

# Käyttöohje

## Graafinen käyttöliittymä

Graafinen käyttöliittymä avautuu, kun ohjelman suorittaa ilman argumentteja.

Ensimmäisessä ikkunassa tulee valita browse-painikkeesta syötekuva, sekä aloitus- ja lopetuspiste joko klikkaamalla valittua syötekuva tai syöttämällä koordinaatit suoraan vasemman alakulman tekstilaatikoihin.

Toisessa ikkunassa valitaan oikealla olevasta listasta suoritettava algoritmi sekä lisävalintoja sen suorittamiseen. Time limit määrää, kuinka kauan algoritmia yritetään suorittaa ennen kuin se pysäytetään. Visualisointitilan ollessa päällä ohjelma piirtää kaikki algoritmin välivaiheet. Tässä tilassa suoritusaikaa ei mitata eikä aikarajoitusta huomioida. Speed-liukusäädin määrää kuinka nopeasti algoritmia suoritetaan visualisointitilassa.

## Komentoriviargumentit

- i *input* Syötetiedosto
- o *output* Tulosteen tallennuskansio (piirretyt polut)
- s *start* Reitinetsinnän aloituspiste, muotoa x,y
- g *goal* Reitinetsinnän lopetuspiste, muotoa x,y
- t *time* Aikaraja algoritmin suoritukselle sekunteina
- c *count* Kuinka monta kertaa algoritmi tai algoritmit suoritetaan
- a *algorithm* Suoritettava algoritmi. Ilman tätä argumenttia ohjelma suorittaa kaikki tuntemansa algorimit. Sallitut vaihtoedot:
  - A\_STAR
  - DIJKSTRA
  - JUMP\_POINT\_SEARCH
  - BREADTH\_FIRST\_SEARCH
  - DEPTH\_FIRST\_SEARCH
- less Tämän argumentin ollessa asetettu ohjelma tulostaa vain suoritukseen kuluneet ajat.

Ohjelmalle on annettava vähintään syötetiedosto sekä aloitus- ja lopetuspisteet. Muut argumentit ovat valinnaisia.

Esimerkkejä:

```
java -jar Tiralabra.jar -i input.png -s 0,0 -g 199,199
```

```
java -jar Tiralabra.jar -i input.png -o output -s 0,0 -g 199,199 -t 10
```

```
java -jar Tiralabra.jar -i input.png -s 0,0 -g 199,199 -c 10 -a DIJKSTRA --less
```

# Syötekuva

Ohjelma hyväksyy syötteeksi minkä tahansa kuvatiedoston, jonka Javan ImageIO osaa tulkita kuvaksi. Ohjelma muuttaa syötekuvan reitinetsintäalgoritmien käyttämäksi verkoksi seuraavin perustein:

- Kaikki valkoiset pikselit (väriarvo 0xFFFFFFFF) ovat avoimia solmuja
- Kaikki muut (ei-valkoiset) pikselit ovat suljettuja solmuja
- Kaksi solmua ovat vierekkäin, jos niitä vastaavat pikselit ovat vierekkäin vaaka-, pysty- tai vinosuunnassa
- Kahden vierekkäisen avoimen solmun välillä on kaari
- Suljetuilla solmuilla ei ole kaaria vierekkäisiin solmuihin
- Millään kahden ei-vierekkäisen solmun välillä ei ole kaarta

Käytännöllisesti sanoen tämä tarkoittaa, että valkoiset pikselit luovat alueen, jonka läpi reitinhakualgoritmi etsii reittiä maaliin, ja muunväriset pikselit ovat esteitä, joiden läpi reitinetsintä ei voi kulkea vaan ne on kierrettävä.

## Jar ja testitiedostot

Suoritettava jar-paketti löytyy kansioista Tiralabra/, ja testaukseen käytetyt kuvat löytyvät kansioista Tiralabra/test-images/.