

Best Chess

Wednesday, 29 July 2015

It's on Kongregate:

<http://www.kongregate.com/games/pippinbarr/best-chess>

Released into the wild in general today, completely forgetting to notify Kill Screen, like a klutz.

Nice to have it tidied away though. Seems like I designed and built and released this in a week in July? Beat the July deadline to have something out and about. Feel good about that. Next up is? Probably a breakout game. BREAKSOUT might be the most fun? Need to solve the mobile problem or at least try to.

That's that.

Sunday, 26 July 2015

Okay I think it works. I still need to

- ~~* Explain the mechanism to Rilla~~
- ~~* Examine the transcript just a bit more (with other white starting moves) to make sure it's behaving~~
- ~~* Add ability for white to actually play a move at the start~~
- ~~* Add ability for black to offer draw in UI~~
- ~~* Add ability for white to reply to black move (in the hypothetical case black plays a move)~~
- ~~* Add ability for black to calculate a reply based on white rejecting a draw and playing~~

- * Add some UI feedback while calculating black move (probably display game outcomes?)
- * Make it look alright on mobile if possible

Saturday, 25 July 2015

Why don't I understand anything easily?

Friday, 24 July 2015

Ah, propagation.

(Also, remember that at the very top level you need to remember the ACTUAL MOVE to make in the case of the WIN or DRAW outcomes, in the first case because you play on, and in the second I suppose if white refuses the draw...)

SO

10 Get all possible moves in the active position

20 IF the 'next move' indicator is still valid (there is another move in the list to try)

 IF this is the first possible move in the list

 push current value of "?" to this depth

 make the next move

 increment next move indicator

 IF the move means game over

 IF active player WIN

 Change current value of this depth to "b" or "w" (depending on
winner)

 (And remember WHICH MOVE this is, so need to store pairs)

 Stop searching at level of this move (which is now a win)

 // This is integrate up (special short circuit case)

Undo move (because no more searching at this level)

pop value (and remember)

Propagate value up

IF current level == "?" THEN make it the popped value

(This is the only case, because we're talking about the value of

the

opponent's move here, and if there's already a draw you'd keep it

and

if there were already a win, we wouldn't have got to this search)

IF active player LOSS or DRAW

IF current value == "?" && LOSS

Change current value of this depth to "b" or "w" (depending on

loser)

ELSE IF current value == "?" || loss for this level && DRAW

Save this level as a "d"

Undo move

GO TO 10

ELSE IF game not over

GO TO 10?

30 ELSE IF the 'next move' indicator is too big
(you have exhausted all possible moves at this level)

IF you are at the very top

note final outcome (WIN, LOSS, DRAW)

```
IF win THEN make winning move
ELSE IF loss THEN resign
ELSE IF draw THEN offer draw
```

```
ELSE IF not at the top
```

```
// This is basically the 'integrate up' function
Pop the depth value
Undo the last move (to go one up the depth)
Integrate the depth value at this level based on active player
    (e.g. active win means stop searching, draw > loss)
propagate the current 'final' outcome UP
```

```
GO TO 10
```

--

Both sides are assumed to play perfectly (even though white might well not, being a human). Thus after white's move either:

- * Black resigns (because every move results in a white win with perfect play)
- * Black offers a draw (because every move results in a draw with perfect play)
- * Black plays a move (which leads in every case to a black win)

How does black establish which of these three things to do?

Basically you need to go to the bottom of the game tree and then propagate wins, draws, or losses back up the tree, reacting appropriately to each one:

- * If you find a win for BLACK then remember this as the 'current state of the game', but assume white won't make their previous move, undo it, and try the next one.
- * If you find a win for WHITE then remember this as the current state of the game, but black won't make its previous move, so undo it and try the next one

* If you find a DRAW then remember this as the current state of the game and keep searching?

Is that it? I think it's probably worse than this.

--

There's something else here, which is that black is searching only for WINS or DRAWS, as soon as it finds a LOSS in any game it has to give up on the move that it made that led to it completely (i.e. you don't need to check any of WHITE's other moves because you assume white would play perfectly and find that winning move). You basically have to search every version until you find a LOSS, but if you don't and you found one DRAW, then that move is a DRAW move, and if it's ALL WINS, then that's the perfect move, but you then have to go on considering because of course then WHITE wouldn't have played the move that led to that branch (and, rather, ALL moves ensuing from white's move must lead to a WIN for black, or draws...)

So when you find a BLACK LOSS you then UNDO white's move (to win) AND back's move (to get into that position in the first place)... but isn't this what I'm already doing? I'm confused?

ON WHITE MOVE: GO TO 1

1 CHECK current move index

* IF IN RANGE

MAKE MOVE

INCREMENT MOVE INDEX

PUSH NEW MOVE INDEX OF 0 TO INDEXES

* IF GAME OVERx

UNDO MOVE

POP MOVE INDEX

GO TO 1

* ELSE

GO TO 1

* ELSE IF NOT IN RANGE

(played all possible moves at this depth)

* IF DEPTH IS 0

(all games examined)

MAKE HIGHEST VALUED MOVE (RANDOM TIE-BREAK)

(should this actually be offer a draw, resign, etc.?)

WAIT FOR WHITE MOVE

* ELSE IF DEPTH IS > 0

UNDO MOVE

POP MOVE INDEX

GO TO 1

In The Beginning:

You play as white, you make a move, the game solves chess before replying.

EVALUATION

If you do -1 for a loss, 0 for a draw, and 1 for a win, then you might imagine that you'd be able to store the 'value' of any particular tree of play (from the second move) within an integer maybe? Which would make life easier. Is that a safe assumption? Anyway the big question is whether you can actually store the value of one of the trees coming from white's first move. Seems like there's on the order of 10^{120} games all up, how much is this reduced by knowing the first move?

MOVE LIMIT

It would be an option to set a 30-40 move limit (reaching it being a draw) to control the really deep stuff that explodes everything (even more)? I guess that would help out? Or just live with the 50 'eventless' move limit thing?

TECH LIMIT

Rather than worry about storage could also just embrace the potential impossibility of storing the score successfully and write it down to a technological limitation (much like the time limit?) – but this is kind of unsatisfying because I kind of want the point to be the tech ISN'T limited, that it will just go ahead with it and "would be able to" if given enough time.

In fact it's probably fine to 'encode' the number in a string? But then I'd have to implement the addition/subtraction myself? Which sounds assy? But is probably not that bad? And then it would be 'no problem' to store large enough numbers.