Software Engineering – Assignment 3

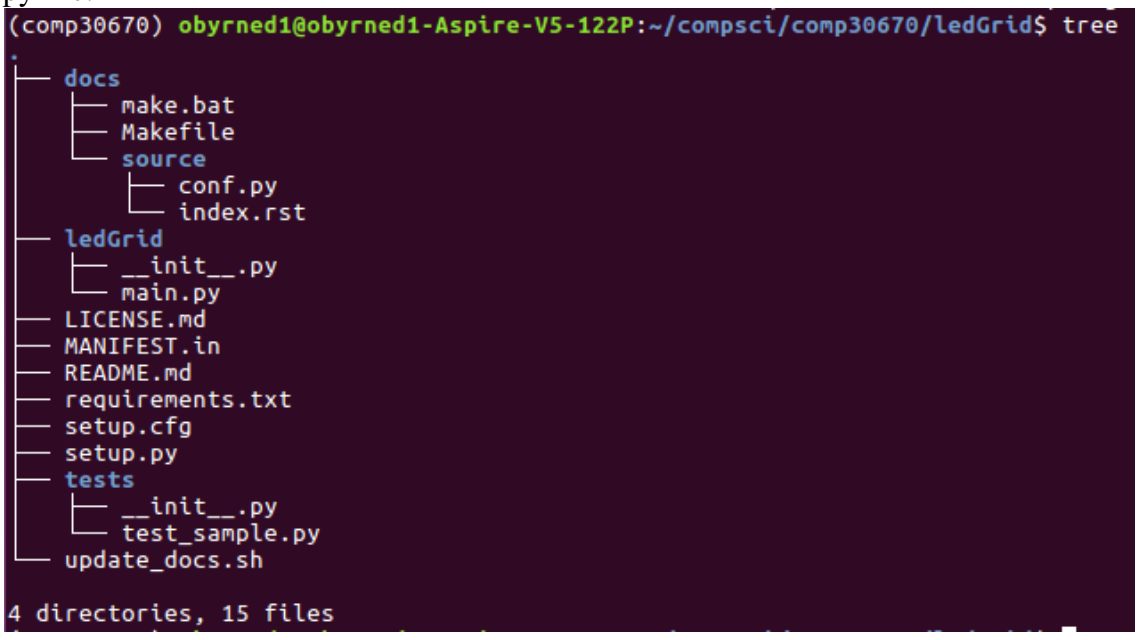
Daniel O'Byrne – 17205389

GitHub address: https://github.com/obyrned1/Assignment3

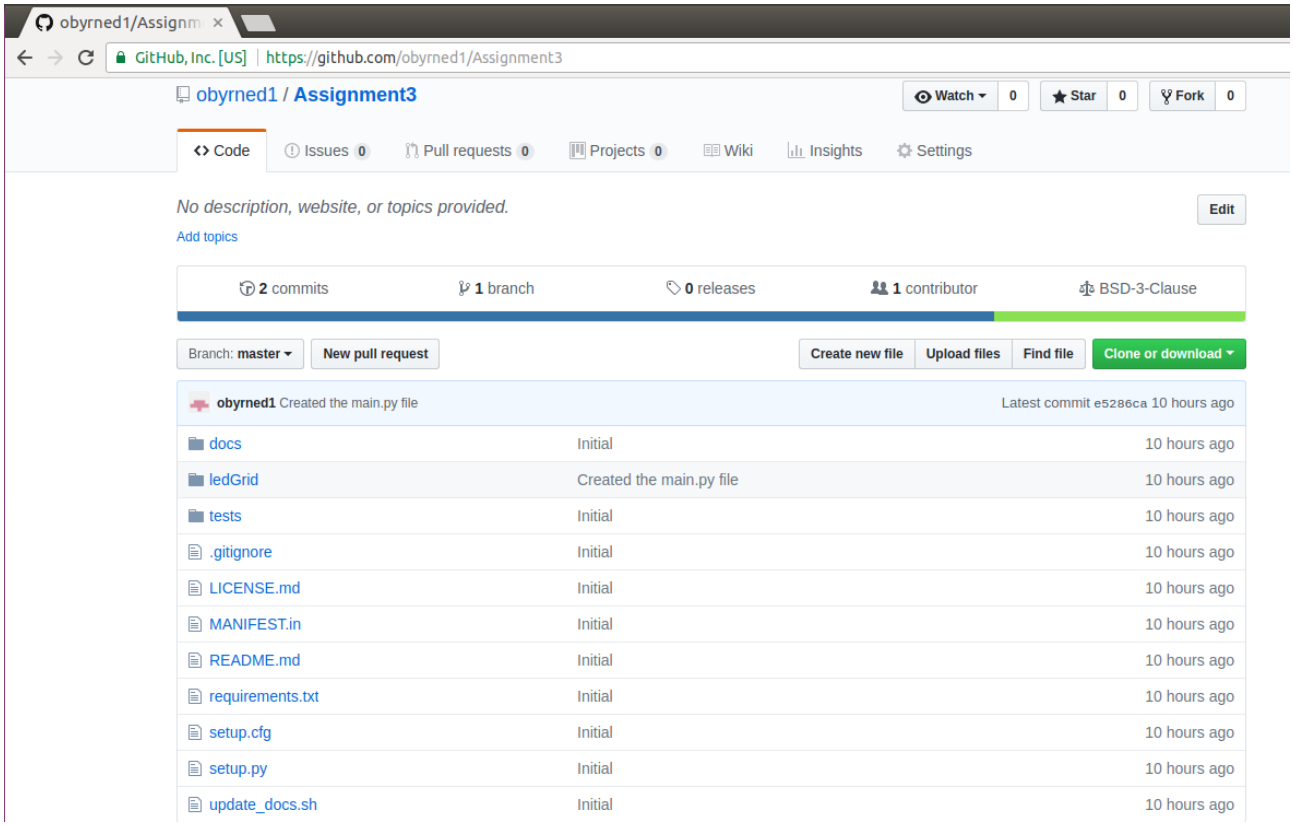
(a) How I will solve this problem:

- Firstly, I will have to create a function that will read in the contents of a local file or file retrieved from a network address.
- When I have read in this file, it should be in the form of "command" "x1" "y1" "x2" "y2", where x1,y1 is the stop point and x2, y2 is the stop point. These elements will have some sentence structure around them. I will have to figure out how to specifically get the contents in this form with only five elements, separating each group of five elements in order to carry out specific functions. I will use some form of regular expression technique to do this, creating one large array of commands. The regex technique must consider that we are only considered with commands of turn on/off or switch. Any other commands must be ignored.
- When I have the all the commands in one array, it will be simple to loop through it and access each element and sub-element.
- In a class , LightTester(), I will initialize the size of the grid and occupy each element with the value False. For example if the size of the grid is 1000, there will be 1000 arrays, with 1000 elements in each, all with the value False. The size of the grid is on the first line of the files, thus this will have to be extracted in the main method and used to create said arrays.
- My main method will then go through each element in the array until there are no more elements/commands. At each element, there will be a check to see what the command is i.e. turn on/off or switch. The relevant method should be called depending on the command, using the start and stop points within the element. There must be some check that all points don't have any coordinates that are below zero or above the size of the grid of lights.
- When all elements in the array have operated, a counting method will be called that will loop through the arrays initialized to have all False values, and will count every occurence of True in them. This indicates the number of lights that have been turn on from the given commands.

(b) I used cookiecutter as per assignment2, using the template provided at https://github.com/wdm0006/cookiecutter-pipproject. The following tree structure is the result, which satisfies the criteria of a package directory, a tests directory, a README.md file, and a setup.py file:



(c) I created a new GitHub repository called Assignment3. I then pushed the contents of my ledGrid folder created by cookiecutter, to it. Setting this as the origin master, I can open the files in Eclipse and successfully commit and push changes to GitHub.



(e) I found that using the Test Driven Development approach in this project, greatly assisted me in understanding the code I needed to write, and how to write it most efficiently. Continuously running tests, failing and rewriting them until they passed, provided a gradual learning process about how to best construct my code.

I created tests to:

- Check that the file/network address exists.
- Check that the contents of the file was successfully converted into a strcutured array.
- Check the coordinates from the array were converted to their own array, with both coordinates as integers.
- Check that coordinates outside the grid size were converted to the bounds of the grid.
- Check that the turn_on method changed an element from False to True, or kept True as True.
- Check that the turn_off method changed an element from True to False, or kept False as False.
- Check that the switch method changed an element from False to True, and from True to False.
- Check that the light_count method successfully counted all occurences of True in the arrays that make up the grid.

All tests eventually passed:

```
obyrned1@obyrned1-Aspire-V5-122P:~/compsci/comp30670/ledGrid$ py.test --verbose tests/*
============================= test session starts ==============================
platform linux -- Python 3.6.3, pytest-3.2.1, py-1.4.34, pluggy-0.4.0 -- /home/obyrned1/anaconda3/bin/python
cachedir: .cache
rootdir: /home/obyrned1/compsci/comp30670/ledGrid, inifile:
collected 8 items

tests/tests.py::test_file_exists PASSED
tests/tests.py::test_string_convert PASSED
tests/tests.py::test_coordinates PASSED
tests/tests.py::test_within_grid PASSED
tests/tests.py::test_turn_on PASSED
tests/tests.py::test_turn_off PASSED
tests/tests.py::test_switch PASSED
tests/tests.py::test_light_count PASSED

========================== 8 passed in 5.01 seconds ============================
```

However, there were many failed test along the ways and it took a lot of tweaking to get them all passing. For example my first step was to check if the file given is a legitemate network or local address [1].

```
def file_exists(filename):
    '''Checks if a file exists in the local address or a network address that is given'''
    if filename.startswith('http://'):
        returned_file = urllib.request.urlopen(filename)
    else:
        returned_file = open(filename, 'r')
    return returned_file


def test_file_exists():
    test_file = "http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt"
    assert file_exists(test_file) == None
```

I wrote this test to fail, once I set the != None, the test passes

```
=================================== FAILURES ===================================
_____________________________ test_file_exists _____________________________

    def test_file_exists():
        test_file = "http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt"
>       assert file_exists(test_file) == None
E       AssertionError: assert <http.client.HTTPResponse object at 0x7f04fd68a470> == None
E        +  where <http.client.HTTPResponse object at 0x7f04fd68a470> = file_exists('http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt')

tests/tests.py:9: AssertionError
========================== 1 failed in 10.82 seconds ===========================
```

Similarly, further on in the testing when I was cheking that the contents of the file were convreted to a structured string, I encountered issues. My test was to check the the array began with (*'switch'*, *'109'*, *'360'*, *'331'*, *'987'*). After some syntax adjustments, it passed.

```
=================================== FAILURES ===================================
_____________________________ test_string_convert _____________________________

    def test_string_convert():
        test_file = "http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt"
        output = string_convert(test_file)
        assert output[0] == "('switch', '109', '360', '331', '987')"
>       assert ('switch', '109', '360', '331', '987') == "('switch', '109', '360', '331', '987')"

tests/tests.py:18: AssertionError
========================== 1 failed, 1 passed in 1.59 seconds ==========================
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~/compsci/comp30670/ledGrid$ py.test --verbose tests/*
============================= test session starts ==============================
platform linux -- Python 3.6.4, pytest-3.4.1, py-1.5.2, pluggy-0.6.0 -- /home/obyrned1/anaconda3/envs/comp30670/bin/python
cachedir: .pytest_cache
rootdir: /home/obyrned1/compsci/comp30670/ledGrid, inifile:
collected 2 items

tests/tests.py::test_file_exists PASSED                                  [ 50%]
tests/tests.py::test_string_convert PASSED                               [100%]

========================== 2 passed in 0.71 seconds ============================
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~/compsci/comp30670/ledGrid$
```

[1] https://www.tutorialspoint.com/python/string_startswith.htm

The full development of my tests and my code should be evident through the commits and associated messages evident on my GitHub repository. I aso documented some of the steps I had taken in creating my code and tests:

- Now that I have the contents of the file in a string, I have to split each command into its own, so i can read what each one is asking. Firstly I have to only look at parts of the string that begin with:
  - "turn on"
  - "turn off"
  - "switch"
    - As given by the question, all ther commands should be ignored [2].
- I used one of the txt files to call the function created, string_convert. Running this yields the following output. Thus i can see that the return string has been divided up into parts(instructions), each part with 5 sub parts(details of instructions).

```
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~/compsci/comp30670/ledGrid/ledGrid$ python main.py
[('switch', '109', '360', '331', '987'), ('switch', '8', '315', '550', '764'), ('turn on', '867', '334', '914', '847'), ('switch', '618', '963', '737', '996'), ('
turn on', '981', '695', '992', '754'), ('switch', '578', '650', '722', '841'), ('turn off', '382', '796', '631', '915'), ('switch', '499', '730', '588', '897'), (
'switch', '606', '749', '879', '796'), ('turn on', '290', '458', '923', '797'), ('switch', '544', '149', '900', '340'), ('switch', '732', '9', '979', '725'), ('tu
rn off', '433', '258', '584', '672'), ('turn off', '330', '344', '758', '707'), ('switch', '952', '261', '960', '708'), ('turn on', '546', '467', '797', '918'), (
'switch', '876', '503', '883', '744'), ('switch', '600', '462', '951', '635'), ('turn off', '4', '469', '477', '575'), ('turn on', '325', '432', '690', '467'), ('
switch', '236', '298', '741', '931'), ('switch', '482', '134', '704', '614'), ('turn on', '388', '540', '731', '813'), ('turn off', '98', '365', '478', '800'), ('
switch', '322', '371', '379', '921'), ('turn off', '286', '245', '363', '719'), ('turn on', '619', '181', '736', '944'), ('turn off', '907', '350', '912', '553'),
 ('turn on', '688', '665', '958', '671'), ('switch', '195', '254', '923', '729'), ('turn off', '821', '494', '904', '896'), ('turn off', '986', '131', '987', '659
'), ('turn on', '422', '425', '948', '521'), ('turn on', '76', '846', '398', '938'), ('turn on', '558', '42', '931', '823'), ('switch', '607', '982', '673', '997'
```

- Now I have to write functions that can be called to split up this array appropriately

```
def coordinates(string):
    '''takes in a string which is parsed from the output of string_convert and assigns coordinates to start and stop values
    start = int(string[0])
    end = int(string[1])
    point = [start,end]
    return point
```

- This will be used to access each point, where the start of end value. These ints can then be used to light up specific areas
- I then set up the function to adjust points that fall outside of the grid size. Any point that has a value greater then the specified size, is set t and any value less then 0 is set to 0.
- I then created funcitons to turn on, off and switch lights. They are very similar, using for loops to loop through each array and each element in those arrays. True and False values are assigned to each depending on the command.
- Once I created the count method and it passed the test I had written, I had to create the main method that calls each relevant method logically.
- Having completed this main method, I needed to adjust the entry points to take arguments from the command line i.e. so that the setup.py installs a script which reads input from a file, specified as a command line argument (as per question)
- How to add an entry point successfully, to take command line arguments was adapted from:
  - https://chriswarrick.com/blog/2014/09/15/python-apps-the-right-way-entry_points-and-scripts/

[2] https://www.youtube.com/watch?v=sZyAn2TW7GY

- Having adjusted the entry points, I used pip install -e . . This command allows for updates of the installed module i.e ledGrid when the scripts are adjusted in Eclipse for example.
- Running all the txt files using the module gives the following results:

```
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  400410
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_a.txt
There are  5000  X  5000  lights in this grid
The number of lights turned on:  24999879
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_b.txt
There are  11000  X  11000  lights in this grid
The number of lights turned on:  29942250
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_c.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  477452
(comp30670) obyrned1@obyrned1-Aspire-V5-122P:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  349037
```

(f) I then launched my EC2 instance, and as per the previous assignment for systeminfo, I pip installed the ledGrid module from the GitHub address (after final commit and push):

```
(comp30670) ubuntu@ip-172-31-45-34:~$ pip install git+https://github.com/obyrned1/Assignment3
Collecting git+https://github.com/obyrned1/Assignment3
  Cloning https://github.com/obyrned1/Assignment3 to /tmp/pip-2xtub9g_-build
Collecting sphinx (from ledGrid==0.0.1)
  Using cached Sphinx-1.7.1-py2.py3-none-any.whl
Collecting sphinx_rtd_theme (from ledGrid==0.0.1)
  Using cached sphinx_rtd_theme-0.2.4-py2.py3-none-any.whl
Collecting nose (from ledGrid==0.0.1)
  Downloading nose-1.3.7-py3-none-any.whl (154kB)
    100% |████████████████████████████████| 163kB 3.3MB/s
Collecting coverage (from ledGrid==0.0.1)
  Downloading coverage-4.5.1-cp36-cp36m-manylinux1_x86_64.whl (202kB)
    100% |████████████████████████████████| 204kB 3.4MB/s
Collecting pypi-publisher (from ledGrid==0.0.1)
Collecting six>=1.5 (from sphinx->ledGrid==0.0.1)
  Using cached six-1.11.0-py2.py3-none-any.whl
Collecting Jinja2>=2.3 (from sphinx->ledGrid==0.0.1)
  Using cached Jinja2-2.10-py2.py3-none-any.whl
Collecting imagesize (from sphinx->ledGrid==0.0.1)
  Using cached imagesize-1.0.0-py2.py3-none-any.whl
Collecting snowballstemmer>=1.1 (from sphinx->ledGrid==0.0.1)
  Using cached snowballstemmer-1.2.1-py2.py3-none-any.whl
Collecting alabaster<0.8,>=0.7 (from sphinx->ledGrid==0.0.1)
  Using cached alabaster-0.7.10-py2.py3-none-any.whl
Collecting packaging (from sphinx->ledGrid==0.0.1)
  Using cached packaging-17.1-py2.py3-none-any.whl
Collecting babel!=2.0,>=1.3 (from sphinx->ledGrid==0.0.1)
  Using cached Babel-2.5.3-py2.py3-none-any.whl
Collecting sphinxcontrib-websupport (from sphinx->ledGrid==0.0.1)
```

ledGrid was successfully as a module on the instance, as seen using pip list:

```
(comp30670) ubuntu@ip-172-31-45-34:~$ pip list
DEPRECATION: The default format will switch to
under the [list] section) to disable this warni
alabaster (0.7.10)
Babel (2.5.3)
certifi (2018.1.18)
chardet (3.0.4)
coverage (4.5.1)
docutils (0.14)
gitdb (0.6.4)
GitPython (0.3.6)
idna (2.6)
imagesize (1.0.0)
Jinja2 (2.10)
ledGrid (0.0.1)
```

Running the same tests, I yielded the same results, except for b:

```
(comp30670) ubuntu@ip-172-31-45-34:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  400410
(comp30670) ubuntu@ip-172-31-45-34:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_a.txt
There are  5000  X  5000  lights in this grid
The number of lights turned on:  24999879
(comp30670) ubuntu@ip-172-31-45-34:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_b.txt
There are  11000  X  11000  lights in this grid
Killed
(comp30670) ubuntu@ip-172-31-45-34:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_c.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  477452
(comp30670) ubuntu@ip-172-31-45-34:~$ ledGrid --input http://claritytrec.ucd.ie/~alawlor/comp30670/input_assign3_d.txt
There are  1000  X  1000  lights in this grid
The number of lights turned on:  349037
```

The result from b of ‘Killed’ is likely due to the size of the computation that is required.

(g) On reflection of my test and subsequent code, I felt that my code could have been written in a more efficient manner with roughly 90 lines of code including doc strings. This could be a reason the process was Killed on my EC2 instance. However I feel that the code that I have written is simple and well structured, following a logical approach. I feel that more lines of easy to understand code is more appropriate, so others can understand the functionality of it.

Firstly within_grid and coordiantes methods could have merged together, returning an adjusted/non-adjusted coordinate. However I felt that if needs be, the user could print both the coordinate and within_grid output to highlight where points have been changed that were outside the grid. Therefore, I felt that keeping both methods separate was more user friendly and effective.

Similarly using a method to convert the string output of the file into an array, using regex, could have been streamlined. I could have instead had a line in the main method that converted the string and assigned it to a varibale. Again however, I felt that keeping the converting in it’s own method made it easier to see the steps of progression through the algorithm.