

## Reittihaku ja Minimikeko

Toteutettuani minimikeon tein nopean tehokkuusvertailun javan PriorityQueueeen. En tehnyt mitään teknisesti hienoa, ajoinpahan vain tuhansia reittihakuja eri kokoisilla, randomgeneroidulla kartoilla käyttäen eri tietorakenteita ja katsoin, kumpi on nopeampi. Alla hieno taulukko tuloksista.

Kartan koko	Toistoja	Keko	PriorityQueue	Kekoetu
3-5	100000	567ms	710ms	~25%
3-10	10000	112ms	147ms	~31%
10	10000	224ms	299ms	~33%
3-100	1000	1066ms	6566ms	~515%
100	1000	2993ms	27655ms	~824%
1000	1	1261ms	216292ms	~17052% (?!?)

Kuten huomaamme, keko on selkeästi tehokkaampi pienilläkin kartoilla, mutta suuremmilla se on aivan ylivoimainen. En tosin ole täysin varma, toimiiko algoritmi edes oikein jos kartat ovat kovin suuria. Ongelmia tulee viimeistään silloin, kun reittien pituus lähestyy miljardia ( $\infty$  on algoritmissani  $\text{max-int}/2$ ).

'Hupaisasti' algoritmin suhteellisesta tehokkuudesta on rajallisesti hyötyä, koska

1. tekoälyn hitaus ei liity reitinhakuun lainkaan ja joka tapauksessa
2. tehokkuudesta isoilla kartoilla (jos sellaisille nyt edes viitsisi pelata) ei ole mitään hyötyä, koska nopeusrajoitus rajaa tutkittavan osan karttaa neliöön, jonka sivujen pituudet ovat reittiä hakevan  $(\text{yksikön nopeus} * 2) + 1$  eli luokkaa 10.

## Tekoäly

Tekoälyn 8-versio (kuvastaa sen älyn kokoa) on toteutettu min-max-sovelluksena, mikä tarkoittaa että se on älyttömän raskas. Joka vuorolla joka yksiköllä on kymmeniä ellei jopa satoja eri tapoja toimia, ja tekoäly käy ne kaikki läpi jokaiselle yksikölleen. Ainakin

tämänhetkinen tekoäly toimii hyvänä demonstraationa eksponentiaalistien algoritmien ongelmallisuudesta...

Teknisesti algoritmi osaa laskea arvoja vaikka sadan vuoron päähän, mutta käytännössä enemmän kuin 3 on täysin käyttökelvoton vaikka yksiköjä olisi vain muutamia.

Tekoälyllä on neljä parametriä: aggressiivisuus, puollustavuus ja sattumanvaraisuus, jotka vaikuttavat sen tapaan arvioida eri toimintojen keskinäistä paremmuutta, sekä vuorot, joka vaikuttaa siihen, kuinka monta simuloitua kierrosta tekoäly pelaa kerrallaan.

Testasin tekoälyä pelauttamalla sitä eri parametreilla, vaihtelevilla yksikkömäärillä ja karttakoilla sekä toimintojen arvot määrittävää ValueLogic -luokkaa virittelemällä. AI se toimii kohtalaisen loogisesti, ainakin sitten kun lopulta huomasin että olin sotkenut new Tekoäly():ssä parametrien järjestyksen. (Se mitä luulin aggressiivisuudeksi olikin sattumanvaraisuus ja se mitä luulin sattumanvaraisuudeksi olikin defense.)