VILNIUS GEDIMINAS TECHNICAL UNIVERSITY

Aurimas PETROVAS

COMPUTATIONAL CREATIVITY IN VIDEO GAME SCENE GENERATION BY GENETIC MULTI-CRITERIA DECISION-MAKING METHODS

DOCTORAL DISSERTATION

TECHNOLOGICAL SCIENCES
INFORMATICS ENGINEERING (T 007)

Vilnius, 2024

The doctoral dissertation was prepared at Vilnius Gediminas Technical University in 2019–2024.

Supervisor

Prof. Dr Habil. Romualdas BAUŠYS (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

The Dissertation Defence Council of the Scientific Field of Informatics Engineering of Vilnius Gediminas Technical University:

Chairman

Prof. Dr Dalius MAŽEIKA (Vilnius Gediminas Technical University, Informatics Engineering – T 007).

Members:

Prof. Dr Habil. Gintautas DZEMYDA (Vilnius University, Informatics Engineering – T 007),
Dr Morteza YAZDANI (Valencian International University, Spain, Informatics Engineering – T 007),
Prof. Dr Arnas KAČENIAUSKAS (Vilnius Gediminas Technical University, Informatics Engineering – T 007),
Prof. Dr Rytis MASKELIŪNAS (Kaunas University of Technology, Informatics Engineering – T 007).

The dissertation will be defended at the public meeting of the Dissertation Defence Council of Informatics Engineering in the SRA-I Meeting Hall of Vilnius Gediminas Technical University at **10 a.m. on 2 May 2024**.

Address: Saulėtekio al. 11, LT-10223 Vilnius, Lithuania.
Tel.: +370 5 274 4956; fax +370 5 270 0112; e-mail: doktor@vilniustech.lt

A notification on the intended defence of the dissertation was sent on 29 March 2024.
A copy of the doctoral dissertation is available for review at the Vilnius Gediminas Technical University repository (https://etalpykla.vilniustech.lt), at the library of Vilnius Gediminas Technical University (Saulėtekio al. 14, LT-10223 Vilnius, Lithuania), and at the library of Kaunas University of Technology (K. Donelaičio st. 20, LT-44239 Kaunas, Lithuania).

Vilnius Gediminas Technical University book No 2024-002-M

doi:10.20334/2024-002-M

aurimas.petrovas@vilniustech.lt

VILNIAUS GEDIMINO TECHNIKOS UNIVERSITETAS

Aurimas PETROVAS

SKAITMENINIS KŪRYBIŠKUMAS ŽAIDIMŲ SCENOS GENERAVIME TAIKANT GENETINIUS DAUGIAKRITERIŲ SPRENDIMŲ PRIĖMIMO METODUS

DAKTARO DISERTACIJA

TECHNOLOGIJOS MOKSLAI,
INFORMATIKOS INŽINERIJA (T 007)

Vilnius, 2024

Disertacija rengta 2019–2024 metais Vilniaus Gedimino technikos universitete.

Vadovas

prof. habil. dr. Romualdas BAUŠYS (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007).

Vilniaus Gedimino technikos universiteto Informatikos inžinerijos mokslo krypties disertacijos gynimo taryba:

Pirmininkas

prof. dr. Dalius MAŽEIKA (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007).

Nariai:

prof. habil. dr. Gintautas DZEMYDA (Vilniaus universitetas, informatikos inžinerija – T 007),
dr. Morteza YAZDANI (Valensijos tarptautinis universitetas, Ispanija, informatikos inžinerija – T 007),

prof. dr. Arnas KAČENIAUSKAS (Vilniaus Gedimino technikos universitetas, informatikos inžinerija – T 007),
prof. dr. Rytis MASKELIŪNAS (Kauno technologijos universitetas, informatikos inžinerija – T 007).

Disertacija bus ginama viešame Informatikos inžinerijos mokslo krypties disertacijos gynimo tarybos posėdyje **2024 m. gegužės 2 d. 10 val.** Vilniaus Gedimino technikos universiteto SRA-I posėdžių salėje.

Adresas: Saulėtekio al. 11, LT-10223 Vilnius, Lietuva.
Tel. (8 5) 274 4956; faksas (8 5) 270 0112; el. paštas doktor@vilniustech.lt

Pranešimai apie numatomą ginti disertaciją išsiųsti 2024 m. kovo 29 d.

Disertaciją galima peržiūrėti Vilniaus Gedimino technikos universiteto talpykloje https://etalpykla.vilniustech.lt, Vilniaus Gedimino technikos universiteto bibliotekoje (Saulėtekio al. 14, LT-10223 Vilnius, Lietuva) ir Kauno technologijos universiteto bibliotekoje (K. Donelaičio g. 20, LT-44239 Kaunas, Lietuva).

Abstract

Procedural generation in video games refers to the automatic creation of game content, such as levels, environments, and characters, through algorithmic processes rather than manual design. This approach enables developers to achieve diverse video game scene patterns, enhancing player experiences. Multi-criteria decision-making methods are employed in procedural generation to balance multiple objectives, such as gameplay variety, aesthetics, and a fluid combination of abstract video game-level features. Neutrosophic sets, a mathematical framework dealing with indeterminate and uncertain information, offer a way to handle ambiguous elements in procedural generation, adding a unique creative dimension to the process.

The dissertation consists of an introduction, three main chapters, general conclusions, and a list of references. The first chapter performs a literature review on creative procedural generation methods for video games and formulates the dissertation's objectives. The second chapter proposes a novel approach for procedural video game scene generation, which uses genetic algorithms, employs MCDM methods for fitness function, and models creativity-based criteria. Proposed methods include WASPAS-SVNS and CoCoSo fitness functions for the genetic algorithm, regional object morph algorithm and modelling of Gestalt design principles for the fitness functions.

The third chapter evaluates, explores and presents the generated result artefacts of the proposed creative procedural generation method. The case study results show how the algorithm can increase the creative value of the generated artefacts and reduce the time for manual decision-making of creative tasks. The method reduces the number of repetitive game scene patterns and generates a significant number of unique game object layout patterns. MCDM methods and neutrosophic sets ensure the combination of fluid-conflicting criteria. Generated artefact features are easy to distinguish and do not make generated iterations chaotic by not employing every criterion identically in a single algorithm run. One generated game scene can employ more than one visual design pattern if there is a possibility in the initial genetic algorithm seed and random mutation direction. When combined for different design rules, cellular automata-based rules with local neighbourhood check agents can generate varied video game scene patterns relatively quickly. The final algorithm employs an above-average ability to generate creative value.

Reziumė

Procedūrinis generavimas žaidimuose apima automatinį žaidimo turinio, tokio kaip lygiai, aplinkos ir veikėjai, kūrimą algoritminiais procesais. Tokia strategija leidžia žaidimų kūrėjams padidinti įvairovę generuojant žaidimų turinį ir taip pagerinti žaidėjų patirtį. Procedūrinio generavimo metu gali būti taikomi daugiakriteriai sprendimų priėmimo metodai, siekiant subalansuoti skirtingus tikslus, tokius kaip žaidimo įvairovė, estetika ir abstrakčių žaidimo lygio savybių sklandus sujungimas. Neutrosofinės aibės padeda apdoroti neapibrėžtą ir netikslią informaciją procedūrinio generavimo procese ir padeda lengviau apibrėžti kūrybiškumo elementus. Disertacija susideda iš įvado, trijų pagrindinių skyrių, apibendrinamųjų išvadų ir literatūros šaltinių sąrašo. Pirmajame skyriuje atliekama literatūros apžvalga, kurioje aprašomas kūrybinių procedūrinio generavimo metodų taikymas žaidimuose, ir suformuluojami disertacijos tikslai. Antrajame skyriuje pasiūlomas naujas procedūrinio žaidimo scenų generavimo būdas, pasitelkiant genetinius algoritmus, daugiakriterius sprendimų priėmimo metodus ir kūrybiškumu grindžiamų kriterijų modeliavimą. Siūlomi metodai apima WASPAS-SVNS ir CoCoSo funkcijas genetiniams algoritmams, regioninių objektų morfavimo algoritmus ir geštalto dizaino principų modeliavimą tikslo funkcijose.

Trečiajame skyriuje įvertinami ir pristatomi sukurtų rezultatų artefaktai, gaunami taikant pasiūlytą kūrybinio procedūrinio generavimo metodą. Atliktas tyrimas rodo, kaip algoritmas gali padidinti sukurtų artefaktų kūrybinę vertę ir sumažinti rankinio kūrybinių užduočių sprendimo laiką. Šis metodas sumažina pasikartojančių žaidimo scenų skaičių ir generuoja didelį unikalių žaidimo objektų išdėstymo modelių kiekį. Daugiakriteriai sprendimų priėmimo metodai ir neutrosofinės aibės užtikrina sklandų konfliktinių kriterijų derinimą. Sukurti artefaktai būna lengvai atpažįstami ir nesukelia chaotiškų algoritmo rezultatų. Kriterijų elementų balansas būna nevienodas kelių algoritmo aktyvavimų metu. Vienoje algoritmo iteracijoje galima pamatyti daugiau negu vieną vizualaus dizaino elementą, jei tai leidžia pirminiai atsitiktiniai genetinio algoritmo duomenys. Ląstelinio automato pagrindu paremtos taisyklės su lokalių kaimynų patikros agentais, kai derinamos skirtingos dizaino taisyklės, gali gana greitai generuoti skirtingus žaidimo scenų modelius. Galutinis algoritmas pasižymi didesniu negu vidutinis kūrybinės vertės generavimo gebėjimu.

Notations

Symbols

m – aesthetic symmetry (liet. *Estetinė simetrija*);
x – grid length (liet. *Tinklelio ilgis*);
x_i – grid x dimension element (liet. *Tinklelio x dimensijos elementas*);
y – grid height (liet. *Tinklelio aukštis*);
y_i – grid y dimension element (liet. *Tinklelio y dimensijos elementas*);
s – binary value representing symmetric objects (liet. *Binarinė reikšmė nurodanti simetrišką elementą*);
e – empty space ratio (liet. *Tuščios erdvės santykis*);
t – the sum of empty objects (liet. *Tuščių objektų suma*);
d – player-exit distance (liet. *Atstumas tarp žaidėjo ir išėjimo*);
z – safe zone (liet. *Saugi zona*);
X – a set of objects (liet. *Objektų rinkinys*);
v – criteria value (liet. *Kriterijaus reikšmė*);
v_{max} – highest possible criteria value (liet. *Aukščiausia įmanoma kriterijaus reikšmė*);
N – neutrosophic number (liet. *Neutrosofinis skaičius*);
S – crisp number (liet. *Skaliarinis skaičius*);
Q – criteria representation in the WASPAS algorithm (liet. *Kriterijaus reprezentacija WASPAS algoritme*);
t – truth (liet. *Tiesa*);
i – intermediacy (liet. *Neapibrėžtumas*);

f – falsehood (liet. *Netiesa*);

R – weighted comparability (liet. *Svorio koeficientas*);

P – power weight of the comparability sequences (liet. *Laipsniu pakelto svorio koeficiento seka*)

λ – weight constant (liet. *Svorio konstanta*);

k_{ia} – arithmetic mean of the sums of WSM and WPM (liet. *Aritmetinis WSM ir WPM sumų vidurkis*);

k_{ib} – the sum of the relative scores of WSM and WPM compared to the best (liet. *Reliatyvių WSM ir WPM verčių suma lyginant su aukščiausia verte);*

k_{ic} – balanced compromise of the WSM and WPM model scores (liet. *WSM ir WPM verčių subalansuotas rezultatas*);

k_i – ranking of alternatives (liet. *Alternatyvų rikiavimas*);

l – number of generations for the genetic algorithm (liet. *Genetinio algoritmo generacijų skaičius*);

A – cellular automata agents vision centre (liet. *Ląstelio automato agentų matymo centras*);

B – cellular automata agents neighbour (liet. *Ląstelinio automato agento kaimynas*);

s_m – general normalised Gestalt criteria (liet. *Standartinis ir normalizuotas Geštalto kriterijus*);

r – agent neighbours (liet. *Agento kaimynai*)

t_1 – similarity (liet. *Panašumas*);

t_2 – proximity (liet. *Artumas*);

t_3 – continuity (liet. *Tęstinumas*);

t_4 – focal points (liet. *Židinio taškas*);

t_5 – common regions (liet. *Bendras regionas*);

s_i – focal aesthetic criteria value (liet. *Lokalaus estetinio kriterijaus reikšmė*);

v_i – global aesthetic criteria value (liet. *Globalaus estetinio kriterijaus reikšmė*);

f_1 – functional criteria value (liet. *Funkcinio kriterijaus reikšmė*);

c_1 – binary functional criteria value (liet. *Binarinio funkcinio kriterijaus reikšmė*);

w – single criteria weight (liet. *Vieno kriterijaus svoris*);

R – random number (liet. *Atsitiktinis skaičius*);

N_1 – noise (liet. *Triukšmas*).

Abbreviations

2D – 2-dimensional (liet. *2 dimensijų*);

3D – 3-dimensional (liet. *3 dimensijų*);

CoCoSo – Combined Compromise Solution;

GHz – Gigahertz (liet. *Gigahercas);*

MCDM – Multi-Criteria Decision-Making (liet. *Daugiakriteris sprendimų priėmimo metodas*);

CPU – Central Processing Unit (liet. *Procesorius*);

PCG – Procedural Generation (liet. *Procedūrinis generavimas*);

PCGML – Procedural Content Generation via Machine Learning (liet. *Procedūrinis turinioo generavimas pasitelkiant mašininį mokymą*);

SVNS – Single-Valued Neutrosophic Sets (liet. *Vienos vertės neutrosofinė aibė*);

WASPAS – Weighted Aggregated Sum Product Assessment method;

WASPAS-SVNS – the WASPAS method, modelled under the Single-Valued Neutrosophic Set environment;

WPM – Weighted Product Model (liet. *Svertinis daugybos modelis*);

WSM – Weighted Sum Model (liet. *Svertinis sumos modelis*).

Contents

Introduction

Problem Formulation

The popularity of creativity modelling is on the rise, with its application spreading across various fields. Despite its popularity, there is a lack of a universal definition of creativity, which varies among different fields. The abstract nature of creativity further adds to the challenge of interpretation. Creativity generally encompasses diverse creative intelligence abilities, making simulation a complex technical task (Colton and Wiggins, 2012; Wiggins, 2006). It indicates that human creative abilities are not easily understandable and replicable in a model. The perspective of creativity modelling seeks to understand and extract creative traits from existing knowledge, works, or the environment. Valuable information about creativity is implicitly encoded in the surrounding world. Therefore, grasping how creativity is defined is crucial for effectively generating creative systems.

Computational creativity is an area of artificial intelligence that explores the development of computational systems capable of exhibiting creative behaviour. One aspect of the creative process involves combining goals of varying natures. In modern applications of computational creativity, generative art takes centre stage. Generative art refers to artefacts created through algorithms which resemble artistic motivations (Boden and Edmonds, 2009). The approach is primarily used to produce various media forms. Examples of generated content include sound or music, images derived from text or specialised pixel art sheets, and text generation. Generative art finds particular relevance in the digital world and game-level design, where procedural generation is frequently employed.

Search-based procedural generation in video games involves using a fitness function to evaluate and rate the generated assets or compositions. The evaluation determines how well the generated artefacts perform based on the fitness function, making the construction of the fitness function a crucial factor in the success of the generator. The advantage of using a search algorithm lies in its ability to find existing solutions consistently based on the fitness function. Two main types of fitness criteria are used: aesthetic and functional. Aesthetic criteria focus on the visual aspect, ensuring the generated level has a pleasing appearance. On the other hand, functional criteria assess adherence to rules, such as the existence of key object elements or the proper usability of the artefacts for their intended purpose. One of the challenges faced in the process is the seamless integration of both criteria types into the final result. Sometimes, the aesthetic and functional criteria may conflict, making it difficult to strike the right balance (Han et al., 2021).

Another challenge in modelling creative elements arises from their abstract nature, which makes it difficult to define them precisely for mathematical algorithms. The research aims to address this issue by providing exact definitions for algorithm elements. Additionally, to ensure that the generated artefacts possess high creative value, methods must be devised to incorporate high-level aesthetic concepts and introduce noticeable differences between iterations of the same algorithm. Multi-criteria Decision Making (MCDM) is employed as one of the solutions to effectively combine the criteria of the fitness function. MCDM helps model strategies for selecting alternatives from a finite pool of possible solutions (Zavadskas et al., 2014). Genetic algorithm operators can be optimised by applying these methods to the genetic algorithm's fitness function. Non-determinism can also increase in some variations of MCDM methods, such as using fuzzy sets instead of crisp numbers. Overall, the primary objective of the dissertation is to enhance the creativity aspect of game-level object layout procedural generation using genetic algorithms.

Relevance of the Dissertation

Automation has significantly enhanced the efficiency and quality of people's lives (Filip, 2021). However, creativity and intelligent problem-solving have traditionally been handled by humans. Researchers have recently devoted considerable attention to mathematically modelling creativity problems. Using algorithmic solutions for creative tasks is becoming more popular, with creativity traits typically involving novelty and the value of the creations (Pichot et al., 2022). Multimedia creation is a resource-intensive process, often requiring substantial human effort. Hence, the potential to automate more complex creative tasks could greatly accelerate digital content generation. As a diverse multimedia branch, video games heavily rely on computer capabilities, resulting in a wide variety of creative content types. However, traditional procedural generation is often associated with random approaches, leading to less diverse and authentic computer-generated content. The precise definition of creativity and its evaluation remains unclear in the branches of computer science. In parallel with assessing creative value, it is essential to comprehend what constitutes creativity and how it can be effectively modelled.

Research Object

The object of the dissertation involves employing multi-criteria decision-making to create object layouts within genetic procedural game scenes, with a particular emphasis on the fitness function.

Aim of the Dissertation

The main goal is to save game design development time by proposing creative pattern generation for video game scenes by devising new methods to increase the variance of game object pattern composition.

Tasks of the Dissertation

To achieve the objective, the following problems had to be solved:

1. To review common video game scenes or creative artefact procedural generation methods.
2. To develop new methods for video game scene object layout procedural generation by combining genetic algorithms with MCDM methods and neutrosophic sets.
3. To develop a fitness function and criteria for the genetic algorithm, which is focused on the creative and game design value.
4. To develop an engine for game scene generation and experiment with different sets of rules and procedural generator extensions to generate a set of game scene layouts, which aims to increase automated creative and game design value.

Research Methodology

The dissertation applied literature analysis for the investigation of the existing video game artefact generation with a high focus on game scene object layout composition and problem formulation. Genetic algorithms, multi-criteria decision-making, fuzzy logic, and cellular automata methods were applied to develop creative game scene generation strategies.

Scientific Novelty of the Dissertation

The dissertation introduces the following scientific novelty:

1. Adaptation of WASPAS-SVNS method and CoCoSo enhancement for the genetic procedural game scene layout generation fitness function was created, which allows for increased indeterminacy and criteria combination effectiveness and, thus, increases the amount of distinct generated artefacts.
2. Quantisation and fitness criteria were modelled with Gestalt principles for the genetic procedural game scene layout generation to increase the aesthetic value generation for the game scenes.
3. The WASPAS-SVNS-based procedural generator was extended with locally morphing game objects to increase the variety and composition of game object clusters.

Practical Value of the Research Findings

The research findings can be applied to generating video game-level environments, where the focus is on the aesthetic value of game object layouts. A practical application of the proposed algorithms can be used to assist a game-level designer with a system that automates a part of the low-level game object placement creative task.

Defended Statements

The following statements based on the results of the present investigation may serve as the official hypotheses to be defended:

1. WASPAS-SVNS and CoCoSo methods for the genetic procedural game scene layout generation create a varied set of game scene layouts. It successfully creates functional aesthetic patterns without specifically defining the shapes of the combination of game objects.
2. The modelling approach based on Gestalt principles applies abstract rules discreetly, resulting in the creation of game scenes that exhibit Gestalt principle-based arrangements of game objects.
3. Locally morphing game object clusters adds more visual object variations to a functional game object type by adding an array of possible visual variations and, thus, increases the aesthetic value of the generated game scene.

Approval of the Research Findings

Research results on the dissertation topic were published in five scientific publications. Three were published in the reviewed scientific journals, which are indexed in Web of Science databases (Petrovas and Bausys, 2022; Petrovas, Bausys, Zavadskas and Smarandache, 2022; Petrovas, Bausys and Zavadskas, 2023); and two were published in proceedings of international conferences (Petrovas and Bausys, 2022; Petrovas and Bausys, 2019).
The author made three presentations at international scientific conferences:

- International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022.
- International Workshop Data Analysis Methods for Software Systems (DAMSS), Druskininkai, Lithuania, 2019.
- Open Conference of Electrical, Electronic and Information Sciences
- (eStream), Vilnius, Lithuania, 2019.

The Structure of the Dissertation

The scope of the dissertation consists of an introduction, three chapters, general conclusions, a list of references, and a list of publications by the author. The scope of the dissertation is 93 pages, 44 equations, 54 figures and two tables. The dissertation makes 101 related research references.

1

Overview of Creative Procedural Generation Methods for Video Games

The chapter reviews creative procedural generation methods and focus areas for video games. It discusses common and emerging strategies for procedural game scene generation and current problems that should be considered when building a generation model. The proposed approach focuses on MCDM-based fitness functions for genetic algorithms combined with an abstract criteria digitisation process. Creativity, procedural generation, MCDM, and fuzzy set approaches are reviewed for the process of creating video game scenes.

The main research results of this chapter were published in five author’s scientific publications (Petrovas and Bausys, 2022; Petrovas, Bausys, Zavadskas and Smarandache, 2022; Petrovas, Bausys and Zavadskas, 2023; Petrovas and Bausys, 2022 July; Petrovas and Bausys, 2019), and findings of the research were presented at three international conferences (International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022; International Workshop Data Analysis Methods for Software Systems (DAMSS), Druskininkai, Lithuania, 2019; Open Conference of Electrical, Electronic and Information Sciences (eStream), Vilnius, Lithuania, 2019.

1.1. Computational creativity

In the field of artificial intelligence, researchers are constantly making discoveries fuelled by advancements in computing power, storage, and data volumes (Dick, 2019). One emerging approach gaining momentum is Computational Creativity, which involves solving creativity problems through computation-based systems that attempt to simulate creative work. However, there is no widely agreed-upon engineering-specific definition of creativity. Nevertheless, understanding the essence of the concept is essential for effectively modelling a system based on creative principles. The definition of creativity can be broken down into several parts, depending on how creative work is assessed or generated. Four types of creativity modelling targets are typically considered: person, process, product, and press. The most common targets in machine learning tasks are product and process (Lamb et al., 2018). The product target evaluates completed works and aims to replicate them by combining and expanding elements from previous works. The process target simulates logical loops used in creative work generation. The person target, which involves simulating creative agents or individuals, is used less frequently. The press target is common in filtering creative and impactful work, such as internet content scans. Among these targets, research focused on the process-related target, which tends to generate more example-independent results, a crucial aspect of computational creativity. It means that the generated work differs more from the training data set, enabling more original and unique creations (Carballal et al., 2019).

Research can break creativity down into various classifications to comprehend its structure, which is crucial for developing a model. Creative value can be defined by the following key aspects: usefulness, aesthetics, originality, relevance to the task, and surprise (Ventura, 2016). These aspects are often considered when determining the value of a creative outcome. Creativity involves a blend of expertise, chance, and intuition. Integrating these traits into a system generally enhances the likelihood of producing results with greater creative value (Lamb et al., 2018). Examples of cognitive creativity approaches include concept combination, concept expansion, imagery, metaphor, and divergent thinking (Cook et al., 2013). The cognitive approach is often likened to a heuristic search. Evaluating the outcomes is a crucial task in defining creativity. Common evaluation methods include Classification, Regression, Predictive Models, and Generative Models. However, these methods typically focus on replicating creative content rather than exploring new spaces. On the other hand, transformational creativity systems aim to autonomously determine what is creative using more abstract evaluation techniques. The most effective evaluation often comes from external sources, such as feedback from other creative systems (Taivonen and Gross, 2015). Nonetheless, transformational systems are not widely employed or fully realised at present. The challenges for evaluating creativity can be grouped into two categories: the generation of transformational creative content that adds new value to existing results and the issue of generated results being too similar across multiple iterations of the same model. Few models specifically designed for creativity exist currently. Although creative models can generate artwork, they often lack contextual creative value (Franceschelli and Musolesi, 2021).

Creating algorithms for creativity presents a significant challenge due to the complexities of constructing a mathematical model for this purpose. Traditional algorithms are often ill-suited for handling creative tasks. Despite advancements in computing power and novel approaches to address creativity, many new methods are emerging to tackle creative tasks. Nevertheless, in most cases, it remains arduous to rival human creativity (Franceschelli and Musolesi, 2021).

One of the most prevalent perspectives on creativity modelling involves dividing it into four main categories. These categories are person (characteristics of the creative agent and the model used), process (actions taken during the creative process), product (the resulting creative artefacts), and press (meta-information indirectly related to the work and the cultural context surrounding the outcome) (Cook et al., 2019; Walia, 2019). The scientific community often employs reverse engineering systems to lay the groundwork for creativity simulations, which align with the product-oriented approach (Srinivasan and Uchino, 2021). A current computational creativity challenge revolves around generating visually appealing and functional creative outcomes. In the proposed research, the focus is on analysing and modelling the creative process. Usually, defining creativity entails the creation of new concepts, which can be difficult to learn solely from existing data (Srinivasan and Uchino, 2021). A creative layout is a design that uses visual elements to create an attractive and engaging presentation of information or content.

1.1.1. Computational creativity in video games

Automated game-level generation has become a topic of growing interest. While machine learning algorithms excel at specific computational tasks, they still struggle to replicate human creativity perfectly. The main objective of research in the area is to identify measurements of creativity and apply them to automated content generation. Currently, the results in this field are mainly exploratory and often cannot fully replace creative work. However, they gradually assist creators by handling simple creative tasks. One of the tools aiding in work generation is PCGML, which involves four modelling steps: problem identification, solution, results, and the application of the generated outcomes.

In game design, several research examples address the challenge of computer-generated creativity. One such example involves generating physical puzzle game levels, focusing on the feasibility and stability of objects (Pereira et al., 2016). The final fitness of the levels is determined by an agent that plays the game. The approach reduces computational costs and emphasises new solutions to calculate rewards for the genetic algorithm while minimising penalties. Another example is a level generator for a game similar to Lode Runner. Playability and connectivity are assessed using the "A *" algorithm, and an autoencoder with a multi-channel approach analyses 150 pre-made levels. Evolutionary algorithms are utilised to encode the levels into multi-channel strings, adding an element of unpredictability. Performance evaluation involves comparing the generated levels' similarity to the original game levels (Thakkar et al., 2019). A framework for generating general 2D games, primarily top-down adventures, is also present (Zafar et al., 2020). The framework evaluates levels of symmetry, balance, density, and reachability, focusing on aesthetics and difficulty. It derives three different fitness values (Score Difference Fitness, Unique Rule Fitness, and Metric Based Fitness) and calculates the average value for the final fitness. The research aims to apply procedural video game generation to various games with differing rules. Another notable example concentrates on creative patterns (Volz et al., 2020). It uses a match-3-type game as the foundation for the generator, employing visual pattern recognition and line symmetry for evaluation. Expert study analysis is used to judge the results. The generator learns from existing content and employs pattern-aware PCGML, random Markov fields with symmetric positional information, and visual analysis to create larger structural patterns. In the case of a Pac-Man arcade-type game (Safak et al., 2016), playability, object spread, and ratios are evaluated, and levels evolve using a genetic algorithm. The objective is to generate unique levels with each iteration of the algorithm. Finally, another more generalised example involves generating verticality for primarily flat surfaces on grid-based platforms (Petrovas and Bausys, 2019).

Most examples found in the literature involve experimentation in a 2D space and focus on games from the 1980s or simple game levels designed specifically for particular tasks. The typical objects used for creating game levels include empty spaces, walls, players, goals, collectables, and hazards. Common evaluation criteria include guidance, progression, aesthetics, safe zones, and pace-breaking. The current state of computational creativity in the field of video games is relatively young and has not yet been extensively applied to large and complex games. Additionally, it is challenging to develop systems that can fully replicate the manual creative work of human designers. As the complexity of the resulting game increases, it becomes easier to distinguish between synthetic and human-generated creativity.

Object variety and the patterns in which they are combined are crucial factors that define the value of game levels. Multiple-level design rules can be employed to generate these patterns. The initial step in creating a layout involves establishing relative patterns between objects and considering the function of each object type. The subsequent stage is to introduce variety by incorporating alternative looks for specific game object types, thereby enhancing the aesthetic appeal of the game level (Alvarez et al., 2018; Schrum et al., 2020; Atkinson and Parsayi, 2021). The proposed research is centred on the next layer of game scene generation, where the functional level is expanded through regionally morphed zones. Within these zones, arrays of alternative visual objects are used to fulfil the specific functions of game objects.

1.2. Procedural generation in video games

Procedural game content generation using machine learning models on existing content (PCGML) is a method employed to address content generation challenges in the gaming domain (Summerville et al., 2018). The utilisation of procedural content generation is on the rise in the game industry, and researchers are continuously exploring innovative approaches to create high-quality content. Assistance levels for content generation can be classified as partial, complete, autonomous, interactive, and guided. Game content is categorised into functional and cosmetic elements. However, the main challenges with the procedural generation machine learning approach include training on limited datasets, the absence of suitable data, parameter adjustment, and other related issues (Togelius et al., 2013). Procedural content generation methods (PCG) often lack comprehensive evaluation, and the objectives are typically defined by designers. The use cases for PCGML encompass autonomous generation, artificial intelligence-assisted design, repair, analysis, and data compression. The proposed research mainly focuses on autonomous generation, creating game content without human interaction by combining algorithms with fitness functions. Video games, as a widely used form of multimedia, require a broad scope of machine-learning approaches. Game design inherently demands levels that are both playable and aesthetically pleasing (Risi and Togelius, 2019). However, there is currently no standardised approach to dataset standardisation and performance evaluation for game design problems (Summerville et al., 2018; Isaksen et al., 2015; Liapis et al., 2012). The objectives of PCG in a game-level generation are to enhance replayability, reduce the demand for creators' time, save storage space, and enable specific aesthetics (Risi and Togelius, 2019).

Procedural Content Generation (PCG) is an algorithmic method used to generate assets or compositions of assets. The outcomes typically consist of computer-generated content that can be utilised at various stages of the game development process. This approach is frequently employed to produce creative results and can modify diverse video game elements. One common element used in procedural game content generation is the arrangement of game objects. The primary motivations behind using PCG are personalisation, replayability, and cost-effectiveness (Togelius et al., 2011). This research focuses on enhancing replayability by creating unique and diverse game scenes using the same algorithm. Replayability implies novelty in the context of creativity. Consequently, the new outcomes are sufficiently different to be considered interesting, even after multiple instances of the same game level. Ensuring replayable content is crucial for video games when using the same PCG algorithm for level generation.

Procedural generation has become a widely used tool in game development for generating content in games efficiently. It enables the creation of multiple game assets in a relatively short period, provided the generation tool is already set up. In contrast, human-crafted game levels are often more unique but may lack randomness or the natural patterns observed in nature. Procedurally generated levels can incorporate reactivity to player actions and personalised game content (Short and Adams, 2017). Procedural generation can be defined in various ways; some authors describe it as automatically generated assets with limited input data or amplification algorithms, while others define it as a fully automatic generation algorithm with no input data. The primary focus for automatic generation is minimal input (Freiknecht and Effelsberg, 2017). The proposed research aims to minimise input data while generating unique layouts for game levels.

The popularity of research on autonomous game-level layout generation systems has been on the rise. Various methods are being proposed, often combined with evolutionary algorithms, to incorporate creativity traits with minimal or no pre-generated data (Thakkar et al., 2019; Zafar et al., 2020; Volz et al., 2020; Safak et al., 2016; Petrovas and Bausys, 2022). These methods typically operate by combining game functionality requirements with the application of aesthetic rules for generating levels. The game prototypes used in these algorithms are usually minimalistic and employ one game object type for each game function. The proposed algorithm is an extension of one of these algorithms, namely Genetic WASPAS-SVNS (weighted aggregated sum product assessment in a single-valued neutrosophic set environment), which is used for game layout generation (Petrovas and Bausys, 2022). However, these types of algorithms are often constrained by the size of the criterion array. As the ratio between the game scene grid size and the number of conflicting criteria increases, it becomes challenging to distinguish the features of these criteria effectively.

This research aims to enhance the aesthetic quality of the generated levels by incorporating an additional layer of algorithms that introduce regionally morphing areas using an array of alternative visual objects. This algorithm represents a significant advancement in automated game design, reducing the need for extensive designer supervision while increasing the automated creative value. A distinguishing feature of these algorithms is the absence of a starting seed or pattern, a common practice in procedural generation for game-level design (Nenad, 2018; Khalifa et al., 2019a; Khalifa et al., 2019b).

Procedural generation is a data creation technique that utilises algorithms and automation to produce outcomes (Short and Adams, 2017). As a practical concept in games, it has been in use for over three decades, with one of the first notable games employing this approach being “Elite” (Braben and Bell, 1984). The notion of generating distinct and non-repetitive visual results remains innovative even today. There are various procedural generation techniques in games, including search-based methods (often using genetic or fitness-based algorithms), solver-based approaches, rule-based systems with strict control, and grammar-based algorithms (originating from linguistics) (Liu et al., 2021; Nyholm and Nilsson, 2017). Some examples of procedural generation in games involve using generative adversarial networks (GANs) to train and generate new game levels (Torrado et al., 2020; Giacomello et al., 2018). Others use cellular automata, with evaluation agents iterating over batches of game elements (Earle et al., 2022). Certain methods even analyse gameplay videos to train their models (Guzdial and Riedl, 2016). Additionally, some systems directly apply game design patterns to search-based algorithms (Baldwin et al., 2017). These systems often analyse hand-crafted or indexed creative artefacts to learn and generalise pattern generation. The applications and methods of procedural generation in games are diverse, but the common goal is to create results that are indistinguishable from or assist the creation process compared to manually crafted artefacts. Examples in the literature usually present abstract visual results.

1.2.1. Fitness criteria modelling in video games

The fitness function used in game design and computational creativity typically comprises a subjective combination of criteria, and its quantification is still in the early stages of research (Cook et al., 2019). No widespread consensus exists on how results should be compared in these domains. Game design can be divided into several parts, categorising games based on their objectives. The conversion of fitness criteria varies depending on the game type. Patterns in game design are elements that appear in levels across multiple games rather than being recurring features within the same game title. These patterns are typically classified into various types. “Guidance” patterns direct players in an intended direction, while the “safe zone” pattern denotes an area where players are not exposed to negative interactions. “Foreshadowing” patterns provide hints about events later in the game. “Layering” patterns involve combining multiple objects to create a new experience. “Branching” patterns offer players multiple paths to achieve their objectives, and “pace-breaking” patterns involve altering elements of the game to achieve a creative objective (Khalifa et al., 2019a).

Most automated game design approaches are based on reverse engineering principles, often utilising datasets obtained by analysing existing games. With this approach, there is no need to explicitly define fitness criteria for the generations (Togelius and Schmidhuber, 2008). However, a significant drawback of this method is the lack of novelty and new concepts in the generated content (Toivonen and Gross, 2015). The ultimate goal is to establish objective formulas based on game design principles to generate game levels. In the literature, game flow strategy is proposed as a measure of game design. This concept combines various criteria, including concentration, challenge, player skills, control, clear goals, feedback, immersion, and social interaction (Sweetser and Wyeth, 2005). Some authors attempt to assess game engagement by analysing difficulty and applying constraints to ensure playable levels (Sorenson and Pasquier, 2010). The present research strives to quantify abstract creativity criteria applicable to real-world digital applications.

Multiple methods are available for generating procedurally generated content. To increase the diversity of levels, the model can employ genetic algorithms with an abstract fitness function to enhance the uniqueness of the generated game level scenes (Herrmann, 1999; Pereira et al., 2016; Thakkar et al., 2019; Safak et al., 2016; Beukman et al., 2022). Unique content contributes to replayability, as each instance of the generated level varies sufficiently to remain interesting. The fitness function plays a crucial role in genetically created levels as it guides the evolution of our task. Our problem involves optimisation, aiming for non-deterministic results while ensuring functionality. The fitness function must be somewhat ambiguous to generate a broad range of satisfying results. At the same time, research must define criteria for the generated level, typically categorised as functional and aesthetic criteria. Functional criteria set the rules and constraints of game design, which can be optimised, and limitations that must always hold. On the other hand, aesthetic criteria determine the visual appeal of the generated levels or objects (Statham et al., 2022). Combining all these criteria into a single fitness number requires employing multi-criteria decision-making algorithms (MCDM) (Zavadskas et al., 2014) in this mathematical model.

The original algorithm by Petrovas and Bausys (2022) employs a genetic algorithm to generate game scene layouts. The fitness function incorporates a combination of functional and aesthetic game design criteria, utilising a multi-criteria decision-making algorithm with neutrosophic sets. The WASPAS algorithm is utilised to reconcile conflicting criteria and introduce neutrosophic sets, enhancing the algorithm’s unpredictability and randomness. This base algorithm is commonly applied to various engineering and design problems, such as architectural, construction, environmental sustainability, image processing, or pathfinding tasks (Lescauskiene et al., 2020; Morkunaite et al., 2019; Bausys et al., 2020a; Bausys and Kazakeviciute-Januskeviciene 2021).

Functional criteria are relatively easier to incorporate compared to aesthetic criteria because aesthetic criteria follow more abstract rules and are more challenging to integrate seamlessly. There are various approaches or rules to consider for achieving aesthetic criteria results (Cook et al., 2019; Karth, 2019), such as Apollonian order, Dionysian chaos, Gestalt, individual and repetition shape, style, multiplicity, or cohesion variations. These methods can be utilised to establish aesthetic criteria for procedural generation. However, one of the challenges associated with this choice is to model their implementation, as they are often defined in abstract terms, but for mathematical algorithms, precise definitions of algorithm elements are required, which is the specific focus of this research. Simultaneously, to ensure that the artefacts have a high creative value, methods must be devised to apply high-level aesthetic concepts effectively.

This research introduces a game scene procedural generator that enhances the aesthetic value of levels using low-abstraction level building blocks for high-level aesthetic criteria.

1.3. Multi-criteria decision-making in the genetic algorithm context

Fuzzy logic can be used to express lists of criteria (Lara-Cabrera et al., 2014). Another emerging approach involves combining neutrosophic sets with multi-criteria decision-making (MCDM), which has not been extensively explored in the machine-learning field but holds the potential to enhance the creativity of such models (Lescauskiene et al., 2020; Morkunaite et al., 2019; Bausys et al., 2020a; Bausys and Kazakeviciute-Januskeviciene, 2021). Moreover, combining MCDM algorithms with iterative optimisation algorithms is not commonly employed (Semenas et al., 2021; Bausys et al., 2020b). In some SVNS applications in the literature, there is a strong emphasis on addressing uncertainty (Ali et al., 2020). The following paragraphs provide a more detailed explanation of related work, methodology, the developed framework, obtained results, and concluding remarks.

Many contemporary MCDM algorithms leverage fuzzy sets, allowing for non-deterministic outcomes that enhance the replayability of generated game levels. There are numerous instances of fuzzy MCDM methods applied in research, mostly in static environments and single iterations (Peng and Li, 2021; Peng and Garg, 2021; Yazdani et al., 2021; Yousefi et al., 2021; Svadlenka et al., 2020; Kieu et al., 2021; Zavadskas et al., 2021; Zhao et al., 2021; Wang et al., 2022; Filip, 2022; Semenas et al., 2021). However, due to the genetic algorithm's nature, achieving proper convergence often necessitates multiple subsequent iterations. This research proposes a novel approach that enables MCDM for a series of subsequent iterations. It explores the fuzzy limitations of MCDM and adapts general algorithms to suit the creative PCG task. Non-deterministic creativity problems can be effectively modelled using various versions of modern fuzzy sets.

Multi-criteria decision-making (MCDM) provides an effective approach to integrating fitness function criteria. It offers strategies for selecting alternatives from a finite pool of potential solutions (Zavadskas et al., 2014). These methods can be employed in the fitness function of the genetic algorithm to enhance the optimisation of genetic operators. By utilising fuzzy sets instead of crisp numbers, certain variations of MCDM methods can introduce increased non-determinism.

The research proposes an innovative extension for the neutrosophic CoCoSo method specifically designed for the genetic scene layout generator, where the fitness function is computed using this approach. The research explored adaptation aspects for the CoCoSo method to address the content generation problem. Typically, normalisation is computed within the local Min-Max range, involving a single algorithm iteration. The research updates the normalisation process with global Min-Max values, encompassing all possible value ranges to enhance applicability (Choi and Moon, 2003). This allows each criterion to be normalised within the appropriate value range. Another important consideration is the algorithm’s speed. Since the genetic algorithm calculates fitness using the CoCoSo method for each chromosome in every generation, it can be computationally intensive for repeated calculations. Certain values with a limited impact on the result can be replaced with constant values to optimise performance. Furthermore, a conversion from linear, crisp numbers to neutrosophic sets is necessary. Even the slightest non-proportional change in values can lead to unsuitable outcomes and may exceed mutation differences, affecting evolution (Herrmann, 1999). Hence, careful handling of the conversion is essential to ensure the effectiveness of the genetic algorithm.

1.4. Fuzzy sets theory

Fuzzy set theory, established almost 60 years ago by Zadeh in 1965, has seen successful applications emerging since the 1980s. This theory continuously evolves, and researchers constantly find new ways to redefine and update it (Kahraman et al., 2016; Wu et al., 2021). Examples of applications include software selection problems, intuitionistic linguistic aggregation, human resources management, smartphone selection, and more (Haque et al., 2020). Among the various types of fuzzy sets, neutrosophic sets stand out as they can increase nodeterminism. Neutrosophic sets are defined by three numbers representing true, indeterminacy, and false values, each being independent (Broumi et al., 2018; Smarandache, 1999). Neutrosophy deals with neutralities and their interaction with ideational spectra. The research employed the neutrosophic set environment in the CoCoSo method (Yazdani et al., 2018; Turskis et al., 2022) to calculate the final fitness score for each chromosome in the genetic algorithm.

Zadeh introduced the concept of fuzzy sets in 1965(Zadeh, 1965), which allows elements to have degrees of membership ranging from 0 to 1, depending on how closely they match the set's criteria. This novel idea, known as "fuzzy sets", paved the way for various applications in control theory, decision-making, and pattern recognition, as outlined in Zadeh's seminal paper from 1973 (Zadeh, 1973). In 1999, Florentin Smarandache extended fuzzy logic to introduce neutrosophic sets, which provide a mathematical framework for representing uncertain or indeterminate information (Smarandache, 1999). One of the extensions of neutrosophic sets is Pythagorean neutrosophic sets, which offer a more flexible approach to representing uncertainty compared to regular neutrosophic sets (Wang et al., 2010).

1.5. Conclusions of the First Chapter and formulation of the dissertation tasks

The key observations and conclusions were formulated following the literature review:

1. Different targets exist for computational creativity modelling, including person, process, product, and press. These targets offer unique challenges and opportunities for modelling creativity in various domains, such as art and game design. Defining and measuring creativity in computational systems is a challenge. The varying definitions of creativity and the lack of widely agreed-upon engineering-specific definitions pose difficulties in effectively modelling and evaluating creative work.
2. Fitness criteria are significant in procedural content generation for video games. Incorporating high-level aesthetic concepts into procedural content generation remains a challenge. Abstract aesthetic criteria, such as balance, harmony, and visual appeal, must be translated into precise mathematical algorithms for effective integration.
3. Evaluating creativity is complex, especially in procedural content generation. While various evaluation methods exist, assessing and comparing creative outcomes objectively and comprehensively remains a challenge, especially when dealing with multiple criteria. Using fuzzy sets and multi-criteria decision-making methods, particularly neutrosophic sets, offers a

way to handle uncertainty and introduce non-determinism in creative algorithms. These approaches contribute to enhancing the diversity and uniqueness of generated outcomes.

4. The challenge of balancing generating novel and unique content while ensuring that the generated outcomes remain replicable and consistent. Striking this balance is crucial for enhancing replayability and maintaining player engagement in video games. A well-defined fitness function that incorporates functional and aesthetic criteria is crucial to ensuring the quality and creativity of generated content.

Based on the performed literature survey, the following tasks were formulated to achieve the study aims:

1. To develop a framework for game scene generation.
2. To develop new methods for video game scene object layout procedural generation by combining genetic algorithms with MCDM methods and neutrosophic sets.
3. To develop a fitness function and criteria for the genetic algorithm, focused on the creative and game design value.
4. To experiment with different sets of rules and procedural generator extensions to generate a set of game scene layouts, aiming to increase automated creative and game design value.

2

Genetic Multi-Criteria Decision-Making Methods and Criteria Modelling for Video Game Scene Generation

This chapter discusses and investigates genetic neutrosophic MCDM methods and criteria modelling for video game scene generation. It defines the creativity problems in video game content generation and proposes a novel algorithm for game scene generation with WASPAS-SVNS and CoCoSo-based fitness functions. It also introduces the application of Gestalt design rules for fitness criteria using cellular automata agents. Finally, it explains how game scene objects can be morphed into new objects and offers conclusions for the final generation model.

The main research results of this chapter were published in four author’s scientific publications (Petrovas and Bausys, 2022; Petrovas, Bausys, Zavadskas and Smarandache, 2022; Petrovas, Bausys and Zavadskas, 2023; Petrovas and Bausys, 2022), and the research findings were presented at one international conference (International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022).

2.1. Scene layout modelling and optimisation

The research proposes an automated framework for game scene layout generation based on PCGML (Procedural Content Generation using Machine Learning). The framework utilises a mathematical model that includes a fitness function employed by a genetic algorithm to assess the population. The research employs the Multi-Criteria Decision Making (MCDM) utility function for the genetic algorithm fitness function. Specific criteria parameters are fixed for difficulty, playability, and size adjustments. In each algorithm iteration, the game-level grid is populated and subsequently evaluated. The evaluation process calculates the fitness for each game-level grid, enabling the selection of the best-performing grids. The approach generates diverse and unexpected results as the generation seed is randomly chosen and further refined by the algorithm.

The research integrates various measurements for level design criteria into a multi-criteria decision-making table to define the problem. The overall fitness of the game scene is determined by aggregating the scores for each criterion. This allows for evaluating different generated scenes by using alternative options on one axis and their corresponding fitness scores on the other axis in the table. Based on the table results, the algorithm can then select the most suitable alternatives as a foundation for further scene generations. By calculating the fitness score for each criterion and representing them as fuzzy sets, the research utilises the Weighted Aggregated Sum Product Assessment with Single-Valued Neutrosophic Sets (WASPAS-SVNS) method to handle situations with conflicting criteria (Lescauskiene et al., 2020; Morkunaite et al., 2019; Bausys et al., 2020a; Bausys and Kazakeviciute-Januskeviciene, 2021; Semenas et al., 2021; Bausys et al., 2020b). Fuzzy logic with neutrosophic sets is employed for the calculations (Stanujkic, 2021).

From a computational creativity perspective, the proposed approach incorporates elements of the creative process, including considerations of usefulness, aesthetics, and chance, which together form the constraints and criteria set for the mathematical model. The framework is designed to generate video game-level layouts by initially creating random levels, then refining them using a genetic algorithm, and finally evaluating them with a weighted aggregated sum product assessment to identify the most promising alternatives. Moreover, the framework is flexible and can be expanded by adding additional requirements and fitness criteria, and many parameters can be adjusted to meet specific needs. The detailed explanation of the proposed approach is organised into four chapters: game scene modelling methodology, game scene procedural generation criteria list, the proposed extension of the genetic algorithm through WASPAS-SVNS, and the application of the WASPAS-SVNS utility function to calculate the fitness function.

2.1.1. Game scene encoding modelling

A standard set of game objects is employed, chosen based on the game-level design principles. These objects are represented by various numbers in the matrix, where each number corresponds to a different object type. The game scene layout is discretised into a grid, and each cell can accommodate only one object. A single genetic algorithm chromosome corresponds to a single scene layout. The objects and their respective numbers are as follows:

- Player (number 0) – signifies the starting position of the player, the character intended to play the game.
- Exit (number 1) – marks the location that the player must reach to complete the game.
- Empty space (number 2) – represents traversable and unoccupied areas through which the player can move.
- Wall (number 3) – refers to an object that obstructs the player's movement.
- Hazard or enemy (number 4) – represents a traversable object that poses a danger to the player.
- Collectable (number 5) – denotes a desirable object that the player can collect during gameplay.
- Ground – while not encoded in the chromosome matrix, the object is used in the 3D projection visualisation step as the floor layer.

The information of a single chromosome is stored in a 2D numerical grid, as shown in Figure 1.1 (Fig. 2.1). For experiments, the research utilises a matrix with dimensions of ten units in width and ten units in length. Each object type is represented by a distinct number in the grid. To visualise the outcomes, research projects them into the 3D space by adding a ground layer below the grid and converting the numerical values into corresponding 3D objects on the main grid.

3	3	3	3	3	3	3	3	3	3
3	0	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	1	3
3	3	3	3	3	3	3	3	3	3

Fig. 2.1. Single-chromosome data example

2.1.2. Criteria modelling for the fitness function

Following discretisation, the research establishes a set of criteria that outline the requirements for the game layout. The research proposes the utilisation of four fitness criteria functions and three constraint functions. These functions employ their respective results to calculate the overall fitness value for each iteration of the genetic algorithm using the WASPAS-SVNS algorithm. Aesthetics are evaluated through the symmetry and empty-space balance criteria, while usefulness is assessed using the safe zone and player exit distance criteria. The selection of these criteria is based on their recurrence in the literature (Pereira et al., 2016; Thakkar et al., 2019; Zafar et al., 2020; Volz et al., 2020; Safak et al., 2016), their alignment with game design principles, and their relation to creativity definitions (Lamb et al., 2018; Ventura, 2016; Cook et al., 2013). If any of the constraint functions fail to meet their conditions, the total fitness value is multiplied by zero.

The outcomes of the criteria are standardised to fall within the range of 0 to 1, providing a common reference point for various criteria metrics (Sorenson and Pasquier, 2010). Here, 0 represents the poorest possible value, while 1 indicates the optimal value. To avoid potential biases in the evaluation process using neutrosophic sets, the final values for each criterion are scaled by 0.9, preventing them from getting too close to 1. Scalar values are transformed into single-valued neutrosophic sets during the evaluation. The fitness functions are as follows:

- Aesthetic symmetry is calculated to determine the level of symmetry in the chromosome grid. The grid is divided into four smaller 5×5 grids by crossing it with a horizontal and vertical slice. Each object in the 5×5 grid is examined to see if it has an identical symmetrically matching object both vertically and horizontally (Figs. 2.2 and 2.3). The final symmetry score is computed by dividing the number of symmetrical matches by the maximum number of possible matches, where each object can have two matching objects in the adjacent 5×5 grids that share a boundary:

$$m = \frac{\sum_{i=0}^{2xy} s}{2xy}. \tag{2.1}$$

  The grid size is denoted by “x” and “y”, and “s” is a binary value representing symmetry. If an object does not have a matching pair, “s” is set to 0. Each object is assessed twice for both horizontal and vertical axes;

- The balance criteria used for aesthetic purposes measures the proximity of the ratio between the count of empty game objects and the total count of objects to 50% (Fig. 2.4);

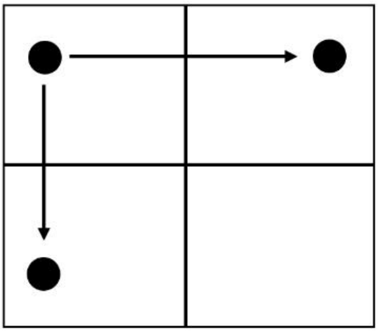


Fig. 2.2. Symmetry calculation

- Mathematically, it can be expressed in these steps:

$$\mathrm{e} = \frac{t}{\frac{t}{\frac{1}{2}xy}}. \tag{2.2}$$

- In this context, the variable e represents the total empty space ratio, which is normalised to a range between 0 and 1. The variable t corresponds to the sum of empty objects, and if this sum exceeds 50% of the grid size, its value is reversed. The variables x and y indicate the grid size, and the variable s takes a binary value of 1 if the object is empty. To calculate t, all empty space objects are counted, and the value is reversed if it surpasses 50% of the grid size:

$$\begin{cases} t = \sum_{i=0}^{n} s_n \\ t = \frac{1}{2}xy - t_1 - \frac{1}{2}xy \end{cases}. \tag{2.3}$$

- The criterion evaluates the distance between the player and the exit game objects. The variables x and y represent the coordinates of the player and exit, respectively. The objective of the rule is to ensure that the player can observe as much of the generated scene as possible while moving towards the exit point:

$$d = \sqrt{(x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1)}\,. \tag{2.4}$$

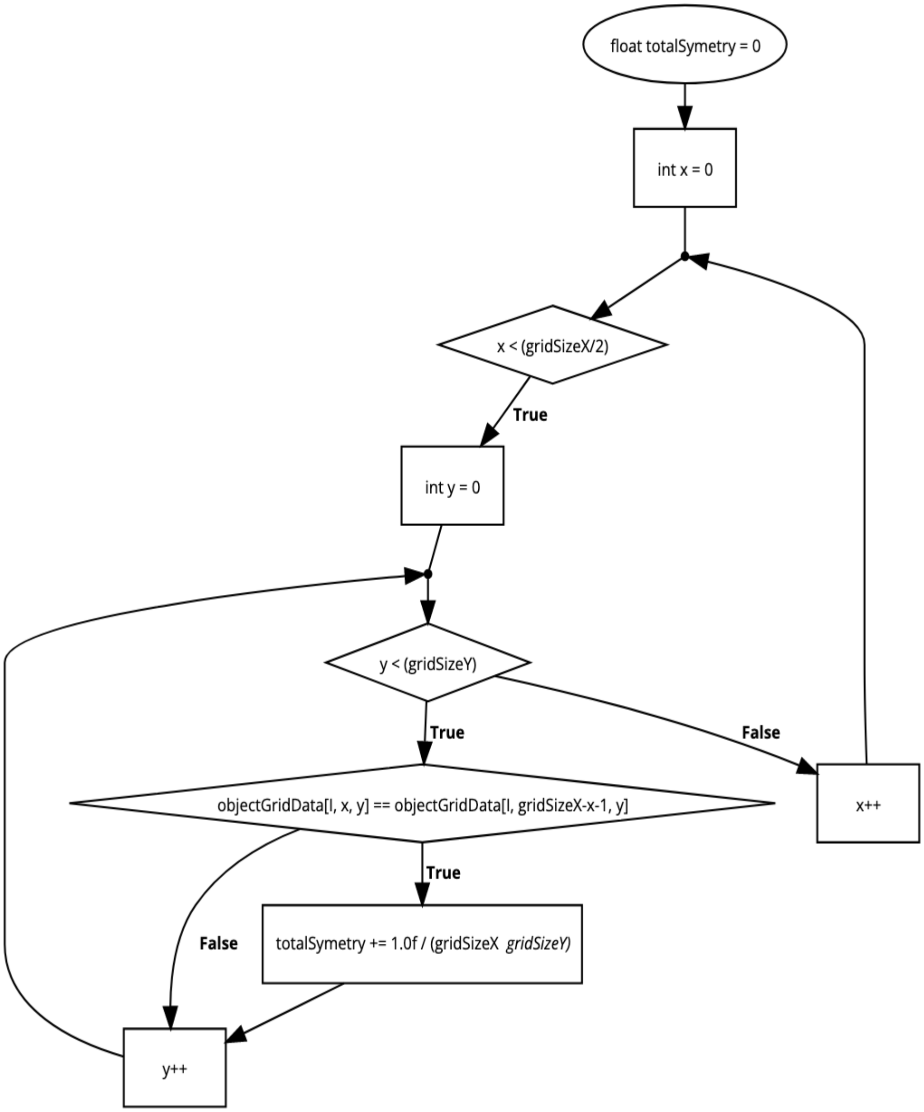

float totalSymetry = 0
int x = 0
x < (gridSizeX/2)
True
int y = 0
y < (gridSizeY)
True
False
objectGridData[l, x, y] == objectGridData[l, gridSizeX-x-1, y]
True
False
totalSymetry += 1.0f / (gridSizeX *gridSizeY)*
x++
y++

Fig. 2.3. Symmetry calculation for a single grid axis

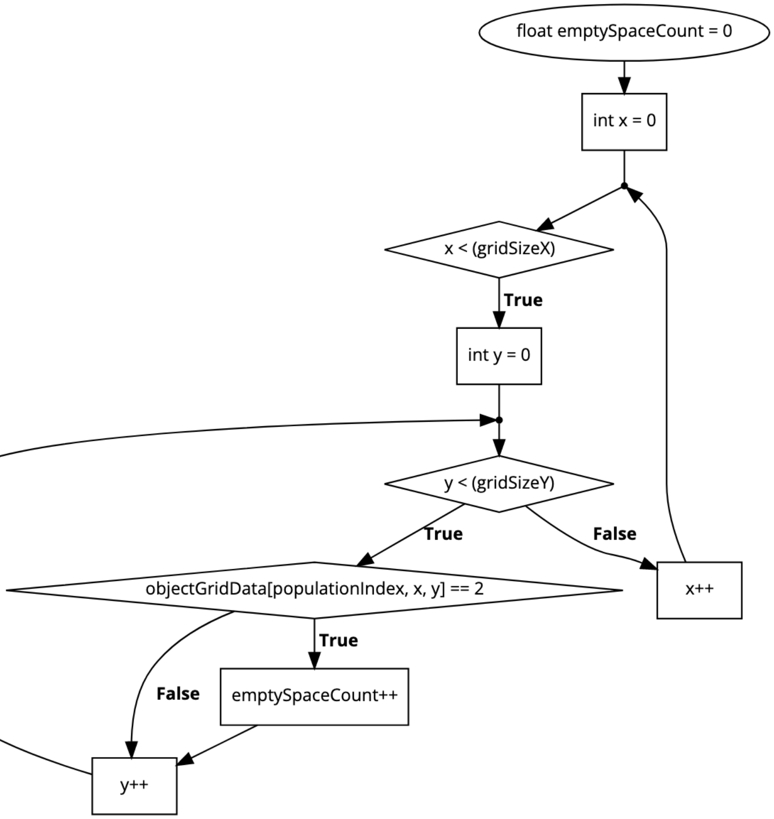


Fig. 2.4. Empty-space balance

- The safe zone criteria assess the density of Hazzard-type objects within a specified square area around the player. The result is obtained by dividing the count of Hazzard-type objects within the square by the total area of the square. The criterion aims to determine how safe the surrounding area of the player is from potential dangers:

$$z = \frac{x_1 y_1}{x_2 y_2}. \tag{2.5}$$

  Each member of the population (Fig. 2.5) has its criteria calculated, and these criteria can be adjusted as needed. The calculations of these criteria serve as the foundational elements of the fitness function.

The constraint functions involve the following checks on the chromosome grid:

- Scanning to ensure the presence of the player object.

- Scanning to ensure the presence of the exit object.
- Using a pathfinding algorithm to verify the existence of a passable route between the player and exit.

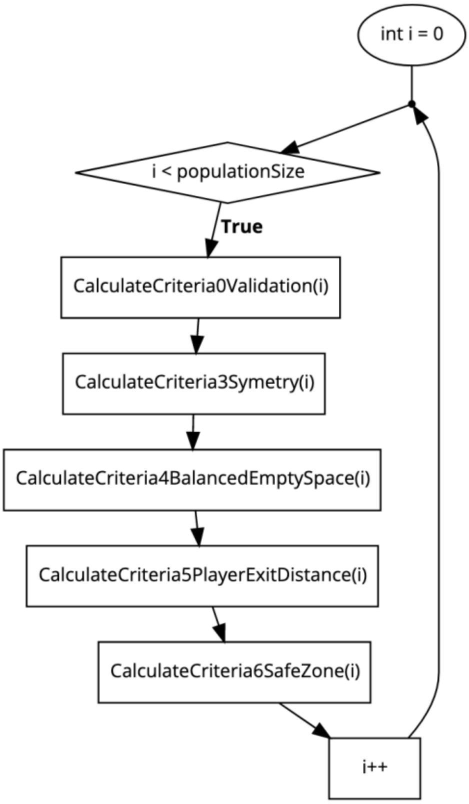

int i = 0
i < populationSize
True
CalculateCriteria0Validation(i)
CalculateCriteria3Symetry(i)
CalculateCriteria4BalancedEmptySpace(i)
CalculateCriteria5PlayerExitDistance(i)
CalculateCriteria6SafeZone(i)
i++

Fig. 2.5. Initial criteria list

2.1.3. Gestalt principles

The aesthetic aspects of created products are often achieved by following visual principles known as Gestalt principles (Wertheimer, 2012). These principles are rules of element organisation and perception, guiding how game objects should be arranged to achieve specific aesthetic effects. Gestalt principles are based on optical perception, where the brain processes complex compositions of objects as a hierarchy of visual elements defined by their size and scope. Deep neural networks and convolutional neural nets also exhibit similar hierarchical abstraction behaviour (Todorovic, 2008; Serb and Prodromakis, 2019; Soleymani et al., 2018; Yu et al., 2017). Compared to the training approach on existing information, the research focuses on new information generation. There are typically between five and ten Gestalt principles, each offering abstract rules for visual input grouping based on specific objectives. While originally not designed for automated tasks, their low-level building strategy makes them valuable for achieving high-quality aesthetic results. Applications of Gestalt principles include visual data representation, network data visualisation, computer interfaces, and image edge detection, where visual grouping enhances information processing efficiency (Nesbitt and Friedrich, 2002; Chang and Nesbitt, 2006; Cao, 2004). The Gestalt principles (Table 2.1) offer a set of rules for visual grouping and abstraction in the context of this research.

Table 2.1. Gestalt principles

Name of the principle	Explanation
Similarity	Similar-looking objects are visually grouped regardless of their proximity to each other (Fig. 2.6).
Proximity	When objects are close to each other, they are perceived as groups (Fig. 2.7).
Continuity	Aligned and smoothest paths are integrated into perceptual wholes (Fig. 2.8).
Focal Point	An object that is different compared to a whole will stand out (Fig. 2.9).
Common Region	When objects are within a closed region, they are perceived as a group (Fig. 2.10).
Closure	Incomplete object patterns are perceived as complete (Fig. 2.11).
Figure Ground	Separation of objects between foreground and background based on their shape and associations (Fig. 2.12).
Common Fate	Objects that point in the same direction are grouped together (Fig. 2.13).

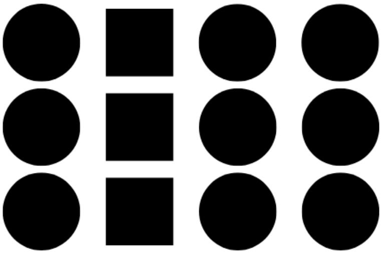


Fig. 2.6. Similarity

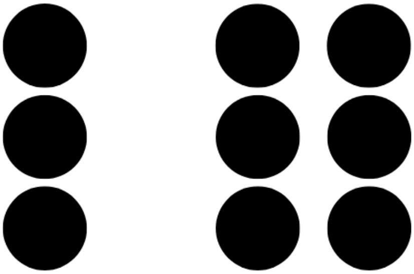


Fig. 2.7. Proximity

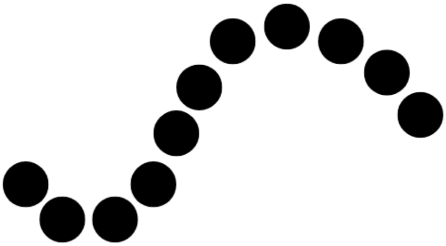


Fig. 2.8. Continuity

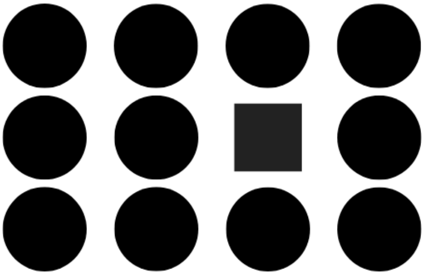


Fig. 2.9. Focal Point

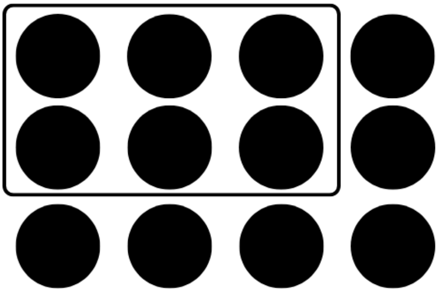


Fig. 2.10. Common Region

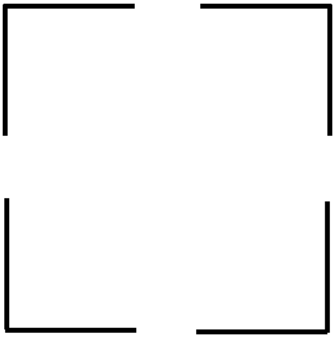


Fig. 2.11. Closure

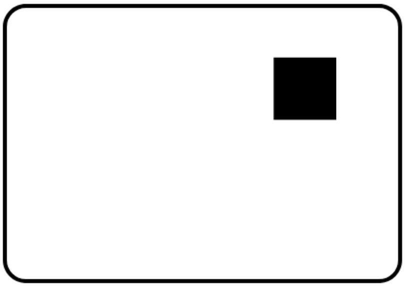


Fig. 2.12. Figure Ground

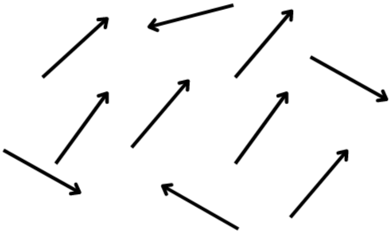


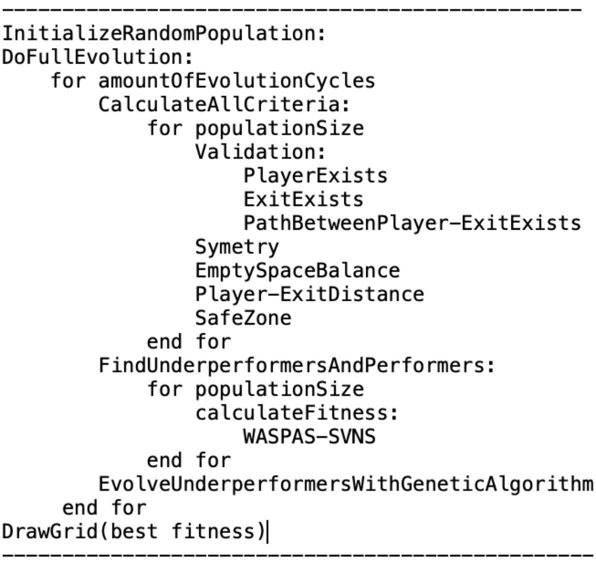
Fig. 2.13. Common Fate

Aesthetic criteria can be classified based on their level of abstraction. High-level criteria have abstract definitions, while low-level criteria are more concrete and can be expressed through direct algorithms. The application of Gestalt principles in-game scene generation represents a novel approach, especially considering their widespread use in interface design. High-level aesthetic criteria offer advantages for modern MCDM technologies, particularly when dealing with vague initial information, which can be achieved through the application of neutrosophic sets. Neutrosophic sets enable the definition and processing of linguistic or abstract value indeterminism. The next step involves presenting the fundamental concepts of extending the genetic algorithm with MCDM, specifically using Pythagorean neutrosophic WASPAS.

2.2. Genetic algorithm for procedural generation in video games

Genetic algorithms are employed to address optimisation problems by leveraging natural selection principles (Whitley, 1994). The approach is advantageous for research purposes as it can explore various local maximums due to its random nature. The algorithm iteratively modifies a population of individual game-level grids, selecting and modifying random grids at each generation. Over time, the population evolves towards both aesthetic and functional solutions. The genetic algorithm offers the advantage of finding good solutions without explicitly designing them. It employs transformative operators on a small set of grid cells. The main advantages of using genetic algorithms over other optimisation methods include non-linear convergence, the ability to evolve multiple solutions in parallel, retention of the best solutions, non-deterministic behaviour due to the use of random numbers, and the generation of different solutions in each run of the algorithm. The fitness function is based on the WASPAS-SVNS algorithm and is used to evaluate and select the best individuals based on multiple criteria. Each grid in the population represents a single solution to the problem, and the size of the population determines the total concurrent grid pool. The best fitness value indicates the most favourable grid designs in the current population. During the calculations, two concurrent snapshots are considered, namely the parent and child generations.

The game level layout is trained using the genetic algorithm (Fig. 2.14), and the evaluation criteria for each iteration are combined using the WASPAS-SVNS algorithm to calculate a single fitness value. The population size is set to 50, and the algorithm is run for 2000 iterations. The initial population is generated randomly, and convergence criteria is the best value of the WASPAS-SVNS evaluated chromosome. The genetic algorithm utilises selection and mutation operators to filter and repopulate the population. To initialise the data, research creates empty chromosomes and fills them with random data, where each object is encoded with integer numbers from 2 to 6 (representing all possible objects except the player (number 0) and exit (number 1)). Research then adds one player and one exit object to each chromosome. In each iteration, research calculates the median fitness value for the entire population and splits the chromosomes into two temporal arrays, one storing chromosomes below the median value and the other storing chromosomes above the median value. The chromosomes below the median value are replaced with chromosomes from the above-median array, and then 5% of the data in this new array is mutated by assigning new random values.



```
------------------------------------------------
InitializeRandomPopulation:
DoFullEvolution:
    for amountOfEvolutionCycles
        CalculateAllCriteria:
            for populationSize
                Validation:
                    PlayerExists
                    ExitExists
                    PathBetweenPlayer-ExitExists
                Symetry
                EmptySpaceBalance
                Player-ExitDistance
                SafeZone
            end for
        FindUnderperformersAndPerformers:
            for populationSize
                calculateFitness:
                    WASPAS-SVNS
            end for
        EvolveUnderperformersWithGeneticAlgorithm
     end for
DrawGrid(best fitness)
-------------------------------------------------
```

Fig. 2.14. Genetic algorithm

2.3. Weighted aggregated sum product assessment algorithm extension for genetic algorithm

In the evaluation phase, the research employs the modified WASPAS-SVNS algorithm (Lescauskiene et al., 2020) to combine the fitness results of the criteria functions. Previous applications of the algorithm mainly involved single iterations (Lesciauskiene et al., 2020; Morkunaite et al., 2019; Bausys et al., 2020b; Bausys and Kazakeviciute-Januskeviciene, 2021). However, this research focuses on an iterative approach using WASPAS-SVNS, which requires adjustments to make it compatible with the genetic algorithm. The section outlines the primary steps of the final evaluation and describes how research integrated it with our procedural generator:

1. The criteria evaluation data is consolidated into a matrix X, where one dimension represents the chromosome index, and the other dimension represents the criteria index. Each element x_{ij} represents a game scene:

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}. \tag{2.6}$$

2. In the original algorithm, data normalisation is performed within the WASPAS-SVNS algorithm. However, for the iterative process, this approach is not suitable because the local min-max and global min-max values differ. As a result, it becomes necessary to establish boundaries before this normalisation step (Padhye and Deb, 2011). Instead, normalisation is carried out in the criteria functions to ensure that the data falls within the range of 0 to 1:

$$\tilde{x}_{ij} = \frac{v}{v_{max}}, \tag{2.7}$$

   v represents the current criteria value and v_{max} is the highest possible value for that criterion for the selected matrix size. $\tilde{x}_{ij}$is a normalised index ij criteria value of matrix X;
3. Neutrosophication. In this phase, research transforms the outcomes obtained from the normalised criteria functions into neutrosophic sets. A neutrosophic set comprises three values: truth (t), intermediary (i), and falsehood (f). To achieve this, research associates the criteria results with neutrosophic numbers through a linear conversion. Research does this to avoid any significant non-proportional shifts that could lead to substantial errors during the evolutionary process. In this context, “N” represents a neutrosophic number, and “S” denotes a scalar number:

$$N(t,i,f) = \begin{cases} S \\ 1 - S \\ 1 - S \end{cases}. \tag{2.8}$$

4. The sum of the overall relative importance of the alternative (represented by a single evolutionary iteration chromosome);
5. The cumulative relative importance of the alternative product;
6. A combined and comprehensive criterion for ranking alternatives, incorporating both step 4 and step 5:

$$\widetilde{Q}_i = 0.5\widetilde{Q}_i^{(1)} + 0.5\widetilde{Q}_i^{(2)} . \tag{2.9}$$

7. Neutrosophic numbers (truth, intermediacy, and falsehood) are transformed into scalar numbers using this formula and subsequently utilised for chromosome evaluation in the genetic algorithm:

$$S(\widetilde{Q}_i) = \frac{3 + t_i - 2i_i - f_i}{4} . \tag{2.10}$$

2.3.1. Genetic weighted aggregated sum product assessment game scene generator

The foundation of the proposed algorithm is the WASPAS-SVNS game scene layout generator. The original algorithm utilises seven types of game objects as fundamental elements for creating layouts. These objects include the player, exit, empty space, wall, hazard, collectable, and ground, each serving a distinct game function. The initial criteria for the fitness function include symmetry, balance of empty spaces, distance between objects, the presence of a safe zone, the existence of the player and exit, and the presence of a path connecting the player and exit. In this approach, research enhances the existing objects by introducing more visual variety and implementing an additional layer of post-processing algorithm on the generated level.

The underlying algorithm begins by initialising a random population of generated level layouts and subsequently launches an evolutionary algorithm. In each evolutionary cycle and for each chromosome, the WASPAS-SVNS iteration is repeated. Individual criteria are computed, starting with the constraining ones that adhere to game design rules and then incorporating the criteria that should be maximised, related to aesthetic rules. Following this, the algorithm identifies the top performers and underperformers within the population. The best chromosomes are further evolved. The ultimate phase of the algorithm involves the level layout map decoder and visualiser (Petrovas and Bausys, 2022). Research enhances the decoder component by introducing regionally morphing visual objects.

2.4. Combined compromise solution method

The general CoCoSo method operates through the following steps. The alternatives in this context represent the population of the genetic algorithm, and each chromosome criterion aligns with the criterion of the CoCoSo method.

1. The criteria evaluation data is consolidated into a matrix X, where one dimension represents the chromosome index, and the other represents the criteria index. Each element x_{ij} represents a game scene. The initial decision-making matrix is determined as follows, where *I* = 1,2 *..., m; j* = 1,2, ... *n*:

$$X_{ij} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}. \tag{2.11}$$

2. The normalisation of the criteria values for the benefit and cost criterion:

$$r_{ij} = \frac{x_{ij} - \min\limits_{i} x_{ij}}{\max\limits_{i} x_{ij} - \min\limits_{i} x_{ij}}, \tag{2.12}$$

$$r_{ij} = \frac{\max\limits_{i} x_{ij} - x_{ij}}{\max\limits_{i} x_{ij} - \min\limits_{i} x_{ij}}. \tag{2.13}$$

3. Next, the research calculates an alternative sum of the weighted comparability and the amount of the power weight of the comparability sequences:

$$R_i = \sum_{j=1}^{n} \left(w_j r_{ij}\right); \tag{2.14}$$

$$P_i = \sum_{j=1}^{n} \left(r_{ij}\right)^{w_j}. \tag{2.15}$$

4. Three appraisal score strategies are used to generate relative weights. k_{ia} expresses the arithmetic mean of the sums of the WSM and WPM scores:

$$k_{ia} = \frac{R_i + P_i}{\sum_{i=1}^{m} (R_i + P_i)}, \tag{2.16}$$

k_{ib} expresses the sum of the relative scores of WSM and WPM compared to the best:

$$k_{ib} = \frac{R_i}{\min\limits_{i} R_i} + \frac{P_i}{\min\limits_{i} P_i}, \tag{2.17}$$

and k_{ic} releases the balanced compromise of the WSM and WPM model scores:

$$k_{ic} = \frac{\lambda(R_i) + (1-\lambda)(P_i)}{\left(\lambda \max\limits_{i} R_i + (1-\lambda) max\limits_{i} P_i\right)}. \tag{2.18}$$

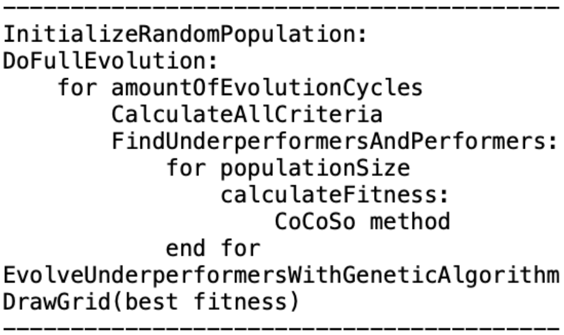
λ is a constant that is manually selected, should be between 0 and 1, and defines the weight between *R* and *P*. The research defines λ as 0.5 to have a neutral impact on both components.

5. Finally, the ranking of alternatives is determined:

$$k_i = (k_{ia} k_{ib} k_{ic})^{\frac{1}{3}} + \frac{1}{3}(k_{ia} + k_{ib} + k_{ic}). \tag{2.19}$$

2.4.1. Neutrosophic genetically combined compromise solution procedural game scene generator

CoCoSo is employed to compute the ultimate fitness score for each chromosome in a genetic algorithm generation. The criteria associated with each chromosome (Fig. 2.16) are utilised to form the decision matrix.



```
-----------------------------------------
InitializeRandomPopulation:
DoFullEvolution:
    for amountOfEvolutionCycles
        CalculateAllCriteria
        FindUnderperformersAndPerformers:
            for populationSize
                calculateFitness:
                    CoCoSo method
            end for
EvolveUnderperformersWithGeneticAlgorithm
DrawGrid(best fitness)
-----------------------------------------
```

Fig. 2.16. CoCoSo method integration

The central component of the procedural generator is a genetic algorithm. The original MCDM alternative(i) and criteria(j) matrix are converted into a three-dimensional grid. In the MCDM algorithm, the alternatives correspond to the population and chromosomes(i) in the genetic algorithm, while the third dimension introduces generations(j) in the time layer (Fig. 2.17).

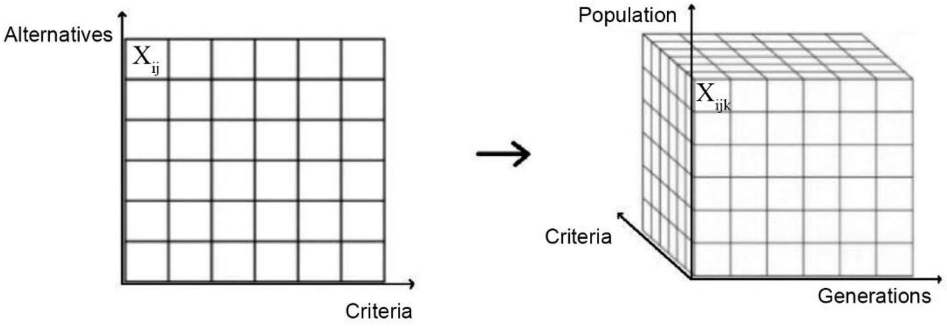


Fig. 2.17. Grid transformation

After a fixed number of generations, the algorithm generates the layouts for the game levels. It starts by initialising random-level layouts and then evolves them through the genetic algorithm. Criteria scores are computed and normalised to fit within the range of 0.1 to 0.9. The research avoids getting too close to 0 or 1 to ensure a more reliable outcome for the MCDM algorithm. The set of criteria comprises seven factors. The first part consists of validation criteria, which verify the existence of the player, exit, and an available path between them. The second part includes optimisable criteria, namely symmetry, empty space balance, player-exit distance, and safe zone calculation. Subsequently, research calculates the final fitness score using the CoCoSo method and applies genetic operators to the population. For each generation, the fitness calculation process is repeated.

The research utilises linear conversion from crisp to neutrosophic numbers to prevent non-linear differences from accumulating over subsequent generations:

$$N = \begin{cases} t \ = \ C \\ i \ = \ 1 - C \\ f \ = \ 1 - C \end{cases}, \tag{2.20}$$

The outcome may become biased since the modification could exceed the evolved difference within a single generation.

The normalisation process is adjusted to utilise the smallest criterion values across all generations rather than relying solely on values generated within a single generation. To achieve this, research manually normalises each criterion beforehand to fit within the 0–1 range. The approach is necessary because future generation values cannot be predicted, and it ensures that normalisation is consistently applied globally for each generation:

$$0 \ < X_{ik} < \ 1. \tag{2.21}$$

There is an option to avoid general normalisation by normalising each criterion independently. This enhancement also enables the bypass of the need to store separate data sets of neutrosophic sets in memory. Instead, research incrementally adds and multiplies them to generate the Ri and Pi values without having to store individual elements for each one:

$$R_i = \sum_{j=1}^{nl} \left(w_j (N_{ij} \right), \tag{2.22}$$

$$P_i = \sum_{j=1}^{nl} \left(N_{ij} \right)^{w_j}, \tag{2.23}$$

where l is the number of generations for the genetic algorithm, and n is the population size. Values r_{ij} are replaced by the values $\mathrm{N_{ij}}$ of the neutrosophic sets.

This improvement also permits us to derive an approximate constant “A” for the divisor in the kia formula, eliminating the need to compute a sum for each chromosome in the generation. As a result, the performance of the iterative algorithm is significantly enhanced without any noticeable compromise in quality.

$$A = \frac{\sum_{i=1}^{nl} x_i}{nl}, \tag{2.24}$$

$$d = nA, \tag{2.25}$$

$$k_{ia} = \frac{S(R_{ik}) + S(P_{ik})}{d}. \tag{2.26}$$

It uses global min (r_1, r_2) and max (p_1, p_2) values of R_i and P_i, which are calculated before revealing future values based on the normalisation of the criteria formula:

$$r_1 = \underset{i \in nl}{min}(R_i), \tag{2.27}$$

$$r_2 = \underset{i \in nl}{min}(P_i), \tag{2.28}$$

$$p_1 = \underset{i \in nl}{max}(R_i), \tag{2.29}$$

$$p_2 = \underset{i \in nl}{max}(P_i), \tag{2.30}$$

so it would be consistent in interactions between generations:

$$k_{ib} = \frac{S(R_{ik})}{r_1} + \frac{S(P_{ik})}{r_2}, \tag{2.31}$$

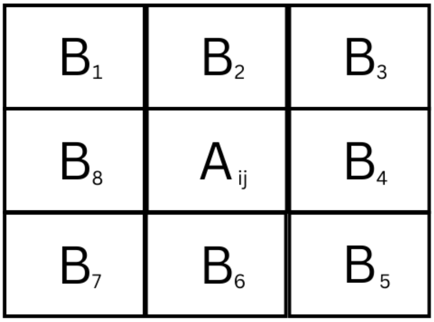
$$k_{ic} = \frac{\lambda S(R_{ik}) + (1-\lambda) S(P_{ik})}{\lambda p_1 + (1-\lambda) p_2}. \tag{2.32}$$

Finally, research uses the original k_i formula to calculate the final fitness for a single chromosome.

2.5. Modelling of Gestalt principles for the fitness function

The computational creativity framework comprises two steps: developing a mathematical model and applying a numerical algorithm. The primary focus in developing the mathematical model is to find a way to algorithmically incorporate abstract visual aesthetic evaluations to automatically evolve functional game levels. A specific set of Gestalt principles is selected to guide this process. However, using too many rules in a confined area can lead to an overcrowded and less distinguishable result. Hence, the chosen principles are those that exhibit the highest visibility and coherence within the context of the tools environment. Five Gestalt principles – Similarity, Proximity, Continuity, Focal Points, and Common Region – are integrated into the model as fitness criteria functions. These principles are applied to a rectangular matrix of symmetric game objects using focal functions. Certain Gestalt principles are omitted from the model due to their complexity and the requirement for more intricate object systems and assignable traits for individual objects. Closure and Figure Ground criteria are skipped as they demand identifiable structures, which can be disrupted by other criteria. The Closure criterion relies on unfinished identifiable structures, which cannot be identified if these structures are absent. The Figure Ground principle requires the separation of foreground and background objects or the identification of larger wholes, which may not be feasible with the abstract patterns generated by the algorithm. Another criterion, Common Fate, necessitates objects with direction vectors, but the objects considered in this research are symmetrical and lack such directional properties. Five compatible Gestalt principles are implemented with the core generation algorithm and object structure to guide the automatic evolution of functional game levels.

The mathematical model for each criterion involves iterating over the entire grid, excluding the bounding rows and columns. Each criterion is represented as a global function using raster algebra. This procedure is applied to all five Gestalt principles. For each cell, research constructs a focal function with a square zone (Fig. 2.18).



B_1	B_2	B_3
B_8	A_{ij}	B_4
B_7	B_6	B_5

Fig. 2.18. Cell vision grid

The matrix cell values represent various situations: 0 denotes the player's starting spot, 1 indicates the exit that the player needs to reach to finish the level, 2 represents empty space, 3 stands for walls that block the player's movement, 4 symbolises enemies or hazardous zones to avoid, and 5 signifies collectable zones that are beneficial for the player to step into. The centre cell in the neighbourhood is marked as "A", while the surrounding cells are marked as "B". The indexes "i" and "j" indicate the location of the cell in the game's scene matrix. The general criteria normalisation formula is the same for all modelled Gestalt criteria and can be expressed as a formula:

$$s_m = 0.9 \frac{t_m}{r * n_x * n_y}. \tag{2.33}$$

The research aims to ensure that the final value for each criterion remains within the range of 0 to 1 without approaching the edges of the range. Research stabilises the neutrosophic algebra by multiplying the value by 0.9.

The members of the formula are defined as follows: "s" represents the single-criteria fitness score, "r" denotes the relevant neighbours (r = 8 for all criteria, except the continuity criterion, where it is r = 1 due to a specific condition that requires exactly two identical neighbours). "nx" and "ny" determine the size of the matrix grid, "t" represents the total relevance of the criteria before normalisation, and "m" represents the index of the criterion from the Gestalt criteria list. The mathematical models for each of the considered Gestalt principles are presented below:

When the object types in the neighbourhood of a cell are the same, the similarity score for that particular cell is increased. Once the iteration over all cells is finished, the number of similar object pairs is divided by the total number of possible pairs (the product of the total matrix cells and eight). This criterion aims to identify clusters or chunks of similar object areas, as depicted below (Fig. 2.19).

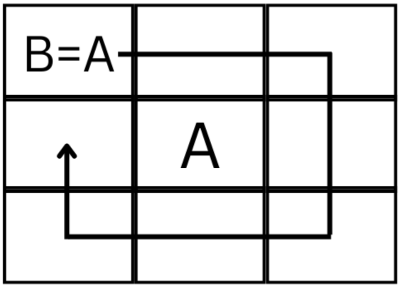


Fig. 2.19. Similarity evaluation for a single cell

$$t_1 = \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \sum_{k=1}^{8} \begin{cases} 1, & B_{ijk} = A_{ij} \\ 0, & otherwise \end{cases}. \tag{2.34}$$

The proximity criterion shares a similar algorithmic basis with the similarity criterion, but in this case, one of the pair members can also be identified as an empty space for the score value to increase (Fig. 2.20). This criterion is designed to detect islands or clusters of similar objects in the area.

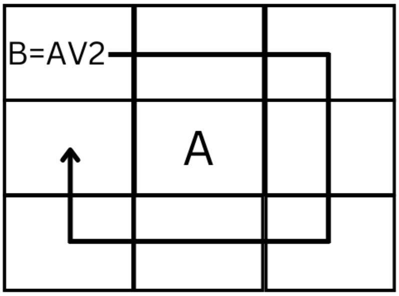


Fig. 2.20. Proximity evaluation for a single cell (number 2 identifies empty space)

$$t_2 = \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \sum_{k=1}^{8} \begin{cases} 1, & B_{ijk} = A_{ij} \vee 2 \\ 0, & otherwise \end{cases} . \quad (2.35)$$

The continuity criterion involves two score values for each level of the cycle. The calculation of the total continuity score follows a similar approach as the total score value in other criteria. Focal continuity increases only if the cell, along with its surrounding cells, meets the continuity requirement. For each cell, it is compared to the adjacent horizontal and vertical cells. Corner cells are excluded from consideration because, in a smaller grid matrix consisting of squares, diagonal continuity might be less apparent as they only touch at a single point, potentially breaking the illusion of continuity (Fig. 2.21).

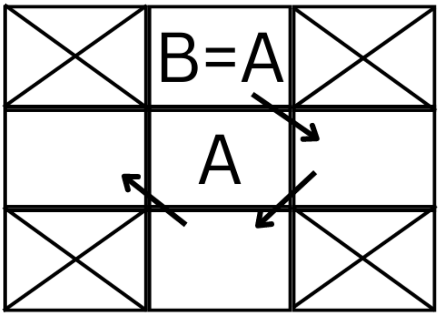


Fig. 2.21. Continuity evaluation for a single cell. Exactly two out of four neighbouring cells must satisfy the B=A criteria

Each cell verifies the presence of exactly two touching similar object types. If there are more or fewer similar touching objects, the cell is not considered continuous (Fig. 2.22).

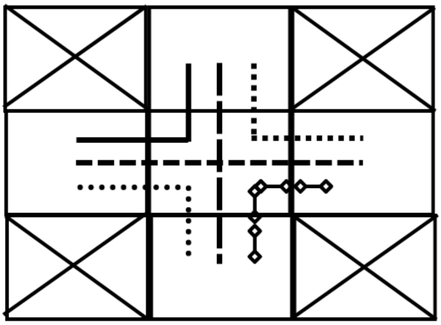


Fig. 2.22. Possible Continuity connections. The lines in the centre cell show which B=A patterns increase continuity

$$t_3 = \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \begin{cases} 1, & (\sum_{k=1}^{4} \begin{cases} 1, B_{ij\,2k} = A_{ij} \\ 0, otherwise \end{cases}) = 2 \\ & 0, otherwise \end{cases} . \tag{2.36}$$

The Focal Point criterion determines the overall extent of the focal point effect. This is accomplished by examining how much each cell is surrounded by the continuity of a different object (where B≠A). The algorithm assesses each object and calculates the number of its repetitions in the surrounding eight cells. The highest number of repetitions is then used as the score for the cell. This algorithm seeks to identify recognisable points or focal points within the game-level pattern (Fig. 2.23).

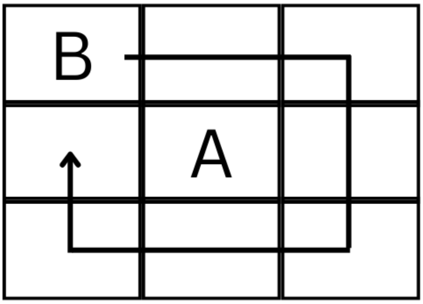


Fig. 2.23. Focal point evaluation for a single cell. B≠A

$$t_4 = \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \max_{l\in[c]} \sum_{k=1}^{8} \begin{cases} 1, (B_{ijlk} \neq A_{ij}) \wedge (B_{ijlk} = B_{ijl}) \\ \quad 0, \quad otherwise \end{cases} , \tag{2.37}$$

c defines the number of possible object types.

The last implemented criterion is Common Region, which operates algorithmically in a similar manner as Proximity, but it searches for boundaries defined by walls instead of empty space (Fig. 2.24). This criterion aims to identify bounded groups or clusters of similar objects within the area.

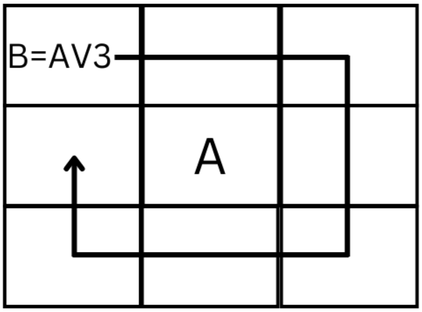


Fig. 2.24. Common region evaluation for a single cell (number 3 identifies walls)

$$t_5 = \sum_{i=2}^{n_x-1} \sum_{j=2}^{n_y-1} \sum_{k=1}^{8} \begin{cases} 1, & B_{ijk} = A_{ij} \vee 3 \\ 0, & otherwise \end{cases} . \tag{2.38}$$

Research can achieve aesthetically pleasing game scene designs by employing the proposed mathematical model based on Gestalt principles. In implementing the computational creative genetic algorithm, research integrates these aesthetic criteria of Gestalt principles into the fitness function. This is done using the Pythagorean neutrosophic WASPAS extension to evaluate the game scenes. The calculated scores are then used to evolve the game scenes through a genetic algorithm.

2.5.1. Game scene generation incorporating Gestalt principles

The game scene generation uses the proposed genetic Pythagorean neutrosophic WASPAS approach. The mathematical model consists of multiple criteria categorised into three types. The first group comprises aesthetic criteria consisting of high-level criteria governed by Gestalt rule derivatives (s) and low-level criteria (v), such as symmetry and empty space balance. The second group includes functional criteria (f), which optimise the evaluations of the game design, such as the distance between key objects and the safe zones around them. The last group comprises constraint criteria (c), which must always hold for the fitness to be non-zero. These constraint criteria involve criteria related to the existence of essential elements and the availability of a feasible path between the player and the exit.

When dealing with multiple criteria, developing a strategy for combining them into a single fitness value becomes essential. Key considerations include determining appropriate weights and choosing the appropriate algebraic method to combine the different criteria. This process is an integral part of fitness function modelling. While most literature examples focus on defining individual criteria, there is untapped potential for improving algorithm effectiveness by exploring the combination of criteria. The proposed modelling approach based on Gestalt principles is integrated into the genetic algorithm. The fitness function incorporates all the criteria from the mathematical model, employing Multiple Criteria Decision Making (MCDM) methods. These methods enable the effective use of numerous criteria for fitness evaluation. To evaluate the generated game level, the criteria and alternative matrix are assembled:

$$X = \begin{bmatrix} S_{11} & S_{12} & \dots & S_{1n} \\ S_{21} & S_{22} & & S_{2n} \\ \vdots & & \ddots & \vdots \\ S_{m1} & S_{m2} & \cdots & S_{mn} \end{bmatrix} . \tag{2.39}$$

Each alternative column Sj is generated using a genetic algorithm and represents a chromosome in the genetically generated population. The complete set of Si for each alternative encompasses 12 criteria (Table 2.2). In the evaluation process, most criteria are normalised to fit within the range of 0 to 1, while constraint evaluations are binary. If the chromosome does not meet the required criteria, the final result for that chromosome is multiplied by 0.

Table 2.2. Full criteria list

Criteria	Type	Value Range
Similarity (s_1), max	Aesthetic value (focal function)	0-1
Proximity (s_2), max	Aesthetic value (focal function)	0-1
Continuity (s_3), max	Aesthetic value (focal function)	0-1
Focal Point (s_4), max	Aesthetic value (focal function)	0-1
Common Region (s_5), max	Aesthetic value (focal function)	0-1
Symmetry (v_1), max	Aesthetic value (global function)	0-1
Empty space balance (v_2), max	Aesthetic value (global function)	0-1
Player-exit distance (f_1), max	More area of the game scene is explored by the player	0-1
Safe space (f_2), max	Key areas do not have hazardous objects nearby	0-1
Player exists (c_1) Boolean	The level is playable	0 or 1
Exit exists (c_2) Boolean	The level is playable	0 or 1
Player-exit path exists (c_3) Boolean	The level is playable	0 or 1

Each criterion value is transformed into a Pythagorean neutrosophic set, which exhibits better correlation compared to a single-valued neutrosophic set (Radha et al., 2021). A Pythagorean Neutrosophic Set combines the concepts of Pythagorean fuzzy sets and neutrosophic sets. Pythagorean fuzzy sets generalise fuzzy sets and assign three values to each element: membership degree, nonmembership degree, and hesitancy degree. On the other hand, neutrosophic sets generalise fuzzy sets to handle uncertain, indeterminate, and inconsistent information. In a Pythagorean neutrosophic set, each element is assigned three values: truth-membership degree, indeterminacy degree, and falsity-membership degree. These values represent the degrees of truth, indeterminacy, and falsity of the element, respectively. The truth-membership degree and falsity-membership degree always sum up to one, while the indeterminacy degree can take any value between 0 and 1. Pythagorean neutrosophic sets have found applications in decision-making problems involving uncertain and inconsistent information, demonstrating promising results in handling such situations (Bausys et al., 2022). The addition of neutrosophic sets to the algorithm increases the nondeterminism of the criteria evaluation. They enable the expression of information about neutrality and generalise fuzzy and intuitionistic fuzzy sets (Smarandache, 1999).

An entity represented as A, which is a Neutrosophic Pythagorean set on the universe R, comprises interdependent Neutrosophic Pythagorean components T and F, along with an independent component U:

$$A = \{< x, T_A, U_A, F_A >: r \in R\}, \tag{2.40}$$

$$(T_A)^2 + (F_A)^2 \leq 1, \tag{2.41}$$

$$(T_A)^2 + (U_A)^2 + (F_A)^2 \leq 2. \qquad (2.41)$$

Here, $\mathrm{T_A}(x)$ is the truth membership, $\mathrm{U_A}(x)$ is the indeterminacy membership, and $\mathrm{F_A}(x)$ is the false membership.

Next, the neutrosophic sets are integrated using a multi-criteria weighted aggregated sum product assessment method (WASPAS) (Petrovas and Bausys, 2022). This approach leads to the calculation of a joint generalised criterion:

$$Q_i = (0.5 \cdot \sum_{j=1}^{n} \tau_{ij} \cdot w_j) \oplus (0.5 \cdot \prod_{j=1}^{n} \tau_{ij} \odot^{w_j}). \tag{2.42}$$

Q is a combined neutrosophic number for one genetically generated alternative. *n* is the number of criteria; *x* is a single-criterion fitness score expressed in a neutrosophic number, and *w* is a single-criterion weight. The Pythagorean neutrosophic number is converted to a crisp number by the score function:

$$S(Q_i) = \frac{3 + 3\xi^2 - 2\vartheta^2 - \eta^2}{6}. \tag{2.43}$$

ξ, ϑ, η are intermediacy, truth, and falsity members of a neutrosophic number. The final WASPAS result is then used as an evaluation score. The algebraic operations are as follows:

$$\tau_1 \oplus \tau_2 = \langle (1 - (1 - \xi_1^2)(1 - \xi_2^2))^{\frac{1}{2}}, \vartheta_1 \vartheta_2, \eta_1 \eta_2 \rangle,$$

$$\tau_1 \otimes \tau_2 = \langle \xi_1 \xi_2, (1-(1-\vartheta_1^2)(1-\vartheta_2^2))^{\frac{1}{2}}, (1-(1-\eta_1^2)(1-\eta_2^2))^{\frac{1}{2}} \rangle,$$

$$\lambda \cdot \tau_1 = \langle (1-(1-\xi_1^2)^{\lambda})^{\frac{1}{2}}, \vartheta_1^{\lambda}, \eta_1^{\lambda} \rangle,$$

$$\lambda \odot \tau_1 = \langle \xi_1^{\lambda}, (1-(1-\vartheta_1^2)^{\lambda})^{\frac{1}{2}}, (1-(1-\eta_1^2)^{\lambda})^{\frac{1}{2}} \rangle.$$

The triplet $\tau = \{\xi,\vartheta,\eta\}$ is called a Pythagorean neutrosophic number, and λ is a real number.

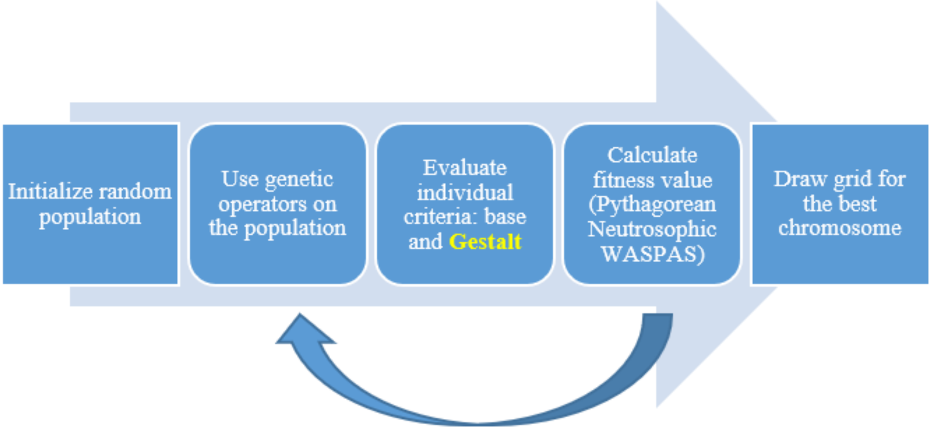

Initialize random population
Use genetic operators on the population
Evaluate individual criteria: base and Gestalt
Calculate fitness value (Pythagorean Neutrosophic WASPAS)
Draw grid for the best chromosome

Fig. 2.25. Game scene generation by Gestalt criteria (made by the author)

Each execution of this algorithm typically results in a different pattern of game object composition due to the various possible fitness points that can be converged upon. This variability leads to creativity through diversity. Additionally, the initial values influence the convergence points, as the answers depend on the relative object positions determined by focal functions. The game scene's initial layouts are initialised with random values and then evolved using a genetic algorithm (Fig. 2.25). The initial population for the genetic algorithm comprises a set of randomly generated game scene levels. Genetic operators, such as selection or mutation, are employed to remove poorly performing levels and generate new ones. The scoring for each population member is determined by the MCDM algorithm using game design criteria. This research proposes novel evaluation criteria based on Gestalt principles for aesthetic pattern evolution, resulting in distinct game object layout patterns. The genetic algorithm loop is repeated for a fixed number of iterations (usually 500, and it rarely goes beyond 2000 with a 10×10 game scene resolution). The functional criteria and restrictions used in the study were reused from previous research.

2.6. Regional object transformation

The proposed algorithm introduces modifications to the “Draw” and “Decode” components of the original algorithm and incorporates a regional morpher. It iterates over the matrix of generated objects and assigns objects to spawn in the game scene. Additionally, each chromosome is assigned a random corner point in the matrix, from which changes radiate towards other areas of the level (Fig. 2.26).

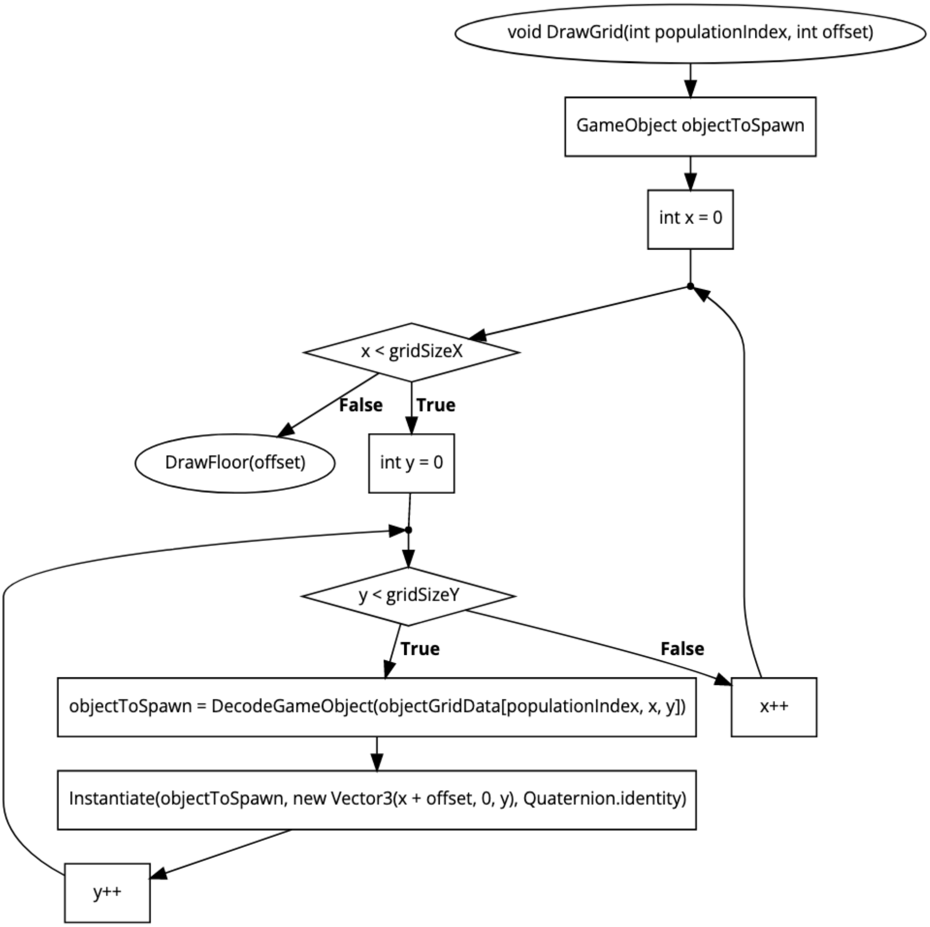

void DrawGrid(int populationIndex, int offset)
GameObject objectToSpawn
int x = 0
x < gridSizeX
False
True
DrawFloor(offset)
int y = 0
y < gridSizeY
True
False
objectToSpawn = DecodeGameObject(objectGridData[populationIndex, x, y])
x++
Instantiate(objectToSpawn, new Vector3(x + offset, 0, y), Quaternion.identity)
y++

Fig. 2.26. Iteration over the game objects grid

In the subsequent modified part of the algorithm, a game object is chosen from the grid, and a model is assigned to it based on the chromosome number associated with each cell of the grid. Instead of visual object selection, an object morph algorithm is utilised. In this experiment, the algorithm is tested using empty space and wall objects (Fig. 2.27). Each identification number corresponds to a specific object type. Arrays of objects in the visual array can be adjusted, and three objects are used for each game object type during testing. Additionally, research varied the resolution of the object grid to observe its impact on the outcome.

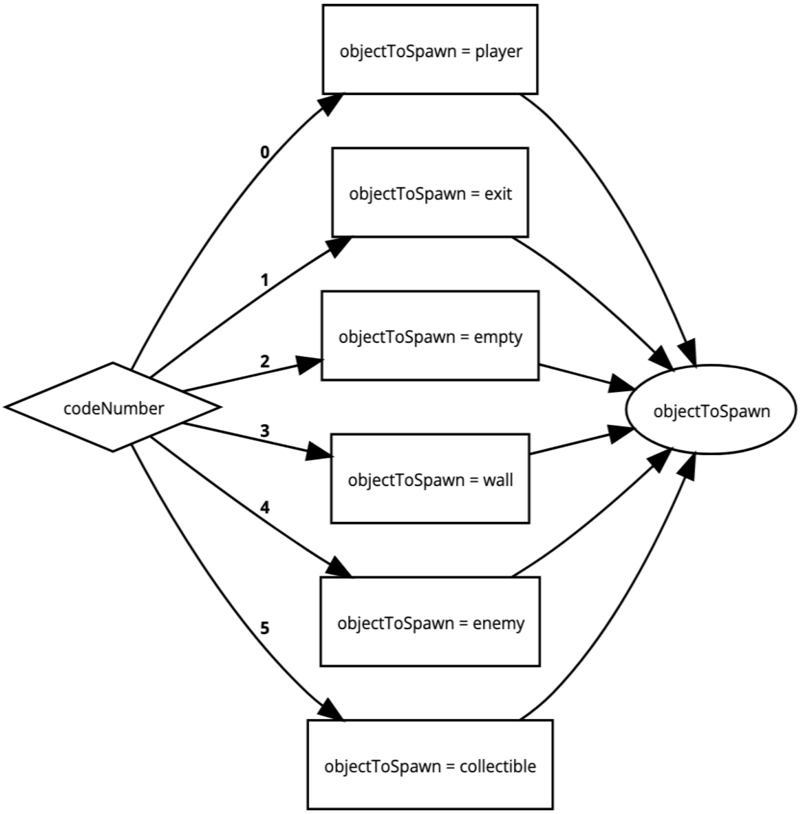


Fig. 2.27. The inclusion of object morphed into a decoder

The morphing algorithm picks a visual object from the pool of available objects and determines its relative position concerning the chosen corner point. This relative position is then multiplied by a pair of random numbers, one ranging from 0 to 1 and the other from 0.5 to 1, to introduce additional variability. For the negative axis sides of the grid, multipliers between 0 to 1 and 1 to 1.5 are used (Fig. 2.28).

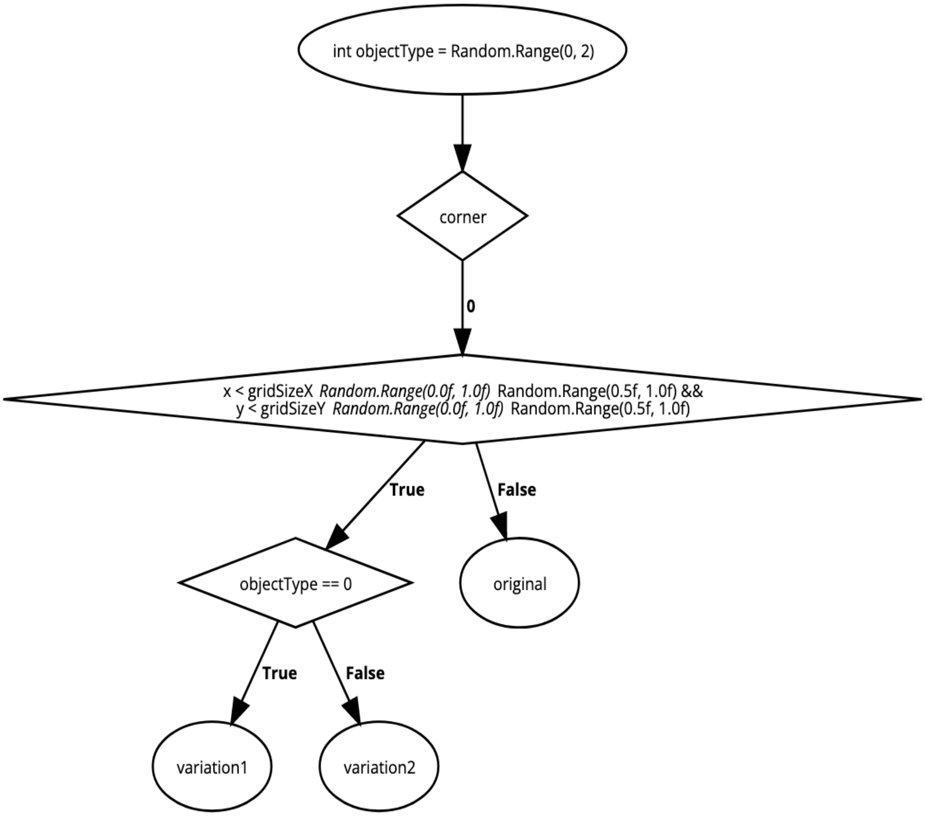

int objectType = Random.Range(0, 2)
corner
0
x < gridSizeX *Random.Range(0.0f, 1.0f)* Random.Range(0.5f, 1.0f) &&
y < gridSizeY *Random.Range(0.0f, 1.0f)* Random.Range(0.5f, 1.0f)
True
False
objectType == 0
original
True
False
variation1
variation2

Fig. 2.28. Visual object selection and noise function

The equation to decide if the object should morph consists of two lines for each axis, which compares object coordinates with the grid length of one axis (Ax, Ay), random number ® and noise (N_1). There are four checks with each possible comparison (*x1>x2*, *y1>y2*; *x1<x2*, *y1<y2*, *x1>x2*, *y1<y2*, *x1>x2*, *y1<y2* (*x1*, *y1* – object coordinates; *x2*, *y2* – randomness element)):

$$\begin{cases} Ax * R * N_1 \\ Ay * R * N_1 \end{cases}. \tag{2.44}$$

By using this approach, research can maintain the integrity of all the original algorithm rules while enhancing the visual diversity aspect. The same algorithm is applied to each type of object, allowing us to expand the variety of visual representations without compromising the underlying rules of the algorithm. The new levels generated with the addition of this algorithm have increased variety due to alternative visual models for the same functional objects.

2.6. Conclusions of the Second Chapter

1. The evaluation phase employs a modified WASPAS-SVNS algorithm, adapting it for iterative use within the genetic algorithm-based framework. The work introduces an iterative approach to enhance compatibility. By consolidating and normalising criteria evaluation data, followed by the transformation into neutrosophic sets, the research establishes a comprehensive method for assessing fitness across various dimensions. This innovative process empowers the genetic algorithm to effectively evaluate and evolve alternative solutions, culminating in the generation of dynamic and engaging game scenes that align with predefined design criteria.
2. The CoCoSo method presents a structured approach to compute fitness scores within the genetic algorithm-based procedural game scene generator. By transforming the MCDM alternatives and criteria into a three-dimensional grid, research establishes a dynamic framework for evolving game-level layouts. Through the iterative application of these techniques, research achieves a refined and efficient generation process, resulting in high-quality game scenes that adhere to defined design criteria. This synergistic fusion of MCDM and genetic algorithms showcases the potential for computational creativity in video game design.
3. The model's core objective is to algorithmically integrate abstract visual aesthetics into the evolution of functional game levels. Guided by select Gestalt principles, this process emphasises visibility and coherence. Five Gestalt principles are embedded as fitness criteria functions in a matrix of symmetric game objects. A mathematical model spanning various criteria is devised using raster algebra. Neutrosophic algebra stabilises the model, ensuring criterion values within a specific range. The process culminates in a genetic algorithm that generates game scenes, reflecting diverse aesthetic patterns and layouts.
4. The proposed algorithm introduces substantial enhancements incorporating a regional morpher for greater visual diversity. The process involves matrix iteration for spawning game objects, with each chromosome assigned a corner point radiating changes across the level. A novel object morphing algorithm replaces visual selection, varying object types using multipliers and random numbers. The algorithm ensures the preservation of original rules while significantly expanding the visual variety applicable across different object types. This approach offers a promising avenue for achieving aesthetic and functional complexity in-game scene generation.

3

Experimentation and Results of the Proposed Procedural Generation Methods

This chapter presents an investigation and experimentation of the proposed genetic neutrosophic MCDM methods with Gestalt criteria-modelled cellular automata agents introduced in the second chapter of this dissertation. Novel procedural generation strategies were developed to address creativity problems: Incorporation of high-level aesthetic concepts into precise mathematical algorithms, scene object pattern diversity and uniqueness, and combination of conflicting aesthetic and functional criteria without breaking the final coherence of a game scene. The research results obtained by assessing the proposed procedural generation strategies are discussed in detail.

The main research results of this chapter were published in four author's scientific publications (Petrovas and Bausys, 2022; Petrovas, Bausys, Zavadskas and Smarandache, 2022; Petrovas, Bausys and Zavadskas, 2023; Petrovas and Bausys, 2022), and findings of the research were presented at one international conference (International Conference on Electrical, Computer and Energy Technologies (ICECET), Prague, Czech Republic, 2022).

3.1. Procedural video game scene generation by genetic and neutrosophic multi-criteria decision-making algorithms

This research developed a framework from scratch using the Unity game engine and obtained visual game object assets from the Unity Asset Store. The results were generated using a custom C# script. The tests were conducted on a computer equipped with a 2.4 GHz 8-Core Intel Core i9 CPU. The procedural generator with neutrosophic evaluation quickly generates rising scores for the first 100–200 generations under the current conditions compared to the summation of individual criteria fitness scores. However, the generator often requires more time to create symmetrical and visually balanced scene layouts while ensuring that game rules are applied. The final fitness score typically settles around 0.75–0.85. Having many local maxima for the game scene generation is crucial, as it ensures that the results are unique and diverse from one another. Numerous possible solutions exist based on the random initial seed and mutations. Examples of fitness scores with different seeds of random initial data and 500 generations are presented in Figure 3.1. The choice of the number of generations was determined based on the point at which the algorithm iterations become stagnant, resulting in either no updates or only negligible updates to the results.

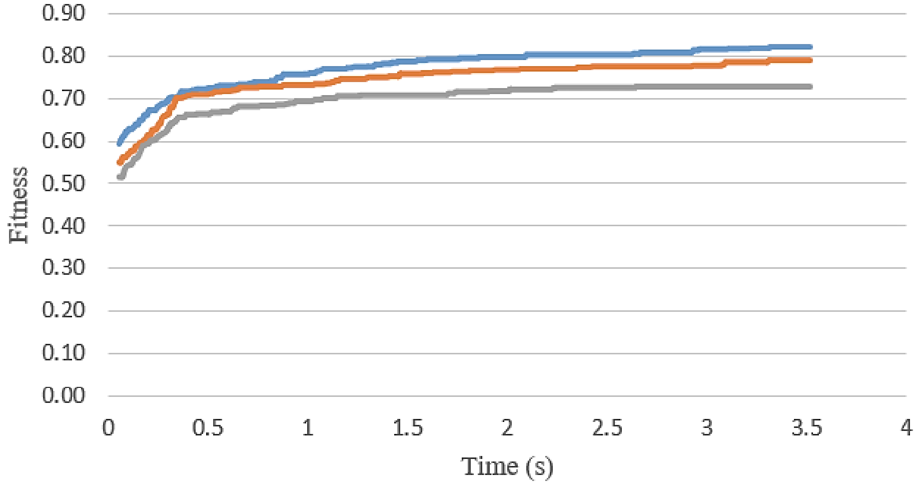


Fig. 3.1. Fitness evolution examples, each curve represents a separate execution of the algorithm

Please note that achieving a fitness close to 1 is not possible since the criteria often conflict with each other. Fitness scores typically start to converge after approximately 500 to 2000 generations. The average time taken to calculate one 10×10 grid level with 2000 generations is around 21 seconds. The initial population of 50 is relatively low compared to the total number of possible scene layouts (7^100), and mutations are set to 5%. The choice of a lower population size aims to produce a broader range of possible solutions. There is a linear dependency of execution time and population size (a population of size 50 executes around 3.5 seconds, and a population size of 100 executes around 7 seconds). The objective is not to optimise the algorithm for a single solution but to generate a diverse set of level layouts that satisfy both creativity and game design criteria.

Visual results demonstrate the generation of aesthetically appealing game scenes with elements of symmetry and space balance. Examples include room-like shapes without specific code defining what constitutes a room (Fig. 3.2). As symmetry conflicts with other criteria and lacks a strict definition, semi-symmetric shapes are also observed (Fig. 3.3). Additionally, the generator can create scenes with a smaller room containing numerous coins or rewards (Fig. 3.4) and game scenes without many walls (Fig. 3.5). These four examples do not use Gestalt principles yet. The generator showcases the ability to produce a diverse array of aesthetic shapes in the game scenes.

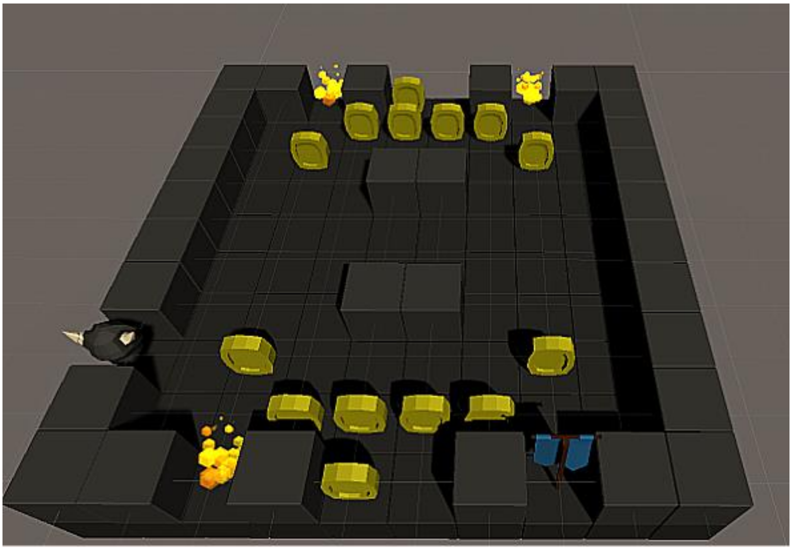


Fig. 3.2. Room-like generated scene

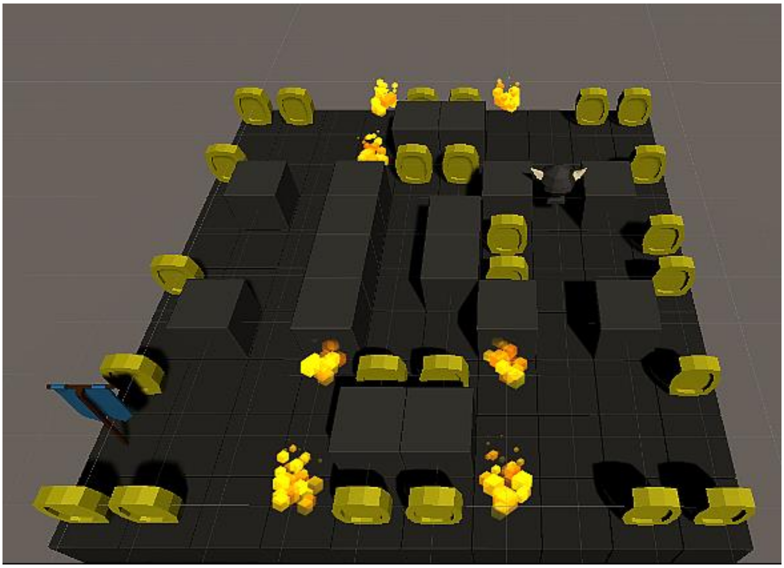


Fig. 3.3. Semi-symmetric results

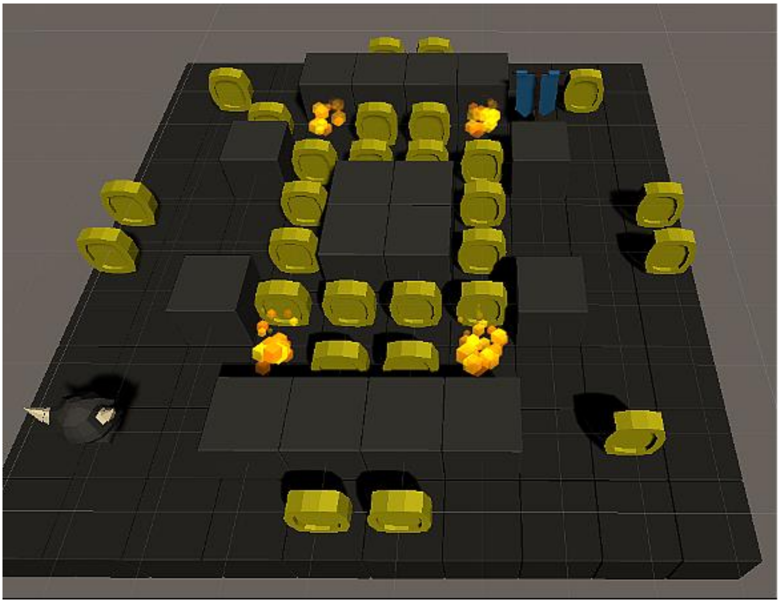


Fig. 3.4. Small room inside the scene

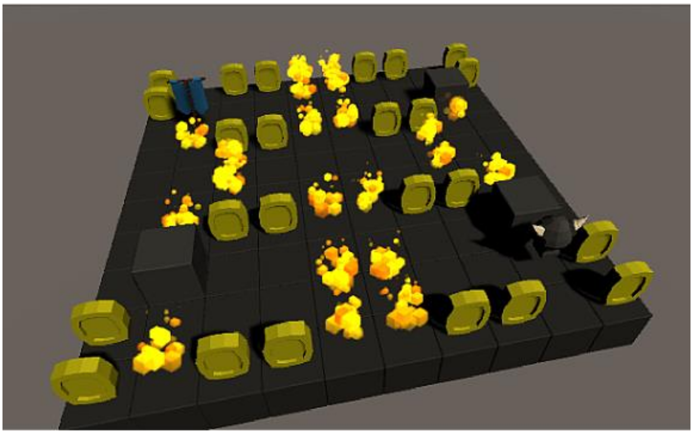


Fig. 3.5. Game scene with almost no walls

An illustration of intermediate evolution results (Fig. 3.6). The grid is displayed every 100 generations. It shows a chaotic layout and rapid progression in the initial stages, followed by fine-tuning and refinement in the later generations.

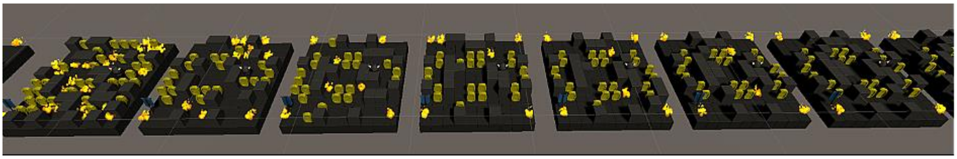


Fig. 3.6. Intermediate evolution results (left to right)

Upon closer examination, the implementation of the aesthetic criterion can be observed in the visual examples (Fig. 3.7), showcasing symmetry and balance in the empty space. Simultaneously, the game design requirements, such as pathfinding, are effectively realised.

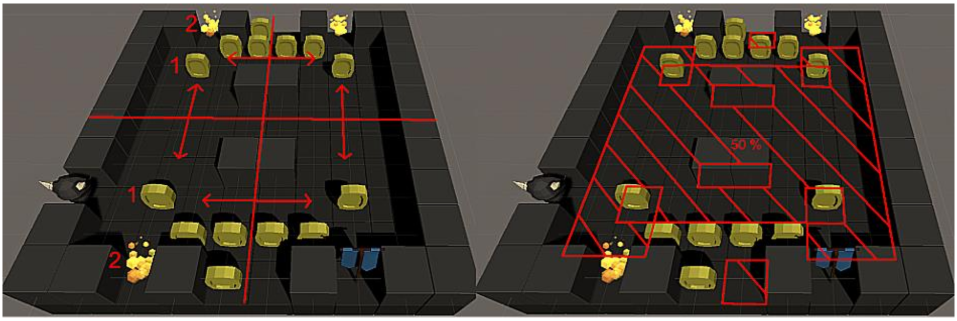


Fig. 3.7. Symmetry and empty space balance

Each result represents a distinct local maximum among numerous possible outcomes. The algorithm's randomness and the vast number of potential solutions make it highly unlikely to generate an identical game level. An alternative approach could involve creating a basic initial room and allowing the algorithm to fine-tune it to meet aesthetic and playability criteria. The optimal solution for this task is achieving complete satisfaction with the proposed criterion under a given random initial seed. Compared to other similar research in the field, the proposed framework generates more visually noticeable aesthetic traits on small object resolutions while maintaining an above-average object pool. This procedural generation approach has the potential to expedite and simplify creative work.

The main problem addressed by the proposed method is how to increase the number of unique and non-repetitive game levels with multiple runs of the same algorithm. The results presented in the study show that the generated levels exhibit interesting game scene layouts that differ with each run. The algorithm can create both aesthetic and functional level layouts simultaneously. Furthermore, developers have the flexibility to interchange visual representations of game assets. The WASPAS-SVNS algorithm facilitates the evaluation of conflicting criteria, enabling a comprehensive assessment of game-level quality. The proposed approach breaks down design principles into primary elements and defines them using the criteria list. The algorithm begins with a random shape and then sculpts a functional and aesthetic game level around that shape. The random nature of the genetic algorithm ensures surprise elements in the levels. Finding a balance between the weights of different criteria and the number of criteria defining a specific objective is crucial to generating a coherent final level. An excessive or insufficient number of features may dull some of the game design elements. Creativity assistance algorithms have the potential to save time for game designers and developers, although currently, most commercial games utilise only light game design assistance tools, such as procedurally generated or handcrafted levels. The proposed work can be expanded by combining it with an algorithm that breaks down design elements from handcrafted game levels and utilises them as a basis for the criteria list. This could further enhance the creative possibilities and efficiency of the procedural generation process.

3.2. Generation of creative game scene patterns by the neutrosophic genetic combined compromise solution method

The algorithm was built upon the original WASPAS procedural scene generator by Petrovas and Bausys (2022). The computations were performed on an 8-core CPU with a speed of 2.4 GHz. The fitness scores start stabilising and converging after approximately 80–120 generations, displaying quicker convergence compared to the WASPAS implementation. However, direct comparison of fitness scores is not feasible as the CoCoSo method produces values outside the 0–1 range. In our demo, the fitness values typically range between 4 and 8, exhibiting a tighter relative value range. A comparison of the fitness curves over time between the CoCoSo and WASPAS methods is depicted in (Fig. 3.8).

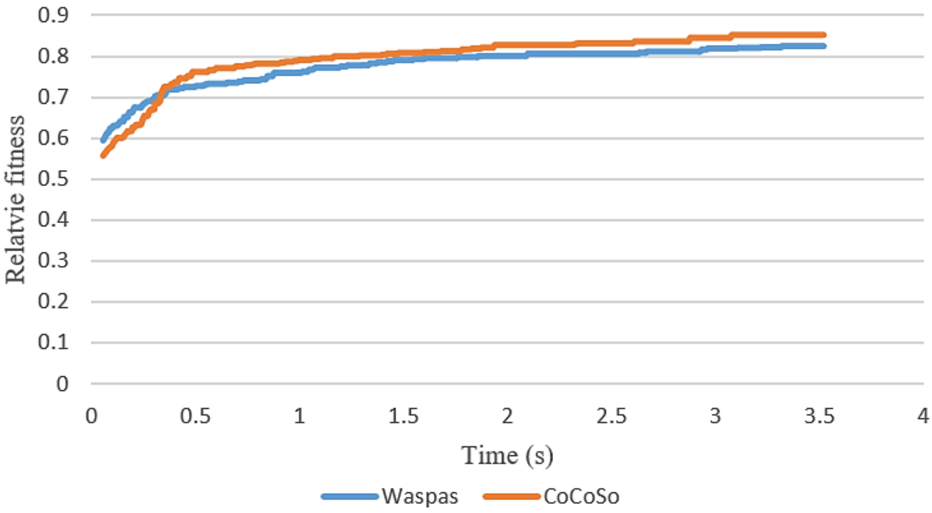


Fig. 3.8. Fitness curve comparison between WASPAS and CoCoSo methods

The execution of 2000 generations typically takes approximately 23 seconds. However, it is usually sufficient to have satisfying results with the CoCoSo method after around 350 generations, allowing for potential time savings with fewer generations. If the number of objects is increased, calculation time increases exponentially. Visual results generated by the algorithm were generated (Figs. 3.9, 3.10, 3.11, and 3.12). It is evident that the generated game scene-level layouts successfully fulfil both aesthetic and functional criteria. Moreover, the patterns for these two simultaneous results differ from each other.

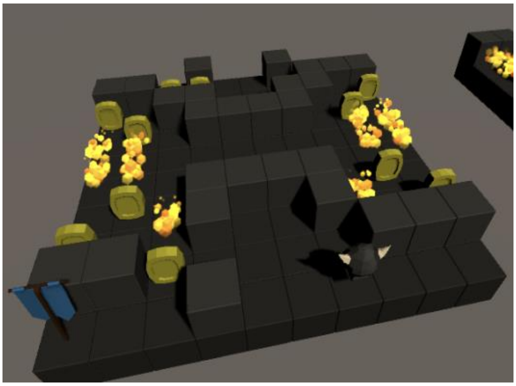


Fig. 3.9. CoCoSo generated example 1

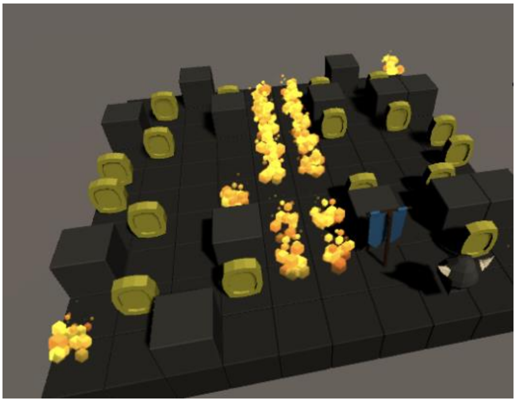


Fig. 3.10. CoCoSo generated example 2

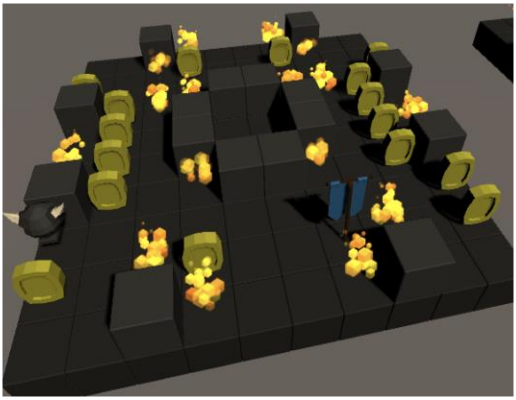


Fig. 3.11. CoCoSo generated example 3

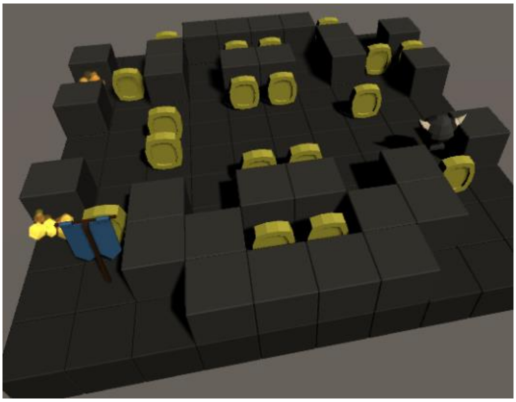


Fig. 3.12. CoCoSo generated example 4

The proposed algorithm can generate intriguing patterns faster than the original algorithm. However, this comes at the expense of approximately 10% increased calculation time for the same number of generations and population. Notably, even in the initial stages of the algorithm with a lower generation count, the desired patterns begin to take shape, as depicted in Fig. 3.13. This algorithm also allows dynamic layout regeneration. If a part of the level is removed, further generations will regenerate the removed part and adapt to the new environment.

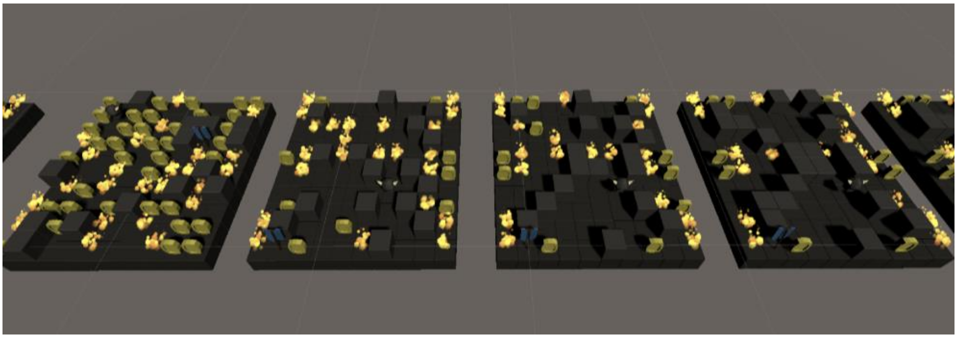


Fig. 3.13. CoCoSo pattern progression over generations

By integrating the CoCoSo method into the genetic procedural scene layout generator, research has extended the algorithm’s capabilities. The CoCoSo method yields satisfying results with fewer generations and considers various aspects of the problem more comprehensively than the WASPAS method. The incorporation of neutrosophic sets enables the effective handling of uncertain information by modelling truth, indeterminacy, and false membership functions. This implementation of neutrosophic sets within the CoCoSo approach enhances the diversity of generated levels. Within the framework of the genetic algorithm, the proposed neutrosophic CoCoSo method outperforms the original CoCoSo approach. Specifically, research improves the method by setting similar divisor values to constant values. The conversions between crisp and neutrosophic numbers have been updated to align with iterative evolution. Additionally, the normalisation procedures have been adjusted to consider global minimum and maximum values, adapting them to the set of generations where values depend on previous generations. The generated patterns meet the replayability principle, satisfying aesthetic and functional criteria. This indicates that the proposed approach creates game levels that are visually appealing, engaging, and enjoyable for players to explore repeatedly.

3.3. Regionally morphing objects for the genetic game scene generation algorithm

The algorithm generates visual results that introduce diversity to the final layout of the game scene by incorporating natural randomness into the visual aspect of the level. Some iterations produce regional grassy fields, while others generate aligned trees that serve as impassable walls without affecting the physical-level boundaries and functionality. Several examples of the final results include a grassy region in the corner with trees (Fig. 3.14), an example with increased resolution (Fig. 3.15), clustered trees (Fig. 3.16), and rocks instead of trees (Fig. 3.17).

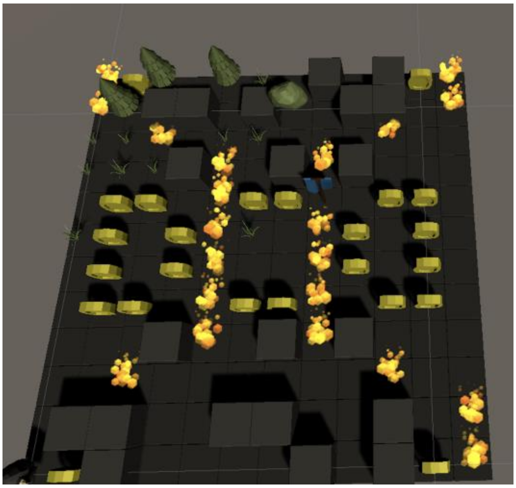


Fig. 3.14. Regional morph results

Increasing the algorithm’s grid size significantly increases the time required to generate a game scene. The additional visual object types need to be logically aligned with their purpose for the algorithm to place them effectively. The final algorithm takes around 6.3 seconds over 800 epochs to generate a level on a 2.4 GHz 8-Core Intel Core i9 CPU. It achieves a fitness value of approximately 0.7 after 100 epochs and reaches around 0.85 after 800 epochs. It should be noted that the algorithm cannot approach a fitness value close to 1 due to the conflicting nature of its criteria and the aim of generating varied results.

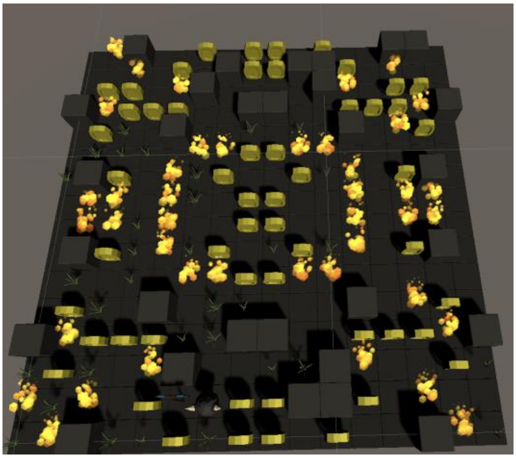


Fig. 3.15. Example with increased resolution

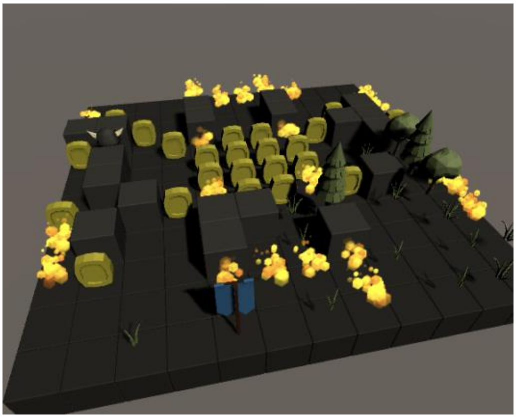


Fig. 3.16. Clusters of objects at the side of the level

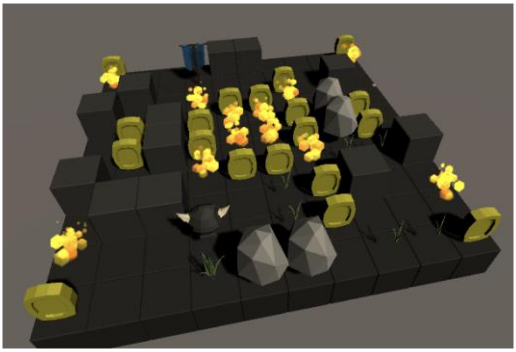


Fig. 3.17. Wall morphs into rocks instead of trees

A creative procedural generation without pre-made content snippets is an emerging field within Computational Creativity. The research introduced and tested an extension for the genetic WASPAS-SVNS game scene generation algorithm by incorporating regionally morphing post-processing for the game objects. This extension enhances the visual diversity of the game-level layout and increases its creative value while ensuring that game design rules remain intact. Each algorithm iteration can produce fresh patterns for the game level, seamlessly integrated into the overall composition. The modular nature of the original procedural generation engine allows for easy integration of new extensions, further enhancing the overall creative value of the generated levels. Each algorithm iteration can produce fresh patterns for the game level, seamlessly integrated into the overall composition. The modular nature of the original procedural generation engine allows for easy integration of new extensions, further enhancing the overall creative value of the generated levels.

3.4. Gestalt principles governed fitness function for genetic Pythagorean neutrosophic game scene generation

The Gestalt principle-based fitness function was integrated into the genetic WASPAS game scene generator. The experiments involved adding a single Gestalt principle to the base fitness function and then combining all Gestalt principles. For reference and comparison, research also provided the generated level without any Gestalt principles (Fig. 3.18).

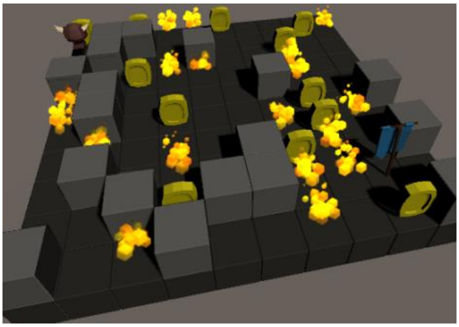


Fig. 3.18. Base result without integration of the Gestalt principles

The common region criteria result in-game objects being enclosed by walls, which creates narrow rooms or corridors in the generated levels. Increasing the weight of this criterion can significantly reduce the amount of empty space in the layout (Fig. 3.19).

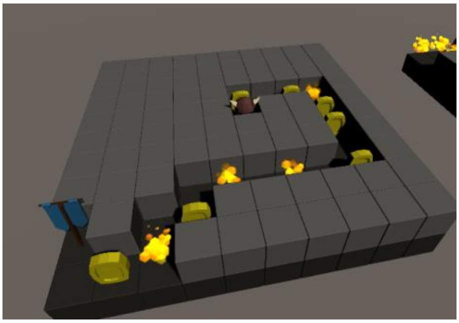


Fig. 3.19. Common region result

The focal-point criteria aim to attract the player’s attention to specific points that stand out in the environment. In the generated levels, it can be observed that the exit (blue flag) and a few flame assets stand out prominently in the surrounding environment (Fig. 3.20).

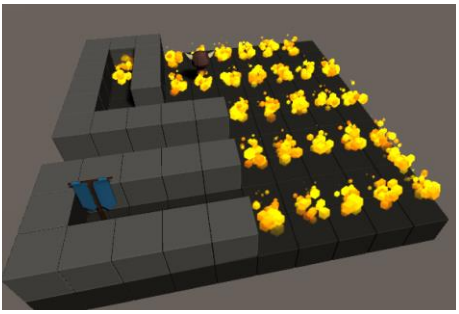


Fig. 3.20. Focal point result

The continuity criterion aims to create continuous lines of objects in the generated levels. As a result, patterns can be observed, where the building blocks are mostly arranged in combinations of lines across the level (Fig. 3.21).

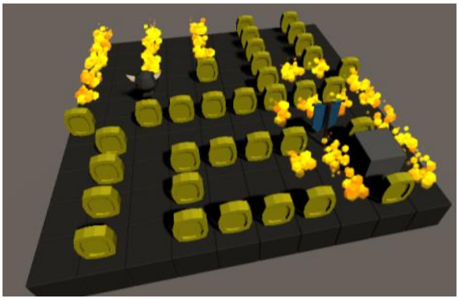


Fig. 3.21. Continuity result

The proximity criteria create isolated islands of objects in the generated levels. In the visual representation, two islands consisting of walls and coins can be observed (Fig. 3.22).

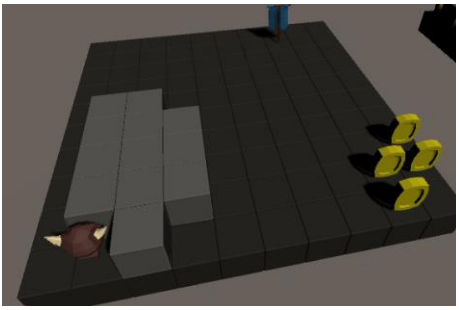


Fig. 3.22. Proximity result

The similarity criteria group similar objects together in the generated levels. In the example, the room is split into four regions, each containing different types of objects (Fig. 3.23).

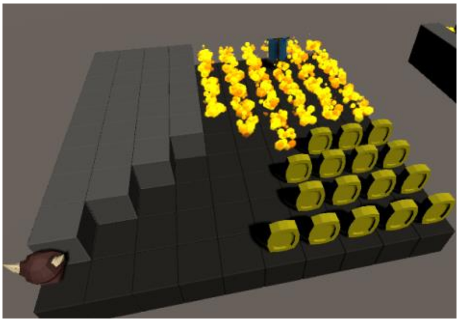


Fig. 3.23. Similarity result

Upon examining how each criterion is integrated into the overall level generator, all the criteria were combined to form the fitness function. The results show the patterns of individual Gestalt rules and the variations between three iterations of the same algorithm. The first example displays an area with balanced empty space, fewer walls, and no hazardous objects (Fig. 3.24). In the second example, there are more walls arranged in an “8” pattern (Fig. 3.25). The third example forms an enclosed space with a cluster of coins in the middle and a hidden exit (Fig. 3.26).

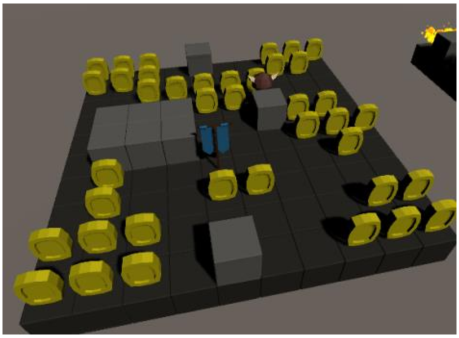


Fig. 3.24. Result of the combined Gestalt principles 1

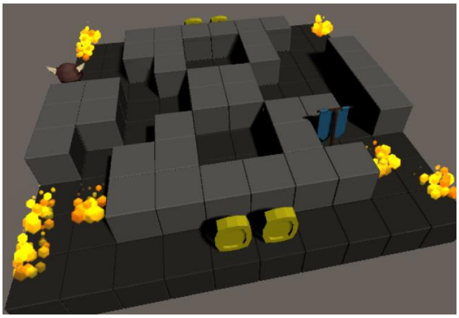


Fig. 3.25. Result of the combined Gestalt principles 2

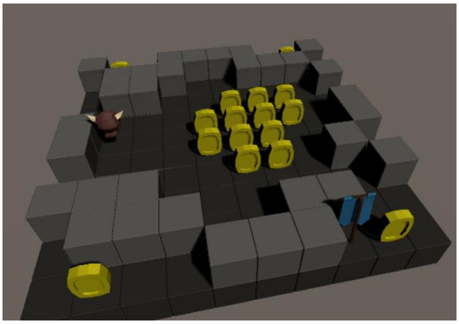


Fig. 3.26. Result of the combined Gestalt principles 3

It was observed that Gestalt principles can create discernible patterns in the standard 10×10 matrix. A higher resolution matrix of 20×20 was tested to investigate further. While pattern formation was still evident, the high-scale presence was not as prominent as in the lower resolution, where the application was based on a small 3×3 matrix focal function (Fig. 3.26).

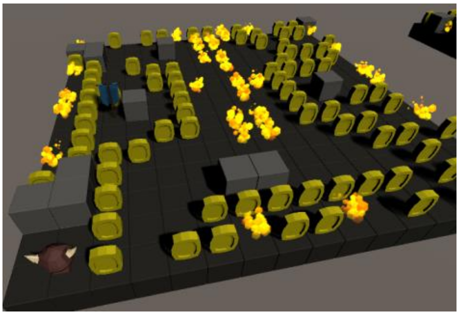


Fig. 3.26. Combined Gestalt principles result in a 20×20 matrix

At the start of the fitness evaluation, there is a rapid increase, which then gradually slows down as the algorithm fine-tunes itself. The initial jump in fitness occurs within approximately 0.5 seconds. The entire evolution process takes around 10 seconds to reach a stagnant state, and the fitness value with the proposed criteria reaches around 0.7 (Fig. 3.27).

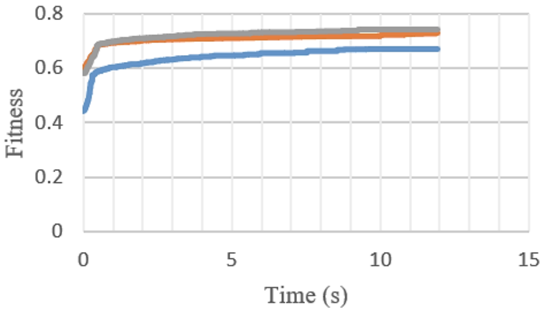


Fig. 3.27. Example evolution curves with separate iterations of the algorithm, each curve represents a separate execution of the algorithm

The fitness cannot reach 1 due to conflicting criteria. As a result of the increased number of conflicting criteria, the final fitness is lower compared to previous experiments (0.85 > 0.7). However, the use of Gestalt principles results in a higher visual value due to the enhanced presence of visual patterns. The model is generative, so its size is only 40 kilobytes. The algorithm has a huge potential for parallelisation because each criterion and each chromosome can be calculated independently. The engine was tested on an Intel i9-9980HK 2.4 GHz CPU.

The proposed fitness function based on Gestalt principles was successfully integrated into the genetic Pythagorean neutrosophic WASPAS game scene generator. The results show that coherent and visually appealing patterns were incorporated into the final generated game scenes. Compared to the originally generated levels, the aesthetic evaluation significantly improves in the final scene generations. Modelling abstract criteria into a mathematical model can be challenging while preserving all the information from the rules. In this case, the mathematical models of Gestalt principles were effectively applied using automated cell neighbourhood algorithms. Both functional and aesthetic criteria were combined into the final algorithm to maintain the original game design rules while enhancing the game scene generation process. The generated result always adheres to the functional criteria due to multiple binary evaluations of these criteria. It means that it is very unlikely to generate impassable levels. Further experimentation with other aesthetic modelling methods may improve the results, but the number of rules and the size of the generated matrix can influence the presence and impact of each integrated rule. If there are too many rules in a small scene matrix, it can lead to chaotic and less desirable results. A higher mutation ratio reduces the game scene fine-tuning ratio but can have a faster initial fitness breakthrough. Results were analysed with 500 engine executions.

An expert survey was conducted to obtain opinions on user interface and user experience from game design experts from the scientific community and the video game industry. The experts were asked to evaluate two generated examples (Figs. 3.25 and 3.26). The first question focused on the actual visibility of Gestalt principles, while the second addressed the aesthetic quality compared to other examples in the literature. S1–S5 correspond to the visibility of Gestalt principles (Similarity, Continuity, Common Region, Proximity, Focal Point), and Q1–Q3 correspond to the subjective creative value compared to the closest examples in the literature (Safak et al., 2016; Zafar et al., 2020; Thakkar et al., 2019). Some experts noted that judging abstract value is not a suitable approach due to different generative goals and other aesthetic qualities like colour and line shapes that cannot be quantified. One expert refused to compare quality for this reason (Table 3.1).

Table 3.1. Quality survey

Sample	S1	S2	S3	S4	S5	Q1	Q2	Q3
3.25	+			+	+	+	+	
	+		+			+	+	
	+			+	+	Refused	Refused	Refused
	+	+	+		+	+	+	+

End of Table 3.1

Sample	S1	S2	S3	S4	S5	Q1	Q2	Q3
	+	+	+	+	+	+	+	+
3.26	+	+		+	+	+	+	
				+	+		+	
		+		+	+	Refused	Refused	Refused
	+	+	+		+	+	+	+
	+	+	+	+	+	+	+	+

The results showed that Similarity received positive feedback in 80% of the responses, Continuity in 60%, Common Region in 50%, Proximity in 70%, and Focal Point in 90%. Aesthetic quality was rated higher in 87.5%, 100%, and 50% of the responses compared to other examples in the literature.

3.5. Conclusions of the Third Chapter

1. The research focused on developing a procedural scene generation framework using the Unity game engine and a custom C# script. The neutrosophic evaluation approach, when applied to the generator, showed improved efficiency in quickly generating rising scores during the initial generations, especially compared to individual criterion fitness scores. However, the process exhibited longer times for creating balanced and symmetrical layouts that adhered to game rules. The approach resulted in fitness scores ranging from 0.75 to 0.85. The algorithm's generation process was diverse due to numerous local maxima driven by random seed variation and mutations. The research demonstrated the potential of combining genetic and neutrosophic algorithms to generate unique and aesthetically pleasing game scenes while respecting game design rules.
2. Building on the original WASPAS generator, the research introduced the CoCoSo method for generating creative game scene patterns. The CoCoSo method exhibited quicker convergence and enhanced calculation efficiency than WASPAS. Implementing neutrosophic sets added diversity to generated levels and facilitated handling uncertain information. The CoCoSo method effectively balanced aesthetic and functional criteria, creating visually appealing and engaging game scenes. The research demonstrated that this method could expedite the creative process and produce satisfying results in fewer generations.

3. This research extended the genetic WASPAS-SVNS game scene generator by incorporating regionally morphing post-processing for game objects. The algorithm introduced diversity to final layouts through natural randomness. The resulting game scenes exhibited varied elements, such as grassy fields, trees, and rocks, contributing to enhanced visual diversity without compromising game design rules. The extension demonstrated the feasibility of integrating regionally morphing objects into procedural generation engines, facilitating creative and dynamic game-level design.
4. The research integrated Gestalt principles into the genetic Pythagorean neutrosophic WASPAS game scene generator, aiming to enhance the aesthetic and visual aspects of generated levels. By applying different Gestalt principles individually and then combining them, the study showcased the impact of these principles on layout patterns. The incorporation of Gestalt principles led to improved visual aesthetics in the generated levels. However, due to the conflicting nature of the criteria, achieving a fitness score close to 1 was not possible. Nevertheless, the research highlighted the potential of using Gestalt principles to create visually appealing game scenes.
5. The expert survey showed that Similarity received positive feedback in 80% of the responses, Continuity in 60%, Common Region in 50%, Proximity in 70%, and Focal Point in 90%. Aesthetic quality was rated higher in 87.5%, 100%, and 50% of the responses than other examples in the literature.

General Conclusions

1. The CoCoSo method, an enhancement to the WASPAS generator, enabled quicker convergence (~80% faster initial bump in convergence) and efficient calculation. Incorporating neutrosophic sets improved diversity (~35% more unique artefact generation) and balanced aesthetic and functional criteria. The approach accelerated creativity within a reduced number of generations.
2. The addition of regionally morphing objects diversified game-level layouts by introducing natural randomness. Grass, trees, and rocks varied final scenes while preserving design rules (~25% of the scenes used a number of possible variations that were 100% times higher). This extension showcased the feasibility of dynamic and creative game-level design.
3. Gestalt principles were integrated into the genetic Pythagorean neutrosophic WASPAS game scene generator. While enhancing visual aesthetics, challenges persisted due to conflicting criteria. Despite this, using Gestalt principles demonstrated potential for creating visually appealing game scenes (250% more automatic aesthetic analysis criterion). The expert survey showed that Similarity received positive feedback in 80% of the responses, Continuity in 60%, Common Region in 50%, Proximity in 70%, and Focal Point in 90%. Aesthetic quality was rated higher in 87.5%, 100%, and 50% of the responses compared to other examples in the literature.

References

Ali, J., Bashir, Z., & Rashid, T. (2020). WASPAS-based decision making methodology with unknown weight information under uncer-tain evaluations. *Expert Systems with Applications*, 168, 114143. https://doi.org/10.1016/j.eswa.2020.114143

Alvarez, A., Dahlskog, S., Font, J., Holmberg, J., & Johansson, S. (2018). Assessing aesthetic criteria in the evolutionary dungeon designer. *Proceedings of the 13th International Conference on the Foundations of Digital Games*, 1–4. https://doi.org/10.1145/3235765.3235810

Atkinson, A., & Parsayi, F. (2021). Video games and aesthetic contemplation. *Games and culture*, *16*(5), 519–537. https://doi.org/10.1177/1555412020914726

Baldwin, A., Dahlskog, S., Font, J. M., & Holmberg, J. (2017). Mixed-initiative procedural generation of dungeons using game design patterns. In *2017 IEEE conference on computational intelligence and games (CIG)*, 25–32. https://doi.org/10.1109/CIG.2017.8080411

Bausys, R., & Kazakeviciute-Januskeviciene, G. (2021). Qualitative rating of lossy compression for aerial imagery by neutrosophic WASPAS method. *Symmetry* , *13*(2), 1–26. https://doi.org/10.3390/sym13020273

Bausys, R., Juodagalviene, B., Žiuriene, R., Pankrasovaite, I., Kamarauskas, J., Usovaite, A., & Gaižauskas, D. (2020a). The residence plot selection model for family house in Vilnius by neutrosophic WASPAS method. *International journal of strategic property management 24*)(3), 182–196. https://doi.org/10.3846/ijspm.2020.12107

Bausys, R., Kazakeviciute-Januskeviciene, G., Cavallaro, F., Usovaite, A. (2020b). Algorithm Selection for Edge Detection in Satellite Images by Neutrosophic WASPAS Method. *Sustainability, 12,* 548. https://doi.org/10.3390/su12020548

Bausys, R., Zavadskas, E. K., Semenas, R. (2022). Path selection for the inspection robot by m-generalized q-neutrosophic PROMETHEE approach. *Energies, 15*(1), 223. https://doi.org/10.3390/en15010223

Beukman, M., Cleghorn, C. W., & James, S. (2022). Procedural content generation using neuroevolution and novelty search for diverse video game levels. In *Proceedings of the Genetic and Evolutionary Computation Conference,* 1028–1037. https://doi.org/10.1145/3512290.3528701

Braben, D., & Bell, I. (1984). Elite [Video game].

Broumi, S., Bakali, A., Talea, M., Smarandache, F., Uluçay, V., Sahin, M., Dey, A., Dhar, M., Tan, R,. Bahnasse, A., & Pramanik, S. (2018). Neutrosophic Sets: An Overview. *New Trends in Neutrosophic Theory and Applications*, *2,* 388–418.

Cao, F. (2004). Application of the Gestalt principles to the detection of good continuations and corners in image level lines. *Computing and Visualization in Science*, *7*(1), 3–13. https://doi.org/10.1007/s00791-004-0123-6

Carballal, A., Fernandez-Lozano, C., Rodriguez-Fernandez, N., Castro, L., & Santos, A. (2019). Avoiding the Inherent Limitations in Datasets Used for Measuring Aesthetics When Using a Machine Learning Approach. *Complex.*, 4659809:1-4659809:12. https://doi.org/10.1155/2019/4659809

Chang, D., & Nesbitt, K. V. (2006). Identifying commonly-used gestalt principles as a design framework for multi-sensory displays. In *2006 IEEE International Conference on Systems, Man and Cybernetics*, *3,* 2452–2457.
https://doi.org/10.1109/ICSMC.2006.385231

Choi, S. S., & Moon, B. R. (2003). Normalization in genetic algorithms. *Genetic and Evolutionary Computation Conference*, 862–873. https://doi.org/10.1007/3-540-45105-6_99

Colton, S., & Wiggins, G. A. (2012). Computational creativity: The final frontier?. In *20th European Conference on Artificial Intelligence*, *12,* 21–26.

Cook, M., Colton, S., & Gow, J. (2013). Nobody's A Critic: On The Evaluation Of Creative Code Generators - A Case Study In Video Game Design. *ICCC*, 123–130.

Cook, M., Colton, S., Pease, A., & Llano, M. T. (2019). Framing In Computational Creativity - A Survey And Taxonomy. *ICCC*, 156–163.

Dick, S. (2019). Artificial Intelligence. *Harvard Data Science Review*, *1*(1), 1–9. https://doi.org/10.1162/99608f92.92fe150c

Earle, S., Snider, J., Fontaine, M. C., Nikolaidis, S., & Togelius, J. (2022). Illuminating diverse neural cellular automata for level generation. In *Proceedings of the Genetic and Evolutionary Computation Conference*, 68–76. https://doi.org/10.1145/3512290.3528754

Filip, F. G. (2021). Automation and computers and their contribution to human well-being and resilience. *Studies in Informatics and Control, 30*(4), 5–18. https://doi.org/10.24846/v30i4y202101

Filip, F. G. (2022). Collaborative Decision-Making: Concepts and Supporting Information and Communication Technology Tools and Systems. *International Journal of Computers, Communications and Control, 17*(2), 1–9. https://doi.org/10.15837/ijccc.2022.2.4732

Franceschelli, G., & Musolesi, M. (2021). *Creativity and Machine Learning: A Survey*. arXiv:2104.02726.

Freiknecht, J., & Effelsberg, W. (2017). A survey on the procedural generation of virtual worlds. *Multimodal Technologies and Interaction*, *1*(4), 27. https://doi.org/10.3390/mti1040027

Giacomello, E., Lanzi, P. L., & Loiacono, D. (2018). Doom level generation using generative adversarial networks. In *IEEE Games, Entertainment, Media Conference (GEM),* 316–323. https://doi.org/10.1109/GEM.2018.8516539

Guzdial, M., & Riedl, M. (2016). Game level generation from gameplay videos. In *Twelfth artificial intelligence and interactive digital entertainment conference, 12*(1), 44–50.

Han, J., Forbes, H., & Schaefer, D. (2021). An exploration of how creativity, functionality, and aesthetics are related in design. *Research in Engineering Design*, *32*(3), 289–307. https://doi.org/10.1007/s00163-021-00366-9

Haque, T. S., Chakraborty, A., Mondal, S. P., & Alam, S. (2020). Approach to solve multicriteria group decision-making problems using exponential operational law in a generalised spherical fuzzy environment. *CAAI Transactions on Intelligence Technology*, *5*(2), 106–114. https://doi.org/10.1049/trit.2019.0078

Herrmann, J. W. (1999). A genetic algorithm for minimax optimization problems. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99,* 2, 1099–1103. https://doi.org/10.1109/CEC.1999.782545

Isaksen, A., Gopstein, D., Togelius, J., & Nealen, A. (2015). Discovering Unique Game Variants. *Computational Creativity and Games Workshop at the International Conference on Computational Creativity*.

Kahraman, C., Öztayşi, B., & Çevik Onar, S. (2016). A comprehensive literature review of 50 years of fuzzy set theory. *International Journal of Computational Intelligence Systems, 9*(1), 3–24. https://doi.org/10.1080/18756891.2016.1180817

Karth, I. (2019). Preliminary poetics of procedural generation in games. *Transactions of the Digital Games Research Association*, *4*(3), 245–285. https://doi.org/10.26503/todigra.v4i3.106

Khalifa, A., de Mesentier Silva, F., & Togelius, J. (2019a). Level design patterns in 2D games. *IEEE Conference on Games (CoG)*, 1–8. https://doi.org/10.1109/CIG.2019.8847953

Khalifa, A., Green, M. C., Barros, G., & Togelius, J. (2019b). Intentional computational level design. *Proceedings of The Genetic and Evolutionary Computation Conference,* 796–803. https://doi.org/10.1145/3321707.3321849

Kieu, P. T., Nguyen, V. T., Nguyen, V. T., & Ho, T. P. (2021). A Spherical Fuzzy Analytic Hierarchy Process (SF-AHP) and Combined Compromise Solution (CoCoSo) Algorithm in Distribution Center Location Selection: A Case Study in Agricultural Supply Chain. *Axioms*, *10*(2), 53. https://doi.org/10.3390/axioms10020053

Lamb, C., Brown, D. G., & Clarke, C. L. A. (2018). Evaluating Computational Creativity: An Interdisciplinary Tutorial. *ACM Computing Surveys., 28,* 1–34. https://doi.org/10.1145/3167476

Lara-Cabrera, R., Cotta, C., & Fernández-Leiva, A. J. (2014). On balance and dynamism in procedural content generation with self-adaptive evolutionary algorithms. *Nat Comput, 13,* 157–168. https://doi.org/10.1007/s11047-014-9418-9

Lescauskiene, I., Bausys, R., & Zavadskas E. K., Juodagalviene B. (2020). VASMA weighting: survey-based criteria weighting methodology that combines ENTROPY and WASPAS-SVNS to reflect the psychometric features of the VAS scales. *Symmetry* (*12*)10, 1–20. https://doi.org/10.3390/sym12101641

Liapis, A., Yannakakis, G. N., & Togelius, J. (2012). Limitations of Choice-Based Interactive Evolution for Game Level Design. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* (*8*)5. https://doi.org/10.1609/aiide.v8i5.12571

Liu, J., Snodgrass, S., Khalifa, A., Risi, S., Yannakakis, G. N., & Togelius, J. (2021). Deep learning for procedural content generation. *Neural Computing and Applications, 33*(1), 19–37. https://doi.org/10.1007/s00521-020-05383-8

Morkunaite, Z., Bausys, R., & Zavadskas, E. K. (2019). Contractor Selection for Sgraffito Decoration of Cultural Heritage Buildings Using the WASPAS-SVNS Method. *Sustainability,* (*11*)22: 6444. https://doi.org/10.3390/su11226444

Nenad, M. (2018). Designing game worlds. Coherence in the design of open world games through procedural generation techniques. *Proceedings of the 2018 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts,* 353–363. https://doi.org/10.1145/3270316.3270319

Nesbitt, K. V., & Friedrich, C. (2002). Applying gestalt principles to animated visualizations of network data. *In Proceedings Sixth International Conference on Information Visualisation,* 737–743. https://doi.org/10.1109/IV.2002.1028859

Nyholm, O., & Nilsson, P. (2017). A Comparison Between Evolutionary and Rule-based Level Generation [Dissertation]. Malmö högskola/Teknik och samhälle.

Padhye, N., & Deb, K. (2011). Multi-objective optimisation and multi-criteria decision making in SLS using evolutionary approaches. *Rapid Prototyping Journal*, *17*(6), 458–478. https://doi.org/10.1108/13552541111184198

Peng, X., & Garg, H. (2021). Intuitionistic fuzzy soft decision making method based on CoCoSo and CRITIC for CCN cache placement strategy selection. *Artificial Intelligence Review, 55,* 1567–1604. https://doi.org/10.1007/s10462-021-09995-x

Peng, X., & Li, W. (2021). Spherical fuzzy decision making method based on combined compromise solution for IIoT industry evaluation. *Artificial Intelligence Review, 55,* 1857–1886. https://doi.org/10.1007/s10462-021-10055-7

Pereira, L. T., Toledo, C., Ferreira, L. N., & Lelis, L. H. (2016). Learning to speed up evolutionary content generation in physics-based puzzle games. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI),* 901–907. https://doi.org/10.1109/ICTAI.2016.0139

Pereira, L. T., Toledo, C., Ferreira, L. N., & Lelis, L. H. (2016). Learning to Speed Up Evolutionary Content Generation in Physics-Based Puzzle Games. *IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI),* 901–907. https://doi.org/10.1109/ICTAI.2016.0139

Pichot, N., Bonetto, E., Pavani, J. B., Arciszewski, T., Bonnardel, N., & Weisberg, R. W. (2022). The construct validity of creativity: empirical arguments in favor of novelty as the basis for creativity. *Creativity Research Journal, 34*(1), 2–13. https://doi.org/10.1080/10400419.2021.1997176

Radha, R., Mary, A. S. A., Prema, R., & Broumi, S. (2021). Neutrosophic Pythagorean Sets with Dependent Neutrosophic Pythagorean Components and its Improved Correlation Coefficients. *Neutrosophic Sets and Systems*, *46*, 77–86.

Risi, S., & Togelius, J. (2019). Increasing generality in machine learning through procedural content generation. *Nature Machine Intelligence*, 1–9.

Safak, A. B., Bostanci, E., & Soylucicek, A. E. (2016). Automated maze generation for Ms. Pac-Man using genetic algorithms. *International Journal of Machine Learning and Computing*, *6*(4), 226–240. https://doi.org/10.18178/ijmlc.2016.6.4.602

Safak, A. B., Bostanci, E., & Soylucicek, A. E. (2016). Automated maze generation for Ms. Pac-Man using genetic algorithms. *International Journal of Machine Learning and Computing*, *6*(4): 226–4. https://doi.org/10.18178/ijmlc.2016.6.4.602

Schrum, J., Gutierrez, J., Volz, V., Liu, J., Lucas, S., & Risi, S. (2020). Interactive evolution and exploration within latent level-design space of generative adversarial networks. *InProceedings of the 2020 Genetic and Evolutionary Computation Conference*, 148–156. https://doi.org/10.1145/3377930.3389821

Semenas, R., Bausys, R., & Zavadskas, E. K. (2021). A novel environment exploration strategy by m-generalised q-neutrosophic WASPAS. *Studies in informatics and control, 30*(3), 19–28. https://doi.org/10.24846/v30i3y202102

Serb, A., & Prodromakis, T. (2019). A system of different layers of abstraction for artificial intelligence. *arXiv:1907.10508.*

Short, T., & Adams, T. (2017). *Procedural generation in game design*. CRC Press. https://doi.org/10.1201/9781315156378

Smarandache, F. A. (1999). Unifying Field in Logics. Neutrosophy: Neutrosophic Probability, Set and Logic. *Book Chapter*. American Research Press.

Soleymani, S., Dabouei, A., Kazemi, H., Dawson, J., & Nasrabadi, N. M. (2018). Multilevel feature abstraction from convolutional neural networks for multimodal biometric identification. In *2018 24th International Conference on Pattern Recognition (ICPR),* 3469–3476. https://doi.org/10.1109/ICPR.2018.8545061

Sorenson, N., & Pasquier, P. (2010). Towards a Generic Framework for Automated Video Game Level Creation. *EvoApplications,* 6024, 131–140. https://doi.org/10.1007/978-3-642-12239-2_14

Srinivasan, R., & Uchino, K. (2021). Biases in generative art: A causal look from the lens of art history. *In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 41–51. https://doi.org/10.1145/3442188.3445869

Stanujkic, D., Karabaševic, D., Popovic, G., Pamucar, D., Stevic, Z., Zavadskas, E.K., & Smarandache, F. A. (2021). Single-Valued Neutrosophic Extension of the EDAS Method. *Axioms*, *10,* 245. https://doi.org/10.3390/axioms10040245

Summerville, A., Snodgrass, S., Guzdiaz, M., Holmgård, C., Hoover, A.K., Isaksen, A., Nealen, A., & Togelius, J. (2018). Procedural content generation via machine learning (PCGML). *IEEE Transactions on Games*, *10*(3), 257–270.
https://doi.org/10.1109/TG.2018.2846639

Svadlenka, L., Simic, V., Dobrodolac, M., Lazarevic, D., & Todorovic, G. (2020). Picture Fuzzy Decision-Making Approach for Sustainable Last-Mile Delivery. *IEEE Access, 8,* 209393-209414. https://doi.org/10.1109/ACCESS.2020.3039010

Sweetser, P., & Wyeth, P. (2005). GameFlow: a model for evaluating player enjoyment in games. *Computers in Entertainment (CIE)*, *3*(3), 3.
https://doi.org/10.1145/1077246.1077253

Thakkar, S., Cao, C., Wang, L., Choi, T. J., & Togelius, J. (2019). Autoencoder and evolutionary algorithm for level generation in lode runner. In *2019 IEEE Conference on Games (CoG),* 1–4. https://doi.org/10.1109/CIG.2019.8848076

Thakkar, S., Cao, C., Wang, L., Choi, T. J., & Togelius, J. (2019). Autoencoder and evolutionary algorithm for level generation in lode runner. *IEEE Conference on Games (CoG),* 1–4. https://doi.org/10.1109/CIG.2019.8848076

Thakkar, S., Cao, C., Wang, L., Choi, T. J., & Togelius, J. (2019). Autoencoder and Evolutionary Algorithm for Level Generation in Lode Runner. *IEEE Conference on Games (CoG)*, 1–4. https://doi.org/10.1109/CIG.2019.8848076

Todorovic, D. (2008). Gestalt principles. *Scholarpedia, 3*(12), 5345.
https://doi.org/10.4249/scholarpedia.5345

Togelius, J., Champandard, A.J., Lanzi, P. L., Mateas, M., Paiva, A., Preuss, M., &Stanley, K. O. (2013). Procedural content generation: goals, challenges and actionable steps. *Dagstuhl Follow Ups*, *6,* 61–75.

Togelius, J., Kastbjerg, E., Schedl, D., & Yannakakis, G. N. (2011). What is procedural content generation? Mario on the borderline. In *Proceedings of the 2nd international workshop on procedural content generation in games,* 1–6. https://doi.org/10.1145/2000919.2000922

Togelius, J., & Schmidhuber, J. (2008). An experiment in automatic game design. *IEEE Symposium On Computational Intelligence and Games,* 111–118. https://doi.org/10.1109/CIG.2008.5035629

Toivonen, H., & Gross, O. (2015). Data mining and machine learning in computational creativity. *WIREs Data Mining Knowl Discov, 5,* 265–275. https://doi.org/10.1002/widm.1170

Toivonen, H., & Gross, O. (2015). Data mining and machine learning in computational creativity. *WIREs Data Mining Knowl Discov, 5,* 265–275. https://doi.org/10.1002/widm.1170

Torrado, R. R., Khalifa, A., Green, M. C., Justesen, N., Risi, S., & Togelius, J. (2020). Bootstrapping conditional gans for video game level generation. *In 2020 IEEE Conference on Games (CoG),* 41-48. https://doi.org/10.1109/CoG47356.2020.9231576

Turskis, Z., Bausys, R., Smarandache, F., Kazakeviciute-Januskeviciene, G., & Zavadskas, E. K. (2022). M-generalised q-neutrosophic extension of CoCoSo method. *International Journal of Computers, Communications and Control, 17*(1), 1–12. https://doi.org/10.15837/ijccc.2022.1.4646

Ventura, D. (2016). Mere generation: essential barometer or dated concept? *Proceedings of the Seventh International Conference on Computational Creativity ICCC,* 17–24.

Volz, V., Justesen, N., Snodgrass, S., Asadi, S., Purmonen, S., Holmgård, C., Togelius, J., & Risi, S. (2020). Capturing local and global patterns in procedural content generation via machine learning. *IEEE Conference on Games (CoG),* 399–406.

Volz, V., Justesen, N., Snodgrass, S., Asadi, S., Purmonen, S., Holmgård, C., Togelius, J., & Risi, S. (2020). Capturing Local and Global Patterns in Procedural Content Generation via Machine Learning. *IEEE Conference on Games (CoG),* 399–406. https://doi.org/10.1109/CoG47356.2020.9231944

Walia, C. (2019). A dynamic definition of creativity. *Creativity Research Journal, 31*(3), 237–247. https://doi.org/10.1080/10400419.2019.1641787

Wang, H., Smarandache, F., Zhang, Y., & Sunderraman, R. (2010). Single valued neutrosophic sets. *Infinite study, 12.*

Wang, Y., Xu, Z., & Filip, F. G. (2022). Multi-Objective Model to Improve Network Reliability Level under Limited Budget by Considering Selection of Facilities and Total Service Distance in Rescue Operations. *International Journal of Computers, Communications and Control, 17*(1), 1–18. https://doi.org/10.15837/ijccc.2022.1.4573

Wertheimer, M. (2012). *Investigations on Gestalt principles*. 127–183. MIT Press.

Whitley, D. (1994). Genetic algorithm tutorial. *Statistics and Computing 4,* 65–85. https://doi.org/10.1007/BF00175354

Wiggins, G. A. (2006). Searching for computational creativity. *New Generation Computing, 24,* 209–222. https://doi.org/10.1007/BF03037332

Wu, Z., Liao, H., Lu, K., & Zavadskas, E. K. (2021). Soft computing techniques and their applications in intelligent industrial control systems: A survey. *International journal of computers communications and control, 16*(1), 1–28. https://doi.org/10.15837/ijccc.2021.1.4142

Yazdani, M., Mohammed, A., Bai, C., & Labib, A. (2021). A novel hesitant-fuzzy-based group decision approach for outsourcing risk. *Expert Systems with Applications, 184*(115517). https://doi.org/10.1016/j.eswa.2021.115517

Yousefi, S., Valipour, M., & Gul, M. (2021). Systems failure analysis using Z-number theory-based combined compromise solution and full consistency method. *Applied Soft Computing, 113*(107902), . https://doi.org/10.1016/j.asoc.2021.107902

Yu, Q., Yang, Y., Liu, F., Song, Y. Z., Xiang, T., & Hospedales, T. M. (2017). Sketch-a-net: A deep neural network that beats humans. *International journal of computer vision, 122,* 411–425. https://doi.org/10.1007/s11263-016-0932-3

Zadeh, L. A. (1965). Fuzzy sets. *Information and control, 8*(3), 338–353. https://doi.org/10.1016/S0019-9958(65)90241-X

Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions on systems, Man, and Cybernetics, 1,* 28–44. https://doi.org/10.1109/TSMC.1973.5408575

Zafar, A., Mujtaba, H., & Beg, M. O. (2020). Search-based procedural content generation for GVG-LG. *Applied Soft Computing*, 86:105909. https://doi.org/10.1016/j.asoc.2019.105909

Zafar, A., Mujtaba, H., & Beg, M. O. (2020). Search-based procedural content generation for GVG-LG. *Applied Soft Computing, 86*. https://doi.org/10.1016/j.asoc.2019.105909

Zavadskas, E. K., Turskis, Z., & Kildienė, S. (2014). State of art surveys of overviews on MCDM/MADM methods. *Technological and economic development of economy, 20*(1), 165–179. https://doi.org/10.3846/20294913.2014.892037

Zhao, H., Zhang, R., Zhang, A., & Zhu, X. (2021). Multi-attribute Group Decision Making Method with Unknown Attribute Weights Based on the Q-rung Orthopair Uncertain Linguistic Power Muirhead Mean Operators. *International Journal of Computers Communications and Control, 16*(3), 1–20. https://doi.org/10.15837/ijccc.2021.3.4214

List of Scientific Publications by the Author on the Topic of the Dissertation

Papers in the reviewed scientific journals

Petrovas, A., & Bausys, R. (2022). Procedural Video Game Scene Generation by Genetic and Neutrosophic WASPAS Algorithms. *Applied Sciences*, *12*(2): 772. https://doi.org/10.3390/app12020772

Petrovas, A., Baušys, R., Zavadskas, E. K., & Smarandache, F. (2022). Generation of creative game scene patterns by the neutrosophic genetic CoCoSo method. *Studies in informatics and control*, *31*(4), 5–11. https://doi.org/10.24846/v31i4y202201

Petrovas, A., Bausys, R., & Zavadskas, E. (2023). Gestalt Principles Governed Fitness Function for Genetic Pythagorean Neutrosophic WASPAS Game Scene Generation. *International journal of computers communications & control*, *18*(4). https://doi.org/10.15837/ijccc.2023.4.5475

Papers in other editions

Petrovas, A., & Bausys, R. (2022), July. Regionally morphing objects for the genetic WASPAS-SVNS game scene generation algorithm. In *2022 International Conference on*

Electrical, Computer and Energy Technologies (ICECET), 1–4. https://doi.org/10.1109/ICECET55527.2022.9872949

Petrovas, A., & Bausys, R. (2019). Automated Digital Terrain Elevation Modification By Procedural Generation Approach. *Open Conference of Electrical, Electronic and Information Sciences (eStream)*, 1–5. https://doi.org/10.1109/eStream.2019.8732171

Summary in Lithuanian

Įvadas

Problemos formulavimas

Šiuo metu kūrybiškumo modeliavimo populiarumas sparčiai auga, o jo taikymas plinta įvairiose srityse. Nepaisant šio populiarumo, nėra universalaus kūrybiškumo apibrėžimo ir apibrėžimas varijuoja tarp skirtingų mokslo sričių. Abstrakti kūrybiškumo prigimtis lemia jo vertinimo iššūkius. Kūrybiškumas apima įvairius intelektinius gebėjimus, todėl jo simuliacija yra sudėtinga techninė užduotis (Colton & Wiggins, 2012; Wiggins, 2006). Tai rodo, kad žmonės kūrybai naudoja gebėjimus, kurie nėra lengvai suprantami ir todėl juos sunku atkartoti dirbtinio kūrybiškumo modelyje. Kūrybiškumo modeliavimo užduotimi siekiama suprasti ir išskirti kūrybines savybes iš esamų žinių, kūrinių ir kūrinių aplinkos. Vertinga informacija apie kūrybą yra netiesiogiai užkoduota mus supančioje aplinkoje, todėl svarbu suvokti, kaip apibrėžiamas kūrybiškumas, siekiant efektyviai generuoti dirbtines kūrybiškas sistemas.

Skaitmeninis kūrybiškumas yra dirbtinio intelekto sritis, tyrinėjanti skaitmeninių sistemų, kurios geba parodyti kūrybiškumo savybes, kūrimą. Vienas pagrindinių kūrybiškumo aspektų yra skirtingų prigimčių tikslų derinimas. Taikant šiuolaikinius skaitmeninio kūrybiškumo metodus dažniausiai naudojamas generatyvinis menas. Generatyvinis menas apibrėžia algoritmiškai sukurtus artefaktus, kurie rodo meninius motyvus (Boden & Edmonds, 2009). Šie metodai dažniausiai taikomi kuriant įvairias medijos formas. Generuoto turinio pavyzdžiai yra garsas, muzika, vaizdai ar tekstas. Generatyvinis menas yra naudinga sritis kuriant skaitmeninius žaidimų pasaulius ir projektuojant žaidimų lygius, kur yra dažnai pagalbai pasitelkiama procedūrinė generacija.

Paieškos metodus taikanti procedūrinė generacija vaizdo žaidimams įtraukia sugeneruotų žaidimo objektų ar jų kompozicijų vertinimą pasitelkdama tikslo funkciją. Toks automatizuotas vertinimo būdas padeda nuspręsti, kaip gerai sugeneruoti artefaktai yra reitinguojami pagal tikslo funkciją. Todėl yra labai svarbu sukurti gerą tikslo funkciją, kuri nulems, kaip gerai procedūrinis generatorius atliks savo užduotį. Pagrindinis paieškos algoritmų privalumas yra galimybė reguliariai surasti naujus sprendimus pagal tikslo funkcijos reikalavimus. Taikomi du pagrindiniai tinkamumo kriterijų tipai: estetiniai ir funkciniai. Taikant estetinius kriterijus vertinami rezultatai vizualiniu aspektu, užtikrinant, kad sukurtas lygis atrodytų patraukliai. Taikant funkcinius kriterijus vertinama rezultato atitiktis funkcinėms taisyklėms, kurios skirtos teisingam vaizdo žaidimo veikimui. Funkcinių kriterijų pavyzdžiai gali būti svarbių žaidimo objektų egzistavimas ar galimybė juos naudoti pagal paskirtį sugeneruotoje žaidimo scenoje. Vienas pagrindinių iššūkių šiame procese yra geras funkcinių ir estetinių kriterijų sujungimas. Kartais estetiniai ir funkciniai kriterijai gali būti prieštaringi vienas kitam, todėl gali būti sunku pasiekti tinkamą jų pusiausvyrą (Han et al., 2021).

Kitas iššūkis modeliuojant kūrybinius elementus kyla iš jų abstrakčios prigimties, dėl to yra sunku tiksliai apibrėžti juos matematiniais algoritmais. Šiuo tyrimu siekiama spręsti šią problemą pateikiant tikslias algoritmo elementų apibrėžtis. Be to, norint užtikrinti, kad sukurti artefaktai turėtų kūrybinę vertę, reikia sukurti metodus, kurie galėtų įtraukti aukšto lygio estetinius elementus ir sukurti pastebimus skirtumus tarp kelių to paties algoritmo iteracijų. Siekiant efektyviai sujungti tikslo funkcijos kriterijus, vienas iš sprendimų yra daugiakriterių sprendimų priėmimo metodai (MCDM). MCDM padeda ieškoti sprendinio renkantis alternatyvas iš riboto galimų sprendimų rinkinio (Zavadskas et al., 2014). Taikant šiuos metodus genetinio algoritmo tikslo funkcijai, galima optimizuoti genetinio algoritmo operatorius. Kai kurios neapibrėžtus skaičius naudojančios MCDM metodų variacijos gali padidinti rezultato neapibrėžtumą. Apibendrinant, pagrindinis šio disertacijos tyrimo tikslas yra pagerinti žaidimų lygio objektų išdėstymo procedūrinės generacijos kūrybinį aspektą pasitelkiant genetinius algoritmus.

Darbo aktualumas

Automatizavimas gali žymiai pagerinti šiandieninį žmonių gyvenimo produktyvumą ir kokybę (Filip, 2021). Tačiau viena sritis, kurią įprastai tvarko žmonės, yra kūrybiškumas ir problemų sprendimas. Pastaraisiais metais mokslininkai daug dėmesio skiria kūrybiškumo problemų matematiniam modeliavimui. Algoritminių sprendimų taikymas kūrybinėms užduotims tampa vis populiaresnis, o kūrybiškumo bruožai paprastai yra susiję su naujumu ir kūrinių verte (Pichot et al., 2022). Multimedijos kūrimas yra daug išteklių reikalaujantis procesas, dažnai reikalaujantis didelių žmogaus pastangų. Taigi galimybė automatizuoti sudėtingesnes kūrybines užduotis gali labai paspartinti skaitmeninio turinio generavimą. Vaizdo žaidimai yra viena daugiausia skirtingų medijų sujungianti multimedijos šaka, kuri priklauso nuo kompiuterinio galimybių atkurti skirtingas medijos formas, todėl generuojant jų turinį galima pasiekti įvairų kūrybinį turinį. Tačiau tradicinis procedūrinis generavimas dažnai siejamas su atsitiktinumu ir chaosu, todėl kompiuteriu sukuriamas turinys yra ne toks įvairus ir autentiškas. Informatikos mokslo šakose tikslus kūrybiškumo apibrėžimas ir jo vertinimas yra neapibrėžtas. Kartu svarbu yra ne tik įvertinti kūrybinę vertę, bet ir sukurti efektyvų kūrybinės vertės modeliavimą.

Tyrimo objektas

Šio tyrimo objektas yra procedūrinis objektų išdėstymas žaidimo scenose pasitelkiant daugiakriterius sprendimų metodus modeliuojant genetinio algoritmo tikslo funkciją.

Darbo tikslas

Pagrindinis šio darbo tikslas yra sutrumpinti žaidimų dizaino kūrėjo darbo laiką sukuriant kūrybišką objektų išdėstymo generacijos algoritmą vaizdo žaidimams, padidinant objektų išdėstymo struktūros įvairovę.

Darbo uždaviniai

Darbo tikslui pasiekti buvo sprendžiami šie *uždaviniai*:

1. Atlikti vaizdo žaidimų scenų ir kūrybiškų artefaktų procedūrinio generavimo srities literatūros analizę.
2. Sukurti naujus metodus, skirtus žaidimo scenų objektų išdėstymo procedūriniam generavimui sujungiant genetinius algoritmus, daugiakritierius metodus ir neutrosofinius skaičius.
3. Genetiniam algoritmui sukurti tikslo funkciją ir kriterijus, kurie yra sufokusuoti didinti kūrybišką žaidimų dizaino vertę.
4. Sukurti žaidimų scenų generavimo variklį ir eksperimentuoti su skirtingais taisyklių junginiais ir procedūrinio generavimo patobulinimais siekiant sugeneruoti žaidimų scenas, kurios padidina automatizuotai sugeneruoto rezultato kūrybinę ir žaidimų dizaino vertę.

Tyrimų metodika

Šioje disertacijoje išanalizuotas esamų vaizdo žaidimų artefaktų generavimas susifokusuojant į žaidimų scenos objektų išdėstymo kompoziciją ir problemos formulavimą. Kuriant kūrybiškas žaidimo scenų generavimo strategijas, buvo pasitelkti genetiniai algoritmai, daugiakriteriai sprendimų metodai, abstrakčių skaičių logika ir ląsteliniai automatai.

Darbo mokslinis naujumas

1. Pritaikytas WASPAS-SVNS metodas ir CoCoSo patobulinimas procedūrinei genetinei žaidimo scenų struktūros generavimo tikslo funkcijai, todėl padidinamas neapibrėžtumas, kriterijų sujungimo efektyvumas ir sugeneruotų artefaktų įvairumas.

2. Kvantifikuoti ir sumodeliuoti tikslo funkcijos kriterijai pasitelkiant geštalto principus žaidimo scenų struktūros genetinio procedūrinio generavimo tikslo funkcijai siekiant padidinti estetinę scenos struktūros vertę.
3. Išplėstas WASPAS-SVNS procedūrinis generatorius su lokaliai morfuojančiais žaidimo objektais siekiant padidinti objektų tipų ir jų klasterių įvairovę.

Darbo rezultatų praktinė reikšmė

Tyrimo rezultatai gali būti taikomi žaidimų lygio aplinkoms generuoti, kai dėmesys skiriamas žaidimo objektų išdėstymo estetinei vertei. Pasiūlyto algoritmo praktinis pritaikymas gali būti naudojamas padėti žaidimų dizaineriui atliekant žemo lygio kūrybines žaidimo objektų išdėstymo užduotis.

Ginamieji teiginiai

1. WASPAS-SVNS ir CoCoSo metodai, pritaikyti genetiniame procedūriniame žaidimo scenų struktūros generavimo procese, sukuria kardinaliai skirtingų struktūrų aibę. Jie sėkmingai sukuria funkcionuojančias ir estetiškai įvairias žaidimo scenas neapibrėždami sugeneruotų formų algoritmo viduje.
2. Abstrakčiais geštalto principais sumodeliuota matematinė tikslo funkcija padeda sugeneruoti šiuos principus atitinkančias žaidimo objektų struktūras.
3. Lokaliai morfuojantys žaidimo objektų klasteriai padidina žaidimo scenų įvairovę padidindami galimų vizualių variacijų skaičių.

Darbo rezultatų aprobavimas

Tyrimų rezultatai disertacijos tematika buvo išspausdinti penkiose publikacijose. Trys straipsniai išspausdinti recenzuojamuose moksliniuose žurnaluose, indeksuotuose WoS duomenų bazėse (Petrovas & Bausys, 2022; Petrovas, Bausys, Zavadskas & Smarandache, 2022; Petrovas, Bausys & Zavadskas 2023); ir dviejose publikacijose, išspausdintose remiantis pranešimo medžiaga (Petrovas & Bausys, 2022 July; Petrovas & Bausys, 2019). Tyrimų rezultatai buvo pristatyti trijose tarptautinėse konferencijose:

- Tarptautinė elektros, kompiuterių ir energijos technologijų konferencija (*International Conference on Electrical, Computer and Energy Technologies* (ICECET)), Čekijos Respublika, 2022 m.
- Tarptautinės duomenų analitikos metodų ir programų sistemų dirbtuvės (*International Workshop Data Analysis Methods for Software Systems*) (DAMSS), Lietuva, 2019 m.
- Atvira elektros, elektronikos ir informatikos mokslų konferencija (*Open Conference of Electrical, Electronic and Information Sciences* (*eStream*)), Lietuva, 2019 m.

Disertacijos struktūra

Darbą sudaro įvadas, trys pagrindiniai skyriai, bendrosios išvados, literatūros sąrašas ir autoriaus publikacijų disertacijos tema sąrašas. Darbo apimtis − 97 puslapiai, tekste yra 44 formulės, 54 paveikslai ir 2 lentelės. Rašant disertaciją buvo pacituotas 101 literatūros šaltinis.

1. Procedūrinio generavimo metodų vaizdo žaidimams apžvalga

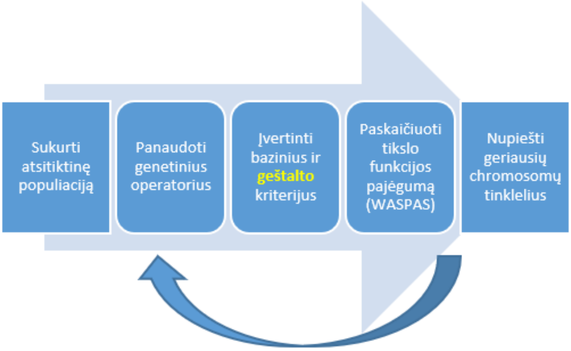
Šiame skyriuje apžvelgiami kūrybiško procedūrinio generavimo vaizdo žaidimams metodai ir pateikiama juos veikiančių sričių apžvalga. Jame aprašomos įprastos ir populiarėjančios strategijos, skirtos procedūriniam žaidimo scenų generavimui. Taip pat aprašomos dabartinės problemos, į kurias svarbu atkreipti dėmesį kuriant generacinį modelį. Siūlomas algoritmas fokusuojasi ties MCDM pagrindu sukurta genetinio algoritmo tikslo funkcija, kurios abstraktūs kriterijai yra skaitmenizuojami. Apžvelgtas kūrybiškumas, MCDM metodai, neapibrėžtumo aibės žaidimo scenų kūrimo procese.

Keturi pagrindiniai skaitmeninio kūrybiškumo tyrimo objektai yra asmuo, procesas, produktas ir spauda. Kiekvienas iš jų sukuria skirtingus iššūkius ir galimybes modeliuojant kūrybiškumą skirtingose srityse, tokiose kaip menai ar žaidimų dizainas. Pamatuoti ir apibrėžti kūrybiškumą yra sudėtinga skaičiuojamosiomis sistemomis. Didelė kūrybiškumo apibrėžimų įvairovė ir inžinerinių apibrėžimų nebuvimas sukuria sunkumus siekiant efektyviai modeliuoti ir vertinti kūrybišką rezultatą. Procedūrinio generavimo vaizdo žaidimams metu yra svarbu gerai sumodeliuoti tikslo funkciją. Aukšto lygio estetinių kriterijų inkorporavimas procedūriniam turiniui generuoti yra sudėtingas procesas. Abstraktūs estetiniai kriterijai, tokie kaip balansas, harmonija, vizualus patrauklumas, turi būti paversti tiksliais matematiniais algoritmais siekiant juos integruoti į efektyvų generavimo procesą. Nors ir abstraktūs vertinimo kriterijai egzistuoja, objektyvus jų vertinimas, ypač taikant daugiau negu vieną kriterijų, yra sudėtingas. Abstrakčių neutrosofinių aibių ir daugiakriterių sprendimų metodų taikymas suteikia galimybę valdyti abstraktumą kūrybiškuose algoritmuose. Šie metodai padeda sustiprinti sugeneruotų rezultatų įvairovę ir unikalumą. Taip pat yra sudėtinga surasti balansą tarp sugeneruoto turinio unikalumo ir sugeneruoto turinio funkcionalumo. Rasti šį balansą yra svarbu, nes sugeneruotos scenos turi būti unikalios, kad išlaikytų žaidėjo susidomėjimą žaidimo metu. Tikslo funkcija turi inkorporuoti ir estetinius, ir funkcinius žaidimų dizaino kriterijus tam, kad išlaikytų sugeneruoto turinio kokybę.

Atsižvelgiant į literatūros analizę sudaryti uždaviniai: sukurti žaidimų scenų generavimo karkasą, kuris leistų inkorporuoti kūrybinių sprendimų metodus; sukurti naujus žaidimo scenų objektų išdėstymo algoritmus pasitelkiant genetinius algoritmus, MCDM metodus ir neutrosofines aibes; sukurti tikslo funkciją ir kriterijus genetiniam algoritmui, kurie yra sufokusuoti į kūrybinę ir žaidimų dizaino vertę; optimizuojant generavimo taisykles ir sukurto algoritmo papildinius sugeneruoti žaidimų scenas, kuriomis bandoma pasiekti aukštą kūrybiškumo ir žaidimų dizaino vertę.

2. Genetiniai daugiakriterių sprendimų priėmimo metodai ir kriterijų modeliavimas vaizdo žaidimų scenoms generuoti

Antrajame skyriuje aprašomi ir tiriami genetiniai neutrosofiniai MCDM metodai ir jų kriterijų modeliavimas žaidimo scenoms generuoti (S2.1. pav.). Jame aprašomos kūrybiškumo problemos žaidimų turinio generavimo procese ir pasiūlomi nauji algoritmai žaidimų scenoms generuoti pasitelkiant WASPAS-SVNS ir CoCoSo pagrindu sukurtas tikslo funkcijas. Taip pat pristatomas geštalto dizaino taisyklių pritaikymas tikslo funkcijai pasitelkiant ląstelinio automato agentus. Galiausiai aprašomas būdas, kaip morfuoti žaidimo objektus, ir aprašomos išvados galutiniam generavimo modeliui.


Sukurti atsitiktinę populiaciją
Panaudoti genetinius operatorius
Įvertinti bazinius ir geštalto kriterijus
Paskaičiuoti tikslo funkcijos pajėgumą (WASPAS)
Nupiešti geriausių chromosomų tinklelius

S2.1 pav. Žaidimo scenų generavimas

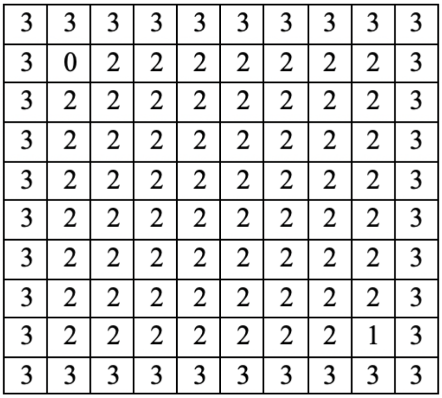
Tyrime siūloma automatizuota žaidimo scenos išdėstymo generavimo sistema, pagrįsta PCGML (procedūrinio turinio generavimas naudojant mašininį mokymąsi). Mūsų sistemoje naudojamas matematinis modelis, apimantis tikslo funkciją, kurią taiko genetinis algoritmas populiacijai įvertinti. Genetinio algoritmo tikslo funkcijai taikoma daugiakriterė sprendimų priėmimo (MCDM) naudingumo funkcija. Nustatomi konkretūs sudėtingumo, žaismingumo ir tinklelio dydžio koregavimo kriterijų parametrai. Kiekvienoje algoritmo iteracijoje žaidimo lygio tinklelis užpildomas ir vėliau įvertinamas. Taikant vertinimo procesą apskaičiuojamas tinkamumas kiekvienam žaidimo lygio tinkleliui, kad būtų galima pasirinkti našiausius tinklelius. Šis metodas generuoja įvairius ir netikėtus rezultatus, nes pirminiai generacijos duomenys yra atsitiktinai parinkti ir toliau tobulinami algoritmu.

Naudojamas standartinis žaidimo objektų rinkinys, parinktas pagal žaidimo lygio projektavimo principus. Šiuos objektus vaizduoja įvairūs skaičiai matricoje, kur kiekvienas skaičius atitinka skirtingą objekto tipą. Žaidimo scenos išdėstymas yra diskretizuotas į tinklelį, o kiekviename langelyje gali tilpti tik vienas objektas. Viena genetinio algoritmo chromosoma atitinka vienos scenos išdėstymą. Objektai ir atitinkami jų numeriai yra tokie:

- Žaidėjas (skaičius 0) – nurodo žaidėjo pradinę padėtį, veikėją, skirtą žaisti žaidimą.
- Išėjimas (skaičius 1) – pažymi vietą, kurią žaidėjas turi pasiekti, kad užbaigtų žaidimą.
- Tuščia vieta (skaičius 2) – reiškia neužimtas vietas, per kurias žaidėjas gali judėti.
- - Siena (skaičius 3) – reiškia objektą, kuris trukdo žaidėjo judėjimui.
- Pavojus arba priešininkas (skaičius 4) – reiškia objektą, kuris kelia pavojų žaidėjui.

- Surenkama esybė (skaičius 5) – žymi pageidaujamą objektą, kurį žaidėjas gali rinktis žaidimo metu.
- Paviršius – nors ir nėra užkoduotas chromosomų matricoje, šis objektas naudojamas 3D projekcijos vizualizacijos etape kaip grindų sluoksnis.

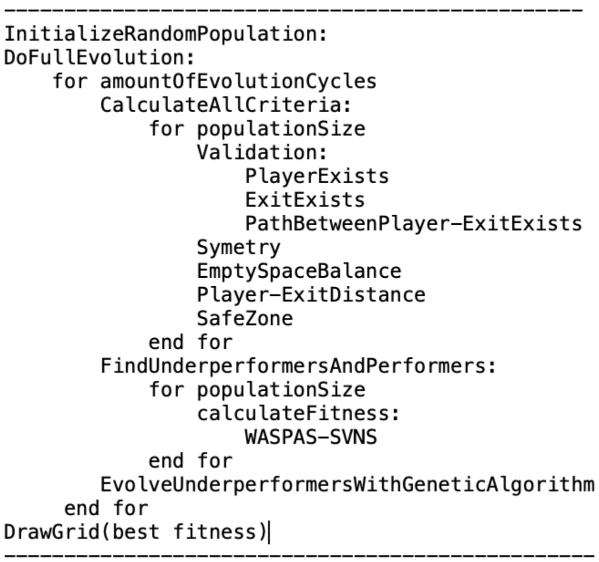
Vienos chromosomos informacija saugoma 2D skaitmeniniame tinklelyje, kaip parodyta S2.2 pav. Savo eksperimentams naudojame matricą, kurios matmenys yra 10 vienetų pločio ir 10 vienetų ilgio. Kiekvieną objekto tipą vaizduoja atskiras skaičius tinklelyje. Norėdami vizualizuoti galutinius rezultatus, projektuojame juos į 3D erdvę, pridėdami žemės sluoksnį po tinkleliu ir konvertuodami skaitines reikšmes į atitinkamus 3D objektus pagrindiniame tinklelyje.



3	3	3	3	3	3	3	3	3	3
3	0	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	2	3
3	2	2	2	2	2	2	2	1	3
3	3	3	3	3	3	3	3	3	3

S2.2 pav. Vienos chromosomos duomenų pavyzdys

Žaidimo lygio išdėstymas treniruojamas naudojant genetinį algoritmą (S2.3 pav.), o kiekvienos iteracijos vertinimo kriterijai sujungiami naudojant WASPAS-SVNS algoritmą vienai tinkamumo vertei apskaičiuoti. Aibės dydis nustatomas į 50, o algoritmas vykdomas 2000 iteracijų. Genetinis algoritmas naudoja atrankos ir mutacijų operatorius, kad filtruotų ir iš naujo apgyvendintų populiaciją. Norėdami inicijuoti duomenis, sukuriame tuščias chromosomas ir užpildome jas atsitiktiniais duomenimis, kur kiekvienas objektas yra užkoduotas sveikaisiais skaičiais nuo 2 iki 6 (atstovaujantis visiems galimiems objektams, išskyrus žaidėją (skaičius 0) ir išėjimą (skaičius 1)). Tada prie kiekvienos chromosomos pridedame vieną žaidėją ir vieną išėjimo objektą. Kiekvienoje iteracijoje apskaičiuojame visos populiacijos tinkamumo medianą ir padalijame chromosomas į dvi laikinąsias matricas, kurių viena saugo chromosomas žemiau vidutinės vertės, o kita saugo chromosomas virš vidutinės vertės. Chromosomos, esančios žemiau vidutinės vertės, pakeičiamos chromosomomis iš aukščiau esančios medianinės matricos, o tada 5 % šio naujo masyvo duomenų mutuoja priskiriant naujas atsitiktines reikšmes.



```
------------------------------------------------
InitializeRandomPopulation:
DoFullEvolution:
    for amountOfEvolutionCycles
        CalculateAllCriteria:
            for populationSize
                Validation:
                    PlayerExists
                    ExitExists
                    PathBetweenPlayer-ExitExists
                Symetry
                EmptySpaceBalance
                Player-ExitDistance
                SafeZone
            end for
        FindUnderperformersAndPerformers:
            for populationSize
                calculateFitness:
                    WASPAS-SVNS
            end for
        EvolveUnderperformersWithGeneticAlgorithm
     end for
DrawGrid(best fitness)
-------------------------------------------------
```

S2.3 pav. Genetinis algoritmas

Žaidimo scenos generavimas atliekamas taikant siūlomą genetinį Pitagoro neutrosofinį WASPAS metodą. Naudojamas matematinis modelis susideda iš kelių kriterijų, suskirstytų į tris tipus. Pirmąją grupę sudaro estetiniai kriterijai, susidedantys iš aukšto lygio kriterijų, kuriuos reglamentuoja geštalto taisyklės dariniai (s), ir žemo lygio kriterijų (v), tokių kaip simetrija ir tuščios erdvės balansas. Antroji grupė apima funkcinius kriterijus (f), kurie optimizuoja žaidimo dizaino vertinimus, tokius kaip atstumas tarp pagrindinių objektų ir saugios zonos aplink juos. Paskutinė grupė yra apribojimų kriterijų (c), kurie visada turi būti teisingi, kad tinkamumas būtų ne nulis. Šie apribojimo kriterijai apima kriterijus, susijusius su esminių elementų egzistavimu ir įmanomo kelio tarp žaidėjo ir išėjimo buvimu (S2.1 lentelė).

S2.1 lentelė. Visas kriterijų sąrašas

Kriterijus	Tipas	Verčių rėžiai
Panašumas (s_1),	Estetinė vertė (lokali funkcija)	0–1
Artumas (s_2),	Estetinė vertė (lokali funkcija)	0–1
Tęstinumas (s_3),	Estetinė vertė (lokali funkcija)	0–1
Centriniai taškai (s_4),	Estetinė vertė (lokali funkcija)	0–1
Bendras regionas (s_5)	Estetinė vertė (lokali funkcija)	0–1

S2.1 lentelės pabaiga

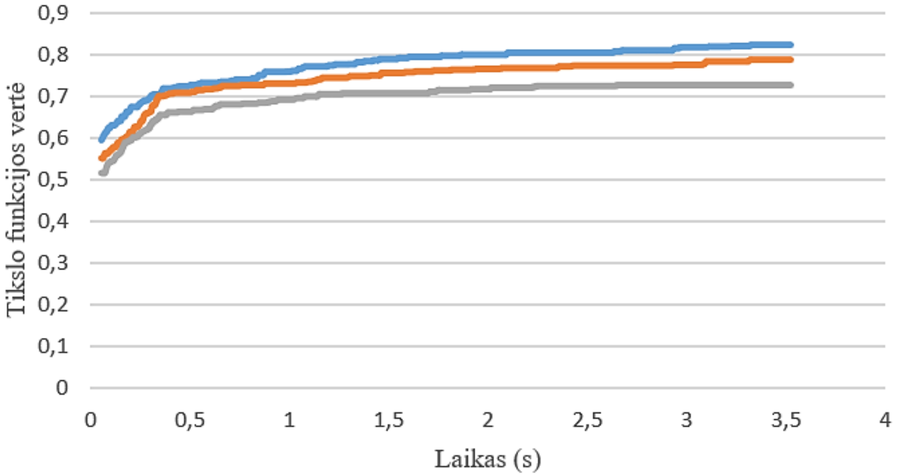
Kriterijus	Tipas	Verčių rėžiai
Simetrija (v_1),	Estetinė vertė (lokali funkcija)	0–1
Tuščios erdvės balansas (v_2)	Estetinė vertė (lokali funkcija)	0–1
Atstumas tarp žaidėjo ir išėjimo (f_1)	Tyrinėjimas	0–1
Saugi erdvė (f_2)	Saugumas	0–1
Žaidėjo egzistavimas (c_1)	Lygis pereinamas	0 arba 1
Išėjimo egzistavimas (c_2)	Lygis pereinamas	0 arba 1
Išėjimo kelio egzistavimas (c_3)	Lygis pereinamas	0 arba 1

3. Pasiūlyto procedūrinio generavimo metodo eksperimentavimas ir rezultatai

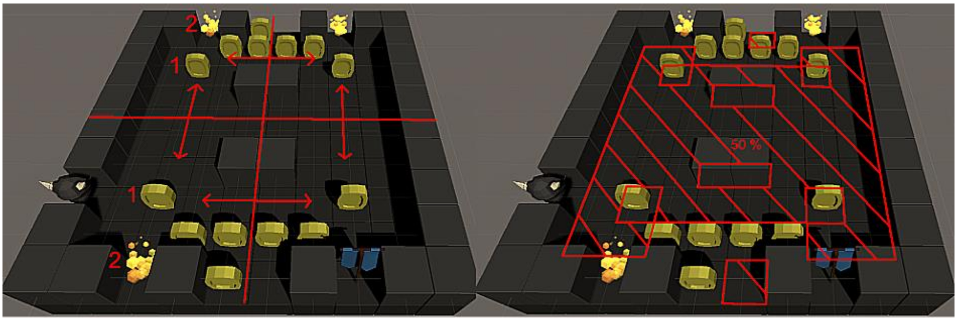
Trečiajame skyriuje pristatomi pasiūlytų genetinių neutrosofinių MCDM metodų ir geštalto principų pagrindu sumodeliuotos ląsteliio automato agentų tyrimai ir eksperimentavimų rezultatai. Sukurtos naujos procedūrinio generavimo strategijos siekiant išspręsti kūrybiškumo problemas, tokias kaip: aukšto lygio estetinių kriterijų inkorporavimas į tikslias matematines funkcijas, automatinis objektų struktūros diversifikavimas ir unikalumas, konfliktuojančių estetinių ir funkcinių kriterijų sujungimas nesugadinant galutinės scenos darnos. Skyriuje detaliai aprašomi sugeneruoti rezultatai.

Šiam tyrimui sukurta nauja sistema, naudojant *Unity* žaidimo variklį ir panaudojant vaizdinių žaidimų objektus iš *Unity Asset Store*. Bandymai buvo atlikti kompiuteryje, kuriame yra 2, 4 GHz 8 branduolių *Intel Core i9* procesorius. Procedūrinis generatorius su neutrosofinio vertinimo funkcija greitai generuoja didėjančius rezultatus pirmiesiems 100–200 kartų. Tačiau generatoriui dažnai reikia daugiau laiko, kad būtų galima sukurti simetriškus ir vizualiai subalansuotus scenų išdėstymus, kartu užtikrinant, kad būtų pritaikomos žaidimo taisyklės. Galutinė rezultato reikšmė paprastai nusistovi tarp 0,75 ir 0,85. Labai svarbu turėti daug lokalių maksimumų žaidimo scenai generuoti, nes tai užtikrina, kad rezultatai būtų unikalūs ir skirtingi. Egzistuoja daugybė galimų sprendimų, pagrįstų atsitiktiniais pradiniais duomenimis ir juos sekančiomis mutacijomis. Tikslo funkcijos reikšmė su skirtingomis atsitiktinių pradinių duomenų ir 500 genetinio algoritmo kartų variacijomis pateikti S3.1 paveiksle. Kartų skaičiaus pasirinkimas buvo nustatytas atsižvelgiant į momentą, kuriuo algoritmo iteracijos stagnuojasi, todėl rezultatai nėra atnaujinami arba atnaujinami tik nežymiai.

Estetinių kriterijų įgyvendinimą galima pastebėti vaizdiniuose pavyzdžiuose (S3.2 pav.): simetriją ir pusiausvyrą tuščioje erdvėje. Tuo pačiu metu efektyviai įgyvendinami žaidimo dizaino reikalavimai, tokie kaip kelio paieška ar svarbių objektų egzistavimas.

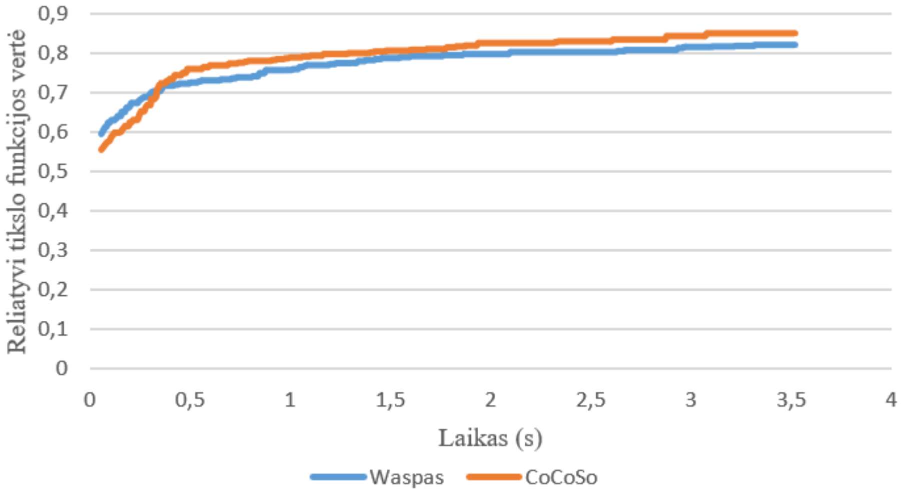


S3.1 pav. Tikslo funkcijos pavyzdžiai, kiekviena kreivė reprezentuoja atskirą algoritmo iteraciją



S3.2 pav. Simetrija ir tuščia erdvė

Tikslo funkcijos reikšmė pradeda stabilizuotis ir sulėtėti maždaug po 80–120 kartų, o tai rodo greitesnę konvergenciją, palyginti su WASPAS įgyvendinimu. Tačiau tiesioginis vertės palyginimas neįmanomas, nes taikant CoCoSo metodą gaunamos vertės, nepatenkančios į 0–1 diapazoną. Mūsų demonstracinėje versijoje tikslo funkcijos vertės paprastai svyruoja nuo 4 iki 8. Tikslo funkcijos kreivių palyginimas laikui bėgant tarp CoCoSo ir WASPAS metodų pavaizduotas S3.3 pav.



S3.3 pav. Tikslo funkcijų kreivių palyginimas tarp WASPAS ir CoCoSo metodų

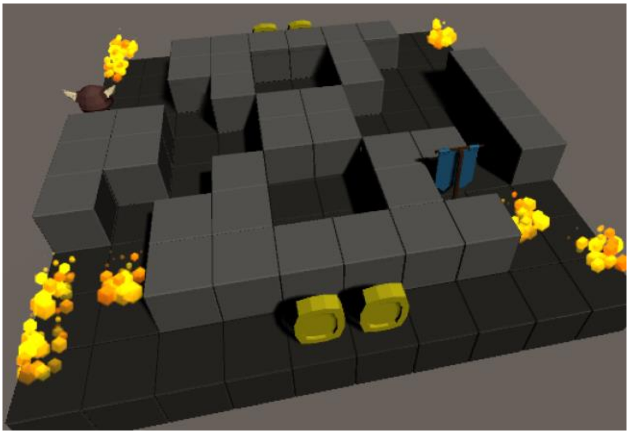
Išnagrinėjus, kokią įtaką kiekvienas kriterijus turi bendrame lygių generatoriuje, visi kriterijai buvo sujungti, kad būtų suformuota tikslo funkcija. Rezultatuose galime stebėti atskirų geštalto taisyklių modelius ir pastebėti skirtumus tarp atskirų to paties algoritmo iteracijų. Pirmajame pavyzdyje galima pastebėti sienas, išdėstytas „8“ skaičiaus forma (S3.4 pav.). Antrasis pavyzdys sudaro uždarą erdvę su monetų grupe viduryje ir paslėptu išėjimu (S3.5 pav.).

S3.1 lentelė. Expertų apklausa

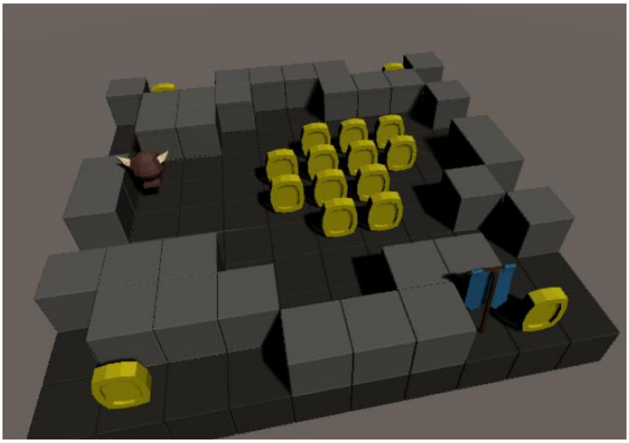
Sample	S1	S2	S3	S4	S5	Q1	Q2	Q3
3.25	+			+	+	+	+	
	+		+			+	+	
	+			+	+	Atsisakė	Atsisakė	Atsisakė
	+	+	+		+	+	+	+
	+	+	+	+	+	+	+	+
3.26	+	+		+	+	+	+	
				+	+		+	
		+		+	+	Atsisakė	Atsisakė	Atsisakė
	+	+	+		+	+	+	+
	+	+	+	+	+	+	+	+

Apklausa buvo atlikta siekiant sužinoti ekspertų nuomones, įskaitant vartotojo sąsają, vartotojo patirtį ir žaidimų dizaino ekspertus iš mokslo bendruomenės ir vaizdo žaidimų pramonės. Ekspertų buvo paprašyta įvertinti du sugeneruotus pavyzdžius (S3.4 ir

3.5 pav.). Pirmasis klausimas buvo skirtas faktiniam geštalto principų matomumui, o antrasis – estetinei kokybei palyginti su kitais literatūros pavyzdžiais. S1–S5 atitinka geštalto principų matomumą (panašumas, tęstinumas, bendras regionas, artumas, židinio taškas), o Q1–Q3 atitinka subjektyvią kūrybinę vertę, sulygintą su artimiausiais literatūros pavyzdžiais (Safak et al., 2016; Zafar et al., 2020; Thakkar et al., 2019). Kai kurie ekspertai pabrežė, kad abstrakčios vertės vertinimas nėra tinkamas dėl skirtingų generatyvinių tikslų ir kitų estetinių savybių, tokių kaip spalvų ir linijų formos, kurių negalima kiekybiškai įvertinti. Dėl šios priežasties vienas ekspertas atsisakė palyginti kokybę (žr. S3.1 lentelę). Rezultatai parodė, kad panašumas buvo matomas 80 % atsakymų, tęstinumas – 60 %, bendras regionas – 50 %, artumas – 70 %, o židinio taškas – 90 %. Estetinė kokybė buvo įvertinta 87,5 %, 100 % ir 50 % atsakymų, lyginant su kitais literatūros pavyzdžiais.



S3.4 pav. Galutinio algoritmo rezultatas sujungiant geštalto principus 1



S3.5 pav. Galutinio algoritmo rezultatas sujungiant Geštalto principus 2

Bendrosios išvados

1. WASPAS generatoriaus patobulinimas CoCoSo metodu pagerino tikslo funkcijos didžiausiąją vertę ir skaičiavimo greitį (~80 % greitesnis konvergacijos šuolis). Neutrosofinių aibių inkorporavimas padidino sugeneruotų artefaktų įvairovę (~35 % dažnesni sugeneruoti unikalūs artefaktai) ir subalansavo estetinius ir funkcinius kriterijus. Kūrybiškų sprendimų įžvelgimas padidėjo ankstyvose algoritmo generacijose.
2. Lokaliai morfuojančių objektų algoritmo inkorporavimas padidino žaidimo scenų įvairovę pasitelkiant natūralų atsitiktinumą. Galutinės scenos pridėjo naujų objektų variacijas, tokias kaip žolė, medžiai ir akmenys, sugeneruotoms scenoms išlaikant visas žaidimo dizaino taisykles (~25 % scenos sugeneruoja 100 % didesnį objektų variacijų skaičių). Šis algoritmo papildymas parodė jo galimybę padidinti žaidimo scenų kūrybiškumą ir dinamiškumą.
3. Geštalto principai buvo integruoti į pitagorinį neutorosofinį WASPAS žaidimo scenų generatorių. Šis sprendimas padidina vizualinę sugeneruotų scenų vertę, tačiau konfliktuojantys kriterijai gali išblukinti kai kurių estetinių kriterijų vertę. Geštalto principų integravimas parodo potencialą vizualiai gerų žaidimo scenų generavimo procese (250 % didesnis automatinės estetinės analizės kriterijų skaičius). Ekspertų apklausa parodė, kad panašumas buvo matomas 80 % atsakymų, tęstinumas – 60 %, bendras regionas – 50 %, artumas – 70 %, o židinio taškas – 90 %. Estetinė kokybė buvo įvertinta kaip geresnė 87,5 %, 100 % ir 50 % atsakymų, lyginant su kitais literatūros pavyzdžiais.

Aurimas PETROVAS

COMPUTATIONAL CREATIVITY IN VIDEO GAME
SCENE GENERATION BY GENETIC
MULTI-CRITERIA DECISION-MAKING METHODS

Doctoral Dissertation

Technological Sciences,
Informatics Engineering (T 007)

SKAITMENINIS KŪRYBIŠKUMAS ŽAIDIMŲ SCENOS
GENERAVIME TAIKANT GENETINIUS DAUGIAKRITERIŲ
SPRENDIMŲ PRIĖMIMO METODUS

Daktaro disertacija

Technologijos mokslai,
informatikos inžinerija (T 007)

Lietuvių kalbos redaktorė Dalia Markevičiūtė
Anglų kalbos redaktorė Jūratė Griškėnaitė

2023 03 29. 9,3 sp. l. Tiražas 20 egz.
Leidinio el. versija *https://doi.org/10.20334*/2024-002-M
Vilniaus Gedimino technikos universitetas
Saulėtekio al. 11, 10223 Vilnius
Spausdino UAB „Ciklonas“,
Žirmūnų g. 68, 09124 Vilnius