On Mixed-Initiative Content Creation for Video Games

Gorm Lai, Frederic Fol Leymarie, and William Latham

Abstract—**In this article, we present a survey of mixed-initiative methods for the creation of content for video games. We also propose a definition of what mixed initiative implies, as the term lacks a clear specification. The survey includes works not directly aimed at video games but which create content that can potentially be used in games, such as art programs utilizing mixed initiative. Furthermore, we highlight research areas that overlap wholly or partly with mixed initiative, such as casual creators, explainable artificial intelligence, or interactive evolutionary computation. We examine these and several other topics in the context of mixed initiative. Finally, we provide a catalogue of typical techniques and challenges connected with mixed initiative before considering future directions.**

Index Terms—**Co-creativity, experience-driven procedural content generation (EDPCG), interactive evolutionary computation (IEC), mixed initiative, mutant shopping, procedural content generation (PCG), procedural content generation via machine learning (PCGML), user fatigue, video games.**

I. Introduction

As SOFTWARE-DRIVEN and artificial intelligence (AI)-supported systems become more popular, there has been an increased focus on our relationship with these systems as humans and users. Mixed initiative concerns itself with the interaction between human and machine, such that either human or machine can take or give back the initiative in an interaction. As we will see in this study, in the world of video games research, mixed-initiative methods have often been combined with procedural content generation (PCG), where one seeks to automate the production of content (such as graphical or audio assets). Mixed-initiative methods offer a way of interacting and directing the underlying algorithms, for example, through a user interface (UI) or a computational agent.

In our survey, we are especially focusing on mixed-initiative applications that can produce content for video games. This includes common tools, such as level editors. However, we will also consider tools developed in other disciplines, which can impact video game content production. A good example is the "Creative Sketching Partner" (see Section VI). While some studies have focused on procedural content generation via machine learning (PCGML) [1] or computational creativity for PCG [2], our survey is the first to explicitly focus on mixed-initiative methods for games, as the number of reported works in this area has significantly increased in recent years.

As we will see in Section III, mixed initiative has its background in dialogue systems and text-based contexts, where it is relatively well defined. However, when transposed over to PCG, several authors have combined the two concepts in the sense that they do not clearly distinguish what PCG is from mixed initiative. To address this potential confusion in terms, we will narrow down the definition of mixed initiative (see Section III-D).

In Section IV, we will identify research areas that overlap wholly or in part with mixed initiative, such as casual creators, experience-driven PCG, and others. In Section V, we introduce the scope of the study, a methodology for selecting works, and which categories of content are covered. This serves to support, in Section VI, the core of this survey: that of mixed-initiative systems for digital content creation (CC) that can be used for video games. Finally, we will consider commonly used techniques (see Section VII), challenges (see Section VIII), and future perspectives (see Section IX) on mixed-initiative systems.

II. Related Recent Studies

Summerville *et al.* [1] produced a large study on the use of PCGML for video game content. In addition, they included a section on the possible use of PCGML for mixed initiative and co-creativity. They refer to Tanagra [3], [4], Sentient Sketchbook [5], [6], and Ropossum [7]–[10] as earlier examples that do not use machine learning (ML) and mention inpainting [11] as a somewhat related work that uses PCGML and which could be transposed over to level generation. Petrovskaya *et al.* [12] created a typology of applications designed for casual creators (see Section IV-A). Most of these are designed to assist the user and, thus, have mixed-initiative elements. Barreto *et al.* [2] surveyed the use of computational creativity in PCG, extending into mixed initiative by covering works, such as Tanagra (see Section VI-I10) and the Sentient Sketchbook (see Section VI-I8).

III. What is Mixed Initiative?

Carbonell [13] is the first author to mention the term mixed initiative. However, this is introduced indirectly by describing the software SCHOLAR as mixed initiative. SCHOLAR was a text-based dialogue computer system for interrogating students about their knowledge of the geography of South America.

Manuscript received 5 August 2021; revised 3 March 2022; accepted 6 April 2022. Date of publication 23 May 2022; date of current version 15 December 2022. This work was supported in part by the Engineering and Physical Sciences Research Council Centre for Doctoral Training in Intelligent Games and Game Intelligence under Grant EP/L015846/1. *(Corresponding author: Gorm Lai.)*

The authors are with the Goldsmiths, University of London, SE14 6NW London, U.K. (e-mail: glai001@gold.ac.uk; f.leymarie@gold.ac.uk; w.latham@gold.ac.uk).

Digital Object Identifier 10.1109/TG.2022.3176215

Carbonell explains how "SCHOLAR is capable of maintaining a mixed-initiative dialogue with the student, with questions asked by either side and answered by the other," and goes on to say that "in a mixed-initiative ISO system like SCHOLAR questions are asked to and by the system." Thus, according to Carbonell, mixed initiative emphasizes that both the sides can initiate a dialogue and respond. Carbonell also describes three modes of interactions with SCHOLAR: 1) a *testing* mode, where only the computer can ask questions; 2) a *Q&A* mode, where only the student can ask questions; and 3) a *mixed initiative,* where both the sides can engage and respond. Thus, for Carbonell, it is not enough that one side can ask questions and another respond. Rather, both the sides need equal capabilities to query and respond. We will return to such a definition later in this section.

Novick and Sutton [14] examine the question of what is mixed initiative by summarizing earlier uses of the term and exploring what it means to have initiative. They produce a model of initiative involving the interplay of three factors: 1) *choice of task*; 2) *choice of speaker*; and 3) *choice of outcome*. Although their work mainly focused on dialogue, we can apply their ideas to general CC tools. The choice of task 1 is not only about an overall task but also about advancing through subtasks needed in creating a particular piece of content. This choice may not even be a linear progression through subtasks, as specific changes may require either a human or a CA to go back and change earlier stages. This iteration can be because such a change is necessary or because the human or the CA realizes that the content is no longer suitable for what was initially needed.

Many mixed-initiative applications feature explicit turn taking [15]–[17]. In contrast, others have an explicit choice of a speaker (2), as a human and a computer are able to work seemingly simultaneously in real time without interruption as the computer adapts to the user's changes [3]–[6]. The choice of outcome (3) can often be split into more granular levels, as control is traded for the ease of creation. For example, Tanagra (see Section VI-I10) allows the CA to create a complete level at the touch of a button. However, the user can also directly edit the level or part of the level, resulting in a more time-consuming process while giving complete control to the human.

So far, we have only talked about the principle of mixed initiative in the context of a conversation. In the context of PCG, Liapis *et al.* [18] give a comprehensive introduction to mixed-initiative interfaces. They suggest that mixed-initiative algorithms can be put on a continuum between two extremes. At one end, the human has full control as the creator, and the computer comes up with suggestions. Gmail and most popular text messaging applications are examples of this: as the user types, the CA comes up with alternative recommendations and means of autocompleting words and sentences. At the other end of the spectrum are games like *Civilization IV* [19], where the user provides the initial input and the CA creates the entire game setup from there. In this context, the human is always the initiator of the process. The continuum between the two extremes indicates how much the CA is involved in the process: as it increases, the human's involvement correspondingly decreases.

Deterding *et al.* [20] also create a 1-D continuum with two extremes, where the creator is either entirely the CA or the human, and such that the middle parts of the spectrum are designated as mixed initiatives. Liapis and Yannakakis [21] create an identical spectrum to illustrate mixed-initiative co-creativity.

Yannakakis *et al.* [22] define mixed-initiative co-creativity by considering "both the human and the computer *proactively* making contributions to the problem solution." This definition is close to Carbonell [13]'s, where both the sides (human and CA) should be able to take the initiative and respond.

A. Content Iteration Versus Fire & Forget

Liapis *et al.* [18] use *Civilization IV* as an example. Applying Carbonell's model, we might say that the human has initiative as they input the original values. The CA responds by creating some content and takes some initiative by adding random seeds. However, the human never has a chance to respond. This is what we will identify as the "Fire & Forget" technique in Section VII. Though it has become the norm, we question the usefulness of classifying methods utilizing Fire & Forget as mixed initiative, as it will automatically define all the PCG works that take input as mixed initiative, even if they do not feature a graphical UI.

Considering again Carbonell [13], we argue that in order to allow both the humans and the CAs to take initiative and respond, mixed-initiative methods need to allow more than one iteration of what Compton [23] calls the *Grokloop*. The Grokloop is the iterative work process that Compton distills into four parts: *build a hypothesis, modify the model, evaluate the result,* and *update the result*. Participants in the creative process iterate through the Grokloop until they are happy with the content. Other authors have introduced similar concepts, such as Fails and Olsen [24] who, in the context of interactive machine learning (IML), introduced a three-stage loop: *manually classify, generate classifier,* and *evaluate classifier*.

Note that this model of a Grokloop indicates the CA has a memory of the result of the previous iteration for it to be classified as mixed initiative. For example, if we imagine a landscape generator that resets its random seeds every time it goes through the Grokloop, then it does not make sense for an agent to respond to that, as the random values will be different the next time it goes through the loop. Essentially, a Fire & Forget PCG tool does not become mixed initiative just because we give the user the chance to press *generate* a second time.

We can also safely exclude from being mixed initiative the large group of PCG visualizing tools, such as Shader Toy [25] that allows you to write and visualize the output of shaders in a web browser. This is due to the fact that the CA takes no real initiative and does not make an independent contribution to the visualization. This would be to the very end of the left scale of [20, Fig. 1]. A similar argument applies to the end of the right-hand side toward computational creativity. Works such as ANGELINA [26] by Cook and Colton, which is an automatic game design tool that is designed to work without the assistance of a human designer, do not fall into the category of mixed initiative. In this context, it can be helpful to think of this continuum between human and computer initiatives, as described independently by Deterding *et al.* [20] and Liapis and Yannakakis [21] as open intervals. In mathematics, an open interval does not include end values, while a closed interval does. Therefore, for a process to be considered as mixed initiative, it needs to be part of an open interval limited by human and computer initiatives. It cannot be exclusively of one type or the other.

B. Graphical UIs and Direct Manipulation

We find it meaningful to differentiate between the many PCG tools that have user input and possibly feature a visual preview and those that are mixed initiative. This is in contrast to, for example, Togelius *et al.* [27], who suggest that any tool featuring generative content with indirect input is defined as mixed initiative:

> "The difference between mixed-initiative PCG and level editors seems to be the directness of the editing. In a typical level editor, there is a direct, immediate and typically local connection between what the user does and how the content changes. In contrast, a (mixed-initiative) PCG system involves a non-trivial amount of computation (the "procedural" part of PCG) between the user input and the content change; the change might not be immediate, not local and not direct in the sense that it is not straightforward for the human to predict the exact changes that will come about as a result as a particular input."

While this quote shows that Togelius *et al.* [27] argue that many mixed-initiative interfaces do not feature or utilize direct manipulation, already in 1999, Horvitz [28] published a paper focusing on mixed-initiative direct manipulation techniques. Direct manipulation can also be seen in most text editing applications featuring sentence prediction.

C. Co-Creative Agents

If we envision co-creation as a collaborative effort, then it can be argued that one end goal of a mixed-initiative system is to come as close to the idea of collaborative work as possible—that constant exchange of ideas, brainstorming, and iterating, where all the involved parties act as creators, critics, and gardeners. This type of fusion of human and machine has been expected at least as far back as 1960, when Licklider [29] wrote "(hu)man-computer symbiosis is an expected development in cooperative interaction between men and electronic computers." However, Licklider did not foresee that CAs could become decision makers: "In the anticipated symbiotic partnership, men will set the goals, formulate the hypotheses, determine the criteria, and perform the evaluations. Computing machines will do the routinizable work that must be done to prepare the way for insights and decisions in technical and scientific thinking."

Lubart [30] identified four different collaborative roles for a CA: nanny, pen-pal, coach, or colleague. The most advanced role for a CA that Licklider foresaw is what Lubart identifies as the *coach*. However, Yannakakis *et al.* [22] make the step that Licklider did not consider, and define mixed-initiative co-creativity as Lubart's *colleague* role, describing a situation where both the human and the CA can take the initiative in the process of creation. In more recent work, Guzdial *et al.* [31] have taken a different approach by identifying four different ways users expected to interact with an assistive AI: friend, collaborator, student, or manager.

In 2007, Horvitz [32] provided an overview of challenges in designing mixed-initiative systems while introducing the concept of *grounding:* the process of reaching a common understanding of the project, its goals, processes, and so forth. Horvitz aims to achieve the goal of having a CA similar to Lubart's colleague [30]. For example, Horvitz discusses the need of moving past a turn-based approach, and how the goal (or "dream" as he phrases it) is to support an "elegant problem-solving dance among parties, where the nature and timing of human and machine contributions are coordinated carefully" [32].

D. Toward a Definition of "Mixed Initiative"

Dissecting the term mixed initiative, the first part—mixed—refers to a collaboration between a human and a machine, while the latter part indicates that both the parties need to be able to take initiative. The main question then becomes, *what does it mean to have initiative*? Looking at the iterative loop described earlier (see Section III-A), we see that an initiative can either be taken at the start of the loop or as a response at any iteration in the loop. As we have already excluded Fire & Forget methods from our definition, we must include a memory of the state of the system at the outcome of the previous iteration. Another point to consider is *whether a human has to be aware of the actions of a CA for it to be mixed initiative*. It is a question that deserves a longer study; therefore, we will just start the discussion here. It can be argued that the answer is context dependent, as if the agents (humans and CAs) are working together on a game or set of levels, such as a real-world team of game developers, then they do not need to know who they are collaborating with or even what exactly has been changed. Sometimes, there is a *manager* of some sort, whose job would be to facilitate communication between sets of agents. However, if a set of agents are working together on a very specific component, then changes need to be communicated, as otherwise the agents would not know if they are getting closer to their goal. All of the works presented in this article, except for perhaps Refraction (see Section VI-I6), fall into the latter category. Optimally, we would also require the reasons for any changes to be communicated to all the agents [see Section IV-C on explainable AI (XAI)]. We, therefore, propose the following generic definition.

Definition [Mixed initiative for content generation (MI4CG)]: An MI4CG consists at its basis in an iterative loop, where at least one agent (human or artificial) can take the initiative by starting the CC process and all the involved agents (human and artificial), having memory of the current state or of past states of the system, can contribute to the content and respond at least once. An informed MI4CG has the additional requirement that its agents can be made aware of the other agents' actions.

IV. Related Topics

In this section, we give an overview of concepts that can overlap with the MI4CG.

A. Casual Creators

Casual creators is a term first coined by Compton and Mateas [33] who defined it as "an interactive system that encourages the fast, confident, and pleasurable exploration of a possibility space, resulting in the creation or discovery of surprising new artifacts that bring feelings of pride, ownership, and creativity to the users that make them." This definition implies that the focus is not only on the produced artifact but more so on autotelic creation. In order to make creation pleasurable, most casual creators utilize generative and mixed-initiative techniques often trading expressiveness for accessibility. Many works directed toward making game development accessible, such as Germinate [34], covered in Section VI-H, are based on the principles of casual creation.

B. Experience-Driven Procedural Content Generation (EDPCG)

An EDPCG, introduced by Yannakakis and Togelius [35], presents the idea of adjusting the content in a game in real time according to the player's experience. As the game adjusts the content, so will the player's experience change, and the two—the game and the player—enter into a long loop of adjusting to each other. This concept is similar to the collaboration present in mixed-initiative applications for CC, where the agent and the user enter into iterative loops, working off of each other's ideas. However, there are also potential differences. EDPCG content changes are directly applied without communication between agents, such that the human user, in particular, can remain unaware of changes made.

C. Explainable AI

A feeling of frustration can arise when working with someone if we do not understand their motivations or get no explanation for their actions. This feeling can be aggravated when working with a CA, as often there is no way to communicate the frustration of not understanding why something happens. Many will have experienced the urge to "scream at the computer" at least once because it does something we do not understand at an unwanted time.

XAI is an area that focuses on opening up the often closed box of algorithms to communicate to the human user the *whys* and *hows* of what is being generated. Biran and Cotton [36] give an introduction to XAI focused toward ML. In ML applications, one often finds systems that are hard to understand; therefore, XAI in ML has become a research area of interest. They introduce the idea that a system needs to be able to *explain* the decisions and actions that are made, as well as being *interpretable* by a human. In lieu of being able to interpret a system, in the case of an opaque closed box system, we might instead fall back on being able to *justify* the actions taken. To tackle such an issue of opaqueness, Guzdial *et al.* [37] trained a PCGML system on game design patterns from *Super Mario Bros* [38] labeled by human game designers. These patterns became a means of providing meaningful communication between the human and the computer. For example, the system can be given an image of a section of a level and a label for a game design pattern to apply to that section to improve it according to the commonly agreed semantics of the label. *Danesh* by Cook *et al.* [39] is a tool for assistive design that can be used to explore the generative space of a PCG method. It works by searching for specific C# attributes that it recognizes in the code base and then, in the spirit of XAI, puts together a UI, where the user can explore variables and see a visual output of the algorithm. It even features an "autotune" method that can be used for tuning and directing the output of a PCG tool toward a set of user-defined metrics. Zhu *et al.* [40] introduce the idea of *XAI* for Designers. They first divide XAI into black-box (closed) and white-box (open) methods. The former focuses on methods that visualize or highlight information, such as a heatmap or a relationship between neurons in a neural network. The latter attempts to illustrate the inner workings of a CA, for example, by providing the graph of a decision tree. The authors define explainability through a series of axioms, such as "explanation without introspection is not explanation."

If we want a computational agent to take on some of the roles described in Section III-C, such as a *colleague*, then we face some of the explainability issues as emphasized above with ML. Why has the CA taken a particular set of actions? Is it trying to consistently direct the content in a specific direction? If yes, why? Can the human designer communicate intent to the CA to get it to change direction? All these are questions that are relevant to mixed-initiative methods and can benefit from approaches developed for XAI.

D. Interactive Evolutionary Computation (IEC)

Another related topic is IEC [41], [42]. Takagi identifies Dawkins' Biomorphs [15] as the first IEC application. Todd and Latham's evolutionary art [16] is another early example of IEC. Takagi says IEC "is simply an EC technique whose fitness function is replaced by a human user." In this way, mixed initiative and IEC overlap, as EC techniques are necessarily generational by definition. With mixed initiative, the user and the agent enter into a loop, iterating on the content together. However, mixed initiative need not use evolutionary techniques; for example, Tanagra uses constraint programming (see Section VI-I10).

E. Fluidic Games

Related to the concept of casual creators (see Section IV-A), fluidic game was first introduced by Gaudl *et al.* [43] and refers to a "mobile application which positions itself between a mobile game, a game design tool and a casual creator." We later cover their initial work *Gamaika* (see Section VI-H).

V. Scope, Methodology, and Content Categories

A. Scope

A relatively small portion of creating content for video games involves directly interacting with game making tools. Combining assets into actual video game content is usually one of the last steps in the game creation pipeline. Instead, creators spend a lot of time using art creation programs, sound design programs, IDEs, and so forth. These programs are often not specifically designed for game making. For example, both Adobe Photoshop [44] and Autodesk 3ds Max [45] are programs that are used across a diverse set of industries and not just games. Similarly, where a mixed-initiative method demonstrates an interesting technique, a related perspective has led us to include works that create art but are not directly meant for CC. The reasoning is that many works designed directly for video games do not have any functionality or support for using the content in-game either. Instead, the work is intended to demonstrate a technique that could potentially be used to create content for games. Barreto *et al.* [2] surveyed the use of computational creativity in PCG (see Section II) and also extended their study into mixed initiative by covering works, such as Tanagra (see Section VI-I10) and the Sentient Sketchbook (see Section VI-I8). Later, Deterding *et al.* [20] and Liapis and Yannakakis [21] argued that mixed-initiative and computational creativity methods exist on the same continuous spectrum (see Section III).

The scope of our survey boils down to the following question: *"Does this work demonstrate a technique that can fit our mixed-initiative definition while being potentially used to create content for video games?"* We, thus, have to put ourselves in the shoes of a game developer and evaluate if a publication or a software has such potential. We emphasize that, as the authors of this study, we have collectively more than 30 years of game development experience between us.

We have grouped the works by type of content, including code, 2-D images, 3-D models, level design, etc., resulting in a taxonomy similar to that of Hendrikx *et al.* [46].

B. Methodology

By applying our definition of mixed initiative (see Section III) to the above scope, we find that some of the works that might traditionally fall within the category of mixed initiative are instead left out.

For example, works that solely use Fire & Forget (see Section VII) do not qualify as mixed initiative in our present study. Traditionally, there is potentially a considerable overlap between mixed initiatives and conventional PCG methods. By requiring that both the human and the computer have to be able to respond to proposed outcomes, most of the traditional PCG works are left out. One such example is *Civilization IV* [19] which Liapis *et al.* [18] mention as a mixed-initiative application (albeit at the furthest end of the spectrum). By applying Carbonell's [13] original definition that both the sides need to be able to take the initiative and respond, we see that for a game like *Civilization IV*, the computer creates (takes the initiative) and responds to the human player's original request and input. Still, there is no room for the human to respond to what the computer has made. In effect, we are not able to continue in the Grokloop [23], and thus, Civilization and other Fire & Forget methods fall outside our definition of mixed initiative.

Similarly, many assistive applications (see Section III), where the computer is unable to take its own initiative, such as Shader Toy [25], will not qualify as mixed-initiative systems.

C. Content Categories

In the following, we survey works from a broad set of categories that could be used for assets for video games: 2-D and 3-D models, 2-D art, scripts, animation, level design, and more. A lot of recent mixed-initiative works for video games focus on level and game designs. However, based on our past professional experience, we have done our best to include as many categories of assets as possible. At times, this meant including works primarily directed toward computational creativity, but which do demonstrate important mixed-initiative principles and which outputs could conceivably be used for games if the right plugins were added. This is, for example, the case with some of the works described in the categories Early Works, 2-D Images, and even Game Design (Gamaika [47]).

VI. Survey

This survey covers a number of different important asset types. However, as we have limited space, we are unable to cover all the types used in the games industry. We have, thus, aimed at providing a comprehensive and diverse selection of works.

A. Early Works

Mixed-initiative systems have existed before the term became commonplace. For example, in 1991, the UI described by Haggerty [84] based on a program developed by Todd and Latham [16] presents the user with nine possible phenotypes to choose from and is using a mixed-initiative approach such that the further development and evolution of the forms happen in collaboration between a CA and the designer. Even earlier, in The Blind Watchmaker, Dawkins [15] presented a similar interface for creating his *biomorphs*. Compton and Mateas [33] called this way of choosing between forms "mutant shopping," while Latham and Todd called it "gardening." We will discuss these terms more in-depth in Section VII. More works, such as PicBreeder [17], also use mutant shopping.

Carbonell [13] is the first author to mention the term mixed initiative. However, the earliest work using the term that is relevant for this review is by Horvitz [28] who lists 12 principles for direct manipulation. Horvitz aims to combine two seemingly opposing schools of thought, both focused on improving the user experience. The first wanted to promote automation by learning user behavior, while the second focused on improving the affordances of direct manipulation UIs. Horvitz's insight was to bridge these two ideas by arguing for introducing a CA that could help the user perform desired actions.

Doherty *et al.* [85] focus on the use of online mixed initiative in video games. While briefly touching upon general ideas, such as improving interfaces for strategy games, they quickly focus on creating a mixed-initiative-based framework for open-world character-based games. They argue that such a framework needs five major components: "a system for knowledge representation, a system for mixed initiative, a group of CAs that can take advantage of the represented knowledge, a method for translating those interactions into human-readable text, and an interface that permits the user to communicate through interactions without any direct knowledge of their structure." Unfortunately, this article only covers knowledge representation and a system for mixed initiative. The former focuses on different types of knowledge representation and the idea of *sensory honesty* [86], which signifies an effort to make a character's perception realistic. The latter focuses on the importance of multiagent interaction, including CA to CA, and the related challenges.

TABLE I
OVERVIEW OF SURVEYED WORKS AND THE RELATED TOPICS AND MAIN TECHNIQUES USED

Title	**Reference**	**Mixed-Initiative Techniques / Related Topics**
Early Works		
SCHOLAR	Carbonell [13]	
Biomorphs	Dawkins [15]	Mutant Shopping (see section VII), IEC (see section IV-D)
Mutator	Todd and Latham [16]	Mutant Shopping (see section VII), IEC (see section IV-D)
2D Images		
	Karth and Smith [48]	
	Karimi *et al.* [49]	
Drawing Apprentice	Davis *et al.* [50]	
Genetic Images	Sims [51]	Mutant Shopping (see section VII), IEC (see section IV-D)
PicBreeder	Secretan *et al.* [17]	Mutant Shopping (see section VII)
	Jaksa and Takagi [52]	IEC (see section IV-D), Grading (see section VII)
2D Models		
	Liapis *et al.* [53]–[55]	Mutant Shopping (see section VII)
3D Models		
GenTree	Mazza and Congdon [56], [57]	IEC (see section IV-D), Grading (see section VII)
	Machwe and Parmee [58]	IEC (see section IV-D), Grading (see section VII)
Subversion City Generator	Martin *et al.* [59]	IEC (see section IV-D)
Animation		
bRigNet	Acampora [60]	
Code & Scripting		
Copilot	Github [61]	Mutant Shopping (see section VII)
Squeezer	Johansen *et al.* [62]	
Dialogue, Narrative & Stories		
	Takahashi *et al.* [63]	EDPCG (see section IV-B)
Mimisbrunnur	Stenisson and Thue [64]	
Writing Buddy	Samuel *et al.* [65]	
Game Design		
AGE	Craveirinha and Roque [66]	
CADI	Mobramaein *et al.* [67]	Fluidic design techniques (see section IV-E)
CICERO	Machado *et al.* [68]	Mutant Shopping (see section VII)
Galactic Arms Race	Hastings *et al.* [69]	EDPCG (see section IV-B), Promote often used content (section VII)
Gamaika	Nelson *et al.* [47]	Fluidic Game (see section IV-E)
Germinate	Kreminski *et al.* [34]	Casual creator (see section IV-A)
Level Design		
	Liapis *et al.* [70]	Rank-Based Interactive Evolution (seetion VII)
	Schrum *et al.* [71]	Mutant Shopping (see section VII), Overriding the CA (see section VII)
Anhinga	Sturtevant *et al.* [72]	
Baba Is Y'all	Charity *et al.* [73]	
Evolutionary Dungeon Designer	Dahlskog *et al.* [74], Baldwin *et al.* [75], [76], Alvarez *et al.* [77]–[79]	Mutant Shopping (see section VII), Overriding the CA (see section VII)
Morai Maker	Guzdial *et al.* [80], Hoyt *et al.* [81]	
Refraction	Butler *et al.* [82]	
Ropossum	Shaker *et al.* [7]–[10]	
Sentient Sketchbook	Liapis *et al.* [5], [6]	Mutant Shopping (see section VII), Overriding the CA (see section VII)
SuSketch	Migkotzidis and Liapis [83]	
Tanagra	Smith *et al.* [3], [4]	Overriding the CA (see section VII)

B. 2-D Images

Co-creative interfaces for sketching and drawing 2-D images is an active area of research. We present a selection of works.

Karth and Smith [48] introduce a method that is directly generalizable to any systems using wave function collapse [87] (WFC). Their working example is based on the creation of images of 2-D flowers, which is a sample taken from the WFC repository. The main point addressed is that it is usually very expensive (in needed resources) and time consuming to generate the training data for ML-based methods. It often proves more efficient to do the work manually. This argument is similar to the one made for the design pillar, "Respect the Creative Process," by Lai *et al.* [88]. The authors show how a constraint solver for WFC can be iteratively guided by a human, using a small number of positive and negative training examples, where the user feeds a set of examples to the CA, who responds by giving an output.

Karimi *et al.* [49] developed a co-creative sketching system, focused on novelty, to assist the user in creating new ideas and "generating conceptual shifts." When the user draws a sketch, the CA will respond with an alternative sketch as inspiration for the next iteration. The system is based around two neural network architectures: visual- and conceptual-similarity modules. Using a convolutional neural network long short-term memory (LSTM), the former module takes the user's drawing and extracts features and matches with the top 20 resulting categories from an image database. The last module extracts descriptive words—using the simpler word2vec [89]—and then, by applying a measure of both visual and conceptual similarity, suggests conceptual shift candidates. Both the modules can be configured by how large a conceptual shift is desired—low, medium, or high. Finally, a high novelty mode is demonstrated.

Drawing Apprentice is a mixed-initiative web app for collaborative free-form drawing, by Davis *et al.* [50], where a user and a CA take turns working on a drawing. The user can set the agent's creativity (from 1 to 100) as well as up- and down-vote its creations. Based on a deep neural network, the CA is represented via a virtual pen as an avatar, a choice based on feedback from users who preferred the CA being personified. This CA is collaborative in the sense that it mimics or is inspired by what the user did on their turn. Two different user studies were conducted: one with a CA and one in a Wizard of Oz style, where a human pretended to be the CA. From these studies, the authors obtained several requirements for a CA to be participatory. Noticeably, users tended to get annoyed if the CA drew on top of their lines, and there was a need for a sense of memory in case the human user went back and amended earlier parts of the drawing. Users also expected the system to react immediately to the down-vote button and expressed they thought it had little or no effect.

Together with the pioneering work "Genetic Images" by Sims [51], PicBreeder [17] has become one of the most famous genetic art programs based on mutant shopping (see Section VII). While many IEC-based works, such as by Sims [51], were based on genetic programming, PicBreeder uses compositional pattern producing networks [90] based on NeuroEvolution of augmenting topologies [91] (CPPN-NEAT), where the network takes as input a 2-D pixel position as well as the distance to the center of the image and outputs a color. The authors conclude that CPPNs are suitable for creating spatial structures as they implicitly encode symmetry and repetition. Another original feature is that it is a web application; therefore, online visitors can continuously evolve the stored images. In effect, there is only one creator, the CA, and many gardeners (see Section VII) who curate together in an asynchronous fashion.

Jaksa and Takagi [52] developed an IEC-based method for image processing. They created filters to adjust image brightness and contrast in two ways: based on an EC or an IEC. For the EC version, the user adjusts the distributions of intensities in a histogram before starting the algorithm, while in the IEC case, the user numerically grades candidate images. The authors find that the IEC version gives better image quality. To counter user fatigue (see Section VIII) in the IEC case, the authors introduce a hybrid version of the two methods.

C. 2-D Models

Inspired by Galactic Arms Race (GAR) (see Section VI-H), Liapis *et al.* [53]–[55] worked on evolving the shapes of 2-D spaceships using a blend of mixed-initiative and experience-driven PCG techniques. They focused on evolving the shapes of the spaceships instead of the movement and color of the particles of the weapon in GAR. However, just like in GAR, they used the CPPN-NEAT algorithm to generate content [90], [91]. The networks take in a circle of 64 equidistant points as inputs and output a vector list of 2-D vertices outlining the spaceship. Each vertex has a value indicating if it has a thruster or weapon attached. The first work in the series is not interactive [54], but it lays the basis for the follow-ups [53], [55]. The authors use the feasible–infeasible two-population genetic algorithm (FI-2Pop GA) [92] to evolve the population and assure diversity.

Seven different tests, such as variations on speed to reach goals, collision avoidance, combat, and chasing a target, are used to measure the spaceship's fitness. The authors successfully evolve unconventional looking but functional spaceships for use in games. In the next work [53], Liapis *et al.* replace game-engine-specific tests with aesthetic tests based on symmetry, distribution of weight, hull direction, as well as containment within a triangle. A user study combined two automatic offline tests, where one selected their favorite spaceship among all candidates. New instances were then generated for five generations based on a fitness score derived from the selected candidate as well as the discarded ones. This approach was later called choice-based interactive evolution (CIE) [55] (see Section VII).

The final work in this series [55] incorporated a functional feasibility test together with an aesthetic evaluation. The initial test rejects spaceships hulls that would not work at all or work poorly in the game engine. The second test improves the aesthetic appeal in comparison to the previous work in the series. A combination of seven different aesthetic fitness measures is used: symmetry across the x-axis, containment within a triangle, simplicity, jaggedness, and three different weight distribution metrics. Two more different offline experiments were also performed, testing the optimization of a single aesthetic heuristic versus multiple heuristics in various combinations. CPPN-NEAT was also compared against four other neural-network-based constraint optimization algorithms. CPPN-NEAT consistently converges faster than its competitors. Unsurprisingly, incorporating multiple heuristics makes their algorithm converge slower than when only taking one heuristic into account.

D. 3-D Models

Built upon the Genesis system [93], Mazza and Congdon [56] created GenTree, a system for generating 3-D polygonal trees using genetic algorithms and mixed initiative. The genotype of each tree is encoded in a bitstring, and the fitness of each tree is provided by the human user, using a rating-based approach (see Section VII). In later work, the user can also interactively adjust the parameters [57].

Machwe and Parmee [58] employed mixed-initiative methods to develop 3-D designs of public seating environments. The evolutionary algorithm uses a weighted fitness function consisting of five different parts: 1) engineering fitness based on structural integrity; 2) rule-based aesthetic fitness based on the uniformity of seating elements; 3) a cantilever deflection analysis; 4) minimization of materials usage; and 5) a user-assigned grading of designs from low to best (see Section VII). Evolved seating arrangements are presented, which, it is claimed, can only be obtained due to the human factor.

Martin *et al.* [59] made an interactive tool for generating 3-D cities for the game Subversion [94]. It allows the user to navigate a 3-D city, where each building, described via a textual markup language, represents an individual in a population. Between evolutionary runs, the user can select buildings of interest and change their parameters impacting crossover and mutation. Different methods for performing such crossover and mutation were tested, and the least successful one proved to be choices made at random.

E. Animation

BRigNet [60], a plugin for Blender, is used for the automatic creation of animation skeletons for 3-D characters. Based on RigNet by Xu *et al.* [95], it adds an integration into the well-known open-source Blender game engine, together with a GUI, such that parameters can more easily be adjusted and passed on to the underlying ML algorithm.

F. Code and Scripting

When the first author of this survey started working in the video games industry, it was a surprise to discover code being marked as a class of assets. Intuitively, one would have thought that code is only used to create a game, while assets are used in potentially many games. However, this turns out to be true only for some code, as most modern games use large amounts of scripted code in a textual or visual format. Such scripting code can then be treated as a class of assets.

Many tools already exist for assistive coding, such as Visual Studio's Intellisense or Unreal Engine's blueprints. However, while helpful, nearly all of these tools do not possess their own initiative, and thus, they do not correspond to our definition of mixed initiative (see Section III). One work that falls inside our category is Copilot [61], which is based on the OpenAI Codex by Chen *et al.* [96]. Copilot can automatically create a function based on the user's comments or based on the function name. Copilot also allows the user to choose between different suggested solutions using mutant shopping (see Section VII). Squeezer by Johansen *et al.* [62], a tool for adding Juice [97] or Game Feel [98] to a game, is another example of mixed initiative. It adds effects to the game by hooking into the game's code in Unity. It also has the capability to organize and schedule these effects into a more complex sequence.

G. Dialogue, Narratives, and Stories

Dialogue systems in games are one of those cases where nonplayer characters (NPCs) often have more decision power than the player. NPCs often decide what the conversation is about, and at times even when the conversation ends. Existing in a domain overlapping both EDPCG and mixed initiative, Takahashi *et al.* [63] created a system with an adaptive mixed-initiative dialogue system, where both the NPC and the player can pick *leading* as well as *nonleading* choices.

Mimisbrunnur is a mixed-initiative tool designed by Stefnisson and Thue [64] for creating story outlines. It has three subeditors used for setting an initial state, designing actions, and fixing a goal. The initial state editor is where the characters or entities and their details can be defined. The action editor lets the designer define actions, which have preconditions and an effect. An example of this, given by the authors, is the action *Steal*, which would have the preconditions "Robin has gold" and "John does not have gold." The effect would be to exchange the conditions of Robin and John and add that "John stole gold from Robin." The goal editor is used for combining states and actions in a way the designer would like the end state to be. The mixed-initiative mainly works via a *suggestion generator* that can propose actions based on the initial state and goal editors.

Writing Buddy by Samuel *et al.* [65] is a system for player-designed stories. It first assigns the user a goal and then lets them create beats or state changes. An example of a beat is a character that stops being hungry. A story is basically a series of beats that continue until the goal has been reached. Besides beats, the player also has to create the action that causes the state change in beat. Two technologies, PlaySpec and Ensemble [99], help put the story together. While PlaySpec is a technical evaluation of the beats, existing preconditions, actions, and so forth, Ensemble is what the authors call a *social physics system* and can assess characters' volition and, therefore, if a given action and change of state makes sense.

H. Game Design

The Authorial Game Evolution (*AGE*) [66] is a mixed-initiative application for producing variations on a game design. AGE consists of three parts: a Base Game, Design Goals, and Game Variations. The Base Game is a prototype with variables left exposed so that the designer can test out different Game Variations via the AGE interface. The Design Goals is a set of measures used to calculate a score to measure the success of a Game Variation. AGE features a genetic algorithm to optimize the game design toward the Design Goals. Through iterative user tests, with a single user who was an experienced game designer, they found that the user/designer solely used the system to explore their designs and never allowed the system to apply its algorithm. Even so, the user/designer enjoyed the system. The authors concluded that systems such as AGE could offer both exploratory and optimization stages.

CADI is a conversational mixed-initiative application by Mobramaein *et al.* [67] for generating customized Pong [100] levels. Using speech recognition, their implementation extracts compound features from a command spoken by the user. Examples given by the authors include "make the arena round" and "make the game look cold," which, respectively, picks out the compound features *round* and *cold*. In the first instance, this results in the gameplay being made circular, while in the second case, a bluish palette is applied and particles added.

CICERO by Machado *et al.* [68] is a mixed-initiative interface for creating any game that can be built with the General Video Game AI (GVGAI) [101] framework, which itself depends on the video game description language (VGDL) [102], [103] used for describing complete games by defining sprites, level, interactions, a win condition, this in a concise manner. CICERO provides three features: a game mechanics recommender, rules statistics, and level visualization. Sprites in VGDL include a visual image and functionality. The mechanics recommender works by comparing sprites and interactions from the VGDL library with interactions and sprites used in the game being built. It ranks these through a metric and presents the best and the least matching candidates to the user; the latter is presented to encourage diversity in game design. Game rules statistics highlights rules and sprites by playing the game with a GVGAI agent multiple times. A heat map is built in a similar manner with a GVGAI agent.

GAR [69] is a 2-D "shoot'em up" game by Hastings *et al.* where the content adapts to the user's play style. Specifically, particle-based weapons are constantly evolving based on what the player uses. It is not the graphical style of the weapon that evolves; instead, it is the ammunition, the color, and the movement pattern of the emitted particles. The particles are evolved using the CPPN-NEAT [90], [91] method. The networks take a 2-D position, a distance to the original firing position, and a bias as an argument and output a 2-D velocity and 3-D color variable. The fitness of a weapon is based on how often it is used, and it degrades over time if not used. Weapons are selected as parents for the NEAT [91] algorithm, either because of their fitness or because they belong to a pre-evolved spawning (diversity) pool of content curated by the developers. The spawning pool helps to ensure diversity in the population. The authors call their evolving algorithm "cgNEAT" (for content generation NEAT), which consists of the following rules: 1) content is evolved through CPPN-NEAT; 2) content needs to be used not only picked up to be eligible as a parent; 3) fitness is based on how often a piece of content is used; 4) the chance to be chosen as a parent is proportional to the fitness; and 5) diversity is assured through a spawning pool. The fitness of a weapon depends on how much it is used. *Gamaika* is a fluidic game (see Section IV-E) technology for designing 2-D games, built by Nelson *et al.* [47]. It parameterizes a game via 284 features. *Cillr* is their front end built on top of *Gamaika*, exposing its capabilities. Using Cillr, which has a tool for customising sprites, levels, and so forth, two applications were built: *Wevva* and *No Second Chance* that each explore a subspace of *Gamaika* parameters. By letting the user tweak rules and tokens, these applications allow the user to build specific games. Wevva also has a customisable co-AI player and a design screen for randomizing games.

Germinate [34] is an intent-focused tool for casual creators (see Section IV-A), built on top of a game creation tool called *Gemini* [104]. In this setup, Germinate provides the front end and Gemini the back end. Instead of wrestling with technical details, Germinate presents a minimalistic UI, where the designer can express their intent at a high level. A game consists of entities, resources, relationships, and triggers. Entities correlate with sprites that can move around and, in some cases, be controlled by the player. Resources are quantifiable and are shown via energy or resource bars. Relationships exist between a resource and a resource, a resource and an entity, or two entities. An example of a relationship could be *goblin (an entity) consumes happiness (a resource)*. Finally, triggers listen for specific events, and when such an event takes place, something happens. For example, a mouse click (an event) moves a player unit (an entity). After setting up these four types, the designer can generate games that match. The authors ran a preliminary study with four expert users who found the tool at least "somewhat fun to use," a key criterion for a casual creation tool.

I. Level Design

Mixed-initiative tools for level design is a popular genre in research with many widely cited publications. Some papers focus on making tools for specific games, while others focus on general level design tools for groups of games such as games with dungeons, strategy games, and 2-D platformers.

1) Anhinga: Using exhaustive PCG (EPCG), Sturtevant *et al.* [72] built a mixed-initiative-level editor, called Anhinga, to create and solve *Snakebird* [105] levels. The authors argue that since *Snakebird* is a complex puzzle game, it is too easy to make a level unplayable with a random change and so chose to use EPCG and avoided other more common techniques such as ML. The editor allows the user to do manual editing, play the game, use automatic placement of designer chosen gameplay tokens, and calculate the shortest solution path. The EPCG algorithms consists of a generator and an evaluator. The evaluator can build longest and shortest path solutions for all the possible combinations of generated tiles and positions.

2) Baba Is Y'all: Charity *et al.* [73] wrote a mixed-initiative-based tool to help design levels for Baba Is You [106], a 2-D top-down puzzle game where one modifies the rules as part of the core gameplay. The authors' tool is a web app and consists of four main modules: game, editor, generator, and map module. The first module is used for simulating the rules of the game, enabling levels to be played either by a human or a CA. The CA uses a tree search algorithm based on the weighted city-block distance between three main tokens in the game. The editor is for humans only, while the generator module is where the CA edits levels. The fitness measure is based on a version of the Kullback–Leibler divergence algorithm [107]. The map module can be thought of as an archive where levels are stored, with the map position being calculated based on a variant of the MAP-Elites algorithm [108].

3) Evolutionary Dungeon Designer (EDD): Developed through a series of works, the EDD is a mixed-initiative application for creating procedurally generated dungeons. The first work in the series by Dahlskog and Togelius [74] introduces design patterns—micro, meso, and macro—for the PCG of dungeons. These game design patterns were distilled from the design of the dungeons in 91 games. Baldwin *et al.* [75] built the first version of the EDD using a search-based algorithm based upon the FI-2Pop GA [92]. The user gives inputs through a parameter-based UI. Baldwin *et al.* [76] expanded the EDD with support for autodetecting mesa patterns and highlighting these to the user. Using the EDD, the designer can customize a dungeon room and replace the current design with one generated by the search algorithm. Surveyed users' suggestions were implemented by Alvarez *et al.* [77]. They added three improvements to the EDD: the ability to lock down a part of the map and fitness scores measuring symmetry and similarity. The symmetry fitness works by measuring the similarity in micropatterns between the authored solution and the ones suggested by the search-based algorithm. The lockdown functionality required the authors to go from a direct gene encoding to an indirect tree-structure representation of the genome. Later, Alvarez *et al.* [78] added further improvements such as the ability to design entire dungeons, not only rooms, and a view that provides a coherent insight into how the different rooms are connected. Finally, Alvarez *et al.* [79] expanded the EDD with an implementation of MAP-Elites [108], which provides the designer with an interactive overview of alternative layouts.

4) Latent Variable Evolution: Schrum *et al.* [71] added a graphical tool on top of a latent variable evolution algorithm to explore *Super Mario Bros* and *Zelda levels*. Designers can combine manually exploring the space using sliders with mutant shopping (see Section VII). The surveyed participants generally preferred manually exploring the space, though overall preferred being able to combine the two techniques.

5) Morai Maker: Guzdial *et al.* [80] designed *Morai Maker*, a general mixed-initiative tool for grid-based 2-D platformer *Super Mario Bros* [38] inspired games, where co-creative agents can be plugged in and tested. Their level editing interface is turn-based between human users and CAs. To minimize user frustrations, the agent is restricted to making additions only, whereas humans can also delete elements from the level. In a follow-up work, Guzdial *et al.* [109] survey Morai Maker using three different types of CAs based on Markov chains, Bayes Nets, and LSTM recurrent neural networks. They conclude that there is no significant difference in users' experience with the agents and argue that instead of adapting existing PCGML methods for co-creative mixed-initiative works, a new method is needed as the levels created do not resemble any existing Super Mario Bros levels. They then introduce a *semi-Markov decision process* with concurrent actions and show that this method learns faster than the previous three CAs. Hoyt *et al.* [81] added A*-based [110] reachability and survivability checks to *Morai Maker*. The reachability check uses A* to verify if a possible playable path exists between two points on the map and visually highlight one of those paths. Survivability is a measure of the difficulty of that path, as indicated by 100 CAs trying to play it through successfully.

6) Refraction: All the other tools in this section focus on mixed-initiative techniques for designing a single level and do not assist in configuring the overall level progression for a game, nor do they provide an easy way of readjusting difficulties and content simultaneously across all levels. Butler *et al.* [82] set out to remedy this by creating a mixed-initiative tool that can be used for designing a progression system across a group of levels. To this end, the authors use the tool for creating levels for *Refraction*, a puzzle game designed to teach fractional math using constraint solving. They describe four different types of constraints: prerequisites, corequisites, concept count, and concept introduction rate. *Prerequisites* is a constraint describing if a concept A must be introduced in a different level before concept B, while *corequisites* tells us if B is a corequisite of A in a level. For example, a 2-D platformer might require springs to be present if there are floating platforms. *Concept count* measures the complexity or difficulty of a level by showing the target number of concepts, while *concept introduction rate* describes the number of concepts that should have been introduced at that point. The designer can get an overview of prerequisites and corequisites by looking at a 2-D matrix presented by the tool, showing to what degree those types of constraints are upheld. The designer uses a different tool with a spline editor to indicate how many concepts should be introduced per stage.

7) Ropossum: Created by Shaker *et al.*, Ropossum [7]–[10] is a mixed-initiative tool for creating and solving Cut the Rope [111] levels. A context-free grammar is used for describing the levels. The first work [9] describes two ways of generating levels. In the first method, levels evolve according to a set of fitness measures with a focus on the optimal placement of game elements, making it *likely* that the levels are playable. In the second method, an action-based method is combined with physical simulation. Examples of actions are *void*, *press air cushion*, and *cut the rope*. In the next work [8], the limited action-based method is extended with a reasoning component that uses physics simulation to focus on actions that make sense in a given context.

8) Sentient Sketchbook: This work by Liapis *et al.* [5], [6] is a mixed-initiative tool for creating top-down strategy games, mainly based on resource management and conquering the enemy base. The user can edit a high-level map sketch, place bases, resources, impassable tiles, etc. The user is editing the map, while the co-creative CA is simultaneously coming up with alternative map suggestions and judging the current map via six separate fitness functions, which include measures for how many resources are close to the base, the safety of the surrounding area, and how difficult the enemy base is to reach. The CA will also constantly test if the map is playable. The alternate map suggestions are generated via a FI-2Pop GA [92]. In the initial work [5], surveyed users said it was hard to see why the agent gave the suggestions that it did, so in a later work [6], Liapis *et al.* made an improvement where the CA learns the style of the human user in two different ways: 1) by capturing the fitness weights in the map suggestions the user picks and using those weights for generating new suggestions that the user can choose as a basis for further edits and 2) by learning from what the user is currently doing and using that to adapt the weights of the suggestions. To avoid overconvergence, the new version shows six new weighted suggestions and six other suggestions generated through a novelty search algorithm.

9) SuSketch: This mixed-initiative tool by Migkotzidis and Liapis [83] is for creating levels for a first-person shooter. Besides the basic map editor, the tool provides several features, categorized via three different tabs: map, predictions, and suggestions. The map tab calculates other metrics, such as the number of various tiles and the distance between player bases and tile types. It also checks whether the level is well formed. The predictions tab helps the designer balance the level by building a death heat map, calculating who will dominate, how long the match will last, and so on. SuSketch introduces two novel features: a dramatic arch calculation (how the kill ratio develops) and the combat pace (the total kills in the last 2 min). The suggestions tab highlights ideas for power-up placements and suggested classes for the level match being designed. Most of these features build on a surrogate model, which, in turn, keeps calculation times down. A user study with 16 participants indicated that the system was easy to use. However, some of the features, including the dramatic arc and combat pacing graphs, were not well understood, while the balancing suggestions were not used much.

10) Tanagra: Smith *et al.* produced a series of works on *Tanagra* [3], [4], which is a mixed-initiative tool for creating and playtesting side-scrolling 2-D platform games. Tanagra allows the user to design parts of a level and then have the co-creative AI fill in the rest. Using a constraint solver, the tool guarantees that the autogenerated part of a level is always playable. For example, if the user manually edits part of a level, the surrounding regions are automatically regenerated to satisfy certain constraints. Tanagra uses a leniency parameter to calculate a level's complexity, giving more points to platforms and stompers. Except for the enemies, none of the level elements in Tanagra moves during the game. For example, there are no moving platforms, teleporters, or triggers that can change the level. The editor allows the designer to modify the pacing of existing parts of a level, for example, by adding or removing jumps to a jumping section of the level.

VII. Common Mixed-Initiative Techniques

Having surveyed and summarized several published works, we can now start to identify similarities and differences in the techniques used. Previously, Compton and Mateas [33] identified a number of design patterns appropriate for casual creators (see Section IV-A). Many of these patterns, such as *no blank canvases*, *entertaining evaluations*, and *instant feedback*, relate to autotelic creation and sharing among users. However, as we will see in the next section, the pattern Compton and Mateas identify as mutant shopping is also used in general mixed-initiative applications.

CIE, gardening, and mutant shopping: These are similar to an IEC technique (see Section IV-D), where the program presents several content candidates to the user who then selects the candidate they like best. The program evolves a new generation based on that choice, which is again presented to the user, and so forth. In a review of Sims' Genetic Images [51] in *Wired Magazine* [112], Kelly describes the interplay between algorithm and human as "a joint product of breeding machine and human gardener." Such thinking had already been formulated by Todd and Latham [16] a couple of years earlier, when they said that "evolutionism further splits the traditional role of the artist into two, [that of] the creator and the gardener." In this case, the algorithm is the creator and the human the *gardener*. In 1986, Dawkins [15] used a similar technique for developing biomorphs. Liapis *et al.* [55] call this approach CIE. The term *mutant shopping* was first coined circa 2015 by Compton and Mateas [33]. Liapis *et al.* [113] speculate on the limitations of such a technique, using as case study the Sentient Sketchbook (see Section VI-I8).

Grading: An alternative technique closely related to mutant shopping is *Grading*, where instead of choosing one or more favorites, the user grades the content that the CA presents. Grading is more time consuming than simple mutant shopping and prone to user fatigue (see Section VIII), as the user has to consider grading each piece of content. Jaksa and Takagi [52] attempt to overcome this user fatigue by adding automatic grading of content. Modifications are then only needed where the user disagrees with the CA-based evaluation.

Rank-based interactive evolution (RIE): In a work on PCG for strategy games, Liapis *et al.* [70] introduce RIE, which is closely related to grading. Results are ranked relative to each other without giving an absolute score. The authors compare a number of IEC selection strategies by defining ten different fitness functions measuring "gameplay and aesthetic features." They then create a model of real users' preferences in strategy games to construct *simulated users*, who, in turn, are used for testing out the IEC selection strategies, including RIE, CIE, and a bog-standard rating-based IE where eight evolved candidates are given the weights $\frac{1}{8}$, $\frac{2}{8}$,..., $\frac{8}{8}$, respectively. The authors report that RIE converges faster than CIE in nearly all the tests.

Overriding the CA: There are cases where the user is satisfied with a subpart of a piece of content and might want to change other parts independently. In such cases, a user will want the ability to override the CA in some form, either by locking down and/or by directly editing subparts. The EDD (see Section VI-I3) takes both approaches, allowing the user to not only manually edit the map but at later iterations also lets the user mark parts of the map that will be locked down and not changed by further generational suggestions created by the CA. Along a similar vein, Tanagra (see Section VI-I10) lets the user lockdown parts of the map. As the designer edits part of the locked level, the CA will adjust the surrounding area, so the level is always playable. The Sentient Sketchbook (see Section VI-I8) lets the human user do manual edits.

Promote often used content: This is an approach taken by GAR (see Section VI-H) where the usefulness or popularity of a piece of content can be measured by how often it is used.

Fire & Forget: Fire & Forget falls outside of our definition of mixed initiative and is listed here for the sake of completeness (see Section III). The user gives the initial parameters, and the CA creates or adjusts the content from there on, with no possibility of the user entering into a co-creative loop with the agent. The way the player sets up parameters before beginning a game of *Civilization IV* [19] is an example of Fire & Forget. Similarly, in a work by Jaksa and Takagi [52], the user sets up an evaluation function that the EC-based algorithm can compare against as it iterates on the image filters. Again, the user has no way of interacting with the agent until the process has completed.

VIII. Challenges

Many mixed-initiative systems show similar weaknesses. Some of these are fundamentally UX issues but are nonetheless worth mentioning as they are consistent across many different applications. Some of these issues have also been highlighted elsewhere [33], [88].

Lack of Control: While allowing the rapid generation of procedural content, some mixed-initiative systems fall short in giving designers enough precision to get out their vision [33], [88]. The standard technique in several works to remedy that lack of control is to allow the user to disable the CA and let the designer directly edit the content.

Interactivity: Some mixed-initiative tools use techniques that require significant amounts of computation time, such as the simulation of different outcomes [73], [75], paths for agents [81] or time to train an artificial neural network. SuSketch (see Section VI-I9) actively seeks to overcome this issue by using a surrogate model built from previous simulations.

Fatigue: User fatigue is a common issue when using interfaces that encourage iteration, such as those featuring mutant shopping (see Section VII). Repetitively choosing candidates for an exploratory activity can be fine. Still, if the user is trying to exploit the algorithm to go in a certain direction, then repetition and lack of control can lead the user to lose patience. Several authors mention user fatigue as an issue [18], [41], [42], [52], [88], [114]. Takagi [41], [42] writes of human fatigue in his work on IEC (see Section IV-D), where he suggests that 10–20 iterations are the limits of repetition for most users. To remedy such fatigue, he further suggests to improve [41]: the input and display interfaces, as well as the performance of the evolutionary computation. Improving the input interface is a typical UX issue. For example, consider reducing a large input scale to something more meaningful for a human, quoting Takagi: "we cannot exactly distinguish the difference between 62 and 63 points in 100 levels rating," which can be the source of "psychological fatigue." Improving the display interface is also a UX issue related to how information is conveyed to the user. One example Takagi gives is when we can predict results well enough to display them in an order similar to what a human user would do, making it less taxing. A related form of user fatigue can set in if each iteration takes too long, such as when there are too many parameters to change at every iteration or if the CA is slow to produce a new generation.

Reducing fatigue is discussed further in several other works. Lai *et al.* [88] discuss this as part of the design pillar *Respect the Creative Process*. Compton [23] introduces the idea of the *Grokloop*, which conceptualizes the iterative loop for a generative system. Fails and Olsen's work [24] on IML introduces a similar loop with the concept of "fast and focused."

IX. Conclusion

In the first part of this survey, after proposing a definition of "mixed initiative" (see Section III), we discussed related research areas (see Section IV), such as casual creators, XAI, and fluidic games. In the core part (see Section VI), we provided an extensive survey on the use of mixed initiative in CC tools with a focus on their use for video games. Table I gives an overview of surveyed works and the related topics & main techniques used. Building on that survey, we identified a set of mixed-initiative techniques and associated challenges (see Sections VII and VIII). The most active current area of investigation is focused on mixed initiative for different genres, such as shooters, puzzles, and strategy games. In the following, we discuss other perspectives for the future applications and improvements of mixed-initiative systems.

A. Future Directions

1) Direct Manipulation: As highlighted by Togelius *et al.* [27], most mixed-initiative works in games have focused on parameterized UIs [115], featuring numbers and sliders where the connection with the content is abstract at best. Most modern CC packages provide direct manipulation of assets. Research on this area remains sparse and should be explored further.

2) Disagreeable Agents: Intuitive and helpful CAs may seem like an attractive idea that leaves the designer in an unperturbed creative workflow. However, Smith [116] asks "how would mixed-initiative systems work if they were designed to be disagreeable with their users?" and argues if, in fact, it is in our interest to design tools that are based solely on "productivity and efficiency." Smith proposes that if we want a reflective design process, then it might be in our interest to design a mixed-initiative tool that encourages deliberation and for the design process to take time.

3) Feminist Interaction Design: Linked to the idea of tools that are not all based around improving efficiency [116], there is but a short leap to a feminist approach to designing mixed-initiative interfaces. Bardzell [117] writes about designing interfaces that are sensitive "to the central commitments of feminism–agency, fulfillment, identity and the self, equity, empowerment, diversity, and social justice." Specializing that human–computer interaction angle to mixed-initiative, we can speculate on the properties of feminist mixed-initiative systems. What would a CA with feminist principles suggest? What would the purpose of a feminist CA be? Using a feminist perspective, Phillips *et al.* [118] analyze PCG for video games, covering topics such as how the gendering of CAs affects the human perception of the CA, a discussion of ownership of the labor of CAs, and related topics.

4) Invisible Agents: EDPCG [35] is characterized by an "invisible hand" that assists in CC. In effect, the player often will not realize that a CA is guiding the CC. This is a player-focused technique, but can we transplant this idea of an invisible hand assisting in CC to a designer facing application? Gingold's *magic crayons* [119] fall into this category.

5) Expert Systems: While traditional expert systems seem to have almost been abandoned in AI, there is still a need for computer systems that simulate or incorporate expertise in a large body of knowledge. For example, consider a CA that advises a designer of household items. It might be an expert in the affordances for the Design of Everyday Things [120], and it could assist by commenting on the user's designs and propose improvements.

6) Roles for AI and Humans: Works such as Bartle's taxonomy of player types [121] can be applied to mixed initiative. For online content, a generative CA could adapt to the style of the player. Similarly, in the context of the design of assets, some users might have a more exploratory approach, while others might be very goal oriented, knowing exactly what they want. In the former case, we want an assistant AI to help with exploration and stick closer to diversity search algorithms. In contrast, in the latter case, it is the task of the AI to second-guess the user so that they can complete the task as effectively as possible. A user could also exhibit both an exploratory and a goal-oriented behavior at different stages of the CC process. This last idea is echoed by the authors of AGE [66] (see Section VI-H).

7) Virtual Reality (VR): Not many authors have looked into mixed-initiative interaction in VR. Novick *et al.* [122] made an online mixed-initiative system in VR, where interaction is triggered by proximity and gaze detection. Users reported that they found the interactive mixed-initiative system at least as engaging as a noninteractive narrative scene. Many mixed-initiative interfaces only let you interact indirectly with the CA, for example, by changing numbers in a GUI or choosing a preferred phenotype via mutant shopping (see Section VII). In VR, however, users expect actions to match closer to their real life counterpart; therefore, interaction is expected to be more direct and immediate. If we hypothesize that users in VR expect interfaces to primarily be of the egocentric direct manipulation type, we need to imagine how interaction with a CA in VR will look like.

References

[1] A. Summerville *et al.*, "Procedural content generation via machine learning (PCGML)," *IEEE Trans. Games*, vol. 10, no. 3, pp. 257–270, Sep. 2018.

[2] N. Barreto, A. Cardoso, and L. G. Roque, "Computational creativity in procedural content generation: A state of the art survey," in *Proc. Conf. Sci. Art Video Games*, 2014.

[3] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: A mixed-initiative level design tool," in *Proc. 5th Int. Conf. Found. Digit. Games*, 2010, pp. 209–216.

[4] G. Smith, J. Whitehead, and M. Mateas, "Tanagra: Reactive planning and constraint solving for mixed-initiative level design," *IEEE Trans. Comput. Intell. AI Games*, vol. 3, no. 3, pp. 201–215, Sep. 2011.

[5] A. Liapis, G. N. Yannakakis, and J. Togelius, "Sentient sketchbook: Computer-aided game level authoring," in *Proc. 8th Int. Conf. Found. Digit. Games*, 2013, pp. 213–220.

[6] A. Liapis, G. N. Yannakakis, and J. Togelius, "Designer modeling for sentient sketchbook," in *Proc. IEEE Conf. Comput. Intell. Games*, 2014, pp. 1–8.

[7] M. Shaker, N. Shaker, and J. Togelius, "Ropossum: An authoring tool for designing, optimizing and solving cut the rope levels," in *Proc. 9th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2013, pp. 215–216.

[8] M. Shaker, N. Shaker, and J. Togelius, "Evolving playable content for cut the rope through a simulation-based approach," in *Proc. 9th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2013, pp. 72–78.

[9] M. Shaker, M. H. Sarhan, O. A. Naameh, N. Shaker, and J. Togelius, "Automatic generation and analysis of physics-based puzzle games," in *Proc. IEEE Conf. Comput. Intell. Games*, 2013, pp. 1–8.

[10] M. Shaker, N. Shaker, J. Togelius, and M. Abou Zleikha, "A progressive approach to content generation," in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2015, vol. 9028, pp. 381–393.

[11] C. Guillemot and O. Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Process. Mag.*, vol. 31, no. 1, pp. 127–144, Jan. 2014.

[12] E. Petrovskaya, C. S. Deterding, and S. Colton, "Casual creators in the wild: A typology of commercial generative creativity support tools," in *Proc. 11th Int. Conf. Comput. Creativity*, 2020, pp. 65–72.

[13] J. R. Carbonell, "Mixed-initiative man-computer instructional dialogues," Ph.D. dissertation, Dept. Elect. Eng., Massachusetts Inst. Technol., Cambridge, MA, USA, 1970.

[14] D. Novick and S. Sutton, "What is mixed-initiative interaction?," in *Proc. Spring Symp. Comput. Models Mixed Initiative Interact.*, 1997, pp. 114–116.

[15] R. Dawkins, *The Blind Watchmaker*. New York, NY, USA: Norton, 1986.

[16] S. Todd and W. Latham, *Evolutionary Art and Computers*. San Diego, CA, USA: Academic, 1992.

[17] J. Secretan *et al.*, "Picbreeder: A case study in collaborative evolutionary exploration of design space," *Evol. Comput.*, vol. 19, no. 3, pp. 373–403, Sep. 2011.

[18] A. Liapis, G. Smith, and N. Shaker, "Mixed-initiative content creation," in *Procedural Content Generation in Games* (Computational Synthesis and Creative Systems). Berlin, Germany: Springer, 2016.

[19] Firaxis Games, "Civilization IV," Game [*Microsoft Windows, MacOS*], 2K Games, Aspyr, Sparks, MD, USA, 2005.

[20] S. Deterding *et al.*, "Mixed-initiative creative interfaces," in *Proc. CHI Conf. Extended Abstr. Hum. Factors Comput. Syst.*, 2017, pp. 628–635.

[21] A. Liapis and G. N. Yannakakis, "Boosting computational creativity with human interaction in mixed-initiative co-creation tasks," in *Proc. ICCC Workshop Comput. Creativity Games*, 2016.

[22] G. N. Yannakakis, A. Liapis, and C. Alexopoulos, "Mixed-initiative co-creativity," in *Proc. Found. Digit. Games*, 2014.

[23] K. Compton, "Casual creators: Defining a genre of autotelic creativity support systems," Ph.D. dissertation, Univ. California, Santa Cruz, CA, USA, 2019.

[24] J. A. Fails and D. R. Olsen, "Interactive machine learning," in *Proc. 8th Int. Conf. Intell. User Interfaces*, 2003, pp. 39–45.

[25] I. Quilez and P. Jeremias, "Shader toy," Feb. 2013. Accessed: Jul. 26, 2021. [Online]. Available: https://www.shadertoy.com/

[26] M. Cook and S. Colton, "Multi-faceted evolution of simple arcade games," in *Proc. IEEE Conf. Comput. Intell. Games*, 2011, pp. 289–296.

[27] J. Togelius, E. Kastbjerg, D. Schedl, and G. N. Yannakakis, "What is procedural content generation?: Mario on the borderline," in *Proc. 2nd Int. Workshop Procedural Content Gener. Games*, 2011, Art. no. 3.

[28] E. Horvitz, "Principles of mixed-initiative user interfaces," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst. CHI Limit*, 1999, pp. 159–166.

[29] J. C. Licklider, "Man-computer symbiosis," *IRE Trans. Hum. Factors Electron.*, vol. HFE-1, no. 1, pp. 4–11, Mar. 1960.

[30] T. Lubart, "How can computers be partners in the creative process: Classification and commentary on the special issue," *Int. J. Human-Comput. Stud.*, vol. 63, no. 4/5, pp. 365–369, Oct. 2005.

[31] M. Guzdial *et al.*, "Friend, collaborator, student, manager: How design of an AI-driven game level editor affects creators," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, 2019, pp. 1–13.

[32] E. Horvitz, "Reflections on challenges and promises of mixed-initiative interaction," *AI Mag.*, vol. 28, no. 2, pp. 19–22, 2007.

[33] K. Compton and M. Mateas, "Casual creators," in *Proc. 6th Int. Conf. Comput. Creativity*, 2015, pp. 228–235.

[34] M. Kreminski, M. Dickinson, J. C. Osborn, A. Summerville, and M. Mateas, "Germinate: A mixed-initiative casual creator for rhetorical games," in *Proc. 16th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2020, pp. 102–108.

[35] G. N. Yannakakis and J. Togelius, "Experience-driven procedural content generation," *IEEE Trans. Affect. Comput.*, vol. 2, no. 3, pp. 147–161, Jul.–Sep. 2011.

[36] O. Biran and C. Cotton, "Explanation and justification in machine learning: A survey," in *Proc. Workshop on Explainable Artif. Intell.*, 2017.

[37] M. J. Guzdial *et al.*, "Explainable PCGML via game design patterns," in *Proc. CEUR Workshop*, 2018, vol. 2282.

[38] *Super Mario Bros*, Nintendo, Kyoto, Japan, Sep. 1985.

[39] M. Cook, J. Gow, and S. Colton, "Danesh: Helping bridge the gap between procedural generators and their output," in *Proc. 7th Workshop Procedural Content Gener.*, 2016, pp. 1–16.

[40] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, "Explainable AI for designers: A human-centered perspective on mixed-initiative co-creation," in *Proc. IEEE Conf. Comput. Intell. Games*, 2018, pp. 1–8.

[41] H. Takagi, "Interactive evolutionary computation: System optimization based on human subjective evaluation," in *Proc. IEEE Int. Conf. Intell. Eng. Syst.*, 1998, pp. 17–19.

[42] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. IEEE*, vol. 89, no. 9, pp. 1275–1296, Sep. 2001.

[43] S. E. Gaudl *et al.*, "Exploring novel game spaces with fluidic games," in *Proc. Artif. Intell. Simul. Behav. Annu. Conv.*, 2017, pp. 344–347.

[44] *Adobe Photoshop*. (Feb. 1990), Adobe Inc., San Jose, CA, USA. Accessed: Jul. 17, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Adobe_Photoshop

[45] *Autodesk 3ds Max*. (Apr. 1996), Autodesk Inc., San Rafael, CA, USA. Accessed: Jul. 17, 2021. [Online]. Available: https://en.wikipedia.org/wiki/Autodesk_3ds_Max

[46] M. Hendrikx *et al.*, "Procedural game content generation: A survey," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 9, no. 1, 2013, Art. no. 1.

[47] M. J. Nelson, S. Meijer, J. Van Der Velden, and A. Iosup, "Mixed-initiative approaches to on-device mobile game design," in *Proc. CHI Workshop Mixed-Initiative Creative Interfaces*, 2017.

[48] I. Karth and A. M. Smith, "Addressing the fundamental tension of PCGML with discriminative learning," in *Proc. 14th Int. Conf. Found. Digit. Games*, 2019, Art. no. 89.

[49] P. Karimi, M. L. Maher, N. Davis, and K. Grace, "Deep learning in a computational model for conceptual shifts in a co-creative design system," 2019, *arXiv:1906.10188*.

[50] N. Davis, C.-P. Hsiao, K. Y. Singh, L. Li, and B. Magerko, "Empirically studying participatory sense-making in abstract drawing with a co-creative cognitive agent," in *Proc. 21st Int. Conf. Intell. User Interfaces*, 2016, pp. 196–207.

[51] K. Sims, "Genetic images," in *Revue Virtuelle Notebook 5*. Paris, France: Centre Georges Pompidou, 1993.

[52] R. Jaksa and H. Takagi, "Tuning of image parameters by interactive evolutionary computation," in *Proc. IEEE Int. Conf. Syst., Man, Cybern. Conf. Theme—Syst. Secur. Assurance*, 2003, pp. 492–497.

[53] A. Liapis, G. N. Yannakakis, and J. Togelius, “Optimizing visual properties of game content through neuroevolution,” in *Proc. 7th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2011, pp. 152–157.
[54] A. Liapis, G. N. Yannakakis, and J. Togelius, “Neuroevolutionary constrained optimization for content creation,” in *Proc. IEEE Conf. Comput. Intell. Games*, 2011, pp. 71–78.
[55] A. Liapis, G. N. Yannakakis, and J. Togelius, “Adapting models of visual aesthetics for personalized content creation,” *IEEE Trans. Comput. Intell. AI Games*, vol. 4, no. 3, pp. 213–228, Sep. 2012.
[56] R. Mazza and C. B. Congdon, “Towards a genetic algorithms approach to designing 3D Polygonal Tree Models,” in *Proc. Congr. Evol. Comput.*, 2002, pp. 1842–1847.
[57] C. B. Congdon and R. H. Mazza, “GenTree: An interactive genetic algorithms system for designing 3D polygonal tree models,” in *Proc. Genet. Evol. Comput. Conf.*, 2003, pp. 2034–2045.
[58] A. Machwe and I. Parmee, “Integrating aesthetic criteria with evolutionary processes in complex, free-form design—An initial investigation,” in *Proc. IEEE Int. Conf. Evol. Comput.*, 2006, pp. 165–172.
[59] A. Martin, A. Lim, S. Colton, and C. Browne, “Evolving 3D buildings for the prototype video game subversion,” in *Proc. Eur. Conf. Appl. Evol. Comput.*, 2010, pp. 111–120.
[60] P. Acampora, “bRigNet,” 2020. Accessed: Jul. 30, 2021. [Online]. Available: https://github.com/pKrime/brignet
[61] *Copilot*, (Jun. 2021), Github, San Francisco, CA, USA. Accessed: Jul. 30, 2020. [Online]. Available: https://copilot.github.com/
[62] M. Johansen, M. Pichlmair, and S. Risi, “Squeezer—A mixed-initiative tool for designing juice effects,” in *Proc. 16th Int. Conf. Found. Digit. Games*, 2021, Art. no. 11, doi: 10.1145/3472538.3472575.
[63] T. Takahashi, K. Tanaka, and N. Oka, “Adaptive mixed-initiative dialog motivates a game player to talk with an NPC,” in *Proc. 6th Int. Conf. Human-Agent Interact.*, 2018, pp. 153–160.
[64] I. S. Stefnisson and D. Thue, “Mimisbrunnur: AI-assisted authoring for interactive storytelling,” in *Proc. 14th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2018, pp. 236–242.
[65] B. Samuel, M. Mateas, and N. Wardrip-Fruin, “The design of writing buddy: A mixed-initiative approach towards computational story collaboration,” in *Proc. Int. Conf. Interact. Digit. Storytelling*, 2016, pp. 388–396.
[66] R. Craveirinha and L. Roque, “Exploring the design-space: The authorial game evolution tool case-study,” in *Proc. 13th Int. Conf. Adv. Comput. Entertainment Technol.*, 2016, Art. no. 21.
[67] A. Mobramaein *et al.*, “CADI—A conversational assistive design interface for discovering pong variants,” in *Proc. 14th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2018, pp. 194–200.
[68] T. Machado *et al.*, “CICERO: Computationally intelligent collaborative EnviROnment for game and level design,” in *Proc. ICCC Comput. Creativity Games Workshop*, 2017.
[69] E. J. Hastings, R. K. Guha, and K. O. Stanley, “Evolving content in the galactic arms race video game,” in *Proc. IEEE Symp. Comput. Intell. Games*, 2009, pp. 241–248.
[70] A. Liapis, H. P. Martánez, J. Togelius, and G. N. Yannakakis, “Adaptive game level creation through rank-based interactive evolution,” in *Proc. IEEE Conf. Comput. Intell. Games*, 2013, pp. 1–8.
[71] J. Schrum, J. Utierrez, V. Volz, J. Liu, S. Lucas, and S. Risi, “Interactive evolution and exploration within latent level-design space of generative adversarial networks,” in *Proc. Genet. Evol. Comput. Conf.*, 2020, pp. 148–156.
[72] N. R. Sturtevant, D. Nicolas, A. Tripodi, C. Yang, and M. Guzdial, “A demonstration of anhinga: A mixed-initiative EPCG tool for snakebird,” in *Proc. AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2020, pp. 328–330.
[73] M. Charity, A. Khalifa, and J. Togelius, “Baba is Y’all: Collaborative mixed-initiative level design,” in *Proc. IEEE Conf. Games*, 2020, pp. 542–549.
[74] S. Dahlskog and J. Togelius, “Patterns, dungeons and generators,” in *Proc. Found. Digit. Games Conf.*, 2015, Art. no. 30.
[75] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Mixed-initiative procedural generation of dungeons using game design patterns,” in *Proc. IEEE Conf. Comput. Intell. Games*, 2017, pp. 25–32.
[76] A. Baldwin, S. Dahlskog, J. M. Font, and J. Holmberg, “Towards pattern-based mixed-initiative dungeon generation,” in *Proc. 12th Int. Conf. Found. Digit. Games*, 2017, pp. 1–10.
[77] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, and S. Johansson, “Assessing aesthetic criteria in the evolutionary dungeon designer,” in *Proc. 13th Int. Conf. Found. Digit. Games*, 2018, Art. no. 44.
[78] A. Alvarez, S. Dahlskog, J. Font, J. Holmberg, C. Nolasco, and A. Österman, “Fostering creativity in the mixed-initiative evolutionary dungeon designer,” in *Proc. 13th Int. Conf. Found. Digit. Games*, 2018, Art. no. 50.
[79] A. Alvarez, S. Dahlskog, J. Font, and J. Togelius, “Empowering quality diversity in dungeon design with interactive constrained MAP-Elites,” in *Proc. IEEE Conf. Games*, 2019, pp. 1–8.
[80] M. J. Guzdial, J. Chen, S.-Y. Chen, and M. O. Riedl, “A general level design editor for co-creative level design,” in *Proc. AAAI Workshop AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2017, pp. 2013–2015.
[81] A. Hoyt *et al.*, “Integrating automated play in level co-creation,” pp. 2016–2018, 2018.
[82] E. Butler, A. M. Smith, Y.-E. Liu, and Z. Popovic, “A mixed-initiative tool for designing level progressions in games,” in *Proc. 26th Annu. ACM Symp. User Interface Softw. Technol.*, 2013, pp. 377–386.
[83] P. Migkotzidis and A. Liapis, “SuSketch: Surrogate models of gameplay as a design assistant,” *IEEE Trans. Games*, early access, Mar. 23, 2002, doi: 10.1109/TG.2021.3068360.
[84] M. Haggerty, “Evolution by Esthetics, an Interview with W. Latham and S. Todd,” *IEEE Comput. Graph. Appl.*, vol. 11, no. 2, pp. 5–9, Mar. 1991.
[85] L. Doherty, M. Whitney, and J. Shakya, “Mixed-initiative in computer games: Algorithmic content creation in open-ended worlds,” in *Proc. AAAI Fall Symp.*, 2005, pp. 46–50.
[86] B. Blumberg, “New challenges for character-based AI for games introduction: AI’s next steps,” in *Proc. AAAI Spring Symp.*, 2002, pp. 41–45.
[87] M. Gumin, “Wave function collapse,” 2016. Accessed: Jul. 21, 2021. [Online]. Available: https://github.com/mxgmn/WaveFunctionCollapse
[88] G. Lai, W. Latham, and F. F. Leymarie, “Towards friendly mixed initiative procedural content generation: Three pillars of industry,” in *Proc. Int. Conf. Found. Digit. Games*, 2020, Art. no. 11.
[89] word2vec. Accessed: Jul. 31, 2021. [Online]. Available: https://code.google.com/archive/p/word2vec
[90] K. O. Stanley, “Compositional pattern producing networks: A novel abstraction of development,” *Genet. Program. Evolvable Mach.*, vol. 8, no. 2, pp. 131–162, 2007.
[91] K. O. Stanley and R. Miikkulainen, “Evolving neural networks through augmenting topologies,” *Evol. Comput.*, vol. 10, no. 2, pp. 99–127, Jun. 2002.
[92] S. O. Kimbrough *et al.*, “On a feasible–infeasible two-population (FI-2Pop) genetic algorithm for constrained optimization: Distance tracing and no free lunch,” *Eur. J. Oper. Res.*, vol. 190, no. 2, pp. 310–327, Oct. 2008.
[93] J. Grefenstette, “Genesis: Genetic search implementation system.” Accessed: Mar. 1, 2021. [Online]. Available: https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/0.html https://www.cs.cmu.edu/afs/cs/project/ai-repository/ai/areas/genetic/ga/systems/genesis/0.html
[94] *Subversion Game [Release suspended]*. Introversion Softw., Walthon-On-Thames, U.K., 2011.
[95] Z. Xu, Y. Zhou, E. Kalogerakis, C. Landreth, and K. Singh, “RigNet: Neural rigging for articulated characters,” *ACM Trans. Graph.*, vol. 39, no. 4, 2020, Art. no. 58.
[96] M. Chen *et al.*, “Evaluating large language models trained on code,” 2021, *arXiv:2107.03374*.
[97] M. Jonasson and P. Purho, 2012. Accessed: Jul. 30, 2021. [Online]. Available: https://www.gdcvault.com/play/1016487/Juice-It-or-Lose
[98] S. Swink, *Game Feel: A Game Designer’s Guide to Virtual Sensation*. Evanston, IL, USA: Routledge, 2008.
[99] B. Samuel, A. A. Reed, P. Maddaloni, M. Mateas, and N. Wardrip-Fruin, “The ensemble engine: Next-generation social Physics,” in *Proc. 10th Int. Conf. Found. Digit. Games*, 2015, pp. 22–25.
[100] *Pong*, Game [Arcade], Atari Japan, Nov. 1972.
[101] D. Perez-Liebana *et al.*, “The 2014 general video game playing competition,” *IEEE Trans. Comput. Intell. AI Games*, vol. 8, no. 3, pp. 229–243, Sep. 2016.
[102] M. Ebner, J. Levine, S. M. Lucas, T. Schaul, T. Thompson, and J. Togelius, “Towards a video game description language,” *Dagstuhl Follow-up*, vol. 6, pp. 85–100, 2013.
[103] T. Schaul, “A video game description language for model-based or interactive learning,” in *Proc. IEEE Conf. Comput. Intell. Games*, 2013, pp. 1–8.
[104] A. Summerville, C. Martens, B. Samuel, J. Osborn, N. Wardrip-Fruin, and M. Mateas, “Gemini: Bidirectional generation and analysis of games via ASP,” in *Proc. 14th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment*, 2018, pp. 123–129.
[105] Noumenon Games, “Snakebird,” Game [iOS, Android, Microsoft Windows, Mac OS, Linux], 2015.
[106] Hempuli, “Baba Is You,” Game [Microsoft Windows, MacOS, Linux, Nintendo Switch], Mar. 2019.
[107] S. M. Lucas and V. Volz, “Tile pattern KL-divergence for analysing and evolving game levels,” in *Proc. Genet. Evol. Comput. Conf.*, 2019, pp. 170–178.

[108] J.-B Mouret and J. Clune, “Illuminating search spaces by mapping elites,” 2015, *arXiv:1504.04909*.
[109] M. Guzdial, N. Liao, and M. O. Riedl, “Co-creative level design via machine learning,” in *Proc. CEUR Workshop*, 2018.
[110] P. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. SSC-4, no. 2, pp. 100–107, Jul. 1968.
[111] *Cut the Rope*, Game [iOS, Android, Windows Phone, Nintendo DS, Nintendo 3DS, MacOS, Microsoft Windows], Chilingo, Activision, ZeptoLab, Moscow, Russia, Oct. 2010.
[112] K. Kelly, “Genetic images,” *Wired*, Sep. 1994.
[113] A. Liapis, G. Yannakakis, and J. Togelius, “Limitations of choice-based interactive evolution for game level design,” in *Proc. 8th AAAI Conf. Artif. Intell. Interact. Digit. Entertainment Workshop*, 2012, pp. 33–36.
[114] A. Lomas, “Morphogenetic vase forms,” in *Proc. Conf. Artif. Life*, 2019, pp. 523–530.
[115] T. Betts, “Procedural content generation,” in *Handbook of Digital Games*. Hoboken, NJ, USA: Wiley, 2014, pp. 62–91.
[116] G. Smith, “What do we value in procedural content generation,” in *Proc. Int. Conf. Found. Digit. Games*, 2017, Art. no. 69.
[117] S. Bardzell, “Feminist HCI,” in *Proc. 28th Int. Conf. Hum. Factors Comput. Syst.*, 2010, Art. no. 1301.
[118] A. Phillips, G. Smith, M. Cook, and T. Short, “Feminism and procedural content generation: Toward a collaborative politics of computational creativity,” *Digit. Creativity*, vol. 27, no. 1, pp. 82–97, Jan. 2016.
[119] C. Gingold, “Miniature gardens & magic crayons: Games, spaces, & worlds,” Master’s thesis, School Literature, Culture, Commun., Georgia Inst. Technol., Atlanta, GA, USA, 2003.
[120] D. A. Norman, *The Design of Everyday Things*. New York, NY, USA: Basic Books, 2002.
[121] R. Bartle, “Hearts, clubs, diamonds, spades: Players who suit MUDs,” 1996. Accessed: Sep. 11, 2020. [Online]. Available: https://mud.co.uk/richard/hcds.htm
[122] D. Novick, L. J. Hinojos, A. E. Rodriguez, A. Camacho, and M. Afravi, “Conversational interaction with multiple agents initiated via proxemics and gaze,” in *Proc. 6th Int. Conf. Human-Agent Interact.*, 2018, pp. 356–358.

Gorm Lai received the B.Sc. and M.Sc. degrees in computer science from Aarhus University, Aarhus, Denmark, in 2001 and 2004, respectively. He is currently working toward the Ph.D. degree with the EPSRC Centre for Doctoral Training in Intelligent Games and Game Intelligence, Goldsmiths, University of London, London, U.K.

He was a Cooperating Research Fellow with the Graduate School of Informatics, Nagoya University, Nagoya, Japan, from June 2019 to March 2020. Before rejoining academia for his Ph.D. studies, he was a Senior Graphics, Audio and Game Engine Programmer specialized in cross-platform and experimental game development. He has worked on 17 commercial games since 2004; some of them are multiple-award-winning games. His research interests focus on how mixed-initiative interfaces can assist in creating procedurally generated creatures for use in video games. Besides video games, his portfolio includes virtual reality, interactive installations, and other applications.

Frederic Fol Leymarie received the Ph.D. degree from the Engineering Division, Brown University, Providence, RI, USA, in 2003.

He joined the Computing Department, Goldsmiths, University of London, London, U.K., to develop and contribute to novel (or less explored) areas of research and teaching in computing and beyond. Together with W. Latham, he launched the first M.Sc. in computer games programming in the greater London area in 2008. This activity is now complemented by an M.A. in computer games art and design. Both programs are industry facing; more than half the lecturers they recruited are from the industry. He explores the areas of research bringing together creative artistic activities, vision, and perception understanding, with computing. His recent collaborations with artists have included W. Latham, P. Tresset (creator of Paul the robot), D. Berio, and T. Broad. In 2011, he, together with W. Latham, launched London Geometry Ltd., London, which conducts projects focused on interactive visualization, games, science, and art. In 2015, he, together with Prof. S. Rueger, launched DynAIkon Ltd., London, which focuses on biodiversity monitoring and machine learning.

William Latham received the bachelor’s degree in fine art from the University of Oxford, Oxford, U.K. in 1982 (attending Christchurch College), and the masters degree in fine art from the Royal College of Art, London, U.K. in 1985.

He is well known for his pioneering Organic Computer Art created in the late 1980s and early 1990s when he was a Research Fellow with IBM, Winchester, U.K. His book with Stephen Todd *Evolutionary Art and Computers* (New York, NY, USA: Academic, 1999) on interactive evolutionary design covering the period at IBM is cited as a leading publication in this domain. In 1993, he moved into Rave Music creating album covers and videos for bands such as The Shamen. He then moved into computer games development for ten years as a creative director creating games published by Universal Studios, Microsoft, and Warner Bros. In 2007, he became a Professor in Computing with Goldsmiths, University of London, London, working as Co-Investigator on new research projects in Protein Docking with Imperial College Bioinformatics Group and DNA/RNA visualization with the Weatherall Centre of Molecular Medicine, University of Oxford. His Mutator VR Art Experience on HTC Vive developed with long-term collaborator Stephen Todd and with Lance Putnam has been exhibited to much acclaim at the Pompidou Centre in Paris, France, and Ars Electronica in Linz, Austria, in Venice, Italy, and in St. Petersburg, Russia (linked to The Hermitage Museum). He is the Director and co-founder of SoftV Ltd., Hove, U.K. (developing Neuroscience Patient Apps for the NHS) and London Geometry Ltd., London.