The deadline for this exercise sheet is **Monday, 26.06.2018, 14:00.**

Introductory Words

Remember that you need proper documentation to pass the homework. The documentation doesn't need to be *perfect*, but everything that needs a docstring, should have a docstring.
Make sure that your code does not crash on execution! If your code is not runnable – and especially if it is only a typo, and there are no comments to explain your idea or why it was not fixable **we will consider that particular task as failed**.
Task 1 and 2 are not dependent on each other. We supplied data to do task 2.

Precautions

Install one of the frameworks – either PsychoPy or Expyriment.
You can install them via pip easily by using either `pip install expyriment` or `pip install psychopy`. In the best case this will run without errors, and the library should be installed and you can proceed with setting up your experiment!

1 The Great Experiment

Let's design a small experiment: We want to find out whether we take longer to react to a stimulus the farther it is away from the center of our vision.
The experiment will go as follows: we are constantly showing a fixation stimulus in the middle of the screen and then display a circle after a (seemingly) random amount of time between 1 and 6 seconds at a (seemingly) random position. The subject will keep their eyes fixated on the stimulus in the middle and press the left or the right key whenever they notice a circle appearing. After the key press, the offset of the stimulus from the middle of the screen and the reaction time (in milliseconds) will be logged to a .csv file.

1.1 Guidelines and technical stuff

- Make the experiment be displayed in a `1024x768` WINDOW with a light grey background.
- The fixation stimulus is a black small circle or cross in the center of the window.
- You can choose the exact size of the circles, but the radius should be around 25px. The circles will be completely black, meaning black outline and fill.

- There will be **one block** with **10 trials** (for simplicity). You predefine the single trials – each subject has to see the exact same combinations of circle positions and offset times (the times after which the circles are shown), but for each subject you randomize order in which these position-timing tuples will appear.
- The experiment will exit after the user pressed the 'q' key (for 'quit')
- We came up with some conditions for you so you don't have to. The time values for the pause are in seconds, and the circle position is in pixels, Set the units in your code and configurations accordingly where appropriate.

time offset (pause)	circle position x	circle position y
4.5	128	64
2.5	200	200
5	-256	-192
1	256	-350
5	-200	200
2.25	-128	350
2.5	200	-200
1	-128	-64
4	-450	0
2	450	0

- Most importantly: You can choose whether you want to use PsychoPy or Expyriment to implement this program! Depending on which you choose, pay attention to the programming and structure guidelines we've formulated for the respective library.

1.2 Onto the Coding

Now that everything important has been said, you would already be free to program your experiment. With the following subtasks we are providing a guideline of how you could tackle this task. You don't have to follow these, but we strongly recommend it. You can also use these as checkpoints for your task.

a) The Window
No experiment without a window. First, define your experiment window (and other necessities you might need to principally run the experiment). Take a look at the guidelines for what the window should look like and implement it accordingly.
Run your experiment to make sure it works (at this point, not much will happen – this is to make sure your code is not crashing).

b) Stimuli Part 1
Now define the fixation stimulus. Draw it in the middle of the screen. Make the program wait for a key input from the user after drawing, so you can check if it worked. So the experiment should open, you should see a window with the fixation stimulus in the middle and upon pressing some key, the

program should close again.
Run your code to see if the experiment behaves correctly.

c) Stimuli Part 2
Now define the circle stimulus. For now, draw it somewhere on the screen, but at a different position from the cross, so they don't overlap each other. Run your code and make sure the fixation and the circle can be shown at the same time.

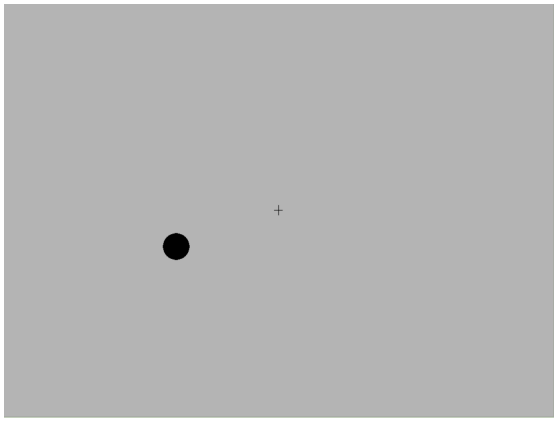

Figure 1: An example how it could look after Stimuli Part 2

d) The Loop
Include the circle position and time offset conditions into your code. You might want to use a dictionary and a list for this. In the list, store the keys of the dictionary. This is what you will iterate over in your experiment. You can simply use the integer from 1 to 10 (the trial numbers) as keys. The corresponding values for each trial key, are tuples consisting of the horizontal circle position, the vertical circle position and the time offset before the circle in this trial is shown.
Now write a loop to repeat for as many times as you have conditions. In each iteration draw the fixation cross/point and circle at the same position as you have done in the previous step, and wait for a key press from a user. You should be able to press keys 10 times before the experiment ends.
Test your program.

In each iteration step, you get the next index from your list (you may start doing this sequentially). Then use this index to access the corresponding dictionary value. For now, merely print the dictionary contents to the terminal to check that the values are retrieved correctly.
Again, test your program.

Now change it up! Instead of going through the indices sequentially, obtain the next index randomly. Depending on the framework you chose, there are

different ways to implement this. Make sure that each index is used exactly once (for example by printing the used index to the terminal).
And one more time. Test your program.

e) Using the values
Next up: Moving the circle. Get the new circle position from the tuple, that you got out of the dictionary. Change the circle position and draw the circle at its new position – don't forget the fixation point!
Test your program. Does it move?
One value remains. Get the time offset and include it as a pause in your loop. The pause should hold as many seconds as the time offset states and needs to be held *before* the circle is drawn.
Test again. Are the pauses working? If they do: Awesome! The hardest part is done!

f) The time
The next step is to measure the reaction time and convert it into a millisecond value. Cut off all decimal places and make it an integer value.

g) Data logging
Using either the file writing methods you are familiar with or the io capabilites of the library you are working with, prepare a csv file. This means including headers in the first row that define the table. Make sure there are no unnecessary spaces in the file especially around the commas!
In this file, log the x and y position of the circle and the reaction time of the user, that they needed until they pressed a key, after each trial.
Did the writing work?

h) Final touches
Add the global quitting key 'q', so that the experiment can be cancelled at any time. Change the key presing function, so that it will wait for a key press of the 'left' or 'right' key.
Clean up your code. Remove reamining print functions, and add documentation where missing. It is an option to encapsulate part of the work in functions, but these kind of scripts are also working well as a procedural script simply running from top to bottom. Hence, do not worry if you do not find ways to split your code in a way that makes sense and is efficient.

Alright, you got the experiment set up and going. Awesome! Now we need to actually collect some data with this experiment. Run the experiment and either do it by yourself or get someone else to do it for you.

2 Looking at Data

If you successfully completed the first part you will have a csv file with your experiment data, otherwise you can use the csv file from the .zip archive, which contains sample data.

Write a new script `analysis.py`, in which you import pandas, numpy and pyplot. Using pandas read in the csv file you created into a dataframe. Inspect the dataframe and check whether the data was loaded correctly.
Since we are interested in the reaction time in dependency of the distance of the circle from the fixation cross, we need to calculate this first. We have the x and y coordinates and can use the euclidean distance formula: $d = \sqrt{x^2 + y^2}$
Since numpy can work on all entries of a series at once, use numpys square root function to calculate this. Pyplot works well with Pandas! You can use the newly calculated distance as the x values, and the reaction times as y values in a scatter plot.
Maybe you can identify a trend?

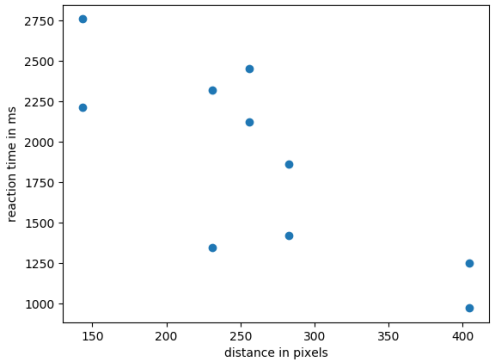

Figure 2: An example result of plotting the reaction time against the circle distance