Luleå Tekniska Universitet **8 oktober 2014**
Laborationsrapport

Laboration 3

D0036D

JavaGameServer & Client

Namn Magnus Björk
E-mail magbjr-3@student.ltu.se

Handledare
Örjan Tjernström

Innehåll

1 Introduktion

Laborationsuppgiften var att programmera ett nätverksspel. Syftet med uppgiften var att man skulle utvidga sin kunskap gällande:

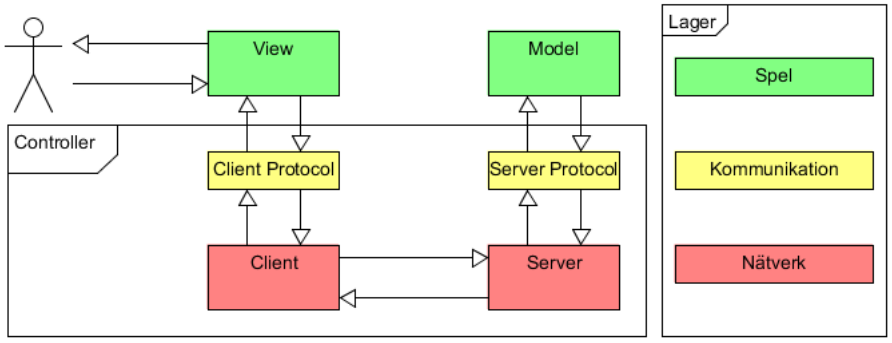
- **Nätverksprotokoll**
  - **TCP**
    Pålitligt protokoll för dataöverföring. Lämpligt att använda vid viktiga tillfällen i spelet. *T.ex. när en klient ansluter eller lämnar en server.*
  - **UDP**
    Opålitligt men snabbt protokoll för dataöverföring. Snabbare än TCP på grund av att det dels inte skickar tillbaka något ACK-meddelande till sändaren samt att UDP har en mindre 'frame'. Lämpar sig bäst för uppdateringar av förflyttningar på grund av sin snabbhet.
- **Sockets**
  Spelet skulle innehålla kommunikation över nätverket, detta sköts av diverse sockets:
  - **DatagramSocket**
    Hanterar trafik av protokollet UDP. Kan skicka unicast till en annan DatagramSocket eller multicast genom att skicka till en multicast-grupp. *T.ex. Önskemål om förflyttning från klient till server.*
  - **MulticastSocket**
    Hanterar inkommande multicast-trafik av protokollet UDP. En MulticastSocket går med i en multicast-grupp för att få ta del av dess trafik. *T.ex. uppdateringar av spelarnas position från server till klienter.)*
  - **ServerSocket**
    Serversocket tar emot trafik av protokollet TCP. Ansluter en klient till servern kan man sedan skapa ett Socket objekt för att hantera strömmen.
  - **Socket**
    Hanterar TCP-trafik. Skapas av ServerSocket på server-sidan. Klienten skapar den socket som tar kontakt med servern.
- **Trådning**
  En del moment måste köras parallellt och då kan man använda trådar. *T.ex. så har varje klient en tråd på servern som den kommunicerar med.*

2 Metod

Min strategi för att lösa uppgiften var att dela upp koden i 3 lager. Detta för att göra det enkelt att lägga på funktioner. *T.ex. Uppgiften att flytta uppdateringsmeddellanden från TCP till UDP var relativt smärtfri, hade spelmekanik varit inblandad i denna del hade det varit betydligt värre.*

- **Nätverk** Nätverkskoden skall endast skicka meddellanden mellan servern och klienten. Ingenting som har med spel-lagret att göra skall finnas här.
- **Spel** Spelets logik och utseende skall endast påverkas av kommunikationslagret av programmet.
- **Kommunikation**
  Det lager som sköter kommunikation mellan Spel- och Nätverks-lagren.

2.1 Design



Jag ville skapa ett dataflöde liknande bilden ovan.

1. Spelaren trycker på en tangent för att flytta sin markör.
2. Klient-protokollet skapar ett meddelande med användarens indata och skickar detta till nätverkslagret.
3. Nätverkslagret skickar nu meddelandet från klienten till servern.
4. Server-protokollet tar emot och bearbetar modellen efter anvisningar i meddelandet. Skapar ett nytt meddelande som skickas tillbaka till klienten via nätverkslagret.
5. Klient-protokollet tar emot meddelandet och ändrar eventuellt vyn som visas för spelaren.
6. Spelaren upptäcker förflyttningen (Om den var giltig) och kan nu ge ny indata om den så vill.

3 Resultat

3.1 Användarhandledning

3.2 Handskakning

3.3 Applikationsprotokoll

3.3.1 Klient

3.3.2 Server

4 Diskussion

4.1 Synchronize