**Artificial Intelligence**

**Fall 2020**

**Project #1**

**Grading Sheet**

Name: Allister Liu

Email: liu14@cooper.edu

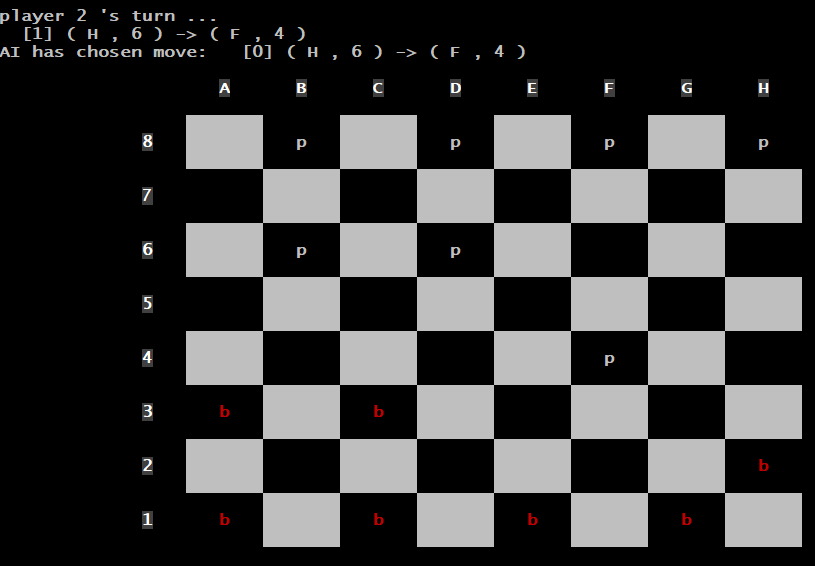
Grade: 88 – 2 (one day late) = 86

Notes:

You have implemented Checkers. I am testing it using Cygwin on my home desktop. I had to make a couple of minor changes to compile (e.g., I had to include <climits> for the use of INT\_MAX in one file). It was not difficult to get it to compile and run.

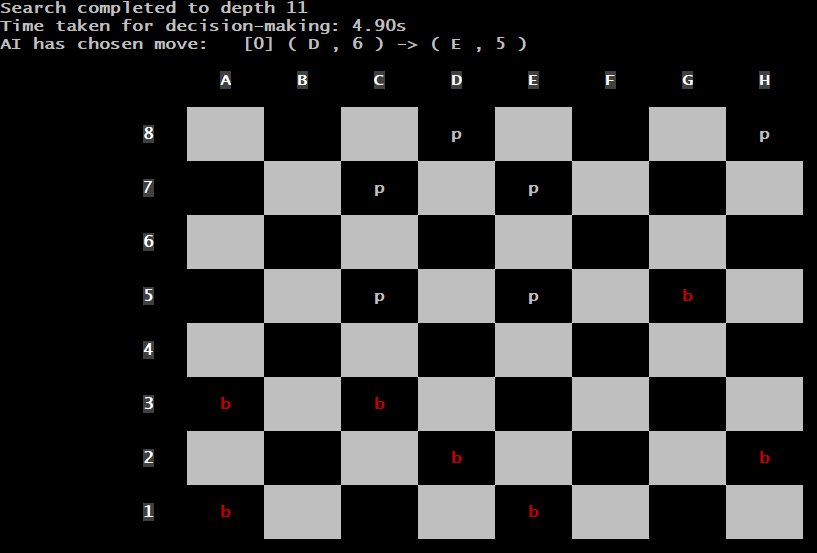
The program lets me decide both which player to play as, and which player goes first. I am playing my first game as red, giving the program 5 seconds per move. The interface looks good. I like that there are options to forget and save the game (I will not forfeit though, as I want to make sure the program knows how to win, and I haven't tested saving the game yet). For the first few move, the program is searching to depths ranging from 11 to 13, which if accurate, is quite deep! The initial moves seem fine.

Some trades happen early. When the program has a forced move, it plays it right away. One trade leads to a program's piece being pushed far forward; maybe too much. If I attack it, I may be able to capture it, but I am not sure. Here is the current board on my turn:

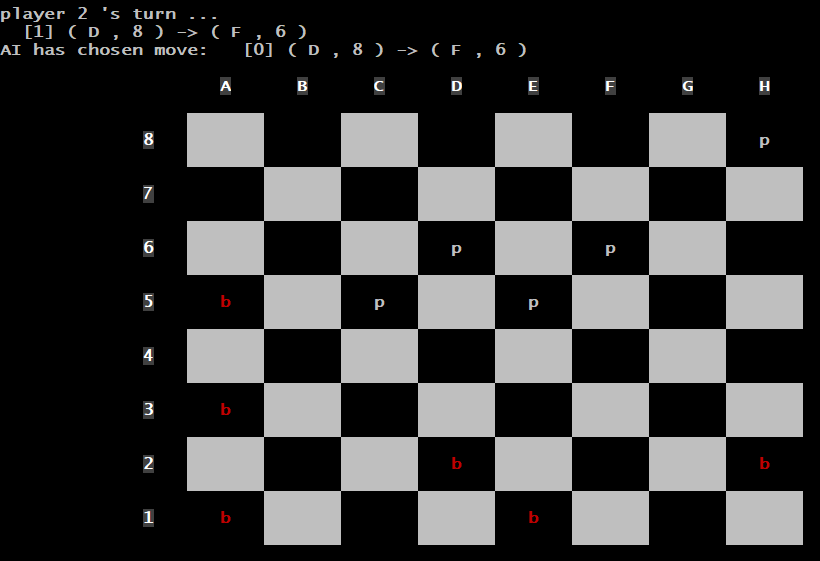


Even if I pull ahead material from this, it does not mean there is a bug. The last moved was forced, and the search before that was to depth 9. If I pull ahead material, it will take more plies than that from the point where it made its last search.

I do wind up pulling ahead a piece. I don't think there is a bug that led to this, it was probably inevitable from the previous position. Here is the updated board:



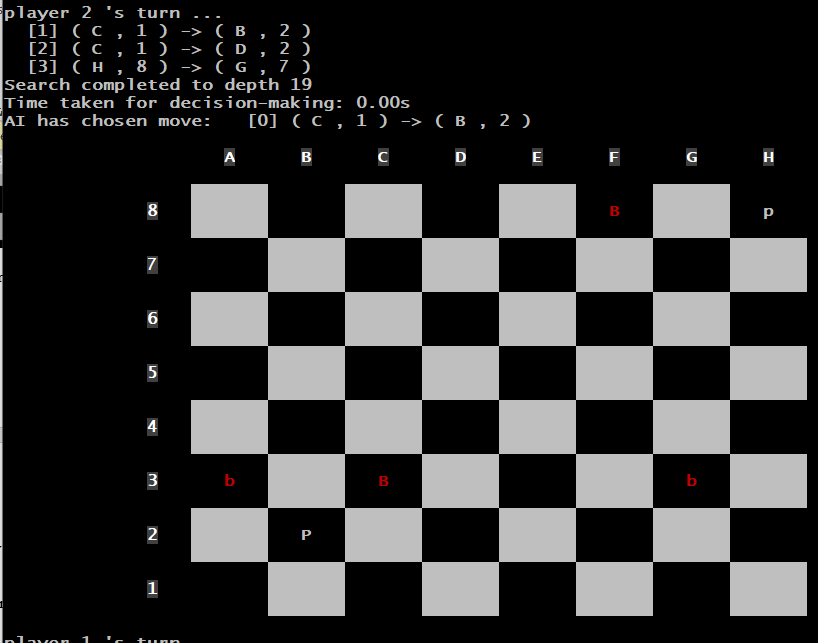
A bit later, the program forces a trade, and in the process, it gives me a free path to get a king. I don’t think that getting a king for me was inevitable within its depth search (which was 11 before the trade), but I am not sure. Here is the next updated board:



At this point, I should win unless I make a large blunder. I get my first king a few moves later (change my red 'b' to a red 'B').

I start attacking with my king, and eventually I gain another piece. I am careful in the process to not let the program get a king.

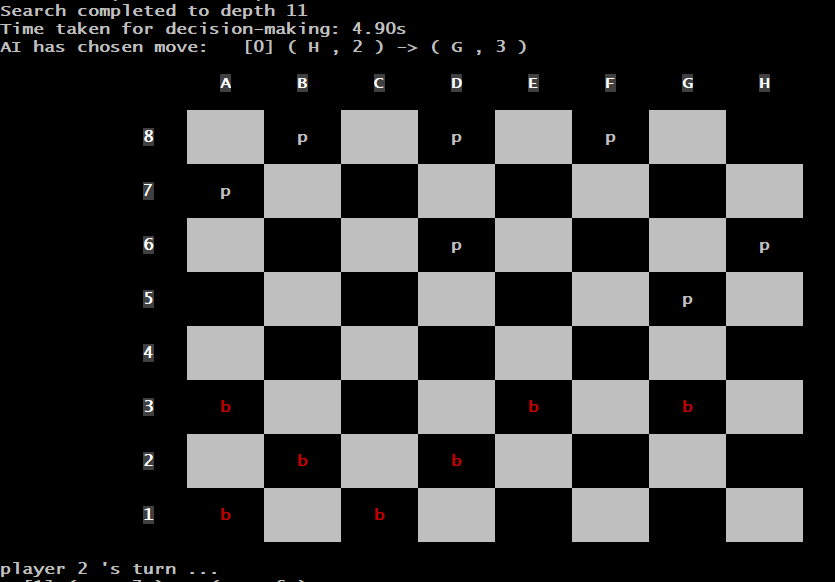
I go on to win pretty easily from there. The program does push off its inevitable loss as long as it can. Here is one very minor criticism:



Note the program made its move almost immediately, but it is strange that it reports a depth search to 19, when inevitably the full game tree will only extend to a depth of 4, no matter what we both do. Anyway, I win on my next move; the program exits gracefully.

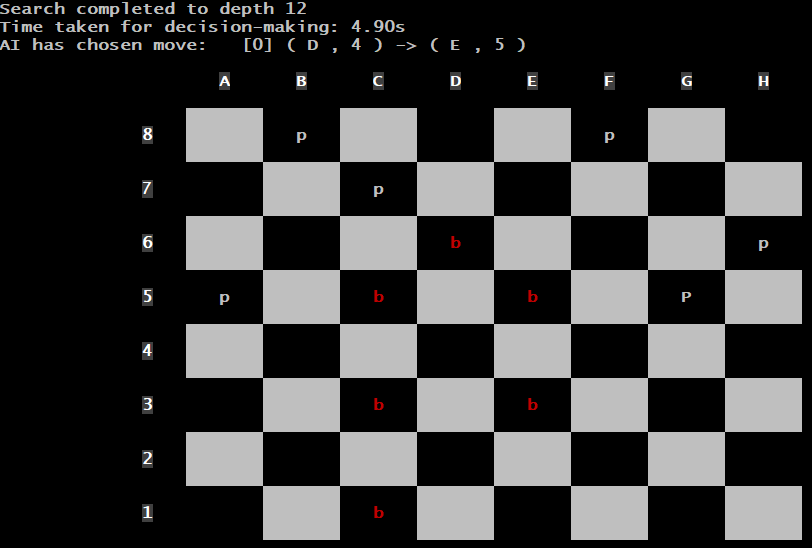
I start second game, this time moving second as white. The program is playing fine at the start. One minor criticism is that it moved one of its back-row pieces very early when it didn't need to. (In the previous game, it did a good job keeping the back row protected for a long time.)

A bit later, it has moved two back-row pieces, and now it makes a move that seems to be a mistake to me:

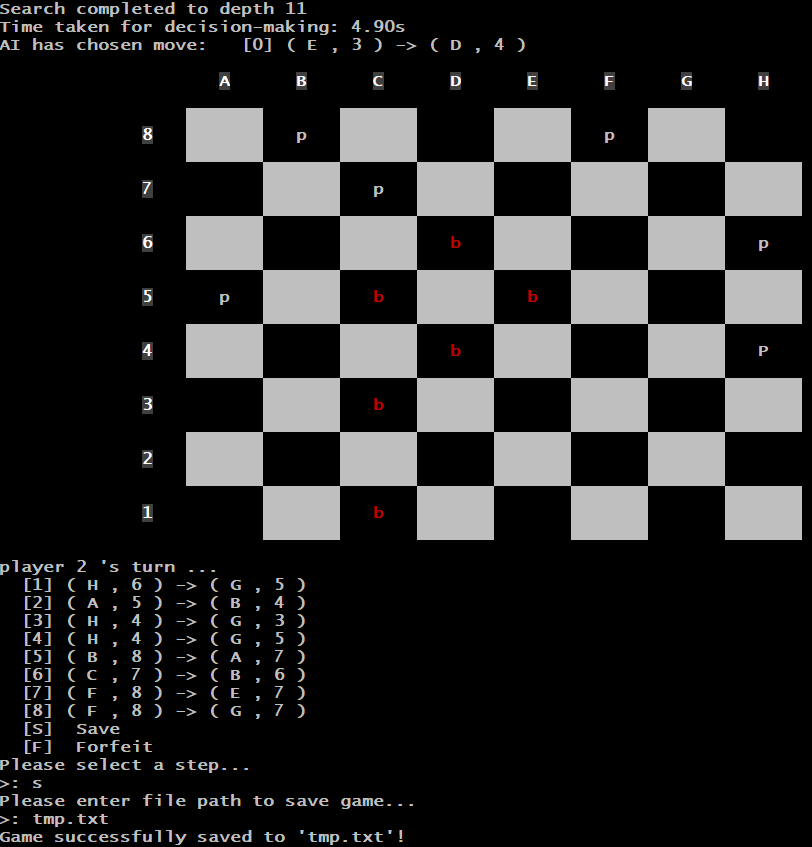


Now, I will move from g5 to h4, and soon I will get a king (and within the program's depth search). There is no way that this was inevitable. Unless I am missing something, it feels like a bug, or a flaw in the heuristic. (I wonder if the program might play better as white than as red? In the previous game, I saw no clearly bad move like this.)

I have been able to convert my king advantage to a piece advantage. So far, I have been unable. At one point, I thought I was going to pull ahead, but the program defended in a way that I would call clever if a person did it! Here is the updated board:



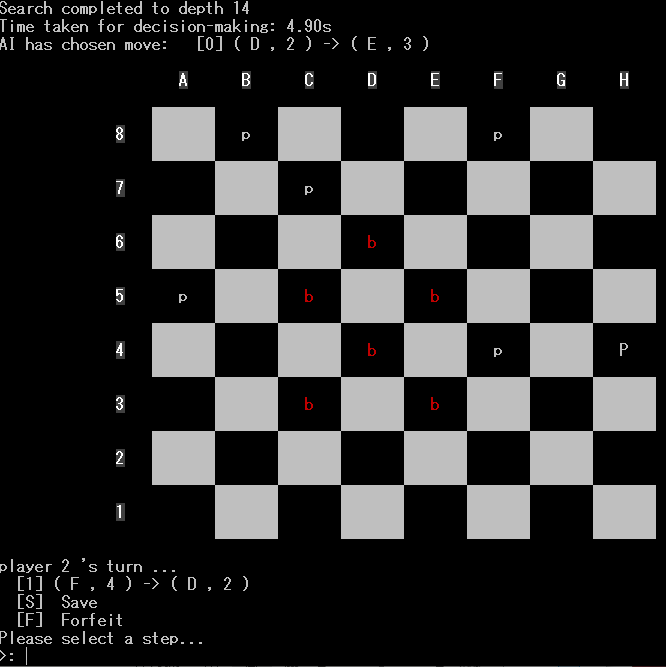
A couple moves later, I save the game, just to see if it works! It seems to work fine; later I'll try loading it back to make sure that it worked.



I'm afraid of c5->b6 here. I move from h6 to g5, and the program moves from c1 to d2 (I'm not sure why it doesn't move to b6 then; it probably sees something I don't.) Still, I may lost material soon.

One minor issue I am now noting: It is hard to distinguish the lowercase ps from the capital Ps! I almost forgot which of my characters was a king (the one at H4 above). This was not an issue for bs vs Bs. I change my font at this point so that the distinction is clearer:

A couple of moves later, the program ahs not moved to b6 (which I think would gain material for the program), and it now makes what seems like a terrible move:



It is forcing me to jump it, and I'll soon get another king. Instead of pulling ahead, it will be losing. I really doubt this is the start of a clever tactic; it seems like a clear bug. I gain another piece a few moves later (that one may have been inevitable). Soon I am ahead five pieces to one. Its piece is a king, which it moves to a double corner. It does not put off the win as long as possible, though; when I am close (it may see by then that a loss is inevitable, as it claims to be searching to depth 19), it just comes out and forces me to capture it and win quickly. (It should put off the loss as long as possible.)

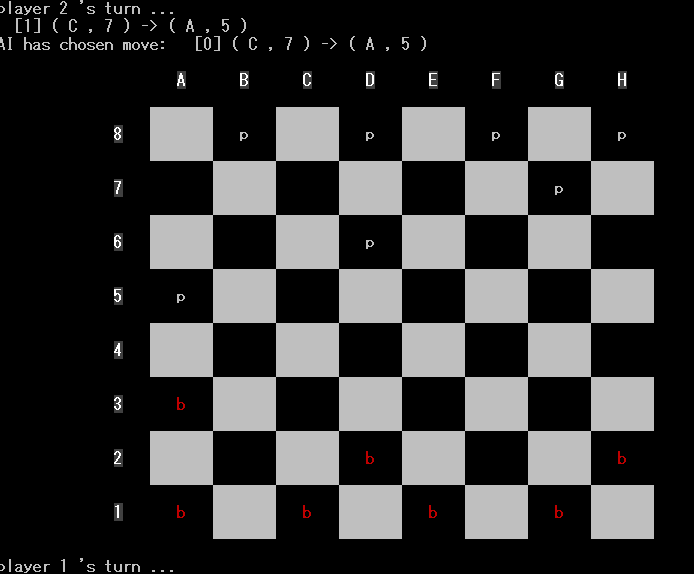
I try loading back the position I saved earlier, just to make sure it worked. It works fine. I will also use the saved board to figure out the file format, so I can try loading some of my own positions. (BTW, you were supposed to include a short writeup with he project, which could have explained things like this, but you did not.)

I recreate one of my crazy boards with lots of complex, multiple jumps available. Your program correctly detects all of the legal moves! (I try the forfeit feature after loading this board; it works fine.)

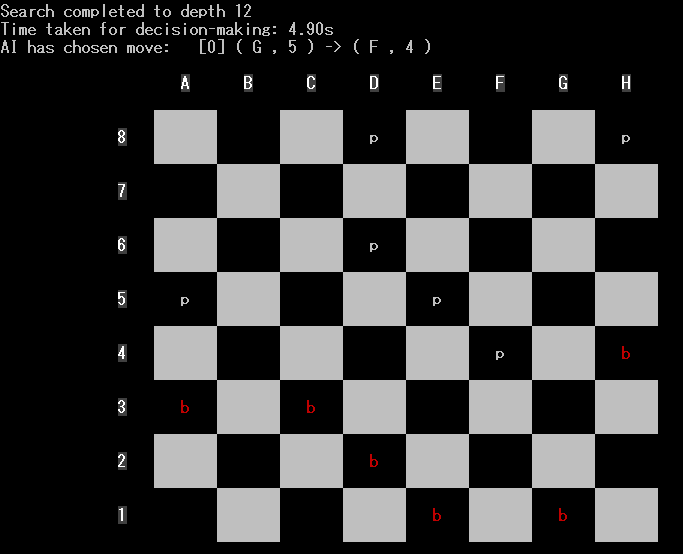
Next, I load a board where I have one king in a double corner, and the program has two kings starting far away. I give the program 5 seconds per move. I move back and forth in the double corner. At first, it seems that one of the program's kings is brought closer to mine; but then, it starts moving that piece back and forth close to me. The second king never moves. It is not making any progress, so I stop the game.

I watch the program play a game against itself, giving it three seconds per move. The feature works fine. I can't really follow the game at this speed. At the middle game, red pulls ahead a piece. (That's interesting, because in my games, I felt it might play better as white.) Ultimately, red is ahead three pieces (all kings) to two (with one king); but white's king is in a double corner, and both sides are moving back and forth. Red makes no attempt at progress, so I have to stop this game.

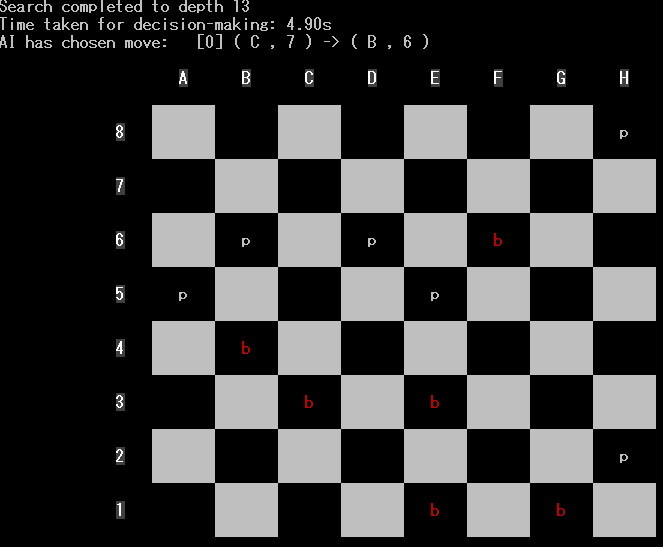
I'll play one more game as red, since the program gave me more trouble as white; I'll move second as red this time. Again, I am giving the program 5 seconds per move. A lot of trades happen early. Here is our current position:



The program continues to play well, and we are still even in the middle game:

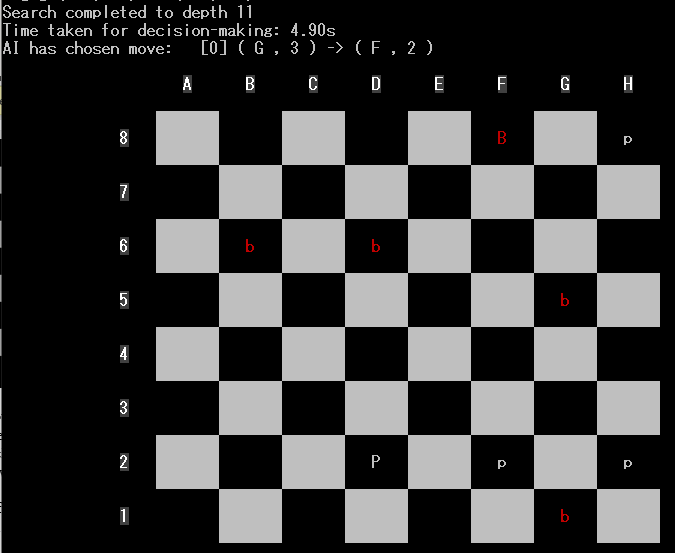


A bit later, I think I will pull ahead material soon:



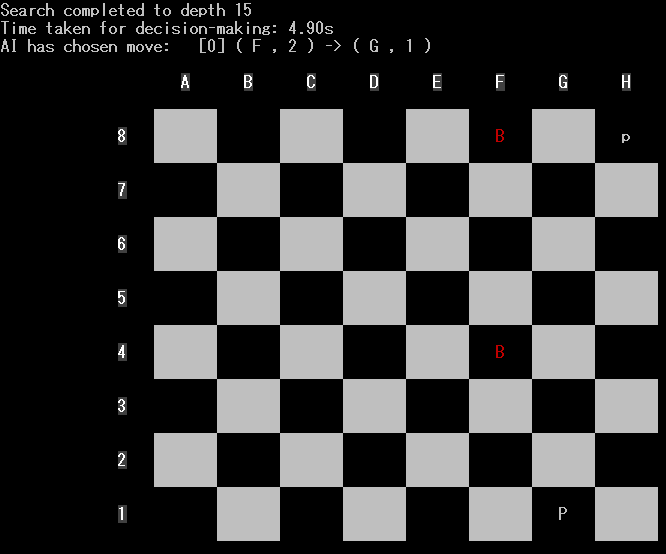
I will move from e3 to d4; I think its only move to avoid losing material right away would then be e5 to f4; but then after I move from d4 to e5, I believe I will gain material on my next move. (And since the program will see this, it may choose to lose material some other way.) Of course, its possible I am missing something. Nope, I was wrong; the next two moves go as I expect, but if I move to e5, it would move from b6 to c5, and we'd trade. So, I will progress to get a king instead (getting it a couple of moves ahead of the program). A couple of moves later, it lets me take a piece (probably seeing within its depth search that a loss of material was inevitable).

Ugh. I was ahead five pieces to four, and I should have had an easy win. But, I made a mistake! The program found a tactic to catch up. It will also even up the position in the process:



My last move allowed this. Oh well, I will continue; this should go to a draw now if neither side makes a mistake! (I'll have three kings to the program's two kings and one regular piece, but that should not be enough for me to win.)

After another trade, the program moves its king to a bold corner. I think this is still a theoretical draw, but I will try to force it out. Perhaps since its other piece is not a king, it may wind up trapped in some weird way after it starts to advance it. I think this will be a draw, but here is the current situation (with me as red):



We play some more, and eventually, there is another trade, and I cannot prevent the program from getting its king. It is a clear draw, one king a piece. I am stopping it.

I have enough to evaluate. The project is overall good. It plays a (mostly) solid game of Checkers. I played three games, achieving two wins and one draw. I should have won the third game too, but I made a mistake (but that's part of the game, and the program showed it doesn't miss a good tactic!). The program is capable of finding all legal moves, including complex multiple jumps. It usually makes good moves. In one game (the second), there are two moves that felt like bugs, especially the second one where it seemed to give up a piece for no reason (when I think it had a tactic that it could have used to pull ahead a piece). There is probably a subtle bug somewhere that only occurs in very specific sorts of situation. The program is weak in the end game. When it is ahead a piece, it doesn't know how to force an opponent out of a double corner. All in all, based on the strength of play (including not knowing how to win when it is ahead a piece in the endgame), I would give the program a B++. I would count that as an 88. The project was submitted one day late, so you also lose to points for that, bringing the project grade to an 86.