Born in Fair Haven: Procedural Generation of Cultural Sign Systems

Craig Paul Green, Callum Nash, Jo Briggs
School of Design
Northumbria University
Newcastle-upon-Tyne, UK
craig.p.green@northumbria.ac.uk, callum.nash@northumbria.ac.uk,
jo.briggs@northumbria.ac.uk

Xiaoming Dong
Independent
xiaomingdong.joy@gmail.com

ABSTRACT
We propose the term *procedural cultural sign systems generation* to help define the emergent practice of procedurally generating games that seek to mimic or in some way signify fictitious culture. We borrow from cybernetics to understand procedural cultural sign systems generation and exemplify this with our *V*Machine algorithm. We argue the need for procedural content generation algorithms that are readily implementable, and generalizable for multiple use cases. In this paper, we discuss cybernetics in the context of procedural content generation followed by an overview of recent games that generate elements of culture as part of gameplay. We then outline our *V*Machine Algorithm and our method, before demonstrating practical applications in two short case studies based on games we are currently developing.

Keywords
Procedural cultural sign systems generation, cybernetics, *V*Machine Algorithm, procedural culture, generating culture.

INTRODUCTION
The conference reviewers aptly pointed out that this paper could have been split into three; one presenting case studies of what we call Procedural Cultural Sign System Generation or procedural culture for short, another applying cybernetic principles to explain this practice, and a third detailing a working example of an algorithm that attempts to demonstrate these principles. In presenting these in a single paper, our aim is to encourage thinking about how the case studies interrelate and encourage critical discussion towards sketching out a theoretical framework by 'providing a context in which to think'.

In the popular television series *Star Trek*, the holodeck can generate vast environments and narratives from the utterance of a few words. Projecting the audience to the year 2376 (though filmed in 2000), the crew of USS Voyager, for respite, created 'Fair Haven', an interactive 19th century Irish village populated by hundreds of characters going about their everyday routines and activities, interacting with each other and with Voyager's crew members. We reference this village in our title because it represents an emergent culture and associated mythology, yet one that is a stereotypical toffee tin picture of Irish society in the 19th century. This illustrates a key limitation when attempting to procedurally generate culture. No matter how sophisticated or how much variety is produced for these cultures, what is produced will carry the inherent biases and ideology of the designer-developers. Perhaps, Fair Haven exemplifies what to avoid when procedurally generating culture.

The holodeck's advanced level of immersive, narrative driven cultural sign generation and representation as envisioned in Star Trek, is of course still not possible today. However, the generation of causally related assets, using Procedural Content Generation (PCG), particularly in AAA open-world, massive multiplayer online games, is slowly becoming normalised as the technology becomes more reliable. These causally generated assets promise players exploration of vast living environments. They in part respond to calls from gamer and developer communities for multi-level multi-content PCG (Togelius et al. 2013) or perpetual uniqueness (Short and Adams 2019, 15), where wildly different game worlds with all the associated content are generated at the click of a button, with memorable characters guaranteed to appear on demand. There is also community appetite for general content generators (Cook et al. 2019; Togelius et al. 2013), which can be used in multiple contexts, and are easily deployable for a wide range of projects. These help to proliferate PCG techniques and systems to different groups of designer-developer users, both in industry and academia. One of the motivations of this paper, then, is to advocate for these principles while contributing a practical solution.

We begin the paper by discussing our motivation, before introducing cybernetics as a way of explaining how PCG systems handle complexity. We then give an overview of well received contemporary games that generate elements of culture as part of gameplay, which we refer to as cultural sign systems informed by Roland Barthes (1977). We go on to outline our algorithm, which uses cybernetics to inform the handling of complexity—in cybernetics' terms 'variety'—to contribute an approach to the proliferation of large amounts of variety in procedural culture and beyond. The algorithm provides a low-barrier-to-entry technique and can be used in multiple contexts.

By 'algorithm' we refer to a finite sequence of well-defined, computer-implementable instructions that typically solve a class of specific problems, or that can perform a computation (Math Vault n.d.; Merriam Webster Dictionary n.d.). While loosely synonymous with complexity, we use variety in reference to Ashby's (1956), mathematical description of the total number of possible states within a system (e.g., a flipped coin has a variety of two, a dice a variety of six).

BACKGROUND AND MOTIVATION

PCG systems for video games take an amount of input variety and through an algorithmic process create a variety of output content that the player interacts with through gameplay. The central problem we explore in this paper is this: too little variety limits the experience of novelty and too much produces incoherent, unusable output (Short and Adams 2019). Using limited or excessive variety for PCG systems is especially noticeable in AAA, open-world massively multiplayer online games where thousands of players are exploring one game environment together. As the scale of game environments has increased to accommodate ever-larger player communities, the challenges for procedural generation have similarly multiplied, and will continue to do so potentially exponentially. For example, *Elite Dangerous* (2014) claims to have a 1:1 scale Milky Way Galaxy replete with 400 billion explorable systems. If only 0.1% of these contained unique civilisations, the games designer-developers would need to create 400,000,000 separate civilisations. The scale of this problem rapidly increases if these individual civilisations are designed to emulate elements of culture in such a way that is meaningful to players, to have unique varieties of cultural signs that are interactive, that develop over time and interrelate.

We were motivated to appropriate cybernetics, specifically Ashby's Principle of Requisite Variety (1956) and Stafford Beer's Viable System Model (1984), as a means of balancing novelty and coherency in output variety, using alien civilisations and their cultural sign systems for demonstration. Furthermore, designing PCG algorithms with limited variety means that they are often not reusable outside of their original purpose (Togelius et al. 2013). We were motivated to investigate to what extent cybernetics could help us design more generalisable, easy to use PCG.

Culture in Video Games

Our proposed heuristic model alludes to cultural signs comprising physical manifestations that signify the values and behaviours of a society (Barthes 1977; Danesi 2013). While exactly what constitutes culture within a game environment is a topic for rich discussion beyond the scope of this paper, we advocate for establishing better-informed debate on how narrative-driven games inherently express and reinforce values and ideologies (Ash and Gallacher 2011; Longan 2008).

Drawing on Cybernetics: Ashby's Law of Requisite Variety

A universal principle in cybernetics is that for a system to be viable and function as expected it must be able to handle a variety of states (parameters) and operate within a defined set of states. If a state is exceeded, such as the external temperature being too low for an organism to maintain homeostasis, the system must adapt to remain viable. It does this by amplifying (adding to) its variety of states. For example, human beings could be said to amplify variety to deal with cold weather by putting on a jacket. Variety-handling on the part of the system can also attenuate—or reduce—its number of states to remain viable. Take for example the Earth's atmosphere, where human production creates carbon that cannot be effectively absorbed by nature. People must both attenuate the production of atmospheric carbon whilst also amplifying carbon storage to ensure the viability of natural systems upon which humankind depends.

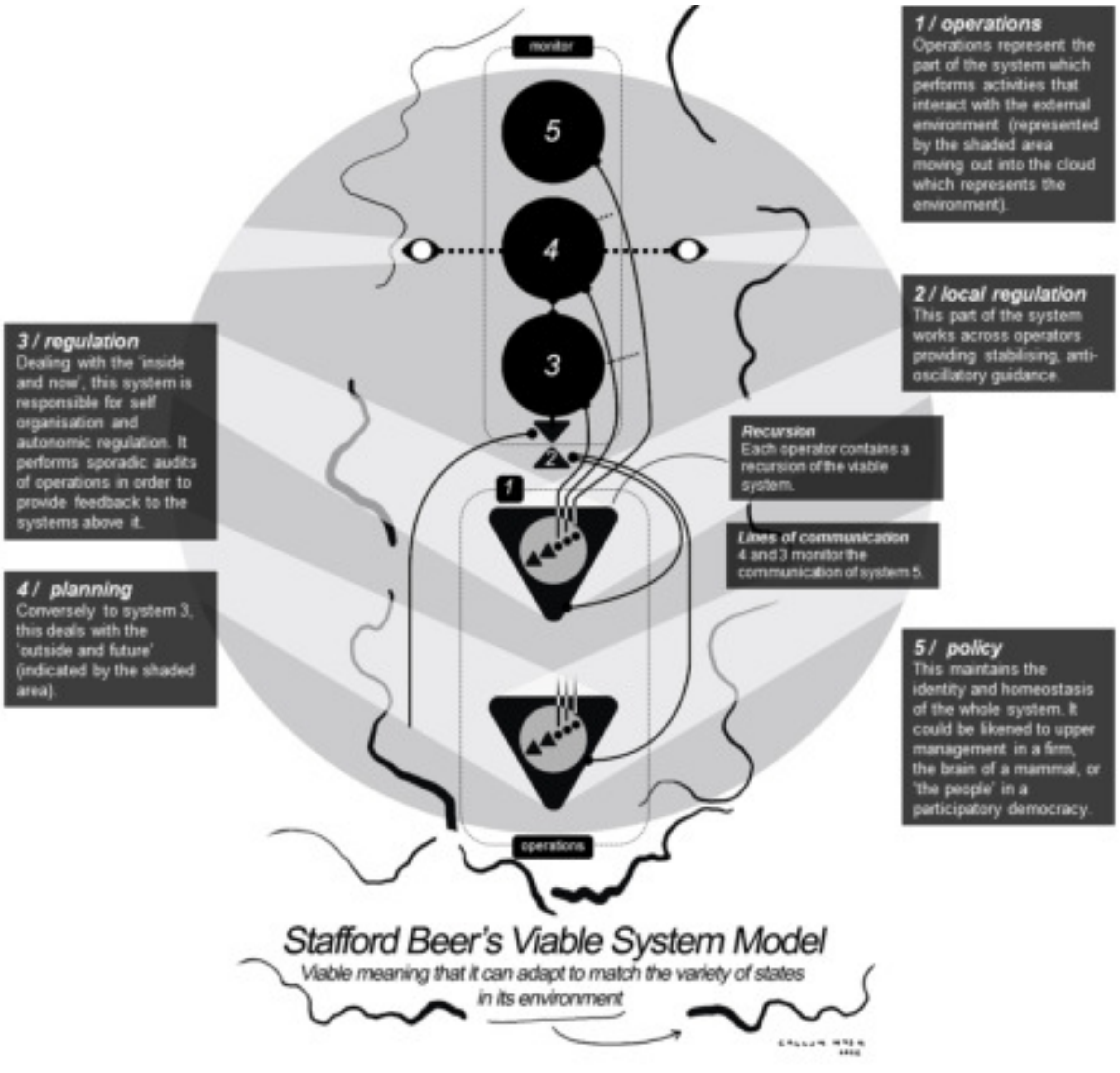
This way of understanding the viability of both simple and complex systems is described in Ashby's principle of requisite variety: every system must contain the requisite variety of states to match those of its environment. This can be achieved through attenuating and/or amplifying the systems number of states. Ashby therefore states that "Variety can destroy variety" (1956, 207), using the example of two players in a game. For one to win, they must create a variety of states that their player opponent cannot handle by luck, knowledge, or some external means. On the other hand, if the opponent finds a way to increase or attenuate their variety to handle these states they remain in the game. The end of the game is reached when one player's variety has absorbed that of the other player with gameplay essentially a dialectic opposition of two sets of variety. If Ashby's principle has universal application, then the question we examine is how this principle applies to PCG systems. PCG systems can produce a variety of states, but it is the players who must ultimately handle this variety in both narrative and gameplay.

Too much variety can lead to output that is incoherent and not fit for the designer's intended purpose e.g., "One would not want airplanes to change in one step into wheelbarrows" (Togelius et al. 2013, 66). Using a cybernetics lens, the player can be understood as a system containing a certain amount of variety. Some players will be able to narratively contextualise the appearance of the wheelbarrow—in what, Jason Grinblat, the designer of the emergent narrative game Caves of Qud (2015) refers to as Apophenia, the tendency for people to perceive meaningful connections between unrelated things (GDC 2019; Grinblat and Bucklew 2017). Other players will reject it outright. This might be considered a simple content design problem; why would the designer provide an airplane with the possibility of turning into a wheelbarrow? Limiting the variety works to ensure coherency of output. However, in our case of procedural culture for alien civilisations, it is much harder for the designer to manually restrict, in design time, the huge variety of outputs required if each new alien civilisation is to be considered novel by players.

Consider for example the children's television program Transformers (1984) in which it is perfectly plausible that an airplane should suddenly 'transform' into a wheelbarrow. However, this action is contextualised and made acceptable by the narrative, provided that the audience have suspended their disbelief (Coleridge 1817). If initial design input is limited, the opportunity to create unique civilizations and construct compelling narratives around them is limited. Counter to this PCG variety problem is Kate Compton's oatmeal problem (Short and Adams 2019, 15). Here, procedural generation produces highly varied parameters for a predictable outcome. In this case, the players easily absorb the variety generated by the PCG system but do not experience novelty. Therefore, from the perspective of Ashby's principle, procedural generation can only be perceived as novel (instead of random) if the variety of PCG in the narrative and the player's capacity to make sense of that PCG within the narrative *remain in step*.

VARIETY AND VIABILITY – THE VIABLE SYSTEM MODEL

According to Ashby’s principle, when two systems can absorb each other’s variety, they are viable. So, when we talk about the variety produced by a system and variety that could be absorbed by the player as remaining in step, we can understand this in terms of Ashby’s statement that ‘variety destroys variety’. Stafford Beer’s Viable System Model (1984) operationalises this principle in a generalisable system model for any viable system, from the autonomous nervous system to entire economies.



monitor
5
4
3
2
1
operations
1 / operations
Operations represent the part of the system which performs activities that interact with the external environment (represented by the shaded area moving out into the cloud which represents the environment).
2 / local regulation
This part of the system works across operators providing stabilising, anti-oscillatory guidance.
3 / regulation
Dealing with the 'inside and now', this system is responsible for self organisation and autonomic regulation. It performs sporadic audits of operations in order to provide feedback to the systems above it.
4 / planning
Conversely to system 3, this deals with the 'outside and future' (indicated by the shaded area).
5 / policy
This maintains the identity and homeostasis of the whole system. It could be likened to upper management in a firm, the brain of a mammal, or 'the people' in a participatory democracy.
Recursion
Each operator contains a recursion of the viable system.
Lines of communication
4 and 3 monitor the communication of system 5.
Stafford Beer's Viable System Model
Viable meaning that it can adapt to match the variety of states in its environment

Figure 1: Diagram of Stafford Beer’s Viable System Model.

We take from Beer the notion that we can design PCG systems to better handle variety, and that any complex PCG system (such as the holodeck from which we draw inspiration) must contain ‘variety handling apparatus’ to realise the full potential of PCG systems as being able to provide ongoing novelty to players in gameplay and audiences in narratives. The Vmachine we shall go onto present operationalises a small part of this model in recursively attenuating variety, and we present the Viable System Model here to illustrate our design inspiration and encourage further discussion on the application of this and other cybernetic lens.

PROCEDURAL CONTENT GENERATION METHODS

PCG in video games is a well-established practice with a long history stretching back to the early years of computing. There are many approaches, each with respective advantages and limitations (Azad and Martens 2019; Bontchev 2016; Harrell 2005; Hartsook et al. 2011; Karth 2019; Kreminski and Wardrip-Fruin 2018; Balali Moghadam and Kuchaki Rafsanjani 2017; Salminen et al. 2017; Togelius et al. 2011; Togelius, Justinussen, and Hartzen 2012). Current approaches tend to focus on generating content that designer-developers can guarantee will produce outputs relevant to the generation context, achieved by limiting the design input. Because of this, systems produced are rarely multiple or general-purpose. Similar to our approach (Togelius et al. 2011) established the idea of fitness functions in search-based procedural generation enabling the algorithm to attenuate selection of procedurally generated content to that perceived as having meaningful connections. These mechanisms include direct fitness functions whereby generated content is assigned a fitness value. This is useful for content that has a specific function such as the number of doors or exits in a dungeon. Simulation-based Fitness Functions that use artificial agents to establish the viability of content are useful to determine if generated content is usable. Meanwhile, Interactive Fitness Functions define the fitness of generated content by monitoring how players interact with it during gameplay. There are also examples of PCG systems that produce a wide range of encyclopaedic content (Ryan 2014) that could be useful for the generation of a large variety of different content types. Generating content from large datasets such as Wikipedia is also a viable approach (Barros et al., 2016)

To create an algorithm for the generation of procedural culture for fictional alien civilisations, large databases that readily communicate with each other are useful for diversifying output. This means that any selection process must recognise meaningful connection possibilities for extremely large amounts of diverse content. In our attempts to address limitations of database handling and to create a flexible, easy-to-use, generalisable system we contribute the *V*Machine Algorithm (see later). This is a probabilistic constraint solving algorithm (Short and Adams 2019, 9) that amplifies and/or attenuates variety recursively, to provide a constraint solver that can utilise extremely large databases, without the need for high levels of computation or wo/manpower. The algorithm is heuristic and readily adaptable and aims to be suitable for the generation of procedural culture and potentially other PCG tasks.

State of the Art Procedural Culture Generation in Video Games

First, in this section we summarise how procedural culture generation works in three different games to demonstrate the problem space and application of our *V*Machine Algorithm. Although the associated algorithms of these games are generally not public knowledge, there are insights provided by the game's developers, through Youtube videos, academic papers, and lectures. We also examine their output through gameplay, as we are examining PCG in these cases from the players' perspective—and direct experience of the variety produced—rather than solely examining the underlying tools.

No Man's Sky's (Hello Games 2016) worlds feature generated terrain, foliage, creatures, weather conditions, mineral placement, sites of interest, etc. (GCD 2017). The game also generates fragments of culture, in the form of 'ancient' ruins, which afford players opportunities for interpretation from the perspective of exploration and promotes the imagining of a universe full of sentient life capable of producing advanced culture. This is constrained by the limited content available and relies heavily on players wanting to explore. In their study of this feature Catherine Flick et al. (2017) refer to players as archeologist gamers or 'archaeogamers' who explore, catalogue, and analyze found objects as they are generated and discovered in the game, and who then seek to set acode of conduct echoing those used in real-world archaeological sites. There are also limited examples of pre-designed cultural sign systems that are not the remnants of past civilisations, including the “five sentient species that can be encountered throughout the universe, each with distinct languages, lore, and technology” (‘No Man’s Sky Wiki’ n.d.).

Dwarf Fortress (Bay 12 Games 2002) uses algorithms that define the narrative space in which generated worlds exist supported by societal histories. Day-to-day interactions between characters are factored in, creating a rich emergent backstory. This includes defining the potential of the character agents (Aylett 1999) to act, which provides near-endless narrative content to explore, interpret, and perceive as emerging from or cocreated by player input (Kreminski and Mateas 2021). However, there are also constraints, most notably concerning the predefined character types: dwarves, humans, elves, and goblins, etc., and their typical behaviour. This is a simple way of establishing meaning for players but is limited by pre-design. This approach would be inadequate if applied to procedurally generating novel alien civilisations from beyond the common fantasy canon. The implication is that the game’s generation process relies on typical character behaviors designed by the developer rather than generated by algorithms. On the other hand, the game provides a large variety of cultural signification, which draws players into an emergent narrative as they are surprised by the seeming autonomy of the dwarves and the complexity of their generated history. Dwarf Fortress provides unexpected variety, creating the feeling of emergence (Walsh 2011) as players must both interpret and contextualise the evolving cultural space.

Blaseball (The Game Band 2020) is a relatively new online game that simulates a fantasy baseball league. Players interact by picking a favourite team and selecting a fictional player from this team to idolize. These team players have combinations of obscure metrics such as number of fingers, their ‘shakespearianism’ and ‘unthwackability’ (People Make Games 2020) which provides data for comparison of fictional team player abilities and their effectiveness against other teams. *Blaseball’s* concept design is rooted in surrealism, providing interesting characterisation and situations, which are used by game players to inform and develop their own stories and culture. This approach goes beyond the core gameplay loop, enabling game players to make sense of the game’s surrealist content while interacting with and contributing to it, driving its emergent collective fanfiction created by player communities. This promotes ongoing dialogue between the fans of the game and the game’s developers, mediated by the game’s unique abstract setting, logic, and game mechanics. In comparison to the other games discussed, *Blaseball* gives little variety, comprising mostly of the procedural generation of statistics, though the game’s community of players amplifies this variety and through collective fiction have established a different social form of procedural culture.

	No Man's Sky	***Dwarf Fortress***	***Blaseball***
Procedural Culture	Fragments scattered around generated terrain at specific sites of interest	Created as a part of game world generation	Created by player communities, based on in-game content, from which designers then add new parameters
How the player engages with the procedural culture	Exploration and gameplay	Core gameplay and exploration of information about the generated world	Players form communities to engage with and develop procedural culture
Required level of player interpretation of procedural culture	Medium	Low	High

Table 1: Summary of procedural culture in the three games discussed.

THE *V*MACHINE ALGORITHM

The *V*Machine Algorithm is a stochastic variety handling algorithm utilising a Bayesian network (Ben-Gal 2008) that determines whether node states are likely to exist in relation to other node states. States are specific content, and nodes are state containers and state selectors. The algorithm is designed to foster meaningful connections between individual nodes. When the algorithm runs, nodes can use any other nodes that have a selected state to attenuate their own state selection. Once this process is complete, and all nodes have selected their states the resulting content is consolidated and used for the intended purpose. This could involve producing content that is then used for the creation of graphical content or defining alien behaviours (see later). Each node has a scale (between 1 and a user-defined integer) that all possible states must reference to provide a set reference points for attenuation between nodes (see Figure 2).

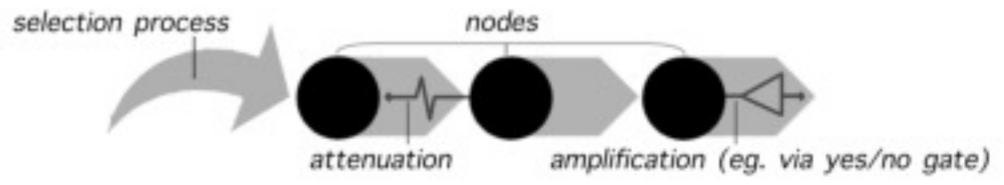


Figure 2: Selection, attenuation, and amplification between nodes i.e. state containers and state selectors.

The Mathematical Method

In this Section we present the mathematics required to build the algorithm, this is for designer-developers who are interested in building and using the algorithm for their own projects/research. To set up the algorithm, a designer-developer must create a set of nodes and associated potential states. Each state is assigned an appropriate state scale integer. Designer-developers select which nodes attenuate which other nodes. When the algorithm runs, nodes pass their state selection along a chain, which is then processed by receiving nodes. Each receiving node randomly selects from its own list of states, takes this state's scale number, and deducts it from the received state's scale number. In the following example A represents the state scale number received by the node, B represents the state scale number selected by the node, and X represents the difference between A and B, with X taking the absolute value of A minus B.

$$X = |A - B|$$

In our example the sample space is thus: A {1,2,3,4,5}, B {1,2,3,4,5}, X {0,1,2,3,4}. X is a discrete variable over a particular range of real values from 0 to 4. The probability distribution of X is shown below.

X	X1 = 0	X2 =1	X3 = 2	X4 = 3	X5 = 4
P	P(X=x1) = $\frac{1}{5}$	P(X=x2) = $\frac{8}{25}$	P(X=x3) = $\frac{6}{25}$	P(X=x4) = $\frac{4}{25}$	P(X=x5) = $\frac{2}{25}$

Table 2: Possible X values and their associated probabilities.

P(R) is the probability of reselecting the node's state, P(N) is the probability of not reselecting the node's state, P(X|R) is the conditional probability of reselecting when X happens, e.g., P(x=0|R) represents the probability of reselecting when X = 0, hence P (x= 0|R) = $\frac{1}{100}$. Individual nodes can be attenuated by groups of nodes. This is done by querying each node in the group individually. So, if node A is attenuated by node B and node C, A must produce positive results on both B and C's content to finalise its selection.

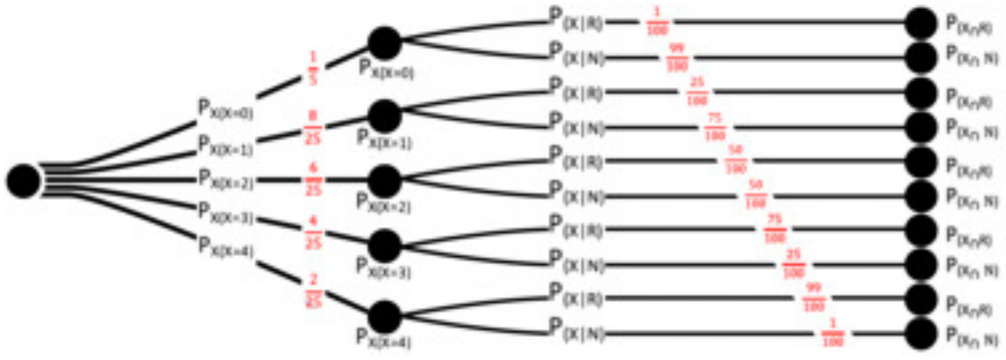


Figure 3: The probability tree diagram for the *V*Machine Algorithm output, P(X|R) is the probability of a node reselecting its state, and P(X|N) is the probability of a node not reselecting its state.

DEMONSTRATING THE *V*MACHINE ALGORITHM

We now outline two case studies that we have implemented and that are currently testing the algorithm in games developed by the first two authors. The first concerns the context of procedural culture and the second generates behaviours for a fictional character. Both comprise unique instances of using the algorithm to demonstrate its ease of use and generalizability in multiple contexts.

Procedurally Generating Cultural Content: Fictional Civilisations

This project has been developed to generate backend content, i.e., not seen by players but useful in generating a wide variety of frontend content for player consumption. For demonstration purposes, a piece of text generation software is available using browsers compatible with WebGL, at www.AlienGenerator.com [accessed 12 October 2021]. Although we have attempted to make this text generator consistent and easily readable, we stress that it is purely to demonstrate the algorithm's varied output and not intended as final player ready. To utilise the algorithm involved a few key steps: first, a scale of 'closeness to how humans experience the universe' was defined (1 being very close and 5 is far from it) to help us create potential node states and associated scale number (e.g., the state 'earth-like planet' would use 1, as it is very close to the human experience of the universe). Defining nodes was the next step e.g., Planet of Origin has 5 possible states (though any number can be used), with an allocated number: earth-like planet = 1, water planet = 2, rocky planet = 3, gas planet = 4, and nebula = 5. In total, 15 nodes (see Figure 4) are used: planet of origin (1), variety of biomes (2), ubiquitous materials (3), development of the means of production (4), species physiology (5), primary sensory apparatus (6), larger or smaller than the typical human (7), consumption method (8), number of senses (9), social viability (10), if they wear clothing? (11), This acts as a variety amplifier involving additional garments (12).

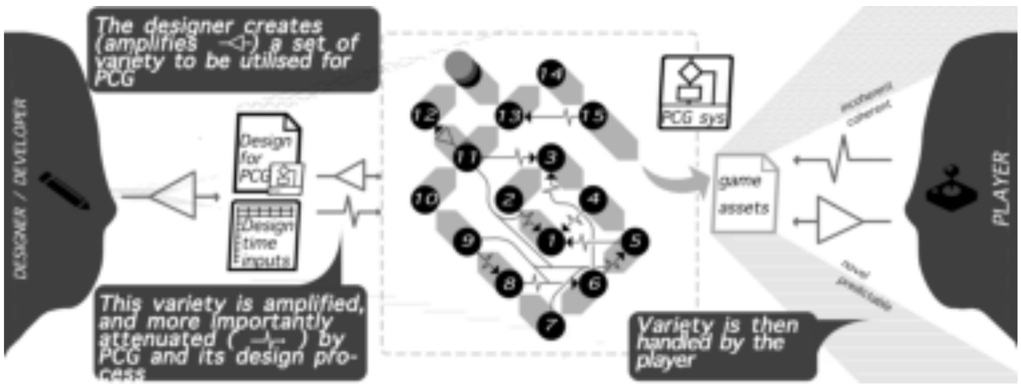

The designer creates (amplifies ◁) a set of variety to be utilised for PCG
DESIGNER / DEVELOPER
Design for PCG
Design time inputs
This variety is amplified, and more importantly attenuated ( ) by PCG and its design process
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
PCG sys
game assets
incoherent coherent
novel predictable
Variety is then handled by the player
PLAYER

Figure 4: Variety proliferation throughout procedural culture, considering designer input and player interpretation.

After this, the selection process moves onto homogeneity (13), recorded history age in human years (14), and rate of change (15). The arrows indicate the occurrence of variety handling apparatus that attenuate the scale choices. These nodes include a very small amount of civilization variety to allow us to start defining higher-level cultural nodes e.g., garments. By using Species Physiology, Planet of Origin, Rate of Change, Homogeneity, and the Development of the means of production for attenuation of this node a general description of garments worn by the fictitious species is created (e.g., garment decoration, functionality, or necessary for survival). With enough nodes, it will be possible to take this much further defining content such as colour palettes, e.g., based on how the society creates dyes, and materials with the most value e.g., for jewellery, etc.

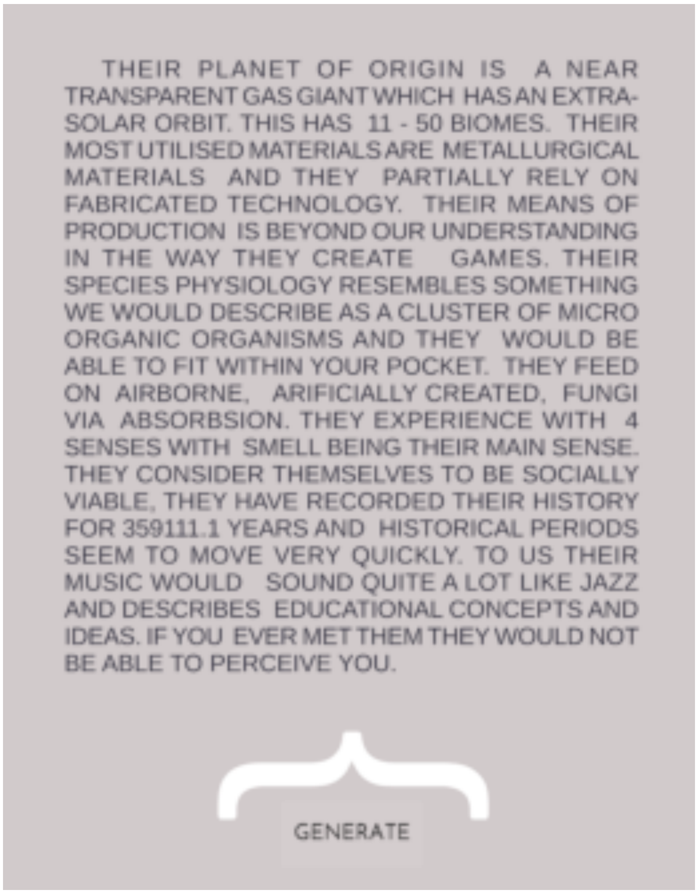

THEIR PLANET OF ORIGIN IS A NEAR TRANSPARENT GAS GIANT WHICH HAS AN EXTRA-SOLAR ORBIT. THIS HAS 11 - 50 BIOMES. THEIR MOST UTILISED MATERIALS ARE METALLURGICAL MATERIALS AND THEY PARTIALLY RELY ON FABRICATED TECHNOLOGY. THEIR MEANS OF PRODUCTION IS BEYOND OUR UNDERSTANDING IN THE WAY THEY CREATE GAMES. THEIR SPECIES PHYSIOLOGY RESEMBLES SOMETHING WE WOULD DESCRIBE AS A CLUSTER OF MICRO ORGANIC ORGANISMS AND THEY WOULD BE ABLE TO FIT WITHIN YOUR POCKET. THEY FEED ON AIRBORNE, ARIFICIALLY CREATED, FUNGI VIA ABSORBSION. THEY EXPERIENCE WITH 4 SENSES WITH SMELL BEING THEIR MAIN SENSE. THEY CONSIDER THEMSELVES TO BE SOCIALLY VIABLE, THEY HAVE RECORDED THEIR HISTORY FOR 359111.1 YEARS AND HISTORICAL PERIODS SEEM TO MOVE VERY QUICKLY. TO US THEIR MUSIC WOULD SOUND QUITE A LOT LIKE JAZZ AND DESCRIBES EDUCATIONAL CONCEPTS AND IDEAS. IF YOU EVER MET THEM THEY WOULD NOT BE ABLE TO PERCEIVE YOU.

Figure 5: Latest example output for the text generation program that is available www.AlienGenerator.com (requires WebGL) [accessed 12 October 2021]. This is an example project for generating procedural culture.

Reflection

The algorithm was easy to deploy for the text generator, (see Figure 5), and relatively easy to design for, although we found that nodes attenuated by more than one other node sometimes added more complexity depending on how much fine control over output was required/wanted. Fine control here, means control over the most likely and least likely/impossible outputs. For a high amount we needed to spend more time understanding individual connections for each attenuating node to guide outputs to what we considered to be intelligible/meaningful connections for player interpretation. Of course, with a huge amount of variety this approach becomes less feasible, and so, depending on requirements, and the amount of variety in any given system, we recommend focusing more on a loose approach, such as the contextualisation of the stochastic scale approach i.e., ‘closeness to how humans experience the universe’.

We found that the stochastic scale approach was excellent for the designing process, as it gave us a simple way to quickly determine each states associated scale number. Yet, it also had certain limitations in that some states were difficult to measure using this scale. We also found that higher numbers of potential states gave much better results, while insufficient states crashed the system as the nodes sometimes struggled to reach a positive result. Limiting the number of checks a node can make before randomly picking any available state helped to address this issue whilst also providing fewer desirable outputs. Ultimately, a mixture of fine control and likely fit worked well for us in our specific project, but we feel that the likely fit would create the most interesting results, even if they are not entirely believable/interpretable, this is where research into how we might convert generated information into different procedurally generated representational forms/narrativisation to help guide players in the interpretation process, could provide an interesting future research direction.

Procedurally Generating Fictional Character Behaviours

For the second case study, we deployed the algorithm in a video game, working title *Void* and currently under development. This is a single-player game whereby the player must collect objects to appease a mysterious entity whilst also trying to determine what type of entity it is. This entity can be one of several types selected at random at the start of each game, for example, a hostile alien. The entity's identity is not revealed until the end of the game once the player has met the win or lose conditions. The first step of implementation is determining what the entity wants for each specific round by using the algorithm to attenuate selection. We first set the node state (A) (see The Mathematical Method section), which represents the entity to 1 and for each of the potential objects that the entity could want, we set a corresponding number (B) which is based on the type of object of which there are 8 types. It is simple to design how likely entity types are to ask for specific object types from the player e.g., a hostile alien is more likely to want military-type objects determining those objects are set to 2 for a hostile alien. Conversely, a hostile alien might not be very interested in cultural object types determining that this is set to 5 for a hostile alien and making it unlikely that cultural objects would be selected for that entity's requirements.

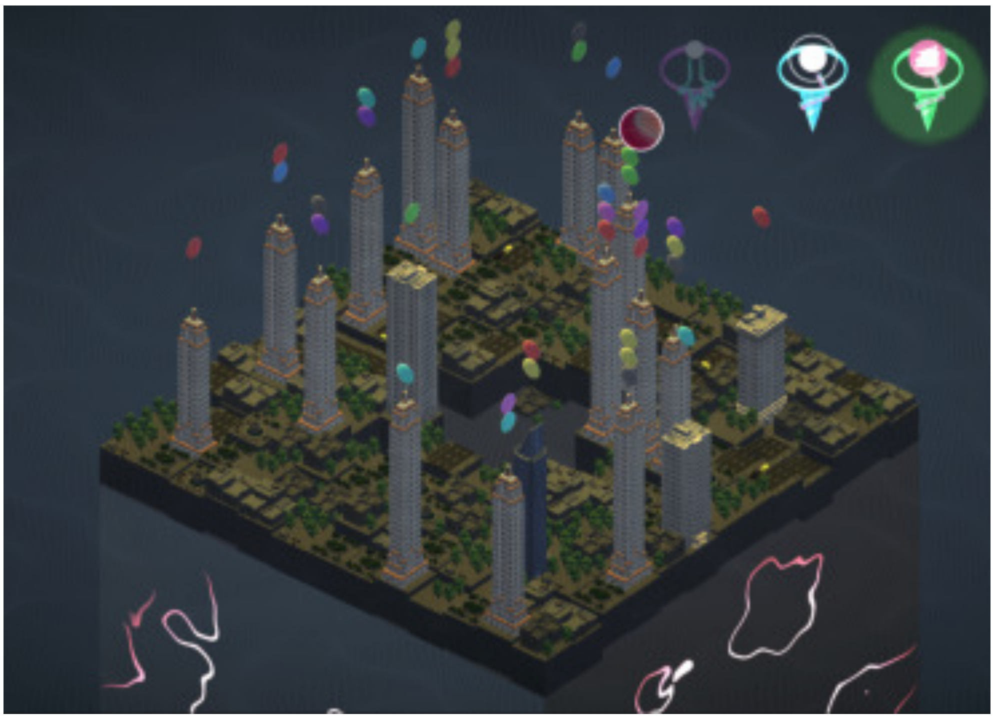


Figure 6: Void game screenshot showing (red) sphere entity representation.

In gameplay, the entity is represented as a moving and visually changing sphere (see figure 6) that follows the players as they move around the map. Its type influences its behaviours. A hostile alien has a higher probability of showing more aggression if the player fails to collect the required items. These cues support the player in making informed decisions and determining the type of selected entity. Behaviours are signalled by changes in the sphere's colour, sounds, movements, size, and texture. We manage all these behaviours using our *V*Machine Algorithm, which uses metrics such as entity type, the number of objects left to collect and how much time the player has left to collect them, to determine the entity's behaviours. Similar to the process for selecting objects, the entity's current state (A) is determined by these metrics and checked against a list of all possible behaviours (B). The first behaviour in the list to gain a positive result is then used as the entity's next behaviour.

Reflection

This approach was useful, allowing for easy implementation within the game's already-established state machine architecture. It provided a way to create semi-predictable behaviours for all entity types. This was ideal for our use case, as the entity needed to remain mysterious yet predictable enough to allow players to make informed decisions on its nature. By providing this entity with the possibility to act outside of a strict set of parameters per type, the game became more challenging. This is due to all available entity actions taken by any entity type being determined by probabilities based on the current state of the game. There were limitations in that, there was only a small variety of possible entity actions, and metrics on which selections were made. This created a situation similar to the previous case study, whereby, sometimes it would prove difficult for the algorithm to gain a positive result.

DISCUSSION AND CONCLUSION

We chose to focus on creating cultural signs systems for science fiction precisely because it demonstrates a variety problem: Thinking about extra-terrestrial life amplifies this challenge taking us immediately into the unknown. Before design even begins, possibilities for what can be generated for alien civilisations is limited by our inability to envision and comprehend a full variety of outcomes for what is in essence beyond our subjective experience as designers-developers. To be self-critical, this problem of procedural culture being limited by the subjective cultural grounding of its designers-developers warrants further consideration. Games are inherently ideological (Ash and Gallacher 2011; Longan 2008); while they may not be motivated by the explicit values, they are grounded in and reflect those of their designers-developers. If we are to create procedural culture that is meaningful and novel to marginalized people and reflects their values and experiences, having hegemonic normative initial variety in procedural culture is immediately problematic. This is where the creation of easily implementable and user-friendly, general content generators (Togelius et al. 2013) could be most valuable, allowing for a diverse set of designer-developers to easily input into procedural culture systems, and without the need for and access to advanced coding skills.

Ashby's law (1956) provides a universally applicable analytical and evaluative lens for the design of procedural culture, to promote understanding of the relationship between algorithm, setting, and audience perception. Any system that attempts multi-level multi-content PCG (Togelius et al. 2013) must navigate the tension produced by the player's ability to absorb variety and the system's ability to attenuate and amplify it for narrative coherence (Karhulahti 2012) —and novelty. As discussed, the latter is curtailed by the designer-developer's biases and values. Handling variety here should include understanding one's views and biases; when taking a cybernetics view, the system is only as good as our model for the system (Von Foerster 2003). As designer-developers it is important to be reflective and self-critical and seek to design procedural cultures which are inclusive of atypicality. For future work, exploration of how the generated content could be used for the creation of game assets is an important step. The exploration of how to enhance our ability to create meaningful connections for players is also very important, this might involve narrative generation, and understanding how assets are presented to the audience. Finally, exploration of creating open source, open access, simple to use procedural generation systems that utilise the *V*Machine Algorithm as a way of democratising and diversifying their inputs and outputs could yield interesting research, resulting in a more inclusive approach to procedural generation in mainstream games and beyond.

REFERENCES

Adams, Zach, and Tarn Adams. 2002. *Dwarf Fortress*. Mircosoft Windows. Bay 12 Games. http://www.bay12games.com/dwarves/.

Ash, James, and Lesley Anne Gallacher. 2011. 'Cultural Geography and Videogames'. *Geography Compass* 5 (6): 351–68.

Ashby, W. Ross. 1956. *An Introduction to Cybernetics.* New York: J. Wiley.

Aylett, Ruth. 1999. 'Narrative in Virtual Environments -Towards Emergent Narrative'.

Azad, Sasha, and Chris Martens. 2019. 'Lyra: Simulating Believable Opinionated Virtual Characters'. *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment* 15 (1): 108–15.

Balali Moghadam, Arman, and Marjan Kuchaki Rafsanjani. 2017. 'A Genetic Approach in Procedural Content Generation for Platformer Games Level Creation'. In . Iran, Kerman. https://doi.org/10.1109/CSIEC.2017.7940160.

Barros, Gabriella A B, Antonios Liapis, and Julian Togelius. 2016. 'Playing with Data: Procedural Generation of Adventures from Open Data'. *Proceedings of 1st International Joint Conference of DiGRA and FDG*, 16.

Barthes, Roland. 1977. *Elements of Semiology*. Farrar, Straus and Giroux.

Beer, Stafford. 1984. 'The Viable System Model: Its Provenance, Development, Methodology and Pathology'. *Journal of the Operational Research Society* 35 (1): 7–25. https://doi.org/10.1057/jors.1984.2.

Ben-Gal, Irad. 2008. 'Bayesian Networks'. In *Encyclopedia of Statistics in Quality and Reliability*. American Cancer Society. https://doi.org/10.1002/9780470061572.eqr089.

Bontchev, Boyan. 2016. 'MODERN TRENDS IN THE AUTOMATIC GENERATION OF CONTENT FOR VIDEO GAMES'. *Serdica Journal of Computing* 2: 133–66.

Coleridge, Samuel. 1817. 'From Biographia Literaria, Chapter XIV'.

Cook, Michael, Simon Colton, Jeremy Gow, and Gillian Smith. 2019. 'General Analytical Techniques For Parameter-Based Procedural Content Generators'. In *2019 IEEE Conference on Games (CoG)*, 1–8. London, United Kingdom: IEEE. https://doi.org/10.1109/CIG.2019.8848024.

Danesi, Marcel. 2013. 'On the Metaphorical Connectivity of Cultural Sign Systems'. *Signs and Society* 1 (1): 33–49.

Flick, Catherine, Andrew D. Reinhard, and L. Megan Denis. 2017. 'Exploring Simulated Game Worlds: Ethics in the No Man's Sky Archaeological Survey'. *The Orbit Journal* no. 2: 1–13.

Foerster, Heinz von. 2003. 'Cybernetics of Cybernetics'. In *Understanding Understanding: Essays on Cybernetics and Cognition*, edited by Heinz von Foerster, 283–86. New York, NY: Springer. https://doi.org/10.1007/0-387-21722-3_13.

Frontier Developments plc. 2014. 'Elite Dangerous'. 2014. https://www.elitedangerous.com/.

GCD. 2017. *Building Worlds in No Man's Sky Using Math(s)*. YouTube Video. https://www.youtube.com/watch?v=C9RyEiEzMiU.

GDC. 2019. *Math for Game Developers: End-to-End Procedural Generation in Caves of Qud*. YouTube Video. https://www.youtube.com/watch?v=jV-DZqdKlnE&t=422s.

Grinblat, Jason. 2015. *Caves of Qud*. PC. Freehold Game LLC. https://www.cavesofqud.com/.

Grinblat, Jason, and C. Brian Bucklew. 2017. 'Subverting Historical Cause & Effect: Generation of Mythic Biographies in *Caves of Qud*'. In *Proceedings of the 12th International Conference on the Foundations of Digital Games*, 1–7. Hyannis Massachusetts: ACM. https://doi.org/10.1145/3102071.3110574.

Harrell, D. Fox. 2005. 'Shades of Computational Evocation and Meaning: The GRIOT System and Improvisational Poetry Generation'. In *In Proceedings, Sixth Digital Arts and Culture Conference*, 133–43.

Hartsook, Ken, Alexander Zook, Sauvik Das, and Mark O. Riedl. 2011. 'Toward Supporting Stories with Procedurally Generated Game Worlds'. In *2011 IEEE Conference on Computational Intelligence and Games (CIG'11)*, 297–304. Seoul, Korea (South): IEEE. https://doi.org/10.1109/CIG.2011.6032020.

Hello Games. 2016. 'No Man's Sky'. 2016. https://www.nomanssky.com/.

Karhulahti, Veli-Matti. 2012. 'Suspending Virtual Disbelief: A Perspective on Narrative Coherence'. In *Interactive Storytelling*, edited by David Oyarzun, Federico Peinado, R. Michael Young, Ane Elizalde, and Gonzalo Méndez, 1–17. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer.

Karth, Isaac. 2019. 'Preliminary Poetics of Procedural Generation in Games'. *Transactions of the Digital Games Research Association* 4 (3). https://doi.org/10.26503/todigra.v4i3.106.

Kreminski, Max, and Michael Mateas. 2021. 'A Coauthorship-Centric History of Interactive Emergent Narrative'. In *Interactive Storytelling*, edited by Alex Mitchell and Mirjam Vosmeer, 13138:222–35. Lecture Notes in Computer Science. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-92300-6_21.

Kreminski, Max, and Noah Wardrip-Fruin. 2018. 'Sketching a Map of the Storylets Design Space'. In *Interactive Storytelling*, edited by Rebecca Rouse, Hartmut Koenitz, and Mads Haahr, 160–64. Lecture Notes in Computer Science. Cham: Springer International Publishing.

Longan, Michael W. 2008. 'Playing With Landscape'. *Aether: The Journal of Media Geograohy*, 18.

'Merriam Webster Dictionary'. n.d. Definition of ALGORITHM. Accessed 13 October 2021. https://www.merriam-webster.com/dictionary/algorithm.

'No Man's Sky Wiki'. n.d. No Man's Sky Wiki. Accessed 12 October 2021. https://nomanssky.fandom.com/wiki/Species.

People Make Games. 2020. *What Is 'Blaseball' and Why Is It Taking over the Internet?* https://www.youtube.com/watch?v=Y5t8DwnDE1k.

Ryan, James. 2014. 'DIOL / DIEL / DIAL'. Jamesryan.World. 2015 2014. https://www.jamesryan.world/projects#/diol/.

Salminen, Joni, Sercan Sengün, Haewoon Kwak, Bernard Jansen, Jisun An, Soon-Gyo Jung, Sarah Vieweg, and D. Fox Harrell. 2017. 'Generating Cultural Personas from Social Data: A Perspective of Middle Eastern Users'. In *2017 5th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW)*, 120–25.

Short, Tanya X., and Tarn Adams. 2019. 'Procedural Generation in Game Design'. Routledge & CRC Press. 2019. https://www.routledge.com/Procedural-Generation-in-Game-Design/Short-Adams/p/book/9781498799195.

'The Definitive Glossary of Higher Math Jargon | Math Vault'. n.d. Accessed 13 October 2021. https://mathvault.ca/math-glossary/.

The Game Band. 2020. 'BLASEBALL'. 2020. https://www.blaseball.com/.

Togelius, Julian, Alex J. Champandard, Pier Luca Lanzi, Michael Mateas, Ana Paiva, Mike Preuss, and Kenneth O. Stanley. 2013. 'Procedural Content Generation: Goals, Challenges and Actionable Steps'. Application/pdf, 15 pages.

Togelius, Julian, Tróndur Justinussen, and Anders Hartzen. 2012. 'Compositional Procedural Content Generation'. In *Proceedings of the The Third Workshop* *on Procedural Content Generation in Games - PCG'12*, 1–4. Raleigh, NC, USA: ACM Press.

Togelius, Julian, Georgios N. Yannakakis, Kenneth O. Stanley, and Cameron Browne. 2011. 'Search-Based Procedural Content Generation: A Taxonomy and Survey'. *IEEE Transactions on Computational Intelligence and AI in Games* 3 (3): 172–86.

Walsh, Richard. 2011. 'Emergent Narrative in Interactive Media'. *Narrative* 19 (1): 72–85.

Welker, Frank, Peter Cullen, and Corey Burton. 1984. *Transformers*. Animation, Action, Adventure. Sunbow Productions, Marvel Productions, Hasbro.