

Earle Harrison
November 2, 2013
CS 4635

How to Play

1. Enter “java Board” into the command prompt (assuming your in the correct directory) and press enter to have a Thoughtful and Naïve agent play Tic-Tac-Toe
2. When the game is over, enter 1 or 2 to ask question 1 or 2 to the Thoughtful agent. Type “next” to ask the Naïve agent the same questions

Architecture

The architecture is similar to the one used in project 3, however, an array of moves has been added to the Agent class

The Tic-Tac-Toe board is represented by an array of characters of size 9. The agents are written to return a number which will index an ‘x’ or ‘o’ in the board array. There are 2 types of agents which play this game: the Naïve agent and the Thoughtful agent. The Naïve agent always goes first and is represented by ‘o’, and the Thoughtful agent follows by playing ‘x’. The Board class, which is the “main” class, also uses a “move” variable which keeps track of the moves which were made.

Both agents extend the “Agent” abstract class, which carries information about the current state and methods on how to play (the “Offense” and “Defense” methods). Since only the Thoughtful agent uses a strategy, the “Offense” and “Defense” methods are only used by this agent. It also checks if a square is empty with the “checkLegalMove” method.

The Naïve agent picks its moves by returning a random number generated by the “board” class. This agent uses the checkLegalMove method to check if the square is empty. If the square is empty, the Naïve agent picks the square using the pickRandomMove method, updates the current state, and ends its turn as the Board class increments the “move” counter. It also uses sets the “lastmove” variable, which stores the last move the Naïve agent has made.

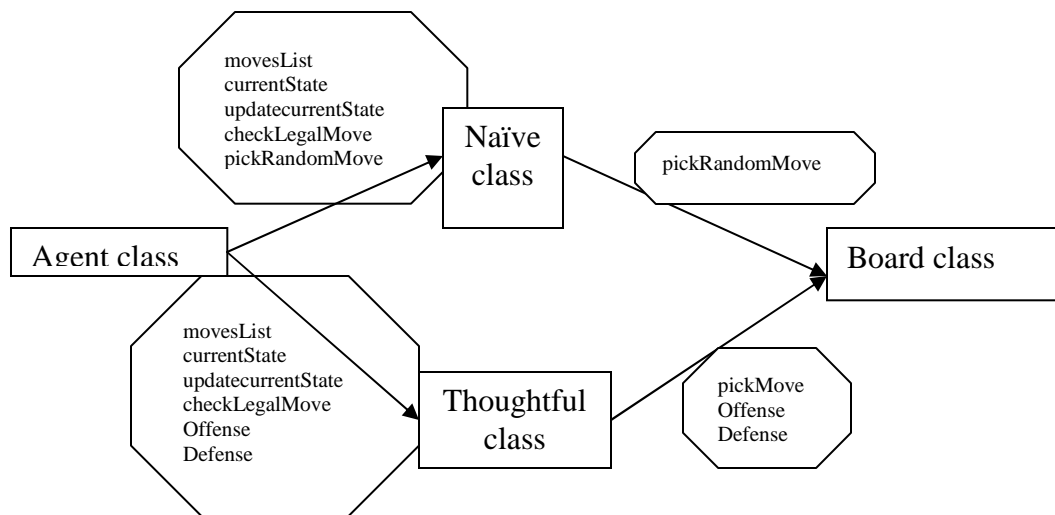
The Thoughtful agent plays after the Naïve agent (to make the game fair for the Naïve agent). This agent is written to pick the middle square on its first move; if checkLegalMove returns false, it picks the left-most top square, updates the board state, and ends the first turn as the Board class increments the “move” counter. It also uses sets the “lastmove” variable, which stores the last move the Thoughtful agent has made.

After its next turn, it plays offensively by calling the “Offensive” method, which takes in the random number generated by the Board class. This method checks the current state of the board for a possible win, by using the last move it played and checking if the square is empty. If it finds a winning square, it picks that square, updates the state, and ends its turn as the Board class increments the “move” counter.

If it doesn't find a winning square, it calls the "Defensive" method, which takes in the random number. This method finds an empty square, using the current state and checkLegalMove method, which will block a winning move for the other agent. If it finds one, it will operate similar to the "Offensive" method. If it can't find one, it will pick a random empty square chosen by the Board class.

After both agents have played, or if the "move" counter has reached "10", the Board class checks for a horizontal, vertical, or diagonal win. If there is no win, and the counter is less than 10, then the agents play again.

The game prints out the board after each move (which is the Domain knowledge trace) and the Strategic knowledge traces (a list of positions the agent has played and positions the agent believes it should play) after the game is done.



This figure of the architecture shows the relationship between each class and how a class would use a method originating from another class

Taxonomy

Questions the user will be able to ask:

1. **What action did you take on move "x"?**
The game will use the moves list from the agents to answer this question
2. **Why did you take the action you did at move "x"?**

Experiment

There were three games in this experiment, and Thoughtful won vertically on all three games.

Conclusion

Since the user can ask the agents questions about how they played, this should give someone who knows nothing about the game an idea of how its played, at least from the Thoughtful agent, since that's the agent which uses strategy. The agents are asked how they played on certain moves to show that they do know the rules of Tic-Tac-Toe and aren't just playing randomly (at least Thoughtful isn't playing randomly). They are also asked why they picked certain squares to show that they are making strategic decisions (at least Thoughtful is).