

Aineopintojen harjoitustyö: TiRa
Suunnitteludokumentti
Juha Inkinen
01.11.2012

Sisällysluettelo

1. Projektin aihe.....	3
1.1 Alustava logiikka.....	3
2. Käytettävät algoritmit.....	3
3. Syötteet.....	3
4. Tavoitteet.....	3
4.1 Tekoäly.....	3
4.2 Aikavaativuudet.....	3
4.3 Tilavaativuudet.....	3
5. Lähteet.....	4

1. Projektin aihe

Tarkoituksena on toteuttaa tekoäly shakki-peliin.

Käytössä tulee näillä näkymin olemaan shakin pienempi variaatio, luultavasti *Simpler chess* -nimellä tunnettu 6x6 laudalla pelattava versio, jossa jätetään pois 2 sotilasta sekä hevoset molemmilta pelaajilta. [1]

1.1 Alustava logiikka

Käytössä on ruutukeskeinen logiikka, jossa pelilauta tuntee jokaisen nappulan sijainnin ja tyyppin. Pelilauta tuntee myös pelin säännöt ja osaa arvioida onko tuleva siirto laillinen.

Pelilaudan toteutus tulee luultavasti olemaan matriisi-mallinen.

2. Käytettävät algoritmit

Seuraavan siirron valintaan käytetään alpha-beta-kruunausta hyödyntävää MINIMAX-algoritmia, joka tarvitsee toimivan puurakenteen. Puun toteutuksen täytyy sisältää ainakin puun luomiseen vaadittava algoritmi, sekä puun läpikäyntiin käytettävät ominaisuudet.

Lisäksi tarvitaan järkevä tapa rajata arvioitavan siirron laatua, jotta aikavaativuus ei kasva liian suureksi. Luonnollisesti pitää myös laskea siirtojen laatu.

Lienee kannattavaa myös suunnitella alkupeleihin valmiit logiikat, koska pelien aloituksessa on shakki-teorian mukaisesti tunnetusti hyviä aloitustaktiikoita. [2][3] Näin välttyään alkupelissä, jolloin myös vaihtoehtoja on suuri määrä, ylimääräiseltä ”tyhmien” siirtojen laskemiselta.

3. Syötteen

Riippuen siitä, että asetetaanko ohjelma pelaamaan itseään vai pelaajaa vastaan syötteiden määrä voi olla hyvinkin minimaalinen.

4. Tavoitteet

4.1 Tekoäly

Tavoitteena on saada aikaan toimiva tekoäly, joka on kykenevä pelaamaan itseään vastaan järkevästi ja tehokkaasti.

4.2 Aikavaativuudet

Algoritmi tulee vaatimaan eksponentiaalisen määrän aikaa, jotta voidaan huomioida kaikki mahdolliset siirtoyhdistelmät. Näin ollen rajausta tiettyyn syvyyteen on tärkeää ja pyritään luomaan algoritmeja, jotka kykenevät pelaamaan riittävän nopeasti välttämättä häviöt, jotka johtuisivat shakkikellon mittaaman ajan loppumiseen.

4.3 Tilavaativuudet

Puurakenne, johon arviot siirtojen laaduista syötetään voi kasvaa suureksi riippuen kuinka syväksi seuraavan siirron arviointia rajataan.

Lisäksi lienee kannattavaa tallentaa tietyn nappulan/ruudun mahdolliset siirrot, jotta niitä ei

tarvitse laskea jokaisen vuoron alussa uusiksi niiden nappuloiden osalta, joihin ei vaikutettu.

5. Lähteet

- [1] <http://en.wikipedia.org/wiki/Minichess> (31.10.2012)
- [2] <http://fi.wikibooks.org/wiki/Shakki> → Pelipuu (1.11.2012)
- [3] <http://fi.wikipedia.org/wiki/Shakkiavaus> (1.11.2012)