MÔ HÌNH TCP/IP(CLIENT-SERVER)CỜ CARO

// Do em không chạy được file trong bài thực hành nên em tạo 1 chương trình chạy với mô hình client – server .

File caroclient.java

**package** caro;

**import** java.awt.Color;

**import** java.awt.Font;

**import** java.awt.Graphics;

**import** java.awt.Point;

**import** java.awt.event.MouseEvent;

**import** java.awt.event.MouseListener;

**import** java.io.DataInputStream;

**import** java.io.DataOutputStream;

**import** java.net.Socket;

**import** java.util.ArrayList;

**import** java.util.List;

**import** javax.swing.JFrame;

**public** **class** carooffline **extends** JFrame **implements** MouseListener, Runnable {

**public** **static** **void** main(String[] args) {

**new** carooffline();

}

**int** n = 15;

**int** s = 40;

**int** os = 50;

List<Point> dadanh = **new** ArrayList<Point>();

DataOutputStream dos;

DataInputStream dis;

**public** carooffline() {

**this**.setSize(n \* s + 2 \* os, n \* s + 2 \* os);

**this**.setTitle("Caro");

**this**.setDefaultCloseOperation(3);

**this**.addMouseListener(**this**);

// Ket noi toi server

**try** {

Socket soc = **new** Socket("localhost", 80);

dos = **new** DataOutputStream(soc.getOutputStream());

dis = **new** DataInputStream(soc.getInputStream());

} **catch** (Exception e) {

e.printStackTrace();

}

**new** Thread(**this**).start();

**this**.setVisible(**true**);

}

**public** **void** paint(Graphics g) {

g.setColor(Color.***WHITE***);

g.fillRect(0, 0, **this**.getWidth(), **this**.getHeight());

g.setColor(Color.***BLACK***);

**for** (**int** i = 0; i <= n; i++) {

g.drawLine(os, os + i \* s, os + n \* s, os + i \* s);

g.drawLine(os + i \* s, os, os + i \* s, os + n \* s);

}

g.setFont(**new** Font("arial", Font.***BOLD***, s));

**for** (**int** i = 0; i < dadanh.size(); i++) {

**int** ix = dadanh.get(i).x;

**int** iy = dadanh.get(i).y;

**int** x = os + ix \* s + s / 2 - s / 4;

**int** y = os + iy \* s + s / 2 + s / 4;

String st = "o";

**if** (i % 2 != 0)

st = "x";

g.drawString(st, x, y);

}

}

@Override

**public** **void** mouseClicked(MouseEvent e) {

**int** x = e.getX();

**int** y = e.getY();

**if** (x < os || x >= os + n \* s)

**return**;

**if** (y < os || y >= os + n \* s)

**return**;

**int** ix = (x - os) / s;

**int** iy = (y - os) / s;

**for** (Point d : dadanh) {

**if** (ix == d.x && iy == d.y)

**return**;

}

**try** {

dos.writeUTF(ix + "");

dos.writeUTF(iy + "");

} **catch** (Exception e1) {

}

}

@Override

**public** **void** mousePressed(MouseEvent e) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** mouseReleased(MouseEvent e) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** mouseEntered(MouseEvent e) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** mouseExited(MouseEvent e) {

// **TODO** Auto-generated method stub

}

@Override

**public** **void** run() {

**while**(**true**) {

**try** {

**int** ix = Integer.*parseInt*(dis.readUTF());

**int** iy = Integer.*parseInt*(dis.readUTF());

dadanh.add(**new** Point(ix,iy));

**this**.repaint();

}**catch**(Exception e) {

}

}

}

}

File caroserver.java

package caro;

import java.awt.Point;

import java.io.DataInputStream;

import java.io.DataOutputStream;

import java.net.ServerSocket;

import java.net.Socket;

import java.util.ArrayList;

import java.util.List;

import java.util.Vector;

public class CaroServer {

public static void main(String[] args) {

new CaroServer();

}

static int n = 15;

static List<Point> dadanh = new ArrayList<Point>();

static Vector<Xuly> clients = new Vector<Xuly>();

public CaroServer() {

try {

ServerSocket server = new ServerSocket(80);

Socket p1 = server.accept();

// Player 1

Xuly c1 = new Xuly(p1);

clients.add(c1);

c1.start();

// Player 2

Socket p2 = server.accept();

Xuly c2 = new Xuly(p2);

clients.add(c2);

c2.start();

// Viewer

while (true) {

Socket s = server.accept();

Xuly v = new Xuly(s);

clients.add(v);

}

} catch (Exception e) {

}

}

}

class Xuly extends Thread {

Socket soc;

DataOutputStream dos;

DataInputStream dis;

public Xuly(Socket soc) {

try {

this.soc = soc;

dos = new DataOutputStream(soc.getOutputStream());

dis = new DataInputStream(soc.getInputStream());

for (Point p : CaroServer.dadanh) {

dos.writeUTF(p.x+"");

dos.writeUTF(p.y+"");

}

} catch (Exception e) {

}

}

public void run() {

try {

loop: while(true) {

int ix = Integer.parseInt(dis.readUTF());

int iy = Integer.parseInt(dis.readUTF());

///Can phai lam gi??!!!!!

// kiem tra ca 2 client da vao hay chua?

if (CaroServer.clients.size()<2) continue;

// Kiem tra co phai luot danh cua client nay hay khong!!

if (this == CaroServer.clients.get(0) && CaroServer.dadanh.size()%2!=0) continue;

if (this == CaroServer.clients.get(1) && CaroServer.dadanh.size()%2==0) continue;

// Kiem tra toa do co phu hop khong

if (ix<0 || ix>=CaroServer.n) continue;

if (iy<0 || iy>=CaroServer.n) continue;

for (Point d : CaroServer.dadanh) {

if (ix == d.x && iy == d.y)

continue loop;

}

CaroServer.dadanh.add(new Point(ix,iy));

// Gui toa do cho tat ca client dang co!!

for (Xuly c : CaroServer.clients) {

try {

c.dos.writeUTF(ix+"");

c.dos.writeUTF(iy+"");

}catch(Exception e) {

}

}

}

}catch(Exception e) {

}

}

}

Chạy chương trình :

