

Nätverksprogrammering, EDA095

Projekt: Chess game, 2013-05-21

Handledare: Roger Henriksson

Axel Hildingsson, a.hildingson@gmail.com

Hoang Huyuh Truong, artiq90@yahoo.se

Lisa Lindberg, rys07lli@student.lu.se

Michal Sadowski, misad90@gmail.com

Bakgrund

Tanken med gruppuppgiften i EDA095 är att sätta den teori som gått igenom under kursen i praktik genom valfritt projekt. Fokus ska ligga på själva nätverkskommunikationen som kan realiseras i form av exempelvis ett chattsystem, ett spel eller internettelefoni. Upplägget av projektet bör i samtliga fall utgöras av implementation av en lite enklare grundidé som därefter utvecklas med passande features. Samtliga system bör omfatta mer än endast kommunikation mellan 2 användare, klienter. För att nå en lämplig nivå bör exempelvis i ett spel antingen detta möjliggöra spel för multiplayer eller att flera deltagare kan vara involverade på något sätt. Uppgiften innebär att upprätta lämplig server och klient som kan hantera funktioner som deltagarlistor, gränssnitt, regelverk för hur användaren får bete sig m.m. Redovisning sker i form av en demonstration av det färdiga systemet, en hemsida som presenterar syfte, upphovsmän och projektet i allmänhet samt en övergripande skriftlig rapport.

Kravspecifikation

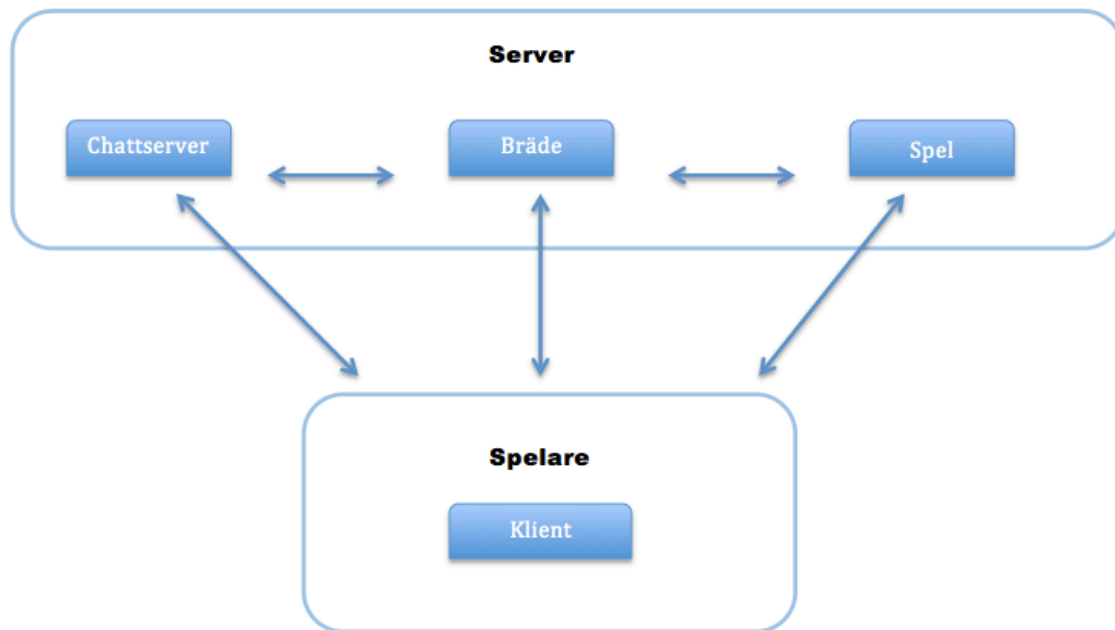
Efter att ha valt att göra ett spel i någon form blev den slutgiltiga projekttiden att konstruera ett schackspel. "Chess" grundläggande funktioner kom att definieras som ett ordinärt schackspel där användare av Chess kan spela ett parti schack med varandra. Man har även möjlighet att endast observera spel och med andra observatörer diskutera spelet medan det förlöper. Även spelarna kan under spelet chatta med varandra.

- Vi uppstart av programmet ska en lista över aktiva spel visas. Denna ska för varje spel innehålla ett spel ID, huruvida spelet är igång eller i väntan på en motståndare samt den angivna tiden för varje spelares drag.
- En användare ska kunna starta ett nytt schackspel där denne inväntar en motståndare.
- En användare ska kunna ansluta till ett spel som väntar på en motståndare och starta detta spel.
- En användare ska kunna ansluta till ett påbörjat spel för att endast åskåda detta.
- En användare ska vid start av ett nytt spel kunna välja vilken färg denna vill ha.
- Gränssnittet ska utgöra regelboken som avgör huruvida ett drag är giltigt eller ej.
- De två spelarna ska kunna skriva till varandra i en chatt som inte är synlig för åskådarna.
- Åskådarna ska kunna chatta med varandra i en chat som inte är synlig för spelarna.
- Alla drag som görs ska publiceras för samtliga anslutna samt om de är utförda av svart eller vit spelare.
- Vid "schack" ska spelaren som befinner sig i just schack endast kunna genomföra drag som motverkar schack.
- I en spelomgång har spelarna max 10 minuter var på sig att slutföra spelet där varje spelares återstående tid ska visas för samtliga anslutna.
- Går tiden ut för en spelare förlorar denna.
- En användare ska då den ansluter till ett spel kunna välja ett användarnamn.

- En användare ska då den startar systemet få en lista på alla aktiva spel och kunna välja vilket denna vill ansluta till.
- Listan med aktiva spel ska innehålla spelets ID, hur många aktiva användare som är anslutna och huruvida det finns en eller två spelare.
- Alla deltagare i ett aktivt spel ska visas i en lista i gränssnittet.

Modell

Genomgående i programmet skickas all data via TCP.



De viktigaste klasserna i programmet är

Klass: chess_main

Mainmetoden som startat upp chess servern. Lyssnar efter nya användare och då en användare ansluter skapas i chess_main ett user-objekt som tilldelas en userthread.

Klass: user

Definierar en användare genom att tilldela denna ett användarnamn, en socket för chatt och en för spel. En user kopplas till ett board då denna har valt vilket spel denna vill ansluta till.

Klass: userthread

Hanterar näst intill samtliga kommandon som klienten skickar. Hanterar inte chatten.

Klass: boardList

Liknande en databas för att hålla reda på alla pågående spel samt användare. Chess_main skapar en boardList som skickas ut till/används av alla userthreads.

Klass: board

Omfattar bräde, drag och användare för varje spel. Skapar ytterligare, bredvid brädet i GUI:t, ett bräde som håller reda på drag som skickas ut till spectators och huruvida ett spel tar slut. Då ett bräde skapas upprättas även en ny chattråd. Board, håller även reda på vilken färg en användare har.

En session börjar med ett meddelande från klienten, ett s.k. "Hej"-meddelande. Servern svarar med en lista över alla användare. Därefter skickar klienten sitt användarnamn som matchas mot användarlistan för att förhindra att dubletter läggs in. Då användaren har valt ett användarnamn som inte existerar får denna av servern en lista med pågående spel. Därefter väntar servern på svar från klienten som kan välja att starta ett nytt spel, gå med i ett befintligt och därmed välja ett spelID eller lämna alla aktivitet genom disconnect.

Beroende på vilket kommando servern mottar fortlöper kommunikationen på följande sätt:

Nytt spel

Ett boardobjekt skapas dit användaren kopplas samt startas en chattservern igång för spelet. Därefter väljer användaren en färg och väntar på att en motståndare ska ansluta.

SpelID

Användaren har valt att ansluta till ett spel där en användare ligger och väntar på en motståndare. Servern svarar med en lista över alla användare ansluta till det valda spelet samt vilken annan användare som startat spelet och vilken färg denna har valt. Klienten skickar antingen den motsatta färgen eller "spectate".

Disconnect

Break, sessionen avslutas.

Spectate

Har användaren valt spectate kommer den endast ta emot de drag spelarna gör fram till det att spelet avslutas. Klienten kommer även lyssna efter nya användare i spelet.

Spelomgången

Spelaren som valt vit färg gör första draget och skickar till servern koordinaterna från vilken plats samt till vilket plats en pjäs ska flyttas. Spelaren skickar även hur lång tid spelaren tog på sig innan denna gjorde draget. (Överstiger tiden 10 minuter vinner motståndaren och spelet avslutas.) Servern skickar vidare draget samt tiden till alla eventuella spectators och den andra spelaren. Därefter svarar den svarta spelaren med liknande information.

Chatten

Parallellt med spelet startas en chattsession där andra socket:ar (än de i spelet) används. Chattservern har två listor, en för spelarna och en för åskådare. Spelare chattar med spelare och åskådare med åskådare.

Användarhandledning

Genom att starta Chess fås en lista med alla aktiva spel. Varje spels ID visas samt tiden varje spelare har på sig att slutföra en omgång respektive spel. Alternativt finns inget aktivt spel vilket innebär att enda alternativen för en ny användare är att vänta på att ett spel startas eller starta ett nytt själv.

Fortsättningsvis kan användaren alltså starta ett nytt spel genom att klicka på *create table* och därefter välja vilken färg denna vill spela. Görs inget val får spelaren vit färg. Därefter är det bara för spelaren att vänta på att en motståndare ansluter.

En användare kan också välja att endast åskåda ett spel som *spectator* genom att välja *spectate*.

Vill användaren utmana en spelare som väntar på en motståndare trycker man på *join* och därefter *sit down*.

Som spelare gör man ett drag genom att i gränssnittet flytta pjäserna på brädet. Ogiltiga drag går ej att utföra.

I spelet finns en chatt för spelare och en för åskådare där spelare kan chatta med spelare och åskådare med åskådare genom att helt enkelt skriva i respektive chattfält.

När ett spel avslutas raderas spelare och åskådare från listan över anslutna i det avslutade spelet. Man kan också lämna ett spel genom att stänga ner det.

Utvärdering

Projektet har fortlöpt i enlighet med vår kravspecifikation och vi anser att vi uppfyllt samtliga krav. All funktionalitet finns i systemet tillsammans med ett tydligt gränssnitt. Gränssnittet hade dock kunnat utvecklas och förfinats mot ett större tidsspann. Fokus har, enligt projektets syfte, legat på funktionalitet och kommunikation server-klient. Flera features hade kunna adderats för att ge ett mer informativt gränssnitt som exempelvis att man skriver ut vilken spelare som gör vilket drag, hur många spel en användare har spelat och hur många vinster/förluster denna har. Ytterligare en feature vi hade under diskussion vid projektets uppstart var en eventuell röstkommunikation mellan spelarna. Liknande Skype skulle dessa kunna spela och chatta, som i vårt aktuella spel, men till det också få

möjlighet till att prata med varandra. Denna röstkommunikation påbörjade vi aldrig p.g.a. tidsbrist men hade varit en rolig extrafunktion att implementera som hade tillfört ytterligare en dimension till vårt spel. En implementation av detta hade inneburit ytterligare en kanal att lyssna på i form av input från externa komponenter vi hade behövt införskaffa.

Samtliga i gruppen är generellt nöjda med projektuppgiften som vi anser har bidragit till väsentliga kunskaper inom kursens ramar. Friheten i val av projekt gör den mer intressant då man får möjlighet att själv implementera något man vill lära sig eller tycker är extra kul. Nivån har också varit rimlig då labbarna har varit bra som förberedande underlag samt eftersom friheten också här bidrar till en individuellt anpassningsbar uppgift där man gruppvis kan lägga ribban så högt man önskar.

Programlistor

Se bifogad fil i mejlet.