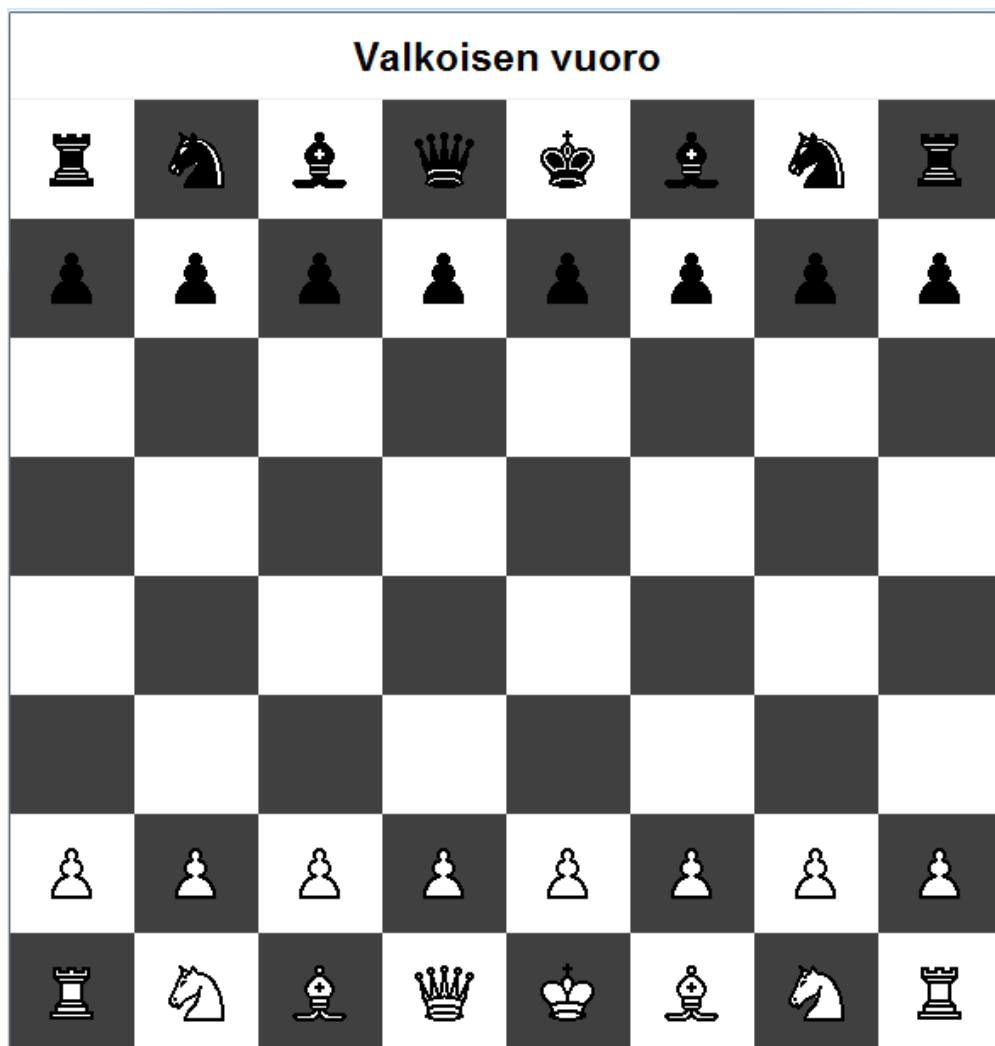


PAAVO LYYTIKÄINEN

Kaksinpelattava shakki



Käyttöliittymät

Ohjelma on ensisijaisesti suunniteltu toimimaan graafisen käyttöliittymän kautta, mutta myös tekstikäyttöliittymää voi tarvittaessa käyttää. Tekstikäyttöliittymässä ei erota selvästi nappuloiden reittejä, koska mustat ja valkoiset ruudut näkyvät samanlaisina viivoina. Lähinnä tekstikäyttöliittymän tarkoitus olikin toimia suunnittelun apuna eikä sen pelattavuuteen tämän vuoksi käytetty enempää vaivaa. Itse pidin erittäin tärkeänä, että saisin pelistäni nimenomaan ulkoisesti shakin näköisen.

Molemmat käyttöliittymät saavat parametrina Pelilauta-luokan, joka alustetaan. Käyttöliittymä käynnistetään mainissa, jossa kutsutaan myös run-metodia, joka varsinaisesti laittaa käyttöliittymän pyörimään.

Pelilauta

Pelilaudan alustuksessa luodaan pelilaudalle nappulat NappulatAlkutilanteessa-luokan kautta. Jos haluaa pelata eri shakkivariaatioita, kuten linnoitusshakkia tai kulmashakkia, onnistuu se kätevästi muokkaamalla nappuloiden koordinaatteja NappulatAlkutilanteessa-luokassa. Pelilaudassa on toteutettu lähes koko sovelluslogiikkaosio. Pelilauta tuntee nappuloiden sijainnit ja eri pelitilanteet. Tärkein metodi koko pelissä on liikutaNappulaa, joka pirstottunakin näyttää hurjalta. Metodin toiminnallisuutta on pyritty kuvaamaan javadocissa mahdollisimman tarkasti. Tähänkin metodiin saattaa minulle vielä joskus löytyä keinoja ilmaista asiat helpommin.

Nappula

Nappula-luokka on abstrakti, jolla on koordinaatit. Eri nappulat perivät Nappulan ja toteuttavat omalla tavallaan voikoLiikkua-metodin ja kulkureitin tallentamisen. VoikoLiikkua ei ota yhtään kantaa siihen mikä tilanne on pelilaudalla, koska nappulat eivät tunne pelilautaa. Metodi ainoastaan selventää jokaiselle nappulalle miten ne kykenevät oikeaoppisesti liikkumaan.

Suurimmat poikkeuksellisuuden nappuloissa on kuninkaassa ja sotilaassa. Sotilas on siitä erityinen, että se pystyy syömään kulmittain mutta liikkumaan ainoastaan eteenpäin suoraa linjaa. Kuningas taas ei ole syötävissä oleva nappula, jota itse nappula ei tosin tiedä. Pelilauta määrittelee sen, ettei kuningasta voi ikinä syödä.

Reitin taulukkoon tallennus näyttää erityisesti kuningattaressa aika sotkuiselta. En kuitenkaan osannut tätä pilkkoa eri osiin ilman, että se olisi menettänyt loogisen kokonaisuuden, joten päädyin pitämään yhden hieman hankalamman näköisen kokonaisuuden rakenteessa. Kenties tulevaisuudessa tämäkin metodi pienenee muutama riviin.

Ohjelman toiminta

Shakki toimii shakkisääntöjen mukaisesti siten, että valkoinen aloittaa joka kerta. Ensimmäinen hiiren klikkaus valitsee siirrettävän nappulan ja toinen kohdan, johon nappula halutaan siirtää. Shakkitarkistus

vaatii sen, että vuoron jälkeen omalle kuninkaalle ei aiheudu shakkia, joten se ei hyväksy kuin siirrot, jotka ohjaavat pois shakista. Peli voi päättyä joko matti- tai patti-tilanteeseen, jolloin toisella pelaajista ei ole enää siirtoja jäljellä. Mikäli molemmilla pelaajilla on jäljellä enää kuninkaat tai kuninkaan lisäksi ainoastaan yksi hevonen tai lähetti, ei kumpikaan pelaajista kykene enää voittamaan peliä paitsi äärimmäisen harvinaisessa tilanteessa, jossa on itse ajanut kuninkaansa nurkkaan.

Pelinappuloita voi kuitenkin siirtää ikuisesti eikä peli ilmoita missään vaiheessa olevansa loppunut, vaan se on käyttäjän vastuulla. Käyttäjinä toimii kaksi ihmispelaajaa, joten ihmisten pelinhahmotuskykyyn on luotettu. Amatööripelaajille saattaa tulla tilanteita, että eivät tiedä mitä ovat tehneet tai miksi jotain ei voi tehdä, mutta vähän enemmän pelanneilla ei ole mitään ongelmia tämän suhteen.

Ohestalyöntiä ei ole käsitelty tässä versiossa, vaan päätin jättää sen toteutuksen myöhemmäksi, jotta nykyiset toiminnallisuudet toimisivat oikein, eikä olisi liikaa poikkeustilanteita sekoittamassa peliä. Tornituksenkin toteutus jäi puuttumaan tästä versiosta, vaikka sen toteutuksesta minulla onkin suhteellisen selvät sävelet. Eli tarvitsee ainoastaan tallentaa tornien ja kuninkaiden sijainnit Pelilaudassa oliomuuttujiin sekä laatia jokaiselle onkoLiikkunut-metodi, joka tarkistaa, onko kyseinen nappula liikkunut. Näin ollen tilanteissa, joissa yritetään siirtää kuningasta kaksiaskelta sivulle, tarkistetaan metodi voikoTornittaa. Ja jos voi, niin suoritetaan tornitus.

Omat kokemukset

Täytyy sanoa, että projekti oli kyllä erittäin opettavainen. Joutui pohtimaan alusta lähtien, miten lähtisi toteuttamaan ohjelmaansa. Paljon tuli mutkia matkalle, mutta aina se palkitsi yhtä paljon, kun sai ongelman ratkaistua. Suurimmat ongelmat kohtasin kuljettavan reitin tarkistuksessa sekä shakkitarkistuksessa. Nappuloista sotilas oli kaikkein viheliäin, koska se syö eri tavalla kuin liikkuu. Tämä aiheutti paljon harmia, kun pitkään peli oli siinä vaiheessa, että kuningas ei päässyt siirtymään suoraan sotilaan eteen. Projektin viimeinen viikko oli itselleni kaikkein tuottoisin ja sain tehtyä hienomman pelin kuin olisin itse voinut kuvitella.

Täytyy siis sanoa, että ilman ohitusta ja tornitustakin olen ylittänyt huomattavasti omat odotukseni. Ohjaus kurssin aikana oli erittäin hyvää, joten siitä iso kiitos. Välillä tietenkin tuskastutti, kun kotona tehdyt kyhäelmät eivät enää yliopistolla vaikuttaneetkaan yhtä fiksulta. Kun nyt olen saanut toimivan shakin tehtyä, niin totta kai viimeistelen vielä ohestalyönnin ja tornituksenkin jossain vaiheessa.