

Työn aihe: A* -reitinhakualgoritmia käyttävä ohjelma joka muodostaa kaksiulotteisen kartan ja etsii (lyhimmän) reitin kahden annetun pisteen välillä.

Käytetään reitinhakuun Manhattan metodia:

$$H = 10 * (\text{abs}(\text{currentX} - \text{targetX}) + \text{abs}(\text{currentY} - \text{targetY}))$$

koska se vaikuttaa selkeimmältä ja suurin osa esimerkeistä käyttää sitä.

Toiminnalliset vaatimukset

Algoritmin tulee toimia oikein, eli löytää lyhin (ts. tehokkain) reitti kahden annetun pisteen välillä. Pisteet annetaan muodossa (x,y).

Algoritmin tulee löytää reitti vaikka lyhimellä polulla olisi esteitä.

Mikäli lyhintä reittiä ei ole mahdollista löytää, ohjelma ei saa kaatua hallitsemattomasti.

Ohjelma osaa muodostaa kaksiulotteisen kartan annetun testidatan tai kuvatiedoston pohjalta. Testidatassa muodostetaan X*Y -kokoinen kaksiulotteinen taulukko johon asetetaan esteitä merkitsemällä karttaan pisteitä joiden läpi ei voida liikkua. Kuvatiedostosta generoitaessa tulkitaan kuvasta vastaavat tiedot ja muodostetaan taulukko samoin periaattein.

Ohjelma käyttää itse toteutettua tietorakennetta (keko), joka korvaa Javan LinkedListin/PriorityQueuen käytön.

Toiminnalliset lisävaatimukset

Vaihtoehtoinen toteutus toisella tietorakenteella keon asemesta ja toteutusten tehokkuuden vertailu.

Muut vaatimukset

Julkiset metodit ja muuttujat kommentoitava JavaDocilla.

Ohjelma kattavasti yksikkötestattu.

Ohjelma kattavasti testattu ja suorituskäytetty.

Aikavaativuus: PriorityQueueella reitin etsinnän aikavaativuus on $N(\log N)$, jossa $\log N$ on yksittäisen noodin ympärysnooidien läpikäynti. Reitinhaussa noodeja joiden ympärysnoodeja läpikäydään on N kappaletta. Kartan alustuksen aikavaivuutta tässä ei huomioida, mutta se on luokkaa n^2 , eli kaksi sisäkkäistä looppia.

Kekototeutuksen on pääpiirteissään samanlainen kuin PriorityQueue, joten myös sen aikavaivuuden tulisi olla $N(\log N)$, eli suorituskäytetyn suoritusajojen tulisi molemmilla toteutuksilla kasvaa suunnilleen samassa tahdissa.

Käytetyt lähteet

Tietorakenteet -kurssi, kevät 2011. Luentokalvot.

Kurssin aihe -sivulla linkitetty artikkeli "Introduction to A*".

Erilaiset hakukoneille löydetyt A*:n toteutukseen liittyvät kysymykset, vastaukset ja esimerkit. Varsin usein stackoverflow.com:in kautta löydetyt.