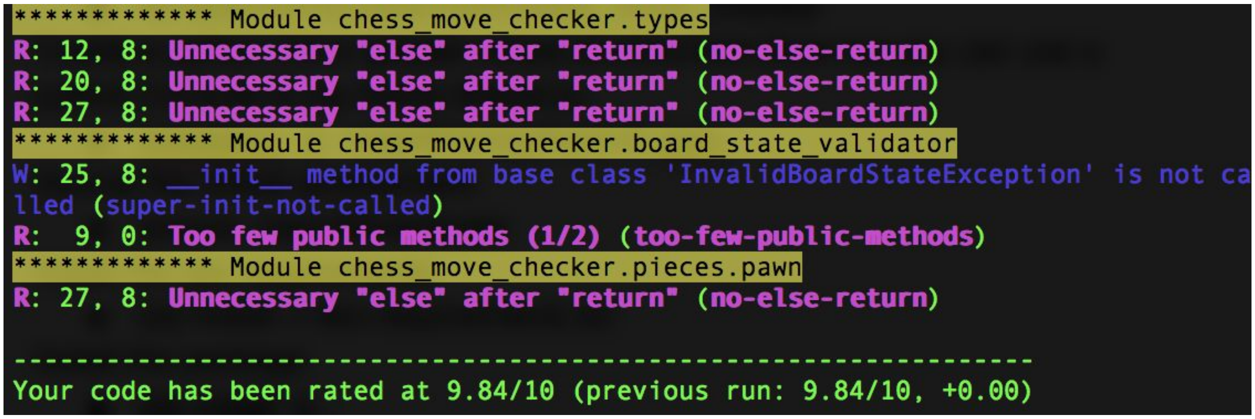
Cameron Graybill
CS 576
08/19/2018

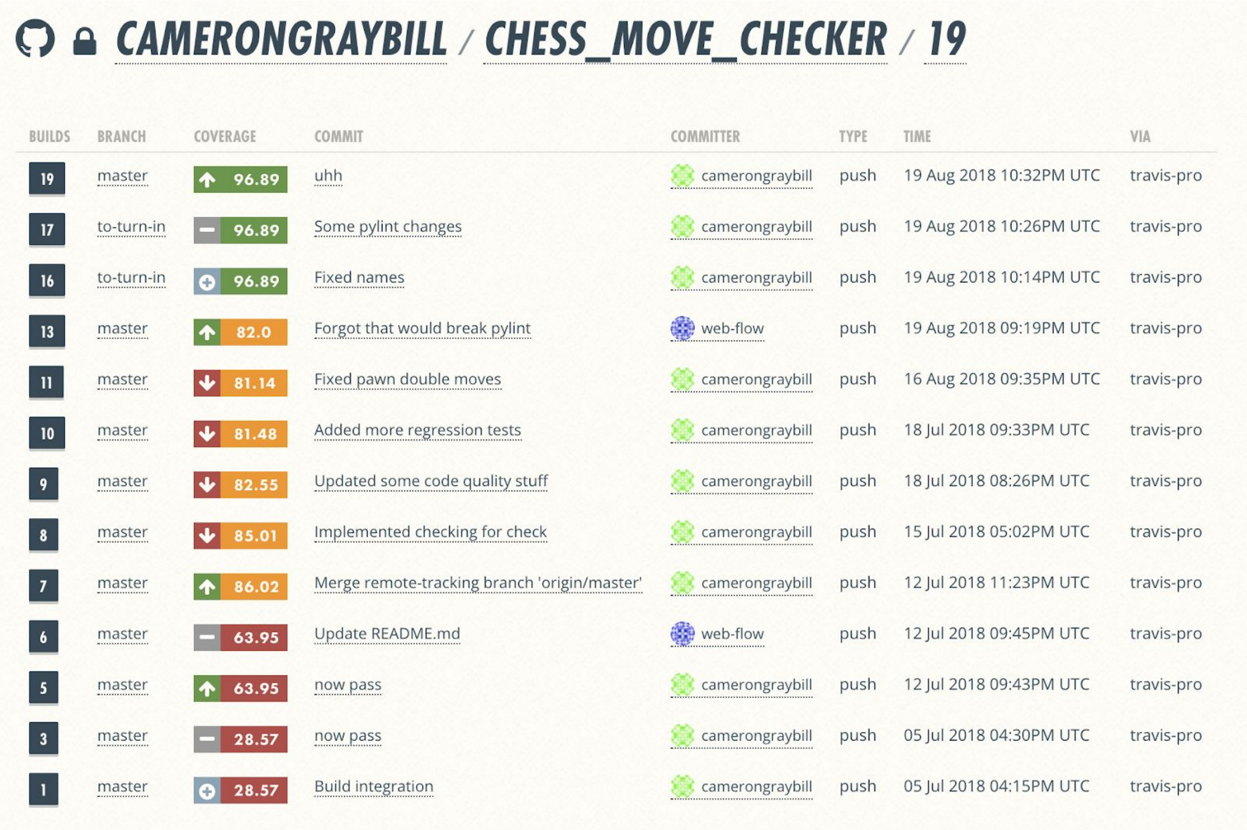
- General Notes:
    - I did not implement castling because we do not know if the king or rook has moved yet this game. Castling is only valid if neither piece has moved, and we have no sense of what actions took place in turns before this one.
    - I also did not implement en passant for the same reason. We do not know previous moves so we cannot make this move.
- Source Code is included in “chess_move_checker” directory. To build and run the source on linux:
    - Change to the source directory
        - cd /path/to/my/folder/root/chess_move_checker
    - Create a python 3.6 or greater virtual environment (from tux you can use a python 3.6 build in my home directory)
        - virtualenv -p ~csg58/bin/python3.6 venv
    - Source the virtual environment
        - source ./venv/bin/activate
    - Install the dev requirements
        - pip install -r dev-requirements.txt
    - Install the package
        - pip install -e .
    - Run the program! (validate_move should be in your path now, example input files are in tests/regression_tests/inputs)
        - validate_move OR validate_move -i input_file
    - If validate_move is not in your path, the path to the executable is ./venv/bin/validate_move
- For test cases, I had regression and unit tests. I used Unit tests to fill in any areas that my regression tests did not cover. I mainly had unit tests because the assignment does not specify what kind, and it is fairly easy to come up with test cases that cover every piece’s possible movement pattern.
    - Regression tests are stored in chess_move_checker/tests/regression_tests
    - Unit tests are stored in chess_move_checker/tests/unit_tests
    - Run all tests by running (assuming you are in the directory with setup.py in it):
        - python setup.py test
    - To run tests with code coverage, you must run:
        - nosetests --with-coverage --cover-package=chess_move_checker
- Here is some output from the static analysis tool pylint that I ran against my code:



```
************* Module chess_move_checker.types
R: 12, 8: Unnecessary "else" after "return" (no-else-return)
R: 20, 8: Unnecessary "else" after "return" (no-else-return)
R: 27, 8: Unnecessary "else" after "return" (no-else-return)
************* Module chess_move_checker.board_state_validator
W: 25, 8: __init__ method from base class 'InvalidBoardStateException' is not ca
lled (super-init-not-called)
R:  9, 0: Too few public methods (1/2) (too-few-public-methods)
************* Module chess_move_checker.pieces.pawn
R: 27, 8: Unnecessary "else" after "return" (no-else-return)

-------------------------------------------------------------------
Your code has been rated at 9.84/10 (previous run: 9.84/10, +0.00)
```

- From that output, I went back and updated the code in a few places to remove unnecessary “else” after returns, particularly in types.py. I also used the built-in static analysis tool in my IDE, Pycharm, as I developed. This tool helped me avoid syntax errors and would give type suggestions do some static analysis as I wrote code.
- My final code coverage was about 97%. As I developed, I used the tools “Github”, “Travis CI” and “Coveralls” to build my code and run my tests as I developed it. Every time I would push a commit to my Github repository, Travis CI would run all of my tests and upload the results to Coveralls. Here is the history of code coverage as I worked on the project and committed to my repository.



CAMERONGRAYBILL / CHESS_MOVE_CHECKER / 19

BUILDS	BRANCH	COVERAGE	COMMIT	COMMITTER	TYPE	TIME	VIA
19	master	↑ 96.89	uhh	camerongraybill	push	19 Aug 2018 10:32PM UTC	travis-pro
17	to-turn-in	− 96.89	Some pylint changes	camerongraybill	push	19 Aug 2018 10:26PM UTC	travis-pro
16	to-turn-in	⊕ 96.89	Fixed names	camerongraybill	push	19 Aug 2018 10:14PM UTC	travis-pro
13	master	↑ 82.0	Forgot that would break pylint	web-flow	push	19 Aug 2018 09:19PM UTC	travis-pro
11	master	↓ 81.14	Fixed pawn double moves	camerongraybill	push	16 Aug 2018 09:35PM UTC	travis-pro
10	master	↓ 81.48	Added more regression tests	camerongraybill	push	18 Jul 2018 09:33PM UTC	travis-pro
9	master	↓ 82.55	Updated some code quality stuff	camerongraybill	push	18 Jul 2018 08:26PM UTC	travis-pro
8	master	↓ 85.01	Implemented checking for check	camerongraybill	push	15 Jul 2018 05:02PM UTC	travis-pro
7	master	↑ 86.02	Merge remote-tracking branch 'origin/master'	camerongraybill	push	12 Jul 2018 11:23PM UTC	travis-pro
6	master	− 63.95	Update README.md	web-flow	push	12 Jul 2018 09:45PM UTC	travis-pro
5	master	↑ 63.95	now pass	camerongraybill	push	12 Jul 2018 09:43PM UTC	travis-pro
3	master	− 28.57	now pass	camerongraybill	push	05 Jul 2018 04:30PM UTC	travis-pro
1	master	⊕ 28.57	Build integration	camerongraybill	push	05 Jul 2018 04:15PM UTC	travis-pro