# 

# Jäljittäjä

# A \* - algoritmiin perustuva polunetsintä liikkuvaan maaliin

## Harjoitustyö: Tietorakenteet ja Algoritmit

Jouni Männistö

Heinolankatu 6 A 17

00520 Helsinki

jouni.mannisto@cs.helsinki.fi

# 

# Määrittelydokumentti

# 

## Sisällysluettelo

[Sisällysluettelo](#h.d7v0x5vjjbf)

[Johdanto](#h.8j5dd31ljroc)

[1.1. Vaatimusdokumentin tarkoitus](#h.rauog8y2mczq)

[Yleiskuvaus](#h.2n6n60d522iz)

[2.1. Ohjelman toiminta](#h.g32ukzn43rxm)

[2.2. Algoritmit ja tietorakenteet](#h.mcjqr61xuk4b)

[2.3. Ohjelman syötteet](#h.qkfnc4y6plvf)

[Vaatimukset](#h.ndibv5cgnc1p)

[3.1. Toiminnalliset vaatimukset](#h.fbm9i2a3oyy8)

[3.2. Aika- ja tilavaatimukset](#h.lozmvqhlnlia)

[Lähteet](#h.g06ti2d9n69t)

[Versiohistoria](#h.gkcuicpcsd9s)

# 

# 

# Johdanto

### 1.1. Vaatimusdokumentin tarkoitus

Tämän dokumentin tarkoitus on kuvata Tietorakenteet ja Algoritmit - kurssin harjoitustyön määrittelyt. Määrittelyt voivat tarkentua työn edetessä.

# 

# Yleiskuvaus

### 2.1. Ohjelman toiminta

Ohjelma etsii lyhimmän polun pisteiden A ja B välillä suuntaamattomassa, painotetussa verkossa, jossa A on lähtöpiste ja B on määränpää. B-pisteen sijainti muuttuu polun etsinnän aikana. Jokaisella solmulla on kustannus, joiden vertailun perusteella polun muodostavat solmut valitaan.

Maalin eli B-pisteen liikehdintä (vauhti ja matka) tarkentuu kehityksen edetessä.

Ohjelman suoritusta kuvataan välitulostuksin. Mikäli aikaa on riittävästi, on tavoitteena kuvata polunetsintää myös graafisesti.

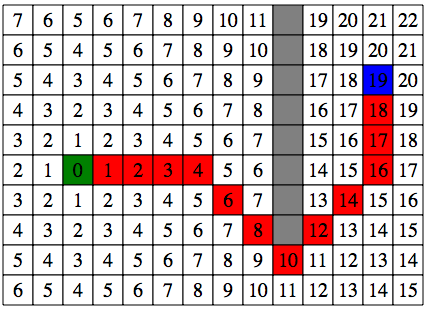
### 2.2. Algoritmit ja tietorakenteet

Polunetsintä perustuu A\*-algoritmiin (kuva 1), jossa lyhimmän reitin päättelyyn käytetään heuristista arviota kustannuksesta ja jäljellä olevaa matkaa kohteeseen.

Polunetsinnän heuristinen arviointi perustuu ns. “Manhattan-etäisyyteen”, joka tarkoittaa liikkumisen sallimista vain pysty- ja vaakasuunnassa. Ts. diagonaalista liikettä ei alustavasti sallita.

Koska tarkoitus on simuloida liikkuvan maalin jäljitystä, myös maalin, eli pisteen B, liikkuminen toteutetaan jollakin myöhemmin päätettävällä logiikalla.

Varsinainen A\*-algoritmi käyttää tietorakenteinaan listoja (mm. käydyt solmut eli suljettu lista) sekä prioriteettijonoa (avoin lista), jonka priotiteetti perustuu siihen talletettavan solmun kustannukseen.



Kuva 1. A\* - algoritmin löytämä polku. Esimerkissä liikehdintä tapahtuu

diagonaalisesti.

### 2.3. Ohjelman syötteet

Ohjelman syötteenä on matriisi (verkko) sekä alku- ja kohdesolmut. Matriisi rakennetaan ohjelmallisesti ja siihen voi muodostaa esteitä.

### 

# Vaatimukset

### 3.1. Toiminnalliset vaatimukset

Tavoitteena on saavuttaa “liikkuva maali” eli piste B pienimmin mahdollisin kustannuksin, tilanteessa, jossa B on mahdollista saavuttaa.

### 3.2. Aika- ja tilavaatimukset

Perus A\*-algoritmin tehokkuus riippuu käytetystä heuristiikasta. Jos arviointi osuu oikeaan, vähentää se läpikäytävien solmujen määrää.

Tässä ohjelmassa ns. liikkuva maali (pisteen B muuttuva sijainti) aiheuttaa sen suorituskyvylle huomattavaa lisähaittaa. Reittiä täytyy päivittää aina B:n liikkuessa, jolloin jo läpikäytyjen solmujen kustannukset täytyyt laskea uudelleen. B:n liikehdintä saattaa olla myös edullista polunetsinnälle (esim. maali tulee lähemmäs), mutta sen hyödyntäminen edellyttää uudelleen laskentaa.

Tällöin ohjelman tehokkuuteen vaikuttaa käytetyn heuristiikan (“paras arvaus”) lisäksi myös uudelleen laskennan tiheys. Kuinka usein kannattaa päivittää tiedot?

Aikavaatimus ohjelmalle on , jossa n on solmujen määrä. TODO:

Ohjelman muistin käytölle ei ole vaatimuksia.

# Lähteet

[http://en.wikipedia.org/wiki/A\*\_search\_algorithm](http://en.wikipedia.org/wiki/A*_search_algorithm)

<http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html>

<http://en.wiktionary.org/wiki/Manhattan_distance>

# Versiohistoria

|  |  |  |
| --- | --- | --- |
| 16.3.2015 | Jouni Männistö | ensimmäinen versio |
| 18.3.2015 | Jouni Männistö | täydennetty / muokattu |
| 19.3.2015 | Jouni Männistö |  |

### 