

## Testausdokumentti

Kirjoitan testausdokumentin hieman viikkoraportin tapaan, sillä jonkinlainen testaus on toki kokoajan työtä tehdessä läsnä ja oletan sen nimenomaan ajan kuluessa hieman vaihtelevan riippuen siitä, millaisessa kehitysvaiheessa ohjelmisto on.

### Viikko 1

Ensimmäisinä päivinä testaamiseksi riitti täysin manuaalinen seuranta mitä koodissa ja sen suorituksessa tapahtuu. Esimerkiksi Javan PriorityQueueea käyttäessä tuli ongelmia. Kun koitin ensimmäisen kerran ajaa hienoa  $A^*$  -algoritmiäni, se näytti löytävän mitä sattuu. Lopulta ymmärsin sen johtuvan siitä, että lisäsin nodeja PriorityQueueeen ENNEN kuin olin laskenut niille etäisyysarviot aloitussolmusta ja maalisolmusta. Tämä aiheutti sen, että PriorityQueueeen lisätyt solmut eivät enää olleetkaan oikeassa järjestyksessä sillä kyseinen jono ei varmastikaan uudelleenjärjestä itseään sitä mukaa kun tiedot muuttuvat.

Tämän seikan huomasin puhtaasti koodia silmäillessäni koittaen kohta kohdalta miettiä päässäni miten suoritus rivi riviltä etenee. Varmastikin olisin bugin löytänyt myös sopivilla Junit-testeillä, mutta ainakaan vielä ei ollut tarvetta niihin turvautua.

Toinen tässä työssä hyvin toimiva testaustapa on puhdas visuaalinen havainnointi, kiitos työn aiheen joka on paitsi  $A^*$  -algoritmi niin myös sen visualisointi. Tästä seuraa se, että jos algoritmi toimii jotenkin toisin kuin odottaisi, sen suurella toden näköisyydellä myös huomaa löytyneestä polusta. Näin kävi esimerkiksi kun olin lisännyt mahdollisuuden vinottaisliikkumiseen (sekä statusviestit jotka näyttivät lopullisen polun pituuden); polku ei ollutkaan saman pituinen tie-breakerilla ja ilman sitä eli algoritmissani oli selvästi bugi. Olin unohtanut käyttää eri heuristiikkaa vinottaisliikkumistapauksessa.