

Universitatea Alexandru Ioan Cuza

Facultatea de informatică

Disciplina: Retele de calculatoare

Proiect : Connect4

Grigoras Marian-Gabriel Anul II, Grupa A7

Anul universitar 2016-2017

marian.grigoras@info.uaic.ro

Cuprins:

- 1.Introducere
- 2.Tehnologii utilizate
3. Arhitectura aplicatiei
4. Concluzii
5. Bibliografie

Introducere

Proiectul are ca scop implementarea jocului Connect4 pe un server si crearea unui client compatibil cu server-ul.

Descrierea jocului

Connect4 se joaca in doi, fiecare jucator primeste jetoane de cate o culoare, galben sau rosu, pe care le pot introduce intr-un tabel de dimensiunea 6x7. Tabelul e asezat perpendicular pe pamant astfel incat jucatorii pot alege doar coloana pe care sa aseze piesa nu si linia, aceasta fiind determinata de numarul de piese asezate anterior pe coloana respectiva. Primul jucator care reuseste sa aseze 4 jetoane in linie, coloana sau diagonala, castiga;

Tehnologii utilizate

Pentru implementarea proiectului dat se utilizeaza protocolul TCP (de la engl. Transmission Control Protocol , Protocolul de control al transmisiei), care este un protocol folosit de obicei de aplicații care au nevoie de confirmare de primire a datelor. Efectuează o conectare virtuală full duplex între două puncte terminale, fiecare punct fiind definit de către o adresa IP și de către un port TCP. Protocolul corespunde nivelului transport din stiva TCP/IP.

Biblioteci utilizate

- pentru implementarea server-ului

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

- pentru eventualele erori pe care vreau sa le prind

```
#include <errno.h>
```

(exemplu: cod server linia 53)

- pentru functia “fork()”, apelurile de asteptare a copiilor si de terminare fortata a lor

```
#include <sys/wait.h>
#include <signal.h>
```

(exemplu de utilizare: cod server linia 258)
(exemplu de utilizare: cod server linia 259)

- pentru functiile jocului

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>
#include <stdbool.h>
```

Arhitectura aplicatiei

Aplicatia va folosi o arhitectura de tipul CLIENT/SERVER TCP concurent. Serverul va primi cererile de conectare de la clienti si va raspunde cu un mesaj care anunta jucatorul ca trebui sa astepte pentru conectarea unui adversar, iar dupa ce vor fi doi clienti conectati va porni jocul.

Dupa ce porneste jocul, clientii sunt anuntati de culoarea cu care vor juca si ordinea in care vor porni, apoi informati la fiecare mutare daca este sau nu randul lor.

Tabla de joc va fi actualizata dupa validarea din partea server-ului a deciziei clientului, si trimisa ambilor clienti.

Daca unul dintre clienti castiga, sau se umple tabla si nu mai pot fi introduse piese, runda se termina dar jocul poate fi continuat la decizia clientului, in cazul in care unul dintre clienti renunta, va fi deconectat si celalalt.

Concluzii

Nu voi mai folosi niciodata declarari de genul "tip_variabila *nume_variabila". Proiectul mi s-a parut destul de simplu, deoarece am facut intai jocul in C++ cu rulare locala, iar apoi l-am integrat in server si am facut sincronizare read write cu clientul.

Cea mai grea parte, sau mai bine zis cea care m-a incurcat cel mai mult a fost continuarea jocului la decizia clientilor (linia 192 server, 122 client). Desi nu era precizat in enunt m-am gandit ca ar fi correct ca jocul sa continue doar daca ambii jucatori sunt deacord.

Cea mai inovativa parte a proiectului este folosirea lui XOR pentru a determina adversarul fara a folosi "select()", deoarece clientii sunt pusi intr-un vector in care se ocupa doar pozitiile 0 si 1, iar facand XOR cu 1oricarui client ii obtin adversarul.

Bibliografie

- Cursurile disciplinei “Rețele de calculatoare ” de Lenuta Alboaie.
- <http://stackoverflow.com/>
- <http://www.cplusplus.com/>
- <https://www.google.ro/>