



HAL
open science

Computer Simulations as Political Manifestos

Pierre Depaz

► **To cite this version:**

Pierre Depaz. Computer Simulations as Political Manifestos. Games and Politics, Goethe-Institute, 2016. hal-04276949

HAL Id: hal-04276949

<https://hal.science/hal-04276949v1>

Submitted on 9 Nov 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License

Computer Simulations as Political Manifestos

Pierre Depaz

Goethe Institute 2016

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, without or without direct human input¹. 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. As self-contained systems, they 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 Rancière², are inherently political. Rancière’s work was particularly focused on how representation amounts to the subjective understanding of an objective phenomenon, and as such represents 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 sciences³ and international relations⁴, in order 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.

¹ KITCHIN, R., DODGE, M., *Code/Space: Software and the everyday life*, MIT Press, 2011

² RANCIÈRE, J., ZIZEK, S., *The Politics of Aesthetics: The Distribution of the Sensible*, Continuum International Publishing Group, 2004.

³ EPSTEIN, J., *Modeling Civil Violence: An Agent-Based Computational Approach*, Proc. Natl. Acad. Sci. USA. 2002

⁴ CEDERMAN, L.-E., *Emergent actors in world politics: how states and nations develop and dissolve*, Princeton University Press, 1997

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 hold 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 are computer simulations 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 can computer simulations 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 -as the former couldn't exist without the latter.

I. The political characteristics of building a simulation

At the core of any game system lies the idea of rule-bounded interaction⁵. 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 law⁶. As representations of a social phenomenon allowing for, 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 stance⁷. It becomes apparent that computer simulations are self-contained pieces of political philosophy, as they represent the world, but only do so in specific, ontologically limited ways.

Even if games are defined by the fact that they are interactive, the introduction of automation and computing in digital games have a very specific consequence. While play as a historical activity implied the constant renewal of an implicit agreement between players to follow the

⁵ SALEN, K., ZIMMERMAN, E., *Rules of Play*, MIT Press, 2003

⁶ WIENER, N., *the Human use of human beings: Cybernetics and Society*, Da Capo Press, 1988

⁷ RANCIERE, J., ZIZEK, S., *The Politics of Aesthetics: The Distribution of the Sensible*, op. cit.

rules⁸, the automation of those rules by computers redefine player agency. Instead of enacting those 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 an individual does not constantly renew its 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 assessment⁹. Action is evaluated through number-crunching, and whether these numbers correspond to the expected output or not. To that extent, the game *Perfect Woman*¹⁰ 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 systems who determine 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 mensurations 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, and still not the frame itself. In the example of *Grand Theft Auto V*¹¹, 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 is the actual interaction (it is impossible to have a conversation with any non-playable character). If the system does not allow for it, it is as if it doesn't 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 then always altered by the algorithm through which it is processed, and any invalid input is rejected as such and the playful

⁸ HUIZINGA, J., *Homo Ludens: A Study of the Play-Element in Culture*, Martino Fine Books, 2014

⁹ FOUCAULT, M. *Discipline and Punish*, Random House, 1975

¹⁰ LU, P., SCHÖNEFELD, L., *Perfect Woman*, UCLA Game Lab, 2014

¹¹ Rockstar Games, *Grand Theft Auto V*, 2013

experience comes to a halt. Any action supposes then a response 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 it 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 each other. 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 Renaissance¹².

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*¹³ to *Super Hexagon*¹⁴), simulations have roots in the shared knowledge of phenomena and organizations. They do so by combining 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*¹⁵ to the fictional teams composed of real players in the *NBA 2K*¹⁶ games series.

¹² LA BOETIE, E., *Discourse on Voluntary Servitude*, retrieved from http://www.constitution.org/la_boetie/serv_vol.htm

¹³ PAJNITOV, A., *Tetris*, 1984

¹⁴ CAVANAGH, T., *Super Hexagon*, 2012

¹⁵ CHEN, J., *Flower*, 2009

¹⁶ ELECTRONIC ARTS, *NBA 2k (series)*, 1999-current

This relationship between a preexisting phenomenon and its representation through computer games can be seen as a remediation¹⁷. 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, its linearity, are no longer defined, and the player, if limited in the actions that are 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 since, by offering more choices, the choices which are left out of the game become more apparent to the player.

*Papers, Please*¹⁸ 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 leaving enough agency to the player so that she can decide to not play by the rules presented to her, but by her own rules, enough this leads to losing that game. Because these simulations are built upon the foundations of our external knowledge (how a city works, how a human behaves, how an athlete dribbles, ...), 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, than we veer

¹⁷ BOLTER, D. J., *Remediation: Understanding New Media*, MIT Press, 1998

¹⁸ POPE, L., *Papers, Please*, 2013

away from the traditional action - reaction - adjustment that happens with other types of game that are more action-intensive, and 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.

When the starting dataset of a simulation is processed and outputs a situation that does not conform to those assumptions, the usual reaction of the player is not to understand what was wrong in the execution of the action, but in the assumption behind that action. The believability of that world increases the feeling of uncanniness, of discrepancy between an assumed worldview and its programmed counterpart. The uncanny valley¹⁹, the indistinguishable feeling of the lack of reality forces us to question ourselves, or to question the person responsible for the concept presented. From an instinct could form 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 Weizenbaum²⁰ 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 digital 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 doesn't fit the player's own model. Computer simulations, then, go a step further in that they allow to confront, in real-time, 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.

¹⁹ FREUD, S., *The Uncanny*, retrieved from <http://web.mit.edu/allanmc/www/freud1.pdf>

²⁰ WEIZENBAUM, J., *Computer Power and Human Reason: From Judgment to Calculation*, W.H. Freeman and Company, 1976

The way the system would either confirm or infirm the assumptions of the player is not objective. As every other piece of software, it is written by humans, and its features and functionalities are based on what is deemed necessary by the developer. As the developer encodes their worldview in the source code, it gives an objective 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, from the developer's stand point, how they are received by the player, and how this dialogue comes with a new perspective to depict and understand our assumptions.

II. The computer code as a new 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 chips²¹, 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.

As such, 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, which 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, in which 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 them²². From the urban planning of Robert Moses designed to

²¹ KITTLER, F., *There Is No Software*, 1995, retrieved from www.ctheory.net/articles.aspx?id=74

²² NISSEMBAUM, H., FLANAGAN, M., *Values at Play in Digital Games*, MIT Press, 2015

prevent certain segments of the population from accessing particular geographic areas²³ to internet submission forms only specifying a certain perspective on gender and race, designers embed their worldview within their product, whether or not they do it consciously. Digital games are no exception to the rule. Furthermore, belonging to the more emotional gradient of software products, they achieve subjective connections with their players through the medium of what is ultimately written code. This emotional connection tends to obfuscate the possibility for self-reflexion and political inquiry. If the designer's intent is to create such an environment based on his conception of what can bring joy, fear, excitement or surprise, the problem remains in the implementation of this intent.

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. Indeed, the process in which the developer engages when converting his abstract ideas is both close to sampling and to the cartesian method. 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 complicated²⁴. 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, which I define here as the representation of the world using data structures. On the one hand, there is this necessity of breaking down what we consider human concepts widely used in computer games, such as friendship, antagonism, exploitation, etc. 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 players²⁵. The attitude with which the developer approaches that

²³ KROES, P., VERBEEK, P.-P., *The Moral Status of Technical Artifacts*, Springer, 2014

²⁴ DESCARTES, R., *Discours de la méthode*, retrieved from www.gutenberg.org/etext/13846

²⁵ PERSSON, M., *Minecraft*, Mojang, 2009

implementation problem is inherently political. Writing concepts to which the players can relate in the medium of computer code opens this possibility for self-reflection, that is both expressive (because of the fact that the code runs, because it *executes*) in the present moment, but can also be referenced in the future, since it is written down and archived.

This practice of coding concepts 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, allowing to build an executable from source 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, 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, then, 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*, *The Leviathan* or *The Two Treatises of Government* were endeavors in philosophical thought where the authors 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 human²⁶²⁷²⁸. If we look at the written source code, then 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 assumptions-based rules, and how they would impact a given

²⁶ ROUSSEAU, J.-J., *Du Contrat Social*, retrieved from http://classiques.uqac.ca/classiques/Rousseau_jj/contrat_social/Contrat_social.pdf

²⁷ LOCKE, J., *The Two Treatises of Government*, retrieved from <http://oll.libertyfund.org/titles/222>

²⁸ HOBBS, T., *The Leviathan*, retrieved from <https://www.gutenberg.org/files/3207/3207-h/3207-h.htm>

population of digital agents. The most obvious of these manifestos is Will Wright's *SimCity*²⁹, 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 is the nature of the thing they represent. 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. On the opposite, games such as the *Civilization* series or the *Total War* series are based upon an acute understanding of the numbers that are presented to the player. From that point of view, they 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 of this desire to mediate even the system of the game itself, 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 re-introduces a level of interpretation that was absent to begin with, and further pushes the player's conception towards a worldview that is pre-established (in the case of *SimCity*, urban sustainability²⁴ 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 and 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

²⁹ KOLSON, K., *The politics of SimCity*, American Political Science Association, 1996

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 that simulation. However, since dialectical thought is one of the cornerstones of western classical philosophy and interaction one of the pillars of computer games, then 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.

III. 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.

Ludo-narrative dissonance, a topic well-discussed in game studies, is the phenomenon of realizing the discrepancy between action and concept³⁰ –that is, what is being afforded by the designer against what is presented by that same designer. However, if we look at it from the perspective of the player, we have that discrepancy 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 Minecraft³¹, 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, that is, 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

³⁰ UPTON, B., *The Aesthetics of Play*, MIT Press, 2015

³¹ PERSSON, M., *Minecraft*, op. cit.

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 wasn't hard-coded by the designer, there are very few games that modify its actual source code at runtime -Zach Gage's *Lose/Lose*³² 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), 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 acquisition³³. If the newtonian revolution led to a deeper and broader understanding of physics, then it is possible that player curiosity and ludo-narrative 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 they 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 forces³⁴), is essentially an intervention from the player's end based on her

³² GAGE, Z., *Lose/Lose*, 2009

³³ KANT, I, *What Is Enlightenment?*, retrieved from <http://www.columbia.edu/acis/ets/CCREAD/etscc/kant.html>

³⁴ HALL, J., *The Videogame That Allows You To Play As An ISIS Fighter*, retrieved from <http://www.dailymail.co.uk/news/article-2937641/ISIS-fighters-distributing-video-game-allows-players-play-role-Islamist-kill-Westerners.html>

disagreement with the presented simulation. This possibility doesn't 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 has been examples of open sourced code bases, these have mostly motivated by technical advances. The release of *Doom*³⁵ 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 world³⁶. 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. Examining the code through its step-by-step instructions act as the reading of a manifesto, while acknowledging this role as a cultural and, ideally, public impact would force the designer to phrase their logic, their system in a way that is both readable and executable.

In conclusion, we've seen that computer simulations are political artifacts as they represent livable and relatable phenomena. They are remediating that phenomenon and representing it to the player so that the player can start experimenting with it based on her previous knowledge. As all computer games, simulations are based on a loop of inputs and outputs. It is the nature of

³⁵ id Software, *DOOM*, 1993

³⁶ COX, A., *Speaking Code*, MIT Press, 2014

the input and the process of the output that differentiates simulations from the rest. As such, simulations represent the meeting point of two perspectives on the represented situation, that of the designer and that of the player, coming from separate directions with different subjectivities. This political process of representing the world as a simulation comes, for the designer, through the writing of code. This act of coding concepts can be understood through the dual lens of sampling and of cartesian method. The break down of continuous, human concepts into unambiguous source code allows for a historically unique communication of one's ideas and point of view. Source code is objectively understandable and not objectively true, and as such can therefore yield a different approach to examining political concepts as they are enacted by a computer. Yet, this process is still hidden to the player, where what is represented procedurally isn't always what is represented visually or semantically. The player is then left on her own side of this potential dialogue, where the only possible philosophical thought is self-reflexive. Therefore it seems that, in order to be fully realized as a political discourse, developers must make their work accessible, so that close reading by the public becomes possible and that players can put in perspective their particular political stance on the represented world with both static and running code, which embodies the belief of another subjectivity and contributes to an ever-improving public dialogue.