

A*- ja Dijkstra- reitinhakualgoritmien testausdokumentaatio

Tietorakenteiden harjoitustyö
Santeri Martikainen
014152902
Helsingin yliopisto
Kesä 2013

Mitä on testattu ja miten tämä tehtiin?

Testauksen kohteena on kolme luokkaa: Astar, Dijkstra ja Minimikeko.

Astar

Tarkoitus on testata algoritmin suorituskykyä tekemällä sarjana vähintään satoja hakuja satunnaisten (ja jokaisen haun jälkeen muuttuvien) lähtö- ja maalipisteiden välillä, sekä laskea näistä keskimääräinen hakuaika.

Dijkstra

Luokka on miltei täysin identtinen Astar-luokan kanssa, joten testaus noudattaa matalimman aidan periaatetta tapahtuen lähinnä Astar-testiluokan kautta.

Minimikeko

Tälle luokalle on jo olemassa kahdeksan testiä, jotka nähdäkseni kattavat vähimmäismäärän, mitä luokan testaukselta voi vaatia. Tarkoitus on jatkaa tämän laajentamista, kunhan edelliset kaksi luokkaa ovat saaneet ensin huomiota osakseen.

Minkälaisilla syötteillä testaus tehtiin?

Reitinhakualgoritmeille käytetään syötteinä satunnaisesti poimittuja halutun reitin lähtö- ja maalipisteitä.

Minimikeko-luokkaa sen sijaan testataan tällä hetkellä 'kovakoodatusti' antamalla sille joukko objekteja, ja vertaamalla sen metodien tuloksia kynää ja paperia käyttäen saatuihin tuloksiin.

Miten testit voidaan toistaa?

Testit voi toistaa avaamalla projektin Netbeansissa ja valitsemalla valikosta Run → Test Project.

Tulosten esittäminen graafisessa muodossa

Koodattavana olevassa pelissä on toiminnallisuudet reitinhakuun ja sen visualisointiin pelin sisältä käsin. Nämä toimivat painamalla **a**(star)- ja **d**(ijkstra)-näppäimiä.

Tämän kurssin aihepiiriin liittyvät myös näppäinkomennot **c** (poistaa edellisen reitinhaun visualisoinnin kuluttamatta pelimaailman aikaa) ja **q** (näyttää koordinaatit missä pelaajahahmo paraikaa sijaitsee).

Pelin sisällä voi asettaa päähenkilölle uuden maalin painamalla **m** ja antamalla sitten suuntakomennon nuoli-/numpad-näppäimillä. Nyt a- ja d-näppäimet luovat reitinhaun päähenkilöltä vasta asetettuun maaliin.