# ***Progetto: FORZA 4***

A cura di Luca Song, Radoslaw Jakub Zak

## *Introduzione:*

Progettazione del gioco "Forza 4", mediante l'uso di socket dei server concorrenti, scritto in linguaggio java.

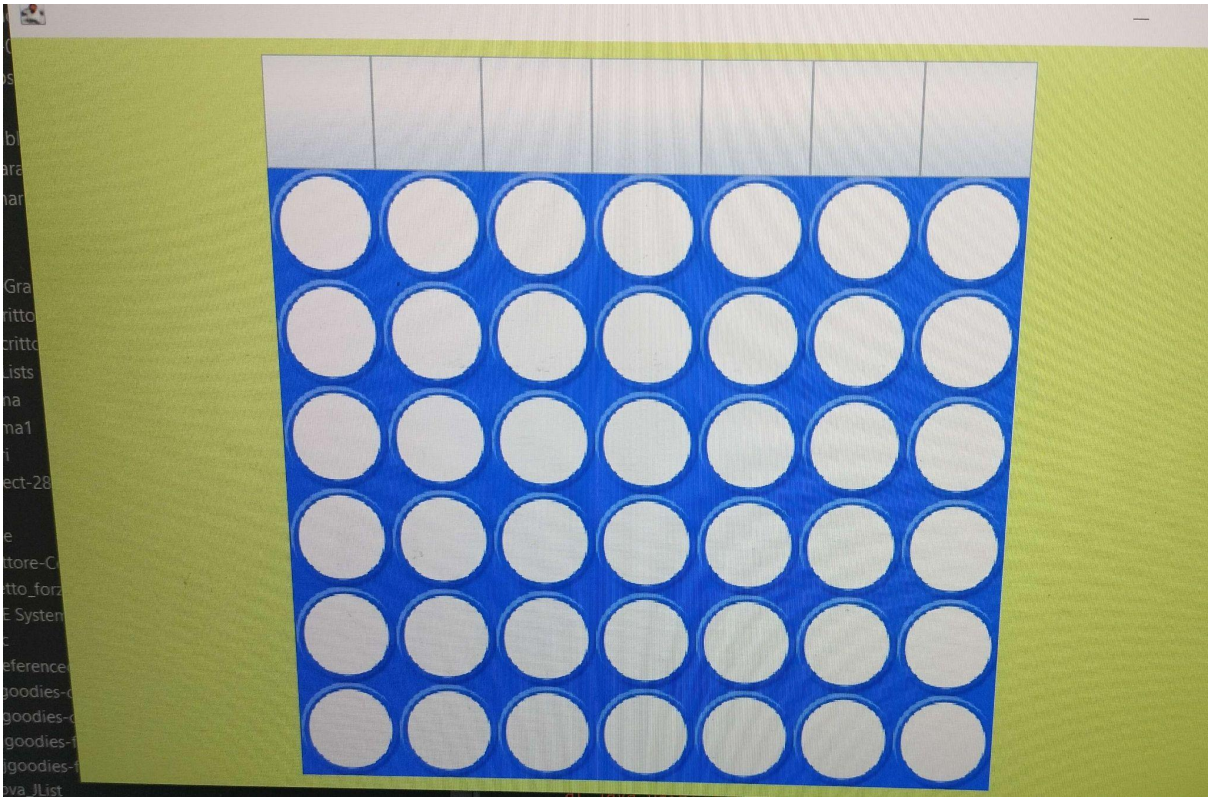
## *Descrizione generale:*

- Una volta che avremo fatto partire il due server(ciascuno legato al proprio client), l'utente dovrà inserire l'indirizzo ip corretto per poter accedere alla partita


(la schermata iniziale)

- dopo aver effettuato il login, il giocatori vedranno la schermata principale in cui vedranno la schermata principale del gioco, il cui

vedranno la tabella dei gettoni con il pulsanti soprastanti



la generazione della tabella viene effettuata tramite un metodo chiamato “genera_celle”;

```java
public void genera_celle(int x, int y, JPanel contentPane, final int dim, String[] mosse) {
    int coordinata_x = x;
    int coordinata_y = y;
    int i = 0;
    for (int riga = 0; riga < 6; riga++) {
        for (int colonna = 0; colonna < 7; colonna++) {
            JLabel cella = new JLabel(mosse[i]);
            String[] cont = mosse[i].split("-");
            switch (cont[1]) {
            case "g":
                cella.setIcon(new ImageIcon(dirBase + "/src/img/giocatore1.jpg"));
                cella.setBounds(coordinata_x, coordinata_y, dim, dim);
                coordinata_x += dim;
                contentPane.add(cella);
                break;
            case "r":
                cella.setIcon(new ImageIcon(dirBase + "/src/img/giocatore2.jpg"));
                cella.setBounds(coordinata_x, coordinata_y, dim, dim);
                coordinata_x += dim;
                contentPane.add(cella);
                break;
            default:
                cella.setIcon(new ImageIcon(dirBase + "/src/img/cella_vuota.jpg"));
                cella.setBounds(coordinata_x, coordinata_y, dim, dim);
                coordinata_x += dim;
                contentPane.add(cella);
                break;
            }
            i++;
        }
        coordinata_x = x;
        coordinata_y += dim;
    }
}
```

in cui vedrà passare le due coordinate, il pannello di riferimento, che dovrà ospitare la tabella, la dimensione(indicata come una costante) delle immagini e l’array delle mosse in cui il giocatori si scambiano quando posizionano il loro gettone.

- quando uno dei due giocatori clicca uno dei bottoni soprastanti, la cella più bassa della tabella(in cui non è stato occupato) prende colore da uno dei due giocatori

```java
private void preparazioneDati(String mossa) {
    Dati dato1 = new Dati(mossa, operatore, Dati.Operazione.MOS);
    invioDati(dato1);
}

private void calcMossa(int colonna) {
    for (int i = 0; i < 42; i++) {
        String controllo[] = mosse[i].split("-");
        int integer = Integer.parseInt(controllo[0]); // %10 = colonne,/10= righe
        if (integer % 10 == colonna && controllo[1].equals("n") == true) {
            preparazioneDati(integer+"-"+operatore);
            break;
        }
    }
}

private void listener() {

    interfaccia_user.getBtnNewButton().addActionListener(this);
    schermoVittoria.getBtnNewButton().addActionListener(this);
}
```

```java
if (e.getSource() == interfaccia_user2.getBtnNewButton_1()) {
    calcMossa(1);
}
```

```java
public String controlloUsers() {
    if (!buffer.getGiocatore1()) {
        operatore = "g";
        return "g";
    } else if (!buffer.getGiocatore2()) {
        operatore = "r";
        return "r";
    }
    operatore = "n";
    return "n";

}
```

- per controllo della vittoria, prende il dato necessario dal buffer condiviso(ovvero il giocatore che è riuscito a soddisfare la condizione della vittoria, che verrà indicata come “n”, nel caso che la cella non fosse occupata, “g” oppure “r” quando viene occupato da giocatore 1 o giocatore 2) e confronta se il requisiti fossero soddisfatti, se uno dei due vince, allora verrà mandata ai due host la rispettiva schermata di vittoria / sconfitta, altrimenti se nessuno riesce a vincere, verrà

mandato a entrambi la condizione del pareggio

```java
public void run() {
    try {

        while (true) {
            inputStream = socket.getInputStream();
            objectInputStream = new ObjectInputStream(inputStream);
            Dati datoInput = (Dati) objectInputStream.readObject();
            if (datoInput.getOp() == Dati.Operazione.GIO) {
                dati = datoInput;
            } else if (datoInput.getOp() == Dati.Operazione.ERR) {
                System.out.println(datoInput.getMessErrore());
            } else if (datoInput.getOp() == Dati.Operazione.PAR) {
                interfaccia_user2.setVisible(false);
                schermoVittoria.setVisible(true);
                schermoVittoria.getLblNewLabel().setText("pareggio");
            } else if (datoInput.getOp() == Dati.Operazione.PER) {
                interfaccia_user2.setVisible(false);
                schermoVittoria.setVisible(true);
                schermoVittoria.getLblNewLabel().setText("Sconfitta");
            } else if (datoInput.getOp() == Dati.Operazione.VIT) {
                interfaccia_user2.setVisible(false);
                schermoVittoria.setVisible(true);
                schermoVittoria.getLblNewLabel().setText("Vittoria");
            } else if (datoInput.getOp() == Dati.Operazione.NOM) {
                operatore = datoInput.getTesto();
            }else if (datoInput.getOp() == Dati.Operazione.MOS) {
                aggionaBuffer(datoInput);
                interfaccia_user2.setMosse(mosse);
            }
            Thread.sleep(250);
        }
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```java
public boolean confronto4Str(String a, String b, String c, String d) {
    if (a.compareTo(b) == 0 && b.compareTo(c) == 0 && c.compareTo(d) == 0 && a.compareTo("n") != 0) {
        return true;
    }
    return false;
}

public void controlloVincita(String[] mosse) {
    Dati dato1;
    for (int j = 0; j < 42; j++) {
        Dati dati1;
        String[] cont = buffer.getmosse()[j].split("n");
        int asd = Integer.parseInt(cont[0]);
        // controllo vittoria in riga orizzontale
        if (j < 39) {
            String controllo[] = buffer.getmosse()[j].split("-");
            String controllo1[] = buffer.getmosse()[j + 1].split("-");
            String controllo2[] = buffer.getmosse()[j + 2].split("-");
            String controllo3[] = buffer.getmosse()[j + 3].split("-");
            if (confronto4Str(controllo[1], controllo1[1], controllo2[1], controllo3[1])) {
                if (controllo[1].compareTo(operatore) == 0) {
                    invioDatiGenerico(dato1 = new Dati("server", Dati.Operazione.VIT));// in caso di vittoria di un
                                                                                       // giocatore 1 o 2;
                } else {
                    invioDatiGenerico(dato1 = new Dati("server", Dati.Operazione.PER));// in caso di sconfitta
                                                                                       // dell'altro giocatore;
                }

            }

        } else {
            invioDatiGenerico(dato1 = new Dati("server", Dati.Operazione.PAR)); // nel caso in cui nessuno dei due
                                                                                // giocatori vince
        }
```

(esempio di controllo vincita, per riga orizzontale)

- alla fine della partita, il giocatore vedrà la schermata finale, in cui viene linkato un bottone che farà tornare alla schermata iniziale