DAYANANDA SAGAR UNIVERSITY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
SCHOOL OF ENGINEERING
DAYANANDA SAGAR UNIVERSITY
KUDLU GATE
BANGALORE - 560068

MINI PROJECT REPORT

ON

"TicTacToe using TELNET Protocol (java socket programming)"

SUBMITTED TO THE 6TH SEMESTER
UNIX SYSTEMS PROGRAMMING -2020

BACHELOR OF TECHNOLOGY
IN
COMPUTER SCIENCE & ENGINEERING

Submitted by

K.HIMABINDU - (ENG17CS0109)
KIRAN KUMAR M- (ENG17CS0110)
KIRAN M - (ENG17CS0111)
KULSUM KHAN - (ENG17CS0112)

Under the supervision of

Prof. Shivaprasad Ashok Chikop, Assistant Professor

DAYANANDA SAGAR UNIVERSITY

School of Engineering, Kudlu Gate, Bangalore-560068

CERTIFICATE

This is to certify that Ms. K.Himabindu bearing USN ENG17CS0109, Mr. Kiran kumar M bearing USN ENG17CS0110 , Mr. Kiran M S bearing USN ENG17CS0111, Ms. Kulsum Khan bearing USN ENG17CS0112 has satisfactorily completed his/her Mini Project as prescribed by the University for the semester B.Tech. programme in Computer Science & Engineering during the year at the School of Engineering, Dayananda Sagar University., Bangalore.

*Date:*_____

_____

Signature of faculty in-charge

Max marks	*Marks obtained*

_____

Signature of Chairman
Department of Computer Science and Engineering.

DECLARATION

We hereby declare that the work presented in this mini project entitled-"TicTacToe using TELNET PROTOCOL ", has been carried out by us and it has not been submitted for the award of any degree,diploma or the mini project of any other college or university.

K.HIMABINDU - (ENG17CS0109)
KIRAN KUMAR M- (ENG17CS0110)
KIRAN M - (ENG17CS0111)
KULSUM KHAN - (ENG17CS0112)

ACKNOWLEDGEMENT

The satisfaction that accompanies the successful completion of a task would be incomplete without the mention of the people who made it possible and whose constant guidance and encouragement crown all the efforts with success.

We are especially thankful to our **Chairman ,Dr. M K Banga,** for providing necessary departmental facilities, moral support and encouragement.

We are very much thankful to **Prof. Shivaprasad Ashok Chikop, Assistant Professor ,** for providing help and suggestions in completion of this mini project successfully.

We have received a great deal of guidance and co-operation from our friends and we wish to thank all that have directly or indirectly helped us in the successful completion of this project work.

Table Of Contents

Table of Figures

ABSTRACT

The aim of this project is to demonstrate a part of RFC 854 which implements the Telnet Protocol. The Telnet Protocol is a standard network protocol used to provide a bidirectional interactive text-oriented communication. The term telnet is also used to refer to the software that implements the client part of the protocol. Telnet client applications are available for virtually all computer platforms. Telnet is also used as a verb. To telnet means to establish a connection using the Telnet protocol, either with a command line client or with a graphical interface.

I. INTRODUCTION

TELNET (TELecommunication NETwork) is a network protocol used on the Internet or local area network (LAN) connections. It was developed in 1969 beginning with RFC 15 and standardized as IETF STD 8, one of the first Internet standards. It is a network protocol used on the Internet or local area networks to provide a bidirectional interactive communications facility. Typically, telnet provides access to a command-line interface on a remote host via a virtual terminal connection which consists of an 8-bit byte oriented data connection over the Transmission Control Protocol (TCP). User data is interspersed in-band with TELNET control information. The user's computer, which initiates the connection, is referred to the local computer.

The computer being connected to, which accepts the connection, is redirected to as the remote computer. The remote computer can be physically located in the next room, the next town or in another country.The network terminal protocol (TELNET) allows a user to log in on any other computer on the network. We can start a remote session using client/server programs that allow a user on the client site to log into the computer at the server site and use the services available there. To use TELNET we are using java network programming/ Socket programming to write a TicTacToe program that executes across multiple devices (computers), in which the devices are all connected to each other using a network.

Applications:

The java.net package of the J2SE APIs contains a collection of classes and interfaces that provide the low-level communication details, allowing you to write programs that focus on solving the problem at hand.

The java.net package provides support for the two common network protocols which are a part of telnet −

TCP − TCP stands for **Transmission Control Protocol,** which allows for reliable communication between two applications. TCP is typically used over the Internet Protocol, which is referred to as TCP/IP.

UDP − UDP stands for **User Datagram Protocol,** a connectionless protocol that allows for packets of data to be transmitted between applications.

II. PROBLEM STATEMENT

To build a telnet interface that allows the user to play a game (tic tac toe).The application will use a client-server paradigm. It listens for two clients to connect, and spawns a thread for each: the first is Player X and the second is Player O. The client and server send simple string messages back and forth to each other messages correspond to the Tic Tac Toe protocol. The server provides each client with the total information of the game for each move the text is displayed on the terminal for both the clients.

III. OBJECTIVE

The main aim of the project is to establish a bi-direction server/client connection between 2 terminals or 2 devices so TicTacToe can be played as a 2 player game. This is established using java socket programming(java.net) as the programming language. We use a socket in Java which is one endpoint of a two-way communication link between two programs running on the network. An endpoint is a combination of an IP address and a port number. .

IV. METHODOLOGY

IV. I. Block Diagram

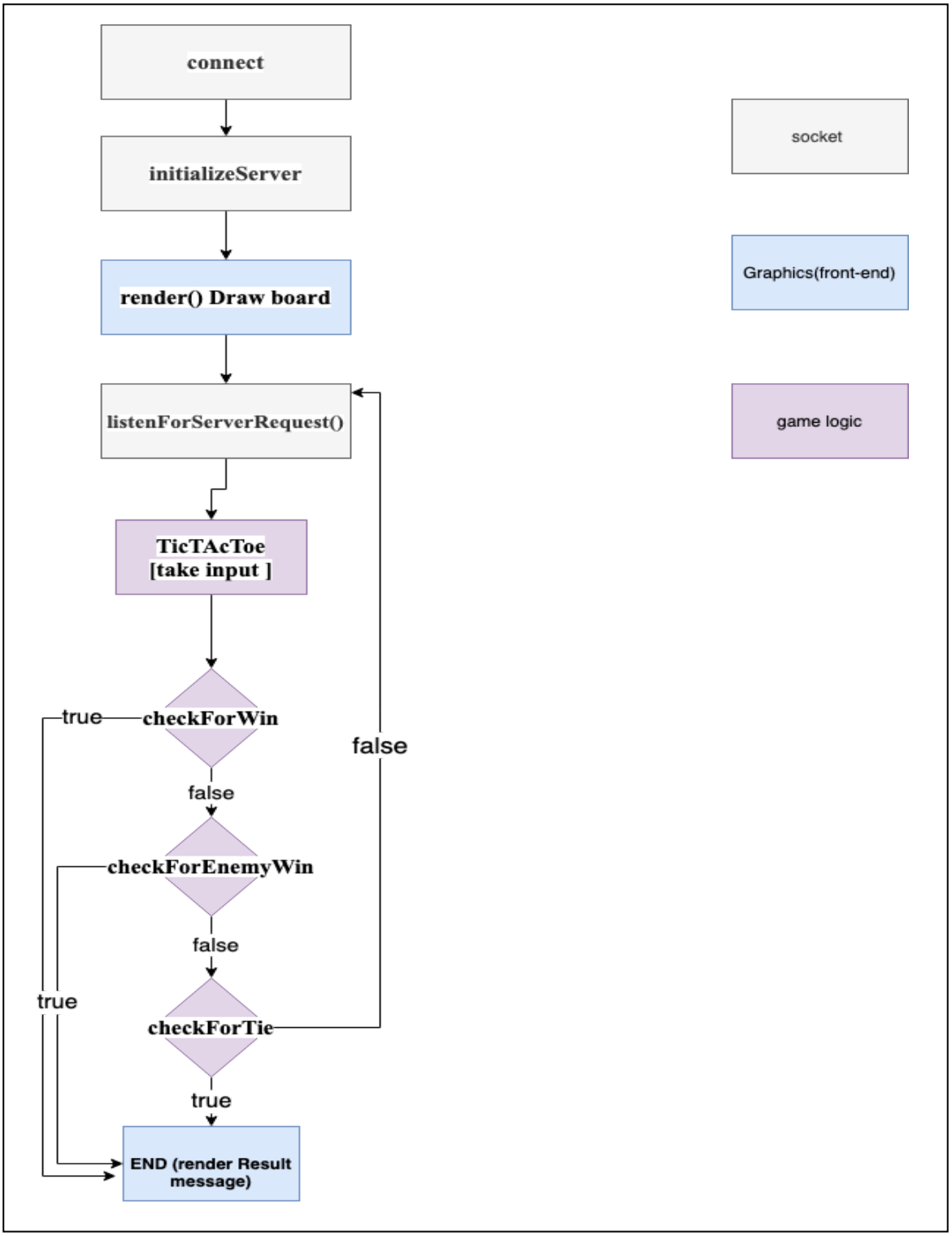

Fig 4.1 - block Diagram

IV.II .Proposed Work

Front end or displaying the board:

For the "console" version into a "graphics" version - a Java Swing application, is used as illustrated. In this initial design, we do not separate the cell and board into dedicated classes, but include them in the main class. We used an inner class DrawCanvas (that extends JPanel) to do the custom drawing, and an anonymous inner class for MouseListener.

The content-pane (of the top-level container JFrame) is set to BorderLayout. The DrawCanvas (JPanel) is placed at the CENTER; while a status-bar (a JLabel) is placed at the SOUTH (PAGE_END).

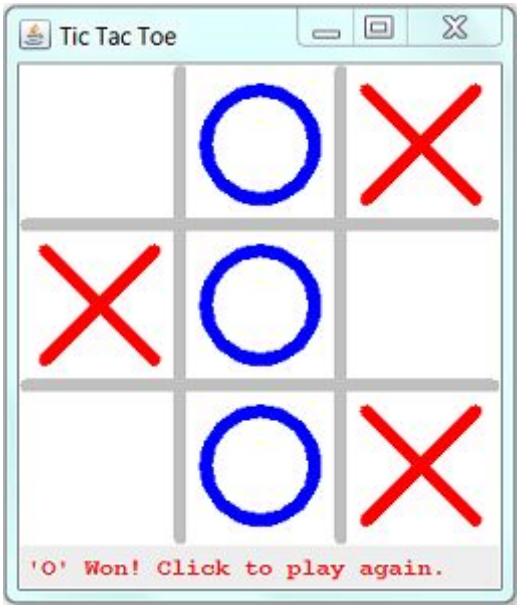

Fig 4. 2 - The graphics used for front end

Methods and Packages Used:

1. **Frame package** is used to display the window frame.
    a. **render()** : draw the board if the connection is established.There are 2 parts of the drawing:
    b. When the game is going on the board is displayed, and whenever the players make moves, it is displayed simultaneously on both screens.
    c. When the game ends: if one of the players wins or it's a tie, the board grid is removed and the corresponding message is displayed.
2. **loadImages()** : is used to load the basic shape used to to draw the board and the X and O in the colors red and blue.

3. **Painter()** : is used to set focus, background and add a 'this' pointer to make mouse movements on the screen.
4. **mousePressed()** : is used to track mouse movements.

BACKEND : Establishing a connection

The TicTacToe function is used to establish a connection. For a device to be able to connect, it needs to have a JRE . It will ask for IP and a port number, after which it will connect if both are correct. For every move a player makes, it shows on both the device screens. The socket is waiting for every key stroke by checking if the variable unableToCommunicatWithOpponent is True or not.

As long as the connection is working , it takes moves as input and checks the tick() function.

Methods:

1. **TicTacToe()** : It will ask for IP and a port number, after which it will connect if both are correct.
2. **run():** if tick is true, the run method uses listenForServerRequest method to keep taking in input until the game ends.
3. **connect()** : uses public Socket(InetAddress host, int port) throws IOException. This method attempts to connect to the specified server at the specified port. If this constructor does not throw an exception, the connection is successful and the client is connected to the server. The host is denoted by an InetAddress object.
4. **listenForServerRequest() :** used Socket accept() which waits for an incoming client. This method blocks until either a client connects to the server on the specified port or the socket times out, assuming that the time-out value has been set using the setSoTimeout() method. Otherwise, this method blocks indefinitely.
5. **initializeServer():** uses public ServerSocket(int port, int backlog, InetAddress address) throws IOException which Attempts to create a server socket bound to the specified por. The InetAddress parameter specifies the local IP address to bind to. The InetAddress is used for servers that may have multiple IP addresses, allowing the server to specify which of its IP addresses to accept client requests on.

BACKEND : tictactoe

To play the game, the program checks the Tick() method if the player is O or X. accordingly decides the game. Methods used are:

1. **tick** : If yourTurn and unableToCommunicateWithOppnonent is not True, it takes the move of the player, checks for player's win or the opponent's win or checks if there is a tie and BREAKS else continues.
2. **checkForWin:** if the Player is O, check if there is a line or cross with 3 O's , else if there is a line or cross with 3 X'x, if True the player WINS and the game ends

3. **checkForEnemyWin**:if the Player is O, check if there is a line or cross with 3 X's , else if there is a line or cross with 3 O'x, if True the opponent WINS and the game ends.
4. **checkForTie**: if the board is filled with no 3 consecutive O's or X's in a line or cross, then it results in a tie.

V. SOFTWARE REQUIREMENT

5.1.Back-end

5.1.2. JAVA socket programming :

- Java.io packages
  - DataInputStream
  - DataOutputStream
  - IOException
- Java.net packages
  - InetAddress
  - ServerSocket
  - Socket

5.2. Front-end

5.2.1. JAVA socket programming

- Javax packages
  - ImageIO
  - JFrame
  - JPanel
- Java.awt packages :
  - BasicStroke
  - Color
  - Dimension
  - Font
  - Graphics
  - Graphics2D
  - RenderingHints
  - Toolkit
  - event.MouseEvent
  - event.MouseListener
  - image.BufferedImage

VI. RESULTS

We open both the server and client to the left and right respectively by running the program in 2 terminals.It asks for the host ip and port number:

1.

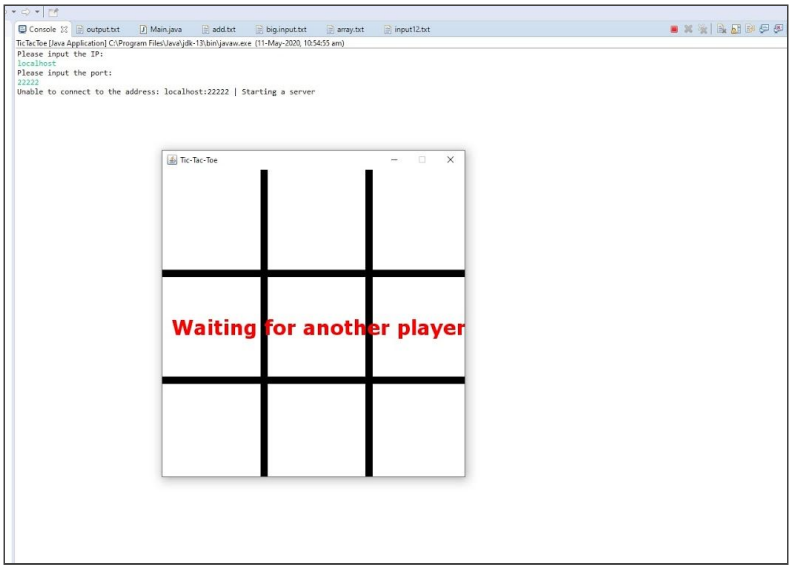

Fig 6.1 - server connects

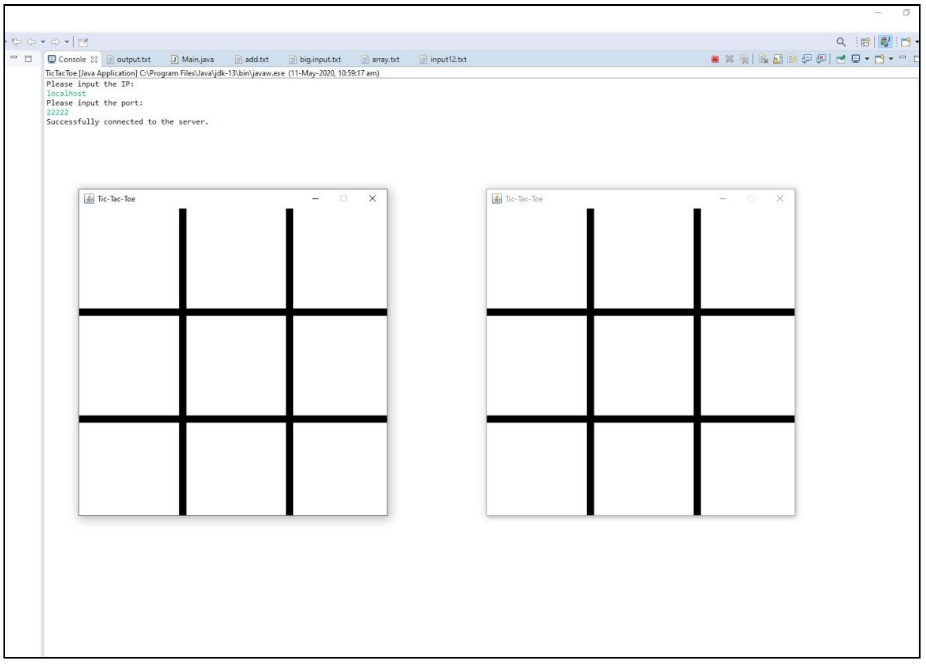

Fig 6.2 - client connects, connection establishes

2.Test case 1 : player 1 wins

As player one gets a 3 consecutive X's in a line , it wins

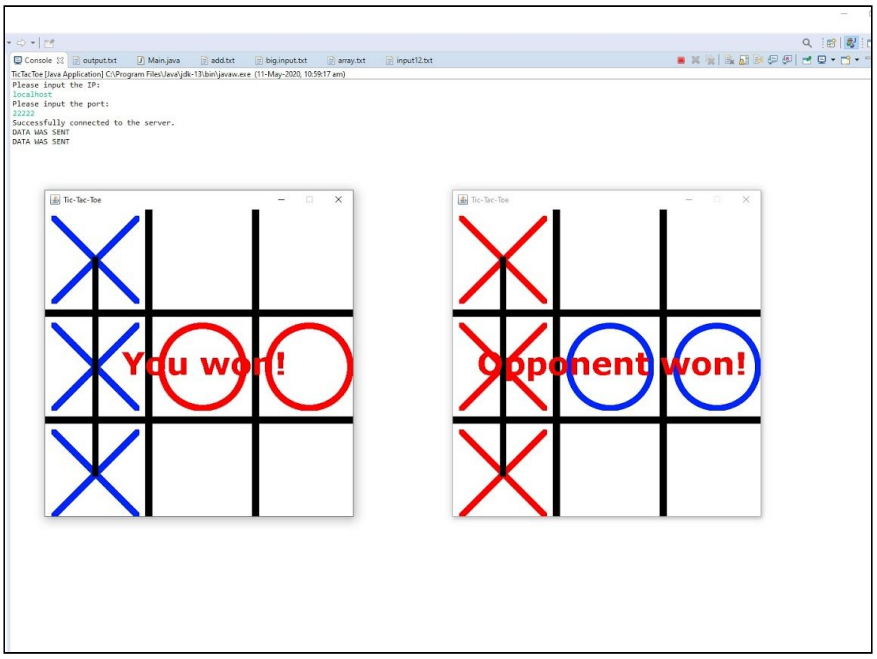

Fig 6.3 - Player 1 WINS

3.Test case 1 : it is a tie

As all spaces have been used up, a tie is declared.

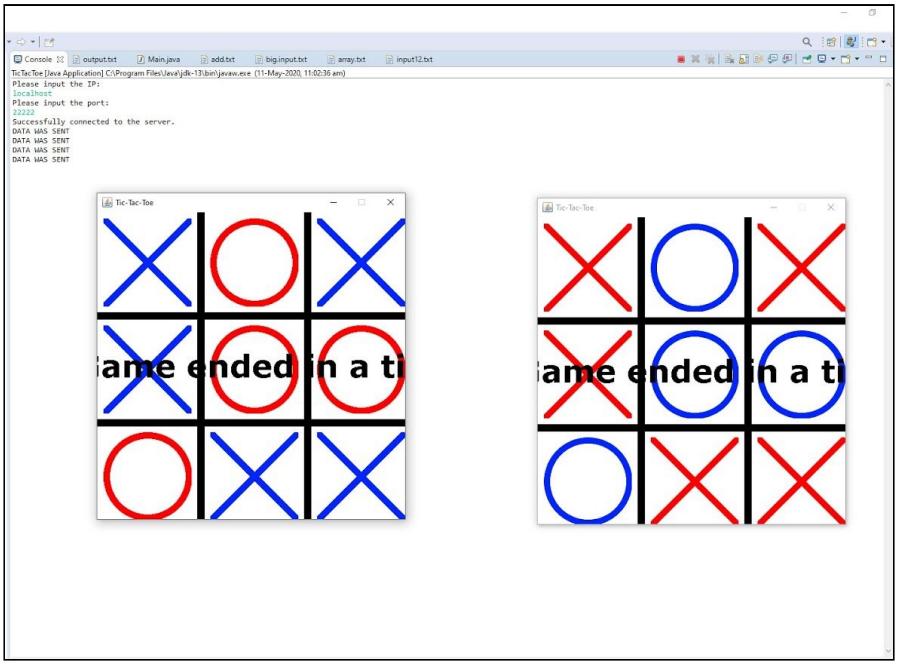

Fig 6.4 - Its a tie

VII. Conclusion

By doing this project we implemented a server/client telnet protocol using java socket programming. As we did the program, we learnt a lot about server client networks in java and also learnt how java can be used as both a font end and back end language.

For future works we can learn to make a 4 dot game using the same connection. This project can be played on 2 devices provided they have the same JRE version, we can try implementing it on different JDK versions.

VIII. References

[1] Java Network Programming, Elliotte Rusty Harold, 1997

[2] TCP/IP Sockets in Java,Second Edition. Practical guide for programmers.

[3] Graphic Java, 1.2. Mastering the JFC (David M Gallery)

[4] Java Socket programming from Tutorialspoint

https://www.tutorialspoint.com/java/java_networking.htm

[5] Data Communication and Networking ,Behroz A. Forouzan