

Otto Ville Nykänen

**Koneoppimiseen perustuva proseduraalinen generaatio
peliympäristöjen tuotannossa**

Tieto- ja ohjelmistotekniikan kandidaatintutkielma

30. toukokuuta 2024

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Otto Ville Nykänen

Yhteystiedot: `otvinyka@student.jyu.fi`

Työn nimi: Koneoppimiseen perustuva proseduraalinen generaatio peliympäristöjen tuotannossa

Title in English: Using PCGML in creation of game environments

Työ: Kandidaatintutkielma

Sivumäärä: 24+0

Tiivistelmä: Tämä tutkimus käsittelee koneoppimiseen perustuvan proseduraalisen generaation (PCGML) hyödyntämistä peliympäristöjen tuotannossa. PCGML on proseduraalisen generaation muoto, jossa sisältöä tuotetaan hyödyntämällä koneoppimismenetelmiä. Tutkimuksessa analysoidaan PCGML:ään liittyvää viimeaikaista tutkimusta kirjallisuuskartoituksen muodossa. Tutkimuksessa selvitetään miten PCGML:ää on hyödynnetty peliympäristöjen tuotannossa, millaisia hyötyjä PCGML tarjoaa ja millaisia haasteita PCGML:n viimeaikaisessa tutkimuksessa on löytynyt.

Avainsanat: proseduraalinen generaatio, koneoppiminen, koneoppimiseen perustuva proseduraalinen generaatio, pelit

Abstract: This research explores the use of Procedural Content Generation via Machine Learning (PCGML) in creation of game environments. PCGML is a method of Procedural Content Generation (PCG) where content is generated using machine learning methods. Literature review is used as the main research method to survey recent academic research related to PCGML. This research attempts to find out how PCGML has been applied to generation of game environments, what benefits PCGML has been found to have and how researchers have addressed the challenges related to PCGML.

Keywords: procedural content generation, PCG, machine learning, procedural content generation via machine learning, PCGML, games

Sisällys

1	JOHDANTO	1
2	PELIYMPÄRISTÖT JA PROSEDURAALINEN GENERAATIO.....	3
3	PCG:N JA PCGML:N HYÖDYT JA HAASTEET	6
	3.1 PCGML:n tarpeet ja hyödyt peleissä	7
	3.2 PCGML:n haasteet	7
4	PCGML:N TUTKIMUS.....	9
	4.1 Koneoppimisen vaatiman datan puute.....	9
	4.2 Laatuvaatimukset ja ohjattavuus.....	11
	4.3 Suuret kielimallit	12
	4.4 PCGML:ään perustuvat suunnittelutyökalut	14
	4.5 Tutkimuksen aukot	15
5	YHTEENVETO.....	17
	LÄHTEET	18

1 Johdanto

Automaattisella sisällöntuotannolla (engl. *content generation*) on monia käyttötarkoituksia pelien tuotannossa. Peleistä puhuttaessa sisällöllä viitataan pelin tasoihin, sääntöihin tai pelimekaniikkoihin, tekstuureihin, tarinaan, esineisiin, musiikkiin, hahmoihin, ym. (Shaker, Togelius ja Nelson 2016). Sisällön tuottaminen automaattisesti voi helpottaa pelien tuotantoa vähentämällä pelin tekemiseen kuluvaan aikaan ja työpanosta, mutta automaattista sisällöntuotantoa voidaan myös hyödyntää esimerkiksi pelin uudelleenpelattavuuden lisäämiseen tekeillä jokaisesta pelikerrasta erilainen.

Sisällön tuottamiseen on tyypillisesti käytetty algoritmipohjaisia ratkaisuja, proseduraalista generaatiota (engl. *procedural content generation*, PCG) (Shaker, Togelius ja Nelson 2016, luku 1.1), mutta tekoälyn ja koneoppimisen nousussa suosioon datan tuotannossa viime vuosina koneoppimiseen pohjautuva proseduraalinen generaatio (engl. *procedural generation via machine learning*, PCGML) on kiinnostava tutkimuksen kohde pelien tuotannossa (A. Summerville ym. 2018).

Tämän tutkimuksen tarkoituksena on kartoittaa koneoppimiseen pohjautuvasta proseduraalisesta generaatiosta tehtyä tutkimusta ja selvittää miten PCGML:ää on käytetty ja voitaisiin käyttää osana pelimaailmojen, -ympäristöjen ja -tasojen tuotantoa. Tässä tutkimuksessa pelillä tarkoitetaan kokonaisuutta, jossa yksi tai useampi toimija, pelaaja, on vuorovaikutuksessa pelin kanssa joidenkin sääntöjen puitteissa. Tuloksia voi myös olla mahdollista yleistää simuloitujen tietokone mallien, kuten tulvasimulaatioiden ympäristöjen tuotantoon. Tutkimus keskittyy pääasiassa digitaalisiin peleihin. On myös olemassa muita pelejä, joissa voitaisiin käyttää proseduraalisesti luotuja ympäristöjä, kuten esimerkiksi pöytäroolipeli jossa pelaajat seikkailevat proseduraalisesti luodussa maailmassa.

Tutkimusmenetelmänä toimii kirjallisuuskartoitus. Kirjallisuuskartoituksen tarkoitus on kartoittaa PCGML:n mahdollisuuksia peliympäristöjen tuotannossa. Lähteinä käytetään proseduraalista generaatiota ja koneoppimistoimista proseduraalista generaatiota käsitteleviä artikkeleita ja aiheeseen liittyviä oppikirjoja. Luvussa 2 selitetään tutkimuksessa käytettyjä käsitteitä, kuten peliympäristön käsite ja kerrotaan proseduraalisen generaation ja koneop-

pimiseen perustuvan proseduraalisen generaation taustasta. Luvussa 3 kerrotaan klassisen PCG:n hyödyistä ja haasteista peliympäristöjen tuotannossa verrattuna PCGML:n hyötyihin ja haasteisiin. Luvussa 4 analysoidaan, miten PCGML:ää on hyödynnetty peliympäristöjen tuotannossa, miten hyödyt ovat ilmenneet, miten liittyvä tutkimus on ratkaissut edellä mainittuja ongelmia ja mitä aukkoja tutkimuksesta löytyy.

2 Peliympäristöt ja proseduraalinen generaatio

Pelimaailmalla tai peliympäristöllä ei ole tarkkarajaista määritelmää, mutta termeillä voidaan tarkoittaa interaktiivista laajempaa ympäristöä, joka koostuu esimerkiksi maastosta, kasvillisuudesta, rakennuksista, vihollisista, hahmoista, vaaroista (engl. *hazards*) ja pulmista (engl. *puzzles*). Pelimaailmasta tai -ympäristöstä puhutaan usein avoimempien seikkailu- ja roolipelien yhteydessä verrattuna tasoihin, joista puhutaan yleensä esimerkiksi tasohyppelypelien yhteydessä. Joskus voidaan puhua myös tyrmistä tai luolastoista (engl. *dungeons*). Jotta tasoja, tyrmii ja kolmiulotteisia pelimaailmoja ei tarvitsisi mainita erikseen jokaisessa asiayhteydessä, tässä tutkimuksessa näistä kaikista käytetään yleisnimityksenä termiä *ympäristö* tai *peliympäristö*. Joissain tapauksissa voi silti olla aiheellista tarkentaa mitä peliympäristön alatyyppejä tarkoitetaan.

Tietotekniikassa proseduraalisella generaatiolla (engl. *procedural generation*, usein myös *proseduraalinen sisällöntuotanto*, *procedural content generation*, PCG) tarkoitetaan datan tuottamista hyödyntämällä algoritmeja. Proseduraalisella generaatiolla on sovelluksia esimerkiksi simulaatiossa ja tietokonegrafiikassa, mutta eräs tutkituimmista ja yleisimmistä proseduraalisen generaation sovelluksista on proseduraalisen generaation hyödyntäminen peleissä. Peleistä puhuttaessa PCG voidaan määritellä pelisisällön algoritmisenä tuotantona ilman suoraa ihmisen syötettä (Shaker, Togelius ja Nelson 2016, luku 1.1).

Proseduraalisen generaation sovelluksien analyysin helpottamiseksi voi olla hyödyllistä kategorisoida PCG:n lähestymistapoja. Shaker, Togelius ja Nelson (2016, luku 1.6) esittelevät metriikoita, joita voidaan käyttää peleissä käytettyjen proseduraalisen generaation sovelluksien luokitteluun:

- *Online/Offline*: Tässä termillä *online* tarkoitetaan sitä että sisältöä luodaan proseduraalisesti sovelluksen ajon aikana verrattuna *offline*-lähestymistapaan, jossa proseduraalisesti tuotettu sisältö on generoitu ennalta, esim. pelin kehitysvaiheessa.
- *Välttämättömyys/Valinnaisuus*: Proseduraalisesti tuotettu sisältö voi olla sovelluksen toiminnan kannalta välttämätöntä sisältöä (esim. pelitason perusrakenne) tai valinnaista sisältöä kuten sivutehtävien palkintoja.

- *Ohjattavuus*: Proseduraalisen generaation ohjattavuuteen vaikuttaa miten paljon ja helposti käyttäjä tai kehittäjä voi vaikuttaa sovelluksen tuottamaan tulosteeseen. Sovellusta voidaan ohjata esimerkiksi mukautettavilla parametreilla tai näennäissatunnaisgeneraattorin lähtöarvolla (*seed*).
- *Geneerisyys/Adaptiivisuus*: Peleistä puhuttaessa adaptiivinen proseduraalinen generaattori ottaa pelaajan toiminnan huomioon tulosteen tuotannossa verrattuna geneeriseen proseduraaliseen generaatioon, jossa pelaajan toiminta ei vaikuta tulosteeseen.
- *Deterministisyys/Stokastisuus*: Deterministinen PCG tuottaa aina saman tulosteen samoilla lähtöarvoilla verrattuna stokastiseen generaatioon, jonka tulostetta voi olla vaikea ennustaa.
- *Konstruktiivinen lähestymistapa/Generoi-ja-testaa-lähestymistapa*: Generoi-ja-testaa-lähestymistavassa tulostetta testataan ja uudelleengeneroidaan haluttujen laatuksien täyttymisen varmistamiseksi verrattuna konstruktiiviseen metodiin, jossa tuloste luodaan kerran.
- *Automaattinen generaatio/Mixed authorship*: Sisältöä voidaan tuottaa täysin automaattisesti tai mixed-authorship -menetelmällä jossa pelaajan tai kehittäjän sisältöä yhdistetään PCG:n luomaan sisältöön.

Tekoälyyn ja koneoppimiseen liittyvä tutkimus on ollut jatkuvassa kasvussa. Tekoäly on laaja käsite, jolla selitetään menetelmiä, joissa kone oppii, tekee päätöksiä ja ratkoo ongelmia ihmisten toiminnan kaltaisesti. Koneoppiminen on tekoälyn alakäsite, joka kattaa järjestelmät, joissa kone oppii sille syötetystä datasta. Koneoppimismalleja voidaan kategorisoida niiden koulutustavan mukaan kolmeen kategoriaan: valvottuun oppimiseen perustuviin malleihin (engl. *supervised learning*), valvomattomaan oppimiseen perustuviin malleihin (engl. *unsupervised learning*) ja vahvistusoppiviin malleihin (engl. *reinforcement learning, RL*). (Hiran ym. 2021, luku 1.)

Valvottuun oppimiseen perustuvissa malleissa malli koulutetaan hyödyntämällä valmiiksi kategorisoitua dataa, minkä jälkeen malli pystyy kategorisoimaan sille syötettyä tuntematonta dataa. Valvomattomaan oppimiseen perustuvissa malleissa mallille syötetään joukko kategorisoimatonta dataa, josta malli löytää piileviä samankaltaisuuksia, joiden perusteella se klusteroi dataa (*clustering*) ja löytää klustereiden välisiä yhteyksiä (*association*). RL-

malleissa malli oppii kokemuksesta: malli käyttää hyväksi jotain automatisoitua toimijaa, jonka onnistunutta toiminta-askelta palkitaan ja epäonnistunutta askelta rangaistaan. (Hiran ym. 2021, luku 1.)

Kun proseduraalisessa generaatiossa hyödynnetään koneoppimismenetelmiä, puhutaan koneoppimiseen pohjautuvasta proseduraalisesta generaatiosta (engl. *procedural generation via machine learning*, PCGML). PCGML voidaan määritellä pelisisällön tuotantona olemassa olevan pelisisällön avulla koulutettujen koneoppimismallien tuottamana sisältönä (A. Summerville ym. 2018), tosin määritelmää saattaa olla syytä laajentaa myös PCG:n tekoälypohjaiseen laadunvalvontaan tai pelisisältöön, joka tuotetaan koneoppimismalleilla peleihin liittymättömän koulutusmateriaalin pohjalta. Koneoppimiseen perustuvan proseduraalisen generaation toteutuksessa voidaan käyttää useita erilaisia koneoppimismenetelmiä.

Peliympäristöjen datan representaatiolla ei ole vakiintunutta toteutustapaa. Peliympäristöjen data voidaan esittää useissa eri muodoissa joihin lukeutuvat esimerkiksi jonot (engl. *sequence*), ruudukot (engl. *grid*), graafit (engl. *graph*) (A. Summerville ym. 2018) sekä moniulotteiset matriisit ja muut erilaiset useita ulottuvuuksia sisältävät tietorakenteet. Jonossa ympäristön piirteet esitetään lineaarisesti peräkkäin, esimerkiksi 2-ulotteinen taso voitaisiin esittää jonona jakamalla se jonoksi pystysuoria suikaleita. Koska moni kaksiulotteinen peli koostuu ruudukkoon asetelluista kuvahahmoista (engl. *sprite*), ruudukko soveltuu hyvin kaksiulotteisten ympäristöjen esittämiseen, mutta ruudukkoa voidaan käyttää myös kolmannessa ulottuvuudessa (esim. kokeuskartta). Graafilla voidaan esittää ympäristön piirteitä (engl. *attributes* tai *features*), mikä tekee siitä hyödyllisen useiden erilaisten ympäristöjen esittämiseen toisessa ja kolmannessa ulottuvuudessa. Joissain tapauksissa dataa voidaan esittää monitasoisena kokonaisuutena (esim. pelin taso, jossa perusgeometria esitetään ruudukkona, johon on lisätty graafilla esitettyjä piirteitä).

3 PCG:n ja PCGML:n hyödyt ja haasteet

Peleissä käytetyn proseduraalisen sisällöntuotannon kehityksen tarve syntyi aikaisten kotitietokoneiden tallennustilan puutteesta (Shaker, Togelius ja Nelson 2016, luku 1.3). Proseduraalisen sisällöntuotannon käyttö pelien kehityksessä ja siihen liittyvä tutkimus ovat lisääntyneet ja PCG:tä käytetään nykyään pelimekaniikkana pelien uudelleenpelattavuuden lisäämiseen, tuotantokulujen vähentämiseen ja tallennustilavaatimusten vähentämiseen (A. Summerville ym. 2018). Shaker, Togelius ja Nelson (2016, luku 1.2) mainitsevat myös pelien kehityskulujen ja -ajan minimoimisen erääksi tärkeimmistä PCG:n hyödyistä, sekä lisäävät, että PCG:n avulla voidaan toteuttaa pelejä, joissa proseduraalisen generaation avulla pelaajan kokemusta voidaan räätälöidä pelaajan mieltymysten mukaiseksi.

Proseduraalisen generaation hyödyntämiseen liittyy myös haasteita. Peliympäristöjen tuotannossa on huomionarvoista, että koska peliympäristö on osa pelin funktionaalista kokonaisuutta, peliympäristöllä on lähes aina pelin pelattavuuteen vaikuttavia laatukriteerejä. Esimerkiksi voi tapahtua niin, että proseduraalisesti luodulla tyrmällä ei ole sisäänkäyntiä tai uloskäyntiä (esimerkki lainattu: Shaker, Togelius ja Nelson 2016, luku 1.5) tai Perlinin kohinakartalla luodussa komiulotteisessa ympäristössä on liian jyrkkiä rinteitä, joita pelaaja ei voi kiivetä löytääkseen pelin kannalta tärkeitä esineitä. Tällöin luodut peliympäristöt ovat laadultaan kelvottomia. Kriteerien toteuttamiseksi peleissä käytetään usein haku- ja ratkaisualgoritmeihin perustuvia proseduraalisen generaation menetelmiä, jolloin kehittäjät joutuvat lisäämään PCG-mallien tuottaman datan laatukriteerit manuaalisesti (A. Summerville ym. 2018). Oikeellisen datan luotettavuuden lisäksi haasteena voi olla PCG-järjestelmän nopeus, monipuolisuus ja luovuus (Shaker, Togelius ja Nelson 2016, luku 1.5).

Seuraavissa alaluvuissa esitellään, millaisia hyötyjä ja haasteita koneoppimiseen pohjautuvalla proseduraalisella generaatiolla on ja miten PCGML voisi ratkaista perinteisen PCG:n ongelmia.

3.1 PCGML:n tarpeet ja hyödyt peleissä

Koneoppimiseen perustuva proseduraalinen generaatio jakaa suurimman osan hyödyistään perinteisten proseduraalisen generaation menetelmien kanssa erityisesti autonomisessa sisällöntuotannossa ja sovelluksen tallennustilan tarpeen pienentämisessä. Tosin PCGML tarjoaa uusia hyötyjä, esimerkiksi yleiskäyttöisempiä koneoppimismalleja, jotka pelin kehittäjä voi opettaa joukolla halutunlaista dataa sen sijaan, että halutun ulostulon laatukriteerit joudutaan ohjelmoimaan käytettyyn malliin (A. Summerville ym. 2018). PCGML-metodien on teoretisoitu pystyvän oppimaan laatukriteereihin vaikuttavia tekijöitä sille syötetystä datasta, jolloin tuotettu data voi olla myös parempilaatuista perinteisellä proseduraalisella generaatiolla tuotettuun dataan verrattuna. Tällöin PCGML voisi mahdollistaa konstruktiivisen lähestymistavan käyttämisen hitaamman iteratiivisen generoi-ja-testaa-lähestymistavan sijaan.

Koneoppimiseen perustuvilla malleilla on perinteisestä PCG:stä poiketen kyky tunnistaa, analysoida ja kritisoida pelisisältöä. Tunnistavilla, analysoivilla ja kritisovilla menetelmillä voidaan esimerkiksi korjata ja parantaa hakupohjaisen PCG:n tulostetta tai ihmisen luomaa sisältöä (A. Summerville ym. 2018). PCGML on hyvin soveltuva *mixed-initiative*-sovelluksiin, joissa ihminen ja tietokonesovellus toimivat yhdessä sisällön tuottamiseksi (A. Summerville ym. 2018) edellä mainittujen ominaisuuksiensa ansiosta. *Mixed-initiative*-sisällöntuotantomallissa (*Mixed initiative for content generation*) jokin ihmis- tai tekoälyosallinen voi aloittaa sisällöntuotantoprosessin, minkä jälkeen kaikki osalliset osallistuvat iteratiivisesti tulosteen kehittämiseen tietoisena järjestelmän tilasta tai historiasta (Lai, Leymarie ja Latham 2022).

3.2 PCGML:n haasteet

Kun proseduraalista generaatiota käytetään peliympäristöjen tuotannossa, eräs suurimmista ongelmista liittyy koneoppimismallien koulutus- ja syötedatan puutteellisuuteen ja peliympäristöjen datan vaihtelevaan representaatioon. Koneoppimisen tutkimuksessa vallitsee yleinen käsitys siitä, että datan suurella määrällä on enemmän merkitystä datan laatuun tai käytettyyn koneoppimismenetelmään verrattuna (A. Summerville ym. 2018). Summervillen ym. (2018) mukaan yleisesti käytettyjen tekstiä ja kuvia tuottavien koneoppimismallien koulu-

tuksessa käytettyyn teksti- ja kuvadataan verrattuna pelien ympäristöihin liittyvää dataa on suhteellisen vähän ja se voi olla vaikeammin hyödynnettävissä, esimerkiksi julkisen datan puutteen vuoksi. Toinen ongelma liittyen datan hyödynnettävyyteen on pelien ympäristöjen datan representaatio. Saman peligenren tai usein myös saman pelisarjan eri julkaisut jakavat datan tallennustapansa tai semantiikkansa vain harvoin (A. Summerville ym. 2018) ja eri pelien välisen datan hyödyntäminen voi olla haasteellista pelien toiminnallisen luonteen vuoksi. Etenkin pelien sääntöjen ja ympäristöjen interaktiivisten mekaniikkojen erilaisuus vaikeuttaa pelien välisen datan hyödyntämistä. Koulutusdatan määrän lisäksi koulutusdatan representaatio (esim. ruudukko vs. graafi), voi vaihdella ja eri tavoilla esitettyä dataa voi olla vaikeaa hyödyntää yhdessä.

Usein proseduraaliselta sisällöntuotannolta halutaan sitä, että luotu tuloste on tarpeeksi monipuolista, luovaa ja uskottavaa (Shaker, Togelius ja Nelson 2016, luku 1.5). Monipuolisen tulosteen tuottamista vaikeuttaa joissain tapauksissa koneoppimiseen liittyvä ylisovittumiseksi (engl. *overfitting*) kutsuttu ongelma, jossa malli saattaa tuottaa lähes täysin syötedatan kaltaista dataa (Sudhakaran ym. 2023; Halina ja Guzdial 2023).

Ohjattavuus on eräs proseduraalisen generaation suotuisista ominaisuuksista (Shaker, Togelius ja Nelson 2016, luku 1.5), mikä luonnollisesti pätee myös PCGML:ään. Koneoppimiseen perustuvien järjestelmien tuloste perustuu pääasiassa koneoppimismallin koulutusdataan mutta on ymmärrettävää, että suunnittelija saattaa haluta kontrolloida tulosteen ominaisuuksia (A. Summerville ym. 2018). Käytetty koneoppimismalli voi vaikuttaa tulosteen ohjattavuuteen. Ohjattavuuden toteuttamiseen voi vaikuttaa koneoppimisen yleisesti liittyvä *mustan laatikon* ongelma. Monet koneoppimismallit ovat niin sanottuja mustia laatikoita, joiden toiminnan selittäminen on ihmisille vaikeaa (Rudin 2019). Mustiksi laatikoiksi määritellään koneoppimismallit, jotka ovat joko liian monimutkaisia ihmisen selitettäväksi (esim. rekursiiviset syväoppimismallit), tai kaupalliset yksinoikeutetut suljetun lähdekoodin mallit (Rudin 2019). Mustan laatikon ongelma korostuu erityisesti lääketieteessä käytetyn koneoppimisen kaltaisissa kriittisissä käyttötarkoituksissa, mutta ongelma on relevantti myös muilla aloilla, kuten pelien tuotannossa käytetyssä PCGML:ssä. Selitettävä koneoppimismalli voi olla paremmin ohjattavissa ja sen toimintaa on helpompaa analysoida mustaan laatikkoon verrattuna.

4 PCGML:n tutkimus

Tässä luvussa tarkastellaan peliympäristöjen tuotantoon käytetyn koneoppimiseen perustuvaan proseduraaliseen generaatioon liittyvää viimeaikaista tutkimusta. PCGML:ää ei ole tutkimuksen määrän perusteella vielä sovellettu yhtä aktiivisesti peliympäristöjen tuotantoon verrattuna perinteisiin proseduraalisen generaation menetelmiin. Aiheen akateemisen tutkimuksen analyysillä voidaan selvittää millaisien peliympäristöjen tuotantoon PCGML:ää on jo käytetty, miten PCGML:n hyödyt ilmenevät käytännössä, millaiset koneoppimismenetelmät soveltuvat PCGML:ään ja miten PCGML:n haasteita on yritetty ratkaista. Kirjallisuuskartoituksen tarkoituksena on myös löytää mahdollisia aukkoja tutkimuksen kattavuudesta.

4.1 Koneoppimisen vaatiman datan puute

Koneoppimismallien koulutusdatan puutteen ongelmalle on kehitetty useita mahdollisia ratkaisuja. Eräs lähestymistapa ongelman ratkaisemiseksi on peliympäristöihin liittyvän datan kokoaminen julkiseksi kokoelmaksi. Tutkijoiden A. J. Summerville ym. (2016) kokoama *The Video Game Level Corpus*, (VGLC) on eräs esimerkki tällaisesta kokoelmasta. *The Video Game Level Corpus* sisältää 428 tasoa 12 pelistä, joiden datan representointiin on käytetty ruudukko-, graafi- ja vektoripohjaisia esitystapoja. Vastaavat peliympäristöjen datan kokoelmat voisivat toimia hyvin PCGML:ssä käytettyjen koneoppimismallien koulutusdatana mutta lisää työtä on vielä tehtävänä kokoelmien kasvattamiseksi; A. Summerville ym. (2018) kritisoivat VGLC:n kokoa ja datan häviöllisyyttä (esimerkiksi *Super Mario Bros.*-pelin kaikki eri viholliset on esitetty samoina *enemy*-nimellä esiintyvinä dataobjekteina).

Peliympäristöjen datan keräämistä helposti hyödynnettävien, jono, ruudukko ja graafipohjaisten, datan representaatioiden sijaan toissijaisista lähteistä, kuten internetistä löytyvistä pelivideoista, on myös kokeiltu. Dataa voidaan muuttaa koneoppimismalleille hyödynnettävään muotoon käyttämällä esimerkiksi konenäköratkaisuja, kuten avoimen lähdekoodin konenäkökirjasto OpenCV:tä ("About - OpenCV", n.d.).

Guzdial ja Riedl (2021) käyttivät OpenCV:tä datan keräämiseen internettiin ladatuista "Let's Play"- ja "Long Play"-videoista. Konenäöllä tasoista kerätty data koottiin ruudukkopohjai-

sesti listaamalla jokaisen tasossa esiintyneen kuvahahmon sijainti jokaista videon yksittäistä ruutua (engl. *frame*) kohden ja samankaltaisen rakenteen jakavat ruudut yhdistettiin tason segmenteiksi. Koneäöllä pelivideosta voitiin myös tallentaa pelaajan käyttämä aika tason tietyissä vaiheissa, jota tutkijoiden kehittämä malli pystyi hyödyntämään segmenttien kategorisointiin tasogeneraattorissa käytettyä mallia varten. Se, että pelaajaan toiminta voidaan erotella videolta, tekee videosta hankitusta datasta mielenkiintoisen tutkimuskohteen. Guzdialin ja Riedlin (2021) kehittämä malli ei ole välttämättä helposti sovellettavissa useisiin peleihin, koska malli voi silloin vaatia muutoksia koodiin ja datan hankkimisessa käytettyyn kuvahahmokokoelmaan (engl. *spritesheet*) (Mirgati ja Guzdial 2023). Mirgatin ja Guzdialin (2023) kehittämä variaationaalisen autoenkooderin (engl. *Variational Autoencoder, VAE*) ja generatiivisen vastaverkon (engl. *Generative Adversial Network, GAN*) yhdistävä VAE-GAN-malli tähtää tämän ongelman ratkaisemiseen. Mallin variaationaalinen autoenkooderi ja generatiivinen vastaverkko voidaan kouluttaa joukolla koneäöllä eroteltua dataa, minkä jälkeen mallin variaationaalinen autoenkooderi pystyy muuttamaan malliin syötettyä videomuotoista dataa mallille entuudestaan tuntemattomasta pelistä mallin generatiiviselle vastaverkolle soveltuvaksi piileväksi tilaksi, jonka klustereiden avulla malli voi tuottaa syötetyn videon kaltaisia tasoja.

Jotkin peliympäristöjen tuotantoon käytetyt menetelmät pystyvät toimimaan pienemällä määrällä koulutus- tai syötedataa tai joissain tapauksissa jopa ilman sitä. Tällaisia malleja ovat esimerkiksi vahvistusoppivat RL-mallit. Proseduraaliseen sisällöntuotantoon käytetyssä vahvistusoppimisessa (*PCGRL*) vahvistusoppimiseen perustuva koneoppiva tekoälytoimija tekee iteratiivisia muutoksia luotavaan sisältöön. Jokaisen iteraation jälkeen muuttunut sisältö arvioidaan joitain kriteerejä arvioivalla funktiolla, jossa arviointifunktion arviota parantavat muutokset palkitsevat- ja arviota huonontavat muutokset rankaisevat RL-toimijaa ohjaten toimijaa tekemään parempia muutoksia jokaisessa iteraatiossa (Khalifa ym. 2020).

Khalifa ym. (2020) kokeilivat tutkimuksessaan RL-sisällöntuotantomallia kolmen eri ylhäältä kuvatun kaksikulotteisen ruudukkopohjaisen ympäristön tuottamiseen: kaksikulotteisiin labyrinthteihin, *The Legend of Zelda*-tyylisen pelin tyrmiin ja *Sokoban*-laatikontyöntöpulmapelin pulmiin. Tutkimuksessa RL-mallin palkitsemisjärjestelmä palkitsi RL-toimijaa, jos toimijan tekemät muutokset paransivat tason läpäisemiseen liittyviä kriteerejä, mikä ohjasi mallia

tuottamaan laadukkaampia ympäristöjä. Khalifan ym. (2020) esittelemän RL-mallin hyödyntämistä vaikeuttaa se, että mallin tekemät muutokset ja tulosten arviointifunktio joudutaan räätälöimään halutun tulosten mukaan. Bontrager ja Togelius (2021) esittelevät *Generative Playing Networks*-mallin (GPN), jossa asetellaan vastakkain pelaava RL-toimija ja generatiivinen RL-toimija. Pelaava RL-toimija pyrkii määrittelemään ne pelitilanteet, jotka ovat sadan prosentin varmuudella voitettavissa, kun taas generatiivinen RL-toimija pyrkii muokkaamaan ympäristöä niin, että pelaavan RL-toimijan arvio on mahdottoman ja täysin varmasti läpäistävässä olevan pelitilanteen välillä. Kun pelaava toimija kehittyä generatiivisen toimijan generoimien ympäristöjen läpäisemisessä, generatiivinen toimija joutuu luomaan vaikeampia ympäristöjä. Malli ei tarvitse syötteenä peliympäristöjen dataa, koska mallin tulee ainoastaan tietää millaisia muutoksia ympäristöön voidaan tehdä ja mitä toimintoja pelaaja voi suorittaa.

4.2 Laatuvaatimukset ja ohjattavuus

Laadukkaan datan tuottaminen automatisoiduilla metodeilla voi olla haastavaa. Lähes kaikkien luotujen peliympäristöjen tärkeimpänä laatukriteerinä voidaan pitää pelattavuutta pelin sääntöjen sallimissa rajoissa. Tulosteelle halutaan usein myös asettaa muita vaatimuksia, joista yleisin lienee syötedataan verrattava monipuolisuus ja mahdollisimman vähäinen plagiointi (esim.: Sudhakaran ym. 2023; Halina ja Guzdial 2023; Shu, Liu ja Yannakakis 2021, ym.). Halutut laatukriteerit vaihtelevat usein luodun ympäristön tyyppin mukaan. Esimerkiksi tuotetuilla tasohyppelypelin tasoilla tai toimintaroolipelien tyrmillä voidaan haluta olevan tietty vaikeustaso tai fysiikkapohjaisen pulmapelin tasojen alkutilanteen halutaan olevan vakaa. Myös subjektiivisempia laatukriteereitä kuten ympäristössä pelatessa koettua hauskuutta tai turhautumista voidaan haluta mitata ja ohjata (Guzdial ja Riedl 2021).

Laatuvaatimuksien evaluointiin käytetään usein automatisoituja kelpoisuusfunktioihin (engl. *fitness function*) perustuvia menetelmiä. Pelattavuuden arvioinnissa kelpoisuusfunktiona käytetään useimmiten reitinetsintämenetelmiä kuten A*-toimijoita (esim.: Sudhakaran ym. 2023; Sarkar ja Cooper 2021; Mirgati ja Guzdial 2023, ym.), joiden rajoituksiksi on ohjelmoitu pelin säännöt ja pelaajan mahdolliset toiminnot. Pelaajilla teetätetty tuotetun sisällön laadun arviointi voi olla erittäin arvokasta mallien laadunvalvonnassa (Shaker, Togelius ja Nelson

2016, luku 12.4), josta esimerkkinä Guzdial ja Riedl (2021) toteuttivat käyttäjätutkimuksen, jossa tutkijoiden tasogeneraattorin tuottamia *Super Mario Bros.*-tasoja verrattiin aikaisempiin vastaavan kaltaisten tasogeneraattoreiden tuottamiin tasoihin ja *Super Mario Bros.*-pelin alkuperäisiin tasoihin kriteereiden, kuten esimerkiksi *Mario-kaltaisuuden*, *hauskuuden* ja *turhautuniesuuden* kautta (tarkempi tutkimusasetelma ks. Guzdial ja Riedl 2021).

PCGML-mallien ohjattavuuteen on tyypillisesti vaikutettu integroimalla malliin korkean tason suunnitteluparametreja, esimerkiksi tyrmän huoneiden määrä, vihollisten määrä, tason pisimmän hypyn pituus, ym. Parametrien toteutumisen tarkistamiseen joudutaan kuitenkin usein käyttämään haku- ja ratkaisijapohjaisia menetelmiä (A. Summerville ym. 2018), kuten mallien evaluoinnissa. Koneoppimismallin koulutukseen vaikuttavia parametreja on kokeiltu, kuten Gisslénin ym. (2021) toteuttamassa tutkimuksessa käytetty vahvistusoppivan mallin palkitsemisfunktioon liitetty vaikeustasoparametri, joka vaikuttaa koulutuksessa käytetyn reitinetsintätoimijan suoriutumiseen. On tosin epäsuotuisaa, että haku- tai ratkaisijajärjestelmien kelpoisuusfunktiot ja vahvistusoppimisen palkitsemisfunktiot joudutaan ohjelmoimaan malliin kehittäjien toimesta. Valvomattomaan oppimiseen suunniteltujen autoenkooderien ja generatiivisten vastaverkkojen hyödyntämistä ympäristöjen tuotannossa on tutkittu (Thakkar ym. 2019; Mirgati ja Guzdial 2023; Abraham ja Stephenson 2023), mutta valvomattoman oppimisen kykyä löytää laatuvaatimukseen vaikuttavia piileviä muuttujia ja tulosten ohjaamista laatuvaatimukset täyttävillä esimerkkipeliympäristöillä ei ole vielä laajalti tutkittu.

4.3 Suuret kielimallit

Suurten kielimallien, valvomattomasti tai osittain valvotusti suurella määrällä tekstimuotoista dataa koulutettujen koneoppimismallien, ja muiden suurilla datamäärillä koulutettujen koneoppimis- ja tekoälyjärjestelmien hyödyntäminen pelisisällön tuotannossa on myös tutkimuksen kohteena. Suurilla kielimalleilla (engl. *Large Language Model*, *LLM*) on emergentejä ominaisuuksia, mikä tarkoittaa sitä, että tarpeeksi laajan skaalan suuret kielimallit pystyvät suorittamaan tehtäviä, joita niitä ei ole suoranaisesti koulutettu suorittamaan (Abdullah ym. 2024).

Sudhakaran ym. (2023) tutkivat generatiivisten ennakkokoulutettujen muuntajamallien (engl.

Generative Pre-trained Transformer, GPT) hyödyntämistä *Super Mario*-tasojen tuotannossa (*MarioGPT*). Tutkimuksessa käytettiin GPT-2 mallista muokattua avointa DistilGPT-mallia jonka ennakkokoulutuksessa käytetyn yleispätevän tekstidatan lisäksi mallin koulutuksessa käytettiin 37 *Super Mario Bros.* ja *Super Mario Bros.: the Lost Levels* tason viidenkymmenen ruudun levyisten osien merkkijonoiksi muunnettua dataa, johon on myös merkitty pelaajan kulkema reitti. Mallin koulutuksessa alkuperäisten tasojen osista koottuja neljän osan tasoja mutatoitiin evolutionäärisellä menetelmällä koulutusdatan lisäämiseksi. Tutkimuksessa mallia kokeiltiin kouluttaa myös pelkällä tasojen datalla ilman DistilGPT:n ennakkokoulutusta ja huomattiin että yleismuotoinen ennakkokoulutusdata paransi mallin tulostetta tutkijoiden asettamilla laatukriteereillä.

Sudhakaran ym. (2023) kokeilivat myös MarioGPT-mallin ohjattavuuden lisäämistä sisäistämällä malliin kehoitteita (engl. *prompt*) käsittelevän mallin, joka tunnistaa syötetystä kehoitteesta avainsanojen (esim. *blocks, enemies*) ja määrää kuvaavien suureiden (esim. *some, many*) pareja, jotka muunnetaan GPT-2 mallin tulosteeseen vaikuttavaksi ominaisuusvektoriksi. Sudhakaran ym. (2023) laskivat ominaisuuksien esiintyvyyksiä tuotetuissa tasoissa verrattuna käytettyyn kehoitteeseen ja huomasivat, että kehoite vaikuttaa avainsanojen kuvaamien ominaisuuksien ilmenemiseen tulosteessa, mutta malli ei aina noudata kehoitetta.

Mallille syötetyllä kehoitteella ohjattavien yleiskäyttöisten suurten kielimallien emergenttejä ominaisuuksia voi olla mahdollista hyödyntää peliympäristöjen tuotannossa ilman tehtävään erikoistunutta koulutusta. Abdullah ym. (2024) tutkivat *Angry Birds*-pelin tutkimuskäyttöön suunnatun kloonin *Science Birds* tasoissa esiintyvien rakennelmien luomista kilpailun muodossa. Kilpailijoiden tehtävä oli hyödyntää erilaisia kehoitesuunnittelun (engl. *prompt engineering*) menetelmiä luodakseen kehoitteen, jolla ChatGPT:n GPT-3.5, GPT-3.5F, GPT-4 ja GPT-4F mallit luovat vakaita englannin kielen aakkoston kirjaimien muotoisia rakennelmia. Jotta tuloste voitaisiin muuttaa rakennelmiksi, rakennelmien tuli esiintyä tulosteessa *Science Birds*-tasojen palikoiden sijaintia ja tyyppiä kuvaavina koodifunktioina. Kilpailijoiden tuottamien 16 hyväksytyyn kehoitteen tuottamia rakennelmia arvioitiin rakennelman vakauden ja halutun kirjaimen samankaltaisuutta arvioivien algoritmien avulla. Kaksi parasta kehoitetta tuottivat vakaita rakennelmia yli 97 prosentin todennäköisyydellä, mikä kertoo että emergentti LLM voi suoriutua monimutkaisista proseduraalisen generaation tehtävistä.

4.4 PCGML:ään perustuvat suunnittelutyökalut

Automatisoidun sisällöntuotannon lisäksi PCGML:n metodeja voidaan hyödyntää peliympäristöjen suunnittelutyökaluissa hyödyntämällä mixed-initiative-malleja. MI-työkaluja voidaan esimerkiksi hyödyntää ympäristöissä esiintyvien aukkojen tai ei-haluttujen osien täydentämiseen ja ympäristöjen laajentamiseen hyödyntämällä täydentämismenetelmiä (engl. *inpainting*) (Jara Gonzalez ja Guzdial 2023), työkalut voivat kritisoida ja analysoida suunnittelijan tekemiä muutoksia (A. Summerville ym. 2018) ja ehdottaa korjauksia tai työkalu voi luoda valmiita pohjia kehittäjän suunnitteluprosessin alkupisteeksi.

Guzdial, Liao ja Riedl (2018) suorittivat käyttäjätutkimuksen, jossa 84 osallistujalle annettiin tehtäväksi luoda kaksi *Super Mario Bros.*-tasoa MI-tasontuotantomallien tutkimuskäyttöön suunnatulla *MoraiMaker*-tasoeditorilla (Guzdial ym. 2021), ollessa vuorovaikutuksessa molemmilla tasoluontikerroilla eri koneoppimismenetelmään perustuvan MI-toimijan kanssa. Guzdial, Liao ja Riedl (2018) kokeilivat tutkimuksessa kolmea koneoppimismenetelmää, Markov-ketjua, Bayes-verkkoa ja LSTM-mallia (*Long Short-Term Memory recurrent neural network*), mutta päätyivät johtopäätökseen, että mallien tuloste ei vastaa hyvin käyttäjien visiota ja, että käytetyllä mallilla oli vain vähän eroja käyttäjien suhtautumiseen. Tutkimuksen johtopäätöksenä todettiin, että autonomiseen sisällöntuotantoon suunnitellut koneoppimismallit eivät pysty oppimaan käyttäjän visiota tarpeeksi nopeasti, minkä vuoksi MI-sisällöntuotantoon käytetyt mallit tulisi kouluttaa erityisesti MI-tehtäviä varten.

Analysoivaa ja kritisoivaa MI-mallia on kokeiltu esimerkiksi tutkijoiden Migkotzidis ja Liapis (2022) kehittämässä *SuSketch* työkalussa, jolla voidaan kehittää *Team Fortress II*-ammuntapelin kaltaisten ammuntapelienvä ympäristöjä mixed-initiative-mallisesti. *SuSketch*-työkalu käyttää koneoppimisalgoritmia näyttääkseen käyttäjälle tietoja esimerkiksi ympäristön pelattavuudesta sekä ennustettuja tilastoja esimerkiksi pelierän kestosta ja missä osassa karttaa suurin osa eliminaatioista tapahtuu. Malli voi myös tarjota ehdotuksia pelin tehosteiden, kuten ammuslaatikoiden ja vahinkotehosteiden sijainneille ympäristön tasapainottamiseksi reiluksi molemmille pelin tiimeille. Tutkijoiden Migkotzidis ja Liapis (2022) ennakkoon koneilla pelisuunnittelijoilla teetättämä käyttäjätutkimus antoi osviittaa siitä, että MI-mallin tarjoamat ennusteet ja ehdotukset olivat käyttäjille hyödyllisiä. Kehitettäväksi jäi työkalun kokeileminen kokenemattomampien käyttäjien keskuudessa sekä ennusteiden mahdollinen

epätarkkuus johtuen siitä, että kolmiulotteisen ensimmäisen persoonan ammuntopelin esittäminen työkalussa käytetyssä kaksiulotteisessa ruudukossa ei välttämättä vastaa todellisten pelitilanteiden dynamiikkaa.

4.5 Tutkimuksen aukot

Huomattava määrä peliympäristöjen tuotantoon liittyvästä PCGML:n tutkimuksesta keskittyy nykyisellään kaksiulotteisiin tasoihin, kuten esimerkiksi *Super Mario Bros.* pelin tasoihin (esim.: Sudhakaran ym. 2023; Guzdial ja Riedl 2021; Jara Gonzalez ja Guzdial 2023; Guzdial, Liao ja Riedl 2018, ym.), mikä selittynee aikaisemman tutkimuksen määrällä, helposti hyödynnettävällä koulutusdatalla (esim.: VGLC A. J. Summerville ym. 2016) ja tutkimustyökalujen, kuten valmiiden reitinetsintätoimijoiden saatavuudella. Tutkimusta on myös tehty muiden kaksiulotteisten tasohyppelypelien tasojen tuotannossa (esimerkiksi *Lode Runner* (Thakkar ym. 2019) ja *Kid Icarus* (Sarkar ja Cooper 2021)), kaksiulotteisten toimintaroolipelien tyrmien tuotannossa (esimerkiksi *The Legend of Zelda* (Bontrager ja Togelius 2021; Khalifa ym. 2020, ym.) ja kaksiulotteisten pulmapelien tasojen ja pulmien tuotannossa (esimerkiksi *Sokoban* (Khalifa ym. 2020) ja *Angry Birds* (Abraham ja Stephenson 2023)).

Kolmiulotteisten ympäristöjen tuotannossa PCGML-metodeja on tutkittu esimerkiksi kolmiulotteisen maaston (Tang ja Sweetser 2023), kolmiulotteisten tasohyppelypelien tasojen (Gisslén ym. 2021) ja ajopelien kilparatojen tuotannossa (Gisslén ym. 2021). Voidaan huomata, että tutkimus on painottunut tietyn tyyppisten ympäristöjen ympärille ja, että PCGML:n hyödyntämisessä kolmiulotteisten ympäristöjen tuotannossa on vielä tutkittavaa. Myös monikäyttöiset useiden eri tyyppisten ympäristöjen tuotantoon helposti sovellettavat mallit, kuten Bontragerin ja Togeliuksen (2021) esittelemä RL-malli, ovat mielenkiintoinen tutkimuskohde.

Mallien tulosten ohjattavuudessa on myös vielä tutkittavaa. Nykyisten mallien tulostetta ohjataan vielä usein generoi-ja-testaa menetelmällä hyödyntämällä tulosteelle haku- ja ratkaisija-algoritmeja (ks. 4.2), mutta tuotannon nopeuden ja laadun tehostamiseksi voisi olla hyödyllistä kehittää konstruktiivinen malli, joka pystyisi noudattamaan kehittäjän määrittelemiä laatuvaatimuksia jokaisella tulosten generointikerralla. Mallin sisäiseen tilaan vai-

kuttavat parametrit (Gisslén ym. 2021) ja kehotekäsittelijät (Sudhakaran ym. 2023) voisivat tarjota paremman ohjattavuuden ja vaativat lisää tutkimusta. Toinen lähestymistapa olisi tutkia pystyvätkö valvomattomaan oppimiseen perustuvat koneoppimismenetelmät löytämään laatukriteerit täyttävästä esimerkkitiedosta kriteereihin vaikuttavia piileviä ominaisuuksia ja sitä kautta parantamaan tulosteensa laatua. Myös suurten kielimallien hyödyntäminen (Abdullah ym. 2024) proseduraalisessa generaatiossa ja kielimallien ohjaaminen kehotteella on mielenkiintoinen tutkimuskohde.

5 Yhteenveto

Tässä tutkimuksessa tarkasteltiin koneoppimiseen perustuvaa proseduraalista generaatiota peliympäristöjen tuotannossa. Tutkimuksessa selvitettiin mitä on koneoppimiseen perustuva proseduraalinen generaatio ja pohdittiin PCGML:n hyötyjä ja haasteita peliympäristöjen tuotannossa. PCGML:n hyötyjä ja haasteita myös verrattiin perinteisten proseduraalisen generaation vastaaviin. Tutkimuksen kirjallisuuskartoituksessa tarkasteltiin, miten tuore PCGML-menetelmiä tutkiva tutkimus on ratkaissut tunnettuja ongelmia ja mitä hyötyjä PCGML:stä on havaittu.

Löydösten perusteella koneoppimismenetelmiä on mahdollista hyödyntää proseduraalisen sisällöntuotannon tehtävissä, ja PCGML-menetelmät pystyvät tuottamaan pelattavia ympäristöjä. Koneoppimismenetelmien kyvystä analysoida ja kritisoida sisältöä on myös lupaavaa näyttöä minkä ansiosta PCGML:ää voidaan hyödyntää mixed-initiative suunnittelutyökaluissa. Tutkittavaa on vielä PCGML:n soveltamisessa useiden erilaisten ympäristöjen tuotannossa, koneoppimisen vaatiman datan puutteen ratkaisemisessa ja PCGML-mallien ohjattavuudessa. On myös epäselvää, miten paljon koneoppimista on hyödynnetty peliteollisuudessa akateemisen tutkimuksen ulkopuolella.

Lähteet

Abdullah, Febri, Pittawat Taveekitworachai, Mury F. Dewantoro, Ruck Thawonmas, Julian Togelius ja Jochen Renz. 2024. “The 1st ChatGPT4PCG Competition”. *IEEE Transactions on Games*, 1–17. <https://doi.org/10.1109/TG.2024.3376429>.

“About - OpenCV”. n.d. Viitattu 7. toukokuuta 2024. <https://opencv.org/about/>.

Abraham, Frederic ja Matthew Stephenson. 2023. “Utilizing Generative Adversarial Networks for Stable Structure Generation in Angry Birds”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 19, numero 1 (lokakuu): 2–12. <https://doi.org/10.1609/aiide.v19i1.27496>.

Bontrager, Philip ja Julian Togelius. 2021. “Learning to Generate Levels From Nothing”. Teoksessa *2021 IEEE Conference on Games (CoG)*, 1–8. <https://doi.org/10.1109/CoG52621.2021.9619131>.

Gisslén, Linus, Andy Eakins, Camilo Gordillo, Joakim Bergdahl ja Konrad Tollmar. 2021. “Adversarial Reinforcement Learning for Procedural Content Generation”. Teoksessa *2021 IEEE Conference on Games (CoG)*, 1–8. <https://doi.org/10.1109/CoG52621.2021.9619053>.

Guzdial, Matthew, Jonathan Chen, Shao-Yu Chen ja Mark Riedl. 2021. “A General Level Design Editor for Co-Creative Level Design”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 13, numero 2 (kesäkuu): 81–83. <https://doi.org/10.1609/aiide.v13i2.12980>.

Guzdial, Matthew, Nicholas Liao ja Mark Riedl. 2018. *Co-Creative Level Design via Machine Learning*. arXiv: 1809.09420 [cs.AI].

Guzdial, Matthew ja Mark Riedl. 2021. “Game Level Generation from Gameplay Videos”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 12, numero 1 (kesäkuu): 44–50. <https://doi.org/10.1609/aiide.v12i1.12861>.

Halina, Emily ja Matthew Guzdial. 2023. “Tree-Based Reconstructive Partitioning: A Novel Low-Data Level Generation Approach”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 19, numero 1 (lokakuu): 244–254. <https://doi.org/10.1609/aiide.v19i1.27520>.

Hiran, Kamal Kant, Ritesh Kumar Jain, Kamlesh Lakhwani ja Ruchi Doshi. 2021. *Machine Learning: Master Supervised and Unsupervised Learning Algorithms with Real Examples (English Edition)*. BPB Publications.

Jara Gonzalez, Johor ja Matthew Guzdial. 2023. “Reconstructing Existing Levels through Level Inpainting”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 19, numero 1 (lokakuu): 276–283. <https://doi.org/10.1609/aiide.v19i1.27523>.

Khalifa, Ahmed, Philip Bontrager, Sam Earle ja Julian Togelius. 2020. “PCGRL: Procedural Content Generation via Reinforcement Learning”. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 16, numero 1 (lokakuu): 95–101. <https://doi.org/10.1609/aiide.v16i1.7416>.

Lai, Gorm, Frederic Fol Leymarie ja William Latham. 2022. “On Mixed-Initiative Content Creation for Video Games”. *IEEE Transactions on Games* 14 (4): 543–557. <https://doi.org/10.1109/TG.2022.3176215>.

Migkotzidis, Panagiotis ja Antonios Liapis. 2022. “SuSketch: Surrogate Models of Gameplay as a Design Assistant”. *IEEE Transactions on Games* 14 (2): 273–283. <https://doi.org/10.1109/TG.2021.3068360>.

Mirgati, Negar ja Matthew Guzdial. 2023. “Joint Level Generation and Translation Using Gameplay Videos”. Teoksessa *2023 IEEE Conference on Games (CoG)*, 1–10. <https://doi.org/10.1109/CoG57401.2023.10333193>.

Rudin, Cynthia. 2019. *Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead*. arXiv: 1811.10154 [stat.ML].

- Sarkar, Anurag ja Seth Cooper. 2021. “Generating and Blending Game Levels via Quality-Diversity in the Latent Space of a Variational Autoencoder”. Teoksessa *Proceedings of the 16th International Conference on the Foundations of Digital Games*. FDG '21. Montreal, QC, Canada: Association for Computing Machinery. ISBN: 9781450384223. <https://doi.org/10.1145/3472538.3472545>.
- Shaker, Noor, Julian Togelius ja Mark J. Nelson. 2016. *Procedural Content Generation in Games*. 237. Computational Synthesis and Creative Systems. Cham: Springer International Publishing. <http://dx.doi.org/10.1007/978-3-319-42716-4>.
- Shu, Tianye, Jialin Liu ja Georgios N. Yannakakis. 2021. “Experience-Driven PCG via Reinforcement Learning: A Super Mario Bros Study”. Teoksessa *2021 IEEE Conference on Games (CoG)*, 1–9. <https://doi.org/10.1109/CoG52621.2021.9619124>.
- Sudhakaran, Shyam, Miguel González-Duque, Matthias Freiberger, Claire Glanois, Elias Najarro ja Sebastian Risi. 2023. “MarioGPT: Open-Ended Text2Level Generation through Large Language Models”. Teoksessa *Advances in Neural Information Processing Systems*, toimittanut A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt ja S. Levine, 36:54213–54227. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2023/file/a9bbeb2858dfbdbd4c19814e5d80ec60-Paper-Conference.pdf.
- Summerville, Adam, Sam Snodgrass, Matthew Guzdial, Christoffer Holmgård, Amy K. Hoover, Aaron Isaksen, Andy Nealen ja Julian Togelius. 2018. “Procedural Content Generation via Machine Learning (PCGML)”. *IEEE Transactions on Games* 10 (3): 257–270. <https://doi.org/10.1109/TG.2018.2846639>.
- Summerville, Adam James, Sam Snodgrass, Michael Mateas ja Santiago Ontañón. 2016. *The VGLC: The Video Game Level Corpus*. arXiv: 1606.07487 [cs.LG].
- Tang, Runze ja Penny Sweetser. 2023. “Multilayer Map Generation Using Attribute Loss Functions”. Teoksessa *Proceedings of the 18th International Conference on the Foundations of Digital Games*. FDG '23. , Lisbon, Portugal, Association for Computing Machinery. ISBN: 9781450398558. <https://doi.org/10.1145/3582437.3587175>.

Thakkar, Sarjak, Changxing Cao, Lifan Wang, Tae Jong Choi ja Julian Togelius. 2019. “Autoencoder and Evolutionary Algorithm for Level Generation in Lode Runner”. Teoksessa *2019 IEEE Conference on Games (CoG)*, 1–4. <https://doi.org/10.1109/CIG.2019.8848076>.