

Määrittelydokumentaatio: opinto-ohjelmalle tietojenkäsittelytieteen kandidaatti

Tulen käyttämään Algoritmit ja tekoäly kurssin harjoitustyössä python ohjelmointikieltä, ja hallitsen itse tarpeeksi hyvin vertaisarviointia varten C#, JS, Python ja Javaa.

Teen 2048 pelin version, jossa tietokone tekee joka toisen siirron, ja joka toinen siirto taas on randomilla tehty siirto. Tekoälyn on tarkoitus minmax algoritmin avulla pärjätä random siirtoja "vastaan".

Ratkaisen miten voin minmax algoritmin avulla saada pelattua 2048 peliä random muuttujan vaikeuttamana.

Ohjelma saa syötteenä käyttäjän hiiren klikkauksen merkkinä seuraavasta siirrosta. Eli kun klikkaat hiirellä, tietokone tekee oman siirtonsa, ja kun klikkaat uudestaan "vastus" eli random siirto tapahtuu. Täten käyttäjän pystyt seuraamaan hyvin pelin tapahtumia vaikkeet itse niihin pääsekään vaikuttamaan.

Minmax on rekursiivisesti toimiva algoritmi, joka tarkastelee useita pelisiirtoja kerrallaan eteenpäin puumaisessa kaaviossa, etsien mikä seuraava mahdollinen siirto johtaisi meidät sellaiseen pelitilanteeseen jonka haaralla on parhaat mahdollisuudet onnistua. Algoritmi on helposti paljon aikaa vaativa. Aikavaativuus riippuu pelin "syvyyden" (kuinka pitkälle minmax puussa voidaan mennä) ja mahdollisten siirtojen määrästä.

Arvoisin aikariippuvuudeksi $O(x^y)$ jossa:

X = haarautumis- aste, tässä pelissä 4, jokaisessa vaiheessa voidaan mennä yhteen neljästä suunnasta

Y = puun syvyys, eli kuinka monta siirtoa eteenpäin peli tutkii, tulee vaihtelevaan pelin edetessä.

Tilavaativuus olisi $O(x*y)$ koska muistiin tullaan tallettamaan kaikki tutkittavat pelitilanteet.

Harjoitustyön ydin on tekoäly, joka osaa pelata 2048 pelin pelilogiikkaa randomsiirtojen vaikeuttamana. Työhön tulee kuulumaan: Pelin perustoteutus, satunnaisten siirtojen logiikka ja tekoälyn toteutus minimax-algoritmilla. Projektia itseään hallitaan GitHub-repositoriossa.

Tässä tekstissä tai työn määrittelyssä ei käytetty kielimalleja.