

Project #3: TAC Travel Agents

Your assignment is to design a team of agents that can trade for travel goods in order to satisfy the preferences of a number of clients. We will use the simulator designed for the Classic Trading Agent Competition (TAC) to test your agents by pitting them against each other in a tournament during our scheduled final exam period. This will be a team project with three teams of two or three people each.

Important dates:

Form your team	Monday, Nov. 21
Preliminary test	Friday, Dec. 16, 11pm
Tournament	Wednesday, Dec. 21, 9-11am (our scheduled exam time)

Forming teams:

You must decide on who your teammates will be by Monday, Nov. 21. As I said before, each team should consist of two or three people. Please notify me of both your team name and the names of your team members. If any of your team members does not have an account on the department's Solaris machines, please mention that and provide their Lehigh user ids.

Getting started:

In order to do well on this project, you will have to spend a lot of time designing and programming, so get started right away. I highly recommend that every team use the TAC Classic AgentWare (in Java) as a starting point for this agent. You can download this code from the TAC website. I also recommend that you review some of the publications on the TAC website in order to learn about different strategies teams have used in the competition. As you develop your agent, you should use the TAC Servers (available from the TAC website) or the server I've set up for the class to test your agent. You can play against the default dummy agents, any other agents you can find executables for, or schedule games against other teams. Do not assume that your agent will do well in the tournament simply because it performs well against the dummy agents. Keep in mind that your agent cannot make any assumptions about the power of the machine it will run on, the load on that machine, or about network latency. Thus, one of your agent's activities should be to determine its bidding cycle time.

Rules:

Your agents are subject to the rules set for the TAC Classic 2004 competition. As long as your agent plays by these rules, you are free to design your agent in whatever way you think gives you the best chance of winning. You are also free to discuss any aspect of the project with other teams, but since you will be competing against them, you may lose advantage by doing so. The only code you may reuse is the TAC Classic AgentWare. If you have chosen not to program in Java, and think you have found an equivalent base implemented in your language, then please seek my approval before you use it.

Preliminary Test:

In order to make sure that every team has a fieldable agent by the competition date, there will be a preliminary test of your agent in the competition environment. No later than 11pm Friday, Dec. 16, you will need to run your agent from one of the Sun Xeon workstations in Packard 122,

connect to the TAC server at `cse431tac.cse.lehigh.edu`, and win at least two out of three consecutive games against the default dummy agents. Once you have accomplished this feat, send an e-mail to me. Note, I will check your agent's performance by reviewing the game logs. Meeting this standard on time will account for 10% of your final grade.

The Lehigh TAC Tournament:

In order to put your agents to the test, we will have a tournament in which they face off against each other. For the tournament, the server will be hosted at `cse431tac.cse.lehigh.edu`, each team will run two agents on one of the other Xeon machines. By running two agents, we'll be able to average out some of the "luck of the draw." I will supply additional agents to play any slots not occupied by an actual team. We will play the game between and five to eight times. After each game, any team can request up to a 10 minute "timeout" before the next game is commenced. This can be used to adjust parameters in your agent, but probably is not enough time to make any coding changes. Your score for each game will be the average of each of your agents' scores. The winner of the tournament will be the team who has the highest sum of scores over all games. The tournament is scheduled for Wednesday, Dec. 21 from 9-11am (during our scheduled final exam period). I strongly suggest that all members of your team be present.

Final Submission:

Your project is due at the start of the tournament. You must provide electronic versions of source code, a Solaris executable (if your agent is in Java, then an executable JAR file will suffice), and a five to ten page report, double-spaced. The report should describe your team's design, and should include a short section on what each person contributed to the project. Be specific about individual contributions, as this will factor into each team member's individual grade. In addition, each team member will be asked to fill out a team evaluation form in which they rate their own contributions and that of the other members on the team. These evaluations will also play a factor in your individual grade.

Grading:

Grading will be based on completing the preliminary test, the quality and originality of your design, how your team performs in the tournament, the quality of your report, and your individual contribution to the team. Part of your individual grade will also be based on your own self evaluation and the evaluation of your performance by your teammates. The breakdown of grades is as follows:

Preliminary Test	10%
Design	40%
Performance	25%
Report	15%
Individual grade	10%