

## Self-awareness in software

The realms of software, cyberspace and beyond, is often considered to be 'unreal' or at the very least, some other kind of real. Yet at the same time no small number of computer related terms and concepts are borrowed directly from the 'analogue' world. A whole slew of metaphors serves to bridge the conceptual gap between the world(s) inside and out of the screen; desktop, window, folders, and so on. This is a curious paradox of sorts. It is as if the traditions and practices of software developers are considered opaque or even occult. A black art, of sorts. And by extension, the highest virtue for developers (of milquetoast consumer software, anyway), it seems, is to create something so intuitive and elegant that it is prematurely branded as "simply natural."

Of course, the question of what is or is not natural is deeply complex and it is not within the scope nor ambition of this paper to tackle that particular challenge. However, some software does not strive for 'naturalness', but on the contrary insists on highlighting or bringing to attention its own artificiality. This is seen in many different types of software, ranging from virtual assistants (ask Siri certain questions and it will deliberately shatter the illusion of its "realness") to video games and, uh, also other examples. Within the world of programming and software development itself, self-awareness is arguably a key component in the development of software. Concepts like metadata and recursion are examples of how some degree of referring-to-onself is necessary for certain functions.

To be clear, the concept of self-awareness is different from that of consciousness. I do not wish to discuss questions regarding the possibility of a truly sentient or conscious artificial intelligence. Rather, the aim is to examine and analyze the ways in which developers of software can or try to express self-awareness in or through software itself. Also, how this self-awareness can be seen or used as an opportunity (or an express order, even) to reflect on the artificiality and materiality of software. To examine what constitutes self-awareness is also, perhaps inadvertently, to define what is not self-awareness. That is, what is 'ordinary' or 'unawareness'. This is arguably a 'fringe' topic and perhaps not at the heart of the paper. Still, it could be of some relevance to make such considerations.

In terms of placing this topic within the confines (though not exclusively) of the course syllabus, I think the texts regarding fun in software are applicable. Self-awareness can be used as an expression of humor, in and outside of software. The particular materiality (-ies) of software engender certain ways of expressing the self-awareness. As such, I might also make use of the Dourish text on shifting materialities. I might also draw upon texts relating to generativity and the role or importance of self-

awareness in generative art. Again, I am not interested in questions of authorship or artificial intelligence, but to what degree an overt expression of its own artificial nature (eg, “this generative art was created by a computer by following so-and-so procedures, whoa!”) is important to the aesthetic qualities of the software in question.

It seems self-awareness can be used for many different purposes. To ridicule, to lampshade, to make fun of “itself”. Or perhaps to make some deeper, more profound comment on the shifting nature of reality and the increasingly blurred lines between the “natural” and the “artificial”. In any case, I want to include certain case examples to be subjected to my own analysis – itself of course informed by the writings of others. In terms of non-syllabus literature, I might draw upon the writings of Dunne & Raby, who introduce the notion of ‘parafunctionality’, a sort of design philosophy in which the artifact (software or otherwise) serves a trojan horse for something else and upon the discovery of which, the user is forced to reexamine their own concept of what is or is not supposed to be the “real” product. Pardon the overuse of scare quotes.

The notion of self-awareness is also somewhat closely associated with the concept of postmodernism. As such, I might track down various texts and sources discussing postmodern software or software’s role in postmodernism. A proposed structure for the argument of the text might be this: here’s an account of what is meant by self-awareness; here’s a bunch of specific examples of software exhibiting self-awareness; here’s a consideration of how and why this self-awareness is expressed; here’s a conclusion following the discoveries made during the paper. Something like that.

In short, what I want to consider is this: how can software express self-awareness and how can this self-awareness serve as a jumping off point to consider the material and aesthetic qualities of software (in particular or in general)?