



Tekoäly pelituotannossa

Juho Kivinen

Haaga-Helia ammattikorkeakoulu

Tradenomi

Opinnäytetyö

2023

Tiivistelmä

Tekijä(t) Juho Kivinen
Tutkinto Tradenomi
Raportin/Opinnäytetyön nimi Tekoäly pelituotannossa
Sivumäärä 26
<p>Tekoälyn käyttö on merkittävästi kasvanut viime vuosina. Tähän on vaikuttanut se, että kone- ja syväoppimisessa on otettu suuria harppauksia, joten pelikehittäjät ovat voineet valjastaa tekoälyn paremmin käyttöönsä luodessaan uusia pelejä. Tutkimuksessa tutkittiin sitä, miten tekoälyn avulla voidaan parantaa käyttäjäkokemusta ja mitä haasteita sen käyttöön liittyy pelikehityksessä.</p> <p>Merkittävimpiä pelimoottoreita ovat Unity ja Unreal Engine. Näistä Unity on helpompi opeteltava sekä soveltuvampi mobiilipelien luomiseen, kun taas Unreal Engine mahdollistaa hyvien grafiikoiden luomisen, jonka takia suurimmat pelistudiot sitä käyttävät.</p> <p>Proseduraalinen generointi on tekoälytekniikka, jolla kokonainen pelimaailma tai muu pelin sisältö luodaan algoritmien avulla joka kerta uniikkina siemenarvon avulla. Se eroaa satunnaisgeneroinnista siten, että satunnaisgeneroitu data saadaan valmiiksi määritetystä joukosta arpomalla. Tällöin pelikokemusta pystytään hallitsemaan paremmin, kuin kokonaan tyhjästä luodussa maailmassa.</p> <p>Toinen merkittävä tekoälytekniikka on mukautuva vaikeusjärjestelmä. Nimensä mukaisesti järjestelmän avulla pelin vaikeustasoa voidaan muokata reaaliajassa pelaajan suorituksen mukaan. Täten pelikokemus pysyy sopivan haastavana ja sujuvana.</p> <p>Pelien testauksessa tekoäly pystyy tunnistamaan virheet jo lähdekoodissa ja ehdottaa korjauksia. Tekoälyn avulla voidaan myös luoda monipuolisia testitapauksia, joita voidaan käyttää ehkäisemään virheitä.</p> <p>Tutkimuksessa päästiin lopputulokseen, jossa proseduraalinen generointi ja mukautuva vaikeusjärjestelmä koettiin merkittävimmiksi tekoälytekniikoiksi pelikehityksessä. Aiheesta riittää yhä tutkittavaa ja jatkotutkimuksessa voitaisiin tutkia esimerkiksi tekoälyn luonnollisten kielten käsittelyä, joka on esillä peleissä, jossa tekoälyn kanssa käydään keskustelua.</p> <p>Tutkimus kehitti tekijän taitoa kirjoittaa puhtaasti lähteisiin perustuvaa tekstiä. Työ antoi myös tekijälle avaramman käsityksen tekoälyn toiminnasta ja sen käyttötavoista, joka on oleellista kirjoittajan ammatillisen kehityksen puolesta.</p>
Asiasanat Tekoäly, videopelit, koneoppiminen, syväoppiminen, käyttäjäkokemus

Sisällys

1	Johdanto	1
2	Pelimoottorien yleiskuvaus	3
2.1	Unity	3
2.1.1	Unity pelimoottorina	3
2.1.2	Merkittäviä Unity-pelejä	4
2.2	Unreal Engine	5
2.2.1	Unreal Engine pelimoottorina	5
2.2.2	Merkittäviä Unreal Engine-pelejä	6
2.3	Unity vai Unreal Engine	7
3	Tekoälytekniikat	8
3.1	Proseduraalinen vai satunnaisgenerointi	8
3.1.1	Esimerkkejä proseduraalisesta generoinnista	8
3.1.2	Esimerkkejä satunnaisgeneroinnista	9
3.2	Käyttätymispuut	10
3.3	Mukautuva vaikeusjärjestelmä	11
4	Tekoäly pelin testausvaiheessa	14
4.1	Test case generation	14
4.2	Tekoäly virheiden havainnoinnissa	14
5	Eettisyys tekoälyn käytössä	16
5.1	Tekoälyn luoma sisältö	16
6	Tutkimus ja tulokset	17
6.1	Tutkimusmenetelmä	17
6.2	Tulokset	17
7	Johtopäätökset ja pohdinta	19
7.1	Johtopäätökset ja suositukset	19
7.2	Opinnäytetyö ja oma oppiminen	19
	Lähteet	20

1 Johdanto

Tekoälyä käytetään nykyisin yhä enemmissä määrin pelituotannossa niin tasojen, hahmojen kuin tehtävien luomiseen. Opinnäytetyössä tutkin tekoälyn käyttöä pelituotannossa historian saatossa ja pyrin selvittämään, miten tekoäly voidaan valjastaa luomaan peleihin ominaisuuksia pelikokemuksen parantamiseksi.

Oma kiinnostukseni aiheeseen juontaa juurensa vuosien kokemuksiin tekoälyn ja pelien käyttäjänä. Viimeisen parin vuoden aikana tekoäly on tullut itsellenikin hieman tutummaksi, joten tutkimus on hyvä tapa yhdistää nämä kiinnostuksenkohteet.

Aihe on erityisen ajankohtainen, sillä kone- ja syväoppimisessa on otettu suuria harppauksia viime vuosina. Pelikehittäjät pystyvät täten ottamaan tekoälyn entistä tehokkaammin käyttöön peliä luodessa. Pelimarkkinat ovat myös yksi suurimmista ja nopeimmin kasvavista viihdeteollisuuden osa-alueista, jonka takia kehittäjät haluavat luoda tekoälyn kanssa menestyviä pelejä ja lisätä tuottoa.

Tutkimuksessa selvitän, kuinka tekoälyä voidaan käyttää pelituotannossa siten, että se parantaa käyttäjäkokemusta ja mitä haasteita sen käytössä on pelin tekijöille? Aiheeseen liittyy muun muassa seuraavat alaongelmat.

- Mitä eettisiä asioita tulee ottaa huomioon, kun tekoäly otetaan avuksi pelituotantoon?
- Mitä testausvälineitä tekoäly tuo pelien testausvaiheeseen ja miten tekoäly tunnistaa ongelmia sekä ilmoittaa niistä ohjelmoijille.

Aiheen tutkimiseen on käytetty OpenAi:n GPT-3.5-tekoälyä. Tekoälyllä kartoitettiin tutkimusongelmaa sekä sen alaongelmia.

Tutkimusmenetelmä on kirjallisuuskatsaus, jossa perehdyn aiheesta löytyviin lähteisiin. Tutkimuksen tavoitteena on lähteitä hyödyntämällä pystyä vastaamaan tutkimusongelmaan sekä sen alaongelmiin.

Keskeisiä käsitteitä:

Koneoppiminen = Tietokoneen kyky oppia ja parantaa suorituskkyä kokemuksen kautta, ilman varsinaista ohjelmointia (Nieminen 2023).

Käyttäytymispuut = Hierarkkisia solmuja sisältävä käyttäytymispuu kontrolloii tekoälyn tekemiä päätöksiä (Simpson 2014).

Luonnollisen kielen prosessointi = Tekniikoiden ja työkalujen tutkimista ja soveltamista, joiden avulla tietokoneet voivat käsitellä ja perustella ihmisten kieltä (Nelson,2022).

Siemenarvo = Satunnainen merkkijono, jota käytetään luomaan uniikkia sisältöä. Samaa siemenarvoa käyttämällä saadaan aina sama lopputulos. (Wicklin 2017.)

Syväoppiminen = Tekoälymenetelmiä, jotka hyödyntävät lukuisia neuroverkkokerroksia ja massiivisten aineistojen prosessointia (Kärkkäinen & Hänninen 2023).

Tekoäly = Tarkoittaa koneen kykyä jäljitellä inhimillistä päättelyä.

Proseduraalinen generointi = Datan tuottamista algoritmien avulla sen sijaan, että dataa tuotettaisiin manuaalisesti (Kenny 2021).

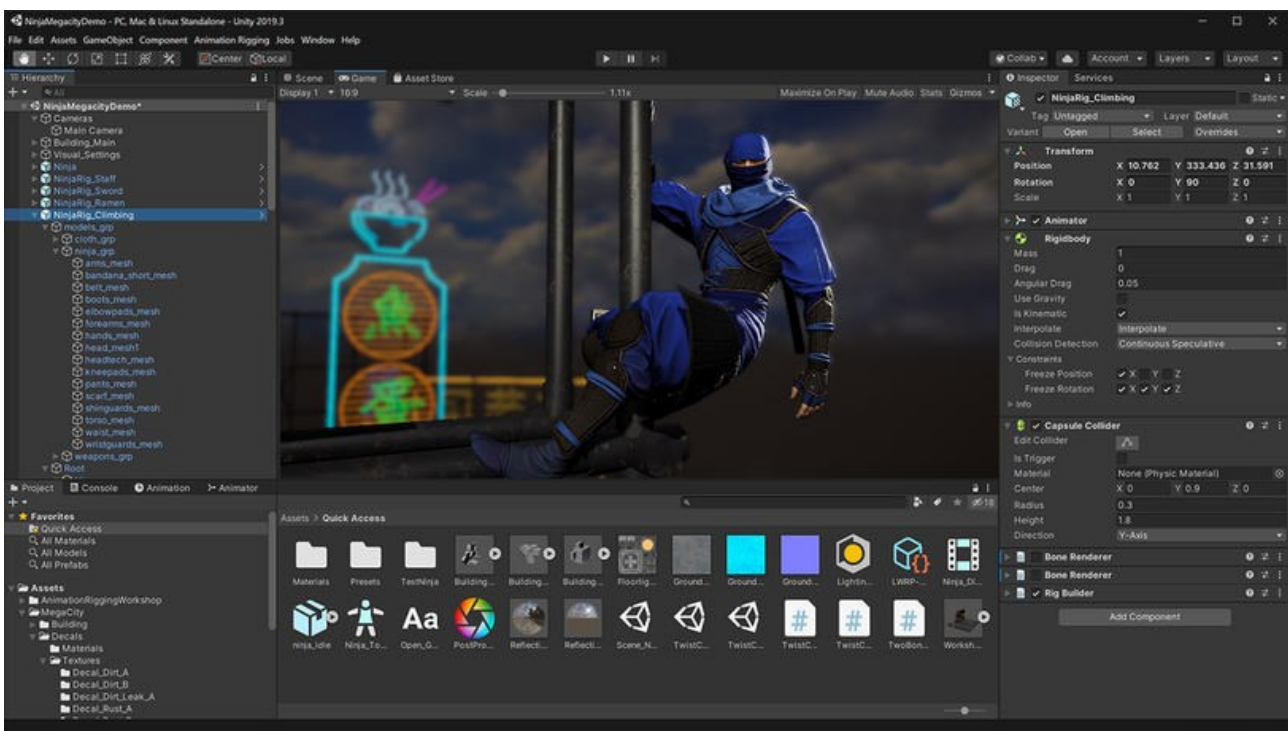
2 Pelimoottorien yleiskuvaus

Tässä luvussa tutustutaan kahteen merkittävään pelimoottoriin nimeltä Unity ja Unreal Engine. Tutustutaan pelimoottorien ominaisuuksiin sekä vertaillaan pelimoottoreita vastakkain. Tarkastellaan myös merkittävimpiä videopelejä, jotka on luotu näitä pelimoottoreita hyödyntäen.

2.1 Unity

Unity on alustariippumaton pelimoottori, jonka on kehittänyt Unity Technologies. Unity tarjoaa tukea yli 25 pelialustalle, joiden joukkoon kuuluu muun muassa pelikonsolit Xbox ja Playstation. Useat pelikehittäjät ovat ottaneet Unityn käyttöönsä, sillä tällä pystytään luomaan 3D-, 2D- sekä VR-pelejä. (Arora 2023.)

2.1.1 Unity pelimoottorina



Kuva 1 Unity pelimoottori (Unity)

Unity tarjoaa pelinkehittäjille runsaasti hyviä ominaisuuksia. Pelimoottori tukee niin 2D- kuin 3D-grafikoita. Jokaisen pelikentän luomiseen on tarjolla hyvät työkalut. Pelikentät on jaettu niin sanottuihin kohtauksiin, jossa on käytössä muun muassa pelihahmoja, vihollisia tai taustoja. Pelimoottori mahdollistaa myös useamman objektin kuten aseiden tai vaatekappaleiden lisäämisen pelihahmoon. (Zenva 2023.) Käyttöliittymän keskellä näkyy esikatselu luotavasta pelistä. Alavalikossa sijaitsee pikavalinnat eri elementeille kuten valaistukselle ja rakennuspalikoille. Vasemmalta valikosta

voidaan vaihtaa muokattavan kohtauksen välillä, muokaten esimerkiksi yksittäistä objektia pelihahmon sijaan. (Kuva 1. Unity-käyttöliittymä)

Unityn avulla pelejä voidaan myös luoda usealle eri alustalle. Pelit voidaan viedä pelikonsolien lisäksi Androidille ja iOS:lle. Unity onkin nykypäivänä enimmäkseen suosittu mobiilipelien luomiseen sekä pienempien pelistudioiden käyttöön. (Akhil 2022.)

2.1.2 Merkittäviä Unity-pelejä

Muutaman viime vuoden suosituimpia pelejä ollut Fall Guys on luotu Unity-pelimootoria käyttäen. Peli saavutti kesällä 2022 50 miljoonan pelaajan rajapyykin kahteen viikkoon. Samana kesänä kehittäjät muuttivat pelin ilmaiseksi, jonka sisällä tapahtuvilla mikrotransaktioilla pelaaja voi ostaa itselleen lisäsisältöä ja täten tukea pelikehittäjiä. Hahmoille puettavat vaatekappaleet ovat suuri osa pelikokemusta. (Warner 2022.) Kuvassa pelaajien ohjaamat hahmot kisailevat esterataa pitkin. Ensimmäinen maalissa on voittaja. (Kuva 2. Fall guys-hahmoja)



Kuva 2 Fall guys-hahmoja (Fall guys)

Yksi maailman menestyneimpiä mobiilipelejä on Unity-pelimootorilla luotu Pokemon GO, joka julkaistiin vuonna 2016. Pelissä hyödynnetään lisättyä todellisuutta tavalla, jossa pelissä napattavia

otuksia asetetaan puhelimen kameran myötä oikeaan maailmaan. (Chamary 2018.) Vuonna 2021 peliä ladattiin 572 miljoonaa kertaa ja se tuotti 1.21 miljardia (Iqbal 2023). Peliä pelataan liikkumalla luonnossa, sekä osoittamalla mobiililaitteen kameraa kohti sijaintia, jossa pokemon sijaitsee. Tällöin pokemon ilmestyy niin sanotusti oikeaan maailmaan, jolloin se on mahdollista napata. (Kuva 3. Pokemon luonnossa)

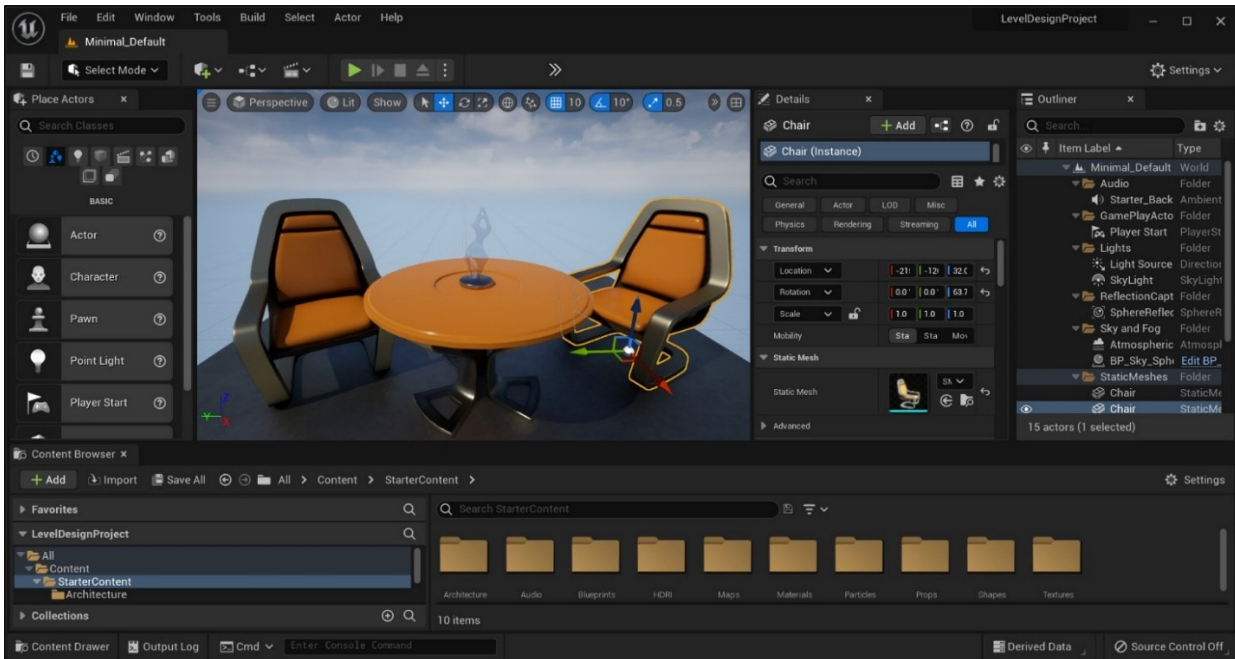


Kuva 3 Pokemon luonnossa

2.2 Unreal Engine

2.2.1 Unreal Engine pelimoottorina

Unreal Engine on videopelien kehitystyökalu, jonka on kehittänyt ohjelmistokehittäjä Epic Games. Työkalulla on mahdollista luoda simulaatioita, editoida video- tai ääniklippejä sekä renderöidä animaatioita (Moss s.a.). Lähes kaikki AAA-pelejä luovat pelikehittäjät käyttävät nykypäivänä Unreal Engineä sen tarjoamien mahdollisuuksien takia. Unreal Engine 5 käyttää tekoälyyn perustuvaa Lumen-systeemiä, jonka ansiosta kehittäjien ei tarvitse manuaalisesti luoda heijastuksia tai valojen yksityiskohtia. (Akhil 2022.) Käyttöliittymän keskellä näkyy esikatselu muokattavasta kohtauksesta. Alavalikossa voidaan valita eri tekstuureja tai materiaaleja. (Kuva 4. Unreal Engine 5-level editor)



Kuva 4. Unreal Engine 5-level editor (Unreal Engine)

2.2.2 Merkittäviä Unreal Engine-pelejä

Edelleen 6 vuotta julkaisunsa jälkeen Epic Gamesin luoma Fortnite toimii yhtenä maailman pela-tuimmista videopeleistä. 2023 pelillä on 500 miljoonaa rekisteröityä käyttäjää, joista 231 miljoonaa on kuukausittain aktiivisia pelaajia. Arvioiden mukaan peli on tuottanut 2017–2022 välillä yli 23 miljardia dollaria. (Shewale 2023.) Pelin suosioon vaikuttaa epäilemättä sen olevan ilmainen. Lisäksi peli on tarjolla joka alustalle mobiililaitteita myöten. Peli on mekaniikoiltaan lähellä muita saman genren pelejä. Battle royale-genressä pelaaja kilpailee yksin tai joukkueen kanssa muita pelaajia vastaan koko ajan kutistuvalla peliarenalla, kunnes pelin päätteeksi yksi pelaaja tai joukkue on jäljellä. Fortnite eroaa mekaniikoiltaan siinä, että pelissä on mahdollista asettaa ansoja tai rakentaa erilaisia rakennelmia, joiden avulla on mahdollista saada etulyöntiasema muihin pelaajiin verrattuna. (Roach 2021.) Kuvassa pelaajan ohjaama hahmo on etuosassa oleva pinkki nallekarhu, taustalla näkyy muita pelaajia ampumassa sekä rakentamassa siltaa talon katolle. (Kuva 5. Tais-telu)



Kuva 5. Taistelu (Epic Games)

2.3 Unity vai Unreal Engine

Aloitellessa pelikehittämistä, suositellaan Unityn käyttöä. Unityn käyttö on helppo oppia ja käyttäjä voi mahdollisesti jo viikonlopun mittaisen kurssin jälkeen luoda ja julkaista yksinkertaisia pelejä. Unreal Engine käyttää C++ -ohjelmointikieltä, joka on hankalampi opetella kuin Unityn käyttämä C#. Alustaa pohtiessa tulisi myös miettiä, minkälaisia pelejä luodaan ja mille alustalle niitä halutaan julkaista. Sekä Unity, että Unreal Engine kykenevät luomaan pelejä uusille pelialustoille, mutta Unityllä on helpompi luoda 3D-pelejä, jonka takia sitä suositellaan mobiilialustoille luotaviin peleihin. Mikäli halutaan luoda hyvillä grafiikoilla varustettu peli, suositellaan Unreal Enginen käyttöä. Suurin ero alustoiden välillä onkin juuri grafiikoiden tasoero. Unity on kehittänyt omia grafiikoitaan vuosien saatossa, mutta yhä suurimmat pelistudiot käyttävät Unreal Engineä sen tarjoamien grafiikoiden takia. (Tristem s.a.)

3 Tekoälytekniikat

3.1 Proseduraalinen vai satunnaisgenerointi

Proseduraalinen generointi pelituotannossa on datan tuottamista algoritmien avulla sen sijaan, että data tuotettaisiin manuaalisesti. Ensimmäiset merkittävät pelit, jotka käyttivät proseduraalista generointia olivat vuonna 1980 julkaistu Rogue sekä 1984 julkaistu Elite. Peleissä kuljetaan labyrinttien kaltaisissa pelikentissä, jotka on luotu algoritmien avulla helpottaen kehittäjien työtaakkaa. Pääosin proseduraalista generointia käytetään pelin uudelleenpelaamisen arvon nostamiseksi, sillä pelikentät luodaan joka pelikerralla erilaisina. (Kenny 2021.)

Satunnaisesti generoidussa pelituotannossa nimensä mukaisesti pelimaailman sisältö luodaan valmiiksi koodatuista elementeistä satunnaislukugeneraattoreita hyödyntäen. Tätä tekniikkaa hyödyntäen peliin saadaan uudelleenpelausarvoa samalla kontrolloiden pelikokemusta ennalta määritettyjen elementtien avulla. (Bycer 2015.)

3.1.1 Esimerkkejä proseduraalisesta generoinnista

Minecraft-pelissä pelaajan luodessa uuden maailman, pelaaja saa täysin uuden ympäristön seikkailtavakseen. Pelin viholliset, resurssit sekä päivä-yö sykli on ennalta koodattu, mutta pelimaailma luodaan tyhjästä aina, kun uusi maailma aloitetaan. (Bycer 2015.) Pelimaailman luonnissa käytettävä siemenarvo on lyhyt numerosarja tai teksti. Minecraft maailman luodessa 64-bittinen siemenarvo kutsutaan, joka puolestaan luo pelimaailman tyhjästä. Minecraftin lähdekoodi tunnistaa 4 miljardia sanaa, jonka ansiosta pelkkiä sanoja käyttämällä on mahdollista luoda 4 miljardia teoreettista pelimaailmaa. (Varikuti 2022.) Kuvassa Minecraft-maailma, joka on luotu siemenarvoa käyttäen. (Kuva 6. Minecraft-maailma)



Kuva 6. Minecraft-maailma (Varikuti, 2022)

2016 julkaistu No Man's Sky käyttää proseduraalista generointia pelimaailman luomiseen. Pelimaailma koostuu 18.4 kvintiljoonasta planeetasta. Pelimaailmaan luotuun sisältöön lukeutuvat yksityiskohtaiset kasvustot, erilaiset avaruusoliot sekä täysin toimivat avaruusaluukset ja rakennukset. Pelimaailma luodaan siemenarvolla, joka algoritmeja hyödyntäen määrittelee jokaisen pelin osan alueen. (Press 2016.) Vaikka pelaaja vierailisi yhdellä planeetalla joka sekunti, jokaisen pelin planeetan vierailuun kuluisi noin 584 miljoonaa vuotta (Good 2014). Jokainen planeetta on proseduraalisesti generoitu tyhjästä. Kuvassa yksi kyseisistä planeetoista, jossa esiintyy kasvillisuutta sekä muutama avaruusalus. (Kuva 7. Proseduraalisesti generoitu planeetta)



Kuva 7. Proseduraalisesti generoitu planeetta (Good 2014)

3.1.2 Esimerkkejä satunnaisgeneroinnista

Binding of Isaac on esimerkki pelistä, jossa pelimaailman elementit on koodattu valmiiksi kehittäjien toimesta. Kaikki huoneet on ennalta luotu ja pelin edetessä ne nidotaan yhteen maailman luomiseksi. Vaikka huoneiden järjestys on jokaisella pelikerralla erilainen, huoneita on luotu rajallinen määrä. (Bycer 2015.) Myös huoneissa kerättävät tavarat generoidaan satunnaisesti ennalta määritetystä joukosta. (Kuva 8. Satunnaisgeneroitu huone)

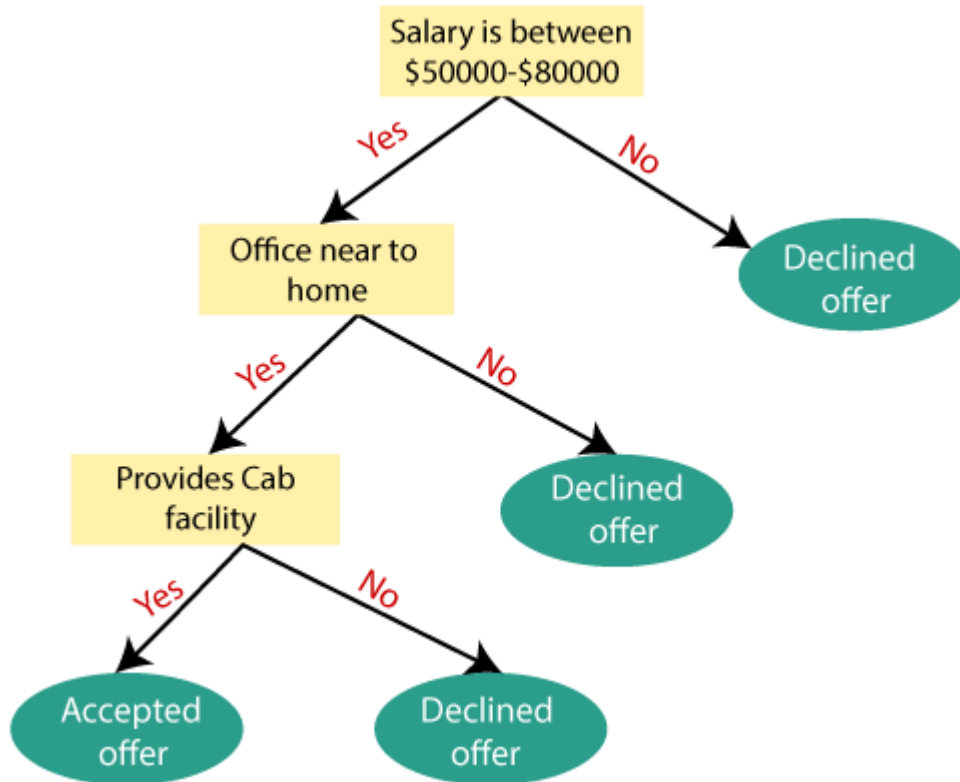


Kuva 8. Satunnaisgeneroitu huone (Polygon 2014)

3.2 Käyttäytymispuut

Käyttäytymispuita on koneoppimisessa monenlaisia, mutta kaksi päätyyppiä ovat luokittelu sekä regressio. Luokittelumallisissa käyttäytymispuissa vastaukset kysymyksiin valitaan aina ennalta määritetystä joukosta dataa. Regressiomallisissa puolestaan tuotetaan vastaukseksi lukuarvo, jonka ei tarvitse olla kokonaisluku. Regressiomallilla voidaan vastata tilanteisiin, jossa arvioidaan etäisyyttä esteeseen tai tulevan elokuvan lipputuloja. (Gopalan 2020.)

Käyttäytymispuut näyttävät vuokaavioilta, jotka alkavat juurisolmusta. Alussa pyydetään vastausta kysymykseen, joka johtaa eri lopputulemia tai seuraavia askeleita sisältäviin haaroihin. Tämä toistuu, kunnes saavutetaan niin sanottu terminaalisolmu, joka lopettaa käyttäytymispuun. (Coursera 2023.) Yksinkertainen käyttäytymispuu on nähtävissä kuvassa 9. Käyttäytymispuun alkuvaiheessa tekoäly kysyy yksinkertaisen kysymyksen, tässä tapauksessa sen, onko palkka 50000–80000 dollarin välillä. Mikäli käyttäjä vastaa negatiivisesti, käyttäytymispuu päättyy terminaalisolmuun ja kysely lopetetaan. Käyttäjän vastatessa positiivisesti, jatketaan kysymysten kysymistä ennalta määritettyyn pisteeseen saakka tai kunnes päädytään uuteen terminaalisolmuun.



Kuva 9. Käyttäytymispuu (Javapoint)

3.3 Mukautuva vaikeusjärjestelmä

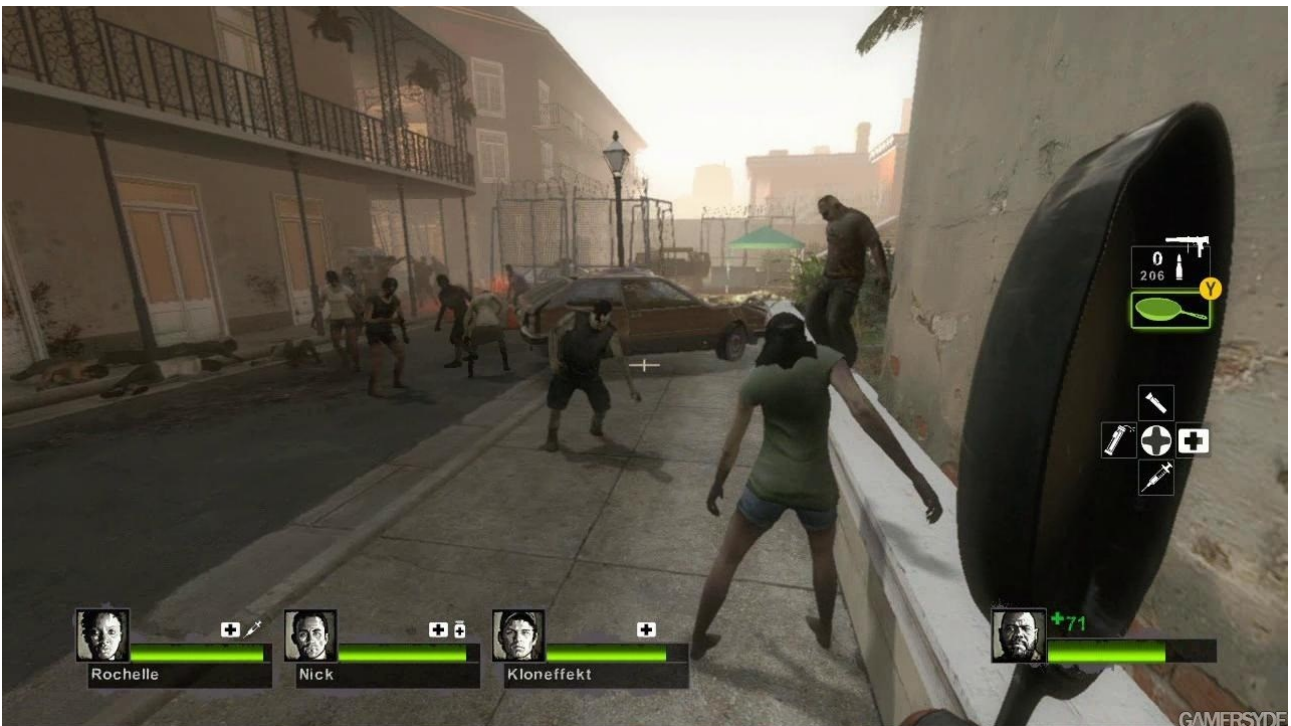
Videopelien maailmassa vaikeusaste on yksi suurimmista pelikokemusten muovaajista. Pelaajan viihtyvyyden ja sopivan haasteen tasapainottaminen on haaste pelikehittäjille. Tekoälyn noustua pintaan on pelien vaikeustasoa mahdollista muovata pelin edetessä pelaajan kykyjen mukaan. Peli reagoi pelaajan taitotasoon ja muokkaa vihollisten ja haasteiden vaikeustasoa sen mukaan, miten hyvin pelaaja etenee. Näin pelikokemus saadaan pidettyä personalisoituna sekä sopivan haastavana jokaiselle pelaajalle. (Fintelics 2023.)

Mukautuvan vaikeusjärjestelmän suurin hyöty onkin pelaajien turhautumisen vähentäminen. Kiinteässä vaikeustasossa monet pelaajat saattavat jäädä jumiin tiettyyn pelin vaiheeseen, eivätkä täten pääse etenemään pelissä. Dynaamisesti muokkaamalla vaikeustasoa, pääsevät pelaajat etenemään omassa tahdissaan jäämättä jumittamaan tiettyihin kohtiin. (Frackiewicz 2023.)

Mukautuva vaikeusjärjestelmä tuo mukanaan myös haasteita. Pelaajat saattavat tuntea pelin epäreiluksi, mikäli vaikeustason vaihtelu on liian suurta pelin sisällä. Vastatakseen tähän, monet pelikehittäjät antavat pelaajalle vaihtoedon mukautuvan ja kiinteän vaikeustason välillä.

Tekoälyjärjestelmälle voi myös olla haastavaa arvioida pelaajan taitotasoa tarkasti. Tämä on erityisen haastavaa peleissä, joissa on monimutkaisia mekaniikoita tai monia tapoja lähestyä pelitilanteita. Tekoäly kuitenkin kehittyy jatkuvasti, joka tarkoittaa, että nämä järjestelmät tulevat entistä tarkemmiksi ja virheiden mahdollisuus pienenee. (Frąckiewicz 2023.)

Yksi tunnetuimmista peleistä, joka käyttää mukautuvaa vaikeusjärjestelmää on Left 4 Dead-pelisarja. Järjestelmä monitoroi pelaajan suoritusta ja synnyttää enemmän tai vähemmän vihollisia riippuen siitä, kuinka hyvin pelaaja pärjää. Tekoäly myös tarjoaa ylimääräisiä resursseja ja jopa muo-
vua pelialuetta pitääkseen kokemuksen mielenkiintoisena ja pelaajan taitotasoon muokattuna. (Frąckiewicz 2023.) Kuva pelistä Left 4 Dead 2, nähtävissä suuri määrä vihollisia, joiden määrä perustuu pelaajan taitotasoon. (Kuva 10. Left 4 Dead.)



Kuva 10. Left 4 Dead

Toinen esimerkki mukautuvaa vaikeusjärjestelmää käyttävästä pelisarjasta on Forza Motorsport. Pelin Drivatar-niminen tekoälyjärjestelmä muovaa vastakilpailijoiden tasoa vastaamaan pelaajan omaa tasoa. Drivatar ei ainoastaan kopioi asioita, joita pelaaja itse tekee, vaan se osaa myös ennakoita miten pelaaja toimisi tietyissä tilanteissa. (Thompson 2021.) Tämän ansiosta kilpailut tuntuvat realistisemmilta, sillä tekoälyn ohjaamat kilpailijat ajavat lähes oikeiden ihmisten lailla. Vaikeustaso on myös tällöin hyvin vertailtavissa itse pelaajaan. (Frąckiewicz 2023.)

Kuvassa pelaajan ohjaama keltainen auto jahtaa tekoälyn ohjaamia kilpailijoita, joiden ajotyyli ja taitotaso määräytyy pelaajan omien taitojen perusteella. (Kuva 11. Kilpailu)



Kuva 11. Kilpailu (Golson 2014)

4 Tekoäly pelin testausvaiheessa

4.1 Test case generation

Tekoälyn käyttö testitapauksien luomisessa sisältää syväoppimisalgoritmeja, luonnollisen kielen käsittelyä sekä muita tekoälytekniikoita. Näitä tekniikoita käytetään analysoimaan sovelluksen käyttäytymistä, käyttäjävirtoja sekä koodin rakennetta. Tekoälyn avulla voidaan luoda laajoja sekä tehokkaita testitapauksia. (Cser 2023.)

Tekoälyä käytettävissä testitapauksissa luomisprosessi sisältää yleensä seuraavia askeleita:

1. Datan keräys ja analysointi. Tekoälyjärjestelmä kerää dataa useista lähteistä, kuten käyttäjien vuorovaikutuksesta toisiinsa, sovelluksen lokitiedoista sekä koodivarastoista. Datan avulla tekoäly tunnistaa käyttäjien käytöstä sekä mahdollisia riskejä.
2. Testitapauksen luonti. Kerättyyn dataan viitaten, tekoälyjärjestelmä luo lukuisia eri testitapauksia, joissa se käy läpi mahdollisia ongelmia ja skenaarioita. Tekoäly käyttää luonnollisen kielen käsittelyä koodin jäsentelyyn ja mahdollisten skenaarioiden tunnistamiseen sekä syväoppimisalgoritmeja ennakoimaan ja luomaan relevantteja testitapauksia.
3. Tekoälyn ohjaama testitapausten luontityökalu optimoi luodut testitapaukset tunnistamalla niistä turhat sekä priorisoimalla kriittiset skenaariot. Tällä varmistetaan testitapausten olevan tehokkaita
4. Jatkuva oppiminen ja sopeutuminen. Sovellusten kehittyessä tekoälyjärjestelmä oppii uudesta datasta kuten koodimuutoksista, bugiraporteista sekä käyttäjien palautteesta. Näin tekoälyllä luodut testitapaukset voivat sopeutua muutoksiin sovelluksessa ja testin tulokset pysyvät ajankohtaisina. (Cser 2023.)

4.2 Tekoäly virheiden havainnoinnissa

2019 julkaistu Clever-Commit on Ubisoftin kehittäjien luoma työkalu virheiden tunnistamiseen. Ubisoftin mukaan, järjestelmä käyttää viimeisimpiä koneoppimistekniikoita analysoimaan aikaisempia virheitä ja arvioimaan, tuoko uusi koodi mukanaan uusia virheitä. Työkalu vapauttaa arviolta 20 prosenttia pelikehittäjien ajasta. (Cardoza 2019.)

Alkuvuodesta 2023 Twitterissä esiteltiin virheenkorjaustyökalua nimeltä Wolverine. Käyttäjän luoma koodi ajetaan ohjelmassa ja sen kaatuessa virheen vuoksi ohjelma tunnistaa missä virhe sijaitsee. Tämän jälkeen ohjelma korjaa koodin ja selittää mikä koodissa oli väärin. Vaikka virheitä olisi useita, ohjelma ajaa korjauksia niin kauan, kunnes kaikki bugit on saatu korjattua. (Obedkov 2023.) Kuvassa 12 näkyy selitys virheelle ja tekoälyn tarjoamat korjaukset koodiin.

```
Explanations:
- The error occurs because the subtract_numbers
  pt.
- Variable 'res' is not defined, should use 're:

Changes:
--- +++ @@ -4,6 +4,7 @@ def add_numbers(a, b):
    return a + b

+
def subtract_numbers(a, b):
    return a - b

    def multiply_numbers(a, b):
        return a * b
@@ -25,7 +26,7 @@     else:
        print("Invalid operation")

-     return res
+return result
```

Kuva 12. Wolverine (Obedkov 2023)

5 Eettisyys tekoälyn käytössä

5.1 Tekoälyn luoma sisältö

Pelikehittäjien käyttäessä tekoälyä yhä enemmän luodakseen realistisia ja koukuttavia pelikokemuksia, nousee esiin kysymys eettisyydestä. Käsitys realismista on yksi suurimmista huolenaiheista, koska tekoäly mahdollistaa yhä enemmän realististen pelimaailmojen luomisen. Tämä voi aiheuttaa sekaannusta tai vääristymiä oikean maailman ja pelimaailman välillä joillekin pelaajille. (Frąckiewicz 2023.)

Tekoäly saattaa myös luoda sisältöä, joka ei ole realistista, tai sisältöä, joka on rasistista, seksististä tai väkivaltaista. Nämä sisällöt voivat vaikuttaa negatiivisesti pelaajan asenteeseen tai arvo maailmaan. Pelikehittäjien tulisikin pitää mielessä eettiset vaikutukset luodessaan sisältöä tekoälyä hyödyntäen. Tekoälyn käyttö herättää myös kysymyksen mahdollisesta valvonnasta ja datan keräilystä. Monet tekoälyä hyödyntävät pelit vaativat pelaajaa syöttämään henkilökohtaista dataa kuten ikää, sukupuolta tai sijaintia. Pelikehittäjien tulisikin olla avoin datan käytön suhteen ja kertoa pelaajalle, mihin tarkoitukseen pyydettyä dataa käytetään, sekä tarjota pelaajalle mahdollisuus kieltäytyä datan keräämisestä. (Frąckiewicz 2023.)

6 Tutkimus ja tulokset

6.1 Tutkimusmenetelmä

Tutkimuksessa halusin selvittää, miten tekoäly voidaan valjastaa käyttöön pelikehityksessä ja mitä hyötyä siitä on pelikehittäjille. Selvitän myös tekoälyn tuomia eettisiä ongelmia pelikehityksessä sekä sitä, miten tekoälyä hyödynnetään pelien testausvaiheessa. Muun muassa näihin asioihin etsin vastauksia lähdemateriaaleista.

Opinnäytetyöhön on tiukka aikataulu, jonka takia tutkimusmenetelmä on kvalitatiivinen kirjallisuuskatsaus, jossa tarkoituksena on kerätä olemassa olevaa tietoa tekoälyn käytöstä pelituotannossa. Oleellista kyseisessä tutkimusmenetelmässä on kerätä laaja kokoelma kirjallisuutta ja tutustua aiheeseen näiden avulla perusteellisesti.

Tietoa haettiin lähinnä elektronisista lähteistä, joita olivat erilaiset blogipostaukset, tieteelliset tutkimukset sekä pelimoottorien omat kotisivut. Merkittäviä lähteitä tutkimuksen kannalta ovat proseduraalista ja satunnaisgenerointia vertaileva blogi (Bycer 2025) sekä mukautuvaa vaikeusjärjestelmää käsittelevä uutinen (Frąckiewicz 2023).

6.2 Tulokset

Tutkimuksessa haluttiin selvittää, miten tekoälyä käytetään pelikehityksessä ja mitä hyötyä siitä on kehittäjille. Proseduraalinen generointi on merkittävä tapa hyödyntää tekoälyä luomaan pelille uudelleenpelausarvoa (Kenny 2021). Lisäksi pelien vaikeustasoa voidaan mukauttaa reaaliajassa vastaamaan pelaajan omaa tasoa, jolloin pelikokemus on mukavampi ja sopivan haastava koko kokemuksen ajan. (Frąckiewicz 2023.) Itse pyrin aina valitsemaan mukautuvan vaikeustason, mikäli sellainen on pelissä tarjolla. Tällöin pelikokemukseni ei tunnu liian helpolta tai vaikealta, enkä joudu itse säätämään vaikeustasoa kesken pelin. En muista kertaakaan huomanneeni, että vaikeustaso ei mukautuisi omaan tasooni. Proseduraalinen generointi on minulle tuttua muun muassa Minecraft-pelistä, jossa pelimaailma luodaan aina niin sanotusti tyhjästä. Toivoisinkin, että useammat pelit hyödyntäisivät tätä tekoälytekniikkaa, sillä pelikokemus tuntuu joka kerta virkistävältä ja uudelleenpelausarvo on korkea.

Lisäksi selvitettiin, mitä eettisiä asioita tulee ottaa huomioon, kun käytetään tekoälyä pelien luomiseen. Pääpointit tekoälyn eettisyydessä liittyvät sen luomaan sisältöön. Sisältö voi pahimmillaan olla hyvin rasistista, seksististä tai muuten epäeettistä, jonka takia pelikehittäjien tulee valvoa tekoälyn luomaa sisältöä erittäin tarkasti. (Frąckiewicz 2023.) Itse en ole juurikaan törmännyt

pelatessani sisältöön, jonka olisin ajatellut olevan jollakin tapaa epäeettistä. Uskon, että tekoälyä valvotaan hyvin tarkkaan, jolloin näitä teemoja ei juuri nähdä nykypäivän videopeleissä.

Selvitettiin myös sitä, kuinka tekoälyä hyödynnetään pelien testausvaiheessa. Tekoälyä käytetäänkin luomaan testitapauksia, jossa simuloidaan mahdollisia tapahtumia pelin edetessä. Tekoäly mahdollistaa laajojen ja tehokkaiden testitapausten luomisen. Tekoäly myös oppii jatkuvasti, jolloin testitapaukset ja täten testien tulokset pysyvät ajankohtaisina. Tekoäly mahdollistaa myös virheiden tunnistamisen jo kehitysvaiheessa ja sitä voidaankin käyttää tarjoamaan vaihtoehtoisia ratkaisuita ja korjauksia pelin koodiin. (Cser 2023.)

7 Johtopäätökset ja pohdinta

7.1 Johtopäätökset ja suositukset

Tutkimuksessa selvisi, että proseduraalinen generointi on merkittävä osa pelituotantoa. Pelin uudelleenpelausarvoa nostamalla saadaan pelaaja pelaamaan peliä enemmän, joka puolestaan nostaa tuottoja. Pelikehittäjien kannattaisikin tutustua aiheeseen enemmän ja miettiä, miten proseduraalisen generoinnin voisi valjastaa omaan peliinsä.

Toinen merkittävä tutkimuksessa selvinnyt asia on mukautuva vaikeusjärjestelmä. Tutkimuksessa selvisi, että tällä tavoin pelikokemuksesta saadaan entistä sulavampi sekä mielenkiintoisempi. En näe haittaa siinä, että vaikeustasoa ei saisi jatkossa yhä useammassa pelissä valittua mukautuvaksi.

Yleisesti tuloksista voidaan päätellä, että tekoälyyn tulee panostaa pelikehittäjien osalta tulevaisuudessa yhä enemmän. Tekoälyn tuottamaa sisältöä tulee myös tulevaisuudessa tarkkailla tarkasti, jotta peleihin ei pääse epäeettistä sisältöä. Uskallan väittää, että tekoälyn käyttö kasvaa jatkossa merkittävästi sen tarjoamien mahdollisuuksien takia.

Tutkimuksen tulokset ovat linjassa omien ajatusteni kanssa. Ennen tutkimusta tiesin proseduraalisesta generoinnista sekä mukautuvasta vaikeusjärjestelmästä. Nämä asiat olivat jo ennestään sellaisia, jotka saavat minut kiinnostumaan jostakin pelistä toisen sijaan. Koen, että tulokset ovat siis varsin luotettavia.

Tekoälyn käytöstä pelituotannossa riittää tutkittavaa. Tutkimusta voisi jatkaa esimerkiksi tutustumalla tekoälyn luonnollisten kielten käsittelyyn tarkemmin. Luonnollisten kielten käsittelyä käytetään usein peleissä, joissa tekoälyn kanssa käydään dialogia. Tekoälyä käytetään myös runsaasti erilaisten animaatioiden luomiseen, joka on osa-alue, johon tutkimusta voitaisiin jatkaa.

7.2 Opinnäytetyö ja oma oppiminen

Opinnäytetyötä kirjoittaessa huomasin haasteita etenkin siinä, että jokaisen faktan yhteyteen tuli lähdemerkintä. Varsinkin opinnäytetyön alussa kirjoitin tekstiä ja vasta jälkikäteen muistin mennä tarkistamaan mistä lähteestä tieto olikaan peräisin. Työn edetessä kuitenkin lähteiden lisäämisestä tuli helpompaa.

Työ opetti itseäni kirjoittamaan lähteisiin perustuvaa tekstiä, sillä en juuri ole joutunut sitä harjoitteluun. Työ antaa tulevaisuutta varten hyvät työkalut vastaavanlaisen tekstin kirjoittamiseen. Aiheena tekoäly pelituotannossa oli tuttu lähinnä omien pelikokemuksieni pohjalta, joten sain aiheesta myös syvempää osaamista, jolle on varmasti jatkossa käyttöä.

Lähteet

- Akhil, G. 2022. Why Unreal Engine is preferred for creating AAA games. Luettavissa <https://timesofindia.indiatimes.com/business/india-business/why-unreal-engine-is-preferred-for-creating-aaa-games/articleshow/89163555.cms> Luettu 13.9.2023
- Arora, S. K. 2023. Unity vs Unreal: Which Game Engine Should You Choose? Luettavissa <https://hackr.io/blog/unity-vs-unreal-engine> Luettu 1.11.2023
- Bycer, J. 2015. Procedural vs. Randomly Generated Content in Game Design. Luettavissa <https://www.gamedeveloper.com/design/procedural-vs-randomly-generated-content-in-game-design> Luettu 5.10.2023
- Cardoza, C. 2019. Clever-Commit coding assistant uses AI to protect against bugs. Luettavissa <https://sdtimes.com/ai/clever-commit-coding-assistant-uses-ai-to-protect-against-bugs/> Luettu 1.11.2023
- Chamary, J. 2018. Why 'Pokémon GO' Is The World's Most Important Game. Luettavissa <https://www.forbes.com/sites/jvchamary/2018/02/10/pokemon-go-science-health-benefits/> Luettu 13.9.2023 Luettu 25.9.2023
- Coursera. 2023. Decision Trees in Machine Learning: Two Types (+ Examples). Luettavissa <https://www.coursera.org/articles/decision-tree-machine-learning> Luettu 6.11.2023
- Cser, T. 2023. AI Testing Revolution: The Implications of ML in Software Testing. Luettavissa <https://www.functionize.com/automated-testing/ai-testing-revolution-the-implications-of-ml-in-software-testing> Luettu 1.11.2023
- Fall guys*. s.a. Luettavissa <https://www.fallguys.com/en-US> Luettu 25.9.2023
- Fintelics. 2023. AI in Game Difficulty Adjustment: Adapting Challenges to Player Skill Levels. Luettavissa <https://fintelics.medium.com/ai-in-game-difficulty-adjustment-adapting-challenges-to-player-skill-levels-b7f7767c96b> Luettu 5.10.2023
- Frąckiewicz, M. 2023. Adaptive Difficulty: How AI is Changing the Way We Play Video Games. Luettavissa <https://ts2.space/en/adaptive-difficulty-how-ai-is-changing-the-way-we-play-video-games/> Luettu 5.10.2023

- Frąckiewicz, M. 2023. Ethical Implications of AI-Enhanced Video Games: Reality, Representation, and Responsibility. Luettavissa <https://ts2.space/en/ethical-implications-of-ai-enhanced-video-games-reality-representation-and-responsibility/> Luettu 1.11.2023
- Golson, J. 2014. In Forza Horizon 2, Computers Finally Drive as Crazy as Humans. Luettavissa <https://www.wired.com/2014/09/forza-horizon-2-drivatars/> Luettu 5.10.2023
- Good, O. S. 2014. It's impossible to visit every planet in No Man's Sky. Luettavissa <https://www.polygon.com/2014/8/19/6045933/its-impossible-to-visit-every-planet-in-no-mans-sky> Luettu 25.9.2023
- Gopalan, B. 2020. Is Decision Tree a classification or regression model? Luettavissa <https://www.numpyninja.com/post/is-decision-tree-a-classification-or-regression-model> Luettu 6.11.2023
- Hamilton, T. 2023. What is Test Script? Luettavissa <https://www.guru99.com/test-script.html> Luettu 1.11.2023
- Ijarotimi, T. 2023. Gaming Intelligence: How AI is revolutionizing game development. Luettavissa <https://interestingengineering.com/innovation/gaming-intelligence-how-ai-is-revolutionizing-game-development> Luettu 17.9.2023
- Iqbal, M. 2023. Pokémon Go Revenue and Usage Statistics. Luettavissa <https://www.businessofapps.com/data/pokemon-go-statistics/> Luettu 25.9.2023
- Javapoint. s.a. Decision Tree Classification Algorithm. Luettavissa <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm> Luettu 6.11.2023
- Kärkkäinen, T., & Hänninen, J. 2023. Tekoäly ei tarvitse syväoppimista - yksinkertaisemmat mallit tuottavat parempia tuloksia. Luettavissa <https://www.jyu.fi/fi/ajankohtaista/arkisto/2023/07/tekoalymallit-toimivat-myos-ilman-syvaoppimistamalleja-yksinkertaisemmat-mallit-ovat-eettisempia-ja-saastavat-resursseja> Luettu 6.11.2023
- Kenny. 2021. Procedural Generation: An Overview. Luettavissa <https://kentpawson123.medium.com/procedural-generation-an-overview-1b054a0f8d41> Luettu 5.10.2023

- Khare, Y. 2023. How AI Is Revolutionizing Game Testing in 2023. Luettavissa <https://www.analyticsvidhya.com/blog/2023/04/how-ai-is-revolutionizing-game-testing-in-2023/> Luettu 1.11.2023
- Kollar, P. 2014. Best of the rest of 2014: The Binding of Isaac: Rebirth may be my most-played game in 2015. Luettavissa <https://www.polygon.com/2014/12/27/7455023/2014-year-in-review-the-binding-of-isaac-rebirth> Luettu 25.9.2023
- Gamersyde. s.a. *Left 4 Dead*. Luettavissa https://www.gamersyde.com/video_left_4_dead_2_gameplay_part_2-13447_en.html Luettu 25.9.2023
- Mascellino, A. 2023. We Need To Address The Ethics of AI in Video Games. Luettavissa <https://www.techopedia.com/ethics-of-ai-in-video-games> Luettu 1.11.2023
- Mosobalaje, M. A. 2023. The Future of Gaming with New AI-Powered Anti-Cheats. Luettavissa <https://hackernoon.com/the-future-of-gaming-with-new-ai-powered-anti-cheats> Luettu 1.11.2023
- Moss, E. s.a. What Is Unreal Engine? Luettavissa <https://www.bairesdev.com/blog/what-is-unreal-engine/> Luettu 13.9.2023
- Nelson, D. 2022. Mikä on NLP (Natural Language Processing)? Luettavissa <https://www.unite.ai/fi/mik%C3%A4-on-luonnollisen-kielen-k%C3%A4sittely/> Luettu 6.11.2023
- Nieminen, K. 2023. Mitä on koneoppiminen? Luettavissa <https://markkinointitrendit.fi/koneoppiminen/> Luettu 6.11.2023
- Obedkov, E. 2023. Dev used GPT-4 to make AI bug fixing tool named after Marvel's Wolverine. Luettavissa <https://gameworldobserver.com/2023/04/14/gpt-4-bug-fixing-ai-wolverine-python> Luettu 1.11.2023
- Press, R. 2016. The algorithms of No Man's Sky. Luettavissa <https://www.rambus.com/blogs/the-algorithms-of-no-mans-sky-2/> Luettu 25.9.2023
- Roach, J. 2021. What is Fortnite? Luettavissa <https://www.digitaltrends.com/gaming/what-is-fortnite/> Luettu 25.9.2023
- Shewale, R. 2023. Fortnite Statistics In 2023. Luettavissa <https://www.demandsage.com/fortnite-statistics/> Luettu 25.9.2023

- Simpson, C. 2014. Behavior trees for AI: How they work. Luettavissa
<https://www.gamedeveloper.com/programming/behavior-trees-for-ai-how-they-work> Luettu
6.11.2023
- Thompson, T. 2021. How Forza's Drivatar Actually Works. Luettavissa
<https://www.gamedeveloper.com/design/how-forza-s-drivatar-actually-works> Luettu
5.10.2023
- Tristem, B. s.a. Unity vs. Unreal: Which Game Engine is Best For You? Luettavissa
<https://blog.udemy.com/unity-vs-unreal-which-game-engine-is-best-for-you/> Luettu
1.11.2023
- Unity*. s.a. Luettavissa www.unity.com Luettu 13.9.2023
- Unreal Engine*. s.a. Luettavissa www.unrealengine.com Luettu 13.9.2023
- Varikuti, M. 2022. How Exactly Does Minecraft World Generation Work? Luettavissa
<https://medium.com/mlearning-ai/how-exactly-does-minecraft-world-generation-work-e50c07693c9a> Luettu 5.10.2023
- Warner, N. 2022. Fall Guys raced past 50 million players two weeks after going free-to-play.
Luettavissa <https://www.destructoid.com/fall-guys-raced-past-50-million-players-two-weeks-after-free-to-play-release/> Luettu 25.9.2023
- Wicklin, R. s.a. How to choose a seed for generating random numbers in SAS. Luettavissa
<https://blogs.sas.com/content/iml/2017/06/01/choose-seed-random-number.html> Luettu
6.11.2023
- Zenva. 2023. What is Unity? – A Top Game Engine Solution for Video Games. Luettavissa
https://gamedevacademy.org/what-is-unity/#Key_Features Luettu 13.9.2023