

Testausdokumentti

1. Mitä ja miten on testattu

Tekoälyn pelimenestystä on testattu 1000 kierroksen peleillä joita on toistettu 5 kertaa ja otettu keskiarvo kaikkien pelissä voitettujen ja hävittyjen erien suhteesta.

1.) Yleinen ohjelman tehokkuus:

- 1 000 000 kierrosta ai vs ai peliä (molemmilla 3 strategiaa käytössä) suoriutui keskimäärin noin 900:ssa millisekunnissa.

2.) Algoritmien kykyä pelata toisiaan vastaan ilman metastrategioita:

- MarkovFirst voittaa 60% MarkovSecond:iä vastaan pelatuista kierroksista.
- MarkovSecond voittaa 100% StupidAi:ta vastaan pelatuista kierroksista.
- Muut algoritmit olivat tasavahvuisia keskenään.

3.) Metastrategioiden määrän vaikutus pelimenestykseen toista tekoälyä vastaan:

- Yleisesti ottaen suurempi määrä metastrategioita antaa suuremman todennäköisyyden voittaa pelin
- 1 vs 4-6 metastrategiaa häviää 100% kierroksista.
- 2 vs 5-6 metastrategiaa häviää 80% kierroksista.
- 3 vs 4-6 metastrategiaa häviää hieman yli puolet kierroksista.
- Loput yhdistelmät olivat tasavahvuisia keskenään.

4.) Decayn vaikutus pelimenestykseen toista tekoälyä vastaan:

- Kun kaksi identtistä tekoälyä pelaa keskenään yhdellä metastrategialla, decay:tä käyttävä häviää noin 55% kierroksista.
- Kun kaksi identtistä tekoälyä pelaa keskenään kuudella metastrategialla, decay:tä käyttävä voittaa noin 55% kierroksista
- Johtopäätöksenä voidaan todeta, että decay:n käyttäminen tuo etua kun käytetään monta metastrategiaa.

5.) Eri algoritmiyhdistelmien pelimenestys ihmispelaajaa vastaan

- Automatisoitu testaus vaikeaa, koska tekoälyn ensimmäiset siirrot ovat satunnaisia ja voivat vaikuttavat pelin lopputulokseen yllättävän paljon.
- Tällä hetkellä TestPlayer (pelaa vanhoja pelejä automaattisesti) voittaa lähes aina tekoälyn tietyllä valintayhdistelmällä. Ilmeisesti yksikään strategia ei vielä havaitse toistuvia kuvioita tarpeeksi tehokkaasti.
- MarkovFirst ja MarkovSecond yhdistelmää käyttävä tekoäly voittaa noin 60% itseäni vastaan pelatuista kierroksista ja 70% testihenkilöä vastaan pelatuista kierroksista.
- Yleensä kaikkien kuuden metastrategian käyttö on turhaa ja jopa haitallista, koska tekoäly voi "huomata" että pelaaja:lla olisi jokin monimutkainen strategia käytössä, vaikka todellisuudessa tätä ei olisi tapahtunut. Tällöin eri metastrategioiden välinen vaihtelu voi tuottaa epäloogisia valintoja pelaajan näkökulmasta.

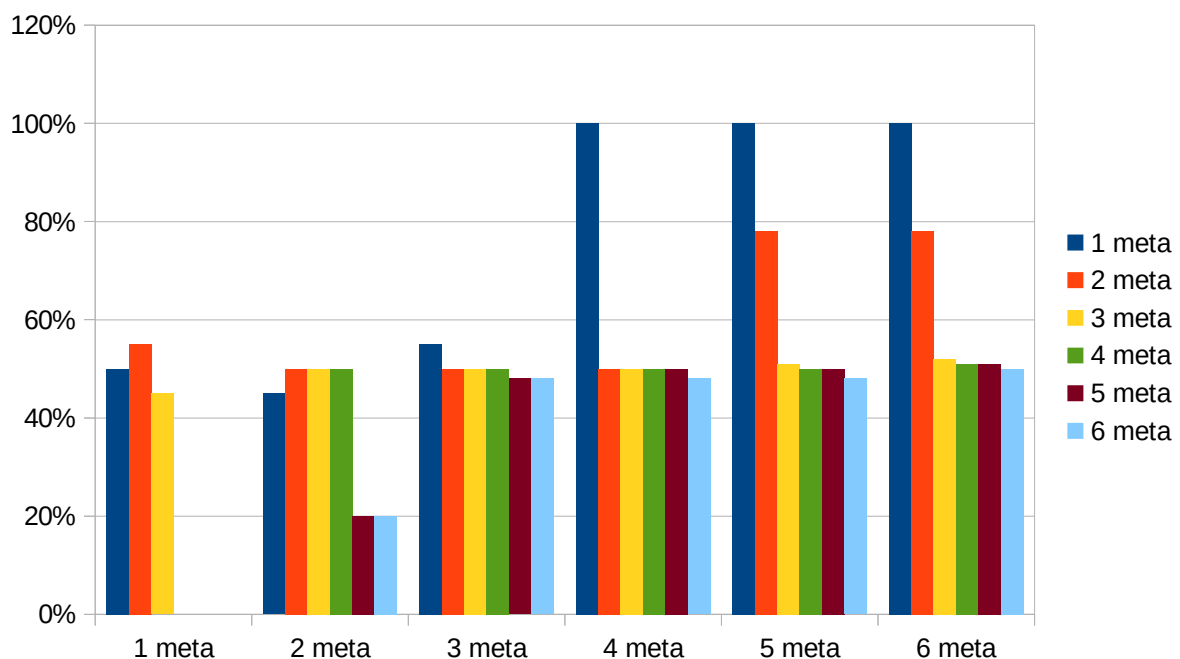
6.) Yksikkötestausta:

- Testattu että markovin malleihin perustuvat algoritmit antavat aina todennäköisimmän valinnan.
- Peli toimii oikein: tarkistaa pelaajien kädet ja päivittää tulokset.
- StrategyHandler pisteyttää strategiat viime kierroksen tulosten perusteella ja osaa valita parhaiten menestyneen metastrategian ehdotuksen.

3. Miten testit voidaan toistaa

Tekoäly vs Tekoäly-testien toisto vaatii Game-luokassa kahden StrategyHandlerin luomisen, tarvittavien parametrien käytön ja tarvittavien algoritmien lisäyksen StrategyHandleriin. Jos haluat testata itse tekoälyä, niin Game-luokassa täytyy lukea "p1 = new Player();" ja "p2 = new StrategyHandler(2, metastrategioiden_määrä, 0.95, onkoDecayPäällä);" sekä lisätä tarvittavat algoritmit p2:lle käsin. Tämän voisi tehdä valikkojen kautta tulevaisuudessa.

4. Tilastoja



Taulukossa tulokset eri metastrategioiden vaikutuksesta voittoprosenttiin kun käytössä on yksi algoritmi. Alhaalla on "pelaajan" metastrategioiden määrä, palkeissa on voittoprosentti eri metastrategioiden määrää vastaan. Esim "4 meta"-kohdasta nähdään kuinka 4 vs 1 metastrategia voittaa aina, koska neljäs metastrategia olettaa, että vastustaja käyttää samaa algoritmia pelaajaa vastaan.