

## Reitinhaku videopeleissä

Hyvin monissa videopeleissä toimintaympäristönä on jonkinlainen kenttä, jolla pelin tapahtumat visualisoidaan ja jossa pelissä esiintyvät toimijat tai hahmot (agent) liikkuvat. Kentät voivat pelkistetyimmillään olla vain kaksiulotteisia tasoja, tai toisaalta hyvinkin uskollisesti realismia tavoittelevia kolmiulotteisia pelimaailmoja. Riippumatta kenttien kompleksisuuden asteesta, on tavanomaista että kenttien rakenne ei ole täysin avoin, vaan niissä on erilaisia esteitä. Pelkistetyissä pelimaailmoissa kenttä saattaa esimerkiksi olla jaettu pelaajalle näkyväksi ruudukoksi, jossa osa ruuduista puuttuu ja on sellaisia joihin ei voi siirtyä. Realismia tavoittelevissa peleissä puolestaan esteitä voivat olla puut, rakennukset, kivet ja muut reaali maailmasta tutut elementit, ja kuljettavia alueita ruohokentät, tiet ja niin edelleen. Siinä missä pelaajahahmon liikkuminen on usein jätetty pitkälti pelaajan itsensä vastuulle, hoitaa peliin ohjelmoitu tekoäly luonnollisesti muiden pelihahmojen liikkeen. Aivan yksinkertaisimpia pelejä lukuunottamatta tämä tarkoittaa käytännössä jonkinlaisen reitinhakualgoritmin käyttämistä. Näitä algoritmeja varten pelimaailma mallinnetaan usein solmuista ja kaarista koostuvaksi graafiksi, missä solmut edustavat liikkumisen kohteena olevia kiintopisteitä, ja kaaret näiden välisiä yhteyksiä mahdollisine liikkumiskustannuksineen.

Reitinhaussa käytetään yleisesti erilaisia A\*-algoritmin variantteja. A\* on niin kutsuttu paras ensin -reitinhakualgoritmi, joka pyrkii ensisijaisesti suoraan maalia kohti ja esteen kohdatessaan alkaa käydä läpi niitä solmuja, joiden laskettu etäisyys kohteeseen on pienin. Etäisyys lasketaan kentän topologiaan sovitettulla heuristiikkafunktiolla, jolla voidaan ottaa huomioon kentän ympäristötekijöitä. Käytännön esimerkkinä voidaan ajatella vaikkapa suota jonka yli halutaan mennä ja jonka lävitse kulkee kapea polku, mutta polun alkupäähän päästäkseen pitäisi valita suoraa linjaa pidempi reitti. Jos tässä

kuvittelemassamme tilanteessa tavoitellaan realismia, niin on ilmeistä että polkua pitkin kulkemisen pitäisi johtaa nopeammin perille kuin suon läpi kahlaaminen, mikäli polku ei sijaitse kovin kaukana.

Reitinhaun odotetaan yleisesti täyttävän tiettyjä reunaehtoja. Ensinnäkin mikäli kentällä on jokin olemassaoleva reitti haluttuun kohteeseen, sen tulee olla löydettävissä. Tässä on haasteena kentällä mahdollisesti tapahtuvat muutokset, jolloin optimaalinen – tai jopa ainoa – reitti saattaa olla väliaikaisesti tukossa, mutta algoritmin pitäisi pystyä ottamaan tämä huomioon. Toisekseen algoritmi ei saisi johtaa tarpeettomiin silmukoihin, eli hahmon liikkumiseen samojen sijaintien välillä. Ajatellaan tilannetta, missä kohteeseen voi päästä vain kahden toisistaan etäällä olevan oviaukon kautta. Emme varmaankaan halua, että ensimmäisen oviaukon liepeillä odottava hahmo pysähtyy reitin ollessa tukossa, lähtee vapaata oviaukkoa kohti, ja kääntyy sitten takaisin huomattessaan ensimmäisen reitin vapautuvan. Haluamme todennäköisesti mieluummin nähdä hahmon odottavan paikallaan ensimmäisen oviaukon liepeillä, mikäli ennakoitu odotusaika olisi lyhyempi kuin kiertotien käyttämiseen menevä aika.

Reitinhaun toteuttaminen yhtäaikaisten toimijoiden tapauksessa onkin varsin ongelmallista, ja tarkoitukseen on kehitetty useita monen toimijan (multi-agent) reitinhakualgoritmeja. Nämä pyrkivät ratkaisemaan yhteentörmäysongelmia, edellämäinittuja silmukoita, sekä säästämään laskentaresursseja. Lähestymistapoja ovat muun muassa yhteentörmäyksien ennakoiminen ja reitin muuttaminen sen pohjalta, abstraktiomallit reitinhaun apuna, muiden toimijoiden väistäminen, odottaminen, sekä reitin laskennan rajoittaminen johonkin etäisyyteen.

Reitinhaku videopeleissä on siis ongelmana varsin monisyinen, ja sopivan lähestymistavan löytäminen riippuu täysin pelin rakenteesta ja sen halutusta toimintatavasta.