

Assignment 2 Report

by Rhys Treharne

Submission date: 13-May-2025 02:47PM (UTC+0100)

Submission ID: 259080545

File name: 36124_Rhys_Treharne_Assignment_2_Report_664977_627138170.docx (1.29M)

Word count: 1331

Character count: 6893



IMPLEMENTATION REPORT



Rhys Treharne 2120039
ASSIGNMENT 2 Tim Bashford

Abstract

The networked application is an interactive Star Wars themed trivia quiz game to test the knowledge of Star Wars fans. The application is built in the C++ language and includes key practices to successfully build the trivia quiz. The goal of the quiz is to answer as many questions as possible correctly, the game includes a scoring system to track how many questions the client gets correct.

Table of Figures

Figure 1: Questions.h	5
Figure 2: Questions.cpp	6
Figure 3: Server Code Receiving Clients Name	6
Figure 4: Client Code Sending Client's Name	7
Figure 5: Client Name Outcome	7
Figure 6: Server Sending Questions to Client	8
Figure 7: Client Receiving Questions	8
Figure 8: Sending & Receiving Questions	8
Figure 9: Client Sending Their Response Code	9
Figure 10: Server Receiving the Response and Determining if the answer is correct or incorrect	10
Figure 11: Sending the Result To The Client	10
Figure 12: Expected Outcome For Sending Client Answer & Server's Response To The Answer	10
Figure 13: Scoring System Server Side	11
Figure 14: Expected Outcome Scoring System	11
Figure 15: Server Sending Final Score to Client	12
Figure 16: Final Score Client Side	12
Figure 17: Expected Outcome Final Score	12
Figure 18: Error Handling Example 1	13
Figure 19: Error Handling Example 2	14
Figure 20: Client Joining Server Message (Server)	14
Figure 21: Client Disconnecting From The Server	15
Figure 22: Completed Trivia Quiz (Final Score)	16

Contents

Abstract.....	1
Table of Figures.....	2
Contents.....	3
Introduction	4
Features.....	5
Question Handling	5
User Interaction	6
Scoring System.....	10
Design Decisions & Best Practices	12
Language.....	13
Code Organisation.....	13
Error Handling/Input Validation	13
Testing	15
User Experience	15
Conclusion	16
References.....	17

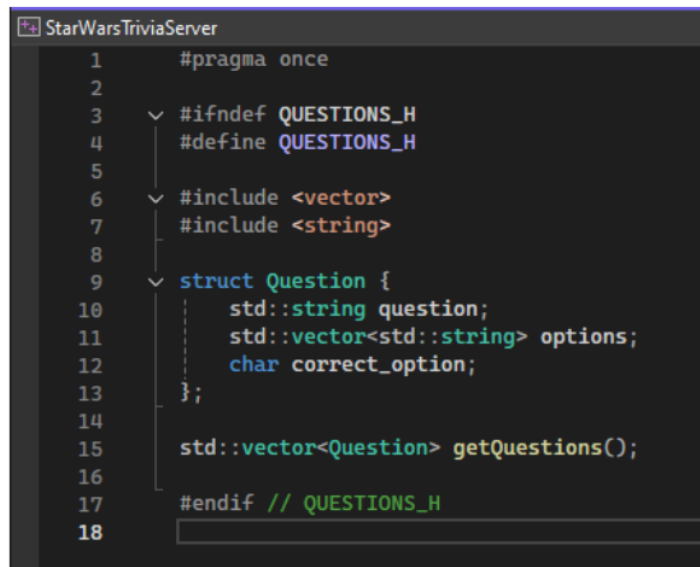
Introduction

The purpose of this implementation is to exchange information from a server through to the client and vice versa using sockets. This report will describe the features, design decisions and best practices. The trivia quiz is based on a Star Wars theme, each correct answer and the client will receive a point. At the end of the game, the client will receive their final score.

Features

Question Handling

The questions are stored in two files, one is a header file, and the other is a cpp file. The reasoning behind separating the questions into individual files is to ensure they're maintainable. If the questions were to be further developed, this approach ensures that the rest of the code won't be affected. The question files are stored on the server side due to the server sending the questions and the client to receive the question. Below are snippets of the code used to create and manage the questions:

A screenshot of a code editor window titled "StarWarsTriviaServer". The editor displays the content of a header file named "Questions.h". The code is written in C++ and includes preprocessor directives, standard library includes, and a struct definition. The lines are numbered from 1 to 18 on the left side of the editor. The code is as follows:

```
1  #pragma once
2
3  #ifndef QUESTIONS_H
4  #define QUESTIONS_H
5
6  #include <vector>
7  #include <string>
8
9  struct Question {
10     std::string question;
11     std::vector<std::string> options;
12     char correct_option;
13 };
14
15 std::vector<Question> getQuestions();
16
17 #endif // QUESTIONS_H
18
```

Figure 1: Questions.h

```

1 #include <vector>
2 #include <string>
3
4 struct Question {
5     std::string question;
6     std::vector<std::string> options;
7     char correct_option;
8 };
9
10 std::vector<Question> getQuestions() {
11     std::vector<Question> questions = {
12         {"1. Who is Luke Skywalker's father?", {"a) Obi-Wan Kenobi", "b) Yoda", "c) Anakin Skywalker", "d) Mace Windu"}, 'c'},
13         {"2. What is the name of Han Solo's ship?", {"a) Star Destroyer", "b) Millennium Falcon", "c) X-Wing", "d) TIE Fighter"}, 'b'},
14         {"3. Who trained Luke Skywalker in 'The Empire Strikes Back'?", {"a) Obi-Wan Kenobi", "b) Yoda", "c) Mace Windu", "d) Qui-Gon Jinn"}, 'b'},
15         {"4. What is the name of the Wookiee home planet?", {"a) Endor", "b) Tatooine", "c) Kashyyyk", "d) Naboo"}, 'c'},
16         {"5. Who is the bounty hunter who captures Han Solo?", {"a) Boba Fett", "b) Jango Fett", "c) Greedo", "d) Ic-88"}, 'a'},
17         {"6. What is the name of the Death Star's commander?", {"a) Grand Moff Tarkin", "b) Darth Vader", "c) Emperor Palpatine", "d) General Hux"}, 'a'},
18         {"7. Who is the mother of Luke and Leia?", {"a) Padme Amidala", "b) Shmi Skywalker", "c) Mon Mothma", "d) Ahsoka Tano"}, 'a'},
19         {"8. Who was Qui Gon Jins master?", {"a) Yoda", "b) Jar Jar Binks", "c) Count Dooku", "d) Obi Wan Kenobi"}, 'c'},
20         {"9. What is the name of the ice planet in 'The Empire Strikes Back'?", {"a) Hoth", "b) Endor", "c) Tatooine", "d) Dagobah"}, 'a'},
21         {"10. Who initialised Order 66?", {"a) Darth Vader", "b) Darth Sidious", "c) Darth Maul", "d) Mace Windu"}, 'b'},
22     };
23     return questions;
24 }
25

```

Figure 2: Questions.cpp

User Interaction

Once the connection between the server and the client has been established, the user will be prompted to enter their name:

Server code that receives the client's name:

```

void handleClient(SOCKET client_socket, const std::vector<Question>& questions) {
    char buffer[1024] = { 0 };

    // Receive client's name
    int valread = recv(client_socket, buffer, 1024, 0);
    std::string client_name;
    if (valread > 0) {
        client_name = std::string(buffer, valread);
        std::cout << "Client Has Joined The Server, Welcome " << client_name << std::endl;
    }
}

```

Figure 3: Server Code Receiving Clients Name

Client code sending the client's name:

```
// Prompt user for their name
std::string name;
std::cout << "Enter your name: ";
std::getline(std::cin, name);

// Send name to server
send(sockfd, name.c_str(), name.length(), 0);
std::cout << "Name sent\n";

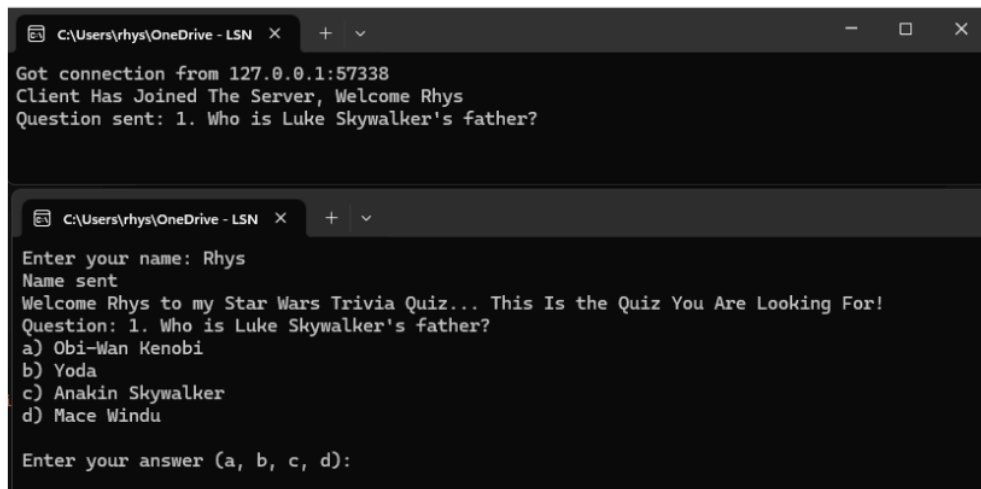
// Display welcome message
std::cout << "Welcome " << name << " to my Star Wars Trivia Quiz... This Is the Quiz You Are Looking For!\n";

// Handle trivia questions
handleTrivia(sockfd, name);

WSACleanup();
return 0;
}
```

Figure 4: Client Code Sending Client's Name

Expected Outcome:



```
C:\Users\rhys\OneDrive - LSN >
Got connection from 127.0.0.1:57338
Client Has Joined The Server, Welcome Rhys
Question sent: 1. Who is Luke Skywalker's father?

C:\Users\rhys\OneDrive - LSN >
Enter your name: Rhys
Name sent
Welcome Rhys to my Star Wars Trivia Quiz... This Is the Quiz You Are Looking For!
Question: 1. Who is Luke Skywalker's father?
a) Obi-Wan Kenobi
b) Yoda
c) Anakin Skywalker
d) Mace Windu
Enter your answer (a, b, c, d):
```

Figure 5: Client Name Outcome

Next, the client will be prompted with a question. The client has four options to choose from (one of them being the correct answer). As mentioned previously, the questions are stored in sperate files. The server will send the questions to the client and wait to receive the client's response, below are code snippets of this:

Server Sending Questions:

```
for (const auto& q : questions) {  
    // Send question and options to client  
    std::string message = q.question + "\n";  
    for (const auto& option : q.options) {  
        message += option + "\n";  
    }  
    if (send(client_socket, message.c_str(), message.length(), 0) == SOCKET_ERROR) {  
        std::cerr << "Send failed with error: " << WSAGetLastError() << std::endl;  
        closesocket(client_socket);  
        return;  
    }  
    std::cout << "Question sent: " << q.question << std::endl;
```

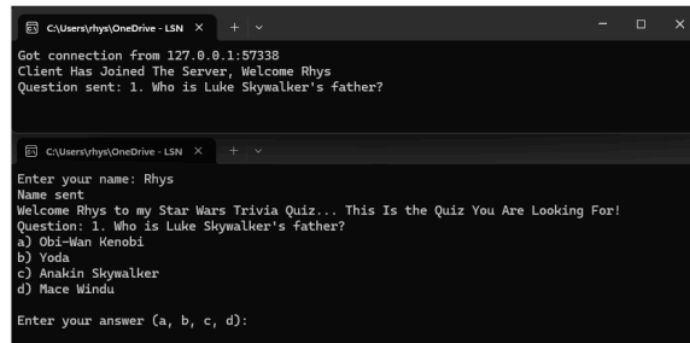
Figure 6:Server Sending Questions to Client

Client Receiving Questions:

```
void handleTrivia(SOCKET sock, const std::string& name) {  
    char buffer[1024] = { 0 };  
    int valread;  
  
    while (true) {  
        // Receive question and options from server  
        valread = recv(sock, buffer, 1024, 0);  
        if (valread <= 0) break; // Exit if no more questions
```

Figure 7: Client Receiving Questions

Expected Outcome:



```
C:\Users\rhys\OneDrive - LSN > +  
Got connection from 127.0.0.1:57338  
Client Has Joined The Server, Welcome Rhys  
Question sent: 1. Who is Luke Skywalker's father?  
  
C:\Users\rhys\OneDrive - LSN > +  
Enter your name: Rhys  
Name sent  
Welcome Rhys to my Star Wars Trivia Quiz... This Is the Quiz You Are Looking For!  
Question: 1. Who is Luke Skywalker's father?  
a) Obi-Wan Kenobi  
b) Yoda  
c) Anakin Skywalker  
d) Mace Windu  
Enter your answer (a, b, c, d):
```

Figure 8: Sending & Receiving Questions

Once the Client has received the question, the client will send a response to the server. Once the server has received the response, the server will send the client a message stating whether their answer is correct or incorrect:

Client Sending Their Response Code:

```
// Get answer from user
std::string answer;
while (true) {
    std::cout << "Enter your answer (a, b, c, d): ";
    std::cin >> answer;

    // Convert answer to lowercase for consistency
    std::transform(answer.begin(), answer.end(), answer.begin(), ::tolower);

    // Validate input
    if (answer == "a" || answer == "b" || answer == "c" || answer == "d") {
        break;
    }
    else {
        std::cout << "Invalid input, please try again.\n";
    }
}

// Send answer to server
send(sock, answer.c_str(), answer.length(), 0);
std::cout << "Answer sent\n";
```

Figure 9: Client Sending Their Response Code

Server Receiving the Response and Determining if the answer is correct or incorrect:

```

// Receive answer from client
valread = recv(client_socket, buffer, 1024, 0);
if (valread == SOCKET_ERROR) {
    std::cerr << "Client Has Left The Server: " << WSAGetLastError() << std::endl;
    closesocket(client_socket);
    return;
}

char client_answer = buffer[0];
std::cout << "Client answered: " << client_answer << std::endl;

// Determine if the answer is correct
std::string response;
if (client_answer == q.correct_option) {
    response = "Well done " + client_name + ", you got that answer correct!";
    score++;
    response += "\nYou received a point!";
}
else {
    response = "Unlucky " + client_name + ", that is the incorrect answer!";
    response += "\nYou did not receive a point.";
}

```

Figure 10: Server Receiving the Response and Determining if the answer is correct or incorrect

Once the server has determined if the answer provided by the user is correct or incorrect, the server will send a response to the client stating the result:

```

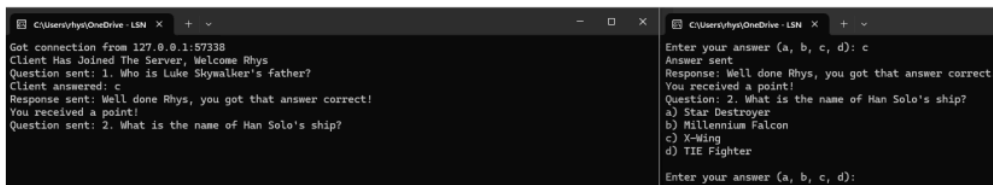
// Send response to client
if (send(client_socket, response.c_str(), response.length(), 0) == SOCKET_ERROR) {
    std::cerr << "Send failed with error: " << WSAGetLastError() << std::endl;
    closesocket(client_socket);
    return;
}

std::cout << "Response sent: " << response << std::endl;

```

Figure 11: Sending the Result To The Client

Expected Outcome:



```

C:\Users\rhys\OneDrive - LSN x + v
Got connection from 127.0.0.1:57338
Client Has Joined The Server, Welcome Rhys
Question sent: 1. Who is Luke Skywalker's father?
Client answered: c
Response sent: Well done Rhys, you got that answer correct!
You received a point!
Question sent: 2. What is the name of Han Solo's ship?

C:\Users\rhys\OneDrive - LSN x + v
Enter your answer (a, b, c, d): c
Answer sent
Response: Well done Rhys, you got that answer correct!
You received a point!
Question: 2. What is the name of Han Solo's ship?
a) Star Destroyer
b) Millennium Falcon
c) X-Wing
d) TIE Fighter
Enter your answer (a, b, c, d):

```

Figure 12: Expected Outcome For Sending Client Answer & Server's Response To The Answer

Scoring System

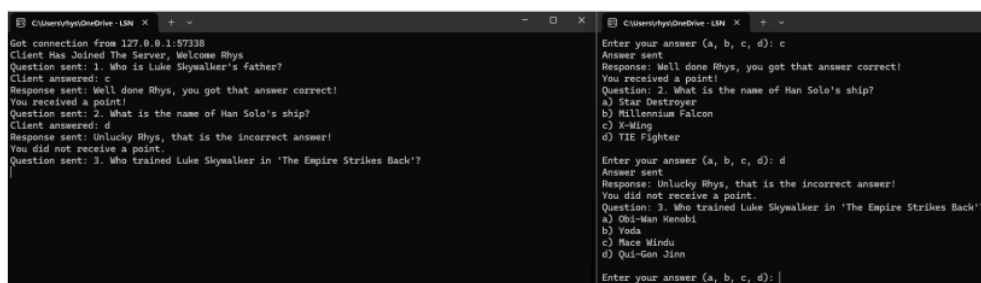
Each question is worth one point providing the answer is answered correctly, if the question is answered incorrectly the client will receive no points. Below is a snippet of code on how the scoring system works:

Server Code:

```
// Determine if the answer is correct
std::string response;
if (client_answer == q.correct_option) {
    response = "Well done " + client_name + ", you got that answer correct!";
    score++;
    response += "\nYou received a point!";
}
else {
    response = "Unlucky " + client_name + ", that is the incorrect answer!";
    response += "\nYou did not receive a point.";
}
```

Figure 13: Scoring System Server Side

After each question, the server will send a response to the client stating whether they received a point or not:



The figure shows two terminal windows side-by-side. The left window, titled 'C:\Users\rhys\OneDrive - USN x + v', shows the server's logs: 'Got connection from 127.0.0.1:57338', 'Client Has Joined The Server, Welcome Rhys', 'Question sent: 1. Who is Luke Skywalker's father?', 'Client answered: c', 'Response sent: Well done Rhys, you got that answer correct!', 'You received a point!', 'Question sent: 2. What is the name of Han Solo's ship?', 'Client answered: d', 'Response sent: Unlucky Rhys, that is the incorrect answer!', 'You did not receive a point.', 'Question sent: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?'. The right window, also titled 'C:\Users\rhys\OneDrive - USN x + v', shows the client's input and output: 'Enter your answer (a, b, c, d): c', 'Answer sent', 'Response: Well done Rhys, you got that answer correct!', 'You received a point!', 'Question: 2. What is the name of Han Solo's ship?', 'a) Star Destroyer', 'b) Millennium Falcon', 'c) X-Wing', 'd) TIE Fighter', 'Enter your answer (a, b, c, d): d', 'Answer sent', 'Response: Unlucky Rhys, that is the incorrect answer!', 'You did not receive a point.', 'Question: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?', 'a) Obi-Wan Kenobi', 'b) Yoda', 'c) Mace Windu', 'd) Qui-Gon Jinn', 'Enter your answer (a, b, c, d): |'.

Figure 14: Expected Outcome Scoring System

Once the client has answered all the questions, the server will send the user a final score statement which suggest the end of the game. Below are snippets of code showing how this works:

Server Side:

```

// Send final score to client
std::string final_score = "Your final score is: " + std::to_string(score);
if (send(client_socket, final_score.c_str(), final_score.length(), 0) == SOCKET_ERROR) {
    std::cerr << "Send failed with error: " << WSAGetLastError() << std::endl;
    closesocket(client_socket);
    return;
}

std::cout << "Final score sent: " << final_score << std::endl;

closesocket(client_socket);
}

```

Figure 15: Server Sending Final Score to Client

Client Side:

```

// Check if the message is the final score
if (std::string(buffer).find("Your final score is:") != std::string::npos) {
    std::cout << buffer << std::endl;
    break; // Exit the loop after displaying the final score
}

```

Figure 16: Final Score Client Side

Expected Outcome:

```

C:\Users\pdp\OneDrive - LHM > .\
Not connection from 127.0.0.1:57330
Client has joined the Server. Welcome Rhye
Question sent: 1. Who is Luke Skywalker's father?
Client answered: c
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 2. What is the name of Han Solo's ship?
Client answered: d
Response sent: Well done Rhye, that is the incorrect answer!
You did not receive a point.
Question sent: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?
Client answered: b
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 4. What is the name of the Wookiee home planet?
Client answered: c
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 5. Who is the bounty hunter who captures Han Solo?
Client answered: a
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 6. What is the name of the Death Star's commander?
Client answered: a
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 7. Who is the mother of Luke and Leia?
Client answered: a
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 8. Who was Qui Gon Jinn master?
Client answered: c
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 9. What is the name of the ice planet in 'The Empire Strikes Back'?
Client answered: a
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Question sent: 10. Who initialised Order 66?
Client answered: b
Response sent: Well done Rhye, you got that answer correct!
You received a point!
Final score sent: Your final score is: 9

```

Figure 17: Expected Outcome Final Score

Design Decisions & Best Practices

Language

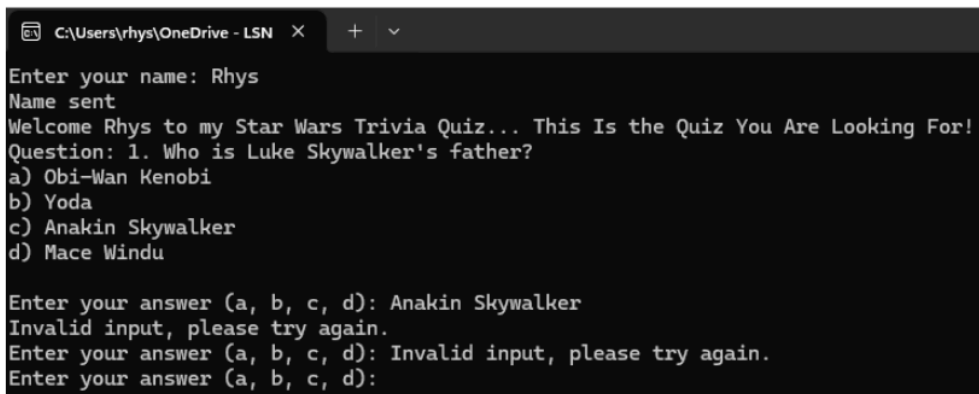
The chosen language to create the networked trivia quiz was C++. C++ is a fast speed and low-level memory management language which is ideal for building applications [1].

Code Organisation

The code is split into different files which makes things easier to manage. By separating the server side and client side into separate solutions/projects in Visual Studio 2022 it made it easier to develop. As mentioned in the features section, the questions were stored in their own files. This is more practical than having it all in one file.

Error Handling/Input Validation

When the client is answering the questions, the developer needed to ensure that the answers received from the client were valid. For example, if the client was to answer question 1 and enter something different from a,b,c and d, the client will receive a message stating that the answer is invalid please try again:



```
C:\Users\rhys\OneDrive - LSN X + v
Enter your name: Rhys
Name sent
Welcome Rhys to my Star Wars Trivia Quiz... This Is the Quiz You Are Looking For!
Question: 1. Who is Luke Skywalker's father?
a) Obi-Wan Kenobi
b) Yoda
c) Anakin Skywalker
d) Mace Windu

Enter your answer (a, b, c, d): Anakin Skywalker
Invalid input, please try again.
Enter your answer (a, b, c, d): Invalid input, please try again.
Enter your answer (a, b, c, d):
```

Figure 18:Error Handling Example 1

Also, the developer allowed the client to enter uppercase and lowercase letters to answer the questions:

```
C:\Users\rhys\OneDrive - LSN x + v
Enter your name: Rhys
Name sent
Welcome Rhys to my Star Wars Trivia Quiz... This Is the Quiz You Are Looking For!
Question: 1. Who is Luke Skywalker's father?
a) Obi-Wan Kenobi
b) Yoda
c) Anakin Skywalker
d) Mace Windu

Enter your answer (a, b, c, d): c
Answer sent
Response: Well done Rhys, you got that answer correct!
You received a point!
Question: 2. What is the name of Han Solo's ship?
a) Star Destroyer
b) Millennium Falcon
c) X-Wing
d) TIE Fighter

Enter your answer (a, b, c, d): B
Answer sent
Response: Well done Rhys, you got that answer correct!
You received a point!
Question: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?
a) Obi-Wan Kenobi
b) Yoda
c) Mace Windu
d) Qui-Gon Jinn

Enter your answer (a, b, c, d):
```

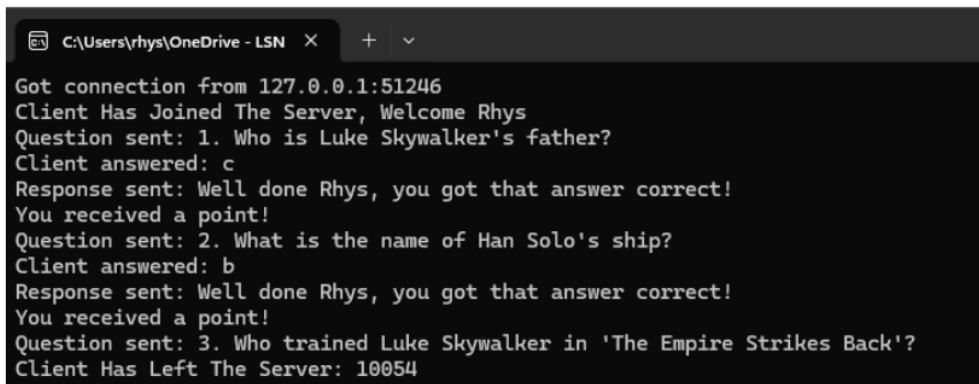
Figure 19: Error Handling Example 2

Once the client has joined the server, the server will process a message stating the client (name) has joined the server. Below is a snippet of this:

```
C:\Users\rhys\OneDrive - LSN x + v - C:\Users\rhys\OneDrive - LSN x
Got connection from 127.0.0.1:51246
Client Has Joined The Server, Welcome Rhys
Enter your name: Rhys
Name sent
```

Figure 20: Client Joining Server Message (Server)

Also, if the client unexpectedly left or disconnected from the server, the server will display a statement:



```
C:\Users\rhys\OneDrive - LSN X + v
Got connection from 127.0.0.1:51246
Client Has Joined The Server, Welcome Rhys
Question sent: 1. Who is Luke Skywalker's father?
Client answered: c
Response sent: Well done Rhys, you got that answer correct!
You received a point!
Question sent: 2. What is the name of Han Solo's ship?
Client answered: b
Response sent: Well done Rhys, you got that answer correct!
You received a point!
Question sent: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?
Client Has Left The Server: 10054
```

Figure 21: Client Disconnecting From The Server

By including error handling will make the application more user friendly for the client. Without having any error messages, the client will be confused if something was entered wrong etc.

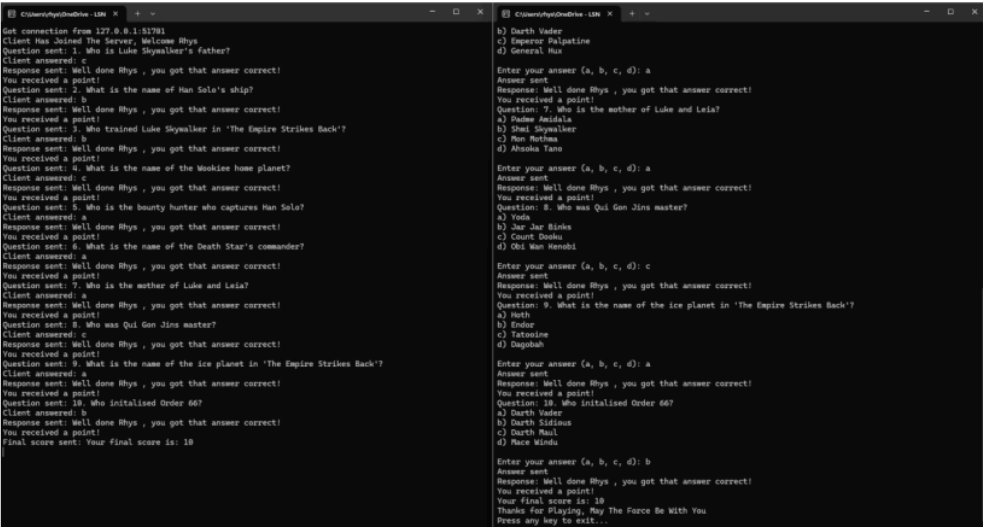
Testing

The application was thoroughly tested by the developer to begin with, once the developer was happy with the first version other clients tested the application. The feedback suggested the implementation of a scoring system to make the tracking of correct answers easier. The developer implemented the recommendation which improved the application for the clients.

User Experience

The purpose of the application is for the client to enjoy the game and have fun with it. Hence why the scoring system was implemented into the application. By having a

scoring system more than one player (on one client connection) can play the game and contest to see who has the more knowledge on the chosen topic.



```
Get connection from 127.0.0.1:51791
Client has joined The Server, Welcome Rhys
Question sent: 1. Who is Luke Skywalker's father?
Client answered: c
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 2. What is the name of Han Solo's ship?
Client answered: b
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 3. Who trained Luke Skywalker in 'The Empire Strikes Back'?
Client answered: b
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 4. What is the name of the Wookiee home planet?
Client answered: c
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 5. Who is the bounty hunter who captures Han Solo?
Client answered: a
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 6. What is the name of the Death Star's commander?
Client answered: a
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 7. Who is the mother of Luke and Leia?
Client answered: a
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 8. Who was Qui Gon Jinn master?
Client answered: c
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 9. What is the name of the ice planet in 'The Empire Strikes Back'?
Client answered: a
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Question sent: 10. Who initialised Order 66?
Client answered: b
Response sent: Well done Rhys , you got that answer correct!
You received a point!
Final score sent: Your final score is: 10
```

```
b) Darth Vader
c) Emperor Palpatine
d) General Hux

Enter your answer (a, b, c, d): a
Answer sent
Response: Well done Rhys , you got that answer correct!
You received a point!
Question: 7. Who is the mother of Luke and Leia?
a) Shmi Skywalker
b) Shmi Skywalker
c) Han Solo
d) Ahsoke Tano

Enter your answer (a, b, c, d): a
Answer sent
Response: Well done Rhys , you got that answer correct!
You received a point!
Question: 8. Who was Qui Gon Jinn master?
a) Yoda
b) Jar Jar Binks
c) Count Dooku
d) Obi Wan Kenobi

Enter your answer (a, b, c, d): c
Answer sent
Response: Well done Rhys , you got that answer correct!
You received a point!
Question: 9. What is the name of the ice planet in 'The Empire Strikes Back'?
a) Hoth
b) Endor
c) Tatooine
d) Dagobah

Enter your answer (a, b, c, d): a
Answer sent
Response: Well done Rhys , you got that answer correct!
You received a point!
Question: 10. Who initialised Order 66?
a) Darth Vader
b) Darth Sidious
c) Darth Maul
d) Nace Winda

Enter your answer (a, b, c, d): b
Answer sent
Response: Well done Rhys , you got that answer correct!
You received a point!
Your final score is: 10
Thanks for Playing, May The Force Be With You
Press any key to exit...
```

Figure 22: Completed Trivia Quiz (Final Score)

Conclusion

In conclusion, the application successfully provides a trivia quiz experience to the clients. However, there are areas that could be improved on. The first improvement

would be to allow more than one connection onto the server. This would allow more than one client to play the game. The application does comply with best practices especially in areas such as code organisation and error handling. Overall, the implementation process was successful.

References

[1] GeeksForGeeks, "C++ Programming Language," GeeksForGeeks, 15 04 2025.
[Online]. Available: <https://www.geeksforgeeks.org/c-plus-plus/>. [Accessed 12 05 2025].

Assignment 2 Report

GRADEMARK REPORT

FINAL GRADE

GENERAL COMMENTS

A nice trivia game application, well considered and implemented. While not the most complex application, it's fairly complete and overall well designed. A good effort.

72/100

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

