

Arash Parnia
Dr. Gordon
April 3, 2015

SHAH the GAS TANK chess

your program's NAME

Shah (Šâh) ([/ˈʃɑː/](#); [Persian](#): شاه [\[ʃɒːh\]](#), "king") is a title given to the emperors/kings and lords of Iran a.k.a. [Persia](#).

Shah is Farsi version of the King piece in chess.

Instructions on how to install, run, and play against your program

Program will run in windows and Macintosh by running shah.out in Unix based platform and shah.exe in windows. If the executable did not run then the code need to be compiled. To compile the program you need to have C++ compiler. After downloading and installing the compiler use `g++ gt.cpp -o shah.out` to compile the program. (Hopefully there will not be any errors) then run the program as instructed. Note that all the files must be in the same folder and you must be in the folder where the files are. To successfully compile the program. The files that are absolutely needed for the program to compile are gt.cpp, gt.h and Timer.h. Other files are just experimental functions and alternative but proven to be less efficient methods.

What language it is written in

The program is written in C++, it uses mostly C with minimal use of objects for timer implementation.

Which techniques you used (alpha-beta, iterative deepening, history tables, etc.)

Minimax with alpha beta pruning is implemented. To improve alpha beta the move generator will sort the moves based on capture, out of gas and friendly capture. Enemy capture will be the first moves in the sorted list because it's more likely that they will have a higher score. Then minimax with alpha beta pruning will use quiesce to avoid quiet moves. There was no visible need to implement history table and killer move since the search depth is high enough after alpha beta pruning.

How deep it typically searches (how many plies), from the opening position, and then later in the game

initial result : DEPTH 7 SCORE 0 in 5000 m/sec with 381797 evaluations

endgame result: DEPTH 9 SCORE -5600 in 1 m/sec with 56 evaluations

the game is designed not to go beyond 10 fold deep.

This might be a bug and it seems like the game is making good decisions up to level 7 only.

A brief explanation of the factors your program considers in its terminal evaluation function

A loop will assign predefined weights for each piece that is not captured and is not out of gas.

Then another function will generate all the moves from that state of the board and check if any of those moves will attack the king (king in check) and it will increase the score accordingly.

Below are the predefined weights in an array corresponding to every piece.

```
enum{EMPTY,WhiteKnightLeft,WhiteBishopLeft,WhiteQueen,WhiteKing,WhiteBishopRight,WhiteKnightRight,BlackKnightLeft,BlackBishopLeft,BlackQueen,BlackKing,BlackBishopRight,BlackKnightRight,};
```

```
int pieceValue[13]= {0,300,300,900,5000,300,300,-300,-300,-900,-5000,-300,-300};
```

How strong you think it is be specific (can it beat you? has it played any of the other programs?)

My program seems to have some kind of bug that will do strange moves in end game. Sometimes it plays good and sometimes it plays stupid. I might be able to find the bug before the tournament.

Anything unusual or unique about your program

I think perhaps pre sorting my move when I am generating my moves is unique to my program and it really speeds up the alpha beta. I have implemented bit board for move generation but after checking the time it seemed like that bit boards take longer than a normal array. But I have included bit board implementation code in other files accompanied

Any bugs you are aware of that cause it to crash or make illegal moves?

If input is not a proper format the program will get into loop.. Easy fix but wrong format for moves should not be entered anyway. The program will detect illegal moves and show appropriate message. It's the move format that cause crash for example if you enter c45cc35 it will get into a loop

Any bugs you are aware of that cause it to sometimes make bad moves?

This might or might not be a bug but at end game the program seems to be in favor of checking the opponent rather than saving its king