“Close to the Code, Down by the River”, Minecraft Symposium, Concordia University (December 21, 2015)

9 February 2015

Close to the Code, Down by the River

Bart Simon, TAG

So aside from my amazingly clever nod to the 1970’s band YES here is my hypothesis in a nutshell. Minecraft is to the IBM PC as Dragon Age is to the Mac. Seems pretty obvious right? At least there are still a couple of folks in the room who remember Umberto Eco’s famous newspaper column from 1994.

> The fact is that the world is divided between users of the Macintosh computer and users of MS-DOS compatible computers. I am firmly of the opinion that the Macintosh is Catholic and that DOS is Protestant. Indeed, the Macintosh is counterreformist and has been influenced by the "ratio studiorum" of the Jesuits. It is cheerful, friendly, conciliatory, it tells the faithful how they must proceed step by step to reach - if not the Kingdom of Heaven - the moment in which their document is printed…. DOS is Protestant, or even Calvinistic. It allows free interpretation of scripture, demands difficult personal decisions, imposes a subtle hermeneutics upon the user, and takes for granted the idea that not all can reach salvation. To make the system work you need to interpret the program yourself: a long way from the baroque community of revelers, the user is closed within the loneliness of his own inner torment.

Ignore for the moment, Eco’s conflation of the computer with its operating system. A similar articulation was made by Sherry Turkle in 1996 when she wrote about the social construction of the computer user in *Life on the Screen*. In her case the IBM PC was Modern; the Mac quintessentially postmodern and she writes, “The Mac encouraged users to stay at a surface level of visual reproduction and gave no hint of inner mechanisms… the user was presented with a scintillating surface on which to float, skim and play. There was nowhere visible to dive” (34). So the story goes, the modern hermeneutic aesthetic of DOS/PC takes on an increasingly embattled position with the rise of the Anglican and seemingly impenetrable Windows interface (which creates a surface layer over MS-DOS and therefore can only cause frustration). At the same time we can observe a concurrent lock-down on proprietary hardware as the IBM beige box and its contents flirt with plug and play. The whole thing comes to fruition in our contemporary moment with mobile hardware and operating systems with which we barely interact at all, touch screen interfaces that give us the sense that we are doing something when we are not (the obvious extension of Apple’s transparent ‘look but do not touch’ aesthetic is now superceded by ‘touch but do not feel aesthetic’), and ubiquitous, miniature and cloud computing in which the CPU is removed completely from our awareness altogether.

Design scholars like Donald Norman proclaim these shifts as advances in the usability and user friendliness of computer systems but in his essay,‘There is no software’ Friedrich Kittler has refers to this movement as a kind of friendly secrecy system, “Firstly, on an intentionally superficial level” he writes, “perfect graphic user interfaces… hide the whole machine from its users. Secondly, on the microscopic level of hardware itself, so-called protection software has been implemented in order to prevent ‘untrusted programs’ or ‘untrusted users’ from any access to the operating system’s kernal.” Anyone who knows Kittler’s work will know this is the beginning of a line of critique of the computer industry – users “cannot be trusted” and are “locked out” of the technology for their own good (but in fact it’s for the good of the existing military-industrial-entertainment power complex).

This discussion is kind of how I opened my old analysis of case modding in the context of LAN party culture in the mid-2000s. Back then, I was interested in how case modders unintentionally challenged and sometimes upended the cultural logics underwriting Kittler’s friendly secrecy system by dremelling their PC cases, overclocking their CPUs (using dip switches at that time), innovating cooling systems and installing cold cathode ray tubing. I was interested in the subcultural practices that give us access to what I then called the material guts of virtuality (something I have since gone on to argue is one of the basic premises of maker culture – the thingification of the digital). In one old paper, I focused my analysis on ‘lapping’ which is a mostly forgotten and warranty-voiding skill having to do with smoothing the surface of a CPU or GPU so as to eek out ever so slightly more thermal conductivity with a heat sink, thus lowering the temperature of the CPU, which allows for higher clock speeds, faster processing and therefore better LAN gaming performance (a subject of much “heated” debate).

In my analysis this has everything to do with game culture but never had anything to do with modders’ ability to make their machines more efficient for gaming (the physics actually barely supports them in this). Instead, lapping becomes a way for the modder to gain a measure of Protestant-like intimacy, if not a sense of control, over her machine. So much so, that in addition to the step by step ‘how-tos’ and sumptuous pictures of lapped cores (everything is about the mirror finish and its display) a very common activity at LAN parties would be CPU drag races – the ultimately display of macho bravado with the promise of the spectacle of the chemical smoke and sometimes flame of melting motherboards.

Whether or not it works, lapping renders the abstraction of computation and its effects as material, tangible, sensual and accessible. The tactile smoothness of the handmade mirror surfaces of the intel chip become a palpable contrast to the capacitive touchscreens of our ipads. This is what it meant to be “close to the machine” and playing with the “guts of virtuality” – to bypass the friendly secrecy system in order to mess with (and even undermine) the material conditions of the production of virtual experience/immersion as social control – all for the sake of playing a better game (rather than social justice as was the case with early hackers and hacktivists). This is my wacky version of ludology.

Fast forward now to Minecraft and what was old for me has become new again. This time I am less obsessed by material practices of computing and more obsessed by the displaced sensuality of those practices as a matter of game design and gameplay. Let us assume that we are already all Catholic and the guts of our machines are now just fashion if they are there at all. For Kittlerians the only route left would probably be something like RaspberryPI and Linux and there are indeed signs of a Protestant reformation on the horizon but for the rest of us our route must be via the software. The code.

From Lessig, through Alex Galloway, Adrian Mackenzie, Miguel Sicart and many others… Code is law, code is rule, code is control, code is… basically evil. Of course I exaggerate but it is consistent with the only Kittlerian line of defense – we must all learn to code. In machine language no less. There is absurdity in this but it is deadly serious – more so than ever, the material conditions of our existence depend on who is closest to the machine. Critical theory updated for today's operating systems have us, following Adorno or Marcuse, swimming in a sea of superficial menu choices and button presses in the name of liberation (liberation from the machine/from our bodies/from our selves) while the power complex of the State absentmindedly mines the data we can't be bothered to think about.

In such a world, video games can never be more than a distraction – an obfuscating layer more pernicious than Siri and nothing at the moment epitomizes Catholicism of our i-world better than Dragon Age – an immerse world of pure surface. I will not spend time on this because I can probably be easily dissuaded over beer but DA will do fine as today's foil for Minecraft. This should take me directly to a discussion of how Minecraft was originally designed and coded and then it should take me to a discussion of Minecraft modding but I am going to bracket that as the easy case to save for later.

Instead, I am interested in the subjectivities of Minecraft players. It would be pretentious and foolish to claim that Minecraft gives players access to the guts of their machines. It is an (extremely) object oriented operating system of course and the experience certainly fits Eco's model for MS-DOS "To make the system work you need to interpret the program yourself… the user is closed within the loneliness of his own inner torment." Some other game experiences come close to this but to see Minecraft as a materialization of Eco's intuitions about MS-DOS seems a perfectly reasonable place to start.

But Eco's parable about PCs and Macs is not about absolutes but about trajectories. Just as CPU lapping is not about efficiency but about desire. If we follow this pathway we need not worry too much that we will never make Kittler happy but the problem now is where to look – if Minecraft can be interpreted as a material(ized) practice in the way that case modding obviously is – where is its "lapping?" That's the sense that this paper is nothing but an invitation for further discussion but I have just enough time to end with some speculation.

I see in case modding a generalized desire to unpack the black box sensibility (or rather beige box) that defines our relation to computing and technology. To make apparent, sensible and tangible the chains of actants and mediations that compose our systems. Pure deconstruction is untenable because it renders those systems inert and results in inaction and the collapse of the very games that the case modders want to play. This is where Minecraft is suggestive for me. I now see the game as a kind of unholy simulation where the drive is not to reproduce or even oppose the social-technical systems that compose our world but to re-articulate material chains of actants as fictional systems – natural environments, houses, roads and railways, electric grids, factories, weapon systems and computers themselves. It is a machine within a machine that drives desire beyond the reach of friendly secrecy systems with unknown consequences. Eco would play Minecraft, and you know… probably so would Kittler.