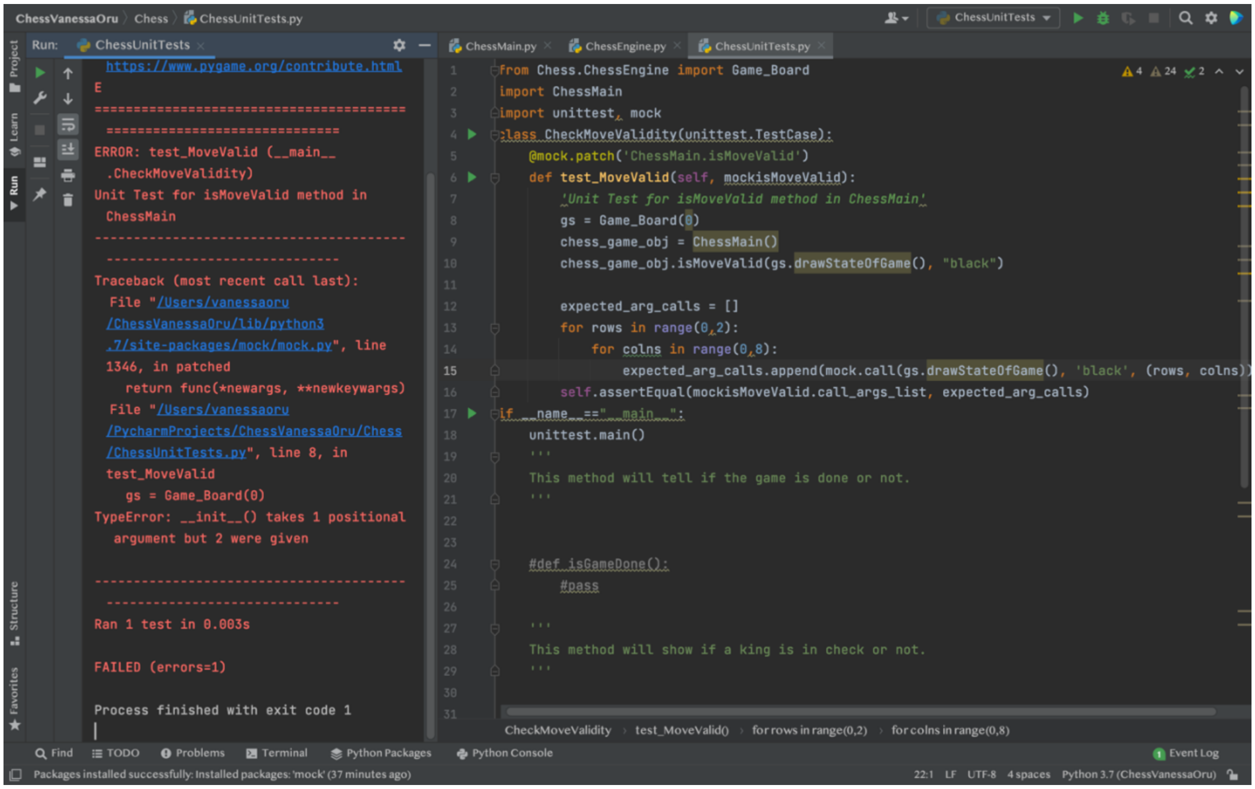
I did a unit test for the method that checks if a move is valid. I had to install the unittest.mock python package. This is a mockup package that runs the tests as mock objects of your original objects. The bugs found in my Chess game project can be seen in the picture below. I am unable to fix them now because I have to finish developing those methods in the Chess Main file. Also, my board still will not show but I am working on fixing it.



```
  https://www.pygame.org/contribute.html
E
========================================
  ==============================
ERROR: test_MoveValid (__main__
  .CheckMoveValidity)
Unit Test for isMoveValid method in
  ChessMain
----------------------------------------
  ------------------------------
Traceback (most recent call last):
   File "/Users/vanessaoru
  /ChessVanessaOru/lib/python3
  .7/site-packages/mock/mock.py", line
  1346, in patched
     return func(*newargs, **newkeywargs)
   File "/Users/vanessaoru
  /PycharmProjects/ChessVanessaOru/Chess
  /ChessUnitTests.py", line 8, in
  test_MoveValid
     gs = Game_Board(0)
TypeError: __init__() takes 1 positional
   argument but 2 were given


----------------------------------------
  ------------------------------
Ran 1 test in 0.003s

FAILED (errors=1)

Process finished with exit code 1
```

```python
from Chess.ChessEngine import Game_Board
import ChessMain
import unittest, mock
class CheckMoveValidity(unittest.TestCase):
    @mock.patch('ChessMain.isMoveValid')
    def test_MoveValid(self, mockisMoveValid):
        'Unit Test for isMoveValid method in ChessMain'
        gs = Game_Board(0)
        chess_game_obj = ChessMain()
        chess_game_obj.isMoveValid(gs.drawStateOfGame(), "black")

        expected_arg_calls = []
        for rows in range(0,2):
            for colns in range(0,8):
                expected_arg_calls.append(mock.call(gs.drawStateOfGame(), 'black', (rows, colns))
        self.assertEqual(mockisMoveValid.call_args_list, expected_arg_calls)
if __name__=="__main__":
    unittest.main()
    '''
    This method will tell if the game is done or not.
    '''


    #def isGameDone():
        #pass

    '''
    This method will show if a king is in check or not.
    '''
```

Packages installed successfully: Installed packages: 'mock' (37 minutes ago)

UPDATE: BOARD IS SHOWING AND CODE RUNS JUST FINE.