

Muistipeli

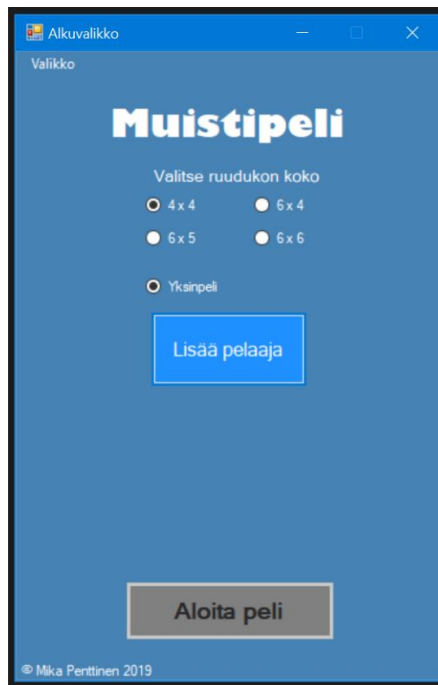
Ohjelmointi II, Käyttöliittymäohjelmointi

25.11.2019

Yleiskuvaus

Valitsin harjoitustyöaiheista muistipelin, koska se kuulosti mielenkiintoisimmalta ja haastavimmalta vaihtoehdolta. Peliä voi pelata yksin- tai kaksinpelinä. Yksinpelin päättyessä ilmoitetaan, kuinka monella klikkauksella pelaaja löysi kaikki parit. Kaksinpelin lopussa ilmoitetaan kunkin pelaajan pisteet (1 piste = 1 löydetty pari) ja tämän jälkeen voitot sekä tappiot kirjoitetaan ylös .XML-tiedostoon. Tasapelejä ei kirjata ylös. Pelitilastoja voi tarkastella alkuvalikon menusta. Pelialustavaihtoehtoja on 4 kpl, joista pienin on 4 x 4 (8 paria) ja suurin puolestaan 6 x 6 (18 paria).

Käyttöohje

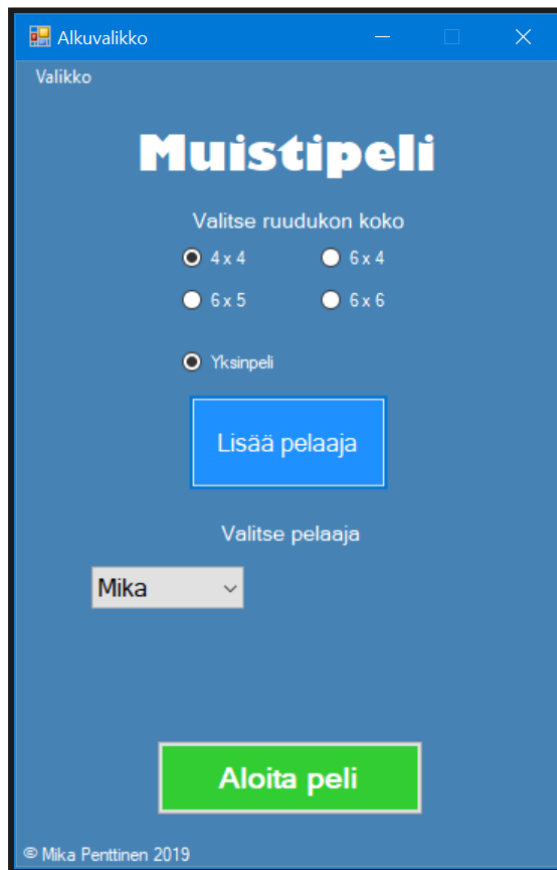


Kuva 1. Alkunäkymä

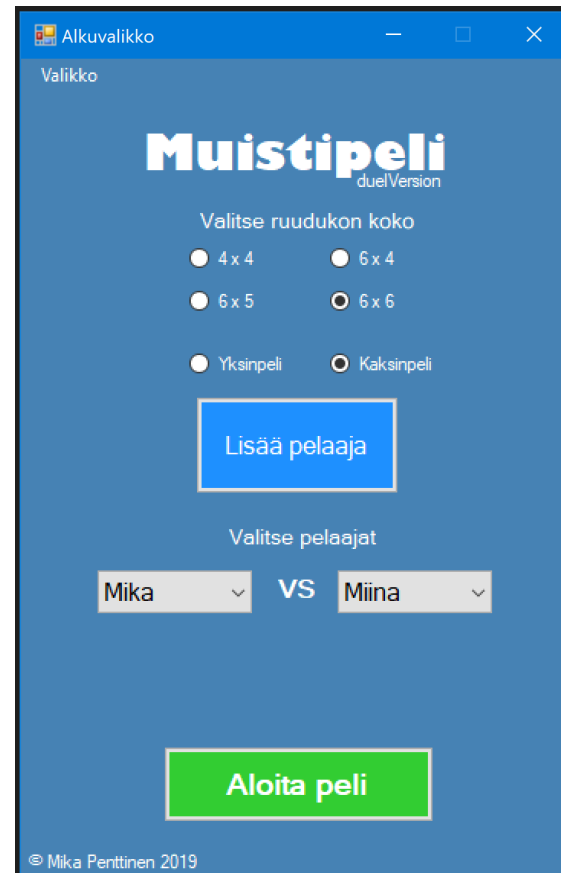


Kuva 2. Pelaajan lisäys

Kuvassa 1. on alunäkymä, kun ohjelma ensimmäisen kerran avataan. Pelaajia ei ole vielä lisätty, joten ”Aloita peli” -painike on inaktiivinen, sekä kaksinpelivaihtoehto on piilossa. ”Lisää pelaaja” -painike avaa kuvan 2. näkymän, johon syötetään pelaajan nimi, jonka jälkeen ”Lisää” -painikkeella lisätään pelaaja listaan. Saman nimistä pelaajaa ei voi lisätä listalle kahdesti (tästä ilmoitetaan popupilla).



Kuva 3. Yksi pelaaja lisätty

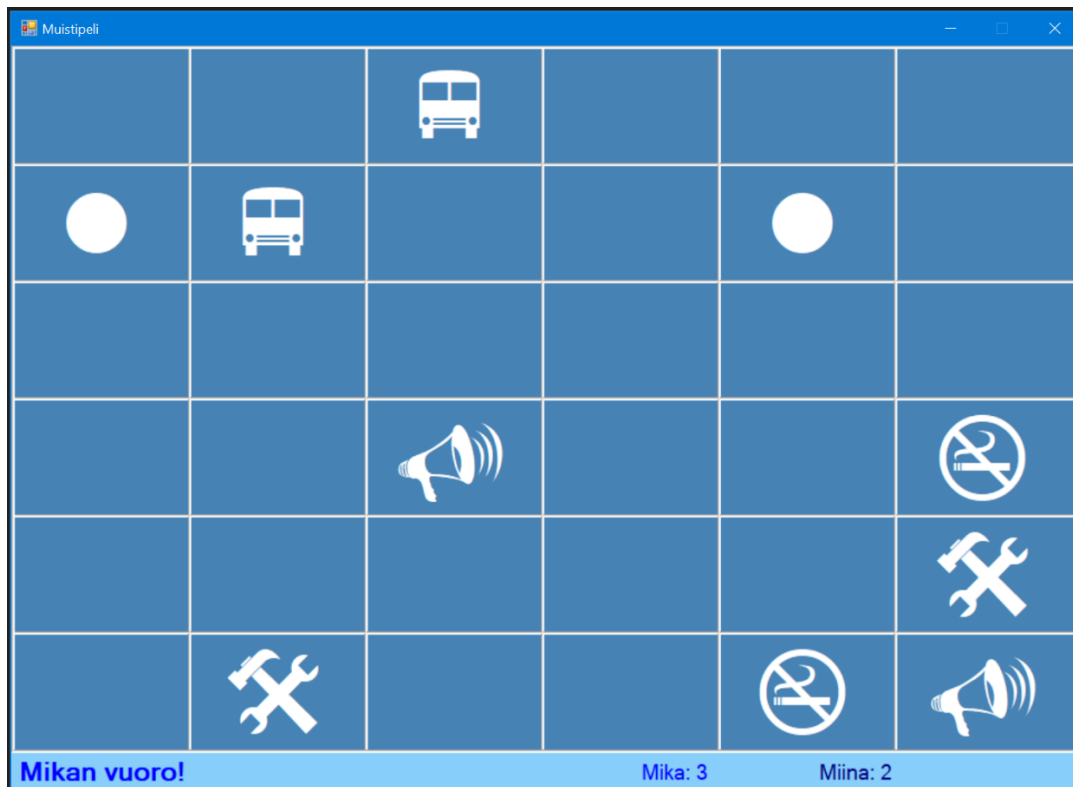


Kuva 4. Kaksi pelaajaa lisätty

Kuvassa 3. on lisätty yksi pelaaja listaan ja "Aloita peli" -painike on aktivoitunut. Peliä voi nyt pelata yksinpelinä, jolloin pelin päättymisen jälkeen ilmoitetaan kuinka monella klikkauksella pelaaja ratkaisi pelin. Näitä klikkauskmääriä ei kirjoiteta tiedostoon, toisin kuin kaksinpelin voitto/häviötilastot.

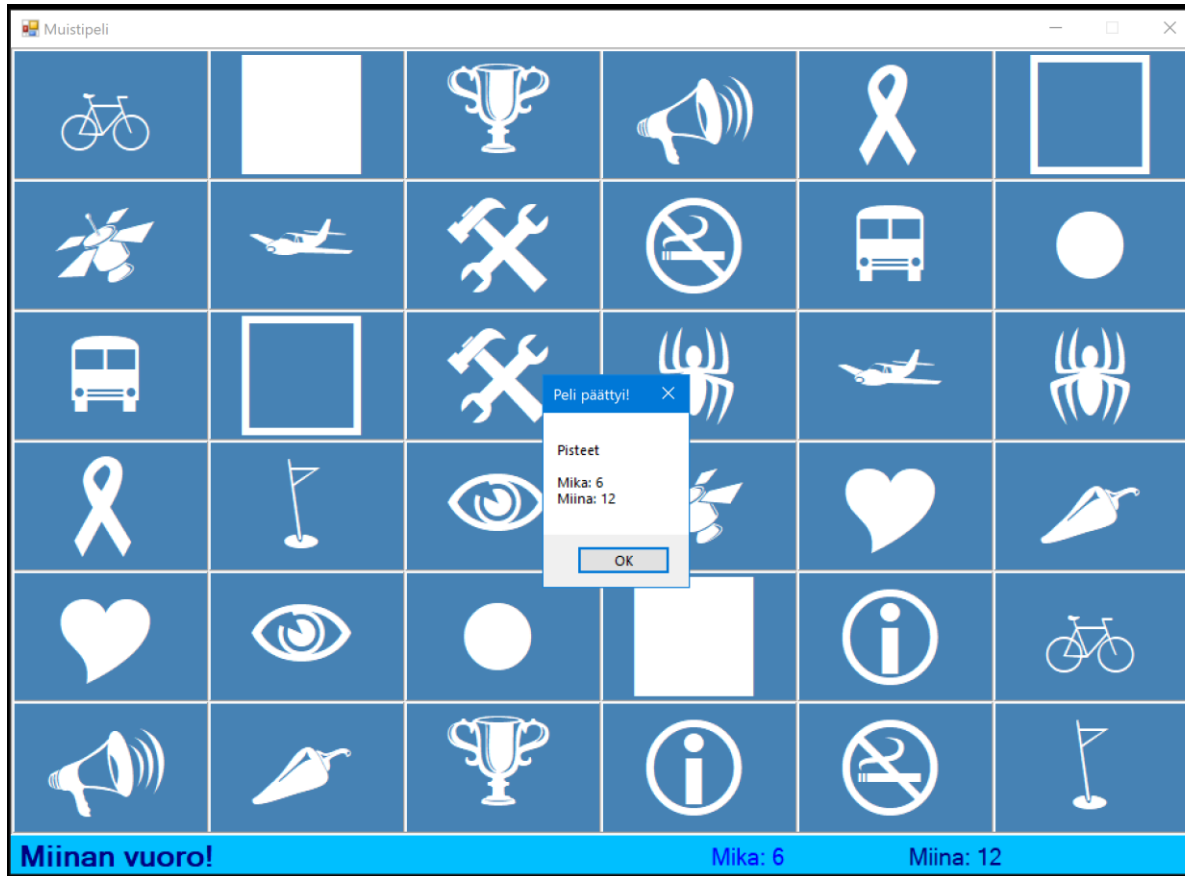
Kuvassa 4. kaksi pelaajaa on lisätty listaan, ja täten kaksinpelivaihtoehto on tullut esille. Tämän jälkeen jos valitsee yksinpelin, piilotetaan "VS"-label ja toisen pelaajan Combobox näkymästä pois.

Jos käyttäjä yrittää aloittaa pelin niin, että pelaaja 1. ja pelaaja 2. ovat sama pelaaja, "Aloita peli" -painike inaktivoituu. Täten estetään pelaamasta kaksinpeliä yhdellä ja samalla pelaajalla.



Kuva 5. Kaksinpeli käynnissä

Kuvassa 5. on näkymä kaksinpelitilanteesta 6 x 6 pelialusta valittuna. Alapalkissa näkyy kenen pelaajan vuoro on menossa, sekä kummankin pelaajan sen hetkiset pistemäärät. Alapalkin vuoroteksti sekä alapalkin sävy muuttuu kunkin pelaajan vuoron mukaan.

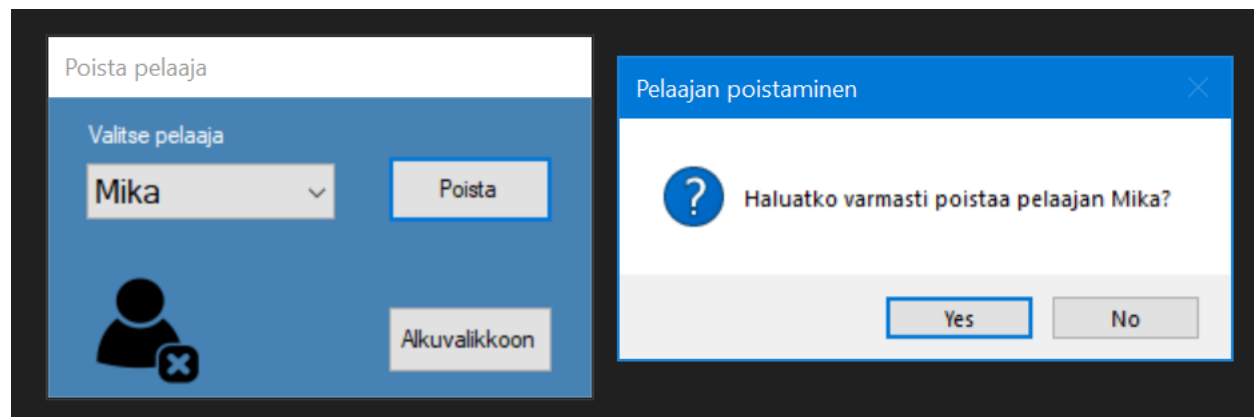


Kuva 6. Kaikki parit löydetty, kaksinpeli päättynyt

Kuvassa 6. kaikki parit on löydetty ja kaksinpeli on päättynyt. Peli ilmoittaa kunkin pelaajan pistemäärät vielä erillisessä messageboxissa. Tässä vaiheessa voitot ja häviöt kirjataan ylös pelaajalistaan ja täten myös .XML-tiedostoon.



Kuva 7. Valikko → Pelaajatilastot

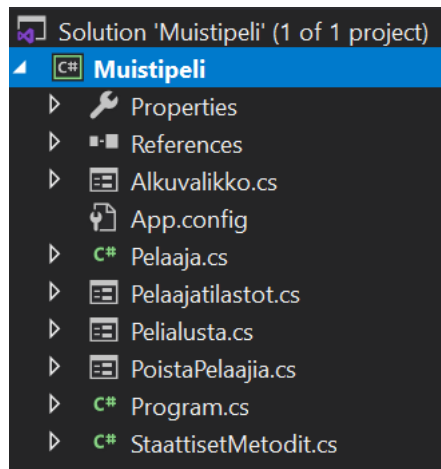


Kuva 8. Poista pelaaja -ikkuna sekä poistamisen varmistaminen

Pelin alunäkymän Valikko-menusta löytyy ”Pelaajatilastot”, ”Poista pelaajia” sekä ”Poistu” vaihtoehdot. Kuvassa 7. on pelaajatilastot auki alunäkymän päällä. Tilastot luetaan .XML-tiedostosta ja ne generoidaan kyseiseen ikkunaan ohjelmallisesti, jolloin ikkunaan saadaan aina kaikki tiedoston tiedot listattuna. Kuvassa 8. on avattuna ”Poista pelaajia” vaihtoehto. Tämän ikkunan kautta voidaan poistaa pelaajia listalta. Poista napin painaminen avaa vielä varmistusikkunan, jonka jälkeen pelaaja poistetaan listalta, sekä .XML-tiedosto uudelleenkirjoitetaan.

”Alkuvalikkoon” painikkeesta palataan pelin alkunäkymään. Jos käyttäjä poistaa pelaajalistan ainoan pelaajan, ”Poista pelaaja” -ikkuna sulkeutuu ja palataan automaattisesti pelin alkuvalikkoon. Täten estetään listan nolaindeksin poistamisyritys ja ohjelman kaatuminen.

Tiedostot



Alkuvalikko.cs: Tämä aukeaa, kun peli avataan. Alkuvalikosta käyttäjä valitsee pelikentän koon, pelimoodin ja pelaajat. Alkuvalikossa on menu, jonka kautta pääsee pelaajatilastoihin, poistamaan pelaajia sekä poistumaan pelistä.

Pelaaja.cs: Tehtävänannosta (structit) poiketen päädyin käyttämään Pelaaja-luokkaa, jota käytetään pelaajia lisättäessä ja pelaajatilastoja ylläpidettäessä. Sisältää muuttujat: nimi, voitot, häviöt, sekä konstruktorit ja getterit/setterit.

Pelaajatilastot.cs: Alkunäkymän valikon kautta avautuva pelaajatilastonäkymä. Labelit generoidaan ohjelmallisesti, joten tieto mahtuu aina ikkunaan, vaikka pelaajalista olisi todella suuri (scrollbar ilmestyy ikkunaan tietyn pelaajamäärän jälkeen).

Pelialusta.cs: Pelin alusta ja itse pelin toiminnallisuus. Pelialustan generointi valitun koon mukaan, vuorojenvaihtuminen, klikkausten ja kulloisenkin vuoron seuranta.

Poistapelaajia.cs: Alkunäkymän valikon kautta avautuva ikkuna, jossa voidaan poistaa pelaajia pelaajalistalta.

Program.cs: Tästä tiedostosta löytyy ohjelman Main() -metodi, jossa Application.Run(new Alkuvalikko()); käynnistää itse pelin alkuvalikon. Tämä ajetaan aina ensimmäiseksi, kun peli käynnistetään.

StaattisetMetodit.cs: Keräsin tähän tiedostoon staattisia metodeja muista tiedostoista, jotta koodin lukeminen olisi helpompaa. Tässä tiedostossa on myös .XML-tiedoston lukemiseen ja kirjoittamiseen tarvittavat metodit.

Itsearviointi

Aikaa tähän projektiin kului yhteensä n. 25 tuntia, josta noin puolet oli tiedonhakua ja loput itse ohjelman rakentamista. Matkan varrella tuli haettua tietoa todella paljon eri toiminnoista ja tapahtumista. Tutuksi tuli stackoverflow.com ja lukuisat muut internetin tietolähteet. Täytyy todeta, että lausahdus ”ohjelmoimaan oppii vain itse ohjelmoimalla” on enemmän kuin totta.

Projekti oli mielenkiintoinen ja haastava, koska halusin lisätä peliin kaikki tarpeelliset ominaisuudet. Alkuvalikon ulkoasuun kului paljon aikaa, koska halusin tehdä siitä ”single page app” -tyylisen näkymän, eli toimintoja piilotetaan/näytetään yhdessä ikkunassa sitä mukaa, mitä käyttäjä on tekemässä. Tämä tekee alkunäkymästä miellyttävän ja selkeän käyttää, koska uusia ikkunoita/popuppeja ei aukea eri toimintoja suoritettaessa.

Comboboxien ja pelaajalistan yhteentoimivuus aiheutti päänvaivaa ja siihen kuluikin paljon aikaa. Se ei edelleenkään toimi ehkä täydellisesti, mutta mielestäni riittävän hyvin ja selkeästi. Try-catch lohkoja käytin metodeissa joissa luettiin tai kirjoitettiin tiedostosta/tiedostoon. Muut toiminnot sain rakennettua siten, että käyttäjä ei voi tehdä esim. virheellisiä tietojen syöttämisä. Tämän sain aikaan toimintojen rajoittamisella eri tilanteissa: esim. ”Poista pelaaja” -ikkunasta poistutaan heti, kun pelaajalistan viimeinen olio on poistettu, jolloin tyhjän listan nollaindeksin poistamista ei pääse tapahtumaan edes vahingossa. Testailin ohjelmaa aina kun lisäsin uusia ominaisuuksia, ja suurempia bugeja ei projektissa pitäisi enää olla. Jos kuitenkin sellaisia löytyy, otan mielelläni tiedon vastaan ja yritän ratkaista kyseisen ongelman!

Projektia tehdessä opin paljon uusia asioita, joista päällimmäisinä jäivät mieleen kontrollien ja elementtien ohjelmallinen luominen. Pelialusta generoidaan ohjelmallisesti sen mukaan, minkä kokoisen pelin käyttäjä valitsee. Tämän vuoksi jokaiselle pelialustakoolle ei tarvitse tehdä omaa formia, vaan silmukoiden avulla saadaan generoitua pelaajan valitsema pelialustakoko. Pelaajatilastot-ikkunan tiedot näytetään myös silmukan avulla, jolloin ikkunaan ei tarvitse käsin määritellä labeleita jne.

Projektia voisi laajentaa lisäämällä esim. klikkaustilastot yksinpeleille. Tilasto ilmoittaisi kunkin pelaajan ennätyksen (alhaisin klikkausmäärä) kullakin pelialustakoolla. Tämä onnistuisi helposti Pelaaja-luokkaa laajentamalla, eli tekisi pelaajalle yhden uuden muuttujan per pelialustan koko, johon tallentaisi alhaisimman klikkausmäärän minkä pelaaja on saavuttanut. Keskityin tässä projektissa kuitenkin enemmän kaksinpeliin ja sen tilastointiin, joten yksinpelin laajentaminen on tulevaisuuden hommia. Kipinä ohjelmointiin kasvoi koko tämän projektin myötä ja C# -kieli on alkanut selkeytymään päivä päivältä. Paljon on vielä opittavaa, mutta alku on ollut miellyttävää ja erityisesti haastavaa tekemistä. Haasteiden voittaminen ja pitkän vääntämisen jälkeen toimiva koodi on ohjelmoinnin suola!