KESKENERÄINEN !!!

TOTEUTUSDOKUMENTTI 14.6.2013
Fifteen Puzzle – tietorakenteiden harjoitustyö

@Tero Mäntylä

Mitä tein
Totetutin Fifteen Puzzle -palapeliin erilaisia ratkaisu algoritmeja, jotka etsivät satunnaisesti sekoitetulle palapelille pienimmän siirtosarjan jolla sen voi ratkaista.
Viimeisimpänä tein A*-algoritmin pohjalta toimivan toteutuksen. Sitä varten koodasin itse minimikeon ja hajautustaulun. Hajautustaulussa tosin käytin Javan omaa Arrays-luokan deepHashCode()-metodia.
Tärkein työn kohde oli kirjoitta IDA*-algoritmin ideaan perustuva palapelin ratkaisija. Algoritmin tehokkaan toiminnan kannalta tärkeässä osassa oli hyvän heuristiikan löytäminen ja sen koodaaminen toimimaan tehokkaati. Tätä osaa varten ei tarvinnut tehdä mitään Tietorakenteet ja algoritmit kurssilla esiteltyä tietorakennetta.
Iso osa alun työstä oli toteutta tietorakenne palapelille itselleen. Se on tehty Puzzle-luokkaan.

Tarkastellaan toteutettuja tietorakenteita yksi kerrallaan

Puzzle-luokka ja palapelin sekoittaja
Tämä tietorakenne muistaa viimeisimmän tilanteen - palojen paikat palapelissä erityisesti tyhjän paikan sijainnin ja viimeisimmän tehdyn siirron. Luokka tarjoaa myös toiminnalisuuden palojen hyväksyttäviin siirtoihin sekä palapelin sekoittamiseen satunnaiseen järjestykseen.
Palojen paikkatieto on talennettu kaksiulotteiseen byte-muuttuja tauluun. Koska palat pystytään kuvaamaan 16 numerolla, voisi tämän tiedon pakata 4 bittiin. Ja koko palapeli mahtuisi yhteen 64 bittiseen long-muuttujaan. Päätin kuitenkin yksinkertaisuuden vuoksi käyttää kuuttatoista 8 bittistä byte-muuttujaa, jotka vievät tilaa 128 bittiä. Eli muistitilan haaskaus on 100 prosentia. Muut tietorakenteen luokkamuuttujat ottavat tilaa 146 bittiä. Tällä ei IDA* toteutuksessa ole merkitystä, mutta A* kyllä tästä kärsii muisti intensiivisenä algoritmina.

Satunnainen sekoitus …..

Minimikeko

Hajautustaulu

A*-algoritmi

IDA*-algoritmi

Puzzle luokka ja palapelin sekoittaja