

Gabi Kinkela

Tilastotieteelliset menetelmät pelikehityksessä

Tietotekniikan Kandidaatintutkielma

30. huhtikuuta 2022

Jyväskylän yliopisto

Informaatioteknologian tiedekunta

Tekijä: Gabi Kinkela

Yhteystiedot: gagawaki@student.jyu.fi

Ohjaaja: Sanna Juutinen

Työn nimi: Tilastotieteelliset menetelmät pelikehityksessä

Title in English: Statistical methods in game development

Työ: Kandidaatintutkielma

Opintosuunta: All study lines

Sivumäärä: 17+2

Tiivistelmä: Tässä kandidaatintutkielmassa selvitetään mitä kaikkia tilastotieteellisiä menetelmiä tai tekoälyä voidaan käyttää pelikehityksessä mahdollisimman tyydyttävän pelikokemuksen luomiseen pelaajalle. Tutkimustulokset osoittivat menetelmien olevan hyödyllisiä apukeinoja pelien parantamiseen pelikehityksen kannalta.

Avainsanat: tekoäly, pelikehitys

Abstract: This thesis studies how gamedevelopers can create the most satisfying experience for players using statistical methods or artificial intelligence as tools in game development. According to the research results, statistical methods are shown to be useful tools in game development.

Keywords: Artificial Intelligence, Game Development

Esipuhe

Jo pienestä pitäen tiesin, että haluan päästä työskentelemään pelialalle. Aikaisemmin tähtäsin pelikehittäjäksi, mutta yliopistossa opiskelun myötä pelitutkimus niin menetelmien optimoinnin kuin psykologisten vaikutuksien kannalta on alkanut kiinnostamaan myös. Toivon, että saattamalla tämän työn loppuun pystyn tuottamaan enemmän tieteellisiä tekstejä tai muita projekteja.

Haluaisin kiittää vanhempiani jotka tukivat uraani pelaajana, ystäviäni jotka ovat auttaneet vaikeina aikoina ja kaikkia opettajia jotka opettivat erilaisia tapoja tutkia maailmaa.

Jyväskylä, 30. huhtikuuta 2022

Gabi Kinkela

Kuviot

Kuvio 1. Esimerkkejä aaltofunktion romahduksen käytöstä (Gumin 2021)	14
--	----

Sisällys

1	JOHDANTO	1
2	TILASTOTIETEELLISET MENETELMÄT	3
2.1	Aaltofunktion romahdus	3
2.2	Proseduraalinen generointi.....	5
2.3	Geneettiset algoritmit	5
2.4	Vahvistettu oppiminen	6
3	TILASTOTIETEELLISTEN MENETELMIEN HYÖDYNTÄMINEN PELIKE- HITYKSESSÄ	7
4	YHTEENVETO.....	9
	LÄHTEET	10
	LIITTEET.....	13

1 Johdanto

Nykyään pelikehityksestä on tehty niin helppoa, että kuka tahansa voi tehdä pelin. Tämän seurauksena yhä useammat pelit ovat yksinkertaisempia ja tylsiä. Suurempi osa pelaajista arvostaa haasteita jolloin pelin viihdyttävyyden paraneekin (Bopp, Mekler ja Opwis 2016). Jos haluaa tuottaa yksin sisällöllisesti laadukkaan pelin, joka haastaa pelaajaa jatkuvasti esimerkiksi fyysisesti reaktionopeuksilla tai mentaalisesti pulmilla ja tilan hahmottamisella, kehittämiseen menee paljon aikaa ja työmäärä vaikuttaa suurelta.

Mielestäni hyvästä pelistä voidaan löytää neljä pääpiirrettä: tekemisen nauttiminen, syy palata takaisin, haasteiden kohtaaminen sekä niiden ylitse pääseminen ja ulkopuolinen pohdinta. Tekemisen nauttimisella tarkoitetaan sitä, että peli tuntuu tyydyttävältä pelata. Peli sisältää erilaisia pelituntoon (engl. *game feel*) vaikuttavia asioita, kuten ruuduntärinä osumasta viholliseen tai jonkun tietyn äänen päästäminen pääosumasta (Swink 2008). Vaikka pelituntoon vaikuttavat asiat kuulostavat turhilta, ilman niitä hyvät pelit muuttuvat keskinkertaisiksi.

Perinteisesti kun peli pelataan läpi, se unohdetaan ja siirrytään seuraavaan. Luomalla syitä palata takaisin, lyhyistäkin peleistä voidaan tehdä paljon monipuolisempia. Peli voi itsessään sisältää jotain vaihtoehtoisia pelimuotoja tai niitä voidaan luoda lisää joko virallisesti tai epävirallisesti modaamalla peliä (Sotamaa 2010, s. 239-255).

Pelit joissa pelaajalle annetaan työkalut ja joku opittava taito, antavat haasteita jotka tuntuvat hyviltä ylittää. Jos opittava taito on toteutettu tarpeeksi hyvin, pelaajaa ei tarvitse palkita mitenkään pelinsisäisesti, vaan palkitsevuus tulee oman taitavuuden tajuamisesta (Swink 2008). Jos pelaajalta ei vaadita mitään, niin mitä eroa pelaamisella ja elokuvan katsomisella on.

Kun pelaajia haastateltiin haastavuudesta ja heidän kokemuksistaan eri vaikeustasoilla Unreal Tournament 2 videopelissä, tulokset osoittivat pelaajien suosivan helppoutta (Klimmt ym. 2009). Tutkimuksessa otettiin huomioon vain miten pelaaja pärjäsi tässä tietyssä pelissä. Haastavuuteen vaikuttavat myös asiat kuten pelaajahahmon ohjattavuus, ympäristön ymmärtäminen, kokemus kyseisestä pelistä ja vihollisen tekoälyn reiluus, mitä ei otettu huomioon. Henkilökohtaisesta kokemuksesta miellyttävältä tuntuva vaikeaa peliä on nautinnollisempi

pelata, kuin huonolta tuntuva helppoa peliä.

Ulkopuolisella pohdinnalla tarkoitetaan sitä, että peli jää pyörimään mieleen vielä kun pelin sulkee. Tämä on hieman abstraktimpi, sillä jokainen edellä mainittu hyvän pelin piirre voi antaa pohdiskeltavaa. Pohdittavan asian ei ole pakko olla joku pelimekaniikka, vaan se voi olla myös pelin maailma, musiikki tai tarina.

Hyvän pelin luomista varten voidaan hyödyntää tilastotieteestä tuttuja menetelmiä, jotka voidaan yksinkertaistaa tekoälyksi. Sen sijaan, että luodaan itse käsintehty maailma, annetaan mallille parametrit minkälainen on hyvä maailma ja annetaan tekoälyn toteuttaa se haluamallaan tavalla. Jos halutaan lisätä hirviö, on järkevämpää luoda kone joka luo hirviöitä kuin pelkkä hirviö. Yksinkertaistamalla pelikehityksen eri osa-alueiden hienosäätöä, voidaan pelikehityksessä keskittyä enemmän edellä esitettyjen pääpiirteiden kehitykseen. Tässä tekstissä käsitellään aaltofunktion romahdusta, proseduraalista generointia, geneettisiä algoritmeja sekä vahvistettua oppimista ja miten niitä voidaan hyödyntää ainakin haasteiden ja takaisin palaamisen syiden kannalta.

2 Tilastotieteelliset menetelmät

Tilastotieteellisillä menetelmillä tarkoitetaan tässä tutkielmassa tekoälyä, jolla tarkoitetaan malleja jotka oppivat jollain menetelmällä. Välttelen tekoäly-termiä, sillä pelien kontekstissa hahmo joka kävelee kohti pudotusta ja kääntyy ympäri, on periaatteessa tekoäly. Kaplanin ja Haenleinin (2019) mukaan tekoäly on järjestelmän kyky ymmärtää ulkoista tietoa, oppia siitä ja käyttää sitä hyväkseen mukautuakseen haluttuun lopputulokseen. Tässä tutkielmassa käsittelen, miten tilastotieteellisiä menetelmiä voidaan käyttää työkaluina pelikehittäjän kannalta soveltamalla niitä peleihin sekä käyttämällä niitä työkaluina kehitysvaiheessa.

Näitä menetelmiä voisi myös pelillistää, sillä peli saa tietoa pelaajalta hänen käyttäytymisestään ja tätä tietoa voidaan hyödyntää pelin muuttamisessa pelaajalle sopivaksi. Tämä ei ole uusi idea sillä *Left 4 Dead* (2008) niminen videopeli sisältää tekoälyohjajaan joka mukautuu pelaajan käyttäytymiseen helpottamalla tai vaikeuttamalla pelikulkua (Agis, Cohen ja Martínez 2015).

2.1 Aaltofunktion romahdus

Aluksi käsitellään aaltofunktion romahdus (engl. *Wave Function Collapse*) joka saa inspiraationsa kvanttimekaniikkasta (Karth ja Adam Marshall Smith 2021). Menetelmä täyttää rajatun tilan sääntöpohjaisesti ja sitä voidaan käyttää esimerkiksi kaksiulotteisen tasohyppeleypelin tason luomiseen. Pseudokoodi algoritmille on seuraava:

1. Aseta jokainen solu superpositioon eli jokaiseen mahdolliseen tilaan
2. Valitse vähiten entropiaa eli mahdollisia tiloja sisältävä solu ja aseta siihen joku tila
3. Romauta solut eli käy läpi solut jotka riippuvat asetetusta tilasta ja poista sääntöjen vastaiset tilat niistä
4. Jatka kohdasta 2 kunnes kaikilla soluilla on joku tila

Paras tapa hahmottaa algoritmin läpikäynti on ajatella sudoku-peliä (Donald 2020). Pelissä on yhdeksän 3x3 ruudukkoa, jotka täytyy täyttää luvuilla yhdestä yhdeksään siten, että jokainen luku esiintyy vain kerran yhdessä ruudukossa, rivillä ja sarakkeella. Pseudokoodi

sudokun ratkaisemiseen olisi seuraava:

1. Aseta jokaiseen soluun jokainen mahdollinen tila (1-9)
2. Valitse solu jolla on vähiten mahdollisia tiloja ja aseta siihen joku mahdollinen luku
3. Käy läpi saman 3x3 ruudukon solut, sekä pysty- että vaakarivin solut ja poista niistä käytetty luku
4. Jatka kohdasta 2 kunnes kaikki solut ovat täytetty

Varsinaisessa aaltofunktion romahduksen käytössä kaksiulotteiselle mallille voidaan antaa esimerkkipkuva, mistä algoritmi hahmottaa säännöt solujen mahdollisille tiloille, katso liite 1. Karth ja Adam M Smith (2019) toteaa, että perinteisten koneoppimismenetelmien vaatima laaja esimerkkiaineisto heikentää niiden mahdollisia hyötyjä, mutta pienemmät esimerkiksi kehittimet kuten aaltofunktion romahdus pystytään yhdistämään erotteleviin oppimisstrategioihin jotta artisti voi intuitiivisesti tehdä kohdistettuja muokkauksia vaarantamatta olemassa olevia ratkaisuja. Esitetyt esimerkit ovat olleet kaksiulotteisia, mutta aaltofunktion romahdusta voidaan käyttää myös kolmannessa ulottuvuudessa (Donald 2020). Tällöin ei enää voida käyttää esimerkkipkuvia, vaan tilojen säännöt on määriteltävä kuutioiden sivujen mukaan. Vaikka menetelmä toimii pienillä esimerkeillä, siihen voidaan lisätä sääntöjä kuten tavaroiden sisällyttäminen tai tiettyjen tilojen suosiminen painoilla. Menetelmä on myös sen verran kevyt, että sen suorittaminen on mahdollista ajonaikana (Sandhu, Chen ja McCoy 2019).

Aaltofunktion romahdukseen pätee ongelmat kuten ylioppiminen ja samankaltaisuus (Bucklew 2019). Ylioppimisella tarkoitetaan sitä, että tulokset ovat liian samankaltaisia, lähes kopioita alkuperäisestä. Bucklew (2019) esittää, että eräs ratkaisu on tasoittaa aaltofunktion romahduksia siten, että ensimmäisellä kerralla luodaan esimerkiksi kylä, minkä jälkeen ajetaan toinen romahdus, joka käsittelee kylän talojen sisätilat. Samankaltaisuudella tarkoitetaan rajattomuutta, sillä kaikki solut joudutaan romauttamaan samalla tavalla, jolloin on vaikeampi määrittää suurempia rakennelmia. Bucklewin mukaan ongelma voidaan ratkaista jakamalla solut eri funktioiden romautettavaksi, kuten kaikki solut ovat osana metsää, mutta metsän sisällä voi olla eri joukkoja määriteltynä raunioille tai kylille (Bucklew 2019).

Työmäärä maailmojen luomiseen on suhteellisen pieni, sillä kehittäjä joutuu rakentamaan

kaikki osat yhteensopiviksi ja luomaan säännöt niiden välille. Lopputuloksena on juuri sellainen kartta kuin halutaan, eikä mahdollomat poikkeukset ole mahdollisia toisin kuin seuraavaksi käsiteltävässä menetelmässä.

2.2 Proseduraalinen generointi

Tässä kappaleessa käsitellään proseduraalista generointia, joka on tuttu ja tunnettu tapa, miten pelikehittäjät lisäävät satunnaisuutta peleihin (Frade, Vega ja Cotta 2012). Kuten nimi kertoo, kyseessä on menetelmällinen tai sääntöihin pohjautuva generointi, kuten aaltofunktion romahdus, mutta käyttötarkoitus on yleispätevämpi (Galín ym. 2010). Siinä missä aaltofunktion romahdus on hyvä joukossa, proseduraalista generointia voidaan hyödyntää loputtomasti.

Esimerkiksi halutaan luoda kolmiulotteiseen peliin aavikkotasoa, jossa on kyliä satunnaisin välimatkoin. Aaltofunktion romahduksessa olisi määriteltävä ihan kaikki eri dyynien muodot, mikä olisi äärimmäisen työlästä. Proseduraalisella generoinnilla voidaan asettaa kylät paikoilleen ja käyttää jotain kohina-algoritmia luomaan erinäköisiä ja -kokoisia dyynejä kylien välille (Parberry 2014).

2.3 Geneettiset algoritmit

Käydään seuraavaksi läpi geneettiset algoritmit, jotka ottavat mallia биологиasta ratkaisujen löytämiseen evoluution, periytyminen ja mutaatioiden avulla (Stanley ja Miikkulainen 2002, s. 99-127). Periaate on yksinkertainen. Oletetaan eliö, millä on kromosomeja jotka vaikuttavat sen toimintoihin ja sopivuusfunktio tai pisteyttäjä, joka kertoo miten hyvin eliö on pärjännyt. Algoritmi toimii luomalla joukon eliöitä satunnaisilla kromosomeilla ja seuraamalla niitä. Tämän jälkeen jotain valintamenetelmää hyödyntäen osa eliöistä uusitaan. Valintamenetelmää valittaessa täytyy ottaa huomioon lokaali maksimi, jolla tarkoitetaan suurinta pistemäärää jonka eliö sen hetkisillä kromosomeilla voi saada (Mathew 2012). Tämä pistemäärä ei välttämättä ole suurin mahdollinen mihin eliö pystyy, minkä takia eri valintamenettelyjä kannattaa kokeilla.

Yksi valintamenettely voisi olla parhaimpien valitseminen, missä valitaan aina parhaat pisteet saaneet eliöt ja loput uusitaan täysin. Tämä valintamenettely on erittäin altis lokaalille maksimille. Yksi tapa välttää lokaali maksimi on käyttää mutaatiota, jossa valitaan parhaat pisteet saavuttaneet eliöt ja niille tehdään satunnaisesti pieniä muutoksia kromosomeihin (Mathew 2012).

Esimerkiksi halutaan luoda botti, joka osaa selvittää tasohyppelypelissä tason. Botilla on sensoreita, jotka kertovat onko edessä ilmaa, seinä vai vihollinen. Sopivuusfunktio on matka maaliin. Kun botti epäonnistuu, se muistaa miten hyvin se pärjasi sopivuusfunktion avulla ja yrittää päästä pidemmälle aloittamalla alusta pienin muutoksin, esimerkiksi jos vihollinen tai seinä on edessä, niin kannattaa hypätä. Tällainen käytös olisi mahdollista ohjelmoida suoraan, mutta tarkoituksena on että botti löytää vastaavia ratkaisuja itse (Baldominos ym. 2015).

2.4 Vahvistettu oppiminen

Vahvistetussa oppimisessa (engl. *Reinforcement Learning*) luodaan agentti, joka osaa havaita ympäristöään ja saada palautetta siltä (Sutton ja Barto 2018). Sen sijaan, että meillä on haluttu syöte ja tulos, vahvistettu oppiminen on sopiva tehtäviin, millä ei ole mitään yhtä tiettyä ratkaisua, kuten pelitestaaminen (Gordillo ym. 2021). Vahvistettu oppiminen koostuu melkein samoista osista kuin geneettiset algoritmit, jolla tarkoitetaan käytäntöjä miten se reagoi ulkomaailmaan, sopivuusfunktioita joka kertoo miten hyvin se pärjää ja arvofunktiota joka vertailee miten ratkaisut vaikuttavat pidemmällä aikavälillä. Toisin kuin geneettinen algoritmi, jota muutetaan sopivammaksi käytön jälkeen valintamenettelyllä, vahvistettua oppimista voidaan kouluttaa ajonaikana, jolloin se oppii suoraan sen vaikutuksista ympäristöön (Sutton ja Barto 2018).

3 Tilastotieteellisten menetelmien hyödyntäminen pelikehityksessä

Sen sijaan että pohditaan mitä aaltofunktion romahduksen käytettäväksi luodaan, yksi tapa tutkia sen käytettävyyttä on käyttää sitä jo olemassaolevassa pelissä. Mawhorter ym. 2021 käyttää tutkimuksessaan Super Nintendolle julkaistua Super Metroid peliä esimerkissään siten, että peli toimii myös alkuperäisellä alustalla. Tulokset osoittavat että menetelmä saa luotua alkuperäisiltä vaikuttavia tasoja, mutta se ei huomioi pelattavuutta asettamalla vääränlaisia palikoita jotka vaikuttavat esimerkiksi kameran liikkumiseen tai käyttämällä salaseiniksi tarkoitettuja palikoita joiden läpi voi kulkea kävelypolkuna. Korjaamalla virheet menetelmää voidaan käyttää tasojen satunnaistamiseen, joka antaa syitä palata takaisin kyseiseen peliin.

Aaltofunktion romahdus voidaan yhdistää myös erilaisiin menetelmiin, kuten kasvaviin neuroverkkoihin. Nordvig Møllerin, Billeskovin ja Palamasin (2020) tutkimuksessa pyydettiin ihmisiä vertailemaan pelkällä aaltofunktion romahduksella luotuja kartoja niihin, jotka oli luotu sen lisäksi käyttäen kasvavia verkkoja. Tuloksissa yhdistelmän luodut kartat olivat helpompia hahmottaa. Hyödyntämällä näitä menetelmiä kehittäjä voi luoda loputtomasti samantyyllisiä kartoja ja valikoida sieltä mieleisensä.

Freiknecht 2021 käyttää proseduraalista generointia realistisen kaupunkien luomiseen ajamisen harjoitteluun autokouluissa siten, että kuski voi ajaa pitkiä matkoja näkemättä samoja teitä. Toiseksi generointia käytetään rakennusten luomiseen todellisuuteen ja käytettävyyteen keskittyen. Käyttämällä menetelmää kartojen luomiseen esimerkiksi ensimmäisen persoonan ammutapeleissa (engl. *First Person Shooter*) pystytään luomaan osittain muuttuvia kartoja jotka haastavat pelaajaa totutteleamalla uuteen ympäristöön.

Geneettisiä algoritmeja voidaan käyttää pelin tekoälyyn vastustajana suoraan, mutta sen lisäksi sitä voidaan käyttää myös pelin tasapainottamiseen. Antamalla algoritmin pelata itseään tai tutkijaa vastaan ja tutkimalla pelistä saatuja tilastoja peliin voidaan tehdä tarvittavia tasapainomuutoksia (Wong ja Fang 2012, s. 145-159). Antamalla tekoälylle vapaat kädet, se voi keksiä erikoisempia ei niin optimaalisia taktiikoita ja ratkaisuja jotka saattavat yllättää ja haastaa pelaajaa.

Gisslén ym. 2021 käyttää tutkimuksessaan vahvistettua oppimista kouluttaakseen sekä ratkaisijan että generaattorin ralli- ja tasohyppelypeleihin. Tutkimus näyttää, että proseduraalisesti luodussa ympäristössä koulutetut ratkaisijat pärjäsivät paremmin kuin kiinteissä tasoissa koulutetut. Tasohyppelypeleiden ratkaisijat hyötyivät enemmän monipuolisista ympäristöistä, sillä ajamisen säännöt voidaan yleistää, kun taas tasohyppelypeleissä voi olla enemmän erilaisia tilanteita (Gisslén ym. 2021). Menetelmää voidaan käyttää myös kolmiulotteisen kartan tutkimiseen pelaajan jumittumisen estämiseksi. Gordillo ym. (2021) menetelmässä käytetään hupia ja uteliaisuutta mittareina agenteille. Tutkimalla agenttien liikeratoja kehittäjä voi tehdä muutoksia karttaan jotka estävät pelaajahahmon jumittumisen, sekä muokata kohtia joihin pelaajan ei kuuluisi päästä saavuttamattomaksi (Gordillo ym. 2021). Pelien tutkiminen on nautittavampaa, kun estetään pelaajahahmon tahallinen ja/tai tahaton jumittuminen.

4 Yhteenveto

Tässä tutkielmassa käsiteltiin erilaisia menetelmiä mitä pelikehittäjät voivat hyödyntää osana pelinkehitystä. Lisäämällä automatisoituja luomismenetelmiä pelikehittäjä voi keskittyä enemmän osa-alueisiin, joihin hän itse haluaa panostaa. Menetelmiä voidaan käyttää joko osana pelinä dynaamisten karttojen luomiseen tai kehittäjän työkaluna erilaisten tulosten valikoimiseen.

Aaltofunktion romahduksella pystytään luomaan pienistä esimerkeistä suurempia sääntöjen mukaisia kokonaisuuksia. Kerrostamalla romahduksia pystytään lisäämään yksityiskohtia ja rajaamaan alueita samanlaisuuden rikkomiseksi.

Proseduraalista generointia voidaan hyödyntää niillä osa-alueilla, joilla yhtenäisyydellä ei ole väliä. Menetelmälle annetaan paikka mistä aloittaa ja loput tapahtuu automaattisesti.

Geneettisiä algoritmeja ja vahvistettua oppimista voidaan käyttää pelillisesti kehittyvinä haastajina pelaajalle tai pelien testaamiseen luomalla testiaineistoa.

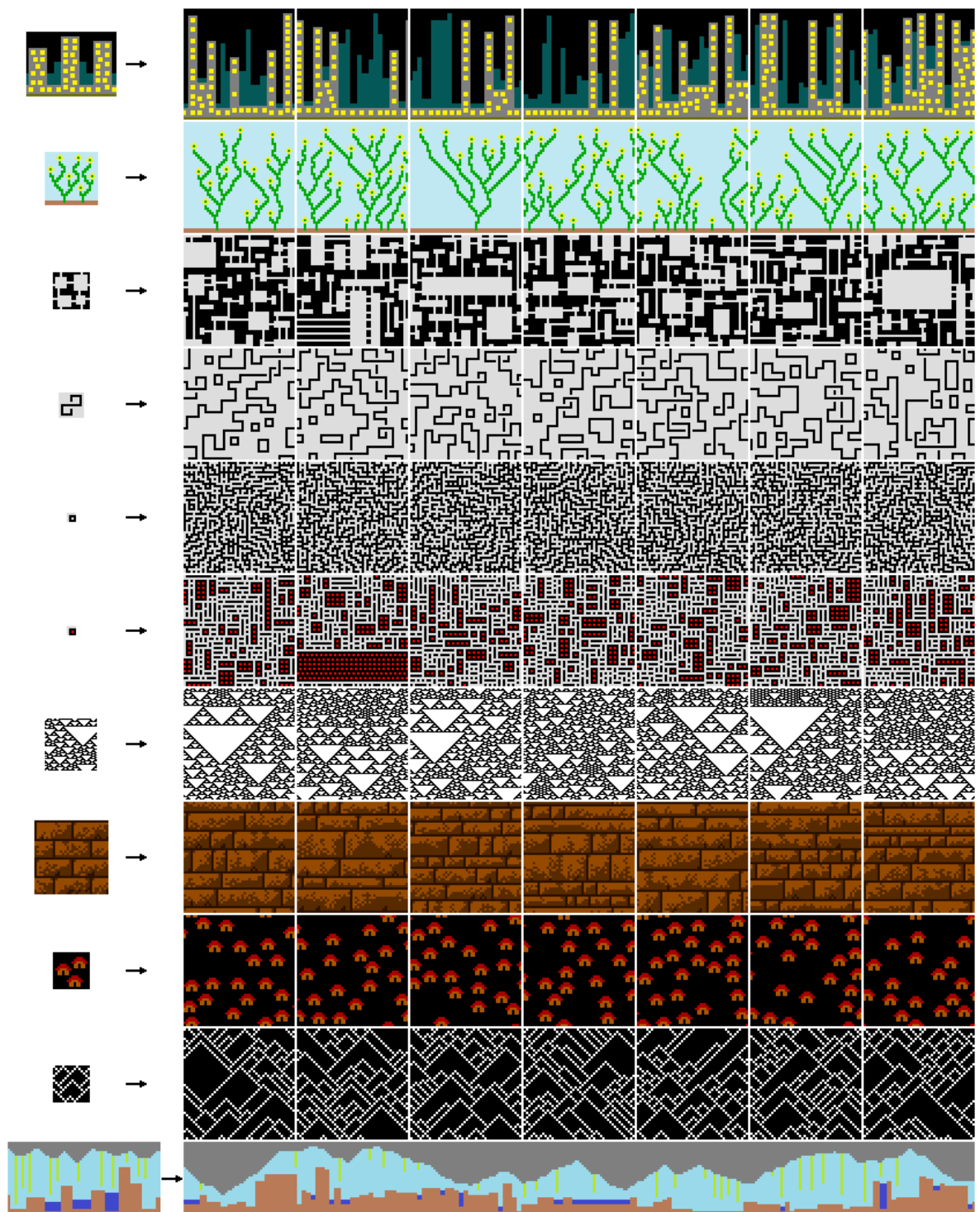
Lähteet

- Agis, Ramiro A, Andrea Cohen ja Diego C Martínez. 2015. “Argumentative AI Director Using Defeasible Logic Programming”. Teoksessa *Computer Games*, 96–111. Springer.
- Baldominos, Alejandro, Yago Saez, Gustavo Recio ja Javier Calle. 2015. “Learning levels of mario ai using genetic algorithms”. Teoksessa *Conference of the Spanish Association for Artificial Intelligence*, 267–277. Springer.
- Bopp, Julia Ayumi, Elisa D Mekler ja Klaus Opwis. 2016. “Negative emotion, positive experience? Emotionally moving moments in digital games”. Teoksessa *Proceedings of the 2016 CHI conference on human factors in computing systems*, 2996–3006.
- Bucklew, Brian. 2019. “Dungeon Generation via Wave Function Collapse”. Viitattu 10. maaliskuuta 2022. <https://youtu.be/fnFj3dOKcIQ>.
- Donald, Martin. 2020. “Superpositions, Sudoku, the Wave Function Collapse algorithm.” Viitattu 10. maaliskuuta 2022. <https://youtu.be/2SuvO4Gi7uY>.
- Frade, Miguel, Francisco Fernandez de Vega ja Carlos Cotta. 2012. “Automatic evolution of programs for procedural generation of terrains for video games”. *Soft Computing* 16 (11): 1893–1914.
- Freiknecht, Jonas. 2021. “Procedural content generation for games”.
- Galín, Eric, Adrien Peytavie, Nicolas Maréchal ja Eric Guérin. 2010. “Procedural generation of roads”. Teoksessa *Computer Graphics Forum*, 29:429–438. 2. Wiley Online Library.
- 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. IEEE.
- Gordillo, Camilo, Joakim Bergdahl, Konrad Tollmar ja Linus Gisslén. 2021. “Improving Playtesting Coverage via Curiosity Driven Reinforcement Learning Agents”. *CoRR* abs/2103.13798. <https://arxiv.org/abs/2103.13798>.

- Gumin, Maxim. 2021. “WaveFunctionCollapse”. Viitattu 10. maaliskuuta 2022. <https://github.com/mxgmn/WaveFunctionCollapse>.
- Kaplan, Andreas, ja Michael Haenlein. 2019. “Siri, Siri, in my hand: Who’s the fairest in the land? On the interpretations, illustrations, and implications of artificial intelligence”. *Business Horizons* 62 (1): 15–25. ISSN: 0007-6813. <https://doi.org/https://doi.org/10.1016/j.bushor.2018.08.004>.
- Karth, Isaac, ja Adam M Smith. 2019. “Addressing the fundamental tension of PCGML with discriminative learning”. Teoksessa *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 1–9.
- Karth, Isaac, ja Adam Marshall Smith. 2021. “WaveFunctionCollapse: Content Generation via Constraint Solving and Machine Learning”. *IEEE Transactions on Games*.
- Klimmt, Christoph, Christopher Blake, Dorothee Hefner, Peter Vorderer ja Christian Roth. 2009. “Player Performance, Satisfaction, and Video Game Enjoyment”. Teoksessa *Entertainment Computing – ICEC 2009*, toimittanut Stéphane Natkin ja Jérôme Dupire, 1–12. Berlin, Heidelberg: Springer Berlin Heidelberg. ISBN: 978-3-642-04052-8.
- Mathew, Tom V. 2012. “Genetic algorithm”. *Report submitted at IIT Bombay*.
- Mawhorter, Ross, Batu Aytemiz, Isaac Karth ja Adam Smith. 2021. “Content reinjection for super metroid”. Teoksessa *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, 17:172–178. 1.
- Nordvig Møller, Tobias, Jonas Billeskov ja George Palamas. 2020. “Expanding wave function collapse with growing grids for procedural map generation”. Teoksessa *International Conference on the Foundations of Digital Games*, 1–4.
- Parberry, Ian. 2014. “Designer worlds: Procedural generation of infinite terrain from real-world elevation data”. *Journal of Computer Graphics Techniques* 3 (1).
- Sandhu, Arunpreet, Zeyuan Chen ja Joshua McCoy. 2019. “Enhancing wave function collapse with design-level constraints”. Teoksessa *Proceedings of the 14th International Conference on the Foundations of Digital Games*, 1–9.

- Sotamaa, Olli. 2010. "When the game is not enough: Motivations and practices among computer game modding culture". *Games and Culture* 5 (3): 239–255.
- Stanley, Kenneth O, ja Risto Miikkulainen. 2002. "Evolving neural networks through augmenting topologies". *Evolutionary computation* 10 (2): 99–127.
- Sutton, Richard S, ja Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.
- Swink, Steve. 2008. *Game feel: a game designer's guide to virtual sensation*. CRC press.
- Wong, Sai-Keung, ja Shih-Wei Fang. 2012. "A study on genetic algorithm and neural network for mini-games". *Journal of information science and engineering* 28 (1): 145–159.

Liitteet



Kuvio 1. Esimerkkejä aaltofunktion romahduksen käytöstä (Gumin 2021)