

SAVONIA

ammattikorkeakoulu

PROJEKTIN LOPPURAPORTTI - AMMATTIKORKEAKOULUTUTKINTO
TEKNIIKAN JA LIIKENTEEN ALA

KÄYTTÖLIITTYMÄOHJELMOINTI

Ristinolla

TEKIJÄ: Kalle Koljonen

SISÄLTÖ

1	OHJELMAN YLEISKUVAUS	3
2	KÄYTTÖOHJE	4
2.1	MainPage	4
2.2	Start	6
2.3	ScoreBoard	7
3	TÄRKEIMMÄT SOVELLUKSEEN LIITTYVÄT TIEDOSTOT	8
3.1	MainPage.xaml ja MainPage.xaml.cs	8
3.2	Start.xaml ja Start.xaml.cs	8
3.3	ScoreBoard.xaml ja ScoreBoard.xaml.cs	9
3.4	InfoStruct.cs	9
3.5	Players.json	9
4	ITSEARVIO JA ARVIO TYÖHÖN KÄYTETYSTÄ AJASTA	10

1 OHJELMAN YLEISKUVAUS

Harjoitustyöksi valikoitui toteuttaa ristinolla peli. Sovellus käynnistyy MainPagella, jossa kysytään käyttäjältä halutut henkilötiedot, sekä vastustaja, jota vastaan käyttäjä haluaa pelata. Vastukseen käyttäjä voi valita joko toisen pelaajan tai tekoälyn. Tämän lisäksi etusivulta löytyy Scoreboard-nappi, jota painamalla siirrytään toiselle sivulle, josta löytyy tilastotietoa rekisteröityneistä pelaajista, sekä heidän aiemmista peleistään. Etusivulta pääsee aloittamaan pelin painamalla StartGame-nappia, jonka jälkeen käyttäjä ohjataan varsinaiselle pelisivulle. Pelisivulla on ajastin, josta näkee pelin keston, sekä NewGame-nappi, josta voi missä tahansa kohtaa nollata pelin ja aloittaa alusta.

MainPage-sivulla tallennetaan pelaajan henkilötiedot tietueeseen, ja pelisivulla tallennetaan pelien tulokset sekä kokonaisaika, jota peliä on pelattu. Nämä tiedot esitetään Scoreboard sivulla, kun käyttäjä navigoi sinne.

2 KÄYTTÖOHJE

2.1 MainPage

Seuraavassa kuvataan ohjelman aloitussivun ominaisuuksia ja toimintaa liitekuvien, sekä selitysten kautta. Ohjelman aloitussivu on koodissa nimellä MainPage.

KUVA 1.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent:

Ohjelma käynnistyy MainPagelle ja näkymä avautuu kuvan 1 osoittamalla tavalla. Layout on rakennettu XAML puolella grid muotoon, ja aloitussivulla käytetään toimintoja, joita on ohjelmoitu nappeihin, entry-komponentteihin sekä picker-komponenttiin.

KUVA 2.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent:

Tarkoituksena on syöttää uuden pelaajan tiedot niille osoitettuihin kenttiin. Tiedot tallentuvat json-muodossa välimuistiin, ja rekisteröity pelaaja löytyy seuraavalla kerralla sovelluksen aukaisemisen jälkeen myös picker-komponentin valikosta.

KUVA 3.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent:

KUVA 4.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent:

Picker-komponenttia käyttäessä saadaan esille lista aiemmin rekisteröityneistä pelaajista. Käyttäjä voi halutessaan valita pelaajan listalta, jolloin valitun pelaajan tiedot täyttyvät automaattisesti niille osoitettuihin kenttiin.

KUVA 5.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent

Kun kaikki kentät ovat täytetty, pitää käyttäjän vielä valita itselleen vastustaja. Vaihtoehtoja on kaksi, joko tekoäly AI tai toinen pelaaja Player 2. Klikkaamalla jompaakumpaa vaihtoehtoa, valinta osoitetaan keltaisilla ääriviivoilla. Vastustajaksi valikoituu se vaihtoehto, joka on aktiivisena, sillä hetkellä, kun start game-nappia painetaan.

KUVA 6.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent

Error
Fill every field with appropriate information

Kuvissa 6-8 havainnoidaan poikkeustilanteita ja tyyliä jolla ne on huomioitu MainPagella.

Kuva kuusi näyttää tilanteen, jossa käyttäjä yrittää aloittaa pelin ilman, että on täyttänyt mitään vaadittua kenttää. Sama tilanne ilmenee, mikäli joku kentistä on tyhjä.

KUVA 7.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

Choose opponent

Error
you must select an opponent

Kuvassa seitsemän on syötetty kaikki tiedot niille osoitettuihin kenttiin oikein, mutta käyttäjä yrittää käynnistää pelin ilman, että hän on valinnut vastustajaa.

KUVA 8.

Information page

Existing Player:

Enter first name:

Enter surname:

Enter birth year:

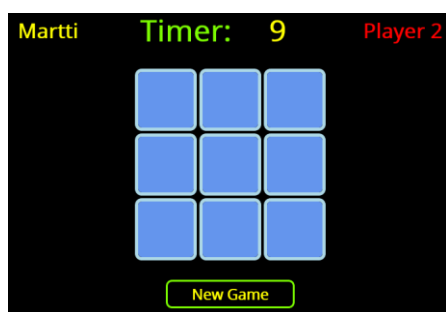
Choose opponent

Error
Give birth year in form of XXXX. Make sure you use only numbers.

Kuvassa kahdeksan käyttäjä on syöttänyt syntymä vuotta kysyvään kenttään jotain muuta kuin numerotietoa. Kyseinen kenttä on hyvin jäykkä, ja ottaa tiedon vastaan vain nelinumeroisena kokonaislukuna.

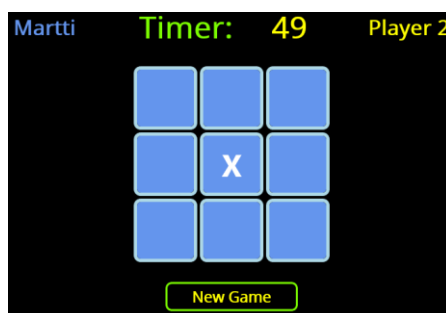
2.2 Start

KUVA 9.



Pelisivu kulkee ohjelmassa nimellä Start. Sivun aukeaa kuvassa yhdeksän näkyvään tilaan. Sivun auetessa Timer alkaa laskea pelin kestolle aikaa, joka pelin päätyttyä tallennetaan rekisteröityneen pelaajan tietoihin. Pelaajan vuoro indikoidaan nimen keltaisella värillä. Rekisteröitynyt pelaaja aloittaa pelin ja hänen merkkinsä pelissä on X. Tavoitteena on saada kolmen suora omia merkkejä vaaka-, pysty- tai viistosuunnassa.

KUVA 10.



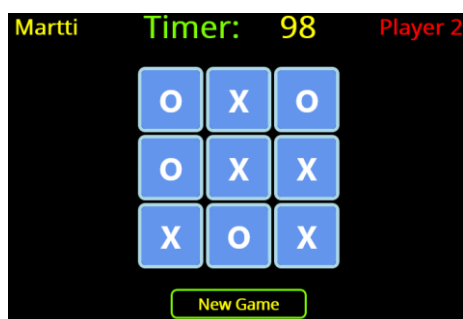
Kuvassa kymmenen pelaaja Martti on suorittanut ensimmäisen siirtonsa. Vuoro siirtyy hänen valitsemalleen vastustajalle, joka esimerkkitapauksessa on Player 2, eli toinen rekisteröimätön ihmispelaaja.

KUVA 11.



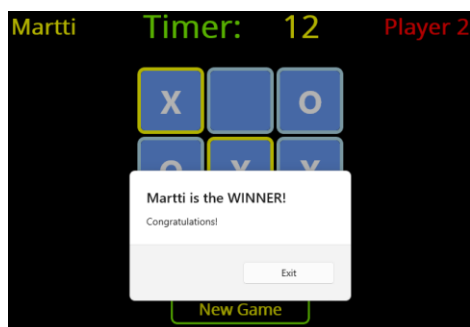
Tämä esimerkki päättyi tasapeliin, josta ohjelma antaa ilmoituksen. Start over-nappia painamalla saadaan suljettua ilmoitus.

KUVA 12.

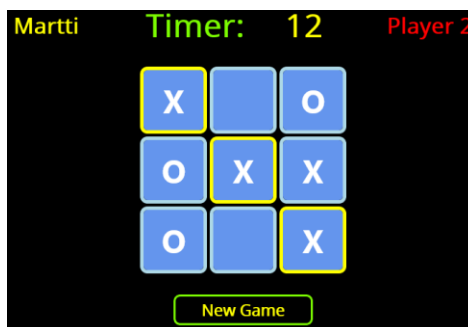


Ilmoituksen poistuttua, pystytään vielä tarkastella pelin lopputulosta. Pelin päätyttyä rekisteröityneen pelaajan tiedot päivitetään. Tässä tapauksessa Martin tietoihin lisättäisiin yksi tasapeli lisää, sekä kokonaispelaikaan 98 sekuntia. Seuraavaksi voitaisiin aloittaa uusi peli New Game-napista, tai navigoida yläpalkin nuolilla takaisin aloitus sivulle (nuolet eivät näy kuvassa).

KUVA 13.

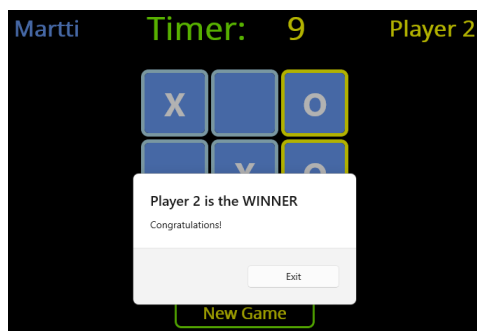


KUVA 14.



Kuvissa 13 ja 14 havainnoidaan vielä, kuinka voitto ilmoitetaan pelissä. Pelin päätyttyä ilmaantuu jälleen ilmoitus, joka kertoo voittajan. Exit-nappi sulkee ikkunan. Voittajan löytyessä, voittosuora ilmaistaan keltaisella värillä. Tässä esimerkissä Martin tietoihin lisättäisiin yksi voitto lisää, sekä 12 sekuntia lisää kokonaispelaikaan.

KUVA 15.



Mikäli voittaja ei olekaan käyttäjä, ohjelma toimii silti samalla tavalla, tällä kertaa ilmoittaen voittajan olevan Player 2 eli rekisteröimätön pelaaja. Tässä tapauksessa rekisteröityneen käyttäjän tietoihin lisätään yksi tappio ja tässä esimerkissä 9 sekuntia lisää kokonaispelaikaan. Jos käyttäjä olisi valinnut vastustajakseen tekoälyn, toimisi ohjelma vastaavalla tavalla, mutta tekoäly tekisi vastustajan siirrot omalla vuorollaan pienen miettimisajan jälkeen. AI ohjaa Player 2 tavoin aina O-merkkiä.

2.3 ScoreBoard

KUVA 16.

 A screenshot of the 'Information page' for the game. It has a dark background with yellow and red text. The form includes:

- 'Existing Player:' followed by a dropdown menu.
- 'Enter first name:' followed by a text input field.
- 'Enter surname:' followed by a text input field.
- 'Enter birth year:' followed by a text input field with 'xxxx' as a placeholder.
- 'Choose opponent' with two buttons: 'AI' (red) and 'Player 2' (green).
- A yellow button labeled 'START GAME'.
- A yellow arrow pointing to a yellow button labeled '!!!Scoreboard!!!'.

MainPagelta eli ohjelman aloitussivulta löytyy myös Scoreboard-nappi. Nappia voi painaa milloin tahansa käyttäjän ollessa aloitussivulla. Tämän napin kautta käyttäjä pääsee ScoreBoard-sivulle, jossa hän pystyy tarkastelemaan rekisteröityneiden pelaajien tilastoja, eli tietoja, joita kullekin pelaajalle on tallennettu.

KUVA 17.



The screenshot shows a window titled "Score board" with a standard Windows-style title bar. Inside the window is a table with the following data:

Firstname	Surname	Birthyear	W	L	I	Time played s.
Kyösti	Kallio	1873	3	1	0	35
Risto	Ryti	1889	1	3	2	35
Urho	Kekkonen	1900	4	2	1	64
Mauno	Koivisto	1923	2	3	1	130
Martti	Ahtisaari	1937	3	3	1	263

Kuvassa 17 näkyy kooste, joka on otettu ScoreBoard-sivulta. Tältä sivulta ei löydy muuta toiminnallisuutta, kuin tulosten esittäminen. Sivulla näytetään rekisteröityneen käyttäjän tiedot, jotka on poimittu ensimmäisen kerran talteen aloitussivulla, sekä tilastot, jotka tälle pelaajalle päivittyvät jokaisen pelatun pelin jälkeen.

3 TÄRKEIMMÄT SOVELLUKSEEN LIITTYVÄT TIEDOSTOT

3.1 MainPage.xaml ja MainPage.xaml.cs

Ohjelman aloitussivun ulkoasu on luotu MainPage.xaml tiedostoon. Tiedoston koodissa sivun rakenne luodaan grid-elementin sisään. Tällä tavoin label-tekstit, entry-kentät, button-komponentit, sekä picker-komponentti saadaan järjestettyä keskenään samaan linjaan. Tästä tiedostosta käsin asetetaan myös sivulla käytettävät oletusvärit, niin taustan kuin kaikkien muidenkin elementtien osalta.

MainPage.xaml.cs toimii edellä mainitun tiedoston parina. Tässä tiedostossa kirjoitetaan taustakoodi, joka linkittyy xaml-tiedostossa luotuihin rakenteisiin. Tämä tiedosto luo kyseiselle sivulle sen varsinaisen toiminnallisuuden, kuten esimerkiksi tapahtumat, jotka seuraavat napin painalluksia. Tältä sivulta kutsutaan myös funktioita, joiden avulla tietoa siirretään eri sivujen välillä, unohtamatta tietojen tallennusta, joka on myös ohjelman yksi keskeisimmistä toiminnoista.

3.2 Start.xaml ja Start.xaml.cs

Start.xaml oli ensimmäinen osa, jota lähdin ohjelmaa rakentaessa hahmottelemaan. Tässä tiedostossa luodaan xaml-koodin avulla varsinaisen pelisivun ulkoasu. Samaan tyyliin, kuin aiemmin kuvasin, myös tämä sivu rakennettiin grid-elementin sisälle. Sivulla on pelkkiä label ja button komponentteja. Label-elementtejä ovat kaikki teksti joka sivulla näkyy, ja pelialusta on luotu button-elementteistä. Näiden lisäksi löytyy myös nappula, josta pelin voi aloittaa uudestaan.

Start.xaml.cs tiedosto on koodi määrältään suurin tiedosto, joka pitää sisällään itse pelin mekaniikan ja paljon muuta. Tiedosto toimii Start.xamlin parina, ja tapahtumien kautta muokkaa tilanteen mukaan näkyvää sivua vastaamaan kulloistakin tilannetta, joka pelissä syntyy. Sivulla muun muassa tallennetaan tietoa, rakennetaan tekoälyn sekä pelin toimintalogiikka, hallinnoidaan ajastimen toimintaa sekä tallennetaan tietoa.

3.3 ScoreBoard.xaml ja ScoreBoard.xaml.cs

ScoreBoard.xaml tiedostossa luodaan jälleen ulkoasu käyttäjälle näkyvälle sivulle. Tällä kertaa se on rakennettu kahden erillisen grid-elementin sisälle. Ensimmäiseen gridiin syötetään vain yhteen riviin tulostaulun otsikot. Tämän jälkeen toinen grid asemoidaan siten, että halutut tiedot näkyvät kunkin otsikon alapuolella. Tällä tavalla saadaan aikaan helppolukuinen taulukkomainen näkymä, jota korostetaan pelin teemaväreillä.

ScoreBoard.xamlin parina toimii ScoreBoard.xaml.cs tiedosto. Tiedoston tehtävänä on tuoda sivulle lista tietueita, joissa rekisteröityneiden pelaajien tiedot ovat tallessa. Nämä tiedot esitetään gridissä, joka on rakennettu xaml-koodissa otsikko-gridin alapuolelle.

3.4 InfoStruct.cs

InfoStruct.cs-tiedostossa luodaan julkinen InfoStruct niminen tietue, johon tallennetaan käyttäjän syöttämiä tietoja, kuin myös pelin rekisteröimiä tietoja. Käyttäjä itse syöttää etu- ja sukunimen, sekä syntymä vuoden, jotka tallennetaan tietueeseen. Näiden tietojen lisäksi rekisteröityneen pelaajan tietueeseen tallennetaan aina pelien yhteydessä niiden tulos eli voitto, häviö tai tasapeli, sekä aika joka peliin on kokonaisuudessaan käytetty.

3.5 Players.json

Kuten aiemmin kuvattu ohjelma tallentaa tietoa rekisteröityneistä pelaajista. Tietoja tallennetaan väli muistiin kahdella eri sivulla. Tallentamisen lisäksi tietoja myös haetaan players.json tiedostosta. Mikäli tallennetut tiedot haluaisi poistaa, tulisi käyttäjän paikantaa koneeltaan kyseinen tiedosto ja manuaalisesti poistaa siellä olevat tiedot.

4 ITSEARVIO JA ARVIO TYÖHÖN KÄYTETYSTÄ AJASTA

Omat valmiuteni kurssille olivat rakentuneet täysin ohjelmointi I kurssin pohjalle. Kurssin aihepiiri oli erittäin kiinnostava, ja käyttöliittymäohjelmointi tuntui heti alusta alkaen mielekkäältä eikä vähiten siksi, että monessa tilanteessa tehdyt muutokset ohjelmassa näkyivät ja tuntuivat konkreettisesti. Kurssin kanssa oli itselläni ongelmia ajankäytön kanssa, sillä pättitilanteet tuntuivat pysäyttävämiltä, kuin esimerkiksi ohjelmointi I kurssilla. Tiedon etsimiseen kului todella paljon aikaa, ja aina sekään ei tuonut tulosta. Jos olisi ollut enemmän aikaa budjetoida tälle kurssille olisin halunnut tehdä kaikki tehtävät, mutta jouduin tekemään epämiellyttävän päätöksen ja panostaa enemmän aikaani harjoitustyöhön. Mikäli olisin saanut tehtyä kaikki tehtävät ennen harjoitustyön aloittamista, olisi se varmasti avittanut myös sen työstössä.

Tehtävät, joita tein kurssin aikana antoivat hyvää pohjaa lähteä työstämään harjoitustyötä. Omista lähtökohdistani tulevalle opiskelijalle olisi kuitenkin ollut hyötyä, jos olisi saanut vielä enemmän konkreettisia esimerkkejä. Etenkin xaml-puolen asioista oli välillä haastava löytää hyviä esimerkkejä, joiden pohjalta olisi voinut alkaa itse rakentamaan tiettyjä toimintoja. Kehityin kurssin aikana tiedonhaussa ja Microsoftin omien ohjeiden läpikäymisessä. Koen, että tarvitsen silti lisää harjoitusta etenkin Microsoftin omien ohjeiden tulkitsemisessä ja hyödyntämisessä. Xaml oli itselleni melko selkeää kirjoittaa. Pidän ulkoasujen rakentamisesta, ja oli mielekästä päästä luomaan tapahtumia eri komponenteille. Koin oppivani paljon juuri tapahtumien luomisesta.

Lopputulokseen olen itse erittäin tyytyväinen. En olisi uskonut, että saan näinkin hienon sovelluksen aikaiseksi. Aluksi lähdin tähtäämään siihen, että saan rakennettua toimivan pelin. Työn edetessä itselleni valkeni, ettei peliosuuden tekeminen ollutkaan haastavin osuus ohjelmassa. Selkeästi eniten ongelmia tuli tallennuksen rakentamisen kanssa. Etenkin näihin ongelmiin hain apuja, ja sainkin ne loppuen lopuksi parsittua kasaan.

Ohjelmaan jäi vielä useita kohtia, joita olisin halunnut viilata. Esimerkiksi vaikeusasteen luominen tekoälylle olisi ollut hauska vielä rakentaa. Ajatuksena saman kaltaisella toteutuksella, kuin voittojen tarkastus. Olisin tehnyt toiminnon, joka tarkastaa myös kahden suorat, ja mikäli niitä löytyy, olisi tekoäly asettanut merkkinsä estämään kolmen suoran synnyn jollakin tietyllä todennäköisyydellä, riippuen vaikeusasteesta. Myös asetteluun saatan vielä tehdä tämänkin kirjoittamisen jälkeen muutoksia, mikäli aikaa jää.

Olen arvioinut ajankäyttöäni läpi projektin, aluksi pidin ihan kirjaa tunteista, sillä tuntui, että käytin kaiken opiskeluaikani nimenomaan ohjelmointiin. Suunnitteluvaiheelle arvioin noin 6h, tähän sisältyi visuaalisen ilmeen hahmottelua paperille, sekä toiminnallisuuksien toteuttamisen suunnittelua. Etusivun xaml, sekä pelisivun xaml ilman varsinaisia toimintoja otti aikaa noin 8h. Picker-komponentin kanssa painimiseen menikin jo paljon enemmän aikaa, samoin tallennusten kanssa. Etenkin tallennukset teettivät töitä, tein ne alun perin temp-kansioon, mutta oppitunnilla käytyjen asioiden tähden lähdin sitä korjaamaan. Tallennusten tekemiseen meni kaiken kaikkiaan arviolta useampi täysi opiskelupäivä. Sivujen välistä tiedonkulkua tuli myös tehtyä ja suunniteltua pitkään. Loppuarvio kokonaisuudesta, jonka käytin ohjelman tekemiseen, raportointiin ja kommentointiin on noin 80h.