

**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG KHOA CÔNG NGHỆ THÔNG TIN**

----------

**BÁO CÁO BÀI TẬP LỚN**

**LẬP TRÌNH MẠNG**

**GAME CỜ CARO ONLINE**

**Sinh viên thực hiện:**

B21DCCN172 – Dương Thái Bình

**B21DCCN393 – Phạm Việt Hoàng**

B21DCCN489 – Vũ Hữu Hoài Linh

B21DCCN525 – Nguyễn Chí Minh

**Nhóm lớp: 02**

**Nhóm Bài tập lớn: 05**

**Giảng viên hướng dẫn: Nguyễn Hoàng Anh**

Thành viên và phân công nhiệm vụ

|  |  |
| --- | --- |
| Nguyễn Chí Minh | Lên kế hoạch và phân công nhiệm vụ  Thiết kế giao diện chính Code và xử lý chức năng chơi game  Logic trò chơi |
| Vũ Hữu Hoài Linh | Code và xử lý phần: Đăng nhập, đăng ký, đăng xuất  Xem danh sách trạng thái hoạt động  Xem thông tin hồ sơ chi tiết các  người chơi  Mời chơi, chấp nhận lời mời , vào phòng  Code các chức năng xử lý Logic trò chơi  Thiết kế CSDL cho toàn hệ thống. |
| **Phạm Việt Hoàng** | **Logic trò chơi**  **Danh sách bạn bè**  **Code và xử lý chức năng tạo phòng** |
| Dương Thái Bình | Tìm hiểu cách kết nối nhiều thiết bị (Radmin VPN) Logic trò chơi  Xử lý kết bạn và nhận lời kết bạn  Code và xử lý chức năng xem bảng xếp hạng |

**I. Mở đầu**

**1. Giới thiệu game**

* Game Cờ Caro là một trò chơi trực tuyến, nơi người chơi đăng nhập từ máy client và tham gia vào các trận đấu thông qua server. Hệ thống bao gồm một server quản lý các trận đấu và lưu trữ thông tin người chơi. Sau khi đăng nhập, người chơi có thể mời người khác tham gia trận đấu.Trong mỗi trận đấu,server sẽ cung cấp một màn hình 15x15 các ô vuông và người chơi cần phải cố gắng để có thể điền được 5 quân liên tiếp.Luật chơi tuy rất đơn giản nhưng lại đòi hỏi khả năng tính toán và tư duy của cả 2 người chơi. Điểm số và xếp hạng của các người chơi sẽ được cập nhật và lưu trữ trên server.
* Công nghệ sử dụng:
  + Ngôn ngữ lập trình Java
  + UI:Java swing
  + TCP/IP
  + MVC cải tiến
  + JDBC
* Database: MySQL

**2.Phân tích yêu cầu hệ thống**

**a. Yêu cầu với Server**

* **Quản lý người chơi**:
  + Lưu trữ thông tin người chơi, bao gồm tên, tổng số điểm hiện tại, và trạng thái (online, offline, đang chơi).
  + Cung cấp khả năng đăng nhập và xác thực người chơi.
  + Phát hiện và quản lý trạng thái kết nối của người chơi (online/offline).
* **Quản lý trận đấu**:
  + Khởi tạo và theo dõi trạng thái của từng trận đấu, bao gồm:
    - Danh sách người chơi tham gia.
    - Trạng thái hiện tại của ván cờ (trong quá trình chơi, hoà, hoặc có người thắng).
    - Kiểm tra điều kiện chiến thắng: người chơi có đạt được 5 quân liên tiếp trên hàng, cột hoặc đường chéo không.
    - Quản lý thời gian cho mỗi lượt đi (30 giây mỗi lượt).
    - Quyết định người chiến thắng hoặc hòa sau khi trận đấu kết thúc.
    - Lưu trữ kết quả trận đấu (thắng/thua/hòa) và cập nhật điểm số người chơi.
* **Tính năng gửi lời mời chơi**:
  + Cho phép người chơi gửi lời mời chơi đến những người khác từ danh sách trực tuyến.
  + Khi một người chơi chấp nhận lời mời, server sẽ thông báo và bắt đầu trận đấu.
* **Lưu trữ kết quả và xếp hạng**:
  + Cập nhật điểm số và số trận thắng của mỗi người chơi.
  + Cung cấp giao diện cho phép người chơi xem bảng xếp hạng với thông tin tổng điểm và số trận thắng.
  + Lưu trữ kết quả trận đấu và thống kê người chơi.
* **Xử lý kết nối**:
  + Quản lý kết nối và ngắt kết nối của client.
  + Đảm bảo rằng server có thể xử lý nhiều client đồng thời mà không bị gián đoạn.

**b. Yêu cầu với Client**

* **Đăng nhập và giao diện người chơi**:
  + Cung cấp giao diện đăng nhập cho người chơi để xác thực tài khoản của mình.
  + Hiển thị danh sách người chơi trực tuyến, cho phép người chơi xem thông tin về tên, điểm số và trạng thái (online, offline, đang chơi).
* **Bắt đầu trận đấu**:
  + Hiển thị danh sách người chơi trực tuyến cho phép người chơi gửi lời mời chơi đến người khác.
  + Khi nhận được lời mời chơi, người chơi có thể chấp nhận hoặc từ chối lời mời.
  + Sau khi trận đấu bắt đầu, giao diện sẽ chuyển sang màn hình chơi cờ.
* **Giao diện chơi Cờ Caro**:
  + Hiển thị bàn cờ 15x15.
  + Cho phép người chơi chọn ô để đặt quân (X hoặc O) trong trận đấu.
  + Hiển thị nút **“Gửi”** để gửi nước đi và nút **“Thoát”** để thoát khỏi trận đấu.
  + Sau mỗi lượt đi, cập nhật giao diện để hiển thị nước đi của cả hai người chơi.
* **Quản lý thời gian và thông báo**:
  + Hiển thị thời gian đếm ngược 30 giây cho mỗi lượt đi.
  + Thông báo kết quả trận đấu khi có người chiến thắng hoặc hòa.
  + Cung cấp thông báo khi người chơi bị xử thua vì không đưa ra nước đi trong thời gian quy định.
* **Thoát trận đấu**:
  + Cung cấp tính năng cho người chơi thoát khỏi trận đấu bất cứ lúc nào. Tuy nhiên, nếu thoát trước khi kết thúc trận đấu, người chơi sẽ không nhận được điểm.
* **Xem bảng xếp hạng**:
  + Cho phép người chơi xem bảng xếp hạng của tất cả người chơi, bao gồm điểm số tổng và số trận thắng.

**II. Thiết kế**

**1. Kiến trúc tổng quan**

* Server: Xử lý data và gửi lại client để hiện thị kết quả
* Client: Tiếp nhận các thao tác của người dùng
* ClientHandle: Xử lý các thao tác của người dùng và gửi về server xử lý data

**A diagram of a server and client

Description automatically generated**

**2.Sơ đồ khối và các chức năng của client và server**

A diagram of a server

Description automatically generated

**3. Usecase tổng quan**

**A screenshot of a computer

Description automatically generated**

**4. Use case chi tiết**

**A diagram of a game

Description automatically generated**

**5. Biểu đồ lớp**

**Client:**

A diagram of a computer

Description automatically generated

**Server:**

A diagram of a computer program

Description automatically generated with medium confidence

**6. Biểu đồ tuần tự**

Sequence diagram play game:

A diagram of a project

Description automatically generated

**7. Sơ đồ thực thể quan hệ**

**A screenshot of a computer

Description automatically generated**

**III. Kết quả**

**1. Kiến trúc ứng dụng**

Kiến trúc ứng dụng cho dự án Game ghi nhớ ô nhớ có thể được chia thành ba lớp chính:

**Giao diện người dùng (UI Layer)**

* **Công nghệ**: Java Swing.
* **Chức năng**:
  + Màn hình **đăng nhập**, **danh sách người chơi** (hiển thị tên, điểm, trạng thái).
  + Bàn cờ **15x15** với nút **Gửi** và **Thoát**.
  + Hiển thị thời gian 30s/lượt, người chơi hiện tại (X/O).
  + Màn hình **xếp hạng** (theo điểm).

**Logic trò chơi (Game Logic Layer)**

* **Chức năng**:
  + Quản lý bàn cờ (15x15), cập nhật trạng thái nước đi.
  + Kiểm tra thắng/thua/hòa (5 quân liên tiếp trên hàng, cột, chéo).
  + Xử lý điểm: Thắng (+2), hòa (+1), thua (0).
  + Đếm thời gian lượt chơi, tự xử thua nếu quá 30s không đi.

**Kết nối mạng (Networking Layer)**

* **Công nghệ**: TCP Sockets.
* **Chức năng**:
  + Kết nối **Client-Server**.
  + Đồng bộ nước đi và trạng thái bàn cờ giữa hai người chơi.
  + Xử lý lời mời chơi, cập nhật danh sách người chơi.
  + Lưu kết quả và điểm số vào cơ sở dữ liệu (MySQL).

**2. Cài đặt triển khai ứng dụng và kết quả**

**2.1 Chức năng đăng nhập**

a, Mô tả

Người chơi vào hệ thống giao diện đăng nhập hiện ra người chơi nhập username, password và ấn đăng nhập. Hệ thống tiến hành kiểm tra thống tin nếu thông tin đúng thì giao diện chính người chơi hiển thị, nếu sai hiện thông báo sai thông tin đăng nhập

b, Giao diện

A screenshot of a computer

Description automatically generated

**2.2 Chức năng đăng kí**

a, Mô tả

Người chơi chưa có tài khoản ấn vào nút đăng kí trên màn hình. Hệ thống hiện giao diện đăng ký gồm các thông tin(username, password, nickname) và nút đăng ký. Người chơi tiến hành điền các thông tin và ấn vào nút đăng ký. Hệ thống lưu thông tin về người chơi vào csdl.

b, Giao diện

A screenshot of a cartoon

Description automatically generated

**2.3 Màn hình chính người chơi khi đăng nhập**

a, Mô tả

Sau khi đăng nhập thành công hệ thống hiển thị giao diện chính của người chơi

b, Giao diện

A screenshot of a chat

Description automatically generated

**2.4 Chức năng kết bạn**

a, mô tả

* Là chức năng kết bạn giữa 2 người chơi

b, giao diện

**Screens screenshot of a computer

Description automatically generated**

ấn kết bạn ở ô thông tin đối thủ, người kia sẽ nhận được thông báo kết bạn

Sau khi trở thành bạn bè, danh sách bạn bè được cập nhật.

**2.5 Chức năng xem BXH**

a,mô tả: là chức năng xem bảng xếp hạng người chơi

Giao diện

A screenshot of a computer

Description automatically generated

**2.6 Logic trận đấu**

a, Mô tả

**Cờ Caro** là một trò chơi bàn cờ dành cho hai người, thường được chơi trên lưới kẻ ô vuông (15x15 hoặc lớn hơn).

**Luật chơi:**

1. **Mục tiêu**: Người chơi đầu tiên sắp xếp được **5 quân liên tiếp** (theo hàng ngang, hàng dọc, hoặc đường chéo) sẽ chiến thắng.
2. **Người chơi**:
   * Một người sử dụng quân **X**.
   * Người kia sử dụng quân **O**.
3. **Luân phiên đánh**: Hai người chơi lần lượt đặt quân vào các ô trống trên bàn cờ.
4. **Kết quả**:
   * **Thắng**: Đạt 5 quân liên tiếp.
   * **Hòa**: Hết nước đi mà không ai thắng.

b, Giao diện

**A screenshot of a computer

Description automatically generated**

**IV. Danh sách bạn bè**

Mô tả Chức năng

Chức năng Danh sách bạn bè trong game Cờ Caro Online là một phần quan trọng của giao diện người dùng. Nó cho phép người chơi:

1. Xem danh sách bạn bè cùng trạng thái hiện tại (online, offline, đang chơi).
2. Gửi yêu cầu thách đấu cho bạn bè trực tuyến không trong trận đấu.
3. Cập nhật danh sách bạn bè theo thời gian thực để đồng bộ với trạng thái của server.

Giao diện Người dùng

* Giao diện bảng hiển thị danh sách bạn bè:
  + Cột 1: ID của bạn bè.
  + Cột 2: Nickname của bạn bè.
  + Cột 3: Icon trạng thái:
    - Online: Hiển thị icon "kiếm sẵn sàng".
    - Đang chơi: Hiển thị icon "hai thanh kiếm".
    - Offline: Hiển thị icon "offline".
  + Hành động: Nhấn vào một dòng để gửi yêu cầu thách đấu.
* Giao diện được xây dựng bằng Java Swing với bảng JTable để hiển thị dữ liệu danh sách bạn bè.

Cách Hoạt Động

1. Khởi tạo giao diện:
   * Khi mở giao diện, bảng sẽ hiển thị tiêu đề "Danh sách bạn bè" và tự động gửi yêu cầu đến server để lấy dữ liệu danh sách bạn bè.
2. Cập nhật danh sách bạn bè:
   * Sử dụng phương thức requestUpdate() để gửi lệnh "view-friend-list," đến server qua socket.
   * Server phản hồi danh sách bạn bè dạng JSON, được chuyển đổi thành đối tượng User và cập nhật bảng hiển thị.
3. Xử lý hành động thách đấu:
   * Khi người chơi nhấn vào một dòng trong bảng:
     + Kiểm tra trạng thái của bạn bè (online, offline, đang chơi).
     + Nếu bạn bè đủ điều kiện, hiển thị hộp thoại xác nhận thách đấu.
     + Gửi yêu cầu thách đấu đến server với lệnh "duel-request,<ID>".
4. Đồng bộ trạng thái bạn bè theo thời gian thực:
   * Chức năng sử dụng luồng riêng (Thread) để liên tục cập nhật danh sách bạn bè từ server mỗi 500ms nếu giao diện đang hiển thị.

Cách Thực Hiện

1. Xây dựng giao diện (UI):

* Dùng JPanel, JTable, và JScrollPane để tạo giao diện.
* Dữ liệu danh sách bạn bè được hiển thị trong bảng JTable, với mỗi dòng gồm 3 cột (ID, Nickname, Icon trạng thái).

2. Xử lý cập nhật trạng thái bạn bè:

* Hàm requestUpdate() gửi lệnh "view-friend-list," đến server qua socket để lấy danh sách bạn bè.
* Server trả về danh sách bạn bè, được xử lý bằng hàm updateFriendList().

3. Kiểm tra trạng thái và thách đấu:

* Khi nhấn vào một dòng trong bảng:
  + Kiểm tra điều kiện:
    - Nếu bạn bè offline: Hiển thị thông báo lỗi.
    - Nếu bạn bè đang chơi: Hiển thị thông báo lỗi.
    - Nếu bạn bè đủ điều kiện: Hiển thị hộp thoại xác nhận thách đấu.
  + Gửi yêu cầu thách đấu đến server.

Cập nhật danh sách bạn bè:



Xử lý nhấn vào danh sách bạn bè:



Kết Quả

* Giao diện danh sách bạn bè hoạt động mượt mà, cập nhật trạng thái bạn bè theo thời gian thực.
* Hỗ trợ người chơi gửi lời mời thách đấu nhanh chóng và trực quan.

Hạn Chế

* Việc cập nhật liên tục mỗi 500ms có thể gây tải lớn lên server nếu số lượng người dùng đông.
* Chưa hỗ trợ tìm kiếm hoặc phân loại danh sách bạn bè.
* Hiện tại không có tính năng thông báo trạng thái bạn bè thay đổi (chỉ cập nhật khi giao diện mở).

Đề Xuất Cải Tiến

1. Tối ưu hóa cập nhật trạng thái bạn bè:
   * Sử dụng cơ chế WebSocket hoặc thông báo sự kiện từ server để giảm tải cập nhật liên tục.
2. Bổ sung tính năng:
   * Tìm kiếm bạn bè.
   * Thông báo trạng thái bạn bè thay đổi ngay cả khi giao diện không mở.
3. Tăng cường bảo mật:
   * Kiểm tra và xác thực yêu cầu thách đấu từ server trước khi hiển thị thông báo.

**V. Kết luận**

Hoàn thiện

* Hoàn thành các chức năng theo đề bài đề ra.
* Chương trình giao diện thân thiện dễ giao tiếp, dễ sử dụng.
* Server có thể quản lý số lượng máy lớn.
* Giao diện đẹp, dễ nhìn.

Hạn chế

* Chưa phân tách dữ liệu thành nhiều bảng để quản lý khoa học hơn.
* Nghiệp vụ dự án còn một vài thiếu sót .
* Code chưa clean.

**VI. Tài liệu tham khảo**

**Tài liệu tham khảo**

* [**https://studyandshare.wordpress.com/category/java-programming-**](https://studyandshare.wordpress.com/category/java-programming-language/java-swing/)[**language/java-swing/**](https://studyandshare.wordpress.com/category/java-programming-language/java-swing/)
* [**https://niithanoi.edu.vn/huong-dan-thao-tac-voi-jtable-lap-trinh-java-swing.html**](https://niithanoi.edu.vn/huong-dan-thao-tac-voi-jtable-lap-trinh-java-swing.html)
* [**https://studyandshare.wordpress.com/category/java-programming-language/lap-**](https://studyandshare.wordpress.com/category/java-programming-language/lap-trinh-mang/)[**trinh-mang/**](https://studyandshare.wordpress.com/category/java-programming-language/lap-trinh-mang/)
* [**https://stackoverflow.com/a/4685606**](https://stackoverflow.com/a/4685606)
* [**https://stackoverflow.com/questions/201287/how-do-i-get-which-jradiobutton-**](https://stackoverflow.com/questions/201287/how-do-i-get-which-jradiobutton-is-selected-from-a-buttongroup)[**is-selected-from-a-buttongroup**](https://stackoverflow.com/questions/201287/how-do-i-get-which-jradiobutton-is-selected-from-a-buttongroup)
* [**https://shareprogramming.net/tag/java-socket/**](https://shareprogramming.net/tag/java-socket/)

[**https://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading**](https://gpcoder.com/3484-lap-trinh-da-luong-trong-java-java-multi-threading)**/**