

Polunetsintä verkossa

Valitsemani harjoitustyön aihe on polunetsintä verkoissa. Ajatuksena on toteuttaa A*-algoritmi, joka on periaatteessa laajennos Dijkstran algoritmille. A* eroaa Dijkstrasta sen käyttämän heuristiikan osalta, joka laskee jokaiselle nodelle etäisyyden maalisolmusta. Tämän avulla A*:lla löydetään optimaalinen polku, joka tapahtuu usein huomattavasti Dijkstraa nopeammin.

Verkon mallintamisen olen alustavasti ajatellut toteuttaa bittikarttakuvasta tuotetulla kaksiulotteisella char-taulukolla, mutta tämä saattaa tarkentua vielä projektin edetessä. Ajatuksena on myös, että GUI:n kautta voisi itse piirtää karttoja, joihin voi asettaa esteitä, lähtösolmun sekä maalisolmun.

Jos ylimääräistä aikaa jää, aion tämän lisäksi toteuttaa myös Jump Point Searchin (JPS), joka on suhteellisen uusi polunetsintäalgoritmi. En ole vielä hirveästi ehtinyt tutustumaan JPS:ään, mutta ilmeisesti se löytää aina optimaalisen reitin ja se hyötyy suoritusajan suhteen, mikäli käsiteltävässä verkossa on halutun reitin ympärillä paljon avaraa tilaa.

A*- ja JPS -algoritmeja varten minun täytyy toteuttaa binäärikeko-tietorakenne, jonka toteutan omana luokkanaan.

A*-algoritmin aikavaativuus on sama kuin Dijkstrassa eli $O((|E| + |V|) + \log|V|)$ ja tilavaativuus on $O(|V|)$, missä V on verkossa olevien solmujen määrä ja E on kaarien lukumäärä.

JPS:n aika- ja tilavaativuudesta on ole vielä löytänyt varmaa tietoa, mutta se varmasti tarkentuu vielä mikäli aika riittää kyseisen algoritmin toteuttamiseen.

LÄHTEET:

<http://www.cs.helsinki.fi/u/floreen/tira2012/tira.pdf>

<http://harablog.wordpress.com/2011/09/07/jump-point-search/>

http://en.wikipedia.org/wiki/A*_search_algorithm