The Cargo Cult: Real Time AI Live Coding using Evolutionary Algorithms

Damian Dziwis[1,2] and Christoph Pörschmann[1]

[1] TH Köln - University of Applied Sciences
[2] Technical University of Berlin
damian.dziwis@th-koeln.de

Abstract. In this paper, we present the piece and the underlying software system of "The Cargo Cult" by Damian T. Dziwis. The piece is an audio-visual live coding performance using an autonomous programming machine learning (ML) agent in the context of contemporary music. The ML-agent is a genetic programmer system from the field of evolutionary algorithms. It is designed to live code in a proprietary and specially developed esoteric-like programming language, resulting in real-time audio-visual events presented to the audience. In this case, the usually used fitness function of the executing algorithm is replaced by a text-based, chat-like input interface. This creates a genuine human-computer interaction between the autonomous agent and the artist on stage, reflecting the theme of "cargo cults" emerging in society in the context of interaction with artificial intelligence systems. The piece was developed for the "die digitale" festival (Düsseldorf/Germany) in 2018.

Keywords: artificial intelligence, live coding, autonomous programming, machine learning, contemporary music, audio-visual art

1 Introduction

During World War II, by observing stationed Western soldiers being supplied with food, clothing, and all manner of goods by cargo planes, the indigenous people of Melanesia developed a religious cult. Hoping that the "new gods" would also reward them with all the goods falling from the sky, they began to ritually imitate the observed behavior - they began to imitate hand-moving land signals, lit signal fires, built meaningless runways, towers, and even carved wooden headphones with no functionality. This phenomenon was later called "cargo cult" and has also become a metaphor for (imitated) questionable symbolic actions with no conscious causal connection to expected outcomes. In our digital world, "cargo cults" are omnipresent in symbolic acts towards Big Data networks, search engines or social media algorithms. Always hoping that the new "digital gods" will reward us with a better user reach, likes, rankings or scores - be it through (independently) learned behavior patterns or following the prophecies of social media and SEO gurus.

"The Cargo Cult" stages this seemingly spiritual dialogue as an interaction with a self-programming artificial intelligence (AI), resulting in a live coding performance with audio-visual artifacts. While currently commonly used artificial neural networks (ANN) are based on pre-selected training data, resulting in generated results that are rather style imitations than genuine works, the approach taken here, instead is based on a machine learning (ML) model called genetic programming. These evolutionary algorithms allow the ML-agent in "The Cargo Cult" to be taught a specially designed esoteric programming language that allows the agent to program its own audio-visual patterns without projecting the creator's biases and ideas of art and music onto it.

2 Software Development

In 2017, Becker and Gottschlich presented with the "AI Programmer"(Becker & Gottschlich, 2017) a first-of-its-kind ML system based on genetic algorithms (GA). They demonstrate the advantages of genetic programming (GP) (Koza & Poli, 2005) using a reduced instruction set programming language, as common in esoteric programming languages, in this case *Brainf*ck* [3].
This concept of GP using an esoteric programming language is adapted as the underlying ML system in "The Cargo Cult". Unlike approaches of autonomous live coding using ANN systems (Navarro Del Angel & Ogborn, 2017; Stewart & Lawson, 2019; Stewart & Lawson, 2020), the use of GP allows for the creation of an ML-agent with minimal human input. The knowledge base (KB) of a GP system is not trained from human-generated examples, but rather the rules of the corresponding programming language are embedded in the algorithm. The system is then able to generate programs and solve problems on its own. Specifically on the artistic level, this approach helps decouple the AI-generated output from previously human-generated works and the stylistic/aesthetic as well as ideas and biases they contain. This decoupling leads to genuine AI works instead of variations or style limitations of previously trained human art.

Operator	Instruction
>	push next value from stack
<	push fromer value from stack
|	random choose of two values to put on stack
+	increments current value
-	decrements current value
⌜	copy current stack to stack
⌟	copy stack in reversed order to stack
⫤	randomly scramble the values in stack

Table 1. Instruction set of the designed language.

[3] `https://en.wikipedia.org/wiki/Brainfuck`, accessed: 2021-06-10

2.1 Genetic Programmer

A GP is a type of genetic algorithm (GA) from the evolutionary algorithm (EA) family. They are ML models with no previously trained data. A GP starts with no KB besides the rules and instructions of the corresponding programming language (PL). Instead, an initial set of values and operators, called the "seed", is randomly selected. This initial genome already represents a complete program in the PL. The program evolves in each generation by randomly mutating the current genome or applying a crossover to another genome. This leads to a tree-like structure of child and parent genomes representing a resulting program. As this process performs as-fast-as real-time, it can be presented as live coding to a audience. To guide the evolutionary process of mutation and genome crossover, usually a fitness test is used to evaluate how well a child generation performs. Because an artistic process is not given a specific "target" or an estimated outcome, it is not possible to calculate a fitness score by comparing a given outcome to an expected one.

2.2 Chat System

Instead of using a fitness function to guide the evolutionary process of the GP, the ML-agent in "The Cargo Cult" uses a chat interface to provide text-based feedback to the algorithm. In keeping with the artistic concept of the piece, the user sends messages to the ML-agent and expects a response in the form of a live coded program by the agent. The ML-agent has no semantic understanding of the user message, instead it substitutes a fitness value from it by analyzing the message. It looks for keywords like "continue, follow, more, less, try, etc..." to guide the evolution process. Other keywords like "hello world" or "build all" are used to trigger control sequences of the algorithm, e.g. initialization of a new DNA sequence and the like.

2.3 Esoteric Programming Language

For this piece, a specially designed language inspired by esoteric programming languages is created. Similar to esoteric languages like Brainf*ck, the rudimentary language developed here is stack-based and consists of a minimal instruction set with only 8 operators (see Tab. 1). The available data values in the language are characters from the higher unicode[4] spectrum (mostly from the Japanese alphabet, one Korean character): ;アエウイオトゲツゼピレニ
The interpreted program code results in sequences of numerical values that trigger patterns of events in the audio and visual engine.

2.4 Sound and Visuals Engines

The software system developed for "The Cargo Cult" consists of several implementations in different programming languages and environments (see Fig. 1).

[4] `https://home.unicode.org/`, accessed: 2021-06-10

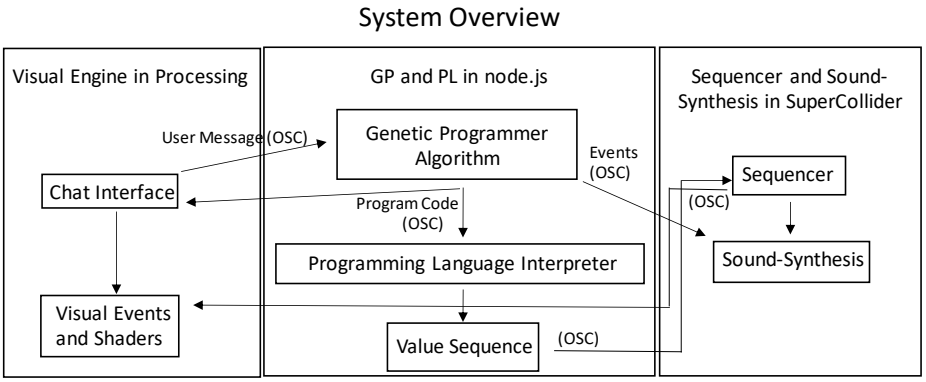

Fig. 1. The diagram is showing the system architecture used in "The Cargo Cult".

The GP algorithm and the embedded programming language interpreter are implemented in JavaScript using the Node.js runtime environment. They are connected to a visuals and sound engine via the Open Sound Control (OSC) protocol. The chat interface and the visuals engine are implemented in Java using the Processing framework. The visuals engine is used to render 3D objects and glitch effects programmed in the OpenGL Shading Language (GLSL). The sequences of data values generated by the GP are played by the sequencer in SuperCollider and create audio-visual patterns by triggering events in the sound synthesis engine and visual effects in Processing.

3 Artistic Concept

"The Cargo Cult" was commissioned for the "die digitale" festival (Düsseldorf / Germany) in 2018. The theme of the festival was "digital gods", so the piece was developed with the idea of reflecting "cargo cults" in society created by interaction with the "new digital gods" such as Big Data networks and other AI systems. The piece is staged as a dialogue with a binary goddess, with humans sending "prayers" in the form of text messages and the goddess responding with a live coded audio-visual world using an evolutionary algorithm. The resulting audio-visual world consists of various digital cult objects and artifacts. The overall setting engages internet phenomena in the context of "New Aesthetics"(Berry et al., 2012). The piece refers to digital stylistics such as Vaporwave, glitch art and Clicks'n'Cuts, as well as to technological artefacts such as the Nefertiti 3D heist[5], the Utah teapot[6] or samples from numbers stations[7].

[5] `https://alloversky.com/puzzlepieces/the-other-nefertiti`, accessed: 2021-06-10

[6] `https://www.computerhistory.org/revolution/computer-graphics-music-and-art/15/206/557`, accessed: 2021-06-10

[7] `https://www.numbers-stations.com/`, accessed: 2021-06-10

References

Becker, K., & Gottschlich, J. (2017). *AI Programmer: Autonomously Creating Software Programs Using Genetic Algorithms.*

Berry, D. M., van Dartel, M. D., Kasprzak, M., Muller, N., O'Reilly, R., & José, L. D. V. (2012). New Aesthetic, New Anxieties. *E-Publication, V2*.

Koza, J., & Poli, R. (2005, 01). Genetic programming. In (p. 127-164). doi: 10.1007/0-387-28356-0_5

Navarro Del Angel, L., & Ogborn, D. (2017). Cacharpo: CO-PERFORMING CUMBIA SONIDERA WITH DEEP ABSTRACTIONS. In *International Conference on Live Coding, ICLC* (pp. 1–12).

Stewart, J., & Lawson, S. (2019). Cibo: An Autonomous TidalCyles Performer. In *International Conference on Live Coding, ICLC* (pp. 1–9).

Stewart, J., & Lawson, S. (2020). Cibo v2: Realtime Livecoding A.I. Agent. In *International Conference on Live Coding, ICLC* (pp. 1–10).