Code Inspection

Name of the Reviewer: Ian Ferguson
Code Under Review: github_repo.py
Date of the Review: November 9, 2016
Name of the Author of the Code under review: Ahsan Zia

- Bugs
  - Checking for an existing directory bug, that is, if the repo that is to be cloned exists then the GitRepo function fails to initialize and throws git clone error. The problem occurs once the GitRepo call is made inside oec_git_test.py at line 45. The simple fix to the problem is to check if the cloned repo already exists delete the existing repo before initializing the call to GitRepo.
  - Pull requests are deleted when the branch is deleted from the forked repo. To produce the bug, we run oec_git_test.py for a sample test.csv and wait for the program to finish, once completed their should be several pull request to a dummy GitHub account but none are shown. That is the bug at line 191 of github_repo.py removes all the branches in the forked GitHub which allows the pull requests to be deleted because the way API handles pull requests. In order to fix the bug, the program should first detect if the pull request is still active if so do not delete the branch but just update/add the files (if needed) and since a pull request already established it should not send another. However, if the pull request is rejected, we delete the local branch on the local forked repo and make a new branch with updated values and send a new pull request.

- Poor code logic
  - No redundant or duplicated code found.
  - No dead code located listed inside github_repo.py
  - Code is straightforward, easy to read and understand.
  - Clean up function inside github_repo.py may need modifications based on the 2nd bug.

- Poor coding style
  - Document is minified under pep8.
  - The checkout function can be made smaller.

- Missing documentation
  - The code is fully documented with python's docstring and commented throughout the class for easy to understand hard code and functions.

- Unreadable code:
  - Code is clean and concise throughout the class

- Vulnerabilities in code
  - Directory waiting to be cloned is not checked before cloning another copy which throws a clone error.
  - Pull requests are not established correctly as they are deleted when the branches are deleted from the local fork.

Code Inspection

  - Curl is used in pull request but on line 163 is not valid way to check for JSon output request.

- Poor testing
  - The test cases are given based on the specification and requirements provided by the GitHub API documentation.

Code Inspection

Name of the Reviewer: Ahsan Zia
Code Under Review: system_xml.py
Date of the Review: November 9, 2016
Name of the Author of the Code under review: Ian Ferguson

- Bugs
  - This is more of a conversion bug, but the conversion function currently has no guards for improperly formatted strings inputted from the csv files. As a result, the program can crash if add_element or update_element method is called and the string passed to the conversion function from the csv contains a typo or incorrect format. The string either needs to be sanitized or error handling added (i.e. skip adding incorrectly formatted data if not convertible/sanitizable).
  - When the xml creation function new_system_xml in xml_functions.py is run on a SystemXML object with a binary xml tag, it does not update xml subelements of the binary tag, instead creating new tags outside of the binary tags as though it is not a binary system.
  - When an existing XML document is updated all new code is properly formatted however the existing xml text seems to be outputted with correct indentation but two spaces between each line. This bug appears on intermittent on documents (not all xml documents have the spacing issue) but consistently within the same document on subsequent runs (always affects the same documents).

- Poor code logic
  - Instead of having individual methods to add stars, planets, etc. why not have one add subelement method that can be given a tag value as each method is nearly identical except for the tag values. Also seems odd to have those as XML file methods instead of SystemXML and Star methods respectively, if both methods are kept they should be moved to the child classes.
  - The pull_msg and commit_msg methods should adjust their output if the source and or reference list is empty.
  - Is it confirmed that each System XML can only have one last update date, or is it possible that different exoplanets in the system might have been updated on different dates. If not then we may need to add a method to return the last update of each planet.

- Poor coding style
  - Style follows Pep8 and there is a good use variable/method names.
  - As mentioned under code logic, the code to add stars/exoplanets is very similar and could be combined or made submethods of the child classes.

- Missing documentation
  - All classes have type contracts and documentation.

- Unreadable code
  - All code is readable, no dead code.

Code Inspection

- Vulnerabilities in code
  - As mentioned under bugs, data from csv is not checked before being passed to conversion function. This should be done at one of three points, when csv data is read in, when it is passed into the add_element or update_element methods, or when received by the conversion function to_oec_value in the conversions.py file.
  - Currently invalid parameters are not guarded against, however as the only user input these functions receive is from the csv files, and mapping tuples (issues with csv already mentioned) and are only intended to be called by other program functions within the code with well defined calls/values, using a contract based design (i.e. undefined behaviour for invalid inputs) does not seem unreasonable for everything except for csv values, and mapping values. Mapping values will throw an error (unhandled: ValueError if tag or attribute tuple contains too few or too many values, TypeError if the tuple does not have a second element that is not an iterable, and IndexError if tuple does not contain at least 2 elements) if incorrect, however it would be better to handle potential errors and log message to update with correct mapping.

- Poor testing
  - Current testing (white box) is comprehensive for valid input for init/methods, however the unit tests currently contain no defensive testing for invalid input from csv or mapping tuples which could both potentially be not properly formatted. All other input is preformatted by calling code and should be caught before passed to the function and is safe to assume it is correctly formatted; therefore, existing testing should be sufficient for these inputs.

Code Inspection

Name of the Reviewer: Jubin Patel
Code Under Review: mappings.py
Date of the Review: November 9, 2016
Name of the Author of the Code under review: Lucy Xing

- Bugs
    - No bugs detected

- Poor code logic
    - Data structure to store mappings is overly complicated
    - Too many global variables, code becomes difficult to follow

- Poor coding style
    - Over complicated variable names
    - Too much repetition and redundancy in global variable names

- Missing documentation
    - Parts of code missing documentation
    - Over complicated variable names
    - Too much repetition and redundancy in global variable names

- Unreadable code:
    - Very hard to follow code

- Vulnerabilities in code
    - How to confirm correct mapping?
    - Very hard to maintain code
    - Change of column names by the monitored catalogue will result in the mapping of the OEC Synchronizer to lose effect

- Poor testing
    - Test not possible for this part of code

Code Inspection

Name of the Reviewer: Lucy Xing
Code Under Review: csv_downloader.py
Date of the Review: November 9, 2016
Name of the Author of the Code under review: Marhababanu Chariwala

- Bugs
  - No major bugs.
  - exoplanet data are downloaded correctly in csv format and stored in appropriate files.

- Poor code logic
  - No redundant or duplicate code in the module.
  - Logic is working because the both the monitored sites have ability to download data in csv format. Logic might fall apart if additional site that does not csv format are required to monitor.

- Poor coding style
  - does not have repeated code
  - Follows pep8 conventions
  - Variable and method are named properly. No confusion in variable
  - Code is formatted and readable

- Missing documentation
  - Docstring for functions are provided
  - Comment are inserted where necessary
  - Description of the function can be made more comprehensive by adding how exceptions handled
  - Example are missing in the docstring

- Unreadable code:
  - Code is readable
  - Contain no dead code

- Vulnerabilities in code
  - Invalid inputs like Boolean or integer are not handled
  - Error and exception are handled efficiently

- Poor testing
  - Unit test do not include Timeout scenario
  - Current testing is based on black box testing but some categories of the test case are not covered because of limitation in setting up environment required for that particular cases

Code Inspection

Name of the Reviewer: Marhababanu Chariwala
Code Under Review: conversions.py
Date of the Review: November 9, 2016
Name of the Author of the Code under review: Jubin Patel

- Bugs
  - Functions do not handle invalid inputs (i.e. strings)

- Poor code logic
  - Distance function is never used since the "keplermag" constant is mapped to None, but could be used when additional catalogues are added in

- Poor coding style
  - Exceptions should be implemented so invalid inputs are handled in case they somehow enter the csv tables under the columns where you don't expect those sort of inputs
  - Pep8 style is implemented and variables are named accordingly
  - A special function for NASA data (i.e. nasa_to_oec_declination) is probably not needed and could be combined with the other declination function

- Missing documentation
  - Functions are documented sufficiently to let the reader know exactly what the function does and its output

- Unreadable code
  - Code is readable
  - No dead code present

- Vulnerabilities in code
  - Input data is not checked
  - Invalid parameters not handled as mentioned above

- Poor testing

Current unittest don't address invalid inputs, but covers every other numeric input possible and ensures the correct output is displayed

Code Inspection Meeting Video: https://www.youtube.com/watch?v=tmOooxXFnhM