CHARLESTON SOUTHERN UNIVERSITY Term Project Task 1 **CSCI332**

Computer and Information Sciences (CSCI)

League: An Online Game

Introduction

This lab introduces how to use UDP/IP to send and receive data between computers in a local network. This also demonstrates the use of threads to send and receive data so that the user interface doesn't get stuck while sending and receiving data.

Background

This program uses SDL2 for graphics and **linux UDP socket** to send and receive data. The main advantage of UDP is that it works three times faster than TCP. One of the problems of UDP is that it is not reliable and data loss can occur. Data loss in LAN environment is rare but it may happen. We will not deal with it in this project. We will also use P2P architecture to build this game.

One of the main problems of pure P2P is that there is no centralized server, players need to find each other inorder to join the game. To overcome this problem, we will broadcast a UDP packets to all players in the local area network.

The first step of broadcasting packets to local area network is to find the broadcast IP. The broadcasting of data packets is the same as sending data packets to another computer except we use a LAN broadcast IP. For example, if your computer has IP of 192.168.200.123 and the netmask is 255.255.255.0, then the broadcast IP will be 192.168.200.255. After the broadcast IP is found, we will then use broadcast IP for communication instead of host IP.

P2P needs to be able to send and receive packets at the same time. It will be almost impossible to receive and send at the same time if the program runs linearly, instead, multi-threads is required. Furthermore, the program is a game, the program needs to intercept I/O inputs like mouse and keyboard events and update graphics. All these tasks need to run at the same time thus you will need to create a thread for each task. Fortunately, I have designed and created threads for you. All what you need to do for this project is to complete the following sections:

Task I.(30%)

1. Download the sample code from Blackboard. Make sure the firewall setting allows your program to access network, otherwise you will not be able to receive and send messages.
2. Install the required library for SDL2 as what follows:

```
sudo apt-cache search libsdl2
sudo apt-get install libsdl2-dev
```

sudo apt-get install libsdl2-image-dev
sudo apt-get install libsdl2-ttf-dev

3. You will need to create a Final Project Task 1 page on your webserver and present answers on it.

Part A. (10%) Extracting Tokens

In this problem, you are going to demonstrate the understanding of how strtok works in c++ under **Ubuntu**.

1. Create a c++ Console program.
2. Allow the user to enter a string.
3. Use strtok to split the string into a vector.
4. Use a loop to display elements in the vector.
5. On your website, explain how strtok works. Show screen shots of your program running with inputs and proper outputs.

Part B. (10%) Finding Tokens

In this problem, you are going to demonstrate the understanding of how to find a specific token in a token vector in c++ under **Ubuntu**.

1. Based on Part A, create a C++ console program.
2. Allow the user to enter a string.
3. Use Part A to split the string into a vector.
4. Allow the user to enter a keyword.
5. Use range based for loop to find the keyword in the vector. Specifically, use
   for (auto an_iterator : a_vector) and *strncmp()*
   where an_iterator is an iterator of a_vector and strncmp allows you to compare two strings.
6. On your website, explain how you find a keyword. Show screen shots of your program running with inputs and proper outputs.

Part C. (10%) Constructor and vectors

In this problem, you are going to demonstrate how to dynamically add an object into a vector.

1. Create a c++ Console program.
2. Download *Test_Constructor.h* and include *Test_Constructor.h* in your program.
3. Use a for loop from step 2 to step 4
4. In each iteration, asks for Name, Role, X, Y, Health and use these values to create a character object.
5. Add the object into a vector.

6. After 3 objects is added into the vector, use another loop to display the values of each object in a vector.
7. On your website, explain how to add an object into a vector. Show screen shots of your program running with inputs and proper outputs.

Part D. (10%) Update objects in a vector

In this problem, you are going to demonstrate how to replace values of an objects in a vector.

1. Based on Part C, create a C++ console program.
2. Create three objects and add them into a vector.
3. Ask for a Name and then use the name to find the object in the vector.
4. Ask for new Role, X, Y, and Health.
5. Update the object with these new values.
6. If the user enters *Quit!*, exit your program.
7. On your website, explain how to find an object in a List. Show screen shots of your program running with inputs (including *Quit!*) and proper outputs.

Part E. (10%) Broadcasting Message

In this problem, you are going to demonstrate how to broadcast messages in C++. Use the sample Code from class.

1. Based on Part D and create a c++ Console program.
2. Add the code we did in class to this part and make changes so you are able to broadcast to the LAN where your computer is connected to.
3. Send all three player's name, role, x, y, health to your partner's server in a loop with
   for (auto an_iterator : a_vector)
4. Show screen shots on your website.

Part F. (50%) Partner Evaluation

After completing the group project task, you will evaluate your partner according to the rubric on the next page. Compress the form and report documents (in web page format) into a single compressed file and submit to Blackboard.

If you have more than one partner, use an evaluation sheet for each partner. You must use my form attached at the end of the handout! Failed to do so, you and your partner will receive 0 for evaluation.

What to turn in?

1. The zipped whole project folder including webpages. Make sure it is executable when you decompress it on another computer.
2. Your team's demonstration. Show me how you operate your program.
3. You must demonstrate both you and your partner are able to receive and update characters on screen
4. Submit a separate

Demonstration

Online students: You are going to record your project and go through each part of the task. Clearly demonstrate what I ask for. If I feel like you are skimming some parts, I will assume you either do not understand or did not implement the part then you will receive a 0 for it. Both you and your partner must connect to VPN and play game at the same time to demonstrate your program is able to send and receive messages.

Face-to-Face students: I will use a grading sheet to go through each part of the task at your seat. Please sit with your partner and have both computers connected before grading.

Evaluation Sheet

Please indicate how many points your partner receives in each category.

Partner Name: ____________________

	Below Expectations (1-3)	Average (4-6)	Exceeds Expectations (7-8)
1. Produces research information for team.			
2. Demonstrates understanding of team roles when assigned.			
3. Shares in work of team and demonstrates time management skills.			
4. Demonstrates good listening skills.			
5. Demonstrates understanding of the qualities of a successful team and explain why those qualities are important to success			

Comment: