

Literate programming and documentation of artistic processes

Henrik Frisk, henrik.frisk@kmh.se

Abstract

The general question of how to document artistic processes is largely unsolved, mainly because each process is relatively unique (See Frisk, 2018, for a related discussion on documentation). The particular case of documenting compositional processes in electronic music, let alone attempting to document the actual sonic objects that are the result of such processes, has its own set of challenges. Not only due to the fact that it is difficult, the benefit of documenting is sometimes less than the time consumed by doing it. There are at least two aspects to this: (i) In a world where almost everything is constantly recorded or potentially so, meticulous documentation of ones process may seem ubiquitous. Almost everything leaves a digital trace and that this may be difficult to trace later is easily forgotten. (ii) Even if the compositional process *may* be of interest to the composer or others, it is still not necessarily worth the while. Recreating from memory may be just as valid, and perhaps artistically more apt. After all, the documenting force is to some degree rooted in a modernistic view where the key to the work lies in the material with which it was created.

Though the elements of the process as material for an interpretation is questioned, an assumption in this paper is that the artistic work process constitutes knowledge and that this knowledge may be useful for the development of the process. This kind of knowledge is difficult to grasp and assess, due to its elusiveness and refusal to adopt to predefined categories. Part of the ambition here, however, is to show that the method presented may not only exemplify a method for documentation, but that it may also create an understanding of artistic knowledge.

The method is built on using a technique referred to as literate programming. Literate programming was proposed by Donald Knuth in 1984 (Knuth, 1984) to allow for a different structure for programming, one that is more closely related to natural language than machine code, and that is

structured the way that humans understand logic rather than how machines understand it. Needless to say, what this meant in 1984 is quite different from what it may mean today and the method here may appear arcane to users of GUI based systems for composition such as some of the modern DAWs or software like Max or PureData that has improved the situation for human friendly computing. Knuth's ideas, however are interesting in the context of music composition precisely because it allows the user to structure the music programming process in a way that is less governed by the logic of the language and more structured in a way that improves structure and understanding.

Practical examples from an ongoing compositional process is presented and discussed and the particular feature of literate programming that allows documentation be embedded with the code is demonstrating the self documenting aspect of the method.

References

- Frisk, H. (2018). "ArtDoc - An Experimental Archive and a Tool for Artistic Research". In: *Music Technology with Swing*. Ed. by Aramaki M. et al. Vol. CMMR 2017. Lecture Notes in Computer Science 11265. Springer, Cham.
- Knuth, D. E. (1984). "Literate Programming". In: *The Computer Journal* 27.2, pp. 97–111. DOI: 10.1093/comjnl/27.2.97. URL: <https://doi.org/10.1093/comjnl/27.2.97>.