TDT4190

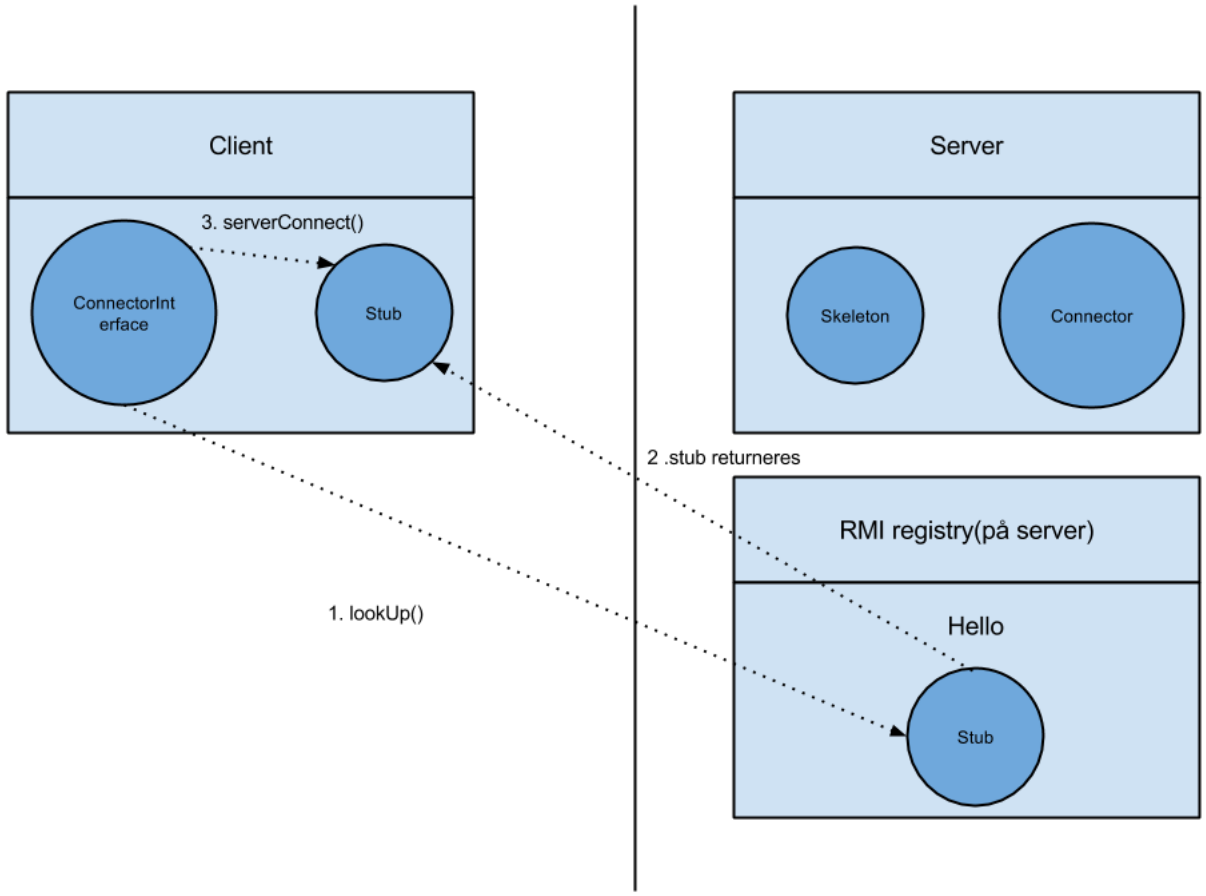
Øving 1

Kjetil Sletten, Simen Skoglund og Christian Peter

February 7, 2014

1 Kort beskrivelse

Vi har valgt å lage en egen klasse som implementer Java RMI. Hver klasse er beskrevet nedenfor. Her er et overblikk over det som skjer:



Klienten må ha tak i stub-objektet, for å få tak i dette må klienten da gjøre et lookup. Siden RMI registry allerede har laget stuben som klienten leter etter, trenger klienten bare å spørre om denne stubben finnes. Deretter vil RMI reqgistry returnere stubben, som klienten da vil snakke sammen ved å overføre de metodene som er blitt definert i interfacet.

2 Connector.java

public void serverConnect()

Denne koden vil bli kjørt hvis det ikke finnes noen server. Da vil isServer bli satt til true slik at vi kan bruke dette til å bestemme hvilket merke den skal bruke. Vi benytter oss av Registry der vi åpner det på port 3050 for øyeblikket.

public ConnectorInterface clientConnect()

Hvis det ikke går ann å sette opp en server, vil clientConnect kjøres. Denne metoden vil enten retunere servers grensesnittet eller klient grensesnittet slik at klassen TicTacToe.java kan bruke denne.

public boolean setMark(int x, int y, char mark)

Metode for å sette merke(O eller X) på motstanderens skjerm.

public boolean serverTurn(boolean server)

Denne metoden setter om det er serveren sin tur til å spille.

public boolean isServer()

En enkel metode for å se om man selv er server eller klient.

public void setOpponent(ConnectorInterface opponent)

Når klienten har koblet seg opp, er det ikke noen automatikk i at serveren vet hvem denne klienten er. Derfor lagde vi denne metoden som setter motstanderen for serveren så den kan sende merker tilbake til klienten når det spilles.

public void setGameIsWon(boolean gameIsWon)

For å hindre at det skal kunne spilles videre når en har vunnet må man si ifra til motstanderen at man har vunnet. Dette gjør denne metoden.

public void resetGame(int boardSize)

Dette er en ekstra funksjonalitet som vi har lagt til. Det vi gjør her er å starte spillet på nytt på både server og klient.

3 ConnectorInterface.java

Grunnen til at vi har med denne er at vi trenger et interface som kan brukes av klienten på den andre siden. Dette grensesnittet implementerer Remote. Metodene som ligger her er beskrevet i Connector.java seksjonen.

4 BoardModel.java

public void cleanBoard()

Denne metoden måtte jeg legge til for å fjerne alle kryss og sirkler. Ingenting som har med RMI er gjort her.

5 TicTacToe.java

I denne klassen er det også implementert noen hjelpe metoder for sette motstander, om spillet er vunnet og hvem sin tur det er.

public void valueChanged(ListSelectionEvent e)

I denne metoden har vi lagt til en del nye ting. Den største forskjellen nå er at man bare kan gjøre et trekk hvis det er din tur. Det er også i denne metoden at vi sier ifra til motstanderen hva som har skjedd samt at vi oppdaterer vårt eget brett som før. Det er heller mulig å spille videre hvis en spiller har vunnet.

public void actionPerformed(ActionEvent e)

Denne metoden er ikke en del av øvingen. Det som skjer her er at vi starter spillet på nytt igjen. Vi sier ifra til motstanderen om at spillet er startet på nytt.