2017.xCoAx.org

Lisbon
Computation
Communication
Aesthetics
& X

0—255

SUSANA SANCHES
susana.sanchesp@gmail.com

Lisbon, Portugal

Abstract

0—255 is an interactive installation that explores the role and meaning of human execution in the enactment of algorithmic artworks as participative aesthetic events. It proposes the use of human interpretation in order to understand, experience and perceive its expressive potential within rules-based systems. It aims to engage the audience in procedurally reversing simple algorithms that have been investigated within computer space, back onto the physical space. This approach follows an on-going research that approaches strategies analogous to both real and artificial systems, aiming to contribute to an understanding of software code as a creative

Keywords
Cellular Automata
Computation
Interactivity
Procedurality
Reversal
Simulation

1. BACK ONTO THE PHYSICAL SPACE

Janet Murray (1997) defines procedurality as the computer's "defining ability to execute a series of rules". This term points to the formalization of abstract processes, which we call algorithms (as treatable procedures or methods); abstractions, which can be considered independently from both programming languages, and the machines that execute them (Goffey 2008:15-16). In the arts, this notion took shape in computational terms as new media artists started to think in terms of both human and code's performativity, and their combined role in the enactment of the artwork.

In line with this view our research develops along the idea of procedural simulation and its subsequent reversal. As *simulation* we consider the process of formalization of real world phenomena made according to standardized "digital data structures" (Berry 2008); and by reversal we assume the process of translation of "phenomena based on certain laws that have been investigated within computer space" back onto the physical space (Miwa 2007). Focusing on the latter, we explore human performance as a computational agent, assuming the "open gaps" inherent to the reversal process from the virtual to the physical realm. We consider this an important variable in the enactment of an artwork, as human's subjective interpretation may lead to the emergence of behaviors that generate novelty and unpredictability at each execution.

Our first approach to this concept was *Simulate-Reverse Play* (Sanches et. al. 2014),[1] an interactive installation that considers a type of play that emerges from the simulation and reversal of a set of procedures inspired by the *Game of Life's* (Conway 1950) algorithm. Presenting two layers that combined both real and virtual dimensions communicating in an interdependent feedback loop, this project allowed us to explore the creative potential of code when extended outside the computer. Based on these same overall guidelines we developed *0—255* as a second stage of this research.

2. CONCEPT

0—255[2] is a project about procedural simulation and its subsequent reversal back onto the physical realm by means of human mediation. It explores an understanding of code as a conceptual notation that conflates with execution. It resorts to human interpretation in order to understand, experience and perceive the "translation quality" of code from human-readable delegated code to machine-readable prescriptive code (Berry 2008), and vice-versa.

The main idea underlying this project is the notion of computation and the effects of code's actualization process. Following Stephen Wolfram's approach to "computational irreducibility" in the evolution of both computational and natural systems, it explores Cellular Automata (CA) as "simple computer programs", analog to the complexity and unpredictability in life, nature, and the "apparent freedom of human will" (2002, 637-750). As simple examples of *simulation*, CA allow us to explore how algorithmic systems are "by no means limited to formal instructions for computers", as long as their rules "meet the requirement of being executable by a human being as well as by a machine" (Cramer 2002). Thus, through CA, we approach software code's inherent performative dimension both on a computational and human level—a quality that is emphasized whenever code is "enacted or actively performed anew" (Salter 2010, 26).

[1] *Simulate-Reverse Play* (SRP) is a project developed in 2014 under the MA in Communication Design and New Media at FBAUL (Lisbon, PT).

[2] *0—255* started being developed at V2_ La for the Unstable Media (Rotterdam, NL) where it was exhibited for the first time. This project was produced as part of the Summer Sessions Network for Talent Development residencies, in partnership with Associação Arquivo 237 (Lisbon, PT) in 2016. Teaser of the project's development: www.vimeo.com/186410587.

Consequently, this project explores the evolution of CA systems through an ongoing process of algorithmic simulation and its subsequent reversal. We propose an interactive exchange in which, just like in irreducible computation, "the only way [for the audience] to work out how the system will behave is essentially to perform" it's computation (Wolfram 2002, 750) along with the machine. In this sense, this work reflects on the difference between human and machine algorithmic execution, allowing the audience to explore its role as the enacting agent in algorithmic artworks. By assuming human interpretation as a variable in this process, we intend to reveal how the nuances of human execution—such as, time of reaction, focus, learning ability and interpretational and physical coordination—can be incorporated and become expressive within rules-based systems, playing an important role in the enactment and meaning of the artwork as a participatory aesthetic event.

3. IMPLEMENTATION

3.1. Algorithm

A CA consists in a grid of cells, each representing a 1-bit of memory that can be updated to a binary state of 0 or 1 (black or white). The system evolves based on its rules and initial conditions, as "at every step there is then a definite rule that determines the color of a given cell from the color of that cell and its immediate left and right neighbors on the step before" (Wolfram 2002, 24).

We decided to work with the 256 "elementary" rules, which are "by most measure the simplest possible" (Wolfram 2002, 60). They present all the possible values or configurations (ranging from 0 to 255) for 1 byte (8-bits); the smallest addressable unit of digital information. These are algorithms that have already been exhaustively tested within computer space, and are defined by Wolfram as containing all "the essential ingredients needed to produce even the most complex behavior" (2002, 62). Considering the process of reversal, we opted for this type of algorithms due to their simplicity, linear progression and the fact that, just like any other CA, "their behavior can readily be presented in a visual way" (Wolfram 2002, 24).

3.2. Outcome

The project consists is an interactive installation, representing, in a process of constant actualization, all the 256 possible combinations for elementary CA and its corresponding patterns of behaviour. The set up is composed by a light projection on a black wall/screen and a plinth sustaining a keyboard placed in front of the projected area. The layout is divided in two areas. On one side, the rules and two lines of 8 cells are displayed—the first corresponds to the machine computation and the second to the human participant's computation, inserted via keyboard. On the other side a grid is presented; a zoom out of the overall pattern of behaviour generated by the human and machine interdependent computations, as they alternatively respond to each other's input.

3.3. Interaction and Feedback

The audience is invited to choose and insert a number between 0 and 255 with the keyboard and to press "Enter". By doing this, the participant selects one of the 256 rules in display, a terminal opens and the computational process of the selected rule is initiated. The terminal presents the chosen decimal number and its conversion into its 8-bit binary form of 0s and 1s. The system's initial conditions are the visual representation of the generated binary number. The participant can either compute the rule starting from these initial conditions or, if there have already been previous computations in previous interactions, he picks up at the point where the last participant left the computation.

The machine gives the first input, the participant responds, and together they fill up the lines of squares alternatively. When the participant succeeds in following the rules and activates the right cell the system paints it white. Otherwise, when activating the wrong cells, the system paints it grey. This visual response gives the participant a hint of how the execution must be made. Due to its linear progression it is not possible for the participant to correct his past computations, as the system will interpret the grey cells as part of the new conditions with which it has to work; an error that may influence the evolution of the system giving space for new patterns to emerge.

The interaction is over when the automaton *dies* or when the participant reaches a level of "disengagement" (Costello, et al. 2005, 55). When this happens the system goes back to its initial state, displaying the 256 rules in choice. The last actualization of the computed rule is stored in the system, either completed or waiting for another participant to carry on its evolution.

3.4. First Experiments

The first prototype presented a similar system, having in addition a printer and a wooden square (30 x 30 cm). The audience was invited to insert a decimal number into the keyboard (between 0 and 255) that was graphically converted into a binary number represented as a pattern of black and white cells, defining the rules the participant had to execute. The rows were filled up alternatively—first by the machine, then by the human—and, once the grid was completed, the system printed a mapping of the moment of the interaction. It replicated the results of a long espoused image that was being captured by a camera, giving the audience a graphical representation of the overall performance of the computation.

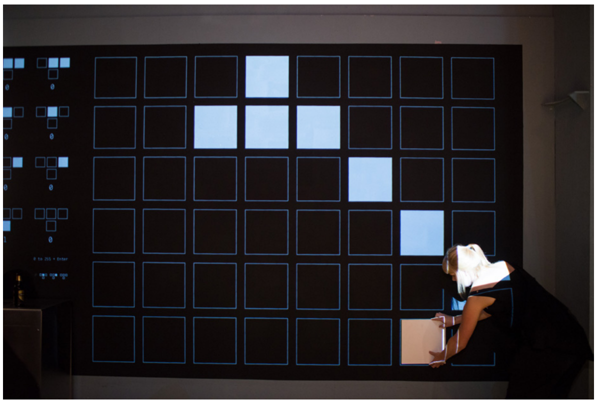
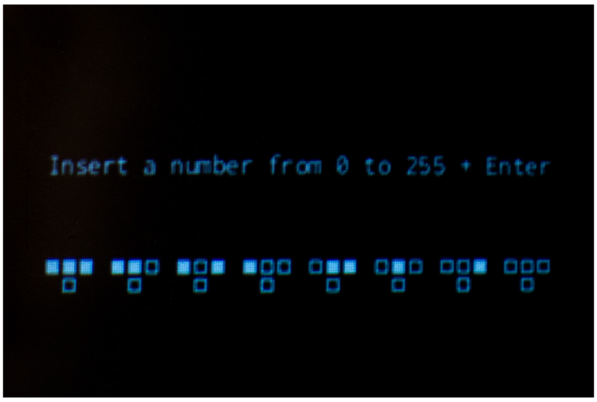


Fig. 1
Presentation at V2_
(08.09.2016).



Insert a number from 0 to 255 + Enter

Fig. 2
Presentation at V2_
(08.09.2016).

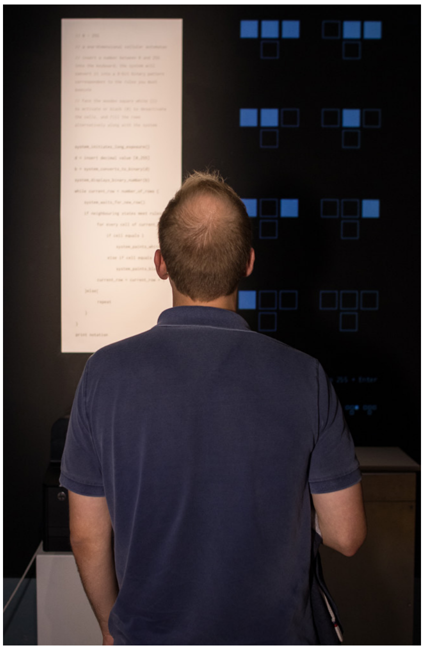


Fig. 3
Presentation at V2_
(08.09.2016).

3.5. Results

0—255 is a work in progress. In our first experiment we explored the performative experience of the audience in executing elementary CA rules. By seeking a playful approach that intended to recall certain emerging pleasures of play, as defined by Edmonds and Costello (2007, 79-80)—in particular the pleasures of *exploration, discovery, difficulty* and, eventually, the pleasure of creation associated with the pleasure of subversion—, we concluded that the complexity of the rules prevented the audience to reach a level of understanding of the system and of their enactive role in it. Consequently, in this second approach, we decided to simplify the legibility of the rules and bring to evidence the relations of cause and effect between the audience's actions and their effective results in the system's evolution. To do this, we decided to focus on the nuances of human interpretation in algorithmic execution, instead of human interpretation as corporeal performativity. This project aims to make reference on a conceptual and practical level to computation and its basic principles. By considering the procedural reversal of computational process, it proposes a representation that explores an understanding of code as a conceptual notation that conflates with human execution. In this sense it aims to bring to the fore a contemporary approach to conceptualism in relation to computing and coding, and establish analogies between human and artificial systems. And also propose an understanding of code as a creative medium not only inside, but also outside the computer.

REFERENCES

Berry, David M. 2008. "FCJ-089 A Contribution Towards a Grammar of Code." *The Fibreculture Journal* (13: After Convergence).

Costello, Brigid, and Ernest Edmonds. 2007. A Study in Play, Pleasure and Interaction Design. Sydney, Australia: DPP '07.

Cramer, Florian. 2002. Concepts, Notations, Software, Art. In *Software Art: Thoughts'*, edited by State Center for Museums and Exhibitions. Moscow: Rosizo: Read_me festival 1.2.

Goffey, Andrew. 2008. "Algorithm." In *Software Studies: A Lexicon*, edited by Mathew Fuller, 15-20. Cambridge, Massachussets: The MIT Press.

Miwa, Masahiro. 2007. "Reverse Simulation Music part 1/2: Reverse Simulation Music as a new musical methodology." http://www.youtube.com/watch?v=UX6vXiqpyJQ

Murray, Janet. 1997. *Hamlet on the Holodeck: The Future of Narrative in Cyberspace*. Cambridge, Massachussets: The MIT Press.

Salter, Chris. 2010. Entangled: Technology and the Transformation of Performance.

Sanches, Susana, Luísa Ribas, and Miguel Cardoso. 2014. "Procedural Simulation and Its Reversal Applied to Game Dynamics: Exploring Rules-Based Systems Through Play." xCoAx, 2nd Conference in Computation, Communication, Aesthetics and X, Oporto, Portugal.

Wolfram, Stephen. 2002. *A New Kind of Science*: Champaign, IL: Wolfram Media.