

Toteutusdokumentti

Ohjelman yleisrakenne

Olen yrittänyt tehdä ohjelman mahdollisimman helposti jatkettavaksi, joten toteutin esimerkiksi algortimit niin, että ne perivät abstraktin luokan Hakualgoritmi, joka tarjoaa tarvittavat julkiset metodit kyseisten hakualgortimien käyttäjille. Keon toteutin myös niin, että sille annetaan parametrina Comparator-olio, joka määrittelee kekoon laitettavien alkoiden ”arvot”. Tämä mahdollistaa sen, että sama kekoa voidaan käyttää minimi- tai maksimikekona tarpeen vaatiessa. Tarkoituksena olisi toteuttaa keko myös niin, että sille voidaan antaa mikä tahansa object-luokan ilmentymä.

Aikavaativuudet

Algortimien aikavaativuudet vaikuittaisi olevan samat, kuin pseudokoodista katsottuna, eli Dijkstralla, A*:lla ja jps:lla kaikilla sama $O((|E| + |V|) + \log(|V|))$ ja tilavaativuus $O(|V|)$, missä V on solmujen lukumäärä ja E kaarien lukumäärä. Keon lisäysoperaatio on $O(\log n)$ ja poistaminen $O(\log n)$ myös, missä n on alkoiden määrä. Tilavaativuus keolla on luonnollisesti $O(n)$.

Puutteet ja parannusehdotukset

Tiukoista ponnisteluista huolimatta A* algoritmiin on jäänyt pieni bugi, jonka takia se ei tuota optimaalista reittiä kaikilla syötteillä, ilmeisesti tämä johtuu siitä, jos jo keossa olevalle nodelle löytyy lyhyempi reitti valmiiksi keossa olevan toisen noden kautta, niin silloin tämän lyhyemmän reitin päivittämisessä tapahtuu jotain pientä häikkää.

Graaffisessa käyttöliittymässä olisi paljonkin korjailtavaa, johon ikävä kyllä aikani ei millään riittänyt. Esimerkiksi uusien karttojen piirtäminen suoraan käyttöliittymästä, käsiteltyjen nodejen piirtäminen pelkän reitin piirtämisen ohelle, jump point searchissa reitin piirtäminen esim. viivalla, kun tällä hetkellä jps:n reitti näkyy pelkkinä jump pointteina.

Lähteet

<http://www.cs.helsinki.fi/u/floreen/tira2012/tira.pdf>

<http://zerowidth.com/2013/05/05/jump-point-search-explained.html>

<http://gamedevelopment.tutsplus.com/tutorials/speed-up-a-star-pathfinding-with-the-jump-point-search-algorithm--gamedev-5818>

<http://harablog.wordpress.com/2011/09/07/jump-point-search/>