

## **ASSIGNMENT 5: QUORIDOR (Phase 2)**

**Objective:** We will play three tournaments on Quoridor( $n,m,k$ ). We will play three configurations of the game: Quoridor(9,9,10), Quoridor(11,11,12) and Quoridor(13,13,14).

As a reminder here are the notes for computing the margin of victory:

1. If player 1 wins then margin of victory is computed by number of moves taken by player 2 to reach their goal. Player 1 is allowed to pass or add a wall on their turn.
2. If the game hasn't completed and player 1 times out or makes an invalid move, player 2 gets 80 as the margin of victory.
3. If the game finished, player 1 is the winner, in computing margin of victory player 2 either times out or plays an invalid move or oscillates (so that the current number of steps post win become greater than 80) player 1 gets 80 as margin of victory.
4. If the game finished, player 1 is the winner, in computing margin of victory player 1 either times out or plays an invalid move, player 1 wins with margin of victory computed as follows:
  - reset the game position to when player 1 had made the winning move (i.e., remove all walls added by player 1 and move player 2's piece at the original position too)
  - compute shortest path to get to goal for player 2 -- that is the margin of victory.

The winner will get positive margin of victory as their score, and the loser will get negative margin of victory as their score.

### **What is being provided:**

Your new assignment packet contains three server codes for these three configurations.

### **Evaluation Scheme**

1. We will run three tournaments of this game. The allocated time for these games will be 2 minute, 3 minute and 4 minutes per player for the three games.
2. We will create a league fixture per game. In each league there will be many small groups of size about 5-6 teams. Each group will play a round robin and the top team will move to pre-quarter finals. After that it will be a standard tournament with pre-quarter final, quarter final, semi-final and final. In each game there will be two scores depending upon who plays first.
3. A team A will win a game against another team B if both games are won by A. If one game is won and other is lost then the total margins of victory will determine who won. Each win will get 2 points. In the rare event of a draw there will be 1 point.

4. The top team in a group will be decided as the team which has the maximum points. If multiple teams has maximum points the team that has the highest score (total margin of victory/loss) will move to the next level. If in a rare event two teams had the same score then the team that performed better playing first will be preferred. If that also can't decide we will toss.
5. Each league performance is worth 5 points. All teams which do not timeout and don't send invalid moves will get a score of 1.5 per game. Their performance in their group will determine the next bin score which can be a maximum of 2. All teams losing in pre-quarter final will get a total score of 3.5. All teams losing in quarter final will get 4. All teams losing in semi-final will get 4.5. The runners up team will get a 5. The winner will get a 15 and won't participate in the next league (giving a chance to two other teams to get a 15).

**Code:** Your code must compile and run on **machine named 'todi' or any machine with similar configuration present in GCL**. Please supply a compile.sh script for compilation. Also supply three shell scripts run1.sh, run2.sh and run3.sh for the three games. Executing the command ./runi.sh server port should start your player and start interacting with game server. The server will send instructions about whether the client is the first player or the second.

### **What to submit?**

1. Submit your code for your game player. **The code should be contained in zip file named in the format <EntryNo>.zip**. If there are two members in your team it should be called <EntryNo1>\_<EntryNo2>.zip
2. Submit at-most 1 page writeup (10 pt font) describing your choices and rationale for your player. This is not graded but failure to submit a satisfactory writeup will incur negative penalty of 20% of total score.

### **What is allowed? What is not?**

1. **THE LATE SUBMISSION OF THIS ASSIGNMENT IS NOT ALLOWED.**
2. You must use either the same partner as in Phase 1. Or you may work alone.
3. It is recommended that you work in C++. You may choose to not use the sample code. As an experiment we also allow the use of python. Using a different programming language makes it difficult to do a fair comparison, so we strongly recommend C++. If you choose to use python we will not be responsible for inefficiencies of the compiler.
4. Your code must be your own. You are not to take guidance from any general purpose AI code or problem specific code meant to solve this or related problem.
5. It is preferable to develop your algorithm using your own efforts. However, we will not stop you from google searching.
6. You must not discuss this assignment with anyone outside the class. **Make sure you mention the names in your write-up in case you discuss with anyone from within the class outside your team.** Please read academic integrity guidelines on the course home page and follow them carefully.
7. You get a zero if your player does not match with the interaction guidelines in this document.
8. We will run plagiarism detection software. Any team found guilty will be awarded a suitable penalty as per IIT rules.