

## Information technologies and multimedia Informacinės technologijos ir multimedija

### SKAITMENINIO KŪRYBIŠKUMO TAIKYMAS GENERUOJANT ŽAIDIMO OBJEKTUS

Aurimas PETROVAS\*

*Vilniaus Gedimino technikos universitetas, Vilnius, Lietuva*

Gauta 2019 m. gegužės 15 d.; priimta 2019 m. gegužės 27 d.

**Santrauka.** Viena sunkiausiai atliekamų užduočių mašinai yra kūrybiškumas. Skaitmeninis kūrybiškumas yra terminas, apibūdinantis kompiuterio galimybę atlikti kūrybines užduotis. Pagrindiniai metodai, generuojant turinį, yra šie: tiriamasis, sujungiantis ir transformacinis. Procedūrinio generavimo būdu naudojantis dirbtinio intelekto agentais, galima kurti turinį žaidimams. Straipsnyje aprašytas skaitmeninis kūrybiškumas, procedūrinis generavimas ir jo naudojimas žaidimuose. Buvo panaudotas, ištirtas ir išplėstas procedūrinis lygiu generatorius, siekiant padidinti jo išraišką ir įvairovę. Pagrindinės modifikacijos yra susietos su medžių ir aplinkos tekstūrų generavimu.

**Reikšminiai žodžiai:** skaitmeninis kūrybiškumas, procedūrinis generavimas, žaidimas, tekstūra, žaidimo lygis, žaidimų dizainas.

#### Įvadas

Iš praktinės pusės kūrybiškumas yra galimybė sukurti neįprastas idėjas, siekiant tam tikro tikslo. Labai nedaug naudingų sprendimų gali būti priimama pasikliaujant tik griežta logika. Prieš atliekant sprendimą visada reikia įvertinti esama situaciją ir veiksmus, kuriuos galima atlikti. Įprastai kūrybines užduotis atlieka žmonės. Dabartinės technologijos ir didelė kompiuterių skaičiuojamoji galia suteikia galimybę naudoti mašinų pagalbą kūrybiškiems sprendimams priimti. Kūrybiškumas įprastai yra didelių laiko išteklių reikalaujantis procesas. Mašinos yra atsparios nuovargiui, baimei ir stresui, todėl jas yra daug paprasčiau valdyti. Šiuo metu vienas iš efektyviausių modelių yra pagalbinis skaitmeninio kūrybiškumo modelis. Informacijos repozitoriją naudoja žmonės ir mašinos. Reikalavimus ir tikslus sugalvoja žmonės ir perduoda mašinai, kuri analizuoja turinį ir siūlo sprendimus. Skaitmeninis kūrybiškumas yra nauja ir eksperimentinė sfera, todėl ji vis dar neturi daug realaus pasaulio programėlių. Pagrindinės naudojimo sritys yra sprendimų pasirinkimas, algoritmų tobulinimas, skaitmeninio turinio generavimas ir marketingas. Kūrybiškumas skirstomas į tris dalis: tiriamasis, sujungiamasis ir transformacinis. Tiriamasis kūrybiškumas ieško naujovių ir vertės iš anksto nustatytoje paieškos erdvėje, sujungiamasis sujungia pažįstamas idėjas ir gauna naują rezultatą, o transformacinis pakeičia paieškos erdvės

rėžius ir prideda naujus koncepcinius modelius (Gustavi ir Jändel, 2013).

Vienas iš skaitmeninio kūrybiškumo metodų yra teorių sujungimas. Šis metodas veikia panašiai, kaip kad mąsto žmogus. Originaliai ši koncepcija buvo sukurta norint paaiškinti, kaip žmonės galvoja, tačiau dabar jos pritaikymas yra tyrinėjamas mašinų kūrybiškumo sferoje. Ši teorija sujungia skirtingas informacijos erdves tarpusavyje. Taip pat dažnai pasitelkiamas procedūrinis generavimas ir sujungiamos įvairios informacijos bei turinio detalės, pasitelkiant tam tikras taisykles ir atsitiktinius skaičius, tačiau įprastai kelios generacijos sukuria gan panašius rezultatus ir, siekiant gauti labiau išsiskiriantį rezultatą, kelių generacijų metu reikia modifikuoti patį algoritmą. Šio tyrimo metu buvo bandoma pritaikyti skaitmeninio kūrybiškumo metodus ir modifikuoti lygių procedūrinį generatorių pridėdam jam estetinių elementų, tekstūrų generavimą ir medžių generavimą. Generavimas atliekamas pasitelkiant dirbtinio intelekto agentus, kurie keliauja pirminiu žaidimo lygiu ir skaido jį į mažesnes dalis kuriant landšaftą ir generuojant tuščias erdves. Antra generavimo fazė užpildo karkasą papildomu turiniu, optimizuoja aplinką ištaisant klaidas, tokias kaip nevientisa erdvė arba nuo aplinkos atskirti kambariai. Panaudoti keturi turinio rinkiniai ir integruoti į procedūrinio generavimo algoritmą. Galutinis algoritmas sukuria realiam pasauliui artimesnį rezultatą.

\*Autorius susirašinėti. El. paštai: [glalrfl1@yahoo.com](mailto:glalrfl1@yahoo.com); [aurimasp@unity3d.com](mailto:aurimasp@unity3d.com)

Pagrindiniai elementai, kurie buvo pagerinti, yra išraiška, įvairovė ir kūrybiškumas. Šie elementai sunkiai įvertinami, nes jie subjektyvūs. Vienas iš būdų patikrinti jų lygį – naudoti objektyvių žmogiškų įvertinimų vidurkį.

## 1. Procedūrinis turinio generavimas žaidimuose

Procedūrinis turinio generavimas yra algoritminis žaidimo turinio kūrimas, naudojant kompiuterinę programinę įrangą. Dažniausiai ji būna automatinė arba nurodant nedidelį žmogaus ar dizainerio indėlį. Žaidimo turinį sudaro žaidimo sudėties elementai. Šie elementai gali būti lygiai, žemėlapiai, taisyklės, istorija, garsas, veikėjai ar bet kuris kitas žaidimo turinio vienetas (Togelius, Shaker ir Nelson, 2016).

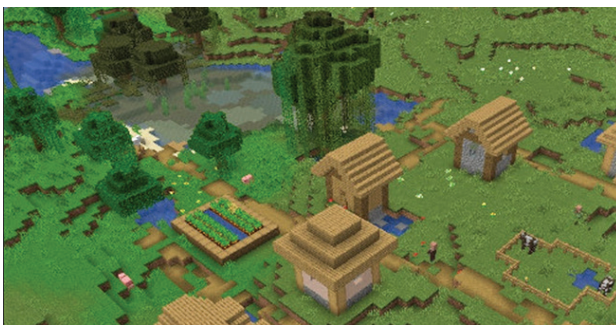
Žaidimų turinio generavimo pavyzdžiai:

- programinės įrangos įrankis, kuris generuoja lygius žaidimus;
- metaduomenų rinkimas ir įdomiausių žaidimo ginklų registravimas. Naujų ginklų generavimas atsižvelgiant į surinktus duomenis;
- programa, kuri generuoja nesudėtingas stalo žaidimų taisykles;
- vegetacijos populiovimo sistema žaidimo pasaulyje. Algoritmas, kuris generacijos plotą užpildo objektais, reprezentuojantis augalus ar medžius;
- žaidimo taisyklių balanso patarimo priemonė žaidimo turinio kūrėjui.

Pagrindinis automatinio turinio generavimo privalumas yra galimybė sumažinti žmogaus indėlio ir darbo kiekį kūrybiniame procese. Žmogiškieji išteklių yra brangūs ir lėtas (programos atžvilgiu) išteklius. Didžioji dalis žaidimų kūrėjų žaidimo kūrimo procese yra artistai. Procedūrinio generavimo metodai skirti turinio generavimo problemų sprendimams rasti.

Kokybės matavimo vienetai generuojant turinį:

- Greitis – laikas, kurį programa užtrunka generuodama tam tikrą turinio vieneta. Realus laiko generavimas dažniausiai užtrunka milisekundes, o išankstinis turinio generavimas gali užtrukti nuo sekundžių iki mėnesių, priklausomai nuo užduoties sudėtingumo.



1 paveikslas. „Minecraft“ žaidimo sugeneruotas pasaulis (Mojang, 2011)

Figure 1. World, generated in the Minecraft video game (Mojang, 2011)

- Patikimumas – žaidimo turiniui svarbių taisyklių laikymasis, artefaktų ar kitų netikslumų skaičius galutiniame turinyje. Žaidimo lygio generacija su nepasiekiamų išėjimų dažniu yra patikimumo kriterijaus pavyzdys.
- Kontrolės lygis – galimybė keisti žaidimo turinio generavimo taisykles.
- Išraiška ir įvairovė – turinio skirtumas kiekvienos generacijos metu. Vienoje spektro pusėje gali būti atsitiktinės spalvos pasirinkimas vienam konkrečiam elementui generacijoje, kitoje – atsitiktinių elementų sujungimas.
- Kūrybiškumas – sugeneruoto turinio panašumas į žmogaus sukurtą turinį.

Procedūrinis generavimas išskiria žaidimo turinį į atskirus elementus ir pritaiko generavimo algoritmus pavieniems elementams. Dažniausiai automatiškai sugeneruoto turinio kokybė nusileidžia žmogaus sukurtam turiniui. Programos sugeneruotas turinys dažniausiai nusileidžia savo patikimumu, nuspėjamumu, vizualia kokybe ir tikroviškumu. „Minecraft“ (Mojang, 2011) žaidime galima aptikti netikroviškų aplinkos junginių ar generacijos klaidų, tokių kaip skraidančios salos ar pastatų įėjimo užtvėrimas (1 pav.). Siekiant kuo labiau išvengti tokių neatitiktinių, reikia atsižvelgti į žaidimo mechanikas ir principus. Tarkime, jeigu žaidimo aplinka nesukuria tam tikro tikslo ir gali būti žaidėjo ignoruojama, ji nebus traktuojama kaip pozityvi patirtis žaidėjui. Ši problema nagrinėjama procedūrinio generavimo tyrimais. Turinio generavimas gali būti sujungtas su dizainerio pasirenkamomis žaidimo savybėmis (Karavolos, Bouwer ir Bidarra, 2015).

„Dwarf Quest“ (Wild Card Games, 2013) yra požemių tyrinėjimo žaidimas, kuriame žaidėjas tyrinėja atsitiktinai sugeneruotus požemius ir renka įvairius objektus. Kambariuose gali būti sugeneruotas vienas iš kelių elementų: spąstai, renkami objektai, priešininkai, durys, tiltai, raktai arba svirtys. Taip pat kambarys gali neturėti jokių objektų. Dizaineriui pasirinkus generacijos nustatymus, vieno mygtuko paspaudimu gali būti sukurtas labirintas. Smulkesni generavimo objektai yra sukuriami automatiškai, o dizaino kryptį nustato dizaineris. Toks pasirinkimas leidžia sudėtingesnes ir kūrybiškumo reikalaujančias užduotis atlikti žmogui. Dizaineris gali aprašyti tam tikrus susidūrimų scenarijus, kurie vėliau gali būti jungiami ir modifikuojami procedūrinio generavimo programos. Sugeneravus turinį, galima paprašyti dizainerio grįžtamojo ryšio, tokiu būdu pakeičiant algoritmą, kad nebūtų kartojamos buvusios klaidos. Tokiu principu sujungiami automatizuoto ir mechaninio generavimo privalumai, sukuriama interakcija tarp dizainerio ir programos, tačiau ji nepaliečia žaidėjo. Tolimesnis tyrimas gali prijungti žaidėją turinio tobulinimo procese. Tokiu būdu automatizaciją galima tobulinti atliekant žaidėjų testus ir galiausiai pasiekti geresnę žaidimo kokybę bei žaidimo populiarumą.

## 2. Generavimo agentai, kuriant trijų matmenų žaidimo lygius

Vienas iš metodų, skirtų procedūriniais lygiams generuoti, yra agentų lygių generatorius. Toks generatorius turi vieną arba daugiau tinklu keliaujančių agentų, kurie gravituoja ir užpildo žemėlapius turiniu. D. Karavolos (2017) generatoriaus pavyzdys yra atvirojo kodo ir turi nemažai galimybių varijuoti algoritmą. Taip pat yra kelios klaidų mažinimo patikros. Generavimui pasibaigus, atliekama paieška ir palyginamas kambarių vientisumas. Jeigu kuri nors generacijos dalis yra atskirta, tarp agentų sukuriama koridorius. Žemėlapių dydis, kambarių dydžiai, kambarių persikirtimai gali būti keičiami algoritmo viduje. Viena iš pavyzdinių generacijų naudoja du agentus, piešia žemėlapių dviem lygiais ir juos sujungia laiptais (2 pav.).

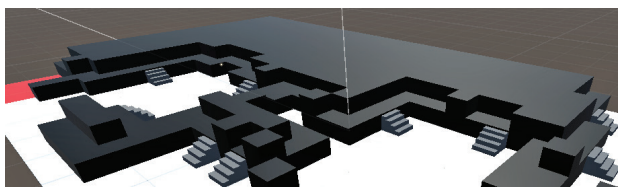
Logika aprašyta devyniomis klasėmis. „IntVector“ klasė aprašo vektorių struktūrą. „LDCell“ sukuria erdvinį elementą. „LDRoom“ išgravituoja erdvę aplink agentą. „LDRoom-Settings Settings“ pakeičia paviršiaus medžiagos savybes. „GridDirection“ aprašo vektorių kryptis, posūkius ir skirtingų duomenų konversiją. „DiggingAgent“ inicializuoja ir valdo agentą. „LevelDigger“ ir „MultiAgentDigger“ aprašo visą agentui suteiktą lygio generavimo logiką. „GameManagerLD“ inicializuoja pirminius duomenis ir valdo sceną.

Generatoriuje galima naudoti ir modifikuoti kambarių spalvų masyvą, labirinto dydį, agento darbo laiką, posūkių dažnį, kambarių dydžių režius, kambarių susipynimą, startines agentų pozicijas, pradžios ir pabaigos veiksmus, laiptų ir papildomų aukštų generavimo dažnius, agentų skaičių, pradinių kambarių dydžius ir pradinius kryptių vektorius.

Pagrindinis agento algoritmas inicializuoja duomenis, sugeneruoja pirminius kambarius ir tam tikram žingsnių kiekiui kartuoja pagrindinį generavimo ciklą. Šis ciklas stumdo agentus pusiau atsitiktinėmis kryptimis ir generuoja objektus. Agentui išgravitavus kambarius sukuriama papildomas lygio gylis. Algoritmas patikrina kambarių vientisumą, prideda laiptus ir sukuria antrą aukštą, kuris yra sumažinta pirmo aukšto dalis. Viskas paverčiama statišku elementu ir apskaičiuojamos tuščios erdvės. Tokiu būdu nupiešiamas galimos navigacijos tinklas, kuriuo remiantis galima sukurti automatiškai tinkle judančius veikėjus.

Alternatyvūs procedūrinio generavimo metodai yra:

- Atsitiktinės reikšmės ir interpoliacija. Generuojami objektai ir priskiriamos vidutinės reikšmės vidiniams



2 paveikslas. Procedūrinio generavimo agentais sugeneruotas žaidimo lygis (Karavolos, 2017)

Figure 2. Game level, generated using procedural generation agents (Karavolos, 2017)

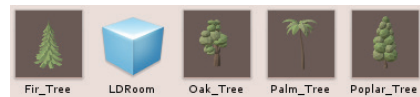
- objektams. Galutinis rezultatas turi mažiau klaidų ir objektai būna priklausomi vieni nuo kitų, tačiau kelių generacijų rezultatai mažiau skiriasi vienas nuo kito.
- Fraktalai. Algoritmas yra kartojamas skirtingais masteliais. Sukuriami natūralūs paviršiai, tačiau objektų yra daug daugiau.
- Paieškos metodas. Pasirenkami paieškos režiai ir, atliekant genetines funkcijas, yra kuriamos atsitiktinės generacijos. Procesas kartojamas, kol tikslo funkcija pradeda konverguoti.

## 3. Eksperimentinių bandymų aprašymas

Generatoriaus transformacija į žaidimui artimesnę aplinką. Pirmu žingsniu pakeisti laiptų objektai medžių objektais. Paveiksle nurodomi medžių objektų masyvo elementai (3 pav.). *Resources* direktorijoje pridėti medžių modeliai ir programos kode pakeistos objektų nuorodos. Pakeistas dangaus kubinis žemėlapis ir dangaus tekstūra. Generuojamiems landšafto objektams priskirtos tekstūros (DARTH\_ARTISAN, 2017). Paveiksle matomas generavimo agentas ir sugeneruotos tekstūros (4 pav.). Jos išskaidytos į tris skirtingas rūšis: atvirų erdvių, elevacijos ir uždarytų erdvių.

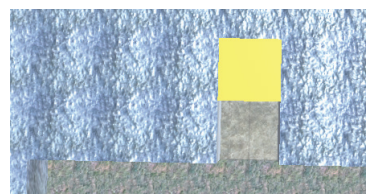
4 paveiksle matome scenos paviršių, landšafto sluoksnį ir generavimo agentą. Agentas naikina landšafto blokus ir palieka tuščias erdmes. Atliekant judėjimo funkcijas, agentas paviršių nuspalvina smėlio tekstūra. Kai kambario sukūrimo skaičiaus reikšmė viršija pasirinktą slenkstį, sukuriama kambarys ir nuspalvinamas naudojant žemės tekstūrą. Skaičiaus reikšmė yra nusprendžiama naudojant skaičių generatorių. Visiems nepaviršiniams žemės sluoksniams sukuriami kalnų tekstūra. Paveikslėlyje matome tris tekstūras.

Pritaikyta simuliacija medžiams generuoti. Du kartus išplėstas generavimo plotas. Medžių generavimo šansas padidintas 4,5 karto. Įdėtos keturios medžių rūšys. Panaudotas atsitiktinių skaičių generatorius naudojamų objektų tipui nustatyti. Skirtingos reikšmės sukuria skirtingus medžius. Agentų skaičius pakeistas nuo dviejų iki keturių. Kiekvienam agentui nustatomos pradinės kryptys,



3 paveikslas. Medžių objektai

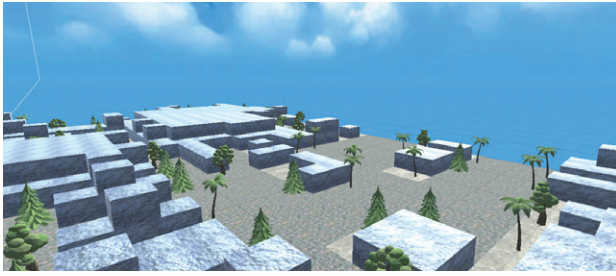
Figure 3. Tree objects



4 paveikslas. Tekstūrų tipai ir procedūrinio generavimo agentai

Figure 4. Texture types and procedural generation agents





5 paveikslas. Atnaujintas procedūrinio generavimo variklis

Figure 5. Updated procedural generation engine

pirminių kambarių dydis ir koordinatės. Agentai pastatyti keturiuose scenos kampuose. Medžių generacija nevykdoma sujungimų laukeliuose tarp agentų sukurtų kambarių, todėl nėra užtveriamas kelias.

Papildomi pakeitimai suteikia scenai įvairovės ir kūrybiškumo. Paveikslėlyje matoma galutinė generavimo scena (5 pav.). Dangaus tekstūrą sudaro kubinis žemėlapis. Kubo sienos yra deformuotos ir jų vidinė dalis padengta tekstūra. Apatinėje paveikslėlio dalyje matome agento suformuotą landšaftą, jo paviršių, padengtą tekstūromis, ir objektais užpildytą turinį. Atnaujinti ir papildyti elementai pagerina šiuos rodiklius: išraišką, įvairovę ir kūrybiškumą. Generatoriaus bazę galima plėsti toliau. Potencialūs pakeitimai gali būti papildomų elevacijos aukštų pridėjimas, objektų geometrinių savybių pakeitimas, papildomas objektų generavimas. Taip pat galima keisti algoritmo intervencijos ir pradinių duomenų santykį, siekiant keisti objektų apstatymo ir graviravimo lygį. Pakeitimai buvo atliekami naudojant „Unity“ žaidimų variklį ir programuojama C# programavimo kalba.

Eksperto tikslas – sukurti žaidimo aplinkos generatorių, kuris gebėtų kurti skirtingas aplinkas kelių generacijų metu. Modifikacijų metu buvo išplėstos algoritmo turinio kiekio ir įvairovės generavimo galimybės. Algoritmas pritaikytas saugoti daugiau negu vieną komponentą kiekvienam objektui. Naujų komponentų masyvus sudaro medžiai ir paviršių tekstūros. Siekiant gauti įvairesnius landšafto modelius, išplėsti generatoriaus sprendimų rėžiai. Generatorius sukuria daugiau estetinių elementų ir taip turinį padaro artimesnį kūrybinio proceso rezultatui.

## Išvados

Tyrimo metu buvo atliekamas automatinis turinio kūrimas naudojant generavimo agentų algoritmą.

1. Du kartus padidintas simuliacijos dydis ir pridėti du papildomi procedūrinio generavimo agentai. Sugeneruoto turinio kiekis per tą patį laiką išlaikytas

toks pats, tačiau 100 % didesnis skaičiavimo išteklių kiekis generavimo metu.

2. Išplėstas ir modifikuotas laiptų generavimo algoritmas. Sukurtas 4 skirtingų medžių masyvas ir jo elementais užpildoma tuščio ploto erdvė (5 pav.).
3. Tekstūrų generavimas pritaikytas juodiems statybiniams blokams (2 pav.). Generatorius sukuria žolės, kalnų ir smėlio tekstūras (5 pav.).
4. Galutinis algoritmas sukuria realiam pasauliui artimesnį rezultatą (5 pav.). Pagrindiniai elementai, kurie buvo pagerinti, yra išraiška, įvairovė ir kūrybiškumas.

## Literatūra

- DARTH\_ARTISAN. (2017). *Free Trees*; G.E. TEAM. 2018. *Fantasy Skybox*; LOWLYPOLY. 2017. *Hand Painted Grass Texture*; GLOVER GAMES. 2019. *GloverGames Free Ground Textures*. Unity Asset Store. Retrieved from <https://assetstore.unity.com/>
- Gustavi, T., & Jändel, M. (2013). *Computational creativity: novel technologies for creative decision making*. FOI.
- Karavolos, D. (2017). *Agent based level generator*. Retrieved from <https://autogamedesign.eu/>
- Karavolos, D., Bouwer, A., & Bidarra, R. (2015). Mixed-initiative design of game levels: integrating mission and space into level generation. *Paper presented at the Proceedings of the International Conference on the Foundations of Digital Games*.
- Mojang. (2011). *Minecraft*. Retrieved from <https://minecraft.net>
- Togelius, J., Shaker, N., & Nelson, M. J. (2016). *Procedural content generation in games: a textbook and an overview of current research* (Chapter 1). Springer.
- Wild Card Games. (2013). *Dwarf Quest*. Retrieved from <https://www.dwarfquestgame.com/>

## COMPUTATION CREATIVITY USING PROCEDURAL GENERATION FOR GAME OBJECT GENERATION

A. Petrovas

Abstract

One of the hardest task for a machine is creativity. Computational creativity defines creative task completion for a machine. Three main creative content generation methods are: exploratory, combinatorial and transformational. Video game content can be generated using procedural generation. Computational creativity, procedural generation, and application are explained in this paper. Procedural level generator is used as a base and additional features are built on top of it. The main goal of this research and modification is to increase application creative value, variety and expression. Additional functionality consists of tree and texture generation.

**Keywords:** computational creativity, procedural generation, game, texture, game level, game design.