

Projekt 4 gewinnt:

(Von Matthias König, Dominik Schramek, Khalil Hanna)

Das Spiel 4 gewinnt soll mithilfe von TCP/IP-Sockets implementiert werden. Die Steuerung des Spieles übernimmt ein Server-Prozess. Der erste sich verbindende Prozess übernimmt den Spieler 1 und bekommt das Symbol Kreuz (X). Der 2. sich verbindende Prozess übernimmt den Spieler 2 und das Symbol Kreis (O). Mit Spieler 1 beginnend wird nun abwechselnd eine Spalte gewählt in der ein Stein platziert werden soll. Der Spieler der zuerst 4 Steine in eine Reihe bringt gewinnt. Nach jedem Spielzug soll das Spielfeld in der Konsole ausgegeben werden. Sollte das Spielfeld voll sein ohne, dass jemand 4 Steine in einer Reihe hat, geht die Runde unentschieden aus. Nutzereingaben werden direkt an den Server geschickt, der diese auswertet.

Nach Beendung der Verbindung zum Client soll auf eine neue Verbindung gewartet werden.

Das Spielfeld besteht aus 6 Zeilen und 7 Spalten. Aus ASCII Zeichen soll ein symbolischer Spielplan aufgezeichnet werden. Die Nutzer Eingabe erfolgt über eine Zahl von 1-7 welche je für eine Spalte steht, in welcher der Stein auf den untersten freien Slot platziert wird.

Sehr geehrte Team-Mitglieder,

Das Thema "4-Gewinnt" wurde öfters gewählt, lässt sich aber mit unterschiedlichen Features umsetzen. Da Sie eine 3er-Gruppe sind, fügen Sie folgende Anforderungen hinzu:

- 1) es sollen mehrere Spiele parallel stattfinden können
- 2) legen Sie fest wie Spieler zugeordnet werden, wenn mehrere Spiele parallel stattfinden
- 3) die Anzahl an gleichzeitig stattfindenden Spielen wird am Ende jeden Spiels neben dem Spielausgang auch angezeigt
- 4) es soll eine Maximalzeit geben, um einen Zug durchzuführen, ansonsten hat der Spieler verloren
- 5) der Sieger wird erst nach einer vorgegebenen Anzahl von Spielen bestimmt
- 6) zwischen den Spielen muss eine maximale Zeit für den nächsten ersten Zug eingehalten werden, ansonsten wird das Spiel abgebrochen
- 7) die Applikationen sollen wie folgt ausgeführt werden können:
viergewinnt_server [-p <port>][-n <number of games>]
viergewinnt_client [-i <server ip>] [-p <server port>]

Der Parameter für die -p-Option gibt den UDP-Port des Sockets an, über den kommuniziert werden soll, die -n-Option gibt die Anzahl der durchzuführenden Spiele an. Der Server soll auf allen möglichen Schnittstellen/IPs lauschen. Beim Client soll zusätzlich mittels -i Parameter die IP-Adresse des Servers angegeben werden können. Sollte dies fehlen, wird 127.0.0.1 verwendet. Wird kein Port übergeben, soll ein passender, vordefinierter Port verwendet werden, der in Server und Client gleich ist (z.B. die User-ID eines Autors). Sollten ungültige Parameter übergeben werden, soll zumindest eine usage message ausgegeben werden.

Ich habe Ihnen die Team-Kennzeichnung "G" zugeordnet.

mit freundlichen Grüßen

Herbert Newald