Peer-review of assignment 4 for *INF3331-anettfre*

Reviewer 1, kasperfu, kasperfu@student.matnat.uio.no
Reviewer 2, magnumhu, magnumhu@student.matnat.uio.no
Reviewer 3, krisgarv, krisgarv@student.geo.uio.no

October 12, 2018

1 Review

VERSION: This review is done on windows 10, with python version 3.6.5, Anaconda, with visual studio code.

Assignment 4.1

Works well. The code is well-documented, and you have docstrings for all the functions. Although the convention for multi-line docstrings in python is to have a short summary one the first line, then have a blank line followed by a more elaborate explanation. See: https://www.python.org/dev/peps/pep-0257/
The code is nice and compact, and easy to understand. The variable- and method-names are good. I have no suggestions for improving this part, well done!

Assignment 4.2

Assignment 4.2 works great. The code is well explained and documented, but the same point about docstrings from 4.1 applies here. Example:

```python
def make_set(min1, min2, max1, max2, z1=1000, z2=1000, numberOfIerations=1000):
    '''Makes points and calls mandelbrot and color function, returns values.

    Since for-loops take time i will try to awoid them since numpy can add array whitout loops
    To create c I make array b imaginary by multiplying b with i
    z1 and z2 is resolution
    '''
    a = np.linspace(min1, max1, z1) #real
    ...
```

Otherwise well done! You have successfully vectorized the computations in the functions and achieved a much quicker run-time. The code is easy to read and names are good. I don't have any suggestions for a simpler solution.

Assignment 4.3

Well done.

Assignment 4.4

Not answered.

Assignment 4.5

There is a small error in the program that prevents you from specifying the filename, if you call the program with only a filename argument. In mandelbrot.py line 32, in the showImage(set, filename) function, you use quotes around filename in the call to plt.savefig(). This causes the output-filename to always be "filename" instead of whatever filename the user provided. Just remove the quotes and it works perfectly. Also, the case where you invoke the program with all possible arguments, ie. method, region, resolution etc. the program crashes. Also the output-filename would be incorrect here, since it is never set.
The problem here is line 66, where you have saved the program version to run as a string. But then you are trying to call the make set(....) method directly on the string. This doesn't work since string have no such method. You have to use the same technique that you used where the user only gives method-name. An alternative version:

```
elif len(sys.argv) > 8: #all
    method = sys.argv[1]
    min1 = float(sys.argv[2])
    min2 = float(sys.argv[3])
    max1 = float(sys.argv[4])
    max2 = float(sys.argv[5])
    z1 = int(sys.argv[6])
    z2 = int(sys.argv[7])
    numberOfIerations = int(sys.argv[8])
    filename = sys.argv[9]

    if method == "mandelbrot_1":
        set = mb1.make_set(min1, min2, max1, max2, z1, z2, numberOfIerations)
    elif method == "mandelbrot_2":
        set = mb2.make_set(min1, min2, max1, max2, z1, z2, numberOfIerations)
    else:
        set = mb3.make_set(min1, min2, max1, max2, z1, z2, numberOfIerations)
    showImage(set, filename)
    sys.exit()
```

Also, I would recommend you to take a look at argparse: https://docs.python.org/3/library/argparse.html By using arparse you can implement the UI in a more elegant way, as this implementation is somewhat verbose. Argparse automates a lot of help-messages, which gives you a nice UI.

Assignment 4.6

The packaging is ok but the unit tests are not quite right. You are supposed to test your functions from the earlier parts of the assignment, but currently you have hard-coded in a function that you are testing, which means that your tests will never fail, even if you change your mandelbrot-functions. Instead, import your modules in test-mandelbrot.py, and place your functions inside the assert-statements. Otherwise you would have to manually change the code in test-mandelbrot.py every time you changed your other code.

Assignment 4.7

Nice, but missing support for the user to choose different color scales as arguments.

Assignment 4.8

In repeat-good(), line 33, there is a typo. Should be:

```
if counter < len(someString)-1:
```

Otherwise it works fine. I think the bad-version is actually quite good, as it is the same as the good-version but with shorter variable names. However the short variable names still makes sense. I think you could have made this even worse by for instance choosing more confusing variable names. For instance, all variable names could have been a, aa, aaa, aaaa etc. Or you could have swapped names on the variables, so that someString was called counter and vice versa. This makes the code really confusing and hard to read.

General feedback

Overall good job, however there were some few errors. I also think you can improve the help-messages in mandelbrot-py when called with –help, as I thought they were a little bit unclear. Also you could be a little more presise in your docstrings on specifying the data types of the arguments and return values, as these can sometimes be really important.

Good luck on the next assignment!