Discussing the Creativity of AUTOMATONE: an Interactive Music Generator based on Conway's Game of Life

Garrit Schaap
Department of Computer Science and Informatics
Jönköping University
Sweden
garrit.schapp@ju.se

Maria M. Hedblom
Jönköping AI lab
Jönköping University
Sweden
maria.hedblom@ju.se

Abstract

In recent years, generative AI systems for music composition have transformed not only music generation but the field of computational creativity as a whole. In contrast to the black-boxes of deep learning techniques, classic algorithms offer a transparent alternative to music generation that does not require training data and, due to the autonomous process, such systems could be argued to reflect a more genuine creative process. One such algorithmic system is cellular automata. Designed as grids of binary nodes, cellular automata use mathematical rules to transition between different states that can be used to generate music. The initial state and the particular transition rules allow different patterns to emerge which can be translated into musical compositions. In this paper, we introduce AUTOMATONE, a semi-interactive music generator based on the cellular automaton Conway's Game of Life. To ensure the quality of the music output, AUTOMATONE is based on pentatonic scales and uses four different state-transition systems to generate beats of different tempos.

Introduction

Music generation systems based on deep learning (DL) techniques (e.g. see Briot, Hadjeres, and Pachet (2020); Carnovalini and Rodà (2020) for overviews) have become an increasing sight not only in scientific communities like that of computational creativity (CC) but also in commercial settings (e.g. OpenAI's Jukebox (Dhariwal et al., 2020) and Google's MusicML (Agostinelli et al., 2023)). Today, auto-generated music systems can create interesting and aesthetically pleasing music with little to no inference from human users. Neural networks are trained on different musical styles and genres, and as a result, they can produce seemingly innovative compositions that replicate the characteristics of particular styles. Despite some remaining challenges with semantic content like lyrics or sophisticated rhythmic patterns, it is hard to diminish the artistic quality of musical products from generative AI. Thus, these systems fulfil Boden (1992)'s creativity criteria of being *novel and valuable*. However, such DL models are trained on a large dataset of human-made music that guides the creation of new compositions. Arguably, these compositions perform style reproduction of existing music and the 'creative' process exists as black-boxes without intentionality and explainability. This means, that even if such processes in some aspects could be considered creative, it is by no means possible to evaluate this using some classic methodology (e.g. Rhodes (1961)).

In contrast, traditional methods for music generation focus on designing algorithms that create music based on a set of generation and combination rules. While subject to less broad and 'generalisable results,' such methodologies are transparent and explainable.

One prototypical method to design musical rules with enough variability to be considered 'self-creative' while simultaneously relying on transparent and easy-to-understand rule systems, is *cellular automata* (CA) (Von Neumann, 2017). Originally introduced to model self-replicating systems, CA are state-transition systems in the format of grids (typically 2 dimensional) of binary cells. Following predefined rules, the cells in the grid transform into new state constellations based on the states of the neighbouring cells. One such rule system is Conway's Game of Life (GOL) which, simplified, simulate evolutionary reproduction.

In this paper, we introduce the music-generating system AUTOMATONE based on GOL and discuss its creative character within the setting of autonomous creativity.

Theoretical and Technical Foundation

Creative Products and Creative Processes

Classical definitions of creativity tend to focus on the output to be both novel and valuable (Boden, 1992). While this makes CC a straightforward endeavour, it has had the consequence that many systems that simulate creativity can focus exclusively on the product and ignore that the process of generating the products also influences the creative character of the system. For music generation systems it becomes of particular importance that the process generating the output also displays some level of creativity as music is experienced over time and not as a finished product. The importance of also considering the creative process has been highlighted by several researchers (e.g. Csikszentmihalyi, Abuhamdeh, and Nakamura (2005); Rhodes (1961) but it is not clear how the process of music generation could be evaluated in relation to 'good-old-fashioned' CC systems as well as generative AI systems respectively.

Despite the complexity, a few different models to investigate the creative process of the system have been proposed. Referred to as 'computational creativity desiderata' by Brown and Jordanous (2022)' such evaluation methods include: (Colton, 2008)'s tripod model, in which the system needs to display skill, appreciation and imagination; and (Jordanous, 2012)'s SPECS step-evaluation method that highlights the importance of defining creativity and using standardised tests.

Algorithmic Music Composition and Cellular Automata

Fundamental to human expression, music has been of particular interest to formally simulate. Due to the systematic patterns that are identifiable in (most) music genres, algorithmic composition traditionally focused on mathematical models, knowledge base systems, grammars and evolutionary algorithms (see Papadopoulos and Wiggins (1999) for a historic overview).

Here the prototypical stochastic method for music composition is to use Markov chains (e.g. Scirea et al. (2015); Shapiro and Huber (2021)). Based on probabilities from learned data, the music generation is based on a state-transition system in which the upcoming node depends on the previous one. Markov chains are particularly suitable for learning rhythms as they easily capture repetitive patterns. Naturally, as with DL models, Markov chains are trained on existing musical data. But in comparison, they limit the probabilities for the node generation to the previous states rather than the whole model.

From the perspective of evolutionary algorithms, cellular automata (CA) is a commonly used methodology for music generation (see examples in (Burraston and Edmonds, 2005; Miranda, 2001)). CA are discrete dynamical systems that act as state transitions over time intervals. In contrast to Markov chains, CA do not rely on probabilities or training data, but the state transitions are exclusively based on the current state. Created to simulate cellular evolution, CA typically consist of 2-dimensional grids where the cells can either be 'alive' or 'dead'. For 2D CA, there is a possible 256 different transition rules. Regardless of the chosen rule, each CA is given an initial state that due to the lack of implemented global trends, develops emergent behaviours. This means that music generation based on CA independently develops its own patterns that when translated into tones and chords create unique melodies.

The problem with these algorithms is that the degree of freedom and randomness of the system tends to be quite high. Consequently, the quality of the translated musical composition tends to be quite low. For this, it is important that the translation from CA cell grids into musical notes makes sense from the perspective of music theory.

AUTOMATONE: System Description

AUTOMATONE was built as an object-oriented program made in JavaScript with p5.js[1], and Tone.js[2] as the main coding frameworks. The latest version of the system is available via GitHub Pages[3] and the source code to the program is available on the corresponding GitHub Repository[4].

The System's Backbone: Conway's Game of Life

The baseline in AUTOMATONE is the auto-generated patterns extracted from CA, in particular, GOL that was introduced by John Horton Conway in 1970. The underlying idea of GOL was to create a complex 'organic' system following simple mathematical rules. The rules state that in a 2D grid, each pixel - a 'cell', can only have two states; 'dead' or 'alive'. Simplified, the rules state that 1) a cell 'dies' when it has fewer than two living neighbours or more than three living neighbours, and 2) is 'born' when it has exactly three living neighbours. And 3) for any setting between rules 1) and 2), the cell stays 'alive.' The initial state of the system can be random or fixed.

As a CA, GOL has some particularly interesting emergent features. Based on the format of the initial state, the transformation patterns can result in three different end states. First, they can fade away completely, either by overcrowding the boards or becoming too sparse. Second, the patterns can settle into a stable configuration without any further changes. Finally, it can enter into an oscillating phase in which the pattern repeats in an infinite loop.

There are known initial states that will produce a specific output, either still or oscillating that are of particular interest as they are known to develop particular patterns and emergent properties as the game develops. In AUTOMATONE, we ignore these pre-defined initial states for enhancing the creative process and for each initiation of music generation, the boards are given a randomised initial state.

AUTOMATONE consists of four different grid boards (see the different rows in Figure 1) that compose the basis of the music generation system. Each grid spans 8x8 cells that adhere to the logical rules of the GOL algorithm. As a human user, it is possible to manipulate the grids in real time to 'kill' or 'awaken' some of the cells in the different grids. That way, it is possible to alter the course of the generation process into new and unexpected patterns even if the system had ended up in one of the GOL end states.

Cells to Notes: Making Music with Pentatonic Scales

One of the most challenging aspects of making music from a cellular grid is to ensure that the binary cells correspond to something musically meaningful. Randomly assigning notes to the cells results in a cacophony of noise that hardly could be considered music and would fail on the creativity criterion of being perceived as 'valuable'.

In AUTOMATONE, we deal with the translation of grids into music in two ways. The first is to add a level of rhythmic complexity and the second is to ensure a harmonic output.

Rhythmic complexity: Each of the four grids has a counter, (see the small turquoise dot at the bottom in Figure

[1] https://p5js.org/

[2] https://tonejs.github.io

[3] https://pixelkind.github.io/automatone/

[4] https://github.com/pixelkind/automatone/

1) that dictates the beat of the particular grid by transitioning through the vertical lines in a step-wise fashion. From left to right, the speed increases. The slowest grid plays a note for 4 seconds, the second grid plays 4 notes at the same time, the third plays 8 notes at the same time, and the fourth plays 24 notes at the same time. This melodic beat provides depth to the generated music.

This is motivated by how in Western music, the typical beat is a 4/4 rhythm Powell (2010). Following musical convention, the first board is the dominant component of the melody, followed by the second, third and fourth boards representing weaker beats. Each board's time is aligned to ensure that the beats enhance each other and do not cause any distortions in the melodies.

Harmonic tones: To deal with the 'quality' of the musical output it is possible to rely on musical scales to ensure that the generated combinations are within 'reasonable' creative limits. For this purpose, Pentatonic scales are particularly useful. Referred to as "the mother of all scales" Powell (2010) and commonly used in many types of music, pentatonic scales are musical compositions limited to five notes per octave ordered based on musical keys (eg. C, D, E, G and A).

The most useful thing about the pentatonic scales in relation to auto-generated music is that their members have a basic mathematical relationship to one another. This means that they form an excellent self-sufficient group that can either be played simultaneously to provide harmony or be played in a melodic sequence to generate a melody. For AutomaTone, this means that the pentatonic scales are used both as a melodic component as part of the sequential steps that each of the boards goes through (cf. the turquoise dot), as well as a harmonic component in the way that they are played simultaneously as the cells are activated in the individual boards as well as over the four boards.

Practically, this means that each board is a tone matrix (TM) with a pentatonic scale, its own synthesizer connected with it and its individual speed. In the boards, the pentatonic scale goes from top to bottom with a play head that plays one column at a time.

Since we are using a 8x8 TM and have only five notes per octave in a pentatonic scale, we combine different octaves for each TM to fill it up (eg. C3, D3, E3, G3, A3, C4, D4, E4).

AutomaTone: The Complete Picture

AutomaTone starts with four different GoL systems. Every system has its own board that consists of an 8x8 grid of cells that represents a single instance of a GoL simulation. Vertically the rows are assigned to a specific 'note', while horizontally, the columns define the time dimension. During setup, every cell gets randomly assigned a state, either dead or alive. Before the system starts, the user has the opportunity to manipulate the initial input for every board. Every board has its individual speed, tone scale, and synthesizer. In the initial setup, the 'slowest' board, plays the deepest notes, while the 'fastest' board plays the highest notes.

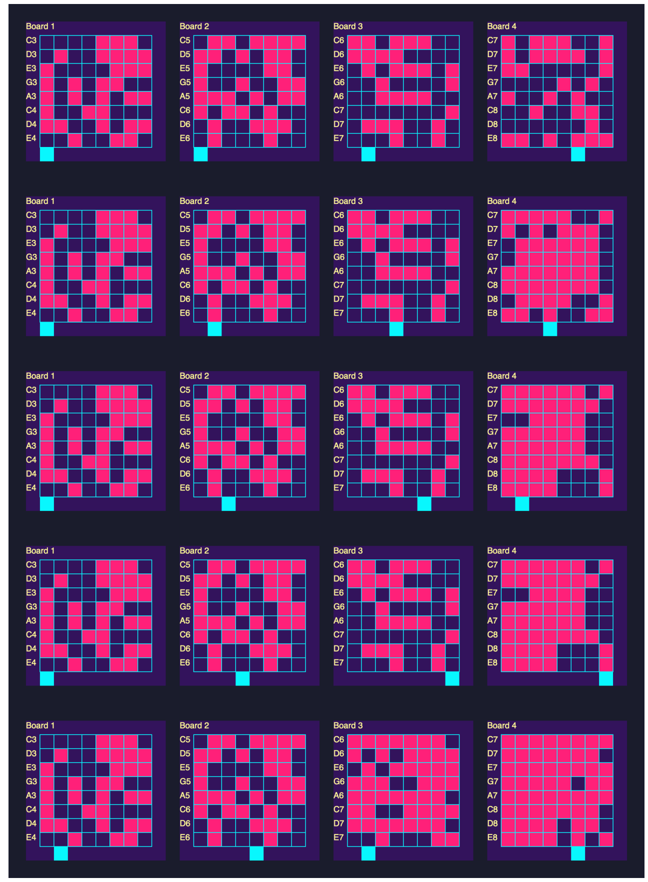

Figure 1: AutomaTone generating music over 5 seconds. Each row represent a new second for each of the boards.

A board consists of a two-dimensional array of cells and a reference to a synthesizer. In the current system implementation, one synthesizer is used for the first board, while the other three boards share another synthesizer.

Technically, every cell is treated as an individual object with the ability to: 1) know its state, 2) alter its state (alive/dead), 3) play the tone that is connected to its position in the grid, 4) assess if it has been manipulated by a user and transform accordingly.

System Demonstration

The interface of AutomaTone can be accessed through a web browser on a desktop computer with the mouse as an input device for optional manipulation of the cells (you can run it on Github[5]). The rows in the four grids are labelled according to the note and the octave with respect to the pentatonic scale. The columns represent the time dimension and a turquoise square at the bottom shows which of the columns are currently played.

In Figure 1, the music generation over 5 seconds is demonstrated.

[5]https://github.com/pixelkind/automatone/

Discussion and the Bigger Picture

Positioning AUTOMATONE in Music Generation

Despite being a unique application, the foundation of AUTOMATONE is not a novel approach as CA has been used for music generation for decades. One of the most famous usages is perhaps avant-garde composer Iannis Xenakis, famous for using cellular automata in part of his compositions Solomos (2005). While the elementary algorithms for CA are of a finite set, there is a wide variety in which the method can be used to translate into music.

Developed in stages over several decades, the work by Miranda (e.g. Miranda (2001)) is perhaps one of the most long-lived and scientifically impactful projects on using CA for music generation in different ways. Building on Conway's GOL, the music generator CAMUS translates the grids into music based on a coordinate system of three notes. Similarly, Morris and Wainer (2012) also uses Conway's GOL to generate music, but here the translation from cells to music is done by mapping a Chord Progression Model directly onto the cell spaces.

While it is impossible to compare the quality of the musical output, the main difference between our method and these two examples is that our method simultaneously uses four TM grids of different tempos to create a beat of some sort. This multidimensional component enriches the music's complexity in a way that easily can be transposed to represent different musical instruments. Simultaneously, our harmonics are based on pentatonic scales which ensures that the nodes played are compatible with one another.

Another noteworthy thing is that due to the simplicity and transparency of the logical rules underlying the methodology, it is not only possible to combine CA with a range of musical translation methods. It is also possible to combine CA music generation with a plethora of other computational methodologies - further improving the output and the complexity. For instance, in Delarosa and Soros (2020), a neural network was trained on the 256 elementary CA rules and MIDI files to be able to generate music. Likewise, in Lo (2012), machine learning fitness models were used to guide the CA music generation. In Arshi and Davis (2017), the authors combine CA with the stochastic neural networks Boltzmann machines for generating Persian music. Further, in Phon-Amnuaisuk (2010) heterogeneous cellular automata (hetCA) were used in combination with time-delay neural networks to generate chorale melodies in the style of Bach. Finally, Miranda and Miller-Bakewell (2022) investigated the compatibility of using CA with quantum computing. Each of these studies demonstrates the complexity and scientifically interesting application area of using CA for music generation.

How Creative is AUTOMATONE?

One of our main claims is that due to the transparency of the process behind the music generation of AUTOMATONE, it is worthy to be considered a creative system in its own right. From the perspective of Colton (2008), a creative system is required to display skill, appreciation and imagination. While it is possible to (in a liberal sense) argue for both skill and imagination, we have not provided the system with any level of self-assessment worthy of the epithet of 'appreciation'. Following Jordanous (2012)'s SPECS evaluation, we have defined creativity as being of equal parts a novel and valuable product and displaying a creative process. This means that the system must be considered creative. However, we have yet to conduct empirical investigations as to whether this impression of the perceived creative ability extends to other than us designers of the system.

Diverging from creativity assessments in the CC domain, we look to Rhodes' Rhodes (1961) classic four-component analysis of creativity: *Product, Person, Process* and *Press*.

As for the *product*, the music is novel and (we think) quite pleasant to listen to, fulfilling Boden (1992) requirement. As for the *person*, while the system lacks an embodied 'self' highlighted to be of importance for creative ability Merlini and Nicoletti (2020), the system can together with the human user autonomously produce an infinite number of musical compositions without any requirement for training data. As for the *process*, the transformation rules of GOL might seem deceptively simple but combined with the method by which the cells are turning into musical tones, it resembles the generation of electronic music. Arguments we think are in support of the creative character of AUTOMATONE. The final component *press*, requires a reception of an audience and we leave it to future work to evaluate this component (see below).

Conclusion and Future Work

In this paper, we introduced AUTOMATONE, a generative music system based on CA theory that acts as a partially interactive music generation system. Based on Conway's game of life, AUTOMATONE's algorithms can generate an infinite number of musical melodies comparable to the melodies generated by generative AI's methods. In comparison, AUTOMATONE is entirely transparent and explainable, and there is no 'artistic theft' involved in the training as the quality of the melodies is based on music theory, in particular, pentatonic scales. Further, due to the compositional design of the system, AUTOMATONE generates four different melodies at different beats to provide depth and sophistication to the generated music.

AUTOMATONE is still in an early stage of development in which we have several planned directions for improvement, among others *technical* and *evaluative*.

As for technical improvements, due to the compositional design of the system, it is possible to alter AUTOMATONE to simulate multiple different instruments. By utilising and expanding on the existing TM boards and assigning them the sounds of different musical instruments, AUTOMATONE would be able to create increasingly complex and sophisticated music pieces by simulating bands and orchestras.

As for evaluative assessments of the system's usefulness, aesthetic value and creative character, our initial plan is to conduct an on-site study at ICCC. As AUTOMATONE is a music generator with an emphasis on a transparent process, any such study is required to have a live audience.

Author Contributions

Garrit Schaap developed the system and co-authored the paper. Maria M. Hedblom supervised the implementation and composed the main body of the paper.

References

Agostinelli, A.; Denk, T. I.; Borsos, Z.; Engel, J.; Verzetti, M.; Caillon, A.; Huang, Q.; Jansen, A.; Roberts, A.; Tagliasacchi, M.; et al. 2023. Musiclm: Generating music from text. *arXiv preprint arXiv:2301.11325*.

Arshi, S., and Davis, D. N. 2017. Capturing the dynamics of cellular automata, for the generation of synthetic persian music, using conditional restricted boltzmann machines. In *Artificial Intelligence XXXIV: 37th SGAI International Conference on Artificial Intelligence, AI 2017, Cambridge, UK, December 12-14, 2017, Proceedings 37*, 72–86. Springer.

Boden, M. 1992. *The Creative Mind*. London: Abacus.

Briot, J.-P.; Hadjeres, G.; and Pachet, F.-D. 2020. *Deep learning techniques for music generation*, volume 1. Springer.

Brown, D. G., and Jordanous, A. 2022. Is style reproduction a computational creativity task? 220–229.

Burraston, D., and Edmonds, E. 2005. Cellular automata in generative electronic music and sonic art: a historical and technical review. *Digital Creativity* 16(3):165–185.

Carnovalini, F., and Rodà, A. 2020. Computational creativity and music generation systems: An introduction to the state of the art. *Frontiers in Artificial Intelligence* 3:14.

Colton, S. 2008. Creativity versus the perception of creativity in computational systems. In *AAAI spring symposium: creative intelligent systems*, volume 8, 7. Palo Alto, CA.

Csikszentmihalyi, M.; Abuhamdeh, S.; and Nakamura, J. 2005. Flow. *Handbook of competence and motivation* 598–608.

Delarosa, O., and Soros, L. B. 2020. Growing midi music files using convolutional cellular automata. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, 1187–1194. IEEE.

Dhariwal, P.; Jun, H.; Payne, C.; Kim, J. W.; Radford, A.; and Sutskever, I. 2020. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*.

Jordanous, A. 2012. A standardised procedure for evaluating creative systems: Computational creativity evaluation based on what it is to be creative. *Cognitive Computation* 4:246–279.

Lo, M. Y. 2012. *Evolving cellular automata for music composition with trainable fitness functions*. Ph.D. Dissertation, University of Essex.

Merlini, M., and Nicoletti, S. M. 2020. Of flesh and steel: Computational creativity in music and the body issue. *INSAM: Journal of Contemporary Music, Art and Technology* 4:24–42.

Miranda, E. R., and Miller-Bakewell, H. 2022. Cellular automata music composition: From classical to quantum. In *Quantum Computer Music: Foundations, Methods and Advanced Concepts*. Springer. 105–130.

Miranda, E. R. 2001. Evolving cellular automata music: From sound synthesis to composition. In *Proceedings of 2001 Workshop on Artificial Life Models for Musical Applications*.

Morris, H., and Wainer, G. A. 2012. Music generation using cellular models. In *Proceedings of the 2012 Symposium on Theory of Modeling and Simulation-DEVS Integrative M&S Symposium*, 1–8. Citeseer.

Papadopoulos, G., and Wiggins, G. 1999. AI methods for algorithmic composition: A survey, a critical view and future prospects. In *AISB symposium on musical creativity*, volume 124, 110–117. Edinburgh, UK.

Phon-Amnuaisuk, S. 2010. Investigating music pattern formations from heterogeneous cellular automata. *Journal of New Music Research* 39(3):253–267.

Powell, J. 2010. *How music works: The science and psychology of beautiful sounds, from Beethoven to the Beatles and beyond*. Hachette UK.

Rhodes, M. 1961. An analysis of creativity. *The Phi delta kappan* 42(7):305–310.

Scirea, M.; Barros, G. A.; Shaker, N.; and Togelius, J. 2015. Smug: Scientific music generator. In *The Sixth International Conference on Computational Creativity*, 204–211.

Shapiro, I., and Huber, M. 2021. Markov chains for computer music generation. *Journal of Humanistic Mathematics* 11(2):167–195.

Solomos, M. 2005. Cellular automata in xenakis's music. theory and practice. In *International Symposium Iannis Xenakis (Athens, May 2005)*, 11–p.

Von Neumann, J. 2017. The general and logical theory of automata. In *Systems Research for Behavioral Science*. Routledge. 97–107.