```
'''
Created on Feb 22, 2015

@author: mroch
'''

import time
import datetime
import human  # human - human player, prompts for input
import boardlibrary  # might be useful for debugging
import checkerboard


# tonto - Professor Roch's not too smart strategy
# You are not given source code to this, but compiled .pyc
 files
# are available for Python 3.5 and 3.6 (fails otherwise).
# This will let you test some of your game logic without
having to worry
# about whether or not your AI is working and let you pit
your player
# against another computer player.
# initializing tonto
import imp
import sys
major = sys.version_info[0]
minor = sys.version_info[1]
modpath = "__pycache__/tonto.cpython-{}{}.pyc".format(
major, minor)
tonto = imp.load_compiled("tonto", modpath)


def elapsed(earlier, later):
    """elapsed - Convert elapsed time.time objects to
duration string

    Useful for tracking move and game time.  Example
pseudocode:

    gamestart = time.time()

    while game not over:
        movestart = time.time()
        ...  logic ...
        current = time.time()
        print("Move time: {} Game time: {}".format(
            elapsed(movestart, current), elapsed(gamestart
, current))

```

```python

    """
    return time.strftime('%H:%M:%S', time.gmtime(later - earlier))


def Game(red=human.Strategy, black=tonto.Strategy,
         maxplies=5, init=None, verbose=True, firstmove=0):
    """Game(red, black, maxplies, init, verbose, turn)
    Start a game of checkers
    red,black - Strategy classes (not instances)  # Not invoked
    maxplies - # of turns to explore (default 10)
    init - Start with given board (default None uses a brand new game)
    verbose - Show messages (default True)
    firstmove - Player N starts 0 (red) or 1 (black). Default 0.
    """

    # Example of creating a game
    #ai_player = ai.strategy('r', checkerboard.CheckerBoard, maxplies)  # todo
    #  create a checkerboard with this particular state
    red_player = red('r', checkerboard.CheckerBoard, maxplies)
    black_player = black('b', checkerboard.CheckerBoard, maxplies)

    board = checkerboard.CheckerBoard()

    board.turncount = 0
    while board.is_terminal()[0] is False:
        if board.turn_count % 2 == 0:
            [board, red_action] = red_player.play(board)
            print("Red player moved {}".format(red_action))
            print(board)
        if board.turn_count % 2 != 0:
            [board, black_action] = black_player.play(board)
            print("Black player moved {}".format(black_action))
            print(board)
        board.turn_count += 1
    print("Winner chicken dinner is: " + str(board.is_terminal()[1]))

```

```python

if __name__ == "__main__":
    Game()
```