Rock-Paper-Scissors-Lizard-Spock

Aim of the Project:

To create a Rock-Paper-Scissors-Spock-Lizard game played against a computer. And then among the persons using network connected computer. The game should be written in C++ language. The players' scores should be serialised to storage using boost serialisation library with appropriate STL containers. The communication between two instances of the game should be done using boost ASIO library. Computer should display a leaderboard of the players with their scores in % wins.

Objectives of the Project:

1. Determine Rock-Paper-Scissors-Spock-Lizard win as each player attack the other with one of the five pawns. The winner of the two is determined as below

    Rock blunts Scissors and crushes Lizard.

    Paper covers Rock and disproves Spock.

    Scissors cuts Paper and cuts Lizard.

    Spock vaporises Rock and breaks Scissors.

    Lizard eats Paper and poisons Spock.

2. Display a leader board of the players with their % of win.
3. Display number of scenarios where player chooses each of the five pawns.
4. Serialise game state to storage device and vice versa.
5. Create a server listening on port 7000 that can play against a client player across a network.

Scope of the project

- Program runs on windows machines connected with IPv4 network connection.
- Boost Library should be installed. And its library path should be added to the system path.
- Player names conflict in not handled and should be resolved by the players.
- Its a console application.

Algorithm:

1. Create a map entry for each pawn that gives defeaters and winners. And a map that gives winning action between any two pawns.

2. Deserialise “Players.bin” to load all players scores and info.

3. Computer should prompt the player to input his name & password.

1. Program should start listening on the port 7000 for the other players to connect asynchronously, stack the connections and display the connected players’ names.

2. Computer should prompt to input the name of the opponent or the network address of the opponent.

3. If player name is computer then play against the computer. If player name is among the list of connected players then choose the player’s connection for the communication. And acknowledge the opponent with the player’s info. Else if the input is network address then connect to the remote computer, send the player’s info and wait for the opponent to choose the player.

4. Once the handshake is complete the game begins by prompting the player to input his pawn.

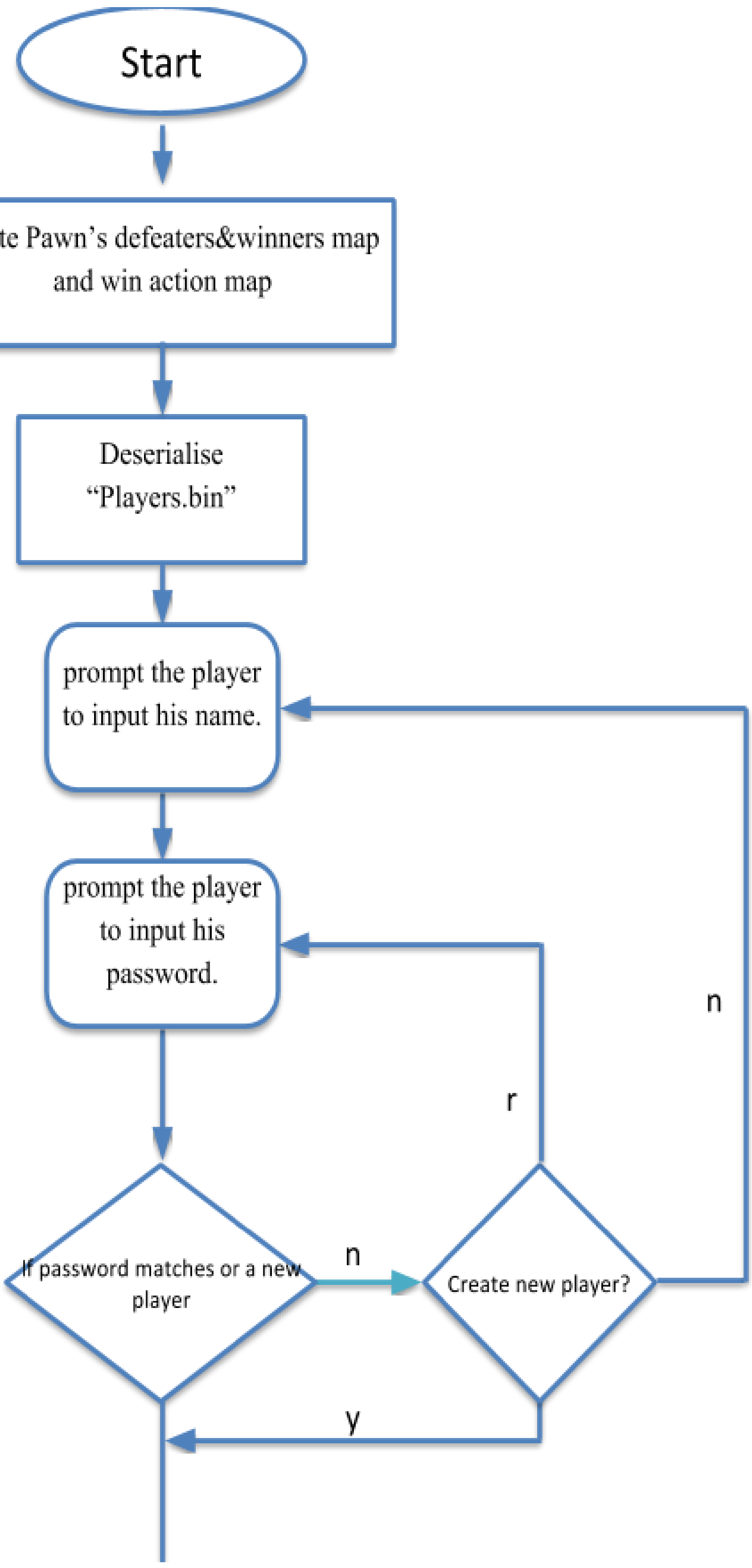
5. If played against the computer, the computer should pick a random pawn else the choosen pawn should be sent to the opponent and receive the opponent’s pawn.

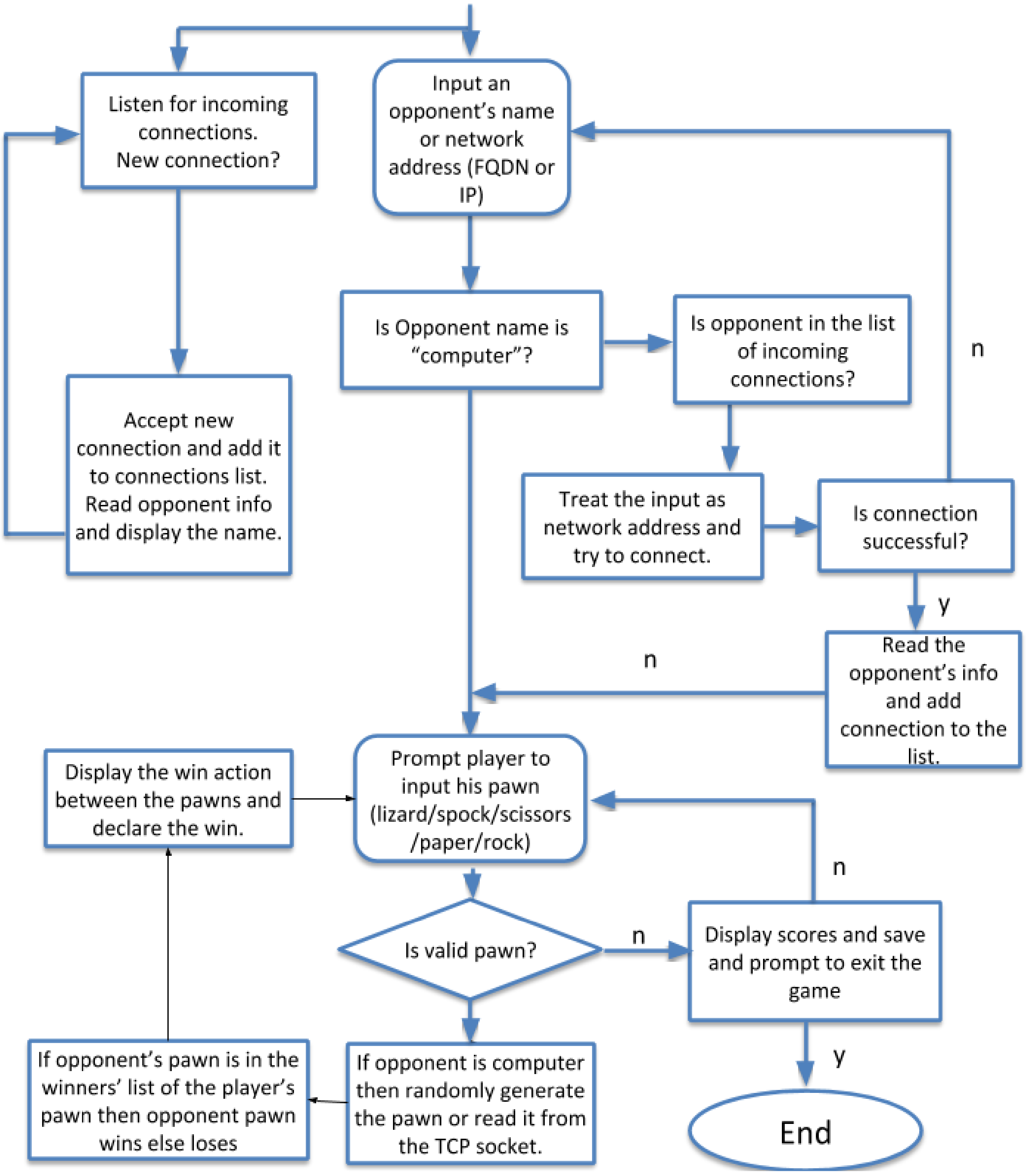
6. Action of the one pawn against the other should be displayed and the win or loose should be declared.

7. If the choosen pawn is not valid then prompt the player if he wants to exit.

8. If yes display and save the scores, close all the connections and exit the game else prompt for the pawn again.

Flowchart:


Start
Create Pawn's defeaters&winners map and win action map
Deserialise "Players.bin"
prompt the player to input his name.
prompt the player to input his password.
If password matches or a new player
n
Create new player?
r
n
y


Listen for incoming connections. New connection?
Input an opponent's name or network address (FQDN or IP)
Accept new connection and add it to connections list. Read opponent info and display the name.
Is Opponent name is "computer"?
Is opponent in the list of incoming connections?
n
Treat the input as network address and try to connect.
Is connection successful?
y
Read the opponent's info and add connection to the list.
n
Display the win action between the pawns and declare the win.
Prompt player to input his pawn (lizard/spock/scissors /paper/rock)
n
Is valid pawn?
n
Display scores and save and prompt to exit the game
y
If opponent's pawn is in the winners' list of the player's pawn then opponent pawn wins else loses
If opponent is computer then randomly generate the pawn or read it from the TCP socket.
End

Testing and results

1. Correct pawn actions? Yes.

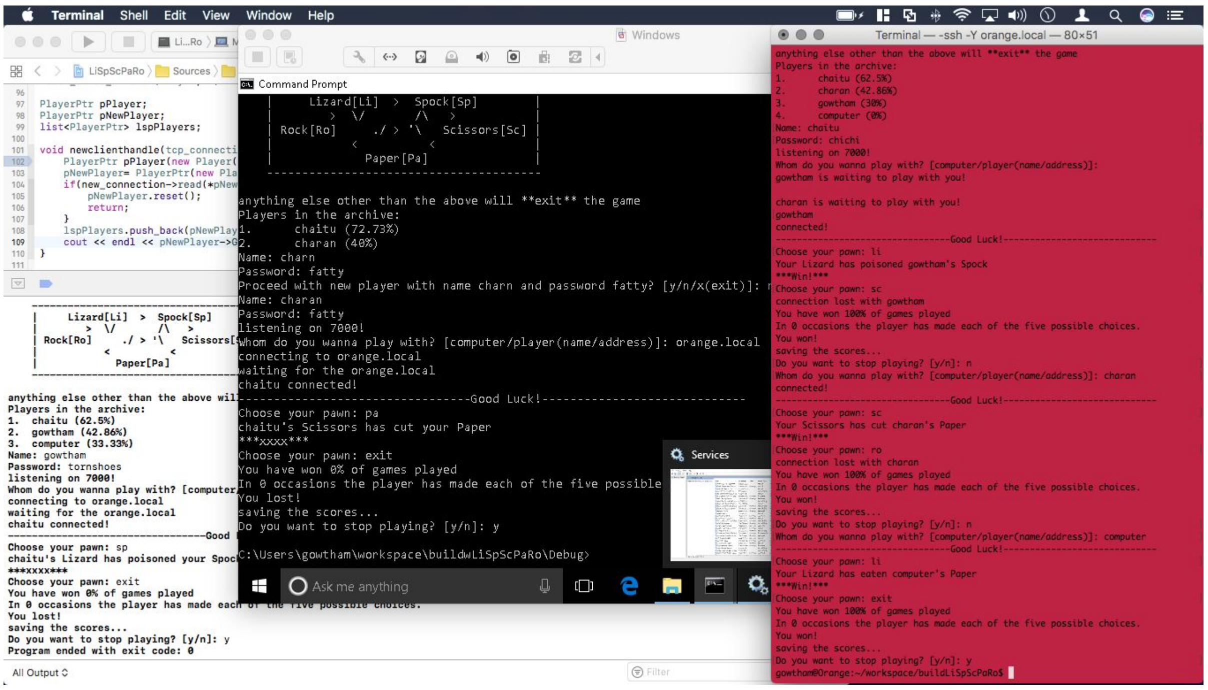
2. Scores saved and loaded? Yes.

3. Win % computed correctly? Yes.

4. Able to play with remote players? Yes.

5. Leader board shows correct scores every time game loads? Yes.

Screenshots/Snapshots



```
PlayerPtr pPlayer;
PlayerPtr pNewPlayer;
list<PlayerPtr> lspPlayers;

void newclienthandle(tcp_connecti
    PlayerPtr pPlayer(new Player(
    pNewPlayer= PlayerPtr(new Pla
    if(new_connection->read(*pNew
        pNewPlayer.reset();
        return;
    }
    lspPlayers.push_back(pNewPlay
    cout << endl << pNewPlayer->G
}
```

```
 ------------------------------------
 |     Lizard[Li]  >  Spock[Sp]
 |         >  \/        /\   >
 | Rock[Ro]     ./ > '\   Scissors[
 |           <            <
 |              Paper[Pa]
 ------------------------------------

anything else other than the above wil
Players in the archive:
1.  chaitu (62.5%)
2.  gowtham (42.86%)
3.  computer (33.33%)
Name: gowtham
Password: tornshoes
listening on 7000!
Whom do you wanna play with? [computer
connecting to orange.local
waiting for the orange.local
chaitu connected!
----------------------------------Good
Choose your pawn: sp
chaitu's Lizard has poisoned your Spoc
***xxxx***
Choose your pawn: exit
You have won 0% of games played
In 0 occasions the player has made each
You lost!
saving the scores...
Do you want to stop playing? [y/n]: y
Program ended with exit code: 0
```

Command Prompt

```
    |      Lizard[Li]  >  Spock[Sp]        |
    |         >  \/        /\   >          |
    | Rock[Ro]     ./ > '\   Scissors[Sc]  |
    |            <           <             |
    |              Paper[Pa]               |
    ---------------------------------------

anything else other than the above will **exit** the game
Players in the archive:
1.      chaitu (72.73%)
2.      charan (40%)
Name: charn
Password: fatty
Proceed with new player with name charn and password fatty? [y/n/x(exit)]: 
Name: charan
Password: fatty
listening on 7000!
Whom do you wanna play with? [computer/player(name/address)]: orange.local
connecting to orange.local
waiting for the orange.local
chaitu connected!
---------------------------------Good Luck!-----------------------------
Choose your pawn: pa
chaitu's Scissors has cut your Paper
***xxxx***
Choose your pawn: exit
You have won 0% of games played
In 0 occasions the player has made each of the five possible
You lost!
saving the scores...
Do you want to stop playing? [y/n]: y

C:\Users\gowtham\workspace\buildwLiSpScPaRo\Debug>
```

Terminal — -ssh -Y orange.local — 80×51

```
anything else other than the above will **exit** the game
Players in the archive:
1.      chaitu (62.5%)
2.      charan (42.86%)
3.      gowtham (30%)
4.      computer (0%)
Name: chaitu
Password: chichi
listening on 7000!
Whom do you wanna play with? [computer/player(name/address)]:
gowtham is waiting to play with you!

charan is waiting to play with you!
gowtham
connected!
--------------------------------Good Luck!----------------------------
Choose your pawn: li
Your Lizard has poisoned gowtham's Spock
***Win!***
Choose your pawn: sc
connection lost with gowtham
You have won 100% of games played
In 0 occasions the player has made each of the five possible choices.
You won!
saving the scores...
Do you want to stop playing? [y/n]: n
Whom do you wanna play with? [computer/player(name/address)]: charan
connected!
--------------------------------Good Luck!----------------------------
Choose your pawn: sc
Your Scissors has cut charan's Paper
***Win!***
Choose your pawn: ro
connection lost with charan
You have won 100% of games played
In 0 occasions the player has made each of the five possible choices.
You won!
saving the scores...
Do you want to stop playing? [y/n]: n
Whom do you wanna play with? [computer/player(name/address)]: computer
--------------------------------Good Luck!----------------------------
Choose your pawn: li
Your Lizard has eaten computer's Paper
***Win!***
Choose your pawn: exit
You have won 100% of games played
In 0 occasions the player has made each of the five possible choices.
You won!
saving the scores...
Do you want to stop playing? [y/n]: y
gowtham@Orange:~/workspace/buildLiSpScPaRo$ 
```