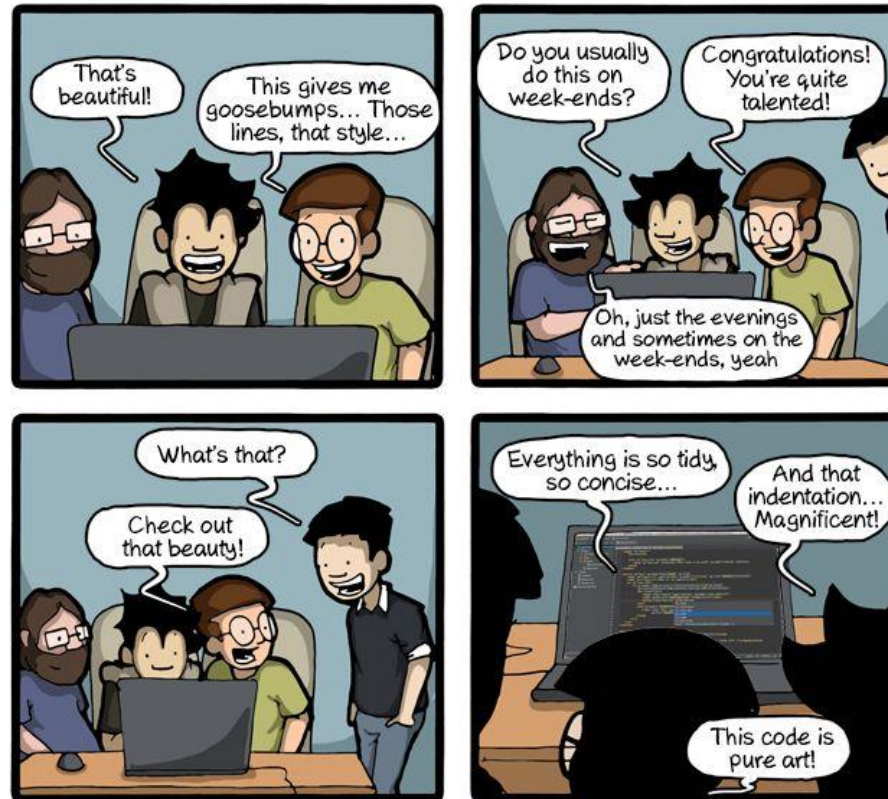


CS 115, Autumn 2021

Lecture 36: GUIs



CommitStrip.com

Thanks to Marty Stepp and Stuart Reges for parts of these slides

Exercise: `drawing_panel`

- We used `drawing_panel` a lot this quarter. How does it work?
 - It is a window filled with a drawing `canvas`
 - Syntax:
panel_variable =
 `add_canvas`(**container**, **width**, **height**, **background_color**)
 - You can call any function on **panel_variable** that you could call on `drawing_panel`.

Reacting to Clicks

- In order to be able to create Paint program, we will need to make the `canvas` react when we click.
 - Add a reaction by setting the function that should run when a particular user action occurs:

panel_variable [event_type] = function

- Event types:
 - "press" – mouse button is pressed down
 - "release" – mouse button is lifted up
 - "move" – mouse is moved

Exercise: drawing

- How can we draw a line that follows the mouse when the user holds it down and drags?
 - What mouse event types will we need to react to?

Parameters and events

- What happens if we write the following?

```
hello['command'] = print_something("hi")
```

- This calls the function `print_something` passing it "hi" right now – it doesn't set it to be called when `hello` is clicked
- How can we pass parameters without parentheses?
 - We can't so we need lambda notation

Lambda functions

- Lambda functions – small anonymous function that can be stored in variables
 - You can call another function from within one and have access to the variables in the function you declare it in

- Example:

```
from ECGUI import *

def print_something(name):
    print("hello " + name.get())

def main():

    window = make_window()
    hello = add_button(window, "hello")
    name = add_entry_box(window)
    hello['command'] = lambda: print_something(name)
    window.mainloop()
```