

Rinnakkainen reitinhaku tietokonepeleissä

Santeri Martikainen

Aine
HELSINGIN YLIOPISTO
Tietojenkäsittelytieteen laitos

Helsinki, 19. lokakuuta 2016

Sisältö

1 Johdanto	1
2 Rinnakkainen reitinhaku	2
2.1 Local Repair A*	← suomeksi / nimeäminen?
2.2 Cooperative A*	
2.3 Hierarchical Cooperative A*	
2.4 Windowed Hierarchical Cooperative A*	
2.5 Conflict-based Search	
3 Yhteenveto	
4 Lähteet	

1

Johdanto

Reitinhakua tarvitaan useilla eri tahoilla niin reaalimaailmassa kuin virtuaaliympäristössäkin. Tyypillisiä käyttökohteita ovat muun muassa viranomaisten etsintä- ja pelastustoimet, joissa käydään läpi rakennuksia, sekä maa- ja vesialueita apua tarvitsevien ihmisten löytämiseksi. Tällöin on pelastustoimien onnistumisen kannalta keskeistä, että läpikäytävä kohde saadaan tutkittua mahdollisimman tehokkaasti ja kauttaaltaan. Niinkään erilaisissa partointitehtävissä voidaan pyrkiä optimoimaan resurssien käyttöä suunnittelemalla kuljettava reitti ennalta haluttujen painotusten mukaan.

Myös teollisuudessa hyödynnetään reitinhakua monin eri tavoin. Sovelluskohteita ovat esimerkiksi tavarakuljetukset, kalastusalusten ja harvestereiden ohjaaminen ja erilaiset automatiolaitteet ja robotit. Tavallisen kuluttajan näkökulmasta kenties tutuin käyttökohde on autojen navigaattorit, joiden avulla kuljettaja voi ilman perinteistä karttaa suunnistaa haluamaansa kohteesseen. Reitinhakua sovelletaan myös laajalti tietotekniikassa, kuten verkkoliikenteen reitityksessä ja tietokonepelien saralla. [1][2][3] (**varmaan karsittava itse aiheeseen liittymättömiä rinnastuksia, kirjoitan tämän joka tapauksessa vielä kertaalleen uusiksi**)

Tässä aineessa keskitytään tarkastelemaan reitinhakua tietokonepelien osalta ja siinä erityisesti tilannetta, missä on monta eri toimijaa joille kaikille on löydettävä reitti tietyin reunaehdoin.

2

Rinnakkainen reitinhaku (tietokoneleissä?)

Mistä videopelien reitinhaussa oikeastaan on kysymys? (**ovatko retoriset kysymykset huonoa tyylilä?**) Pelimaailman topologia, ts. kenttä, alue, jolla pelaaja ja mahdolliset ei-pelaajahahmot voivat liikkua, voi olla ulkoasultaan hyvin abstrakti ja pelkistetty, tai toisaalta vaikuttaa hyvin realistiselta maastolta puineen, vesistöineen, rakennuksineen ja niin edelleen. Rakenteellisesti kenttä muodostuu tyypillisesti joko verkosta tai soluista, jotka käsittävät koko pelialueen, ja joista osa toimii esteinä.[6]

Realismia tavoittelevassa pelimaailmassa esteitä ovat luonnollisesti kaikki sellaiset objektit, jotka olisivat reaalimaailmassakin esteitä, kuten puut tai seinät. Siten pelissä liikkuva hahmo ei usein voi kulkea kahden pisteen välillä suoraan, vaan esteiden ympäri on löydettävä jokin reitti, jonka löytämiseen sovelletaan reitinetsintääalgoritmeja. (**pitäisikö tämän oikeastaan kuulua johdantoon?**)

Reitinetsintä kokonaisuutena jakaantuu kahteen vaiheeseen: Toimintaympäristöstä muodostetaan ensin yksinkertaistettu malli, minkä jälkeen sitä käydään läpi jollakin algoritmilla halutun reitin löytämiseksi. Tunnetuin näistä algoritmeista on nimeltään A*, ja siitä on kehitetty lukuisia eri variantteja eri toimintaympäristöjä silmälläpitäen. [6]

A* (eli A-star tai A-tähti) toteuttaa niin sanottua paras ensin-tyyliä, jossa jokaisen solmun tai solun kohdalla pyritään ensiksi etenemään suoraan kohti maalia. Jos tiellä on jokin este, algoritmi pyrkii kiertämään sen pyrkimällä viereisiin soluihin ja sieltä jälleen maalia kohti kunnes joko kohteesseen päästään, tai reittiä ei voida löytää. A* on variantti Dijkstran algoritmista (**ehkä tarpeetonta**)

(**Verkkojen ja solujen aukikirjoittaminen?**)

Videopeleissä on yleensä useita, jopa satoja tai tuhansia, eri toimijoita, joille on löydettävä reitit kohteisiinsa. Tällainen monen toimijan rinnakkainen reitinhaku (multi-agent pathfinding) tuo yksittäisen toimijan, reitinhakuun verrattuna uusia ongelmia, kuten sallitaanko reittien risteäminen,

voiko kaksi tai useampi toimija olla samaan aikaan samassa paikassa ja tuleeko liikkumista porrastaa odottamalla että reitti edessä vapautuu? [1]
Näiden ongelmien ratkaisemiseen on kehitetty useita eri algoritmeja.

(kuvien käyttö havainnollistukseen, ok vai ei?)

2.1 Local Repair A*

2.2 Cooperative A*

- [1] Erdem, E.; Kisa, D.G.; Oztok, U.; Schüller, P.
A General Formal Framework for Pathfinding Problems with Multiple Agents
AAAI, 2013
- [2] Sharon, G.; Stern, R.; Felner, A.; Sturtevant, N.R.
Conflict-based search for optimal multi-agent pathfinding
Artificial Intelligence, 2015
- [3] Standley, T. S.
Finding optimal solutions to cooperative pathfinding problems.
In Proc. of AAAI, 2010
- [4] Silver, D.
Cooperative Pathfinding
In Proc. of AIIDE, 117-122, 2005
- [5] Cui, X.; Shi, H.
A*-based Pathfinding in Modern Computer Games
IJCSNS International Journal of Computer Science and Network Security, VOL.11
No.1, January 2011
- [6] Algfoor, Z.A.; Sunar, M.S.; Kolivand, H.
A Comprehensive Study on Pathfinding Techniques for Robotics and Video Games
International Journal of Computer Games Technology Volume 2015, Article ID
736138