**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI**

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

**BỘ MÔN TRUYỀN THÔNG VÀ MẠNG MÁY TÍNH**

---------------o0o---------------

****

**BÁO CÁO BÀI TẬP LỚN**

**LẬP TRÌNH MẠNG**

**ĐỀ TÀI: XÂY DỰNG CHƯƠNG TRÌNH MÔ PHỎNG GAMESHOW “HÃY CHỌN GIÁ ĐÚNG” TRỰC TUYẾN**

**GVHD: Ths. Bùi Trọng Tùng**

**Họ và tên: MSSV:**

**Ngô Thái An 20172930**

**Trần Minh Quang 20173329**

**TP. HÀ NỘI, THÁNG 7 NĂM 2020**

MỤC LỤC

[DANH MỤC HÌNH ẢNH 3](#_Toc44255293)

[LỜI NÓI ĐẦU 4](#_Toc44255294)

[PHÂN CÔNG CÔNG VIỆC 5](#_Toc44255295)

[CƠ SỞ LÝ THUYẾT 6](#_Toc44255296)

[1. Tổng quan về mô hình mạng 6](#_Toc44255297)

[2. Lâp trình Socket cho ứng dụng 7](#_Toc44255298)

[3. Tiến trình Client 7](#_Toc44255299)

[4. Tiến trình Server 7](#_Toc44255300)

[5. Lập trình song song 7](#_Toc44255301)

[GIAO THỨC 9](#_Toc44255302)

[1. Biểu đồ hoạt động 9](#_Toc44255303)

[1.1 Biểu đồ hoạt động mở ứng dụng: 9](#_Toc44255304)

[1.2 Biểu đồ hoạt động Chơi game: 10](#_Toc44255305)

[1.3 Biểu đồ hoạt động Xem điểm cao: 12](#_Toc44255306)

[2. Thông điệp 13](#_Toc44255307)

[2.1 Thông điệp gửi từ Client đến Server 13](#_Toc44255308)

[2.2 Thông điệp gửi từ Server về Client 13](#_Toc44255309)

[THUẬT TOÁN XỬ LÝ SERVER 15](#_Toc44255310)

[1. Xử lý câu hỏi không trùng nhau 15](#_Toc44255311)

[2. Xử lý điểm cao 15](#_Toc44255312)

[GIAO DIỆN 16](#_Toc44255313)

[1.Giao diện Menu: 16](#_Toc44255314)

[2. Giao diện điểm cao 17](#_Toc44255315)

[3. Giao diện Màn hình nhập tên 18](#_Toc44255316)

[4.Giao diện chơi game 19](#_Toc44255317)

[5. Giao diện kết thúc game 20](#_Toc44255318)

[Tài liệu tham khảo: 21](#_Toc44255319)

# DANH MỤC HÌNH ẢNH

[Hình 1: Mô hình các tầng mạng 6](#_Toc45050586)

[Hình 2: Biểu đồ hoạt động mở ứng dụng 9](#_Toc45050587)

[Hình 3: Biểu đồ hoạt động chơi game 10](#_Toc45050588)

[Hình 4: Biểu đồ hoạt động Xem điểm cao 12](#_Toc45050589)

[Hình 5: Giao diện Menu 16](#_Toc45050590)

[Hình 6: Giao diện điểm cao 17](#_Toc45050591)

[Hình 7: Giao diện màn hình nhập tên 18](#_Toc45050592)

[Hình 8: Giao diện chơi game 19](#_Toc45050593)

[Hình 9: Giao diện kết thúc game 20](#_Toc45050594)

# LỜI NÓI ĐẦU

Trong suốt quá trình học tập môn học Lập trình mạng, nhóm chúng em đã luyện tập và phát triển được game Hãy chọn giá đúng. Báo cáo này cung cấp những cái nhìn, từ tổng quát đến chi tiết về quá trình làm, từ xây dựng các giao thức đến phát triển và hoàn thiện ứng dụng cũng như các chức năng được cung cấp.

Ai trong chúng ta cũng từng ít nhất một lần xem gameshow Hãy chọn giá đúng và mong muốn được tham gia một lần. Vì vậy chúng em quyết định chọn đề tài này để đem đến cho người chơi tựa game với một lối chơi mới.

Với việc được tiếp thu những kiến thức mới từ môn Lập trình mạng, dưới sự hướng dẫn của thầy Bùi Trọng Tùng, sau một thời gian tìm tòi và phát triển chúng em đã hoàn thành game.

Do thời gian có hạn nên có thể có những hạn chế mà chúng em chưa thể nhận ra cũng như chưa thể giải quyết. Tuy nhiên, nhóm sẽ nghiên cứu, đề ra hướng giải quyết và hoàn thiện trong thời gian sớm nhất. Trong quá trình thực hiện đề tài không tránh khỏi những sai sót, nhóm rất mong sẽ nhận được sự quan tâm góp ý và đánh giá của thầy.

Nhóm chúng em xin chân thành cảm ơn!

# 

# PHÂN CÔNG CÔNG VIỆC

Ngô Thái An:

+ Lập trình giao diện Client.

+ Lập trình xử lý thông điệp Client.

+ Xây dụng giao thức.

+ Viết báo cáo.

Trần Minh Quang:

+ Lập trình xử lý thông điệp Server.

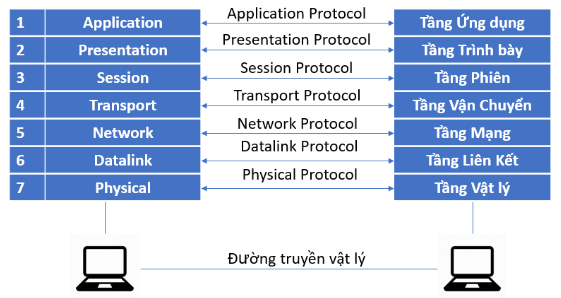
+ Xử lý thuật toán Server.

+ Comment code

+ Kiểm thử.

# CƠ SỞ LÝ THUYẾT

## Tổng quan về mô hình mạng



Hình 1: Mô hình các tầng mạng

Là một cấu trúc phả hệ có 7 tầng, nó xác định các yêu cầu cho sự giao tiếp giữa hai máy tính. Mô hình này đã được định nghĩa bởi Tổ chức tiêu chuẩn hoá quốc tế (International Organization for Standardization) trong tiêu chuẩn số 7498-1 (ISO standard 7498-1). Mục đích của mô hình là cho phép sự tương giao (interoperability) giữa các hệ máy đa dạng được cung cấp bởi các nhà sản xuất khác nhau. Mô hình cho phép tất cả các thành phần của mạng hoạt động hòa đồng, bất kể thành phần ấy do ai tạo dựng. Vào những năm cuối thập niên 1980, ISO đã tiến cử việc thực thi mô hình OSI như một tiêu chuẩn mạng. Mô hình OSI gồm 7 tầng phân chia chức năng của một giao thức ra thành một chuỗi các tầng cấp. Mỗi một tầng cấp có một đặc tính là nó chỉ sử dụng chức năng của tầng dưới nó, đồng thời chỉ cho phép tầng trên sử dụng các chức năng của mình:

Tầng Vật lý (Physical Layer): Chuyển dữ liệu ở dạng bit thành tín hiệu và truyền.

Tầng Liên kết dữ liệu (Datalink Layer): Hỗ trợ việc truyền thông cho các thành phần kế tiếp trên cùng một mạng.

Tầng Mạng (Netwwork Layer): Chọn đường và chuyển tiếp gói tin giữa các máy, các mạng.

Tầng Giao vận (Transport Layer): Điều khiển việc truyền dữ liệu giữa các tiến trình của tầng Ứng dụng.

Tầng Phiên (Session Layer): quản lý phiên làm việc, đồng bộ hoá phiên, khôi phục quá trình trao đổi dữ liệu

Tầng Trình diễn (Presentation Layer): được dùng để chuẩn hoá biểu diễn dữ liệu trong các ứng dụng ở tầng Ứng dụng một cách chính xác, rõ ràng.

Tầng Ứng dụng (Application Layer): cung cấp các ứng dụng trên mạng.

Trong khuôn khổ môn lập trình mạng, ta đi sâu vào tầng Ứng dụng trong mạng với các giao thức cho phép thực hiện mô hình kết nối Client – Server trong trò chơi.

## Lâp trình Socket cho ứng dụng

Socket là một khái niệm cho phép giao tiếp giữa các tiến trình khác nhau trên cùng hoặc trên nhiều máy khác nhau. Nói cách khác, nó là cách để các máy tính giao tiếp, trò chuyện với nhau.

Socket được sử dụng trong các ứng dụng dạng Client – Server. Server có thể coi là tiến trình cho phép thực hiện các hàm để đáp ứng lại các yêu cầu từ tiến trình client. 1 socket bao gồm hai thành phần: Địa chỉ IP và cổng. Từ đây, các máy tính có thể truyền chính xác thông điệp cho các tiến trình trên các máy khác nhau.

## Tiến trình Client

Là tiến trình thực hiện các yêu cầu (request) lên Server để nhằm thu được thông tin mong muốn. Sau khi thu được phản hồi, tiến trình này có thể sẽ kết thúc (terminate), hoặc thực hiện các tác vụ khác với thông tin thu được.

## Tiến trình Server

Là tiến trình thực hiện các tác vụ xử lý yêu cầu từ các tiến trình Client. Sau khi thu được thông tin từ Client, tiến trình này sẽ xử lý, cập nhật thông tin và gửi thông điệp phản hồi trở lại cho Client. Khi đã sẵn sàng, tiến trình Server sẽ có thể thực hiện tiếp các quá trình xử lý nhằm phục vụ các yêu cầu của tiến trình Client khác. Tiến trình này cần luôn sẵn sàng để được gọi nhằm phục vụ các yêu cầu được gửi đến.

## 5. Lập trình song song

- Lập trình song song là một hình thức sao cho Server có thể thực hiện được nhiều công việc trong cùng một thời điểm.

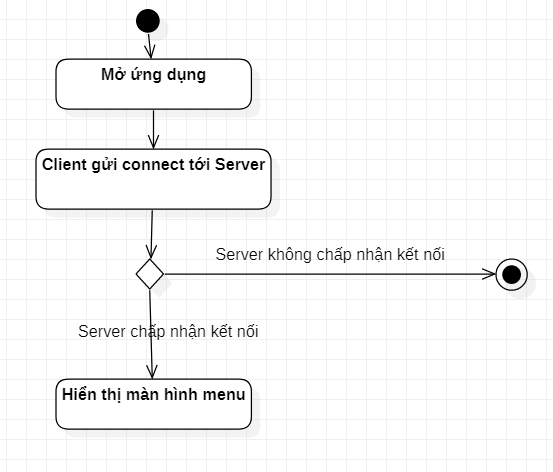
- Việc lập trình song song được xử lý bằng kỹ thuật đa luồng.

- VD: Khi Server nhận một request từ Client thì sẽ tạo ra một Thread riêng để xử lý tính toán. Như vậy thì khi có 2 hay nhiều Client cùng đồng thời truy cập thì Server cùng có thể xử lý đồng thời được cho tất cả các client kết nối đến bằng các xử lý chúng riêng ở từng Thread.

# GIAO THỨC

## Biểu đồ hoạt động

### 1.1 Biểu đồ hoạt động mở ứng dụng:



Hình 2: Biểu đồ hoạt động mở ứng dụng

Mô tả: Khi người dùng mở ứng dụng, Client sẽ gửi một kết nối tới Server và chờ nghe kết nối. Nếu Server chấp nhận kết nối, thì Client sẽ hiển thị màn hình Menu, ngược lại kết thúc.

### 1.2 Biểu đồ hoạt động Chơi game:

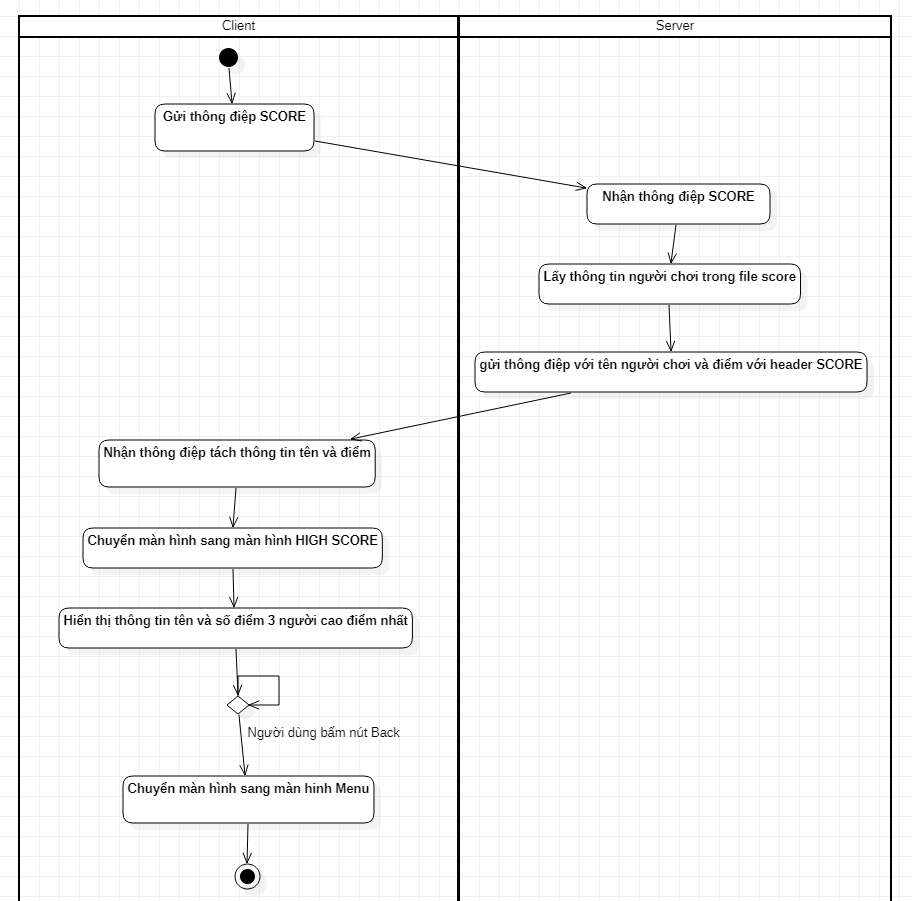
A flock of white map

Description automatically generated

Hình 3: Biểu đồ hoạt động chơi game

Mô tả: Khi người dùng bấm vào nút START game, người dùng sẽ được chuyển đến một màn hình khác được gọi là màn hình nhập tên. Tại đấy người dùng sẽ nhập tên cho phiên chơi đó, tên đó sẽ gắn với người chơi đến khi kết thúc phiên chơi. Sau khi nhấn Enter, sẽ có một thông điệp với header START kèm với tên người chơi được gửi tới Server, khi đó game sẽ bắt đầu. Server sẽ random ra câu hỏi và gửi về cho Client thông điệp QUEST kèm số thứ tự câu hỏi, câu hỏi, các đáp án. Client sau khi nhận được thông điệp sẽ chuyển màn hình sang màn hình Gameplay, tách thông điệp để lấy chỉ số câu hỏi từ đó lấy hình ảnh của câu hỏi trong folder câu hỏi, tách câu hỏi và các đáp án rồi hiển thị lên màn hình. Khi người chơi chọn đáp án, đáp án đó được gửi cho Server, Server sẽ so sánh đáp án được gửi lên với đáp án chính xác, nếu đúng điểm người chơi sẽ được tăng thêm một, sau đó quay lại bước Random câu hỏi và thực hiện tuần tự như vậy, nếu sai Server sẽ so sánh điểm của người chơi với điểm trong file điểm cao để cập nhật lại file điểm cao, rồi gửi cho Client thông điệp POINT cùng điểm số của người chơi. Khi người Client nhận được thông điệp này Client sẽ chuyển màn hình đến màn hình Gameover và hiển thị điểm người chơi. Tại đây người chơi có thể lựa chọn Retry để tiếp tục chơi hoặc Menu để quay lại màn hình menu, Quit để thoát game.

### 1.3 Biểu đồ hoạt động Xem điểm cao:



Hình 4: Biểu đồ hoạt động Xem điểm cao

Mô tả: Khi người dùng bấm Score Client sẽ gửi một thông điệp SCORE tới Server. Khi Server nhận được thông điệp SCORE, Server sẽ lấy thông tin người chơi và điểm số trong file score và gửi lại thông điệp SCORE cùng điểm số và tên người chơi cho Client. Client tách tên người chơi và điểm số từ thông điệp SCORE. Client chuyển màn hình sang màn hình HIGH SCORE, và hiển thị tên và điểm số đã lấy được từ thông điệp. Nếu người dùng bấm Back thì chuyển màn hình sang màn hình Menu.

## Thông điệp

**Khuôn dạng thông điệp: Xâu ký tự, phân cách giữa các trường bằng ký tự ‘|’.**

## 2.1 Thông điệp gửi từ Client đến Server

#### 2.1.1 Thông điệp Bắt đầu game:

START|Tên người chơi

Trong đó:

Tên người chơi: Do người chơi nhập đại diện cho người chơi trong phiên chơi.

Khi chọn START, màn hình nhập tên người chơi xuất hiện .Sau khi nhập tên và ấn nút Enter thông điệp được gửi tới Server thông báo bắt đầu game lúc này Server lấy tên người chơi kèm thông điệp gửi để quản lý phiên của người chơi.

#### 2.1.2 Thông điệp trả lời câu hỏi

ANS |Đáp án

Trong đó:

Đáp án: Số từ 1 đến 4 trong đó 1 tương ứng với đáp án thứ nhất, 2 đáp án thứ hai, 3 đáp án số 3, 4 đáp án cuối cùng.

Thông điệp được gửi đi sau khi người dùng nhấn chọn đáp án. Đáp án được gửi cùng thông điệp với header ANS\_\_ được gửi tới Server.

#### 2.1.3 Thông điệp lấy thông tin điểm cao

SCORE

Tại màn hình Menu, nếu người chơi chọn HIGH SCORE thông điệp sẽ được gửi tới Server để truy vấn điểm của những người có điểm cao nhất được lưu trong file score.

## 2.2 Thông điệp gửi từ Server về Client

#### 2.2.1 Thông điệp gửi câu hỏi**:**

QUEST|id|quesText|firstAns|secondAns|thirdAns|fourthAns

Trong đó:

Id: số thứ tự câu hỏi trong list câu hỏi

quesText: Mảng char chứa nội dung câu hỏi

firstAns: Đáp án thứ nhất

secondAns: Đáp án thứ hai

thirdAns: Đáp án thứ ba

fourthAns: Đáp án thứ tư

Thông điệp được gửi khi nhận được thông điệp với header START gửi từ Client (thông điệp báo bắt đầu game mới) hoặc sau khi kiểm tra kết quả đáp án của người chơi là đúng. Nếu trước đó Client gửi thông điệp có header ANS\_\_ kèm đáp án thì thông điệp này tương ứng với thông báo đáp án trước đó là chính xác.

#### 2.2.2 Thông điệp gửi điểm cho người chơi

POINT|point

Trong đó:

point: Số điểm đạt được của người chơi.

Thông điệp được gửi khi Server kiểm tra đáp án người chơi gửi là sai, thông điệp được gửi để báo cho Client biết đáp án là sai và kết thúc việc truyền đáp án về Client đến khi nhận được thông điệp có header START mới.

#### 2.2.3 Thông điệp gửi điểm cao về cho người chơi

SCORE|hsName1|score1|hsName2|score2|hsName3|score3

Trong đó:

hsName1: Tên người chơi có điểm cao nhất

score1: Điểm của người chơi có điểm cao nhất

hs Name2: Tên người chơi có điểm cao thứ hai

score2: Điểm của người chơi có điểm cao thứ hai

hsName3: Tên người chơi có điểm cao thứ ba

score3: Điểm của người chơi có điểm cao thứ ba

Thông điệp được gửi khi Server nhận được thông điệp có header SCORE từ client.

# THUẬT TOÁN XỬ LÝ SERVER

## Xử lý câu hỏi không trùng nhau

Vấn đề: Khi người chơi bắt đầu game, Server sẽ random các câu hỏi để gửi về cho Client. Vấn đề đặt ra là làm sao trong một phiên chơi các câu hỏi không được trùng nhau.

Giải quyết: Tạo một vector lưu trữ các chỉ số (index) câu hỏi đã dùng. Sau mỗi lần câu hỏi được gửi sẽ cập nhật index thêm vào vector lưu trữ. Mỗi lần random câu hỏi sẽ kiểm tra trong vector đã có chỉ số câu hỏi đó chưa, nếu đã có random ra chỉ số mới, nếu chưa chọn số đó và gửi về cho Client.

## Xử lý điểm cao

Vấn đề: Khi người chơi đạt điểm cao hơn so với người chơi trong phần điểm cao, thì bảng điểm cao phải cập nhật, nếu điểm của người chơi cao hơn người có điểm cao thứ ba thì điểm của người chơi sẽ được cập nhật ở vị trí số ba, nếu điểm của người chơi cao hơn điểm của người chơi ở vị trí thứ hai thì điểm của người chơi sẽ được cập nhật ở vị trí thứ hai còn điểm người chơi ở vị trí số hai ban đầu sẽ được cập nhật lại ở vị trí số ba. Tương tự với vị trí thứ nhất.

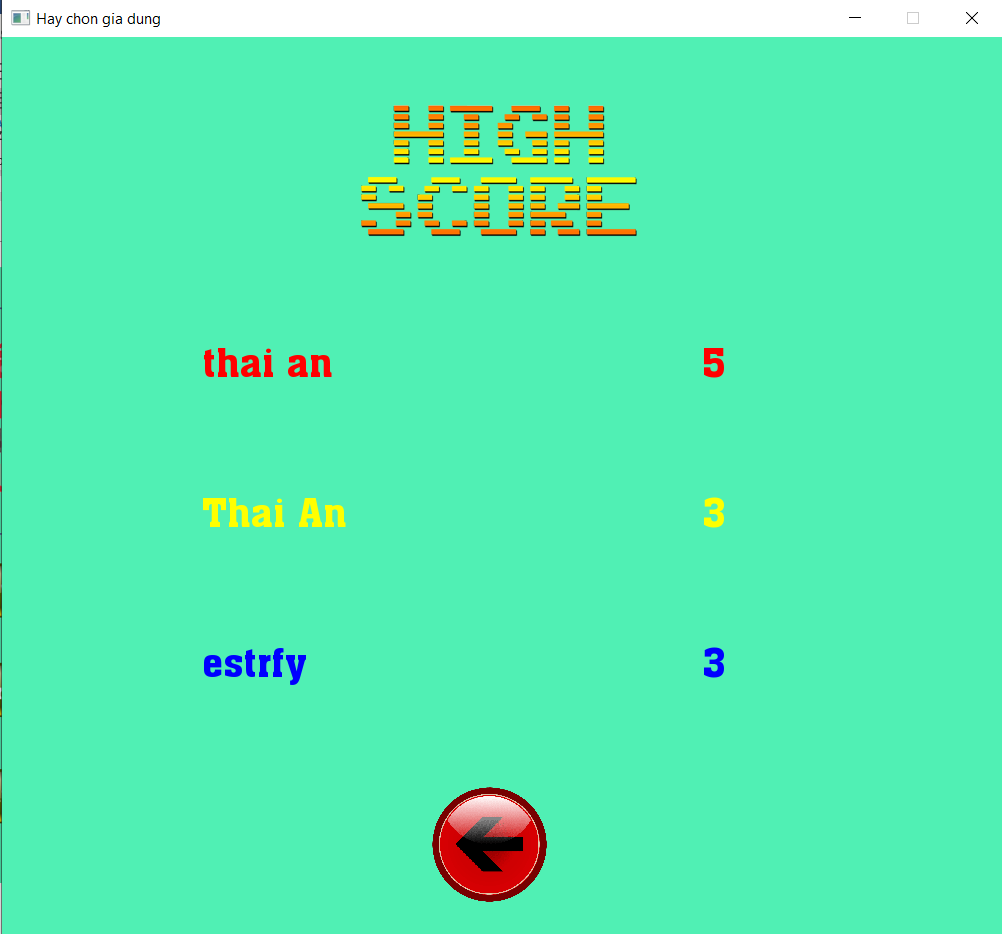
Giải quyết: Tạo ra một struct lưu trữ tên người chơi và điểm cao. Từ file điểm cao tạo ra một list điểm cao gồm ba phần tử lưu trữ điểm của ba người cao nhất. Sau khi người chơi thất bại sẽ kiểm tra điểm của người chơi với điểm của trong list điểm cao và cập nhật lại list. Sau đó list này sẽ được cập nhật vào file điểm cao.

# GIAO DIỆN

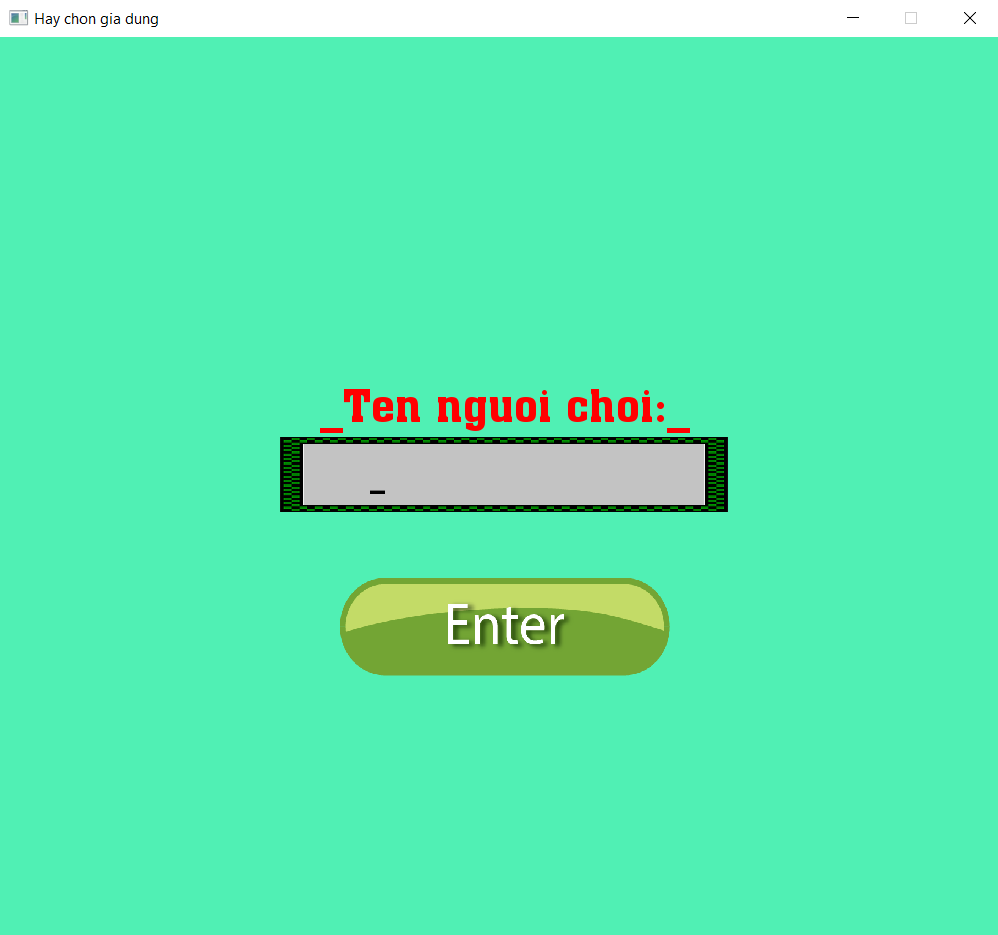
## 1.Giao diện Menu:



Hình 5: Giao diện Menu

2. Giao diện điểm cao

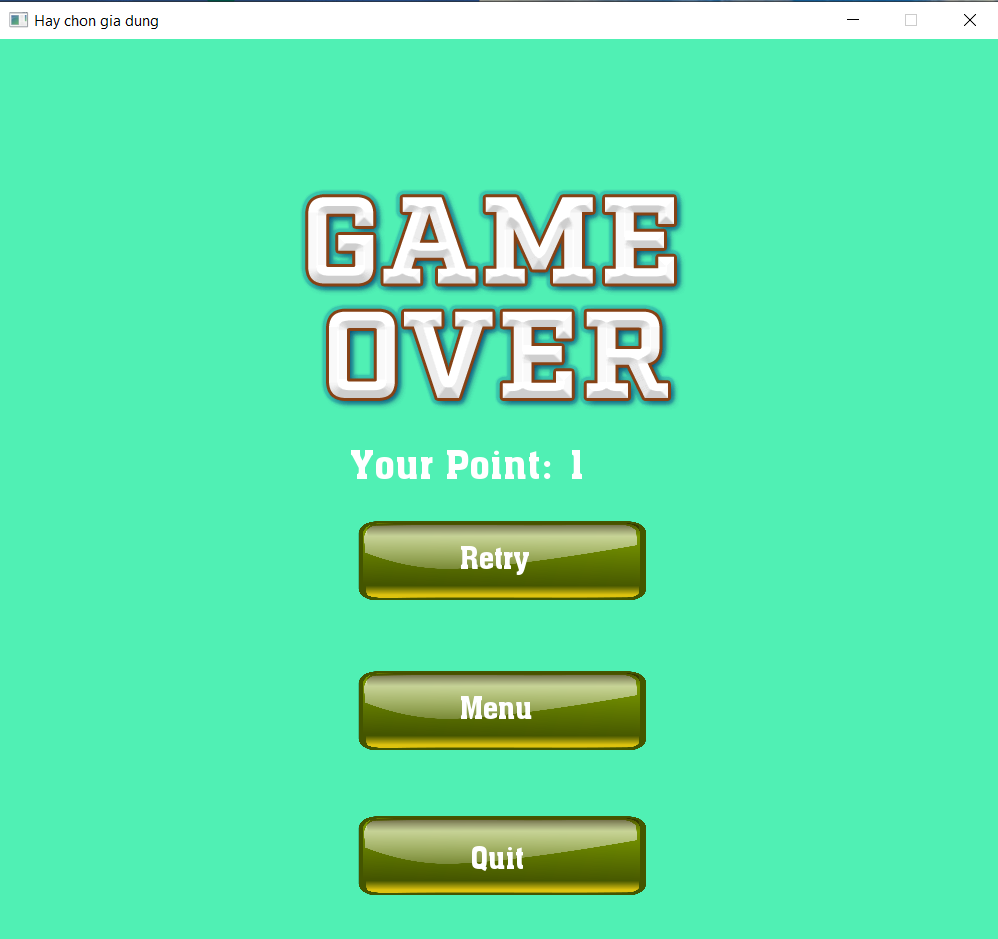
Hình 6: Giao diện điểm cao

3. Giao diện Màn hình nhập tên

Hình 7: Giao diện màn hình nhập tên

4.Giao diện chơi game

Hình 8: Giao diện chơi game

5. Giao diện kết thúc game

Hình 9: Giao diện kết thúc game

# Tài liệu tham khảo:

Slide lập trình mạng. ThS. Bùi Trọng Tùng

Sfml Game Development Ebook. Artur Moreria, Henrik Vogelius Hansson, Jan Haller