



HAL
open science

Assessing the Support for Creativity of a Playground for Live Coding Machine Learning

Francisco Bernardo, Chris Kiefer, Thor Magnusson

► **To cite this version:**

Francisco Bernardo, Chris Kiefer, Thor Magnusson. Assessing the Support for Creativity of a Playground for Live Coding Machine Learning. 20th International Conference on Entertainment Computing (ICEC), Nov 2021, Coimbra, Portugal. pp.449-456, 10.1007/978-3-030-89394-1_38. hal-04144397

HAL Id: hal-04144397

<https://inria.hal.science/hal-04144397v1>

Submitted on 28 Jun 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



This document is the original author manuscript of a paper submitted to an IFIP conference proceedings or other IFIP publication by Springer Nature. As such, there may be some differences in the official published version of the paper. Such differences, if any, are usually due to reformatting during preparation for publication or minor corrections made by the author(s) during final proofreading of the publication manuscript.

Assessing the Support for Creativity of a Playground for Live Coding Machine Learning

Francisco Bernardo, Chris Kiefer, and Thor Magnusson

Experimental Music Technology Lab
School of Media, Arts and Humanities
University of Sussex, Brighton, UK
{f.bernardo, c.kiefer, t.magnusson}@sussex.ac.uk

Abstract. We present the ongoing research around the design of Sema, a live coding environment aimed at supporting live coding with machine learning in the modern web browser. Sema integrates custom dashboards with code editors, debugging and visualisation tools, reference documentation and interactive tutorials. We analyse survey findings applying the Creativity Support Index, which aimed at understanding how well Sema supports creativity across its subsystems, and discuss how the insights we obtained contributed to inform the following design iteration.

Keywords: Live Coding · Sound, Music & Performance · Machine Learning · Creativity Support Tools

1 Introduction

Live coding practitioners typically engage in real-time composition and performance by programming with a domain-specific language (DSL) and community-developed open-source tools for audio and visual synthesis, algorithmic creation and instrument design [7]. Such tools can be considered as Creativity Support Tools (CST), a class of tools designed to assist users in creative work that includes, for instance, environments for software development, mathematics, music production, animation and visualisation [9]. Designing complex applications to support creativity with highly-technical domain-specific workflows presents difficult challenges, nevertheless, particularly when assessing prototypes for informing future design iterations. We present an assessment of the creativity support of Sema, a playground for live coding music and machine learning (ML) in the Web, using the Creative Support Index [4] with a sample of workshop participants that comprised novices and expert live coders.

2 System Overview

Sema [1,2] is an open source live coding system—such as SuperCollider [5], Tidal-Cycles [6] or Gibber [8]—where code is used as the language and medium for real-time composition and performance. Sema aims to support different musical

practices and cultures, and empower users to shape the language to their own musical needs. It also aims to support users wanting to apply machine learning models that may generate interesting musical behaviour in their performance. Such models can be trained on small data sets interactively and in real time, or previously trained on big data sets and used through transfer learning.

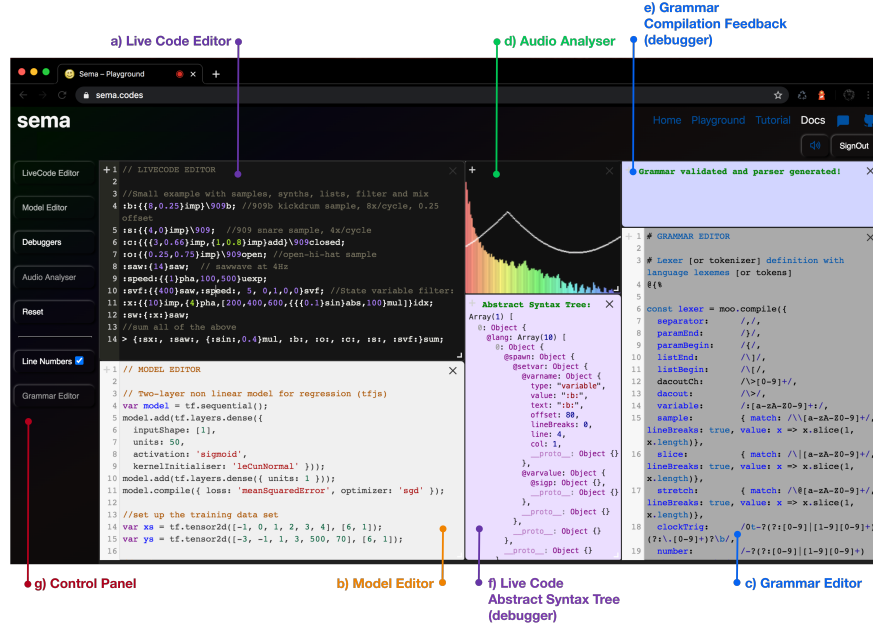


Fig. 1. Sema's playground GUI

Previous user-centred design iterations led us to integrate machine learning and language design workflows with a high-performance signal engine for real-time interactive audio in a low-entry-barrier environment available in the Web browser [1,2]. Sema integrates third-party ML libraries—e.g. Tensorflow.js [11] and Magenta.js [10]—that enable users to design and evaluate end-to-end ML pipelines. This includes creating training datasets, bespoke model architectures and hyper-parameters, and input and output channels that enable communication between the live coding languages and the signal engine. The user interface to ML is a code editor similar to Jupiter Notebooks where users can write and evaluate JavaScript code blocks in a non-linear fashion. The system provides machine learning exemplars for sound and music applications, including interactive machine learning for FM synthesis, reinforcement learning for sequencing beats, or pre-trained Magenta models for note sequencing. Sema also enables users to customise or create their live coding languages by inspecting, editing or

specifying the different components of a language grammar (i.e. lexer, grammar rules and semantic actions) using the Bachus-Naur Form metasyntax notation.

Sema’s user interface—version 0.5.0, used in the workshop (Figure 1)—allowed users to navigate between two main areas: the Playground and the Tutorial. In the Playground, users could access a dashboard of customisable layouts and a library of dynamic components—code editors, visualisers, and debugging tools—and engage in a free, open-ended exploration of live coding, language design and ML workflows. In contrast, the Tutorial area makes a strong proposition with a constrained-layout environment and content structured with chapters and sections with gradually increasing challenge and granularity. The live coding editor (Fig. 1–a) provided users with the main tool to interact with the live coding languages, key bindings for code evaluation and real-time audio playback, suspension and resume, code commenting and text search.

3 User Study

The overall goal of the mixed-methods user study was to evaluate the creativity support, usability and utility of Sema. This study succeeded a previous study which assessed the first design iteration of Sema through a week-long co-located workshop [2]. For the scope of this paper, we focus on creativity support and delimit the description of methods, results and discussion accordingly, while providing as much context about the workshop as it is deemed reasonable.

3.1 Workshop

The “Sema: Live Coding With Machine Learning Workshop” was as a two-week long online workshop in late June–July 2020, which was conducted as part of a mixed-methods user-centred design study. The workshop was advertised using a blog post¹ that was disseminated on social media and personal contact networks two weeks before the event. The blog post described the target audience of beginners, with beginner-level JavaScript programming skills, and without required experience in Sema, machine learning, language design or music. The blog post described some of the affordances of Sema in live coding, language design and machine learning, and provided demonstration videos of live coding with three languages in Sema. Additionally, it described the workshop programme, with schedule and topics to cover, and included details about the registration and a link to the registration form.

In the first week of the workshop, we employed a flipped-learning approach [3] where, every day, we released introductory tutorial videos and provided two-hour long synchronous Zoom sessions (one hour of workshop tutorial followed by an hour of Q&A) and asynchronous Slack channel support. Participants would explore Sema and the instructional content on their own individual time. The

¹ Sema: Live Coding With Machine Learning Workshop <http://www.emutelab.org/blog/semaworkshop>

second week of the workshop switched to hands-on project development, with three synchronous Zoom sessions and continued asynchronous Slack channel support over the week for people developing their projects.

The workshop was designed to give us data for further development of the project. For data collection, we employed several techniques that yielded a very substantial and rich primary data set comprising:

1. a record with all the interactions between workshop participants and facilitators in the Slack channel, with live code samples produced and shared by participants,
2. video recordings of the Q&A sessions and of semi-structured interviews conducted with a selection of 10 participants at the end of the workshop,
3. an online survey that included sections with the Creativity Support Index completed by 23 participants by the end of the workshop.

The full analysis of all the elements of this data set is beyond the scope of this paper and will be published separately. Here we focus our analysis on the last element of the data set, the results of the Creativity Support Index.

3.2 Creativity Support Index

The Creativity Support Index (CSI) is a survey designed for the psychometric evaluation of the support for creativity of digital tools [4]. The CSI enables researchers to understand how well a tool supports creative work and what aspects of creativity support need improvement, through a quantitative assessment of six dimensions of creativity support—Exploration, Expressiveness, Immersion, Enjoyment, Results, Worth Effort, and Collaboration.

The CSI was administered to the participants at the end of the workshop, as part of an online survey in the Qualtrics platform. A preface survey section comprised a summary of the study goals, the participant information sheet, and a consent form that conditionally unblocked access to the remainder of the survey. Part A of the survey assessed participant demographics, background and experience on live coding, machine learning and language design. Part B.1 was compulsory and presented CSI agreement statements in two pages (with six items per page) followed by 15 pages of paired-factor comparisons. All paired-factor comparisons had 15 pages, with one pair per page. The remainder of the survey presented an optional section with open-ended qualitative questions.

3.3 Sample

The sample of participants consisted mostly of novices to Sema, but with extensive coding experience, particularly in live coding and creative coding tools, and a varying degree of machine learning experience. Twenty-one workshop participants engaged with the Qualtrics survey (17 males and 3 females, 1 non-binary, with an average age of 36 years old). The average reported coding experience was 9.14 years ($SD = 8.71$). One participant had participated on an early Sema

workshop and had advanced skills in live coding, machine learning and language design. One participant reported having no live coding experience. The remainder of the respondents reported having experience with at least one live coding system or a combination of several (e.g. ‘Chuck’, ‘Csound’, ‘Gibber’, ‘Supercollider’, ‘TidalCycles’, etc.). Seven respondents reported never having used machine learning tools, while 14 respondents reported having experienced different creative ML tools, illustrating with examples (e.g. ‘Wekinator’, ‘ml5js’, ‘Magenta’, ‘RunwayML’, ‘ML4A’, etc). Two respondents reported having had previous experience designing a programming language. Respondents reported having spent 19.62 hours working with Sema on average (SD = 12.79).

3.4 Results

All the participants completed Part B.1 of the survey, which referred to the CSI of Sema’s subsystem for live coding with the default language. Part B.1 generated an average score of 56.86 (SD = 20.84), which is a moderate score with a high standard deviation. This score indicates that the prototype assessed at the workshop provided support for creativity to novices to the system, in the task of live coding with the default language. The high standard deviation value indicates high variability across participants. On further inspection, Table 1 shows the average factor counts, average factor score, and average weighted factor score for each of the six factors of the CSI for Part B.1.

Table 1. CSI Results from the Live Coding with the Default Language (N=21)

Scale	Avg. Factor Counts (SD)	Avg. Factor Score (SD)	Avg. Weighted Factor Score (SD)
Exploration	3.71 (1.10)	12.34 (5.00)	47.72 (26.42)
Expressiveness	3.55 (1.23)	11.72 (5.18)	39.83 (25.29)
Results worth effort	3.17 (1.47)	12.33 (5.58)	32.07 (29.60)
Enjoyment	2.48 (1.12)	14.19 (5.57)	34.9 (22.38)
Immersion	2.11 (1.08)	6.63 (4.43)	10.50 (10.26)
Collaboration	2.11 (1.54)	6.04 (6.02)	5.56 (13.29)

The average factor counts for live coding with the default language in Sema (Table 1, first column) show the importance that participants have given to the different CSI factors (in a scale between 0 and 5) independently of how well Sema supported them. Exploration and Expressiveness we considered the most important factors, followed by Results Worth Effort and Enjoyment, with more moderate importance. Collaboration and Immersion were considered the less important factors for the task by novice users. The factor score metrics indicate how well Sema supported the six factors for the task of live coding with the default language, in a scale between 0 and 20. The weighted factor scores show sensitivity to the importance that participants assigned to specific factors

(factor counts) for the live coding task and how they affect factor scores, in a scale between 0 and 100.

The Enjoyment dimension received the highest score albeit a moderate good one. This indicates that, to some extent, participants became adept of live coding with the default language, despite having faced difficulties. These could relate to the design, level of polish of the system, or its complexity and steep learning curve. In a system with complex workflows and a steep learning curve, enjoyments ratings usually start low and increase over time, as the user learning experience progresses [4].

The Exploration, Expressiveness, and Results Worth Effort dimensions received moderate scores. The Exploration score (12.34) suggests that users found challenging to try out new ideas, different possibilities or versions, and that system support for versioning, branching, and duplication of user content should be developed. The Expressiveness score (11.72) suggests that users found limitations in the space of possibilities offered by the system, or in the ability to express their ideas when live coding with the system. It also suggests the need for developing system features that align with the goals and needs of novice users of a live coding system. The Results Worth Effort score (12.33) suggests that, for the the results obtained, the effort required to live code with the default language may be too high, or involve too many steps. This could point to the verbosity or low abstraction level of the language. Both Collaboration and Immersion paired-factor comparisons scored very low against the remaining factors. Collaboration received a low value with 8 respondents marking the collaboration statements as non-applicable (which coded Collaboration statements as 0). Given that the prototype was not designed with collaboration features (except for the ordinary Copy & Paste to share code) this score was expected. Nevertheless, we decided to keep this factor in the survey to apply a standard CSI version. The low Immersion score points out to potential disruptions to the workflow, which could be due to bugs, lack of feedback or visibility, or the necessity to look for help resources outside the system.

In general, we found the results obtained through the CSI were reasonable given the level of system complexity and skill levels of the sample of participants. However, we also recognise that these point out mostly to the need for deeper research around the causes of the moderate factor scores and to general improvements to the system.

4 Discussion

The results of the CSI score and factor breakdown analysis indicated that Sema required substantial improvements to achieve a good creativity support level. This was anticipated as, at the time of the workshop, despite successfully integrating rather complex subsystems and workflows (to a good extent), Sema was a functional high-fidelity prototype that presented ‘rough edges’ and bugs of varying severity (e.g. UI layout rendering, code editor interactions, debugger compilation error reporting, a malfunctioning delay line). More critically,

there were important functional and non-functional features lacking, such as a well-designed and encompassing documentations system, comprehensive error handling and feedback in all workflows, and data persistence mechanisms.

Despite these shortcomings, participants showed interesting accomplishments, including creating a new language (e.g. STX), porting an existing language to Sema (e.g. QuaverSeries), extending ML examples to new application (e.g. neural arpeggiator), integration of an euclidean rhythm generator, etc. Five participants used Sema in a live coding performance at the Networked Music Festival, one week after the workshop. All these examples of issues and of successful usage were captured through complementary methods (interviews, open-ended sections of the survey, Slack content) as part of the broader mixed-methods user study. The CSI was useful as a structured framework and complementary design assessment tool for uncovering how novices valued the different factors for creativity support, and to use the resulting insights to influence the directions of improvement and re-design.

Worthwhile of reflection and note are the challenges found with the application of the CSI. The CSI has been previously applied to CSTs with relatively simple tasks, where even the most complex examples are typically well-defined and self-contained tasks with high-level interfaces. In contrast, in CSTs with highly complex workflows and subsystems that interact, that are difficult to learn and master, and that aim to support end-user goals with non-linear, variable or unknown underlying tasks like Sema, the CSI provides an analysis that seems to park at a rather superficial level. CSI authors [4] point out that tool complexity and user skill level can impact CSI scores, and they acknowledge that new system complexity metrics are required to support deeper investigations into these relationships.

Furthermore, it was difficult to account for the effect that the communication used in the workshop advertisement might have had in building up overly inflated user expectations about the tool, and how this in turn may have affected the creativity scores. On experimental designs with more scoped tasks and constrained environments, this might be easier to account for. However when applying discount usability methods or testing CSTs in the wild with workshops [2] this seems to be an additional aspect to consider in the final evaluation outcome.

5 Conclusion

We have applied the CSI and computed scores for the task of live coding with the default language in Sema. We also carried out a breakdown analysis of the CSI factors. The scores were moderate which could be explained by the shortcomings of the high-fidelity prototype. They also suggested room for substantial improvements to the creativity support of Sema. On one hand, the CSI is relatively straightforward to apply, to analyse, and provides a limited but useful set of insights. On the other hand, we question the efficacy and applicability of CSI in assessing highly complex, hard to learn workflows and systems, which aim

to support unknown, non-linear or variable tasks, such as live coding, language design or machine learning.

For future work, we will release the more extensive results of the mixed methods user study, further identify and improve on the issues that undermine creativity support in Sema, and apply the CSI as a tool to complement assessment of the different workflows in future design iterations.

Acknowledgements

We would like to thank the workshop participants and our MIMIC project colleagues. This work was supported by two UKRI/AHRC grants: MIMIC—Musically Intelligent Machines Interacting Creatively (ref: AH/R002657/1)—and Innovating Sema—Community-building of Live Coding Language Design and Performance with Machine Learning (ref: AH/V005154/1).

References

1. Bernardo, F., Kiefer, C., Magnusson, T.: A Signal Engine for a Live Coding Language Ecosystem (2020) In: *Journal of the Audio Engineering Society* 68 (10), 756-766.
2. Bernardo, F., Kiefer, C., Magnusson, T.: Designing for a Pluralist and User-friendly Live Code Language Ecosystem with Sema (2020) In: *Proceedings of International Conference on Live Coding*, 41-57.
3. Bergmann, J., Sams, A.: Flip your classroom: Reach every student in every class every day. (2012) (pp. 120-190). Washington DC: International Society for Technology in Education
4. Carroll, E., Latulipe, C.: Quantifying the Creativity Support of Digital Tools through the Creativity Support Index. In: *ACM Transactions on Computer-Human Interaction*. 21, 4, Article 21 (May 2014), 25 pages. DOI: <http://dx.doi.org/10.1145/2617588>
5. McCartney, J.: Rethinking the computer music language: SuperCollider (2002) *Comput. Music J.* 26, 61–68. <https://doi.org/10.1162/014892602320991383>
6. McLean, A.: Making Programming Languages to Dance to: Live Coding with Tidal (2014) in: *Proceedings of the First ACM SIGPLAN Workshop on Functional Art, Music, Modeling – FARM’14*. pp. 63–70. <https://doi.org/10.1145/2633638.2633647>
7. Magnusson, T.: Herding Cats: Observing Live Coding in the Wild (2014). In: *Comput. Music J.* 38, 91–101. DOI: <https://doi.org/10.1162/COMJ>
8. Roberts, C., Kuchera-Morin, J.A: Gibber: Live coding audio in the browser. (2012) *Proc. Int. Comput. Music Conf.* 2012 64–69.
9. Shneiderman, B.: Creativity Support Tools: Accelerating Discovery and Innovation. (2007) *Commun. ACM* 50, 12, 20–32.
10. Roberts, A., Hawthorne, C., Simon, I. Magenta.js: A JavaScript API for Augmenting Creativity with Deep Learning. (2018) *Proceedings of the 35th Joint Workshop on Machine Learning for Music (ICML)*
11. Smilkov et al.: TensorFlow.js: Machine Learning for the Web and Beyond (2019) *Proceedings of the 2nd SysML Conference*