Abstract

Computer games have increasingly been the focus of user-led innovations in the form of game mods. This paper examines how different kinds of socio-technical affordances serve to organize the actions of the people who develop and share their game mods. The affordances examined include customization and tailoring mechanisms, software and content copyright licenses, game software infrastructure and development tools, career contingencies and organizational practices of mod teams, and social worlds intersecting the mod scene. Numerous examples will be used to ground this review and highlight how such affordances can organize, facilitate or constrain what can be done. Overall, this study helps provide a deeper understanding of how a web of associated affordances collectively serve to govern what mods get made, how modding practices emerge and flourish, and how mods and the game industry serve each others’ interests, though not always in equivocal terms.

Contents

Introduction
Mods, modders, modding, and the mod scene
Discussion
Conclusions

Introduction

Computer game mods are a leading form of user-led innovation in game design and game play experience. But modded games are not stand-alone systems, as they require the user to have an originally acquired or authorized copy of the unmodded game. Thus, there are questions of not only who creates what and who owns such modified games, but also whether or how the practice of game modding is controlled or governed by external parties to ultimately exploit the efforts of game modders. Conversely, if game mods are co-created by game studio developers and user-modders, then it may also be the case that modders create or enable developers, and thus modders govern some of the actions and choices of developers. In this article, we examine these notions by investigating an array of loosely coupled affordances among people, information resources, computing technologies, and institutional arrangements that are expressed or embodied through game mods, modding practices, modders, and the mod scene.

Affordances refer to situated, interactional properties between objects (tangible or symbolic) and actors that facilitate certain kinds of social interactions within a complex environment. The concept of affordances seeks to characterize aspects of complex work settings that facilitate how people interact though socio-technical systems like computer games. Computer-supported game play environments, when effective, afford new ways and means for collaborative learning (Gee, 2008; Hayes and Games, 2008; Scacchi, et al., 2008). However, affordances are neither universal nor ubiquitous, nor are they experienced in the same way in different settings. Accordingly, an affordance that enables or encourages certain kinds of actions in one game mod setting, may inhibit or discourage similar actions in another setting. Modding practices observed for one game may differ from another game, so details and settings matter. Subsequently, the focus in this article is directed at the interactions that facilitate collaborative activities between game modders who are geographically dispersed but share access to online artifacts, networked information repositories and communication infrastructures, such as Web pages, Web sites, source code version servers, distributed file servers, and virtual private networks.

Affordances are enablers of collaboration and governance. For example, the online game modding artifacts, circumstances, and boundary objects can make different forms of governance obscure or coordination/control less visible (Star and Strauss, 1999) and more remote. However, the information resources, social arrangements, technological system configurations, personal and professional interests that surround game mods, modding, and modders continually emerge as a web of associations or interrelated affordances that stabilize actions and practices (Latour, 2003). Consequently, our interest lies in examining where such associations stabilize and entrench the social surround of computer games in different settings, and at different levels of analysis (Kling and Scacchi, 1982; Scacchi, 2004; Scacchi, 2008).

Governance can therefore be perceived through a web of affordances that either push it into a nearly invisible background, or into a visible confrontational foreground, in ways that may be historically situated or
Mods, modders, modding, and the mod scene

Depending on who you ask or where you look, game mods mean different things to different people. So if we seek to understand how game mods as objects, modding as activities or practices, modders as people working in particular situations, or the mod scene as the surrounding social world, we frame our study as one encompassing multiple affordances that naturally arise at different levels of granularity. Accordingly, we choose levels that span (a) game mod embodiments, (b) game software and content licenses, (c) game modding software infrastructure and development tools, (d) career contingencies and organizational practices of game modders, and (e) social worlds intersecting the mod scene. While no claim is made that such levels are unique or definitive, their role is to help ground and reveal how governance of game mods, modding practices, modders, and mod scene occurs across a society of games.

We thus turn to examine modding at each of these levels, starting with what constitutes a mod.

Customizing, tailoring, and remixing game embodiments

Understanding what constitutes a game mod is not as simple as it might seem. Said differently, if in some way you alter a game, how its played, or on what it is played, have you created a game mod? Since we prefer not to presume some prior definition of game mods that might privilege the inclusion or discount the exclusion of some mods, modding practices, or modders, we employ a wide-ranging view. So to start, we conceive of game mods as covering customizations, tailoring, and remixes of game embodiments, whether in the form of game content, software, or hardware denoting our space of interest.

As scholars like Dourish (2001) and Suchman (2007) have observed with other computing technologies, understanding game mods starts from observing how players interact with and reconfigure the game embodiments at their disposal. Computer games and game mods take on tangible form that reflect certain kinds of social computing experiences that are mediated by game system platforms (Montfort and Bogost, 2009). Modding is a ‘Do It Yourself’ approach to technology personalization that can establish both socio–technical and distributed cognitions for how to innovate by restating control over technology design from their producers. Modding is a form of meta–gaming — playing games that play with the game systems. At least five types of game mods can be observed: user interface customization; game conversions; machinima and art mods; game computer customization; and, game console hacking. Each enables different kinds of affordances that govern mods, modding practices, and modders.

User interface customizations act as the medium through which game development studios can support mass customization of their game products, as a strategy for increasing the likelihood of product success through user satisfaction (Pine, 1993). User interfaces to games embody the practice and experiences of interfacing game players to the game system and play experience designed by game developers. Game developers act to constrain and govern what users can do, and what kinds of experience they can realize. Some users in turn seek to achieve some form of competitive advantage during game play by modding the user interface software for their game, when so enabled by game developers, to acquire or reveal additional information that the users believe will help their play performance and experience.

Three kinds of user interface customizations can be observed. First and most common, is the player’s ability to select, alter, or accessorize the characters that embody a player’s in–game identity. Second, is for players to customize the color palate and representational framing borders of the their game display within the human–computer interface, much like what can also be done with Web browsers and other end–user software applications [1]. Third, are user interface add–on components that modify the player’s in–game information management dashboard that do not necessarily modify game play rules or functions. These add–ons provide additional information about game play that may enhance the game play experience, as well as increasing a player’s sense of immersion or omniscience within the game world through sensory or perceptual expansion. This in turn enables awareness of game events not visible in the player’s current in–game view (Taylor, 2006), as the cognition required for sustained game play become more distributed (Taylor, 2009). Kow and Nardi (2010) describe their study of add–on modding practices for World of Warcraft in a cross-cultural setting, which helps reveal how cultural practices in different nations may encourage or not encourage such modding, as a reiteration of norms that tacitly govern social collectivity and conformity, and that eschew radical individuality and innovation.

Game Conversion Mods are perhaps the most common form of game mods [2]. Most such conversions are partial, in that they add or modify (a) in–game characters including user–controlled character appearance or capabilities, opponent bots, cheat bots, and non–player characters, (b) play objects like weapons, potions, spells, and other resources, (c) play levels [3], zones, terrains, or landscapes, (d) game rules [4], or (e) play mechanics [5]. Some more ambitious modders go as far as to accomplish either (f) total conversions that create entirely new games from existing games of a kind that are not easily determined from the originating game, or (g) parodies that implicitly or explicitly spoof the content or play experience of one or more other games via reproduction and transformation.

One of the most widely distributed and played total game conversions is the Counter–Strike (CS) mod of the Half–Life first–person action game from Valve Software. As the success of the CS mod gave rise to millions of players preferring to play the mod over the original HL game, then other modders began to access the CS mod to further convert in part or full, to the point that Valve Software modified its game development and distribution business model to embrace game modding as part of the game play experience that is available to players who acquire a licensed copy of the HL product family [6]. Valve has since marketed a number of CS variants that have sold over 10M copies as of 2008, thus denoting the most successful game conversion mod, as well as the most lucrative in terms of subsequent retail sales derived from a game mod [7].

Other player–modders have benefited from access to meta–mods, such as Garry’s Mod[8] of Half–Life 2. Garry’s Mod has evolved into a modding toolkit [9] that has generated hundreds of game conversions and inventive gameplay mechanics. Though the success of a single game conversion may not foretell a similar success of any subsequent mod (as no game modding to date has reached anywhere the global success attained by CS), other game conversions represent innovations in game design and re–purposing. For example, the game Chex Quest is a conversion of the first–person shooter, Doom, into a “non–violent” game that was distributed in the Chex brand of cereal boxes on a CD targeted to young people and gamers [10].

Another example is found in games converted to serve a purpose other than entertainment, such as the development and use of games for science, technology, and engineering applications. For instance, the FabLab game (Scacchi, 2010) is a conversion of the Unreal Tournament game, from a first–person action
Finally, another class of game conversions are found in the form of game parodies — games created to reproduce and reframe the game play concepts and mechanics found in one or more popular games. For example, the game WPF?! is a 2D, Flash–based, side-scrolling, parody of the 3D World of Warcraft game that shares exportable character information from WoW. But WPF?! introduces new characters like Sigmund Freud, Karl Marx, and Mary Daly who serve to mod the WoW play experience through commentaries and quest directives that act to direct the player’s attention respectively to construction of identity of player game characters, alternatives to capitalistic resource accumulation as game play goal, and dealing with the objectification of the female form within games.

**Machinima and Art Mods** can be viewed as the product of modding efforts that intend to mod play experience [11]. These products employ computer games as their creative media, such that these new media are mobilized for some other purpose (e.g., creating online cinema or interactive art exhibition). Machinima focuses attention to playing and replaying a game for the purpose of story telling, movie–making [12], or retelling of daunting or high efficiency game play experience (Lowood, 2008). Machinima is a form of the retelling experience for game play through a recording of its visual play session history so as to achieve some other ends beyond the enjoyment (or frustration) of game play. These play session histories could then be modded via video editing or remixing with other media (e.g., audio recordings) to better enable cinematic storytelling or creative performance documentation.

Art mods modify the game play experience through manipulation, intervention, appropriation, or other creative transformation of a game’s original visual content as it is consumed by users during a play session. Artists working in the interactive medium of games have explored appropriation and intervention as tactics for using modded games as static, dynamic, or performance art works [13]. Artists were among the first community of game players to embrace game hacks and patches: modifications to original game software functionality as a strategy for creating new works of art intended for curation and exhibition. The 1999 “Cracking the Maze” art exhibit curated by Anne–Marie Schleiermacher helped bring these modified games–like–designs light through an online exhibit that was structured as a C++ source code program for viewing or interaction in an Emacs–like text editor [14]. Art mods can also incorporate alternative game control devices and user interface non–sequiturs [15]. Concepts, artworks, and game mods incorporating such capabilities were first showcased at the Beall Center for Art and Technology in the “Shift–Cntl” exhibit in 2000, in conjunction with the Center’s opening and first art exhibit (LaFarge and Nideffer, 2002).

**Custom gaming PCs** are another expression of game system modding practices. Here players direct their efforts not to merely accessorizing their game platforms, but to go under the hood and to assemble, reassemble, or otherwise reconfigure their general–purpose personal computer into one that is specialized for competitive game play or display in a multi–player environment (cf., Suchman, 2007). What results are something similar to “hot rod” and “custom car” PCs, computer hardware platforms modded for either maximum speed and performance, or special appearance for case mods (Simon, 2007), rather than for mere operation of mainstream software applications or casual game play. Hot rod game PCs feature high performance mainboard and overclock–able GPUs with liquid–cooling devices, custom–timed on–board memory chipssets, multiple GPU cards, and a variety of accelerator cards for in–game physics, networking, game audio, and so forth [16]. These PCs embody a desire to achieve competitive game play advantage for some, or sheer display of performance or signification of performance potential, much like automobiles are modded by their owners for the purpose of increasing their driving performance on the road, or as might be called for in a street racing situation [17]. Though public practice of such situations with hot rod street cars is often isolated and criminally penalized when excessive, PC platform hot rodding is yet to achieve such distinction, public recognition, consequence, or cinematic embodiment.

PC case mods, on the other hand, generally do not modify game functionality or performance. Instead they serve to signify a game player’s interest or technological projection of self–identity onto their game play platform. Such projection denotes an unabashed choice to display one’s enthusiasm, alignment, and commitment to game play as more than just entertainment, but as part of one’s personal identity, fetish, cultural experience and life-style preference. Nonetheless, enthusiasts who do so mod their computers, also showcase off their machines in more public venues like regional LAN parties or game play conferences like QuakeCon [18]. Custom gaming PCs thus represent a strategy for game play enthusiasts to proudly assert their embrace of game culture and the vendors that supply the means to display such embrace, though while being able to pose in stance that signifies independence from the governance telegraphed by corporately designed PCs.

**Game console hacking** is a practice whose purpose often times seems to be in direct challenge to the authority of game console manufacturers that represent large, global corporate interests. Console hacking, in contrast to custom PC (re)configuration, is often focused not so much on how to improve competitive advantage in multi–player game play, but instead is focused on expanding the range of experiences that users may encounter through a game console (Consalvo, 2007; Huang, 2003). For example, Huang’s (2003) treatise on how to hack into a Microsoft Xbox console stresses how to unlock the potential of the Xbox as an alternative personal computer that could also support personal productivity applications or services, along with playing proprietary games acquired from Microsoft game vendors. However, Huang’s study instead focuses on the practice of “reverse engineering” as a strategy to understand both how a computing platform was designed and how it operates in fine detail, as a basis for developing new innovative modifications or original platform designs, such as installing and running a Linux open source operating system, in place of Microsoft’s proprietary closed source offering [19].

While many game developers seek to protect their intellectual property from reverse engineering through end–user license agreements, whose terms attempt to prohibit such action under threat of legal action, reverse engineering is not legally prohibited nor discouraged by the Courts. Consequently, the practice of game console modding is often less focused on enabling players to achieve competitive advantage when playing retail computer games, but instead may encourage those few so inclined in the practice for how to understand and ultimately create computing innovations through reverse engineering [20] or other DIY hardware modifications [21]. Console modding is thus an expression of game players who are willing to fore go the “protections” and quality assurances that console developers provide through product warranties, in order to experience the liberty, skill and knowledge acquisition, as well as potential to innovate, that mastery of reverse engineering affords. So while game console developers and retailers may seek to govern the actions that discourage users from looking inside of the gaming platforms, players who are willing to take responsibility for their actions (and not seek to defraud vendors due to false product failure warranty claims), can enjoy the freedom to learn how their gaming systems work in intimate detail and to potentially learn about hardware innovation with the support of others like–minded.

**Software and content licenses**

Players acquire computer games through retail outlets (online or off–line stores). But what they pay for when they “buy” a game is a license to use the game product or product media. This license is commonly expressed as a copyright–based, end–user license agreement (EULA), much like it is for most other software that...
consumers acquire. The terms and conditions of such license stipulate what rights and obligations a user of a licensed game copy realizes, as the game software and content is owned by the game's development studio, publisher, or distributor. So here we examine how a game's software and content licenses can govern mods, modding, and modders.

First, for game conversion mods, it is common practice that the underlying game engine has one set of license terms and conditions to protect original work (e.g., no redistribution), while game mod can have a different set of terms and conditions as a derived work (e.g., redistribution allowed only for a game mod, but not for sale). In this regard, software licenses embody the business model that the game development studio or publisher seeks to embrace, rather than just a set of property rights and constraints. For example, in Aion, an MMOG from South Korean game studio NCSoft, no user created mods or add-ons are allowed, so attempting to incorporate such changes would conflict with its EULA and subsequently put such user–modders at risk of losing their access to networked Aion multi–player game play. In contrast, WoW, also an MMOG, allows for UI customization mods and add-ons only (Kow and Nardi, 2010), but no other game conversions, no reverse engineering game engine, and no activity intended to bypass WoW's encryption mechanisms. And, in one more variation, for games like Unreal Tournament, Half-Life, NeverWinterNights, Civilization and many others, the EULAs encourage modding and the free redistribution of mods without fee to others who must have a licensed copy, but no reverse engineering or redistribution of the game engine required to run the mods. This restriction in turns helps game companies realize the benefit of increased game sales by players who want to play with known mods, rather than with the unmodified game as sold at retail. Mods thus help improve game sales, revenue, and profits for the game development studio, publisher, and retailer (see note 23 for details).

Second, it is sometimes noted that game software licenses fundamentally differ based on whether they seek to insure copyright or copyleft, where the former provides closure and prevents certain freedoms to act, while the latter issues openness and certain license freedoms — freedoms of choice, freedom of expression, and freedom of modify and redistribute, plus obligation to propagate these freedoms. So it is sometimes stated that game conversion mods can be viewed and treated as free software. But such a position is an overstatement of fact in that free/open source software that is protected with a GPL license covers the source code and embedded documentation, but not game assets, contents, play mechanics, or rules of play, which can also be subject to protection through copyright, trademark, patent, secret use, and restricted use.

Finally, “open gaming licenses” have begun to appear which allow restrict game configurations that include open source software along with proprietary data assets/artwork, game play rules, or game product identity. In sum, game software and content licenses act to govern what mods can or cannot be made, by whom, and with what subsequent usage rights and contract/social obligations. But it is also clear that such governance is only effective where considered legitimate, as there are those who will ignore declared IP restrictions or act to crack open games to mod them in some way, and then share their results with others. Thus we should recognize that there is also resistance to governance, in the form of those who choose to ignore the license terms and conditions that come with closed, restricted games.

### Game software infrastructure and development tools

Games are most often modded with tools that provide access to an internal, unencrypted representation of the game software or game platform. While it might seem the case that game vendors would seek to discourage users from acquiring such tools, we observe a widespread contrary pattern. Game system developers are increasingly offering software tools for modifying the games they create or distribute, as a way to increase game sales and market share. SDKs and other modding tools provided to users by game development studios represent a contemporary business strategy for engaging users to help lead product innovation from outside the studio (von Hippel and Katz, 2002; Jeppesen, 2005). Once id Software and Epic Games, maker of the Unreal series of game, started to provide game play enthusiasts (who were also literate in computer programming) with software tools that would allow users to edit game content, play mechanics, or other functionality, other competing game development studios were pressured to make similar offerings or face a possible competitive disadvantage in the marketplace. However, these tools do not provide access to the underlying source code that embodies the game engine — a large software program infrastructure that coordinates computer graphics, user interface controls, networking, game audio, access to middleware libraries for game physics, and so forth.

A tool support platform allows a game development studio to offer game products and services that users can mod. At the same time, studios seek to control access to the core IP that enables game play and modification. Accordingly, we can observe different strategies for how game development studios configure and support game modding platforms. For example, World of Warcraft from Blizzard Entertainment, provides a UI Customization tool and a governance policy that enable WoW add–ons to reconfigure a WoW player's user interface dashboard, which in turn may provide an enhanced game play experience for technically accomplished players. But these add–ons do not modify or convert the WoW game into something entirely different and seeks to assure and control that users with or without add–ons have access to the same WoW in–game play setting and game play mechanics. In contrast, BioWare offers its NeverWinterNights game series with its Aurora Toolset and module construction toolkit (FAQs, online tutorials, and in–game character profiles to help would be user–made content creators to modify NWN game play levels and characters. But creating new characters for such mods requires additional tools (e.g., modeling software like Autodesk 3D Studio) and corresponding specialized skills. Similarly, Epic Games recently announced that it is distributing a suite of a dozen or so game development tools as the Unreal Development Kit (SDK), which is said to be the same SDK that Epic previously marketed and distributed only to other game development studios that acquired its commercial game software development licenses. However, the complexity and capabilities of such a tool suite mean that any one, or better said, any game development or modding team, can now access the tools to build high quality games, though this master requires these games to be a significant undertaking likely to be only of interest to highly committed, would–be game developers who are self–supported or self–organized.

Subsequently, we observe a range of corporate governance and control schemes that vary across game development firms, from (a) highly focused and limited at Blizzard for WoW, to (b) supported mediation at NWN by Bioware, to (c) do whatever you can if you have the skill, resources, team, and other resources to do so and sell your results, if you pay for such right with Epic Games. But all these schemes still stipulate control over the underlying game engine, such that it is out of reach to game players, as a way to protect key IP resources of the game studios.

In contrast to game modding platforms provided by game development studios, there are also alternatives...
provided by the end-user community. One approach can be seen in facilities provided in Garry’s Mod, a mod-making package that you can use to construct a variety of fanciful contraptions, to create comic books, to program game conversions, or to produce other kinds of user-created content. But this package requires that you own a licensed game like Counter-Strike: Source, Half-Life 2, or Day of Defeat: Source from Valve Software. A different approach to end-user game development platforms can be found arising from free/open source software games and game engines. As noted above, the Doom and Quake games and game engines were released as free software, and made subject to the GPL, once they were seen by id Software as having reached the end of their retail product cycle. Hundreds of games/engines have been developed and released for download starting from the free/open source software that was the platform of the original games. However, the content assets for many of these games (e.g., in-game artwork) are not covered by the GPL, and so developers still must acquire a licensed copy of the original game if its content is to be reused in some way.

Nonetheless, some variants of the user-created GPL’d games now feature their own content that is limited/protected by Creative Commons licenses.

**Career contingencies and organizational practices of modders alone and together**

Who gets “paid” to make games when comparing developers versus modders? Developers who work for established game studios get paid a monetary salary and sometimes stock options (financial capital) to make game content, software, or play devices. Modders in general do not get a paid to make games, unless they are unusually successful by winning an open modding competition (Sotamaa, 2007) or landing an angel/venture capital investment. However, modders can earn or acquire other kinds of career-oriented resources of professional or occupational value. These resources include deep game development knowledge, skills for how and when to apply such knowledge, status among peers, community recognition and reputation, and other forms of social capital for the gifts of labor, creativity, expertise, and collaboration efforts they contribute (Portes, 1998). In this regard, both employed game developers and independent game modders can build on the efforts of one another in ways that are asymmetric and non-equivalent, yet beneficial to each. Nonetheless, such a relationship is valuable to those who are committed to a career in game development inside or outside of the computer game industry. This is similar to what has been observed in a number of studies of open source software development projects, where in some cases, corporations that sponsor an open source project look to recruit and hire project contributors who make substantial and highly regarded software contributions to a project (Scacchi, 2007). Furthermore, it has also been observed that developers who have a track record of success in open source software projects often get paid a higher monetary salary compared to their workmates who lack comparable experience (Hann, et al., 2002). So it may be the case that game modders have the potential to invest in themselves as a way to acquire professional identity, accomplishments, and compensation if they choose to direct their career trajectory towards becoming a game developer who works in a game studio. However this remains an open question for further study to substantiate, refine or refute. In the interim, what kind of investment is required of a game modder to enable such career contingencies?

To begin, why can’t you get a job working for a game company if you are a software developer or artist? In simple terms, the computer game industry often looks to hire experienced game developers, rather than just game players or fans. Commercial game development is a risky venture, as most retail games often fail to recoup their development costs (as do many commercial software development efforts). Failure to produce a profitable game is all too familiar of an outcome, even for established game studios with skilled developers. Adding new staff with little or no prior game development experience implies the need for training and socialization, which is time-consuming and may be unproductive. Game studios tend to hire developers or artists who have some sort of record of game development. So without prior industry experience, starting as an independent game modder is the easiest way to determine whether one has the skill, disposition, and commitment to make games, even without a job to do so. Consequently, game modders must start and follow a self-motivated path.

The most direct way to become a game modder is through self-tutoring and self-organizing practices. Modding is a form of learning — learning how to mod, learning to be a game developer, learning to become a game content/software developer, learning computer game science outside or inside an academic setting, and more (El-Nasr and Smith, 2006; Hayes and Games, 2008; Scacchi, 2004). Modding is also a practice for learning how to work with others, especially on large, complex games/mods. Many would-be modders seek out others who have gone before them, much like game players at lower game play levels sometimes seek advice from players at higher levels for how best to proceed. In acting to become a modder or game developer, it may be reasonable to search out others like minded by joining a modding team, learning modding practices and mod team roles, and why you can’t be in charge until you are recognized as being capable of leading. Mod team efforts may also self-organize around emergent leaders or “want to be” game component development leaders, for example, as highlighted in the Planeshift open source MMOG development/modding project.

Modding is also a viable way to get a job in the game development industry. Epic Games for many years has encouraged would-be game developers to download its game editing tool — UnrealEd, the early precursor to what is now the UDK described earlier), start modding an Unreal game, and then perhaps get hired by a game development studio as an experienced game developer. Similarly, to stand behind its advocacy of such a career pathway into a game development studio, Epic Games has also sponsored a series of mod-making contests, most recently under the name of “Make Something Unreal” that provides sizable cash prizes (with co-funding from Intel Corporation), or a full commercial license for UDK, so that a game conversion mod can be marketed and sold. Such an outcome essentially provides the winning mod team with venture seed capital, but without the venture capitalist taking partial ownership of the new venture.

Making game conversion mods may also be open pathways for modders from a small game studio with limited resources to move upward into a large game studio with ample resources and multiple game development projects. However, some scholars (Kücklich, 2005; Somataa, 2007) are skeptical about the value of modding, which they see as “playbour” (a coercion of leisurely play into work) or a commodization of leisure time, and subsequently conclude that modding work primarily benefits game companies by exploiting the productive efforts of game modders. However, if game modders act out of self-interest to establish a career path into jobs at game companies, and if they get to create conversions that open new career opportunities in other industries, then it would seem that modders use modding as a way to exploit game development studio’s desire to hire skilled game developers. Consequently, our view going forward may now need to reflect that modding as work can be productive to both independent mod makers and commercial game studios, but in different ways that may further reinforce and reproduce one another.

So our position is the marketplace of career contingencies for game developers and game modders governed through an invisible hand of individual, team, and corporate self-organizing interests to help select and guide who will be able to work for paid salary or who will accumulate skill, experience, and social capital as a game or game mod developer.

**Social worlds intersecting the mod scene**

Computer gaming is such a large-scale, globally diverse social endeavor that involves hundreds of millions of people as to merit its conception as a computer-centered social movement, a computationization movement...
Let’s consider how the computer game mod scene intersects with the Warez scene, and with the world of free/open source software development [38].

The Warez scene [39] focuses attention to activities that endeavor to modify protected software and other digital media (recorded music, movies, console circuit designs) so as to make them widely available for unlettered access and consumption. Games are one of the most commonly modified types of software that are transformed in the Warez scene into “pirated games” or game warez that are “illegally downloaded.” The Warez scene in a sense is focused on engaging a kind of meta–game that involves modifying game software products. Game piracy has thus become recognized as a collective, decentralized and placeless endeavor (i.e., not a physical organization) that relies on torrent servers as its underground distribution venue for game warez. Game development studios, publishers, and some retailers all frequently lament their belief that game piracy is a kind of theft that is reducing their game sales and profits, as well as circumventing extant Digital Rights Management mechanisms (e.g., Greenburg and Irwin, 2008). As recent surveys of torrent-based downloads reveals, in 2009 the top 10 pirated games represented about 9M downloads, while in 2009 the top five pirated games represent more than 13M downloads, suggesting a substantial growth in interest in and access to such modded game products [40]. And these figures do not account for sharing or exchange of game warez on private peer-to-peer networks or darknets. Finally, it seems that in parts of the physical world like India and throughout Asia, game consoles can be purchased directly from retailers already modded with modchips that allow them to run game warez, including games not available in their national retail markets. This is not to impune some negative social connotation on other cultures, but instead to point out that not all national cultures share Western values or beliefs about what can be shared or what piracy means, and under what conditions or corporate institutions may be affected.

In the case of game warez and modded game consoles, it may be the situation that the developers and distributors of such products still financially benefit from pirated warez, if they can substantiate that such piracy is theft of products with economic value. Such theft can be treated as a legitimate business deduction when determining tax payments to national tax ministries. That is, piracy reduces the amount of profits that are subject to tax, since established losses due to theft (e.g., number of pirated games downloaded from torrents) may be tax deductible. Eliminating all game piracy might then actually work against the financial interests of game studios or publishers, if the people who download do not otherwise act to buy games that would now be unavailable to them. Consequently, downloading pirated games from torrents may just represent another channel for distributing products and realizing benefits, with certain costs.

The Warez scene appears increasingly effective at cracking closed games and distributing modded games (or if you prefer, pirated games) and game platforms compared to the capabilities of game studios, publishers, and console manufacturers. So is such accomplishment merely a sign of growing resistance on the part of game consumers, or does it represent an emerging transformation of the marketplace for how games and mods, modding practices, and modders are developed, distributed, played, remixed, reproduced and re–purposed? Is it for or against the interests of game studios and publishers to distribute game software whose DRM techniques can be readily cracked and bypassed? It appears that many interests are served by the ways and means of the Warez scene.

Game mods, modding practices, and modders are in many ways quite similar to their counterparts in the world of free/open source software development. Modding is to games, like FOSS development is to software — they are increasingly becoming a part of mainstream technology development culture and practice. Modders are players of the games they construct, just like FOSS developers are users of the systems they develop. There is no systematic distinction between developers and users in these communities, other than there are users/players that may contribute little beyond their usage, word of mouth they share with others, and their demand for more such systems. At FOSS portals like SourceForge.com, which in December 2009 indicates more than 380K projects are registered in its repository, the domain of “games” appears as the third most popular project category with over 38K projects, after the domains of “software development” and “Internet.” Of these game-focused FOSS projects which develop either FOSS-based games, game engines, or game tools/SDKs, all of the top 50 have logged more than 1M downloads. So the intersection of games and FOSS covers a substantial social plane, as both modding and FOSS development are participatory, user–led modes of system development, and both rely on continual replenishment of new participants joining and migrating through project efforts, as well as new additions or modifications of content, functionality and end–user experience (Nieburg, 2005; Scacchi, 2004; Scacchi, 2007; Scacchi, 2008). Mods and FOSS develop projects are in many ways socio–technical experiments to prototype alternative visions of what innovative systems might be in the near future [41], and both are widely embraced and practiced primarily as a means for learning about new technologies, new working relationships with potentially unfamiliar teammades from other cultures, and more (cf., Scacchi, 2007).

Overall, game mods, game warez, and FOSS can be recognized as expressive forms of participatory culture. Though as we have found above, there are different modes for governing such participation depending on the affordance of hand. But in each situation, governance realizes an emerging transformation the marketplace of ideas and the means of production from centralized corporate authorities towards a more decentralized commons–based peer production (Benkler, 2006). Furthermore, sometimes such production may seek its own non–profit foundation or for–profit incorporation, or it may eschew any established organizational form in order to flourish in an alternative market that may not yet be recognized, or one that is being pursued in hope of preventing its move into a position of dominance.

**Discussion**

One way for viewing and understanding game mods is through the lens of ownership. Ownership influences or controls the allocation of game modding resources, markets or venues where game mods are transacted, or other decisions pertaining to which game mods are created, shared, played, and remixed/remodded by who, when, and where. Such a perspective centralizes governance with those who control ownership of property (game engines, licenses, and SDKs) and the means of game mod production. Alternatively, a decentralized view of governance draws attention to intrinsic governance activities, patterns, and practices, where ownership is grounded in individual, small group, or project team action and resources. Based on the preceding examination of game mod embodiments, game software licenses, modding infrastructure and tools, career opportunities, and social worlds intersecting game modding, it seems that it is not the case that one of these two structural perspectives is more spot on compared to the other, as conditions and circumstances that...
Conclusions

In this study, we started with the quest to examine how the mod squad constituted from game mods, modding practices, and modders are governed, and to what ends. Affordances of five different types serve as guideposts at different levels of analysis that helped to navigate the path forward. These affordances center about game mod embodiments, game licenses, game modding tools and infrastructure, career contingencies of game modders as would-be game developers, and social worlds that impinge on the game modding culture. Along the way, we examined, employed, or referenced the kinds of analytical constructs or results that others have previously employed in their studies of game mods.

We have seen that there are no single category of object, activity or relationship that uniquely denotes what a game mod is. Instead, mods can appear as game user interface add-ons, game conversions of various kinds and complexity, machinima and interactive art, custom gaming PCs, or hacked gaming consoles. Viewing and locating game mods within such a technological ecosystem provides a better perspective for understanding what mods are or can be, who makes them, and why. Mods or mod embodiments are becoming actors that participate in governing the mod squad.

We have seen that game software licenses (and by implication, game platform and console hardware designs) embody the business models of game development studios. Further, game licenses are increasingly multimodal and heterogeneous, thus challenging nearly all to determine the rights and obligations of who can mod what with what. This suggests that such licenses will become a fertile ground for debate regarding what kinds of mods can be made, what modding practices and tools will be provided or tolerated, and who can mod what. Such debates over licenses as a mode of governance and resistance may increasingly spill over into the Courts as well.

Game modding tools, SDKs, modchips, and impinging open source software alternatives for content creation or game engine transformation are collectively an emerging socio-technical media. And as new media, they serve stimulate, enact, or communicate continuing innovation, growth, and diversification by game development studios, game modders and others. The technologies of game modding thus help govern what kinds of mods and modders will emerge, flourish, or wither over time, as well as how the game industry and modding communities can cultivate and grow new markets for new game products.

Game modders are sometimes viewed as leisure laborers whose playful modding actions primarily contribute to the growth of economic rents collected by game studios. However, modders can also be viewed as independent and self-organizing actors who are acquiring the means for producing their own games or game conversions through self-serving investment in skill and knowledge acquisition, time, effort, and socialization with others like-minded. Modders can sometimes become entrepreneurial, and thus transform themselves from game consumers into producers. Consequently, the career contingencies of game modders cannot be simply assumed to be fully covered by some pre-determined set of producer-consumer arrangements or outcomes. Modders make mods through modding practices and technical means, and this in turn helps to produce more modders. Modding careers, contingencies, and pathways/trajectories thus help govern who becomes a modder and to what end.

The social world of game modding provides a system of beliefs, values, and norms that mod makers and users assimilate and reproduce in online modding forums and Web sites. This culture of game mods is enabled and embraced in the social arrangements and technical system configurations that frame what it means to be a game modder, to engage in game modding, and to make and play with a game mod. However, the world of game modding is not an island onto itself. Instead, like all social worlds, it is one that intersects some other worlds, and where and how these intersections occur is a locus of collective social movement, resource redistribution, market transformation and institutional conflict. Games are being modded by actors in the Warez scene in ways that are transforming who get access to acquire and play what games where. This outcomes of this might be lamented as piracy or recognized as tax shelter on accumulated economic rents. Game modding is also influenced by, and influencing, the world of free/open source software development and its diverse software development projects and practices. Thus where, how, when, with whom, and with what the world of game modding intersects other social worlds that impinge on its boundaries will govern and frame why such intersections will arise, and to what end.

About the author

Walt Scacchi is a senior research scientist and research faculty member in the Institute for Software Research, and also Director of Research at the Center for Computer Games and Virtual Worlds, both at University of California, Irvine. He received a Ph.D. in Information and Computer Science at UC Irvine in 1981. From 1981–1998, he was a professor at the University of Southern California. Dr. Scacchi returned to UC Irvine in 1999. His research interests include open source software development, computer game culture and technology, virtual worlds for modeling and simulating complex engineering and business processes, developing decentralized heterogeneous information systems, software acquisition, and organizational analysis of system development projects. Dr. Scacchi is an active researcher with more than 150 research publications, and has directed more than 50 externally funded research projects. He also has had numerous consulting and visiting scientist positions with more than 25 firms or institutes, including four start-up ventures. His recent research activities and publications can be found at [http://www.ics.uci.edu/~wscacchi](http://www.ics.uci.edu/~wscacchi).
Acknowledgements

Support for this research is supported through grants from the National Science Foundation #0534771 and #0807873, and also the Digital Industry Promotion Agency, Global Research and Development Collaboration Center, Daegu, South Korea. No review, approval, or endorsement implied. Also, Garnet Hertz at UCI and the reviewers provided helpful comments on an earlier version of this article.

Notes

1. For example, the Firefox Web browser provides “themes” and “personas” as a way to customize the appearance of a browser’s user interface. See http://support.mozilla.com/en-US/kb/Using+themes+with+Firefox?bl=n&s=themes, accessed 30 March 2010.

2. Much, but not all of, the legacy of game conversion mods originates from the practices of the game software development studio, id Software, with legendary game developers John Carmack and John Romero. They tired of responding to game players requests for additional game features in their existing game products like Wolfenstein 3–D and Doom, while they sought to focus their attention to new games in development. Subsequently, they chose to hand over the source code to the Doom game to their end–user community, and then make the resulting source code subject to free software licensing rights, while still obligating user–modders to acquire the original retail version of the game to access in–game content assets. Many eager users embraced this open sharing of the Doom game engine code, as well as the modding of a Doom engine source code, who was also a software developer, how the game engine was designed and coded, and thus how to modify or reproduce such an engine, or to build a new one, based on knowledge of the source code and skill in software development. The details of this story are more comprehensively described elsewhere, such as in Kushner (2003), and also in studies by Au (2002), Kücklich (2005), and Morris (2003) among others.

3. For example, the game Little Big Planet, which features game level creation as one of its core game play mechanics, has realized the creation and upload/sharing of 1M levels within its first nine months of release, and now offers more than 1.3M user–rated levels. See http://www.littlebigplanet.com/en-us/game_guide/ps3/playing/community_levels, accessed 30 March 2010.

4. Nomic games (http://www.nomic.net/, accessed 30 March 2010) are those where changing the rules of a game is move, such that rule changing, generally with the agreement or vote of the game’s players, becomes a play mechanic.

5. A popular Web portal where modders share and discuss their mods with modding fans who download, play, and rate the mods, can be found at http://www.moddb.com. These fans in turn may be just mod players, or they may be other modders who seek to learn about what others have accomplished, with whom to consult for advice, or with whom to try to join with in developing more ambitious game conversion mods. Mod fans or users also may consult http://www.usercreated.org for news and reviews of current mods, accessed 30 March 2010.


8. Game modding toolkits are also identified as “software development toolkits” (SDKs). However, they are not general purpose software development environments, but instead are specific to a game or family of games that share a common game engine. Software engineers designate such toolkits as domain–specific SDKs that center about the use of a domain–specific programming language, like UnrealScript for the Unreal game engine.


10. See http://aoedipus.net/ for access to the WTF?! game, and also to http://www.nideffer.net/promo/proj/wtf.html for an overview. The SDK also supports subsequent modding of WTF?! see http://nideffer.net/promo/proj/wufe.html, accessed 30 March 2010.


13. Velvet–Strike is an example of an interventionist art mod that alters the CS mod of Half–Life by artists Anne–Marie Schleiner, Joan Leandre and Brody Condon in 2002, so as to allow for inclusion of anti–war graffiti within the tactical small–group battleground of a CS game level/world. See http://www.openconspiracy.net/velvet-strike/, accessed 30 March 2010.


16. Print–based and online periodicals like MaximumPC (http://www.maximumpc.com/), accessed 30 March 2010) and CustomPC (http://www.custompc.co.uk, accessed 30 March 2010) focus attention and provide illustrated guidance for how to modify PCs to improve game run–time performance, and routinely employ performance intensive computer games like Crysis and Cry Cry 2 (both of which can be modded) to demonstrate and document the results of PC systems before and after modding.

17. This kind of technological expression, identification, and eros is dramatized as popular culture in feature

19. It should also be noted that in response to such dissemination of knowledge for how to hack into the Xbox system, that Microsoft invested a great deal of effort and resources to insure its subsequent Xbox 360 game platform could not be so readily hacked, and it seems to have mostly succeeded in these efforts, though to the disadvantage of its users, who must consequently look elsewhere to learn about reverse engineering and platform hacking as a basis for system innovation or customization. In this regard, Microsoft’s actions may suggest it does not trust the consumers of its products to engage in activities that might improve their personal situation, but instead acts to protect what Microsoft perceives to be its corporate interests within its dispersed intellectual property.

20. Such practice is to be distinguished from suspect or illegal misappropriation of trade secrets through covert acquisition of materials or documents from insiders or other illicit sources, rather than as a consumer who is inquisitive and willing to void their product warranty to look inside the back box to see what’s there, and who is willing to exercise the freedom to figure out how it does or doesn’t operate, and who can create or modify what is found to achieve new operational capabilities.


22. Prior to the release of WoW, Blizzard Entertainment actively pursued and ultimately shut down the independent development of a Battlenet server portal called bnetd.org that was claimed to have been reverse engineered from public information. But Blizzard successfully prosecuted that such modding entailed work-around or breakage of in-game anticircumvention mechanism, and that such infringement made the bnetd.org modders legally culpable. The Court then ruled in favor of Blizzard Entertainment and bnetd.org was shut down. See Davidson & Assoc. et al., v. Jung et al., 422 F.3d 630, 633 (8th Cir. 2005).

23. Kücklich (2005) reports the EULA for the Half-Life SDK “… hereby grants Licensee a nonexclusive, royalty-free, terminable, worldwide, non-transferable license to: (a) use, reproduce and modify the SDK in source code form, solely to develop a Mod; and (b) reproduce, distribute and license the Mod in object code form, solely to licensed end users of Half-Life, without charge.”

24. Information on the Mozilla tri-license for the Firefox source code (the MPL as a minimally restricted license, the GPLv2 as a reciprocal license, and the LGPLv2.1 as a library-style license) can be found at http://www.mozilla.org/MPL/, accessed 30 March 2010.


26. In a game like Doom, this internal representation is denoted by a WAD file (Kushner, 2003; Morris, 2003).


32. The game development studio, Crytek, has begun to offer job interviews at their company as a prize for winning a game modeling contest that they host. Further information at http://crymod.com/thread.php?threadid=56783, accessed 30 March 2010.

33. For example, the lead designer of the Civilization V game from Firaxis Studios, Jon Shaffer, started his career as a modder of earlier Civilization games. His mods were recognized as being of sufficient quality, and his academic studies in European, nineteenth century German, and World War II history aligned with his Civilization modding interests. Subsequently, he was recruited to work at Firaxis, and has become a lead designer there. See Murdoch and Wilson (2010) for details.


37. The development of the Chex Quest game conversion by a small game studio did allow for one of its first-time mod makers, Chuck Jacobi, to subsequently become lead game artist for Electronic Arts LA, a major game studio. See Lathi (2009) for details.

38. Scacchi (2008) also draws comparison of the world of computer games, FOSS, and scientific research computing.


41. As another major area of FOSS development activity (at SourceForge and elsewhere) is in the domain of Science/Engineering, we should not be surprised to soon see new FOSS game engines or game physics...
libraries being modded to support more complex physical computations beyond forces and motions, and eventually include electro–magnetism, quantum chromodynamics, and celestial mechanics, as well as chemical, biological, geological, and meterological processes. Conversely, as such capabilities come into being, then we expect scientists and students working in such areas to increasingly see their experimental studies employing these new science game mods (Mayo, 2007).

References


This page is for cutscene commands modding with the posemode by kgftbz. This page is allowed to be modified only if the adjustment is correct and positive. Make sure your command is up to date with the newest pose mod or it may crash the game when you try to load your cutscene. Getting Started. For creating cutscenes, the only things you will need are Kgftbz's pose mod and Notepad (or Notepad++). To create a new cutscene, make a text file in the cutscenes folder of the pose mod.

Parameters. When looking through commands, you may see words like bool or int.

A Great Cities Initiative of the University of Illinois at Chicago University Library.

© First Monday, 1995-2019. ISSN 1396-0466.