Institut für Informatik **Ludwig-Maximilians-Universität München**
Prof. Dr. Francois Bry

Higher level languages: Rust, WS 18/19
Tutorial 4
November 07, 2018

Exercise 4-1 Fractal image generator

In this exercise we will get acquainted with the Rust module system. The implementation of a graphical user interface requires the use of several modules, and so we will be implementing a visualization for the Collatz fractal. An overview of the Collatz fractal can be found on the Wikipedia page
`https://en.wikipedia.org/wiki/Collatz_conjecture#Iterating_on_real_or_complex_numbers`.

Iterating the Collatz map

$$\frac{1}{4}(1 + 4z - (1 + 2z)\cos(\pi z))$$

in the complex plane yields the Collatz fractal. The number of iterations required for any point in the plane to establish convergence or divergence determines the hue of the pixel.

The file `collatz.tar.gz` contains a small program that renders the Collatz fractal on the complex plane. However, some elements have been removed from the program. Complete the following tasks to restore the source code and make the program run again. In the source code missing elements are marked with the letter of the corresponding subtask.

a) Find a suitable data type to represent complex numbers (lines 7–22, 49). The crate defining the appropriate data type also needs to be added to the Cargo file. Finally, the definition for the variable `pi` needs to be completed (line 18).

b) Complete the loop that calculates the hue of a certain pixel by iterating the optimized Collatz map (ls. 54–56).

c) Find the correct type and values for the mouse click event handling (ls. 122, 133–137). Hint: Have a look where the function is called. You might also need to take a look at the documentation of the Gtk crate.

d) Complete the `draw` function by filling in the `if`-statement (ls. 154–155) and adding parameters to the call of `set_source_pixbuf` (l. 157).

e) Gtk is driven by an event loop which terminates the program as soon as the event loop is terminated. Find the Gtk function that is responsible for terminating the event loop and add it to the closure which is executed in case the window close button is pressed (l. 169).

f) Bonus exercise: Complete the `resize` function in line 105: The fractal should not become distorted when the window is resized.