Artificial Intelligence – Fall 2017 - Suggested Schedule and Advice for Project #1

I suggest that you spend the first three days thinking about the project and choosing a game. That will leave four weeks, and I suggest following the schedule below. The project is due the night of Sunday, November 5, by 11:59 PM. Barring extreme circumstances, *there will be no extensions*. The late penalty will be 1.5 points per day rounded up (and that can add up fast).

Week #1:
- Implement your data structures and your legal moves function. (This is much harder for Checkers than for Othello.) Efficiency matters more than elegance here!
- Create and test a function to display the board.
- Provide yourself a convenient way to specify any board position and whose turn it is, and test the legal move function thoroughly.

Week #2:
- Implement a function to apply a move to a position.
- Create an ASCII interface (or a GUI) and enable your program to play a complete game:
    - Allow the user to select who makes the first move.
    - On each of the computer's moves, choose between all moves randomly (or choose the move that maximizes a simple heuristic).
    - On each of the players' move, give the option of listing all the legal moves, or list them automatically, and verify that the player selects a legal move.
    - Display the board after every move and check to see if the game is over.
    - Provide an option for the computer to play against itself.

Week #3:
- Implement the alpha-beta search and iterative deepening (this is difficult for both projects):
    - If you implement a single recursive function, make sure you are handling the alpha and beta parameters correctly (especially if the perspective changes, as in negamax).
    - Make sure your program is handling the time limit correctly.
    - Provide yourself the option of producing an ASCII representation of the searched portion of the game space; this can be very useful for debugging.
- Implement a semi-simple heuristic and test your program thoroughly from various starting positions.

Week #4:
- Experiment with your heuristic function and discover one that makes your program play great! (This is somewhat harder for Othello than for Checkers.)
- Remember that the heuristic function should be efficient.
- If you organized your program well, you should be able to make it play itself using various heuristics against each other to help evaluate which ones perform the better than others.
- Reread the project requirements carefully and make sure you are not forgetting anything.

Feel free to read up on the game of your choice (in fact, I strongly advise it). Search for websites and applications that allow you to play against other programs or humans. For Othello, I have

found that Gunnar Andersson's program called Zebra plays very well, and for Checkers, you can play against Chinook on-line! You can also try playing your program against other programs or humans (but in the latter case, it may be unethical if you do not let them know).