

## **Instructions**

This project was created using Java 1.7.0\_10  
to compile the project:  
open terminal to file location  
javac TicTacToe.java

to run the project:  
open terminal to file location  
java TicTacToe x y

where x and y are one of:  
h : for a human player  
t : for thoughtful AI  
n : for naive AI

x is player 1 and y is player 2

ex:

player 1 is naive ai and player 2 is thoughtful ai

java TicTacToe n t

player 1 is thoughtful ai and player 2 is thoughtful ai  
java TicTacToe t t

## **Design**

The design for the project uses case based reasoning for both the thoughtful and the naive agents. I thought that case based reasoning would be good at playing the game because that is the way I normally think of tic tac toe when I play it. I remember which moves to take in certain situations instead of using a pattern based on my opponents last move. Since the rules of tic tac toe are simple and the board is small I did not think there would be a problem collecting enough data to make a thoughtful agent. The agents use a knowledge base of game boards and optimal moves to recognize a state and choose the optimal move for that state. The game board is represented as an integer array of length 9 in the game. There are 0's in the free positions, 1's in the position of player one and 2's in the position of player two. Since the agent is not always the same player it does not save boards as integer arrays. The boards are saved in a text file as a nine character string where 0's are free spaces !'s are the opponents moves and ~'s are the agents moves. The optimal move is saved on the line after the board it is related to. For example, if the agent was X's and this was the board:

X |   | X

-----

  | O | O

-----

X |   |

it would be represented as: ~0~0!!~00, and the optimal move would be position 3 which would cause X's to win while also blocking O's.

Each board represents four different boards since turning the board 90 degrees does not change the optimal move. For example, this board:

```
|  | X
-----
| O | O
-----
X |  |
```

is the same as this board:

```
X |  |
-----
| O |
-----
| O | X
```

but rotated 90 degrees clockwise. The solution to the first board is the same as the second board rotated 90 degrees clockwise. When looking for similar boards in the database the agent will rotate each board up to three times to see if the current board is just a rotation of one it has saved. It keeps track of the number of rotations so that it can do the same number of rotations for the index of the optimal move. Rotating the board significantly cut down the amount of knowledge that needed to be collected. If the agent cannot find a board that matches the current game board it will choose a random, open position. The knowledge was acquired by playing against both an “expert” and a random agent and asking for help when it did not know what to do. The thoughtful agent collected much more data than the naive agent. The naive agent also received some incorrect data. The results of running the agents against each other shows how important it is for a case based AI to have complete and correct knowledge.

## Tests

Thoughtful vs Thoughtful:

Running the thoughtful agent against itself results in a tie every time. It also results in the exact same moves being taken. This was what I expected to happen since there is no randomness to the choices taken. Even though there are four corners to choose from for the second move the agent will always choose the one it “knows” is best. The database does not have any duplicate states to choose from, so every time the thoughtful agent sees a certain state it will choose the exact same move.

This is the way the game will play out every time the thoughtful agent plays itself:

<div>     -----   X  -----     </div>	<div>     -----   X  -----  O    </div>	<div>    X -----   X  -----  O    </div>
<div>O    X -----   X  -----  O    </div>	<div>O    X -----  X  X  -----  O    </div>	<div>O    X -----  X  X  -----  O    </div>
<div>O    X -----  X  X O -----  O  X </div>	<div>O  O  X -----  X  X O -----  O  X </div>	<div>O  O  X -----  X  X O -----  O  X  X</div>

### Naive vs Naive

This test results in an even number of wins for player one and player two and a tie every now and then. The naive agent has some knowledge, but has to choose randomly some times. This leads to good moves occasionally, but most of the time the incorrect data and random choices will lead it to make a bad move. This was what I expected from this agent. It has an incomplete set of knowledge that leads it to make bad decisions, but the knowledge it does have helps it win when it can.

These are some examples of moves the agent chooses:

<pre>                 -----         X         -----       O   O             </pre>	<pre>                 -----         X         -----       O   O   X           </pre> <p>This was a good move, blocking the win.</p>
<pre>                 -----         X         -----       O   O   X           </pre>	<pre>           O       -----         X         -----       O   O   X           </pre> <p>This was a bad move, not seeing the position to block.</p>
<pre>           O       -----         X         -----       O   O   X           </pre>	<pre>           O       -----       X   X         -----       O   O   X           </pre> <p>This was a bad move, did not take the move to win.</p>

### Naive vs Thoughtful && Thoughtful vs Naive

I ran both combinations because the player that goes first normally has an advantage. It did not seem to make a difference in this case. Thoughtful almost always won with the occasional tie. I was expecting thoughtful to win or tie almost every time, but I also expected it to lose when it saw a board it had never seen before. It seems it has collected enough data to be “perfect” in the way that it cannot lose. This was very surprising since I thought it would require much more data than it has to be hard to beat, much less impossible to beat. The games between thoughtful and naive were almost always won on the thoughtful agents third turn so they looked similar to the example above with the exception of thoughtful making the correct choice and winning when it had the option.