ScreenPerfect: Accessible Tools for Web Art Installation .  
A project talk delivered August 2013 to Dames Making Games Toronto (dmg.TO).  
Alex Leitch  
@aeleitch  
alex.leitch@gmail.com

This is a talk that is not about making things, so much as fixing that which is visible until it vanishes. The joy of repair is that, done right, the repair disappears. This is true in plumbing, and it is true in code, which is plumbing for ideas.

There are lots of interesting threads in the disappearance of a language that drives the mechanics of a world. These are languages that build medical devices, cars, and phones. The invisibility of these languages is an interesting problem in the same respect that gendered linguistic assumptions are an interesting problem. Because they are invisible, they are a challenge to overcome. This is where it becomes useful to look outside of computer science to begin to have an idea of how encoded ideals within language can influence the effect one can have one one's world. The best-known and most straightforward theorist on this topic is Hélène Cixous, who wrote in 1964 an essay called the Laugh of the Medusa.

Cixous spent a lot of the Laugh of the Medusa on the idea of direct, physical desire. This has some interesting resonance with the popularization of code-communication, because pornography sprouted in the comfortable anonymity of the internet like mushrooms sprout in forests after the rain, so clearly someone's desires were not being met by the extant system of expression. This resonance is outside the scope of my thesis project, although I believe it is interesting to note that the majority of young women, given a dual-screen game, made games about dating, directly about gender, or in the case of the finest producer - a transhuman - pornography itself. Embodiment is clearly very important to these people. The state of the personal narrative in a new medium is entirely too broad for me to address here, however.

Therefore, the part of Cixous that is interesting for understanding the idea of code as a creative practice of resistance is her insistence that women write, and that women write to resist the language in which they are cast. French is a strongly gendered language. It places a gender on each noun, and a degree of familiarity on the use of the term "you" that we cannot yet articulate within English: the best we can do is "they," which is pluralized in a fashion that "vous" is not. This is particularly interesting in Canada, a country of dual languages, where schoolchildren are trained from a young age to have at least a passing fluency with both worlds. We are expected to learn between, it is illegal to exclude the other from our signage. This is important to my understanding of code as a practice of linguistic structures. Language is a politicized site of visibility within Canada. Cixous' particular articulation of the value of the minority writing their own world, even when forced to use a dominant language that itself is constructed to preclude their existence, is a valuable view of how resistance might be realized.

In code, this comes because code languages, and the libraries that make them work, are released by major corporations on a regular basis. Innovations become dependent on an ability to fall in line with the way of thinking that hundreds or thousands of previous workers have made happen, while still preserving a sense of creativity that permits one to figure out basic problems. These problems are largely technical: how to make best use of a second screen, how to force a specific video to play back in that framework. Everything inside a computer is numbers, however, and the way those numbers line up has been dictated by politics. ScreenPerfect, as psXXYborg, was written to take advantage of the Safari browser - backed by Apple - but webM video files are backed by Google, and are more compact. Due to complex negotiations around copyright, the Apple's webkit solution will not use webM, and Chrome would not play back H.264.

These differences are often put down to simple "technical problems," but this idea is inaccurate, and elides the code that underlies the systems built by large companies to transmit information. Video files are a contested ground at the moment, because in addition to being a convenient communication medium, some video files are incredibly protected forms of information. The litigation surrounding piracy of television and conventional first-run media has been publicly associated with theft: a type of theft where an object's value is removed even while the object itself remains present and resellable. Patent fights about video format, about technical standards, and about open or closed systems, dictate the terms of how media may be distributed and displayed.

screenPerfect gets around many of the restrictions on video by coding a new system for display and interaction, which values a short, engaged experience over the longer-form systems that are already in place. Rather than pursuing the television experience, SP turns video to a system more in line with video games, specifically riffing on an engine popular with independent game-makers. Independent game-makers are concerned with a new medium, the video game, which is made up of the interaction of people with the game system. As Galloway argues in his essay on gaming, the software alone does not count.

Video games are new, and therefore they are as-yet unformalized: the best we get is a system of triple-A games, which are largely concerned with imposing men shooting things with imposing guns. The triple-A framework is not typically particularly feminist, with one or two startling exceptions - the end of Saint's Row 4, which is a feminist game through and through, features the rescue of Jane Austen by space aliens, for example. By and large, the popular and commercial nature of large games precludes the sort of deeply personal representation that Cixious refers to in her own work.

Games - code activated by interaction - are not restricted to only the blockbuster, however, any more than film is. Games are sometimes accused of Cinema Envy, and I believe this is a real problem: because they are not yet taken seriously, games are still free to explore new ideas, to be obscene or funny or reflective of their own culture. Jane Austen is peculiarly popularly persistent, after all: even post-Bridget Jones, the Austen canon continues to inspire new works of popular culture. The important part, however, is that game-makers are still relatively free artists. The code they use, based on massive information systems, still has the potential to make money and drive creativity in the use of and development of technology.

Innovation is not a worthwhile term here, because it has recently been devalued to refer primarily to systems that can be developed quickly and deployed to a broad audience. My interest is not in the broad audience, although it is simple good practice to write code that can be widely distributed. My interest is in the support and production of technology that permits new forms of private expression. Here, too, I do not mean private in the sense of overly limited. I mean private in the sense that something may be presented to a limited audience with the understanding that the complete experience is meant to be retained by that audience. At the same time, the technology is intended to take advantage of systems already in broad use, because these are systems accessible to artists, who must work with what is presently real to define what might become real.

In this, it is important that the privacy of a system not be restricted to exclusively those with the major capital to install and control a given band of technology. While developing the dual-screen technology to run psXXYborg, I was considering a scene from Cory Doctorow's Pirate Cinema (2012), in which a crowd of young film-makers, who make their work entirely from pirated media that has been remixed and repurposed, put up a movie theatre in a forested park by synchronizing the pico projectors on their phones. This was the chief inspiration for screenPerfect: a technology so lightweight that it would require no setup and no particular technical skill to use, which could then permit artists who already had clear vision the advantage of being able to screen their works anywhere, quickly, with nothing to lose.

Pico projectors have not yet taken off in popularity, because they have not yet begun to be built into smartphones and cell phones. That does not matter. We have a laptop on every table, and on those laptops are browsers. To write software that can be understood by the browser, one must interact with a system of capital that is dedicated to the co-option and devaluation of the author at the privilege of the corporation. The point, then, as a practitioner of a form of creativity that has not yet been completely co-opted - for the creativity of someone solving a problem within code, a specific problem especially, is still a creative practice - is that we may resist and open a door to further resistance.

These negotiations around copyright, around patent law and inclusion, pull back a layer on the systems which build the internet. The systems are built by large companies which buy smaller ones and make business decisions about how to include which innovations based entirely on their expense, and what business risk is revealed by including a given technology. Making use of code articulates a subordinate relationship to this machine, with a key difference.

At the moment, internet technologies are functional because they are dependent on massive code-base issues, which are negotiated by corporations in much the same way as the official French language is negotiated by the Academy Française.

That these technologies are invisible is an interesting problem the same way the underlying gendered assumptions in language is an interesting problem to overcome from the point of view of the legal system. It's an interesting problem. What I mean by an interesting problem is a problem that's worth spending a long time on, teasing out all the details. It's like a math problem that's made up of words. Not a word problem, a problem with the words. So, language assumes culture, Javascript assumes callbacks. That means you write it in reverse. It's sort of like being Ginger, you dance backwards in heels the whole time.

As a coder, what you write becomes real in a tangible sense, making interesting new things is difficult, thinking about what they might be is hardest, particularly alone - so, how do you fix this? You find a gang. A girl gang! The arts have an audience, but their audience is limited by technology [\cite{LISA notes}. If you have a particularly brilliant painter or a particularly brilliant graphic designer or an awesomely amazing musician, that is very fine, however, if they cannot record their information, or transmit it to a broader audience, then they will be broke. We know this is a problem. The next problem is how to line up technology so that they don't go broke after they do have the audience. We're still working on that one. It's a scary one.

Technology has a profit margin, but technology, because it builds on itself in a recursive manner, after education and privilege has given people the ability to write it, often has a very limited cultural outlook. People burn out in the tech field all the time. They burn out constantly. Google hires people for the express purpose of burning them out building machines. Because that's what we do. We write documents that are machines. They're laws. They're just laws made of a different kind of language. And they are breakable. There's an entire body of people who work breaking the laws of the language that we write into our documents. We call them hackers. They're very valuable. We need people who can break the laws to show us where they go wrong.

Together, good art and good technology can make good experiences. What I mean by good experiences is experiences that ask questions or display well-rounded characters or use any of the many guidelines and theory that we have that drive the Humanities and the arts more broadly. If we pair people off, then we lose the part where the technologists are scared of the artists because the artists speak a mysterious language that the technologists do not understand, and the artists can learn that what looks like magic - a beautiful glass panel that just does what they say - actually takes a lot of work, it's not work that is unquantifiable either. It's built in code commits and checkins, you can see where the period goes, it isn't being a wizard. On either side. It's being a mechanic. It's assembling things well and putting them together and bolting them down and recording that so that it can be done again by another person. This is true on both sides of the equation. Feminism is equality. Art is technology.

By driving with art, rather than technology, the experience can be newer, and less expected. This is because of the aforementioned recursiveness of technology. Tech builds on tech, almost unquestioned, but art builds on culture and culture has a way of slipping around. When you are not looking at it, it moves in the dark. You think that teddy bear is sitting on your shelf but you wake up and it's at the end of your bed and let me tell you, you will scream. Culture can be seen as nothing but questions, questions like "How might a person be?" I think the answer is better.