

Article

Biome Generation Tool: A Mixed-Initiative Software for the Procedural Generation of Biomes

Gabriel K. Sepúlveda [†], Felipe Besoain [†] , Sven von Brand and Nicolas A. Barriga ^{*} 

Faculty of Engineering, Universidad de Talca, Talca 3460000, Chile; gsepulveda17@alumnos.otalca.cl (G.K.S.); fbesoain@otalca.cl (F.B.); svonbrand@otalca.cl (S.v.B.)

^{*} Correspondence: nbarriga@otalca.cl

[†] These authors contributed equally to this work.

Abstract: This article presents the Biome Generation Tool (BGT), a mixed-initiative procedural content generator plugin for the Unity3D game engine that uses an Evolutionary Algorithm (EA) for the creation of biomes in videogames. A double-blind evaluation was performed with 28 videogame developers split into two groups. The experimental group received a standard BGT, while the control group received a version optimizing for the negative of the fitness function. A survey with semantic differential questions focused on AI impact, productivity, and user satisfaction was applied. The experimental group's appreciation of the AI suggestions, perceived productivity increase, and satisfaction were significantly higher than those of the control group. From the results, we conclude that the BGT succeeds in assisting the development of biomes, generating products of acceptable quality while reducing the designer's workload. In the future, we expect to confirm these results with a separate group of users evaluating the biomes built using BGT.

Keywords: artificial intelligence for videogames; procedural content generation (PCG); evolutionary computation in video games; artificial intelligence to support game and level design



Citation: Sepúlveda, G.K.; Besoain, F.; von Brand, S.; Barriga, N.A. Biome Generation Tool: A Mixed-Initiative Software for the Procedural Generation of Biomes. *Appl. Sci.* **2023**, *13*, 8070. <https://doi.org/10.3390/app13148070>

Academic Editors: Paolo Renna, Laura Anna Ripamonti, Davide Gadia and Dario Maggiorini

Received: 28 April 2023

Revised: 30 June 2023

Accepted: 3 July 2023

Published: 11 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Creating a videogame or interactive experience is a complex process that requires a balance between creativity, narrative, and technological implementation [1]. In this scenario, the map or level is an essential component in various game genres, regardless of their complexity. This is because maps provide coherence between all other components, so they must be designed to support the mechanics and characteristics of the game while providing the proper space to develop a good game flow.

The confection of a map by a designer can take a lot of time and resources [2], even more when considering that a vast number of games have quite a few maps, or the maps used are extensive, or both. Nevertheless, the design and development of game maps are highly time-consuming processes. Therefore, researchers and developers work constantly to simplify these tasks in the development process.

One of the reasons procedural content generation (PCG) systems were created was as an attempt to overcome this complication [3]. PCG can reduce production costs [4], but it comes with a trade-off. For PCG to generate a final product, the algorithm must be sophisticated enough or iterate multiple times and select which one to use among the results. In both cases, the designer has decreased control over the generated product, and even if the result is good, it may end up not being used if it does not follow the designer's intentions. Therefore, creativity and control in the design process become unbalanced and may lead level designers to avoid using these kinds of tools.

How can we benefit from PCG algorithms without losing control over the outcome?

A recent research area in content generation is *mixed-initiative* tools. Mixed-initiative tools delve into the area of Human–AI interaction to create a system in which an algorithm and

the designer collaborate to create the content, thus reducing the workload and keeping a high degree of control.

In this context, we have developed a mixed-initiative tool for the creation of map environments in the Unity3D game engine, called Biome Generation Tool (BGT). A biome is a set of plants and animals that share common characteristics due to jointly inhabiting a particular environment. These organisms usually present multiple interrelationships between themselves and their surroundings. Biomes are important in game map design, particularly in genres such as Real Time Strategy, Adventure, or Survival.

This tool is expected to reduce the workload and time invested in the process of making levels while maintaining the degree of control the designer has over the generated content, allowing them to convey their intent. Most of the mixed-initiative procedural content generation tools described in the literature are standalone research platforms (see for example [5–9]). However, as a differential factor, BGT has been designed to be integrated into a world-class game engine, Unity3D.

In this work, the creation process of BGT is described. BGT is a biome generation tool that lets the designer edit and create map environments for pre-existing maps. It allows the user to decide what objects will be part of the environment, define the area on the map where they will be placed and control the number of elements that will be included.

BGT is also equipped with an AI based on an Evolutionary Algorithm (EA) [10] that evaluates the current state of the designer's work and provides insight or suggestions for modifications. An EA was selected as the AI for BGT due to its robustness and effectiveness. EAs provide good results over large populations and are less susceptible to noise or changes in input throughout the process [11]. Another advantage is that they do not need pre-existing training data as supervised learning methods do. The designer can control the interference of the AI by enabling or disabling it and choosing whether to apply or ignore the modifications.

To validate BGT, a controlled study and a survey were conducted, which measured the usefulness of the AI assistance, potential productivity gain, and overall user satisfaction. Based on this, BGT's ability to serve as an edition software to create biomes is expected to be confirmed, as it should lower costs while maintaining designer control.

The main contributions of the present study are as follows:

1. A double-blind user study showing EA-generated suggestions in a mixed-initiative generation tool are useful to create better biomes and have the potential to increase productivity;
2. A multi-layer architecture that allows more complex interactions between map elements than previous EA mixed-initiative tools;
3. An implementation of an AI-assisted extensible mixed-initiative biome generation tool in Unity3D.

Section 2 presents a literature review of research in the field of mixed-initiative PCG using Evolutionary Algorithms. Section 3 provides an overview of the creation process of the BGT and the user study details. Finally, Section 4 presents the study results, followed by conclusions and future work.

2. Literature Review

Mixed-initiative content generation covers a wide range of systems in which a human creator and a computational creator interact to produce new content [12]. The amount of initiative and impact of the computational side can vary wildly between systems, from light assistance to almost complete independence. However, the software should be a proactive co-creator rather than a tool that merely reacts to user input [13]. This study focuses on tools in the middle of this spectrum, promoting human creativity rather than simply providing assistance [14]. These tools allow creators to be both productive and innovative [15] while avoiding potential risks of fully autonomous PCG, such as repetitive or unplayable content. Out of the vast amount of published mixed-initiative tools for content creation, some of the most relevant applications using Evolutionary Algorithms (EA) include the following:

- **Sentient Sketchbook:** One of the first examples of the use of EAs in mixed-initiative tools is Sentient Sketchbook [9]. Sentient Sketchbook is a tool for creating maps supported by an EA that evaluates the current state and suggests modifications to assist in the process and inspire the designer. With this tool, users can edit low-resolution abstractions of maps with four different elements: passable tiles, impassable tiles, resource tiles, and player bases. The map is evaluated regarding resource safety, safe area, exploration, resource safety balance, safe area balance, and exploration balance. For each one, the tool displays a playability statistic, a map suggestion that maximizes said statistic, and another six suggestions that maximize diversity through novelty search. The tool updates both the statistics and suggestions each time the user performs a change. The suggestions are generated by a two-population EA using a feasible–unfeasible approach. Each population contains 10 chromosomes and runs 10 epochs before selecting a solution. Research concludes that an EA with a feasible–infeasible two-population paradigm [16] is more robust than the other state-of-the-art novelty search approaches of that time for finding feasible solutions. It also states that the tool shows potential due to the usefulness of the suggestions and feedback provided to the designer [9].
- **Baba Is Y'all:** In recent research, use of EA can be seen in a mixed-initiative tool to create maps for the game Baba is You. This study aims to provide a tool where multiple users can interact with a PCG system to create content. Baba is Y'all [6] is a web service tool that allows users to edit, test, evaluate, and upload levels for the Game Jam version of Baba is You. The tool uses an EA to suggest modifications to the user using previously created maps stored in the cloud, including those uploaded by other users, as chromosomes for the initial population. The tool has a Generation Module that can be accessed at any time by the user. This module allows the user to change the variables of the EA, such as the stopping criteria, mutation rate, and initialization method, to customize the EA without changing the algorithm's inner workings. The objective of the EA is to evolve towards diversification and fulfillment of missing requirements on the map. The results of the research were promising but hampered by the lack of maps in the tool cloud and the difficulty of the learning curve. Nevertheless, it worked as validation and a guide to proceed with the efforts to keep working on the tool.
- **Evolutionary Dungeon Designer:** Another project in this field focuses on the subject of map creation in rogue-like games [7]. This research is the expansion of a previous one in which the topic is a mixed-initiative tool for creating rooms in rogue-like games [17]. Evolutionary Dungeon Designer (EDD) is a mixed-initiative tool for the creation of dungeon rooms in videogames. The tool provides an interface where users can edit, generate, save, and load rooms. The rooms are tile grids that can be populated with six kinds of tiles: floor, wall, enemy, treasure, entrance, and door. The tool provides a view in which the user can freely edit the room, and another view to generate rooms using the EA, which only opens at the start of the session. The EA will provide six suggestions for different rooms with the option of re-generating them if none of them are attractive to the designer. After clicking an option, its data will be automatically loaded into the editor view. The EA used to generate the rooms runs by keeping two populations, each one storing either feasible or infeasible solutions. By keeping two populations, it is possible to preserve valuable data that might be lost when discarding the infeasible maps, thus providing a higher level of diversity to the generated solutions. To evaluate the quality of a room, the EA takes into account multiple factors such as the presence of micro patterns such as corridors, intersections, turns, or chambers, the rate of treasure tiles and enemy tiles, the safety of the entrance, and the meso-patterns created by a combination of micro-patterns [17]. In a separate study, the authors modify the tool to enhance it. The new tool is capable of creating whole dungeons of a size ranging from 2×2 to 7×7 rooms instead of independent dungeon rooms. The EDD modification offers a third view, the world

view, in which the user can select a room to disable it, generate suggestions for the room, or open the room editor. The suggestion view works the same as the previous version but it is accessible by the user at any time. The room view is also enhanced by adding a Navigation functionality that enables the user to go from one room to another using either buttons that allow for movement between neighboring rooms or a mini-map to move to any desired room. This view also includes a section to generate suggestions for the current room, allowing the designer to use the EA without needing to change the view. The study concludes that the changes performed to the previous EDD have successfully enhanced the tool by providing further context, feedback, and flexibility [7].

- **Other:** There are several existing tools for biome creation in Unity. Some are fully procedural, while others allow users to select which areas of the world to generate elements in, or to otherwise configure the generation. However, none of them is a mixed-initiative tool, as there is no co-creation of content. Most tools only allow designer intervention before or after the generative step [18]. Examples of these PCG-assisted tools are Gaia Pro (<https://assetstore.unity.com/packages/tools/terrain/gaia-pro-2021-terrain-scene-generator-193476>, accessed on 29 June 2023) and Map-Magic2 (<https://assetstore.unity.com/packages/tools/terrain/mapmagic-2-165180>, accessed on 27 April 2023). Other mixed-initiative Unity plugins exist but do not use Evolutionary Algorithms. An example is Coralize [19], a middleware tool for generating corals and sponges using constructive algorithms, such as L-grammars.

Because mixed-initiative PCG is a recent research field, studies are being conducted not only on the development of new prototype tools, but also on the drafting of how to handle the Human–AI interaction and on the definition of standards to measure the proper behavior of the AI. Current studies are built from theory, but there is also ongoing research focused on analyzing, applying, and evaluating the methodology used in existing mixed-initiative tools [20].

Mixed-initiative tools should be adapted for the designer’s ease of use rather than the other way around. However, while AIs have increased in sophistication, they have also become more difficult to use. Considering this, the creation of a new research field of Explainable AI for Designers (XAID) has been proposed [21]. In this context, [22] Gorm Lai and colleagues present three fundamental pillars that tools must be based on regarding the Human-AI interaction:

- **Respect the designer’s control:** The designer is the one who must control the creation process. The application must provide this control to the designer, and the AI must not overrule the designer’s decisions but work around them.
- **Respect the creative process:** The creative process can be defined as a loop of hypothesis, modification, feedback, and update. The tool must not disrupt this cycle, which means it has to give appropriate feedback to the designer at an appropriate time threshold. A lack of information or a late response can break the creativity flow.
- **Respect the existing work process:** During the creation of a videogame, multiple tasks are being performed simultaneously by different people.

Some of these tasks are dependent on others. The tool must provide an output that does not disturb the production chain, which means the output must be usable by the rest of the processes. The use of the tool should not interrupt or be interrupted by other tasks under normal conditions. Because they are useful in integrating mixed-initiative PCG to the videogame industry, the present work is based on these three pillars.

Finally, a study using Morai Maker, a Super Mario Bros AI-assisted level editor, indicates that users tend to assign roles to AI assistants. They have identified four roles [23]: **Friend**, interaction with the AI is seen primarily as a fun activity. **Collaborator**, the AI is viewed as an equal design partner. **Student**, they expect the AI to follow their lead. **Manager**, they treat the AI as giving instructions and that the human should follow the AI’s lead. This work focuses on the Collaborator and Student roles.

3. Methodology

In this research, the methodology follows the design and creation approach [24] to create and evaluate the tool. First, a mixed-initiative PCG tool was developed for creating biomes in existing maps, with a multi-layered architecture for generating AI suggestions. Second, this tool was evaluated through a double-blind user study designed to assess the impact of the AI assistance on end users in four dimensions: AI suggestions, AI creation features, productivity, and satisfaction. For all four dimensions, the hypothesis is that the means of each dimension from the experimental group will be significantly different and greater than those from the control group.

3.1. BGT

BGT is a Unity package that provides suggestions to a designer creating a biome. These suggestions are produced by an Evolutionary Algorithm, starting from the user's design and maximizing a user-defined fitness function. The map is represented as a 2D top-down height map view divided into separately colored layers for each group of elements. This prototype is preloaded with a sample biome kit with 4 types of elements: trees, plants, animals, and minerals. There are 18 different trees, 19 plants, 16 minerals, and 6 animals to choose from. Each of them contributes something to the biome and may require something else. This contribution or requirement is called a *stat*. For instance, most trees provide *shade*, which is required by any animals included (*lions*, for example), along with *grass*. The specific stats and requirements can be found in the project's repository (<https://github.com/Prion6/BiomeGenerationTool#basic-biome-testing-kit>, accessed on 27 April 2023). These requirements and contributions are taken into account by BGT when generating suggestions. A sample biome kit was built so that the user study procedure would not be overly lengthy. The tool is planned to be distributed along with a few sample biome kits, but users can easily build their own.

3.1.1. Evolutionary Algorithm Core

The base chromosome used is a representation of the layer alpha texture stored in a float matrix of size N^2 , hence the operation time will be of at least $O(N^2)$. Bigger chromosome sizes will mean a higher detail level but also an increase in computing cost. At the start, size $N = 2048$ was used in a population of 10 chromosomes, but the extended execution time of over 2 min to complete 1 generation was unsustainable. Through multiple iterations, a size of $N = 128$ was reached, with an adequate degree of detail and a process running time of less than a second for each generation. The first population of the EA is created based on the current state of the texture scaled to the chromosome size. To generate the first group of chromosomes, the texture is distorted by a shift in position and value using Perlin noise. Three textures of Perlin noise of N^2 size are generated; the first and second determine the x and y shifting in the x,y position, amplified by a set value called amplitude. The third texture determines the shift in the alpha value stored in the chromosome, scaled down by a factor between 0 and 1. The triple shifting results in an unevenly distorted image. In order to fix this, first, each value is replaced with an average between the value and its neighbors. This process is repeated until the desired result is obtained. The process blurs the matrix, providing a more coherent shape but also making an undesired shift in the values. The second fix calculates the proportion between the new and old chromosomes' maximum value, inverts it, and then applies it to all the genes in the new chromosome, reverting the undesired shift in the values. Since the algorithm needs to provide options according to different criteria, each layer runs multiple EAs. Each layer can run a maximum of 8 EAs in parallel threads with a population of 10 chromosomes. The fitness function of each EA is selected by the user.

Also, taking this into account, the degree of change between the chromosomes of one generation and the next needs to be enough to create a difference from the original texture that is noticeable to the user.

3.1.2. Crossover and Mutation

Crossover consists in an exchange of subsections between chromosomes. Both subsections are of size $(N * scale)^2$, and the scale is a number between 0 and 1. To avoid rough edges on the crossover, the exchange is performed through interpolation using the distance from the center of the subsection as the change factor. In this way, the innermost values end up corresponding to those of the foreign subsection, while the furthestmost values remain as the original ones. The mutation modifies the chromosome in one of three possible ways chosen at random in an area of $(N * scale)^2$. The first option works in the same way that the distortion function is used to create new chromosomes. The second option is equivalent to a crossover between the chromosome and a section of the Perlin noise where only the mutated chromosome is modified. The third option works as a crossover between the chromosome and the current state of another layer affecting only the chromosome. The last mutation alternative exists to encourage divergence or convergence among layers based on the fitness function.

Table 1 shows the parameters used by the EA in the tool. Since the objective pursued by the designer can be different depending on the game or their vision, the fitness function field was left open, allowing the game developer to implement their own function.

Table 1. Settings used for the Evolutionary Algorithm properties.

Property	Value
Chromosome size	128 × 128
Chromosome generation	Perlin Noise [21] Distortion
Population size	10
Crossover method	k-point crossover
Crossover section	$(N * scale)^2, N = 128, 0 < scale < 1$, where scale is the squared percentage of the square to recombine
Mutation method	Perlin Noise Distortion, Perlin Noise Partial Crossover, and Inter Layer Partial Crossover
Mutation section	$(N * scale)^2, N = 128, 0 < scale < 1$, where scale is the squared percentage of the square to mutate
Maximum concurrent populations	8
Stop criteria	5 generations

For the mixed-initiative tool to work properly, the algorithm must provide various options distinct enough from each other for the user to notice the difference. Therefore, multiple fitness functions are required to allow the EA to evolve through different branches. In this context, BGT provides a basic set of fitness functions:

- **Connectivity:** Fraction of the layer that does not have elements with a collider (i.e., walkable fraction).
- **Reach Fairness:** If there are more than 2 starting positions, it measures how balanced the walking distances between them are.
- **Stat fairness:** Configurable per stat. Measures how balanced a given stat is between starting positions.
- **Stat gain:** Configurable per stat. Fraction of the layer that provides a given stat.
- **Stat deprivation:** Configurable per stat. Fraction of the layer that does not provide a given stat.
- **Resemblance:** Similarity measure between suggestion and actual layer.
- **MultiLayer:** Applies a given fitness function to multiple layers. Results combined using L2 norm [25].
- **Multievaluation:** Applies multiple fitness functions to a single layer. Results combined using L2 norm.

- **Coherency:** Fraction of the map where the element's stat requirements are met by contributions from other elements in any layer. For instance, lions require shadow and grass. A layer with only lions, and where every lion is in a tile in which another layer is providing it with shadow and grass, will have a coherency of 1 (maximum).

Through the different fitness functions, BGT allows the user to pursue different objectives in the design process. Taking into account that the user may want to explore various options simultaneously, BGT allows the designer to select multiple fitness functions at the same time. For each fitness function selected, the tool will evolve a different population in its own thread. Each population provides the user with a distinct suggestion, avoiding previously reported issues with mixed-initiative tools providing suggestions that were too similar to each other [26].

3.1.3. BGT Interface

BGT runs in a Unity custom window using an EA as the AI. The window is composed of 5 sections:

1. **Draw Area:** Figure 1A shows a 512×512 square area in which the designer can view and edit the current state of the layers. It is a nearest-neighbor upscaling of the 128×128 chromosome. The background texture corresponds to a height-map of the Unity terrain in which the biome is created. The user can modify the textures of the selected layer with a brush, by right-clicking to add color saturation or left-clicking to remove it, thus controlling the density of the elements.
2. **General Settings:** Allows the user to set the starting points of the players on the map and to manage the layers by adding, removing, selecting, and setting them as visible or hidden. Shown in Figure 1B.
3. **Draw Settings:** Figure 1C shows options to change the brush size, brush intensity, painting mode, whether the starting points are displayed, generation density, and background terrain. It also contains the button to generate the biome on the Unity terrain.
4. **Layer Settings:** Figure 1D shows the settings of the currently active layer. It allows users to modify the name, drawing order, and color of the layer along with the map elements that compose it and the fitness functions that will be used to generate new map options.
5. **Options Display:** Figure 1E shows AI-generated suggestions. While the EA runs, it displays option buttons based on the fitness functions selected in the layer settings. The buttons show the suggested new layer state. If the button is pressed, the layer texture is replaced with the button texture. The user can also pause or stop the EA cycle.

3.1.4. Implementation Challenges

The BGT was developed as a plugin for version 2019.4.X of the Unity Engine. As such, this version is required to implement BGT's user interface within the Unity Editor. Although in recent Unity versions the UIBuilder library can be used for such purposes, in 2019.4.X it was still an experimental feature. It is not until version 2021.2.X that UIBuilder is stable and documented. BGT's interface was coded using the GUI and EditorGUI libraries provided by Unity. This is a time-consuming process, and adding flexibility and additional features to the UI is not straightforward.

Another challenge was the chromosome size for the Evolutionary Algorithm (EA). A balance must be achieved between map editing resolution and processing speed. While the concept of the tool is the creation of biomes to encompass large portions of maps, the EA needs to work with a data size that allows for quick user interaction. The tool uses a 512×512 map resolution for the user, but a 128×128 nearest neighbor downsampled chromosome is needed for the EA, in order to alleviate the load without losing too much information.

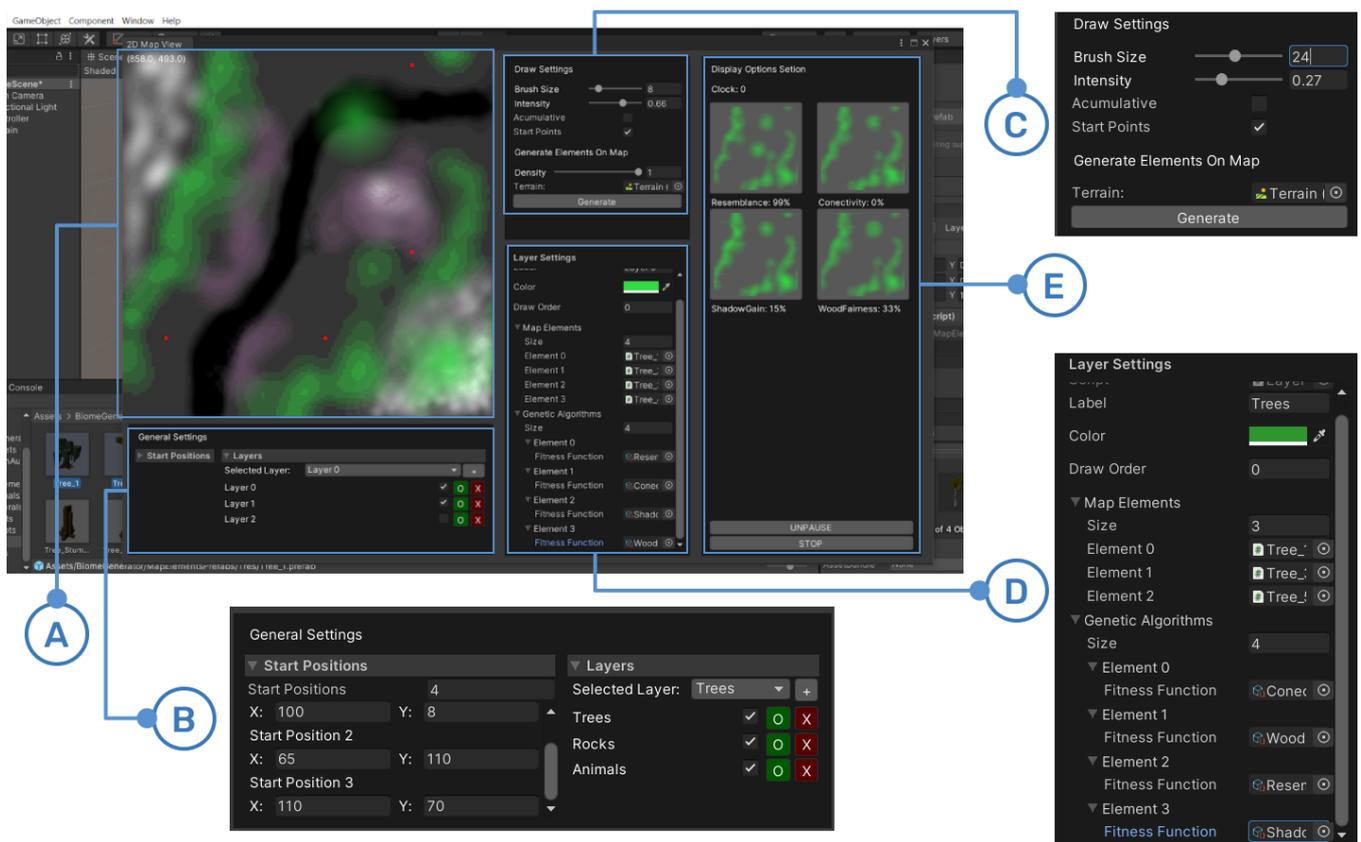


Figure 1. BGT screenshot in the Unity3D engine among its components: (A) Draw Area Panel: the green area corresponds to trees, the purple area to minerals, and the brown areas to animals; (B) General Settings Panel; (C) Draw Settings Panel: Brush size ranges from 1 to 52, intensity from 0 to 1; (D) General Settings Panel: BGT’s display options and layer settings; (E) Layer Settings Panel: Display Options Panel.

3.2. User Study

3.2.1. Participants

A total of 28 undergraduate students (5 female, 22 male, 1 non-binary; age $M = 22.4$, $SD = 2.23$) (32 students were recruited, but 4 were discarded for not following the procedure (using at least 4 fitness functions and generating one map)) from the School of Videogame Development and Virtual Reality Engineering at Universidad de Talca were recruited and asked to review and test the BGT. Students with this background were recruited due to the specificity of the task and as a representation of the game development profession/industry. Completion of the activity and questionnaire occurred in a computer laboratory at the end of their classroom lecture and took approximately 60 min.

3.2.2. Procedure

To conduct a double-blind evaluation, two different BGT versions were built. The experimental group received the tool described in the previous subsections. The control group received the same tool but optimized for the negative of the fitness function. This had the effect of producing suggestions that minimized the user’s chosen feature, instead of maximizing it. For instance, if a user in the control group chose the *connectivity* fitness function (see Section 3.1), the suggestions would have less walkable area than the original layer. This was done because just deactivating the AI suggestions, or providing suggestions with random perturbations was too obvious, thus negating the benefits of a double-blind trial. In this scenario, the BGT interface was the same for both groups.

Both tools were alternately installed in a 40-seat computer lab, 20 experimental and 20 control. All computers were running Unity 2019.4.x on Windows 10 and were equipped with 16Gb of RAM, an Intel Core i5-10400f CPU, and an NVIDIA GTX 1660 super GPU. The person conducting the experiment and interacting with the participants did not know which computer had which version of the tool.

Students were invited to participate in a study for evaluating BGT. They were informed that their task included using and creating map environments using the tool, then answering a questionnaire with different questions related to the experience and BGT. In this context, all participants were given an informed consent form at the beginning of the procedure. Completing and returning this form allowed them to participate in the study.

First, an introduction to the mixed-initiative Biome Generation Tool was presented to the participants with prepared slides and a projection system. This process took about 15 min. Second, participants had 10 min to explore BGT and review the presentation slides. Third, participants were instructed to use at least 4 different fitness functions of the tool and generate at least one map in a 20 min period. After this process, participants had to evaluate BGT in a series of items. Finally, they were thanked for their participation.

3.2.3. Instruments

In this evaluation, there were two sources of data: (1) raw data generated with the BGT (number of fitness functions used, generated maps, and AI suggestions used); and (2) a survey to obtain user perceptions focusing on the AI and the BGT in general, composed of semantic differential questions [27].

- **AI suggestions:** To evaluate the AI suggestions, a construct with five items evaluating AI fitness functions was made. Participants were asked “Do you consider that the suggestions of the AI are: useless/useful, bad/good, inadequate/adequate, not pertinent at all/totally pertinent, undesirable/desirable”. Each question was measured on a seven-point Likert-type scale (i.e., ranging from 1 (useless) to 7 (useful)). Due to high internal correlation, a single AI suggestion index was created by averaging them ($\alpha = 0.921$).
- **AI creation features:** To evaluate the AI creation features, a construct with four items was made. Participants were asked “From your point of view, how much do you consider that the tool helps you develop biomes that are: (1) more natural, (2) more complex, (3) more complete and (4) more varied” on a seven-point Likert-type scale ranging from 1 (a little) to 7 (a lot). Due to high internal correlation, a single AI creation feature index was created by averaging them ($\alpha = 0.814$).
- **Productivity:** To assess the potential impact on productivity, participants were asked “From your point of view, how much do you consider that the tool helps you develop biomes: (1) more quickly, (2) with less effort, (3) with more effort, and (4) makes the process slower”. These questions were evaluated on a seven-point Likert scale from a little (1) to a lot (7). A single productivity index was created by averaging them ($\alpha = 0.605$).
- **Satisfaction:** Three questions were made for this dimension: (1) “In general, my opinion about the tool is:” evaluated on a seven-point Likert scale from “against” to “in favor”; (2) “My level of satisfaction with the use of the tool is” evaluated on a seven-point Likert scale from “totally dissatisfied” to “totally satisfied”; and (3) “In general, I like the tool” evaluated on a seven-point Likert scale from “strongly disagree” to “strongly agree”. A single satisfaction index was created by averaging them ($\alpha = 0.804$).

4. Results

The development process produced the BGT (Available at <https://github.com/Prion6/BiomeGenerationTool>, accessed on 27 April 2023), along with usage instructions, displayed in Figure 1. BGT is one of the first mixed-initiative systems to work in a game engine such as the Unity3D IDE. The tool enables the user to edit and manage a map biome through

a system of layers and is capable of suggesting modifications through an EA, as well as generating the biome in a 3D environment. The generated results for the designer editing in Figure 1A are presented in Figure 2A,B.



Figure 2. (A) Top view of the biome generated by BGT on the Unity editor. (B) Lateral view of the generated biome in the editor. This is the end-user view, where the designer can see the generated environment in 3D.

During the processes of development and testing, the tool demonstrated potential to not only assist in the creation of map environments but, by continuing its development and offering further customization options, also in the creation of any kind of level for any kind of game genre, becoming a generic tool that can be configured to the designer's needs.

User Study Descriptive Analysis

A two-sided two-sample *t*-test was performed to compare the different indices between the experimental group and the control group. Results were analyzed with the SPSS tool with a $p < 0.05$ level of significance, and are shown in Figure 3.

- **AI suggestions:** There was a significant difference in how users perceived the AI suggestions between the experimental group ($M = 4.93$, $SD = 1.02$) and the control ($M = 3.74$, $SD = 1.63$); $t(26) = -2.26$, $p = 0.032$, meaning that participants in the experimental group had a better opinion about the AI suggestions they received, compared to those of participants in the control group about the ones they received.
- **AI Creation Features:** There was not a significant difference in how users perceived that the tool helped them develop biomes between the experimental group ($M = 5.23$, $SD = 1.28$) and control ($M = 4.20$, $SD = 1.50$); $t(26) = -1.93$, $p = 0.064$. It is important to

note that the control group M was 4.20, close to the neutral point of the measurement scale. As a complementary analysis, the difference with the middle point of the scale 4 was tested for, finding that the control group feels neutral about how the tool helped them to develop biomes ($M = 4.20$, $SD = 1.50$, $t(14) = 0.516$, $p = 0.614$). The experimental group showed a positive perception over the neutral point of the scale ($M = 5.23$, $SD = 1.28$); $t(12) = 3.44$, $p = 0.005$), which indicates that the users felt the tool helped them create more natural, complex, complete, and varied biomes.

- **Productivity:** There was a significant difference in perceived productivity, between the experimental ($M = 5.98$, $SD = 0.78$) and control groups ($M = 5.13$, $SD = 1.17$); $t(26) = -2.20$, $p = 0.037$. This means that participants in the experimental group considered that the tool helped them to be more productive than the ones in the control group considered it helped them. It is important to note that both groups' means were over four (which is the mid-point of the seven-point Likert scale). This could be related to the fact that participants were asked about the BGT in general and not only about the AI suggestions. In other words, both groups perceived better productivity, with the experimental group showing a greater effect.
- **Satisfaction:** There was a significant difference in satisfaction with the tool between the experimental ($M = 6$, $SD = 0.82$) and control groups ($M = 4.82$, $SD = 1.40$); $t(23) = -2.747$, $p = 0.011$. This means that participants in the experimental group were more satisfied with the tool than the ones in the control group were. Just as with the last item, it is important to note that the satisfaction of the participants considered the BGT in general and not only the IA suggestions.

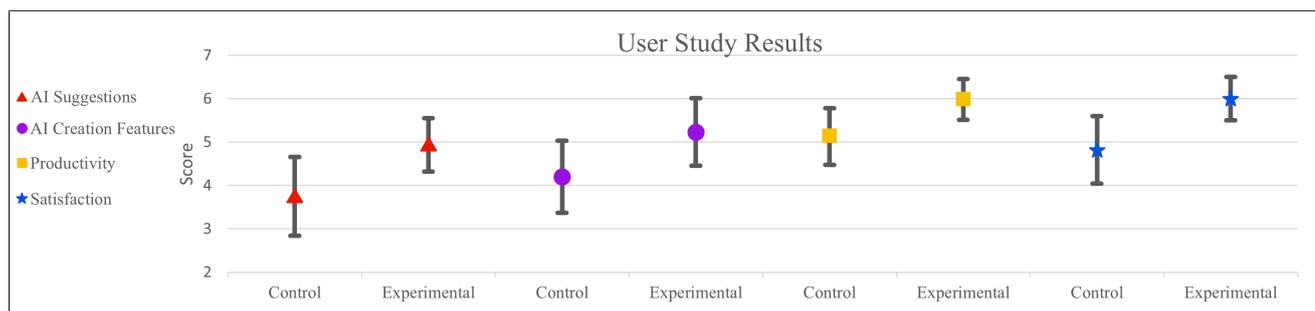


Figure 3. User study results. For each construct, the mean and 95% confidence intervals are shown for the control and experimental groups.

Finally, we compared the number of AI suggestions used by participants in each group. A Mann–Whitney U test revealed that the number of AI suggestions used was not significantly different between the experimental group ($Md = 19$, $n = 13$) and the control group ($Md = 18$, $n = 15$), $U = 78$, $p = 0.368$. This means that both groups interacted with BGT's AI suggestions during the procedure, with no statistical differences. It is important to note that the only difference between groups is the fitness functions provided by BGT. Therefore, the interaction with the tool was the same for all participants.

5. Conclusions and Future Work

In the present document, the development of a mixed-initiative procedural biome generation tool for the Unity game engine has been described. This tool uses a novel multi-layered architecture for generating AI-assisted suggestions. A user study was conducted with the goal of determining the tool's usefulness and user satisfaction.

The users' perceptions of the AI assistance on three out of four dimensions (suggestions, productivity, and satisfaction) were significantly different and greater in the experimental group than those from the control group, as hypothesized. However, results showed no significant differences between the experimental and control groups for the AI Creation Features, even though the experimental group had a favorable perception of the tool, as shown by their rating above the neutral point on the scale. This suggests that users

felt the tool was useful in creating biomes that were more natural, complex, complete, and varied. As a consequence, the user study showed that participants found the suggestions of BGT to be useful, desirable, and pertinent, among other qualities. This means that users recognized the main purpose of the tool. Participants stated that the tool helped them to develop biomes more quickly and with less effort. Additionally, user satisfaction being very high is an important overall indicator for the tool as it is a known predictor of its future use.

In conclusion, it was shown that the BGT has the potential to generate high-quality products for videogame development, and to compete with hand-crafted ones while reducing the workload and providing a high degree of control over the result. Even so, there is still work to do for the tool to be usable in an industry setting.

Further work should be addressed in three main lines:

- **Derived studies:** Evaluation of the product generated using the BGT with an end-user group. In this way, the created biomes will be able to be evaluated from the user's perspective.
- **Improvements of the tool:** Biomes in videogames do not exist in isolation, as they are part of a hierarchy of content. Other generation stages are planned to be added to BGT before and after biome creation; for example: (1) A prior stage will help create the space where the biome exists, and; (2) later stages will populate the scenario with NPCs and assist in creating quests.
- **Application in other fields:** The same concept of the BGT can be applied to other scenarios, such as creating several levels of an indoor building. These tools can be applied for the creation of virtual environments and scenes for interactive software such as virtual museums [1,28], simulators [29,30], and games [31] where both developers and end-users can benefit from the use of this kind of tool, reducing time and resources for the creation process and also obtaining enhanced levels.

Author Contributions: Conceptualization, G.K.S., S.v.B. and N.A.B.; methodology, F.B. and N.A.B.; software, G.K.S.; validation, F.B. and N.A.B.; formal analysis, F.B.; investigation, G.K.S.; resources, N.A.B.; data curation, G.K.S., F.B. and N.A.B.; writing—original draft preparation, G.K.S., F.B. and N.A.B.; writing—review and editing, F.B., S.v.B. and N.A.B.; visualization, G.K.S. and S.v.B.; supervision, N.A.B.; project administration, N.A.B.; funding acquisition, N.A.B. and S.v.B. All authors have read and agreed to the published version of the manuscript.

Funding: This research was partially funded by the National Agency for Research and Development (Agencia Nacional de Investigación y Desarrollo, ANID), Sub-directorate of Applied Research (Subdirección de Investigación Aplicada, SIA), grant number ID21I10363.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Informed consent was obtained from all subjects involved in this study.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AI	Artificial Intelligence
BGT	Biome Generation Tool
EA	Evolutionary Algorithm
EDD	Evolutionary Dungeon Designer
M	Mean
PCG	Procedural Content Generation
SD	Standard Deviation

References

1. Besoain, F.; Jago, L.; Gallardo, I. Developing a Virtual Museum: Experience from the Design and Creation Process. *Information* **2021**, *12*, 244. [CrossRef]
2. Freiknecht, J.; Effelsberg, W. A survey on the procedural generation of virtual worlds. *Multimodal Technol. Interact.* **2017**, *1*, 27. [CrossRef]
3. Barriga, N.A. A short introduction to procedural content generation algorithms for videogames. *Int. J. Artif. Intell. Tools* **2019**, *28*, 1930001. [CrossRef]
4. Togelius, J.; Shaker, N.; Nelson, M.J. Introduction. In *Procedural Content Generation in Games*; Springer: Cham, Switzerland, 2016; pp. 1–16.
5. Saini, V.; Guzdial, M. A Demonstration of Mechanic Maker: An AI for Mechanics Co-Creation. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Virtual, 19–23 October 2020.
6. Charity, M.; Khalifa, A.; Togelius, J. Baba is Y'all: Collaborative Mixed-Initiative Level Design. In Proceedings of the 2020 IEEE Conference on Games (CoG), Osaka, Japan, 24–27 August 2020; IEEE: Osaka, Japan, 2020; pp. 542–549. [CrossRef]
7. Alvarez, A.; Dahlskog, S.; Font, J.; Holmberg, J.; Nolasco, C.; Österman, A. Fostering creativity in the mixed-initiative evolutionary dungeon designer. In Proceedings of the 13th International Conference on the Foundations of Digital Games, Malmo, Sweden, 7–10 August 2018; Association for Computing Machinery: New York, NY, USA, 2018; pp. 1–8. [CrossRef]
8. Delarosa, O.; Dong, H.; Ruan, M.; Khalifa, A.; Togelius, J. Mixed-Initiative Level Design with RL Brush. In Proceedings of the Artificial Intelligence in Music, Sound, Art and Design-10th International Conference, EvoMUSART 2021, Held as Part of EvoStar 2021, Virtual, 7–9 April 2021.
9. Liapis, A.; Yannakakis, G.N.; Togelius, J. Sentient sketchbook: Computer-assisted game level authoring. In Proceedings of the 8th International Conference on the Foundations of Digital Games, Crete, Greece, 14–17 May 2013; Association for Computing Machinery: Chania, Greece, 2013.
10. Russell, S.; Norvig, P. *Artificial Intelligence, A Modern Approach*, 4th ed.; Pearson: Upper Saddle River, NJ, USA, 2022.
11. Alam, T.; Qamar, S.; Dixit, A.; Benaida, M. Genetic Algorithm Reviews, Implementations, and Applications. *arXiv* **2020**, arXiv:2007.12673.
12. Liapis, A.; Smith, G.; Shaker, N. Mixed-initiative content creation. *Proced. Content Gener. Games* **2016**, 195–214. [CrossRef]
13. Liapis, A.; Yannakakis, G.N. Boosting Computational Creativity with Human Interaction in Mixed-Initiative Co-Creation Tasks. In Proceedings of the ICCG Workshop on Computational Creativity and Games, Paris, France, 27 June 2016.
14. Yannakakis, G.N.; Liapis, A.; Alexopoulos, C. Mixed-initiative co-creativity. In Proceedings of the 9th International Conference on the Foundations of Digital Games, Liberty of the Seas, Caribbean, 3–7 April 2014.
15. Karimi, P.; Grace, K.; Maher, M.L.; Davis, N. Evaluating creativity in computational co-creative systems. *arXiv* **2018**, arXiv:1807.09886.
16. Kimbrough, S.; Koehler, G.; Lu, M.; Wood, D. On a feasible-infeasible two-population (fi-2pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch. *Eur. J. Oper. Res.* **2008**, *190*, 310–327. [CrossRef]
17. Baldwin, A.; Dahlskog, S.; Font, J.; Holmberg, J. Mixed-initiative procedural generation of dungeons using game design patterns. In Proceedings of the 2017 IEEE Conference on Computational Intelligence and Games (CIG), New York, NY, USA, 22–25 August 2017; IEEE: New York, NY, USA, 2017; pp. 25–32. [CrossRef]
18. Yannakakis, G.N.; Eladhari, M.; Lopes, P.; Knight, Y.; Liapis, A. Mixed-Initiative Procedural Content Generation. Technical Report, C2Learn Project, 2013. Available online: http://project.c2learn.eu/sites/default/files/C2Learn_D4_3_3_Mixed_Initiative_PCG_Prototype_Report_Final.pdf (accessed on 27 April 2023).
19. Abela, R.; Liapis, A.; Yannakakis, G.N. A Constructive Approach for the Generation of Underwater Environments. In Proceedings of the FDG workshop on Procedural Content Generation, Pacific Grove, CA, USA, 22 June–25 June 2015.
20. Elton-Pym, A. Principles for AI Co-Creative Game Design Assistants. In Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment, Virtual, 19–23 October 2020.
21. Dustler, M.; Bakic, P.; Petersson, H.; Timberg, P.; Tingberg, A.; Zackrisson, S. Application of the fractal Perlin noise algorithm for the generation of simulated breast tissue. In Proceedings of the Medical Imaging 2015: Physics of Medical Imaging, Orlando, FL, USA, 22–25 February 2015; SPIE: Orlando, FL, USA, 2015; Volume 9412, p. 94123. [CrossRef]
22. Lai, G.; Latham, W.; Leymarie, F.F. Towards Friendly Mixed Initiative Procedural Content Generation: Three Pillars of Industry. In Proceedings of the International Conference on the Foundations of Digital Games, Virtual, 15–18 September 2020; Association for Computing Machinery: New York, NY, USA, 2020. [CrossRef]
23. Guzdial, M.; Liao, N.; Chen, J.; Chen, S.Y.; Shah, S.; Shah, V.; Reno, J.; Smith, G.; Riedl, M.O. Friend, collaborator, student, manager: How design of an ai-driven game level editor affects creators. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems, Scotland, UK, 4–9 May 2019.
24. Oates, B.J. *Researching Information Systems and Computing*; Sage: Thousand Oaks, CA, USA, 2005.
25. Giagkiozis, I.; Fleming, P. Methods for multi-objective optimization: An analysis. *Inf. Sci.* **2015**, *293*, 338–350. [CrossRef]
26. Walton, S.; Rahat, A.; Stovold, J. Evaluating mixed-initiative procedural level design tools using a triple-blind mixed-method user study. *IEEE Trans. Games* **2021**, *14*, 413–422. [CrossRef]

27. Osgood, C.; Suci, G.; Tannenbaum, P. *The Measurement of Meaning* (No. 47); University of Illinois Press: Champaign, IL, USA, 1957.
28. Besoain, F.; González-Ortega, J.; Gallardo, I. An Evaluation of the Effects of a Virtual Museum on Users' Attitudes towards Cultural Heritage. *Appl. Sci.* **2022**, *12*, 1341. [[CrossRef](#)]
29. Cervera, M.; Grandon, N.; Rivera, M.; Besoain, F. Improving the selection of IQF raspberries in processing lines: A Virtual Reality approach for training and selecting personnel. In Proceedings of the 2018 IEEE Biennial Congress of Argentina (ARGENCON), San Miguel de Tucuman, Argentina, 6–8 June 2018; pp. 1–7. [[CrossRef](#)]
30. Jago, L.; Gallardo, I.; Besoain, F. Developing a Virtual Reality Experience with Game Elements for Tourism: Kayak Simulator. In Proceedings of the 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Valparaiso, Chile, 13–27 November 2019; pp. 1–6. [[CrossRef](#)]
31. Sepulveda, G.K.; Besoain, F.; Barriga, N.A. Exploring dynamic difficulty adjustment in videogames. In Proceedings of the 2019 IEEE CHILEAN Conference on Electrical, Electronics Engineering, Information and Communication Technologies (CHILECON), Valparaiso, Chile, 13–27 November 2019; pp. 1–6.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.