THE MOD INDUSTRIES?
The Industrial Logic of Non-Market Game Production

Abstract
This article seeks to make the relationship between non-market game developers (modders) and the game developer firm explicit through game technology. We investigate a particular type of modding, i.e. total conversion mod teams, whose organization can be said to conform to high risk, technologically advanced, capital intensive, proprietary practice of the developer firm. The notion ‘proprietary experience’ is applied to indicate an industrial logic underlying many mod projects. In addition to a rather particular user-driven mode of cultural production, mods as proprietary extensions build upon proprietary technology and are not just redesigned games, as modders using a particular marketing and industrial discourse, and a set of industrial-like practices.

Keywords
First Person Shooter, total conversion modification, game engine, proprietary experience, proprietary extension

Bios
David B. Nieborg is a PhD researcher at the Amsterdam School for Cultural Analysis (ASCA) 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.

Shenja van der Graaf is a PhD researcher at the London School of Economics and a fellow at the Berkman Center for Internet and Society at Harvard Law School and the Research Center for Information Law at the 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.

David B. Nieborg
University of Amsterdam
Turfdraagsterpad 9
1012XT Amsterdam
The Netherlands
d.b.nieborg@uva.nl

Shenja van der Graaf
London School of Economics
Houghton Street
London WC2A 2AE
UK
a.c.vandergraaf@lse.ac.uk
Introduction

In 1999 two young enthusiastic amateur developers Minh "Gooseman" Le and Jess "Cliffe" Cliffe and their team brought us the modification (or mod) Counter-Strike, based on the then popular First Person Shooter (FPS) Half-Life (1998). Half-Life, developed by independent software and technology developer Valve, is a skillfully 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 the 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 setup 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’ (TC) 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 by doing so acquiring Counter-Strike’s valuable intellectual property (IP). In 2000 Valve published Counter-Strike as a commercial game while the Counter-Strike mod continuous to be freely available to those who purchase Half-Life. Today, with over seven billion player minutes a month Counter-Strike still is the most played online multiplayer FPS PC game and its player count mirrors the number of players of all other online FPS games combined.

Unsurprisingly, many who seek to address mod culture hail Counter-Strike’s success. The Counter-Strike story - in cultural and economic terms - and its unrelenting and prominent success provided mod practices a rather unequivocal position in FPS game culture which warrants the question: What is Counter-Strike exactly? For millions of gamers worldwide it is a complex social world with its own rituals and social organizations, such as teams or ‘clans’ (Wright, Boria and Breidenbach, 2002). For developers such as Valve, Counter-Strike is part of a successful franchise that generates significant revenue even eight years after its inception proving 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 Counter-Strike seems to have become the example to live up to. These are attractive reasons for researchers to dedicate their attention to Counter-Strike (Dovey and Kennedy, 2006; Jenkins 2006; Kerr, 2006; Sotamaa, 2005a). Counter-Strike 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 endpoint of delivery, while firms 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 e.g. 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, like Valve, pride themselves in actively articulating the creative endeavors of modders. Such novel migratory practices have an economic and cultural impact by foregrounding the increasing pace of innovation and yield insight into trajectories of innovation and commercialization that are a regular occurrence among software developers, modders, and gamers.

Games in general and computer games (PC) in particular are, in many cases, platforms for user creativity and nowadays modding has become an integral part of modern day game culture. Contemporary game technology - particularly FPS PC game technology - has been purposely designed to assist both professionals and amateur developers in unlocking the capabilities of the game’s core technology, i.e. the game engine. Id Software’s engine programmer John Carmack who is responsible for legendary FPS games such as the Doom, Quake and Wolfenstein series, sees his task to develop 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 actively trying to encourage practices such as modding, skinning and modeling by providing the user with the necessary toolkits. Take for example 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 productive engagement of players and design and commercial interests arguing that because of their labor activities, they provide core value to the world (i.e. beyond beta-testing; cf. van der Graaf and Cobarr, 2007). In this volume Humphreys suggests that we should no longer understand players as ‘end-users’,
but we 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 for instance affords players to design their own interface modifications. Especially for heavy users interface mods have become indispensable gameplay aids. In the heavily regulated realms of World of Warcraft however, 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 FPS 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, by 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, u.r.). Large parts of Second Life’s technology are open source, ‘residents’ retain their IP 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 FPS mods.

In this paper, it is not the general practice of modding that is investigated rather we seek to yield insight into mods as the interstices between Valve, its proprietary game technology, and modders. The way Valve is organized and develops its own positions in conjunction with modders and the way 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 suggest 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, and Team Fortress. 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 (SDK) 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 - i.e. modders. Furthermore, Steam is Valve’s digital distribution and communications platform and is used to digitally distribute and manage over 180 PC games such as FPS, role playing games, and other genres and 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 Hdq. in Bellevue, WA. Due to a non-disclosure agreement notes were taken rather than that interviews were recorded. Hence, within the confinements of this article interviews were used as illustrative background data and not as much as primary sources. Questions focused on the role of each employee 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 Valve is organized. Various mod projects were also 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. In the next section we introduce the technological particulars of modding by focusing on the underlying software code, i.e. the game engine. This will be followed by critically approaching the practices of modders invoking thoughts concerning free labor. These two technological and economic streams result in our conceptualization of total conversion mods as proprietary extensions. Then we will take a closer look at some of Valve’s games and mods from a more cultural perspective by analyzing 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 *an sich* we suggest that the relationship between modders and the firm can be made explicit through game technology. Studies on mod culture tend to either focus on Counter-Strike’s cultural or economic impact while the game’s underlying technology is often 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 FPS 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 (TC) modification. Developing and distributing TC 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. Often times the ambitions of TC mod teams are such that there are positions within a mod team like public relations manager, lead tester, (historical) advisor, and community manager. An explanation of TC modding is provided on the Valve development wiki and echoes 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.)”
Our interest in the specifics of TC mods stems from a mild dissatisfaction with previous research that 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. Our research reports on TC mod teams that seem to conform to the corporate logic of the developer firm when we conclude that TC 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. But first we deconstruct previous notions of what a mod exactly is, thereby focusing on the particular technological, and political economic status of mods and what we refer to as proprietary extensions. In order to do so the crucial function of game engine technology in combination with development tools, or SDKs, needs further explanation.

FPS 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 and consists of several components such as the renderer that visualizes the gamespace, a physics engine, networking code, artificial intelligence code, a sound system and other parts. For modders a FPS 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. Or 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.” (Hodgson, 2004: 5).

Modders’ application of game engines and their wider function within mod development is seldom acknowledged by scholars discussing mods. Bogost, one out of a handful scholars, emphasizes the important role of game engines as component based software systems within the wider game industry and singles out FPS game engines as they: “…construe entire gameplay behaviors, facilitating functional interactions divorced from individual games” (2006: 57). Bogost goes on explaining how engines form the basis for other games and as such share the material, functional, and in the case of the FPS games discussed in this paper, 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 IP 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 FPS engines are brought about by multi-million dollar investments by game developers and are considered proprietary technology: “Like component software, game engines are IP” (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. FPS engine development is iterative and engines have an ongoing development cycle where there’s constant tweaking and updating. Just as games have sequels so do engines. The latest Unreal engine is the Unreal Engine 3.0 and id Software’s experimental technology is called id Tech 5. Dovey and Kennedy (2006) talk in this respect 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 rising 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’, i.e. 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 it enables upgrading particular engine parts without ‘breaking the code’ or for third parties to develop plug-ins, ‘sub-engines’ if you wish, 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 used for ingame collision detection and vehicle dynamics, or Interactive Data Visualization’s SpeedTree package designed to render ingame 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 FPS engine like Valve’s Source engine. Valve programmer Tom Leonard (2007): “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”. 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 modeling software have their own plug-ins to interface with popular FPS engines. Modding is to a large extent bound by the engine’s internal logic which is operationalized through its proprietary toolkit. Generally the necessary tools to make mods of original games are provided by the game engine
developer. The domain of the game developer thus consists of the platform, engine and source code while players get access to game code and often an editor and/or toolkit that allows 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. 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 gratis - 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 we intentionally focus 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 FPS 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 existing game material such as textures are used making these mods more iterative than alterations. TC 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 TC 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). Nor, as we have explained, on a technical level, as parts of the engine and tools are closed off.

Scholarly work on mods tends to generalise modding as simply modifying existing game files and broadly define mods as user developed modifications or “gamer made alterations to commercial technology” (Sotamaa, 2004: 2 in Kerr, 2006). As such these notions overlook 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, differ 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 TC mods add a completely new layer of content this additional material is for commercial games required - through elaborate EULAs - to interact with the game’s proprietary engine. Therefore, TC modifications can be said to be more of an addition to proprietary stand-alone 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, as users, in opposition to licensees, are forbidden to derive direct monetary value from their creations, mods are by definition non-market productions. And because of mods’ dependency on proprietary code they are non-commons based. As such mods can be better understood 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 we said earlier, it is suggested that ‘production’ continues well after the release of a game by 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 and especially TC modders, do eagerly anticipate this subsequent process of commodification up to a point where TC mods serve as part of future game developers’ portfolios (De Peuter and Dyer-Whiteford, 2005; Nieborg, 2005).

Because of such strategies and the unique status of mod culture versus the game industry FPS mod culture has been conceptualized as a type of precarious labour or “playbour”, signaling 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 TC modders are understood as “free labour”, a liminal form of work in between paid and voluntary labour which is specific for the digital economy (Terranova, 2000). There is no question that mods do add considerable value for FPS developers. Postigo (2003: 594) frames modders from a purely economic perspective as “video-game programmer hobbyists” operating within the post-industrial economy, 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 firms can make use of external innovation sources including by 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 (Schneiderman, 1999), and spillovers (Breschi and Lissoni, 2001). Consulting with users however, has come more closely into the firm’s focus than ever before (Jenkins, 2006). This coincides with a considerable scholarly interest in innovation that results from rapidly expanding user activities that are facilitated by enhanced connectivity among users.

Comparing the traditional approach to the user-driven approach, Thomke and von Hippel (2002) indicate that in the traditional model the developer firm takes on most of the product development process which results in an unfavourable situation – i.e. in costs and time - for iterations between firm and user. In the customer-as-innovators 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 firms (Humphreys et al., 2005). As a result, what the product is and what it can become comes to light despite missing skills and an incomplete knowledge base internal to the suppliers (Ondrejka, 2005). The firm 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 mods extend the interest of gamers into the original game’s franchise. This goes for 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 serves as a retention tool. Three main ways can be differentiated to attract external innovations: minimizing technical obstacles because modders “can build upon publisher’s proprietary innovation to make a compelling game experience”, creating an infrastructure that facilitates and encourages participation and collaboration, and peer recognition for generated contributions (West and Gallagher, 2006: 98).

By providing gamers with toolkits for modding activities (or 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 is relevant with regard to co-operation and game engine
technology. It points us to, on the one hand, co-operation 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, co-operation among modders and game developers. In-firm 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. Wide scale, real time user contributions to the development of the before mentioned proprietary engines are rare however.

When critically approaching user-innovation studies, we suggest 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. As such creating a proprietary experience that can be characterized by the game developer seeking to minimize technical obstacles (the engine and toolkits) and to create a consistent infrastructure (Steam and the Valve Developer Community) which arguably means that the ‘user-driven’ practice of modding should not be taken for granted as much as it has been. In the following section a more detailed analysis of the techno-economic status of TC modding is offered. We will again draw on Valve’s games and mods such as the development of Counter-Strike, which inhabit an industry-led production context.

@Valve

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” (Hodgson, 2004: 5).

Like 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. By looking at Valve’s position as independent game developer we can yield insight into emerging and implemented trajectories of participation and commercialization.

In recent years there has been a focus on firms that are rethinking their sources of innovation, after realizing that their products can be modified. Firms may encourage user innovation by providing an ‘open system’ and freely available equipment like toolkits that are generally applied for problem-solving (von Hippel, 2005). Not much attention
has been paid to the implications of user innovations when they are a subset of the economic system through sharing and/or commercialization which is the case in TC 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 organisational 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” (Newell, 2007). Over time Valve has, through the words of Newell, stressed its dedication to support 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.” Valve has a clear strategy to steer gamers’ involvement in the game development process, indirectly through the appropriation of successful TC mods and directly through its engine and toolset. The reason, then, why Valve has been able to continuously attract mod teams is both 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 TC mod teams rather they are fans who are 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” (Newell, 2007). The example of Counter-Strike as a truly bottom-up manifestation of user-driven development however, a similar position is pervasive in academic literature, is somewhat misleading in this respect. In the case of the development of TC mods such as Counter-Strike, the development teams seldom are “just guys”. Dovey & Kennedy give a detailed explanation as to why Counter-Strike did not arrive from 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 in 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 IP which Valve also did with other FPS TC mods such as Team Fortress, Day of Defeat and Portal, thereby uniquely positioning itself vis-à-vis its games. The deployment of expo’s or mod contests, such as the million dollar ‘Make Something Unreal’ contest, are two of the many industry strategies to seek out the more mature mod projects (Nieborg, 2005; Sotamaa, 2005b). The overall winners of the Unreal contest, the World War Two mod team of Red Orchestra: Combined Arms, received 50,000 dollars in prize money and an Unreal Engine commercial license. Later on Red Orchestra: Ostfront 41-45 was published as a retail game through Steam. The developers of the contemporary warfare themed Desert Combat TC mod for the World War Two FPS 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 IP and team were acquired by Battlefield 1942’s developer.

Many TC mods however never materialize and the road from a small mod with some homebrew maps and skins to a full blown TC mod is bumpy. Yet, the high profile cases of Counter-Strike, Red Orchestra and Desert Combat seemed to have raised the bar for TC mod teams. Alex "ACPaco" Capriole (2006), a staff writer for a Half-Life community website summarizes this view as the “Wannabe CS 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.” A view that resonates with Valve developer Robin Walker, who started out as a modder himself: “The biggest thing that mod makers are all getting wrong now unfortunately is they’re focussing on becoming more and more like commercial products which basically means they’re trying to develop more and more assets.”

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

As proprietary extensions TC mods feature certain unique techno-economic characteristics on the level of development, marketing and usage, as such structuring the expectations of gamers, modders and developers and prolonging a shared set of discourse, development practices, and aesthetic conventions. An example of a TC mod
being such a prototypical proprietary extension, following the industrial logic of game developers such as Valve Software, is the Source engine powered TC mod Insurgency: Modern Infantry Combat, or INS. 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.” The development on the mod started in 2002 and at the time of writing released its second beta (work in progress), version 1.1. Not knowing that Insurgency is a mod one would think to be looking at 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.”

Players who mod act as voluntary providers of complements which are freely revealed to the entire community (and possibly beyond). TC mods can be seen as a particular subset of modding and are developed, given to the FPS user base, by a relatively small group of generally highly motivated individuals working together in dedicated teams. Insurgency’s mod team consists of an all male team of twelve nationalities (cf. ‘virtual studios’, De Peuter and Dyer Witheford, 2005), most of them from North-America and Northern-Europe, and is split-up 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.

**Concluding remarks**

Modding in general is often invoked by humanity 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 towards 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 thereby purposely blurring the line between the production and consumption and 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 as such structuring user expectations and maintaining certain aesthetic conventions. We believe that discussions from both critical and administrative standpoints within firms 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 structured of the institution and the emergent structure of practice (Wenger, 1998: 244). This tension between the emergent and the structured, between the 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 firms such as Valve to support their customers can best be described by the way the game developer sets technical limits to what the modder can do with the engine, graphics structure, and the toolkit/editor. Ideally the employment of user toolkits and the community support, which is fully part of Valve’s business model, would result in a wide range of user innovations. As TC mod teams are poised to do ‘the next Counter-Strike’ they are not only inherently limited to the techno-economic cadre which characterizes TC modding, they align themselves along the same rigorous development practices as for-profit developers. As part of the game industry, TC 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 both emulating the first party developers’ risk-averse, capital intensive mode of production and within a proprietary context, TC modding has become a ‘proprietary experience’, as modders anticipate the developers’ much looked after act of re-appropriation and subsequent commodification.

Acknowledgments
We are indebted to Tom Leonard at Valve who kindly opened the doors to Valve Hqt. and who 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.

References


