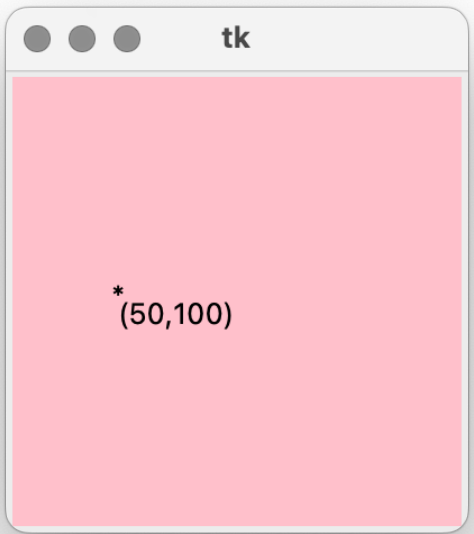


We assume the above has been executed from now on and will *insert new code* before `root.mainloop()`.

tk.Canvas.create_text

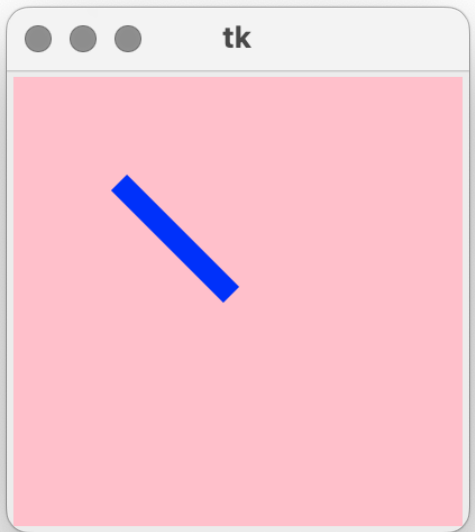
Note we use the *graphics coordinate system* where the top-left corner is designated $(0, 0)$ and increases in the y-coordinate moves you *south*.

```
1  canvas.create_text(
2      50, 100,
3      text="*",
4  )
5
6  canvas.create_text(
7      50, 100,
8      text="(50,100)",
9      anchor=tk.NW
10 )
```



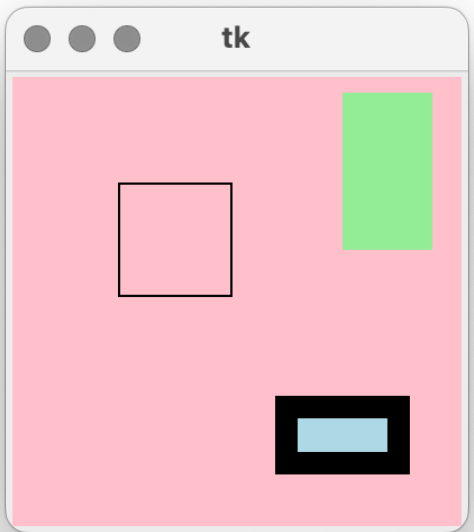
tk.Canvas.create_line

```
1 canvas.create_line(
2     50, 50,
3     100, 100,
4     fill="blue",
5     width=10
6 )
```



tk.Canvas.create_rectangle

```
1  canvas.create_rectangle(
2      50, 50,                    north-west corner
3      100, 100,                  south-east corner
4      width=1)                       border width
5
6  canvas.create_rectangle(
7      125, 150, 175, 175,
8      fill="light blue",             fill colour
9      width=10)
10
11 canvas.create_rectangle(
12     150, 10, 190, 80,
13     fill="light green",
14     width=0)
```



tk.Canvas.create_window

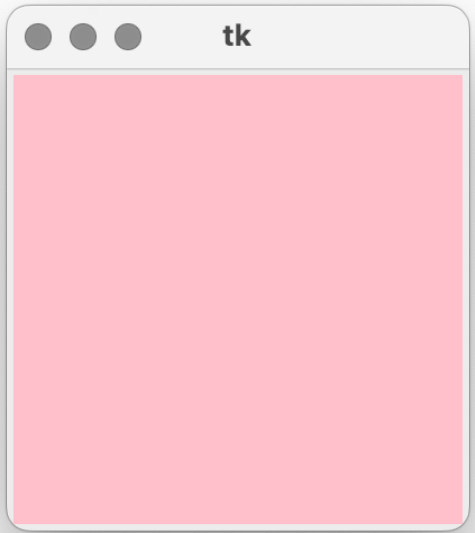
`tkinter` widgets can be placed in canvases using `window`.

```
1  window = canvas.create_window(
2      50, 50,
3      width=100,
4      height=50,
5      window=tk.Button(text="Press Me"),
6                          Any widget can go here.
7      anchor=tk.NW
8  )
9
```

bind in tk.Canvas

We can `bind` inside canvases.

```
1 def handler(event):
2     print(f"event.x event.y")
3 canvas.bind("<Button>", handler)
```



Clicking inside the canvas produces this in the REPL:

```
1 178 155
2 81 89
3 151 83
```

Inheritance

Do not forget that `tkinter` objects are objects which we can be used for inheritance.

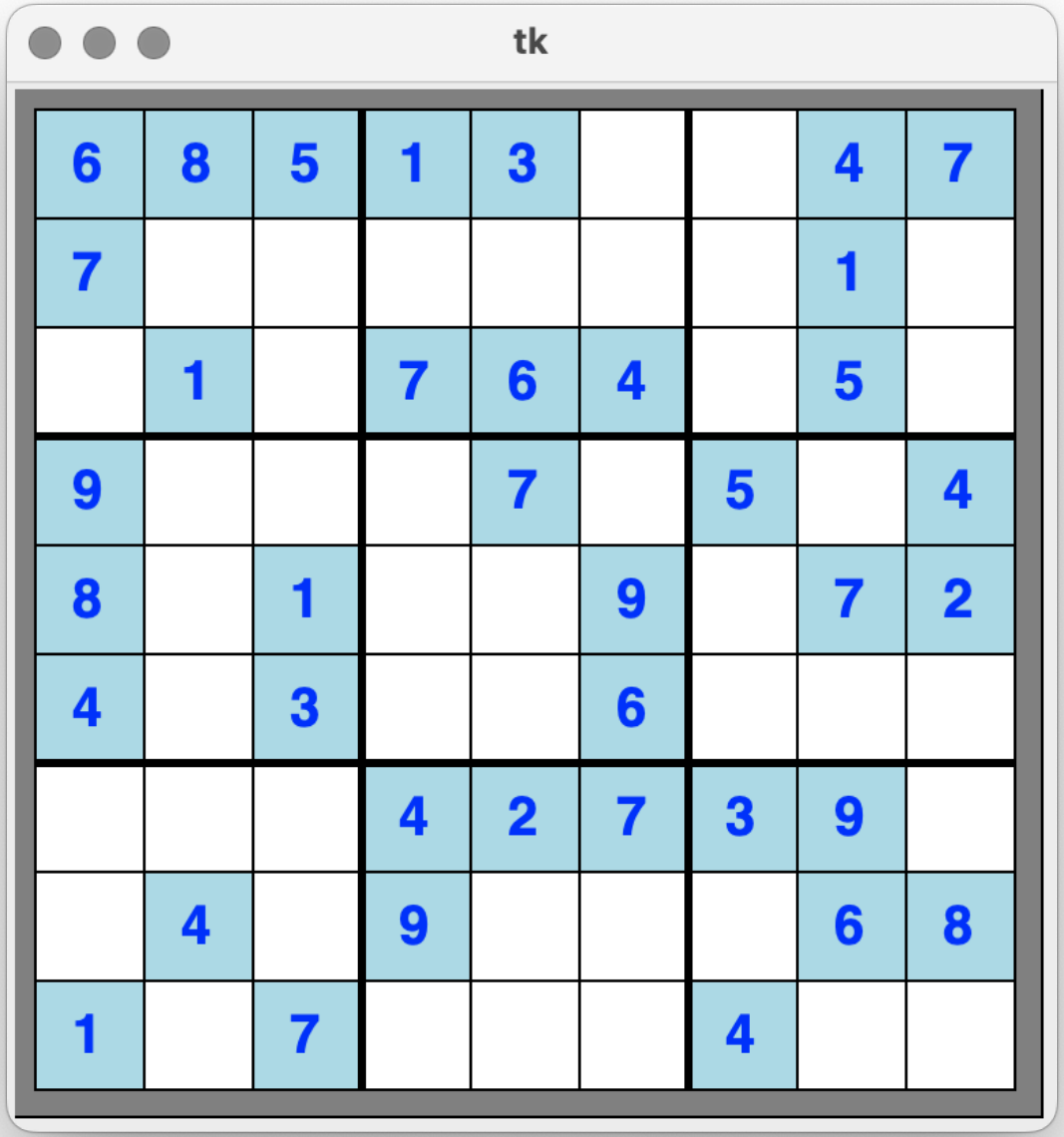
```
1  class View(tk.Canvas):
2      def __init__(self, master: tk.Tk, side: int, **kwargs) -> None:
3          super().__init__(master, width=side, height=side,
4            bg="pink", **kwargs)                          keyword arguments
5
6          self.create_text(50, 50, text="*", fill="black")
7
8  root = tk.Tk();
9  view = View(root, 200)
10 view.pack();
11 root.mainloop()
```

Abstract Grid

We are using an `AbstractGrid` class which inherits from a canvas for assignment 3. Which *extends* a canvas to include methods that interface with the canvas as if it were divided into a *grid* of smaller and equally sized rectangles.

```
1 class AbstractGrid(tk.Canvas):
2     """ A type of tkinter Canvas that provides support for
3         using the canvas as a grid (i.e. a collection of
4         rows and columns). """
```

Suduko from 2022 Sem 2 – A Tkinter canvas



Fortnight in 2023 Sem 1 – A Tkinter canvas