

# Toteutusdokumentti

Ohjelman rakenne on melko suoraviivainen. Main-luokka käynnistää tekstikäyttöliittymäluokan, joka käynnistää pelilogiikkaluokan, joka käynnistää rekursiivisen metodin siirtojen tekemiseen. Siirtoja tekevä metodi kutsuu itseään yhä uudelleen, kunnes peli on päättynyt. Siinä välissä se kutsuu myös toista rekursiivista metodia, joka arvioi parhaan mahdollisen siirron (tekoälyn seuraavaa siirtoa varten) minmax-algoritmilla ja aina tarvittaessa simppeliä tarkistusmetodia, joka tarkistaa, onko peli jo voitettu annetulla peliruudukolla.

Siirtoja arvioiva minmax-algoritmi on pelissä ainoa merkityksellinen tekijä aika- ja tilavaativuuden suhteen. Tilavaativuus on aina  $O(n)$  ( $n$  = ruudukon ruutujen määrä), koska rekursiopino ei koskaan ehdi kasvamaan pidemmäksi kuin ruudukon ruutujen määrä. Aikavaativuus on pahimmassa tapauksessa suunnilleen välillä  $O(n^5)$  ja  $O(n^6)$  - päätellen 3x3-ruudukon mahdollisista pelitilanteista - mikä on vähemmän kuin aluksi ajattelin. Se johtuu siitä, että mahdollisia pelitilanteita karsiutuu suuri määrä pelin loppuessa toisen pelaajan voittoon. Tällainen aikavaativuus toteutuu kuitenkin vain 3x3-ruudukossa ja siinäkin vain osassa tilanteita, kun taas 5x5- ja 7x7-ruudukoissa aikavaativuus on huomattavasti pienempi pääasiassa siirtojen tutkimisen rajoitteiden ansiosta, jotta peli etenisi järjellisessä tahdissa.

Ohjelma alkaa nyt olla aika lailla valmiin oloinen. Siinä on toki vielä hiomista ja tehostamista, mutta pääpiirteittäin kaikki tarvittava toimii. Kenties katson vielä alpha-beta pruning -menetelmää, jolla algoritmi tehostuisi merkittävästi, mutta jonka toteuttamisessa en ole tähän mennessä kunnolla onnistunut. Graafinen käyttöliittymä voisi mahdollisesti olla hyvä lisä, mutta en välttämättä enää ryhdy sellaisen tekemiseen.