

CS4200 -- Project 3: Game

Please use either Java, python or C++

Game: 4-in-a-line

You have an 8x8 board, players take turns placing a piece on any grid. First player to get 4 in a line (either a row, or a column; diagonals are **NOT** counted) wins. The amount of time allowed for generating the next move is 5 seconds. Please make sure that your search will stop when there's no time left. Your program should ask the user to decide who is going to move first.

In order to make the process smooth, you are **REQUIRED** to use the following standard in the movement that your program will generate:

	1	2	3	4	5	6	7	8	Player vs. Opponent
A	-	-	-	-	-	-	-	-	1. e5 d5
B	-	-	O	-	-	-	-	-	2. e4 e3
C	-	-	O	-	-	-	-	-	3. f4 d4
D	-	X	O	O	O	X	-	-	4. e6 e7
E	-	-	O	X	X	X	O	-	5. g4 h4
F	-	-	X	X	-	-	-	-	6. d6 d3
G	-	-	-	X	-	-	-	-	7. d2 c3
H	-	-	-	O	-	-	-	-	8. f3 b3 wins

Strategies and Other Requirements

1. To get a full credit, your program must satisfy the requirements listed above, and **MUST** use **alpha-beta pruning** to determine the computer's move. Note that, you will need to make some changes to the alpha-beta pruning introduced in the book. These changes include: an *evaluation function* so that you can evaluate non-terminal states and a *cut-off test* to replace the terminal test so that your program will return the best solution found so far given a specific period of time.
2. In general, you are allowed **5 seconds** to run your algorithm, which should search at least 5 plies deep. If you use languages other than C or C++, the number of plies may be smaller. The sample file I provided implements iterative deepening search with alpha-beta pruning so that the program will search as deep as possible given different amount of time. But this is optional. **Your program just needs to handle 5 seconds.**
3. Given the same amount of time, if your terminal evaluation function is simple, then your program should be able to search deeper than one with a complex (but maybe better) terminal evaluation function. You can decide whether you want a deeper search and a simple evaluation, or a shallower search with a complex evaluation.
4. Your program is one player, and will attempt to defeat the human operator. Here, "X" means "computer", "O" means "human".
5. The program should detect when an illegal move has been entered and requires the human to re-enter a valid move, for example, trying to place on a non-empty space, or out of bounds.

6. Please refer to the tic-tac-toe.cc which implements MINIMAX as the base skeleton to develop the alpha-beta pruning.

A group of 2 is allowed to work on this project. You can also work individually. Each group needs to have one submission with both names on the report.

What to submit?

- Project report. Please discuss your strategies in generating the evaluation function and implementation of the alpha-beta pruning algorithm. < 3pages, word/pdf format.
- Source code + README (how to compile and run your code).
- Program output: Two sample outputs showing the user interaction with your program: one with the user winning the game; one with the user losing the game.
- Please create a folder called "yourname_4200p3" that includes all the required files and generate a zip file called "yourname_4200p3.zip".
- Please submit your zip file through Canvas.