

Bottien strategisen käyttäytymisen parantaminen tekoälymenetelmien avulla: Esimerkkinä NHL 25 - videopeli

Tietotekniikka
Tietotekniikan laitos, Teknillinen tiedekunta
Kandidaatintutkielma

Laatija:
Leevi Jälkö

Huhtikuu 2025

Kandidaatintutkielma
Tietotekniikan laitos, Teknillinen tiedekunta
Turun yliopisto

Tutkinto-ohjelma: Tietotekniikka

Tekijä: Leevi Jälkö

Otsikko: Bottien strategisen käyttäytymisen parantaminen tekoälymenetelmien avulla:
Esimerkkinä NHL 25 -videopeli

Sivumäärä: 27 sivua, 3 liitesivua

Päivämäärä: Huhtikuu 2025

Tämä kandidaatin tutkielma käsittelee tekoälypohjaisten bottien (tietokoneen ohjaama hahmo) kehitystä ja strategista päätöksentekoa sekä käyttäytymistä videopeleissä. Tutkimuksessa tarkastellaan erilaisia menetelmiä bottien luomiseksi ja analysoidaan, miten tekoälymenetelmiä voidaan akateemisen kirjallisuuden perusteella hyödyntää niiden toiminnan parantamiseksi. Työn tavoitteena on ymmärtää, miten tekoälymenetelmät voivat tehdä boteista luonnollisia ja ihmismäisiä.

Tutkimuksen perusteella keskeisiä menetelmiä bottien kehityksessä ovat tilakoneet, käyttäytymispuut, päätöspuut, neuroverkot sekä vahvistus- ja jäljitelmäoppiminen. Nämä menetelmät tarjoavat eri tasoisia ratkaisuja bottien toiminnan ohjaamiseen ja pelitilanteiden mukauttamiseen. Työssä keskitytään erityisesti NHL 25 -pelin mahdollisiin tekoälymenetelmiin sekä pelin bottien kehitykseen. Boteilla on merkittävä rooli pelinkehityksessä, erityisesti pelin immersiiivisyyden ja haasteellisuuden parantamisessa.

Jatkossa bottien kehitystä voidaan edistää testaamalla tekoälymenetelmiä käytännössä sekä tutkimalla esimerkiksi generatiivisen tekoälyn mahdollisuuksia bottien kehityksessä. Tämän tutkielman tulokset tarjoavat perustan tuleville tutkimuksille ja pelisuunnittelun kehittämislle tekoälyohjattujen bottien osalta.

Avainsanat: tekoäly, botti, koneoppiminen, neuroverkot, käyttäytymispuut, videopelit

Sisällysluettelo

1	Tekoäly ja botit verkkomonipeleissä	1
1.1	Botit pelikokemuksen rikastajina	2
1.2	Eettiset ja käytännölliset kysymykset	3
1.3	Tutkimuskysymykset ja kirjallisuus.....	4
2	Menetelmät bottien luomiseen	6
2.1	Sääntöpohjaiset Järjestelmät ja Tilakoneet	7
2.2	Käyttäytymispuut.....	10
2.3	Päätöspuut	12
2.4	Koneoppiminen ja Keinotekoiset Neuroverkot.....	14
2.5	Vahvistus- ja jäljitelmäoppiminen	16
2.6	Hybridimenetelmät.....	19
3	Kirjallisuudessa esiintyneiden tekoälymenetelmien soveltuvuus NHL 25:n kaltaisiin peleihin	22
4	Yhteenveto	26
	Lähdeluettelo	28

Kuvat

Kuva 1 Vahvistusoppimisen tilat	16
Kuva 2 FALCON menetelmän kolmikanavainen arkkitehtuuri	20
Kuva 3 Näyttökuvasta pelistä NHL 91	22
Kuva 4 Näyttökuvasta pelistä NHL 25	24

Taulukot

Taulukko 1 Bottien luomiseen käytettävät menetelmät	7
Taulukko 2 Kuvaa FSM:n toimintaa	9

1 Tekoäly ja botit verkkomonipeleissä

Tekoäly (engl. *Artificial Intelligence*, AI) ja sen kehitys perustuvat ajatukseen, että koneet voivat jäljitellä ihmisen ajatteluprosesseja. Tämä on vaatinut laajaa tutkimusta siitä, kuinka aivot käsittelevät tietoa, sopeutuvat uusiin tilanteisiin ja ratkaisevat ongelmia. Tästä syntynyt ymmärrys on johtanut merkittäviin teknologisiin innovaatioihin, kuten tekoälypohjaisten tekniikoiden ja sovellusten kehittämiseen. Tässä tutkielmassa perehdytään tietokoneen ohjaamia hahmoja koskeviin tekoälytekniikoihin. Nämä tekniikat kykenevät tunnistamaan malleja, tekemään ennusteita ja tukemaan päätöksentekoa. Tekoällyn nopea kehitys luo jatkuvasti uusia tutkimus- ja innovointimahdollisuuksia ja tuo myös esille eettisiä ja turvallisuuden liittyviä haasteita. (Caltech, 2022)

Tekoällyn osa-alue, koneoppiminen, on nykyään olennainen osa videopelien kehitystä, sillä sen avulla pelihahmoille voidaan luoda älykkäitä ja joustavia käyttäytymismalleja. Koneoppimista hyödynnetään peleissä monin eri tavoin, kuten parantamaan hahmojen kykyä reagoida pelaajan toimintaan, suunnittelemaan strategioita ja haastamaan pelaajaa taitavasti. Koneoppimisen avulla voidaan tarjota esimerkiksi yksilöllisempi ja dynaamisempi pelikokemus, koska sitä käyttämällä tietokoneen ohjaamat hahmot voivat muokata omaa toimintaansa pelaajan käyttäytymisen mukaan. Tekoäly-tekniikoiden kehittyminen on mahdollistanut myös monimutkaisempien ja realistisempien tilanteiden luomisen peleissä, kuten vihollisotilaiden reagoinnin muuttamisen, mikäli ne toistuvasti kärsivät samoista heikkouksista. Koneoppimisalgoritmit tekevät peleistä paitsi viihdyttäviä myös haastavia ja immersiivisiä. Immersiivisyydellä tarkoitetaan kokemusta, jossa pelaaja uppoutuu täysin pelimaailmaan ja kokee sen ympäristön, tapahtumat ja vuorovaikutukset aidon tuntuksina, ikään kuin olisi itse osa peliä. Tällainen syvä uppoutuminen syntyy, kun järjestelmä mukautuu pelaajan taitoihin ja päätöksiin, mikä tekee pelikokemuksesta entistä elämyksellisemmän. (Aydin;ym., 2023)

Aikaisemmin tekoällyn tutkimus on hyödyntänyt kehityksessään yksinkertaisempien pelien esimerkiksi shakin ja Go:n käyttöä. Tekoälykentän voidaan katsoa siirtyneen videopelisiin vuonna 2001, jolloin syntyi perusta pelitekoälylle. Videopelien koneoppimisen varhaisessa vaiheessa keskityttiin erityisesti pelien pelaamiseen, ei-pelattavien hahmojen (engl. *Non playable Character*, NPC) käyttäytymiseen ja reittien suunnitteluun. Koneoppimista tutkittiin muun muassa bottiarkkitehtuurin ja adaptiivisen käyttäytymisen kautta. Viime vuosina nykyaikaisempaa peliä nimeltä StarCraft on aloitettu käyttämään testialustana uusille tekniikoille ja menetelmille. Tutkielma keskittyy nykyaikaisiin menetelmiin bottien kehityksessä. Videopelit onkin vakiinnuttanut asemansa yhtenä suurimpana viihdeteollisuuden alana. Videopelit käyvät myös hyvinä opetusmenetelminä ja monissa yliopistoissa on opetussuunnitelmia, jotka liittyvät videopelien kehittämiseen. (Thompson, 2024)

Bottien asema verkkomoninpeleissä (MMO, engl. *Massively Multiplayer Online game*) on yhä keskeisempi pelikehityksessä ja pelaajien kokemuksessa. Vaikka sekä botit että ei-pelaajahahmot (engl. *Non Playable Characters*, NPC) ovat entiteettejä, joita pelaajat eivät hallitse, niiden toiminta ja tarkoitus eroavat merkittävästi. NPC:t, jotka ovat pelikehittäjien ohjelmoimia ja rajoitetusti vuorovaikutteisia hahmoja, palvelevat usein tarinankerronnan tai pelimaailman elävöittämisen tarkoituksia. Esimerkiksi NPC:t voivat antaa pelaajille tehtäviä tai tarjota kaupankäynti mahdollisuuksia, mutta niiden käyttäytyminen on ennalta määrättyä ja rajoitettua. (Rouse, 2024) Kuitenkin monessa artikkelissa puhutaan NPC-hahmoista bottien sijaan, vaikka artikkeleissa tarkoitetaan peliin mukautuvia sekä itseohjautuvia hahmoja. Sana NPC on tunnetumpi sanamuoto boteille ja monesti nämä saatetaan sekoittaa keskenään. (Rouse, 2024)

Botteja sen sijaan käytetään usein pelaajien korvaamiseksi tai pelin täyttämiseksi, ja ne on ohjelmoitu yrittämään voittaa peli. Botit voivat hyödyntää koneoppimisalgoritmeja, jotka mahdollistavat niiden sopeutumisen ja oppimisen pelaajien toiminnoista. Tämä tekee niistä kykenevämpiä reagoimaan dynaamisesti pelitilanteisiin ja olemaan vuorovaikutuksessa pelaajien kanssa, mikä lisää haasteita ja syvyyttä peliin. Yhteenvedona voidaan todeta, että vaikka NPC:t rikastuttavat pelimaailmaa, botit tuovat mukanaan kilpailullista dynamiikkaa, mikä tekee niistä erityisiä verkkomoninpeleissä. (Rouse, 2024)

1.1 Botit pelikokemuksen rikastajina

Pelejä pidetään yhtenä vanhimmista ihmisten sosiaalisista vuorovaikutustavoista. Pelejä pelataan monista eri syistä, kuten opetus- ja oppimisvälineenä, kulttuurisina ja sosiaalisina sidostapahtumina tai ihan vain nautinnon vuoksi. Tekoälymenetelmien avulla suunnitellut botit mahdollistavat pelaajan ja pelin realistisemmän ja dynaamisemmän vuorovaikutuksen, mikä lisää pelinautintoa. Ihmismäisesti käyttäytyvät botit pystyvät luomaan aidontuntuista kohtaamisista, jotka ovat keskeisiä erityisesti monimutkaisissa roolipeleissä ja toimintapeleissä, joissa bottien vuorovaikutus on keskeinen osa pelin juonta tai tavoitteita. (Milani;ym., 2023)

Bottien tarkoitus on luoda haastetta, koukuttavuutta ja uusia variaatioita verkkomonipeleihin. Immersio on pelien yksi tärkeimmistä ominaisuuksista tuottamaan nautintoa pelaajalle. Mitä suurempi vetovoima, sitä syvempi immersio, riippumatta pelaajan pelikokemuksesta. Kun pelaaja uppoutuu peliin ensimmäisellä kerralla, pitää hänen tuntea haluavansa pelata peliä myös jatkossa. Pelaajien säilyttäminen on tärkeää verkkomonipelien kannalta, sillä monet niistä ovat ilmaisia ladata ja tulonlähde pitää hankkia pelin sisäisillä ostoilla. (Georgios, 2014)

Botit ovat tärkeä osa myös pelien kehittämistä, sillä niiden avulla pelisuunnittelijat voivat vähentää manuaalisen työn tarvetta bottien käyttäytymisen ohjelmoinnissa. Perinteinen

lähestymistapa, jossa botit toimivat ennalta määrättyjen sääntöjen mukaan, on usein aikaa vievä ja rajallinen. Koneoppimistekniikat tarjoavat ratkaisun, joka tekee boteista monipuolisempia ja joustavampia. Tämä mahdollistaa paremmin mukautuvat ja pelaajien toimintaan reagoivat hahmot, mikä on erityisen tärkeää suurissa avoimen maailman peleissä, kuten World of Warcraft ja Grand Theft Auto. Tutkimus tekoälymenetelmien ja ihmisen kaltaisen käyttäytymisen välisestä suhteesta on tärkeää paitsi pelien, myös laajemman tekoälytutkimuksen näkökulmasta. Videopelit tarjoavat erinomaisen ympäristön testata ja kehittää ihmismäisesti käyttäytyviä tekoälymalleja, joiden toiminta voidaan siirtää muihin sovelluksiin, kuten robotiikkaan tai automatisoituun ajamiseen. Ymmärtämällä, miten ihmiset havaitsevat ja arvioivat tekoälybottien ihmismäisyyttä, voidaan parantaa tekoälymallien toimintaa niin peleissä kuin muillakin elämän osa-alueilla. Botit ja niiden kehitys tarjoavat mahdollisuuden luoda pelejä, jotka eivät pelkästään viihdytä vaan myös haastavat ja palkitsevat pelaajia tavalla, joka tuntuu aidolta ja merkitykselliseltä. (Milani;ym., 2023)

1.2 Eettiset ja käytännölliset kysymykset

Bottien suunnittelu ja kehittäminen verkkomoninpeleissä perustuu nykyisin pitkälti älykkäisiin ohjelmointiratkaisuihin ja oppiviin järjestelmiin. Generatiivinen tekoäly (engl. *generative AI*, GT) tarjoaa uusia, vielä tutkimattomia mahdollisuuksia syventää pelaajien immersiota. GT on tekoälyn osa-alue, joka keskittyy uusien, alkuperäisten sisältöjen tuottamiseen sen sijaan, että se vain analysoisi tai luokittelisi jo olemassa olevaa dataa. Generatiivisen tekoälyn soveltaminen NPC:n ja bottien luomisessa tuo esiin mielenkiintoisia eettisiä ja käytännöllisiä haasteita, joita nykyiset tutkimukset eivät ole täysin kattaneet. Yksi tällainen tutkimusaukko liittyy siihen, miten GT voi vaikuttaa pelaajien kokemukseen ja siihen, voivatko pelaajat erottaa todelliset ihmiset tekoälytekniikoilla luoduista boteista. Tämä nostaa esiin kysymyksiä paitsi pelikokemuksen autenttisuudesta, myös eettisistä ja lainsäädännöllisistä seikoista, joita ei ole vielä tutkittu. (Dimov, 2023)

Esimerkiksi pelikehittäjät voivat käyttää generatiivista tekoälyä luodakseen vaikutelman vilkkaammasta pelikokemuksesta MMO-peleissä, mikä voi parantaa pelaajien kokemusta ja pidentää pelien elinkaarta. Toisaalta, jos pelaajille ei tiedoteta esimerkiksi koneoppimisen käytöstä, se voi johtaa ongelmiin avoimuuden ja rehellisyyden kannalta, mikä voi rikkoa kilpailulainsäädäntöä. Tämän tyyppiset haasteet osoittavat, että tutkimusta tarvitaan enemmän generatiivisen tekoälyn käytöstä erityisesti verkkomonipelien kontekstissa. On ratkaisevan tärkeää tutkia, miten tällaiset teknologiat voivat vaikuttaa pelikokemukseen, pelaajien sosiaaliseen vuorovaikutukseen ja pelien kilpailullisiin aspekteihin. (Dimov, 2023) Tämä tutkimusaukko osoittaa, että generatiivisen tekoälyn käyttö

verkkomoninpeleissä ei ole pelkästään tekninen kysymys, vaan siihen liittyy myös monia sosiaalisia ja oikeudellisia ulottuvuuksia, jotka vaativat lisätutkimusta.

1.3 Tutkimuskysymykset ja kirjallisuus

Tutkimuskysymys 1 (TK1): Mitä menetelmiä akateemisessa kirjallisuudessa on esitetty bottien/tietokoneen automaattisesti ohjaamien hahmojen luontiin?

Tutkimuskysymys 2. (TK2): Miten tekoälyä on käytetty parantamaan bottien strategista käyttäytymistä eri videopeligenreissä?

Esimerkkitapaus: Miten tekoälymenetelmiä voidaan soveltaa NHL 25 -pelin boteissa?

Tutkielma tarkastelee bottien ja generatiivisen tekoälyn rooleja videopelien kehityksessä. Se keskittyy bottien luomiseen käytettyihin menetelmiin ja GT:n hyödyntämiseen strategisen käyttäytymisen parantamisessa eri videopeligenreissä. Tutkimuksessa analysoidaan eri lähteistä saatua tietoa, vertaillaan menetelmien vahvuuksia ja heikkouksia sekä pohditaan niiden kehityssuuntia. Lopuksi esitetään tutkimustulokset ja arvioidaan niiden merkitystä aikaisempaan tutkimukseen nähden sekä hahmotellaan mahdollisia jatkotutkimusaiheita.

Johdanto käsittelee bottien kehityksen taustaa ja historiaa sekä kerrotaan aiheen tärkeydestä. Toisessa luvussa analysoidaan erilaisia tekoälymenetelmiä, kuten sääntöpohjaisia malleja, käyttäytymis- ja päätöspuita, koneoppimista sekä neuroverkkoihin perustuvia ratkaisuja. Tarkastelussa huomioidaan eri videopeligenret, kuten rooli-, toiminta- ja strategiapelit, ja miten Generatiivinen tekoäly on parantanut bottien päätöksentekoa ja pelidynamiikkaa. Kolmannessa luvussa keskitytään NHL 25 -pelin botteihin niiden kehitykseen. Käydään läpi pelin uudet järjestelmät, kuten ICE-Q ja Empowered AI, jotka parantavat bottien reagointikykyä ja päätöksentekoa. Lisäksi vertaillaan eri tekoälymenetelmiä ja niiden soveltuvuutta eri pelimuotoihin immersiiivisen pelikokemuksen luomisessa. Tutkielman lopuksi kootaan yhteen tutkimuksen keskeiset löydökset ja vastataan tutkimuskysymyksiin. Kokonaisuutena tutkimus auttaa ymmärtämään bottien ja GT:n vaikutusta pelikehitykseen sekä avaa uusia näkökulmia alan tulevaisuuteen.

Tutkielmassa on hyödynnetty laajaa kirjallisuusaineistoa, joka kattaa niin tieteelliset artikkelit, kirjat kuin verkkolähteetkin. Päälähteinä työssä ovat toimineet erityisesti Yannakakis ja Togeliusen kirja "AI and Games" vuodelta 2018 (Yannakakis;ym., 2018) sekä Muhtar Ç. Uludağlın ja Kaya Oğuzin artikkeli "Non-player character decision-making in computer games" vuodelta 2023 (Uludağlı;ym.,

2023). AI and Games -kirjassa selitetään laajasti, kuinka tekoälytekniikoita voidaan käyttää pelien pelaamiseen, pelisisällön luomiseen ja pelaajien mallintamiseen. Uludağlinin ja Oğuzinin artikkelissa kartoitetaan bottien päätöksenteossa käytettyjä menetelmiä ja tunnistetaan eri kategorioita näistä. Nämä päälähteet ovat melko tuoreita julkaisuja, mutta myös vanhempia lähteitä on otettu huomioon.

Lähdettä (Fink;ym., 2007) on käytetty kuvaamaan koneoppimiseen perustuvaa menetelmää, jolla bottien käyttäytymistä voidaan mallintaa pelin visuaalisen datan avulla. Tässä tutkimuksessa on koettu hyödylliseksi myös vanhempien lähteiden käyttäminen, koska ne tarjoavat historiallista perspektiiviä ja peruseräitä, joille uudemmat tutkimukset ovat perustuneet. NHL 25 -pelistä ei löydy akateemista kirjallisuutta, joten lähteitä on käytetty asiantuntijoiden/pelaajien kirjoittamista artikkeleista. Lähteiden joukossa on myös artikkeleita, joissa kuvataan historiallisia näkökohtia pelibottien kehityksestä. Näissä tuodaan esiin, kuinka perinteisistä yksinkertaisista sääntöpohjaisista boteista on siirrytty kohti monimutkaisempia tekoälymalleja.

Käytettyä kirjallisuutta on muokattu tutkielmaan sopivaksi valikoimalla aineistosta tutkimuskysymyksien kannalta olennaisimmat osiot. Eri lähteistä saatua tietoa on yhdistelty kattavan kuvan luomiseksi tekoälymenetelmien vaikutuksesta bottien strategiseen käyttäytymiseen. Lisäksi tutkimuksessa pyritään vertailemaan eri tutkimusten tuloksia sekä arvioimaan niiden sovellettavuutta erityisesti verkkomoninpeilyympäristöihin, kuten NHL-pelisarjaan. Suuri osa GT:n ja bottien tutkimuksesta on varsin tuoretta ja se kehittyy jatkuvasti. Tämän vuoksi on pyritty löytämään pääosin mahdollisimman uusia ja relevantteja artikkeleita, jotka tarjoavat ajankohtaista tietoa.

2 Menetelmät bottien luomiseen

Bottien luomiseen käytetään monia erilaisia menetelmiä, moniin eri käyttötarkoituksiin. Tässä luvussa perehdymme yleisimpiin menetelmiin ja miten niitä käytetään erilaisilla videopeleissä. Menetelmät voidaan jakaa bottien päätöksen teon mukaan, joiden avulla ne muuttavat pelien kulkua.

Ihmisen kaltaisten bottien rakentaminen vaatii monien haastavien ongelmien ratkaisemista. Näihin kuuluu esimerkiksi pelaajien toiminnan tarkkailu, heidän strategioidensa tunnistaminen, tekoälyteknologian kehittäminen sekä hahmojen uskottavuuden testaaminen. Yksi keskeinen haaste on määrittää, minkä pelaajan käyttäytymistä botin tulisi matkia, ja lisäksi kaikkien mahdollisten toimien manuaalinen ohjelmointi on aikaa vievää ja monimutkaista, varsinkin kun pelitilanteet voivat olla arvaamattomia. (Lauro;ym., 2024)

Taulukko 1 Bottien luomiseen käytettävät menetelmät

Menetelmä	Toiminta	Lähteet
Tilakoneet	Rajoitettu joukko tiloja tilansiirtymiä, jotka määrittävät hahmon käyttäytymisen	(Jagdale, 2021) (Uludağlı;ym., 2023)
Käyttäytymispuut	Toimivat hierarkkisena mallina, jossa botin päätöksenteko ja toiminnot on jaettu loogisiin osiin, kuten ehdollisiin tarkastuksiin ja toimintokutsuihin.	(Colledanchise;ym., 2018) (Matteo;ym., 2022)
Päätöspuut	Tarjoavat päätöksenteossa selkeän ja strukturoituneen tavan mallintaa ja hallita käyttäytymistä eri pelitilanteissa.	(Yannakakis;ym., 2018)
Koneoppiminen	Botit analysoivat dataa, tunnistavat kuvioita ja oppivat toistuvien harjoitusvaiheiden kautta valitsemaan optimaalisia toimintoja eri tilanteissa ilman manuaalisia sääntöjä.	(Fink;ym., 2007) (Uludağlı;ym., 2023) (Heaton, 2017) (Jolly;ym., 2007)
Vahvistusoppiminen	Botti toimii ympäristössä, saa palautetta (palkkioita tai rangaistuksia) ja oppii strategian.	(Yannakakis;ym., 2018)
Jäljitelmäoppiminen	Botti oppii suorittamaan tehtäviä matkimalla ihmisen tai toisen botin suorituksia.	(Lauro;ym., 2024)
Hybridimenetelmät	Botit yhdistävät useita tekoäly- ja koneoppimistekniikoita, kuten sääntöpohjaisia malleja ja syväoppimista, hyödyntääkseen sekä ihmisen määrittelemiä sääntöjä että datasta oppimia kaavoja päätöksenteossa ja käyttäytymisessä monimutkaisissa ympäristöissä.	(Shu;ym., 2016) (Yannakakis;ym., 2018) (Almeida;ym., 2023)

2.1 Sääntöpohjaiset Järjestelmät ja Tilakoneet

Sääntöpohjaisia menetelmiä (engl. *Rule-Based Systems*, RBS) käytettiin tekoälytutkimuksen alkuvaiheissa laajasti. Lähteen (Uludağlı;ym., 2023) mukaan niiden käyttö on kuitenkin vähentynyt, sillä monimutkaisen sääntöpohjaisten järjestelmien toteuttaminen on tehottomampaa ja vaikeampaa kuin esimerkiksi äärellisten tilakoneiden (engl. *Finite State Machine*, FSM) tai päätöspuiden (engl. *Decision Tree*, DT) hyödyntäminen. Sääntöpohjaiset järjestelmät koostuvat kolmesta pääosasta: tietokannasta, sääntökokoelmasta ja näiden välissä toimivasta päättäjstä (eng. *arbiter*). Tietokanta sisältää botin tuntemaan tiedon ympäristöstä ja muista hahmoista, kun taas sääntökokoelma sisältää if-

then-else-tyyppisiä sääntöjä, joiden mukaan toiminta suoritetaan. Päättäjät valitsee sääntökokoelmasta sopivan säännön tietokannan perusteella ja käynnistää siihen liittyvän toiminnon. (Uludağlı;ym., 2023)

Useimmat sääntöpohjaiset järjestelmät hyödyntävät Rete-matching-algoritmia, joka yhdistää oikeat säännöt tietokannan tietojen kanssa. Tunnettuja pelejä, joissa näitä algoritmeja on käytetty bottien käyttäytymisen ohjaamiseen ovat esimerkiksi Baldur's Gate ja Virtua Fighter. Sääntöpohjaisia järjestelmiä on tutkittu myös akateemisessa kontekstissa. Artikkelissa (Uludağlı;ym., 2023) tutkitaan sääntöpohjaisia järjestelmiä autopelissä, jossa päätöksenteossa hyödynnettiin sensoridataa sekä egosentrisestä (itsestä lähtöisin olevaa) että kolmannen persoonan näkökulmasta. Tulokset osoittivat, että kolmannen persoonan sensoritieto voi parantaa päätöksenteon laatua. Vaikka sääntöpohjaiset menetelmät voivat olla tehokkaita yksinkertaisissa tilanteissa, niiden laajamittainen käyttö kohtaa haasteita, kuten monimutkaisuuden kasvun ja tehokkuusongelmat. Tästä syystä nykyaikaisissa tekoälyjärjestelmissä suositaan usein muita menetelmiä, kuten tilakoneita tai todennäköisyyspohjaisia lähestymistapoja. (Uludağlı;ym., 2023)

Äärellisen tilan kone (FSM) on yksi vanhimmista ja eniten käytetyistä tekniikoista videopelien kehityksessä, ja sen juuret ulottuvat klassisiin peleihin kuten Pac-Man ja nykyaikaisempiin peleihin kuten Tomb Raider. FSM:n keskeinen ajatus on parantaa bottien älykkyyttä, jolloin pelistä saadaan dynaamisempi ja kiinnostavampi. Vaikka nykyaikaisessa pelikehityksessä on käytettävissä kehittyneitä koneoppimismenetelmiä. FSM on edelleen tärkeä tekniikka bottien käyttäytymisen toteutuksessa. (Jagdale, 2021)

FSM-malli perustuu joukkoon tiloja, jotka määrittelevät botin senhetkisen käyttäytymisen, sekä ehtoihin, jotka ohjaavat siirtymiä tilasta toiseen. Esimerkiksi Pac-Manissa FSM:t mahdollistivat bottien reaktiivisen käyttäytymisen pelaajan toimintaan. Varhaisessa pelinkehityksessä FSM:t tarjosivat yksinkertaisen, mutta tehokkaan tavan ohjelmoida botteja suorittamaan tehtäviä, kuten jahtaamista ja pakenemista. (Uludağlı;ym., 2023)

Alla oleva taulukko havainnollistaa FSM:n toimintamallia yksinkertaisessa vihollisbottien esimerkissä. Taulukossa toteutetaan botti, joka on pelaajan vihollinen. Botilla on ennalta määritellyt tilat: toimeton, jahdata, hyökätä, puolustaa, paeta ja kuollut. Siirtyminen tilasta toiseen tapahtuu, kun ennalta määritetyt ehdot täyttyvät, kuten pelaajan läsnäolo lähellä bottia, pelaajan hyökkäys tai botin alhainen elämien määrä. Niin kutsuttu pelisilmukka varmistaa jatkuvasti siirtymäehtojen täyttymisen ja päivittää FSM:n tilan. (Jagdale, 2021)

Taulukko 2 Kuvaa FSM:n toimintaa

Nykytila (botti)	Toiminta/Tapahtuma	Lopputila (botti)
Toimeton	Pelaaja on alueella	Hyökkää
Hyökkäys	Pelaaja ei ole enää alueella	Toimeton
Hyökkäys	Pelaaja hyökkää	Puolustautuu
Puolustautuu	Pelaaja pakenee	Hyökkää
Puolustautuu	Botin elämät ovat vähissä	Pakenee
Puolustautuu	Botin elämät ovat loppu	Kuollut

FSM:ien tehokkuudesta huolimatta niillä on myös rajoituksia. Kun tilojen määrä kasvaa, siirtymien määrä kasvaa eksponentiaalisesti, mikä tekee järjestelmästä monimutkaisen ja vaikeasti hallittavan. Tämä on johtanut tutkimuksiin hierarkkisten äärellisten tilakoneiden, (eng. *Hierarchical Finite State Machine*, HFSM) kehittämiseksi. HFSM mahdollistaa tilojen organisoimisen hierarkkisesti ja näin ollen vähentää siirtymien määrää. (Uludağlı,ym., 2023) HFSM mahdollistaa bottien käyttäytymisen joustavan hallinnan, koska se yhdistää eri tilat laajempiin toiminnallisiin kokonaisuuksiin, jotka voivat sisältää alitiloja ja näiden välisiä siirtymiä. Näin HFSM voi yhdistää esimerkiksi "Selviytyminen"-tehtävän, joka koostuu alitiloista "Toimeton" ja "Paeta", sekä toisen "Taistelu"-tehtävän, jossa "Hyökkää" ja "Puolusta" toimivat alitiloina. Tämä rakenne vähentää siirtymien määrää, sillä korkeammalla tasolla voidaan käsitellä laajempia toiminnallisia kokonaisuuksia ja yksinkertaistaa siirtymiä tilojen välillä. Lisäksi HFSM tarjoaa mahdollisuuden tarkastella tiloja joko yleisellä tasolla, jossa vain korkeamman tason tehtävät näkyvät tai tarvittaessa tarkentaa alitiloihin, mikä tekee mallista helpommin hallittavan ja luettavan. (Jagdale, 2021)

Bottien kehityksessä FSM tai HFSM voi olla optimaalinen ratkaisu silloin, kun botin toiminnot ovat ennalta määriteltyjä ja rajattuja. Vaikka koneoppimisen syväoppimistekniikoilla ja muilla tekomenetelmillä voitaisiin luoda dynaamisempia botteja, tällainen lähestymistapa voi tuntua pelaajista liian haastavalta tai epärealistiselta. Lisäksi tuhansien bottien koulutus ja toteutus esimerkiksi syväoppimisella saattaa kuormittaa järjestelmää tarpeettomasti. Siksi tilanteissa, joissa botti suorittaa vain muutamia tehtäviä ja pelaajille riittää säännönmukainen vuorovaikutus, FSM tai HFSM voi olla kustannustehokas ja toimiva valinta. (Jagdale, 2021)

2.2 Käyttäytymispuut

Käyttäytymispuut (*Behavior Trees*, BT) ovat tietokonepeliteollisuudessa kehitetty menetelmä, jonka tavoitteena on lisätä bottien ohjauksrakenteiden modulaarisuutta. Modulaarisuus on avainominaisuus, joka mahdollistaa koodin uudelleenikäytön, toiminnallisuuden inkrementaalisen suunnittelun ja tehokkaan testaamisen, mikä on erityisen tärkeää miljardien dollarien arvoisessa peliteollisuudessa. Käyttäytymispuut ovatkin nousseet vaihtoehdoksi perinteisesti käytetyille äärellisille tilakoneille, jotka ovat pitkään hallinneet bottien ohjausta. (Colledanchise;ym., 2018)

Vaikka FSM:t ovat tehokkaita suoraviivaisissa tilanteissa, niiden rajallisuudet korostuvat, kun tarvitaan monimutkaisia ja joustavia järjestelmiä Petri-verkkojen (eng. *Petri Nets*) tavoin. Petri-verkot tarjoavat vaihtoehdon FSM:ille rinnakkaisten järjestelmien suunnittelussa, käyttäytymispuut tarjoavat vaihtoehdon modulaaristen järjestelmien kehittämisessä. BT:iden avulla voidaan jakaa monimutkaiset toiminnot pienempiin, helposti hallittaviin osiin, joita voidaan yhdistellä ja käyttää uudelleen eri konteksteissa. (Colledanchise;ym., 2018)

Peliteollisuudesta BT:iden käyttö on levinnyt myös akateemiseen tutkimukseen ja muihin sovellusalueisiin, kuten robotiikkaan. Esimerkiksi Carnegie Mellon -yliopistossa käyttäytymispuuta on hyödynnetty robotisoitujen manipulointitehtävien suunnittelussa. BT:iden vahvuutena pidetään niiden modulaarista ja mukautuvaa rakennetta, jonka ansiosta yksittäiset käyttäytymiset voidaan helposti yhdistää osaksi laajempia, korkeamman tason toimintoja ilman, että niiden yhteyttä seuraaviin käyttäytymisiin tarvitsee määritellä yksityiskohtaisesti. Tämä joustavuus on tehnyt BT:istä myös hyödyllisen työkalun ei-asiantuntijoiden käytettäväksi, kuten robottien ohjelmoinnissa tai monimutkaisissa kirurgisissa toimenpiteissä, joissa vaaditaan selkeää ja muokattavaa syntaksia. (Colledanchise;ym., 2018)

BT:iden käyttöä tehostaa niiden selkeä hierarkkinen rakenne, joka mahdollistaa eri toimintojen suorittamisen rinnakkain tai sarjassa, tarpeen mukaan. Tämä hierarkkisuus tekee niistä erityisen tehokkaita dynaamisissa ympäristöissä, joissa olosuhteet voivat muuttua nopeasti. Artikkelissa (Matteo;ym., 2022) esitellyn esimerkin mukaan käyttäytymispuut voidaan helposti mukauttaa monimutkaisiin tilanteisiin lisäämällä niihin uusia haaroja tai muuttamalla olemassa olevien toimintojen prioriteetteja ilman merkittävää uudelleensuunnittelua. Tämä lisää kehityksen joustavuutta ja säästää aikaa, erityisesti suurissa projekteissa.

Lisäksi BT:iden visuaalinen esitystapa, jossa toiminnallisuudet on järjestetty puumaisesti loogisiin kokonaisuuksiin, helpottaa sekä ohjelmoijien että suunnittelijoiden välistä yhteistyötä. Lähteessä mainitaan, että tällainen lähestymistapa on osoittautunut hyödylliseksi erityisesti suurissa kehitystiimeissä, joissa eri asiantuntijat voivat työskennellä samanaikaisesti puun eri osien parissa.

Tämä parantaa paitsi työn tehokkuutta myös sen laatua, sillä muutokset eivät helposti aiheuta odottamattomia virheitä muualla järjestelmässä. (Matteo;ym., 2022)

Käyttäytymispuiden kyky mahdollistaa selkeä ja helposti ymmärrettävä reaktiivisuus tekee niistä ihanteellisen työkalun myös ympäristöissä, joissa bottien on sopeuduttava pelaajien odottamattomiin toimintoihin. Esimerkit osoittavat, että BT:t pystyvät käsittelemään erilaisia skenaarioita siten, että bottien käyttäytyminen vaikuttaa luonnolliselta ja loogiselta. Tämä ominaisuus on keskeinen erityisesti peleissä, joissa käyttäjäkokemus korostuu. Käyttäytymispuut eivät ole pelkästään tehokas työkalu modulaarisuuden ja reaktiivisuuden näkökulmasta, vaan ne myös tukevat käyttäjäystävällistä suunnittelua ja tiimityötä. Tämä tekee niistä merkittävän vaihtoehdon erityisesti myös pelialan ulkopuolella, esimerkiksi robotiikan ja autonomisten järjestelmien suunnittelussa, missä käyttäytymisen ennakoitavuus ja muokattavuus ovat kriittisiä ominaisuuksia. (Matteo;ym., 2022)

Reaaliaikaisissa strategia- ja ensimmäisen persoonan ammutapeleissa (engl. *First-Person Shooter*, FPS) käyttäytymispuut ovat osoittautuneet tehokkaiksi työkaluiksi monimutkaisten päätöksentekojärjestelmien luomisessa. Reaaliaikaisissa strategia peleissä, kuten DEFCON ja StarCraft, BT:tä hyödynnetään useilla eri abstraktiotasoilla yksittäisten yksiköiden tehtävien ohjauksesta aina koko armeijan strategiseen hallintaan. Näissä peleissä päätöksenteko on erityisen haastavaa, sillä pelaajan tulee jatkuvasti sopeutua muuttuvaan pelitilanteeseen ja hallita suuria määriä yksiköitä niiden ominaisuuksien perusteella. BT:t mahdollistavat tämän jakamalla toiminnot loogisiin, hierarkkisiin osiin, jotka tukevat sekä taktisia että strategisia tavoitteita. Esimerkiksi BT:tä voidaan käyttää päättämään, mitä yksiköitä liikutetaan tietylle alueelle, kuinka ne priorisoivat hyökkäyskohteensa tai miten ne muodostavat yhteistyöhön perustuvia ryhmiä. (Matteo;ym., 2022)

FPS-peleissä, kuten Halo 2:ssa, BT:iden rooli korostuu dynaamisissa ympäristöissä, joissa pelaajan vuorovaikutus bottien kanssa on jatkuvaa ja nopeatempoista. Näissä peleissä BT:t voivat hallita hahmojen monipuolisia käyttäytymismalleja, kuten ympäristön havainnointia, riskien arviointia ja toimintaan perustuvaa päätöksentekoa. Lisäksi BT:itä on kehitetty tukemaan uusia toiminnallisuuksia, kuten päätöksentekoa tunteiden, ajan ja riskien perusteella, mikä tekee pelin boteista entistä luonnollisemman tuntuisia. Kehittyneimmissä tapauksissa BT:t oppivat pelaajien toimista ja mukauttavat käyttäytymistään sen perusteella, mitä ne ovat oppineet pelin aikana. (Matteo;ym., 2022)

Näiden esimerkkien perusteella käyttäytymispuut tarjoavat joustavan ja monipuolisen ratkaisun monenlaisiin pelikehityksen haasteisiin. Niiden modulaarisuus ja mukautuvuus tekevät niistä soveltuvia strategiaan ja yksityiskohtaisiin tehtäviin, mikä mahdollistaa entistä realistisemmän ja reaktiivisemmän pelikokemuksen.

2.3 Päätöspuut

Bottien päätöksenteko peleissä perustuu erilaisten hakualgoritmien käyttöön, jotka mahdollistavat optimaalisten valintojen tekemisen monimutkaisessa pelimaailmassa. Verkkomoninpeleissä, kuten Battle Royale- ja MOBA (engl. Multiplayer Online Battle Arena) -peleissä, bottien on tehtävä päätöksiä reaaliaikaisesti, usein yhteistyössä tai vastakkain oikeiden pelaajien kanssa. Tämä asettaa lisähaasteita, sillä pelitilanne voi muuttua nopeasti pelaajien toimien perusteella.

Esimerkiksi yksinpelissä voidaan hyödyntää yksinkertaisia hakualgoritmeja, kuten syvyysuuntaista hakua (engl. *Depth-First Search*, DFS) ja leveyssuuntaista hakua (engl. *Breadth-First Search*, BFS), jotka tutkisivat pelin tiloja ja etsivät parasta mahdollista ratkaisua. DFS lähestyy ongelmaa siten, että se tutkii yhtä polkua niin syväälle kuin mahdollista ennen kuin palaa takaisin ja tarkastelee muita vaihtoehtoja. Tämä voi olla tehokasta, jos ratkaisun syvyys on ennakoitavissa, mutta se saattaa olla tehotonta, jos peliympäristö on laaja ja monitahoinen. Sen sijaan BFS tutkii kaikki mahdolliset toimet tietyssä pelitilassa ennen siirtymistä syvempiin tasoihin. Tämä takaa lyhimmän polun löytämisen, mutta voi olla muistitehokkuuden kannalta haasteellista monimutkaisissa pelitilanteissa, joissa tarvitaan paljon muistia. (Yannakakis;ym., 2018) Moninpeleissä, joissa kartat ja tilanteet ovat jatkuvasti muuttuvia, nämä algoritmit voivat olla hyödyllisiä reittien laskennassa, mutta ne eivät yksinään riitä strategiseen päätöksentekoon.

Näitä menetelmiä voidaan käyttää myös Ms. Pac-Manissa, jossa A*-algoritmi voi ohjata botin liikkumista pelissä. A* ottaa huomioon alkuperäisen solmun etäisyyden ja arvioidun etäisyyden tavoitetilaa, ja sen avulla botin liikkumista voidaan ohjata optimaalisesti, ottaen huomioon pelissä olevat vaarat. Kahden pelaajan vastakkaisessa pelissä, kuten shakissa, bottien päätöksenteko on monimutkaisempaa, sillä molemmat pelaajat vaikuttavat toistensa valintoihin. Minimax-algoritmi on erityisesti suunniteltu tällaiseen vuorovaikutteiseen ympäristöön, jossa pelaajien päätökset eivät ole itsenäisiä, vaan ne riippuvat toisten pelaajien toimista. Minimax tutkii pelipuun solmuja ja arvioi pelin lopputuloksen kullekin mahdolliselle siirrolle, valiten aina sen siirron, joka maksimoi oman pelaajan hyödyn ja minimoi vastustajan hyödyn. Tämä tekee algoritmista tehokkaan työkalu juuri strategisessa päätöksenteossa, mutta sen käyttö monimutkaisemmissa peleissä, voi olla haasteellista. Pelipuun koko kasvaa eksponentiaalisesti, joten kaikkien mahdollisten siirtojen tutkiminen käy nopeasti liian raskaaksi nykyaikaisilla laskentatehoilla. Tämän vuoksi hakua usein rajoitetaan tietyn syvyyden jälkeen, ja pelitilanteita arvioidaan nopeiden arviointifunktioiden avulla, kuten pelissä olevien nappuloiden määrän vertailulla. (Yannakakis;ym., 2018) Nykyisissä verkkomoninpeleissä Minimaxin kaltaiset strategiset päätöksentekomallit voivat auttaa botteja analysoimaan vastustajien liikkeitä ja ennakoimaan, miten pelaajat todennäköisesti toimivat eri tilanteissa, esimerkiksi pelaajan taisteluissa tai resurssien hallinnassa.

Monte Carlo -puuhakeminen (engl. *Monte Carlo Tree Search*, MCTS) on tehokas menetelmä, joka on erityisesti suunniteltu pelien, joissa on suuri haarautumiskerroin tai joissa ei ole helposti määritettävää arviointifunktiota, ratkaisemiseksi. Toisin kuin perinteiset hakualgoritmit esimerkiksi Minimax, MCTS ei etsi kaikkia pelin mahdollisia polkuja tasaisesti, vaan keskittyy lupaavimpaan haaraan. Tämä mahdollistaa syvempien polkujen tutkimisen, vaikka peliympäristö olisi erittäin laaja ja kompleksinen. MCTS käyttää niin kutsuttuja simuloituja pelikerroksia (engl. *rollouts*), jotka arvioivat pelitilan laatua. Simulaatiot pelataan loppuun asti, ja niiden tulokset auttavat arvioimaan, kuinka todennäköisesti peli päättyy voittoon tai tappioon. Tämä menettely on erityisen hyödyllinen, kun pelissä ei ole täydellistä tietoa tai kun tilannetta on vaikea arvioida suoraan perinteisillä arviointifunktioilla. (Yannakakis;ym., 2018) Moninpeleissä, joissa jokainen peli voi olla erilainen pelaajien valintojen takia, MCTS voi auttaa bottien päätöksenteossa esimerkiksi valitsemalla optimaalisia siirtoja tilanteen mukaan ilman tarvetta ennalta määritellyille sääntökokoelmille.

MCTS:n pääpiirre on sen kyky toimia ilman laajaa arviointifunktiota, mikä tekee siitä erinomaisen vaihtoehdon tilanteissa, joissa hyvä tilan arviointi on vaikeaa, kuten monissa epädeterministisissä peleissä. Algoritmin pääsilmut koostuu neljästä vaiheesta: valinta, laajentaminen, simulointi ja takaisinlevitys. Näissä vaiheissa algoritmi valitsee ensin, mihin solmuun tutustutaan seuraavaksi, laajentaa sen, simuloi pelitilanteen loppuun asti ja palauttaa tulokset takaisin ylös hakupuun oksille. Tämä prosessi mahdollistaa pelin dynaamisten tilojen tutkimisen tehokkaasti ja joustavasti, sillä se voi keskeyttää haun milloin tahansa ja toimia "milloin tahansa" -algoritmina. (Yannakakis;ym., 2018) Nykyajan kilpailullisissa verkkopeleissä, kuten FPS- ja RTS-peleissä, MCTS voi auttaa botteja mukautumaan pelaajien päätöksiin ja tekemään realistisia liikkeitä sen sijaan, että ne noudattaisivat ennalta määriteltyjä komentosarjoja.

MCTS on erityisen hyödyllinen peleissä, joissa pelin tilanne voi muuttua nopeasti ja päätöksenteko on usein tehtävä rajoitetussa ajassa. Esimerkiksi moninpeliskenaarioissa, joissa pelaajat tekevät jatkuvia taktisia päätöksiä, kuten hyökkäysten koordinointi tai alueiden hallinta, MCTS voi tarjota boteille realistisen ja adaptiivisen toimintamallin. Erityisesti se voi hyödyntää rajoitettua pelipuuta ja liittymispohjaista pelitilojen esitystapaa, jotka auttavat ratkaisemaan MCTS:n haasteita, kuten laskentatehon rajoitteet ja lyhyen aikarajan pelitilanteiden arvioinnissa. (Yannakakis;ym., 2018) Koska nykyaikaiset moninpelit vaativat bottien mukautumista vaihteleviin tilanteisiin ja eri pelaajien taitotasoihin, MCTS:n kaltaiset menetelmät voivat parantaa bottien realistisuutta ja monipuolisuutta ilman, että ne vaikuttavat pelin tasapainoon.

2.4 Koneoppiminen ja Keinotekoiset Neuroverkot

Koneoppimisen (engl. *Machine Learning*) avulla bottien käyttäytymistä voidaan mallintaa esimerkiksi pelin visuaalista dataa hyödyntäen. Tästä esimerkkinä on tutkimus (Fink;ym., 2007), jossa kehitettiin menetelmä bottien käyttäytymisen rekonstruoimiseksi eli uudelleenmallintamiseksi pelin graafisen ulostulon perusteella. Rekonstruoiminen tässä yhteydessä tarkoittaa sitä, että bottien käyttäytymistä pyritään luomaan uudelleen tarkkailemalla vain pelin visuaalista ulostuloa, kuten näytöllä näkyviä hahmojen liikkeitä ja niiden ympäristöä, ilman että pelin koodiin tai sisäisiin toiminnallisuuksiin on pääsyä. Näin voidaan mallintaa hahmojen toiminnan logiikkaa ja ennustaa niiden käyttäytymistä analysoimalla pelkästään ulkoisia havaintoja. (Fink;ym., 2007)

Tutkimuksessa ehdotettu menetelmä koostuu kolmesta päävaiheesta:

1. **Objektin tunnistus ja seuranta:** Pelin tapahtumat tallennetaan näytön kuvakaappauksina, joista koneoppimismallit tunnistavat ja seuraavat pelin eri objekteja, kuten hahmoja ja liikkuvia esineitä. Tätä varten käytetään kuvankäsittelymenetelmiä, kuten liikkeentunnistusta.
2. **Tilanne-toimintaparit:** Pelin tapahtumat tiivistetään tilanne-toimintapareiksi, joissa yhdistetään tietyt visuaalisesti havaittavat tilanteet ja niitä seuraavat toiminnot. Tämä mahdollistaa hahmojen toiminnan ennustamisen havaittujen tilanteiden perusteella.
3. **Lähimmän naapurin menetelmä:** Ennustamiseen käytetään lähimmän naapurin algoritmia, joka valitsee tilanteeseen parhaiten sopivan toiminnan aikaisempien tilanteiden perusteella. Tämä tekee mallista joustavan ja sovellettavan erilaisiin pelitilanteisiin.

Menetelmän toimivuutta testattiin Pong-pelissä, jossa tietokoneen ohjaaman mailan käyttäytymistä pyrittiin ennustamaan pelin visuaalisen ulostulon perusteella. Vaikka tutkimuksessa ei hyödynnetty pelin sisäistä logiikkaa, kuten fysiikan mallinnusta, saavutettiin ennustuksissa noin 90 % tarkkuus tietokoneohjatun hahmon toiminnassa. Tämä osoittaa, että pelkästään visuaalisen datan pohjalta voidaan saavuttaa korkea ennustustarkkuus. (Fink;ym., 2007)

Koneoppimisen käyttö bottien kehityksessä tarjoaa merkittäviä etuja. Ensimmäinen menetelmä on autonominen, sillä se ei vaadi pääsyä pelin koodiin. Tämä mahdollistaa bottien kehittämisen esimerkiksi suljetuissa järjestelmissä tai peleissä, joiden lähdekoodi ei ole saatavilla. Lisäksi koneoppiminen tekee bottien mukautumisesta ja kehittämisestä joustavampaa, sillä mallit voidaan kouluttaa uudelleen eri peleihin ja skenaarioihin. Toisaalta menetelmän tarkkuutta rajoittavat tietyt tekijät, kuten piilotetut pelitilat ja tilanteet, joita ei havaita koulutusaineistossa. Näin ollen mallinnus

voi olla epätarkkaa erityisesti monimutkaisemmissa peleissä, joissa hahmojen käyttäytymiseen vaikuttavat monet tekijät. (Fink;ym., 2007)

Koneoppimisen avulla voidaan kehittää luonnollisesti käyttäytyviä botteja, jotka tarjoavat haastavampia ja realistisempia pelikokemuksia. Esimerkiksi tutkimuksessa esitelty visuaaliseen dataan perustuva lähestymistapa tarjoaa lupaavan perustan bottien kehitykselle. Jatkossa koneoppimismenetelmiä voidaan edelleen parantaa hyödyntämällä edistyneempiä algoritmeja, kuten syviä neuroverkkoja, ja sisällyttämällä niihin tietoa pelin sisäisistä säännöistä ja toimintalogiikoista. Tämä tutkimus korostaa, kuinka koneoppiminen tarjoaa tehokkaita työkaluja pelialan kehittäjille ja tutkijoille. Se ei ainoastaan mahdollista monimutkaisempien bottien luomista, vaan myös avaa uusia mahdollisuuksia pelien analysointiin ja muokkaamiseen. (Fink;ym., 2007)

Syväoppiminen (engl. *Deep Learning, DL*) on koneoppimisen osa-alue, joka hyödyntää monikerroksisia neuroverkkoja monimutkaisten kuvioiden ja piirteiden tunnistamiseen suurista tietomääristä. Sen sovellukset ulottuvat kuvantunnistuksesta ja luonnollisen kielen käsittelystä reaaliaikaiseen päätöksentekoon. Peliteollisuudessa syväoppimista voidaan hyödyntää esimerkiksi bottien mukautumisessa ihmispelaajien käyttäytymiseen ja strategioihin. Syväoppiminen mahdollistaa järjestelmien oppimisen ilman tarkkaa sääntöpohjaista ohjelmointia, mikä tekee siitä tehokkaan työkalun dynaamisten ympäristöjen, kuten urheilu- ja strategiapelien, tekoälykehityksessä. (Heaton, 2017)

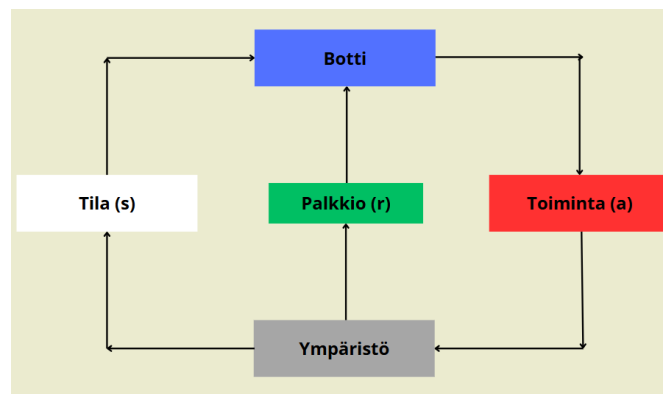
Keinotekoiset neuroverkot (engl. *Artificial Neural Networks, ANN*) ovat laskennallisia järjestelmiä, jotka koostuvat useista prosessoreista ja niiden välisistä yhteyksistä. Niiden toiminta perustuu ihmisen hermojärjestelmän inspiroimaan rakenteeseen, vaikka yhtäläisyydet ovatkin pääasiassa pinnallisia. ANN-järjestelmiä käytetään erityisesti insinööritieteissä, kuten kuvankäsittelyssä ja kaavan tunnistuksessa, sillä ne kykenevät käsittelemään suuria tietomääriä. Neuroverkot koostuvat keinotekoisista neuroneista, jotka sisältävät tietoa, ja näitä yhdistävistä reunoista, jotka välittävät datasiinaaleja. Jokaisella syötesolmulla on painoarvo, joka lasketaan matemaattisen funktion avulla, ja tämä määrittää, mikä neuroni aktivoituu. Verkkojen kerrokset voivat sisältää erilaisia solmuja, jotka muokkaavat syötettä eri tavoin, ja signaalit kulkevat syötekerroksesta lähtökerrokseen, myös piilokerrosten kautta. ANN-verkkoja käytetään pääasiassa koneoppimisessa, mutta niitä voidaan hyödyntää myös bottien päätöksenteossa. Esimerkiksi Madsen ja Adamatti käyttivät ANN-verkkoja sekä tilakoneita roolipelien bottien hyökkäyskäyttäytymisen mallintamiseen, jolloin koulutetun verkon avulla vartijahahmo pystyi päättämään, hyökkääkö vai pakeneeko se pelaajaa. (Uludağlı;ym., 2023)

K.G Jollyn kirjoittamassa artikkelissa (Jolly;ym., 2007) kehittivät he neuroverkkoihin perustuvan päätöksentekomenetelmän robottijalkapalloliigan boteille. Heidän kaksivaiheinen

menetelmänsä sisälsi neuroverkon painomatriisien laskemisen sekä päätösten tekemisen kentän sijainnin perusteella, kuten puolustus-, syöttö- tai hyökkäysalueilla. Tämä lähestymistapa osoittautui tehokkaaksi, sillä simulaatioissa se tuotti erinomaisia tuloksia 3 vastaan 3 robottijalkapallo-otteluissa. (Jolly;ym., 2007) Vaikka kyseiset ottelut olivat oikeiden robottien välisiä kamppailuita, on tämän jälkeen kyseisiä menetelmiä käytetty myös videopelien maailmassa.

2.5 Vahvistus- ja jäljitelmäoppiminen

Vahvistusoppiminen (eng. *Reinforcement Learning*, RL) on koneoppimisen lähestymistapa, joka perustuu käyttäytymispsykologiaan ja erityisesti siihen, miten ihmiset ja eläimet oppivat tekemään päätöksiä ympäristöstään saamien palkkioiden avulla niin negatiivisessa kuin positiivisessakin mielessä. Toisin kuin ohjatussa oppimisessa, jossa on valmiit esimerkit hyvistä toiminnoista, vahvistusoppimisessa botti oppii kokeilemalla ja saamansa palautteen avulla.



Kuva 1 Vahvistusoppimisen tilat

Muokattu lähteestä (Yannakakis;ym., 2018) s. 72

Tietyllä ajanhetkellä (t) botti on tietystä tilassa (s) ja valitsee jonkin toiminnon (a) kaikista käytettävissä olevista toiminnoista kyseisessä tilassa. Ympäristö antaa välittömän palkkion, (r), vastauksena botin toimintaan. Botti oppii jatkuvan vuorovaikutuksen kautta valitsemaan toimintoja, jotka maksimoivat sen saamien palkkioiden summan. Vahvistusoppimista on tutkittu eri tieteenoilta, kuten operaatioanalyysin, peliteorian, informaatioteorian ja geneettisten algoritmien näkökulmista, ja sitä on sovellettu menestyksekkäästi ongelmissa, jotka vaativat tasapainottelua pitkän ja lyhyen aikavälin palkkioiden välillä, kuten bottien ohjauksessa ja peleissä. Vahvistusongelma voidaan konkretisoida esimerkiksi labyrinttinavigointitehtävän kautta. Formaalisti botin tavoitteena on löytää strategia eli tietty politiikka toiminnon valitsemiseksi siten, että pitkän aikavälin palkkio, kuten

odotettu kumulatiivinen palkkio, maksimoituu. Politiikka on strategia, jota botti seuraa toimintojen valinnassa nykyhetken tilassaan. Jos jokaisen toiminnon arvo joko on tiedossa tai opittu, optimaalinen politiikka voidaan johtaa valitsemalla toiminto, jolla on korkein arvo. (Yannakakis;ym., 2018)

Botin ja ympäristön vuorovaikutus tapahtuu diskreeteissä ajanhetkissä ($t = 0, 1, 2, \dots$)

Mallinnetaan Markovin päätösprosessia, MDP (eng. *Markov decision process*):

- S: Tilajoukko $\{s_1, \dots, s_n\} \in S$. Tilajoukko määrittää ympäristön tilat, jotka perustuvat botin saamiin tietoihin ympäristöstä.
- A: Toimintojoukko $\{a_1, \dots, a_m\} \in A$, joka kuvaa jokaisessa tilassa (s) mahdolliset toiminnot.
- P: Siirtymän todennäköisyys (s, s', a) , joka kuvaa siirtymän todennäköisyyttä tilasta (s) tilaan (s') tietyn toiminnon (a) perusteella.
 - Tämä malli seuraa MDP: n ominaisuutta, jossa prosessin tulevat tilat riippuvat vain nykyisestä tilasta ja ovat riippumattomia sitä edeltävästä tapahtumasarjasta.
 - Tämän seurauksena siirtymän todennäköisyys P mahdollistaa 1-vaiheisen dynamiikan ennustamisen.
- R: Palkkiofunktio (s, s', a) , siirryttäessä tilasta (s) tilaan (s') annetulla toiminnolla (a).
 - Kun tilassa (s) oleva botti valitsee toiminnon (a) ja siirtyy tilaan (s'), saa se välittömästi palkinnon (r) ympäristöstä.

Siirtymän todennäköisyys (P) ja palkkiofunktio (R) määrittävät ympäristön dynamiikan sekä pitkän aikavälin palkkion jokaiselle politiikalle. Jos maailmamalli (eng. *World Model*) eli ympäristön säännöt tunnetaan, siirtymätodennäköisyyksiä ja palkkiofunktioita ei tarvitse oppia, vaan optimaalinen strategia voidaan laskea suoraan mallipohjaisiin lähestymistavoin, kuten dynaamisen ohjelmoinnin avulla. Jos taas maailmamalli on tuntematon, siirtymä- ja palkkiofunktioita arvioidaan oppimalla tulevia palkkioita tietyn toiminnon perusteella. Näiden arvioiden perusteella lasketaan politiikka. Oppiminen tapahtuu mallittomien menetelmien, kuten Monte Carlo -haun ja ajallisen eron oppimisen, TDL (eng. *Temporal Difference Learning*) avulla. (Yannakakis;ym., 2018)

Jäljitelmäoppiminen (engl. *Imitation Learning, IL*) on koneoppimisen lähestymistapa, jossa botti oppii mallintamaan ihmisen toimintaa käyttämällä esimerkkietoa ihmisen tekemistä päätöksistä ja toimista. Toisin kuin vahvistusoppimisessa, jossa oppiminen perustuu ympäristön palautteeseen, jäljitelmäoppimisessa botti pyrkii suoraan jäljittelemään asiantuntijan suorittamia toimintoja. Tämä tekee jäljitelmäoppimisesta erityisen hyödyllisen, kun tavoitteena on luoda botteja, joiden käyttäytyminen muistuttaa inhimillistä toimintaa. (Lauro;ym., 2024)

Artikkelissa (Lauro;ym., 2024) esitellään menetelmä, jossa käytetään jäljitelmäoppimista ja pelaajien pelikokemuksista kerättyä tietoa (provenienssidataa) ihmismäisten bottien kouluttamiseen.

Provenienssilla tarkoitetaan tiedon alkuperää ja sen siirtymistä, eli sitä, miten ja mistä tietty tieto on peräisin, sekä millaisia syy-seuraussuhteita siihen liittyy. Tässä tutkimuksessa provenienssidata sisältää yksityiskohtaisia tietoja pelaajien toiminnasta, kuten heidän tekemistään päätöksistä ja niiden vaikutuksista pelimaailmaan. Provenienssidatan avulla on mahdollista kartoittaa pelaajien päätöksentekoprosesseja ja toistuvia käyttäytymismalleja. Tutkimuksessa hyödynnettiin GAIL-kehystä (engl. *Generative Adversarial Imitation Learning*) koulutusprosessin keskiössä, mikä mahdollisti näiden tietojen hyödyntämisen ihmismäisten bottien käyttäytymisen mallintamiseen. Jäljitelmäoppimisen tärkein tavoite on mahdollistaa bottien käyttäytymisen oppiminen ilman tarvetta erikseen käsin ohjelmoida jokaisen tilanteen toimintalogiikkaa. Tämä lähestymistapa sopii erityisen hyvin peliteollisuuteen, jossa boteilta odotetaan yhä monimutkaisempaa ja ihmismäisempää käyttäytymistä. (Lauro;ym., 2024)

Provenienssidatan käyttö jäljitelmäoppimisessa eroaa perinteisestä pelilokeihin perustuvasta datasta, sillä se sisältää syy-seuraussuhteita pelaajien toiminnasta ja niiden vaikutuksista pelimaailmaan. Tämä tieto tallennetaan graafimuotoon, jossa solmut edustavat pelaajien toimintoja, ja kaaret kuvaavat toimien välisiä suhteita. Tämän rikkaan tietorakenteen ansiosta pelikehittäjät voivat tunnistaa tärkeitä pelielementtejä, kuten strategioita ja päätöksentekomalleja, joita voidaan hyödyntää bottien koulutuksessa. Koulutusprosessissa pelien pelaajakohtaiset sessiot toistetaan aiemmin kerätyn provenienssidatan pohjalta, minkä jälkeen GAIL-kehystä käytetään hahmojen toimintapolitiikan oppimiseen. GAIL mahdollistaa bottien oppimisen ilman suoraa vuorovaikutusta asiantuntijan kanssa ja auttaa botteja navigoimaan myös tuntemattomissa tilanteissa, joita ei ole havaittu koulutusaineistossa. (Lauro;ym., 2024)

Tutkimuksessa kehitetty malli testattiin Dodge Ball-pelissä, jossa pelaajat kohtasivat ihmismäisiä botteja, jotka oli koulutettu provenienssidatan avulla. Tulokset osoittivat, että jäljitelmäoppimisen avulla koulutetut botit koettiin pelaajien toimesta uskottavammiksi kuin perinteisillä palkkiojärjestelmillä koulutetut botit. Tämä johtuu siitä, että provenienssidatan avulla koulutetut botit tekevät myös virheitä, kuten ihmispelaajat, mikä lisää niiden autenttisuuden vaikutelmaa. Jäljitelmäoppiminen tarjoaa lupaavan vaihtoehdon vahvistusoppimiselle erityisesti silloin, kun tavoitteena on luoda botteja, joiden käyttäytyminen muistuttaa pelaajien toimintaa ja parantaa pelin immersiivisyyttä. Provenienssidatan hyödyntäminen avaa uusia mahdollisuuksia hahmojen koulutukseen ja tarjoaa tarkemman näkymän pelaajien toiminnasta kuin perinteiset menetelmät. (Lauro;ym., 2024)

2.6 Hybridimenetelmät

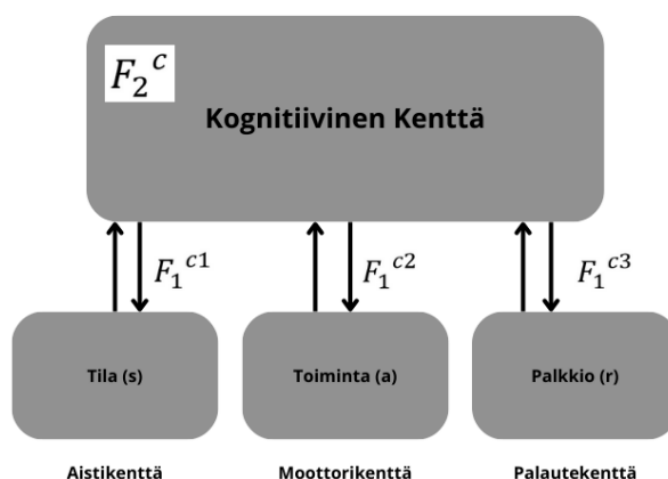
Q-oppiminen on mallivapaa, off-policy-pohjainen vahvistusoppimisen algoritmi, joka perustuu taulukkomuotoiseen Q-arvojen esitykseen. Q-oppimisessa off-policy tarkoittaa, että oppiminen tapahtuu riippumatta siitä, miten toimintaa suoritetaan. Se voi perustua esimerkiksi satunnaisiin kokeiluihin tai johonkin muuhun politiikkaan kuin siihen, jota botti parhaillaan oppii. Tämä erottaa sen on-policy-menetelmistä, joissa oppiminen perustuu ainoastaan niihin toimintoihin, joita botin nykyinen politiikka määrittää. Näillä Q-arvoilla kuvataan, kuinka hyvä tietyn toiminnon a (action) suorittaminen tietyssä tilassa s (state) on. $Q(s+a)$ -arvo edustaa tilan ja toiminnon yhdistelmälle odotettavissa olevaa diskontattua palkkiota r (reward). Q-oppimis-botin tavoitteena on oppia maksimoimaan pitkän aikavälin palkkiot valitsemalla optimaaliset toiminnot jokaisessa tilassa. Oppiminen tapahtuu iteratiivisesti (vaiheittain) päivittämällä Q-aulukon arvoja kokemuksen perusteella: botti valitsee toiminnon, siirtyy uuteen tilaan, vastaanottaa palkkion ja päivittää Q-arvoa niin, että se huomioi uuden palkkion sekä arvioidun tulevien palkkioiden odotusarvon. Tämä tekee Q-oppimisesta tehokkaan algoritmin erilaisten päätöksentekoprosessien oppimiseen, kuten pelistrategioiden kehittämiseen, bottien liikkeenhallintaan tai resurssien optimointiin. Kuitenkin algoritmin käyttöä rajoittaa Q-aulukon kasvu suurissa tila-avaruuksissa, mikä voi vaatia huomattavaa laskentatehoa ja muistia. Näiden rajoitteiden kiertämiseksi voidaan hyödyntää esimerkiksi tekoälypohjaisia neuroverkkoja Q-arvojen approksimointiin. (Yannakakis;ym., 2018)

Syvä Q-oppiminen (engl. *Deep Q-Learning*, DQN) on Q-oppimisalgoritmin laajennus, joka hyödyntää neuroverkkoja Q-funktion approksimointiin. Tämä mahdollistaa tehokkaan päätöksenteon myös suurissa tai monimutkaisissa tilaympäristöissä, kuten kuvapohjaisissa syötteissä. DQN:n keskeisiä ominaisuuksia ovat kokemuksen uudelleentoisto (engl. *experience replay*), joka estää paikallisiin optimeihin jumittumisen, sekä näyttöiden tehokas käyttö koulutuksessa. Haasteina ovat laskennallisesti raskaat vaatimukset ja niin kutsuttu ulottuvuuksien kirous, jossa datan hajonta kasvaa tilaympäristön monimutkaistuessa. DQN on merkittävä parannus verrattuna perinteiseen Q-oppimiseen, erityisesti suurissa ja monimutkaisissa tilaympäristöissä. Perinteinen Q-oppiminen tallentaa Q-arvot taulukkomuodossa, mikä ei ole käytännöllistä, kun tilojen määrä kasvaa eksponentiaalisesti. DQN käyttää neuroverkkoja Q-arvojen approksimointiin, mikä tekee siitä tehokkaamman suurissa tiloissa, kuten kuvapohjaisessa datassa. Näiden ominaisuuksien ansiosta DQN soveltuu monimutkaisiin ympäristöihin, kuten bottien kouluttamiseen, joissa pelkkä Q-oppiminen ei olisi riittävän tehokasta. (Almeida;ym., 2023)

Autonomisten tietojärjestelmien, kuten bottien kehittämisessä käytetään yhä monimutkaisempia oppimismenetelmiä. Näihin kuuluvat erityisesti kaksivaiheinen oppiminen (engl. *Dual-Stage Learning*, DSL) ja sekamallioppiminen (engl. *Mixed Model Learning*, MML), jotka perustuvat

FALCON-nimiseen itseorganisoituvaan neuroverkkomalliin (engl. *Fusion Architecture for Learning and Cognition*). FALCON mahdollistaa useiden oppimisparadigmojen integroinnin yhteen kehykseen, mikä tekee siitä tehokkaan työkalun hybridioppimisen toteutuksessa. (Shu;ym., 2016)

FALCON perustuu ART-neuroverkkoihin (engl. *Adaptive Resonance Theory*) ja yhdistää tilojen, toimintojen ja palkkioiden käsittelyn samanaikaisesti. FALCON-menetelmän keskeinen ominaisuus on sen kyky tukea useita oppimismenetelmiä, kuten ohjattua oppimista, ohjaamatonta oppimista ja vahvistusoppimista. Tämä tapahtuu yhdistämällä kilpailullinen koodaus ja mukautuva verkon laajentaminen. FALCON oppii kognitiivisia solmuja, jotka sisältävät monimutkaisia monimuuttuvaisia riippuvuuksia tilojen, toimintojen ja palkkioiden välillä. FALCON toimii kolmikanavaisella arkkitehtuurilla, jossa kukin kanava vastaa tietyistä syöte- tai palauteparametreista. Tämä mahdollistaa tiedon yhdistämisen eri lähteistä yhteen yhtenäiseen rakenteeseen. Näin saadaan aikaan tehokas alusta, joka tukee sekä jäljittely- että vahvistusoppimista ja mahdollistaa näiden paradigmojen integraation. (Shu;ym., 2016)



Kuva 2 FALCON menetelmän kolmikanavainen arkkitehtuuri

Kuva muokattu lähteestä: (Shu;ym., 2016)

Kuvassa esitetään FALCON:in kolmikanavainen arkkitehtuuri, joka muodostuu kognitiivisesta kentästä ja kolmesta syöttökentästä. Näihin syöttökenttiin kuuluvat aistikenttä, joka vastaa nykyisten tilojen mallintamisesta, moottorikenttä, joka kuvaa toimintoja, sekä palautekenttä, joka edustaa palkkioarvoja. (Shu;ym., 2016)

DSL yhdistää jäljittely- ja vahvistusoppimisen kahdessa vaiheessa. Ensimmäisessä vaiheessa jäljittelyoppimista käytetään luomaan botin alkuperäinen tietämys, joka perustuu ympäristön tiloihin ja niihin liittyviin toimiin. Tämä vaihe antaa botille kyvyn toimia heti ilman pitkää tutkimisvaihetta,

joka on tyypillistä vahvistusoppimiselle. Toisessa vaiheessa vahvistusoppiminen tarkentaa ja laajentaa tätä tietämystä reaaliaikaisen palautteen avulla. DSL nopeuttaa oppimista ja mahdollistaa botin paremman suorituskyvyn dynaamisissa ympäristöissä, kuten Unreal Tournament -pelissä. (Shu;ym., 2016)

MML-menetelmä yhdistää jäljittely- ja vahvistusoppimisen saumattomasti samaan tietorakenteeseen. Toisin kuin DSL, MML ei erottele oppimisvaiheita, vaan molemmat menetelmät toimivat vuorotellen ja jakavat yhteisen tietopohjan. Esimerkiksi negatiivisen palautteen yhteydessä botti voi jäljitellä vastustajan toimintaa ja integroida sen omaan käyttäytymismalliinsa. Tämä strateginen oppimismekanismi mahdollistaa nopean sopeutumisen muuttuvissa olosuhteissa ja parantaa oppimisen tehokkuutta. (Shu;ym., 2016)

1. Nopeampi oppiminen: FALCON-pohjaiset DSL- ja MML-menetelmät tarjoavat nopeamman oppimisnopeuden verrattuna perinteisiin RL- tai IL-menetelmiin.
2. Parempi suorituskyky: Kokeet osoittavat, että DSL- ja MML-menetelmillä luodut botit suoriutuvat paremmin taistelupeleissä, kuten Unreal Tournamentissa, verrattuna yksinomaan RL- tai IL-menetelmien käyttämiin hahmoihin.
3. Sopeutumiskyky: MML-menetelmä mahdollistaa botin reaaliaikaisen sopeutumisen vastustajiin ja ympäristöön, mikä tekee siitä tehokkaan erityisesti dynaamisissa ja kilpailullisissa ympäristöissä.
4. Tiedon yhdistäminen: FALCON-mallin avulla erilaiset oppimisparadigmat voidaan yhdistää yhdeksi yhtenäiseksi järjestelmäksi, mikä tehostaa oppimisen tuloksia ja vähentää päällekkäisyyksiä.

DSL, MML ja FALCON muodostavat tehokkaan yhdistelmän autonomisten järjestelmien, kuten bottien, kehittämisessä. FALCON:in tarjoama joustava tietorakenne mahdollistaa oppimisen, jossa hyödynnetään sekä jäljitelmä- että vahvistusoppimisen vahvuuksia. DSL:n ja MML:n avulla voidaan saavuttaa nopeampi oppiminen, parempi suorituskyky ja parempi sopeutumiskyky monimutkaisissa ympäristöissä. Tämä tekee näistä menetelmistä lupaavia paitsi pelialalla, myös muissa autonomisten järjestelmien sovelluksissa, kuten robotiikassa ja simuloinneissa. (Shu;ym., 2016)

3 Kirjallisuudessa esiintyneiden tekoölymenetelmien soveltuvuus NHL 25:n kaltaisiin peleihin

NHL-videopeli on EA Sportsin kehittämä jääkiekkopelisarja, joka on ollut osa urheilupelien historiaa vuodesta 1991 lähtien. Ensimmäinen peli, NHL Hockey, julkaistiin Sega Genesis-konsolille, ja se saavutti nopeasti suosiota realistisen pelattavuutensa sekä innovatiivisten ominaisuuksiensa, kuten pelaajien yksilöllisten taitojen, ansiosta. Sarja on sittemmin kehittynyt vuosittain julkaistavaksi pelisarjaksi, joka simuloi NHL:n (engl. *National Hockey League*) ammattilaisjääkiekkoa tarkasti ja tarjoaa pelimahdollisuuksia niin yksin kuin moninpelitilassa. (Stephen, 2012)



Kuva 3 Näyttökuvasta pelistä NHL 91

Kuva lainattu lähteestä (GameFabrique)

Nykypäivän NHL-pelit, kuten uusien NHL 25, hyödyntävät edistyneitä tekoölymenetelmiä, jotka parantavat pelimekaniikoiden realistisuutta ja bottiin kykyä mukautua pelitilanteisiin. Generatiivinen tekoöly (GT) mahdollistaa jouhevan ja realistisen pelikokemuksen, jossa botit eivät toimi vain yksinkertaisina vastustajina, vaan ne osaavat reagoida hyökkäys- ja puolustustaktiikoihin pelaajien toiminnan mukaan (The Hockey News, 2024) Generatiivisen tekoölyn rooli on kasvanut huomattavasti ja nykyiset järjestelmät hyödyntävät koneoppimista ja dynaamista päätöksentekoa, mikä tekee bottiin käyttäytymisestä ihmismäisempää ja pelistä autenttisempää.

EA:n luovan johtajan Mike Inglehartin (The Hockey News, 2024) mukaan GT:n käytön aiempi haaste oli epäjohtonmukaisuus, kun GT yritti suorittaa liikaa asioita yhtä aikaa. Vuoden 2025 kehitystyössä painopiste siirtyi GT:n toiminnan selkeyttämiseen ja sen johtonmukaisuuden parantamiseen, mikä näkyy etenkin sen kyvyssä tukea hyökkäyksiä, luoda syöttöjä ja hallita kiekkoa

hyökkäysalueella tehokkaammin. Reaktiiviset toiminnot, kuten pelaajien välisten törmäysten vähentäminen kriittisillä alueilla, osoittavat GT:n kyvyn ymmärtää pelitilanteita entistä paremmin. Pelinopeuden hidastaminen sekä parannukset syöttöjen ajoituksessa ja One-Timer-vedoissa tekevät maalinteosta dynaamisempaa ja hallitumpaa kuin aiemmissa peliversioissa. (The Hockey News, 2024)

NHL 25 tuo mukanaan myös uusia järjestelmiä, kuten ICE-Q:n ja Empowered AI:n, jotka edustavat merkittäviä askeleita GT:n käytössä. (Electronic Arts, 2024) ICE-Q parantaa pelaajien päätöksentekoa ja reagoitukykyä, todennäköisesti hyödyntäen vahvistusoppimista, jossa botteja koulutetaan toistuvien pelisimulaatioiden kautta. Empowered Ai puolestaan vaikuttaa hyödyntävän syväoppimista ja neuroverkkoja, joiden avulla botit osaa ennustaa vastustajien sijoittumista ja mukauttaa omaa sijoittumistaan reaaliajassa. Törmäysten välttely, suunnanmuutokset ja realistiset liikkeet viittaavat liikeoptimointialgoritmien käyttöön, ja ominaisuudet kuten Next Gen Vision Control ja Skill-Based One-Timers voivat perustua simulaatiopohjaisiin fysiikallisiin malleihin, joissa pelaajien liikkeet mukautetaan todentuntuiseksi. Koneäkö ja sensorien tiedon analysointi tukevat luonnollisten liikkeiden hallintaa reaaliaikaisessa ympäristössä.

Mahdollisista tekoälymenetelmistä sääntöpohjaiset lähestymistavat (if-then-else) sopivat hyvin yksinkertaisiin tehtäviin, kuten kiekon liikkeiden ennustamiseen tai perustason puolustukseen. FSM puolestaan voi ohjata pelaajien siirtymistä eri tilojen välillä, kuten hyökkäyksestä puolustukseen. Vaikka FSM tarjoaa selkeän rakenteen, se kärsii joustamattomuudesta ja monimutkaistuu tilojen lisääntyessä. (Jagdale, 2021) Vahvistusoppiminen mahdollistaa NHL:n botteille kyvyn reagoida ja oppia optimaalisia toimintamalleja dynaamisissa pelitilanteissa, kuten hyökkäyksissä ja puolustuksissa, kun taas syväoppimista voidaan käyttää visuaalisessa tunnistuksessa ja ihmispelaajien pelityylin jäljittelyssä. (Aydin;ym., 2023) Hybridimenetelmät yhdistävät sääntöpohjaisen logiikan ja oppimismallit, mahdollistaen strategisen ja reaaliaikaisen päätöksenteon monimutkaisissa pelitilanteissa. (Milani;ym., 2023)

NHL 25:n tekoälyjärjestelmissä voidaan olettaa käytettävän myös päätös- ja käyttäytymispuita. Päätöspuut tarjoavat tehokkaan tavan mallintaa ja jäsentää bottien käyttäytymistä loogisesti erilaisissa päätöksissä, kuten esimerkiksi pitäisikö pelaajan syöttää, laukoa vai hakea parempi paikka. Näin ne voivat toimia tukirakenteina monimutkaisemmille koneoppimisen ratkaisuille, kuten vahvistusoppimiselle. Käyttäytymispuut puolestaan mahdollistavat hierarkkisen ja modulaarisen päätöksenteon, jossa bottien toiminta voidaan jakaa pienempiin, tarkasti määriteltyihin osiin (Matteo;ym., 2022). Esimerkiksi hyökkäystilanteessa botit voi ensin analysoida kiekon hallintaa, sitten päätellä pelaajan sijoittumisen ja lopuksi valita toiminnon, kuten syötön tai laukauksen. Tällaiset

rakenteet tekevät botin käyttäytymisestä helpommin hallittavaa ja ennustettavaa, erityisesti nopeissa ja monivaiheisissa pelitilanteissa.

Lisäksi FALCON-pohjaiset hybridimenetelmät, kuten DSL (kaksivaiheinen oppiminen) ja MML (sekamallioppiminen), tarjoavat lupaavia mahdollisuuksia NHL 25 -tyyppisten pelien bottien kehittämisessä (Shu;ym., 2016). DSL voisi nopeuttaa botin alkuosaamisen muodostamista jäljittelyoppimisen avulla, jolloin botti voi heti pelin alussa hyödyntää ihmisvastustajien pelityylejä. Tämän jälkeen vahvistusoppiminen voi hienosäätää toimintaa reaaliaikaisesti pelitilanteiden mukaan. MML-menetelmä puolestaan mahdollistaa jatkuvasti mukautuvan botin, joka oppii sekä palautteesta että toisten pelaajien strategioista. Esimerkiksi NHL 25 -pelissä botti voisi havaita, miten vastustaja pelaa ylivoimaa ja käyttää tätä tietoa optimoidakseen oman puolustuksensa. FALCON-arkkitehtuuri mahdollistaa tämän, koska se yhdistää aistihavainnot, toimintavaihtoehdot ja saadut palautteet yhdeksi tietorakenteeksi. Näin ollen se soveltuisi hyvin ympäristöihin, joissa peli on jatkuvaa, tilat vaihtuvat nopeasti ja ratkaisut pitää tehdä reaaliaikaisesti. Tällainen lähestymistapa parantaisi botin suorituskykyä ja mukautumista merkittävästi.

Kaiken kaikkiaan NHL 25:n ominaisuudet viittaavat siihen, että pelikehittäjät ovat yhdistäneet useita tekoälymenetelmiä, kuten vahvistusoppimista, syväoppimista ja fysiikkapohjaisia simulaatioita, tuodakseen jääkiekkokokemuksen uudelle tasolle. Nämä menetelmät luovat perustan dynaamiselle ja realistiselle pelitapahtumalle, joka reagoi pelaajien valintoihin entistä älykkäämmin.



Kuva 4 Näyttökuvasta pelistä NHL 25

Kuva lainattu lähteestä (Kinguin, 2025)

Pelimuodoilla on merkittävä vaikutus siihen, mitä tekoälymenetelmiä käytetään. Yksinpelissä bottien tulisi mukautua pelaajan pelityyliin ja taitotasoon, jolloin vahvistusoppiminen ja hybridimenetelmät ovat erityisen tehokkaita. Moninpelissä bottien on pystyttävä jäljittelemään monien ihmispelaajien pelityyliä esimerkiksi jäljitelmäoppimisen kautta, mikä parantaa immersiota ja kilpailullisuutta. Harjoittelutilassa taas voidaan käyttää sääntöpohjaisia ja vahvistusoppimisen menetelmiä, jotka auttavat pelaajaa oppimaan peliä ja tarjoavat progressiivisesti vaikeutuvia haasteita. Generatiivisen tekoälyn (GT) roolia on laajennettu merkittävästi myös pelin Franchise-tilassa, jossa pelaaja toimii joukkueen toimitusjohtajana (engl. *General Manager*, GM). Tässä pelimuodossa GT mahdollistaa sopimuslausekkeiden käsittelyn, pelaajien välisten keskusteluiden simuloinnin sekä pitkän aikavälin tulevaisuuden ennustamisen realistisesti. Esimerkiksi sen arvioimisen, kuka pelaaja saattaa kehittyä NHL:n tähtiluokkaan tai kuka tekee 60 maalia kaudessa. Lisäksi GT mallintaa joukkueiden strategioita ja kehityskulkuja hyödyntäen neuroverkkoja ja hybridimenetelmiä, mikä tekee simulaatioista entistä kiehtovampia ja realistisempia. GT hallinnoi myös pelaajasiirtoja, kokoonpanoja ja joukkuekohtaisia taktiikoita, mikä tukee pelaajan valinnanvapautta ja syventää pelin immersivisyyttä. Franchise-tilan kannalta GT:n johdonmukaisuus ja kyky keskittyä olennaiseen ovat keskeisiä tekijöitä pelin laadun ja pelaajakokemuksen parantamisessa. (The Hockey News, 2024)

NHL-pelin yksi tärkeimmistä ominaisuuksista pelaajakokemuksen muokkaamisessa on niin sanottujen "sliderien" eli liikusäätimien käyttö, joiden avulla pelaajat voivat säätää bottien vaikeustasoa ja pelin dynamiikkaa yksityiskohtaisesti. Slider-asetukset tarjoavat mahdollisuuden hienosäätää pelin eri elementtejä, kuten laukausten kovuutta, luistelunopeutta, syöttöjen tarkkuutta ja voimaa sekä maalivahtien reagoitokykyä eri tilanteissa. Lisäksi pelaajat voivat vaikuttaa jäähyjen herkkyyteen, esimerkiksi kuinka herkästi pelin tuomari tuomitsee jäähyjä kampituksista, estämisistä tai korkeista mailoista. Tämän kaltaiset säädöt mahdollistavat pelin mukauttamisen pelaajan omien taitojen ja mieltymysten mukaan, jolloin peli voi olla sekä realistinen simulaatio oikeasta jääkiekkopelistä että nopeampi arcade-tyylinen kokemus. Sliderien käyttö tuo lisää hallintaa bottien käyttäytymiseen eri pelimuodoissa ja täydentää NHL 25:n tekoälyjärjestelmien, kuten Empowered AI:n ja ICE-Q:n, mahdollistamaa dynaamista reagointia ja realismia.

NHL 25 hyödyntää laajaa valikoimaa tekoälymenetelmiä, joiden avulla peli tarjoaa realistisen, mukautuvan ja immersivisen pelikokemuksen. Botit on suunniteltu reagoimaan älykkäästi pelaajan toimintaan ja kehittymään pelin edetessä, mikä tekee pelistä houkuttelevan niin aloittelijoille kuin kokeneille pelaajille.

4 Yhteenveto

Tässä työssä on käsitelty bottien kehitystä ja niiden strategista käyttäytymistä eri videopeligenreissä. Ensimmäinen tutkimuskysymys keskittyi akateemisessa kirjallisuudessa esitettyihin menetelmiin bottien luomiseksi. Tarkastelussa havaittiin, että keskeisimpiä lähestymistapoja ovat muun muassa tilakoneet, käyttäytymispuut, päätöspuut, neuroverkot sekä vahvistus- ja jäljitelmäoppiminen. Tilakoneet ja käyttäytymispuut tarjoavat sääntöpohjaisia ja modulaarisia ratkaisuja, kun taas neuroverkkoihin ja vahvistusoppimiseen perustuvat menetelmät mahdollistavat adaptiivisemmat ja oppivat botit. Esimerkiksi FPS-peleissä bottien oppiminen voi perustua pelaajien liikkeiden ja toiminnan analysointiin, jolloin botit voivat mukautua pelaajien strategioihin ja tarjota realistisempia haasteita.

Toinen tutkimuskysymys tarkasteli tekoälyn vaikutusta bottien strategiseen käyttäytymiseen eri videopeligenreissä. Tulokset osoittavat, että tekoälymenetelmillä on keskeinen rooli erityisesti peleissä, joissa vuorovaikutus pelaajien ja bottien välillä määrittää pelikokemuksen laadun. Reaaliaikaisessa strategiapelissä käytetään usein suunnittelualgoritmeja, kuten Minimaxia ja Monte Carlo Tree Searchia, jotka auttavat botteja ennakoimaan ja reagoimaan pelitilanteisiin tehokkaasti. Toisaalta ammuskelupeleissä korostuvat reaktiiviset toiminnot ja reitinhaku, joiden avulla hahmot liikkuvat ja toimivat dynaamisissa ympäristöissä. Molemmissa genreissä hyödynnetään myös vahvistusoppimista, jonka avulla botit pystyvät mukauttamaan toimintaansa pelaajien strategioiden ja muuttuvien pelitilanteiden mukaan.

Työ perustuu pääosin kirjallisuuskatsaukseen, eikä siihen sisälly empiirisiä kokeita bottien suorituskyvyn arvioimiseksi käytännön pelikehitysympäristössä. Jatkotutkimuksessa olisi hyödyllistä testata eri tekoälymenetelmiä käytännössä ja vertailla niiden vaikutuksia pelikokemukseen eri genreissä. Erityisesti suurten kielimallien ja generatiivisen tekoälyn yhdistäminen bottien kehitykseen tarjoaa lupaavan tutkimusalueen. Yhteenvetona voidaan todeta, että tämä työ tarjoaa kattavan pohjan bottien tekoälymenetelmien ymmärtämiselle ja soveltamiselle videopeleissä. Tulevaisuudessa teknologian kehittyessä bottien älykkyys ja vuorovaikutusmahdollisuudet voivat edelleen monipuolistua, mikä avaa uusia mahdollisuuksia pelisuunnittelulle ja pelikokemuksen rikastamiselle.

Tulevaisuuden tutkimus voisi keskittyä empiirisiin kokeisiin, joissa eri tekoälymenetelmiä sovelletaan käytännössä ja niiden vaikutuksia analysoidaan pelinkehittäjien ja pelaajien näkökulmasta. Esimerkiksi HFSM:n ja neuroverkkojen suorituskykyä voitaisiin arvioida eri peligenreissä ja eri tasoilla pelaajilla. Lisäksi pelaajien kokemuksia bottien toiminnasta tulisi tutkia tarkemmin. Verkkomonipelit, joissa bottien rooli on merkittävä, tarjoavat erityisen kiinnostavan kontekstin. Pelaajien odotukset ja kokemukset vaikuttavat suoraan siitä, miten onnistuneina botit nähdään. Generatiivisen tekoälyn, kuten tekstin ja kuvien luomiseen erikoistuneiden mallien,

yhdistäminen bottien käyttäytymiseen voisi rikastuttaa pelikokemusta entisestään. Työn tulokset luovat vankan perustan jatkotutkimukselle, joka voi syventää ymmärrystä tekoälymenetelmien ja bottien käyttömahdollisuuksista pelialalla.

Lähdeluettelo

1. **A survey of Behavior Trees in robotics and AI** [Aikakausjulkaisu] / tekijä Matteo Iovino [ym.] // Robotics and Autonomous Systems. - [s.l.] : ScienceDirect, 2022.
2. **AI and Games** [Online] / tekijä Thompson Tommy // How AI is Actually Used in the Video Games Industry. - 28. 2 2024. - <https://www.aiandgames.com/p/how-ai-is-actually-used-in-the-video>.
3. **AI Bots in Online Games** [Lehtiartikkeli] / tekijä Dimov Vitorio // Härting. - 21. 8 2023. - s. 1.
4. **Artificial Intelligence (AI)-Based Self-Deciding Character Development Application in Two-Dimensional Video Games.** / tekijä Aydin Hakan [ym.]. - Istanbul : Bilgi ve İletişim Teknolojileri Dergisi, 28. 6 2023. - ss. 1-19.
5. **Artificial Intelligence and Games** [Kirjan osa] / tekijä Yannakakis Georgios N. ja Togelius Julian // - / kirjan tekijä - - - : Springer Cham, 2018.
6. **Behavior Trees in Robotics and AI** [Kirjan osa] / tekijä Colledanchise Michele ja Ögren Petter. - 2018.
7. **Deep learning** [Raportti] / tekijä Heaton Jeff. - 2017.
8. **EA SPORTS NHL 25 is out now** [Online] / tekijä The Hockey News. - 4. 10 2024. - <https://thehockeynews.com/gaming/latest-news/ea-sports-nhl-25-out-now>.
9. **Extracting NPC behavior from computer games using computer vision and machine learning techniques** [Aikakausjulkaisu] / tekijä Fink Alex, Denzinger Jorg ja Aycock John // IEEE Symposium on Computational Intelligence and Games. - 2007. - ss. 24-31.
10. **Finite State Machine in Game Development** [Aikakausjulkaisu] / tekijä Jagdale Devang // IJARST. - 10 2021. - ss. 384-390.
11. **Intelligent decision making in multi-agent robot soccer system through compounded artificial neural networks** [Aikakausjulkaisu] / tekijä Jolly K.G [ym.] // Science Direct. - 2007. - ss. 589-596.
12. **Kuvankaappaus NHL 25 pelistä** [Kuvat] / artisti Kinguin. - [s.l.] : Kinguin.
13. **Large Language Model based Multi-Agents: A Survey of Progress and Challenges** [Raportti] / tekijä Guo Taicheng [ym.]. - 2024.
14. **Navigates Like Me: Understanding How People Evaluate Human-Like AI in Video Games** [Kokouspöytäkirja] / tekijä Milani Stephanie [ym.]. - Hamburg : ACM Digital Library, 2023. - ss. 1-3.
15. **NHL 25 Gameplay Deep Dive** [Aikakausjulkaisu] / tekijä Electronic Arts // Electronic Arts. - 5. 9 2024. - s. 1.

16. **Non-player character decision-making in computer games** [Aikakausjulkaisu] / tekijä Uludağlı Muhtar, Çağkan ja Oğuz Kaya // Artificial Intelligence Review. - 1. 12 2023. - ss. 14159-14191.
17. **Reinforcement Learning Applied to AI Bots in First-Person Shooters: A Systematic Review** [Aikakausjulkaisu] / tekijä Almeida Pedro, Carvalho Vitor ja Simões Alberto // Algorithms. - 2023.
18. **Screenshot from NHL 91** [Online] / tekijä GameFabrique. - <https://gamefabrique.com/games/nhl-hockey-91/>.
19. **Techopedia** [Online] / tekijä Rouse Margaret // Non-Player Character (NPC). - 26. 4 2024. - <https://www.techopedia.com/definition/1920/non-player-character-npc>.
20. **The interplay between immersion and appeal in video games** [Kirjan osa] / tekijä Georgios Christou // Computers in Human Behavior / kirjan tekijä Christou Georgios. - 2014.
21. **thehockeywriters.com** [Online] / tekijä Stephen Matt. - 31. 8 2012. - <https://thehockeywriters.com/the-evolution-of-the-hockey-video-game/>.
22. **Towards autonomous behavior learning of non-player characters in games** [Kirjan osa] / tekijä Shu Feng ja Ah-Hwee Tan // Expert Systems with Applications. - 2016.
23. **Using provenance data and imitation learning to train human-like bots** [Aikakausjulkaisu] / tekijä Lauro Víctor Ramos Cavadas [ym.] // Entertainment Computing. - 2024. - ss. 1-9.
24. **What Is AI?** [Online] / tekijä Caltech // Science Exchange. - 12. 9 2022. - <https://scienceexchange.caltech.edu/topics/artificial-intelligence-research/artificial-intelligence-definition>.