Due dates:	Monday Oct 19 2015 + Monday Nov 9 2015 + Monday Dec 7 2015
Late submission:	20% per day on each deliverable.
Teams:	Students registered in COMP 472 can do the project individually or in teams of 2. Students registered in COMP 6721 must do the project individually. Teams must submit only 1 copy of the project.
Purpose:	The purpose of this project is to make you develop a heuristic and the minimax algorithm.

In this project, you will implement a minimax algorithm and design a heuristic to play a game called **Hungry Birds**. Hungry Birds is an adversary game between 2 players: a larva, and group of 4 hungry birds.

The game is played on an 8x8 board that represents a yard. Yard "positions" are numbered horizontally from A to H and vertically from 1 to 8. As shown in Figure 1, initially, the larva is located at position D2 (on line 2), and the 4 birds are located at positions A1, C1, E1, G1.
Beyond line 1, there is a fence that birds are not able to cross, but the larva can. The goal of the larva is to reach line 1, before the birds catch it.

	A	B	C	D	E	F	G	H	
8									8
7									7
6									6
5									5
4									4
3									3
2				**L**					2
(fence) 1	**B**		**B**		**B**		**B**		1 (fence)
	A	B	C	D	E	F	G	H	

Figure 1: Initial configuration of the game

The larva can move diagonally forward or backward on empty positions, 1 position at a time.
The birds can only move diagonally forward on empty positions, 1 square at a time.
The larva and any one bird move in an alternate fashion. The larva moves first. Then, one bird moves. Then, it's the larva's turn again, then a bird …

If the larva manages to arrive anywhere on line 1, then it can escape through the fence and wins (birds are bigger and can't go through the fence[1]). If the larva does not get to line 1 and cannot move anymore, then the birds will catch it, eat it and win the game.
Figures 2 and 3 below show two possible endings of the game.

	A	B	C	D	E	F	G	H	
8									8
7					**B**				7
6		**B**							6
5									5
4						**B**			4
3			**B**						3
2									2
1			**L**						1
	A	B	C	D	E	F	G	H	

Figure 2: Example of a win for the larva

	A	B	C	D	E	F	G	H	
8									8
7					**B**				7
6		**B**							6
5	**L**								5
4		**B**				**B**			4
3									3
2									2
1									1
	A	B	C	D	E	F	G	H	

Figure 3: Example of a win for the birds

[1] and no, they cannot fly over the fence!

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 the lab).

Apart from the minimal requirements above (minimax + 3 seconds to decide), your project's grade will depend on the "extra" features you have implemented; for example, the development of a heuristic that takes into account several features, the use of alpha-beta pruning, or...

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 players (i.e. 2 humans playing against each other)
2. manual entry for one player, and automatic moves for the other (i.e. one human playing against your "AI")

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

Programming Details

1. To program the game, you can use Java, C or C++. If you wish to use another language, please check with me first.
2. It is not necessary to have a fancy user-interface. A simple command-line interface with a text-based output is sufficient.
3. In order to facilitate the running of the tournament (see below), we will use the following standards:

    a. The player (human or AI) will indicate its move by entering the coordinates of the moving player followed by the coordinates or where that player should go. For example, `D2 E3` indicates that the player at position `D2` will move to position `E3`.

    b. If a human player enters an illegal move (e.g. `A9, D2 D2, …`), then the player will only be warned and be given a chance to enter another move with no penalty[2]. However, if your AI player generates an illegal move, then it will automatically lose the game.

Tournament
To make the project more fun, we will organize a tournament between all the projects submitted. We will make pairs of AI players play against each other (e.g. your AI against another team's AI). If, at any time, your AI takes more than 3 seconds to decide its next move, it will automatically be eliminated from the game, and your opponent will win.
One bonus point will be allocated to your result in this tournament.
More details about the tournament will be provided later in the semester.

[2] The point of this rule is to avoid losing a game in the tournament because a human did not transcribe the computer's move correctly.

Deliverables
The submission of the project will consist of 3 deliverables + the tournament that will be demonstrated in the lab.

Deliverable	***Functionality***	***Due Date***
Deliverable I	manual game (no minimax, no heuristic)	Monday Oct 19 2015
Deliverable II	Deliverable I + minimax + first version of heuristic	Monday Nov 9 2015
Tournament		Tuesday Nov 24 & Wednesday Nov 25 with a possible final match the next week
Deliverable III	Deliverable II + your final heuristic + report	Monday Dec 7 2015

Report
Your final deliverable must be accompanied by a written report of about 3-5 pages in PDF format. Your report should:

1. Describe your program (how to run it, what the main functions and data structures are, …).
2. Describe and justify your heuristic. In particular, describe how you came up with your evaluation function. For example, what features of the game state you considered and why you chose them; how you balanced speed of the evaluation function with performance, and generally why you settled on the evaluation function you did.
3. Describe and explain your results either at the tournament or when you played against it (why you think your heuristic makes good or bad decisions).

Evaluation Scheme:

Deliverable	***Overall Weight***
Deliverable I	25%
Deliverable II	40%
Result at the Tournament	10% + [0pt...1pt] bonus
Deliverable III	25%

A detailed evaluation scheme for each deliverable will be posted on Moodle later in the semester.

Submission:
Each deliverable must be handed-in electronically by **8pm** on the due date (deliverable III can be submitted by midnight).

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:
  deliverable_N_studentID (for individual work) or
  deliverable_N_studentID1_studentID (for group work)
  where N=1 or 2 or 3
- Upload your zip file at: https://fis.encs.concordia.ca/eas/ as project1, project2, and project3.

Have fun!