

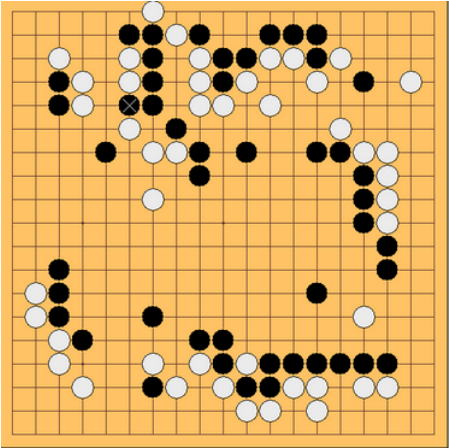
Implementing a Go Class Based on Gnugo

Yuzhe Zhou

May 30, 2014

What is Go? Go is a game played on a 19*19 grid board, consists of the players alternating placing black and white stones down on the intersections of the board; If two stones of the same color are adjacent, they are "connected". If any connected group is not adjacent to any open intersections, they are captured off the board. At the end of the game (when both players pass), you add up the total number of open intersections your stones surround and the number of stones from your opponent that you've captured.

A typical game board in progress



What is Gnugo?

```

White (O) has captured 12 pieces
Black (X) has captured 3 pieces

      A B C D E F G H J K L M N O P Q R S T
19 . . . . . . . . . . . . . . . . . 19
18(O)O . . . . . . . . . . . . O X O X . . 18
17 . . O O . . . . . . . . . . O O X X . . 17
16 . . . O . . . . . . + O . . O X + X . . 16
15 . O O O O . . . . . . . . X . . X . . 15
14 . . . O . . . . . . . . . . X X . . . 14
13 . O O . . . . . . . . . . . . . . . 13
12 O X . . . . . . . . . . . . . . . 12
11 . . . . . . . . . . . . . . . . . 11
10 . . . + . . . O . X . X . . . + . . . 10
9 . . . . . . . . . . . . . . . . . 9
8 . . . . . . . . . . . . . . . . . 8
7 . . . . . . . . . . . . . . . . . 7
6 . . . . . . . . . . . . . . . . . 6
5 . . . . . . . . . . . . . . . . . 5
4 . . . + . . . . . . + . . . . O . . . 4
3 . . O . . . . . . . . . . . . . . . 3
2 . . . . . . . . . . . . . . . . . 2
1 . . . . . . . . . . . . . . . . . 1
      A B C D E F G H J K L M N O P Q R S T

black(51): █
  
```

Last move: White A18

If
you type in Gnugo at the command line terminal, you can use
it right now.

My project aims at getting the output of a gnugo and turns it into a sage class. It now has 3 basic methods: printboard, plot board and findstone, which gives information about the board.