

# Viikkoraportti

## Viikko 3 (ja vähän aiempia)

Tällä viikolla aloitin javadocien täydentämisen ja perehtymisen Junitin yksikkötesteihin. Suuria muutoksia koodin runkoon en ole tehnyt. Joitain muutoksia ohjelmaan olen tehnyt sen testattavuuden vuoksi.

Jonkin verran käytin aikaa graafiseen puoleen, vaikka tiedän, ettei se ole oleellinen osa tätä harjoitustyötä. Se on kuitenkin yksi osa-alue, joka tekee ohjelman suorituksen etenemisestä paljon mielenkiintoisempaa, ”pelimäisempää”. Ja sen avulla ohjelmaa on ehkä helpompi demota, kuin näyttää debuggerilla tai välitulostuksilla. En aio kuitenkaan käyttää siihen hurjasti aikaa, joten siltä osin parhaat käytännöt saattavat jäädä tietutumatta.

Parin ensimmäisen viikon aikana sain työn hyvään alkuun. Tämä oli tärkeää oman mielenrauhan kannalta. Tuskailin mm. kohtuullisen paljon Githubin kanssa; pysrin käyttämään sitä suoraan NetBeansilla, mutta jotenkin sain systeemit solmuun. Nyt olen saanut toimivan ratkaisun käyttämällä Gitin työpöytäsovellusta ja NetBeansia yhdessä.

Viimeisen viikon, kahden aikana olen tietysti perehtynyt A\*-algoritmin toimintaan ja joutunut opettelemaan sen toiminnan kokonaan. Tämä on mm. herättänyt ajatuksia siihen liittyvän heuristisen arvion parantamiseksi.

Yleisesti pieniä vaikeuksia välillä tuottaa Javan käytännöt. Työhommissa olen koodannut 6 – 7 vuotta C#:lla, ja vain harvakseltaan kouluhommissa Javalla. Esim. jotkin hyvin triviaalit asiat itselle tutulla kielellä, vaatii Stack Overflown tai vastaavan käyttöä Javan kohdalla. Mm. *enum*eita ei voikaan muuntaa Javassa helposti integereiksi!

Se millaisella algoritmilla toteutan ”liikkuvan maalin”, vaikuttaa testattavuuteen. Esimerkiksi satunnaisesti liikkuvan maaliin johtava lyhin polku saattaa olla yksikkötestissä mahdoton määriteltävä. Tämän ongelman ratkaisun siten, että liikkuva maali – toiminto on ehdollinen, ja testattaessa aina *false*.

Testaamiseen vaikuttaa myös graafinen käyttöliittymä. Toistaiseksi en tiedä, saako sen yksikkötesteissä disabloitua vai vaatiiko sekin ohjelmaan ehdon.

Seuraavaksi aion tehdä ArrayLististä oman tietorakenteen. Prioriteettijonon tein jo alkuvaiheessa, koska se tuntui olevan helpompaa, kuin opetella Javan PriorityQueueen Comparerin käyttöä. Aion myös kirjoitella yksikkötestejä hiljalleen enemmän.