Jose Saarimaa

Koneoppimisen hyödyntäminen videopeleissä

Tietotekniikan kandidaatintutkielma

3. toukokuuta 2021

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Jose Saarimaa

Yhteystiedot: jose.saarimaa@gmail.com

Ohjaaja: Antti-Jussi Lakanen

Työn nimi: Koneoppimisen hyödyntäminen videopeleissä

Title in English: Utilization of machine learning in video games

Työ: Kandidaatintutkielma

Sivumäärä: 23+0

Tiivistelmä: Tässä kandidaattitutkielmassa käsitellään koneoppimisen hyödyntämistä videopeleissä kirjallisuuskatsauksen muodossa. Ensiksi perehdytään siihen, millainen ympäristö pelit ovat koneoppimisen soveltamiselle ja erityisesti mitä haasteita niihin liittyy. Seuraavaksi käydään läpi yleisimpiä videopelien kontekstissa käytettyjä koneoppimisen tekniikoita, vertaillen niiden heikkouksia ja vahvuuksia. Lopuksi perehdytään mitä käytännön hyötyjä ja sovelluksia tutkimusaineiston perusteella on jo kehitetty ja mikä voisi olla tulevaisuudessa mahdollista.

Avainsanat: koneoppiminen, videopelit, tekoäly, pelikokemus

Abstract: This Bachelors' Thesis covers the utilization of machine learning in the context of video games in the form of a literary review. Games offer a complex environment for applications of machine learning that require multiple different techniques and approaches for the machine learning to be efficient and achieve its goals. This thesis explores these challenges and the commonly used techniques to find out which approaches have yielded successful results and what are the strengths and weaknesses of each, in addition to the practical applications of how machine learning has been used in the context of videogames and what possible applications might be possible in the future.

Keywords: machine learning, video games, AI, gaming experience

Sisältö

1 Johdanto

Videopeleistä on tullut todella suosittu viihteen muoto ja niiden kulutus vaikuttaisi olevan koko ajan kasvavaa. Sitä mukaa kun tietokoneet ja pelikonsolit muuttuvat yhä tehokkaammiksi, on myös peleistä mahdollista tehdä entistä vaikuttavampia niin visuaalisesti kuin muutenkin teknisesti. Pelien fysiikanmallinnuksesta on mahdollista tehdä entistä realistisempaa ja sisältöä yleisesti voi olla enemmän. Yksi suuri osa videopelejä ovat niiden sisältämät tietokoneen ohjaamat hahmot ja ympäristöt. Vaikka onkin olemassa pelejä, joissa pelataan vain muiden ihmisten kanssa täysin staattisissa ympäristöissä, on jonkinlainen tekoäly osa ylivoimaisesti suurinta osaa peleistä. Tekoälyllä ja erityisesti sen ohjaamilla entiteeteillä peleissä on suuri vaikutus pelikokemukseen ja siihen miten hyvänä itse peli koetaan. (Mozgovoy ja Umarov 2011)

Koneoppimisen soveltamisella pelien tekoälyyn on mahdollista saavuttaa useita sellaisia ominaisuuksia, jotka tekevät peleistä pelaajan kannalta parempia. (Mozgovoy ja Umarov 2011) Yksi keskeinen vetovoimatekijä videopeleissä on niiden tarjoama haaste, tähän kytkeytyy keskeisesti pelin vaikeustaso ja sen sopivuus pelaajan taitoihin nähden. Jos peli on liian helppo tai itseään toistava, ei pelaajat todennäköisesti jaksa kovin kauaa motivoitua sen voittamisesta. Samoin liian vaikea peli ei motivoi pelaamaan. Pelaaja voi kokea turhautumista siihen, että ei yrityksestä huolimatta koe saavuttavansa merkittävää edistymistä. (Mozgovoy ja Umarov 2011) Jos peli kykenee reagoimaan pelaajan toimiin ja mukauttamaan omaa toimintaansa sen mukaan, on näin ainakin teoriassa mahdollista tuottaa vaikeustason osalta parempi pelikokemus suurelle osalle pelaajista. Pelikokemuksesta on näin mahdollista saada myös huomattavasti haastavampi niille, jotka niin haluavat, kun tekoälyn toimintatapoja ei pystykään enää oppimaan ulkoa vaan ne oppivat vastaamaan pelaajan tekemisiin. (Tan, Tan ja Tay 2011)

Jokainen videopeli on tekoälyn toimintaympäristönä erilainen, vaikka samantyylisiin peleihin tietenkin pystyy soveltamaan samantyylisiä ratkaisuja. Erilaisia koneoppimiseen pystyviä algoritmeja ja järjestelmiä on lukuisia ja niiden soveltuvuus peleihin vaihtelee myös suuresti. Osa uusimmista esimerkiksi neuroverkkoihin perustuvista toteutuksista pystyy käytännössä oppimaan melkein minkä tahansa roolin missä pelissä tahansa, kun taas toisten mene-

telmiä toimivuus voi olla hyvinkin sidoksissa siihen millaiseen ympäristöön niitä yritetään soveltaa.(Hausknecht ym. 2014) Koneoppimiseen perustuvissa ratkaisuissa on myös muita ominaisuuksia ja tekijöitä jotka vaikuttavat niiden soveltuvuuteen peleissä, esimerkiksi kuinka hyvin oppimista voidaan hallita, kuinka nopeasti ne pystyvät oppimaan ja kuinka ne esimerkiksi reagoivat nopeasti tapahtuviin suuriin muutoksiin.

Pelit myös tarjoavat erinomaisia ympäristöjä koneoppimisen testaamiseen ja tutkimiseen ja niitä on tähän tarkoitukseen käytettykin menestyksekkäästi jo kauan. (Lucas ja Kendall 2006) Tämän tutkielman tarkoitus on luoda yleinen katsaus siihen, millaisia erilaisia koneoppimiseen perustuvia toteutuksia peleissä on jo käytössä ja minkälaisia on hyvien testitulosten perusteella mahdollisesti odotettavissa tulevaisuudessa. Lisäksi tarkoitus on selvittää erilaisten toteutustapojen vahvuuksia ja heikkouksia ja pohtia minkälaisiin tilanteisiin ne parhaiten soveltuvat.

2 Pelit ja niiden haasteet tekoälylle

Tarkasteltaessa koneoppimisen hyödyntämistä videopeleissä on syytä ensin käydä läpi pelejä tekoälyn toimintaympäristönä ja erityisesti niitä seikkoja, jotka tekevät peleistä toimintaympäristönä haasteellisen. Tutkimuksissa tulee pääsääntöisesti esille kolme suurta luokkaa, joiden alle nämä toimintaympäristön haasteet voidaan jakaa, tuntemattomat muuttujat, ristiriitaiset ja muuttuvat tavoitteet sekä ihmismäisen toiminnan jäljitteleminen. (Galway, Charles ja Black 2008) On tietenkin olemassa muitakin seikkoja, jotka vaikeuttavat tekoälyjen soveltamista, mutta ne ovat pääsääntöisesti tapauskohtaisempia, eivätkä näin yleisesti sovellettavissa suuriin joukkoihin pelejä.

Tässä tutkielmassa tekoäly viittaa sellaiseen järjestelmään tai sen osaan, joka kykenee toimimaan itsenäisesti ympäristöstä saatavilla olevan informaation perusteella ja tekemään jollain tavalla perusteltuja valintoja. Koneoppimisella taas tarkoitetaan sellaisia algoritmeja tai niiden muodostamia kokonaisuuksia jotka kykenevät muuttamaan omaa toimintaansa tavoitteenmukaisempaan suuntaan ympäristöstä saadun informaation ja jonkun palautteen perusteella. Oppimismenetelmiä sinällään on hyvin monia erilaisia ja jotkut niistä voivat perustua erilaisten vaihtoehtojen kokeilemiseen, jolloin algoritmin suoriutuminen annetusta tehtävästä voi tilapäisesti jopa huonontua.

2.1 Tuntemattomat muuttujat

Tekoäly on jo pitkään soveltunut hyvin ja menestynyt peleissä, joissa pelaajilla on käytössään kaikki peliin liittyvä informaatio koko pelin kulun ajan. Esimerkkinä tällaisesta pelistä on tammi, jossa tietokone on pärjännyt ihmistä paremmin jo 60-luvulta asti. (Samuel 2000) Sen sijaan pelit, joissa osa tiedosta on pelaajien ulottumattomissa ovat osoittautuneet hyvin hankaliksi ongelmiksi tekoälyn kehittämisen näkökulmasta, esimerkkinä pokeri, jossa neuroverkkoihin perustuva oppiva tekoäly saavutti ihmispelaajien tason vasta äskettäin. (Riley 2017) Saavuttamattomissa olevan tiedon ongelma toistuu hyvin monessa videopelissä ja monesti vielä paljon monimutkaisemmassa ympäristössä, kuin mitä pokeri on.

Perinteisesti pelejä, joissa kaikki tieto on pelaajan saavutettavissa, on pyritty ratkaisemaan

esimerkiksi muodostamalla erilaisista pelitilavaihtoehdoista puita ja sitten tekemään hakuja niihin voittavan/parhaan pelitilan löytämiseksi. Monimutkaisemmissa peleissä, joissa puut tulevat liian isoiksi voidaan käyttää erilaisia heuristiikkoja ongelman välttämiseksi. Puuttuva tieto on haaste tekoälylle, koska silloin parasta mahdollista ratkaisua ei voida arvioida käymällä läpi ja vertailemalla kaikkia mahdollisia potentiaalisia skenaarioita tai vaikka voitaisiin, olisi eri vaihtoehtoja niin paljon, että se ei ole vielä nykyisten koneiden laskentakapasiteetin tavoitettavissa. (Lucas ja Kendall 2006) Esimerkkejä tuntemattomista muuttujista moderneissa videopeleissä ovat esimerkiksi vastustajan käyttämät strategiat peleissä kuten Strarcraft, pärjätäkseen ihmispelaajalle tekoälyn pitäisi kyetä jollain tavalla ennakoimaan ja vastaamaan toisen pelaajan tekemiin valintoihin.

Erityisesti lukuisten eri pelitilojen ja tuntemattomien muuttujien haasteet korostuvat moderneissa peleissä, jotka tulevat koko ajan monitasoisemmiksi ja laajemmiksi. Vaikka esimerkiksi pokeri voi saavuttaa todella suuren joukon erilaisia pelitiloja (game state), ei se ole vielä lähelläkään sitä määrää, mitä moderneissa videopeleissä voi olla. (Lucas ja Kendall 2006) Yleisesti parhaiten tuntemattomia muuttujia sisältävissä peleissä menestyvät yleiskäyttöiset (general purpose) tekoälyt, josta kykenevät oppimaan ja toimimaan jollain tavalla ympäristöstään riippumatta. Tällaisten algoritmien ja tekoälyjen kehittäminen on hyödyllistä myös laajemmin, kuin vain pelien näkökulmasta. On arvioitu, että tekoäly, joka pärjää hyvin pokerissa voisi olla hyödyksi myös esimerkiksi neuvotteluissa tai muissa vastaavissa tilanteissa, joissa ”vastustajan kättä” ei pystytä näkemään. (Lucas ja Kendall 2006)

2.2 Ristiriitaiset ja muuttuvat tavoitteet

Hyvä havainnollistava esimerkki, jossa muuttuvat ja ristiriitaiset tavoitteet tulevat esille haasteena tekoälylle, ovat reaaliaikaiset strategiapelit kuten StarCraft. Niissä pelaajat, ihmisten tai tekoälyjen ohjaamat, voivat liikuttaa useita eri yksiköitä samanaikaisesti, yleensä ilman suuria rajoituksia, rakentaa yksiköitä, johon menee aikaa ja mahdollisesti muita resursseja ja yleensä tehdä vielä monia muitakin valintoja pelin aikana. Kun tähän lisätään se, että kaikki tapahtuu reaaliajassa samaan aikaan, on tekoälylle todella suuri haaste toimia tällaisessa ympäristössä tehokkaasti.(Synnaeve ja Bessiere 2015) Normaali idea siitä, että pyritään oppimaan paras mahdollinen toimintapa toivotun lopputuloksen saavuttamiseksi ei enää pä-

de, koska jatkuvasti muuttuva ympäristö luo tilanteen, jossa tekoäly yrittää oppia koko ajan muuttuvaa parasta tapaa toimia. Tällaisessa ympäristössä toimimiseksi paljon syvempi oppiminen pelin eri tasoista ja osa-alueista on välttämätöntä, jotta toimintatapaa osataan mukauttaa ympäristön mukaan.(Bowling ja Veloso 2002)

Modernien tietokonepelien tavoitteet eivät ole myöskään yhtä suoraviivaisesti määritettävissä kuin esimerkiksi shakissa, joissa vastustajia on aina yksi ja säännöt pelin voittamiseksi ovat lopulta melko suoraviivaiset. Moderneissa tietokonepeleissä taas päämäärä on mahdollista jakaa tilanteesta riippuen hyvinkin moneen erilaiseen osatavoitteeseen, mutta kaikki niistä eivät ole välttämättä läheskään yhtä tärkeitä lopullisen päämäärän saavuttamisen kannalta. (Hagelbäck 2015) Jos ajatellaan esimerkiksi jotakin reaaliaikaista strategiapeliä, osatavoitteita voisivat olla pelissä sillä hetkellä käynnissä olevan yksittäisen taistelun voittaminen, mahdollisen tukikohdan kehittäminen tai toisen pelaajan kimppuun hyökkääminen. Kaikki näistä ovat hyödyllisiä koko pelin voittamisen kannalta, mutta kaikkiin ei pelissä käytössä olevien, rajallisten resurssien vuoksi todennäköisesti ole mahdollista panostaa. Tällöin tekoälyn on kyettävä tekemään valintoja monen eri tavoitteen väliltä, jota kaikki ovat ainakin osittain välttämättömiä voiton kannalta, osatavoitteet myös vaihtuvat koko ajan pelin edetessä.

2.3 Ihmismäisyys

Suuri osa moderneista videopeleistä sisältää hahmoja tai tahoja, jotka ainakin periaatteessa pyrkivät jollain tavalla mallintamaan ihmisen toimintaa jossain tilanteessa. Tietokoneen ohjaamat tahot voivat toimia ihmispelaajan vastustajina, avustajina tai neutraaleina tahoina ja uskottavan pelikokemuksen takaamiseksi niiden kaikkien pitäisi osata toimia ”ihmismäisesti” ja tilanteeseen sopivalla tavalla. (Mozgovoy ja Umarov 2011) Tässä osiossa ei kiinnitetä huomiota sosiaaliseen kanssakäymiseen, vaan pikemminkin siihen, että ihmiset tekevät peleissä esimerkiksi virheitä ja voivat toimia hyvinkin odottamattomilla tavoilla. Tekoälylle tämä on perinteisesti ollut vaikeaa. Erityisesti sellaisen tekoälyn toteuttaminen, joka ihmispelaajan tavoin kykenee muuttamaan jatkuvasti toimintaansa, mutta kykenee silti saavuttamaan jatkuvasti sille asetetut tavoitteet on osoittautunut haastavaksi. (Mozgovoy ja Umarov 2011)

Voidaan ajatella, että tekoälyn ohjaaman hahmon tapauksessa suurin prioriteetti ei välttämättä ole se, kuinka tehokkaasti annettu tehtävä suoritetaan tai kuinka hyvään päätöksentekoon tekoäly kykenee, vaan millaisen kokemuksen se tarjoaa ihmispelaajalle. (Tan, Tan ja Tay 2011) Kuvitellaan esimerkiksi jokin ammuntapeli, jossa tekoälyn ohjaama vastustaja toimii täydellisellä nopeudella ja tarkkuudella, tällöin ihmispelaaja on lähes väistämättä tuomittu häviämään eikä pelikokemus voi olla kovin positiivinen. On käytetty käsitettä tekotyhmyys (Artificial stupidity) (Tan, Tan ja Tay 2011), viittaamaan siihen, että tekoäly ohjataan tarkoituksella tekemään virheitä ja hidastelemaan, että pelikokemus olisi ihmiselle miellyttävämpi.

Oppivien tekoälyjen näkökulmasta tämä on mielenkiintoinen ongelma, koska yleensä oppimisalgoritmi pyrkii nimenomaan oppimaan, kuinka annettu päämäärä saavutetaan mahdollisimman tehokkaasti. Mutta ”ihmispelaajalla täytyy olla mahdollisimman hauskaa” ei ole ainakaan toistaiseksi sellainen tavoite, jonka tekoälylle voi antaa. Näin ollen täytyy käyttää jotain välillisiä tavoitteita, jotka tekoäly voi oppia saavuttamaan, esimerkiksi ennakoida ihmispelaajan toimia, riittävän haasteen tarjoamiseksi tai pelaajan avustamiseksi.

3 Koneoppimisen menetelmiä

Tässä osiossa perehdytään käytännön sovelluksiin ja toteutustekniikoihin joilla tekoälyä on hyödynnetty pelien yhteydessä. Kuusi keskeisintä ja eniten käytettyä koneoppimisen tyyppiä videopeleissä ovat evoluutiopohjaiset (evolutionary computation), vahvistusoppiminen (reinfocement learning), ohjattu oppiminen (supervised learning), ohjaamaton oppiminen (unsupervised learning), suunnitteluun perustuvat ja puurakenteisiin pohjautuva hakeminen. (Yannakakis ja Togelius 2014) Kaikilla näillä tekniikoilla on omat vahvuutensa ja heikkoutensa, joiden perusteella niitä voidaan valita sovellettaviksi eri tilanteissa.

3.1 Puurakenteisiin pohjautuva oppiminen

Koneoppiminen termin alkuperäisenä kehittäjänä pidetään Arthur Samuelia, joka tutki puurakenteisiin pohjautuvan koneoppimisen soveltamista tammen pelaamiseen. (Weiss 1992) Videopelejä ei vielä nykymuodossaan ollut 1950-luvulla, jolloin Samuel kehitti koneoppimisen ensimmäisiä muotoja, mutta samoja tekniikoita käytetään nykypäivänä sähköisten pelien yhteydessä. Perusidea tämäntyyppisessä koneoppimisessa on, että pelin eri tiloista muodostetaan puu siten että jokainen pelilaudan tila vastaa aina yhtä solmua. Solmuista lähtevät haarat vastaavat pelaajien tekemiä valintoja kyseisessä pelitilassa, näin jokainen polku puun juurisolmusta johonkin lehtisolmuun vastaa siis yhtä pelin kulkua alusta loppuun, näin muodostetusta puusta on teoriassa mahdollista valita aina se haara, jossa olevat pelitilat ovat valinnan tekijän kannalta mahdollisimman hyvät. (Samuel 2000)

Käytännössä missä tahansa pelissä on niin paljon mahdollisia erilaisia pelinkulkuja, että kokonaisia puita ei millään pystytä pitämään tietokoneen muistissa, saati sitten käsittelemään. Ratkaisuna tähän ongelmaan on määrittää kullekin pelitilalle jokin arvo perustuen siihen, kuinka tavoiteltava kyseinen pelitila on voittamisen kannalta ja peliä toistuvasti pelaamalla tallentaa muistiin usein toistuvia pelitiloja ja niitä vastaavia arvoja. Mitä enemmän tietoa usein esiintyvistä pelitiloista on opittuna, sitä enemmän puusta voi karsia pois haaroja, joiden valitsemisesta seuraisi epäedullisia pelitiloja. (Samuel 2000) Nykyisin puurakenteisiin pohjautuvaa oppimista on helppoa myös yhdistää muihin koneoppimisen tapoihin, esimerkiksi

käyttämällä jotain muuta oppivaa menetelmää pelitilaan sidottavan arvon määrittämiseen.

3.2 Toistosta oppivat menetelmät

Yleisesti toistosta oppivia menetelmiä (reinforcement-learning) käytettäessä, oppiva algoritmi on yhteydessä toimintaympäristöönsä joidenkin havaintojen ja omien toimintojensa kautta. Jokaisella iteraation askeleella algoritmi saa ympäristöltään tietoa ympäristön tilasta (havainto). Tämän jälkeen algoritmi valitsee jonkin toiminnon, joka muuttaa ympäristön tilaa, tilan muutos aiheuttaa algoritmille jonkinlaisen vahvistussignaalin (reinforcement signal). Algoritmin tavoite on valita suoritettavat toiminnot siten, että vahvistussignaalien suuruus pitkällä aikavälillä on mahdollisimman suuri. Algoritmin toiminnan ydin on siis käytännössä, että virheiden ja onnistumisten avulla siirrytään jatkuvasti kohti optimaalista ratkaisua. (Kaelbling, Littman ja Moore 1996) Tämän toiminnallisuuden eri lailla toteuttavia algoritmeja on lukuisia erilaisia.

Toistossa oppivissa menetelmissä olennaista on, kuinka paljon eri vaihtoehtoja tarvitsee kokeilla, verraten siihen millaisia tuloksia oppiminen alkaa tuottaa. Yleisesti mitä suurempi joukko erilaisia mahdollisia toimintoja voidaan kokeilla, sitä todennäköisemmin löytyy toivotunlaisia lopputulokisa antava strategia, mutta vähänkään monimutkaisimmissa ympäristöissä kokeiltavien vaihtoehtojen määrä kasvaa niin suureksi, että vain satunnaisesti kokeileminen ei ole enää riittävän tehokasta. (Burnetas ja Katehakis 1997) Erot eri toistoon perustuvien oppivien algoritmien välillä on pääasiassa siinä, kuinka ne etsivät uusia strategioita ja miten ne painottavat uusien strategioiden kokeilua jo löytyneisiin jollain tavoin toimiviin strategioihin verrattuna.

3.3 Evolutiiviset

Nimensä mukaisesti evolutiiviset oppivat menetelmät perustuvat optimaalisen ratkaisun etsintään luonnollisen evoluution toimintaperiaatetta jäljittelemällä. Menetelmässä joukkoa mahdollisia ratkaisuja ajatellaan ikään kuin populaationa, jotka ovat muodostettu ja joita käsitellään ja ylläpidetään luonnollista genetiikkaa muistuttavilla operaatioilla. Mahdollisten ratkaisujen sisältö on pakattu sellaiseen muotoon, että tämä onnistuu. Eri ratkaisuvaihtoeh-

tojen soveltuvuutta vertailemalla voidaan iteratiivisesti päästä lähemmäs parasta ratkaisua, mikäli se on tällaisella menetelmällä löydettävissä. (Goldberg ja Holland 1988) Evolutiiviset algoritmit jakautuvat online- ja offline-tyyppisiin, sen mukaan pyrkiikö algoritmi iteroimaan kohti parempaa ratkaisua, kun peli tai muu vastaava oppimisympäristö on käynnissä (online), vai näiden suorituskertojen välissä (offline). (Galway, Charles ja Black 2008) Kummassakin lähestymistavassa on omat vahvuutensa, mutta yleisesti offline-tyyppiset toteutukset sopivat parhaiten sellaisiin ongelmiin, joissa suoritusaika on lyhyt ja suorituskertoja tulee paljon. Offline-tyyppisille algoritmeille soveltuvuusvertailun tekeminen on yksinkertaisempaa, koska koko suorituksen lopputulosta voidaan käyttää yhtenä valintakriteerinä. (Galway, Charles ja Black 2008)

Jotta evolutiivisilla menetelmillä voidaan päästä kohti optimaalista ratkaisua, täytyy evoluutiota varten tarvittavat kandidaatit ensin tuottaa jollain keinoilla. Kandidaattien tuottamisessa voidaan käyttää apuna jotain muuta oppivaa menetelmää tai erilaisiin heuristiikkoihin tai satunnaisuuteen perustuvia vaihtoehtoja. Se millainen tapa kandidaattien tuottamiseen valitaan, riippuu täysin mihin tarkoitukseen evolutiivista oppimismenetelmää ollaan soveltamassa. Videopelien tekoälyvastustajien tuottamisessa on saatu hyviä tuloksia esimerkiksi tuottamalla kandidaatit neuroverkkojen avulla ja sen jälkeen soveltamalla niihin evolutiivista menetelmää. (Zhen ja Watson 2013)

3.4 Neuroverkot

Vaikka keinotekoisiin neuroverkkoihin (artificial neural networks) pohjautuvat toteutukset jakavatkin yleisesti saman biologista neuroverkkoa jollain tavalla jäljittelevän toimintaidean on erilaisia toteutuksia todella suuri määrä ja myös niiden ominaisuudet vaihtelevat paljon. Keinotekoiset neuroverkot on lähtökohtaisesti kehitetty hyvin yleiseksi (general purpose) työkaluksi, jotka kykenevät oppimaan ja mallintamaan laajasti erilaisia asioita. Sovelluskohteissa, joissa mahdollisimman yleinen ja syvä oppiminen on eduksi ovat neuroverkot osoittautuneet melko ylivoimaiseksi työkaluksi verrattain muihin oppimismenetelmiin. (Justesen ym. 2019) Huomioitava on myös, että neuroverkot ovat menetelmänä melko raskas ja monimutkainen ja yleensä niiden suorituskyky tietyssä tehtävässä jää jälkeen nimenomaan tätä kyseistä tehtävää varten laaditusta algoritmista.

Kun keinotekoisia neuroverkkoja tarkastellaan videopelien viitekehyksestä, voidaan ne jakaa kahteen kategoriaan sen mukaan, kuinka ne vastaanottavat syötteenä tulevaa dataa. Feedforward-tyyppiset neuroverkot ottavat vastaan yhden syötteen ja antavat sen perusteella jonkin tuloksen. (Justesen ym. 2019) Syöte voi olla esimerkiksi tämänhetkinen pelitila ja tuloksena vaikka ennustuksia mahdollisten tulevien tapahtumien todennäköisyyksistä tai mahdollisia toimintoja, joita tekoäly voi tehdä. Feedforward-tyyppiset neuroverkot voi jakaa vielä kahteen alatyyppiin, direct encoding- ja indirect encoding-tyyppisiin, sen mukaan, kuinka verkon opettaminen käytännössä toimii. Direct encoding-tyyppisissä verkkoa opetetaan suoraan syötettävän datan avulla, kun taas indirect encoding käyttää apuna toista verkkoa varsinaisen vastauksen tuottavan verkon opettamiseen. (Hausknecht ym. 2014) Direnct encoding-tyyppiset algoritmit suoriutuvat indirect encoding-tyyppisiä paremmin sellaisissa tilanteissa, jossa syötteenä on esikäsiteltyä matalaulottuvuuksista dataa pelin sisällöstä, mutta indirect encoding-tyyppiset esimerkiksi pystyvät oppimaan peleissä toimimisen pelkästään pelkän pikselidatan perusteella. Tämä perustuu siihen, että indirect encoding-lähestymistavan neuroverkkoratkaisut pystyvät oppimaan paljon suurempia verkkoja tietoa, joka on välttämättömyys silloin, kun syötteen perusteella pitää muodostaa monimutkaisempia toimintamalleja. (Hausknecht ym. 2014)

Toinen kategoria ovat recurrent-tyyppiset neuroverkot, jotka toimivat järjestyksessä sarjana tulevalla syötteellä ja käyttävät lisäksi syötteenä verkon aiempaa tilaa, jolloin tilat ovat aina kontekstissa aiempiin syötteisiin. (Justesen ym. 2019) Recurrent tyyppinen verkko on hyödyllinen silloin, kun tämänhetkinen pelitilasta saatava data ei pelkästään riitä antamaan tarpeeksi laajaa kuvaa päätöksentekoa varten. Käytännössä suurimmassa osassa moderneja videopelejä vaaditaan jotain tietoa aiemmista pelin tapahtumista, jotta pelaaja voi tehdä hyviä päätöksiä ja onnistua pelissä. Kun neuroverkkoja sovelletaan videopeleihin, on yleistä, että erityyppisiä neuroverkkoja käytetään yhdessä parhaiden oppimistulosten saavuttamiseksi. (Justesen ym. 2019)

Jos neuroverkkopohjaiseen järjestelmään lisätään vielä ominaisuuksia muista tekoälyistä, on tällaisen yhdistelmän avulla mahdollista rakentaa hyvinkin monimutkaisissa peleissä ihmistä paremmin pelaava tekoäly. Yhdistelemällä parhaita puolia eri menetelmistä voidaan saada neuroverkkojen kyky ratkaista monimutkaisia ongelmia ja kohdennetumpien algoritmien te-

hokkuus. Erityisen tehokasta on esikäsitellä neuroverkon saamaa syötetietoa ja karsia siitä eri menetelmillä ongelman kannalta epäolennaista dataa, tämä tehostaa huomattavasti neuroverkon oppimiskykyä ja tehokkuutta. (Tesauro 2002)

4 Koneoppimisen sovellukset peleissä

Koneoppimista on mahdollista soveltaa videopelien yhteydessä niin pelien sisäisinä komponentteina, ihmispelaajien korvaajina kuin apuna pelinkehityksessä. Tässä luvussa käsitellään sitä, miten koneoppimista on käytännössä hyödynnetty tähän mennessä ja mitä mahdollisia käyttöalueita olisi tehdyn tutkimuksen mukaan olemassa tulevaisuudessa. Sekä videopelit, että koneoppiminen ovat nopeasti kehittyviä tutkimusaloja, joissa uusia ratkaisuja ja mahdollisuuksia syntyy koko ajan.

4.1 Tekoäly pelaajana

Perinteisten algoritmien kehittämiselle, jotka perustuivat puurakenteiden muodostamiseen erilaisista pelitiloista ja hakujen tekemisestä niihin, ovat vanhat ja tunnetut lautapelit olleet todella tärkeitä testausympäristöjä. Tällaisia pelejä, kuten shakki ja tammi, tekoälyt ovat osanneet pelata todella hyvin jo kauan (Lucas ja Kendall 2006), mutta monimutkaisemmat pelit kuten Go ovat olleet ihmispelaajien hallitsemia aivan lähiaikoihin asti, kunnes neuroverkkoihin ja erilaisten koneoppimistekniikoiden yhdistelmiin perustuvat ratkaisut ovat alkaneet päihittää parhaitakin ihmispelaajia. Yksinkertaisemmat puurakenteisiin perustuvat tekoälyt ovat pärjänneet jo jonkin aikaa myös yksinkertaisimmissa videopeleissä, kuten Mariossa ja Pac-Manissa.

Tekoälyn ohjaamat vastustajat ovat hyvin tärkeä osa myös moderneja videopelejä, aivan niin kuin shakissa tai tammessa voi ihmispelaaja pelata tekoälyä vastaan myös Starcraftissa. Perinteisesti tekoälyvastustajat eivät ole onnistuneet pärjäämään hyville ihmispelaajille vielä moderneissa strategiapeleissä, kuten Starcraftissa ilman, että niille annetaan keinotekoinen etulyöntiasema esimerkiksi resursseja tai käytettävissä olevaa tietoa lisäämällä. Erityisesti neuroverkkojen kehittymisen myötä on kuitenkin saavutettu merkittävää edistymistä tekoälyvastustajissa myös näissä vaativimmissa sovelluskohteissa. (Zhen ja Watson 2013) DeepMindin kehittämä, neuroverkkoihin pohjautuva tekoäly on jopa viimevuosina pärjännyt parhaille ihmispelaajille StarCraft II videopelissä. (Vinyals ym. 2019)

Vaikka tutkimuksessa onkin keskitytty paljon strategiapeleihin ja perinteisiin lautapeleihin

koneoppimista käyttävien tietokonepelaajien testaus ja tutkimusympäristönä, on tällaisia tietokonevastustajia tai pelaajia mahdollista soveltaa myös muun tyyppisissä peleissä. Suurin vahvuus koneoppimista hyödyntävissä tekoälyissä johonkin ei-oppivaan tekoälymalliin verraten on se, että kun tekoäly muuttaa toimintaansa, muuttuvat myös sen heikkoudet ja vahvuudet. Näin ollen ihmispelaaja ei voi helposti päihittää tekoälyä joka kerta samalla tavalla tuntemalla sen heikkoudet. (Lucas ja Kendall 2006) Mikäli peli on riittävän monimutkainen, ei välttämättä ole edes mahdollista toteuttaa siinä hyvin toimivaa staattista tekoälyä, näin ollen koneoppiminen voisi mahdollistaa jopa kokonaan uudentyyppisiä pelejä tai mahdollistaa tietokonepelaajien luomisen peleihin, jotka ovat aiemmin vaatineet useamman ihmispelaajan. (Yannakakis ja Togelius 2014)

Kehitys koneoppimisessa, erilaisten koneoppimistapojen yhdistely ja erityisesti neuroverkkoihin pohjautuvien toteutusten muuttuminen paremmiksi on mahdollistanut entistä yleiskäyttöisempien tekoälyjen syntymisen. (Yannakakis ja Togelius 2014) Tällaiset tekoälyt kykenevät pelien sisällä vastaamaan päätöksenteosta ja valinnoista entistä laaja-alaisemmin sen sijaan, että olisi lukuisia erillisiä komponentteja hoitamassa eri osaratkaisuja. Neuroverkkoihin perustuvilla ratkaisuilla on saatu todella hyviä tuloksia myös silloin, kun sama tekoäly laitetaan pelaamaan useampaa erilaista peliä. (Yannakakis ja Togelius 2014) Tällaiselle yleisemmälle tekoälylle ei ole kovin mielekästä testata sitä jotain muun tyyppistä toteutusta vastaan vain yhdessä pelissä, esimerkiksi shakkitietokone pärjää todennäköisesti yleiselle oppivalle tekoälylle hyvin shakissa, mutta ei kykene mitenkään toimivaan esimerkiksi Super Marion kanssa. Näin ollen kriteerit hyvin toimivalle, tai tehokkaalle menetelmälle eivät ole aina niin selkeitä.

4.2 Tekoälyn ohjaamat hahmot

Yleisesti ensimmäiseksi videopelien tekoälystä puhuttaessa tulee mieleen nimenomaan tekoälyn ohjaamat hahmot, eli NPC:t (Non-person characters), joista itsestään käytetään tässä kontekstissa joskus nimeä tekoälyt. Tässä tutkielmassa tekoäly viittaa kuitenkin aina yleisempään mekanismiin, joka pystyy toimimaan itsenäisesti ja tekemään perusteltuja valintoja jossain ympäristössä, eikä pelkästään sen ohjaamiin hahmoihin. Tämä nimitys kuitenkin kuvaa hyvin sitä, kuinka merkittävä sovellusalue NPC:t ovat tekoälyille.

Kaksi tekoälyn ohjaamille hahmoille annettavaa tärkeää laadun vaatimusta ovat uskottavuus ja tehokkuus päätöksenteossa. Nykyisissä suurissa peleissä NPC:t toteutetaan kuitenkin vielä pääsääntöisesti äärellisillä automaateilla ja valmiisiin sääntöihin pohjautuvilla järjestelmillä, vaikka niillä ei tutkimusten perusteella kyetäkään tuottamaan niin hyviä lopputuloksia, kuin mitä oppimiseen perustuvilla toteutuksilla olisi jo nyt mahdollista. (Mozgovoy ja Umarov 2011) Erityisesti koneoppimisen hyödyt korostuvat moderneissa peleissä, joiden monimutkaisuus, monipuolisuus ja sisällön laajuus kasvaa jatkuvasti ja staattisiin malleihin perustuvien toteutus muuttuu koko ajan suuritöisemmäksi.

Näiden algoritmien päätöksentekokykyä koskevien vaatimusten kasvamisen lisäksi nykyisin pelaajat myös odottavat NPC:iltä koko ajan ihmismäisempää ja ”syvällisempää” käyttäytymistä. Syvällisellä tarkoitetaan, että hahmot noudattavat niiden edustamaa roolia yhä yksityiskohtaisemmin ja monipuolisemmin, esimerkkinä erilaiset keskustelut, joita niiden kanssa on mahdollista käydä tai kuinka ne reagoivat pelaajan tekemiin valintoihin. (Mozgovoy ja Umarov 2011) Osa NPC:iden ihmismäisyyteen liittyvää uskottavuutta on tietenkin myös, kuinka hyvin tekoäly pystyy tarpeen vaatiessa suoriutumaan laajastakin joukosta eri tehtäviä, pelistä riippuen tämä voi sisältää esimerkiksi keskustelujen lisäksi, pelitilassa liikkumista, resurssien hallintaa tai jonkinlaisia taistelumekaniikkoja. (Yannakakis ja Togelius 2014) Kuten aiemmin on todettu, koneoppimiseen perustuvalla yleisellä tekoälyllä on hyvin suuria vahvuuksia tällaisessa sovelluskohteissa.

Erityisesti sellaisten NPC:iden tuottamisessa, jotka toimivat ihmispelaajien vastustajina olisi saavutettavissa paljon hyötyä koneoppimisen avulla. Yksi keskeinen uskottavuustekijä tallaisissa tilanteissa on se, että pelaajalle syntyy mielikuva reilusti pelaavasta vastustajasta. (Synnaeve ja Bessiere 2015) Tällä tarkoitetaan sitä, että tekoälyn ohjaamat hahmot eivät esimerkiksi toimi jonkun sellaisen tiedon perusteella, jota ihmispelaajalla ei olisi käytettävissään samassa tilanteessa, tai, että niillä ei ole käytettävissään loputtomia resursseja priorisointiongelmien kiertämiseksi. Muita tällaisia tekijöitä ovat esimerkiksi reagointinopeus ja mekaaninen tarkkuus, joissa koneet tyypillisesti ovat aina parempia kuin ihmiset.

Koneoppimisen avulla voidaan NPC:tä ohjaavalle tekoälylle opettaa oikean ihmispelaajan toimintaa kyseisessä pelissä. Tällöin ei synny tekoälyä, joka toimii mahdollisimman tehokkaasti vaan mallinnetun ihmispelaajan tavoin. Ongelmana tässä kehitystavassa uskottavuu-

den kannalta on se, että NPC:t toimivat ihmispelaajan, eivätkä välttämättä ihmisen tavoin, joissain tilanteissa näillä kahdella voi olla isokin ero. (Yannakakis ja Togelius 2014) Tämä menetelmä itsessään ei myöskään tee tekoälystä vielä oppivaa, vaan siihen on muiden ominaisuuksien saamiseksi lisättävä tarvittavat komponentit. Menetelmää voidaan myös käyttää esiopettamaan oppivia tekoälyn ohjaamia hahmoja. Mitkään näistä menetelmistä eivät kuitenkaan ole vielä kovin laajalti käytössä peliteollisuudessa. (Galway, Charles ja Black 2008)

4.3 Oppiva peli

Tähän asti tekoälyn sovelluksista on puhuttu lähinnä pelien ulkopuolisena mekanismina tai korkeintaan niiden sisällä toimivana vähintään jollain tasolla itsenäisenä osana. Koneoppimista on kuitenkin mahdollista käyttää myös pelissä itsessään, pääasiassa pelikokemuksen parantamiseksi tai kehitys ja ylläpitotyössä tarvittavan datan keräämisen tehostamiseksi. (Tan, Tan ja Tay 2011) Näin peli saadaan reagoimaan pelaajan toimiin, tai johonkin muuhun ympäristön muutokseen ja muuttamaan pelikokemusta tai jotain muuta haluttua pelin ominaisuutta toivottuun suuntaan.

Yksi hyvä käytännön esimerkki on oppimisen avulla toteutettu dynaaminen vaikeustaso, jolloin peli mukautuu käyttäjän taitojen kehittymiseen tai vaikka pelaajan vaihtumiseen niin, että pelikokemus on jatkuvasti mahdollisimman viihdyttävä. (Tan, Tan ja Tay 2011) Koska ihmispelaajat ovat hyvin erilaisia, on kaikille sopivan oppimiskäyrän löytäminen lähes mahdoton tehtävä. (Tan, Tan ja Tay 2011) Yleisesti videopelien vaikeustaso ja kompleksisuus kasvavat pelin edetessä ja tämä puolestaan nostaa pelin vaikeustasoa, oppimiskäyrällä tarkoitetaan tässä sitä, kuinka nopeasti vaikeustaso nousee. Lisäksi jos sama pelaaja pelaa pelin läpi useampaan kertaan, on varsinkin pelin alkuosa monesti hyvin helppo. Perinteisesti tätä ongelmaa eritasoisista pelaajista on kierretty manuaalisesti vaihdettavilla vaikeustasoilla, porrasteisuutensa ja pelaajan omaan arvioon perustuvana tämä ei kuitenkaan aina ole paras mahdollinen ratkaisu. (Tan, Tan ja Tay 2011)

Oppivilla algoritmeilla on mahdollista myös tuottaa pelin sisältöön variaatiota muuttamalla esimerkiksi NPC:iden toimintaa tai muokkaamalla pelimaailmaa sitä mukaa kun pelaaja alkaa oppia aikaisemmat. (Yannakakis ja Togelius 2014) Näin pelistä saa uusia ja mielen-

kiintoisia kokemuksia vaikka sen olisikin pelannut läpi tai käyttänyt siihen muuten paljon aikaa. Tietenkin tämäntyyppisiä mekaniikkoja on mahdollista toteuttaa peleissä ilman oppivia menetelmiäkin, mutta pelaajakohtaisesti yksilöity kokemus ei silloin ole mahdollinen. (Tan, Tan ja Tay 2011) Erityisesti proseduraalinen sisällön muodostaminen, joka on tullut todella suosituksi videopeleissä, voisi hyötyä paljonkin koneoppimisen hyödyntämisestä, koska yleisesti ihmiset ovat hyviä erottamaan mikä on satunnaisesti tuotettua sisältöä ja mikä tarkoituksenmukaisesti suunniteltua. Yksi esimerkki proseduraalista sisällönmuodostamista hyödyntävästä suositusta videopelistä on Minecraft, jossa ei kuitenkaan hyödynnetä vielä koneoppimista.

Se kuinka hyvin koneoppimista saadaan tässä kontekstissa sovellettua, on myös suuresti kiinni siitä, millaisilla mittareilla onnistumista esimerkiksi pelikokemuksen parantamisessa voidaan arvioida. (Tan, Tan ja Tay 2011) Esimerkiksi tällaisista mittareista voivat olla prosentuaalinen onnistumisten määrä pelaajalle, keskimääräiset saavutettavat pisteet tai joku muu vastaavanlainen pelimekaniikasta riippuva seikka. Tällöin on mahdollista myös saada kerättyä dataa pelin vaikeustasoon ja pelaajan käyttäytymiseen liittyen tarkastelemalla minkälaiset muutokset tekoälyn toiminnassa vaikuttavat esimerkiksi voittamisen todennäköisyyteen. (Tan, Tan ja Tay 2011)

5 Yhteenveto

Tämän tutkielman tarkoitus oli selvittää, millaisia eri sovelluskohteita videopeleissä on koneoppimiselle ja mitä erilaisia koneoppimisen muotoja niissä on jo käytössä ja millaisia tulevaisuuden mahdollisuuksia aiheen tutkimuksesta löytyy. Pääasiassa aiheen tutkimus näyttää keskittyvän kahteen pääaiheeseen, pelejä pelaavien tekoälyjen tuottamiseen koneoppimisen avulla sekä mahdollisuuksiin parantaa pelikokemusta koneoppimista hyödyntäen. Lisäksi tietyt pelityypit ja pelit nousivat esille tekoälytutkimuksen kohteina huomattavan usein, osittain sopivien ominaisuuksiensa ja osittain myös niille hyvin saatavilla olevien työkalujen mukaan. Erityisen suosittuja tutkimusalueita olivat strategiapelit StarCraft, StarCraft 2 sekä PlanetWars.

Videopelit ovat haastava sovellus ja testausympäristö koneoppimista hyödyntäville sovelluksille. Osa näistä haasteista on samoja, joita tekoälytutkijat ovat ratkoneet perinteisten lautapelien parissa jo vuosikymmeniä, kuten tuntemattomat muuttujat ja pelin edetessä muuttuvat ja ristiriitaiset tavoitteet. Videopeleissä on myös omat uniikit haasteensa tekoälysovelluksille, kuten ihmismäisen toiminnan mallintaminen ja hyvin monimutkaisissa ympäristöissä toimiminen. Tekoälysovellukset ovat ottaneet merkittäviä edistysaskeleita näiden ongelmin ratkaisemisessa, kun esimerkiksi Go-pelissä ja Pokerissa koneoppimiseen pohjautuvien tekoälyjen ohjaamat pelaajat ovat voittaneet parhaita ihmispelaajia. Myös videopelien alueella StarCraft 2 pelissä tekoäly on pärjännyt parhaille ihmispelaajille, joka lupaa hyvää aihealueen kehitykselle, koska StarCraft 2 on monimutkaisena pidetty toimintaympäristö, niin koneoppimista soveltamalla saadaan varmasti hyviä tuloksia muidenkin pelien kohdalla.

Koneoppiminen on aihealueena hyvin laaja, rajattuna niihin tekniikoihin ja menetelmiin, joita on tutkimuksessa ja käytännössä sovellettu videopeleihin erottuu muutamia lähestymistapoja, jotka toistuvat usein. Varhaisimmat tutkimukset ja sovellukset keskittyivät erilaisten puurakenteiden muodostamisiin mahdollisista pelitiloista, joihin hakuja tekemällä ja eri vaihtoehtoja kokeilemalla on mahdollista oppia pääsemään parhaisiin ratkaisuihin kustakin pelitilasta. Tähän liittyy myös toinen varhaisimmista menetelmistä, eli toistosta oppiminen, sitäkin on käytetty yksinkertaisiin videopeleihin jo kauan ja erityisesti yhdessä muiden menetelmiä kanssa se on osoittautunut hyväksi työkaluksi.

Muita tutkimuksessa usein esiintyviä menetelmiä ovat biologisista prosesseista ideansa saaneet evolutiiviset- sekä keinotekoisiin neuroverkkoihin pohjautuvat oppimismenetelmät. Erityisesti neuroverkkojen hyödyntäminen on tuottanut erittäin hyviä tuloksia varsinkin sovelluskohteissa, jossa koneoppimista hyödyntävä tekoäly toimii itse pelaajana. Lisäksi neuroverkot ovat osoittautuneet varsin ylivoimaiseksi työkaluksi silloin, kun sama tekoäly on tarkoitus opettaa pelaamaan useaa erilaista peliä. Sekä evolutiivisia, että neuroverkkoratkaisuja käytetään hyvin usein yhdistettynä johonkin muuhun koneoppimisen menetelmään.

Läpikäydyn aineiston perusteella vaikuttaa myös siltä, että koneoppimisen avulla olisi mahdollista saada paljon hyötyä pelien kehittämiseen ja pelaajakokemuksen parantamiseen. Erityisen hyviä tuloksia on saatu tekoälyn ohjaamien vastustajien tuottamisessa ihmispelaajille. Muita lupaavia sovelluskohteita ovat tekoälyn ohjaamat hahmot peleissä, sekä peliympäristö itsessään, esimerkiksi pelaajan toiminnan mukaan säätyvä dynaaminen vaikeustaso. Tekoälyn ohjaamien NPC hahmojen osalta erityisesti ihmismäisen toiminnan mallintamiseen koneoppiminen on osoittautunut hyväksi työkaluksi. Kaikkien näiden sovellusten avulla on mahdollista luoda entistä parempia ja moniulotteisempia videopelejä.

Lähteet

Bowling, Michael, ja Manuela Veloso. 2002. “Multiagent learning using a variable learning rate”. *Artificial Intelligence* 136 (2): 215–250.

Burnetas, Apostolos N, ja Michael N Katehakis. 1997. “Optimal adaptive policies for Markov decision processes”. *Mathematics of Operations Research* 22 (1): 222–255.

Galway, Leo, Darryl Charles ja Michaela Black. 2008. “Machine learning in digital games: a survey”. *Artificial Intelligence Review* 29 (2): 123–161.

Goldberg, David E, ja John Henry Holland. 1988. “Genetic algorithms and machine learning”.

Hagelbäck, Johan. 2015. “Hybrid pathfinding in StarCraft”. *IEEE Transactions on Computational Intelligence and AI in games* 8 (4): 319–324.

Hausknecht, Matthew, Joel Lehman, Risto Miikkulainen ja Peter Stone. 2014. “A neuroevolution approach to general atari game playing”. *IEEE Transactions on Computational Intelligence and AI in Games* 6 (4): 355–366.

Justesen, Niels, Philip Bontrager, Julian Togelius ja Sebastian Risi. 2019. “Deep learning for video game playing”. *IEEE Transactions on Games* 12 (1): 1–20.

Kaelbling, Leslie Pack, Michael L Littman ja Andrew W Moore. 1996. “Reinforcement learning: A survey”. *Journal of artificial intelligence research* 4:237–285.

Lucas, Simon M, ja Graham Kendall. 2006. “Evolutionary computation and games”. *IEEE Computational Intelligence Magazine* 1 (1): 10–18.

Mozgovoy, Maxim, ja Iskander Umarov. 2011. “Behavior capture: Building believable and effective AI agents for video games”. *International Journal of Arts and Sciences.*

Riley, Tonya. 2017. “Artificial intelligence goes deep to beat humans at poker”. Viitattu 12. huhtikuuta 2021. http://www.sciencemag.org/news/2017/03/artificial-intelligence-goes-deep-beat-humans-poker.

Samuel, Arthur L. 2000. “Some studies in machine learning using the game of checkers”. *IBM Journal of research and development* 44 (1.2): 206–226.

Synnaeve, Gabriel, ja Pierre Bessiere. 2015. “Multiscale Bayesian modeling for RTS games: An application to StarCraft AI”. *IEEE Transactions on Computational intelligence and AI in Games* 8 (4): 338–350.

Tan, Chin Hiong, Kay Chen Tan ja Arthur Tay. 2011. “Dynamic game difficulty scaling using adaptive behavior-based AI”. *IEEE Transactions on Computational Intelligence and AI in Games* 3 (4): 289–301.

Tesauro, Gerald. 2002. “Programming backgammon using self-teaching neural nets”. *Artificial Intelligence* 134 (1-2): 181–199.

Weiss, Eric A. 1992. “Biographies: Eloge: Arthur Lee Samuel (1901-90)”. *IEEE Annals of the History of Computing* 14 (3): 55–69.

Vinyals, Oriol, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev ym. 2019. “Grandmaster level in StarCraft II using multi-agent reinforcement learning”. *Nature* 575 (7782): 350–354.

Yannakakis, Georgios N, ja Julian Togelius. 2014. “A panorama of artificial and computational intelligence in games”. *IEEE Transactions on Computational Intelligence and AI in Games* 7 (4): 317–335.

Zhen, Jacky Shunjie, ja Ian Watson. 2013. “Neuroevolution for micromanagement in the real-time strategy game StarCraft: Brood War”. Teoksessa *Australasian Joint Conference on Artificial Intelligence,* 259–270. Springer.