Computer Simulations as Political Manifestos

Pierre Depaz
Goethe Institute / ZKM Karlsruhe - 2017

Every piece of software has an impact on how we perceive and act in the world, from payment transactions to social media through hospital registration and election ballots. As common points, all of these are directly referencing our reality, acting upon it to change it, with or without direct human input.[i] Software is built to apply rules, to regulate our world, in an asynchronous fashion. Once the code is written and executed, these rules come into action and we are forced to play the game. On the other hand, interactive software, and more specifically computer games and computer simulations, afford real-time input and feedback for its players, allowing them to explore further the intricacies of the self-contained system it is describing.

The difference between general simulations (used, for example, as scientific tools) and simulations as a sub-genre of computer games lies in the stance taken by both the designer and the user. In the first case, there is a desire to draw objective conclusions from the model built, rigorously set up, framed, and interpreted by following logical guidelines. In the second case, the set up (by the designer) and the interpretation (by the player) are both openly subjective in their perspective on the phenomena simulated. Expanding on Ian Bogost's definition[ii] that suggests that "simulation is a gap between the rule-based representation of a source system and a user's subjectivity," I would add that a simulation is also a catalyzer for the designer's political views. This does align with Frasca's definition of simulation as modeling "a system through a different system which maintains some of the behaviors of the original system."[iii] What we will look at, however, is how those behaviors are modeled.

As self-contained systems, simulations only refer to our reality through metaphor and representation, presenting "a version of." At the frontier between reality and fiction, we start to see hints of a simultaneous presence between 'praxis' and 'poïesis,' between the enactment of the interaction and the abstract creativity of a fictional world. Computer simulations, then, allow us to interact with a responsive representation of our world and these formal representations, according to philosophers like Jacques Rancière,[iv] are inherently political. Rancière's work is particularly focused on how representation amounts to the subjective understanding of an objective phenomenon, and as such actively places the individual political subject within its greater community. Representation through simulation is political speech.

Computer-aided scientific simulations are already in use in multiple fields of the social sciences[v] and international relations,[vi] as a means to organize and monitor the evolution of large amounts of data based on a given set of rules laid out by the researcher. Aiming towards scientific (i.e. quantitative) truth, these simulations'

inner workings are closely documented in order to counter the inherent bias of the researcher herself. On the other hand, however, the actuality of game simulations at runtime also holds a certain worldview regarding the environment in which the player evolves, without revealing its source code or acknowledging its political impact.

This article will examine in which aspects digital games as simulations are political, in how developers design and implement them, and how players interact with them, with a focus on the role of source code in that interaction. Drawing on this examination, I will then outline in what ways these simulations can contribute to political and philosophical thought. Throughout this paper, the terms designer and developer will be used interchangeably as I define them both as authors of a system and a software – since the former couldn't exist without the latter.

The political affordances of a simulation

At the core of any game system lies the idea of rule-bounded interaction.[vii] Interaction as question/response, as a dialogue, becomes automated in digital systems, where rules are no longer enacted by a game master, or by a referee, but by code, acting as immovable barriers which frame the agency of the player. Computer simulations are feedback loops regulated by code, as our social and political systems are regulated by law.[viii] As representations of a social phenomenon, computer simulations bestow rights and duties upon the person playing it, going beyond laying out how the virtual world reacts, but also how the player acts. As a formal representation using written, visual, and sonic cues to represent that world, we can again draw on the work of Rancière to make a parallel between aesthetic appreciation and political stance.[ix] It becomes apparent that computer simulations are self-contained pieces of political philosophy, as they not only represent the world, but only do so in a specific way, that of the automated rule.

Even if games are defined by the fact that they are interactive, the introduction of automation and computing in digital games has a very specific consequence. While play, as a historical activity, implied the constant renewal of an implicit agreement between players to follow the rules,[x] the automation of those rules by computers redefines player agency. Instead of enacting the rules by voluntarily following them, it is the player's actions that are being enacted through validation by a constant and automatic monitoring of the game system, frame by frame. By removing player agreement, digital games improve their affordance for apolitical play. If one understands politics as the confrontation and conciliation of subjective perspectives (developer and player), then removing player agreement only fortifies the dominating opinion. If an individual does not constantly renew her consent to being in a system, then the individual cannot legitimately criticize that system. The parallel with Foucault's theory of biopolitics is here easy to make. One of the ways

to ensure complete control of human action within a given frame is by monitoring that said action through quantitative assessment.[xi] This method of assessment, by generalizing individuals to fit them within a designed system, removes political action from those individuals. Action is thus evaluated through number-crunching, and whether these numbers correspond to the expected output or not. To that extent, the game Perfect Woman is a relevant critique of a society in which the posture of a woman's body is the most important factor in how successful she is seen. The player succeeds whenever her body posture matches the array of numbers that are pre-determined by the developer. Even though the software knows nothing about body-language, it is a remarkable enforcer of some of its aspects. In that case, the game is the practical simulation of a machine-like system which determines the physical behaviors of its players through objective, numerical standards while disregarding the individuality of each player. It echoes the inhuman standards to which women are held up to in modern societies, largely based on mensuration and physical appearance.

While one could argue that player creativity is still alive and well in modern computer games, I would like to point out that it is only the scope of the frame that has been enlarged but not the frame itself. In the example of Grand Theft Auto V, a sandbox game, designers and developers claim near-limitless activities. There is a difference between the objects and actors on which you can apply the allowed interaction (the player can kill anyone, drive anywhere) and what the actual interaction is (it is impossible to have a conversation with any non-playable character). If the system does not allow for an interaction, it is as if it does not exist in the virtual world, echoing the idea that you can only choose from what you are being offered. While the input—the action—comes only from the player herself, the output is always altered by the algorithm through which it is processed, and any invalid input is rejected as such and the playful experience comes to a halt.

Any action supposes a reaction from the system which is essentially a sum of the input action and of the way the system has decided to process it—of how the system has decided to qualify the action based on how the developer has decided to interpret it. As such, the player holds a dual role in the system in which she plays. On the one hand, she is still the actant, the source of input for the system, but on the other hand, she is subject of her own output, once processed by the pre-established algorithms, and then has to alter her behavior to fit the expectations of that system. This duality is necessary in order to allow for a playful experience, to allow for the push/pull metaphor so often used to characterize games.

Here we find our first correlation with political philosophy, and politics in general. One of the main topics of that field is the relationship between the individual and the community, and the relationships of power between the two. How much can an individual sacrifice in order to gain from belonging to a group? How much agency can be removed from the player without removing 'a sense of' agency? The

agency of the player in a computer game can be seen as the freedom of the citizen, able to act differently under different regimes, depending on which rules she plays by. The philosophical questioning of the amount of agency one has while being subject to a rule-based society has been at heart of modern political philosophy since its earliest roots in the Renaissance.[xii]

The specific genre of simulation—as an aesthetic category within the broader field of computer games—exhibits a particular characteristic. While other games can be as abstract as they desire (ranging from Tetris to Super Hexagon), simulations have roots in the shared knowledge of phenomena and organizations. They combine the existing formal structures of certain aspects of our societies and the way we look at them through our collective imaginary. The themes of computer simulations can range widely, from the reenactments of physical phenomenon in Flower to the fictional teams composed of real players in the NBA 2K games series.

This relationship between a preexisting phenomenon and its representation through computer games can be seen as a remediation.[xiii] While this concept has been studied extensively in non-interactive media, the idea of representing a same concept with different affordances yields interesting results when it comes to computer simulations. The concept is not only transposed into a computer program, but it is made manipulable. Its output and its linearity are no longer defined, and the player, if limited in the actions that can be taken, is not limited in terms of the goal that has to be reached within these bounds. The remediated, digitalized product can then be assessed from different angles and, by offering more choices, the choices which are left out of the game become more apparent to the player.

Papers, Please is a simulation of a totalitarian system. By representing the tedious acts of following fixed guidelines, the game puts the player in the shoes of an agent of the system itself. As non-playable characters line-up at the player's booth, she has to enforce the rules dictated by a higher authority. This mechanic allows her to see how unadapted and inadequate those rules are when it comes to dealing with the specificity of individual human beings (interestingly enough, the main quality of 'humans' as they are represented in the game is that they cannot always fit in the system that the player represents). The game itself is a critical simulation of arbitrary guidelines. While it leaves enough agency to the player so that she can decide to not play by the rules presented to her, playing by her own rules leads to losing that game -a conflict of perspectives resulting in failure. Because these simulations are built upon the foundations of our external knowledge (how a city works, how a human behaves, how an athlete dribbles, etc.), there is a constant back and forth that happens between pre-existing knowledge of the theory and exploration of the parameters of the model as simulation. For this relationship to appear and sustain, there needs to be a semblance of believability. There needs to be a formal connection between the

model and the representation which happens in the mind of the player. The model of the phenomenon represented is then projected onto the simulation in the form of user input. User input, indeed, is the starting data set needed to run the simulation, and this data set is mostly based on previous experiences or assumptions. Therefore, having a relatable and believable world allows for a user input that would be more truthful, and whose output would be more interesting to look at from a philosophical point of view and more coherent from the player's point of view. If we then take the fact that this input is based on the beliefs and knowledge of the player, then we veer away from the traditional action-reaction-adjustment that happens with other types of games that are more action-intensive and that focus more on intellectual and moral choices. On the other side, the developer itself also works on her assumptions and uses these to provide a believably functional world.

The believability of the simulated world affects the feeling of uncanniness, of discrepancy between an assumed worldview and its programmed counterpart. The uncanny valley,[xiv] the indistinguishable feeling of the lack of reality, forces us to question ourselves or to question the person responsible for the concept presented. This instinctive reaction asks invites a logical exploration of why the representation of the world that the player is interacting with is not aligned with her expectations. Is the action wrong, or is the model wrong?

Joseph Weizenbaum[xv] explores the interesting differences between a theory and a model. A theory is an encompassing concept, while a model is an implementation of that theory, attempting to prove some or all of its aspects. Computer games here provide a model for any kind of theories, and player entertainment through interaction serves as proof of success or failure. Models, in Weizenbaum's opinion, are likely to fail at representing a holistic version of the theory in question for lack of extensiveness.

The word 'system' here can refer both to our past and present political structures, but also to the smaller, self-contained structure of a simulation. Since the game design behind a simulation is very different from other game design practices, especially in terms of how to present players with a goal and with the tools to reach that goal, it is legitimate and expected for players to try and reach their particular, self-assigned goals within a given toolset in a given sandbox environment. What happens there is the apparition of a conflict in worldview between the player and the designer, as the interaction with the developer's model does not fit the player's own model. Computer simulations, then, go a step further in that they allow the confrontation, in real-time, of two appreciations of the world, and let them interact with each other to explore the spaces where discrepancy and, therefore, dialectics and self-questioning can happen.

As every other piece of software, a simulation is written by humans, and its features and functionalities are based on what is deemed necessary by the developer. The political qualities of a simulation stem from two different sources: it

is political because the developer decides to set up rules and because the player has to play by those rules. The specific affordance of simulations as a political artifact is the dual quality of the worldview embedded in them: they are both manipulable, when the simulation is played, and immutable, as they are written in the source code. As the developer encodes their worldview in the source code, the simulation gives an objective, mechanistic quality to a subjective opinion. Exploring the limits of a simulation, then, amounts to exploring the limits of someone else's beliefs. We are now going to take a look at how those beliefs are implemented on the technical side, how they are received by the player, and how this dialogue comes with a new perspective to depict and understand our assumptions.

Programming a simulation is writing a political manifesto

Historically, computers have been invented to facilitate and automate processes that cannot be easily performed by humans – namely, organizing and re-arranging large amounts of data for scientific endeavors. The corollary of this purpose to simplify tasks comes in the obfuscation of the actual way the processes are implemented by providing the end-user with only the output that the developer wants her to see. The evolution of computer science history is largely defined by the addition of abstraction layers designed to simplify the implementation of non-quantitative concepts and contributing to the rise of Object-Oriented Programming. Similarly to Kittler's claim that the last true writers were the electrical engineers designing the 8086 Intel computer chips,[xvi] on which all digital writing is now taking place, the last act of creative writing is made by the developer, who lays down the rules for what can follow.

The process of implementing the algorithms allowing a simulation to run are very relevant to a new form of philosophical practice. Such a practice happens in three different steps. The first one consists in crystallizing the assumptions of the developer during the design process. This is followed by a second step, the software implementation of these assumptions—that is, the act of programming itself. The last step is then their presentation to the user. It is in this last step that specific choices are made that contribute to the acknowledgment—or lack thereof—of the man behind the machine.

Recent work in science and technology studies has shed light on how designers and designed objects embody values within them.[xvii] From the urban planning of Robert Moses designed to prevent certain segments of the population from accessing particular geographic areas[xviii] to internet submission forms only specifying a certain perspective on gender and race, consciously or not, designers embed their worldview within their product. Digital games are no exception to the rule.

The process in which the developer engages when converting abstract ideas into

a digital world is close to sampling and to the cartesian method. The concept of sampling in signal processing is the reduction of a continuous signal to a discrete signal. Sampling occurs when sound is converted from air pressure modulation or light information of a picture into discrete data stored in bytes. One of the challenges of sampling, then, is to represent, as accurately as possible, the original product, whether it is a sound, an image, or an idea. The cartesian method, in this context, can be viewed from the perspective of subdividing a problem or a difficulty into sub-systems, which are assumed to be less complicated.[xix] This influential paradigm can be seen as the philosophical component of emergence, or digital 'gestalt.'
These two concepts can now help us to examine closer the practice of design implementations in computer simulations. On the one hand, there is a necessity of breaking down what we consider human concepts widely used in computer games, such as friendship, antagonism, or exploitation, to mention a few, into machine-readable instructions; while, on the other hand, this practice allows for a closer reading of what those concepts actually are. As a designer implements a mechanic through which, say, a resource is depleted, the actual process of depleting that resource needs to be carefully written down so that the computer can act upon these instructions. Attitudes can range from simply decreasing a counter by a given amount over time (as in most Real-Time Strategy games), to implementing an algorithm which allows resources to thrive both unattended and when taken care of by the players, as in Minecraft[xx].

This practice of coding concepts, of writing them down in words that have a highly-specific, highly-defined meaning, also allows philosophical discussion in a novel way. While the dialogue of philosophy has been mostly built upon different interpretations of previous works, and close readings of past theories, code is interpreted in a single way. The interpreter, which transforms source code into an executable code, is another piece of software which validates the code it reads and approves it as a running action. That interpreter itself has rules for what is valid or not, and these are the only rules the developer must abide by, not those of logic, ethics, or common sense. For a given source code, there is one and only one possible outcome, one possible interpretation. Therefore, when discussing worldview, as expressed through source code, the discussion relies no longer on what source code means but on how source code acts. Computer simulations afford a more active involvement in philosophical thought, by seeing it in action, seeing it 'realized' and 'actualized,' rather than simply conjectured. Since both states of the designer's intent, the source code and the running executable co-exist side by side, they provide a double lens through which we can look at the designer's assumptions and beliefs.

If the designer is writing down immovable rules for a simulation that is presented

to the player, then we can see a correlation to the political philosophers of the Enlightenment. As thinkers of 17th and 18th century Europe started to conceptualize alternative structures of power, away from the existing monarchical and episcopal models, there was a need for new rules under which humans could live. The political manifestos that followed, such as Du Contrat Social,[xxi] The Leviathan,[xxii] or The Two Treatises of Government[xxiii] were endeavors in philosophical thought that tried to devise the best set of rules under which society could operate, given a certain set of assumptions regarding the behaviors and desires of human. If we look at the written source code, we are looking at the algorithmic equivalent of political philosophy from the designer's point of view. The difference, then, is the ability to witness the consequences of these rules, and how they would impact a given population of digital agents. The most obvious of these manifestos is Will Wright's SimCity, where the very idea that lowering taxes makes people happier is presented within the paradigm of the game as an objective truth, while it is widely accepted to be a subjective policy stance in economics.

The main problem that arises from this is the deliberate obfuscation of the mechanics of the simulations when in a game setting. For the sake of user interface, user experience, and user enjoyment, computer simulations only appear to provide a one-way perspective on a dual phenomenon, repeating this process of abstraction disliked by Kittler. Indeed, there is a tendency for designers to either hide the actual formulation of behaviors, or to show it for their own sake and not for what they represent. Algorithms are designed and tuned for optimal behavior in terms of player enjoyment and not always in terms of what the nature of the thing they represent is. The former approach can be found in simulations such as Spore, where the evolution of species is represented in a strictly formal and visual aspect, drawing on common understanding of linear history to let the player fill in the gaps regarding the actual functioning of the system. An opposite case are games such as the Civilization series or the Total War series that are based upon an acute understanding of the numbers that are presented to the player. From that point of view, those numbers-based mechanics exist for their own sake, as tools for the player to manipulate so that she can reach one of the given victory conditions. The qualitative aspect of this victory (whether by war or by peace) is irrelevant to the core playing of the game. Nowhere in this representation is it confirmed that these are the actual numbers, the actual data structures used to represent that world. Because even the system of the game itself is mediated, there is no immediate way for the player to do a close reading of the designer's political assumptions and opinions in their representation of our world. This additional layer at the interface level introduces an interpretation which was absent at first from the bare design of the game, and further pushes the player's conception towards a worldview that is pre-established (in the case of SimCity, urban sustainability[xxiv] and, in the case of Spore, a superfluous understanding of Darwinistic theories) and actually minimizes the political impact of those products.

We've examined the role of the developer as both cartesian and sampler, in that she holds assumptions and political opinions which, in order to be communicated to the player, have to be broken down, written down in code, and interpreted by another piece of software. This act of writing is essential in the examination of computer simulations and computer games as political manifestos. They are the clear and distinct expression of the designer's political intent and define the existence and agency of the player as she interacts with the simulation. However, since dialectical thought is one of the cornerstones of western classical philosophy and interaction one of the pillars of computer games, it would be possible to elaborate a new form of dialectics when establishing a relationship between the static source code and the running simulation but, this time, from the point of view of the player.

Simulation as a political discussion

Since we've established the possibility of a more direct dialogue between player and developer through acknowledgment of biases, we will now look at some particular examples of how that relationship could produce philosophical thought in the case of interaction with computer simulations.

Ludonarrative dissonance, a topic well-discussed in game studies, is the phenomenon of realizing the discrepancy between action and concept[xxv]—that is, what is being afforded by the designer against what is presented by that same designer. From the perspective of the player, the discrepancy would be between the worldview of the designer and the subjective worldview of the player. If suspension of disbelief happens with reality as a reference point—"this can't be real"—, then that suspension is no longer valid when it comes to idealistic representations of the world. The second step of this dissonance would be to wonder why this dissonance exists—that is, why two conceptions of the world are conflicting, what brought them into existence, and what can be learned by this conflict.

Taking the example of the growth of resources in Minecraft, we can look at how they represent human agency and human necessity in relation to natural resources. One way is that resources grow slowly without direct water supply, while human input can modify the topography to allow for a constant stream of water, i.e. for a higher yield. If some aspects of the simulation are left aside, such as bacteria or seasons, it is to highlight the possibility of man to act upon nature as its main factor. Even though phenomenological thought is a prevalent ideology in digital games—since they are user-centered software—, looking at the rule-based behavior of simulations in relation to the player's ideal behavior can help provide new perspectives on a given phenomenon, such as biological growth and reproduction.

The designer's part of this exchange is, however, still static. Even though

emergent behavior can appear during the simulation that was not hard-coded by the designer, there are very few games that modify its actual source code at runtime—Zach Gage's Lose/Lose is one of these, and eventually leads to a computer crash. Therefore, the only discussion that happens while the simulation runs is between the player and herself. She can reflect on her actions, but since her actions will only yield the same result (i.e. whether that action is afforded by the designer or not), there comes a time for a reflexion around the ontology of these actions. From the "why can't I do this," we move into "why am I not allowed to do this?" and finally get to "why am I trying to do this?"—a series of interrogations particularly well stated in Perfect Woman and Papers, Please alike. As such, the awareness of the player of her own actions can only come through an understanding of the designer's desire to funnel these actions in a particular direction, usually away from her desires.

This desire to uncover how a system works recalls not only the political questionings of the Enlightenment but also the overall paradigm of the Enlightenment in terms of knowledge acquisition.[xxvi] If the Newtonian revolution led to a deeper and broader understanding of physics, then it is possible that player curiosity and Ludonarrative dissonance can lead to a more generalized interest in the technicality of computer simulations and more generally to a higher code literacy as long as they acknowledge the existence, the individuality, and, by extension, the subjectivity of the designer of the game.

One of the avenues for political exchange in computer games is to allow the possibility for the players to create content using similar tools as the designers of that game. The modding phenomenon, in which players decided to modify the source code of a given game in order to fit their own play styles (removing instant travel in the Elder Scrolls series) or their own worldview (integrating Islamic State playable units in Arma III to provide a different representation of geopolitical forces[xxvii]), is essentially an intervention from the player's end based on her disagreement with the presented simulation. This possibility does not actually require the software to be entirely transparent or open-source. However, if computer games are to act as manifestos, then their source code should be as movable as the printing press allowed it when it transformed the European media landscape. If there have been examples of open-sourced code bases, these were mostly motivated by technical advances. The release of Doom was mostly centered around the prowess of real-time 3D computer graphics, while the release of the Limbo source code was mostly focused on sound design implementation. Even the arrival in the public domain of the SimCity, one of the most prominent and evidently political simulations, was greeted by possibilities to 'hack' in the original sense—that is, to tinker with technology for the sake of tinkering with technology instead of tinkering with political statements.

This also implies an awareness from the designer and developer's point of view in terms of their impact in the world. Several authors in the field of software studies

have taken a look at the consequences of coded behaviors in our physical and semantic world. While most of the work does not directly refer to digital games, the rhetoric is the same: intangible code affects the tangible world.[xxviii] Particularly in terms of speech, game designers have to realize that executing code is an action, and that this action has consequences that are, if not material, then at least moral and political, as they bind players in arbitrary rules.

These two opposite perspectives, then, lie dormant within a simulation. Only through mutual desire to acknowledge the role of the other—as designers acknowledge their impact on the player's worldview and ability to interact with that world and as players acknowledge the presence of subjective individualities behind the scripts—can it be possible to achieve a philosophical discussion that resembles that of classical philosophy.
In the end, the written code of the simulation exists as the designer's expression of politics, and as an opportunity for the player to acknowledge that, while interactive and affording some agency, the running code is a subjective perspective on the world around us which should provide more questions than hard-coded answers.

i Kitchin Rob and Martin Dodge. *Code/Space: Software and the everyday life*. Cambridge, Mass.: MIT Press, 2011.

ii Bogost, Ian. *A Response To Critical Simulations*. Retrieved from http://bogost.com/writing/a_response_to_critical_simulat/, 2004

iii Frasca, Gonzalo. "Simulation versus Narrative." Wolf, Mark J. P. and Bernard Perron. *The Video Game Theory Reader*. New York: Routledge, 2003. 221-236.

iv Rancière, Jacques. *The Politics of Aesthetics: The Distribution of the Sensible*. London, New York: Continuum International Publishing Group, 2004.

v Epstein, Joshua. "Modeling Civil Violence: An Agent-Based Computational Approach." *Proceedings of the National Academy of Sciences of the United States of America*. May 14, 2002, Vol. 99: 7243-7250.

vi Cederman, Lars-Erik. *Emergent Actors in World Politics: How States and Nations Develop and Dissolve*. Princeton, N.J.: Princeton University Press, 1997.

vii Salen, Katie and Eric Zimmerman. *Rules of Play*. Cambridge, Mass.: MIT Press, 2003.

viii Wiener, Norbert. *The Human Use of Human Beings: Cybernetics and Society*. New York, N.Y.: Da Capo Press, 1988.

ix Rancière, Jacques. *The Politics of Aesthetics: The Distribution of the Sensible*. Op.cit.

x Huizinga, Johan. (1949) *Homo Ludens: A Study of the Play-Element in Culture*. Eastford, C.T.:Martino Fine Books, 2014.

xi Foucault, Michel. *Discipline and Punish*, Random House, 1975.

xii La Boétie, Étienne., *Discourse on Voluntary Servitude*. Retrieved from http://www.constitution.org/la_boetie/_serv_vol.htm

xiii Bolter, Jay David. *Remediation: Understanding New Media*. Cambridge, Mass.: MIT Press, 1998.

xiv Freud, Siegmund. *The Uncanny*. Retrieved from http://web.mit.edu/allanmc/www/freud1.pdf

xv Weizenbaum, Joseph. *Computer Power and Human Reason: From Judgment to Calculation*. San Francisco, Ca.: W.H. Freeman and Company, 1976.

xvi Kittler, Friedrich. *There Is No Software*. 1995. Retrieved from www.ctheory.net/articles.aspx?id=74

xvii Nissenbaum, Helen and Mary Flanagan. *Values at Play in Digital Games*. Cambridge, Mass.: MIT Press, 2015.

xviii Kroes, Peter and Peter-Paul Verbeek (Eds.). *The Moral Status of Technical Artifacts*. Heidelberg, New York, London: Springer, 2014.

xix Descarted, Réne. *Discours de la méthode*. Retrieved from www.gutenberg.org/etext/13846

xx

xxi Rousseau, Jean-Jacques. *Du Contrat Social*. Retrieved from http://classiques.uqac.ca/classiques/Rousseau_jj/ contrat_social/Contrat_social.pdf

xxii Locke, John. *The Two Treatises of Government*. Retrieved from http://oll.libertyfund.org/titles/222

xxiii Hobbes, Thomas. *The Leviathan*. Retrieved from https://www.gutenberg.org/files/3207/3207-h/3207-h.htm

xxiv Kolson, Kenneth. "The politics of SimCity." *Political Science and Politics*. Mar 1, 1996. Vol 29(1). 43.

xxv Upton, Brian. *The Aesthetic of Play*. Cambridge, Mass.: MIT Press, 2015.

xxvi Kant, Immanuel. *What Is Enlightenment?* Retrieved from http://www.columbia.edu/acis/ets/CCREAD/etscc/ kant.html

xxvii Hall, John. "The Videogame That Allows You To Play As An ISIS Fighter." *Daily Mail*. Retrieved from http:// www.dailymail.co.uk/news/article-2937641/ISIS-fighters-distributing-video-game-allows-players- play-role- Islamist-kill-Westerners.html

xxviii Cox, Geoff and Alex McLean. *Speaking Code*. Cambridge, Mass.: MIT Press, 2014.

Games

Cavanagh, Terry. Super Hexagon. 2012.

Chen, Jenova. Flower. 2009.

Electronic Arts. NBA 2k (series). 1999-current.

Gage, Zach. Lose/Lose, 2009.

id Software, Doom. 1993.

Lu, Peter and Lea Schoenefelder. Perfect Woman. UCLA Game Lab, 2014.

Pajintov, Alexey. Tetris. 1984.

Persson, Marcus. Minecraft. Mojang. 2009.

Pope, Lucas. Papers, Please. 3909. 2013.

Rockstar Games. Grand Theft Auto V. 2013.

Maxis , SimCity, 2016

Bethesda Games , Elder Scrolls, 1997-2014

Maxis , Spore, 2008

Firaxis Games, Civilization, 1999-2017

Assembly Games, Total War, 2002-2017