

Paula Lehtola
2.8.2012

Määrittelydokumentti

1. Mitä algoritmeja ja tietorakenteita toteutat työssäsi

Toteutan 15-pelin ratkaisijan IDA*-algoritmillä. Tai tarkemmin sanottuna N-pelin, koska pelilaudan kokoa voi muuttaa. Aikomukseni on testata ohjelmaa ainakin 3x3- ja 4x4 -pelilautoilla.

Aputietorakenteita: Priority Queue, mahdollisesti linkitetty lista, jokin map tms. Näitä täytyy vielä selvittää, mitkä rakenteet ovat tarkoituksenmukaisimpia. Luulen että asia valkenee pian kunhan pääsen koodin pariin. Aion joka tapauksessa toteuttaa itse ainakin pari eri tietorakennetta.

2. Mitä ongelmaa ratkaiset ja miksi valitsit kyseiset algoritmit/tietorakenteet

Yritän saada aikaan sellaisen ratkaisuohjelman, joka ratkaisee N-pelin mahdollisimman vähillä siirroilla mahdollisimman nopeasti. Aiheesta on olemassa paljon materiaalia ennestään, ja päädyin aiheen tutkimisen ja suositusten perusteella valitsemaan sellaisen algoritmin ja sellaiset heuristiikat, joilla tähän ongelmaan on olemassa tehokas ratkaisu.

Heuristiikka:

Olen alustavasti ajatellut käyttää Manhattan Distancea, johon on lisätty "linear conflict" -tilanteen vaikutus.

Lisäksi jokin toinen heuristiikka, mutta en ole vielä ihan varma, mikä. Mahdollisesti Pattern Database tai Walking Distance.

Kolmanneksi ehkäpä IDA* ilman heuristiikkaa.

3. Mitä syötteitä ohjelma saa ja miten näitä käytetään

Ohjelman syötteenä toimii N-pelin pelitilanne, eli sekoitettu pelilauta, joka on tarkoitus saada järjestykseen käyttämällä pelin sallimia siirtoja mahdollisimman vähäinen määrä. Pelilauta syötetään algoritmilta int-taulukkona, jossa tyhjää ruutua on merkitty esim. nollalla. Alussa ohjelma tarvitsee sekä sekoitetun alkutilanteen että tavoitetilanteen jossa lauta on järjestyksessä.

Ajatuksenani olisi saada ohjelma sellaiseen muotoon, että graafisen käyttöliittymän ja ratkaisualgoritmin saa kytkettyä toisiinsa. Tällöin käyttäjän olisi mahdollista syöttää ratkaistavan pelilaudan koko annettujen (järkevien) vaihtoehtojen puitteissa. N-pelin pelilogiikka ja käyttöliittymä ovat valmiina olemassa Ohjelmoinnin harjoitustyön jäljiltä.

4. Tavoitteena olevat aika- ja tilavaativuudet (m.m. O-analyysi)

IDA*/A* -algoritmin aikavaativuus riippuu käytetystä heuristiikasta. Toistaiseksi en onnistunut löytämään IDA*-algoritmin aikavaativuudesta niin yksiselitteistä tietoa, että olisin sen kunnolla ymmärtänyt. Polynomisesta aikavaativuudesta ainakin puhuttiin jossain. Tavoitteet aika- ja tilavaativuuden suhteen tarkentuvat siinä vaiheessa kun alan oikeasti ymmärtää, mitä olen tekemässä.

5. Lähteet

http://en.wikipedia.org/wiki/Fifteen_puzzle

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

http://en.wikipedia.org/wiki/IDA*

<http://www.heyese-jones.com/astar.html>

<http://heuristicswiki.wikispaces.com>

<http://kociemba.org/fifteen/fifteensolver.html>

Korf, Reid, Edelkamp : Time Complexity of Iterative-Deepening-A* (2001)

<http://www.math.ucf.edu/~reid/Research/Korfedelkamp/index.html>