Aihe: Vuoropohjainen strategiapeli, Tira-osuus

Toteutin tekoälyn pieneen strategiapeliin[1], viriteltyä min-max-hakua hyödyntäen. Lisäksi pelissä tarvitaan reitinhakualgoritminä Dijkstraa.

Käytännössä pelissä ei tarvita reitinhakua kuin lyhyelle matkalle ja lisäksi tarkoitus on nimen omaan saada etäisyys kaikkiin peliruutuihin tietyllä etäisyydellä, jotta pelaaja tai tekoäly tietää, mihin ruutuun siirtyä. Negatiivinen maaston vaikeus olisi myös hölmöä, joten Dijkstra oli hyvä valinta.

Min-maxia käytin, koska se on johdatus tekoälyyn-kurssin ensimmäinen ja helpoin(?) asia, joten pystyin jopa toteuttamaan sen. Se on kuitenkin hyvin raskas, joten arvelin, että on käytännössä pakko viritellä jotenkin tehokkaammaksi. AlfaBetakarsinta oli yksi vaihtoehto.

Tekoäly kuitenkin toimi varsin hyvin vain muutamia tai jopa vain yhden ruudun simuloimalla kiitos yllättävän toimivan arvolaskimen, joten minMaxin hiominen jäi tulevaisuuteen. (Ja aika ja energia loppuivat.)

Lisäksi toteutin pelin vaatimat tietorakenteet eli ArrayListiä vastaavan listarakenteen ja reitinhaun tarvitseman PriorityQueuen korvaavan minimikekorakenteen.

Kaikki algoritmit ja tietorakenteet toteutin pelin ehdoilla mitä syötteisiin tulee. Yritin kuitenkin tehdä niistä ainakin kohtalaisen tehokkaita, eli samaa *O*-luokkaa kuin niiden ‘kuuluukin’ olla, jo siksikin että voin sitten joskus laittaa mielettömiä tuhansien (miljoonien!) yksiköiden AI vs AI-mättöjä pyörimään (mwahaha), vaikka peli periaatteessa onkin tarkoitettu aika pienille yhteenotoille. Tämä siis jos ikinä saan oikeasti kevyen tekoälyn aikaan. Algoritmit ja tietorakenteet myös saavuttivat odotetut vaatimukset.

Lähteet: Kurssimateriaali kursseilta ohjelmoinnin perus- ja jatkokurssi, ohjelmistotekniikan menetelmät, tietorakenteet ja algoritmit, ohjelmistotuotanto, johdatus tekoälyyn. Wikipedia.

[1] Itse pelin määrittelydokumentti: kts. aiheenKuvausJaRakenne.pdf, https://github.com/ohinkkan/tira-labra/tree/master/dokumentointi