CS F303 : COMPUTER NETWORKS

MINI PROJECT

on

"Simulation of multilevel card game (Hearts[2]) over TCP network"

By

Group 6

Pranjal Gupta	2013B4A7470P
Rishabh Jain	2013B3A7473P
Neel Kasat	2013B3A7670P
Akshit Bhatia	2013B3A7722P

February 28, 2017

Table of Contents

1. Problem Statement

Conceptualization, Specification, Design and Implementation of a multi-level game where 4 players will play the game of Hearts. The game will have 3 rounds of play with increasing difficulty, each round having subtle variations in the traditional game play.

2. Introduction

2.1 Purpose

The purpose of this project is to implement a multi-client game of Hearts over TCP network. The game can be played 4 players which will connect to a Host server. The host server synchronizes and alternates between the user(client) activities. Each client should communicate to the server with the current state of its player. The client will prompt its user to take an action (e.g. throw a card) and will communicate the user action to the server. The server should check if the move from a user (through client) is feasible and legit, and consequently continues the same with other clients. The server should also keep scores and communicate the actions of other client to a client. The client shall have a basic user interface which will display scores and actions of other users, as required.

2.2 Scope

The software will be the simulation of the famous card game called Hearts. This game can be played among a group of 4 users(clients) and will be co-ordinated by a server over a TCP network. The software also extends the basic version of the game with increasing difficulty levels which introduces subtle changes to the game rules with its progress.

2.3 Definitions, Acronyms and Abbreviations

1. Server : The host that will synchronize and coordinate between the activities of the client. This will server as an always-on host to which the client will connect.
2. Client : The end-user program which will connect to the server to start playing. The client displays current status of the game to the user and also takes and communicates user responses to the server when prompted by the server.
3. TCP : Transfer control protocol. The underlying protocol to be used by the server and client applications.
4. Socket : It provides a programming interface to a TCP port. The client/server communicates over TCP through a specific port which can be accessed via a socket in C.

2.4 References

Refer to Appendix [6.1]

2.5 Overview

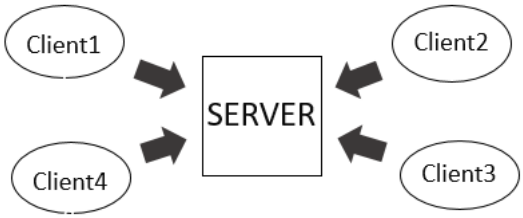
The rest of the document contains the description of the project (including its functions, user characteristics, dependencies) and its other specific requirement. The project as a whole has been

divided into 5 logical modules which describes the stages of the implementation. The designing and implementation of the entire project is planned to be undertaken module-wise in a period 35 days (13th March to 16th April). A tentative weekly plan of development along with module specification and ownership is given, followed by the Appendix.

3. Overall Description

3.1 Product Perspective

The project consists of a server and a client interface, which forms a standalone application communicating to each other from different hosts over the network through TCP connection. The server is an always-on host to which 4 clients can connect to play the game of Hearts together. The client communicates with the server to enforce an interface between other users through their clients.



3.1.1 System Interface

NA

3.1.1 User Interface

- The server application will have CLI-based interface for logging activities about the state of play. The logs will be used to debug and monitor the progress, and also report errors arising during the runtime.
- The client interface will display the activities of all the users and the score at a time in game. The interface will also be used to read user responses when required by the server.

3.1.3 Hardware Interface

The server application uses standard output to display logs. The client application uses the standard output to interact with the user. The user action is read using standard input. This input and output is handled by the standard libraries of C.

3.1.4 Software Interface

Our software will be developed for the Linux-based Operating Systems. It will be programmed in C and we will be making use of network specific C libraries and the standard C library. The code will be compatible with GCC version 5.4.0 (on Debian Ubuntu version 16.04). At this stage, we intend to work on a Linux based platform only for an ease of implementation.

3.1.5 Communication Interface

The server and client applications will utilize the underlying Network Card (WiFi or Ethernet etc.) to connect to the internetwork. Applications will communicate using the Transmission Control Protocol (TCP) over network.

3.2 Product Functions

Functionalities of the server application :

a) Accepts the connection of the clients, up till 4 clients and initiate a game between them.
b) Maintains synchronization of activities among the clients.
c) Provides clients with the information about the current state of play and scores.
d) Error handling, in case of erratic response or connection drop of any of its client.

Functionalities of the client application :

a) Connects to the server application when the user prompts.
b) Communicates the user action to the server when prompted.
c) Displays the current state of the game and score to the user.

4 users through their client interface will be competing against each other in a single game hosted over the server. The game will have 3 levels of increasing difficulty :

- **Level 1 :** simple game implementation with spade queen having 13 points, all hearts cards carrying 1 point and rest other with 0 points. Each user throws a card in each round and prevents acquiring cards with point, to keep their score minimum.
- **Level 2 :** each user will be allowed to give 3 of its cards to its adjacent user. Hence, the users have the option to shed their worst cards.
- **Level 3 :** all the queens will carry variable points depending upon a scoring scheme.

3.3 User Characteristics

The user is expected to have the basic knowledge of cards and the game, Hearts. For the further levels, the required instructions will be provided at appropriate time in game.

3.4 Constraints

- Synchronization Problems due to timing issues on the existing network infrastructure.
- Handling Connection drops due to coagulation on network.

4. Specific Requirements

4.1 External Interface Requirements

NA

4.2 Performance Requirements

The game will require proper synchronisation between the turns of the player , handling connection drops. Static numerical requirements for the game are:

a) 4 users have to play at a time on different machines
b) 4 terminal connected on a server
c) Score of each user, each player move, handling game levels
d) Each user has to wait for the user to complete his turn

4.3 Design Constraints

NA

4.4. Software System Attributes

Since this is only a preliminary game, we are only focusing on synchronization amongst multiple users and handling shared memory for real time game experience which will enhance user experience.

5. Weekly Plan of Work And Module Wise Ownership details

S.No.	DESCRIPTION	MODULE	EST. TIME	MODULE OWNERSHIP
1.	BLUEPRINT OF ALL COMMUNICATION BETWEEN SERVER AND CLIENT [1]	1	13th March to 15th March	Akshit
2.	CLIENT-SERVER ARCHITECTURE : BASIC BUILD UP (WITH SERVER HANDLING MULTIPLE CLIENTS)	2	16th March to 22nd March	Neel
3.	SPECIFIC FUNCTIONS FOR CLIENT AND SERVER FOR IMPLEMENT-ATION OF GAME LOGIC	3	23rd March to 9th April	Pranjal
MID SEMESTER DEMO				
4.	ACTION SYNCHRONIZATION BETWEEN CLIENTS AND SERVER	4	27th March to 9th April	Rishabh
5.	SHARED MEMORY FOR CLIENT SERVER INTERACTION AND BASIC USER INTERFACE	5	10th April to 16th April	Akshit
END SEMESTER DEMO AND SUBMISSION				

6. Appendix

6.1 References

[1]http://www.dreamincode.net/forums/topic/313244-multiple-clients-single-server-fork-sockets/

[2] Description of traditional Hearts game : https://en.wikipedia.or g/wiki/Hearts

[3] cgui.sourcefor ge.net/

[4] Network libraries in C : https://www.gnu.org/software/libc/manual/ html_node/Sockets.html