EUROPEAN JOURNAL OF cultural studies

Vol 11(2) 177–195; 1367-5494
DOI: 10.1177/1367549407088331
www.sagepublications.com

The mod industries?

The industrial logic of non-market game production

David B. Nieborg
University of Amsterdam
Shenja van der Graaf
Berkman Center for Internet and Society, Harvard Law School

ABSTRACT This article seeks to make the relationship between non-market game developers (modders) and the game developer company explicit through game technology. It investigates a particular type of modding, i.e. total conversion mod teams, whose organization can be said to conform to the high-risk, technologically-advanced, capital-intensive, proprietary practice of the developer company. The notion 'proprietary experience' is applied to indicate an industrial logic underlying many mod projects. In addition to a particular user-driven mode of cultural production, mods as proprietary extensions build upon proprietary technology and are not simple redesigned games, because modders tend to follow a particular marketing and industrial discourse with corresponding industrial-like practices.

KEYWORDS first-person shooter, game engine, proprietary experience, proprietary extension, total conversion modification

Introduction

In 1999, two young enthusiastic amateur developers, Minh 'Gooseman' Le and Jess 'Cliffe' Cliffe, and their team brought us the modification ('mod') *Counter-Strike*, based on the popular first-person shooter game *Half-Life* (Valve, 1998). *Half-Life*, developed by independent software and technology developer Valve, is a skilfully designed, narrative-driven single-player game in which the player guides the scientist Gordon Freeman through the Black Mesa Research Facility to find his way to safety after an erroneous teleportation experiment. Purchasing *Half-Life* granted amateur developers access to the game's core technology, its proprietary software engine as well as the free tools provided by the original game's developers. Tapping into online knowledge communities for inspiration, aid and support, the group modified large parts of the original *Half-Life*

and complemented it with an alternative game mode, new code, maps and sounds. The single-player science fiction set-up of *Half-Life* was transformed into a multiplayer game, featuring contemporary themed, fast-paced, multiplayer action. The newly-envisioned mod was called *Counter-Strike* after its counter terrorism-themed gameplay. Although the mod was a complete overhaul, or in gamers' parlance a 'total conversion', of the original game, the new game still focused on 'running and gunning' and the underlying engine technology remained intact.

Counter-Strike proved to be an instant hit among those who owned *Half-Life* and attracted a significant following. Gamers pressed the modders to come up with frequent updates and additional content such as more maps (player levels) and avatar skins. The success of *Counter-Strike* eventually surpassed that of *Half-Life*, and gamers started to buy the original game just to play the mod. Foreseeing a great future for *Counter-Strike* and its developers, *Half-Life*'s original developer Valve responded by offering the mod team a spot on Valve's professional development team and in so doing, acquiring *Counter-Strike*'s valuable intellectual property. In 2000 Valve published *Counter-Strike* as a commercial game while the *Counter-Strike* mod continues to be freely available to those who purchase *Half-Life*. Today, with more than 7 billion player minutes a month, *Counter-Strike* is still the most played online multiplayer first-person shooter game on personal computers (PCs) and its player count is equal to the number of players of all other online first-person shooter games combined.[1]

Unsurprisingly, many who seek to address mod culture hail *Counter-Strike*'s success. In cultural and economic terms, and with its unrelenting and prominent success, the *Counter-Strike* story gave mod practices a rather unequivocal position in first-person shooter game culture, which warrants the question: what is *Counter-Strike*? For millions of gamers worldwide it is a complex social world with its own rituals and social organizations, such as teams or 'clans' (Wright et al., 2002). For developers such as Valve, it is part of a successful franchise that generates significant revenue even eight years after its inception, proving to be an incredibly worthwhile business strategy. In addition, it is opening up (parts of) the game technology to users and providing tools and information to develop new content. For many modders it seems to have become *the* example to which to aspire. These are attractive reasons for researchers to dedicate their attention to *Counter-Strike* (Dovey and Kennedy, 2006; Jenkins, 2006; Kerr, 2006; Sotamaa, 2005a) because it has become the *sine qua non* of modding.

Since the mid-1990s, digital technologies have opened up possibilities for decentralization and diversification by enabling consumers to become participants in the production and distribution of media content, rather than being the endpoint of delivery, while companies have aimed to use and leverage some of these unique qualities of networked technologies by linking consumers directly into the production and distribution of media

content for product development and loyalty building. A rapidly evolving (yet often subtle) relationship of collaboration and cooperation across firm boundaries can be witnessed, where game developers such as Valve pride themselves in actively articulating the creative endeavours of modders. Such novel migratory practices have an economic and cultural impact by foregrounding the increasing pace of innovation, yielding insight into the trajectories of innovation and commercialization that are a regular occurrence among software developers, modders and gamers.

Games in general and computer games in particular are often platforms for user creativity, and nowadays modding has become an integral part of modern-day game culture.[2] Contemporary game technology – particularly first-person shooter PC game technology – has been designed purposely to assist both professionals and amateur developers in unlocking the capabilities of the game's core technology, the game engine. Id Software's engine programmer John Carmack, responsible for legendary first-person shooter games such as the *Doom*, *Quake* and *Wolfenstein* series, sees his task as developing accessible and enabling game technology that serves as 'the ultimate canvas' (Gladstone, 2007). In this study, the key organizing role of the game engine is addressed in innovation practices where the engine is like a canvas for modders and for-profit game developers, and in its political economic role where the engine is a highly-advanced, proprietary piece of technology.

Game developers take different stands with regard to user-created material, varying from not allowing any such content to trying actively to encourage practices such as modding, skinning and modelling by providing the user with the necessary toolkits. For example, take the design of Massively Multiplayer Online Role-Playing Games (MMORPGs): 'Rather than a linear, top-down process, ultimately what we find is a complex co-construction of technologies that occurs between designers, users, and the artifacts themselves' (Taylor, 2006: 2). In her work on the MMORPG *EverQuest*, Taylor signals this uneasy relationship between the productive engagement of players and design and commercial interests, arguing that because of their labour activities, they provide core value to the world (i.e. beyond beta-testing; cf. van der Graaf and Cobarr, forthcoming). In this journal issue, Humphreys suggests that we should no longer understand players as 'end-users', but should frame the activity of playing as immaterial labour, and in doing so we can overcome the unproductive production/consumption distinction.

The popular *World of Warcraft* (Blizzard Entertainment, 2001), for example, allows players to design their own interface modifications. Especially for heavy users, interface mods have become indispensable gameplay aids. However, in the heavily-regulated realms of *World of Warcraft*, interface modding is a deliberate exception. Tinkering with non-interface-related game files is strictly prohibited and 'illegally' altering game files leads to capital punishment (i.e. suspending a player's account).

From this perspective, first-person shooter modding can be seen as the most innovative and sophisticated instance of modding for PC games by providing tools and content, deliberately opening up significant parts of game technology, purposely disseminating knowledge and information and putting up a semi-restrictive legal framework to regulate modding. Conversely, compared to Linden Lab's virtual world *Second Life*, modding for PC games is legally and economically rather limited. Linden Lab built the *Second Life* grid as an open, extensible platform for development rather than a closed proprietary system (yet it still is; van der Graaf, 2007). Large parts of *Second Life*'s technology are open source, 'residents' retain their intellectual property, and while users are still bound by the implicit and explicit technological boundaries of the *Second Life* platform, the world, themes and avatars of *Second Life* are much more diverse than those in many first-person shooter mods.

In this article, it is not the general practice of modding that is investigated; rather it seeks to yield insight into mods as the interstices between Valve, its proprietary game technology and modders.[3] The way that Valve is organized and develops its own positions in conjunction with modders, and the way that modders are organized and develop their own positions by using Valve's proprietary technology, suggest an (industrial) model that can be characterized by granularity and modularity. Valve's labour practices echo modders' development practices, which suggests that if we are interested in this overlapping interest, we should approach developers and modders as components and independents in the process of game development. Valve was chosen as it is one of the most renowned and successful independent game developers worldwide, with titles including *Half-Life*, *Half-Life 2*, *Counter-Strike*, *Day of Defeat* (2003) and *Team Fortress Classic* (1999). Their interest in and support of modding practices are applauded – the former mod teams (or members thereof) of *Counter-Strike*, *Day of Defeat* and *Team Fortress* all work at Valve, and Valve's software development kit is open and available to anyone who purchases Valve's games and is supported by the development community consisting of Valve developers, third-party licensees and amateur developers: modders. Furthermore, Steam is Valve's digital distribution and communications platform, which is used to digitally distribute and manage more than 180 PC games such as first-person shooter, role-playing games and other genres, and which has approximately 13 million active users. Clients include game publishers such as Take-Two Interactive, Eidos Interactive, THQ and id Software.

For this study, semi-structured interviews with 13 Valve employees were conducted at Valve's headquarters in Bellevue, WA. Due to a non-disclosure agreement, notes were taken rather than that interviews were recorded. Hence, within the confines of this article, interviews are used as illustrative background data and not as primary sources. Questions

focused on the role of each interviewee in the game development process, how performance is measured, how teams work, how involved they are in games (fan) communities, their level of interaction with customers and the way that Valve is organized.[4] In addition, various mod projects were examined in order to highlight Valve's approach to steer and facilitate gamers' involvement in the game development process, modding practices and the implications for thinking about the product character of mods as proprietary experiences and extensions.

The next section introduces the technological particulars of modding by focusing on the underlying software code: the game engine. This will be followed by critically approaching the practices of modders, invoking thoughts concerning free labour. These two technological and economic streams result in a conceptualization of total conversion mods as proprietary extensions. The article then takes a closer look at some of Valve's games and mods from a more cultural perspective by analysing non-market game developers (modders), suggesting a consistent relationship between market and non-market game production.

The half-life of a game engine

Rather than looking at mods and mod teams as such, it is suggested that the relationship between modders and the company can be made explicit through game technology. Studies on mod culture tend to focus on *Counter-Strike*'s cultural or economic impact while often the game's underlying technology is taken for granted. *Counter-Strike* is a particular instance of modding (or so we investigate). While thousands, if not hundreds of thousands, of users experiment with first-person shooter game technology and develop their own gameplay mods, such as slightly altered maps or skins, additions such as server tools or single-player missions, *Counter-Strike* is a total conversion modification. Developing and distributing total conversion mods is the most complex and advanced kind of modding, combining various skills that transgress the boundaries of mere tinkering and require, among others, advanced managerial and marketing skills. Sometimes the ambitions of total conversion mod teams are such that there are positions within a mod team such as public relations manager, lead tester, (historical) adviser and community manager. An explanation of total conversion modding is provided on the Valve development wiki, echoing this notion that mod projects are elaborate productions:

> Total Conversion: a mod that either changes or completely rewrites the game mechanics, resulting in a game far different from what it originally was. They typically use new models, new animations, and new code, among other things. They tend to be few and far between because of the work involved, and often fall apart due to internal problems or loss of interest. (The other mod types often suffer this same fate as well.) (http://developer.valvesoftware.com/wiki/Mod)[5]

Our interest in the specifics of total conversion mods stems from a mild dissatisfaction with previous research, which seems to overlook the particular status of these elaborate productions – in particular, their striking resemblance to the organization of the game industry's production and marketing logic. The research here reports on total conversion mod teams that seem to conform to the corporate logic of the developer company, which concludes that total conversion mods are a particular manifestation of mod culture and represent a specific mode of modding; one that is of a high-risk, technologically-advanced, capital-intensive, proprietary practice. However, first this article will deconstruct previous notions of what a mod is exactly, thereby focusing on the particular technological and political economic status of mods, and what is referred to here as proprietary extensions. In order to do so, the crucial function of game engine technology in combination with development tools, or software development kits (SDKs), needs further explanation.

First-person shooter mod culture typifies a specific cultural economy where producers set barriers directly, for example, through End User License Agreements (EULAs) and indirectly through tools, while at the same time encouraging user-developed material (Manovich, 2001). One of the most elemental pieces of game technology facilitating modding is the game engine. The engine is the core piece of game software, consisting of several components such as the renderer that visualizes the game space, a physics engine, networking code, artificial intelligence code, a sound system and other parts.[6] For modders, a first-person shooter game engine is a highly important technological tool allowing mod development, offering users a highly advanced piece of software that serves as a canvas for their ideas. In the words of Valve's managing director, Gabe Newell:

> One of the unique characteristics of games as a medium is that you have to create it in cooperation with the audiences … A game engine is not just the platform for the game itself, but a platform for all the mods that come along to extend the life and enjoyment of the experience. (quoted in Hodgson, 2004: 5)

Modders' application of game engines and their wider function within mod development is seldom acknowledged by scholars discussing mods. Bogost, one of a handful of scholars, emphasizes the important role of game engines as component-based software systems within the wider game industry and singles out first-person shooter game engines as they 'construe entire gameplay behaviours, facilitating functional interactions divorced from individual games' (2006: 57). Bogost goes on to explain how engines form the basis for other games and, as such, share the material, functional and, in the case of the first-person shooter games discussed in this article, intellectual proprietary attributes of the core engine: 'These confines both facilitate and limit discursive production, just as the rules of natural languages bound poetry and the rules of optics bound photography'

(2006: 66). In the particular case of Valve, the Source engine is the main technology that underlies multiple games in various franchises such as the *Half-Life 2* series, remakes such as *Day of Defeat: Source* and *Counter-Strike: Source*, sequels such as *Team Fortress 2* and new intellectual property such as *Portal*.

The game engine offers modders a proprietary development platform which structures and constrains mod development: 'Up to this point in time the engine has been a technologically determining agent in the character of computer games; game engines are not infinitely adaptable or "content neutral"' (Dovey and Kennedy, 2006: 57). Today's first-person shooter engines are brought about by multi-million dollar investments by game developers and are considered to be proprietary technology: 'Like component software, game engines are IP [intellectual property]' (Bogost, 2006: 56). Development powerhouses such as Valve (Source engine), id Software (id Tech), and Epic Games (Unreal engine) purposely specialize in high-risk, capital-intensive game engine development and licensing. First-person shooter engine development is iterative, and engines have an ongoing development cycle where there is constant tweaking and updating. Just as games have sequels, so do engines. The latest Unreal engine is Unreal Engine 3.0, and id Software's experimental technology is called id Tech 5. In this respect, Dovey and Kennedy (2006) speak of game technology as being part of an 'upgrade culture'. The perpetual innovation of gaming hardware, such as console manufacturing, and engine (software) development, has a lot in common with a never-ending arms race, except for the hostile connotation.

The ever-increasing costs of game development – largely due to ongoing investments in game technology – turn engine development into a lucrative business model by licensing the engine to 'third parties': other game developers. Software engines save both engine licensees and modders time by providing a stable development platform and, in turn, 'should allow developers to focus on innovation instead of mechanics' (Bogost, 2006: 60). Engines are purposefully modular in design so that they can enable the upgrading of particular engine parts without 'breaking the code', or third parties to develop plug-ins ('sub-engines') to offload complex software routines. Such sub-parts are also known as 'libraries' and are constantly worked on by developers. Well-known third-party engine libraries are the Havok physics engine, which is used for in-game collision detection and vehicle dynamics, and Interactive Data Visualization's SpeedTree package, which was designed to render in-game foliage. It is partly this modularity that distinguishes modders from licensees and first-party (engine) developers. In many cases modders do not have physical or legal access to certain parts of a first-person shooter engine, such as Valve's Source engine. Valve programmer Tom Leonard notes that:

> Licensees have access to all of our code except for certain physics and sound libraries which we license from other parties. In that case, our licensees can

> either accept binaries from us, or can arrange their own license with the makers of those libraries. (Personal communication, 10 August 2007)

In opposition to licensees, modders only have access to specific parts of the engine, but not to the source code for the renderer, networking, physics, sound system and other core technology pieces of the Valve engine.

To unlock the engine's possibilities a set of tools is required that are inherently part of the engine. Generally, for game development there are two sets of tools. Many third-party toolsets such as graphics editors and 3D modelling software have their own plug-ins to interface with popular first-person shooter engines. Modding is bound to a large extent by the engine's internal logic, which is operationalized through its proprietary toolkit. The necessary tools to make mods of original games are provided by the game engine developer.[7] Thus the domain of the game developer consists of the platform, engine and source code, while players have access to game code and often an editor and/or toolkit allowing players to customize and design essential parts of the gameworld. For example, Valve's Source SDK consists of several proprietary tools such as the Hammer level editor, FacePoser (an advanced choreography tool), the *Half-Life* Model Viewer, and Softimage XSI EXP, a third-party 3D graphics application.[8] The Source SDK is distributed through the Steam platform. There are also quite a number of designers active in open-source projects – both paid and unpaid – to design games and virtual worlds, such as WorldForge.

Mods as proprietary extensions

In order to gain a deeper understanding of both the complex relation between users as modders and developers as facilitators, this article focuses intentionally on developing total conversions as a particular high-profile mode of modding. During the early stages of mod culture, i.e. the days of *Doom* (1993), modding literally meant modifying existing content, such as sprites (two-dimensional images in a three-dimensional space) and textures. A general and unspecified notion of 'mods' as a moniker for all user-created game material misses the finer nuances of the wide range of creative output of amateur developers. Even when concentrating on first-person shooter mods there are many variations, such as client-side mods as user-created maps or skins, or server-side mods such as server plug-ins, which gather player statistics. For user-developed maps, often the existing game material such as textures are used, making the mods more iterative than alterations. Total conversion modders replace the original game's content layer completely with original user-created material. Whatever the scale and scope of a mod, it will always function within the original game's proprietary structure.

For gamers there may not be a significant difference between a well-designed total conversion mod, such as an early version of *Counter-Strike*, or

a commercial game, such as *Counter-Strike: Source*. Even though modders and engine developers have a shared set of development practices and tools, they do not operate on an equal level: not on an economic level, as it is strictly forbidden to sell a mod or to make it work without interacting with the proprietary engine. Modding can be seen as

> a special case where the commercial producer continues to exert constraints on use even as the work gets appropriated by the grassroots community. I can change the fundamental code of the game if I mod it, but at the same time, nobody can play my transformed version of the game unless they become a consumer of the original work. (Jenkins, 2006: 163)

Neither, as we have explained, on a technical level, as parts of the engine and tools are closed off.

Scholarly work on mods tends to generalize modding as simply modifying existing game files, broadly defining mods as user-developed modifications or 'gamer made alterations to commercial technology' (Sotamaa, quoted in Kerr, 2006: 119). As such, these notions overlook the important technological and political economic dimensions of modding: modding is not only a PC-centric affair and as the *World of Warcraft* interface mods indicate, it differs considerably in scope and scale among different PC genres. There are judicial and economic particularities that are part of gaining access to the tools of cultural production. Although total conversion mods add a completely new layer of content, for commercial games this additional material is required to interact with the game's proprietary engine, via elaborate EULAs. Therefore, total conversion modifications can be said to be more of an addition to proprietary standalone software engines. In practice there is only a small difference between an engine licensee and a modder. Yet, following Benkler's (2006) topology on the information economy, users (as opposed to licensees) are forbidden to derive direct monetary value from their creations – therefore mods are, by definition, non-market productions. Due to mods' dependency on proprietary code, they are non-commons-based. As such, mods can be understood better as non-market proprietary extensions.

ComMODifying proprietary innovation

The emergence of mod culture coincided with a trend towards the democratization of innovation: 'User-centered innovation is steadily increasing in importance as computing and communication technologies improve' (von Hippel, 2005: 121). As stated previously, it is suggested that 'production' continues well after the release of a game through user contributions and modding practices (cf. 'play to play' and 'play to develop', Jeppesen and Molin, 2003). Modders seemingly spend infinite hours of unpaid labour on uncertain projects, which makes their work prone to industry appropriation. However, modders – especially total conversion modders – eagerly

anticipate this subsequent process of commodification to a point where total conversion mods serve as part of future game developers' portfolios (De Peuter and Dyer-Witheford, 2005; Nieborg, 2005).

Because of such strategies and the unique status of mod culture versus the game industry, first-person shooter mod culture has been conceptualized as a type of precarious labour, or 'playbour', signalling the uncertain status of work and leisure, copyright issues and the 'ideological masking of modding as a collaborative process' (Kücklich, 2005: 6). Due to the 'playful' nature of modding, discursively constructed by game developers as an extension of playing with the original game, the value-adding practices of total conversion modders are understood as 'free labour', a liminal form of work inbetween paid and voluntary labour that is specific to the digital economy (Terranova, 2000). There is no question that mods do add considerable value for first-person shooter developers. Postigo (2003: 594) frames modders from a purely economic perspective as 'video-game programmer hobbyists' operating within the post-industrial economy and generating considerable monetary value through their unwaged work. However, in practice this value seems to be far more diverse and even more intangible than suggested.

The fundamental challenge for any organization is figuring out how to maximize, motivate, incorporate and allocate and coordinate particular tasks among employees and the gamer base by employing the game engine – and as such guiding and motivating modders. Many models have been constructed to explain how companies can make use of external innovation sources, including highlighting the use of alliances (Gerlach, 1992), imitation of competition (Lieberman and Montgomery, 1998), network structures in open-source movements (von Hippel, 2005), exploitation of outlaw innovators or hackers (Flowers, 2006), user interfaces (Schneidermann, 2002) and spillovers (Breschi and Lissoni, 2001). However, consulting with users is being focused upon by companies more than ever before (Jenkins, 2006). This coincides with a considerable scholarly interest in innovation resulting from rapidly-expanding user activities that are facilitated by enhanced connectivity among users.

Comparing the traditional to the user-driven approach, Thomke and von Hippel (2002) indicate that in the traditional model, the developer company takes on most of the product development process, which results in an unfavourable situation – in terms of costs and time – for iterations between firm and user. In the customer-as-innovator model, users become part of the stages of idea generation and development, facilitated by suppliers' toolkits. In this way, users are presented with a broader design space, shifting the locus of the supplier–user interface, while contributing to the design process in companies (Humphreys et al., 2005). As a result, what the product is, and what it can become, comes to light despite the missing skills or incomplete knowledge base of the suppliers (Ondrejka, 2005).

The company needs insight into results of feature changes in order to make trade-offs during the development process.

It was by having modders interact with innovative technology that game developers

> discovered that the net effect of user-developed mods was positive for them: mods actually increased the sales of their basic software, because users had to buy the vendors' proprietary software engine code in order to play the mods. (von Hippel, 2005: 129)

This is one of the most common economic arguments about mods, that they extend the interest of gamers into the original game's franchise. This applies to publishers of PC games in particular, as they derive direct value from the sales of every game sold. As such, a mod can add to the shelf-life of the original game, build a new brand and subsequently serve as a retention tool. External innovations are attracted in three main ways:

- minimizing technical obstacles so that modders 'can build upon publisher's proprietary innovation to make a compelling game experience'. External innovations are attracted in three main ways (West and Gallagher, 2006: 98);
- creating an infrastructure that facilitates and encourages participation and collaboration; and
- peer recognition for generated contributions.

By providing gamers with toolkits for modding activities (termed 'user innovation' and 'user co-design'), the practice of systematically outsourcing certain design and innovation tasks from the locus of the game developer to the user enables modders to create a mod that corresponds to their individual interests and needs. This approach of user-driven innovation relates to cooperation and game engine technology. On the one hand, it points us to cooperation among different types of gamers, as 'there are disparities between them in terms of their readiness, interest and capabilities' (Jeppesen and Molin, 2003: 20) and, on the other, to cooperation among modders and game developers. In-company development activities are understood to exist and work alongside modding practices bounded by the proprietary game technology, and therefore modders have become intimate with the inner workings of an advanced engine such as Valve's Source engine. However, wide-scale, real-time user contributions to the development of such proprietary engines are rare.

When critically approaching user-innovation studies, it is suggested that user-driven innovation through toolkits is structured by the engine and therefore takes place within the set, capital-intensive boundaries of the proprietary technology. Arguably, creating a proprietary experience which can be characterized by a game developer seeking to minimize technical obstacles (the engine and toolkits), and to create a consistent infrastructure (Steam and the Valve Developer Community), means that the 'user-driven' practice of modding should not be taken for granted as

much as it has been.[9] In the following section a more detailed analysis of the techno-economic status of total conversion modding is offered. Again, it draws on Valve's games and mods, such as the development of *Counter-Strike*, which inhabit an industry-led production context.

Valve

As with many developers, Valve uses both internal and external sources to generate ideas and paths to the games marketplace. Whether it is the creative output of fans in the case of mods, the licensing of advanced proprietary engine technology or internet service providers hosting Valve games, game developers and publishers – and Valve is no exception – are eager to harness external sources of direct and indirect value. As we have seen, games in general and PC games in particular are, in many cases, platforms for user creativity, making modding an integral part of game culture. Looking at Valve's position as independent game developer can yield insight into emerging and implemented trajectories of participation and commercialization.

In recent years there has been a focus on companies rethinking their sources of innovation, after realizing that their products can be modified. They may encourage user innovation by providing an 'open system' and freely available equipment such as toolkits that are applied generally for problem-solving (von Hippel, 2005). Little attention has been paid to the implications of user innovations when they are a subset of the economic system through sharing and/or commercialization, as in total conversion modding. Valve's management constructs strategies to shape the roles of employees and gamers in order to capitalize on the integration of the internal and external labour process, within the organizational dynamics of the games industry.

Valve as a company is trying to be (literally) as open as possible to modders:

> We always have had a good relationship with the mod community. We constantly invite people over to Seattle to have an exchange of ideas and thoughts and to help them with their projects. (personal interview with Gabe Newell, 22 August 2007)

Over time Valve has stressed, through the words of Newell, its dedication to supporting modders:

> Our engine and technology is really open. You are able to access everything. If we have to choose between secrecy and keeping stuff proprietary, we always try to be as supportive to the mod community as possible. There is nothing a modder cannot do. (Personal interview with Gabe Newell, 22 August 2007)

Valve has a clear strategy to steer gamers' involvement in the game development process, indirectly through the appropriation of successful

total conversion mods, and directly through its engine and toolset. The reason, then, why Valve has been able to attract mod teams continuously is by framing modding as an extension of play, and 'the fact that the industry has been careful to project an image of itself that highlights its dedication to high-quality games and deemphasises its dedication to profit' (Kücklich, 2005).

The mod industries

The great majority of modders are not total conversion mod teams; rather they are fans eager to deconstruct and understand Valve's design philosophy and its innovative technology:

> An important reason for modding is to show what you as a modder can do and to learn about our design, tools and technology. The main groups of modders are just guys who are enthusiastic about our technology, such as with *Counter-Strike*. (Personal interview with Gabe Newell, 22 August 2007)

However, the example of *Counter-Strike* as a truly bottom-up manifestation of user-driven development, similar to the position pervasive in academic literature, is somewhat misleading. In the case of the development of total conversion mods such as *Counter-Strike*, the development teams are seldom 'just guys'. Dovey and Kennedy give a detailed explanation of the reason why *Counter-Strike* did not arrive out of the blue:

> It would be a mistake to romanticize Minh Le and his team as a bunch of lucky/talented amateurs; here the distinction between amateur and professional (developer) begins to break down; the *Counterstrike* [sic] production project was a highly skilled, focused collective effort that is typical of the shareware culture that underpinned the development of computer software in its early days and of the internet throughout its history. Nor were the *Counterstrike* team unknowns – the game first saw the light at the Half Life Mod Expo, an event funded by Valve to showcase the best and most interesting modifications of their game being made by these 'player creators'. (2006: 125)

The lessons learned from incorporating the *Counter-Strike* team became a *modus operandi* and an important way to incorporate intellectual property, which Valve also did with other first-person shooter total conversion mods such as *Team Fortress Classic*, *Day of Defeat* and *Portal*, thereby uniquely positioning itself vis-à-vis its games. The deployment of expos or mod contests, such as the million-dollar 'Make Something Unreal' contest, are two of many industry strategies to seek out more mature mod projects (Nieborg, 2005; Sotamaa, 2005b). The overall winners of the Unreal contest, the Second World War mod team of *Red Orchestra: Combined Arms*, received $50,000 in prize money and an Unreal Engine commercial licence. Later, *Red Orchestra: Ostfront 41–45* was published as a retail game through Steam. The developers of the contemporary warfare-themed *Desert Combat* total conversion mod for the Second

World War first-person shooter *Battlefield 1942* followed a similar pattern (cf. Nieborg, 2005). The *Desert Combat* mod attracted such a substantial following, topping that of the original game, that the mod's intellectual property and team were acquired by *Battlefield 1942*'s developer.

However, many total conversion mods never materialize and the road from a small mod with some homebrew maps and skins to a full-blown total conversion mod is bumpy. Yet the high-profile cases of *Counter-Strike*, *Red Orchestra* and *Desert Combat* seemed to have raised the bar for total conversion mod teams. Alex 'ACPaco' Capriole (2006), a staff writer for a *Half-Life* community website, summarizes this view as the 'Wannabe CS [*Counter-Strike*] Syndrome':

> Some mod teams always seem to think that the day they release they're going to be the biggest hit since *Day of Defeat* or *Counter-Strike*, disregarding the fact that those mods were lucky outliers, not the norm. If you look at their website, they treat their mod as if it were a major upcoming commercial release, just waiting to be bought out by Valve. (Capriole, 2006)

This view resonates with Valve developer Robin Walker, who started out as a modder:

> The biggest thing that mod makers are all getting wrong now, unfortunately, is they're focusing on becoming more and more like commercial products. Which basically means they're trying to develop more and more assets. (Personal interview with Robin Walker, 23 August 2006)

Apart from a free set of user-friendly tools, community support and a versatile game engine, why would a modder choose the Source engine over one of the many other first-person shooter engines which have similar traits and treats? The Valve Developer Community wiki reads:

> Valve remains dedicated to supporting the mod community. Valve also has a proven track record for turning mods into full-fledged commercial products. (http://developer.valvesoftware.com/wiki/Building_MODs_with_Valve_Technology)

The all-important currentness of engine technology is crucial both to modders and Valve, and the longevity of both amateur and for-profit productions are counted in player minutes as well as technological iterations. During the proprietary experience that is modding, users wilfully subject themselves to the maelstrom of the technology-driven state of perpetual innovation.

As proprietary extensions, total conversion mods feature certain unique techno-economic characteristics on the level of development, marketing and usage, structuring the expectations of gamers, modders and developers and prolonging a shared set of discourse, development practices and aesthetic conventions. An example of such a prototypical proprietary extension, following the industrial logic of game developers such as Valve, is the Source engine-powered total conversion mod *Insurgency: Modern*

Infantry Combat. The leader of the mod team explains why the Source engine was a deliberate choice:

> We looked at *Half-Life* and saw how the *Half-Life* mods were still alive even after being around for so many years. This is what we wanted, to create a product that can be enjoyed over many years rather than create something for a short-lived game platform. Like we predicted, even as Source is getting outdated, the community is still alive and the engine keeps getting updated. (Yourcatch22, 2007)

The development on the mod began in 2002 and at the time of writing it is still in beta (work in progress). Not knowing that *Insurgency* is a mod, one would think it is a for-profit production. For instance, consider this 'vacancy' on the Insurgency homepage:

> Public Relations – We need a new PR guy to work with me in helping deal with the public. This person should have a very in depth understanding of how the gaming and mod industries work, and have a very strong drive to do anything they can to make the mod prosper. Maturity, experience, intelligence and time are very important. (http://www.insmod.net)

Players who mod act as voluntary providers of complements, freely revealed to the entire community (and possibly beyond). Total conversion mods can be seen as a particular subset of modding and are given to the first-person shooter user base by a relatively small group of generally highly-motivated individuals working together in dedicated teams. Insurgency is an all-male team of 12 nationalities (cf. 'virtual studios'; De Peuter and Dyer-Witheford, 2005), mostly from North America and northern Europe, split into three sub-teams. Modders themselves discursively frame their actions as a dedicated men's job which means serious, plain hard work obscuring the final distinction between grassroots cultural production and the cultural industry.

Conclusion

Modding in general is often invoked by humanities scholars discussing mods and modders for their peculiar cultural status vis-à-vis the game industry. The first-person shooter PC game *Counter-Strike* is singled out as one of the few mods that transgressed its mod status to become a full-blown retail title. Others stress the unique stance of game developers toward their customers, pointing toward the intimate relationship between fans and developers and the co-created nature of the development of a game and its many community services, purposely blurring the line between production and consumption, and between an original game and its modifications. Within this setting, total conversion modifications feature certain unique techno-economic characteristics on the level of development and usage, structuring user expectations and maintaining certain aesthetic conventions. We believe that discussions from both critical

and administrative standpoints within companies reveal a dialectic tension between their emergent and designed properties, where organizational design and practice are viewed as two complementary sources of structuring: 'the organization is therefore the meeting of two sources of structure: the designed structure of the institution and the emergent structure of practice' (Wenger, 1998: 244). This tension between the emergent and the structured, between top-down and bottom-up expressions of power, between institutional design and emergent practices, is a useful way to think about modding in game development.

Against this backdrop, the relation between the employment of user toolkits and the need for companies such as Valve to support their customers can best be described by the way in which the game developer sets technical limits as to what the modder can do with the engine, graphics structure and the toolkit or editor. Ideally, the employment of user toolkits and community support, which is fully part of Valve's business model, would result in a wide range of user innovations. As total conversion mod teams are poised to do 'the next *Counter-Strike*', they are not only limited to the techno-economic cadre which characterizes total conversion modding, but they align themselves along the same rigorous development practices as for-profit developers. As part of the game industry, total conversion modding has become an industry itself: 'Gamework takes place within a rather gendered, producer–consumer collaborative and counter-cultural legacy as well as a corporate, commercial and technology-driven contemporary context' (Deuze et al., 2007: 337). Through emulating the first party developers' risk-averse, capital-intensive mode of production and within a proprietary context, total conversion modding has become a 'proprietary experience', as modders anticipate the developers' act of reappropriation and subsequent commodification.

Acknowledgements

We are indebted to Tom Leonard at Valve who kindly opened the doors to Valve HQ and set up interviews throughout the company. We also acknowledge the British Academy for funding this research project and the stimulating conversations we had at FIR-HSG.

Notes

1. See http://archive.gamespy.com/stats/ and http://steampowered.com/v/index.php?area=stats for statistics.
2. Modding is not unique or reserved to first-person shooter games. Engaging with PC games such as *World of Warcraft*, *The Sims* or *Neverwinter Nights* moves beyond consumption and by developing derivative game material for these games, they become hybrid co-productions rather than finished end products.
3. This article can then be located within the emerging interdisciplinary field of 'game studies', drawing not only on cultural studies but also on innovation studies and legal studies. Through power structures embedded in game technology and processes in corporate capitalism we seek to gain a

deeper understanding of the cultural practices of the industrial logic of non-market game production.

4. An online survey among Valve gamers was conducted in February 2007. We asked about gamer activities, the way they interact with Valve's games and its developers, and their contribution to Valve's development process. See: http://personal.lse.ac.uk/vanderga/valve.html (accessed 7 November 2007).
5. The other two categories are single player and multiplayer mods.
6. As explained by Valve developers Yahn Bernier (personal interview, 22 August 2006) and Mike Durand (personal communication, 10 August 2007); cf. Dovey and Kennedy (2006).
7. Note that for the wider game industry this is rather an exception than a rule. A great minority of game developers share their specialized proprietary (first party) development tools with the community.
8. Historically modders developed their own toolsets and game companies benefited greatly from this. While some of the source code of Valve's tools, such as the map compiler and the model viewer, are available, other tools such as the Hammer level editor are not open.
9. See http://developer.valvesoftware.com for the Valve developer wiki.

References

Benkler, Y. (2006) *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. New Haven, CT: Yale University Press.

Bogost, I. (2006) *Unit Operations: An Approach to Videogame Criticism*. Cambridge, MA: MIT Press.

Breschi, S. and F. Lissoni (2001) 'Knowledge Spillovers and Local Innovation Systems: A Critical Survey', *Industrial and Corporate Change* 10(4): 975–1004.

Capriole, A. (2006) 'Mistakes Mod Teams Make'. [Accessed 20 August 2007: http://planethalflife.gamespy.com/View.php?view=Editorials.Detail&id=20]

De Peuter, G. and N. Dyer-Witheford (2005) 'A Playful Multitude? Mobilising and Counter-Mobilising Immaterial Game Labour', *Fibreculture* 3(5). [Accessed 7 November 2007: http://journal.fibreculture.org/issue5/depeuter_dyerwitheford.html]

Deuze, M., C.B. Martin and C. Allen (2007) 'The Professional Identity of Gameworkers', *Convergence*: *International Journal of Research into New Media Technologies* 13(4): 335–53.

Dovey, J. and H.W. Kennedy (2006) *Game Cultures: Computer Games as New Media*. Maidenhead: Open University Press.

Flowers, S. (2006) 'Harnessing the Hackers: The Emergent and Exploitation of Outlaw Innovation', Centre for Research in Innovation Management. [Accessed 10 September 2007: http://www.sussex.ac.uk/spru/documents/flowers-paper.pdf]

Gerlach, M. (1992) *Alliance Capitalism*. Berkeley: University of California Press.

Gladstone, D. (2007) 'id Software's New Game: Rage'. [Accessed 20 August 2007: http://www.1up.com/do/newsStory?cId=3161743]

Hodgson, D. (2004) Half-Life 2*: Raising the Bar*. Roseville, CA: Prima Games.

Humphreys, S., B. Fitzgerald, J. Banks and N. Suzor (2005) 'Fan Based Production for Computer Games: User Led Innovation, the "Drift of Value"

and the Negotiation of Intellectual Property Rights', *Media International Australia* 114: 16–29.
Jenkins, H. (2006) *Convergence Culture: Where Old and New Media Collide*. New York: New York University Press.
Jeppesen, L.B. and M.J. Molin (2003) 'Consumers as Co-developers: Learning and Innovation Outside the Firm', *Technology Analysis and Strategic Management* 15(3): 363–83.
Kerr, A. (2006) *The Business and Culture of Digital Game: Gamework/Gameplay*. London: Sage.
Kücklich, J. (2005) 'Precarious Playbour: Modders and the Digital Games Industry', *Fibreculture* 3(5). [Accessed 7 November 2007]: http://www.journal.fibreculture.org/issue5/kucklich.html]
Lieberman, M. and D. Montgomery (1998) 'First Mover (dis)Advantage: Retrospective and Link with the Resource-Based View', *Strategic Management Journal* 19(12): 1111–26.
Manovich, L. (2001) *The Language of New Media*. Cambridge, MA: MIT Press.
Nieborg, D.B. (2005) 'Am I Mod or Not? An Analysis of First Person Shooter Modification Culture', paper presented at the Creative Gamers' Seminar: Exploring Participatory Culture in Gaming, University of Tampere, Finland, 11–12 January.
Ondrejka, C.R. (2005) 'Changing Realities: User Creation, Communication, and Innovation in Digital Worlds', 19 January. [Accessed 6 January 2008: http://ssrn.com/abstract=799468]
Postigo, H. (2003) 'From *Pong* to *Planet Quake*: Post-Industrial Transitions from Leisure to Work', *Information, Communication and Society* 6(4): 593–607.
Schneidermann, B. (2002) 'Creating Creativity: User Interfaces for Supporting Innovation', in J.M. Carroll (ed.) *Human–Computer Interaction in the New Millennium*, pp. 235–58. Boston, MA: Addison-Wesley.
Sotamaa, O. (2005a) 'Creative User-Centered Design Practices: Lessons from Game Cultures'. in L. Haddon, E. Mante, B. Sapio, K.-H. Kommonen, L. Fortunati and A. Kant (eds) *Everyday Innovators: Researching the Role of Users in Shaping ICTs*, pp. 104–16. Dordrecht: Springer.
Sotamaa, O. (2005b) '"Have Fun Working with Our Product!" Critical Perspectives on Computer Game Mod Competitions', in *Proceedings of DiGRA 2005 Conference: Changing Views: Worlds in Play* (on CDRom). Vancouver: University of Vancouver.
Taylor, T.L. (2006) *Play between Worlds: Exploring Online Game Culture*. Cambridge, MA: MIT Press.
Terranova, T. (2000) 'Free Labor: Producing Culture for the Digital Economy', *Social Text* 18(2): 33–58.
Thomke, S. and E. von Hippel (2002) 'Customers as Innovators: A New Way to Create Value', *Harvard Business Review* 80(4): 74–81.
van der Graaf, S. (2007) 'Your Second Life: ComMODifying Inventory Toolkits', manuscript under review.
van der Graaf, S. and G. Cobarr (forthcoming) 'The Second Life of Analogue Players in a Digital World', in A. Koohang and K. Harman (eds) *Knowledge Management*. Santa Rosa, CA: Informing Science Press.
von Hippel, E. (2005) *Democratizing Innovation*. Cambridge, MA: MIT Press.

Wenger, E. (1998) *Communities of Practice: Learning, Meaning, and Identity*. Cambridge: Cambridge University Press.
West, J. and S. Gallagher (2006) 'Patterns of Open Innovation in Open Source Software', in H.W. Chesbrough, W. Vanhaverbeke and J. West (eds) *Open Innovation: Researching a New Paradigm*, pp. 82–106. Oxford: Oxford University Press.
Wright, T., E. Boria and P. Breidenbach (2002) 'Creative Player Actions in First-Person Shooter Online Video Games', *Game Studies* 2(2). [Accessed 6 January 2008: http://www.gamestudies.org/0202/wright/]
Yourcatch22 (2007) 'Insurgency Mod: Developer Interview', 4 May. [Accessed 7 November 2007: http://www.clandb.net/featuredcomments&id=6]

Biographical notes

David B. Nieborg is a PhD researcher at the Amsterdam School for Cultural Analysis and a lecturer at the University of Amsterdam. His research focuses on the interaction among participatory culture, game technology and the political economy of the game industry. He contributes to discussions surrounding game culture in various journals, online magazines and national newspapers. ADDRESS: University of Amsterdam, Turfdraagsterpad 9, 1012 XT Amsterdam, The Netherlands. [email: d.b.nieborg@uva.nl]

Shenja van der Graaf is a fellow at the Berkman Center for Internet and Society, Harvard Law School and the Research Center for Information Law, University of St Gallen. Her research focuses on the organization and management of innovation and technology, especially demand-side innovation, product development and media uses in media and software industries. ADDRESS: Berkman Center for Internet and Society, Harvard Law School, 23 Everett Street, Second Floor, Cambridge, MA 02138, USA. [email: shenja@cyber.law.harvard.edu]