

"Minecraft as Statecraft", Festschrift for Chandra Mukerji, UCSD, San Diego, CA (June 6, 2014).

## The Statecraft of Minecraft – or Something Like That

Bart Simon and the Mod Studies Group  
Technoculture, Art and Games, Concordia U., Montreal

---

So this [1] and this [2], [3]. The images are from several projects at various stages of completion aimed at creating the Chateau and Gardens at Versailles in Minecraft.

Minecraft, for those who do not know is a hugely popular multiplayer 3D isometric open world building game created by Swedish indie developer Markus “Notch” Peerson in 2009 and now owned by his game studio Mojang. The game boasts over 100 million registered users and it stands as one of the highest grossing and most widely played video games of all time, across all demographics. The game exemplifies the design principle of being simple to play but difficult to master. In either the survival or creative mode of the game players use a blocky avatar named Steve to interact with a lego-like game environment to create whatever they like, alone, or in groups on shared servers. The basic game is labour intensive and requires many hours spent in the routine extraction of resources (the mining part) for use in building projects which can again take many hundreds of hours (as with the Versailles project). Minecraft is a sandbox game about exploring, making things, and transforming the environment. It is also a social game played by vast communities of players who are doing this work together – instantiating

their collectivities through the virtual material practices that the game affords. That minecraft players are creating literal and figurative gardens in this sense could be our starting point.

Crucially, part of the story of this game is that the environment does not come as a ‘fait complete’. The game has been designed to allow for “mods” or modifications to the composition of the gameworld and Minecraft supports one of the largest modding communities in the world creating, changing and rearranging the conditions of production in the game. So it is a game about making (and therefore supposedly about creativity) but it also a game about making the conditions for making. It is a game about land, infrastructure, logistics and code. It is a game about heterogeneous engineering and I am pretty sure this is Chandra’s kind of video game.

On the surface, Minecraft Versailles should appear as a crude and rather facile reference to *Territorial Ambitions* or maybe just a parody, as pre-modern territorial statecraft gives way to the geeky aspirations and identity work of Minecraft recreations. How better to demonstrate the “engineering” virtuosity of a player (especially a French one) than to attempt a scale recreation of the Chateau and all its gardens? Minecraft is rife with this activity as players rush to reproduce every plausible shared real and fictional iconic piece of architecture and landscape from Versailles, to the Taj Mahal, to Eden, to the entire country of Denmark, to cityscapes of the TV Series Game of Thrones.

But as a digital culture and game studies scholar now I am grasping at something else for which Minecraft might serve as a starting point. Chandra's book presents us with an idea of "the land" and works on the land such as gardens (amongst other things) as the "stuff" of French early-modern territorial statecraft. What if today's land is code (where code is a shorthand for some software/hardware assemblage)? To my mind, Chandra's book, probably filtered through Patrick [Carroll], has always been about "the stuff" of the State but also the "stuff" of power. Power as logistics rather than strategics as Chandra has described it. Minecraft is not the "stuff" of the power perhaps but "the stuff" of Minecraft might be. In this sense, Minecraft's Versailles is an allegory for Chandra's argument in *Territorial Ambitions* or perhaps a runaway version of it complete with 'Notch' as the erstwhile Sun King in a definitively neoliberal rather than a late feudal fantasy that combines pleasure and play, distinctly modernist ideologies of management and control and a model of libertarian self-sufficiency and self-government.

Before I move to trouble this - because one thing we know from Chandra's work is that when dealing with materiality there is always trouble – let me at least try to drive the point home. The world as Minecraft is already consistent with the high modernist schemes James Scott laid out in *Seeing like a State*. We can dispense with the obvious articulation of the totally knowable world and the in-principle identification, classification and organization of every possible object and action. This was already a feature fantasy of virtual and digital worlds perhaps starting with Disney (a nod to Nic

Sammond) or maybe *The Well* (a nod to Fred Turner), wonderfully illustrated in *Second Life* and *World of Warcraft* and reimagined again as a neoconservative panoptic dream in the latest blockbuster game just released last week, *Watchdogs*.

Minecraft is actually more interesting than this because it is a genuine Heideggerian set piece (Vella 2012). All the world rendered (literally), and not just represented, as ready-to-hand. The Minecraft world is not made amenable to vision in the classical constructivist sense so familiar in science studies. The minecraft world is already ready to go. The “land” is composed of cubic blocks of dirt, stone, iron, diamonds, water and more – each understandable in terms of their utility for doing something else in crafting recipes for making more resources, tools, weapons, buildings, farms, gardens and landscapes. In part we are meant to experience this as a kind of fundamental human condition (and it’s interesting to consider Chandra’s *Theory and Society* paper on Christian Humanism in this regard). What choice do we have but to understand the world this way? When you start the game you are plopped in the world in a state of nature. Survival mode. Daytime is edenic and nighttime is hellish as the zombies, creepers and skeletons will make quick work of you if you don’t discover (because there are no instructions) that punching a tree makes wood to be used for building a protective house or crude tools and weapons.

It is not long before each player creates a homestead with the ubiquitous mine for digging ores, a farm growing crops (with seeds scavenged from the wild) a pen for

raising livestock and so on. This is all fine and good but Minecraft comes into its own especially on multiplayer servers where industrious players are able to use some of the (again hidden) cause and effect mechanics or just Colbert's principles of scientific management to facilitate semi-automation thus allowing the creation of railways, hydraulic systems, massive agri-farms, strip mines, managed forests and other high modernist projects (even breeding farms for the villager NPCs in the game). There are no instructions for the game that suggest that players should create these things and at this scale production far exceeds the basic functional need for shelter, food and tools (even as a Durkheimian model of the division of labour). Players do it because they can (or they figure out that they can – which is part of the fun) and each act of creation is a material demonstration of an effect in the world as it is rendered. No wonder then that the primary impulse of the modders continues in this vein – player-modders have taken it upon themselves to design resources and mechanics for sophisticated transport and energy systems, factories, slaughter houses, robot slaves, nuclear weapons, and even fast food restaurants.

What makes this experience as allegory especially effective is how this version of the world is also given to us as a fantasy of code itself. It is easy for players who are especially used to the graphic overproduction of contemporary video games to feel “closer to the code” in the low-res pixelated world of Minecraft. One feels as if one has more control over the conditions that produce the world as you experience it. And with

the rapid convergence of object-oriented programming, GUI heavy, menu-driven game making tools and player customization and modding, it's not difficult to feel that you are the programmer. At least more so than in a game like Watchdogs. [side by side images]

So what you get in the end is a palpable sense that the world you make is the world as you want it to be rather than the world you are pleased or frustrated to consume. I guess I have also not been clear enough that despite these images on the screen this is also not a world for watching – this is not a novel or a film. Players occupy real material space and affect each other and the world over long drawn out periods of time... this has been a consistent position of virtual worlds and games scholars since the beginning. We are not talking about representation in Minecraft, we are talking about material experience. This was already one of Chandra's first points about the material and not just symbolic effects of Versailles as statecraft. I am reminded of the puzzle of the King's orchestrated walks through the garden grounds. These could not be mere symbolic displays of opulence and power but an experience of the King's power as material effect (and thus independent of the King as such). Similarly the ubiquitous youtube videos of players' minecraft accomplishments tell only a partial story (and one too easily told by armchair digital scholars). More important are events like treasure hunts that are designed to move visitors through the player's world.

In Minecraft, each new modernist installation changes the conditions of experience and interaction in the world for one's self and others (with plenty of debate about who's

backyard will host the strip mine). But each installation is also a material demonstration of the player's capacity to code or at least to assume the control over the world that code allows (this can get nuanced in minecraft worlds as players leave but their installations, like so many lost relics and ruins, remain behind). The trouble comes of course with the bedrock upon which all this is built. The material platform for the minecraft world, the java codebase, is an unwieldy mash-up of many iterations of code that many programmers have already criticized for being extremely unwieldy, inelegant and inefficient. And it is a testament to Notch's billionaire eccentricity that he is nostalgic about this and refuses to rewrite the game. So while the pixelated aesthetics suggests a game that should run on 1980s hardware in fact, especially once modded, the game will cause even the very latest machines and networks to slow down.

Modded Minecraft servers are incredible patchworks of the core game plus mod codes written by different people that have to be calibrated. The desire to build the ultimate automatic strip mining machine runs up against the hassle of calibrating various mods written by a host of different coders working under different assumptions [energy systems example]. These worlds are slow to move in, they are lossy, glitchy, and sometimes very unstable (resulting in the potential loss of hundreds of hours of in game work). Crucially these modders are not the developers (Mojang) and indeed they might be exploited labour producing content for the game for free. The modders are "locked" out of the codebase (Minecraft is not open source) and surely if they did have access

things would run more smoothly. Similarly this would be the case if they were vanilla players – [not sure if modders have a corollary in TA but they certainly do in Impossible Engineers].

Minecraft Versailles is about more than the desire to arrange blocks in a likeness of the famous chateau. It is a flexing or testing of a collective engagement with the material conditions of production of the day. Chandra uses Ilyenkov to articulate this idea that “the crafting of the built environment is itself the locus of a kind of conversation. People look at the material results of their productive activities and learn to see things in common” - Surely Minecraft Versailles is not all that distant.