Ray Goh Xin Zhen

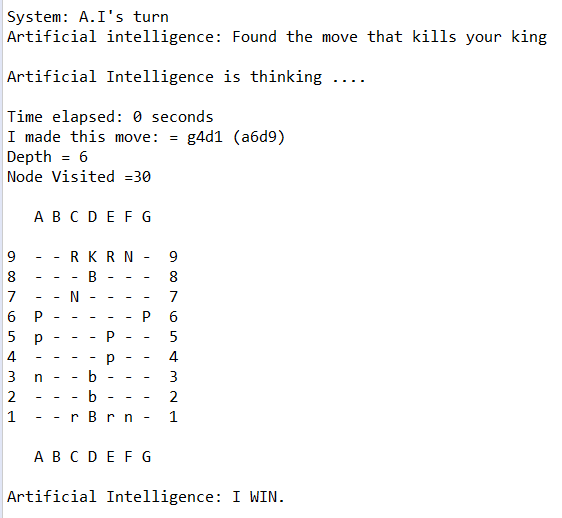
22nd March 2018

CSC 180 Intelligent Systems

Due: 23rd March 2018

**Program Name: CONOR MCGREGOR**

**Typical Mid-game Scene and End Game Scene**



**Instructions on How to Install the Program:**

There would be four files that needs to be compiled in order to run the program:

* AILegalMoves.java
* Newcular.java
* PlayerLegalMoves.java
* Stream.java

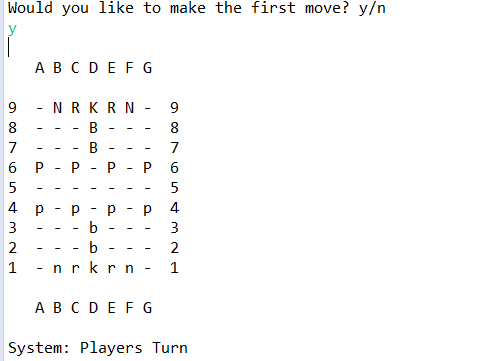
**Step one**: Download Eclipse – IDE for Windows.

* Download link: <https://www.eclipse.org/>

**Step two**: Launch Eclipse, click **File** > **Import** > and navigate to folder that contains the four files mentioned above and import all.

**Step three**: Either click the green play button or hit CTRL + F11 to run the program.

You should get a screen like this if the program compiled successfully:



**Language written in :**

JAVA

**Techniques used to improve search**

Alpha-Beta pruning techniques to work along with Minimax, apart from that it also has a move sort function which sorts all its available moves based on captures/suicidal moves and sort them by evaluation, then do minimax with each move.

It also has a timer to reduce depth if the program is taking too long to search. If it exceeds 4 seconds, it then reduces its depth to 5 (It almost NEVER happens but it is just provides assurance that program would not break the rule and get disqualified). AB-pruning only takes less than a second to complete a full depth 5 search.

**Searching depth**

The program starts of with depth 6 and maintain its depth throughout the game unless certain occasions when the timer exceeds 4 seconds and it switches its depth to 5.

**Terminal evaluation function**

Every piece in the chess board is given a value, with player pieces being negative values and AI pieces being positive values.

The program evaluates its terminal by adding the counts of total pieces remaining on the board.

Highest positive total value on heuristic: Best for A.I

Biggest negative value on heuristic: Best for human

Values of each chess piece are given as below:

Pawn = 10

Knight = 32

Rook = 50;

Bishop = 33;

King = 2000;

For player pieces the values are just negated. For example, a player’s King would have value of -2000.

**How is this program**

I have conducted almost 20+ games with the AI and I have not been able to beat it yet, I was not able to test the program with other’s as for now, but I should be able to play a few games with other programs in the weekend.

**Anything unusual or unique**

Other than alpha-beta algorithm, it has a move sort to sort moves before it runs minimax and a timer that reduces depth if AI is thinking for too long (usually never happens but only as insurance so program will not be disqualified).

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

None as far as I have tested.

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

None yet as far as I have tested and debugged.