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

Author 1, Author 2, and Author 3

Lab X, School of Y, University of Z, Somewhere
{a1, a2, a3}@some.whe.re

Abstract. We present ongoing research around the design of Sema, a web-based coding environment aimed at enabling users to compose and perform music in real time using simple live coding languages and machine learning. Sema integrates custom dashboards with code editors, debugging and visualisation tools, reference documentation and interactive tutorials. Additionally, Sema provides advanced features such as domain-specific language creation, customisation and just-in-time compilation, and integration of external libraries for interactive machine learning workflows with bespoke neural networks and small training data sets. We analyse survey findings applying the Creativity Support Index, which aimed at understanding how well Sema supports creativity across its subsystems, and we 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 [1]. Such tools can be considered as Creativity Support Tools (CST), a class of tools designed to assist users in creative work that includes environments for software development, mathematics, music production, animation and visualisation [2]. Designing complex applications to support creativity with highly-technical domain-specific workflows nevertheless presents difficult challenges, particularly when assessing prototypes that inform future design iterations. We present an assessment of the creativity support of Sema, a web-based live coding playground for music and machine learning (ML), using the Creative Support Index [3] with a sample of workshop participants that comprised users ranging from novices to expert live coders.

* Supported by two UKRI/AHRC grants
System Overview

Sema’s user interface, version 0.5.0 used in the workshop (Figure 1), allowed users to navigate between two main areas: Playground and Tutorial. In the Playground area, users could access a dashboard of customisable layouts and a library of dynamic components—code editors, visualisers, and debugging tools. Users could engage in a free, open-ended exploration of live coding, language design and ML workflows. In contrast, the Tutorials panel presented a constrained environment with a strong proposition. Users could engage in a guided tour with tutorials structured with different levels of granularity and gradually increased challenge. The live coding editor (Fig. 1–a) provided users with the main tool to interact with the live coding languages (the default language and two more), key bindings for manual evaluation, suspension and resume of the audio engine playback, code commenting, and text search functions. Live code was dynamically evaluated to render real-time audio. The generic JavaScript editor (Fig. 1–b) enabled users to live code extended functionality to the live coding language, such as algorithms, ML model architectures, data sets, model training and evaluation routines, and bind data streams and buffers to and from the live code editor. The grammar editor (Fig. 1–c) enabled users to create or customise the live coding language by specifying, inspecting or editing the different components of the user-definable grammar specifications (i.e. lexer, grammar rules and semantic actions). A more detailed technical overview of the system is available in [5].

![Sema’s playground GUI](image)

**Fig. 1.** Sema’s playground GUI
3 User Study

The overall goal of the mixed-methods user study was to evaluate the creativity support, usability and utility of Sema. 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 post\(^1\) 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 \cite{4} 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 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 Creative 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.

\(^1\) Sema: Live Coding With Machine Learning Workshop
http://www.emutelab.org/blog/semaworkshop
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 [3]. 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)

<table>
<thead>
<tr>
<th>Scale</th>
<th>Avg. Factor Counts (SD)</th>
<th>Avg. Factor Score (SD)</th>
<th>Avg. Weighted Factor Score (SD)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploration</td>
<td>3.71 (1.10)</td>
<td>12.34 (5.00)</td>
<td>47.72 (26.42)</td>
</tr>
<tr>
<td>Expressiveness</td>
<td>3.55 (1.23)</td>
<td>11.72 (5.18)</td>
<td>39.83 (25.29)</td>
</tr>
<tr>
<td>Results worth effort</td>
<td>3.17 (1.47)</td>
<td>12.33 (5.58)</td>
<td>32.07 (29.60)</td>
</tr>
<tr>
<td>Enjoyment</td>
<td>2.48 (1.12)</td>
<td>14.19 (5.57)</td>
<td>34.9 (22.38)</td>
</tr>
<tr>
<td>Immersion</td>
<td>2.11 (1.08)</td>
<td>6.63 (4.43)</td>
<td>10.50 (10.26)</td>
</tr>
<tr>
<td>Collaboration</td>
<td>2.11 (1.54)</td>
<td>6.04 (6.02)</td>
<td>5.96 (13.29)</td>
</tr>
</tbody>
</table>

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. The Results Worth Effort and Enjoyment factors followed with 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.

Enjoyment 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 [3]. The moderate score of Exploration (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. Expressiveness obtained a moderate score (11.72), which suggests that users found the space of possibilities offered by the system, or the ability to express their ideas in live coding in the system, limited. This suggests the need for developing system features that align with the goals and needs of novice users of a live coding system. Results Worth Effort also obtained a moderate score (12.33), which 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.

Collaboration received a low value with 8 respondents marking the collaboration statements as non-applicable (which coded Collaboration statements as 0). Both Collaboration and Immersion paired-factor comparisons scored very low against the remaining factors. 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’. These included 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 complete and well-designed documentation, comprehensive error handling and feedback in all workflows, and data persistence mechanisms.

Despite these shortcomings, participants showed very interesting accomplishments, which included creating a new language (STX), porting an existing language to Sema (QuaverSeries), extending ML examples to new application (neural arpeggiator), integrating euclidean rhythm generators, etc. Five participants used Sema in a live coding performance at the Networked Music Festival, one week after the workshop. All these issues and examples 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. However, we found challenges with the application of the CSI that are worthwhile of reflection and note. The CSI has been applied to CSTs with relatively simple tasks, where even the complex examples are mostly well-defined, 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 support end goals with unknown, non-linear or variable underlying tasks, like Sema, the CSI provides an analysis that seems to park at a rather superficial level. CSI authors [3] 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, this might be easier to account for, however when applying discount usability methods or testing CSTs in the wild with workshops [6] this seems to be an aspect to consider.

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.

References

5. Prior work — anonymised for peer-review