

Assignment 5 - A Client-Server Connect 4 Game

In this assignment, we are going to extend the Connect 4 and “TOOT and OTTO” games produced in the last assignment. In this assignment, you are expected to adapt your system to allow distributed play within a geographically confined location; i.e. the 5th floor lab.

Your “games server” must be capable of handling at least ten users on independent machines. In addition, your system should be able to support at least five servers running simultaneously on the network. The users should be divided into multiple games, where each game should run on a single server. Remember servers are virtual not physical ideas.

In addition, you are asked to add two new pieces of functionality with one constraint:

1. Players can agree to stop and save partially completed games with a viewpoint of restarting the game at a later time. Potentially much later.
2. That all games, excluding games involving A.I.s (these are considered as practice games), represent results to be stored, queried, summarized (as league tables, etc), and visualized as entries in a single competition. This competition, or league, has no structure (in terms of regular fixtures, etc) and is considered to run indefinitely. All players must be able to access any required result, or subset of results, at any time.
3. To minimize the price point of your game server, these new requirements cannot use proprietary systems or elements from proprietary systems where any cost could be accrued.

You should consider the following:

- How do I handle starting and serving two different games?
- How do I start new servers?
- How can a client connect to a game?
- What happens when only one client connects, what happens when three or more try to connect?
- What synchronization challenges exist in your system?
- How do I handle the exchange of turns?
- What information does the system need to present to a client, and when can a client ask for it?
- What are appropriate storage mechanisms for the new functionality? (Think CMPUT 291!)
- What synchronization challenges exist in the storage component?
- What happens if a client crashes?
- What happens if a server crashes?
- What error checking, exception handling is required especially between machines?
- Do I require a command-line interface (YES!) for debugging purposes????? How do I test across machines? And debug a distributed program?

- What components of the Ruby exception hierarchy are applicable to this problem? Illustrate your answer. Consider the content of the library at: <http://c2.com/cgi/wiki?ExceptionPatterns> Which are applicable to this problem? Illustrate your answer.

This assignment is worth 20% of the total marks of the CMPE410 assignments; and must be completed in three parts:

Part 1: Your rationale for your design decision must be completed by” **Tuesday March 29th @ midnight**

Part 2: Your System Design (written as a set of assertions) must be completed by: **Tuesday March 29th @ midnight**

Part 3: Your system must be completed by: **Friday April 8th @ midnight**

In addition to the practical demonstration, you are required to hand-in:

- 1) A detailed rational for your decisions with regard to the above design questions.
- 2) A list of deviations in the contracts implemented form the contracts specified in Part 1.
- 3) A copy of the code
- 4) A description of any additional testing beyond that described by your contracts.
- 5) A list of known errors, faults, defects, missing functionality, etc. ...telling us about your system’s limitations will score better than letting us find them!

Please hand in all components by emailing them to {zuhori, jimm}@ualberta.ca and hence all sub-components by definition must be machine readable. In addition,

- The Subject Line should be in a specific format - Assignment 5: Group 1 [, Part 1...]
- Filetypes should be only of .doc or .pdf
- Filenames for code should be in ruby format (all lower cases, with words concatenated with a dash)
- For each assignment with a code section, there must be a primary executable file that is the main entry for other codes. This has to have the format: assignment5_main.rb (where 5 corresponds to the appropriate number in the assignment sequence). This main file should also contain a comment section with a list of the group members if necessary and some description of the runtime requirement of the code (e.g. commandline activation format)