Forza 4

```
package esami.settembreDodici;

import javax.swing.*;

public class Forza4Main {

    public static void main(String[] args){


        Runnable init = new Runnable() {
            @Override
            public void run() {
                new Forza4Frame();
            }
        };

        SwingUtilities.invokeLater(init);
    }
}

package esami.settembreDodici;

import javax.swing.*;
import javax.swing.border.LineBorder;
import java.awt.*;

public class Forza4Frame extends JFrame {

    protected JLabel ipLabel = new JLabel("IP Address");
    protected JLabel portLabel = new JLabel("Port");
    protected JTextField ipBox = new JTextField(10);
    protected JTextField portBox = new JTextField(10);
    protected JButton connect = new JButton("Connetti");
    protected JButton disconnect = new JButton("Disconnetti");
    protected JButton start = new JButton("Start");
    protected JButton stop = new JButton("Stop");
    protected JButton clear  = new JButton("Clear");
    protected JPanel campo = new JPanel(new GridLayout(4,4));
    protected JPanel[][] grid = new JPanel[4][4];
    private Forza4Listener listener;


    public Forza4Frame(){
        super("Dario Pietrosanto");
        Forza4Frame frame=this;
        frame.setResizable(false);
        frame.setLayout(new BorderLayout(10,10));
        frame.setDefaultCloseOperation(WindowConstants.EXIT_ON_CLOSE);
        frame.setLocationRelativeTo(null);
    listener = new Forza4Listener(frame);


    frame.add( top(), BorderLayout.NORTH);
    frame.add( middle(), BorderLayout.CENTER);
    frame.add( bottom(), BorderLayout.SOUTH);

    frame.pack();
    frame.setVisible(true);
  }

  private JPanel top(){
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER,10,10));

    panel.add(ipLabel);
    ipBox.setText("127.0.0.1");
    panel.add(ipBox);
    panel.add(portLabel);
    portBox.setText("4400");
    panel.add(portBox);
    connect.setActionCommand(Forza4Listener.CONNECT);
    connect.addActionListener(listener);
    connect.setEnabled(true);
    panel.add(connect);
    disconnect.setActionCommand(Forza4Listener.DISCONNECT);
    disconnect.addActionListener(listener);
    disconnect.setEnabled(false);
    panel.add(disconnect);

    return panel;
  }

  private JPanel middle(){
    for (int i = 0; i < 4; i++) {
      for (int j = 0; j < 4; j++) {
        JPanel nuovo = new JPanel();
        nuovo.setPreferredSize(new Dimension(150,150));
        nuovo.setBorder(new LineBorder(Color.BLACK));
        nuovo.setBackground(Color.LIGHT_GRAY);
        campo.add(nuovo);
        grid[i][j]=nuovo;
      }
    }
    return campo;
  }

  private JPanel bottom(){
    JPanel panel = new JPanel(new FlowLayout(FlowLayout.CENTER,10,10));

    start.setActionCommand(Forza4Listener.START);
    start.addActionListener(listener);
```

```
        start.setEnabled(false);
        panel.add(start);
        stop.setEnabled(false);
        stop.setActionCommand(Forza4Listener.STOP);
        stop.addActionListener(listener);
        panel.add(stop);
        clear.addActionListener(listener);
        clear.setActionCommand(Forza4Listener.CLEAR);
        clear.setEnabled(true);
        panel.add(clear);

        return panel;
    }

}
package esami.settembreDodici;

import javax.swing.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.io.IOException;
import java.io.OutputStreamWriter;
import java.io.PrintWriter;
import java.net.Socket;
import java.util.Scanner;

import static javax.swing.JOptionPane.CLOSED_OPTION;
import static javax.swing.JOptionPane.YES_OPTION;

public class Forza4Listener implements ActionListener {
    public static final String CONNECT = "connect";
    public static final String DISCONNECT = "disconnect";
    public static final String START = "start";
    public static final String STOP = "stop";
    public static final String CLEAR = "clear";

    private Forza4Frame frame;
    private Socket socket;
    private PrintWriter printer;
    private Scanner scanner;
    private Forza4Worker worker;

    public Forza4Listener(Forza4Frame frame) {
        this.frame = frame;
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String cmd = e.getActionCommand();

        if (cmd.equals(CONNECT)){
```

```
        String ip= frame.ipBox.getText();
        Integer port;
        try {
          port=Integer.parseInt(frame.portBox.getText());
          if (port<0)
            throw new NumberFormatException();
        } catch (NumberFormatException e1){
          JOptionPane.showMessageDialog(frame, "Inserisci un numero di porta valido.",
              "ERRORE", JOptionPane.WARNING_MESSAGE);
          return;
        }
        try {
          socket = new Socket(ip,port);
        } catch (IOException e1){
          JOptionPane.showMessageDialog(frame, "Impossibile connettersi a
"+ip+":"+port+"\nRiprova.",
              "Errore",JOptionPane.WARNING_MESSAGE);
          return;
        }
        try {
          printer = new PrintWriter(new OutputStreamWriter(socket.getOutputStream()));
          scanner = new Scanner(socket.getInputStream());
        } catch (IOException e1){
          JOptionPane.showMessageDialog(frame, "Errore nella connessione al server.\nRiprova.",
              "Errore",JOptionPane.WARNING_MESSAGE);
          return;
        }
        JOptionPane.showMessageDialog(frame, "Connessione riuscita.", "Connessione riuscita",
            JOptionPane.INFORMATION_MESSAGE);

        frame.connect.setEnabled(false);
        frame.disconnect.setEnabled(true);
        frame.start.setEnabled(true);
        return;

      } else if (cmd.equals(DISCONNECT)){
        printer.println(DISCONNECT);
        printer.flush();
        try {
          printer.close();
          scanner.close();
          socket.close();
        } catch (IOException e1){
          JOptionPane.showMessageDialog(frame, "Errore in chiusura della connessione.",
              "Errore",JOptionPane.WARNING_MESSAGE);
          return;
        }
        JOptionPane.showMessageDialog(frame,"Connessione chiusa con successo.",
            "Connessione chiusa",JOptionPane.INFORMATION_MESSAGE);

        frame.connect.setEnabled(true);
        frame.disconnect.setEnabled(false);
```

```
            frame.start.setEnabled(false);
            return;

        } else if (cmd.equals(START)){
            Integer opz = JOptionPane.showOptionDialog(frame,"Scegli il tuo colore.",
                    "Scegli il tuo colore", JOptionPane.YES_NO_OPTION,
                    JOptionPane.QUESTION_MESSAGE, null, new String[]{"Ciano", "Giallo"}, null);

            if (opz.equals(CLOSED_OPTION))
                return;


            clear();

            frame.start.setEnabled(false);
            frame.stop.setEnabled(true);
            frame.disconnect.setEnabled(false);
            frame.clear.setEnabled(false);

            worker = new Forza4Worker(frame, printer, scanner, opz==YES_OPTION?
Color.CYAN:Color.YELLOW);
            worker.execute();
            return;

        } else if (cmd.equals(STOP)){
            if (!worker.equals(null))
                worker.cancel(true);
            return;

        } else if (cmd.equals(CLEAR)){
            clear();
        }
    }


    private void clear(){
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                frame.grid[i][j].setBackground(Color.LIGHT_GRAY);
            }
        }
        return;
    }
}

package esami.settembreDodici;

import javax.swing.*;
import java.awt.*;
import java.io.PrintWriter;
import java.util.NoSuchElementException;
import java.util.Scanner;
import static java.lang.Thread.sleep;

public class Forza4Worker extends SwingWorker<Boolean, Object> {
    private Forza4Frame frame;
    private PrintWriter printer;
    private Scanner scanner;
    private Color user;

    public Forza4Worker(Forza4Frame frame, PrintWriter printer, Scanner scanner, Color colore) {
        this.frame = frame;
        this.printer = printer;
        this.scanner = scanner;
        this.user = colore;
    }

    @Override
    protected Boolean doInBackground() throws Exception {
        if (!isCancelled()){
            printer.println(Forza4Listener.START);
            printer.flush();

            try {
                while (true) {
                    String line = scanner.nextLine();
                    if (line.equals("*"))
                        break;
                    int row;
                    String colore;
                    try {
                         row = Integer.parseInt(line.split(";")[0]);
                         colore = line.split(";")[1];
                    } catch (NumberFormatException e) {
                        continue;
                    }
                    frame.grid[row/4][row%4].setBackground(colore.equals("cyan")?
Color.CYAN:Color.YELLOW);
                }
            } catch (NoSuchElementException e){
                return false;
            }
        }
        return true;
    }

    @Override
    protected void done(){
        printer.println(Forza4Listener.STOP);
        printer.flush();

//        try {
//            for (int i = 0; i < 10; i++) {
```

```
//                if (!scanner.equals(null))
//                    System.out.println(scanner.next().equals("-1"));
//                sleep(100);
//            }
//        } catch (Exception e){
//            e.printStackTrace();
//        }

        frame.start.setEnabled(true);
        frame.stop.setEnabled(false);
        frame.clear.setEnabled(true);
        frame.disconnect.setEnabled(true);

        if (isCancelled())
            checkWinner(null);
        else
            checkWinner(user);
//        JOptionPane.showMessageDialog(frame,"Trasmissione conclusa.","Trasmissione conclusa",
//                JOptionPane.INFORMATION_MESSAGE);
        return;
    }


    private void checkWinner(Color colore){
        if (colore==null){
            JOptionPane.showMessageDialog(frame,"Hai perso.","Sconfitta utente",
                JOptionPane.PLAIN_MESSAGE);
            return;
        }
        int user=0,computer=0;
        int[][] tabella = new int[4][4];

        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                if (frame.grid[i][j].getBackground().equals(colore))
                    tabella[i][j]=1;
                else
                    tabella[i][j]=0;
//                System.out.print("["+tabella[i][j]+"]");
            }
//            System.out.println();
        }
//        System.out.println(colore.toString());


        for (int i = 0; i < 4; i++) {
            if (tabella[i][0]+tabella[i][1]+tabella[i][2]+tabella[i][3]==4)
                user++;
            if (tabella[i][0]+tabella[i][1]+tabella[i][2]+tabella[i][3]==0)
                computer++;
        }
        for (int i = 0; i < 4; i++) {
            if (tabella[0][i]+tabella[1][i]+tabella[2][i]+tabella[3][i]==4)
               user++;
            if (tabella[0][i]+tabella[1][i]+tabella[2][i]+tabella[3][i]==0)
               computer++;
         }

         if (tabella[0][0]+tabella[1][1]+tabella[2][2]+tabella[3][3]==4)
            user++;
         if (tabella[0][0]+tabella[1][1]+tabella[2][2]+tabella[3][3]==0)
            computer++;
         if (tabella[0][3]+tabella[1][2]+tabella[2][1]+tabella[3][0]==4)
            user++;
         if (tabella[0][3]+tabella[1][2]+tabella[2][1]+tabella[3][0]==0)
               computer++;


         if (user>computer){
            JOptionPane.showMessageDialog(frame,"Hai vinto.","Vittoria utente",
                  JOptionPane.PLAIN_MESSAGE);
         } else if (computer>user){
            JOptionPane.showMessageDialog(frame,"Hai perso.","Sconfitta utente",
                  JOptionPane.PLAIN_MESSAGE);
            return;
         } else {
            JOptionPane.showMessageDialog(frame,"Hai pareggiato.","Pareggio",
                  JOptionPane.PLAIN_MESSAGE);
         }
      }
   }
```