

BÁO CÁO THỰC HÀNH

Môn học: **Lập trình mạng căn bản**

Lab 06 – Bài tập tổng hợp

GVHD: Tô Trọng Nghĩa

1. THÔNG TIN CHUNG:

Lớp: NT106.O22.ANTT.1

STT	Họ và tên	MSSV	Email
1	Lê Quốc Ngô	22520951	22520951@gm.uit.edu.vn
2	Trần Tiến Nhật	22521030	22521030@gm.uit.edu.vn

2. NỘI DUNG:

1. *Gameplay*
2. *Giải quyết yêu cầu bài thực hành*
3. *Giải quyết yêu cầu mở rộng*
4. *Demo*

Phần bên dưới của báo cáo này là tài liệu báo cáo chi tiết của nhóm thực hiện.

BÁO CÁO CHI TIẾT

1. Gameplay

Luật chơi cho trò chơi "Đoán số" như sau:

1. Server sẽ quyết định số lượt chơi mỗi ván và thời gian mỗi lượt là 10 giây.
2. Server sẽ đưa ra một khoảng giới hạn và số cần tìm nằm trong khoảng đó. Ví dụ: từ 1 đến 100.
3. Các người chơi sẽ submit con số mà họ cho là đáp án. Mỗi lần submit cách nhau ít nhất 3 giây.
4. Nếu người chơi đoán đúng, họ sẽ được cộng 10 điểm.
5. Nếu người chơi đoán sai, họ sẽ bị trừ 1 điểm.
6. Trò chơi không giới hạn số lần đoán. Người chơi có thể tiếp tục đoán cho đến khi hết thời gian của lượt.
7. Khi có người chơi đoán đúng, trò chơi sẽ bỏ qua các câu trả lời của người chơi khác và chờ đến khi kết thúc thời gian của lượt.
8. Người chơi có thể sử dụng tính năng "Don't have answer". Trong trường hợp này, trò chơi sẽ tự động submit một số trong khoảng giới hạn của trò chơi trong một lượt.
9. Ngoài ra, người chơi cũng có thể sử dụng tính năng "Autoplay". Trong trường hợp này, chương trình sẽ tự động submit các số trong khoảng giới hạn, và người chơi không cần phải tham gia đoán số.
10. Mục tiêu của người chơi là đạt điểm số cao nhất bằng cách đoán đúng số nhanh chóng.

2. Giải quyết yêu cầu bài thực hành

Bên phía Clients, thiết kế giao diện cho phép điền tên người chơi, điền con số dự đoán và thông báo của server (lượt chơi mới bắt đầu, người thắng cuộc, phạm vi số cần đoán,...)

Giao diện phía Client khi bắt đầu:

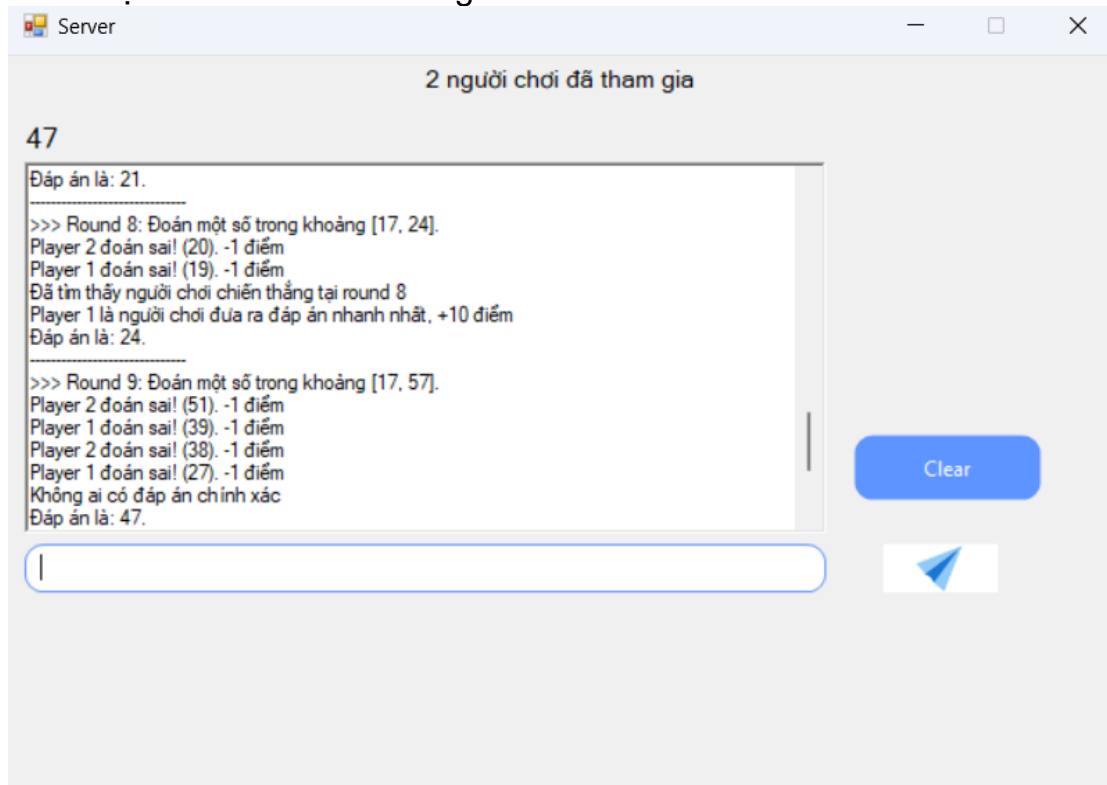
The screenshot shows a window titled "Player 1". Inside, it says "1 người chơi đã tham gia". The text "Số cần đoán nằm trong khoảng:" is followed by a text input field containing "range". Below this, "Thời gian còn lại:" is followed by a text input field containing "x". There is a "Submit" button next to the "Câu trả lời của bạn:" label and a text input field. To the right, there are buttons for "Autoplay", "Don't have answer", "Ready", and "Clear". At the bottom right, there is a blue paper plane icon.

Giao diện phía Client sau khi kết thúc ván chơi:

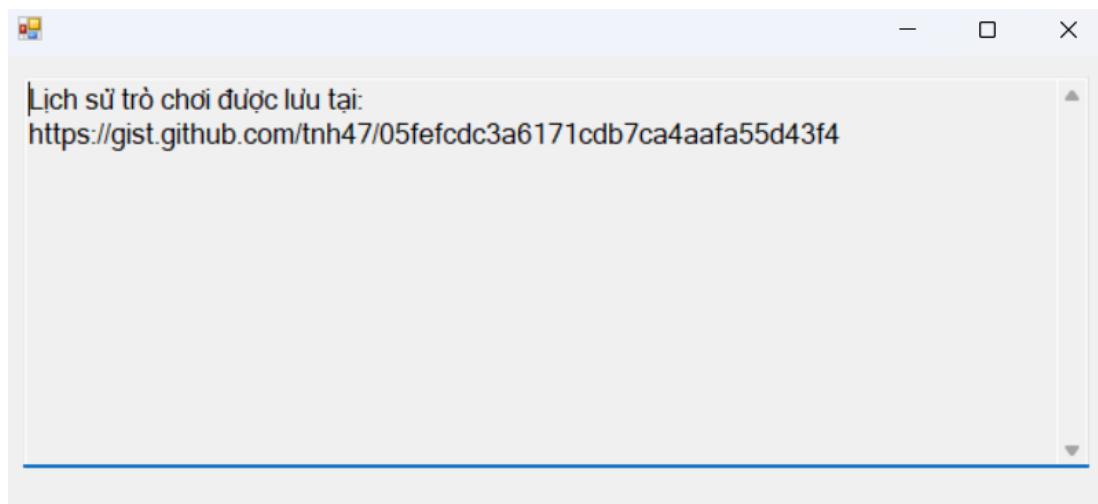
The screenshot shows the same window titled "Player 1", but now it says "2 người chơi đã tham gia". The text "Số cần đoán nằm trong khoảng:" is followed by a text input field containing "[17, 57]". Below this, "Thời gian còn lại:" is followed by a text input field containing "x". There is a "Submit" button next to the "Câu trả lời của bạn:" label and a text input field. To the right, there are buttons for "Autoplay", "Don't have answer", "Ready", and "Clear". At the bottom right, there is a blue paper plane icon. The main text area displays the following text: >>> Round 9: Đoán một số trong khoảng [17, 57]. Player 2 đoán sai! (51). -1 điểm Player 1 đoán sai! (39). -1 điểm Player 2 đoán sai! (38). -1 điểm Player 1 đoán sai! (27). -1 điểm Không ai có đáp án chính xác Đáp án là: 47. Điểm cao nhất là: 7 + Player 1 Điểm của bạn: 7 ===== Tạo game mới... Chờ người chơi tham gia...

Bên phía Server thiết kế giao diện hiển thị các thông tin số đang cần tìm, phạm vi số đang cần tìm, lịch sử trò chơi, số người đang tham gia, số lượt chơi.

Giao diện của Server khi đang chơi:



Sau khi kết thúc ván chơi, một cửa sổ chứa nơi lưu lịch sử tại GitHub Gists hiện lên:



Giới hạn khoảng thời gian giữa 2 lần gửi kết quả dự đoán tối thiểu 3 giây và hiển thị thời gian đếm ngược trên Form của client.

Sau khi submit câu trả lời, nút Submit sẽ Disabled trong vòng 3 giây:

Câu trả lời của bạn:

Câu trả lời của bạn:

Thời gian đếm ngược nằm phía trên thanh Submit:

Player 1

1 người chơi đã tham gia

Số cần đoán nằm trong khoảng: [41, 42]

Thời gian còn lại: 1

Câu trả lời của bạn:

Tạo game mới...
Chờ người chơi tham gia...
>>> Player 1 đã sẵn sàng!
Hiện có 1 người chơi
Trò chơi có 3 round
>>> Round 1: Đoán một số trong khoảng [31, 55].
Không ai có đáp án chính xác
Đáp án là: 38.

>>> Round 2: Đoán một số trong khoảng [14, 28].
Không ai có đáp án chính xác
Đáp án là: 15.

>>> Round 3: Đoán một số trong khoảng [41, 42].

Phát triển tính năng tự động chơi, cho phép client sinh ngẫu nhiên kết quả và tự động gửi (các kết quả tự sinh ra không trùng với kết quả đã thử trước đó và thuộc phạm vi con số cần tìm)

Có 2 lựa chọn cho Client: **Autoplay** và **Don't have answer**

Đối với **Don't have answer**, chương trình sẽ random 1 số trong khoảng giới hạn được cho, và tự động submit 1 lần.

Đối với **Autoplay**, chương trình sẽ random 1 số trong khoảng giới hạn được cho, và tự động submit mỗi 3 giây, có hiệu lực đến hết ván chơi.

Code xử lý:

```

4 references
private void autoSubmit()
{
    if (this.InvokeRequired)
    {
        this.Invoke(new MethodInvoker(delegate ()
        {
            autoSubmit();
        }));
    }
    else
    {
        btnSubmit.Enabled = btnAutoPlaySingleTurn.Enabled = answer.Enabled = false;
        if (isAuto)
        {
            btnAutoplayWholeGame.Enabled = false;
        }
        int val = rand.Next(0, valRange + 1);
        submit(ansList[val]);
    }
}

1 reference
private void autoTurn_Click(object sender, EventArgs e)
{
    autoSubmit();
}

1 reference
private void autoAllGame_Click(object sender, EventArgs e)
{
    isAuto = true;
    autoSubmit();
}

```

Trong trường hợp cả 2 player đều sử dụng tính năng **Autoplay**, chương trình hiển thị như sau:

```

>>> Round 2: Đoán một số trong khoảng [15, 45].
nhát đoán sai! (45). -1 điểm
ngo đoán sai! (37). -1 điểm
nhát đoán sai! (41). -1 điểm
ngo đoán sai! (36). -1 điểm
Không ai có đáp án chính xác
Đáp án là: 20.

```

Server tự quyết định số lần chơi (tối thiểu là 5) để kết thúc trò chơi. Khi trò chơi kết thúc, phía clients sẽ lưu toàn bộ lịch sử thông báo của server thành file `history.txt`, server gửi toàn bộ lịch sử trò chơi lên GitHub Gists

Server quyết định số lần chơi bằng hàm `Random()`, sau đó gửi đến cho người chơi bằng hàm `broadcast()`:

```
private void roundStart()
{
    Thread.Sleep(2000);
    timeupCount = 0;

    if (round == 0)
    {
        round = rand.Next(5, 10);
        broadcast($"mTrò chơi có {round} round");
        currentRound = 1;
    }

    startRange = rand.Next(0, 50);
    endRange = startRange + rand.Next(1, 50);
    ansNumber = rand.Next(startRange, endRange + 1);
    broadcast($"m>>> Round {currentRound}: Đoán một số trong khoảng [{startRange}, {endRange}]." +
        $"n@@@Nextround!@@@{rand.Next(5, 10)}\t{startRange}\t{endRange}\t{ansNumber}");
    currentRound++;
    correctPlayer = "";
}
```

Sau khi trò chơi kết thúc, cả server và client rời khỏi phòng chơi, lịch sử trò chơi sẽ được lưu vào file `.txt` với format **“History_{tên client}.txt”**.

```
try
{
    String path = Path.Combine(
        Path.GetDirectoryName(Application.ExecutablePath),
        $"History_{this.Text}.txt"
    );
    StreamWriter sw;
    if (!File.Exists(path))
    {
        sw = File.CreateText(path);
    }
    else
    {
        sw = File.AppendText(path);
    }
    String hostOrJoin;
    if (this.Text == "Server")
    {
        hostOrJoin = $">>> {time} - Server hosted a connection... <<<";
    }
    else
    {
        hostOrJoin = $">>> {time} - {this.Text} connected to Server... <<<";
    }
    conversation.Text = $"{hostOrJoin}\n\n{conversation.Text}\n>>> Connection closed <<<\n\n\n";
    foreach (String line in conversation.Lines)
    {
        sw.WriteLine(line);
    }
    sw.Close();
}
```

Lịch sử trò chơi của server sẽ được gửi đến GitHub Gists, sau đó hiển thị lại địa chỉ nơi lưu:

```
1 reference
private async void CreateGist(string content)
{
    string gistDescription = "Lịch sử trò chơi";
    bool isPublic = true;
    string personalAccessToken = "ghp_d6XG520NXMdIg0CWuehcse7KHI5euL3Fi8Ls";

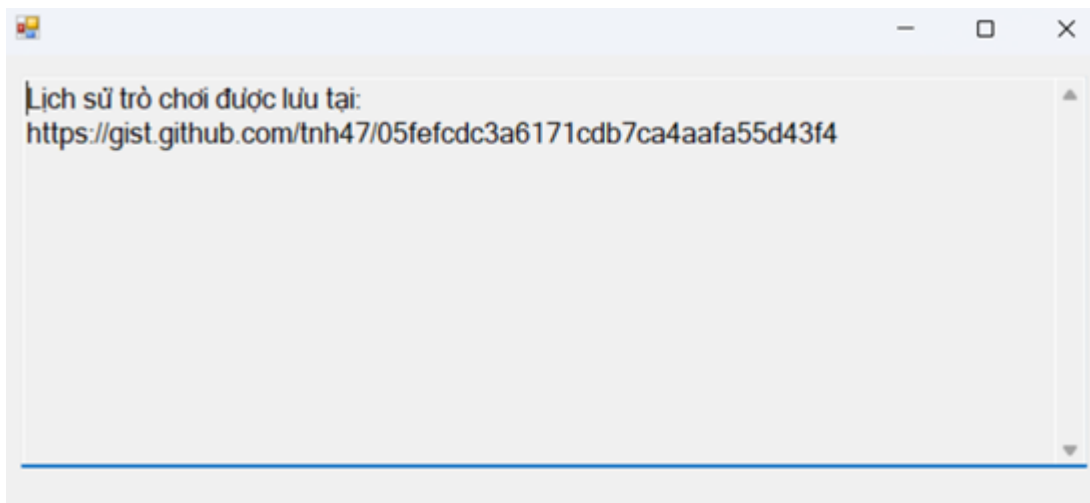
    var gist = new
    {
        description = gistDescription,
        @public = isPublic,
        files = new Dictionary<string, object>
        {
            { "history.txt", new { content = content } }
        }
    };

    string gistJson = JsonConvert.SerializeObject(gist);

    using (HttpClient client = new HttpClient())
    {
        client.DefaultRequestHeaders.Add("User-Agent", "CSharp-App");
        client.DefaultRequestHeaders.Add("Authorization", "token " + personalAccessToken);

        var response = await client.PostAsync(
            "https://api.github.com/gists",
            new StringContent(gistJson, Encoding.UTF8, "application/json")
        );

        if (response.IsSuccessStatusCode)
        {
            string result = await response.Content.ReadAsStringAsync();
            dynamic gistResult = JsonConvert.DeserializeObject(result);
            textBox1.Text = $"Lịch sử trò chơi được lưu tại: \r\n{gistResult.html_url}";
        }
        else
        {
            MessageBox.Show("Lỗi: " + response.StatusCode);
        }
    }
}
```



3. Giải quyết yêu cầu mở rộng

Khi kết thúc trò chơi, thông báo người thắng chung cuộc

Kết thúc trò chơi, người thắng cuộc là người có số điểm cao nhất. Hiện thị thông báo người thắng cuộc, và hiển thị điểm của mình ở màn hình:

```
=====
Điểm cao nhất là: 4
+ nhất
Điểm của bạn: 4
=====
```

Code xử lý:

```
ingame = false;
int highscore = int.MinValue;
foreach (var i in scoreBoard)
{
    if (i.Value > highscore)
    {
        highscore = i.Value;
    }
}

string message = $"mĐiểm cao nhất là: {highscore}\nNgười chơi có điểm cao nhất:\n";
foreach (var i in scoreBoard)
{
    if (i.Value == highscore)
    {
        message += $"m  + {i.Key}\n";
    }
}
```

Thêm tính năng tính điểm cho người chơi (ví dụ: trừ điểm mỗi lần đoán sai, cộng điểm mỗi lần đoán đúng)

Khi người chơi đoán đúng được cộng 10 điểm, và không nhận các submit từ các người chơi khác cho đến hết thời gian của lượt này.

Khi người chơi đoán sai bị trừ 1 điểm, không giới hạn số lần đoán.

```
>>> Round 6: Đoán một số trong khoảng [17, 37].
nhát đoán sai! (35). -1 điểm
ngo đoán sai! (29). -1 điểm
nhát đoán sai! (22). -1 điểm
ngo đoán sai! (20). -1 điểm
nhát đoán sai! (27). -1 điểm
Đã tìm thấy người chơi chiến thắng tại round 6
ngo là người chơi đưa ra đáp án nhanh nhất, +10 điểm
Đáp án là: 34.
```

Code xử lý:

```
if (correctPlayer == "" && timeupCount < readyPlayers.Count)
{
    try
    {
        int ans = Int32.Parse(data.Substring(1));
        if (ans == ansNumber)
        {
            correctPlayer = username;
            scoreBoard[username] += 10;
            broadcast($"mĐã tìm thấy người chơi chiến thắng tại round {currentRound - 1}");
        }
        if (ans != ansNumber)
        {
            broadcast($"m{username} đoán sai! ({ans}). -1 điểm");
            scoreBoard[username]--;
        }
    }
    catch
    {
        broadcast($"m{username} nhập đáp án không hợp lệ. -1 điểm");
        scoreBoard[username]--;
    }
}
```

4. Demo

Link video:

https://drive.google.com/file/d/1hsFHNwNB2TsZh7aPcjMO8EcFwR_T-2R8/view?usp=sharing