Advanced Programming in Python, Group B
Jacobs University Bremen
Dr. Kinga Lipskoch

Course: JTSK-350112
April 27th, 2018

Assignment 4 - Interactive Graphics

- The problems of this assignment must be solved in Python.
- The TAs are grading solutions to the problems according to the following criteria: https://grader.eecs.jacobs-university.de/courses/350112/2018_1gB/Grading-Criteria-Python.pdf

Problem 4.1 *Squares* (1 point)

Presence assignment, due by 18:30 h today
Download the files:
https://grader.eecs.jacobs-university.de/courses/350112/python/graphics/movecircle.py
https://grader.eecs.jacobs-university.de/courses/350112/python/graphics/graphics.py
Change the example program `movecircle.py` in the following way:

a) Make it draw squares instead of circles,

b) Each successive click should create an additional square on the screen (instead of moving it).

Upload one file: `clicksquares.py`.

Problem 4.2 *Draw face* (1 point)

Presence assignment, due by 18:30 h today
Write a program that draws some sort of a human face (for example use circles for head and eyes, triangle for the nose, and combine multiple shapes like circle and rectangle for the mouth, circles or something else for the ears). You can use other shapes as mentioned as well. Minimal requirements: head, eyes, nose, mouth and ears.
Upload one file: `drawface.py`.

Problem 4.3 *Target* (1 point)

An archery target consists of a central circle of yellow surrounded by concentric rings of red, blue, black, and white. Each ring has the same "width", which is the same as the radius of the yellow circle.
Write a function `draw_archery()`, which draws several circles forming the target. The data for the individual circles should be taken from a list of tuples (containing coordinates of the center and the radius), instead of separately hardcoded circles. The values inside of the list of tuples can be hardcoded.
For example call `draw_archery([(100, 100, 15), (100, 100, 30),`
or `(100, 100, 45), (100, 100, 60), (100, 100, 75)])`
instead of `draw_archery(100, 100, 15, 30, 45, 60, 75)` or similar.
Upload one file: `target.py`.

Problem 4.4 *Rectangle by click* (1 point)

Write a program that draws a rectangle by two mouse clicks which determine two opposite corners of the rectangle.
Write also two function for computing and returning the perimeter `2 * (length + width)` and the area `length *width`. The perimeter and area should be is printed on the screen outside of the functions.
Upload one file: `clickrect.py`.

Problem 4.5 *Playing craps with graphics* (1 point)

Download the files:
https://grader.eecs.jacobs-university.de/courses/350112/python/die.py
https://grader.eecs.jacobs-university.de/courses/350112/python/craps.py

Implement the craps game in a graphical manner. Additionally write a program called `game.py` which should create a window of the size of your choice together with three text fields to show the values of the two dice as well as their sum together with corresponding labels as well as a result of the game which will be either ”`Won`” or ”`Lost`”. If the game requires more steps to win or loose then click somewhere in the window for the next roll to happen. Create a field (like a button) with the label ”`Play again`”. Clicking inside of this field should start a new game. Also create a field with the label ”`Exit`”. Clicking inside of this field should stop the application and close the window.
Upload three files: `die.py`, `craps.py` and `game.py`.

Bonus Problem 4.6 *Simple calculator* (2 points)

Write a program with a window, two or more input fields which can act as a simple calculator for floats. You enter two values and an operation (+, −, * or /) and on click the result is showed as text or similar. Use exceptions to make sure that the input is valid (if not repeat) and to make sure that division by zero will not crash your program. The calculator should work for at least one set of input. The program should recompute on new input and/or new click.
Upload one file: `calculator.py`.

How to submit your solutions

Name the programs a4_*x*.py.
Each program **must** include a comment on the top like the following:

```
# JTSK-350112
# a4_1.py
# Firstname Lastname
# myemail@jacobs-university.de
```

You have to submit your solutions via *Grader* at
`https://grader.eecs.jacobs-university.de`.

If there are problems (but only then) you can submit the programs by sending mail to
`k.lipskoch@jacobs-university.de` **with a subject line that starts with `JTSK-350112`**.

Please note, that after the deadline it will not be possible to submit solutions. It is useless to send solutions then by mail, because they will not be accepted.

Your code must compile without any warning under python3.x.

This assignment is due by Wednesday, May 2$^{\text{nd}}$, 10:00 h