

Riku Lehkonen

**Proseduraalinen sisällön tuottaminen videopelien
tasosuunnittelussa**

Tietotekniikan kandidaatintutkielma

31. toukokuuta 2020

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Riku Lehtonen

Yhteystiedot: riku.p.lehtonen@student.jyu.fi

Ohjaaja: Timo Tiihonen

Työn nimi: Proseduraalinen sisällön tuottaminen videopelien tasosuunnittelussa

Title in English: Procedural content generation in video game level design

Työ: Kandidaatintutkielma

Opintosuunta: Tietotekniikka

Sivumäärä: 26+0

Tiivistelmä: Proseduraalisen sisällön tuottaminen on yleistynyt peliteollisuudessa aikaisempiin vuosiin nähden, mutta sitä käytetään yhä usein ainoastaan pelien uudelleenpelattavuuden kehittämiseen, sen sijaan että sitä sovellettaisiin uusiin peli- ja tasosuunnittelun osa-alueisiin. Tekniikan laajempi soveltaminen ja ymmärtäminen auttaisi pelisuunnittelijoita tuottamaan pelitasoissa uudenlaisia pelikokemuksia, ilman laadun heikkenemistä. Tämän takia tässä tutkimuksessa selvitettiin kirjallisuuskatsauksella, mitä mahdollisuuksia kyseisellä tekniikalla on olemassa ja erityisesti, miten proseduraalista sisällön tuottamista käytetään videopelien tasosuunnittelussa. Kirjallisuuskatsauksen tulosten pohjalta vahvistettiin, että proseduraalisella sisällön tuottamisella on enemmän tutkittua potentiaalia, joka olisi peliteollisuuden hyödynnettävissä kuin mitä käytetään.

Avainsanat: Proseduraalinen sisällön tuottaminen, tasosuunnittelu, videopeli, pelisuunnittelu, pelisuunnitteluteoria, MDA viitekehys

Abstract: In this thesis a literature review has been used to find out how procedural content generation is used in the level design of video games and what possibilities exist with that technology. The aim of the thesis is to better game designers' understanding of procedural content generation as well as further outline its possibilities in video game level design, due to its increased use in the gaming industry. Based on the findings garnered from the literature review, it is known that the gaming industry is not using procedural content generation to the

fullest of its researched potential.

Keywords: Procedural Content Generation, level design, video game, game design, game design theory, MDA framework

Termiluettelo

PCG	Proseduraalinen sisällön tuottaminen (procedural content generation). PCG on algoritmista pelin sisällön tuottamista käyttäjän rajallisen tai epäsuoran syötteen ohjaamana.
Roguelike	Roguelike on peligenre, jonka nimi on peräisin Rogue (A.I. Design 1980) -pelistä. Genressä pelaajan pelikokemus pohjautuu pelin uudelleenpelaamiseen. Pelien pelitasot ovat usein proseduraalisesti generoituja jokaiselle uudelle pelikerralle.
Endless runner	Endless runner -pelit ovat tasohyppelypelien alagenre, jossa pelaajahahmo liikkuu jatkuvasti eteenpäin proseduraalisesti kehitetyssä pelitasossa, joka on teoreettisesti loppumaton.
FPS	FPS -pelit ovat ensimmäisen persoonan ammuntapelejä, joissa aseiden käyttö on keskeinen toimintatapa.
RTS	RTS -pelit ovat reaaliaikaisia strategiapelejä, joissa pelaajat toimivat samanaikaisesti, vuorojen sijaan.

Sisältö

1	JOHDANTO	1
2	PROSEDURAALINEN SISÄLLÖN TUOTTAMINEN	2
2.1	Proseduraalisesti tuotettu sisältö	2
2.2	Proseduraaliset sisällön tuottamistavat	3
2.3	Proseduraalisen sisällön tuottamisen lähestymistavat	4
3	PROSEDURAALINEN SISÄLLÖN TUOTTAMINEN PELISUUNNITTELUSSA	8
3.1	Mekaniikkojen proseduraalinen tuottaminen	8
3.2	Sisältöjen dynamiikka ja proseduraalinen tuottaminen	9
3.3	Sisältöjen estetiikka ja proseduraalinen tuottaminen	11
4	VIDEOPELIEN TASOSUUNNITTELU	12
4.1	Proseduraalisuuden hyödyntäminen tasosuunnittelussa	14
4.2	Proseduraalisuuden uudet mahdollisuudet tasosuunnittelussa	15
5	YHTEENVETO	17
	LÄHTEET	18

1 Johdanto

Pelien pelattavuus ja pelikokemuksien määrä ovat olleet peliteollisuuden huomion kohteena jo vuosien ajan. Viime aikoihin saakka pelikoneiden suorituskyky ja peliohjelmistot ovat rajoittaneet pelisuunnittelijoiden mahdollisuuksia tuottaa pelejä. Näiden rajoitteiden poistuttua pelikehityksessä on voitu ottaa käyttöön uusia menetelmiä pelitasojen suunnittelun ja pelien uudelleenpelattavuuden parantamiseen. (Smelik ym. 2014). Viime vuosina on julkaistu enemmän suuria avoimen maailman (open world) pelejä, sillä pelisuunnittelijat ovat voineet hyödyntää enemmän proseduraalista sisällön tuottamista (procedural content generation, PCG) pelitasojensa määrässä ja koossa, kuten vuonna 2016 julkaistussa No Man's Sky (Hello Games 2016) -pelissä (Risi ja Togelius 2019). Proseduraalista sisällön tuottamista on hyödynnetty peliteollisuudessa jo vuonna 1980 julkaistussa Rogue (A.I. Design 1980) -pelissä, jossa tuotettiin proseduraalisesti pelitasoja pienemmässä mittakaavassa. Pelisuunnittelijat ovat kuitenkin usein rajoittaneet proseduraalisesti tuotetun sisällön ainoastaan pelin uudelleenpelattavuuteen, sen sijaan että yrittäisivät laajentaa sen käyttöä uusiin peli- ja tasosuunnittelun osa-alueisiin (Smith 2014a).

Kirjallisuuskartoituksessa on tarkoitus määrittää mitä PCG on, miten sillä voidaan tuottaa tasosuunnittelussa sisältöä, ja miten suunnittelijat ymmärtävät sen. Tavoitteena on lisätä pelisuunnittelijoiden ymmärrystä PCG:n mahdollisuuksista tasosuunnittelussa ja muihinkin tavoitteisiin kuin uudelleenpelattavuuteen.

Kirjallisuuskartoituksen PCG:n keskeisyyden vuoksi toisessa luvussa määritellään mitä PCG on, minkälaisia nämä menetelmät ovat, mitä sisältöä niiden avulla on mahdollista tuottaa ja minkälaisia lähestymistapoja sisällön tuottamiselle on olemassa. Kolmannessa luvussa määritellään proseduraalinen sisällön tuottaminen pelisuunnittelussa, ja miten se vaikuttaa Hunicken ym. (2004) määrittelemään kolmeen osa-alueeseen: mekaniikkaan, dynamiikkaan ja estetiikkaan. Neljännessä luvussa tarkastellaan tasosuunnittelua, miten proseduraalista sisällön tuottamista on hyödynnetty sen osa-alueissa, ja mitä mahdollisuuksia tasosuunnittelun kannalta PCG:llä on olemassa.

2 Proseduraalinen sisällön tuottaminen

Proseduraalista sisällön tuottamista käytetään pelisuunnittelussa nykyään enemmän kuin aikaisemmin. (Smith 2014b). PCG:tä on tutkittu yleisemmin peliteollisuuden ulkopuolella ja ymmärrys siitä on kasvanut, mutta tämä ei ole täysin näkynyt pelisuunnittelussa. (Smith 2014a). Khaled ym. (2013) ehdottavat tämän johtuvan suunnittelijoiden vähäisestä ymmärryksestä PCG:n mahdollisuuksista, jonka vuoksi he kehittivät PCG:lle suunnittelumetaforia. Smithin (2014a) myöhemmän tutkimuksen mukaan suunnittelijat eivät kuitenkaan näytä käyttävän PCG:tä kovin monipuolisesti, joka tarkoittaa Khaled ym. ehdotuksen olevan puutteellinen.

Tämän vuoksi tässä kirjallisuuskartoituksessa tarkastellaan miten pelisuunnittelijat ymmärtävät PCG:n. Lähtökohdaksi valitaan alalla yleisesti hyväksytty Togeliuksen ym. määritelmä: ”PCG on algoritmista pelin sisällön tuottamista käyttäjän rajallisen tai epäsuoran syöteen ohjaamana” (Togelius, Kastbjerg ym. 2011, suomennos minun). Käytännössä kyseinen määritelmä tarkoittaa tietokoneohjelmaa, joka kykenee tuottamaan sisältöä itsenäisesti tai yhteistoimin yhden tai useamman käyttäjän kanssa.

2.1 Proseduraalisesti tuotettu sisältö

Proseduraalisen sisällön tuottamisen määritelmän keskeisin käsite on sisältö. Shaker ym. (2016) ovat rajanneet pelien sisältöä seuraaviin komponentteihin: ”pelitasot, maailmat, pelisäännöt, tekstuurit, tarinat, tavarat, tehtävät, musiikki, aseet, kulkuneuvot, hahmot, ym.” Tämä kartoitus tarkastelee erityisesti PCG:n hyödyntämistä pelitasojen suunnittelun näkökulmasta, mutta on tärkeää myös huomioida, miten pelitasoissa komponentit vaikuttavat prosessiin. Esimerkiksi pelihahmojen kyvyt saattavat vaikuttaa siihen, miten pelitasoissa voi liikkua, kuten kaksoishyppy tasohyppelypeleissä. Jos edellä mainitussa esimerkkipelissä on useita pelihahmoja, joilla on ominaislaatuisia kykyjä, ja joiden välillä pelaaja voi vaihtaa vain etukäteen määritellyissä paikoissa, pelisääntö hahmojen vaihdoksiin tulee vaikuttamaan se, miten kyseinen pelitaso suunnitellaan; onko pelitasolla useita etenemismenetelmiä hahmosta riippumatta vai pakotetaan pelaaja palaamaan takaisin vaihtamaan hahmo, jotta hän voi ede-

tä? Pelitasojen sisällölliset komponentit vaikuttavat tällöin myös pelitason kokonaisuuteen, jolloin nämä komponentit voidaan laskea osaksi pelitasojen suunnittelua.

Pelimoottorit ja NPC -hahmojen käytöksiens generointi eivät sisälly edellä mainittuun Shaker ym. määritelmään, sillä näiden elementtien generointi keskittyy tekoälyn ja laskennallisen älykkyyden hyödyntämiseen, PCG:n sijaan. PCG:n ja tekoälyn välinen raja on kuitenkin epämääräinen, sillä PCG voi esimerkiksi hyödyntää tekoälyä sisällön tuottamisessa (Summerville ym. 2018).

2.2 Proseduraaliset sisällön tuottamistavat

Smith ym. (2012) määrittävät suunnittelijan roolin muuttuvan PCG -pohjaisissa peleissä yksittäisten instanssien suunnittelusta, kokonaisten parametrisoitujen instanssien suunnitteluun. Tämän vuoksi suunnittelijoiden on tärkeä ymmärtää mikä algoritmi sopii projektiin, ja mitä esivaatimuksia, hyötyjä tai haittoja algoritmilla saattaa olla.

Videopelien suunnittelun kannalta PCG:n käyttämät algoritmiset tuottamistavat voidaan jakaa kahteen osa-alueeseen: suunnittelijoiden määrittämiin algoritmeihin, parametreihin, rajoitteisiin ja tavoitteisiin, tai koneoppimismenetelmiin, jotka käyttävät pelin sisältöä oppimismateriaalinaan uuden sisällön tuottamiseen. (Summerville ym. 2018).

Koneoppimisessa riittävän monimutkaisia algoritmeja kutsutaan usein älyllisiksi (Turing 1950), ja sen vuoksi algoritmeille on luotu määritteitä, kuten laskennallinen älykkyys (computational intelligence) ja tekoäly (artificial intelligence). PCG:llä tuotetun sisällön monimuotoisuus riippuu osittain sen tuottamisessa käytettävän algoritmin monimutkaisuudesta. Käyttäjät pystyvät huomaamaan yksinkertaisilla algoritmeilla tuotetun sisällön olevan generoitua, kun taas monimutkaisten algoritmien avulla tuotettu sisältö vaikuttaa suunnittelijan tuottamalta, Turingin testin mukaisesti. (Smith 2014b). Käyttäjän havaitsemaa monimutkaisuutta on myös mahdollista hallita sattumanvaraisuuden määrällä, mutta sen liiallinen käyttö voi tuottaa pelaamattoman pelin. (Shaker, Togelius ja Nelson 2016).

Algoritmin monimutkaisuus vaikuttaa myös sen käytettävyyteen ja pelitasojen suunnitteluun. Puurakennealgoritmeja on esimerkiksi mahdollista käyttää pelitasojen tuottamisessa

pelkästään rakennuslohkojen avulla (Shaker, Togelius ja Nelson 2016), kun taas generatiivinen kilpaileva verkosto, joka hyödyntää kahta kilpailevaa neuroverkostoa, tarvitsee paljon suuremman esimateriaalin sisällön tuottamiseen (Giacomello, Lanzi ja Loiacono 2018).

2.3 Proseduraalisen sisällön tuottamisen lähestymistavat

Proseduraaliselle sisällön tuottamiselle on olemassa useita lähestymistapoja ja luokitteluja. Togelius ym. (2011) ovat jakaneet PCG:n esimerkiksi rakentavaan ja ”generoi-ja-testaa”-lähestymistapaan. Toisaalta, Hendriks ym. (2013) ovat jakaneet PCG:tä esimerkiksi kuvan suodatukseen, kielioppiin perustuvaan generointiin, spatiaalisiin algoritmeihin, ym. Tässä työssä käytetään Smithin (2014b) pelisuunnittelijoille kohdentamia käsitteitä.

Smith (2014b) on jakanut proseduraalisen sisällön tuottamisen lähestymistavat viiteen eri osa-alueeseen: generoivaan optimointiin, rajoiteohjattuun generointiin, kielioppiin perustuvaan generointiin, generointiin sisällön valintana ja generointiin rakentavana prosessina. Näiden lähestymistapojen on tarkoitus kuvastaa millä tavoin on mahdollista generoida sisältöä peleissä. Lähestymistapojen määrittelyssä generoinnin tekniset osatekijät jätetään taustatekijöiksi ja keskitytään suunnittelun kannalta tärkeisiin osa-alueisiin.

Generoivassa optimoinnissa keskitytään sisällön generoinnissa haluttujen ominaisuuksien generointiin ja muotoiluun. Generoitua sisältöä arvioidaan joko laskennallisella järjestelmällä, tai suunnittelijat arvioivat sen tarkkuutta prosessin aikana tai lopussa. Generoidun sisällön määrästä riippuen laskennallinen arviointi saattaa olla työlästä, jolloin suunnittelijoilla on suurempi osuus arviointiprosessissa (Secretan ym. 2011). Inhimillisen osatekijän vuoksi kyseinen metodi tuottaa yksilöllisiä lopputuloksia, eli optimoinnin arvioija tulee vaikuttamaan siihen, minkälainen sisältö on generoinnin jälkeen. Koska suunnittelijat yleensä valvovat generoitavan sisällön laatua, generointi rajoittuu suunnitteluvaiheeseen eikä sovellu pelin aikana käytettäväksi. Smithin mukaan moni optimointigeneraattori hyödyntää evoluutioalgoritmeja, joita kutsutaan usein etsintä-pohjaisiksi PCG:ksi (search-based PCG).

Generointimenetelmää voidaan käyttää eri mittakaavoissa. Togelius ja Schmidhuber käyttivät evoluutiolaskentaa (evolutionary computation) pelisääntöjen ja agenttien logiikan generointiin tyhjästä pohjasta Tetriksen kaltaiselle pelille (Togelius ja Schmidhuber 2008). He

käyttivät sisältöjen laskennallisen arvioinnin pohjana Kosterin: theory of fun -kirjaa (Koster 2013). Kokeen oli tarkoitus todistaa kokonaisten pelisääntöjen generoinnin mahdollisuus ilman, että suunnittelija on osa evoluutioprosessia. Vaikka lopullinen järjestelmä Togeliuksen ja Schmidhuberin mukaan ei ollut ainutlaatuinen, kyseinen koe todistaa, että on mahdollista rakentaa toimivia pelisääntöjä tai pelitasoja hyvin pienestä aloitusjoukosta, jos evoluutiosäännöt ovat sopivat.

Rajoiteohjattu generointi (generation as constraint satisfaction) pohjautuu generoitavan sisällön ominaisuuksien ja rajoitusten tekniseen määrittelyyn. Suunnittelijat pystyvät tuottamaan metodilla sisältöä määritellyn tiedon perusteella, ilman tarvetta määrittellä hakualgoritmien toimintaa. Tämän vuoksi kyseinen metodi on laajasti käytössä suunnittelijoiden käyttämissä työkaluissa (Butler ym. 2013).

Metodin ongelmana on tarvittavien rajoitteiden tekninen määrittely. Koska generoitava sisältö perustuu haluttujen ominaisuuksien generointiin, täytyy olla mahdollista etukäteen luoda rajoitteet, jotka tuottaisivat näitä edellä mainittuja haluttuja ominaisuuksia. Kun rajoitteet ovat monimutkaisia ja toisistaan riippuvia, niiden virheenkorjaus on haastavaa.

Kielioppiin perustuva generointi (generation with grammars) sisältää määrittelevän kieliopin, jonka mukaisesti algoritmi tuottaa ja laajentaa sisältöä. Kyseinen metodi pohjautuu ihmisen luoman sisällön laajentamiseen koneoppimismenetelmin. Smithin (2014b) mukaan lopputuloksena on yhdistelmä suunnittelijan määrittämistä muuttujista ja niiden vuorovaikutuksista, ja tietokoneavusteisesta suunnittelutilan tutkinnasta (exploration of design space), laajentavan algoritmin kieliopin avulla. Metodien hyödyntäminen pelikokemuksen aikana on työlästä, jonka vuoksi se soveltuu offline käyttöön. (Dormans 2010)

Suunnittelijat pystyvät käyttämään kyseistä metodia laajentamaan tietokoneavusteisesti ihmisten tuottamia algoritmin kielioppeja (Müller ym. 2006). Laajentaminen ei kuitenkaan ole välttämättä optimointia, kuten aikaisemmassa generointimenetelmässä, vaan sitä voidaan käyttää pelkästään tuottamissääntönä. Metodien avulla on esimerkiksi mahdollista generoida lisää samankaltaista materiaalia ihmisen suunnitteleman pohjasta, jolloin tuottamissäännöt ovat rakennettu metodin sisälle.

Metodia on mahdollista käyttää pelikentässä sijaitsevien objektien generointiin. Suunnittelijat pystyvät etukäteen kehittämään halutun estetiikan ja tyylin objekteille ja tekstuureille, ja metodi rakentaa näiden tuotettujen materiaalien pohjalta samankaltaisia objekteja, jolloin kehitetty lopputulos säilyttää kokonaisvaltaisen tyylin.

Käytännön tasolla kielioppiin perustuva generointi toimii luonnollisen kieliopin tavoin ja voi haarautua tämän mukaisesti. Dormans (2010) on antanut seuraavanlaisen esimerkin mahdollisista kielioppisäännöistä ja niiden lopputuloksista:

1. Dungeon -> Obstacle + treasure
 2. Obstacle -> key + Obstacle + lock + Obstacle
 3. Obstacle -> monster + Obstacle
 4. Obstacle -> room
-
1. key + monster + room + lock + monster + room + treasure
 2. key + monster + key + room + lock + monster + room + lock + room + treasure
 3. room + treasure
 4. monster + monster + monster + monster + room + treasure

Generointi sisällön valintana (generation as content selection) on yksinkertaisin generointimetodi, ja sen vuoksi on tiedeyhteisö ei ole yksimielinen siitä, että voidaanko se laskea sisällön generointimenetelmäksi (Sullivan 2010). Koska kirjallisuuskartoituksessa PCG on määritelty Shakerin ym. (2016) mukaan, kyseinen generointimenetelmä lasketaan osaksi PCG metodeja. Generointi sisällön valintana eroaa yksinkertaisuutensa takia muista menetelmistä: Se on nopea, mutta toisaalta pelaajat pystyvät myös tunnistamaan helpommin sen ominaispiirteet, kun sillä tuotetaan sisältöä riittävän suuressa mittakaavassa.

Yksinkertaisuutensa ja nopeutensa takia generointi soveltuu Smithin mukaan luomaan sisältöä tehokkaasti pelin pelaamisen aikana, kuten endless runner -peleissä, joissa yleisesti käytetään proseduraalista sisällön tuottamista pelikenttien jatkamiseksi. Toisaalta, menetelmän käyttö suunnitteluvaiheessa antaa suunnittelijoille mahdollisuuden valita pelaamisessa käytettävää materiaalia parametrien perusteella. Tämän vuoksi pelisuunnittelijoiden työkuva muuttuu materiaalien rakentamisesta niiden valintaan ja editointiin.

Lähestymistapa soveltuu tasosuunnittelun kannalta erityisesti suurien avoimien maailmojen generointiin, sillä sen avulla on mahdollista luoda pohja, josta pelisuunnittelija pystyy editoimaan esteettisesti ja tyyllisesti sopivan (Khaled, Nelson ja Barr 2013). Tämänkaltainen hyödyntäminen vähentää pelisuunnittelijan käyttämää aikaa pelitason rakentamisessa, sillä hänen ei tarvitse manuaalisesti rakentaa koko pelimaailmaa, vaan pystyy keskittymään ainoastaan sen sisältämien objektien editointiin.

Viides generointimenetelmä on **generointi rakentavana prosessina** (generation as a constructive process). Lähestymistavassa sisältöä rakennetaan tapauskohtaisesti yhdistelemällä etukäteen suunniteltuja lohkoja. Metodien suunnittelulogiikka on rakennettu sen käyttämän algoritmin sisälle. Smithin (2014b) mukaan generointimenetelmä ei kuitenkaan ole rakennettu ymmärtämään omaa algoritmiaan, kuten esimerkiksi algoritmin kieliopin generointi saattaa olla rakennettu ymmärtämään, vaan se toimii täysin sille etukäteen rakennetun suunnittelulogiikan mukaisesti.

Generointimenetelmän suunnittelulogiikan vuoksi sen käyttö on usein tapauskohtaista; pelien ja pelikertojen väliset erot ovat räätälöidyissä lohkoissa, jolloin yhtä peliä varten suunniteltu generaattori ei toimi samalla tavalla toisen pelin kanssa. Koska generaattori ei ymmärrä omaa algoritmiaan, se ei myöskään pysty tekemään tarvittavia muutoksia rakentamisprosessissa tapausten välillä. Rogue -genre pelit käyttävät tätä metodia erityisesti, sillä pelikentät ja niissä esiintyvät viholliset voidaan rakentaa yhdistelemällä etukäteen räätälöityjä lohkoja, jonka seurauksena jokainen pelikerta on hieman erilainen.

3 Proseduraalinen sisällön tuottaminen pelisuunnittelussa

Hunicke ym. (2004) ovat jakaneet pelisuunnittelun kolmeen eri osa-alueeseen: mekaniikkaan, dynamiikkaan, ja estetiikkaan. Mekaniikalla tarkoitetaan pelin yksittäisiä komponentteja, esimerkiksi Tetris (AcademySoft 1984) -pelin palikat ja niiden liikkuvuus, rivien poistaminen sekä pisteytyksen erot, kun poistetaan useita rivejä. Dynamiikka on mekaniikoista seuraavia reaktioita, Tetriksen tapauksessa pisteytyksen ja palikoiden muotojen eroista seuraa tavoite poistaa neljä riviä samanaikaisesti. Estetiikka kuvaa haluttua pelaamisen aikaista pelaajan kokemaa tunnetta, kun hän pelaa peliä. Tetris -pelissä on tarkoitus haastaa pelaajaa jatkuvasti nopeutuvissa pelitasoissa, pelaajan edistymisen tahtiin, sekä antaa pelaajalle onnistumisen tunteita pelitason läpäisyn tai pisteiden määrän mukaisesti.

Smith (2014b) on luonut viitekehysten, jonka perusteella on mahdollista määrittää PCG elementtien vaikutuksia pelisuunnittelussa. Smithin viitekehys perustuu Hunicken ym. (2004) kolmeen osa-alueeseen, joita on avattu laajemmin PCG:n näkökulmasta. Esimerkiksi mekaniikka, kuten pelikentät, on mahdollista generoida pelaamisprosessia ennen tai sen aikana.

Vaikka pelin mekaanisten komponenttien suunnittelu on tärkeä osa suunnitteluprosessia, niiden määrittely ei ole yhtä tärkeää kuin niistä seuraavat ominaisuudet. Koska suunnittelijoilta puuttuu Smithin (2014a) mukaan ymmärrys PCG:n mahdollisuuksista pelisuunnittelussa, on tärkeämpää käsitellä mitä PCG:n käytöstä voi seurata, kuin miten mekaniikkoja tuotetaan. Tämän vuoksi seuraavat alaluvut keskittyvät erityisesti tuotettuihin dynamiikkoihin ja estetiikkoihin, mekaniikkojen sijaan.

Seuraavat alaluvut pohjautuvat vahvasti Smithin materiaaliin ja niitä tarvitaan luvun neljä pohjaksi.

3.1 Mekaniikkojen proseduraalinen tuottaminen

Smith (2014b) on tarkastellut mekaniikkojen generointia neljästä näkökulmasta: rakennuskomponentit, generoinnin aika, vuorovaikutuksen muoto, ja pelaajan kokemus. Nämä neljä mekaniikkaa määrittävät miten PCG toimii pelissä ja sen suunnittelussa:

- Pelin sisällön generointi voi tapahtua eri kokoisilla komponenteilla; pelitason komponentit voivat olla esimerkiksi laattoja, huoneita tai kokonaisia rakennuksia.
- Sisältöä on mahdollista tuottaa pelikokemuksen aikana tai sen ulkopuolella; Endless runner -peleissä pelitasoa laajennetaan pelaamisen aikana, kun taas roguelike -peleissä pelitaso rakennetaan kokonaisuudessaan ennen kuin pelaaja pääsee pelitasolle.
- Pelaajat voivat olla vuorovaikutuksessa sisällön tuottamisessa käytettävän generaattorin kanssa, ja se voi olla eri tasoista; Aikaisemmin mainituissa roguelike -peleissä generaattori tuottaa usein sisällön ilman pelaajan vuorovaikutusta. Rimworld (Ludeon Studios 2016) -peli antaa pelaajan määrittää pelimaailman alueiden ominaisuuksia, kuten lämpötilaa tai sateen määrää, ennen pelimaailman generointia. On myös mahdollista, että pelaajalla on täysi hallinta generaattorista, kuten Spore (Maxis 2008) -pelin pelihahmon rakennusvaiheessa.
- Generaattorit vaikuttavat pelikokemukseen ja ne voivat tuottaa siihen vaikuttavaa sisältöä epäsuorasti, koostumuksellisesti tai kokemuspohjaisesti; Epäsuorassa tuottamisessa generaattori ei hallitse sisällön tuottamista pelaajan pelikokemusta ajatellen, kuten roguelike -peleissä, joissa pelitasot usein generoidaan ilman pelaajan toivomuksia. Koostumuksellinen tuottaminen tapahtuu generoinnissa, jossa pelaaja voi hallita pelimaailmaan generoitavan sisällön määrää, mutta ei minne tai kuinka kyseinen sisältö sijoitetaan pelimaailmaan. Kokemuspohjaisessa tuottamisessa generaattorilla on suora hallinta pelaajan pelikokemukseen, kuten Warning Forever (Hikware 2003) -pelissä, jossa generaattori kehittyy tuottamaan päävihollisia, jotka pakottavat pelaamisstrategian vaihdoksia.

3.2 Sisältöjen dynamiikka ja proseduraalinen tuottaminen

Pelin käyttämien proseduraalisesti tuotettujen mekaniikkojen dynaamiset ominaisuudet ovat suunnitteluprosessin aikana tärkeämpiä kuin minkälaisia ominaisuuksia näillä mekaniikoilla on, sillä se vastaa suunnittelun aikaiseen kysymykseen: Miten PCG:llä tuotettu sisältö vaikuttaa pelikokemukseen? Dynamiikat voidaan jakaa Smithin (2014b) mukaan useaan eri osa-alueeseen: PCG:n relaatioon muihin mekaniikkoihin nähden, muistiin vs reaktioon, strategisointiin, tutkintaan, harjoitukseen sekä pelaajien väliseen vuorovaikutukseen.

PCG:llä tuotettu sisältö voi olla pelissä keskeinen, rajaava tai koristeellinen. Tasohyppelypeleissä pelitasot ovat tärkeässä asemassa pelikokemusta, kun taas RTS -peleissä pelitasot rajaavat missä tilassa ja miten pelaaja saattaa pelata peliä. Koristeellinen keskeisyys pelitasoissa voi olla esimerkiksi PCG:llä generoidut pelitason tekstuurit.

Muistin ja reaktion tärkeys riippuu Smithin mukaan vahvasti siitä, missä tilassa PCG:tä tuotetaan ja kuinka keskeinen osa peliä tuotettu sisältö on. Tasohyppelypelin pelitasot, jotka tuotetaan suunnitteluvaiheessa ja ovat samoja eri pelikertojen aikana, tukevat pelaajaa muistamaan pelitasot sekä lähestymään näitä pelitasoja opettelemalla ne ulkoa. Toisaalta, jos pelitasot eroavat pelikerrasta toiseen, ne kehittävät pelaajaa pelaamaan pelitasoja reaktionaalisesti, sillä pelissä on vähemmän elementtejä, joita opetella ulkoa.

Strategisoinnilla tarkoitetaan pelaajan tuottamia taktiikoita generaattorin ympärille, joka voi tapahtua pelaajien ymmärtäessä sisällön generointimallin. Pelaajat voivat oppia generaattorin ja pelaajan välisen vuorovaikutuksen vuoksi, miten heidän käyttöksensä vaikuttavat generaattoriin ja siten tuottaa strategioita ohjaamaan generaattoria. Generointialgoritmin yksinkertaisuus saattaa johtaa myös siihen, että pelaajat oppivat uudelleenpelaamisen aikana ennustamaan, miten pelitaso on rakennettu jo ensimmäisten huoneiden perusteella. Tämänkaltaisen yksinkertainen generointitapa voi olla esimerkiksi generointi sisällön valintana.

PCG:n käyttäminen pelitasojen suunnittelussa säästää resursseja ja aikaa, jonka vuoksi suunnittelijat käyttävät sitä suurien pelimaailmojen tuottamisessa. Pelimaailmojen laajenemisen vuoksi pelaajat käyttävät enemmän aikaa sen tutkimiseen menettämättä kiinnostustaan. Kiinnostus menetetään helposti, jos pelaaja oppii pelimaailman tuottamisessa käytetyt mallit. Tämän vuoksi monimutkaisilla algoritmeilla rakennetut pelimaailmat säilyttävät pelaajan seikkailunhalun pidempään.

Harjoituksella tarkoitetaan PCG:n käyttöä pelitaitojen harjoittamiseen. Generaattorilla voidaan rakentaa pelitasoja, jotka kehittävät pelaajan taitoja uudenaikaisessa ympäristössä. Pelitasojen generointi voi olla sattumanvaraista tai pelaaja voi olla vuorovaikutuksessa sen kanssa, joka auttaa pelaajaa kohdentamaan harjoittelua. Esimerkiksi proseduraalisesti kehitetyt pulmat, joilla on yhteinen ratkaisutapa, auttavat pelaajaa ratkaisutavan harjoittelussa.

Viimeisenä dynamiikkana Smith on maininnut pelaajien välisen vuorovaikutuksen ja keskustelun proseduraalisesti tuotetuista mekaniikoista. Dynamiikka on samankaltainen myöhemmän yhteisöllisyyden estetiikan kanssa, mutta se voidaan laskea myös dynamiikaksi, sillä se on pelaajien keskinäinen reaktio mekaniikkaa kohtaan. Toinen syy on keskustelun päämäärä: keskustelu kohdistuu proseduraalisesti tuotettuun sisältöön, pelkän sosiaalisen kanssakäynnin sijaan. Pelaajat ovat esimerkiksi vertailleet proseduraalisesti tuotetun pelin läpipeluuohjeiden eroja, kun pelikertojen välillä on huomattu eroja (Fernández-Vara ja Thomson 2012).

3.3 Sisältöjen estetiikka ja proseduraalinen tuottaminen

Hunicke ym. (2004) ovat jakaneet estetiikan, eli pelin toimintatavan tuottaa nautinnollisen pelikokemuksen, kahdeksaan osa-alueeseen: tunteeseen, fantasiaan, narratiiviseen, haasteeseen, yhteisöllisyyteen, löytämiseen, ilmaisuun ja alistumiseen. Hunicken ym. mainitsevat, että on olemassa muitakin osa-alueita, kuten esimerkiksi kilpailullisuus.

Smith (2014b) on käyttänyt edellä mainituista kahdeksasta osa-alueesta viitekehyksessään kolmea; haastetta, yhteisöllisyyttä, ja löytämistä. Hänen valintansa perustuvat aikaisempiin dynamiikkoihin: Haaste perustuu strategisointiin, muistiin vs reaktioon sekä harjoitteluun, sillä kaikki ovat jonkinlaisia pelaajan kokemia vastuksia. Yhteisöllisyys perustuu yhteisöllisyyden dynamiikkaan, sillä se tukee pelaajien välistä vuorovaikutusta ja keskustelua. Löytämisen estetiikka on pelimaailman ja strategioiden tutkintaa, koska pelaajat yrittävät ymmärtää systeemien rakennetta, joka tuottaa löytämisen tunteen.

Smithin rajauksen ongelma on tunteiden poisjättäminen PCG:n tuottamasta estetiikasta. Gravina ym. (2016) ovat maininneet mahdollisuuden käyttää PCG:tä tietynkaltaisten tunteiden, kuten yllätyksen, tuottamiseen. Tunteiden tuottamista PCG:n avulla on tutkittu tieteen alalla myös myöhemmin (Chakrabortii, Ferreira ja Whitehead 2017).

4 Videopelien tasosuunnittelu

PCG:n määrittelyn yhteydessä tasosuunnittelun todettiin sisältävän myös pelimaailman ja pelitason sisällölliset komponentit. Bjork ja Holopainen (2005) ovat määrittäneet pelin elementit neljään eri luokkaan: pelimaailmaan, objekteihin, abstrakteihin objekteihin ja sijainteihin. Pelimaailma toimii tilana, johon voidaan sijoittaa objekteja ja sijainteja. Objektit ovat pelin elementtejä, joiden kanssa pelaaja on vuorovaikutuksessa suorasti tai epäsuorasti pelitilan muutoksen avulla. Abstraktit objektit ovat taas elementtejä, jotka eivät välttämättä ole konkreettisesti olemassa pelimaailmassa, mutta niitä käytetään kuvastamaan pelin tämänhetkistä tilaa tai abstrakteja arvoja. Sijainnit toimivat elementteinä, joissa pelimaailman tila poikkeaa normaalista. Nämä sijainnit saattavat muuttaa normaaleja toimintoja sen sisällä, toimia turvapaikkoina tai erityislaatuksina pisteinä pelin läpäisyn kannalta. Pelin sisäisiä elementtejä voidaan jakaa edellä mainittuihin luokkiin seuraavalla tavalla:

- pelimaailmat: pelitasot, pelimaailmat, suljetut alueet, ym.
- objektit: viholliset, ansat, esteet, vihjeet, tavarat, hahmot, ym.
- abstraktit objektit: pisteet, elämät, kamerat, ennätysluettelot, ym.
- sijainnit: resurssien sijainnit, tallennussijainnit, turvapaikat, määränpääät, ym.

Kirjallisuuskartoituksen näkökulmasta pelitasot toimivat tällöin runkona, johon sisältöä voidaan sijoittaa. Tällöin peligenrekohtaiset sisällölliset painotuserot vaikuttavat samalla pelitason suunnittelussa. Roolipeleissä pelaajan suorittamat tehtävät toimivat ensisijaisena narratiivisena etenemistapana, jonka vuoksi näitä pelejä kutsutaan usein tehtäväpohjaisiksi peleiksi (Smith ym. 2011). Smith ym. (2011) kuvailevat kyseisiä tehtäviä vahvasti riippuvaisiksi tason suunnitteluun, sillä tehtävät toimivat pelitason fyysisten rajoitteiden sisällä. Hullett ja Whitehead (2010) määrittävät FPS -pelien keskittyvän genrenä toimintaan, joka tapahtuu erilaisten aseiden avulla. He esittävät genressä mahdollisina sijaintietuina esimerkiksi maastolliset pullonkaulat tai tarkka-ammuntapaikat, joissa pelaajien toimintaa saatetaan rajoittaa tai laajentaa normaalista poikkeavaksi. Esimerkin maastollinen pullonkaula ei tuottaisi samanlaista pelaajalle asettamaa haastetta, jos peligenre ei keskity toimintaan.

Pelin sisällölliset elementit ovat myös vuorovaikutuksessa toistensa kanssa, joka on erityisesti nähtävissä peleissä, joissa pelaajat kilpailevat keskenään. Toimintaan keskittyvät FPS -pelit käyttävät aseita toiminnan tapoina, jolloin aseiden ominaisuuksien erot vaikuttavat pelaajien väliseen toimintaan ja pelaajien pelikokemuksen nautintoon. Aseilla voi esimerkiksi olla erilaisia kantomatkoja, jolloin niillä on samalla toisiinsa nähden ominaislaatuiset edut; aseiden kantomatkojen erot tuottavat turvallisia etäisyyksiä vahingoittaa vihollisia. (Karavolos, Liapis ja Yannakakis 2017) Tällöin pelitason tuottamat mahdollisuudet näiden etäisyyksien hyödyntämiseen vaikuttaa siihen, miten aseet ovat tasapainossa toistensa kanssa. Esimerkiksi tarkka-ammuntapaikan korkeus, suoja, näköalue ja puolustusmahdollisuudet vaikuttavat sen toimintaan pelitason sisällä ja aseiden tasapainoon. (Hullett ja Whitehead 2010)

Pelitason tai pelimaailman elementeillä on myös mahdollista ohjata pelaajan liikkeitä. Lemarchandin (2012) mukaan hyvin suunnitellun pelin pitäisi ohjata pelaajan huomio pelikokemusta rikastuttaviin elementteihin, turhien elementtien sijaan. Esimerkiksi Journey (Thatgamecompany 2012) -peli käyttää maastoa, valoa ja näköaloja ohjaamaan pelaajaa avoimessa maailmassa, ilman erikseen luotuja selviä ohjeita, ja ketjuttaa näitä elementtejä ohjaamaan pelaajaa koko pelin aikana. Wintersin ja Zhunin (2014) tutkimuksessa rakenteelliset koostumusmallit, kuten muoto, tila ja järjestely, vaikuttavat pelaajan kulkemiin reitteihin. Esimerkiksi normaalista kaupungista erottuva korkea torni tai pelaajan edessä oleva tie, joka jatkuu kaukaisuuteen, ohjaavat pelaajan liikkeitä.

Kokonaislaatuinen pelikokemus pelille muodostuu edellä mainittujen tapauksien vuoksi Bjorkin ja Holopaisen (2005) tuottaman mallin yksittäisten elementtien holistisena kokonaisuutena. Jos proseduraalista sisällön tuottamista käytetään pelin yhdessä osa-alueessa arvioimattomasti, sen tuottaman lopputuloksen kokonaisuus on sattumanvarainen, joka vaikuttaa negatiivisesti pelaajien nautintoon. Seuraavissa alaluvuissa käsitellään PCG:n tapauksia tasosuunnittelussa sekä mahdollisia uusia suuntauksia PCG:lle tasosuunnittelussa aikaisempien tutkimuksien pohjalta.

4.1 Proseduraalisuuden hyödyntäminen tasosuunnittelussa

Proseduraalinen sisällön tuottaminen mahdollistaa suunnittelijoita tuottamaan suuremman määrän sisältöä vähemmällä työmäärällä, mutta vaatii suunnittelijoiden siirtymistä sisällön tuottamisen hallinnointiin, sen suunnittelun sijaan. Tässä luvussa käsitellään miten PCG:tä on hyödynnetty tasosuunnittelussa.

Smith ym. (2010) ovat luoneet järjestelmän, joka tuottaa tasohyppelypelille pelitasoa pelin rytmin mukaan. Suunnittelijoiden tuottamat tasohyppelypelit sisältävät usein haastavia ja tarkkaan ajoitettuja hyppyjä. Smithin ym. tuottama Launchpad -järjestelmä rakentaa pelitason parametrisoiduista rytmilohkoista, joita yhdistelemällä tuotetaan haluttu pelirytm, riippumatta pelitason geometriasta. Launchpad toimii kaksiosaisena kieliopin generointina; ensimmäisessä osassa järjestelmä tuottaa tahdin, joissa tahditukset vastaavat pelaajien toimintoja, ja toisessa osassa näistä aikaisemmin tuotetuista tahdeista luodaan geometrinen pelitaso. Heidän tuottamansa järjestelmän avulla on mahdollista ohjelmallisesti tuottaa yksilöllisiä pelikenttiä, jotka ovat varmasti pelattavissa, toisin kuin puhtaasti satunnaistamalla yksilöllistetyt kentät. (Shaker, Togelius ja Nelson 2016)

Giacomello ym. (2018) opettivat kahta generatiivista kilpailevaa verkostoa tuottamaan pelitasoja, jotka ovat tyyllisesti samankaltaisia opetusmateriaalin kanssa. He jakoivat opetuksen kahteen osa-alueeseen: ensimmäinen verkosto hyödynsi tavallisia pelitason kuvia ja toinen hyödynsi kuvia sekä erikseen eroteltuja ominaisuuksia. Nämä tapaukset voidaan luokitella ohjattuun ja ohjaamattomaan opetukseen. Heidän molemmat generatiiviset kilpailevat verkostot tuottivat DOOM (id Software 1993) -pelin tyyllisiä pelitasoja, mutta ohjattu oppiminen tuotti korkeatasoisempia pelitasoja, ohjaamattomaan opetukseen verrattuna. Giacomellon ym. opetustapa eroaa normaalista generatiivisen kilpailevan verkoston opetuksesta, sillä se ei vaadi tiedon siirtämistä koodiksi, kuten tavallinen proseduraalisen sisällön tuottaminen koneoppimisen avulla vaatisi.

Lee ym. (2016) hyödynsivät neuroverkkoja ennustamaan StarCraft2 (Blizzard Entertainment 2010) -pelin pelitasojen resurssien sijainteja. Pelitason korkeusvaihtelut rajoittavat ja ohjaavat pelaajan liikkumista sekä resurssien strategista sijoittelua. Neuroverkon opetusmateriaalina käytettiin korkeuskarttoja. Heidän lopullinen neuroverkko tuottaa todennäköisyyksiä

resurssien sijoittumiselle ja sitä voidaan määritellä parametrien avulla. Neuroverkko asettaa sille parametrisoidun määrän resursseja uuteen pelikenttään, sille opetetun esimateriaalin pohjalta. Proseduraalinen resurssipaikkojen generointi todettiin toimivaksi, mutta lopputuloksien laadut erosivat toisistaan kenttäkohtaisesti. Laadullisten erojen oletettiin johtuvan esimateriaalin puutteesta, joka on yleinen ongelma koneoppimisen hyödyntämisessä.

4.2 Proseduraalisuuden uudet mahdollisuudet tasosuunnittelussa

Peliteollisuus hyödyntää PCG:tä lähinnä pelitasojen uudelleen pelattavuuden mahdollistamiseen ja muut näkökulmat jäävät vähemmälle huomiolle. (2014a). Chakrabortii ym. (2017) mainitsevat pelikokemuksien olevan toistuvia, vaikka pelit eivät toista itseään pelitasoissa, joka viittaa uudelleenpelattavan pelikokemuksen olevan toistuvaa. He väittävät tämän johtuvan proseduraalisen sisällön puutteesta luoda tunteellinen yhteys pelaajaan. He ehdottavat yllätyneisyyden olevan yksi tärkeimmistä tunteista tunneyhteyden luontiin, sillä se vahvistaa muita tunnekokemuksia (Luna ja Renninger 2015).

Yannakakis ym. (2016) ovat tehneet tutkimusta psykofysiologian mahdollisuuksista peleissä ja väittävät fysiologian olevan tärkeä elementti tunnepitoisen vuorovaikutuksen saavuttamiseen. Tämän mukaan Chakraborttiin tunnistaman tunnevajeen taustalla on proseduraalisen sisällön puutteellinen kyky tuottaa pelaajissa sopivia fysiologisia reaktioita.

Hunicken ym. (2004) viitekehyksen näkökulmasta puutteellinen tunneyhteys on seurausta siitä, että sisältö ei tuota pelaajalle riittävää esteettistä nautintoa, kuten löytämisen tai haasteen nautintoa. Jos pelisuunnittelijat kykenisivät löytämään mitkä pelin elementit tuottavat pelikokemukselle tärkeitä psykofysiologisia reaktioita, proseduraalisesti tuotettua sisältöä voidaan rakentaa näiden reaktioiden tuottamiseen. Chakraborttiin ym. väittämän valossa tämä tarkoittaisi pelikokemuksien toistuvuuksien vähenemistä, josta vastaavasti seuraa pelaajille uniikkeja pelikokemuksia uudelleenpelaamisen aikana.

Proseduraalista sisällön tuottamista käytetään usein pelitasojen generointiin, mutta ei ole varmuutta siitä, kuinka paljon menetelmää hyödynnetään pelitasojen sisäisen liikkumisen hallintaan. Wintersin ja Zhunin (2014) tutkimus osoittaa rakenteellisten koostumusmallien mahdollisuuksista ohjata pelaajan liikkeitä 3D pelimaailmassa, joita on mahdollista sovel-

taa proseduraalisesti tuotettuihin pelitasoihin. Pelaajan liikkeitä ohjaavien koostumusmallien käyttö proseduraalisesti rakennetuissa pelitasoissa tuottaisi pelaajille uuden luonnollisen lähestymistavan pelitason sisällä liikkumiseen. Tämä vastaavasti vähentäisi pelikokemuksen toistuvuutta, sillä pelaajien toimintatavat pelitason sisällä muuttuisivat.

Mahdollisena ongelmana tämänkaltaiseen johdatteluun ovat kuitenkin pelaajien lähestymistavat; Wintersin ja Zhunin (2014) tutkimuksessa huomattiin joidenkin pelaajien käyttävän esimääriteltyä tapaa liikkua pelitasojen sisällä. Tämänkaltaisen menetelmä on esimerkiksi oikeanpuoleisen seinän seuraaminen, kun pelaaja liikkuu pelitason sisällä. Tällöin proseduraalisesti tuotetussa pelitasossa käytetty epäsuora liikkeiden ohjaaminen ei toimi myöskään silloin, jos pelaajat käyttävät proseduraaliselle sisällölle laadittua ratkaisumenetelmää, koska silloin pelaajat seuraavat pelitason rakennustapaa, sen visuaalisten ohjeiden sijaan. On kuitenkin mahdollista väittää, että näiden ratkaisumenetelmien toistuva käyttö on syy pelikokemusten toistuvuuteen, jolloin jos proseduraalisesti tuotetulle sisällölle ei ole olemassa tämänkaltaista ratkaisumenetelmää, pelikokemus ei olisi toistuvaa. Liiallinen sattumanvaraisuus proseduraalisessa sisällössä ilman pelaajan ymmärrystä vaikuttaa Smithin (2014b) mukaan pelikokemukseen negatiivisesti, joten tämä menetelmä ei välttämättä ole realistinen toteutustapa uudelleenpelattavuuden kehittämiseen.

5 Yhteenveto

Tässä kirjallisuuskartoituksessa käsiteltiin proseduraalisen sisällön tuottamista videopelien tasosuunnittelussa, koska sen käyttö on kasvanut peliteollisuudessa huomattavasti. PCG:n käyttö peliteollisuudessa rajoittuu pääasiassa pelien uudelleenpelattavuuden kehittämiseen, jonka vuoksi PCG:n käyttö peliteollisuudessa on rajoittunutta. Katsauksessa pyrittiin tämän vuoksi käymään läpi proseduraalisen sisällön tuottamisen ominaisuuksia, lähestymistapoja, käytännön sovellutuksia videopeleissä sekä sen muita mahdollisuuksia pelitasojen suunnittelussa, jotta pelisuunnittelijoiden ymmärrys tekniikasta lisääntyisi. Katsauksessa huomattiin mahdollisia sovellutuksia PCG:lle pelitasojen optimoinnissa, uusien pelitasojen kehityksessä opetusmateriaalin pohjalta, epäsuorassa pelaajan ohjaamisessa sekä sen käytössä tunneyhteyksien luomisessa.

Katsaus osoitti, että PCG:tä kehitetään ensisijaisesti peligenren ulkopuolella, mutta menetelmän soveltamista myös peleille on tutkittu. Tutkimustulokset antavat suunnittelijoille pohjan PCG:n soveltamiselle tasosuunnittelussa. Lisäksi on kehitetty valmiita työkaluja, joita suunnittelijat pystyvät käyttämään proseduraalisen sisällön tuottamiseen ilman tarvetta matemaattiseen taustatietoon.

Katsauksen pohjalta tiedetään, että PCG:llä on enemmän tutkittua potentiaalia, joka olisi peliteollisuuden hyödynnettävissä kuin mitä käytetään. Miksi peliteollisuus ei ole tarttunut menetelmän mahdollisuuksiin, olisi lisätutkimuksien arvoista.

Lähteet

A.I. Design. 1980. *Rogue*. [Videopeli].

AcademySoft. 1984. *Tetris*. [Videopeli].

Bjork, Staffan, ja Jussi Holopainen. 2005. *Patterns in game design*. Nide 11. Charles River Media Hingham.

Blizzard Entertainment. 2010. *StarCraft II: Wings of Liberty*. [Videopeli].

Butler, Eric, Adam M Smith, Yun-En Liu ja Zoran Popovic. 2013. “A mixed-initiative tool for designing level progressions in games”. Teoksessa *Proceedings of the 26th annual ACM symposium on User interface software and technology*, 377–386.

Chakrabortii, Chandranil, Lucas N Ferreira ja Jim Whitehead. 2017. “Towards generative emotions in games based on cognitive modeling”. Teoksessa *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–2.

Dormans, Joris. 2010. “Adventures in level design: generating missions and spaces for action adventure games”. Teoksessa *Proceedings of the 2010 workshop on procedural content generation in games*, 1–8.

Fernández-Vara, Clara, ja Alec Thomson. 2012. “Procedural generation of narrative puzzles in adventure games: The puzzle-dice system”. Teoksessa *Proceedings of the The third workshop on Procedural Content Generation in Games*, 1–6.

Giacomello, Edoardo, Pier Luca Lanzi ja Daniele Loiacono. 2018. “DOOM level generation using generative adversarial networks”. Teoksessa *2018 IEEE Games, Entertainment, Media Conference (GEM)*, 316–323. IEEE.

Gravina, Daniele, Antonios Liapis ja Georgios N Yannakakis. 2016. “Constrained surprise search for content generation”. Teoksessa *2016 IEEE Conference on Computational Intelligence and Games (CIG)*, 1–8. IEEE.

Hello Games. 2016. *No Man’s Sky*. [Videopeli].

- Hendrikx, Mark, Sebastiaan Meijer, Joeri Van Der Velden ja Alexandru Iosup. 2013. "Procedural content generation for games: A survey". *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)* 9 (1): 1–22.
- Hikware. 2003. *Warning Forever*. [Videopeli].
- Hullett, Kenneth, ja Jim Whitehead. 2010. "Design patterns in FPS levels". Teoksessa *proceedings of the Fifth International Conference on the Foundations of Digital Games*, 78–85.
- Hunicke, Robin, Marc LeBlanc ja Robert Zubek. 2004. "MDA: A formal approach to game design and game research". Teoksessa *Proceedings of the AAAI Workshop on Challenges in Game AI*, 4:1722. 1.
- id Software. 1993. *DOOM*. [Videopeli].
- Karavolos, Daniel, Antonios Liapis ja Georgios Yannakakis. 2017. "Learning the patterns of balance in a multi-player shooter game". Teoksessa *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–10.
- Khaled, Rilla, Mark J Nelson ja Pippin Barr. 2013. "Design metaphors for procedural content generation in games". Teoksessa *Proceedings of the SIGCHI conference on human factors in computing systems*, 1509–1518.
- Koster, Raph. 2013. *Theory of fun for game design*. "O'Reilly Media, Inc."
- Lee, Scott, Aaron Isaksen, Christoffer Holmgård ja Julian Togelius. 2016. "Predicting resource locations in game maps using deep convolutional neural networks". Teoksessa *Twelfth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- Lemarchand, Richard. 2012. "Attention NOT Immersion: Making your games better with psychology and playtesting, the Uncharted Way". Teoksessa *Game Developers Conference*, 31.
- Ludeon Studios. 2016. *Rimworld*. [Videopeli].
- Luna, Tania, ja LeeAnn Renninger. 2015. *Surprise: Embrace the unpredictable and engineer the unexpected*. TarcherPerigee.
- Maxis. 2008. *Spore*. [Videopeli].

- Müller, Pascal, Peter Wonka, Simon Haegler, Andreas Ulmer ja Luc Van Gool. 2006. “Procedural modeling of buildings”. Teoksessa *ACM SIGGRAPH 2006 Papers*, 614–623.
- Risi, Sebastian, ja Julian Togelius. 2019. “Procedural Content Generation: From Automatically Generating Game Levels to Increasing Generality in Machine Learning”. *arXiv preprint arXiv:1911.13071*.
- Secretan, Jimmy, Nicholas Beato, David B D’Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik ja Kenneth O Stanley. 2011. “Picbreeder: A case study in collaborative evolutionary exploration of design space”. *Evolutionary computation* 19 (3): 373–403.
- Shaker, Noor, Julian Togelius ja Mark J Nelson. 2016. *Procedural content generation in games*. Springer.
- Smelik, Ruben M, Tim Tutenel, Rafael Bidarra ja Bedrich Benes. 2014. “A survey on procedural modelling for virtual worlds”. Teoksessa *Computer Graphics Forum*, 33:31–50. 6. Wiley Online Library.
- Smith, Gillian. 2014a. “The future of procedural content generation in games”. Teoksessa *Tenth Artificial Intelligence and Interactive Digital Entertainment Conference*.
- . 2014b. “Understanding procedural content generation: a design-centric analysis of the role of PCG in games”. Teoksessa *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 917–926.
- Smith, Gillian, Ryan Anderson, Brian Kopleck, Zach Lindblad, Lauren Scott, Adam Wardell, Jim Whitehead ja Michael Mateas. 2011. “Situating quests: Design patterns for quest and level design in role-playing games”. Teoksessa *International Conference on Interactive Digital Storytelling*, 326–329. Springer.
- Smith, Gillian, Alexei Othenin-Girard, Jim Whitehead ja Noah Wardrip-Fruin. 2012. “PCG-based game design: creating Endless Web”. Teoksessa *Proceedings of the International Conference on the Foundations of Digital Games*, 188–195.

- Smith, Gillian, Jim Whitehead, Michael Mateas, Mike Treanor, Jameka March ja Mee Cha. 2010. “Launchpad: A rhythm-based level generator for 2-d platformers”. *IEEE Transactions on computational intelligence and AI in games* 3 (1): 1–16.
- Sullivan, A. 2010. “Content Selection vs. Content Generation”. *Expressive Intelligence Studio*.
- 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.
- Thatgamecompany. 2012. *Journey*. [Videopeli].
- Togelius, Julian, Emil Kastbjerg, David Schedl ja Georgios N Yannakakis. 2011. “What is procedural content generation? Mario on the borderline”. Teoksessa *Proceedings of the 2nd international workshop on procedural content generation in games*, 1–6.
- Togelius, Julian, ja Jurgen Schmidhuber. 2008. “An experiment in automatic game design”. Teoksessa *2008 IEEE Symposium On Computational Intelligence and Games*, 111–118. IEEE.
- Togelius, Julian, Georgios N Yannakakis, Kenneth O Stanley ja Cameron Browne. 2011. “Search-based procedural content generation: A taxonomy and survey”. *IEEE Transactions on Computational Intelligence and AI in Games* 3 (3): 172–186.
- Turing, A. M. 1950. “Computing machinery and intelligence”. *Mind* LIX (236): 433–460.
- Winters, Glenn Joseph, ja Jichen Zhu. 2014. “Guiding players through structural composition patterns in 3D adventure games.” Teoksessa *FDG*.
- Yannakakis, Georgios N, Hector P Martinez ja Maurizio Garbarino. 2016. “Psychophysiology in games”. Teoksessa *Emotion in games*, 119–137. Springer.