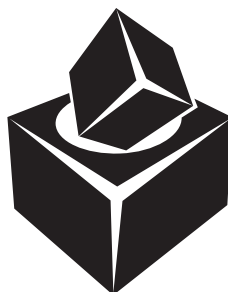


ALEA-Sovellusprojekti

Sovellusraportti

**Arto Hankaniemi
Timo Holopainen
Joni Kivinen
Riku Kuismanen**



Versio: 0.7
Julkinen
24. toukokuuta 2006

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Hyväksyjä	Päivämäärä	Allekirjoitus	Nimenselvennys
Projektipäällikkö	__.__.2006		
Tilaja	__.__.2006		
Ohjaaja	__.__.2006		

Tietoa dokumentista

Tekijät:

- | | | |
|------------------------|--------------------|-------------|
| • Arto Hankaniemi (AH) | arkahonk@cc.jyu.fi | 050-5186436 |
| • Timo Holopainen (TH) | tijuholo@cc.jyu.fi | 041-4368295 |
| • Joni Kivinen (JK) | jomakivi@cc.jyu.fi | 050-5968576 |
| • Riku Kuismanen (RK) | rtkuisma@cc.jyu.fi | 0400-452512 |

Dokumentin nimi: ALEA-Projekti, Sovellusraportti

Sivumäärä: 37

Tiedosto: ALEA_sovellusraportti.tex

Tiivistelmä: Sovellusraportissa kuvataan ALEA-projektin kehittämän WebTalariousovelluksen toteutusta.

Avainsanat: ALEA, sovellusraportti.

Versiohistoria

Versio	Päivämäärä	Muutokset	Tekijät
0.1	9.5.2006	Alustava versio.	AH
0.2	12.5.2006	Katselmoinnissa havaitut virheet korjattu, 4 ensimmäistä lukua.	AH
0.3	12.5.2006	Tarkennuksia sivuille 12,13,14 ja 15.	AH
0.4	17.5.2006	Luvut 5-8 lisätty.	AH/TH/JK
0.5	22.5.2006	Luvut 9-10 lisätty ja lukuihin 5-8 tarkennuksia.	AH/TH
0.6	23.5.2006	Ensimmäisessä katselmoinnissa havaitut virheet korjattu.	AH/TH/JK
0.7	24.5.2006	Toisessa katselmoinnissa havaitut virheet korjattu.	AH/TH/JK

Tietoa projektista

ALEA-projektin tavoitteena on toteuttaa WebTalarius-sovellus Agora Game Labin Talarius-lautapelinsuunnitteluohjelman nykyisen version pohjalta. WebTalarius-sovelluksen toteutuksessa pääasiallinen tavoite oli mahdollistaa pelaaminen ja tiedostojen jakaminen tietoverkossa.

Tekijät:

- | | | |
|------------------------|--------------------|-------------|
| • Arto Hankaniemi (AH) | arkahonk@cc.jyu.fi | 050-5186436 |
| • Timo Holopainen (TH) | tijuholo@cc.jyu.fi | 041-4368295 |
| • Joni Kivinen (JK) | jomakivi@cc.jyu.fi | 050-5968576 |
| • Riku Kuismanen (RK) | rtkuisma@cc.jyu.fi | 0400-452512 |

Tilaaaja:

- | | | |
|----------------------|-------------------------------|-------------|
| • Marja Kankaanranta | marja.kankaanranta@ktl.jyu.fi | 014-2603280 |
| • Tuula Nousiainen | tunousia@cc.jyu.fi | 014-2604652 |
| • Antti Kirjavainen | ajkirjav@iki.fi | 014-2604671 |
| • Heikki Nevala | hemaneva@cc.jyu.fi | 040-8495317 |

Ohjaajat:

- | | | |
|---------------------|--------------------|-------------|
| • Ville Isomöttönen | vilisom@cc.jyu.fi | 014-2604976 |
| • Jukka Miettinen | jupemiet@cc.jyu.fi | 045-6706322 |

Yhteystiedot:

- | | |
|----------------------|---|
| • Sähköpostilistat: | alea06@korppi.jyu.fi,
alea_opetus@korppi.jyu.fi |
| • Sähköpostiarkisto: | https://korppi.jyu.fi/list-archive/alea06/ind.html |
| • Opetusarkisto: | https://korppi.jyu.fi/list-archive/alea_opetus/ind.html |
| • Työhuone: | AgC 225.3 / 014-2604971 |

Sisältö

1	Johdanto	1
2	Termit	2
3	Ympäristö ja vaatimusten toteutuminen	3
3.1	Työkalut	3
3.2	Vaatimusten toteutuminen	3
4	Käyttöliittymä	5
5	Arkkitehtuuri	17
5.1	Yleinen arkkitehtuuri	17
5.2	Sovelluksen pelitiedostojen arkkitehtuuri	18
5.3	Verkkoarkkitehtuuri	19
5.4	XML-paketit	22
5.5	Tiedostopalvelimen XML-paketit	24
6	Luokat	26
6.1	Käyttöliittymien väliset suhteet	26
6.2	Sovelluksen luokat	27
6.2.1	Asiakas	27
6.2.2	Metapalvelin	29
6.2.3	Paketit	30
7	Lähdekoodin ulkoasu	32
8	Testaus	33
8.1	Toteutuksen aikainen testaus	33
8.2	Ensimmäinen testikierros	33
8.3	Toinen testikierros	33
9	Jatkokehitysideat	34
9.1	Tekemättä jääneet vaatimukset	34
9.2	Vaatimusten jatkokehitysideat	34
9.3	Toteutusratkaisujen analyysi	35
10	Lähteet	37

1 Johdanto

ALEA-projektin toteutti WebTalarius-sovelluksen Agora Game Labin Talarius-lautapelinsuunnitteluohjelman nykyisen version pohjalta. WebTalarius-sovelluksen toteutuksessa pääasiallinen tavoite oli mahdollistaa pelaaminen ja tiedostojen jakaminen tietoverkossa. Lisäksi ohjelmaan toteutettiin tiettyjä sovelluksen toimintaa monipuolistavia ja opetuksellisuutta parantavia ominaisuuksia.

Talarius on oppimisohjelmisto, jonka avulla ala-asteikäiset lapset voivat suunnitella ja toteuttaa tietokoneella pelattavia, opetuksellisiin aiheisiin liittyviä lautapelejä. Sovelluksen nykyisellä versiolla voidaan tehdä pelilauta ja kysymyksiä sekä pelata tehtyjä pelejä. Sovellusta voidaan käyttää oppimisen tukena useissa eri oppiaineissa tai esimerkiksi opittujen asioiden kertaamisen apuna. Sovellusta on testattu mm. opetuksellisuuden ja käytettävyyden näkökulmista. Näistä testauksista ja käyttökokeiluista saatiin lähtökohtia WebTalarius-sovellukselle asetettaville vaatimuksille.

Agora Game Labissa toteutukseen osallistui tutkijaryhmä, jonka jäsenet edustavat mm. kasvatustieteiden, luonnontieteiden, tietotekniikan ja tietojenkäsittelytieteiden osaamista. Tutkijaryhmä toimi projektiryhmän tukena. Sovelluksen kehittäminen ja suunnittelu toteutettiin osallistuvan suunnittelun periaatteiden mukaisesti yhteistyössä Agora Game Labin tutkijaryhmän sekä sovelluksen tulevien käyttäjien (mm. koululaiset ja opettajat) kanssa.

Tämä dokumentti on ALEA-projektin sovellusraportti. Luvussa 2 esitellään dokumentin keskeisimmät termit. Luvussa 3 esitellään projektissa käytetyt työkalut sekä mitkä vaatimukset toteutuivat. Luvussa 4 esitellään WebTalarius-sovelluksen käyttöliittymää ja ulkoista ilmettä. Luvussa 5 kuvataan sovelluksen arkkitehtuuri. Luvussa 6 on esitetty luokkien rakenteet kuvineen. Luku 7 keskittyy koodin ulkoasuun ja luku 8 käsittelee sovelluksen toteutunutta testausta. Luvusta 9 löytyvät jatkokehitysideat sekä toteutusratkaisujen analyysi.

2 Termit

Luvussa esitellään dokumentin keskeiset termit.

Delphi 7	on Borlandin kehittämä ohjelmointiympäristö, joka on tarkoitettu graafiseen ohjelmointiin.
Metapalvelin	on erillinen palvelin, joka pitää yllä tietoa tarjolla olevista pelisessiosta ja lisäksi tallentaa pelisessiosta syntyvät lokitiedostot ja pelitallennukset. Metapalvelin pitää yllä myös pelitilejä.
Tiedostopalvelin	on erillinen palvelin, joka tallentaa pelitiedostot.
XML	(Extensible Markup Language) on W3C:n suositus tekstimuotoisesta rakenteellisesta metakielestä, jolla voidaan kuvata muita kieliä. Se on yksinkertaistettu versio SGML-kielestä.

3 Ympäristö ja vaatimusten toteutuminen

Tässä luvussa esitellään projektissa käytetyt työkalut ja vaatimusten toteutuminen.

3.1 Työkalut

Sovellus toteutettiin Borlandin Delphi 7 -ohjelmointiympäristöllä. Sovelluksen pelilogit toteutettiin käyttäen XML-merkkäuskieltä.

3.2 Vaatimusten toteutuminen

WebTalarius-sovellus tehtiin ALEA-projektin toimesta, jossa Talarius-sovellusta jatkokehitettiin. Talarius-sovelluksella voidaan tehdä pelilauta ja kysymyksiä sekä pelata tehtyjä pelejä. ALEA-projekti toteutti WebTalarius-sovelluksen, joka mahdollistaa tietoverkon välityksellä pelaamisen. Ohjelmaan toteutettiin myös joitain sovelluksen toimintaa monipuolistavia ja opetuksellisuutta parantavia ominaisuuksia.

Vaatimusmäärittelystä ei saatu projektin kuluessa toteutettua kaikkia vaatimuksia pääosin siksi, että viestintään valitsemamme komponentti osoittautui projektin kannalta käyttökelvottomaksi. Alkuperäiseen lähdekoodiin tutustuminen ja sen muokkaaminen vaatimusten mukaiseksi vei myös aikaa. Projektikokouksessa 11 priorisoitiin tärkeimmäksi sovelluksen saattaminen virheettömään kuntoon sekä sovelluksen testaaminen. Pakollisista vaatimuksista ei keretty tehdä *tiedostojen siirtoa* (V9.3), eli tiedostopalvelinta, joka varastoi ja jakaa pelitiedostoja. Tiedostopalvelimesta toteutettiin kuitenkin demoversio, jonka jatkokehittämistä tarkastellaan luvussa 9.

Toteutetut pakolliset vaatimukset:

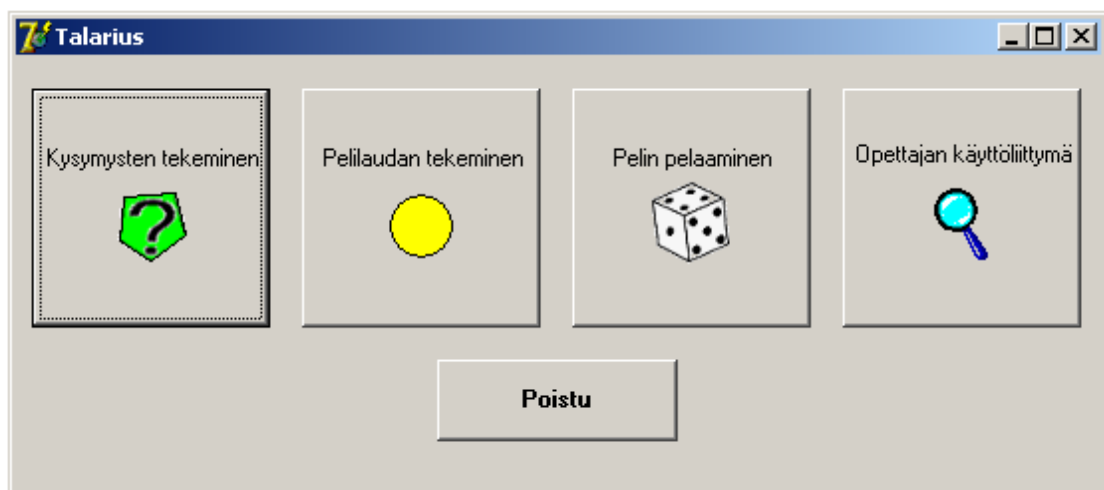
- V1.4 Pelisession maksimipelaajamäärän määrittäminen.
- V1.5 Pelisession käynnistäminen.
- V3.1 Pelaajatilien luominen.
- V3.5 Peliin liittyminen.
- V3.7 Pelisessioiden selaaminen.

- V5.1 Opettajatilin luominen.
 - V5.2 Pelaajatilien luomisen suojaaminen salasanalla.
 - V5.4 Tiedostojen siirron rajoittaminen tiedostopalvelimelle.
 - V6.1 Asetustiedosto.
 - V6.2 Suomenkielinen käyttöliittymä.
 - V6.3 Asennusohje.
 - V6.4 Käyttöohje.
 - V7.1 Tapahtumalokien tallentaminen.
 - V7.4 Tiedostojen jakaminen.
 - V7.6 Metapalvelimella näytettävät pelisessioiden tiedot.
 - V7.8 Yhteyden menettämisen käsittely.
 - V7.9 Pelitilanteen päivittäminen.
 - V7.10 Pelivuoron siirtäminen.
 - V9.1 XML-tiedostomuoto.
 - V9.4 Pelitilien ja -sessioiden hallinta.
- Toteutetut tärkeät vaatimukset:**
- V1.3 Pelisession suojaaminen salasanalla.
 - V4.1 Pelilokien tarkasteleminen.
 - V8.1 Kysymyssarjojen liittäminen.
- Toteutetut mahdolliset vaatimukset:**
- V7.3 Metapalvelimelle pääsyn rajoittaminen.
 - V7.7 Chatti-ikkuna.

4 Käyttöliittymä

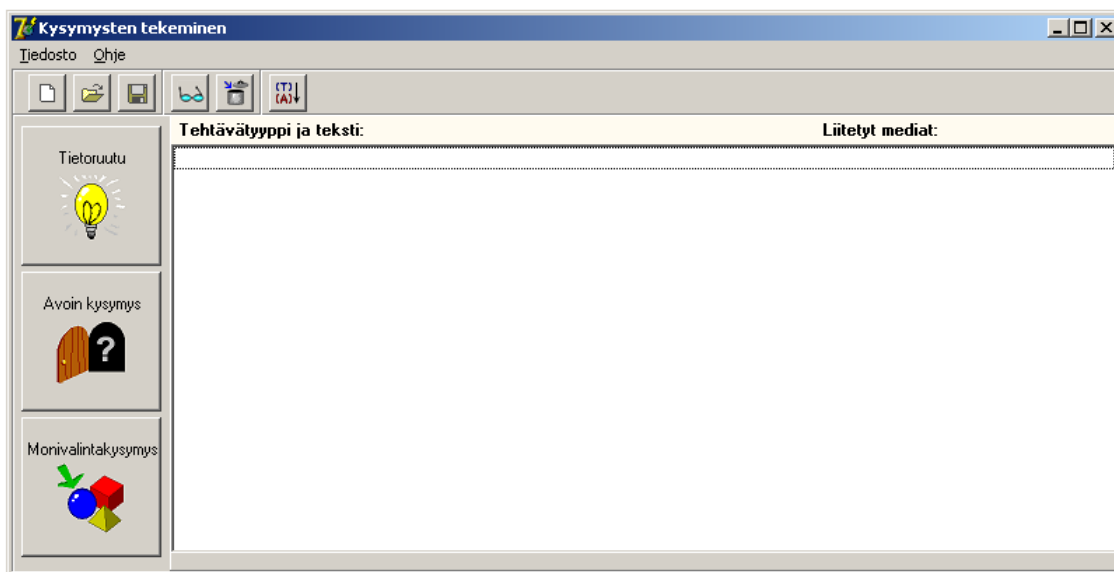
Tässä luvussa esitellään WebTalarius-sovelluksen käyttöliittymien rakennetta.

Kuvassa 4.1 on WebTalarius-sovelluksen *Talarius*-käyttöliittymä, josta sovelluksen käyttö aloitetaan. *Talarius*-käyttöliittymästä voidaan valita kysymysten tekeminen, pelilaudan tekeminen, pelin pelaaminen tai opettajan käyttöliittymä. ALEA-projektissa ei muutettu *Talarius*-käyttöliittymää.



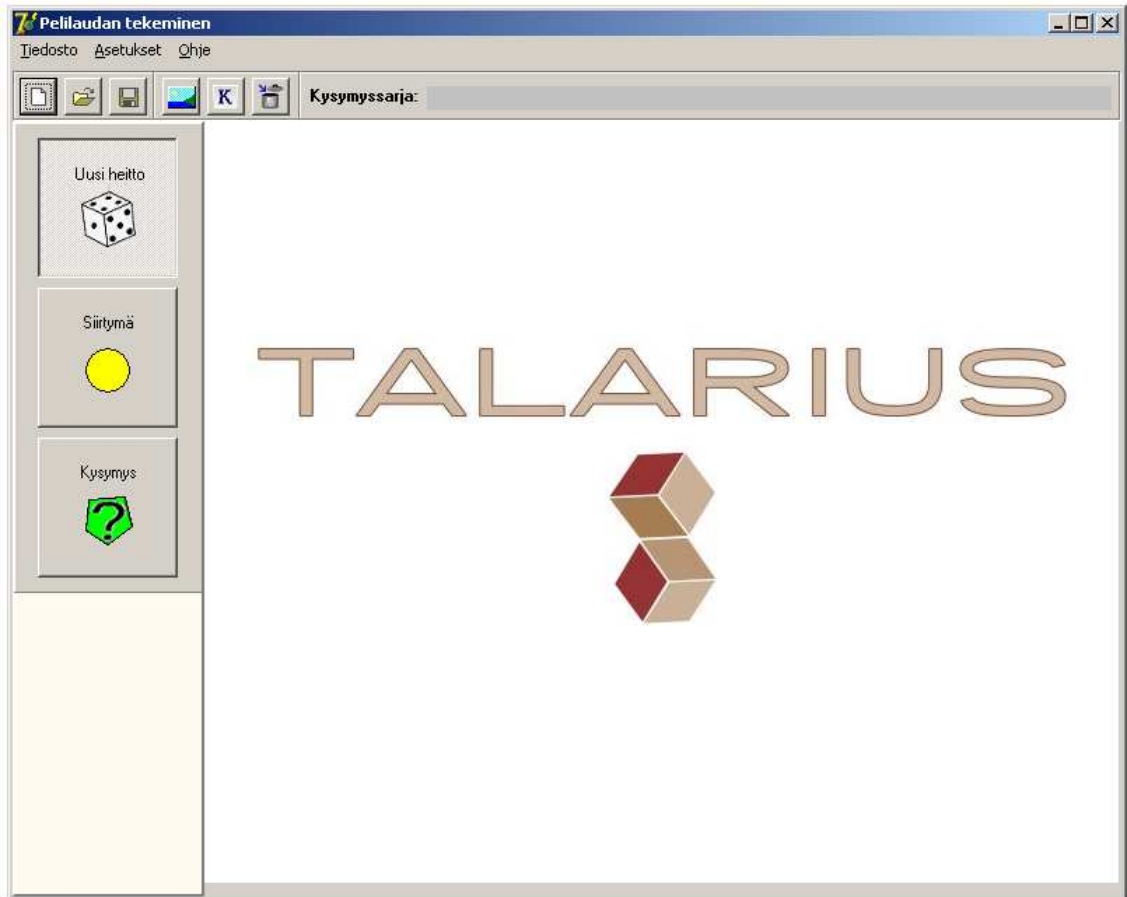
Kuva 4.1: *Talarius*-käyttöliittymä.

Kuvassa 4.2 on *Kysymysten tekeminen* -käyttöliittymä, johon päästään *Talarius*-käyttöliittymän *Kysymysten tekeminen* -painikkeen kautta. Käyttäjä voi luoda tässä käyttöliittymässä erilaisia kysymyksiä peliinsä. Kysymyksiä yhdistelemällä, ja ne tallentamalla saadaan kysymyssarjoja. Erilaisia kysymystyyppejä on kolme: Tietoruutu, avoin kysymys ja monivalintakysymys. ALEA-projektissa ei tämän käyttöliittymän ulkoiseen ilmeeseen puututtu. Ainoa muutos, mitä tehtiin oli sisäinen muutos, eli kysymyssarjojen tiedostojen päätteiden vaihtaminen .ini-päätteisistä tiedoistoista .kys-päätteisiin tiedoistoihin. Tällä haluttiin erottaa kysymyssarjat ja pelilaudat toisistaan.



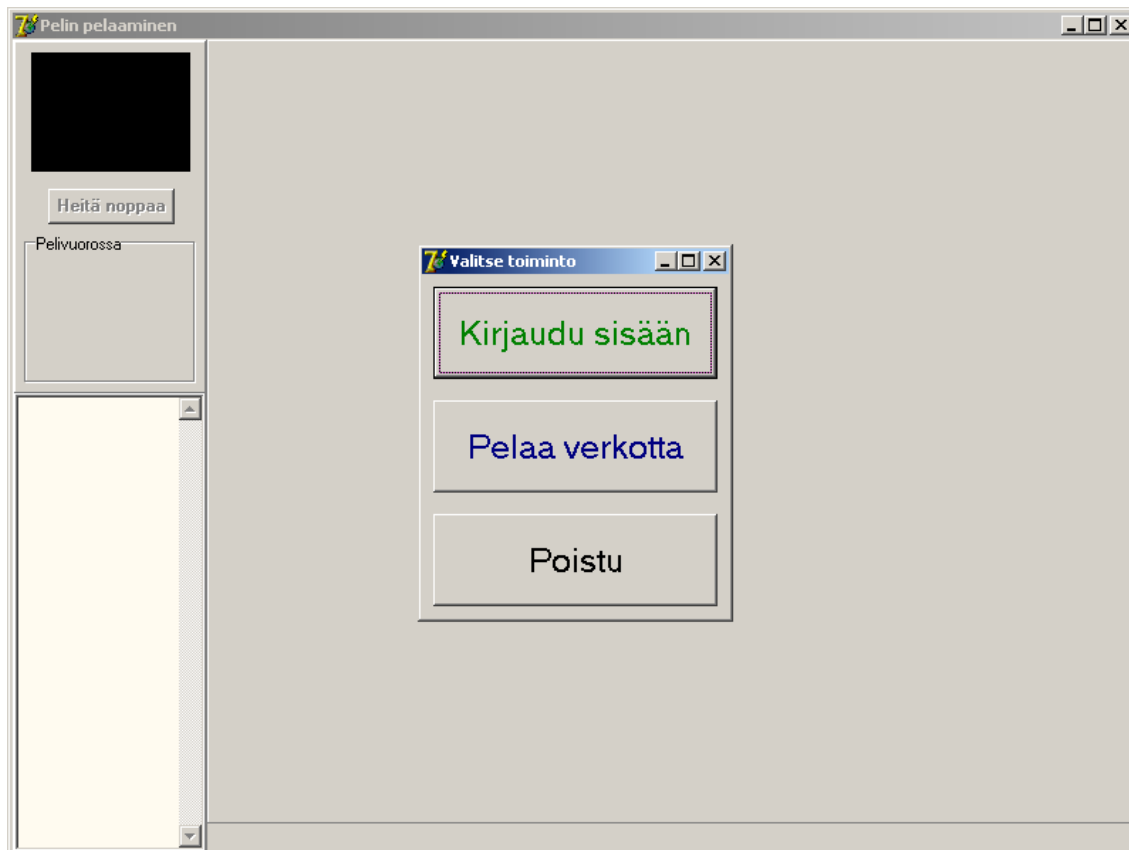
Kuva 4.2: Käyttöliittymä kysymysten tekoon.

Kuvassa 4.3 on *Pelilaudan tekeminen* -käyttöliittymä, johon päästään *Talarius*-käyttöliittymän *Pelilaudan tekeminen* -painikkeen kautta. Käyttäjä voi luoda tässä käyttöliittymässä haluamansa pelilaudan ja liittää siihen haluamansa kysymyssarjan/kysymyssarjat. ALEA-projektissa ei muokattu tämän käyttöliittymän ulkoista ilmettä. Sisäisesti muutettiin pelilautojen tiedostojen päätteet .ini-päätteisistä .pla-päätteisiin tiedoistoihin. Tällä haluttiin erottaa kysymyssarjat ja pelilaudat toisistaan.



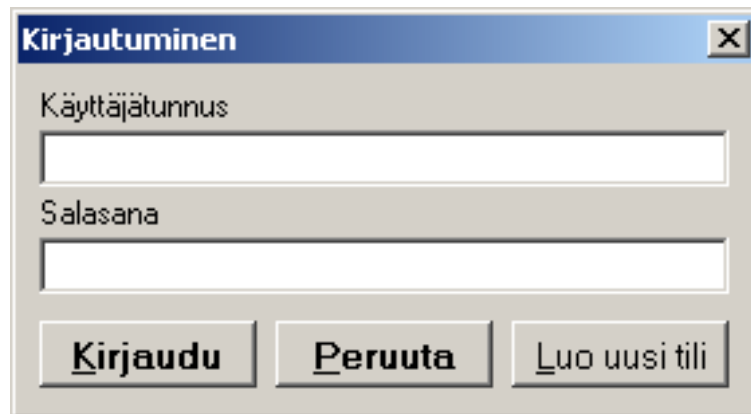
Kuva 4.3: Käyttöliittymä pelilaudan tekoa varten.

Kuvassa 4.4 on *Pelin pelaaminen* -käyttöliittymä, jossa peliä pelataan. Tähän käyttöliittymään päästään *Talarius*-käyttöliittymän *Pelin pelaaminen* -painikkeen kautta. *Valitse toiminto* -ikkunasta voidaan valita halutaanko peliä pelata verkossa vai ei, verkkopelaaminen edellyttää sisään kirjautumista. ALEA-projektissa tehtiin *Valitse toiminto* -ikkuna, ja *Pelin pelaaminen* -käyttöliittymään lisättiin ikkuna, jossa näkyy pelin kulkua kuvaavat viestit (vasen alareuna), muuten *Pelin pelaaminen* -käyttöliittymää ei muutettu.



Kuva 4.4: Käyttöliittymä pelin pelaamista varten.

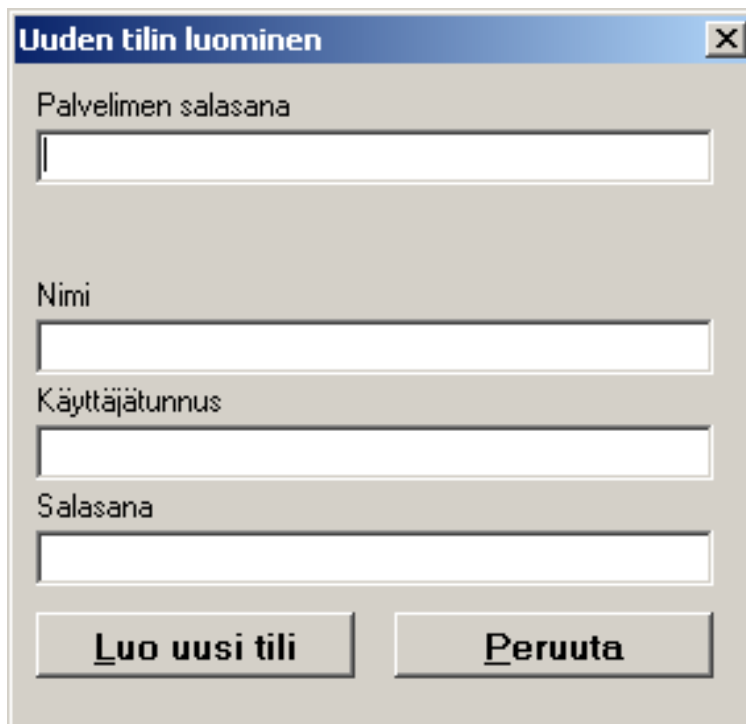
Kuvassa 4.5 on *Kirjautuminen*-käyttöliittymä. Tähän käyttöliittymään päästään *Pelin pelaaminen* - käyttöliittymän *Kirjautu sisään* -painikkeen kautta. Tästä käyttöliittymästä voivat oppilaat kirjautua metapalvelimen hallintakäyttöliittymään. Jos pelaajalta puuttuu tili, niin uuden tilin luominen onnistuu tämän käyttöliittymän *Luo uusi tili* -painikkeen kautta. *Kirjautuminen*-käyttöliittymä on ALEA-projektin tuotos.



The image shows a standard Windows-style dialog box titled "Kirjautuminen". It contains two text input fields, one for "Käyttäjätunnus" and one for "Salasana". At the bottom, there are three buttons: "Kirjautu", "Peruuta", and "Luo uusi tili".

Kuva 4.5: *Kirjautuminen*-käyttöliittymä.

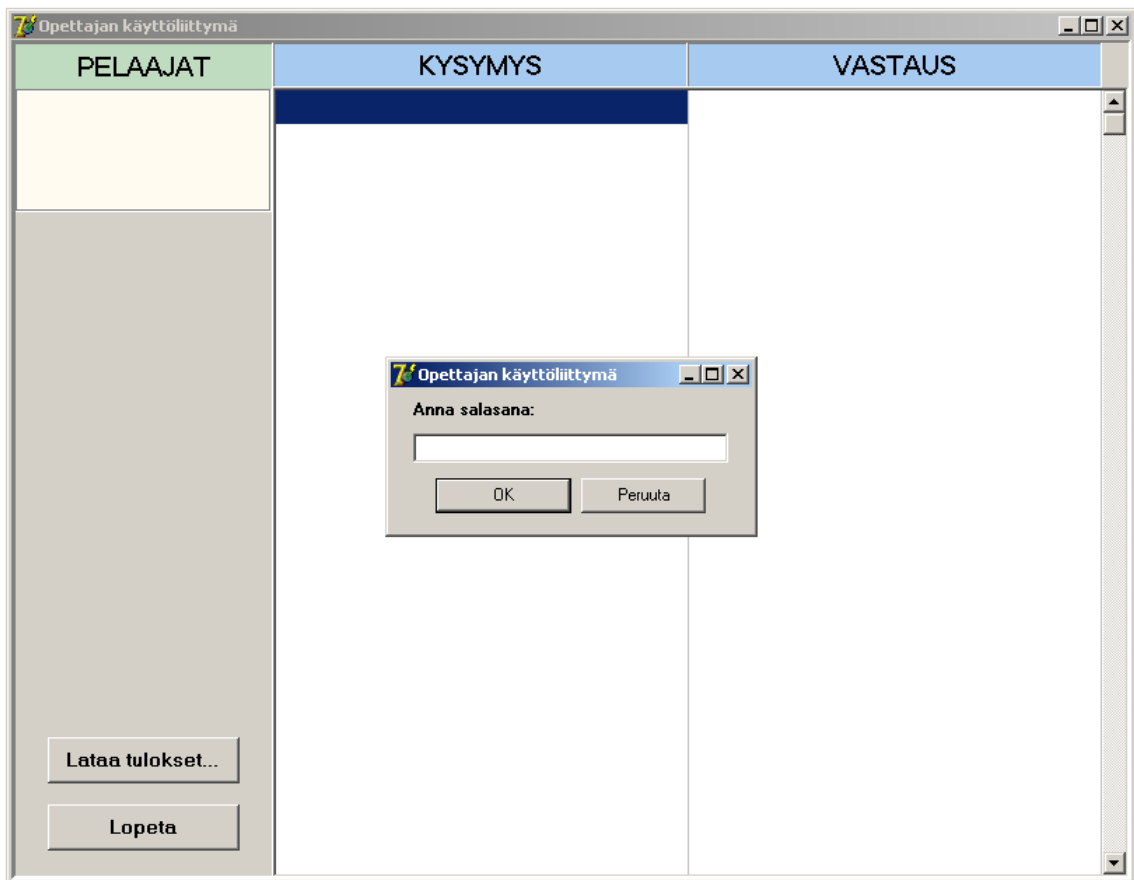
Kuvassa 4.6 on *Uuden tilin luominen* -käyttöliittymä. Tähän käyttöliittymään tullaan *Kirjautuminen*- käyttöliittymän *Luo uusi tili* -painikkeesta. *Uuden tilin luominen* -käyttöliittymästä löytyvät seuraavat kentät: *Palvelimen salasana*, *Käyttäjätunnus*, *nimi* ja *salasana*. Puuttuvat tiedot täytetään ja pelitili luodaan *Luo uusi tili* -painikkeen kautta. Oppilaalle tulee kuittaus, jos pelitilin luonti onnistuu. *Uuden tilin luominen* -käyttöliittymä on ALEA-projektin tuotos.



The image shows a graphical user interface window titled "Uuden tilin luominen". The window contains four text input fields for user registration: "Palvelimen salasana", "Nimi", "Käyttäjätunnus", and "Salasana". At the bottom of the window, there are two buttons: "Luo uusi tili" and "Peruuta".

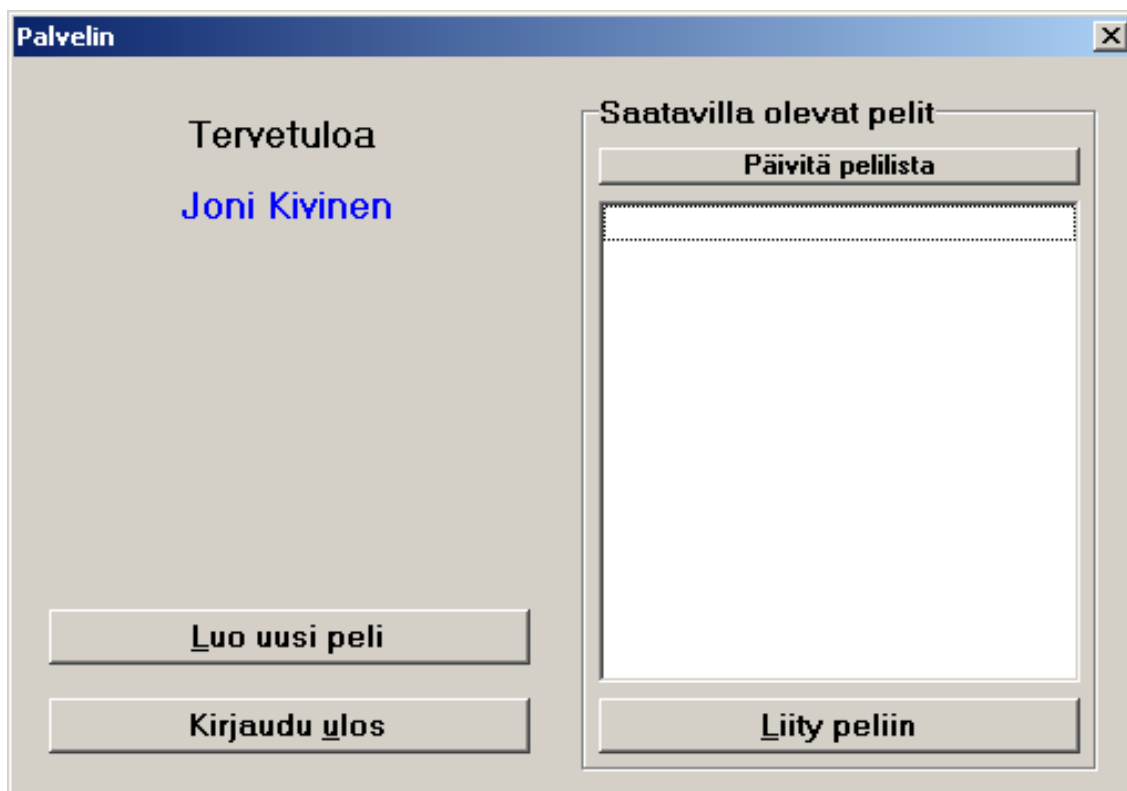
Kuva 4.6: *Uuden tilin luominen* -käyttöliittymä.

Kuvassa 4.7 on *Opettajan käyttöliittymä*, jossa opettaja voi tarkastella pelilokeja oppilas kohtaisesti. Tähän käyttöliittymään päästään *Talarius-käyttöliittymän Opettajan käyttöliittymä* -painikkeen kautta. Opettaja antaa oikean salasanan, jonka jälkeen voidaan *Lataa tulokset* -painikkeesta ladata kaikkien oppilaitten vastaukset. Oppilaitten vastauksia tarkastellaan oppilaittain valitsemalla tietty oppilas. ALEA-projektissa ei muutettu ulkoisesti *Opettajan käyttöliittymää*. Sisäisesti *Opettajan käyttöliittymää* muutettiin siten, että käyttöliittymä osaa lukea XML:ää, koska vaatimusmäärittelyn (V9.1) mukaan pelilokit tallennettiin XML-muotoisena.



Kuva 4.7: *Opettajan käyttöliittymä*.

Kuvassa 4.8 on *Palvelin*-käyttöliittymä. Tähän käyttöliittymään tullaan *Kirjautuminen*-käyttöliittymästä onnistuneen kirjautumisen jälkeen. Tässä käyttöliittymässä näkyy metapalvelimella olevat pelisessiot, kun klikataan *Päivitä pelilista* -painiketta. *Palvelin*-käyttöliittymästä liitytään haluttuun peliin *Liity peliin* -painikkeen kautta tai luodaan oma peli *Luo uusi peli* -painikkeen kautta. Palvelimelta uloskirjautuminen tapahtuu *Kirjaudu ulos* -painikkeen kautta. *Palvelin*-käyttöliittymä on ALEA-projektin tuotos.



Kuva 4.8: *Palvelin*-käyttöliittymä.

Kuvassa 4.9 on *Luo peli* -käyttöliittymä, jossa pelin luoja luo pelisession. Tähän käyttöliittymään päästään *Palvelin*-käyttöliittymän *Luo uusi peli* -painikkeesta. *Luo peli* -käyttöliittymästä valitaan pelisessiolle kysymyssarja / kysymyssarjat *Valitse kysymyssarja omalta koneelta* -painikkeen kautta ja pelilauta *Valitse pelilauta omalta koneelta* -painikkeen kautta. *Luo peli* -käyttöliittymän *Pelin asetukset* -kehyksestä annetaan pelisessiolle nimi, maksimipelaajien määrä sekä halutessaan pelin luoja voi antaa pelisessiolle salasanan. Salasanan käyttö edellyttää *Käytä salasanaa* -valintaruudun valitsemista. *Luo peli* -käyttöliittymän oikeasta alareunasta löytyy valintaruutu, josta voi antaa pelille arvosteluominaisuuden. *Luo peli* -käyttöliittymä syntyi kokonaisuudessaan ALEA-projektissa.

The screenshot shows a Windows-style dialog box titled "Luo peli". It is divided into several sections:

- Kysymyssarjat**: A dropdown menu set to "Järjestä nimen mukaan". Below it, a button "Valitse kysymyssarja palvelimelta" is above a list box containing "Hieno", "Hienempi", and "Hienoin". A button "Valitse kysymyssarja omalta koneelta" is below the list. A text field contains "< Kysymyssarjaa ei ole valittu >". Below that are fields for "Vaikeustaso:" and "Aihealue:" with a "Lisää" button.
- Peliin tulevat kysymyssarjat**: An empty list box with a "Poista valitut kysymyssarjat" button below it.
- Pelilauta**: A dropdown menu set to "Järjestä nimen mukaan". Below it, a button "Valitse pelilauta palvelimelta" is above a list box containing "Hieno", "Hienempi", and "Hienoin". A button "Valitse pelilauta omalta koneelta" is below the list. A text field contains "< Pelilautaa ei ole valittu >". Below that is an "Aihealue:" field.
- Pelin asetukset**: Fields for "Pelin nimi" (containing "WebTalarious peli") and "Salasana". Below these are radio buttons for "Pelaajien lukumäärä" (yksi, kaksi, kolme, neljä) and a checkbox "Käytä salasanaa".
- Bottom**: A "Luo peli" button, a "Peruuta" button, and a checked checkbox "Pyydä arvosana pelin päätyttyä".

Kuva 4.9: *Luo peli* -käyttöliittymä.

Kuvassa 4.10 on *Pelin tarkemmat tiedot* -käyttöliittymä. Tähän käyttöliittymään päästään *Palvelin*- käyttöliittymän *Liity peliin* -painikkeesta. Tästä käyttöliittymästä löytyvät pelin nimi, pelin luoja, liittyneet pelaajat sekä pelattavan pelin tiedot omasta kehiksestään. Jos pelin luoja on suojannut pelisessionsa salasanalla, niin sitä varten löytyy *salasana*-kenttä, johon kirjoitetaan salasana. *Pelin tarkemmat tiedot* -käyttöliittymä on ALEA-projektin tuotos.

Pelin tarkemmat tiedot

Pelin tiedot
Pelin nimi: WebTalarious peli
Pelin luoja: Joni Kivinen

Pelaajat:
Joni Kivinen

Enintään: 3

Tarvitsee salasanan: ei
< kirjoita salasana tähän >

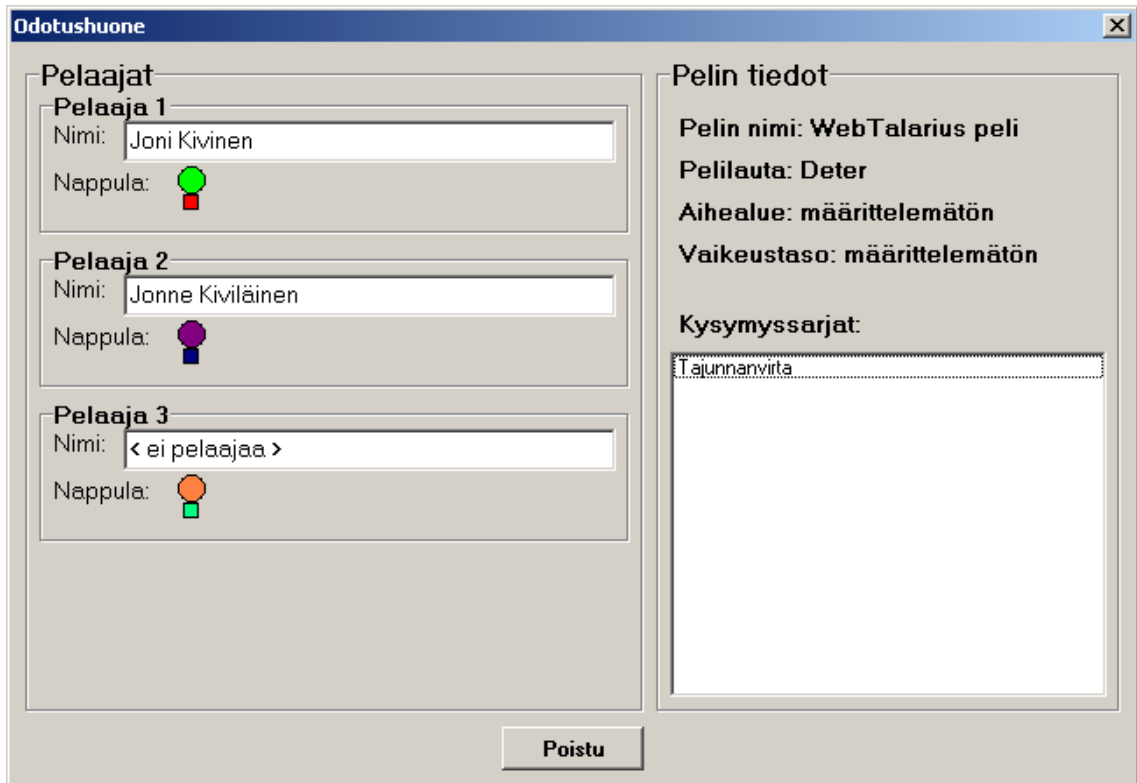
Pelattava peli
Pelilauta: Deter
Aihealue: määrittelemätön
Vaikeustaso: määrittelemätön

Kysymyssarjat:
Tajunnanvirta

Liity peliin **Peruuta**

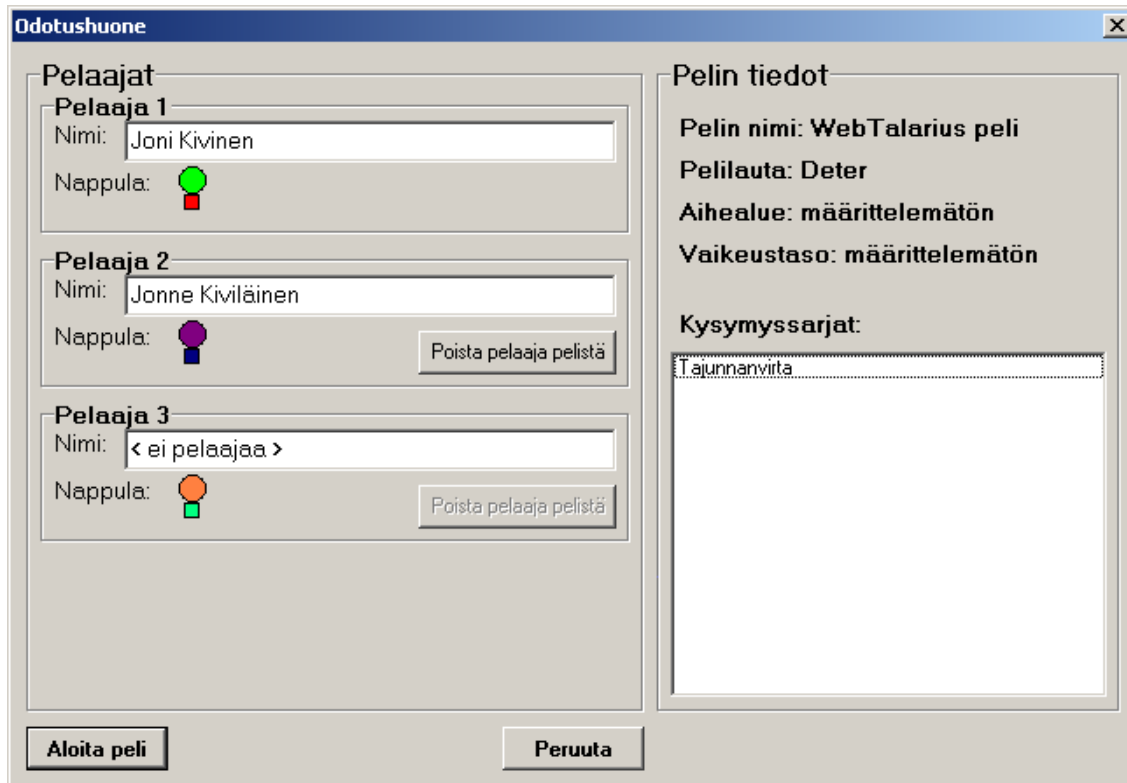
Kuva 4.10: *Pelin tarkemmat tiedot* -käyttöliittymä.

Kuvassa 4.11 on asiakkaan *Odotushuone*-käyttöliittymä. Tähän käyttöliittymään tullaan *Pelin tarkemmat tiedot* -käyttöliittymän *Liity peliin* -painikkeesta. Asiakkaan *Odotushuone*-käyttöliittymän *Pelaajat*-kehyksestä löytyvät pelaajien nimet, jotka ovat liittyneet kyseiseen peliin. *Pelin tiedot* -kehyksessä näkyvät pelin nimi, pelilaudan nimi, aihealue, vaikeustaso ja kysymyssarjat, jotka ovat kyseisessä pelisessiossa. Asiakkaan *Odotushuone*-käyttöliittymä on ALEA-projektin tuotos.



Kuva 4.11: Asiakkaan *Odotushuone*-käyttöliittymä.

Kuvassa 4.12 on pelin luojaan *Odotushuone*-käyttöliittymä. Tähän käyttöliittymän tullaan *Luo peli* -käyttöliittymän *Luo peli* -painikkeesta. Pelin luojaan *Odotushuone*-käyttöliittymässä pelin luoja voi poistaa pelaajia *Poista pelaaja pelistä* -painikkeen avulla ja halutessaan aloittaa pelin *Aloita peli* -painikkeen avulla, muuten pelin luojaan *Odotushuone*-käyttöliittymä ei eroa asiakkaan *Odotushuone*-käyttöliittymästä. Pelin luojaan *Odotushuone*-käyttöliittymä on ALEA-projektin tuotos.



Kuva 4.12: Pelin luojaan *Odotushuone*-käyttöliittymä.

5 Arkkitehtuuri

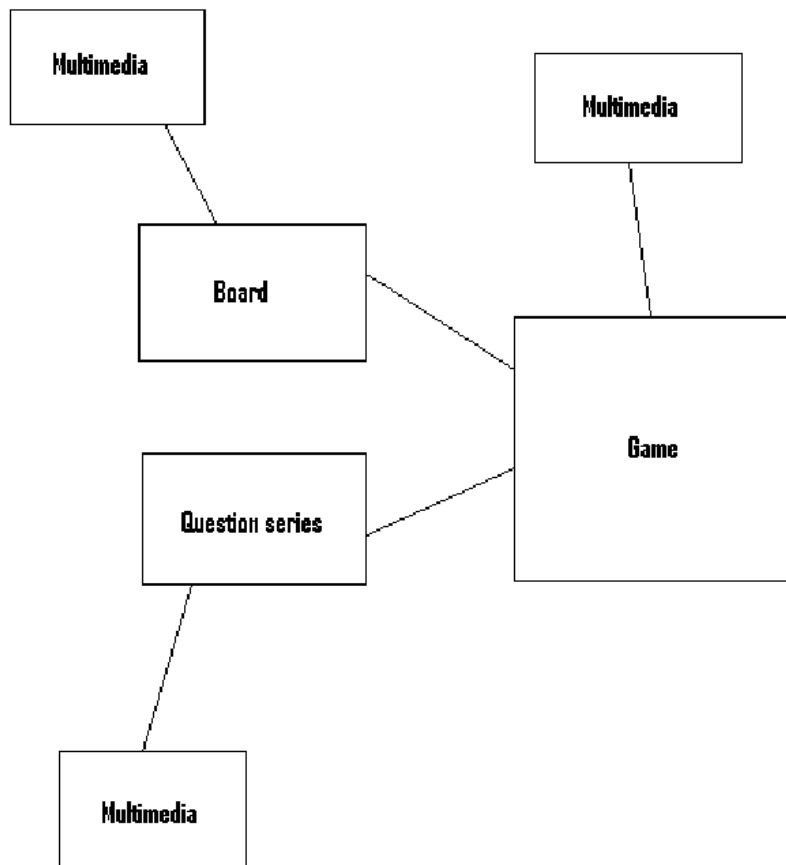
Tässä luvussa esitellään sovelluksen arkkitehtuuri ennen peliä, pelinaikainen arkkitehtuuri, sovelluksen yleinen arkkitehtuuri sekä pelinaikainen verkkoarkkitehtuuri, josta näkyy lähetettävät paketit.

5.1 Yleinen arkkitehtuuri

Sovellus koostuu lomakkeista, joista enemmän luvussa 4. Lomakkeilla asiakkaat pääsevät esimerkiksi liittymään metapalvelimeen ja luomaan pelin. Pelin aikana kaikki pelilogiikka suoritetaan pelin luojaan koneelta. Pelin aikana pelilogiikka hoitaa pelin kulkuun liittyvät viestit kaikkien asiakkaitten välillä, jotka ovat liittyneet pelilogiikkaan (Kuva 5.3). Sovellus sisältää XML-paketteja, joita lähetetään asiakkaiden, pelin luojaan, metapalvelimen ja pelipalvelimen välillä. Kaikki sovelluksen sisältämät XML-paketit löytyvät luvusta 5.4. Kuvassa 5.4 on eritelty tarkemmin, mistä ja mihin pelinaikaiset XML-paketit lähetetään.

5.2 Sovelluksen pelitiedostojen arkkitehtuuri

Kuvassa 5.1 on sovelluksen pelitiedostojen arkkitehtuuri. Peli on yhdistelmä pelilaudasta ja kysymyssarjasta tai -sarjoista. Pelilaudoilla ja kysymyssarjoilla on lisäksi omat multimediatiedostonsa. Pelillä itsellään on omia multimediatiedostoja.



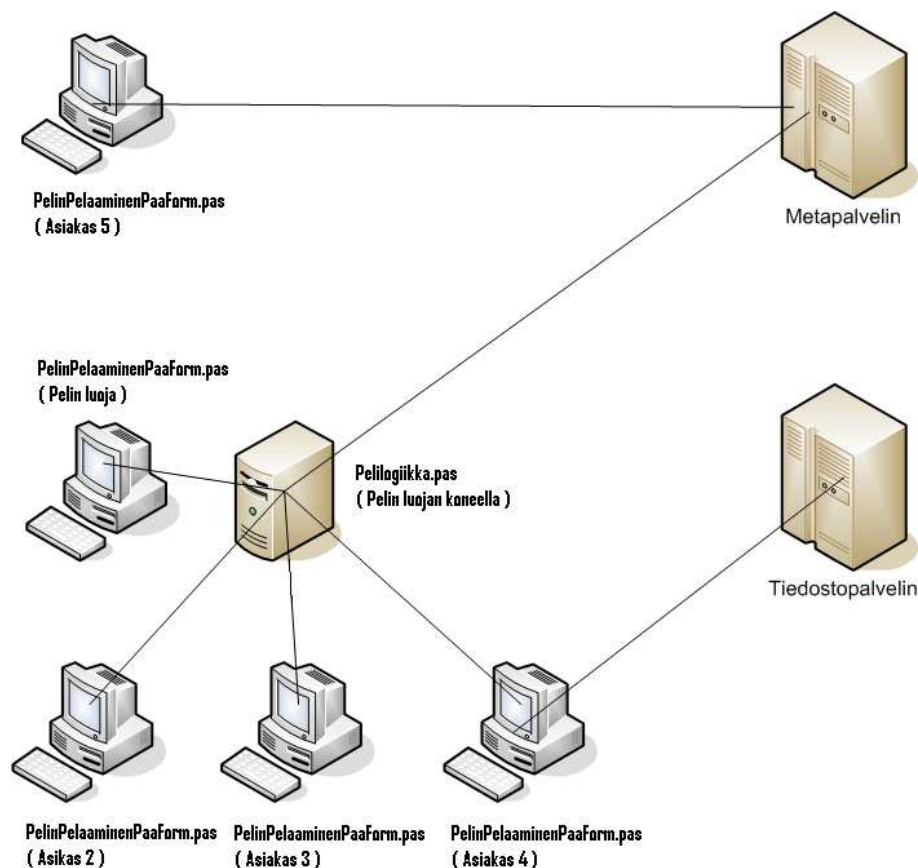
Kuva 5.1: Sovelluksen pelitiedostojen arkkitehtuuri.

5.3 Verkkoarkkitehtuuri

Kuvassa 5.2 on sovelluksen arkkitehtuuri ennen pelin alkua. Sovelluksessa on kaksi palvelinta, meta- ja tiedostopalvelin. Asiakkaat ovat liittyneet yhteen pelilogiikkaan, joka on pelinluojan koneella.

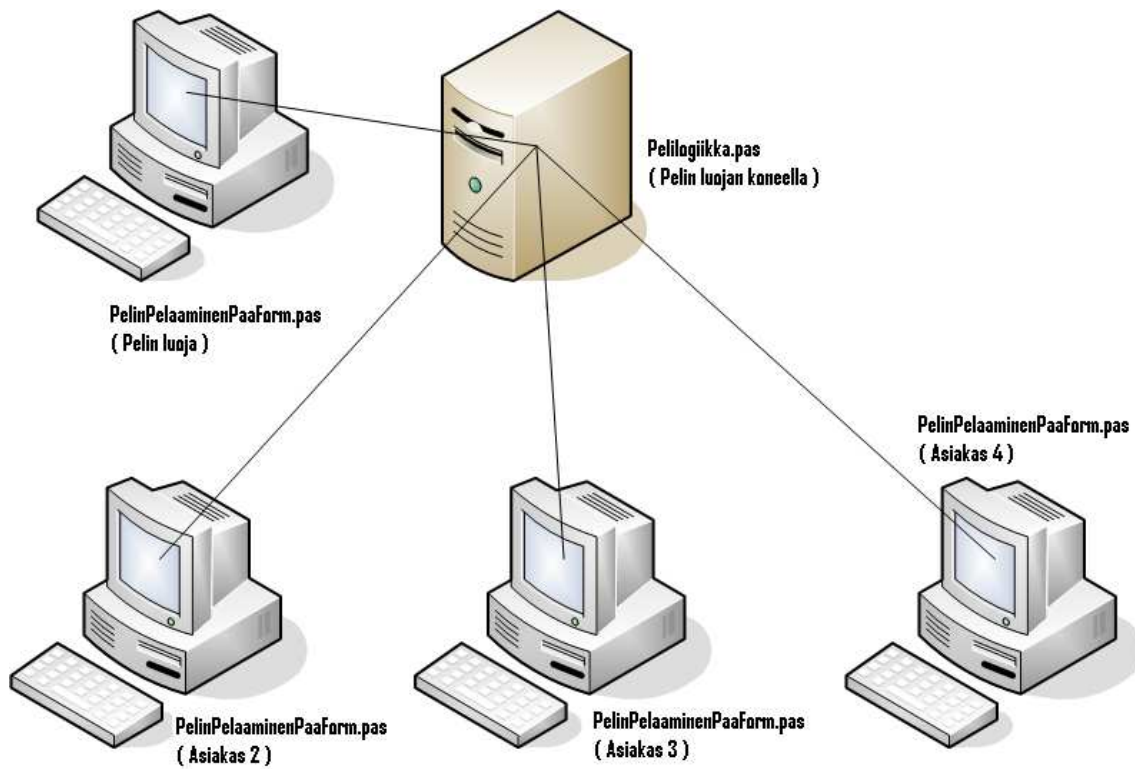
Metapalvelin on erillinen palvelin, joka ylläpitää listaa tarjolla olevista pelisessioista ja lisäksi tallentaa pelisessioista syntyvät lokitiedostot ja pelitallennukset. Metapalvelin huolehtii myös pelaajien pelitileistä. Tiedostopalvelin tallettaa pelisession pelaamiseen tarvittavia pelitiedostoja. Kuvasta 5.3 näkyy pelilogiikan tehtävä pelin aikana.

Pelaajan tunnistus tapahtuu kirjautumalla tunnuksella metapalvelimelle. Pelaajat selailevat luotuja pelejä metapalvelimelta ja liittyvät haluamaansa peliin. Pelaajalle, joka on liittymässä peliin, latautuu ennen pelin alkua kyseisen pelin pelitiedostot tiedostopalvelimelta.



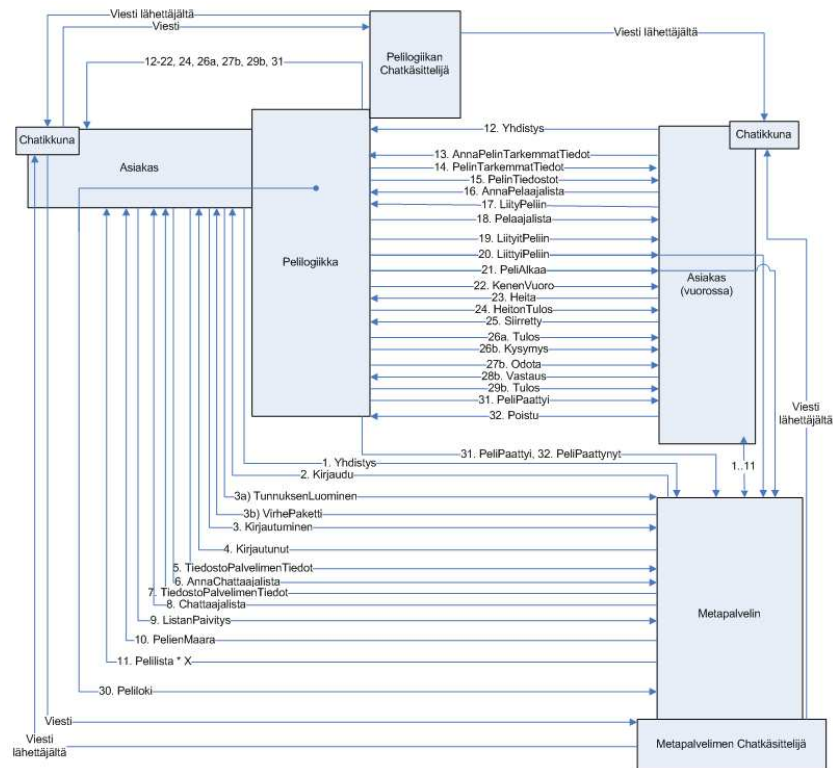
Kuva 5.2: Verkkoarkkitehtuuri ennen pelin alkua.

Kuvassa 5.3 on pelin aikainen arkkitehtuuri. Kaikki pelilogiikka suoritetaan pelinluojan koneelta. Pelaajien syötteet lähetetään asiakasohjelmalla pelilogiikalle ja tilanne päivitetään samantien jokaiselle asiakkaalle.



Kuva 5.3: Verkkoarkkitehtuuri pelin aikana.

Kuvassa 5.4 on pelinaikainen arkkitehtuuri. Kohdissa 1-4 asiakkaat kirjautuvat metapalvelimelle. Kohdissa 5-8 asiakkaat pyytävät metapalvelimelta tiedostopalvelimen tiedot ja chattaajien listan, ja metapalvelin lähettää kyseiset tiedot asiakkaille. Kohdissa 9-11 asiakkaat päivittävät metapalvelimelta pelilistan. Kohdissa 12-15 asiakas pyytää pelilogiikalta pelin tiedot. Kohdissa 16-20 asiakas pyytää pelilogiikalta pelaajalistan, ja tämän jälkeen asiakas liittyy peliin, josta tiedotetaan kaikille muille pelissä oleville asiakkaille ja metapalvelimelle. Kohdassa 21 pelin luoja on päättänyt aloittaa pelin, josta tiedotetaan kaikille asiakkaille ja metapalvelimelle. Kohdissa 22-29b on pelin kulkuun liittyviä viestejä asiakkaitten ja pelilogiikan välillä. Kohdassa 30 pelilogiikka lähettää metapalvelimelle pelilogin kyseisestä pelistä. Kohdissa 31-32 pelilogiikka lähettää asiakkaille sekä metapalvelimelle viestin pelin päättymisestä sisältäen mm. listan pelissä olleista asiakkaista. Asiakas lähettää viestin chat-käsittelijälle, joka lähettää viestin edelleen lähettäjän nimen kera kaikille siihen liittyneille asiakkaille. Kaaviokuvasta puuttuu poistumisviestit, jotka ovat seurausta prosessin keskeyttämisestä. Esimerkiksi pelilogiikka lähettää kaikille asiakkaille poistui- viestin, kun asiakas poistuu kesken pelistä. Kaikki pelilogiikan viestit metapalvelimelle kulkevat pelin luojan asiakkaan kautta. Kuva 5.4 löytyy isommas- sa koossa projektikansiosta.



Kuva 5.4: Sovelluksen verkkoarkkitehtuuri.

5.4 XML-paketit

Seuraavassa kerrotaan, mitä XML-paketteja sovellus sisältää, ja selitetään lyhyesti kyseinen XML-paketti. Pakettien nimien perässä on mainittu paketin lähettäjä ja vastaanottaja, muodossa *lähettäjä->vastaanottaja*. Lyhenteinä on käytetty asiakas (**A**), pelin luoja (**L**), metapalvelin (**M**), pelipalvelin (**P**) ja tiedostopalvelin (**T**).

- 0 **Ping (A->M)**: Varmistetaan, että ohjelmien välillä on liikennettä. Jos liikennettä ei ole Windowsin määräämänä aikavälinä niin Windows katkaisee yhteyden.
- 1 **AnnaChattaajaLista (A->M)**: Asiakas pyytää metapalvelimella olevien käyttäjien listan. Tähän listaan eivät kuulu käyttäjät, jotka ovat pelaamassa.
- 2 **ChattaajaLista (M->A)**: Palautetaan metapalvelimelta asiakkaalle käyttäjien lista. Tähän listaan eivät kuulu käyttäjät, jotka ovat pelaamassa.
- 3 **ChatViesti (A->M tai A->P)**: Viesti, joka halutaan lähettää kaikille, jotka ovat saman palvelimen chatissa.
- 4 **ChatViestiLahettajalta (M->A tai P->A)**: Viesti, joka kulkee kaikille, lähettäjä mukaan lukien, jotka ovat saman palvelimen chatissa.
- 5 **Yhdistys (A->P)**: Yhdistämispyyntö asiakkaalta pelilogiikalle.
- 6 **Kirjautu (M->A)**: Kirjautumispyyntö metapalvelimelta asiakkaalle.
- 7 **Kirjautuminen (A->M)**: Kirjautuminen metapalvelimelle.
- 8 **Kirjautunut (M->A)**: Palauttaa tiedon oliko tiedot oikein, eli onnistuiko kirjautuminen.
- 9 **ListanPaivitys (A->M)**: Pyytää metapalvelinta lähettämään listan saatavilla olevista peleistä.
- 10 **PelienMaara (M->A)**: Metapalvelin lähettää sillä olevien pelien määrän asiakkaalle.
- 11 **Pelilista (M->A)**: Metapalvelimen lähettämä osa pelilistasta saatavilla olevista peleistä, sisältää myös tietoa peleistä.
- 12 **UusiPeli (L->M)**: Pelin luoja ilmoittaa metapalvelimelle uuden pelin luomisesta, sisältää myös tietoa uudesta pelistä.

- 13 **AnnaPelinTarkemmatTiedot (A->P)**: Pyytää pelilogiikkaa kertomaan tarkemmat tiedot pelistä.
- 14 **PelinTarkemmatTiedot (P->A)**: Pelilogiikan lähettämä viesti, jossa on tarkemmat tiedot pelistä.
- 15 **AnnaPelaajalista (A->P)**: Asiakas pyytää pelilogiikalta pelissä mukana olevien listan.
- 16 **Pelaajalista (P->A)**: Pelilogiikka palauttaa pelissä mukana olevien listan.
- 17 **PelinTiedostot (P->A)**: Pelilogiikka lähettää asiakkaalle tiedot pelissä käytössä olevista pelitiedostoista.
- 18 **LiityPeliin (A->P)**: Asiakas yrittää liittyä peliin.
- 19 **LiityitPeliin (P->A)**: Ilmoittaa asiakkaalle peliin pääsystä tai syystä miksi ei päässyt.
- 20 **LiittyiPeliin (P->A)**: Ilmoitus kaikille asiakkaille uudesta peliin liittyneestä asiakkaasta.
- 21 **Poistu (A->P)**: Asiakas ilmoittaa pelistä poistumisesta pelilogiikalle.
- 22 **Poistui (P->A ja L->M)**: Pelilogiikka ilmoittaa kaikissa pelissä mukana olleille ja metapalvelimelle, että pelaaja on poistunut pelistä.
- 23 **PeliAlkaa (P->A ja L->M)**: Ilmoitus pelilogiikalta asiakkaille ja metapalvelimelle pelin alkamisesta.
- 24 **KenenVuoro (P->A)**: Pelilogiikka lähettää kaikille pelaajille tiedon kenen vuoro on kyseessä.
- 25 **Heita (A->P)**: Pelivuorossa oleva pelaaja kertoo pelilogiikalle heitonapin painamisesta.
- 26 **HeitonTulos (P->A)**: Pelilogiikka kertoo kaikille pelaajille, miten paljon heitettiin ja paljonko siirryttiin (oleellista, mikäli vastaan tulee toimintaruutu (V8.2)).
- 27 **Siirretty (A->P)**: Pelivuorossa oleva pelaaja ilmoittaa pelilogiikalle suorittamansa siirtojen lukumäärän.

- 28 **Odota (P->A):** Pelilogiikka ilmoittaa kaikille asiakkaille, että odotetaan pelivuorossa olevan asiakkaan toimintaa.
- 29 **Kysymys (P->A):** Pelilogiikka ilmoittaa pelivuorossa olevalle pelaajalle, mikä kysymys tulee kysyä, mikäli ollaan kysymysruudussa.
- 30 **Vastaus (A->P):** Asiakas lähettää kysymyksen vastauksen pelilogiikalle.
- 31 **Tulos (P->A):** Pelilogiikka ilmoittaa, mikä ruutu tulee näyttää asiakkaille, esimerkiksi vastaus oikein.
- 32 **PeliPaattyi (P->A ja L->M):** Pelilogiikka ilmoittaa asiakkaalle ja pelin luoja ilmoittaa metapalvelimelle pelin päättymisestä.
- 33 **Arvostelu (A->P):** Asiakas antaa pelilogiikalle arvosanan pelatulle pelille.
- 34 **PeliPaattynyt (L->M):** Pelin luoja kertoo metapalvelimelle pelin päättymisestä ja lähettää loppuneen pelin tiedot.
- 35 **KatkaiseYhteys (A->M):** Asiakas lähettää poistumisilmoituksen metapalvelimelle.
- 36 **VirhePaketti (M->A):** Metapalvelin lähettää asiakkaalle kirjautumisessa mahdollisesti ilmenneen virheen.
- 37 **Lokitiedosto (L->M):** Pelin luoja lähettää pelin päätyttyä pelistä syntyneen XML-muotoisen pelilokin metapalvelimelle.

5.5 Tiedostopalvelimen XML-paketit

Seuraavassa jatkokehitystä varten tehdyt tiedostopalvelimen tarvitsemat XML-paketit.

- 38 **KirjaaTiedostopalvelimeksi (T->M):** Metapalvelimella asiakkaana oleva tiedostopalvelin ilmoittaa metapalvelimelle olevansa tiedostopalvelin.
- 39 **AnnaTiedostopalvelimenTiedot (A->M):** Asiakas pyytää metapalvelimelta tiedostopalvelimen tiedot.
- 40 **TiedostopalvelimenTiedot (M->A):** Metapalvelin palauttaa asiakkaalle tiedostopalvelimen tiedot.

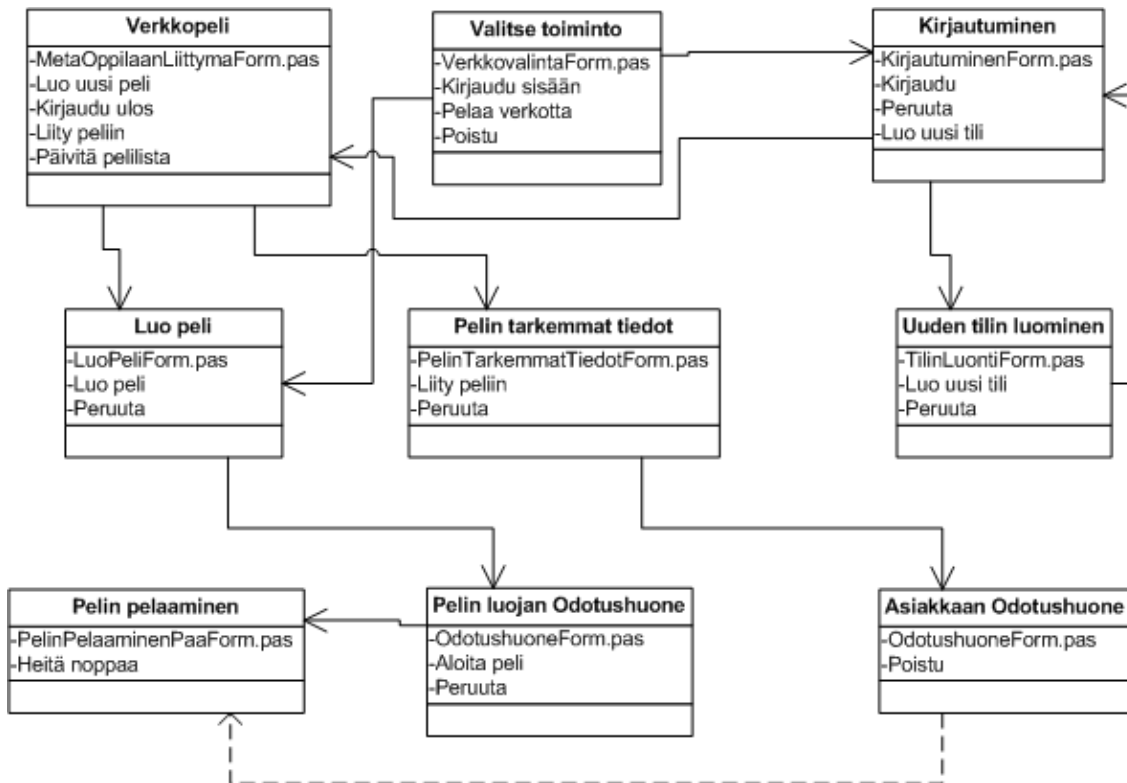
- 41 **AnnaTunnuslista (T->M):** Tiedostopalvelin pyytää metapalvelimelta asiakkaiden käyttäjätunnuslistan.
- 42 **Tunnuslista (M->T):** Metapalvelin palauttaa tiedostopalvelimelle asiakkaiden käyttäjätunnuslistan.
- 43 **LisattyTunnus (M->T):** Metapalvelin lähettää tiedostopalvelimelle asiakkaan luoman uuden tunnuksen.

6 Luokat

Tässä luvussa esitellään käyttöliittymien välisiä suhteita ja sovelluksessa käytettäviä luokkia.

6.1 Käyttöliittymien väliset suhteet

Kuvassa 6.1 on sovelluksen käyttöliittymien väliset suhteet. Ensimmäisellä rivillä on lomakkeen nimi ja toisella rivillä tiedosto, jossa lomake on. Näiden jälkeen on lueteltu painikkeet, joista pääsee siirtymään toisille lomakkeille. Nuolet lomakkeiden välissä kuvaavat, mistä lomakkeelta siirrytään mihinkin lomakkeelle.



Kuva 6.1: Käyttöliittymien väliset suhteet.

Valitse toiminto -lomakkeen *Kirjaudu sisään* -painikkeesta siirrytään *Kirjautuminen*-lomakkeelle, *Pelaa verkotta* -painikkeesta siirrytään *Luo peli* -lomakkeelle ja *Poistu*-painikkeesta ohjelma sulkeutuu.

Kirjautuminen-lomakkeen *Kirjaudu*-painikkeesta siirrytään luoduilla tunnuksilla *Verk-*

kopeli-lomakkeelle. *Kirjautuminen*-lomakkeen *Peruuta*-painikkeesta palataan *Valitse toiminto* -lomakkeelle ja saman lomakkeen *Luo uusi tili* -painikkeesta siirrytään *Uuden tilin luominen* -lomakkeelle.

Uuden tilin luominen -lomakkeen *Luo uusi tili* -painikkeen kautta luodaan pelitili ja palataan takaisin *Kirjautuminen*-lomakkeelle. *Uuden tilin luominen* -lomakkeen *Peruuta*-painikkeesta palataan takaisin *Kirjautuminen*-lomakkeelle.

Verkkopeli-lomakkeen *Luo uusi peli* -painikkeesta siirrytään *Luo peli* -lomakkeelle, jossa peli luodaan. *Verkkopeli*-lomakkeen *Päivitä pelilista* -painikkeesta haetaan metapalvelimelta pelit, ja valitaan listasta peli, johon halutaan liittyä. *Liity peliin* -painikkeesta liitytään haluttuun peliin ja siirrytään *Pelin tarkemmat tiedot* -lomakkeelle. *Kirjaudu ulos* -painikkeesta palataan takaisin *Valitse toiminto* -lomakkeelle.

Luo peli -lomakkeen *Luo peli* -painikkeesta pelin luoja siirtyy *Pelin luojan Odotushuone* -lomakkeelle. *Luo peli* -lomakkeen *Peruuta*-painikkeesta palataan takaisin *Verkkopeli*-lomakkeelle.

Pelin tarkemmat tiedot -lomakkeen *Liity peliin* -painikkeesta asiakas siirtyy *Asiakkaan Odotushuone* -lomakkeelle. Saman lomakkeen *Peruuta*-painikkeesta palataan takaisin *Verkkopeli*-lomakkeelle.

Asiakkaan Odotushuone -lomakkeelta asiakas siirtyy *Pelin pelaaminen* -lomakkeelle, kun pelin luoja käynnistää pelin. *Asiakkaan Odotushuone* -lomakkeen *Poistu*-painikkeesta asiakas palaa takaisin *Pelin tarkemmat tiedot* -lomakkeelle.

Pelin luojan Odotushuone -lomakkeen *Aloita peli* -painikkeesta pelin luoja käynnistää pelin, ja siirtyy *Pelin pelaaminen* -lomakkeelle.

Pelin pelaaminen -lomakkeella tapahtuu itse pelaaminen.

6.2 Sovelluksen luokat

6.2.1 Asiakas

Seuraavassa listattu asiakasohjelman käyttämät luokat.

- 1 **TFormPelinPelaaminen** (*PelinPelaaminenPaaForm.pas*) on asiakasohjelman pääformi. *TFormPelinPelaaminen* pääasiallinen tehtävä on toimia pelinäkäisena näyttönä, mutta se toimii myös pelin toimintojen runkona.

- 2 **TPelaajanVastaukset** (PelaajaVastaukset.pas) sisältää pelaajan tiedot ja taulukon pelaajan vastauksista.
- 3 **TKysymyksenVastaus** (PelaajaVastaukset.pas) sisältää pelaajan vastauksen tiedot.
- 4 **TPelilogiikka** (Pelilogiikka.pas) hoitaa pelilogiikan ja siihen liittyvien viestien lähetyksen ja vastaanoton. Uusi pelilogiikka luodaan aina peliä varten ja tuhotaan pelin päätteeksi. Pelilogiikan luo ja omistaa TFormPelinPelaaminen. Pelilogiikan alustaa TFormLuoPeli uuden pelin luomisen yhteydessä.
- 5 **TKysymyskanta** (KysymysLuokat.pas) sisältää kysymystyyppeihin liittyvien tiedostojen polut.
- 6 **TTietoruutu** (KysymysLuokat.pas) sisältää tietoruudun tekstimuotoisen datan.
- 7 **TAvoinkysymys** (KysymysLuokat.pas) sisältää avoimen kysymyksen tekstimuotoisen datan.
- 8 **TMonivalintaKysymys** (KysymysLuokat.pas) sisältää monivalintakysymyksen vastauksen vaihtoehdot, niihin liittyvät kuvat ja oikean vastauksen.
- 9 **TPelilaudanObjekti** (Objekti.pas) sisältää pelilaudalla olevien ruutujen tiedot.
- 10 **TPelinappula** (Objekti.pas) sisältää pelilaudalla olevan pelaajan nappulan tiedot.
- 11 **THahmo** (Objekti.pas) sisältää pelilogiikassa ja pelin pelaamisessa tarvittavat pelaajan tiedot.
- 12 **TFormChat** (ChatIkkuna.pas) hoitaa chat-viestien lähetyksen ja vastaanotettujen chat-viestien näyttämisen.
- 13 **TFormOdotushuone** (OdotushuoneForm.pas) näyttää peliin liittyneet asiakkaat ja pelin tiedot. TFormOdotushuone käsittelee mm. liittymis- ja poistumisviestit.
- 14 **TFormLuoPeli** (LuoPeliForm.pas) hoitaa pelin luonnin ja alustaa pelilogiikan lomakkeelle laitetuilla tiedoilla.

- 15 **TFormPeliTietoruutu** (PelinPelaaminenTietoruutuForm.pas) on tietoruutu-lomake, joka näyttää tietoruutuun liittyvät tiedot.
- 16 **TFormPeliMonivalintakysymys** (PelinPelaaminenMonivalintakysymysForm.pas) on monivalintakysymys-lomake, joka näyttää monivalintakysymykseen liittyvät tiedot ja käsittelee pelaajan antaman vastauksen.
- 17 **TFormPeliAvoinKysymys** (PelinPelaaminenAvoinKysymysForm.pas) on avoin kysymys -lomake, joka näyttää avoimeen kysymykseen liittyvät tiedot ja käsittelee pelaajan antaman vastauksen.
- 18 **TFormIlmoitus** (IlmoitusForm.pas) näyttää pelaajille kaikki ilmoitukset.
- 19 **TFormPelinTarkemmatTiedot** (PelinTarkemmatTiedotForm.pas) näyttää pelilogiikalta saadut pelin tiedot asiakkaalle.
- 20 **TFormTilinLuonti** (TilinLuontiForm.pas) hoitaa asiakkaan tilin luonnin metapalvelimelle.
- 21 **TFormKirjautuminen** (KirjautuminenForm.pas) hoitaa asiakkaan kirjautumisen metapalvelimelle.
- 22 **TFormMetaOppilaanLiittyma** (MetaOppilaanLiittymaForm.pas) näyttää metapalvelimella olevat pelit ja siitä voi aloittaa pelin luonnin tai pyytää valitun pelin tarkempia tietoja.
- 23 **TFormVerkkovalinta** (VerkkovalintaForm.pas) mahdollistaa kirjautumisen metapalvelimelle ja verkotta pelaamisen.

6.2.2 Metapalvelin

Seuraavassa listattu metapalvelimen käyttämät luokat.

- 1 **TFormMetapalvelin** (MetapalvelinForm.pas) on metapalvelimen päälomake, joka sisältää suurimman osan metapalvelimen toiminnoista.
- 2 **TOpettajaTili** (MetapalvelinForm.pas) varastoi opettajatilin tiedot.
- 3 **TKayttajatunnus** (MetapalvelinForm.pas) varastoi käyttäjätunnuksen tiedot.

- 4 **TAsiakas** (MetapalvelinForm.pas) varastoi kirjautuneen asiakkaan tiedot, ylläpitää verkkosäikeen ja pelaajan tilaa.
- 5 **TFormMetapalvelimenSalasana** (MetapalvelimenSalasanaForm.pas) on käytössä metapalvelimen salasanaa vaihdettaessa ja lukitessa. Sisältää salasana-kentät käyttäjän syötteitä varten, ja suorittaa annetuille syötteille yksinkertaisen tarkistuksen.
- 6 **TFormMetapalvelimenPortti** (MetapalvelimenPorttiForm.pas) on käytössä metapalvelimen porttia vaihdettaessa. Sisältää porttikentän käyttäjän syötettä varten, ja suorittaa annetulle syötteelle yksinkertaisen tarkistuksen.
- 7 **TFormLuoOpettajatili** (LuoOpettajatiliForm.pas) on käytössä opettajatiliä luodessa. Sisältää opettajatunnuskentän ja salasana-kentät käyttäjän syötteitä varten, ja suorittaa annetuille syötteille yksinkertaisen tarkistuksen.

6.2.3 Paketit

Seuraavassa on esitelty pakettiluokka ja esimerkkinä yksi siitä peritty luokka.

TPaketti (Paketit.pas) sisältää attribuutteina *Paketti* (String), *PaketinTyyppi* (Pakettityyppi) ja *XMLTulkki* (IXMLDocument). *Paketti*-attribuutti sisältää lähetettäessä lähetettävän XML-dokumentin, ja vastaanotettaessa sitä ei käytetä. *PaketinTyyppi*-attribuutti sisältää paketin tyyppin arvon. *XMLTulkki*-attribuuttia käytetään vastaanotettaessa vastaanotetun XML-dokumentin tulkinnassa, ja lähetettäessä *XMLTulkin* sisältämään XML-dokumenttiin lisätään lähetettävät tiedot. TPaketti sisältää metodit *VastaanotettavanMuodostaja* ja *LahetettavanMuodostaja*, jotka hoitavat pakettien vastaanoton ja lähetyksen alustamisen. *VastaanotettavanMuodostaja*-metodi alustaa *XMLTulkki*-attribuutin annetulla parametrilla, etsii vastaanotetusta XML-dokumentista paketin tyyppin ja palauttaa *XMLTulkki*-attribuutissa olevan XML-dokumentin juurielementin. *LahetettavanMuodostaja*-metodi alustaa *XMLTulkki*-attribuutin, lisää *XMLTulkki*-attribuutissa olevaan XML-dokumenttiin parametrina annetun paketin tyyppin, sijoittaa *PaketinTyyppi*-attribuuttiin parametrina annetun arvon ja palauttaa *XMLTulkki*-attribuutissa olevan XML-dokumentin juurielementin.

Esimerkkinä TPaketista peritystä luokasta on **TKirjautuminen**, jolla on attribuutteina *Kayttajatunnus* (String) ja *SalasanaMD5* (MD5Digest). Lähetettäessä alustuksessa annetaan parametreinä asiakkaan käyttäjätunnus ja salasanan MD5-tarkistussumma.

Luokka alustaa XML-dokumenttitulkin *LahetettavanMuodostajalla*. Luotuun XML-dokumenttiin lisätään parametrien tiedot ja muodostetaan lähetettävä XML-muotoinen merkkijono. Vastaanottaessa alustuksessa annetaan parametrina XML-muotoinen merkkijono. Luokka alustaa XML-dokumenttitulkin *VastaanotettavanMuodostajalla*. Luodusta XML-dokumentista luetaan paketin tyyppi ja arvot luokan attribuutteihin *Kayttajatunnus* ja *SalasanaMD5*.

7 Lähdekoodin ulkoasu

WebTalarius-sovellus oli jatkokehitys projekti, ja siitä johtuen koodin kommentoinnissa ja nimeämisissä käytettiin Talarius-sovelluksessa käytettyä kommentointi- ja nimeämiskäytäntöä. Sovelluksen kommentointi, luokan metodit ja attribuutit ovat suomeksi.

Attribuutit nimettiin seuraavasti:

MuuttujanNimi: TLuokka;

Metodit nimettiin seuraavasti:

procedure/function KuvaavaNimi(Parametri: TLuokka; Parametri2: TLuokka2);

Esimerkki TFormPelinPelaaminen-luokasta:

```
//ladataan kysymyssarjat-listassa olevat kysymyssarjat muistiin.
procedure TFormPelinPelaaminen.LataaKysymyssarjat();
var I: Integer;
begin
  FreeAndNil(KysymystenDataLista);
  KysymystenDataLista := TObjectList.Create;
  for I:= 0 to Kysymyssarjat.Count-1 do begin
    KysymystenDataLista.Add(LataaKysymyssarja(Kysymyssarjat[I]));
  end;
end;
```


8 Testaus

Tässä luvussa kerrotaan WebTalaria-sovelluksen testauksesta.

8.1 Toteutuksen aikainen testaus

Sovellusta testattiin useassa eri vaiheessa. Luokka- ja integrointitestausta suoritettiin projektiryhmän toimesta sitä mukaa kun sovellukseen tuli uusia toiminnallisuksia, metodeja tai luokkia.

Sovelluksen järjestelmätestaukseen osallistui Keltinmäen koulun 4-luokka. Oppilaat testasivat sovellusta Agoran mikrotietokoneluokassa toisen iteraation lopussa viikolla 16. Ohjaaja Ville Isomöttönen ja tilaaja testasivat sovellusta projektin lopussa.

8.2 Ensimmäinen testikierros

Projektin lopussa projektiryhmä järjestelmätestasi sovellusta testaussuunnitelman [1] mukaisesti. Testaus suoritettiin Windows XP-käyttöjärjestelmällä ja resoluutio oli 1280 x 1024. Testitapauksista testattiin 7 tapausta 18:sta, joiden läpimenoprosentti oli 100%. Koska kaikkia vaatimuksia ei saatu tehtyä, niin ryhmä pystyi testaamaan vain 7 tapausta 18:sta.

Testitapaus Id:11 toimi testitapauskuvauksen mukaisesti. Huomautuksena voidaan todeta, että yhteyden katketessa käyttäjätunnus jää metapalvelimelle voimaan, eli samoilla tunnuksilla ei voida kirjautua takaisin metapalvelimelle. Edellä mainittu ongelma korjattiin metapalvelimelle.

8.3 Toinen testikierros

Projektiryhmä järjestelmätestasi sovelluksen testaussuunnitelman [1] mukaisesti toisenkin kerran. Testausympäristö oli samanlainen, kuin ensimmäisellä testikierroksella. Testitapauksista testattiin 7 tapausta 18:sta, joiden läpimenoprosentti oli 100%.

9 Jatkokehitysideat

Tässä luvussa kerrotaan, mitkä vaatimukset jäivät tekemättä sekä kerrotaan näiden vaatimuksien jatkokehitysideoita.

9.1 Tekemättä jääneet vaatimukset

Ryhmä ei ehtinyt tehdä kaikkia pakollisia vaatimuksia sovellukseen. Tekemättä jäivät pakollisista vaatimuksista: *Sisällön poistaminen (V4.2)* ja *tiedostojen siirto (V9.3)*.

Tärkeistä vaatimuksista tekemättä jäivät *pelin arvostelu pelin päättyessä (V1.6)*, *kysymyssarjojen aihealueen ja vaikeustason määrittäminen (V2.2)*, *pelilautojen aihealueen määrittäminen (V2.3)*, *pelitiedostojen selaaminen ja lataaminen (V3.6)*, *toimintaruudut (V8.2)* ja *mediatiedostojen kokorajoitus (V9.2)*.

Mahdollisista vaatimuksista tekemättä jäivät *pelitilanteen tallentaminen (V1.1)*, *pelin jatkaminen tallennuksesta (V1.2)*, *kysymyksien pisteytys (V2.1)*, *pelien arvostelu (V3.3)*, *pelien ja kysymyssarjojen järjestäminen (V5.3)*, *pelitiedostojen lukitseminen (V7.2)*, *pakollinen pysähtyminen (V8.3)* ja *voittajan määräytyminen (V8.4)*.

Ajan salliessa -vaatimuksista tekemättä jäivät *käsin siirtäminen (V1.7)*, *omien tilastojen tarkasteleminen (V3.1)*, *tiedostojen poistaminen (V3.4)*, *suora tiedostojen siirto asiakkaiden välillä (V7.5)* ja *vaihtoehtoiset palkkiot (V8.5)*.

9.2 Vaatimusten jatkokehitysideat

Seuraavassa mainittavat pakettien tukemiset tarkoittavat, että kyseisessä vaatimuskohtassa tarvittavat paketit on jo toteutettu.

Jatkokehitystä varten on tehty jo valmiiksi metapalvelimelle sisällön poisto tiedostopalvelimelta (V4.2). Tiedostopalvelimen prototyyppi ei osaa vielä käsitellä tulleita poistokäskyjä. Tiedostojen siirtoa (V9.3) varten on tiedostopalvelimella jo valmiina oppilaiden tunnusten kysyminen. Vielä pitää tehdä tiedostopalvelimelle siirrettyjen tiedostojen käsittely.

Pelien arvostelun kysymiselle (V1.6) on valmiina paketit. Vielä pitää tehdä oppilaalle näytettävä arvosanan kysymis -lomake. Paketit tukevat pelilautojen aihe-

lueita (V2.3) ja kysymyssarjojen vaikeustasoja ja aihealueita (V2.2). Pelilautojen ja kysymyssarjojen tekoon ja lukuun pitää lisätä aihealueet ja vaikeustaso. Pelitiedostojen selaamista ja lataamista (V3.6) varten pitää vielä tehdä oppilaalle käyttöliittymä. Osa toimintaruutujen (V8.2) tarvitsemista paketeista on jo toteutettu, mutta pelilogiikasta puuttuu käsittely. Loput toimintaruudut vaativat toimiakseen niiden lisäyksen pelilaudan tekoon ja toteuttaa niiden tarvitsemat paketit ja toiminnallisuuden pelilogiikalle. Tiedostopalvelimelle pitää lisätä tiedoston siirron yhteyteen mediatiedostojen kokorajoituksen (V9.2) tarkistus.

Pelitilanteen tallennusta (V1.1) ja tallennuksesta jatkoa (V1.2) ei katsottu tarpeelliseksi, koska yhden pelin kesto on suhteellisen lyhyt. Kysymysten pisteytykselle (V2.1) on tehty jo tuki paketteihin. Kysymyssarjojen tekoon on vielä lisättävä tuki kysymysten pisteytykselle. Pelin arvostelua (V3.3) varten on valmiina paketit. Metapalvelimelle pitää vielä lisätä arvosanojen tallennus. Pelin luonnissa on valmiina alusvetovalikot pelien ja kysymyssarjojen järjestämiselle (V5.3), mutta niistä puuttuu vielä toiminta. Pelitiedostojen lukitsemista (V7.2) ei katsottu tarpeelliseksi ominaisuudeksi toteuttaa. Pelilogiikka ja paketit tukevat pakollinen pysähtyminen -ruutuja (8.3). Pelilaudan tekoon pitää lisätä vielä pakolliset pysähtymis -ruudut. Pelilogiikka ei tue vielä eri vaihtoehtoja voittajan määräytymiselle (V8.4).

Käsin siirtoa (V1.7) varten on valmiina tuki paketeissa ja pelilogiikassa. Pelilogiikka tallentaa tarvittavan siirron suuruuden. Vielä pitäisi lisätä tuki pelin pelaamisen puolelle. Omien tilastojen tarkastelua (V3.1) varten pitää tehdä vielä paketit, metapalvelimelle tietojen tallennus ja lähetys ja oppilaalle lomake tietojen tarkasteluun. Tiedostojen poistamista varten (V3.4) pitää tehdä paketit, oppilaalle käyttöliittymä ja metapalvelimelle pyyntöjen tallennus. Suora tiedostojen siirto asiakkaiden välillä (V7.5) ei onnistu helposti nykyisillä verkkokomponenteilla, joten sitä ei toteutettu. Vaihtoehtoiset palkkiot (V8.5) tarvitsevat toimiakseen tuen kysymyssarjojen tekoon.

9.3 Toteutusratkaisujen analyysi

Aluksi valitsemamme komponentit (IdTCPClient, IdTCPServer) eivät toimineet, mikä seurauksena menetettiin paljon aikaa ja jouduttiin suunnittelemaan paketit uudelleen. Lisäksi olisi pitänyt hoitaa muistinhallinta ja muuttujien välitys eri metodien välillä huolellisesti alusta saakka.

Valittu ketterä prosessimalli oli ehdottoman hyvä valinta, sillä muuten projektin

puitteissa ei olisi ollut mahdollista saada näin paljon aikaan. Lisäksi projektin aikana oli jatkuvasti seuraava päämäärä suunniteltu, joka lisäsi tehokkuutta.

10 Lähteet

- [1] Hankaniemi Arto, Holopainen Timo, Kivinen Joni, Kuismanen Riku, "ALEA-Sovellusprojekti Testaussuunnitelma", Jyväskylän yliopisto, tietotekniikan laitos, 2006.