

Universitatea „Alexandru Ioan Cuza”

Facultatea de Informatică

Rețele de calculatoare

Tema 2

Connect4 (B)

Tîrnovan Camelia – Elena

An II, grupa A4

Cuprins

1. Introducere
2. Tehnologii utilizate
3. Arhitectura Aplicației
4. Detalii de implementare
5. Concluzii
6. Bibliografie

1. Introducere

Connect Four este un joc de strategie cu 2 jucători. Fiecare dintre aceștia își va alege o culoare și va avea mai multe discuri de culoarea respectivă. Acestea vor fi aruncate din partea de sus într-o grilă suspendată vertical cu șase coloane și șase rânduri, căzând direct în jos. Obiectivul jocului este de a fi primul care formează o linie orizontală, verticală sau diagonală cu patru dintre discurile aruncate.

Jocul Connect Four va fi simulat într-o aplicație client/server. Fiecare client se conectează la server unde acesta va stabili jucătorul care face prima mutare și culoarea cu care va juca fiecare. Serverul are rolul de a ține scorul, jocul putând fi jucat pe mai multe reprize, precum și rolul de a afișa gridul fiecărui jucător, după ce se efectuează o mutare.

2. Tehnologii utilizate

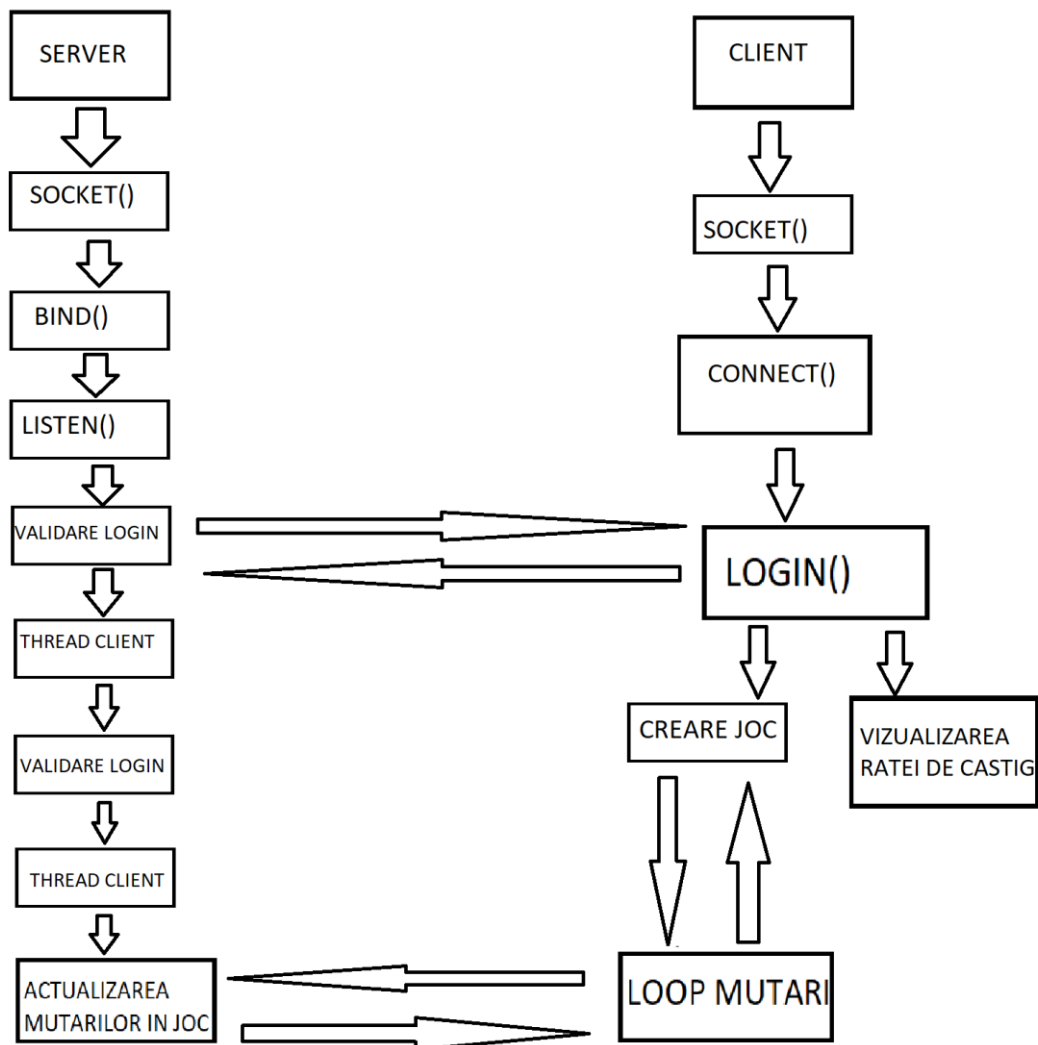
Se va folosi TCP (Transmission Control Protocol), protocol ce efectuează o conectare virtuală între server și client. Acesta va asigura transmiterea corectă a informației la receptor, fără duplicarea sau pierderea acestora.

Nu voi folosi UDP pentru că vreau să evit erorile care pot să apară, chiar dacă acesta este mai rapid.

Pentru verificarea existenței datelor de intrare de la descriptori de citire multipli se va folosi `select()` care permite utilizarea apelurilor blocante pentru un set de descriptori (fișiere, pipe-uri, socket-uri,...) și suspendă programul până când descriptorii din liste sunt pregătiți de operații de I/O.

3.Arhitectura aplicației

Aplicația va folosi o arhitectură de tipul client-server TCP concurent.



4.Detalii de implementare

Pentru a se putea realiza comunicarea dintre server și client se va folosi canalul bidirecțional de comunicare socket(), astfel server-ul va putea să-i transmită clientului/clientilor datele necesare și respectiv, invers.

```
/* crearea unui socket */
```

```
struct sockaddr_in address;
```

```
if ((server_fd = socket(AF_INET, SOCK_STREAM, 0)) == 0)
```

```
{
```

```
    perror("socket failed");
```

```
    exit(EXIT_FAILURE);
```

```
}
```

```
/* conectarea la server */
```

```
if (connect (sd, (struct sockaddr *) &server,sizeof (struct sockaddr)) == -1)
```

```
{
```

```
    perror ("[client]Eroare la connect().\n");
```

```
    return errno;
```

```
}
```

```
/* punem serverul sa asculte daca vin clienti sa se conecteze */  
if (listen (sd, 5) == -1)  
{  
    perror ("[server] Eroare la listen().\n");  
    return errno;  
}
```

6.Concluzii

Una din îmbunătățirile pe care le-aș aduce este o librărie grafică pentru ca aplicația să fie cât mai ușor de utilizat.

7.Bibliografie

https://en.wikipedia.org/wiki/Connect_Four

<https://www.howtogeek.com/190014/htg-explains-what-is-the-difference-between-tcp-and-udp/>

<https://www.geeksforgeeks.org/socket-programming-cc/>

<https://www.geeksforgeeks.org/client-server-model/>