

Työn aihe: Yatzy-peli ja -tekoäly

Ohjelman rakenne: Game-olio käsittelee pelaajien (abstrakti luokka Player, jonka alaluokkia ovat Bot ja Human) vuoroja. Tekoäly (Bot-olio) pelaa yatzya seuraavasti: kullakin vuorolla luodaan TreeBuilder-olio, joka ottaa heitetyistä nopista 8 lupaavinta kombinaatiota ja luo TreeList-olion, joka sisältää kutakin kombinaatiota vastaavan DecisionTree-olion, joka laskee odotusarvot pisteille, jotka saadaan, kun pidetään osa nopista ja heitetään loput uudestaan. DecisionTree koostuu TreeNodeista. Parhaan odotusarvon tuottava puu kertoo, mitkä nopat lukitaan ja mitkä heitetään uudelleen.

Aika- ja tilavaativuudet: Ohjelman aika- ja tilavaativuuksien kannalta mielenkiintoisin osa on DecisionTree-olioiden luonti ja läpikäyminen. Puiden luonti yksinkertaistettuna pseudokoodiesityksenä:

```
luoNodet(int n)
{
    if (n == 5)
    {
        return;
    }
    for (int i = 1; i < 7; i++)
    {
        for (int j = 0; j < n; j++)
        {
            kopioiNopatTaulukkoon();
        }
        linkitäNoppaEdelliseen();
        luoNodet(n+1);
    }
}
```

N on noppien lukumäärä. Kun noppia on 5, ollaan puun lehdessä.

Sisimmäisin for-luuppi suoritetaan n kertaa, ulompi for-looppi vakiomäärä kertoja. Rekursiokutsujen määrä riippuu lineaarisesti syötteestä. Koko puun luonti vaatii siis $O(n^2)$ aikaa ja $O(n)$ tilaa. Käytännössä tämä ei kuitenkaan ole paljoa, sillä n vaihtelee välillä 1-4.

Puutteet: Tekoäly on vielä kaukana optimaalisesta. Myöskin aikavaativuutta voisi jonkin verran parantaa. Tällä hetkellä kunkin mahdollisen käden odotusarvo lasketaan aina uudestaan, ja saman käden eri permutaatiot lasketaan moneen kertaan. Saattaisi olla järkevää laskea käsien odotusarvot etukäteen ja katsoa ne taulukosta.