Code Inspection

Link to download the video:

Since the filesize limit on github is 100mb, we used mega:

https://mega.nz/#!P1xQSISL!DLruJwB_Z6Rru8cagBG6wfdkIYzyPv7gYLYPUNPLKSM

Click “Download through your browser”

Code Review Strategies

Our strategy:

Specifically for this code inspection, there was not a whole lot of core functionality or classes that were added that were able to be thoroughly reviewed, but new methods or features that we chose to inspect.
While we do not look out for style such as PEP8 or google’s programming style, we do consider things such as encapsulation, better ways to complete a task in a cleaner and more efficient manner as well as common programming paradigms and practices. For unit tests under review, we check if the test cover the majority of inputs the user can enter.

autoupdate_daemon.py

Reviewer: Eric Papagiannis
Code under review: autoupdate_daemon.py
Author of the code under review: Tony Wu
Date: Nov 23, 2016

Bugs:

- N/A

Poor code logic:

- Line 37: printing usage string and exiting as opposed to raising exception would be an easier solution.
- The argument to -i is in hours, which could be inconvenient to the user; interpreting the argument as minutes may provide greater flexibility.
- Verbose mode can only be flagged by manually switching a boolean, maybe could have method to toggle that

Poor coding style:

- Line 40: commandstr should not be hardcoded in the middle of the function; Keeping it at the top of the file could improve readability.

Missing documentation:

- Autoupdate_daemon.py could use a more verbose description of the purpose and principle of operation.
- Function main() is missing docstring.
- Inline comments are extremely sparse for the function main().

Unreadable code:

- N/A

Vulnerabilities in code:

- N/A

Poor testing:

- N/A

UnitConverter

Reviewer Name: Vasili Skurydzin, Jerry Cheng
Code under review: UnitConverter class in CSV_data_parser.py (added after last code review)
Date of Review: Nov 23, 2016
Author of the code: Albion Fung

Bugs:

- N/A

Poor code logic:

- ConvertToOpen method - contains nested functions. (state the reason in docstring)
- Nested Functions inside convertToOpen, such as convertDate and convertEURA, contain many hard-coded values that makes the functions non-modular.
- May be better to put UnitConverter class in its own file

Poor coding style:

- If __name__ == "__main__" block contains a code block that is not used.

Missing documentation:

- ConvertToOpen method - docstring is not very informative (description is too vague). Documentation does not specify what is a valid input for the parameters ‘field’, ‘data’ and ‘source’
- ConvertToOpen method - nested functions are missing internal / external documentation.
- Class documentation could be more informative, including the purpose of the class and what it cooperates with.

Unreadable code:

- N/A

Vulnerabilities in code:

- N/A

Poor testing:

- Refer to UnitConverterTest.py

UnitConverterTest.py

Reviewer Name: Jerry Cheng
Code under review: UnitConverterTest.py
Date of Review: November 23, 2016
Author of code: Albion Fung

Bugs:

- N/A

Poor Code Logic:

- N/A

Poor Coding Style:

- N/A

Missing Documentation:

- N/A

Unreadable code:

- N/A

Missing Documentation:

- No documentation for any of the unittests explaining what each of the tests do
- Testcase names in UnitConverterTest do not adequately explain what the test case is testing

Poor Testing:

- No tests to check data input errors in UnitConverter (inputting object types other than Strings)
- No tests to check if the ‘field’ input parameter for convertToOpen is valid
- No tests to check if the ‘source’ input parameter for convertToOpen is valid
- No tests to check different possible date formats the user can enter for convertDate
- No tests to check for entering negative numbers that the user can enter for convertEURA and convertEUDEC
- No tests for convertNASARA and convertNASADEC to check for potential inputs in formats other than ‘**h**m**s’, where n is a single digit number.

buildStar method in CSV_data_parser.py

Reviewer Name: Eric Papagiannis
Code under review: buildStar method in CSV_data_parser.py (added after last code review)
Date of Review: Nov 23, 2016
Author of code: Albion Fung

Bugs:

- N/A

Poor Code Logic:

- Reuse of code from buildPlanet can be refactored into a method they both call to parse data fields and error fields

Poor Coding Style:

- Try excepts have pass on the except statement

Missing Documentation:

- Minimal inline comments

Unreadable Code:

- Some nested statements and try excepts could be harder to read at a glance

Vulnerabilities in Code:

- N/A

Poor Testing:

- Sufficient manual and automated testing

show_range, accept_range, deny_range, postpone_range in Driver_Commands.py

Reviewer Name: Jerry
Code under review: show_range, deny_range, postpone_range, accept_range (methods) in driver_commands.py
Date of Review: 11/28/16
Author of code: Eric

Bugs:

- N/A

Poor Code Logic:

- A lot of code is being reused across all the range functions for input validation. It could be better having a helper function to handle that aspect.

Poor Coding Style:

- Inconsistent variable naming in deny_range: both camelcase and snake_case used.

Missing Documentation:

- No Description on the functionality of deny_range
- No Description on the functionality of postpone_range
- The docString in accept_range has conflicting information regarding valid user input, the description states (int, int, int) is valid, but the type contract asks only for (str, str, int)

Unreadable Code:

- Postpone_range has a pass statement in Docstrings

Vulnerabilities in Code:

- N/A

Poor Testing:

- N/A

Deny_number, postpone_all, Unpack_changes in Driver_Commands.py

Reviewer Name: Tony Wu
Code under review: deny_number, postpone_all, unpack_changes (methods)
Date of Review: Nov 23, 2016
Author of code: Vasili

Bugs:

- N/A

Poor Code Logic:

- In postpone_all, logic flow of code unclear, overly complex
- In postpone_all, reasoning unclear why empty list passed into method
- In unpack_changes, logic flow of code unclear, overly complex
- In deny_number, config was set to both the string “blacklist” and a variable named blacklist

Poor Coding Style:

- For all these methods, driver commands shouldn’t have to worry about printing

Missing Documentation:

- For all these methods, as code is really specific, a more verbose explanation as to how it works would be preferred in their docstrings

Unreadable Code:

- In deny_number, there is spelling error found in in-line comment

Vulnerabilities in Code:

- N/A

Poor Testing:

- N/A