Introduction to ASCII Walkthrough

Frequently Asked Questions

1. Which program and task(s) you have implemented.

The program I implemented was the ASCII studio task. Where I implemented both task 1 and task 2.

2. How the program is started and used.

To Start the program type this command in your terminal when in the directory of the program.

```
>python3 ascii_art_studio.py
```

If successful, you will be greeted with these prompts:

```
>Run file and type help for commands.
>Welcome to ASCII Art Studio!
>AAS Command Input:
```

Where you as the user can input commands. For example if you are unsure what to input. Type **help** when prompted and hit enter.

```
>AAS Command Input: help
```

This would result in:

```
Help Commands:
1. load - load has two sub commands
       1a. load image - you can load an image file into the studio with this
command
       e.g. load image stadshuset.jpg as hus or load stadshuset.jpg
...cut for brevity
```

Or you can simply get going with commands like:

```
>load image slalom.jpg as slalom
>load image stadshuset.jpg as hus
>info
>render
```

3. Which libraries/modules are used and how these are downloaded and installed if they are not part of Python’s standard distribution.

Library Name	Reason	If Not Standard, Instructions
PIL	External library is used for image processing.	If you have pip3 installed, then to install type this command in your terminal: `>pip3 install PIL` or visit https://pillow.readthedocs.io/en/stable/installation.html for further instructions.
pickle	Used for saving the states of the objects to files, to make it easier to load sessions.	Standard
shlex	Used for saving and parsing commands entered by user	Standard
os	Used for error handling of the files	Standard

4. A description of how you structured your program (which files contain what, etc.).

Name	Description
ascii_art_studio.py	This is the main file with the program. Where the code to open up and render images is stored.
report_ascii_kyle_tenn_su.md and report_ascii_kyle_tenn_su.pdf	The text based information stored in md or PDF that you are reading.
All the JPG files	The jpg files are for you to use, but you should be able to put any jpg or png file in the same directory as the main program and be able to use them as well. NOTE **DO_NOT_DELETE_saturn.jpg** is used as a test image
requirements.txt	A file that you can run with python to install the PIL external library. `pip3 install -r requirements.txt`
test_ascii_art_studio.py	This is the main unit test file that is used to check methods and if they are running successfully. I was able to write tests for the methods that changed size, brightness, contrast, loading the image, and creating ascii text. However, one thing

	that could be improved was the test on how the console printed for the ascii render of an image could be tested with a predefined string. My understanding is that you need to use StringIO and sys, and then redirect the output. However, I could not understand how to make that work. I made the trade off by checking the Image class's attribute of the ascii text and comparing that to a predefined string.

Reflections on Code Quality

Code design

I used object orientation to create three modules of code, where the three main modules are

1. Main

This was the class that handled all user input and controlled how the Studio worked through an interactive prompt.

2. ASCII Image

This was the class that handled how the image was modified, manipulated, and stored. The PIL library was important in helping create this class.

3. ASCII Studio

This was the class that stored images, and created an object that was easy to save as a pickle file and load as a pickle file.

The way I handled the input from the user through a while True loop to allow for user commands, was a slightly user friendly way, but I would have preferred to make it all command line based. Then I could have used arg parse or extend the use of shlex to take multiple commands to simplify the main() function.

Which algorithms are used and why?

I used algorithmic thinking and came up with three key ideas. First, I would need characters that would represent a very light pixel to a spectrum of characters that would represent a very dark pixel. I would also need to get the brightness of each exact pixel. And lastly, I would need to, for each pixel across an image, use that pixel and match the brightness with a specific character. This algorithmic thinking led me to writing the convert_to_ascii() function. Where the exact steps and algorithm is written in code. This thinking is also reflected in the code design explained above. Where those three key ideas became three modules of code.

Which data structures are used and why?

I use the main python data structures such as dictionaries, lists, integers, strings, floats, tuples and so on in python. The reasons for using those are basic data storage and manipulation. However, the two data structures or custom classes that I create and instantiate, the studio, and images are more nuanced. I structured the studio with the important attribute of a dictionary which stores the images.