COMP 472/6721 **Project** **Fall 2012**

Due dates: October 22, 2012 + November 12, 2012 + November 26, 2012 + December 5, 2012
Late submission: 20% per day on each deliverable. No late submission accepted for the tournament.
Teams: You can do the project individually or in teams of 2.
Teams must submit only 1 copy of the project.
Purpose: The purpose of this project is to make you develop heuristics and state space search for a game,

---

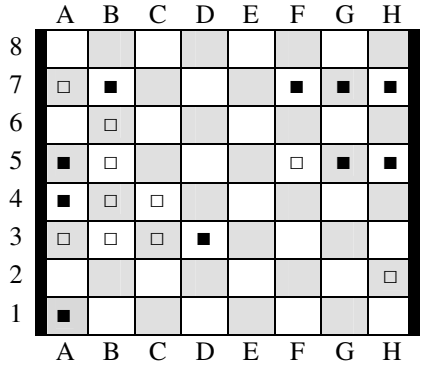
In this project, you will implement a 2-player game called **Magnetic Cave**.

Magnetic Cave is a 2-player adversary game where each player tries to build a "bridge" of 5 magnetic bricks within a cave whose left and right walls are magnetic. For the sake of this project, the bricks of one player will be represented by a ■ and the bricks of the other by a □. The version of Magnetic Cave that you will implement will be played on a regular 8x8 chess board.

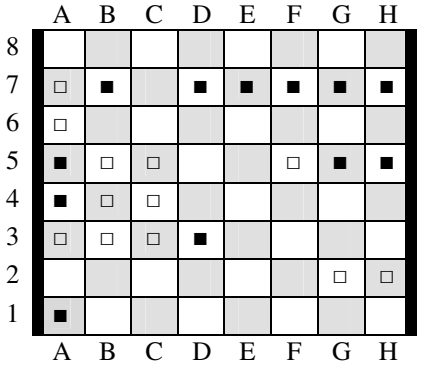
The rules of the game are simple:

- Initially, the cave (the board) is empty.
- Player ■ and player □ move in an alternate fashion, starting with ■. So ■ starts, followed by □, then ■ again, then □ again, ...
- Because there are two big magnets on each side of the cave, a player can only place a brick on an empty cell of the cave provided that the brick is stacked directly on the left or right wall, or is stacked to the left or the right of another brick (of any color).
- As soon as one player is able to align 5 consecutive bricks in a row, in a column or in a diagonal, then this player wins the game.
- If no player is able to achieve a winning configuration and the board is full, then the game stops and there is a tie.

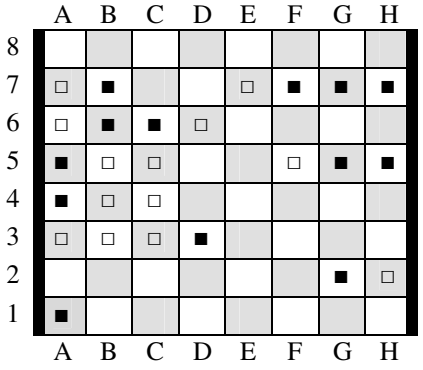
The following figures show possible configurations of the game.



Illegal configuration:
Brick B6 cannot be placed there.



Here, player ■ wins.
He built a bridge from D7 to H7.



Here, player □ wins.
He built a bridge from A3 to E7.

Your Task

In this project, you will implement a minimax algorithm to play this game automatically. You are free to develop the heuristic that you want, but your program must decide on the next move to take in at most 3 seconds (real time). In addition, you are also free to implement an alpha-beta search, as it may help you win the tournament (see below); however, no extra points will be given explicitly for the alpha-beta code.

Play Modes

Your program should be able to run in manual mode and in automatic mode. This means that you should be able to run your program with:

1. manual entry for both ■'s moves and □'s moves
2. manual entry for ■'s moves & automatic moves for □
3. manual entry for □'s moves & automatic moves for ■

After each move, your program must display the new configuration of the board.

Programming details

To program the game, you can use Java, C or C++. If you wish to use another language, please check with me first. It is not necessary to have a fancy user-interface. A simple command-line interface is sufficient

Tournament

To make the project more fun, we will organize a tournament between all the projects submitted. 1 bonus point will be allocated to your result in this tournament. If, at anytime, your program takes more that 3 seconds to decide its next move, you will automatically be eliminated from the game, and your opponent will win.

Report

Your final deliverable must be accompanied by a written report (~3-5pages) that:

- describes your program (how to run it, what the main functions and data structures are, …);
- describes and justifies your heuristic.
- describes and explains your results at the tournament (why you think you lost against your opponent – or why you think you won the tournament).

Deliverables

The submission of the project will consist of 4 deliverables:

Deliverable	*Functionality*	*Weight*	*Due Date*
Deliverable I	Minimax – no interface – no heuristic	40%	Monday October 22
Deliverable II	Minimax + interface + manual play (no heuristic)	50%	Monday November 12
Tournament	Minimax + interface + heuristic	85%	Monday November 26 The tournament will be held in the lab on Nov 27&28. Your results at the tournament will count for [0...1] bonus point on the final grade.
Deliverable IV	Minimax + interface + heuristic + report	100%	Wednesday December 5

Submission:

Each deliverable must be handed-in electronically by midnight on the due date.

1. Make sure that you have signed the expectation of originality form (available on the Web page; or at: http://www.encs.concordia.ca/documents/expectations.pdf) and given it to me.

2. In addition, write one of the following statements on your assignment:
   - For individual work: *"I certify that this submission is my original work and meets the Faculty's Expectations of Originality***",** with your signature, I.D. #, and the date.
   - For group work***:*** *"We certify that this submission is the original work of members of the group and meets the Faculty's Expectations of Originality"*, with the signatures and I.D. #s of all the team members and the date.

3. Hand in each deliverable electronically:
   - Create one zip file, containing all files for your assignment.

   - Name your zip file this way:
     - For individual work: name the zip file: *a1_studentID*, where *studentID* is your ID number.
     - For group work***:*** name the zip file: *a1_studentID1_studentID2*, where *studentID1* and *studentID2* are the ID numbers of each student.

   - Upload your zip file at: https://fis.encs.concordia.ca/eas/ as project1, project2, project3 and project4.

Have fun!