The aesthetic values of source code
Doctoral Symposium

Depaz Pierre
Universite Paris-3 Sorbonne-Nouvelle, Paris, France
pierre.depaz@sorbonne-nouvelle.fr

Abstract. This paper presents the state of my doctoral research on the aesthetics of source code. The approach taken here is that of literary and linguistic analysis**:** by treating source code as text, it becomes possible to identify discourses about what makes source code beautiful, values such as clarity, elegance, simplicity, efficiency. Due to the ambivalent position of source code as language that is understandable both by humans and machines, I aim at highlighting the multiple relationships between those aforementioned values. Through this process of identifying the aesthetic standards of source code, the role and influence of programming languages will also be touched upon, insofar as they provide the linguistic structure within which those aesthetic manifestations take place.

Keywords: Source code, aesthetics, programming languages, elegance, clarity.

1 Purpose of the research, its importance to the field and expected contributions

The purpose of this research is to identify what makes source code beautiful. At first somewhat counter-intuitive - why should something ultimately understood by a machine as binary voltage current be deemed “beautiful” or “ugly”?- the fact that source code can be aesthetically pleasing is acknowledged by programmers, digital artists and hobbyists alike. However, whether there are specific, identifiable formal mechanisms which can elicit beauty in code has yet to be clearly defined. These mechanisms can be engaged with by treating code as literary text, asking questions about form, authorship, and readership in the context of written software, and how they relate to concepts of clarity, elegance,..., which are typically more closely related to code. The purpose of identifying the aesthetic properties of source code is to see to what extent these relate to, or differ from, the aesthetic properties of written language, and what is their relationship with the necessities of running code. By looking at the tension in what is at the same time a machine-readable text and a human-readable text, I aim at showing how different understandings of beauty inform each other.

This research intends to open up new readings of source code, using the lens of aesthetics in order to better understand code as a specific semantic apparatus, and not exclusively as a purely theoretical notion, or as an effective procedure. Particularly, this research contributes to the broader endeavor of demystifying code and bringing forward the shifting, diverse human aspects which come into play during the practice of writing and reading code, while at the same aiming to identify some of the forms that could inherently bring out beauty in what is both a tool for expression and for action.

2 Brief survey of background and related work

Overall, I’ve identified a dynamic which starts from a broad body of theoretical work establishing a definition of software as a socio-technical and cultural object (Mackenzie, 2006), moving towards closer, deliberate readings of source code in more recent work (Montfort et. al., 2012; Paloque-Berges), complemented by an approach from the field of literary studies, particularly around the works of Mark Marino and N. Katherine Hayles. Source code in itself is more and more the object of focus of researchers, particularly in terms of *what written code means*, how it represents, and how it is represented (Cramer, 2005). However, little attention has been given to what formal properties it should exhibit to be qualified as “beautiful”, and what definitions of “beautiful” can source code fulfill.

Among those, the body of works of self-defined *source code poetry* has been given close attention, but yet again mostly in terms of socio-cultural practices; these focus particularly on staples such as Perl Poetry and well-known esoteric languages, and often stop short of linking back their findings to the broader corpus of "everyday source code". Within the realm of "artistic code", Paloque-Berges's work is closest when it comes to analyzing the specific forms that source code poetry takes, and what relationships those forms have with natural languages. She does so by touching upon syntactical tokens that make Perl uniquely suited to this kind of endeavor, but also by providing larger theoretical frameworks, based on the idea of fiction (Goodman, 1978) and of practice (De Certeau, 1990), but not specifically on the *diction* of code, on its formal syntactic and semantic properties.

Among these different approaches, it is the *syntax/semantics* axis which my research aligns itself with. Indeed, I hypothesize that it is possible to touch upon these other thematic axes through the lens of form. According to Cramer, "*formalisms […] have a cultural semantics of their own, even on the most primitive and basic level. With a cultural semantics, there inevitably is an aesthetics, subjectivity and politics in computing"* (Cramer, 2005). A formal analysis of what makes source code beautiful will complement the social, fictional, and practice-based approaches that existing studies have already addressed.

3 Description of the proposed approach and current progress

The proposed approach is threefold: empirical, theoretical and artistic. The basis of this research is constituted of primary sources: source code available online, published in hobbyist magazines, commented in textbooks or published as artworks. The analysis of these texts, along with the discourses around those texts (comments, reviews, discussions) will help develop a heuristic to understand when and how the criteria of the beautiful is summoned among communities of programmers. This corpus, which has already been assembled, encompasses all ranges of project sizes, languages and periods.

The second component of this process will be to examine the findings from the analysis of this corpus in light of the somewhat more traditional concepts of literary theory, such as authorship, reception, rhetorical figures, style, voice and layout, among others. This will establish a framework which takes into account both the practices and appreciations which are intrinsic to the communities of practice of source code, as well as further highlighting some unique aesthetic properties of code via literary methods of analysis. For example, there could be an exploration of the concept of *simplicity* by comparing the programming paradigm *DRY* (Don't Repeat Yourself) and Barthes' *writing degree zero* (Barthes, 1972).

Throughout this process, I intend to complement this approach by writing source code myself, putting into practice the concepts unearthed as the research unfolds, with the triple aim to (1) illustrate concepts developed in the second, theoretical, component of my approach, (2) possibly invalidate such concepts by showing their limitations in a concrete instance and (3) open up new pathways for thinking through what can make code beautiful.

This research project is currently in the phase of analyzing the corpus in order to extract repeating patterns which seem to define beautiful code; the immediate following phase will then be to relate those patterns to existing approaches in literary theory and aesthetic philosophy, in order to investigate if the same word (.e.g "clarity") means the same thing in two different fields, and how possibly divergent meanings could enrich each other.

References

Barthes, Roland, 1972, *Le Degré zéro de l'écriture suivi de Nouveaux essais critiques,* Paris: Editions du Seuil.

Chandra, Vikram, 2014, *Geek Sublime: The Beauty Of Code*, Port Townsend, WA: Graywolf Press.

Cramer, Florian, 2005, *Words Made Flesh: Code, Culture, Imagination*, Rotterdam: Piet Zwart Institute.

De Certeau, Michel, 1990, *L'Invention du Quotidien,* Paris: Editions Gallimard.

Goodman, Nelson, 1978, *Ways of World-making*, Indianapolis, IN: Hackett Publishing Company.

Mackenzie, Adrian. 2006. *Cutting Code: Software and Sociality*. New York, NY: Peter Lang.

Montfort, Nick et. al., 2012, 10 PRINT CHR$(205.5+RND(1)); : GOTO 10, Boston, MA: The MIT Press.

Paloque-Berges, Camille, 2009. *Poétique des codes sur le réseau informatique,* Paris, Editions des Archives Contemporaines.