Construction of Chinese Chess Server

ICTU Student 26
Thai Nguyen University of Information and Communication Technology
Thai Nguyen University, Thai Nguyen, Vietnam

***Abstract*—This thesis documents the design and implementation of a server for an online Chinese chess game using C# and the Microsoft .NET platform. It introduces the client-server model and socket communication, then describes how the server accepts connections, reads client messages, and returns responses. The implemented routines cover user registration, login, logout, message forwarding, game play, and restarting a session. The source also includes board initialization and move-validation code for different chess pieces. These components show the intended flow from a client request to a server action and an updated game state. The thesis provides implementation descriptions and code excerpts, but does not report a controlled performance experiment, a security assessment, or a systematic test with multiple users. Its evidence therefore supports the listed functions and program structure, while reliability under wider deployment remains unmeasured.**

***Index Terms*—Chinese Chess, Client-Server, C#, TCP/IP, Game Server, Programming**

I. Introduction

The project addresses a networked version of Chinese chess in which players connect to a central server. The thesis first surveys the Microsoft .NET environment, the C# language, and the client-server communication pattern. It then motivates a server that can receive player requests and coordinate shared game state. Rather than treating the chess board as a stand-alone local interface, the design places connection management and message exchange on the server side. The source discusses the role of network addresses and ports in establishing a session. Its stated aim is an educational implementation of the game server. The thesis does not present a formal comparison with other chess platforms or a measured account of user demand. [1]

II. Methods

The server was implemented in C#. Client messages are read through a network stream and interpreted according to a message identifier. Separate routines register a user, check login information, remove a logged-out connection, send a message, start game play, and request a restart. The code stores connection and user information in lookup tables so that a request can be forwarded to the intended peer. Additional routines initialize the board and test whether a proposed move is legal for a specified piece and position. The thesis includes source-code excerpts for reading packet sizes, writing responses, and updating play state. These excerpts describe the implementation; they are not a reproducible benchmark or formal verification of every rule. [1]

Supervisor: Nguyễn Văn Tới, Thai Nguyen University of Information and Communication Technology.

III. Results

The source presents a working program structure with server routines for connecting players and handling the main requests of a Chinese chess session. Registration and login routines return responses to the client; logout removes a stored connection. During play, the server reads a message, identifies the receiving player, and forwards the relevant state. The restart routine supports another game after a session. Board initialization and move-checking functions encode the positions and permitted movements of pieces. These are implementation results documented by code and narrative in the thesis. No throughput, latency, error-rate, or multi-user load measurements are provided, so the document does not establish quantitative performance or operational reliability. [1]

IV. Discussion

A centralized server simplifies coordination between two clients because connection records and game messages can be handled in one place. The code excerpts also show how much of the behavior depends on correct parsing of message identifiers, packet lengths, and board positions. The thesis describes this structure but does not include a systematic test matrix for the chess rules or network failure cases. It does not measure the effect of simultaneous users, unexpected disconnections, or malformed requests. Those omissions limit any claim that the implementation is ready for public service. Further work could test rule coverage, session recovery, and input validation against the functions already present in the code. [1]

V. Conclusion

The thesis provides an educational C# implementation of a Chinese chess server. The documented code handles user connections and the main session actions, including registration, login, logout, play, message forwarding, and restart. It also contains routines for initializing the board and checking moves. The reported result is the existence of these designed and implemented components, rather than a measured improvement over another system. Before deployment beyond the described project setting, the server would need documented tests of rules, error handling, security, and behavior under concurrent connections. [1]

References

[1] ICTU Student 26, ”Construction of Chinese Chess Server,” undergraduate thesis (original in Vietnamese), Thai Nguyen University of Information and Communication Technology, n.d.

[2] Đinh Mạnh Tường, Trí tuệ nhân tạo, Nhà xuất bản Khoa học và Kỹ thuật, Hà Nội, 2002

[3] Đỗ Trung Tuấn, Trí tuệ nhân tạo, Nhà xuất bản Giáo dục 1998

[4] Phạm Hữu Khang, Kỹ thuật lập trình ứng dụng C#.net, Nhà xuất bản Lao động - Xã hội 2003

[5] Writing the Server side of a Socket (The JavaTM Tutorials)

[6] http://www.codeproject.com