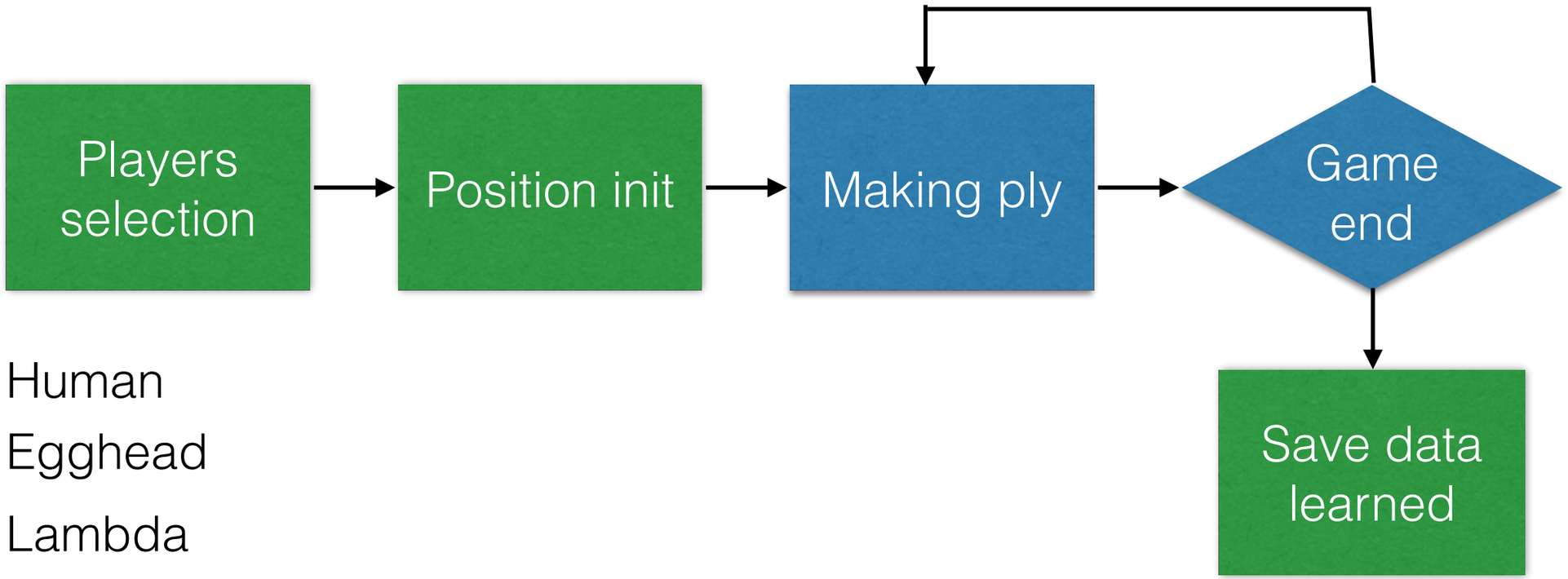
Gomoku

revision
September 2015

Main loop



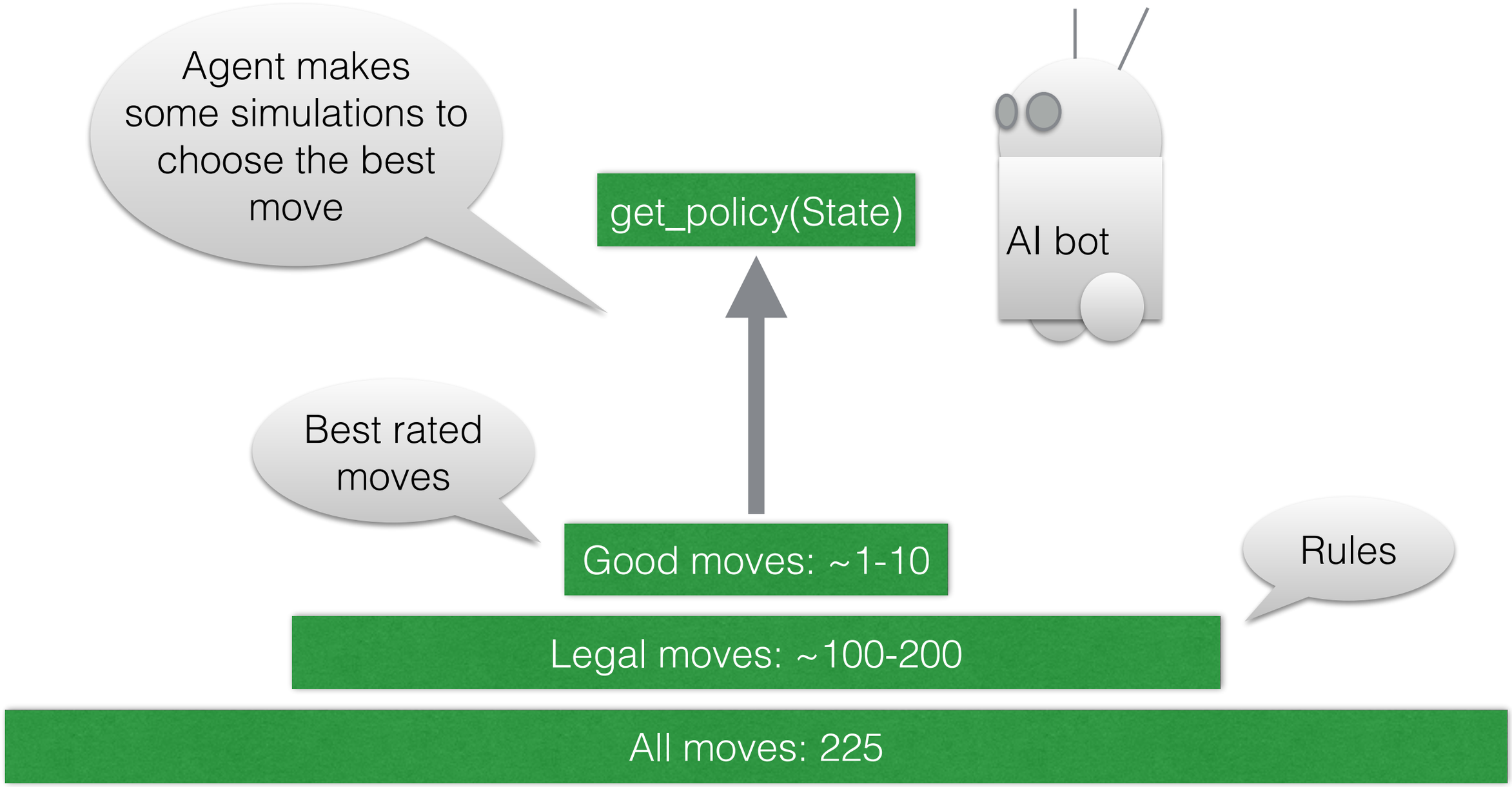
Human

Egghead

Lambda

Random

Choosing the move



AI policy

- If
  one good move -> return this move
  many good moves -> run K simulations of depth D

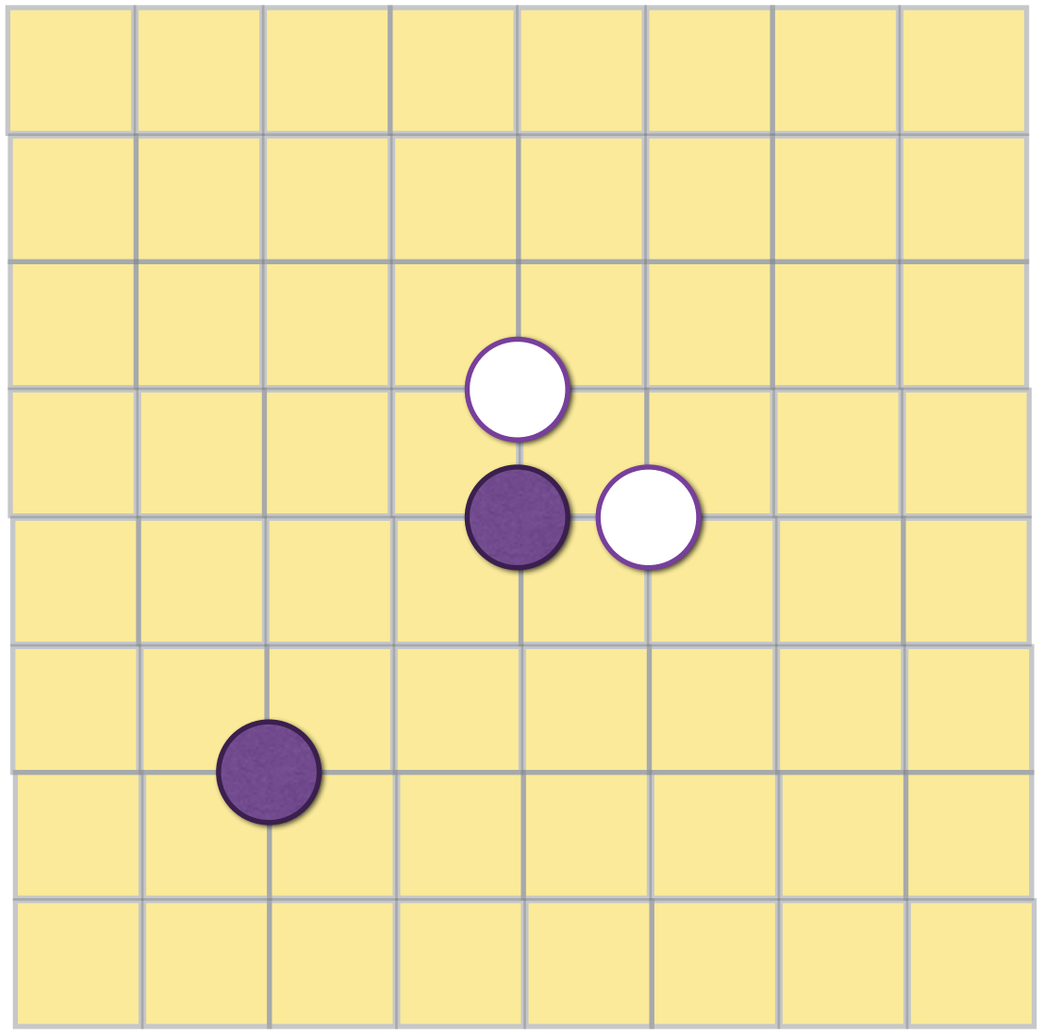
- policy:
  1-eps: get the move with highest score
  eps: get random move among good moves

- Qa table: key={State,Move}, value=Q

AI policy modification

- if state is unknown:
  select good_moves and return randomly one of them

- if state is known use greedy policy with optimistic expectation for unexplored moves or
  use soft-max



Turn

2b	2 x 8 x 1 bits	2 x 8 x 2 bits	. . .	2 x 8 x 7 bits
01	0010001000000000	00000 … 01000000		

State

Board

Board-to-state-to-board

- {State, 0} = board_to_state(Board) % Var = 0

- If State is not in Tab look up other variants
  1. No variants found
  2. {State1,Var1} found with values: [{Move1,V1,N1},…]

- If no variants found, select good_moves; choose any move randomly

- If {State,Var} found, choose optimistic max:
  {_,Mx} = max([{V+1/N, M} || {M,V,N} <— get_moves(State)])

- Find out the move with regard to the variant:
  Move = get_move(Mx,Var)

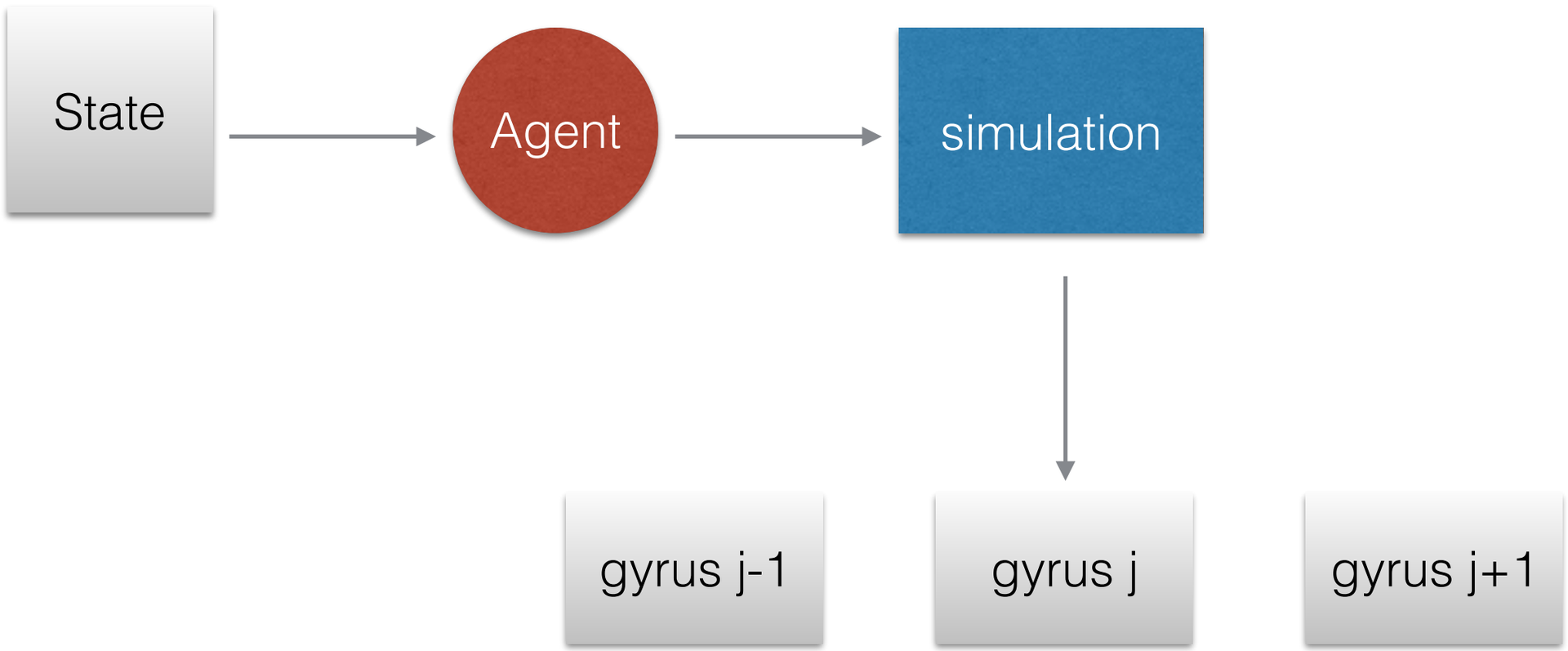
01 0010001000000000 00000 … 01000000

```
next_variant(Var,<<Turn:2/bits>>) -> no_more_variants;

next_variant(8,_) -> no_more_variants;

next_variant(4,<<Turn:2/bits,State>>) ->
  reflect(1,State,<<Turn>>).

next_variant(Var,<<Turn:2/bits,State>>) ->
  rotate(Var,2,State,<<Turn>>).
```

```
rotate(Var,J, <<Line:J/bits,Rest>>,NewState) ->
  rotate(Var,J+2,State,<<NewState,Rest,Line>>);
rotate(Var,J, <<>>,NewState) -> {Var+1,NewState}.


reflect(J,<<Up:J+2/bits,Right:J/bits,
            Down:J+2/bits,Left:J/bits,Rest>>,NewState) ->
  reflect(J+2,Rest,<<NewState,reverse(Down),Right,
                     reverse(Up),Left>>);
reflect(_,<<>>,NewState) -> {5,NewState}.
```

Agent interface

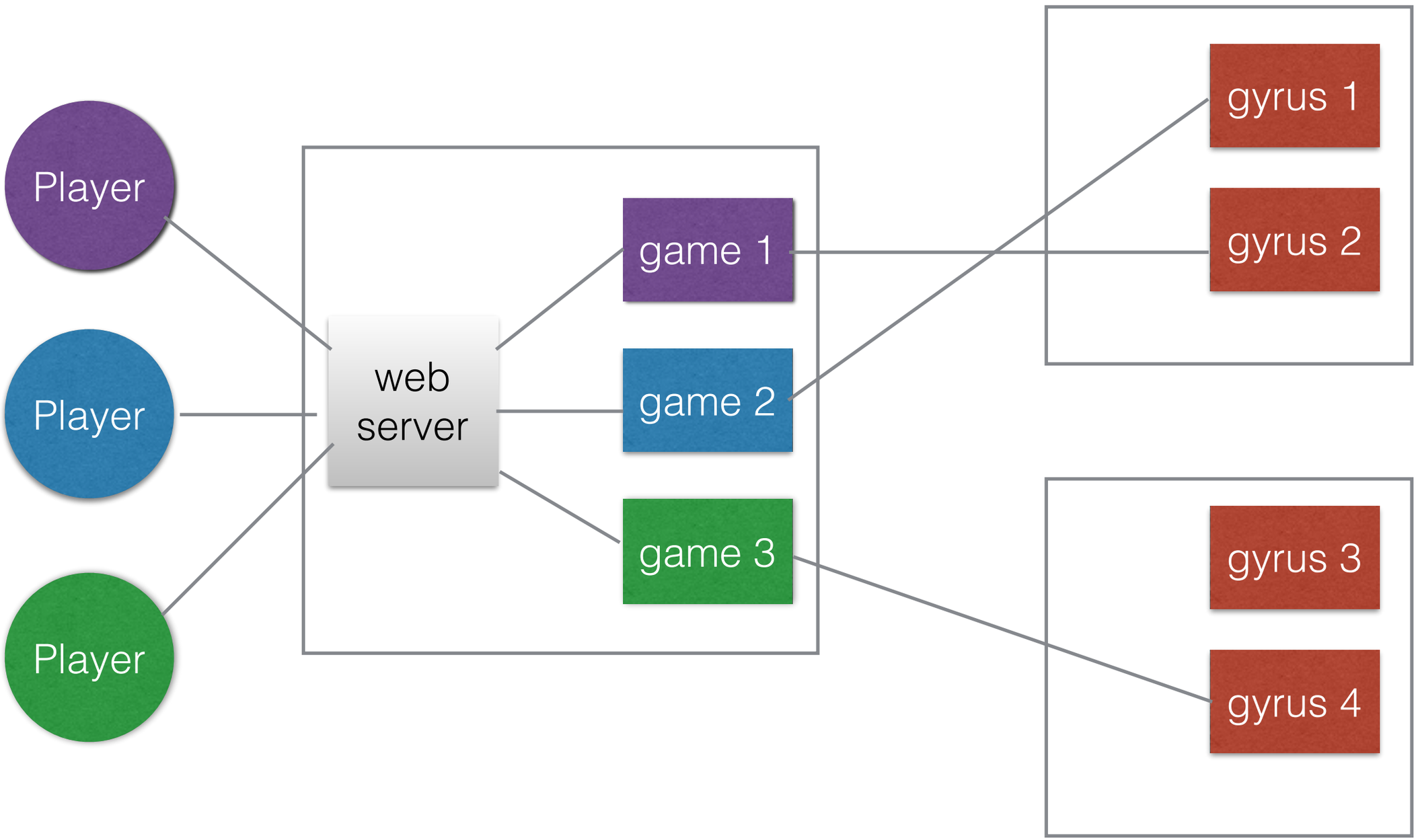
aggressive
strong

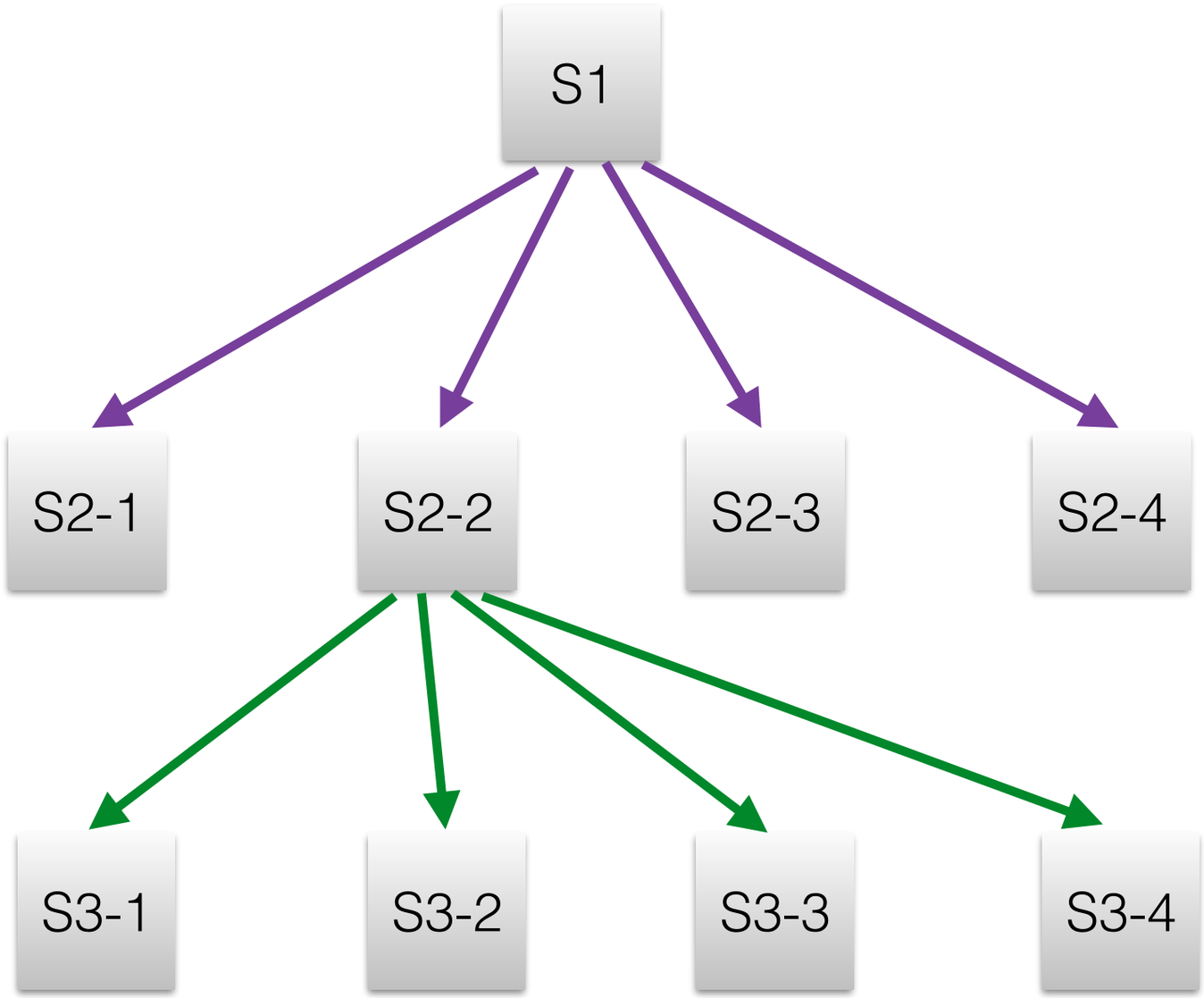


Setup

- The data are contained in the files corresponding to plies; all data are located in the main host
- Main host served as a place for web server and processes spawn for each game
- Other servers served to run gyres a few for each server

Multiple game playing




S1
S2-1
S2-2
S2-3
S2-4
S3-1
S3-2
S3-3
S3-4

Board

State

Lines

{
{e,e,e,e,e,e,e,e,e,e,e,e,e,e,e},
{e,e,e,e,e,e,e,e,e,e,e,e,e,e,e},
{e,e, … },
…
{e,e,e,e,e,e,e,e,e,e,e,e,e,e,,e}
}

<< 0101001000010001000000 … >>

[
{{X1,Y1},{X2,Y2},[e,e,b,e,…]},
…
]

statistics

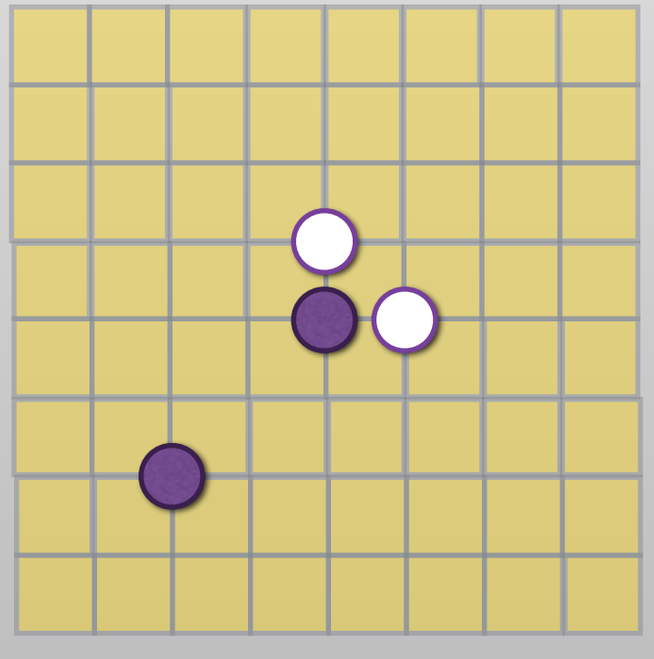
games played by bot: **120,356**

agains people:

Win: **20,356**

Lost: **1,356**

Draw: **657**

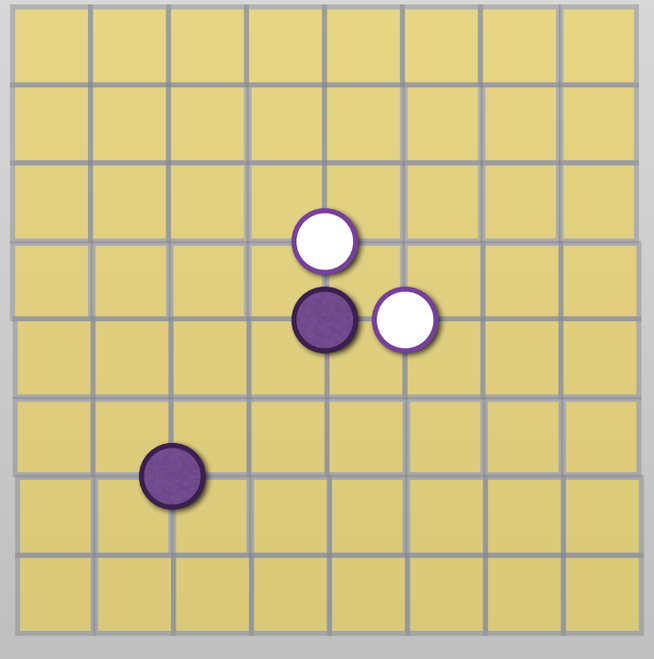


gomoku

Play free-style Gomoku or five-in-a-row board game against AI-bot or watch how it plays against other people.

During each game the Bot learns new tricks, which it will use against you next time.

Enjoy…

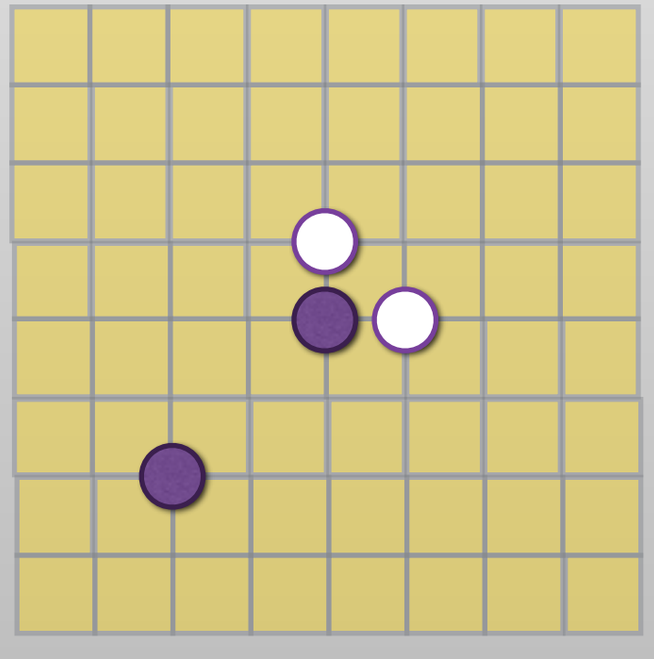


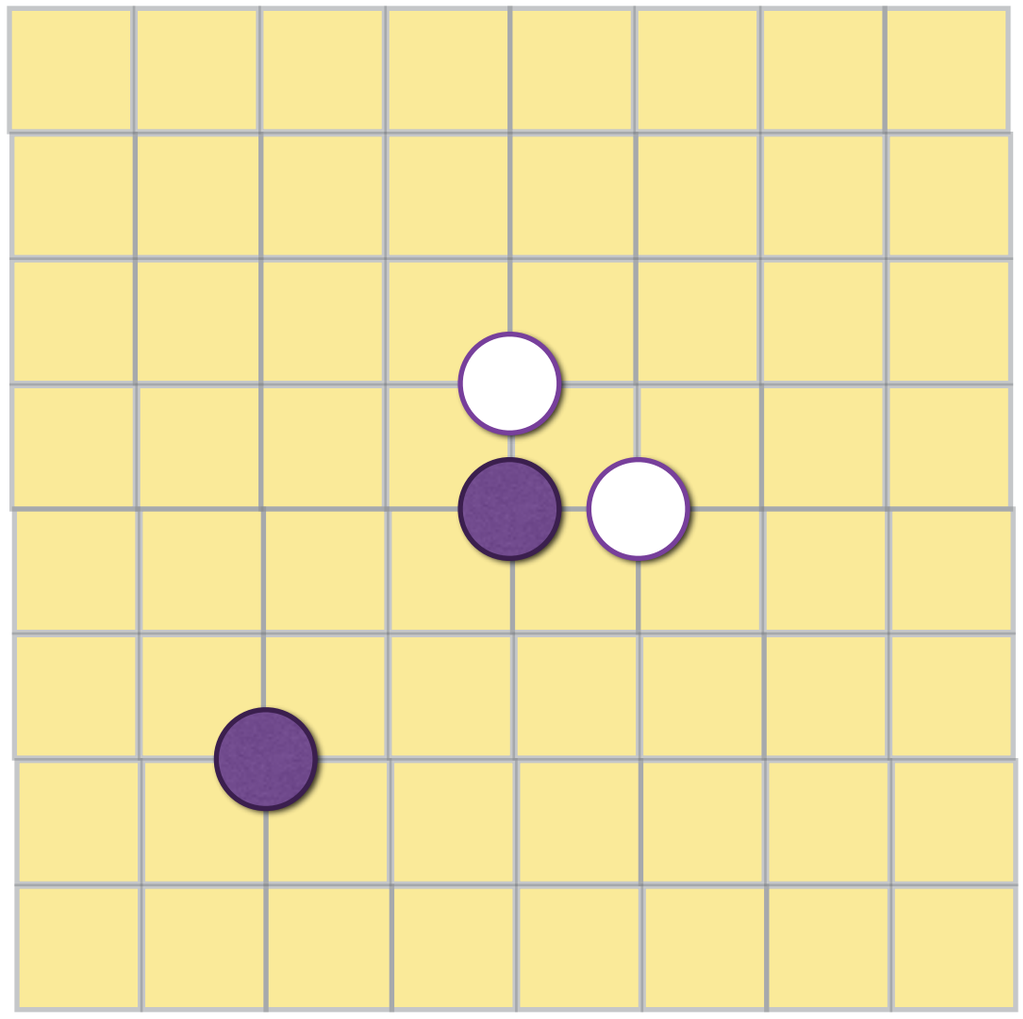
new game

easy

average

hard





your time

00:25

bot's time

00:14

make move