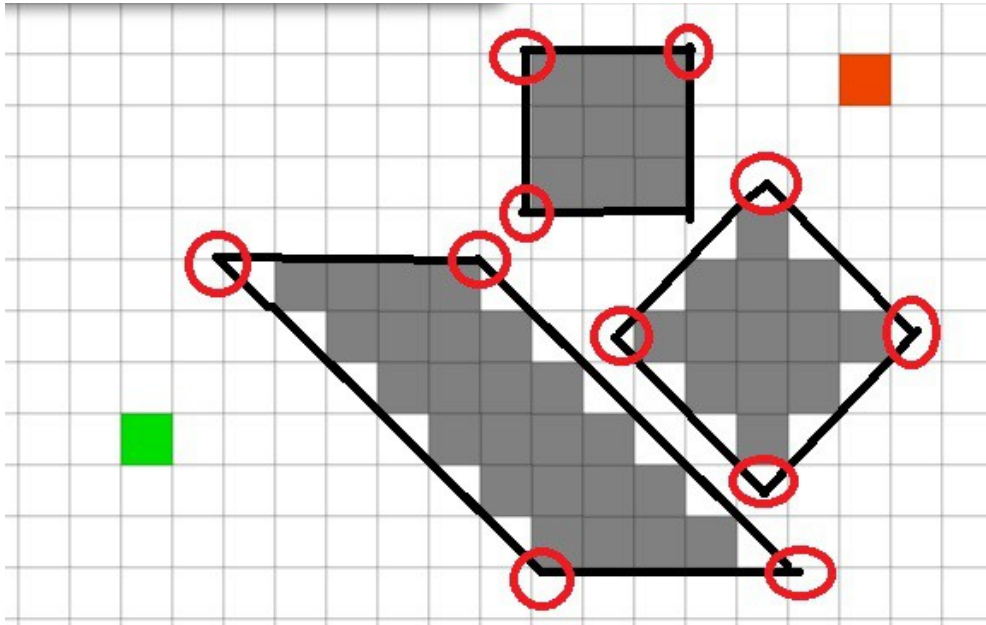


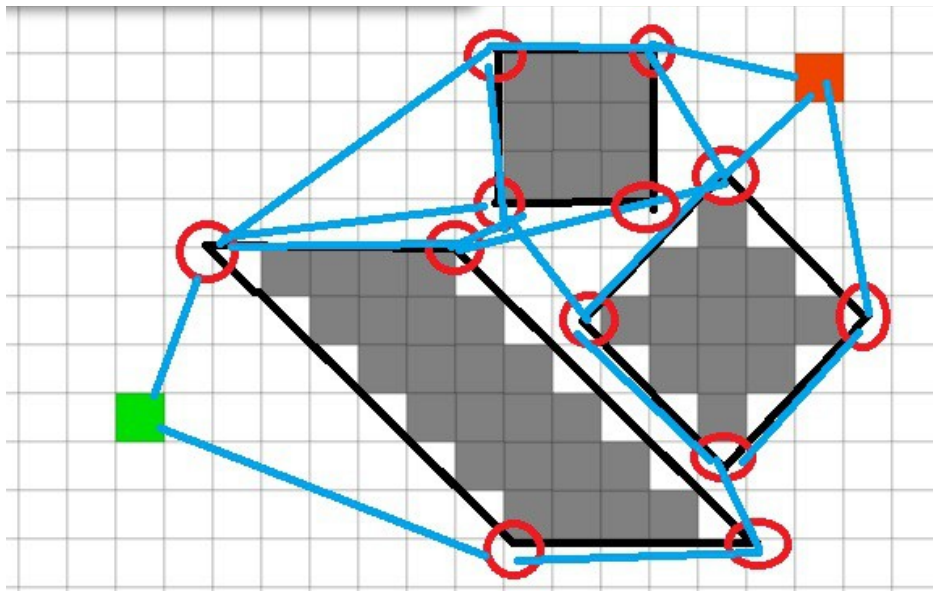
Toteutusdokumentti

- Yleisrakenne:

- 1) Käyttäjä syöttää monikulmiot, alku, sekä loppupisteen
 - 2) Ohjelma laskee käyttäen tätä tietoa Verkon luokassa Jatkuverkkorakenne
- Solmujoukoksi asetetaan monikulmioiden kulmat, alku, sekä loppupisteen.



- Ohjelma käy läpi kaikki solmuparit (n yli 2) ja laskee mitkä solmut näkevät toisensa. Näistä muodostetaan viivajoukko:



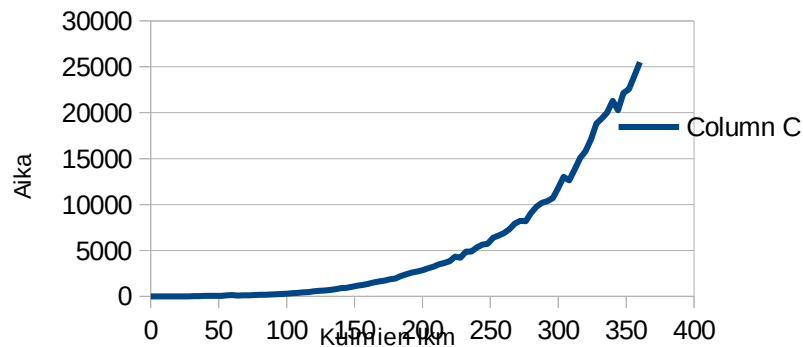
- 3) Ohjelma laskee nopeimman reitin alkusolmusta loppusolmuun käyttäen A*. Ohjelma palauttaa reitin murtoviivana, sekä reitin pituuden

Saavutettu aikavaatiavuus:

n on määritelty monikulmioiden kulmien lukumääränä.

- Verkon muodostamiseen iteroidaan (n yli 2) solmua ja jokaisessa tapauksen kohdalla kuluu korkeintaan $O(n)$ aikaa, sillä käytin LinkedList rakennetta etsintään. Joten rakentamiseen menee korkeintaan $O(n^3)$ (Jos Hashsettejä ja Hashsettejä olisi saanu käyttää tai olisin ehtiny toteuttaa sellaiset aikavaatiavuus olisi $O(n^2)$)
- A* laskee lyhyimmän reitin $O(n^2)$ (Jos verkko on tiheä)
- Siis aikavaatiavuudeksi saadaan $O(n^3)$

Suorituskyky



Saavutettu tilavaatiavuus:

Jokaisen kulman kohdalla säilytetään listaa naapureista. Siis tilavaatiavuus on korkeintaan $O(n^2)$

Työn mahdolliset puutteet ja parannusehdotukset

Ohjelma voisi parantaa seuraavilla tavoilla:

- Aikavaatiavuuden parantaminen $O(n^2)$ luokkaan
- Graafinen käyttöliittymä
- Ohjelma voisi laskea myös approksimaation diskreettisesti
- Unioni operaatio: Silloin kuin lisätään monikulmioita ja jotkin leikkaavat ne voisi korvata yhdellä isommalla monikulmiolla.

Lähteet:

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

<http://www.codeproject.com/Articles/371959/A-brute-force-approach-to-check-if-a-line-segment>