Bora Sahin, 11/01/2015
Backgammon Client / Server Application: System-Wide Requirements

Status of This Memo

This memo defines an experimental protocol and requirements for client and server applications of Backgammon online game.

Abstract

The backgammon application requirements and protocol were developed as part of the term project effort, in partial fullfillment of SWE544 Internet Programming course requirements at Bogazici University.

The application includes a server (i.e. backgammon server) and a client (i.e. backgammon client) implementation as well as the definition of a protocol for client-server communication. The system features playing or watching a backgammon game as online.

Table of Contents

1. INTRODUCTION

Backgammon gaming system (in a simplified form) is an online game application and accompanying protocol for the Internet.

The application features a server (i.e. Backgammon Server, hereforth "BS") that only keeps user credentials (only username in this case) during their connection to the server. Once the connection ends, all the user credentials are removed from the server. BS forms the backbone of the gaming system. All the communication goes through the BS, which translates to no direct communication exists among clients.

The application also features clients (i.e. Backgammon Client, hereforth "BC"), which are Python-based gaming client applications that communicate with BS to log in, and then either play or watch a game.

The two key actors on the system are the User and the System, which can further be broken

down into Playing or Watching Users for gaming, and the client and server-side components of the backgammon gaming system as a whole.

This document is intended for the use of development as well as testing. It contains the elicited system-wide requirements and an analysis of requirements for the client and server side components of the system. The protocol to be used, between the client and the server however, is defined in a separate document.

1.1. Definitions

This document uses the following definitons:

Server, Backgammon server, BS: is the server-side component of Backgammon gaming system that uses BC-BS protocol for messaging and keeps track of the presence of clients during their connection to the BS, along with Internet addresses, some backgammon gaming logic and handles logging

Client, Backgammon client, BC: is the client-side component of the Backgammon gaming system, that allows for logging in to the server and playing or watching a game

Player: The player is any real person playing a Backgammon online game

Watcher: The watcher is any real person watching a Backgammon online game

User: Either player or watcher

Protocol, BC-BS: The text-based messaging protocol, based on TCP, used for the communication of BC and BS components. The protocol is defined in a separate RFC document (SWE544_RFC_2)

System: Refers to the backgammon gaming system as a whole, incorporating both server and client side code

2. SYSTEM-WIDE REQUIREMENTS

2.1. Functional System Requirements

2.1.1. The Backgammon gaming system is a client-server online game system that works on the Internet.

2.1.2. The system utilizes the Transmission Control Protocol (TCP).

2.1.3. A user of the system is any human actor who interacts with the Backgammon gaming system through the use of the client. The users of the system shall be able to:

2.1.3.1. Log in to the system, through their installed client software, by providing:

- A valid username

Validation of username shall be checked by the client. A session shall be opened unless that username is not requested by another client before. If a user loses his/her connection to the server, then all the information attached to the client shall be removed from the server and in the case of a player who currently has a game loses his/her connection, then all the relevant parties (his/her

opponent, watchers) shall be informed and their session shall be closed. The user shall need to re-login using the client software.

2.1.3.2. Play a backgammon game. A player can't choose his/her opponents. Server randomly selects an opponent. A user can play only one game at a time.

2.1.3.3. Wait an opponent. If the server can't find an opponent for a recently-logged-user who wants to play, then it shall put that user into a waiting list. When other users who want to play backgammon log in, the server shall randomly match them together.

2.1.3.4. Watch a game. A match-to-be-watced shall be randomly chosen by the server. A user shall watch only one game at a time. If there is no match in the server, then the user shall be presented with a menu shown when the user first logged in to the system. Watcher should not send any message to the server except leave request.

2.1.3.5. Leave the server. A user shall be able to leave the system only when he/she is either waiting an opponent to play a game or watching a game.

2.1.4. The system assumes all system components are able to listen to and receive traffic from a fixed port number (namely 10001). Seeing as many system components will be behind NAT-enabled routers, the testing and use of the system shall comprise port-forwarding on the routers in order to be able to listen effectively to fixed port number. For the purposes of the Backgammon gaming system, no other port number than those specified shall be used, and this set of requirements **assumes each endpoint of the system is available on a public IP and the NAT layer is fully transparent to the Backgammon gaming system**

2.1.5. In order to test requirement 2.1.4 in a stable environment, the software shall be tested on a LAN, or similar virtual private network environment.

2.2. Non-Functional System Requirements

2.2.1. The system shall implement the BC-BS protocol as defined in SWE544_RFC_2.

2.2.2. The system shall be robust, that is, especially to erroneous user input. Both the server and client (and particularly the server), are expected to graciously handle unexpected protocol messages, protocol message fields, message field contents, unrecognized character encodings, etc.

2.2.3. The movement of checkers should follow backgammon notation.

2.2.4. Message size can't be bigger than 1024 bytes including message body and header.

3. BACKGAMMON SERVER REQUIREMENTS

3.1. Backgammon Server Functional Requirements

3.1.1. The server shall listen to BC-BS protocol. TCP port number 10001 shall be used for the incoming traffic.

3.1.2. The server shall support clients to log in.

3.1.3. The server shall keep track of all the players and watchers currently logged to the system.

3.1.4. The server shall keep client information accessible (keyed) by username and if a client leaves the server for some reason, then the server shall remove all the information pertaining to that client. In this sense, user information kept in the server shall be valid only during their connection to the server.

3.1.5. The server shall send periodic heartbeat messages to the clients every 30 seconds. If a response is not received from a client (2 heartbeats back to back), the server shall remove all the client data from the server. In the case of a player who currently has a game loses his/her connection to the server, then the server shall inform all the relevant parties and close their session(s).

3.1.6. When two players are selected by the server to play a game, one's turn and color shall be decided by the server and each player shall be informed about his/her opponent, color and turn before the game begins.

3.1.7. Every match shall have a watcher list and all the moves shall be sent to all the watchers in the list.

3.1.8. Dice shall be thrown by the server upon the request of a client and be delivered to all the clients of the match including the requestor.

3.1.9. Board state of every match shall be kept.

3.1.10. A move made by a client may be accepted or rejected by his/her opponent. The BS should wait this before updating the board state of the match. Only after the move is accepted by his/her opponent, then the board state shall be updated. If the move is rejected, then the server shall not update the board state and broadcast the current board state to all relevant parties.

3.1.11. Server shall keep track of set and game scores. If a set is won by a player, all the relevant parties shall be informed about the result. If a game is won, all the relevant parties shall be informed about the result as well as closing the sessions of all the relevant parties.

3.1.12. Server should understand when a backgammon game is over..

3.2. Backgammon Server Non-Functional Requirements

3.2.1. The server shall be available through the BC-BS protocol as defined in SWE544_RFC_2.

3.2.2. The server shall be written in the Python programming language.

4. BACKGAMMON CLIENT REQUIREMENTS

4.1. Backgammon Client Functional Requirements

4.1.1. The client shall use the BC-BS protocol. TCP Port 10001 shall be used for the outgoing traffic.

4.1.2. The users shall be able to log in to the Backgammon gaming server by providing an IP address and port number of the server and a username.

4.1.3. The client shall provide an interface to play a backgammon game.

4.1.4. The client shall provide an interface to watch a backgammon game.

4.1.5. The client shall provide an interface to leave the system while waiting an opponent or watching a game.

4.1.6. The client shall provide an interface to throw dice.

4.1.7. The client shall provide an interface to make a move.

4.1.8. The client shall provide a REJECT capability in case his/her opponent makes a wrong move.

4.1.9. The client should provide an interface to see messages sent by the server.

4.2. Backgammon Client Non-Functional Requirements

4.2.1. The client shall be available through the BC-BS protocol as defined in SWE544_RFC_2.

4.2.2. The client shall be written in the Python programming language.

4.2.3. Username shall consists of English letters, digits, and underscores. It's case sensitive and although a username shall only start with an English letter, it may end with a letter or a digit or an underscore.

References

1. SWECHAT Application: System-Wide Requirements:
https://gist.github.com/canerturkmen/8115141

2. SWE544 lectures