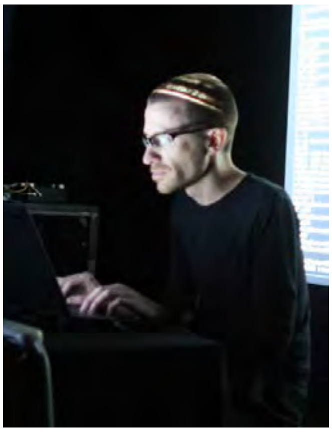
Renick Bell

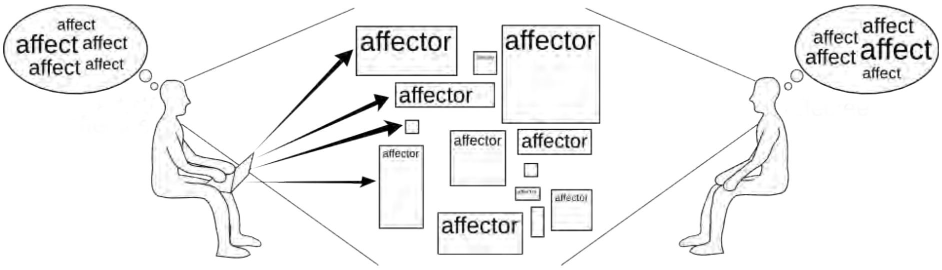
Paper: **Pragmatically Judging Generators**



Topic: Music, Aesthetics

Author:
Renick Bell
Tama Art University,
Tokyo, Japan
www.tamabi.ac.jp

Abstract:
Galanter defines generative art as art practice employing systems with some degree of autonomy to produce art works [4]. Brown describes live coding as a method for interacting with generative processes, clearly demonstrating its membership in the set of generative art practices [2]. McCormack, Bown et al. ask what characterises good generative art and draw attention to the processes involved as a determining factor for such evaluations [5]. The extent to which the generative aspect of live coding influences the aesthetic evaluation of such a performance can be examined using a pragmatic aesthetic framework. Such a framework, based Dewey's concept of an art experience [3], has been described by Bell [1]. Identifying the position of generative processes in the broader context of a live coding performance containing other important features may reveal some directions for aesthetic evaluations of generative processes in other domains. Through this pragmatic approach, improved experiences of live coding and other generative art works can be achieved.



An illustration of live coding as an art experience.

References:
[1] R. Bell, "Towards Useful Aesthetic Evaluations of Live Coding," in Proc. of the Int. Computer Music Conf., 2013.
[2] A.R. Brown and A. Sorensen, "Interacting with Generative Music through Live Coding," Cont. Music Review, vol. 28, no. 1, 2009.
[3] J. Dewey, Art as Experience. Perigee Trade, 2005.
[4] P. Galanter, "What is generative art?...," in GA2003 – 6th Generative Art Conference, 2003.
[5] J. McCormack, O. Bown, et al., "Ten Questions Concerning Generative Computer Art," Leonardo, Feb. 2013.

Contact:
renick@gmail.com

Keywords:
aesthetics, evaluation, live coding, pragmatism

Pragmatically Judging Generators

Renick Bell
Tama Art University, Tokyo, Japan
http://renickbell.net
e-mail: renick@gmail.com

Abstract

Galanter defines generative art as art practice employing systems with some degree of autonomy to produce art works [13]. Brown describes live coding as a method for interacting with generative processes, clearly demonstrating its membership in the set of generative art practices [5]. McCormack, Bown et. al. ask what characterises good generative art and draw attention to the processes involved as being a determining factor for such evaluations [14]. The extent to which the generative aspect of live coding influences the aesthetic evaluation of such a performance can be examined using a pragmatic aesthetic framework. Such a framework, based Dewey's concept of an art experience [12], has been described by Bell [1]. Through this pragmatic approach, improved experiences of live coding and other generative art works can be achieved. Identifying the position of generative processes in the broader context of a live coding performance containing other important features and how they are evaluated may reveal some directions for pragmatic aesthetic evaluations of generative processes in other domains.

1. Introduction

This paper briefly describes live coding as a form of generative art. It then describes a pragmatic aesthetic framework which is a revision of the aesthetic theory of John Dewey, the American pragmatist. That is followed by a summary of his theory of valuation. The paper then analyzes generative art and live coding in terms of the revised aesthetic theory. That approach is used first to discuss a section of a paper by McCormack, Bown, et. al. asking questions about "what characterises good generative art" [14]. It is then used in response to a critical analysis of the generative aspects of live coding written by Brown and Sorensen.

2. Live Coding as Generative Art

This paper defines live coding as the interactive control of algorithmic processes through programming activity, a definition derived from Brown, Collins, and Ward [4, 7, 18]. This paper will not consider programming in front of others as a tutorial or make a distinction between public performances and solitary coding. The relationship between this definition and generative art is made clear below.

The definition of generative art from Galanter is:

> “Generative art refers to any art practice where the artist uses a system, such as a set of natural language rules, a computer program, a machine, or other procedural invention, which is set into motion with some degree of autonomy contributing to or resulting in a completed work of art.” [13]

Though McLean and Wiggins write that live coding differs from generative art in that "generative art is output by programs unmodified during execution," [15], Galanter does not expect complete autonomy from generative art processes, just that the artist relinquishes at least partial control to the generative system.

Brown describes live coding as a method for interacting with generative processes, plainly classifying it as a member of the set of generative art practices [5]:

> “generative music in an expansive sense, where substantial musical outputs are produced by an algorithm... It is the ability to harness generative material that allows live coding performers to participate in a new kind of performance where they exercise indirect, or meta, control over the creation of their music.” [5:3]

Brown describes the performer's position as:

> “The performer is directly embedded within the algorithmic process and is free to guide and directly manipulate the unfolding of processes over time. The generative process exists on two levels, the playing out of the algorithmic potential of the code and the unfolding of the algorithmic opportunities and structural pathways held in the mind of the performer.” [5:7]

Another paper predating the Brown reference above "[advocates] the humanisation of generative music" [18:246]. That humanisation does not preclude the inclusion of live coding as a type of generative art.

Soddu's definition gets at the practical consequences of generative art: "construction of dynamic complex systems able to generate endless variations." [17] Live coding seems to fit this description well. This leads towards consideration of the aesthetics of generative processes in live coding, about which Collins writes that "... generative music is best appreciated when studied closely, when run many times..." and further asks "At a live concert, is generative music a music that says this time is special, now is privileged?" [6:71] These aesthetic questions are considered below.

3. Pragmatic Aesthetic Evaluation

The extent to which the generative aspect of live coding influences the aesthetic evaluation of such a performance can be examined using a pragmatic aesthetic framework. Such a framework, based on the pragmatic philosopher John Dewey's concept of art as experience [12], has been described by Bell [1]. With the intent of improving Dewey's theory, it has been revised by Shusterman [16] for general aesthetics and by McCarthy and Wright to explain interaction with technology [19].

This author presented a revision in [1] and a summary in [3]. Further revisions appeared in [2]. This version contains some additional minor revisions.

4. A Revised Pragmatic Aesthetic Theory

An affect is an emotional state. An affectee is a person experiencing affects in an interaction with affectors. An affector is a percept that stimulates affects in an affectee. A work of art is an affector which in some way was created, organized, or manipulated with the intention of it being an affector. A person involved with the creation or arrangement of an affector is an artist.

An art experience is the experience of affects in an affectee as the result of the affectee's interaction with a network of a potentially infinite number of affectors, with at least one of those affectors being a work of art. The affectors in the network influence each other and function directly or indirectly to stimulate affects in the affectee. The art experience is the experience of those affectors either simultaneously or in sequence. Changing the network of affectors changes the nature of the experience.

5. Dewey's Theory of Valuation

Dewey wrote a considerable amount of material on valuation. His theory of valuation can be summarized as follows. This summary first appeared in [2], but it contains minor revisions.

Value cannot be assigned in a disinterested manner [10]. Value is assigned to an experience according to the context of the experience (including but not limited to the culture it takes place in [10]). Such judgments are always in flux and susceptible to revision based on newly obtained experience.

The value of something derives from how well it suits the achievement of an individual's intentions and the consequences of achieving those ends through those means. The object of an appraisal is also evaluated while considering its consequences with respect to other intentions held by the individual [10].

Everything of value is instrumental in nature. Valuations themselves are instrumental for future valuations and action [9]. Every end is in turn a means for another intention in a continuous stream of experience. Valuations are used to control the stream of an individual's experience [11].

Relating this theory of valuation to the revised aesthetic theory above, it can be said that the value of an affector is connected to the value of an art experience in which it is involved. The value of an art experience is determined by the affects experienced [1] and how well those affects and the other consequences of the experience and its affectors suit the intentions of the affectee [2].

Simple steps for analyzing an experience through this theory are presented in [2].

6. Pragmatic Aesthetic Evaluation of Live Coding

It is useful to consider some aspects of live coding with regards to this theory.

Live coding has many affectors, such as the rhythm and timbre of the output sound, the sound diffusion system, the performance space, the contents of the projection, the programming language and tools used, and so on [1]. An important affector which is felt indirectly in every case and directly perceived in others is abstraction [3]. Generative processes are included in this network of affectors. A more complete exposition of affectors in live coding can be found in [1].

There are also many types of affectees. One simple classification groups them according to three criteria: whether or not they are programmers, musicians, or fans of live coding [1]. When experiencing generators, an awareness of a generator's role in a performance (or lack of it) and the affectee's knowledge modulate that experience.

Live coders possess a large variety of intentions, and the intentions of some can differ or even be in opposition to the intentions of others [2]. When evaluating generators, evaluation depends on the intention of the performer or the audience.

7. Breaking Down Generators into Component Affectors

Generative processes in an art experience have many aspects. In other words, they are compound affectors constituted of many affectors, including:

- the origin of underlying algorithm(s)
- the characteristics of the algorithm(s)
- a mapping of the algorithm to one or more synthesizers (audio in the case of live coding, but a visual or other synthesizer in other fields)
- the design of the internals of the process
- the implementation of the internals of the process, including its efficiency or elegance
- the design of the interface to the process
- the notation in code to express the process
- the degree to which the process has been abstracted and parameterized
- user interaction with the generative process
- the manner in which the generative process fits with other affectors involved in the experience

The necessity of considering all of these affectors in relation to the other affectors experienced follows from the theory above. This is supported by Cox, McLean, and Ward, who write that code should be evaluated both from its appearance as text and in the experience of it running [8].

8. What Makes Generative Art Good?

McCormack, Bown et. al. present a list of ten questions about generative art. One of those questions asks what characterises good generative art [14]. A dialogue providing answers to their questions through use of the theory above shows how it can be applied and might achieve the "more critical understanding of generative art" they say is needed. A selection from their questions and this author's responses follow, with comments related to live coding added.

> “Why is generative art in need of special quality criteria?” [14:9]

Proper consideration of the role of the generative process in the experience is needed. Because a generative process is a compound affector, consideration of all of its components is also necessary. It is also necessary to ask to what degree the affectee is aware of the generative affector and its components.

> “Is it better considered alongside other current practices?" [14:9]

Experiences exist in relation to one another. Past experiences influence present ones. In addition, generative elements appear alongside non-generative elements in every case, and the two have influence on one another. For example, in live coding sometimes audio samples are triggered as a result of a generative process. While the triggering is the result of the generative process, the audio may result from sound design efforts that may or may not rely on generative techniques. In the case that they do not, the practice of creating generative processes is being considered in conjunction with the practice of sound design. Thinking critically about the relationship between these two seems useful.

> “Consider two important properties that differentiate generative art from other practices. The first is that the primary artistic intent in generative art is expressed in the generative process. This process is what the artist creates, and as such should arguably be the subject of scrutiny in appreciation of what it produces.” [14:9]

This seems unnecessary. First, it may be difficult to define a "primary artistic intent" in some cases. Artists frequently possess a variety of intentions. In [2], a variety of intentions possessed by live coders is presented. It may be hard in many cases to choose just one as primary, and it is not certain that doing so is necessary or makes the work any better. For example, eating a meal at a fancy restaurant serves a practical intention of satisfying hunger, and it may also work towards various aesthetic intentions, such as enjoying exquisite flavors, appreciating an environment, engaging in stimulating conversation with friends, and so on.

Further, it seems conceivable that a generative technique might be used as a means towards an end that the artist gives higher priority. For example, consider a live coder whose primary artistic intent is making an audience dance. In this case, the intent to use generative processes is subservient to the primary intention of stimulating and maintaining a full and energetic dance floor.

Using a generative process is just one tool that an artist has for producing a work, along with a collection of other tools. An artist should have the freedom to select appropriate tools in every circumstance. It does seem appropriate, however, that the generative process and its output figure in the evaluation of the work as affector and consequently in the evaluation of the total experience.

> "Secondly, the way this process is interpreted or realised is also the locus of artistic intent, and is intimately intertwined with the first property. The basis of all generative art resides in its engagement with process. So the locus of artistic intent should include the motivations, design and realisation of the process..." [14:9]

While the generative process may or may not be the center of the artist's activity, this point recognizes that the generative process is actually a compound of several factors. Each factor plays a role in evaluation of the generative process and an experience of it.

> "Put simply, the "generative" and "art" parts are inseparable. Process in generative art should be considered the primary medium of creative expression, implying that the exclusive or predominant use of creative software or processes designed by others in one's generative practice is problematic." [14:9]

Calling the use of tools from others problematic is too strong. The total experience should be judged. For example, the use of a standard algorithm but mapped in an original way should still be able to cause affects of admiration of originality, surprise or novelty, or other positive affects. There seems to be no reason that employment of a generative process designed originally by someone else could not be used by another artist. It can be thought of as jazz sax players playing saxes that someone else has manufactured, or singers making use of songs from composers other than the singers themselves. The fact that a painter has not manufactured the paint in her painting is rarely a reason to evaluate the experience of the painting poorly.

> "Understanding an algorithm's subtlety or originality opens a fuller appreciation of the eloquence of a generative work. But this is a significant problem for most audiences, reinforced by focussing on the surface aesthetics of the art object as is often seen in computational generative art, where the computational process is rarely directly perceptible." [14:10]

Collins also notes this problem [6:69]. Any art experience is taken differently by different affectees. Knowledge of an area closely related to an affector changes the experience, but it is too much to ask that every affectee have working knowledge of all aspects of each affector. It is better to accept the knowledge that an affectee brings to the experience and allow that background to give them an authentic experience, even if it is a different experience from an affectee who is an expert. An artist can reveal the generative aspects of an art work, and that transparency can be good as long as it is in harmony with the intentions of the affectee (such as the artist). Providing enough information so that the audience can understand the generative process could be an intention of the artist, but it does not seem to be necessary. Collins suggests good program notes to increase the functioning of generative processes as an affector for audiences [6:68].

Games are one example of an artwork that has generative aspects which are not the main focus of the piece and in which the generative aspects only function indirectly in the experience of affectees. Live coding works similarly for affectees with relatively less knowledge of the means of live coding but possessing intentions such as immersion in electronic music or dancing.

9. Pragmatic Aesthetic Evaluation of Generators in Live Coding

Brown and Sorensen provide a detailed account of their experience with generators in live coding in [5]. Some discussion of those points follows.

> “... the way in which an algorithm is represented can impact upon its utility for the live coder. [5:6]

This certainly seems true. The artist experiences that representation directly, and other affectees in the audience it may experience it directly if that representation appears in the projection or experience it indirectly through its influence on the output sound or projection contents.

> “The description length and complexity of an algorithm plays a large factor in its appropriateness for live coding. Algorithms such as neural networks, evolutionary algorithms, agent based systems, and analytic systems are all affected by issues of description complexity. The longer the description of the algorithm, the more time will pass writing the code in which the programmer is unable to pay attention to other aspects of a performance.” [5:7]

It may then be advisable to use the generators in a generalized sense, meaning already abstracted and available as functions, and code around the parameterized aspects of the generator. The ability to code a generator from scratch could be one intention of a live coder, making this an important point. However, other intentions can make the approach of using an abstracted generator as a library function very effective. While the way that a process is implemented can be factor, it is not always the case that the implementation is the most pertinent affector for an affectee. In many experiences it can be almost invisible.

> “When programmers make a decision to abstract code away into a library, an abstract entity which can only be accessed as a ‘black-box’, the ramification is that they no longer have the ability to directly manipulate the algorithmic description.” [5]

Parameterization might mitigate this problem. A higher-level function can take functions as parameters, in which case some structure is fixed but other structure can be controlled by the programmer in a live setting. The general framework can be coded in advance, leaving a key component to be coded in a performance or to be selected from a body of pre-coded components. It also depends on how the library is accessed, since in some cases that code may still be malleable.

The flexibility of abstractions and code in this way appears as an affector for affectees with programming knowledge, and the resulting affects again are determined partially by intention.

> “Many grammars, pattern matching and analysis systems require a substantial amount of look-ahead for decision making and also often require the generation and scheduling of material into the medium to distant future. We have found these types of algorithms to be not very valuable in practice as they limit our ability to affectively respond to other concurrent processes, input devices and, most importantly, fellow performers.” [5:8]

It seems that some of these processes could be run as if they were non-realtime, that is given their targets in advance and allowed to generate their data silently. Once their output has been generated, the performer can select from or edit it. That data can used by another process. Naturally it would be less responsive to real-time interaction, but it may still be useful. Still, the opinion that Brown and Sorensen express reflects their unique experience of those algorithms. Others may find that such limits are impulses to other creative activity, leading to positive affects.

> “we have identified a set of algorithms that we have found particularly valuable... [including] probability, linear and higher order polynomials, periodic functions and modular arithmetic, set and graph theory, and recursion and iteration.” [5:8]

Here, Brown and Sorensen describe their positive experience with various techniques. It is worth noting that some affectees may lack necessary awareness that such techniques are being used for these affectors to work directly, though their results are certainly felt indirectly.

> “Many simple processes, such as repetition, can become tedious, while others, such as randomness, can seem featureless and uninteresting... This balancing of control and surprise is a constant challenge for generative sound artists and our experience suggests that at present it is better handled by the performer than by some computational ‘agent.’” [5:10]

Brown and Sorensen further describe their experience with various types of generators. It does seem that others might feel differently. Artworks which have been valued highly by many in the past have relied heavily on both repetition (Alvin Lucier's "I am sitting in a room.", for example) and randomness (such as Marcel Duchamp's "Three Standard Stoppages"). This shows the variablity of experience, partly due to the framing of that experience and differing intention.

Brown and Sorensen write that "generative processes should be:"

- "succinct and quick to type"
- "widely applicable to a variety of musical circumstances"

- "computationally efficient allowing real-time evaluation"
- "responsive and adaptive by minimising future commitments"
- "modifiable through the exposure of appropriate parameters" [5:1]

Brown and Sorensen derive actionable criteria from evaluating their experience, which seems productive.

The matter of being succint is up to the notation representing the generative process, rather than particularly being a factor of the generative process itself. Naturally, the larger the number of parameters that that abstraction requires can influence the amount of typing, but even one abstraction with a given number of parameters can be represented notationally in various ways, some of which are more concise than others. This relates directly to their last item, which is unavoidably in conflict with the first point. However, their point is correct in that frequently good abstractions are parameterized well to achieve maximum generality. This leads to their second point.

The width of application is also up to the performer. A generative process which is specific though frequently used may not require application to other cases in order to be evaluated positively. Proper parameterization will increase generality, but some abstractions may still remain fairly specific in their use-cases, which may still not be a factor which would cause a negative evaluation.

Modularity can reduce the commitment required to a generative process. If the process generates data that is in turn read and rendered by another process, then the user simply has to change what the rendering process refers to for its data. This could refer to the commitment required in the generation of the data itself. The need to change the direction of a particular generative process can be reduced if the processes are flexible enough to be discarded freely and instantiated in abundance, or so on. It is likely, though, that only the performer is likely to directly experience these affectors.

It is important to remember that generators can lead to a change in our intentions. Though a generative process may be conceived with a particular goal in mind, after becoming familiar with its output, intentions can change. What may have be imagined as a simple and boring test may be later considered to have interesting output, while an implementation of a generator that perfectly expresses the intention it was begun with may turn out not to have interesting output, causing a change of intentions.

9. Conclusions

Identifying the position of generative processes in the broader context of a live coding performance containing other important features may reveal some directions for aesthetic evaluations of generative processes in other domains. Through this pragmatic approach, improved experiences of live coding and other generative art works can be achieved.

10. Acknowledgements

I would like to thank Professors Akihiro Kubota and Yoshiharu Hamada at Tama Art University for research support.

11. References

[1] R. Bell, “Towards Useful Aesthetic Evaluations of Live Coding,” in Proceedings of the International Computer Music Conference, 2013.

[2] R. Bell, “Considering Interaction in Live Coding through a Pragmatic Aesthetic Theory,” in Proceedings of SI13, NTU/ADM Symposium on Sound and Interactivity, Nanyang Technological University, Singapore: , 2013.

[3] R. Bell, “Pragmatic Aesthetic Evaluation of Abstractions for Live Coding,” The Japanese Society for Sonic Arts, 2013 [Online]. Available: http://www.jssa.info/doku.php?id=journal017

[4] A.R. Brown, “Code jamming,” M/C, vol. 9, no. 6, 2007.

[5] A.R. Brown and A. Sorensen, “Interacting with Generative Music through Live Coding,” Contemporary Music Review, vol. 28, no. 1, pp. 17–29, 2009 [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/07494460802663991. [Accessed: 15-Sep-2013]

[6] N. Collins, “Generative Music and Laptop Performance,” Contemporary Music Review, vol. 22, no. 4, pp. 67–79, 2003 [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/0749446032000156919. [Accessed: 15-Sep-2013]

[7] N. Collins, “Live Coding of Consequence,” Leonardo, vol. 44, no. 3, pp. 207–211, 2011.

[8] G. Cox, A. McLean, and A. Ward, “The aesthetics of generative code,” in Proc. of Generative Art, 2001.

[9] J. Dewey, “Valuation and experimental knowledge,” The Philosophical Review, vol. 31, no. 4, pp. 325–351, 1922.

[10] J. Dewey, “Theory of valuation.” International Encyclopedia of Unified Science, 1939.

[11] J. Dewey, Experience and nature, vol. 1. DoverPublications. com, 1958.

[12] J. Dewey, Art as Experience. Perigee Trade, 2005.

[13] P. Galanter, “What is generative art? Complexity theory as a context for art theory,” in In GA2003 – 6th Generative Art Conference, 2003.

[14] J. McCormack, O. Bown, A. Dorin, J. McCabe, G. Monro, and M. Whitelaw, “Ten Questions Concerning Generative Computer Art,” Leonardo, feb 2013 [Online]. Available: http://dx.doi.org/10.1162/LEON_a_00533. [Accessed: 15-Sep-2013]

[15] A. McLean and G. Wiggins, “Live coding towards computational creativity,” in Proceedings of the International Conference on Computational Creativity, 2010, pp. 175–179.

[16] R. Shusterman, Performing live: Aesthetic alternatives for the ends of art. Cornell University Press, 2000.

[17] C. Soddu, “Generative Art International Conference.” 2013 [Online]. Available: http://www.generativeart.com/. [Accessed: 16-Nov-2013]

[18] A. Ward, J. Rohrhuber, F. Olofsson, A. McLean, D. Griffiths, N. Collins, and A. Alexander, “Live algorithm programming and a temporary organisation for its promotion,” in Proceedings of the README Software Art Conference, 2004.

[19] P. Wright and J. McCarthy, Technology as experience. MIT Press, 2004.