

Play(0), ... player(7) (is_playing)

Small blind: initialized to first person on player list
for 0 to 8

Find next small blind & update after each game)

for Small ... 8: check(is_playing)

for 0 to 9: 11:

Game Starts:

Find UTG

game-1st: (Already implemented)
Small blind = 4
[4, 6, 7, 0, 2]

keep track of 4

Pre flop:

4 check [6, 7, 0, 2, 4]

6 check [7, 0, 2, 4, 6]

7 bet [0, 2, 4, 6, 7] keep track of 7

0 call [2, 4, 6, 7, 0]

2 call [4, 6, 7, 0, 2]

4 raise [6, 7, 0, 2, 4] keep track of 4

6 fold [7, 0, 2, 4]

7 fold [0, 2, 4]

0 call [2, 4, 0]

2 fold [4, 0]

array[0] = 4
Done

Flop: create new game first

for smallB ... 8
is-playing, is-inhand
for 0 ... smallB

[4, 0] Track 4

4 check [0, 4]

0 raise [4, 0] Track 0

4 call [0, 4] DONE

2 bet \$2

3 raise \$4

4 raise \$8

call \$8

raise 16

call 16

Fold

[2, 3, 4]

2 bets [3, 4, 2] Track 2
3 raise [4, 2, 3] Track 3
4 raise [2, 3, 4] Track 4
2 call [3, 4, 2] Track 4
3 raise [4, 2, 3] Track 3
4 call [2, 3, 4] Track 3
2 fold (3, 4) DONE

Stack-\$2 stacks

2 : bet-1[\$2, \$8, \$16]

3 : bet-1[\$4, \$16]

4 : bet-1[\$8, \$16]

keep track max-bet

Call Fold check, Bet

curr bet: \$2

Stack - 2

curr bet: \$4

Pot: \$2

100	-2	-6	-8	
Steve:	[\$2 (bet)]	<u>\$8</u> (call)	<u>\$16</u> (call)]
100				
Pat:	[\$4 (bet)]	\$16 (bet)]
100				
Brandon:	[\$8 (bet)]	\$16 (call)]

After betting:

1. update bet-list
2. update stack/pot-size
 - if len(bet-list) == 1
 - stack = bet-list[-1] / potsize + bet-list[-1]
 - else:
 - stack = (bet-list[-1] - bet-list[-2])
 - potsize + (bet-list[-1] - bet-list[-2])