

Projekt i Nätverksprogrammering EDA095

Adrian Hansson – dic13aha@student.lu.se
Björn Hansson – bjorn.joakim.hansson@gmail.com
Eskil Andersson – tfy11ean@student.lu.se
Jonatan Wede – ada07jwe@student.lu.se

1. Bakgrund

Projeket ämnade att resultera i ett Java-program där kunskap i och förståelse för nätverksbaserad programmering påvisas. Vi valde att skapa en multiplayer-version av det världskända spelet Snake. Utöver spelmekaniken tillkom att konstruera ett TCP-baserat protokoll för att skicka information mellan server och klienter samt att dela upp programmet i två delar: spelmodellen på serversidan och uppritning av spelgrafiken på klientsidan.

2. Kravspecifikation

Systemet består av en server och ett godtyckligt antal klienter som kan koppla upp sig mot denna. Varje spelare (klient) som kopplats upp mot server ges kontroll över en orm i spelet. Klienten kan skicka tangent-kommandon till servern för att ange i vilken riktning dess orm ska färdas. Om en orm kolliderar med en annan dör ormen. Om en orm når ett äpple växer ormen och ett nytt äpple dyker upp. Äpplenas position slumpas ut på spelplanen. Servern hanterar världsmodellen och all spelmekanik. Servern skickar information till klienterna – enbart det som krävs för att klienten ska kunna rita upp spelvärlden på skärmen.

3. Modell

3.1 Klientsidan

Den trådade klassen Client lyssnar efter input från servern och tar den emot sådant skickar den det vidare till Level. Client kan även skicka information till servern – information om vilken knapp på tangentbordet användaren tryckt på.

Level är en utökning av Java-klassen JPanel. Den används som en komponent i GUI:et och innehåller specialiserade metoder för att rita upp ormar. Den lyssnar även efter knapptryckningar och kallar på metoder i Client om en knapptryckning registreras.

View innehåller en main-metod och består av ett antal Swing-komponenter som utgör GUI:et.

3.2 Serversidan

Snake är en klass som beskriver en orm. Där finns metoder för hur ormen ska röra sig och hur förflyttning av segment hanteras. Varje orm tillhör en klient och således finns information om sockets och uppkopplingar här.

Server-klassen är huvudklassen på serversidan och den tar emot uppkopplingar och startar nya Snake-trådar.

3.3 Transport

De två transportprotokoll-klasserna konverterar mellan information att läsa (ints) och bytes att skicka. Vi använder en kombination av TCP och egna regler för hur data skall skickas och avläsas.

3.4 Sammanfattning

Servern hanterar modellen för spelet. Separata trådar körs på serversidan och kommunicerar med var sin spelare. På klientsidan läses knapptryckningar av och servern notifieras om en pil-tangent tryckts ned. Klienten tar även emot data från servern och använder den för att rita upp spelgrafiken. Rendering sker hos klienten medan beräkningar och spelmekanik sker hos servern.

4. Användarhandledning

En klient startar spelet genom att köra klassen View eller Client.jar.

Piltangenterna UPP, NER, HÖGER och VÄNSTER används för att dirigera spelarens orm i respektive riktning.

Målet är att äta de röda äpplena så att ens orm kan växa.

Man ska undvika kollision med sin egen kropp samt med andra ormar.

5. Utvärdering

Målen uppnåddes väl och majoriteten av arbetet flöt på bra. Den stora utmaningen var att bestämma sig för ett väl fungerande transportprokoll, vilket vi fick skriva själva. Skrivandet av protokollet ledde även till insikt om hur koden kunde optimeras för att minska belastningen över nätet.

Det planerades bara för en ganska enkel version av spelet, där mer avancerade funktioner skulle läggas till om det fanns tid över. I efterhand insåg vi att bättre planering av dessa avancerade funktioner kunde gjorts redan innan, vilket avsevärt hade förenklat arbetet när man nått den punkten i projektet.

Projektet i sig var en rolig uppgift och kändes givande. Det var motiverande att man själva fick välja ett projekt, då man kunde jobba på något man själv ville skapa. Uppgiften blev ett effektivt och roligt sätt att låta oss lära oss mer om nätverksprogrammering.

6. Programlistor

<https://github.com/adense13/MultiSnake/>

(Vid icke-fungerande länk kontakta en projektmedlem om ny länk eller en zip-fil)