```
'''
Created on Mar 1, 2015

@author: mroch
'''

import checkerboard

class Strategy:
    """Abstract strategy for playing a two player game.
    Abstract class from which specific strategies should be derived
    """

    def __init__(self, player, game, maxplies):
        """Initialize a strategy
        player is the player represented by this strategy
        game is a class or instance that supports the class or instance method
            game.other_player(player) which finds the name
                of the other player
        maxplies is the maximum number of plies before a cutoff is applied
        """

        # Useful for initializing any constant values or structures
        # used to evaluate the utility of a board
        self.maxplayer = player
        self.minplayer = game.other_player(player)
        self.maxplies = maxplies

    def utility(self, board):
        "Return the utility of the specified board"
        raise NotImplementedError("Subclass must implement")

    def play(self, board):
        """play - Make a move
        Given a board, return (newboard, action) where newboard is
        the result of having applied action to board and action is
        determined via a game tree search (e.g. minimax with alpha-beta
        pruning).
        """

        raise NotImplementedError("Subclass must implement")

```