

The background features a complex geometric design. In the top-left corner, there are overlapping blue and green diagonal stripes. The bottom-left corner shows a circular inset with a detailed image of a printed circuit board (PCB) with various electronic components. The top-right corner displays a series of white, concentric, stepped lines resembling a circuit trace or a stylized architectural element. The word "Go" is centered in a blue, sans-serif font.

Go



Obiettivo

L'obiettivo del progetto era di permettere ad un giocatore di collegarsi ad un server e di giocare contro di lui a go.

Inoltre a livello tecnico dovevamo essere in grado di stabilire una connessione client-server e farli comunicare in modo a noi congeniale, infatti le mosse vengono inviate al server che, dopo aver svolto la sua mossa, invierà la situazione del goban (la scacchiera) al client che si aggiornerà.



Dettagli tecnici



Dettagli tecnici

Per far comunicare il client con il server mi sono avvalso delle socket, che sono un'astrazione software, che simula un canale dedicato nella rete per far comunicare due processi.

Le socket, in quanto astrazioni, vengono gestite dal sistema operativo e questo ci permette di far comunicare anche programmi scritti in linguaggi diversi, ad esempio potremmo creare un client in C# e un server in Java senza aver problemi di incompatibilità.



Dettagli del codice



Parti di codice

Connessione tra client e server

```
bool connected = false;
byte[] msg = Encoding.ASCII.GetBytes(color.ToLower() + "***");
while (!connected)
{
    sok.Connect(remoteEP);
    int bytesSent = sok.Send(msg);
    if (sok.Connected)
    {
        connected = true;
    }
}
```

connessione lato client

```
sok.Bind(localEndPoint);
sok.Listen(1);
handler = sok.Accept();
Console.WriteLine("Connection started");
while (data.IndexOf("***") == -1)
{
    int bytesRec = handler.Receive(bytes);
    data += Encoding.ASCII.GetString(bytes, 0, bytesRec);
}
data = data.Split('*')[0];
Console.WriteLine(data);
```

connessione lato server

Parti di codice

```
int c = int.Parse(sender.Name.Split(';')[0]), r = int.Parse(sender.Name.Split(';')[1]);
if(checksuicide(c, r))
{
    if (sender.BackColor == Color.Transparent)
    {
        sender.BackColor = Color.FromName(color);
        sender.Enabled = false;
        sok.Send(Encoding.ASCII.GetBytes(sender.Name.Split(';')[0] + sender.Name.Split(';')[1] + "***"));
    }
    else
    {
        MessageBox.Show("Non puoi mettere qui la tua pedina");
    }
}
else
{
    MessageBox.Show("Non puoi mettere una pedina dove verrebbe mangiata");
}
```

Scelta mossa client

```
string chmove()
{
    int col, row;
    bool ok = false;
    string s = "";
    Random c = new Random(68), r = new Random(71);
    while (!ok)
    {
        col = c.Next(0, 9);
        row = r.Next(0, 9);
        if (f[col][row].val == 0)
        {
            if (server == "black")
            {
                f[col][row].val = 2;
            }
            else
            {
                f[col][row].val = 1;
            }
            ok = true;
            s = col.ToString() + row.ToString();
        }
    }
    refresh(s, server);
    return s;
}
```

Scelta mossa server

Parti di codice

Aggiornamento goban server

```
status = "";
for (int i = 0; i < 9; i++)
{
    for (int j = 0; j < 9; j++)
    {
        if (f[i][j].t != 0 && f[i][j].b != 0 && f[i][j].l != 0 && f[i][j].r != 0)
        {
            if (f[i][j].val != f[i][j].t && f[i][j].val != f[i][j].b && f[i][j].val != f[i][j].l && f[i][j].val != f[i][j].r)
            {
                if (f[i][j].l != -1)
                {
                    f[i][j].val = f[i][j].l;
                }
                else
                {
                    f[i][j].val = f[i][j].r;
                }
            }
        }
        status += f[i][j].val.ToString();
    }
}
```




Parti di codice

Aggiornamento goban client

```
int k = 0;
for (int i = 0; i < 9; i++)
{
    for (int j = 0; j < 9; j++)
    {
        if (tab.ToCharArray()[k] == '1')
        {
            lb[i][j].BackColor = Color.White;
        }
        else if (tab.ToCharArray()[k] == '2')
        {
            lb[i][j].BackColor = Color.Black;
        }
        else
        {
            lb[i][j].BackColor = Color.Transparent;
        }
        k++;
    }
}
```

Grazie per l'attenzione

Marco Salvi

