

# **Verkkopokerijärjestelmä**

Suunnitteludokumentti, 1. iteraatio

Ryhmä Kanat

Ohjelmistotuotantoprojekti, syksy 2008

## **Projektiryhmä**

Samuli Aalto-Setälä

Jukka Kekäläinen

Jarno Kyykkä

Mika Mielonen

Mårten Smeds

Otto Waltari

## **Ohjaaja**

Paula Kuosmanen

## **Asiakas**

Jussi Kangasharju

## **Projektin tiedot**

### **Kotisivu**

<http://www.cs.helsinki.fi/group/kanat>

### **Versiohistoria**

| Versio | Päivämäärä | Kuvaus  |
|--------|------------|---|
| 0.1    | 27.9.2008  | Ensimmäiset hahmotelmat rajapinnoista               |
| 0.2    | 07.10.2008 | RC1 – Rajapintakuvaukset ja tietoliikenneprotokolla |
| 0.3    | 07.10.2008 | RC2 – Tarkennettu luokkakuvauksia                   |
| 0.4    | 8.10.2008  | RC3 – Sekvenssikaavioita lisätty                    |

# Sisällysluettelo

|   |    |
|---|----|
| Johdanto.....                                 | 5  |
| Sanastoa[1].....                              | 6  |
| Korttikädet .....                             | 6  |
| Pelin kulku / jako.....                       | 7  |
| Muuta.....                                    | 9  |
| Säännöt ja esimerkkejä[2].....                | 9  |
| Ennen kuin kortit on jaettu .....             | 9  |
| Kun kortit on jaettu .....                    | 10 |
| Preflop eli ensimmäinen panostuskierros ..... | 10 |
| Floppi eli toinen panostuskierros.....        | 11 |
| Turn eli kolmas panostuskierros.....          | 11 |
| River eli neljäs panostuskierros.....         | 12 |
| Arkkitehtuurisuunnitelma (1. iteraatio).....  | 13 |
| Palvelin.....                                 | 13 |
| Asiakassovellus ilman tekoälyä.....           | 13 |
| Asiakassovellus tekoälyllä.....               | 13 |
| Rajapintakuvaukset.....                       | 13 |
| Luokkakaaviot.....                            | 13 |
| Palvelin.....                                 | 14 |
| Asiakas ilman tekoälyä.....                   | 14 |
| Asiakas tekoälyllä.....                       | 15 |
| Sekvenssikaaviot.....                         | 16 |
| Palvelin.....                                 | 16 |
| Asiakassovellus.....                          | 17 |
| Luokkametodien kuvaukset.....                 | 17 |
| Palvelin.....                                 | 17 |
| Pelilogiikka.....                             | 17 |
| Pelaaja.....                                  | 18 |
| Tilipali.....                                 | 19 |
| Asiakassovellus ilman tekoälyä.....           | 19 |
| Kasittelija.....                              | 20 |
| UI.....                                       | 20 |
| Tilipali.....                                 | 20 |
| Asiakassovellus tekoälyllä.....               | 21 |
| Tekoaly.....                                  | 21 |

|   |    |
|---|----|
| Tilipali.....                               | 21 |
| PokerEngine (ulkopuolinen komponentti)..... | 22 |
| Deck.....                                   | 22 |
| Card.....                                   | 22 |
| Hand.....                                   | 22 |
| Board.....                                  | 22 |
| HandRank.....                               | 23 |
| Tietoliikenneprotokolla.....                | 23 |
| Viestin rakenne.....                        | 23 |
| Pelikortin esitys verkkoliikenteessä.....   | 23 |
| Esimerkki 1. ....                           | 24 |
| Esimerkki 2.....                            | 24 |
| Viesti tunnukset.....                       | 24 |
| Palvelimen lähettämät viestit.....          | 24 |
| Asiakkaan lähettämät viestit.....           | 25 |
| Lähteet.....                                | 25 |

## Johdanto

Projektin tarkoituksena on kehittää verkkopokerijärjestelmä, jolla voi pelata Texas Hold'em -peliä fixed limit -panostuksella. Sovelluksesta tehdään sekä palvelin-asiakas- että vertaisverkkopohjalla toimiva versio. Palvelinversiossa palvelin hallinnoi pelin etenemistä ja vaatii asiakasohjelmien oikeellisuutta, kun taas p2p-versiossa palvelimen tehtävät hajautetaan vertaisverkon koneille. Tällöin tulee varautua jollain tavalla siihen, että jakaja mahdollisesti huijaa.

Peliin kehitetään myös tekoäly, joka voi toimia ihmispelaajan asemassa. Sen tulee osata pelata ainakin siedettävästi, eli osata tehdä jotakuinkin järkeviä ratkaisuja korttiensa ja panostusten suhteen.

# Sanastoa<sup>[1]</sup>

## ***Korttikädet***

The highest card

Korkein kortti, hai.

Pair

Pari

Two pairs

Kaksi paria, rattaat, tupee.

Three of a kind

Kolmoset, kolme samaa.

Straight

Suora.

Flush

Väri.

Full house

Täyskäsi.

Four of a kind

Neloset, neljä samaa.

Straight flush

Värisuora

Royal flush

Värisuora, reeti, kuningasvärisuora (ässä, kuningas, rouva, jätkä ja kymppi).

Kicker

Esim. 5, 5, J, J, A on kahdet parit ässä-kickerillä, kikkeri. Esim. jos pelaajien kädet olisivat AK ja KQ ja pöytä 5, 5, J, J, Q - ensimmäinen voittaisi "ässä kikkerillä ja kahdella parilla" ja toinen häviää "kuningas kikkerillä ja kahdella parilla".

## **Esimerkkejä korttikäsistä**

Jos pöytä Riverillä olisi esim. 5, 5, J, J, Q ja pelaajilla kädessä:

-AK ja A4: potti jaetaan puoliksi, eli molempien käsi olisi 5, 5, J, J, A

-33 ja 44: potti jaetaan puoliksi, koska "pöytä pelaa" (table plays), eli molempien käsi olisi 5, 5, J, J, Q.

-AK ja 66: jälkimmäinen voittaa, eli kädet olisivat 5, 5, J, J, A ja 6, 6, J, J, Q

-A5 ja AJ: jälkimmäinen voittaa, eli kädet olisivat 5, 5, 5, J, J ja 5, 5, J, J, J

## **Erilaisia panoksia**

Big blind

Mm. Texas Hold'Em:ssa käytettävä alkupanos,

|  |  |
|--|--|
|  | jonka seuraava Small blindista maksaa. "Iso blindi/sokea".   |
| Small blind  | Toinen alkupanos, jonka seuraava Dealer Buttonista aina maksaa. Puolet Big blindista. Jos pelaaja haluaa osallistua peliin, maksaa hän toisen puolen. "Pieni blindi/sokea".                |
| Table stakes   | Tarkoittaa, että vain pöydällä olevilla merkeillä voi pelata. Kesken jaon ei voi lisätä merkkejään, eikä missään vaiheessa saa ottaa merkkejä pois pöydältä ennen pelaamisen lopettamista. |
| <b>Positio</b> (9 pelaajaa, jos vähemmän, poistuvat järjestyksessä MP3, UTG+1, MP2,1, UTG, CO) |  |
| Button / Dealer button   | Mm. Texasissa käytetty valkoinen kiekko, joka osoittaa teoreettisen jakajan paikan. Korttien jakaminen aloitetaan Buttonista seuraavalle pelaajalle. "Nappi".                              |
| Small blind, Big blind   | Buttonista kaksi seuraavaa pelaajaa. Ensimmäinen panostuskierros (Preflop) päättyy Big blindiin. Loput panostuskierrokset (Flop) alkavat Small blindista.                                  |
| UTG, UTG+1   | Big blindista kaksi seuraavaa pelaajaa. Ensimmäinen panostuskierros alkaa UTG:sta, kun Small blind ja Big blind on asetettu. "Under the gun".  |
| MP1, MP2, MP3  | UTG+1:sta kolme seuraavaa pelaajaa. "Middle position".   |
| CO / Cut off   | Viimeinen pelaaja ennen Buttonia.  |

## ***Pelin kulku / jako***

|         |   |
|---------|---|
| Preflop | Ensimmäiset pelaajille jaetut pimeät kortit, joita muut eivät näe. Korttien jakamisen jälkeen ensimmäinen panostuskierros, ennen kuin |
|---------|---|

|       |   |
|-------|---|
|       | ensimmäiset avoimet (yhteiset) kortit (Flop) jaetaan pöytään. |
| Flop  | Ensimmäiset kolme avointa korttia ja toinen panostuskierrös.  |
| Turn  | Neljäs avoin kortti ja panostuskierrös.                       |
| River | Viides avoin kortti ja panostuskierrös.                       |

Board = Pöydällä olevat avoimet kortit.

### Toimintavaihtoehtoja

|                  |  |
|------------------|--|
| Check            | Luovuttaa vuoron seuraavalle pelaajalle, ei panostusta. "Sököttää".                            |
| Bet              | Panos tai lyönti. "Panostaa, lyödä, betsata."  |
| Call             | Edellisen lyönnin maksu. "Maksaa, koolata".  |
| Fold             | Luovuttaa peli ja pelikortit. Ei maksa toisen pelaajan tekemää korotusta. "Foldata, luovuttaa" |
| Raise / Re-raise | Edellisen lyönnin (Bet tai Raise) korottaminen. "Korottaa, reissata, rereissata".              |

### Panostuksesta

|   |  |
|---|--|
| Fixed limit (käytetään meidän pokeriohjelmassa) | Esim. 10/20€ Fixed limitissä panos ja korotus on Preflop 5€, Flop 10€ ja Turn, River 20€. Esimerkiksi pelattaessa 10/20 limit-peliä, kahdella ensimmäisellä panostuskierröksellä panokset ja korotukset tehdään kymmenen ja seuraavilla kierroksilla kahdenkymmenen yksiköinä. Jokaisella panostuskierröksellä voi panosta korottaa maksimissaan kolme kertaa (esim. cap = 10 + 10 + 10 + 10). |
| Cap   | Potti on niin sanotusti lyöty cappiin. Se tarkoittaa sitä, että korottaminen ei ole enää sallittua.  |



Potti

Jokaisella panostuskierroksella sallitaan vain kolme korotusta eli panostuksia on yhteensä siis neljä. Kun potti on lyöty cappiin, jäljellä olevat pelaajat voivat joko luovuttaa tai maksaa.

Jaon aikana maksetut panokset ja tehdyt panostukset sekä korotukset. Esim. Small blind + Big blind + korotus + maksu jne.

## ***Muuta***

Buy-in = Pelimerkkien ostoa pöytään, esim. 50-100€.

Stack = Pelaaalla jäljellä olevan rahan määrä.

Heads-up = Jaossa tai pöydässä on vain kaksi pelaajaa. "Peli on Heads-Up". Mikäli pöydässä on tasan kaksi pelaajaa, Button maksaa Preflopissa pienen blindin ja hänestä seuraava ison blindin. Buttonilla on myös aluksi puheenvuoro, eli hän päättää luovuttaako, maksaa vai korottaako. Preflopin jälkeen kolmella seuraavalla panostuskierroksella (Flop, Turn, River) isolla blindillä on puheenvuoro.

Show down = Kaikkien pimeiden korttien näyttäminen jaon lopussa. "Näyttö".

## **Säännöt ja esimerkkejä<sup>[2]</sup>**

Limit Texas Hold'em -pelissä, joka on Nettipokeri-projektissamme käytettävä pelimuoto, panokset ovat kiinteitä ja määrättyjä. Pelissä on pienet panokset ja isot panokset. Pieniä panoksia käytetään kahdella ensimmäisellä panostuskierroksella (Preflop, Flop) ja isoja panoksia kolmannella ja neljännellä panostuskierroksella (Turn, River). Isot panokset ovat aina tuplakokoisia pieniin panoksiin verrattuna. Panosten suuruus riippuu aina pelin koosta. Pelin koko ilmoitetaan esim €10/€20, eli pieni panos €1 ja iso panos €2. Iso blindi on aina yhtä suuri kuin pieni panos ja pieni blindi puolet isosta blindista. Esim. €10/€20 Limit Texas Hold'emissa pieni panos (eri asia kuin pieni blindi) on €10 ja iso panos on €20. Niinpä iso blindi on siis €10 ja pieni blindi €5.

## ***Ennen kuin kortit on jaettu***

- Yksi pelaajista toimii jakajana. Jakajan paikan, eli jakovuoron, osoittaa Dealer Button. Button määrää, missä järjestyksessä pelaajat toimivat kyseisessä jaossa. Button kiertää myötöpäivään jokaisen pelatun jaon jälkeen.
- Ensimmäinen pelaaja jakajan paikalta (Buttonista) vasemmalle asettaa pöytään pakollisen panoksen, jota kutsutaan pieneksi blindiksi. Toinen pelaaja jakajasta vasemmalle asettaa pöytään myös

pakollisen panoksen, jota kutsutaan isoksi blindiksi.

### ***Kun kortit on jaettu***

- Kortit jaetaan myötöpäivään alkaen pelaajasta, joka istuu ensimmäisenä jakajasta vasemmalle (pieni blindi) päättyen pelaajaan joka toimii jakajana (Button). Jokainen pelaaja saa kaksi pimeää taskukorttia, joita muut eivät näe. Tätä kättä kutsutaan aloituskädeksi.

### ***Preflop eli ensimmäinen panostuskierros***

- Toiminta alkaa pelaajasta, joka istuu isosta blindista ensimmäisenä vasemmalle, eli UTG:sta (paikka nro. kolme). Tällä pelaajalla on vaihtoehtoina joko Fold (luovutus), Call (maksaa iso blindi) tai Raise (korottaa, eli asettaa kaksi pientä panosta eteensä).
- Vuoro siirtyy nyt pelaajalle, joka istuu UTG+1:ssä (paikka numero neljä eli vasemmalle paikasta kolme). Tällä pelaajalla on nyt mahdollisuus luovuttaa, maksaa, korottaa tai uudelleen korottaa (tosin vain jos pottia on korotettu ennen häntä).
- Panostuskierros jatkuu myötöpäivään, kunnes kaikki pelaajat, jotka eivät ole luovuttaneet, ovat laittaneet saman verran rahaa pottiin.

Esimerkiksi:

- Kyseessä on €10/€20 Fixed Limit Texas Hold'em. Pöydässä istuu yhdeksän pelaajaa. Kortit jaetaan ja UTG-position pelaaja (ensimmäisenä vuorossa) päättää korottaa. Hänen pitää siis laittaa pottiin €20 eli kaksi pientä panosta. Pelaajat paikoilta 4,5, ja 6 päättävät luovuttaa. Pelaaja, joka istuu paikalla numero 7, päättää uudelleen korottaa ja hänen on nyt laitettava pottiin €30 (kolme pientä panosta). Pelaaja paikalla numero 8 päättää maksaa ja hänen on siis laitettava pottiin €30 (kolme pientä panosta). Buttonissa eli jakajan paikalla istuva pelaaja luovuttaa.
- Pienessä blindissa istuva pelaaja, joka on jo laittanut pottiin €5, maksaa ja hänen on siis laitettava €25 pottiin (yhteensä €30 eli kolme pientä panosta). Vuoro siirtyy isolle blindille, joka päättää uudelleen korottaa ja hänen täytyy siis laittaa pottiin yhteensä €40 (Tällä pelaajalla oli jo €10 potissa ja hän joutuu siis laittamaan €30 lisää, että kokonaispanostus olisi 40€ eli neljä pientä panosta).
- Ison blindin pelaajan uudelleen korotus tarkoittaa sitä, että potti on niin sanotusti lyöty cappiin (korottaminen ei ole enää sallittua, jokaisella panostuskierroksella sallitaan vain kolme korotusta eli

panostuksia on yhteensä siis neljä). Vuoro palaa alkuperäiselle korottajalle, joka istuu UTG-positiossa.

- Kun ensimmäinen panostuskierros on saatu päätökseen ja kaikki jäljellä olevat pelaajat ovat laittaneet pottiin yhtä paljon rahaa, on aika katsoa floppi.

### ***Floppi eli toinen panostuskierros***

- Blindeja käytetään vain ensimmäisellä panostuskierroksella, pienessä blindissa istuva pelaaja on nyt vuorossa ensimmäisenä (istuu paikalla numero 1). Jos pieni blindi on luovuttanut jo ennen floppia, vuorossa ensimmäisenä flopin jälkeen on iso blindi jne.
- Ensimmäisenä vuorossa olevalla pelaajalla on vaihtoehtoina sököttää (siirtää vuoro myötäpäivään seuraavalle jäljellä olevalle pelaajalle) tai panostaa (jos hän panostaa, on panostettava tasan yhden pienen panoksen verran, eli €10/€20 pelissä €10). Jos kaikki pelaajat sököttävät flopilla, he saavat nähdä neljännen kortin (Turn) ilmaiseksi eli kenenkään ei tarvitse laittaa rahaa pottiin.
- Panostus etenee seuraten samaa kaavaa kuin ensimmäisellä panostuskierroksella. Enimmillään sallitaan siis kolme korotusta tai siis yhteensä neljä pientä panostusta.

Esimerkiksi:

- Kyseessä on €10/€20 Limit Texas Hold'em ja pöydässä istuu kymmenen pelaajaa. Ensimmäisen panostuskierroksen jälkeen pelissä ovat mukana enää pieni blindi, iso blindi ja jakaja. Potti siirretään "pöydän keskelle" ja jakaja jakaa näkyviin kolme yhteiskorttia (Flop). Koska pelivuoro ja panostus siirtyy myötäpäivään, pienessä blindissa istuva pelaaja on vuorossa ensimmäisenä ja hän päättää sököttää. Iso blindi panostaa €10 eli yhden pienen panoksen ja jakaja päättää maksaa eli laittaa potiin myös €10. Nyt on taas pienen blindin vuoro ja hänellä on vaihtoehtoina joko korottaa (laittaa pottiin €20 eli sököttää ensin ja korottaa sitten), luovuttaa tai maksaa (laittaa pottiin €10).
- Kun toinen panostuskierros on saatettu loppuun ja kaikki jäljellä olevat pelaajat ovat sijoittaneet pottiin yhtä paljon, on aika kääntää esiin turn-kortti.

### ***Turn eli kolmas panostuskierros***

- Panostuskierros etenee taas myötäpäivään jakajasta samoin kuten flopilla. Ainoana erona on nyt se, että panokset ovat tuplaantuneet ja niitä kutsutaan isoiksi panoksiksi.

Esimerkiksi:

- Kyseessä on €10/€20 Limit Texas Hold'em ja pöydässä istuu yhdeksän pelaajaa. Flopin jälkeen vain iso blindi ja jakaja ovat enää mukana kädessä. Koska panostus etenee myötöpäivään jakajan paikalta (buttonista) katsoen, iso blindi on vuorossa ensimmäisenä. Iso blindi päättää panostaa €20 eli yhden ison panoksen verran. Jakaja päättää korottaa ja laittaa pottiin siis €40 eli kaksi isoa panosta. Iso blindi päättää maksaa ja laittaa siis pottiin €20 lisää, jolloin hänkin on laittanut pottiin kaksi isoa panosta lisää. Kolmas panostuskierros on nyt saatu päätökseen.
- Kun kolmas panostuskierros on saatu päätökseen ja kaikki jäljellä olevat pelaajat ovat sijoittaneet pottiin saman verran, on river-kortin aika.

### ***River eli neljäs panostuskierros***

Tämän jälkeen panostuskierros etenee taas myötöpäivään jakajasta katsoen samoin kuten flopilla ja turnillakin. Panostukset ovat nyt saman kokoisia kuin turnillakin.

Esimerkiksi:

- Kyseessä on €10/€20 Limit Texas Hold'em ja pöydässä istuu yhdeksän pelaajaa. Turnin jälkeen mukana kädessä ovat enää iso blindi ja jakaja. Panostuskierros etenee jakajasta katsoen myötöpäivään, joten ensimmäisenä vuorossa on isossa blindissa istuva pelaaja. Iso blindi päättää sököttää ja vuoro siirtyy jakajalle. Hän panostaa €20 eli yhden ison panoksen verran. Iso blindi voi nyt luovuttaa, korottaa tai maksaa. Iso blindi päättää maksaa ja hänkin laittaa pottiin €20 eli yhden ison panoksen verran. Jakajan panostus on nyt maksettu ja koska tämä on viimeinen panostuskierros jakaja joutuu näyttämään kaksi taskukorttiaan. Tämän jälkeen hän yhdistää kaksi taskukorttiaan pöydän viiden yhteiskortin kanssa ja muodostaa niistä parhaan mahdollisen viiden kortin pokerikäden. Isolla blindilla on nyt vaihtoehtoina luovuttaa paljastamatta korttejaan (jos hän huomaa, ettei pärjää jakajan kädelle eikä halua paljastaa taktiikkaansa) tai paljastaa kortit näyttääkseen voittavan viiden kortin pokerikäden. Toteuttamassamme Nettipokerissa paras käsi paljastetaan automaattisesti, mutta huonompaa kättä ei ole pakko paljastaa, mikäli on vain maksanut viimeisimmän korotuksen. Molemmat taskukortit täytyy näyttää, jos pelaaja haluaa voittaa potin, vaikka vain yksi kortti riittäisi voittavan käden muodostamiseen.
- Jako päättyy.
- Jakaja siirtää buttonia yhden paikan myötöpäivään. Pakka sekoitetaan ja kortit jaetaan uudestaan. Uusi jako alkaa.

## **Arkkitehtuurisuunnitelma (1. iteraatio)**

Ensimmäisessä iteraatiossa kehitetään kolme eri ohjelmaa, palvelin, ihmisille tarkoitettu asiakassovellus ja tekoälyllä toimiva asiakassovellus. Kaikki eri ohjelmat toteutetaan kaikki Javalla. Ohjelmien välinen keskustelu hoidetaan TCP-socketeilla.

### ***Palvelin***

Peli alkaa, kun se käynnistetään palvelimelta. Peli ei pääty ennen kuin palvelin sammutetaan. Peliin voi liittyä kesken pelin. Palvelimessa on kaikki peliin liittyvä logiikka. Palvelin jakaa kortit asiakasohjelmille. Palvelimella on tallennettuna tieto kaikkien pelaajien korteista, panostuksista ja rahoista. Palvelimen tehtävä on siis vain kysyä asiakkailta mitä he haluavat missäkin tilanteessa tehdä (check/call/raise/fold). Asiakas voi myös missä tahansa tilanteessa kysyä palvelimelta pelin tilaa, jolloin palvelin vastaa kertomalla jaetut kortit, kenen vuoro on pelata, pelaajien rahamäärät ja ovatko he vielä mukana jaossa. Korttien mallinnuksessa ja käsien vertailussa hyödynnetään ulkopuolista kirjastoa. [3]

### ***Asiakassovellus ilman tekoälyä***

Asiakaspalvelin on tarkoitus pitää mahdollisimman yksinkertaisena. Sen takia siihen ei lisätä ollenkaan pelilogiikkaa tai merkittäviä tarkistuksia. Tarkistukset, kuten rahan riittäminen, tehdään kaikki palvelin päässä. Yksinkertaisuudessaan palvelin lähettää asiakkaalle tiedon jaetuista korteista ja kysyy hänen vuorollaan mitä asiakas haluaa tehdä (check/call/raise/fold).

### ***Asiakassovellus tekoälyllä***

Asiakassovellus tekoälyllä on täysin autonominen sovellus. Sovellukselle annetaan palvelimen ip-osoite mihin halutaan liittyä ja tämän jälkeen sovellus pelaa tekoälyn antamien rajojen mukaisesti. Tekoälyssä tullaan hyödyntämään palvelimen käyttämän ulkopuolisen kirjaston tarjoamia käsien vertailuoperaatioita. Tekoälyn on tarkoitus toimia jokseenkin järkevästi ja pyrkiä voittamaan peliin.

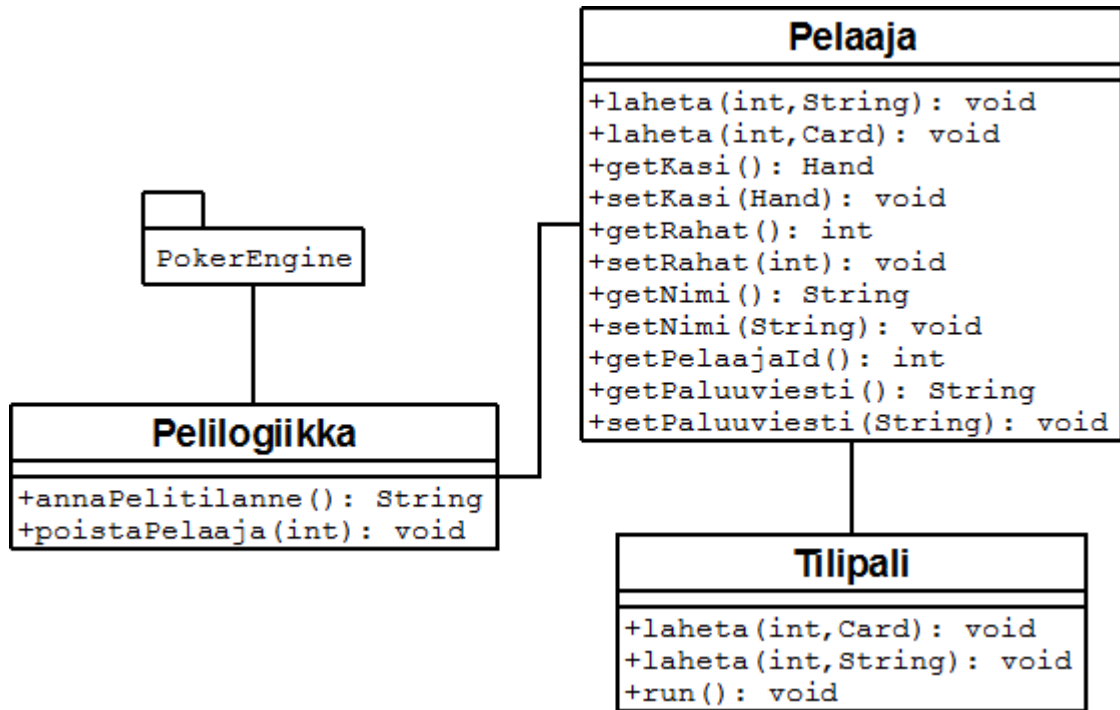
## **Rajapintakuvaukset**

Kuvaus toteutettavien ja valmiiden luokkien rajapinnoista.

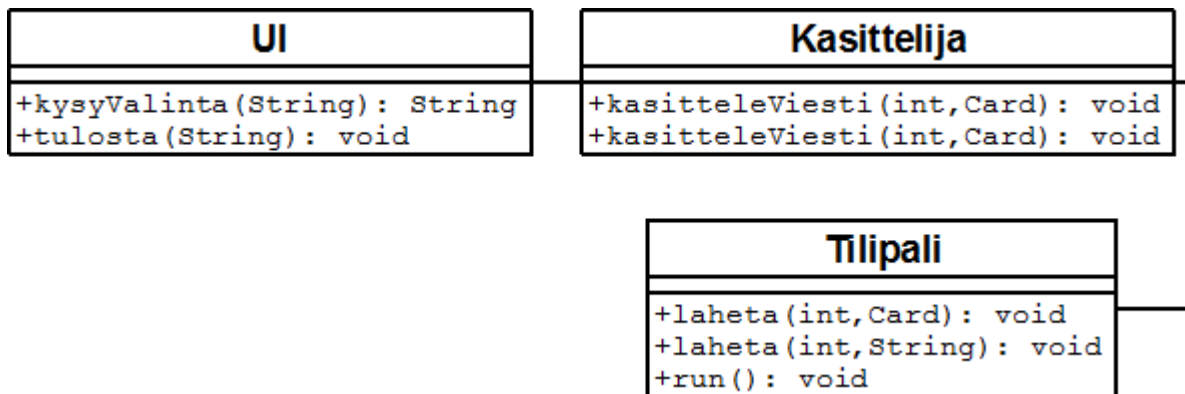
### **Luokkakaaviot**

Luokkakaaviot toteutettavista luokista.

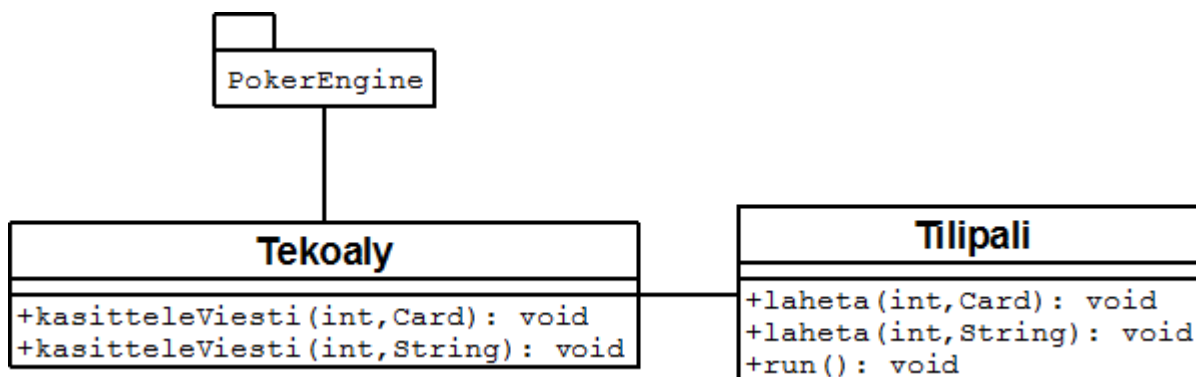
## Palvelin



## Asiakas ilman tekoälyä



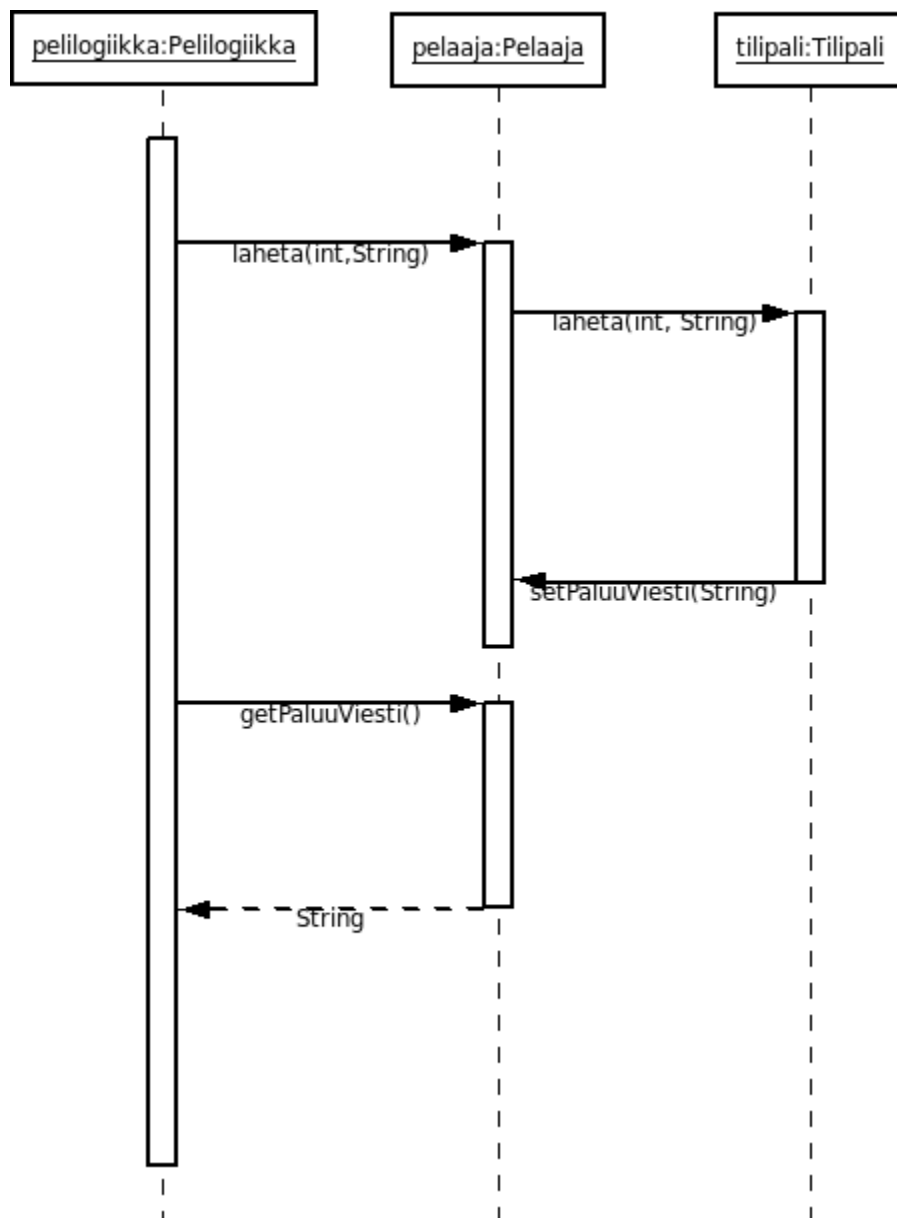
## Asiakas tekoälyllä



## Sekvenssikaaviot

### Palvelin

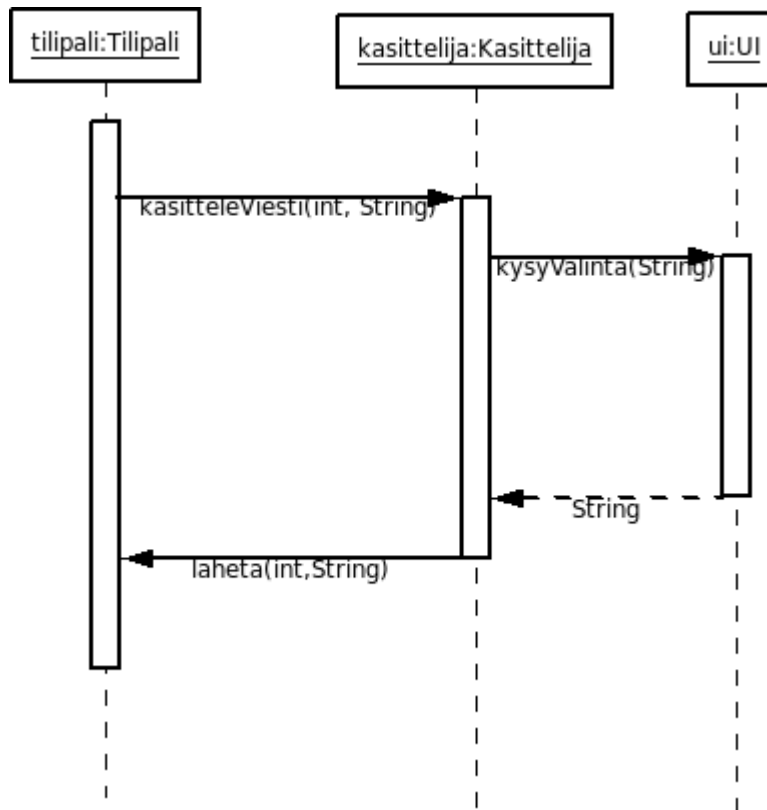
Pelilogiikka kysyy asiakkaalta hänen valintansa. Tämän jälkeen Pelilogiikka hakee asiakkaan lähettämän paluuviestin.





## Asiakassovellus

Palvelin on kysynyt asiakkaalta hänen valintaansa. Kasittelija kysyy käyttöliittymältä asiakkaan syötteen ja lähettää sen takaisin palvelimelle.



## Luokkametodien kuvaukset

Kuvaukset luokkien julkisista metodeista.

### Palvelin

Palvelimen luokat

### Pelilogiikka

Pelilogiikka-luokka huolehtii pelin etenemisestä. Luokka kysyy pelaajilta heidän tekemiä toimintoja, jakaa ja lähettää kortit, päättää ja aloittaa jaot.

Pelilogiikka()

Konstruktori luo uuden pelilogiikan ja alustaa pelaajat arraylistin.

void poistaPelaaja(int)

Pelaajan poistuessa kutsutaan metodia pelaajan id:llä.

String annaPelitilanne()

Palauttaa kortit jotka on pöydällä, potin koon, pelaajien rahat, diilerin paikan.

## Pelaaja

Pelaaja-luokkaan on tallennettuna pelaajasta kaikki tiedot. Pelaaja-luokalla on viite kyseisen pelaajan Tilipali-olioon, mikä hoitaa sockettien käytön. Muuttujaan paluuViesti tallennetaan pelaajan viimeiseksi lähetetty viesti. Pelilogiikka lukee viestin (mikäli se odottaa viestiä) ja sen jälkeen asettaa viestin takaisin nulliin.

Pelaaja()

Luo Tilipali-luokan ilmentymän ja käynnistää sen (Thread). Pelaajaid alkaa kasvamaan yhdestä eteenpäin.

void laheta(int, String)

Välittää viestin Tilipalille lähetystä varten. Kokonaisluku kertoo tapahtumatunnuksen ja String tapahtuman datan.

void laheta(int, Card)

Välittää viestin Tilipalille lähetystä varten. Tapahtumatunnus ja lähetettävä kortti.

Hand getKasi()

Palauttaa pelaajan käden

void setKasi(Hand)

Asettaa pelaajan käden.

double getRahat()

Palauttaa pelaajan rahat

void setRahat(double)

Asettaa rahoille uuden arvon

String getNimi()

Palauttaa nimen

void setNimi(String)

Asettaa nimen

int getPelaajaId()

Palauttaa pelaajaid:n

String getPaluuviesti()

Palauttaa paluuviestin

void setPaluuviesti(String)

Asettaa paluuviestin

## Tilipali

Tilipali-luokka muuntaa lähtevän tiedon tietoliikenneprotokollan mukaiseksi ja vastaavasti muuntaa vastaanotettavan tiedon PokerEngine esittämään muotoon. Vastaanotettaessa paluuviesti Tilipali-luokka kutsuu Pelaajan *void setPaluuviesti(String)*.

Tilipali(TCPSocket, TCPSocket, Pelaaja)

Ulos- ja sisäänmenevät socketit. Pelaaja-viite vastaanotettavien viestien välitystä varten.

void laheta(int, Card)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void laheta(int, String)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void run()

Säie odottaa uusia viestejä. Kutsuu pelaajan *setPaluuViesti(String)* ja asettaa saadun paluuviestin. Mikäli asiakas on pyytänyt pelitilannetta, kutsutaan Pelilogiikan *annaPelitilanne()*-metodia ja lähetetään asiakkaalle pelitilanne takaisin suoraan.

## Asiakassovellus ilman tekoälyä

Asiakassovellus ilman tekoälyä luokat

## Kasittelija

Kasittelija alustaa UI-luokan ja Tilipali-luokan. Kasittelija kasittelee saadun viestin ja sen perusteella tekee tarvittavat tulostukset/kysymykset UI-luokalle.

Kasittelija()

Luo uuden kasittelijan ja alustaa UI-olion.

void kasitteleViesti(int, Card)

Käsittelee palvelimelta saadun viestin.

void kasitteleViesti(int, String)

Käsittelee palvelimelta saadun viestin.

## UI

Luokka huolehtii tulostuksista ja käyttäjän valinnan kysymisestä.

UI()

Luo uuden UI-olion. Alustaa Scannerin.

String kysyValinta(String)

Kysyy käyttäjän valinnan esitettyyn kysymykseen. Vastauksen oikeellisuudesta huolehtii Kasittelija/Palvelin.

void tulosta(String)

Tulostaa annetun parametrin käyttäjälle.

## Tilipali

Socketit palvelimelle ja palvelimelta. Viestin saatuaan kutsutaan Kasittelijan kasitteViesti-metodia.

Tilipali(TCPSocket, TCPSocket, Kasittelija)

Ulos- ja sisäänmenevät socketit. Pelaaja-viite vastaanotettavien viestien välitystä varten.

void laheta(int, Card)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void laheta(int, String)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void run()

Säie odottaa uusia viestejä. Kutsuu Kasittelijaa, kun uusi viesti saatu.

## Asiakassovellus tekoälyllä

Kuvaukset luokista

### Tekoaly

Tekoaly saa viestejä saaduista korteista ja tallentaa ne sisäisiin tietorakenteisiinsa ja tekee vertailuja niiden pohjalta mitä kannattaa tehdä (check/fold/raise/call).

Tekoaly()

Luo Tilipali-olion ja alustaa verkkoyhteydet. Alustaa luokan alkutilaan ja jää odottamaan pelin alkua.

void kasitteleViesti(int, Card)

Käsittelee palvelimelta saadun viestin.

void kasitteleViesti(int, String)

Käsittelee palvelimelta saadun viestin.

### Tilipali

Välittää tekoälylle viestit pelin tapahtumista, joiden perusteella tekoaly vastaa palvelimelle parhaaksi katsomalla tavalla.

Tilipali(TCPSocket, TCPSocket, Tekoaly)

Ulos- ja sisäänmenevät socketit. Tekoaly-viite vastaanotettavien viestien välitystä varten.

void laheta(int, Card)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void laheta(int, String)

Muuttaa tiedon tietoliikenneprotokollan mukaiseksi ja lähettää.

void run()

Säie odottaa uusia viestejä. Kutsuu Tekoalyä, kun uusi viesti saatu.

## **PokerEngine (ulkopuolinen komponentti)**

### **Deck**

void shuffle()

Sekoittaa korttipakan

Card deal()

Otaa pakasta ensimmäisen kortin

### **Card**

int compareTo(Card)

Vertaa kumpi kortti on suurempi

Rank getRank()

Palauttaa kortin numeron

Suit getSuit()

Palauttaa kortin

### **Hand**

void addCard(Card)

Lisää käteen kortin

List<Card> getCards()

Palauttaa käden kortit

HandRank getHandRank(Board)

Parametrina Board-olio, missä on pöydän kortit. Palauttaa käden ja pöytäkorttien yhdistelmän arvon.

### **Board**

List<Card> getCards()

Palauttaa listan pöydällä olevista korteista

void addCard(Card)

Lisää listaan kortin

## HandRank

int compareTo(HandRank)

Vertailee käsien arvoa

Rank getRank()

Palauttaa käden arvon

## Tietoliikenneprotokolla

### Viestin rakenne

Tietoa lähetään TCP-sockettien kautta asiakkaiden ja palvelimen välillä. Lähetettävä viesti lähetetään merkkijonona. Merkkijono koostuu kahdesta osasta. Osat eritellään puolipisteellä. Ensimmäinen osa kertoo tapahtuman tunnuksen kokonaislukuna. Toinen osa kertoo tapahtumaan liittyvän tiedon. Esimerkiksi *0;Pelaaja X liittyi peliin*. Viestin alkuosa kertoo tapahtuman, jonka tunnus on ko. viestissä 0. Edellinen viesti voisi tarkoittaa, vaikka tekstiä mikä lähetetään asiakkaalle ja asiakassovellus tulkitsee 0 tunnukseksi tulleen viestin tulostettavaksi käyttäjälle.

### Pelikortin esitys verkkoliikenteessä

Pelikortti muunnetaan numeroksi ennen lähettämistä ja vastaavasti vastaanotettaessa se muunnetaan takaisin Card-olioksi. Kortin muunnos tapahtuu seuraavasti. Kutakin korttia vastaan yksi numero väliltä 0-51. Kortti voidaan laskea numeroksi seuraavalla tavalla:

Jokaiselle maalle on annettu kerroinluku:

**0 Risti**

**1 Ruutu**

**2 Ruutu**

**3 Pata**

Kortin arvo (1, 2, 4, 13 etc) muunnetaan seuraavasti: Kortin arvosta vähennetään aina 1. Eli kortista kaksi tulisi yksi ja kortista viisi tulisi neljä jne.

Lopuksi edellä laskettuja arvoja hyödyntäen lasketaan kortti numeroksi kaavalla:

***maanKerroin\*13+kortinArvo***

Numero muunnetaan Card-olioksi seuraavalla tavalla:

Maa saadaan selville jakamalla arvo kolmellatoista ja katsomalla saatua kokonaislukua.

Kortin numero saadaan selville jakamalla numero neljälle ja lisäämällä tulokseen yksi.

## **Esimerkki 1.**

Muunnettava kortti pata 10.

Sijoitetaan arvot kaavaan ja saadaan  $3*13+9=48$

Ja takaisin Card-olioksi:

Maa=48/13

Arvo=(48%13)+1

## **Esimerkki 2.**

Muunnettava kortti risti-ässä. Näemme ylhäältä, että ristin kerroin on nolla ja ässän arvo on 1-1=0.

Sijoitetaan kaavaan:  $0*13+0=0$

Takaisin kortiksi:

Maa=0/13

Arvo=(0%13)+1

## **Viesti tunnukset**

Protokollan käyttämät viestitunnukset. Viestit ovat muotoa VIESTITUNNUS;VIESTINDATA.  
Kokonaisluku viestin alussa kertoo viestin tunnuksen, jota seuraa puolipiste ja viestin dataosuus

## **Palvelimen lähettämät viestit**

0 - Viesti käyttäjälle, ei tarvitse jäsentää. Esim: 0;Pelaaja x liittyi peliin

1 - Viesti asiakkaalle, että hänen vuoronsa. Ei dataa. Esim: 1;

2 - Käsikortin lähetys pelaajalle. Esim: 2;43



- 3 - Flopin kortin lähetys. Esim: 3;25
- 4 - Turnin kortin lähetys. Esim: 4;13
- 5 - Riverin kortin lähetys. Esim: 5;46

## Asiakkaan lähettämät viestit

- 0 - Peliin kirjautuminen. Asiakas lähettää datana nimimerkkinsä. Esim. 0;Jarkko
- 1 - Asiakas vastaa palvelimelle mitä hän haluaa omalla vuorolla tehdä. Viestin data voi olla *fold*, *check*, *bet*, *call*. Esim. 1;fold
- 2 – Pyytää palvelinta lähettämään sen hetkisen pelin tilan. Ei dataa. Esim. 2;

## Lähteet

- [1] <http://www.grandcasinohelsinki.fi/index.php/pelit/pelisanastoa/pokeri>
- [2] <http://www.pokerlistings.fi/fixed-limit-texas-hold-em-saannot>
- [3] <http://www.javarants.com/C1464297901/E20070317111158/index.html>