Selected Papers of #AoIR2019:
The 20th Annual Conference of the
Association of Internet Researchers
Brisbane, Australia / 2-5 October 2019

VIDEOGAME ENGINES AND THE POLITICS OF 'DEMOCRATISED' SOFTWARE DEVELOPMENT

Benjamin Nicoll
Queensland University of Technology

Modes of production in the videogame industry have undergone fundamental changes over the past decade. Once upon a time, in the 1990s and early 2000s, videogame studios would create bespoke development tools, assets, and engines from scratch, for use on internal projects. With a few notable exceptions, companies did not share their tools (unless leasing for a fee) or openly trade industry knowledge (O'Donnell, 2014; see also Darchen, 2016: 213-214 for an Australian perspective). For the most part, development was slow, secretive, resource-intensive, and open only to those with access—financially and institutionally—to the proprietary 'toolchains' through which to produce and publish videogame content. Since the early 2000s, however, several low-cost and low-barrier-to-entry game-making 'engines'—such as 'Unity' and 'Unreal'—have transformed the way people make videogames content. A game engine is a *software tool* that enables interactive digital content to be built, and a *code framework* that enables that content to run on different platforms, including consoles, smartphones, and virtual reality devices. Game engines now form the backbone of videogame development and, increasingly, software development more broadly. The Unity engine—a key player in this industry, and the main case study of this paper—aims to 'democratise game development' through an accessible editing interface, a flexible licensing structure, and a toolset that is interoperable with a range of different design tools, middleware software, programming languages, and production workflows.

This paper evaluates the core claim made by and about Unity—that it is has democratised game development—through a framework that analyses the engine's 'articulations' in multiple areas of software culture: design, workflow, education, identity, political economy, and governance. It seeks to illustrate that Unity's perceived 'democratisation' extends not only to its accessibility, or even to the company's decision to provide 'free' licenses for its core software (through a platform-based business model), but also in its efforts to cultivate an 'affective community'—one that builds on long-standing 'communitarian practices' (Guevara-Vallalobos, 2011) in 'indie'

Suggested Citation (APA): Nicoll, Benjamin. (2019, October 2-5). *Videogame engines and the politics of 'democratised' software development*. Paper presented at AoIR 2019: The 20th Annual Conference of the Association of Internet Researchers. Brisbane, Australia: AoIR. Retrieved from http://spir.aoir.org.

development and ‘modding’ scenes—through extensive investment in education, events, and support. Unity’s platform ecology can be described as what Angela McRobbie (2016) calls a ‘creativity *dispositif*’—an affective space where developers are granted a degree of social security to explore possibilities for self-entrepreneurship in what would otherwise be a career path fraught with risk and uncertainty. To make these claims, this paper draws on data from 24 semi-structured interviews with Australian videogame developers, students, and educators, as well as participant observation and ethnographic fieldwork conducted at ‘The Arcade’, a co-working space for (primarily) videogame studios located in Melbourne’s Southbank. It builds on an existing body of research on game engines and related development tools, including research informed by ideas and methods from Actor Network Theory and Object-Oriented Ontology, wherein game engines are understood as ‘actors’ situated in studio environments (see, for example, Whitson, 2018; O'Donnell, 2014; Banks, 2013).

My interview data indicate that Unity is more than just an ‘actor’ in the studio space; it is also a cultural entity whose articulations in broader contexts of design, workflow, education, identity, and governance now need to be considered. These contexts form a ‘circuit of cultural software’, which informs the structure of this paper. My approach here has obvious resonances with the traditional ‘circuit of culture’ method (du Gay *et al.*, 1997), in which a cultural object (the prototypical example being the Sony Walkman) is passed through five interlocking registers—representation, identity, production, consumption, and regulation—and researched accordingly. However, the circuit of culture cannot be applied wholesale to today’s software-based culture, of which game engines are an increasingly important component. As Lev Manovich (2013: 33) argues, design tools such as Photoshop, Blender, and Maya ‘play a central role in shaping both the material elements and many of the immaterial structures that together make up ‘culture’’, and so configure the very circuitry that connect capital, labour, and creativity in today’s economy (see also Qiu, Gregg, and Crawford, 2014).

In the circuit of cultural software, the ‘discourse of democratisation’ is treated as a governing logic within which the interconnected circuits of design, workflow, identity, political economy, and education are situated. Democratisation is a powerful, mobilising, and oftentimes-sensitive concept in videogame development (see Harvey, 2014). Whenever democratisation came up in the interviews—and it often did, unprompted—its status as an ‘empty signifier’ (Laclau, 2007 [1996]) became clear. Respondents linked democratisation to accessibility; versatility; empowerment; diversification of content and culture; recognition of the labour of artists and designers as opposed to just programmers; self-publishing; interfaces that are ‘creative-friendly’ and ‘visual’; the open sharing of knowledge; freedom to create and sell products on Unity’s ‘asset store’ (which functions much like an app store); and care and community. Democratisation was also articulated to contradictory feelings of ‘a lot more content and a lot more competition’; homogenized design practices; threats of monopolisation; Silicon Valley ‘blue-sky’ ideology; and the ongoing presence of ‘computer science baggage’ in ostensibly creative-friendly design tools. One respondent, who was developing a custom engine on a part-time basis, even went so far as to describe Unity’s bid to democratise development as ‘absurd’, on the grounds that ‘democracy would imply that the wider world would decide who’s running Unity, but they don’t’.

Has Unity democratised development, as is often claimed? This is a complex question that requires an equally complex answer—not least because the very notion of 'democracy' is often intertwined with utopian ideals of 'openness' espoused by platforms such as Wikipedia (cf. Tckaz, 2014). However, as several of my respondents made clear, democracy can be a sensitive topic in a cultural context where, even in ostensibly democratic societies, an underlying 'hatred of democracy' (Rancière, 2006) seems to have taken root. My sense is that people feel empowered by Unity not only because of the tools it provides, but also because it makes a sustained commitment to democracy and equality in a political environment where such a commitment is felt to be lacking. There are multiple theories explaining Unity's rapid ascension to a quasi-monopolistic status, but one, perhaps under-acknowledged explanation—and one that will be explored in this paper—is that the company has leveraged the symbolic power of democratisation at a time when developers face increased vulnerability and precariousness because of, for example, the erosion of state welfare and policy support.

References

Banks J (2013) *Co-creating Videogames*. London: Bloomsbury.

Darchen S (2016) "Clusters" or 'communities'? Analysing the spatial agglomeration of video game companies in Australia. *Urban Geography*, *37*(2), 202–222.

du Gay P, Hall S, Janes L, Mackay H, and Negus K (1997) *Doing Cultural Studies: The Story of the Sony Walkman*. London: Thousand Oaks.

Guevara-Villalobos. 2011. Cultures of independent game production: Examining the relationship between community and labour. *Proceedings of DiGRA 2011 Conference: Think Design Play*: 1-18. Retrieved from: http://www.digra.org/wp-content/uploads/digital-library/11307.08157.pdf

Laclau E (2007 [1996]) *Emancipation(s)*. London: Verso.

Manovich L (2013) *Software Takes Command*. Cambridge, MA: MIT Press.

McRobbie A (2016) *Be Creative: Making a Living in the New Culture Industries*. Malden, MA: Polity.

O'Donnell C (2014) *Developers' Dilemma: The Secret World of Videogame Creators*. Cambridge, MA: MIT Press.

Qiu J.L, Gregg M, and Crawford K (2014) Circuits of Labour: A Labour Theory of the iPhone Era. *tripleC* 12(2): 564-581.

Rancière J (2006) *Hatred of Democracy*. Corcoran S (trans.). London: Verso.

Tckaz N (2014) *Wikipedia and the Politics of Openness*. Minneapolis: University of Minnesota Press.

Whitson J (2018) Voodoo software and boundary objects in game development: How developers collaborate and conflict with game engines and art tools. *New Media & Society* 20(7): 2315-2332.