Toteutusdokumentti

Yleisrakenne

MiinaharavanRatkaisija laskee ja päivittää tietoja neljään erilaiseen taulukkoon. Ruudut taulukossa on tieto avatuista ruuduista, miinat taulukossa löydetyistä varmojen miinojen sijainneista, viereistenMiinojenMäärä kertoo avattujen ruutujen viereisen miinojen määrän ja eiloydettyjenViereistenMiinojenMaara sisältää lasketun tiedon kuinka moni ruudun ympärillä olevista miinoista ei ole vielä löydetty.

Ratkaisija tekee aina aloitussiirron keskelle pelilautaa, koska silloin on todennäköisintä saada mahdollisimman iso tyhjä alue avattua ja ratkaistavien siirtojen määrä on mahdollimman pieni.

Algoritmin toiminta

Ratkaisija aloittaa pelilaudan läpikäymisen aina 0,0 koordinaatista, joten sitä mukaan mitä enemmän ylävasemmalta lähtien algoritmi löytää ratkaistavia ruutuja, se lisää ne ratkaistujen siirtojen jonoon. Ratkaistujen siirtojen jonoa lyhennetään vanhemmasta päästä, jotta ratkaisijan siirrot näyttäisivät pelilaudalta järkevältä.

Jokaisen kierroksen aluksi ratkaistessaan peliä ratkaisija päivittää tiedot pelilaudasta, eli hakee avonaisten ruutujen tiedot ja tallentaa ne taulukkoon käyttöön.

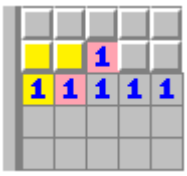
Seuraavaksi ratkaisija etsii täysin varmoja miinoja, jolloin miina ei voi olla muualla kuin ainoassa tai ainoaissa avaamattomissa ruuduissa. Esimerkiksi ruudussa, jonka arvo on 1 ja sen vieressä on vain yksi avaamaton ruutu, ratkaisija tulkitsee avaamattoman ruudun tällöin miinaksi ja liputtaa sen.

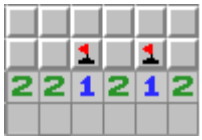
Liputtaminen on ei kuitenkaan ole yksi siirto, vaan se tapahtuu siirtojen ohessa, koska se on vain esteettinen seikka ja auttaa debuggausta. Miinojen löytyessä päivitetään eiLöydettyjenMiinojenmäärä taulukkoon laskemalla tieto löydetyistä miinoista. Kun tieto on nolla, se tarkoittaa sitä että avataanNollat metodi avaa kaikki nolla ruutujen vieressä olevat ruudut, koska ne ovat ilmiselvästi turvallisia avata.

Viereissä kuvassa näkyvät 13231 numerot ovat ratkaisijan tiedoissa 12121, sillä ratkaisija vähentää löydetyn miinan ympäriltä olevista ruuduista löytymättömien ruutujen määrää yhdellä. Tämän ansiosta 121 ja 1221 metodien teko helpottui huomattavasti.

Kun algoritmi jää jumiin eikä löydä varmoja miinoja enää, niin sitten algoritmin viimeinen keino on vielä etsiä löytyisikö 11-, 121- ja 1221-tilanteita pelilaudalta, joiden ratkaisemisen jälkeen normaali tapa ratkaista

peliä toimisi taas. Jos tämäkään keino ei toimi, algoritmini ei osaa ratkaista peliä ilman arvaamista






Aikavaativuus

Ratkaisijan aikavaativuus on O(pelilaudanKorkeus x pelilaudanLeveys x ruudunViereisetRuudut), missä viereistenRuutujen määrä on maksimissaan 8kpl.

Parannettavaa

MiinaharavanRatkaisijan suurin ongelma että ratkaisija metodit käyvät jokaisen ruudun läpi jokaisella kierroksella, vaikka se ei olisi tarpeellista. Jos olisi tiedossa esimerkiksi ArrayListissä kaikki ruudut mitä tarvitsee käydä läpi, raskaimmat metodit eli etsiVarmatMiinat() ja avaaNollat() muuttuisivat kevyemmiksi. Ongelmaa olen jo nyt minimoinut metodien sisällä ruutujen määrää karsivilla if ehdoilla, jotta mahdollismman harvaan ruutuun tulee turhia tarkastuksia ja laskuja.

EtsiVarmatMiinat metodia kutsutaan 70 x 70 kokoisella pelilaudalla noin 15 000 000 kertaa, mutta tarvittavia muutoksia tehdään vain n. 620kpl. Metodia olisi siis mahdollista optimoida jopa 99% nopeammaksi, jos läpikäydyt ruudut olisi tiedossa niin kaikkia ruutuja ei tulisi käytyä jokaisella ratkaisukierroksella läpi uudestan ja uudestaan. Metodin nopeutta voisi nipistää vähän myös lisäämällä taulukkoon tiedon avaamattomien ruutujen määrän, jotta sitä ei tarvitsisi laskea aina uudestaan tarvittaessa.

Miinojen määrää voisi myös hyödyntää viimeisien siirtojen yhteydessä. Sellaisia tilanteita tulee kuitenkin harvoin. Kerran tai kaksi olen törmännyt tilanteeseen jossa siitä olisi ollut hyötyä, eli yleensä nurkkatilanteessa viimeisen miinan yhteydessä.

Algoritmi ei osaa ratkaista kaikkia mahdollisia ratkaistavia tilanteita pelilaudalta, koska en itsekään osaa löytää niitä tai edes ymmärtää miten sellaisia tilanteita tunnistavan metodin tekisin. Posiitivista tässä projektissa on erityisesti ollut että olen oppinut itse pelamaan miinaharavaa paremmin. Minesweeper sivun strategia ohjeista löytyy lisää erikoisemmista ratkaisu tekniikoista. Algoritmin kehittämistä rajoittaa siis oma miinaharavan pelaamistaitoni. Toisaalta näiden tilanteiden määrä on minimaalinen ja algoritmini kattaa ylivoimaisesti yleisimmät tapaukset ja pystyykin parhaimmillaan ratkaisemaan 60% peleistä ilman arvaamista. Erityisesti nurkkiin tai alueisiin missä on paljon miinoja muodostuu helposti 50/50 tilanteita, joissa informaatioita ei ole riittävästi miinojen sijaintien laskemiseen.

Rajoitteet

Toimivuutta testattu Windows 8 ja Ubuntu 12.04 käyttöjärjestelmillä. Toimii Java versiolla 1.7.x tai uudemmalla.