GAME ENGINES AS PREHENSION OF INCOMPUTABLE DATA: THE PROCESSUAL EFFECTIVITIES OF GAME ENGINES

Dr. Damien Charrieras, School of Creative Media, City University of Hong Kong, Level 7, 18 Tat Hong Avenue, Kowloon Tong, Hong Kong
email: <dcharrie@cityu.edu.hk>

Abstract

Our research addresses the impact of technological mediations on the contemporary creative practices of production in interactive games and new media arts. More specifically, our research focuses on game engines. Caught between different actors, cultures, organizations and functions, video game engines are cultural and socio-technical objects whose complex nature mirrors the multiple and competing definitions of video games and has similarly broad ranging cultural, social and economic impacts. The paper highlights the necessity to go beyond the current research on game engines, and outlines a new way to conceptualize them through a discussion of the studies of Ian Bogost and Luciana Parisi.

Keywords: video game engines, circulation, creative practices, video game industries, new media art

What are the specificity/effectivities of interactive games as a medium? This question has been addressed by numerous researchers in the field of game/new media studies who emphasize the procedurality of video games [1] and/or their algorithmic nature [2]. Beyond Game Studies, answering this question pertaining to the practices of video game production and the specificity of technological tools used has larger cultural, social and economic implications [3]. Video game production through specific technological tools is a contested space in which many institutional and non-institutional actors play underappreciated roles [4]. Aware of the latest developments, the originality of our approach, anchored in a material analysis of new technologies [5]–[10], is to address this question through the study of game engines, rather than through the study of the specific form of interactive narratives found in video games [11], or through the ludic qualities of video games [12]. We contend that the effectivities of video games are not solely determined by video game publisher companies, nor by a specific professional culture of video game designers, but also and fundamentally by the specificities of some of the technologies used to produce video game technologies; we focus here on the video game engines.

A game engine is a software framework facilitating the creation of video games through a set of functionalities that enable to automatize some programming tasks involved to handle graphics, sound and artificial intelligence processes during the production of video games. Current research on video game engines consists on the one hand of case studies of game engines used by non-market game developers to modify the mechanics of existing video games [13]. On the other hand, some researchers are studying the internal design of game engines as a software [14]. However, to date there is no study of game engines that links their technological features and their specific mode of being to the ways those game engines circulate and have different ranges of effectivities.

Effectivities of video game engines

In the early days of video games, project teams produced video games directly in machine coding; later a number of middleware tools were created to support the process of production of video games. These technological tools were aimed at reducing the huge number of programming routines required by video games given their computational nature, and to enable a quick prototyping of the game; Aric Wilmunder, one of the coders of the game engine SCUMM, explains in an interview: "One of the great benefits of SCUMM was how quickly a game could be prototyped. The designer would have ideas for rooms and locations and the lead background artist could start doing sketches. When enough of the sketches were done, they would get scanned in and you could very quickly add and connect them up using SCUMM" [15].

Video game engines are complex objects, used inside the video game industries as well as outside of them. Some of them are produced by specialized (middleware) companies [16] which subsequently license their engine to other companies, such as CryEngine, produced by the company Crytek to conceive the game Crysis 2. Some other game engines are freeware - Source, Unity [17] - and can be used by people outside the video game industries, and therefore are at the same time professional tools developed along a business-to-business logic, and end user products taken up by consumers or players to create their own games. Assessing their effectivities is not an easy task, Despite the utopian discourses surrounding their production, they cannot be considered as technologies where engineers enclose scripts of usage to be taken up by users [18], [19]. Nor can it be contended, despite the many tutorials, manuals and formations on game engines (Unity is a case in point), that they are mere tools in the hands of game designers, lending themselves to the transparent actualization of the game designers' (creative) intentions. Whatever the complexities of projections or mutually rationalized anticipations that can be imposed on video game engines (in part due to their strategic position in the financial economy of video game production), their effectivities must be considered beyond the limitative relationship between video game engines as tools and human intentionality.

Mackenzie's definition of software can aid understanding of the complex nature of the video game engine as a techno-cultural object:
"At stake here is an account of software as a highly involuted, historically media-specific distribution of agency. This account diverges from a general sociology of technology in highlighting the historical, material specificity of code as a labile, shifting nexus of relations, forms and practices. It regards software formally as a set of permutable distributions of agency between people, machines and contemporary symbolic environments carried as code. Code itself is structured as a distribution of agency" [20].
What is the media-specific distribution of agency enabled by video game engines? To answer this question we must pay special attention to the algorithmic nature of video game engines themselves, leading us to consider game engines as "mechanisms for the processing and calculation of quantities of data, rather than instruments for the production of qualities/effects." [21], rather than only as tools that mimic the existing worlds through the use of 3D graphics. We contend here that the effectivities of video game engines are related to their specific modes of being (algorithmic objects) that enable what Luciana Parisi and Portanova call an "aesthetic of soft thought". We sketch a new way to conceptualize game engines through a discussion of the research of Ian Bogost on video games, and of Luciana Parisi on contemporary architecture and new media art.

Video game engines and unit operation and the expressive power of video games

For Bogost, procedurality enables video games to be expressive through rules,

interactions between the player and the video game, and processes. Video games differ from other forms of creation in that in the video game "arguments are made not through the construction of words or images, but through the authorship of rules of behavior, the construction of dynamic models" [22]. Hence procedurality is not limited to functional task according to Bogost, and certain procedures related to game playing can convey an expressive power; we use computational power not only to produce better pictures in Photoshop or to write texts in Word, but also to have an aesthetic emotion linked to the expressive power of the procedures we undertake as a player. In other words, "videogames are computational artifacts that have cultural meaning as computational artifacts" [23].

Bogost makes clear that gameplay, hence procedural rhetoric, is linked to the video game engines used to produce games: "The notion of a common substructure for similar games grew into modern game engines, component-based software systems useful not only for rendering background effects like physics, but also for orchestrating the crucial functions of the game- play itself" [24]. To Bogost, video game engines are not just neutral technological tools that will passively house creative expressions of game designers; on the contrary, they play an important role in defining what video game are. "… game engines regulate individual videogames' artistic, cultural, and narrative expression" [25]. The role of game engines in video game production is important for Bogost's development of his theory of unit operations, at the crossing of literary criticism and new media studies. Bogost describes how the poems of Baudelaire were symptomatic of a turn toward "more and more compact modes of representation" in which there occurs a decoupling of human experience and creative work from their continuity in rituals and social abundance [26]:

"My contention is that as this very modern experience moves from an experience of crisis in the mid-nineteenth century to an experience of banality in the twenty-first century, it becomes compressed into more and more compact modes of representation. Baudelaire does not merely author a poem; he also creates a unit of cultural memory, a tool that others can make fungible as a performance of the modern life … . Together, Baudelaire's lyric encapsulates these figures and tropes into a framework, or rule set, for living the modern life. Benjamin calls these rules motifs. I would call them unit operations" [27].

Rather than authoring a poem, Baudelaire is crafting a unit of cultural memory that others can integrate into the performance of modern life [28]; these units of cultural memory are what Bogost calls unit operations. He views the video game engine technologies as central in the contemporary relevance of the concept of unit operation, saying that "the game engine dramatically increases the scope of unit-based abstraction compared to other forms of cultural production" [29]. For him, ultimately, in studying mechanics of video games such as GTA and focusing our attention on unit operations that are the main modes of representation in video games [30], we should strive to "understand and refine each unit operation of our possible actions so we can interrogate and improve the system of human experience." [31]. Bogost invites us to study software itself through a close study of "[p]ublicly documented hardware and software specifications, software development kits, and decompiled videogame ROMs …" [32].

The concepts of procedural rhetoric and unit operation do a wonderful job in accounting for the way in which the player is drawn into an aesthetic experience in a video game, beyond the literary mode of engagement, and the ways in which players can experience emergence through the open ended world of GTA [33]. The concept of unit operation aids understanding of the expressive power of the video game (especially through game engines), where the aesthetic experiences of the players interact with the intentions of the video game designers. Nevertheless, if our main interest lies in the study of the technocultural processes pertaining to video game engines, and how effectivities arise from the interaction of technologies and human practices, we must shift our emphasis from the phenomenology of the players and the game designers that underscore the study of Bogost in *Unit Operation*. The effectivities of game engines must be envisioned beyond their uses as tools by game designers and companies, and beyond their expressive power toward players - beyond modes of representation and human experience; we must examine the unintended consequences of their complex internal dynamism, uncoupled from human agency, unaccountable through phenomenology, and grapple with the complexities of their algorithmic nature - their *soft thought.*

Video game engines as prehension of incomputable data

In Unit Operation, Bogost emphasized the intentionalities of video game designers and the aesthetic experience of the players vis-à-vis unit operations. If we were to use the same theoretical frameworks to study the effectivities of video game engines, we would rely "on the interaction of/with biophysical data in order to explain change." [34] Such a view restricts the range of effectivities displayed by video game engines. The way in which Bogost frames the concept of unit operation doesn't allow us to take into account the internal dynamics of video game engines, the ways in which algorithms become actual entities through the prehension of incomputable data – a process which occurs in a specific spatiotemporal structure that is not the "present state of the world" (duration) in which bodies and matter interact in a linear and efficient causality [35]. Parisi contrasts those conceptions of interaction with the recent development in nanoarchitecture (and especially Anders Christiansen's nanoarchitecture design of *Homeostatic Membrane*):

"As opposed to interactive architecture, according to which spatiotemporal experience is defined by a change in the system induced by biophysical data, nanoarchitectures are spatiotemporal structures of anticipation characterized by incomputable data, corresponding neither to mathematical nor to physical inputs. From ubiquitous computing to the nanofabrication of walls, smart objects, and clothes that sense and anticipate (or productively prerespond to) changes in atmospheric pressures, moods, sounds, images, colors, and movements, incomputable data have infected the general ecology of media systems" [36].

In Parisi's view algorithms aren't just a mode of computation of "real" data overlaying the reality, but constitute building blocks of reality through diverse process of prehension of (incomputable) data. The algorithm isn't just a specific mediation rerouting the existing reality; rather, it generates in and of itself new forms of spatiotemporal realities.

Thus, video game engines shouldn't be considered as a set of predefined possibilities that are materialized through situated practices in a given context; the relationship between them and the con-

texts in which they are produced and used is non-trivial. The internal dynamism of game engines as algorithmic and unstable objects has to be accounted for if we want to understand the whole range of their effectivities; the "soft thought" inherent to them has to be accounted for if we want to understand the ways in which they transduce their diverse contexts of production and circulation.

The effectivities of video game engines as a set of algorithms should not be limited to the intentionalities that the producers of video game engines put into it, or to the state of technology, or to the will of the video game designer using it as a tool. Following Parisi, we have to acknowledge the internal dynamics of video game engines; they have dynamism of their own which escapes their context of use and production, and human perception most of the time - which is why Parisi claims to do a non-phenomenological study of algorithms [37]. They have emergent properties, the singularity of which is linked to the algorithmic nature of video game engines: "I will propose that algorithmic architecture needs to be explained through another kind of aesthetics, relying neither on the beauty of simpler axioms nor on the continual variation of biophysical interactions. On the contrary, algorithmic architecture is important because it offers us an opportunity to discuss another species of actualities: algorithmic objects, the data structures of which now constitute the immanent data of experiences that do not stem from the directly lived" [38].

Parisi and Portanova emphasize the autonomy of code and the specificity of its aesthetics, where aesthetics refers to a mode of "soft thought" inherent to algorithmic objects and data structures, not to the ways in which code is phenomenologically perceived by humans. Game engines are such algorithmic objects and lend themselves to this "soft thought."

What are the effectivities of the production/circulation of game engines inside a complex media ecology that includes companies specializing in their production, game publishers, independent game developers, modders who modify video game content, and digital artist communities using video game engines to produce works of art? [39] This circulation, linked to the internal dynamism of video game engines (the specificities of the aesthetics of their code), continuously redefines the game engine at a technological level (the set of functions they can perform as a software technology) and at a cultural level (the creative practices with which such engines are enmeshed). Video game engines, given their algorithmic nature, shouldn't be considered mere simulations of physical data, but actual entities, becoming actual through prehension of incomputable data [40]. This leads us to focus on the processes that constitute continuously the game engines, their *endurance* [41], as well as on the effectivities of game engines on the media ecology pertaining to its scattering/distribution. A better description of the circulation and effectivities of game engines between internal/external heterogeneous entities is crucial to understand the multiple ways in which interactive games inform, in singular ways, the contemporary forms of cultural production.

References and Notes

1. I. Bogost, *Persuasive games: videogames andprocedural rhetoric*. Cambridge, Mass.; London: MIT Press, 2007.

2. A. R. Galloway, *Gaming: Essays On Algorithmic Culture*, 1st ed. University of Minnesota Press, 2006.

3. E. Castronova, *Synthetic worlds: the business and culture of online games*. Chicago: University of Chicago Press, 2005.

4. C. Poremba, "Discourse Engines for Art Mods," *Eludamos J. Comput. Game Cult.*, vol. 4, no. 1, 2010, pp. 41–56.

5. T. S. Barker, *Time and the Digital: Connecting Technology, Aesthetics, and a Process Philosophy of Time*. Dartmouth, 2012.

6. S. Cubitt, *Digital Aesthetics*, 1st ed. Sage Publications Ltd, 1998.

7. M. Fuller, *Media Ecologies: Materialist Energies in Art and Technoculture*, Illustrated edition. MIT Press, 2005.

8. A. MacKenzie, *Transductions: Bodies and Machines at Speed*, New edition. Continuum, 2006.

9. A. Munster, *Materializing New Media: Embodiment in Information Aesthetics*, Annotated edition. Dartmouth, 2006.

10. L. Parisi, *Contagious Architecture: Computation, Aesthetics, and Space*. MIT Press, 2013, p. 25.

11. M.-L. Ryan, *Avatars Of Story*, 1st ed. University of Minnesota Press, 2006.

12. E. Aarseth, "Computer Game Studies, Year One," *Game Stud. Int. J. Comput. Game Res.*, vol. 1, no. 1, 2001.

13. D. B. Nieborg and S. van der Graaf, "The mod industries? The industrial logic of non-market game production," *Eur. J. Cult. Stud.*, vol. 11, 2008, pp. 177–195.

14. D. S. Evans, A. Hagiu, and R. Schmalensee, *Invisible Engines: How Software Platforms Drive Innovation and Transform Industries*. MIT Press, 2008.

15. M. Bevan, "The SCUMM Diary: Stories behind one of the greatest game engines ever made," *Gamasutra*, 12-Jul-2013. [Online]. Available: http://www.gamasutra.com/view/feature/196009/the_scumm_diary_stories_behind_.php?page=3. [Accessed: 16-Jul-2013].

16. D. Charrieras, "Les médiations sociales, culturelles et technologiques dans la production et l'appropriation des intergiciels de l'industrie du jeu vidéo au Canada," *Commun. Inf. Médias Théories Prat.*, no. Vol. 29/1, Oct. 2011.

17. "List of game engines," *Wikipedia, the free encyclopedia*. 10-Jul-2013.

18. M. Akrich, "The De-Scription of Technical Objects," in *Shaping Technology, Building Society: Studies in Sociotechnical Change*, Wiebe Bijker and John Law. Cambridge, Mass.: MIT Press, 1992, pp. 205–224.

19. L. Suchman, *Human-Machine Reconfigurations: Plans and Situated Actions*. Cambridge University Press, 2007. pp. 191–192.

20. A. Mackenzie, *Cutting code: software and sociality*. New York: Peter Lang, 2006, p. 19.

21. L. Parisi and S. Portanova, "Life and mind: From autopoiesis to neurophenomenology. A tribute to Francisco Varela," *Phenomenol. Cogn. Sci.*, vol. 3, no. 4, 2011, pp. 381–398.

22. I. Bogost, *Persuasive games: the expressive power of videogames*. Cambridge, MA: MIT Press, 2007. p. 29. Cf. also p. 11, p. xiii.

23. Bogost [22] p.ix.

24. I. Bogost, *Unit Operations: An Approach to Videogame Criticism*. MIT Press, 2006, p. 55

25. Bogost [24] p.56.

26. Bogost [24] p.74.

27. Bogost [24] p.73-74.

28. Bogost [24] p.73-74.

29. Bogost [24] p.55.

30. Bogost [24] p.64

31. Bogost [24] p.169

32. Bogost [22] p.63

33. Bogost [24] p.166.

34. Parisi [10] p.25.

35. A. N. Whitehead, *Process and Reality*, 2nd Revised edition. Free Press, 1979, p. 487.

36. Parisi [10] p.26.

37. Parisi [10] pp.69-70.

38. Parisi [10] pp.21-22.

39. P. Krapp, *Noise Channels: Glitch and Error in Digital Culture*. University Of Minnesota Press, 2011.

40. Parisi [10] pp.25.

41. Barker [5]