

Alexi Tamminen

# Hyvä tekoäly reaaliaikaisissa strategiapeleissä

Tietotekniikan kandidaatintutkielma

26. toukokuuta 2018

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

**Tekijä:** Aleksi Tamminen

**Yhteystiedot:** aleksi.a.k.tamminen@student.jyu.fi

**Työn nimi:** Hyvä tekoäly reaaliaikaisissa strategiapeleissä

**Title in English:** good artificial intelligence in real-time strategy games

**Työ:** Kandidaatintutkielma

**Sivumäärä:** 21+0

**Tiivistelmä:** Vaikka reaaliaikaiset strategiapelit ovat olleet olemassa jo pitkän ajan ei niiden tekoälyssä ole saavutettu tilaa mihin voisimme olla kunnolla tyytyväisiä. Tämän kandidaatintutkielman tavoitteena on tutkia, miten voitaisiin luoda hyvä tekoäly reaaliaikaisiin strategiapeleihin ja kuinka tekoäly mahdollisesti paranee tulevaisuudessa. Kandidaatintutkielman muotona on kirjallisuuskatsaus. Tässä tutkielmassa käymme läpi RTS-pelien tekoälyn osa-alueita ja ongelmia, jotka liittyvät näihin osa-alueisiin. Lopuksi käymme läpi koneoppimistekniikoilla suoritettuja tutkimuksia, joita voitaisiin mahdollisesti käyttää parantamaan RTS-pelien tekoälyä tulevaisuudessa.

**Avainsanat:** tekoäly, reaaliaikainen strategiapeli

**Abstract:** Even though real-time strategy games have been around for a long time we still have not achieved a state where we can be content. The goal of this bachelors thesis is to research how to create a good AI to real-time strategy games and how the AI will improve in the future. Form of the bachelors thesis is a literacy review. In this study we will go through some of the parts of RTS-game AI and problems that are included in these parts. In the end we will go through machine learning studies that could possibly be used to improve RTS-game AI in the future.

**Keywords:** artificial intelligence ,real-time strategy games

## Kuviot

- Kuvio 1. (Vasemmalla puolella kuvaa on pelimaailma, mikä näkyy vaikutuskartalle oikeanpuoleisena kuvana .(Liu, Louis ja Ballinger, 2014). Vaikutuskartalla näkyy himmeänä valona pelikartan rajat ja kirkkaalla valolla vihollisen yksiköt.)..... 4
- Kuvio 2. RTS-pelin tekoälyn tasot (Liu, Louis ja Ballinger, 2014). Mikromanagerointi voidaan jakaa reaktioiden hallintaan (engl. reaction control) ja mikromanageroinnin taktiikoihin. Makromanagerointi koostuu yleisestä strategiasta ja siihen sisältyy yksiköiden parannukset ja pelialueen tutkiminen. Makromanageroinnin lopputuloksena on pelaajan armeija, mikä voi jakaa miten pelaaja näkee parhaaksi. Ontañón ym (2013) jakavat tehtävät suuresti samalla tavalla. ....11

## Sisältö

1	JOHDANTO .....	1
2	REAALIAIKAISTEN STRATEGIAPELIEN TEKOÄLY .....	3
	2.1 Reitinhaku .....	3
	2.2 Ryhmässä liikkuminen .....	3
	2.3 Taktinen- ja strateginen tekoäly .....	4
	2.4 Päätöksen teko .....	5
3	MITEN REALIAIKAISTEN STRATEGIAPELIEN TEKOÄLYÄ VOIDAAN PARANTAA TULEVAISUUDESSA .....	7
4	KONEOPPIVA TEKOÄLY JA SEN HYÖDYNTÄMINEN .....	10
	4.1 Mikromanagerointi .....	11
	4.2 Makromanagerointi .....	12
5	YHTEENVETO .....	14
	KIRJALLISUUTTA .....	16

# 1 Johdanto

Tässä kandidatuksielmassa tutkitaan, miten voitaisiin luoda hyvä tekoäly reaaliaikaisiin strategiapeleihin ja kuinka tekoäly mahdollisesti paranee tulevaisuudessa. Reaaliaikaisista strategiapeleistä käytetään lyhennettä RTS-peli (engl. Real-time strategy game) ja tätä lyhennettä käytetään myös tässä tutkielmassa. Kandidatuksielman muotona on kirjallisuuskatsaus.

Tekoäly on laaja käsite ja sen merkitys riippuu paljon kontekstista. Tässä tutkielmassa tekoälyllä tarkoitetaan RTS-pelien sisältämää tekoälyä. RTS-pelien tekoälyyn luetaan mukaan kaikki toiminnot, jotka tarvitsevat tekoälyä toimiakseen tai toimivat tekoälyn ohjaamana, esimerkiksi pelien sisällä olevien ohjattavien yksiköiden tekoäly ja tietokonevastustajan eli botin käyttämä tekoäly. Tietokonevastustaja tekee tekoälyn avulla päätöksiä monella tasolla ohjatessaan hallitsemansa osapuolen yksiköitä, kun taas pelin sisällä oleva yksikkö tekee päätöksiä automaattisesti tilanteen vaatiessa.

RTS-pelien tekoälyä tulisi tutkia, koska hyvän tekoälyn luominen RTS-peleihin on moneen muuhun peligenreen verrattuna paljon vaikeampaa. Esimerkiksi vuoropohjaisissa strategiapeleissä tietokonevastustaja tekee päätöksiä ja liikkeitä vain omalla vuorollaan, kun taas RTS-peleissä tietokonevastustajan täytyy tehdä päätöksiä koko pelin ajan. Videopeliyhtiöt eivät koe tarpeelliseksi kehittää tekoälyä, koska se vaatii paljon aikaa ja resursseja. Hyvä tietokonevastustaja kuitenkin auttaisi lisäämään tulevaisuuden RTS-pelien suosiota. Nykyinen tekoäly RTS-peleissä ei ole kovin kehittynyt ja sitä olisi joillain osa-alueilla helppo parantaa (Buro, 2004). Kehitystä RTS-pelien tekoälyssä on kuitenkin saavutettu Buron artikkelin jälkeen erilaisten kilpailujen avulla. Näitä kilpailuja on pidetty käyttäen esimerkiksi Blizzard Entertainmentin StarCraft peliä alustana (Ontañón ym. 2013).

RTS-peleissä pelaajat käyttävät resurssien keräystä, tukikohdan rakentamista, teknologioiden kehittämistä ja yksiköiden ohjaamista voittaakseen vastustajansa pelimaailmassa, joka on usein jonkinlaisessa sotatilassa (Hagelbäck ja Johansson, 2008).

Reaaliaikaisten strategiapeliä nykyisellä tekoälyllä varustettu tietokonevastustaja saadaan vaikeammaksi antamalla tälle ylimääräisiä etuja. Näin pelaajalle saadaan helposti vaikeammin voitettava vastustaja pelin vaikeusasteen noustessa. Yleisimpiin etuihin kuuluvat nopeampi resurssien keräys ja kartan paljastaminen tietokonevastustajalle.

Luvussa 2 käydään läpi tärkeitä RTS-peleihin kuuluvia tekoälyn osia ja kuinka ne yleensä toimivat. Luvussa 3 tutkitaan, miten reaaliaikaisten strategiapeliä tekoälyä pyritään nykyään parantamaan ja kuinka se tulee vaikuttamaan tulevaisuuden RTS-peleihin. Luvussa 4 käydään läpi koneoppivan tekoälyn hyötyjä ja kuinka sitä voitaisiin käyttää parantamaan RTS-peliä käyttämää tekoälyä.

## 2 Reaaliaikaisten strategiapelien tekoöly

RTS-pelien tekoölyn tärkeimmiksi osa-alueiksi voidaan lukea ainakin reitinhaku (engl. pathfinding), ryhmässä liikkuminen (engl. group movement), taktinen- ja strateginen tekoöly (engl. tactical and strategic AI) ja päätöksenteko (eng decision making) (Millington ja Funge, 2009).

### 2.1 Reitinhaku

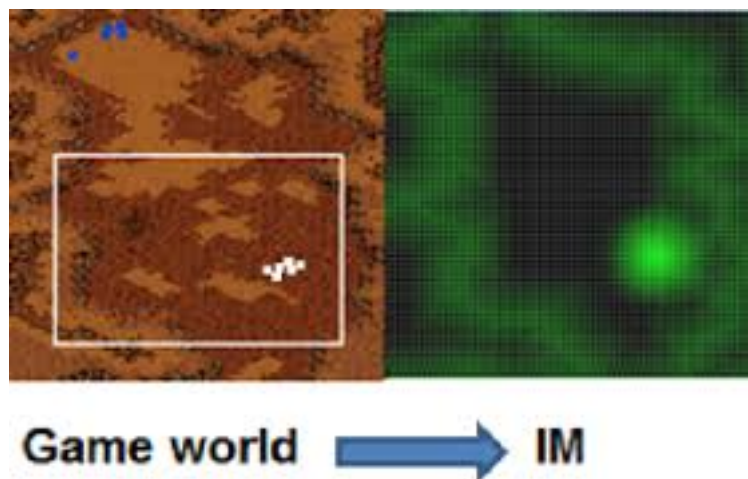
Reitinhaussa etsitään pelikartalta reitti paikasta A paikkaan B. Reitin tulee olla mahdollisimman hyvä, mikä yleensä tarkoittaa nopeinta reittiä. Reitinhaku toimii käyttäen reitinhakualgoritmeja. Yleensä RTS-pelien navigoinnin hoitaa reitinhakualgoritmit kuten A\*, joka löytää parhaan mahdollisen reitin kahden kohteen välillä tarpeeksi lyhyessä ajassa. RTS-pelien dynaamiset pelimaailmat ovat vaikeita käsitellä, mikä tuottaa reitinhakualgoritmeille ongelmia (Hagelbäck, 2016). Reitinhaku ja reitinhakualgortimit vaihtelevat peleittäin ja peligenreittäin. RTS-peleissä maailma jaetaan yleensä ruudukkoihin (engl. grid) ja kartat sisältävät usein kymmeniä tuhansia palasia (engl. tiles). Pelaajan täytyy pystyä siirtämään monia yksiköitä koko kartan halki, minkä takia vanhemmissa RTS- peleissä tehokas reitinhaku oli suurin haaste (Millington ja Funge, 2009). Näille ruudukoille annetaan arvo perustuen siihen, miten helppoa sen kautta on kulkea. Järjestelmän kehittyneisyydestä riippuen reitinhakualgoritmi saattaa ottaa huomioon asioita, kuten vesistöt ja pelikartan korkeuserojen vaikutukset (Schwab, 2009).

### 2.2 Ryhmässä liikkuminen

Ryhmässä liikkuminen tarkoittaa pelissä liikutettavien yksikköjen liikuttamista yhtenä ryhmänä. RTS-peleissä ryhmässä liikkuminen on yleensä tärkeää ja joissain jopa automaattista. Muita tapoja saada aikaan helpompi yksiköiden liikuttaminen ryhmässä, jos se ei pelissä ole automaattista, olisi laittaa peliin ryhmäytymisnappula tai muodostelmajärjestelmä. Ryhmäytymisnappulalla pelaajan valitsemat yk-

siköt menevät samaan ryhmään, kun taas muodostelmajärjestelmässä pelaajan valitseman ryhmän yksiköt menevät ennalta määrättyihin muodostelmiin (Millington ja Funge, 2009). Esimerkiksi pelissä "Age of Empires 2" on neljä eri muodostelmaa valittavissa. Muodostelma otetaan käyttöön kun vähintään 2 yksikköä on samassa ryhmässä. Yksiköt jotka ovat ryhmässä liikkuvat samaa nopeutta kuin ryhmän hitain jäsen. Normaali muodostelmana on valittuna linjamuodostelma, joka muodostuu siten, että edessä on ratsuväki toisessa linjassa on jalkaväki ja takana on ampujaseiteitä käyttävät yksiköt. Toinen muodostelma on neliö missä keskelle menevät heikot yksiköt. Kolmas on hajaantunut muodostelma, joka toimii kuten linja mutta yksiköt ottavat väliä toisiinsa. Neljäs muodostelma on sivusta (eng flank) muodostelma missä valitun ryhmän yksiköt jakautuvat kahteen pienempään ryhmään. Ryhmässä liikumisen esimerkkinä pelissä "Stellaris" pelaajan armeija jakautuu laivastoihin, joiden alukset liikkuvat yhdessä ryhmässä mutta eivät voi erikseen valita muodostelmaa.

### 2.3 Taktinen- ja strateginen tekoäly



Kuvio 1. (Vasemmalla puolella kuvaa on pelimaailma, mikä näkyy vaikutuskartalle oikeanpuoleisena kuvana. (Liu, Louis ja Ballinger, 2014). Vaikutuskartalla näkyy himmeänä valona pelikartan rajat ja kirkkaalla valolla vihollisen yksiköt.)

Taktista- ja strategista tekoälyä on tähän mennessä käytetty enimmäkseen ohjaa-



maan reitinhakua. Toinen asia mihin sitä käytetään on rakennusten sijaintien valitseminen, missä käytetään myös apuna vaikutuskartoitusta (eng influence mapping) (Millington ja Funge 2009). Vaikutuskartalta tekoäly tunnistaa turvallisimmat paikat rakentaa tärkeitä rakennukset. Millington ja Funge (2009) kertovat myös, että RTS-peleissä missä on mahdollista rakentaa muureja tekoäly tutkii niille sopivan sijainnin vaikutuskartalta. Vaikutuskartta on koordinaatisto, joka on laitettu pelimaailman päälle. Sen yksityisille osille, neliöille (square), annetaan vaikutuskartan funktiolla jokin arvo. Vaikutuskarttaa käytetään myös tilallisten ongelmien (engl. spatial problem) ratkaisuun (Liu, Louis ja Ballinger, 2014).

## 2.4 Päätöksen teko

Varsinkin RTS-peleissä päätöksien tekeminen ei ole helppoa, koska koko peli pyörii pelikartalta kerättyjen resurssien ympärillä ja melkein kaikki mitä tekoäly voi tehdä käyttää samoja resursseja. Tekoälyn pitää tietää milloin käyttää resursseja mihinkin asiaan. Milloin ja mihin se rakentaa rakennukset, milloin tutkitaan lisää teknologioita, milloin hyökätään tai puolustetaan ja mitä yksiköitä pitää tilata. RTS-peleissä on monta kohtaa milloin päätöksen teon tulee tapahtua, minkä takia melkein aina tarvitaan moniosainen tekoäly (Millington ja Funge, 2009).

Moniosainen tekoäly on käytössä tietokonevastustajan korkeantason strategisessa tekoälyssä. Tämä korkean tason tekoäly vastaa tietokoneen johtaman maan resursseista. Tekoäly vaihtelee resurssienhallintaa puolustuksen, hyökkäyksen, talouden ja tutkimusten välillä. Korkean tason strategista tekoälyä muokaten voidaan myös helposti luoda tekoäly vastustajista erilaisia. Esimerkiksi antamalla aggressiiviselle maalle X tekoäly, joka näkee parhaaksi tuottaa enemmän sotilaita, kun taas taloudellinen maa Y näkee parhaaksi tehdä vahva talous ennen hyökkäämistä (Schwab, 2009).

Yksinkertaisimmat tekoälyn toiminnat tekevät yksiköt itsekseen. Tämän voi tehdä esimerkiksi yksikkö tai ryhmäpohjaisilla käskyillä. Esimerkiksi monissa peleissä yksikön voi käskä olemaan aggressiivinen, passiivinen tai puolustautuva. Keskitä-

soisiin päätösiin voidaan lukea esimerkiksi yhden ryhmän käyttäytyminen, jolloin kaikki ryhmän jäsenet tekevät ryhmän valitseman toimenpiteen (Millington ja Funge, 2009). Ryhmäpohjaisia käskyjä on esimerkiksi liiku ja hyökkää, jolloin ryhmä liikkuu ja hyökkää kaikkia vastaan tulevia vihollisia vastaan.

Päätöksen teon vaikeuden takia tärkeimmillä RTS-pelin elementeillä on oma tekoäly, jonka monimutkaisuus riippuu pelistä itsestään. Joissain peleissä nämä osaset ovat melko itsenäisiä. Esimerkiksi resursseista vastaava tekoäly ei mieti muita osia vaan kerää kaikkia resursseja mahdollisimman hyvin (Millington ja Funge, 2009). Peleissä joissa tekoäly on yhtenäisempi kaikki tekoälyn osat pyrkivät toimimaan yhteen, esimerkiksi keräämällä enemmän resursseja yksiköihin ja rakentamalla vähemmän (Millington ja Funge, 2009).

### **3 Miten reaaliaikaisten strategiapeliin tekoälyä voidaan parantaa tulevaisuudessa**

Jotta RTS pelien tekoälyä saadaan parannettua tulevaisuudessa, tulee meidän miettiä mitä parannettavaa meillä on nykyisestä tilanteesta. Nykyaikainen tekoäly ei ole kovin hienovarainen, eikä se pysty pelaajiin verrattuna tekemään tärkeitä valintoja, jotka tekevät pelaajia vastaan pelatuista peleistä haastavampia ja nautinnollisempia. RTS-pelien tulevaisuuden kannalta on tärkeää pyrkiä saamaan tekoäly tekemään enemmän valintoja taktiikoissaan ja pelityylissään pelin keskellä. Ontañón ym (2013) jakavat RTS-pelien tekoälyn haastavimmat osa-alueet kuuteen ryhmään. Nämä ryhmät ovat pelissä olevien resurssien käyttö (engl. resource management), valintojen tekeminen epävarmuuden alaisena, ajallinen ja tilallinen järjestyminen, usean tekoälyn yhteistyö, vastustajaan mukautuminen ja oppiminen ja vastakkainen reaaliaikainen suunnittelu (eng. Adversarial real-time planning). Buro ja Furtak (2004) mainitsivat myös reitinhaun yhdeksi ongelmaksi näiden kuuden lisäksi. Reitinhaun on kuitenkin parantunut huomattavasti, joten sitä ei lasketa nykyään suureksi ongelmaksi.

Resurssien käytöllä on suurin vaikutus pelistrategian onnistumiseen, jos pelaaja tai tekoäly ei osaa käyttää resursseja oikeisiin asioihin oikeaan aikaan on häviäminen paljon todennäköisempää. Buro ja Furtak (2004) antavat esimerkkinä tilanteen, missä pelaaja käyttää liian paljon resursseja teknologioiden parantamiseen, mistä johtuen pelaaja ei voi tehdä niin paljon yksiköitä ja hänen puolustuksensa kärsii.

Valintojen tekeminen epävarmuuden alaisena tarkoittaa valintojen tekemistä ilman tarkkaa tietoa. Buro ja Furtak (2004) antavat tästä esimerkkinä pelitilanteen, missä pelaajat eivät tiedä toistensa tukikohtien sijaintia ja joutuvat tekemään päätelmiä siitä, mitä toinen on tekemässä. Ontañón ym (2013) jakavat epävarmuuden kahden osaan, kartan näkyvyyteen ja RTS-pelien vastustaja painoitteisuuteen. Kartan näkyvyysongelmaa voi korjata tutkimalla pelikarttaa ja vetämällä johtopäätöksiä löydetyistä asioista. Vastustaja painoitteisuus tarkoittaa, että peliä pelataan jotakin

vastaan, mikä johtaa tarpeeseen vetää johtopäätöksiä vihollisen tilanteesta samalla tavalla kuin pelaajakin tekee.

Vastasuunnitteluun kuuluu tärkeitä korkeantason valintoja, kuten milloin rakentaa mitä rakennuksia, milloin tutkia teknologioita, missä edetä ja hyökätä. Buro (2004) mukaan jotta tulevaisuuden tekoäly saataisiin niin hyväksi, että se pystyisi voittamaan pelaajan komentotasolla, tulisi sen osata tehdä valintoja dynaamisessa pelimaailmassa. Buro ja Furtak (2004) myös lisäävät, että RTS-pelin pelimaailma tulisi muokata jonkinlaiseen abstraktiin tilaan, missä tekoäly voisi ratkaista ongelmia ja muokata löydetty ratkaisut jonoksi toimintoja, mitkä tekoäly sitten suorittaisi pelissä. Buro ja Furtak (2004) antavat tästä esimerkki ongelman, missä tietokonevastustaja ei ymmärrä hakata puita päästäkseen hakemaan lisää resursseja, kun taas pelaaja ymmärtää tämän helposti.

Vastustajaan mukautuminen ja yleinen oppiminen on yksi suurimmista tekoälyn ongelmista. Uusia koneoppimistekniikoita tulisi kehittämään parantamaan tekoälyn kykyä vastaamaan pelaajan taitoa oppia vihollisen tekniikoista nopeasti ja käyttämään kyseisen strategian heikkouksia pelaajaa vastaan (Buro 2004). Esimerkiksi jos pelaaja rakentaa paljon tankkeja tulisi tekoälyn pystyä vaihtamaan resurssien käytön tankkien vastaisten yksiköiden rakentamiseen. Tekoäly-yhteisöissä on ollut jo pitkään keskustelua taktisen analyysin käytöstä. Käyttäen taktista analyysiä tekoäly voisi suunnitella laajoja yksiköiden liikkeitä ja voisi löytää heikkoja kohtia vihollisen muodostelmasta ja käyttämään niitä hyväkseen (Millington ja Funge 2009).

Ajallinen ja tilallinen järkeily on vaikeaa tekoälylle, koska RTS pelit tapahtuvat reaaliajassa eli tilanne voi vaihtua erittäin nopeasti. Staattisen ympäristön kuten kuristuskohdat tai dynaamiset tilalliset ominaisuudet kuten näkyvyys ja vihollisen vaikutus ovat erittäin tärkeitä asioita, mitkä pitää ottaa huomioon pelitaktikkaa luodessaan RTS pelissä. Nykyiset pelin tekoälyt jättävät nämä asiat monesti huomioimatta, eivätkä pärjää yleisjärjen päättelylle (Buro ja Furtak, 2004). Ontañón ym (2013) lisäävät tähän ongelmaan myös ainakin tukikohdan laajennuksen ja puolustusten rakentamisen. Ajallisesta järkeily on myös tärkeää strategisella tasolla. Ontañón ym (2013) mainitsevat strategisen tason asioista pitkän aikavälin suunnitelmat, jotka ot-

tavat huomioon talouden, teknologiat, rakennusten rakentamisen ja strategian vaihtamisen. Liu, Louis ja Ballinger (2014) käyttivät tutkimuksessaan vaikutuskarttoja hoitamaan tilallisia ongelmia.

Tekoälyn yhteistyö pitää sisällään tietokonevastustajien yhteistyön keskenään tai pelaajien kanssa. Buro ja Furtak (2004) antavat tästä esimerkkinä tilanteen, missä tekoäly pelaa ihmisen kanssa samalla puolella mutta ei hyökkää samaan aikaan pelaajan kanssa. Buro ja Furtak (2004) antavat tähän syyksi sen, että tekoäly ei tutki mitä ihminen tekee. Tulevaisuudessa tekoälylle voisi esimerkiksi ilmoittaa pelaajalle, kun se on valmis hyökkäämään tai pelaaja voisi ohjata tekoälyä toimimaan puolustautuvasti.

RTS-pelin maailmassa tapahtuu paljon asioita, jotka vaikeuttavat reitinhakua. Esimerkiksi haeutulle reitille saattaa tulla liikkuva este mikä estää reitin käytön. A\* algoritmille onkin esitetty ehdotus, että lisättäisiin aika dimensio minkä yksiköt varaisivat tietyksi ajaksi helpottaen liikkumista dynaamisessa pelimaailmassa (Hagelbäck 2016).

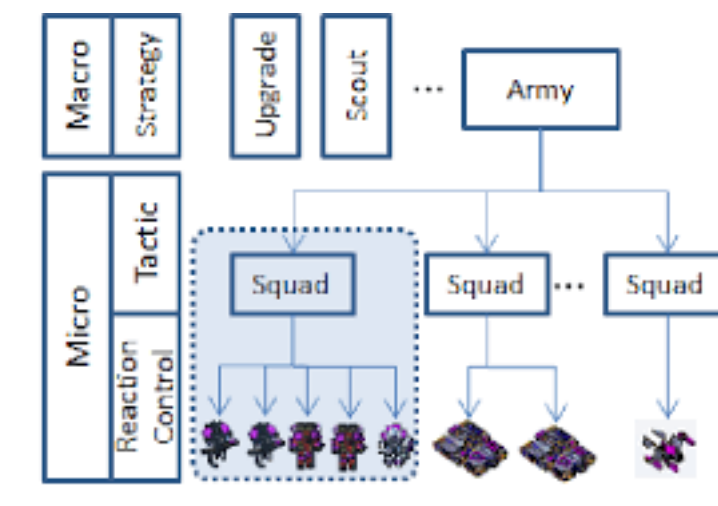
## 4 Koneoppiva tekoäly ja sen hyödyntäminen

Alpaydin (2014) mukaan koneoppiminen on tietokoneiden ohjelmoimista optimoimaan suoriutumisen käyttäen esimerkkitietoa tai aikaisempaa kokemusta. Meillä on malli, jolle annetaan joitain parametrejä ja oppiminen on tietokoneohjelman toteuttama, mallissa olevien parametrien optimointi käyttäen harjoitusdataa tai aikaisempaa kokemusta. Malli mitä tullaan optimoimaan voi olla ennustava, joka pyrkii tekemään oletuksia tulevaisuudesta, tai kuvaileva, joka pyrkii keräämään tietoa datasta. Malli voi olla myös kumpaakin edellämainittua. Alpaydinin (2014) käyttämä optimointi termi tarkoittaa, että mallin opittuaan sen esitys ja algoritminen ratkaisu päätelmään on myös tehokas. Alpaydin (2014) kirjoittaa myös, että joissain tapauksissa oppimisen tehokkuus tai päättely algoritmin ja varsinkin sen ajallinen ja avarudellinen monimutkaisuus voi olla yhtä tärkeää kuin se kuinka tarkkoja sen ennustukset ovat. Koneoppiminen ei ole myöskään pelkkä tietokanta ongelma vaan se on myös osa tekoälyä. Ollakseen järkevä järjestelmän, joka on muuttuvassa ympäristössä pitäisi osata oppia (Alpaydin 2014, sivu 46).

Ponsen (2004) ja Manslow (2002) jakavat oppimisen suoraan ja epäsuoraan oppimiseen. Epäsuorassa oppimisessa muutos käyttäytymiseen perustuu pelimaailman statistiikkoihin. Suorassa oppimisessa optimointialgoritmit tai vahvistusoppimisalgoritmit (engl. reinforcement learning) muuttavat suoraan tekoälyn toimintaa, perustuen suoriutumiseen pelimaailmassa (Manslow, 2002). Ponsenin (2004) mukaan peliyhtiöt ovat kuitenkin varovaisia koneoppimisen kanssa peleissään, koska se on raskasta järjestelmälle, se voi oppia väärin ja sen muokkaaminen saamaan oikeanlainen tulos on vaikeaa. Ponsen (2004) kuitenkin kirjoittaa koneoppimisen kyvystä tehdä tekoälystä parempi ja tätä peliyhteisökin kaipaisi. Manslow (2002) kirjoittaa koneoppimisen hyödyistä ja mainitsee pelien kestävyuden ja pitkäaikaisuuden parantamisen. Ponsen (2004) nimeää koneoppimistekniikoista dynaamisen skriptauksen (engl. dynamic scripting) ja evoluutio algoritmit (engl. evolutionary algorithms)

RTS-pelien tekoälyt voidaan jakaa kahteen osaan, mikromanagerointiin ja makromanagerointiin. Mikromanagerointi on yksiköiden käyttäminen taisteluissa. Mak-

romanagerointiin kuuluu resurssien käyttö eli pelin talous. RTS-peleissä on useasti jonkinlainen resurssienkerääjä yksikkö, joka ei pysty taistelemaan kovin hyvin. Työskentelijä yksikkö yleensä myös toimii tukikohdan rakentajana. Taisteluista vastaavat yksiköt voidaan jakaa eri tavoin riippuen pelin aikakaudesta. Yksiköt voivat olla esimerkiksi jalkaväkeä, lentäviä, tankkeja, vedessä kulkevia tai tykistö yksiköitä.



Kuvio 2. RTS-pelin tekoälyn tasot (Liu, Louis ja Ballinger, 2014). Mikromanagerointi voidaan jakaa reaktioiden hallintaan (engl. reaction control) ja mikromanageroinnin taktiikoihin. Makromanagerointi koostuu yleisestä strategiasta ja siihen sisältyy yksiköiden parannukset ja pelialueen tutkiminen. Makromanageroinnin lopputuloksena on pelaajan armeija, minkä voi jakaa miten pelaaja näkee parhaaksi. Ontañón ym (2013) jakavat tehtävät suuresti samalla tavalla.

## 4.1 Mikromanagerointi

Mitä voidaan tehdä tekoälyn mikromanageroinnin parantamiseksi? Liu, Louis ja Ballinger (2014) tutkivat kuinka geneettisiä algoritmeja voidaan käyttää löytämään strategia voittoon taistelussa. Yksiköiden mikromanagerointi taistelussa pyrkii maksimoimaan viholliselle tehdyn vahinkon ja minimoimaan omille tehdyn vahingon. Yleisesti käytettyjä mikromanagerointi tekniikoita ovat yksiköiden tulen ohjaaminen samaan kohteeseen ja haavoittuneiden yksiköiden pois vetäminen, sekä etäi-

syyden pitäminen viholliseen, jolla on lyhyempi ampumisetäisyys (engl. kiting).

RTS-peleissä on tärkeää tietää missä on paras paikka taistella ja mitkä taistelut pystyy voittamaan. Aikaisemmin mainitsemaani vaikutuskarttaa voidaan käyttää tässä hyödyksi. Liu, Louis ja Ballinger (2014) käyttivät kahta eri vaikutuskarttaa. Toinen laski vihollisten yksiköiden lukumääriä ja merkkasi vaaralliset alueet ja toinen vaikutuskartta merkkasi pelin ympäristöä. Näiden kahden kartan yhdistelmää käytettiin sitten ohjaamaan tekoälyn yksiköiden sijoittelua ja reaktioita pelissä.

Szczepański ja Aamodt (2009) pyrkivät parantamaan mikromanagerointia käyttäen tilannekohtaista järkeilyä (engl. Case-based reasoning CBR) Warcraft 3 pelissä. He pyrkivät voittamaan pelissä jo olleen tekoälyn ja myös tutkimaan, miten CBR voisi auttaa ihmis pelaajia. Algorimina toimi lähin naapuri algoritmi, mutta ilman tavoitteita. Järjestelmän algoritmissä oli viisi kohtaa. Aluksi tallennettiin pelitilanne uudeksi ratkaisemattomaksi tilanteeksi. Toisena verrattiin uutta tilanetta muistissa jo oleviin tilanteisiin ja haettiin kaikista samanlaisin tilanne. Kolmantena kartoitetaan ratkaisu valitusta tilanteesta ja suoritetaan haettu tilanne. Neljännessä kohdassa lisätään uusi tilanne järjestelmään, jos ollaan harjoitustilassa ja ongelma on havainnoitu. Lopuksi odotetaan sekuntti ja palataan ensimmäiseen kohtaan.

## 4.2 Makromanagerointi

Monissa RTS-peleissä on pelitallennus (engl. replay) järjestelmä mistä pelaaja voi käydä katsomassa aikaisemmin pelattuja pelejään. Tätä replay järjestelmää voitaisiin mahdollisesti käyttää opettamaan tekoälyä pelaamaan paremmin. Hsieh ja Sun (2008) analysoivat tutkimuksessaan pelaajien pelitallennuksia ja tutkivat niistä rakennusten rakennusjärjestyksiä ja yksiköiden rakennus strategioita. Tämä on mahdollista, koska pelitalennukset tallentavat pelaajan toiminnot pelitallennus lokiin (engl. replay log). Hsieh ja Sun (2008) käyttivät järjestelmässään tilannekohtaista järkeilyä opettamaan tekoälyä pelaamaan.

Aha ym (2005) tutkivat suunnitelman hakemisalgoritmia, jolla on kolme tietolähdettä. Tilannekohtaisen strategian valinta toimii tilannekohtaisen taktikon (engl. Case-



based Tactician CaT ) avulla. CaT hakee hilasta uuden tilanteen pelin alkaessa ja pelin siirtyessä uuteen vaiheeseen, tarkoittaen tietyn rakennuksen valmistumista. Uuden tilan alussa CaT tallentaa muistiin vihollisen työntekijöiden ja taisteluyksiköiden kuolemat vähennettynä omistakuolemistaan, vihollisen menettämät rakennukset vähennettynä omista tuhoutuneista rakennuksistaan, vihollisen rakentamien rakennusten määrän pelin aikana, vihollisen tekemien taisteluyksiköiden määrän pelin aikana, vihollisen tekemien työntekijöiden määrän pelin aikana, itsensä rakentamien ehjinä olevien rakennusten määrän, itsellään elossa olevien taisteluyksiköiden määrän ja itsellään elossa olevien työntekijöiden määrän. Näiden lukujen avulla valitaan parhaiten toiminut tai mahdollisesti parhaiten toimiva strategia seuraavaan vaiheeseen. CaT käyttää muokattua k-lähin naapuri funktiota (engl. k-nearest neighbor function).

## 5 Yhteenveto

Tämän kandidutkielman muotona on kirjallisuuskatsaus. Tutkielmassa tekoälyllä tarkoitetaan realiaikaisten strategiapeliä eli RTS-pelien sisältämää tekoälyä mihin luetaan kaikki toiminnot, jotka tarvitsevat tekoälyä toimiakseen. RTS-peleissä pelaajat käyttävät resurssien keräystä, tukikohdan rakentamista, teknologioiden kehittämistä ja yksiköiden ohjaamista voittaakseen vastustajansa pelimaailmassa, joka on usein jonkinlaisessa sotatilassa (Hagelbäck ja Johansson, 2008). Tutkielman päätaivite on tutkia kuinka saataisiin hyvä tekoälyvastustaja tulevaisuuden RTS-peleihin. Hyvä tekoälyvastustaja parantaisi RTS-pelien suosiota ja parantaisi näiden pelien elinikää. Tekoälyä on parannettu varsinkin erinäisten kilpailujen avulla, mutta kaupallisiin peleihin ei olla saatu tekoälyvastustajaa, jota ei joutuisi parantamaan antamalla tälle ylimääräisiä etuja.

RTS-pelien tekoälyn tärkeimmiksi osa-alueiksi voidaan lukea ainakin reitinhaku, ryhmässä liikkuminen, taktinen- ja strateginen tekoäly ja päätöksenteko (Millington ja Funge, 2009). Reitinhaku toimii reitinhakualgoritmeilla, jotka etsivät pelikartalla reitin paikasta A paikkaan B. RTS-peleissä pelimaailma on yleensä jaettu ruudukkoihin, joille annetaan arvo riippuen kuinka helppo sen kautta on kulkea. Ruudukot jaetaan vielä pienempiin palasiin, joita kutsutaan tileiksi. Ryhmässä liikkuminen voidaan toteuttaa esimerkiksi ryhmäytymisnapilla tai muodostelmilla. Joskus ryhmäytyminen on automaattista. Taktinen- ja strateginen tekoäly käytetään enimmäkseen ohjaamaan reitinhakua ja rakennusten sijaintien valintaa. Tässä käytetään apuna vaikutuskarttoja. Päätöksen teko on vaikeaa RTS-peleissä ja tämän takia tekoäly on usein moniosainen. Korkeimman tason tekoäly hoitaa tietokonevastustajan resursseja ja Macromanagerointia. Toinen tekoälyn osa johtaa tietokonevastustajan armeijoita taistelussa. Pelissä yksiköt tekevät myös itsekseen päätöksiä tarpeen vaatiessa.

Ontañón ym (2013) jakavat RTS-pelien tekoälyn haastavimmat osa-alueet kuuteen ryhmään. Resurssien käyttöön, valintojen tekemiseen epävarmuuden alaisena, ajalliseen ja tilalliseen järjelyyn, usean tekoälyn yhteistyöhön, vastustajaan mukautu-

miseen ja oppimiseen ja vastakkaiseen reaaliaikaiseen suunnitteluun. Lisäksi vähän reitinhakua. Resurssien käytössä tekoälyn tulisi osata käyttää tilanteen mukaisesti resurssejaan. Valintoja tulisi tehdä useammin ja tietokonevastustajan tulisi osata tutkia pelikartaa saadakseen tietoa näihin valintoihin. Tekoälyn tulisi osata suunnitella pelaajaa vastaan toimiva strategia ja osata ratkoa ongelmia pelissä. Vastustajaan mukautumista on mahdollista parantaa koneoppimis tekniikoilla. Tekoäly yhteisö on myös tutkinut taktista-analyysia. Ajallinen ja tilallinen järjestyksessä tekoälyn tulisi miettiä pitkän aikavälin suunnitelmaa ja varsinkin tukikohdan järkevää laajentamista. Tekoälyn tulisi osata tehdä pelaajan tai toisen tekoälyn kanssa yhteistyötä. Reitinhakua voisi parantaa lisäämällä aika dimensio reitinhakualgoritmeille.

Koneoppiminen on suoriutumisen optimointia käyttäen esimerkkidataa tai aikaisempaa kokemusta. Oppimisen voi jakaa suoraan ja epäsuoraan oppimiseen. Micromanageroinnissa koneoppimistekniikoista on käytetty ainakin geneettisiä algoritmeja ja tilannekohtaista järjelyä. Micromanageroinnin avulla pelaaja pyrkii maksimoimaan vahingon viholliseen ja pitämään vahingot itseensä mahdollisimman pieninä. Macromanageroinnilla tarkoitetaan pelaajan maan ohjaamista. Koneoppimistekniikoita on käytetty ainakin opettelemaan rakennusjärjestystä ja valitsemaan pelitaktiikoita. Pelitallennuksia voidaan käyttää tässä hyväksi. Apuna on käytetty myös lähin naapuri funktiota.

Meidän olisi hyvä tutkia lisää koneoppimistekniikoita, jotta tulevaisuudessa RTS-peleihin todella saataisiin hyvä tekoäly, mutta siihen tarvittaisiin tekniikoita, joita olisi helppo soveltaa uusiin peleihin.

## Kirjallisuutta

Alpaydin, Ethem. 2014. "Introduction to machine learning". MIT press.

Buro, Michael. 2004. Call for AI Research in RTS Games.

David W. Aha, Matthew Molineaux ja Marc Ponsen. 2005. Learning to Win: Case-Based Plan Selection in a Real-Time Strategy Game

Hagelbäck, Johan, ja Stefan J Johansson. 2008. "The Rise of Potential Fields in Real Time Strategy Bots".

Hagelbäck, Johan. 2016. "Hybrid Pathfinding in StarCraft,"in IEEE Transactions on Computational Intelligence and AI in Games

Ji-Lung Hsieh ja Chuen-Tsai Sun. 2008. Building a Player Strategy Model by Analyzing Replays of Real-Time Strategy Games

Manslow, J. (2002). Learning and Adaptation. AI Game Programming Wisdom (ed. S.Rabin), Charles River Media, 2002, pp. 557-566.

Michael Buro ja Timothy M. Furtak. 2004. RTS Games and Real-Time AI Research

Millington, Ian, ja John Funge. 2009. "Artificial Intelligence for Games". Second Edition.

Ponsen, Marc. 2004. Improving adaptive game AI with evolutionary learning

Schwab, Brian. 2009. "AI Game Engine Programming". 2nd Edition.

Tomasz Szczepański ja Agnar Aamodt . 2009. Case-based reasoning for improved micromanagement in real-time strategy games.