Määrittelydokumentti: Tietorakenteiden harjoitustyö

Aika: Periodi 3, kevät 2018

Tekijä: Oskari Koskinen

Opiskelijanumero: 014708767

Johdanto

Tavoitteeni on ohjelmoida tekoäly, joka pystyisi pelaamaan Carcassonne -pelin (ks. Lähteet) yksinkertaistettua versiota kohtuullisen hyvin. Tässä versiossa laattoja on vain kahdenlaisia: peltoja ja linnoja. Pelin aluksi satunnaiseen kohtaan asetetaan yksi pelto-ruutu. Peli etenee seuraavasti: Arvotaan pakkaan satunnainen määrä linnoja ja peltoja. Pelaajat nostavat niitä vuorotellen ja asettavat pelilaudalle kohtaan, jonka vieressä on linna tai pelto. Samaksi linnaksi lasketaan vierekkäiset linna-ruudut, jotka eivät ole kulmittain toisiinsa nähden. Linna lasketaan valmiiksi, kun se on kokonaan peltojen (kulmittain myös linnat käyvät, sekä ruudut, jotka ovat pelin ulkopuolella) ympäröimä. Pelaaja asettaa vuorollaan yhden laatan. Jos laatta on linna, pelaaja voi asettaa siihen yhden seuraajistaan, mutta vain jos samassa linnassa ei ole jo jonkun pelaajan seuraajaa. Seuraaja vapautuu, kun linna on valmis ja linna pisteytetään. Pelaaja jolla on eniten seuraajia linnassa saa siitä pisteet, ja jos seuraajien määrä on tasan, pisteet jaetaan tasan. Linnat, jotka eivät ole valmiita, pisteytetään pelin lopussa.

Tietorakenteet ja algoritmit

Koodaan ohjelman Java -kielellä. Peliä pelataan kaksiulotteisessa mielivaltaisen kokoisessa matriisissa. Matriisi sisältää Integer -tyypin muuttujia. Numero 0 tarkoittaa, että ruutu on tyhjä, numero 1 tarkoittaa, että paikassa on pelto, numero 2 tarkoittaa, että paikassa on linna. Pelaajilla on Integer -tyypin muuttuja, joka kertoo seuraajien määrän. Lisäksi tarvitaan luokka seuraaja, joka kertoo mille pelaajalle

seuraaja kuuluu, sekä rivi- ja sarakesijainnin. Pelaajien seuraajien sijainnit tallennetaan pelaajille taulukkoihin.

Pelaaja voi pelin alussa määrätä matriisin koon ja kaikkien laattojen yhteenlasketun määrän. Pelin alussa arvotaan satunnainen määrä linnoja ja peltoja satunnaisessa järjestyksessä ja laitetaan pino -tietorakenteeseen. Peliä pelataan, kunnes pino on tyhjä tai kartalla ei ole enää yhtään tyhjää ruutua johon voisi sijoittaa laattoja. Näitä toimenpiteitä varten tarvitaan algoritmi, joka arpoo laattoja ja laittaa ne pinoon, sekä algoritmi joka tarkistaa laudan tilanteen jokaisen siirron jälkeen.

Laattojen arpomisalgoritmi toimii seuraavasti: Arvotaan satunnainen luonnollinen luku väliltä [1,2], jos luku on 1 lisätään pinoon pelto, jos luku on 2 lisätään pinoon linna. Tätä jatketaan, kunnes pinossa on laattoja pelaajan haluama määrä. Lisäksi tarvitaan ruudukon läpikäymiseen algoritmi, joka tarkistaa, että ruudukkoon voi vielä sijoittaa uusia laattoja. Ruudukon läpikäymisalgoritmi käy läpi ruudukon jokaisen siirron jälkeen ja lopettaa pelin, jos se ei löydä yhtäkään ruutua, jossa ei olisi jo linna tai pelto.

Tekoäly toimii seuraavasti: Jos tekoäly nostaa linna-laatan, se etsii suurimman linnansa ja jatkaa sitä, jos mahdollista. Tässä käydään koko matriisin kaikki linnat läpi ja lasketaan verkon leveyssuuntaisella haulla linnan ruutujen määrä. Leveyssuuntaista halua varten tarvitaan tietorakenne jono. Jos tekoälyn ei ole mahdollista jatkaa omaa linnaansa, se etsii toisen pelaajan suurimman linnan ja laittaa laatan kulmittain jonkin tämän linnan palan kanssa, mutta vain jos tekoälyllä on vapaa seuraaja. Jos vapaata seuraajaa ei ole, tekoäly asettaa linna-laatan satunnaiseen vapaaseen ruutuun. Jos tekoäly saa pellon, se etsii matriisin läpikäymällä pienimmän linnansa, jonka koko määritetään niin ikään leveyssuuntaisella haulla, ja asettaa pellon tämän linnan viereen satunnaiseen ruutuun. Jos useampi linna on samankokoinen, tekoäly asettaa pellon sen linnan ympärille, jossa on eniten peltoja.

Peli loppuu, kun laattoja sisältävä pino on tyhjä. Lopuksi linnat joita ei ole vielä pisteytetty, pisteytetään. Pelin aikana kirjaa pisteytetyistä linnoista pidetään erillisessä boolean-matriisissa, jotta samaa linnaruutua ei laskettaisi kahteen kertaan. Pisteytys hoidetaan leveyssuuntaisella haulla, joka laskee linnan laattojen määrän ja linnassa olevat seuraajat. Pelaaja jolla on eniten seuraajia, saa linnasta kaikki pisteet.

Algoritmien aikavaativuudet

Laattojen arpominen pinoon on selvästi O(N), jossa N on laattojen määrä. Silmukan sisältö on vakioaikainen, joten aika riippuu vain laattojen määrästä N. Matriisin tarkistaminen on O(n^2), jossa n on matriisin leveys/korkeus. Tekoälyn laatan asettamisen pahin aikavaativuus on O((n^2)x(V+E)), jossa V on solmujen määrä ja E on reittien määrä. Esim. jokaisesta löydetystä linnasta pitää laskea sen koko ja suurimman linnan määrittämiseksi koko matriisin linnat on käytävä läpi.

Lähteet

https://fi.wikipedia.org/wiki/Carcassonne_(peli)

Tietorakenteet ja algoritmit kurssin (syksy 2017) materiaali