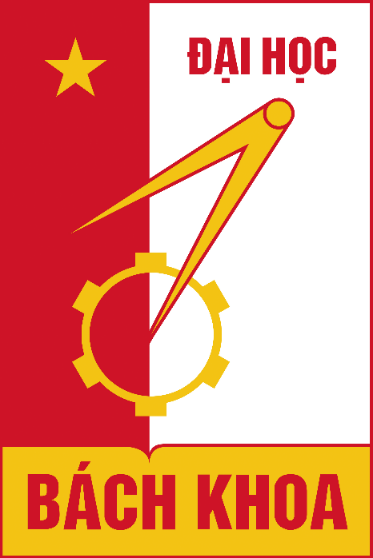
**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

**VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

**🙥🙦✯✯✯🙤🙧**

****

**BÁO CÁO BÀI TẬP LỚN**

**Đề tài: DOTS AND BOX**

**Học phần: *Thực hành Lập trình mạng***

**Giảng viên hướng dẫn: *PGS.TS. Trương Thị Diệu Linh***

**Nhóm: 5**

**Sinh viên thực hiện: *Lê Đức Mạnh - 20176814***

***Mai Quang Huy - 20176785***

***Nguyễn Tiến Anh - 20176681***

***Hà Nội, ngày 09 tháng 01 năm 2020***

**MỤC LỤC**

[**LỜI MỞ ĐẦU 2**](#_Toc61431171)

[**CHƯƠNG 1. MÔ TẢ ĐỀ TÀI 3**](#_Toc61431172)

[**1. Ý tưởng 3**](#_Toc61431173)

[**2. Luật chơi 3**](#_Toc61431174)

[**CHƯƠNG 2. GIỚI THIỆU CHƯƠNG TRÌNH 4**](#_Toc61431175)

[**1. Giới thiệu 4**](#_Toc61431176)

[**2. Chức năng 5**](#_Toc61431177)

[**3. Kiến trúc chương trình 5**](#_Toc61431178)

[**4. Class Diagram 6**](#_Toc61431179)

[**4.1. server.Player Class 6**](#_Toc61431180)

[**4.2. server.PlayerOutput Class 6**](#_Toc61431181)

[**4.3. client.Client Class 6**](#_Toc61431182)

[**4.4. client.Controller Class 7**](#_Toc61431183)

[**4.5. game.Board Class 7**](#_Toc61431184)

[**4.6. game.Edge Class 7**](#_Toc61431185)

[**CHƯƠNG 3. THIẾT KẾ GIAO THỨC TẦNG ỨNG DỤNG 8**](#_Toc61431186)

[**1. Cấu trúc các gói tin 8**](#_Toc61431187)

[**1.1. Các gói tin từ Server 8**](#_Toc61431188)

[**1.2. Các gói tin từ Client 9**](#_Toc61431189)

[**2. Nhóm chức năng liên quan đến xác thực người dùng 10**](#_Toc61431190)

[**2.1. Register 10**](#_Toc61431191)

[**2.2. Login 10**](#_Toc61431192)

[**2.3. Logout 10**](#_Toc61431193)

[**3. Nhóm chức năng liên quan đến phòng chơi 11**](#_Toc61431194)

[**3.1. Danh sách phòng 11**](#_Toc61431195)

[**3.2. Tạo phòng 11**](#_Toc61431196)

[**3.3. Vào phòng 11**](#_Toc61431197)

[**3.4. Rời phòng 11**](#_Toc61431198)

[**3.5. Bắt đầu game 11**](#_Toc61431199)

[**3.6. Chat trong phòng 12**](#_Toc61431200)

[**4. Nhóm chức năng liên quan đến phần chơi 12**](#_Toc61431201)

[**4.1. Thực hiện nước đi 12**](#_Toc61431202)

[**4.2. Trò chơi kết thúc 12**](#_Toc61431203)

[**CHƯƠNG 4. DEMO CHƯƠNG TRÌNH 13**](#_Toc61431204)

[**1. GitHub 13**](#_Toc61431205)

[**2. Demo chương trình 13**](#_Toc61431206)

[**CHƯƠNG 5. KẾT LUẬN 18**](#_Toc61431207)

[**1. Kết quả đạt được 18**](#_Toc61431208)

[**2. Dự định tương lai 18**](#_Toc61431209)

[**TÀI LIỆU THAM KHẢO 18**](#_Toc61431210)

# 

# LỜI MỞ ĐẦU

**Bản báo cáo trình bày quá trình thiết kế, triển khai bài tập lớn đề tài Dots and box áp dụng lập trình socket và kiến thức mạng máy tính.**

**Bản báo cáo được xây dựng trên những kiến thức chúng em được dạy cũng như tự tìm hiểu được. Vì thế trong quá trình làm, nghiên cứu chắc chắn không tránh khỏi một số thiếu sót, mong cô góp ý thêm nhằm giúp cho báo cáo được hoàn thiện hơn nữa.**

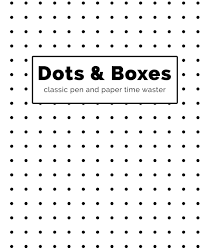
**Qua đây, nhóm chúng em cũng xin được gửi lời cảm ơn chân thành tới cô *Trương Thị Diệu Linh* (giảng viên bộ môn) đã giảng dạy, giải đáp các thắc mắc trong giờ học, cung cấp tài liệu tham khảo để chúng em có thể hoàn thiện bài tập và bản báo cáo này.**

***Hà Nội, tháng 01 năm 2020***

# CHƯƠNG 1. MÔ TẢ ĐỀ TÀI

## Ý tưởng

Game Dots and Box là một trò chơi bằng bút chì cho hai người chơi (đôi khi nhiều hơn). Nó được xuất bản lần đầu tiên vào thế kỷ 19 bởi nhà toán học người Pháp [Édouard Lucas](https://en.wikipedia.org/wiki/%C3%89douard_Lucas) , người gọi nó là la pipopipette .

****

Dots and box là trò chơi trực tuyến. Người chơi có thể đăng nhập vào hệ thống và chơi với bạn bè. Trong đó, mỗi hiệp đấu chỉ dành cho hai người chơi (1vs1) với nhau.

## Luật chơi

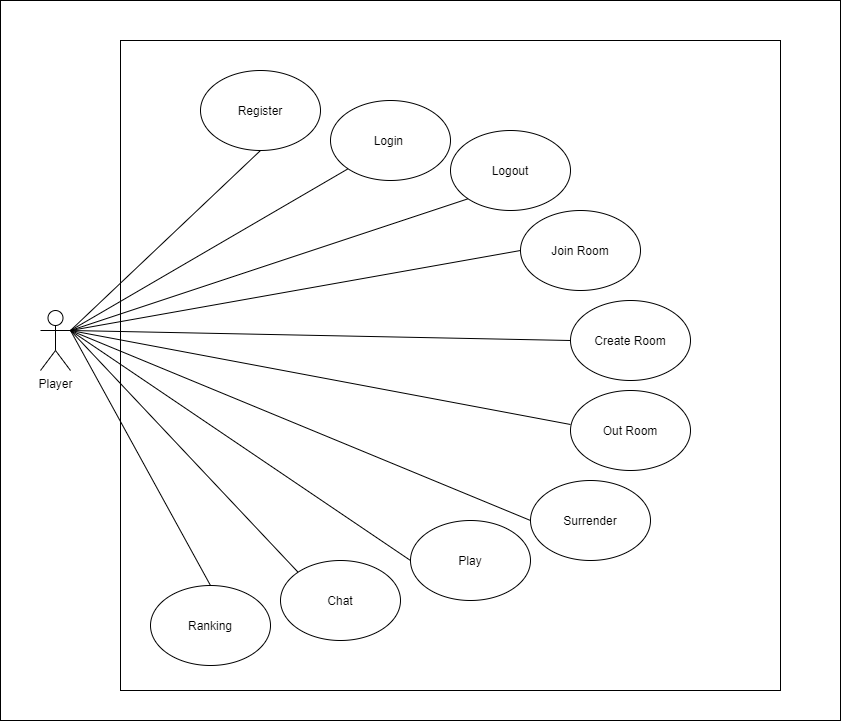
* Số người tham gia mỗi hiệp đấu: 2 người chơi
* Cách chơi: Tấn công theo lượt
* Luật chơi của Dots and Boxes rất đơn giản, 2 người sẽ lần lượt nối các điểm có trên bàn chơi tạo thành các ô vuông. Người chiến thắng sẽ là người có nhiều ô vuông hơn. Trận đấu kết thúc khi tất cả các ô vuông đã được điền kín hoặc có người chơi nhận thua.
* Bàn chơi của Dots and Boxes có nhiều chấm được xếp thành hàng. Hai người chơi lần lượt nối 2 điểm liền kề, có thể theo chiều dọc hoặc chiều ngang và dần dần tạo thành các ô vuông. Người chơi nào vẽ được cạnh cuối cùng của 1 ô vuông được tính là "người khởi tạo" ô vuông đó và sẽ phải tiếp tục vẽ 1 đường tiếp theo. Khi tất cả các ô vuông trên bàn chơi đã được điền kín thì game sẽ tính người chiến thắng dựa trên số ô vuông mà 2 người chơi tạo được. Ai tạo được nhiều ô vuông hơn sẽ là người chiến thắng.

# CHƯƠNG 2. GIỚI THIỆU CHƯƠNG TRÌNH

## Giới thiệu

* Ứng dụng trò chơi Dots and Box có hỗ trợ chơi trực tuyến giữa nhiều người chơi với nhau. Ứng dụng chạy trên hệ điều hành Ubuntu, Window, MacOS. Ứng dụng được lập trình bằng ngôn ngữ Java.
* Người sử dụng có thể đăng nhập bằng username với password đã được đăng kí trước. Nếu chưa có tài khoản người dùng có thể đăng kí một tài khoản mới, mỗi người chơi có một username duy nhất. Người sử dụng có thể tạo phòng/bàn chơi mới và đợi đối thủ. Người chơi khác kết nối đến server có thể lựa chọn một trong số các phòng do người dùng khác tạo để vào chơi hoặc tự tạo phòng mới.
* Trong phòng chơi sẽ hiển thị tên của hai người chơi, hai người chơi có thể chat với nhau. Số người tối đa một phòng là 2, nếu một phòng đang có 2 người chơi thì người chơi thứ 3 không thể vào phòng. Người tạo phòng sẽ là chủ phòng. Chủ phòng sẽ có quyền chọn kích thước bảng chơi. Nếu chủ phòng thoát phòng thì phòng đó cũng bị xóa.
* Trò chơi chỉ bắt đầu khi trong phòng có hai người và chủ phòng đã chọn kích thước bảng và ấn start. Sau mỗi một ván chơi thì phải chọn lại kích thước bảng. Trong quá trình chơi người chơi cũng có thể chọn chế độ đầu hàng, khi đó ván đấu sẽ kết thúc. Mỗi ván game kết thúc thì số điểm của người chơi sẽ được cập nhật lại.
* Ngoài ra khi người dùng sẽ xem được số điểm của mình đạt được cũng như xem được top 5 người chơi có rank cao nhất. Nếu là người chơi mới thì số điểm là 0.

## Chức năng

****

***Hình 1: Biểu đồ use case chức năng***

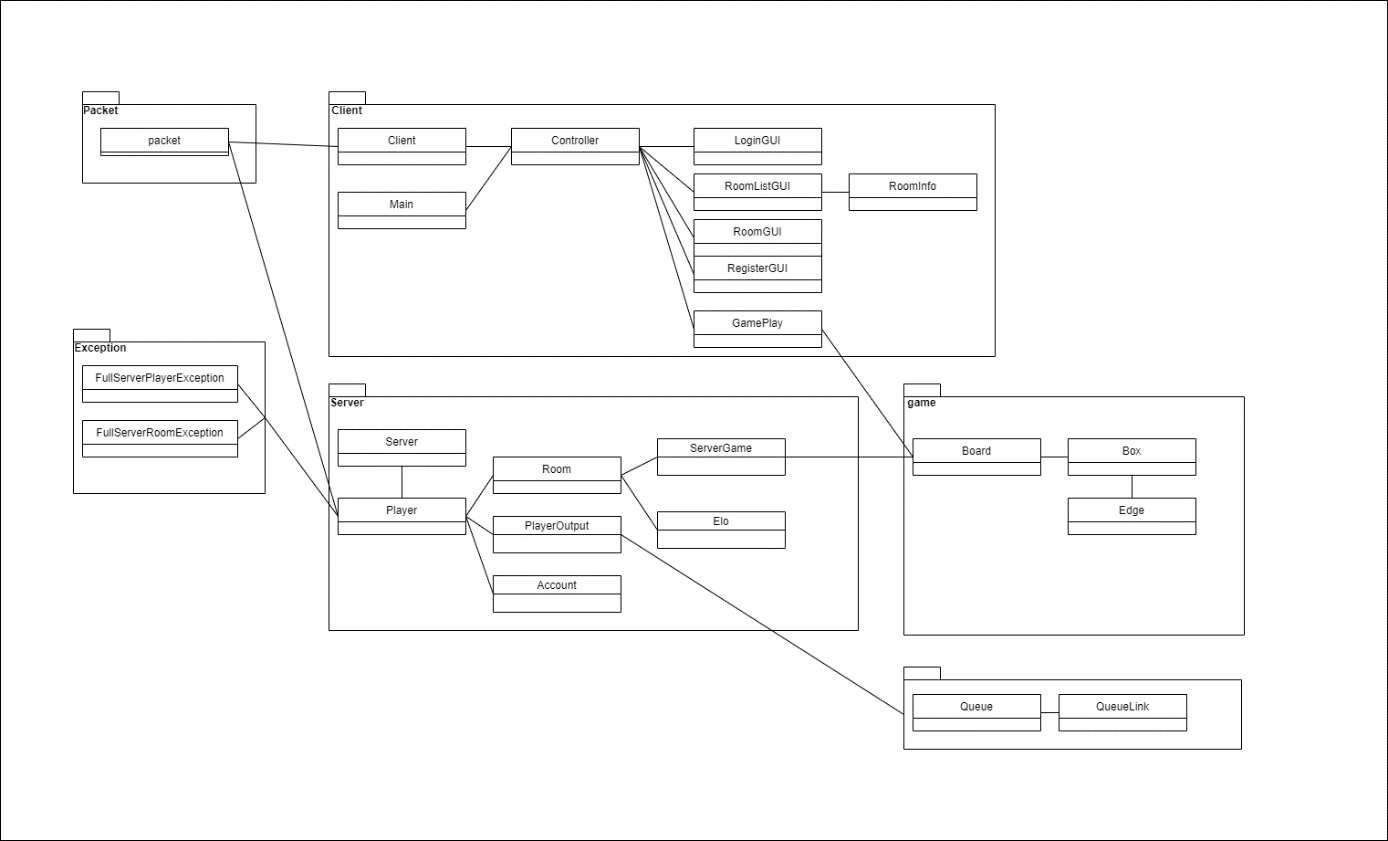
**Ứng dụng có các chức năng sau:**

* Nhóm chức năng liên quan đến xác thực người dùng:
  + Login
  + Logout
  + Register
* Nhóm chức năng liên quan đến phòng chơi:
  + Create Room
  + Join Room
  + Chat
  + Play
  + Out Room
  + Surrender

## Kiến trúc chương trình

* Network model: Client - Server
  + Vai trò của Client: là một người dùng tham gia chơi ứng với một máy tính. Người chơi tương tác, gửi yêu cầu trên client, client gửi thông tin tới server và xử lý thông tin nhận được từ server.
  + Vai trò của Server: là trung tâm xử lý thông tin nhận từ các client (người dùng) và gửi trả kết quả xử lý về client tương ứng. Server cho phép nhiều người chơi tham gia cùng 1 lúc tại các phòng khác nhau.
* Server sockets: Concurrent single TCP server.

## Class Diagram

****

**Các Class quan trọng**

### server.Player Class

* Có chức năng tiếp nhận và xử lý các gói tin từ Client.
* Được kế thừa từ lớp Thread. Tức là đây là một thread.
* Có một số lượng tối đa các đối tượng Player cùng lúc.

### server.PlayerOutput Class

* Cũng được kế thừa từ lớp Thread.
* Có chức năng gửi các gói tin đến cho Client.
* Do đối tượng Player này có thể yêu cầu đối tượng Player khác gửi gói tin (Ví dụ như khi vào phòng, một đối tượng Player sẽ yêu cầu tất cả các đối tượng Player khác gửi gói tin thông báo người chơi vừa vào phòng) nên các gói tin được gửi đi sẽ được thêm vào một Queue. Class PlayerOutput sẽ chạy trên một luồng khác để gửi lần lượt các gói tin từ Queue đó di.

### client.Client Class

* Có chức năng tiếp nhận và xử lý các gói tin từ Server cũng như gửi các gói tin đến Server.
* Cũng được chạy trên một luồng riếng biệt.
* Tùy vào gói tin nhận được sẽ yêu cầu lớp Controller xử lý tương ứng.

### client.Controller Class

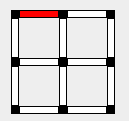
* Có chức năng xử lý giao diện bên phía Client.
* Tùy thuộc vào thao tác của người dùng sẽ yêu cầu Client gửi các gói tin tương ứng đến Server.

### game.Board Class

* Có chức năng xử lý logic trong Game.
* Một Board sẽ quản lý nhiều đối tượng thuộc lớp Edge và Box.

### game.Edge Class

* Được xác định bởi 3 thông số x, y, isHorizontal.
* Giải thích: x, y là tọa độ trên Board, isHorizontal thể hiện cạnh là cạnh ngang hay dọc.
* Ví dụ: Cạnh dưới là cạnh ngang nên có isHorizontal là True. Tọa độ x, y được xác định theo điểm đen bên trái (nếu cạnh dọc sẽ là điểm đen bên trên).

****

# CHƯƠNG 3. THIẾT KẾ GIAO THỨC TẦNG ỨNG DỤNG

## Cấu trúc các gói tin

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Packet\_header** | **DATA\_1** | **DATA\_2** | **...** | **DATA\_N** |

* Packet\_header : tên gói tin
* DATA\_I: Dữ liệu mang thông tin (username, password,...)

### Các gói tin từ Server

|  |  |  |  |
| --- | --- | --- | --- |
| SP\_YOU\_ARE | id | username.length | username |

Gói tin thông tin người dùng

* id: id của người chơi

|  |  |
| --- | --- |
| SP\_LOGIN | id |

Gói tin thông báo kết quả đăng nhập

* id: id của người chơi. id = -1 tức là đăng nhập không thành công.

|  |  |
| --- | --- |
| SP\_LEAVE | id |

Gói tin thông báo người chơi thoát khỏi server:

* id: id của người chơi thoát

|  |  |  |  |
| --- | --- | --- | --- |
| SP\_MESSAGE | message.length | message | seat |

Gói tin thông tin chat của người chơi trong phòng.

* seat: chỗ của người chơi trong phòng

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| SP\_ROOM\_PLAYER | roomId | seat | playerId | username.length | username |

Gói tin thông việc vào, rời phòng

* roomId: id của phòng
* seat: chỗ trong phòng (0 hoặc 1)
* playerId: id của người chơi vừa vào phòng. Nếu Id = -1 nghĩa là có người chơi rời khỏi phòng.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| SP\_GAME\_MOVE | x | y | isHorizontal | seat |

Gói tin thông tin nước đi của người chơi

* x, y: tọa độ của cạnh trong bàn chơi.
* isHorizontal: cạnh là cạnh ngang hay cạnh dọc

|  |  |
| --- | --- |
| SP\_GAME\_START | boardSize |

Gói tin thông tin thông tin bắt đầu ván chơi

* boardSize: kích thước bàn đấu.

|  |  |
| --- | --- |
| SP\_GAME\_WIN | seat |

Gói tin thông tin kết quả ván đấu

* seat: chỗ của người chơi chiến thắng. seat = -1 trận đấu hòa.

|  |  |
| --- | --- |
| SP\_GET\_MMR | mmr |

Gói tin thông tin điểm của người chơi

* mmr: Điểm số hiện tại của người chơi

|  |  |
| --- | --- |
| SP\_REGISTER | success |

Gói tin thông tin thông báo kết quả đăng kí tài khoản

|  |  |  |  |
| --- | --- | --- | --- |
| SP\_TOP\_RANK | usernames.length | usernames | mmrs |

Gói tin thông tin danh sách người chơi có điểm số cao nhất

|  |  |
| --- | --- |
| SP\_ERROR\_PACKET | errorcode |

Gói tin thông báo lỗi

### Các gói tin từ Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CP\_LOGIN | username.length | username | password.length | password |

Gói tin thông tin đăng nhập

|  |
| --- |
| CP\_QUIT |

Gói tin thông báo người dùng logout

|  |
| --- |
| CP\_ROOM\_LEAVE |

Gói tin thông báo người chơi thoát khỏi phòng

|  |  |
| --- | --- |
| CP\_ROOM\_JOIN | roomId |

Gói tin thông tin người dùng muốn join vào phòng có id là roomId

|  |  |
| --- | --- |
| CP\_GAME\_START | boardSize |

Gói tin thông tin người chơi muốn bắt đâu ván đấu với kích thước bảng là boardSize

|  |  |  |  |
| --- | --- | --- | --- |
| CP\_GAME\_MOVE | x | y | isHorizontal |

Gói tin thông tin nước đi của người chơi

* x, y: tọa độ của cạnh trong bàn chơi.
* isHorizontal: cạnh là cạnh ngang hay cạnh dọc

|  |
| --- |
| CP\_GAME\_SURRENDER |

Gói tin thông báo người chơi đầu hàng

|  |
| --- |
| CP\_ROOM\_OPT |

Gói tin thông báo người chơi tạo phòng

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CP\_REGISTER | username.length | username | password.lenght | password |

Gói tin thông tin người dùng đăng kí tài khoản

|  |
| --- |
| CP\_TOP\_RANK |

Gói tin thông báo người dùng muốn xem danh sách những người có điểm cao nhất

## Nhóm chức năng liên quan đến xác thực người dùng

### Register

* Người dùng nhập thông tin đăng kí gồm username, password và xác nhận password
  + Nếu người dùng nhập phần xác nhận lại password không giống với password ở trên thì sẽ được thông báo lỗi cần nhập lại.
  + Nếu người dùng nhập đúng thì bên Client sẽ gửi gói tin CP\_REGISTER gồm các thông tin về username và password đến phía Server
* Server khi nhận được gói tin CP\_REGISTER sẽ gọi đến hàm hdRegister() để xác thực username và password. Sau đó gửi gói tin SP\_REGISTER gồm thông tin thông báo kết quả là true hoặc false đến phía Client.
* Bên phía Client khi nhận được gói tin SP\_REGISTER.
  + Nếu kết quả là True: thì sẽ chuyển đến màn hình login
  + Nếu kết quả là False: thì sẽ hiển thị thông báo đăng kí không thành công

### Login

* Người dùng nhập thông tin đăng nhập gồm username password. Sau đó Client sẽ gửi gói tin CP\_LOGIN gồm các thông tin về username và password đến phía Server.
* Server khi nhận được gói tin CP\_LOGIN sẽ gọi đến hàm hdLogin() để xác thực tài khoản. Nếu tài khoản đúng thì sẽ gửi gói tin SP\_LOGIN và gói tin SP\_YOU\_ARE về phía Client. Nếu username hoặc password sai thì sẽ gửi gói tin SP\_LOGIN với thông tin -1.
* Bên phía Client khi nhận được gói tin SP\_LOGIN
  + Nếu thông tin gói tin là -1 thì hiển thị thông báo đăng nhập không thành công
  + Khác -1 thì sẽ chuyển đến màn hình danh sách các phòng và hiển thị thông tin người chơi

### Logout

* Ngươi chơi khi nhận nút logout phía Client sẽ gửi gói tin CP\_QUIT đến phía Server và ngắt kết nối với Server.
* Server khi nhận được gói tin CP\_QUIT sẽ ngắt kết nối với Client đó.

## Nhóm chức năng liên quan đến phòng chơi

### Danh sách phòng

Danh sách phòng luôn được hiện thị và được tự động cập nhật thông qua gói tin SP\_ROOM\_PLAYER được gửi từ server. Thông tin về người chơi trong phòng cũng được lưu lại và hiện thị nếu cần thiết.

### Tạo phòng

* Người chơi ấn nút “Create Room” để tạo phòng.
* Khi ấn nút Client sẽ gửi gói tin CP\_ROOM\_OPT đến Server để yêu cầu việc tạo phòng. Nếu việc tạo phòng thành công, server sẽ gửi gói tin SP\_ROOM\_PLAYER đến tất cả các client để thông báo về việc tạo phòng. Client gửi gói tin CP\_ROOM\_OPT thấy mình có cùng id với id trong gói tin SP\_ROOM\_PLAYER sẽ chuyển đổi giao diện sang giao diện phòng chơi.
* Nếu gặp lỗi khi tạo phòng, Server sẽ gửi gói tin SP\_ERROR\_PACKET để báo lỗi với Error Code tương ứng.

### Vào phòng

* Người chơi ấn nút “Join” để vào phòng.
* Khi ấn nút Client sẽ gửi gói tin CP\_ROOM\_JOIN đến Server để yêu cầu việc tham gia phòng. Nếu việc Nếu việc vào phòng thành công, server sẽ gửi gói tin SP\_ROOM\_PLAYER đến tất cả các client để thông báo về việc người chơi vừa gia nhập phòng. Client gửi gói tin CP\_ROOM\_JOIN thấy mình có cùng id với id trong gói tin SP\_ROOM\_PLAYER sẽ chuyển đổi giao diện sang giao diện phòng chơi.
* Nếu gặp lỗi khi vào phòng, Server sẽ gửi gói tin SP\_ERROR\_PACKET để báo lỗi với Error Code tương ứng.

### Rời phòng

* Người chơi ấn nút “Quit Room” để rời phòng.
* Khi ấn nút Client sẽ gửi gói tin CP\_ROOM\_LEAVE đến Server để yêu cầu việc ròi khỏi phòng. Nếu việc thoát phòng thành công, server sẽ gửi gói tin SP\_ROOM\_PLAYER đến tất cả các client để thông báo về việc người chơi vừa rời khỏi phòng. Client gửi gói tin CP\_ROOM\_LEAVE thấy mình có cùng id với id trong gói tin SP\_ROOM\_PLAYER sẽ chuyển đổi giao diện.

### Bắt đầu game

* Chủ phòng có thể thiết lập kích thước bàn chơi bắt đầu game bằng cách ấn nút “Start Game”.
* Khi ấn nút Client sẽ gửi gói tin CP\_GAME\_START đến Server để yêu cầu việc bắt đầu chơi. Nếu việc khởi tạo game mới thành công, server sẽ gửi gói tin SP\_GAME\_START đến tất cả các người chơi trong phòng để thông báo về việc khởi tạo game. Client nhận được gói tin SP\_GAME\_START sẽ chuyển sang giao diện chơi game.

### Chat trong phòng

* Client trong phòng có thể chat với nhau.
* Khi nhập nội dung chat rồi ấn enter, client sẽ gửi yêu cầu CP\_MESSAGE tới server để yêu cầu. Nếu việc Chat thành công, Server sẽ gửi gói tin SP\_MESSAGE tới tất cả client trong phòng để thông báo rằng có người chơi vừa gửi chat. Client nhận được gói tin này sẽ hiện thị Chat tương ứng.
* Chat cũng có thể được sử dụng khi chơi game với cách hoạt động như trên.

## Nhóm chức năng liên quan đến phần chơi

### Thực hiện nước đi

* Client sẽ thực hiện nước đi bằng cách gửi gói tin CP\_GAME\_MOVE tới server.
* Server sẽ kiểm tra tính khả thi của nước đi và nếu nước đi được thực hiện thành công, sẽ gửi gói tin SP\_GAME\_MOVE tới 2 client đang chơi. Client khi nhận được gói tin này sẽ hiện thị nước đi tương ứng.

### Trò chơi kết thúc

* Trò chơi kết thúc khi Server nhận thấy không có nước đi nào có thể được thực hiện nữa. Khi đó, Server sẽ gửi gói tin SP\_GAME\_WIN tới 2 client thông báo về kết quả trò chơi. Nếu điểm của người chơi thay đổi, Server gửi thêm gói tin SP\_GET\_MMR để thông báo về điểm mới của người chơi.
* Client có thể đầu hàng sớm. Bằng cách ấn surrender, trò chơi sẽ ngay lập tức kết thúc với chiến thắng của người chơi còn lại.

# CHƯƠNG 4. DEMO CHƯƠNG TRÌNH

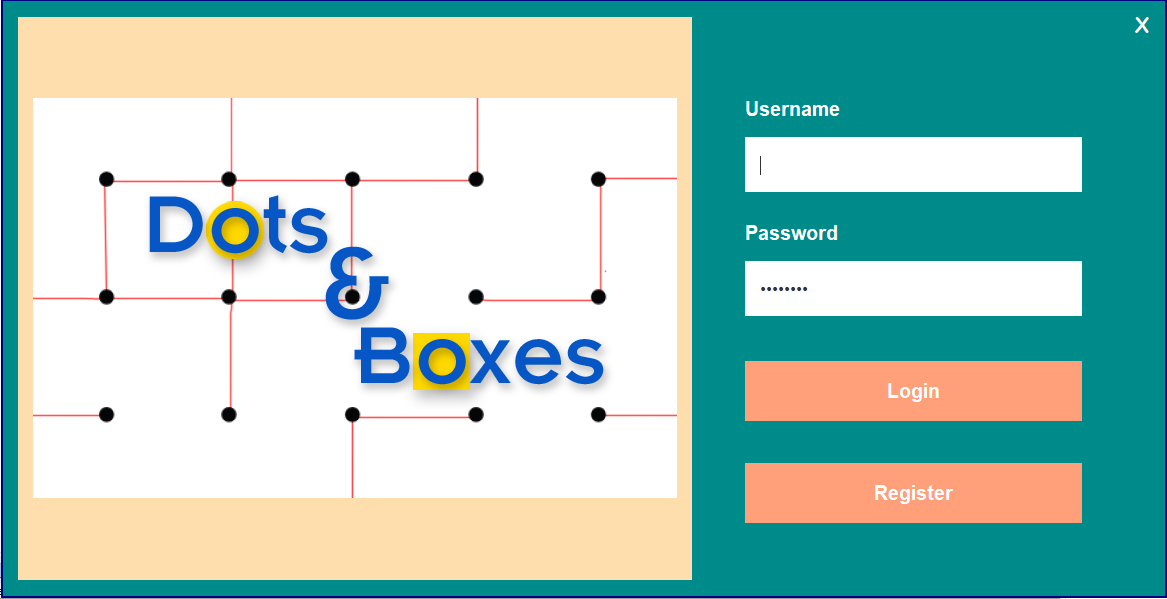
## GitHub

Source code của chương trình có tại:

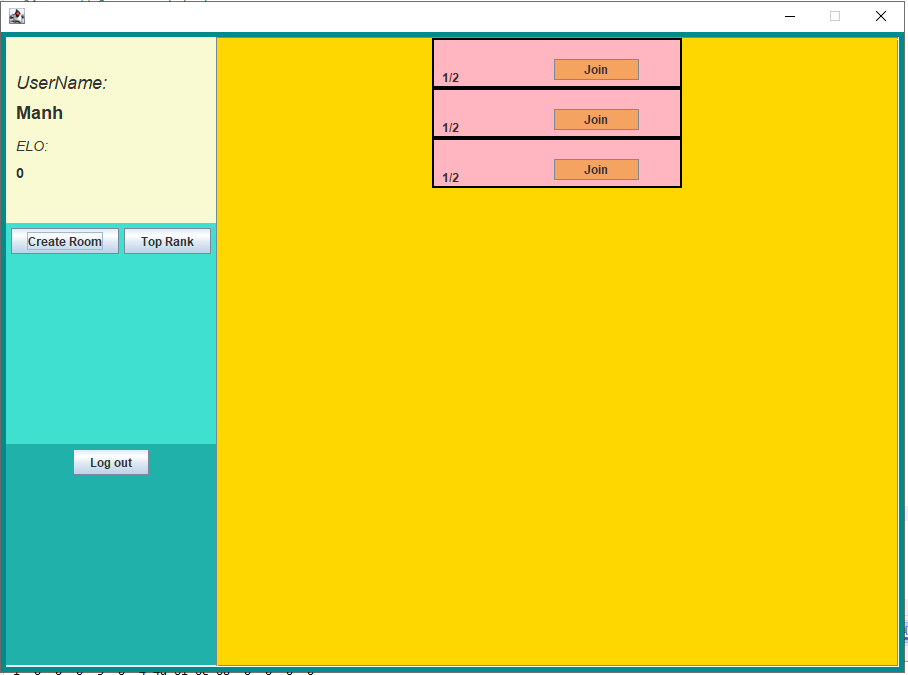
**https://github.com/ManhLeDuc/MultiplayerDotAndBox**

## Demo chương trình

* 1. *Cửa sổ khởi động*

****

* 1. *Cửa sổ Danh sách phòng*

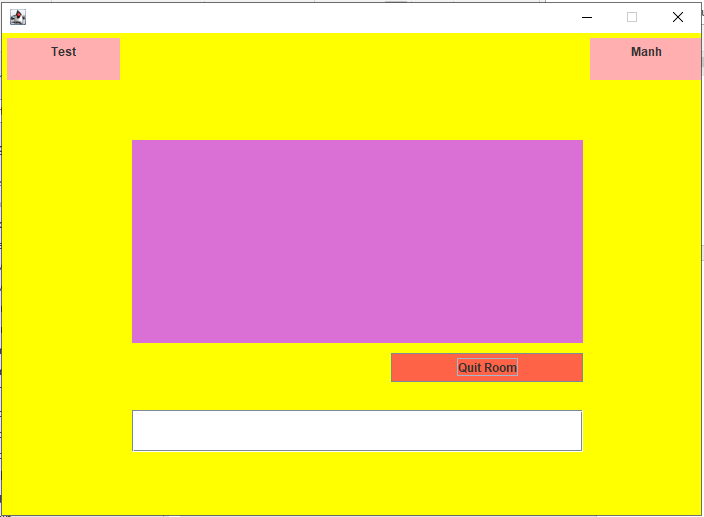
****

* 1. *Cửa sổ phòng*

*Chủ phòng*

****

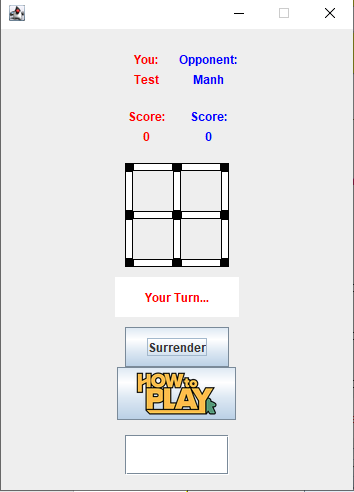
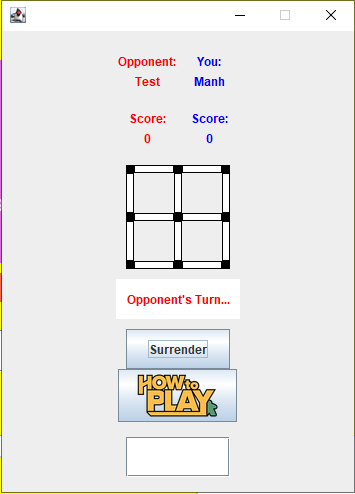
*Khách*

****

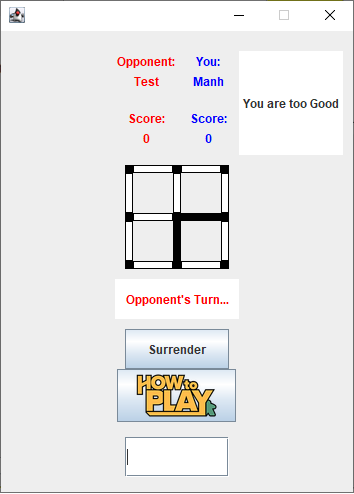
* 1. *Chat trong phòng*

****

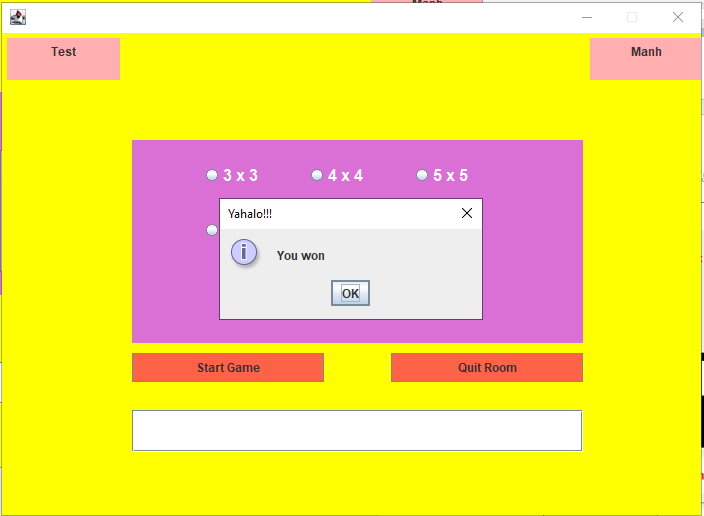
* 1. *Cửa sổ game*

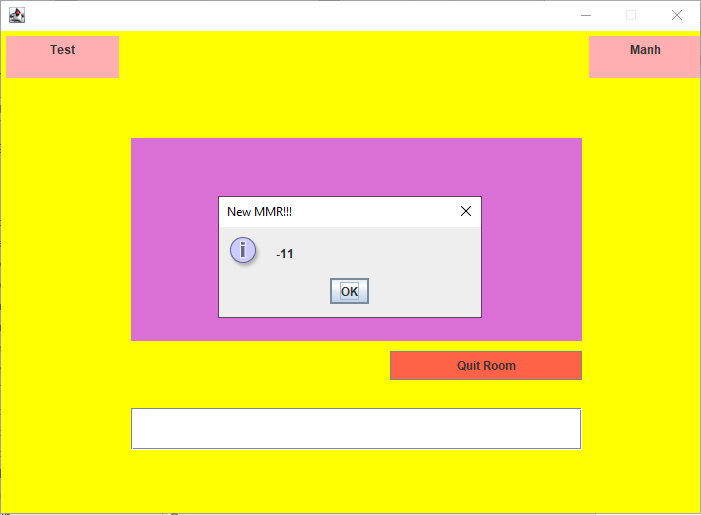
****

* 1. *Chat trong Game*

****

* 1. *Thông báo kết thúc Game*

****

****

# CHƯƠNG 5. KẾT LUẬN

## Kết quả đạt được

* Cơ hội tìm hiểu về cách thiết kế mô hình Client – Server cho ứng dụng game nhiều người chơi
* Cơ hội tìm hiểu về lập trình ứng dụng mạng có đồ họa với Java

## Dự định tương lai

* Tăng số lượng người chơi server có thể quản lý
* Cải thiện các giao thức giữa Server và Client
* Bổ sung thêm các gói tin báo lỗi
* Cải thiện giao diện người dùng

# TÀI LIỆU THAM KHẢO

**Black Art of Java Game Programming by Joel Fan, Calin Tenitchi, Eric Ries**

**Github phiên bản offline: https://github.com/maihuysl1999/DotsAndBoxes của thành viên Mai Quang Huy**