

## CS4200 -- Project 4: Game (ISOLATION)

Due: 5/3/2020 before 11:59 PM

You are only allowed to use Java or C++

### ISOLATION

For this project, you will need to implement a two-player, time-based version of minimax with alpha-beta and an appropriate board evaluation function.

#### The Game:

The game has two players: X and O. The players alternate turns, with one player moving first at the beginning of each game. You can make X the symbol for your computer's player.

The board must be arranged thusly:

	1	2	3	4	5	6	7	8
A	X	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-
D	-	-	-	-	-	-	-	-
E	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	O

The first player to go starts at position A1 and the second starts at H8. In the above board example, the computer displaying the board has the first to move.

For each turn, a player can move like a queen in chess (in any of the eight directions) as long as its path does not cross a square already used or currently occupied. After moving, the space vacated by the player is designated as used and cannot be moved to again. **Note: only the space that was occupied is marked as used, not the entire path.**

You must implement a user interface that receives moves from the keyboard for those made by your opponent. Your computer will make its own move autonomously. In the following example, your computer went first and chose to move to D4:

	1	2	3	4	5	6	7	8
A	#	-	-	-	-	-	-	-
B	-	-	-	-	-	-	-	-
C	-	-	-	-	-	-	-	-
D	-	-	-	X	-	-	-	-
E	-	-	-	-	-	-	-	-
F	-	-	-	-	-	-	-	-
G	-	-	-	-	-	-	-	-
H	-	-	-	-	-	-	-	O

Computer vs. Opponent  
1. D4

Computer's move is: D4

Enter opponent's move:

Assume your opponent's move is E5, after entering the move using the keyboard the display would look like this:

Enter opponent's move: E5

	1	2	3	4	5	6	7	8	Computer vs. Opponent
A	#	-	-	-	-	-	-	-	1. D4    E5
B	-	-	-	-	-	-	-	-	2. D5
C	-	-	-	-	-	-	-	-	
D	-	-	-	#	X	-	-	-	
E	-	-	-	-	O	-	-	-	
F	-	-	-	-	-	-	-	-	
G	-	-	-	-	-	-	-	-	
H	-	-	-	-	-	-	-	#	

Computer's move is: D5

Enter opponent's move:

The game ends when one player can no longer make a move, leaving the other player as the winner.  
The board symbols are as follows:

- is an empty square

# is a used square

X is the current position of the X player

O is the current position of the O player

#### Requirements:

- Your computer is required to prompt for "Who goes first, C for computer, O for opponent: "
- Your computer will be required to make its move within 20 seconds.
- The board will be 8 x 8 with the coordinate A1 indicating the top left hand side of the board.
- The user interface must display the game as demonstrated above
- When a new game starts, It must prompt for the time limit per move and prompt for who goes first (computer or opponent)
- Moves must be entered as shown above
- The game must be implemented using the minimax algorithm with alpha-beta pruning
  - It will help if you also implement this with an iterative deepening.

**What to Submit and How to Submit it:**

- You can work in teams of two. Both team members must submit the project. Teams must be reported by the assigned due date. After the reporting due date, teams cannot be formed
  - You can also work individually if you want
- Source code + README (how to compile and run your code).
  - Do not assume that the grader will use your IDE
  - You need to instruct the grader on how to compile and run your program from the command line
- A report on the strategies that you employed in creating the evaluation function for the implementation of the alpha-beta pruning algorithm. Discuss the problems that you encountered and the steps that you took to resolve them. The report should be  $\leq 3$  pages.
- Create a folder called, "lastname\_firstname\_4200p4", that includes all of the required files, from which, you should generate a zip file called "lastname\_firstname\_4200p4.zip". If you are working on a team, include your teammate's name in the folder name as well. For example, if Jane Doe and Jim Beam are submitting a project, they would name the folder doe\_jane\_jim\_beam\_4200p4. The resulting zip file would be named, doe\_jane\_jim\_beam\_4200p4.zip. Only one team member should submit the project via Blackboard. The project must be submitted before the due date. Make sure to choose the most responsible team member for this task.

**No late submissions will be accepted**

**Grading:**

Minimax/alpha-beta implementation	50%
Including the evaluation function	
User interface	15%
Code quality	20%
Report and README	15%

**Note: You will be graded on your submitted code, If your program does not compile and run, you will not receive any points for the project.**