Chapter 9
Modding as an Open Source Approach to Extending Computer Game Systems

Walt Scacchi
University of California, Irvine, USA

ABSTRACT

This paper examines what is known about the role of open source software development within the world of game mods and modding practices. Game modding has become a leading method for developing games by customizing or creating Open Source Software extensions to game software in general, and particularly to proprietary closed source software games. What, why, and how OSS and closed source software come together within an application system is the subject for this study. Observational and qualitative is used to highlight current practices and issues that can be associated with software engineering and game studies foundations with multiple examples of different game mods and modding practices are identified throughout this study.

INTRODUCTION

User modified computer games, hereafter *game mods*, are a leading form of user-led innovation that generally rely on toolkits (von Hippel, 2001) to support game design and game play experience. But modded games are not standalone systems, as they require the user to have an originally acquired or licensed copy of the unmodded game software.

Modding, the practice and process of developing game mods, is an approach to end-user

game software engineering (Burnett, Cook, & Rothermel, 2004) that establishes both social and technical knowledge for how to innovate by resting control over game design from their original developers. At least four types of game mods can be observed: user interface customization; game conversions; machinima; and hacking closed game systems. Each supports different kinds of open source software (OSS) extension to the base game or game run-time environment. Game modding tools and support environments that support the creation of such extensions also merit attention. Furthermore, OSS game exten-

sions are commonly applied to either proprietary, closed source software (CSS) games, or to OSS games, but generally more so to CSS games. Why this is so also merits attention. Subsequently, we conceive of game mods as covering customizations, tailorings, remixes, or reconfigurations of game embodiments, whether in the form of game content, software, or hardware as denoting our space of interest.

The most direct way to become a game mod developer (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 & Smith, 2006, Scacchi, 2004). Modding is also a practice for learning how to work with others, especially on large, complex games/mods. Mod team efforts may self organize around emergent software development project leaders or "want to be" (W.T.B.) leaders, as seen for example in the *Planeshift* (http://www.planeshift.it/) OSS massively multiplayer online role-playing game (MMORPG) development and modding project (Scacchi, 2004).

Game mods, modding practices, and modders are in many ways quite similar to their counterparts in the world of OSS development, even though they are often seemingly isolated to those unaware of game software development. Modding is increasingly a part of mainstream technology development culture and practice, and especially so for games, but also for hardware-centered activities like automobile or personal computer customization. Modders are players of the games they reconfigure, just as OSS developers are also users of the systems they develop. There is no systematic distinction between developers and users in these communities, other than there are many users/players that may contribute little beyond their usage, word of mouth they share with others, and their demand for more such systems. At OSS portals like SourceForge.net, the domain

of "Games" is the second most popular project category with nearly 42K active projects, or 20% of all projects[1]. These projects develop either OSS-based games, game engines, or game development tools/SDKs, and all of the top 50 projects have each logged more than 1M downloads. So the intersection of games and OSS covers a substantial socio-technical space, as game modding and traditional OSS development are participatory, user-led modes of system development that 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 (Scacchi, 2002, 2004, 2007). Modding and OSS projects are in many ways experiments to prototype alternative visions of what innovative systems might be in the near future, and so both are widely embraced and practiced primarily as a means for learning about new technologies, new system capabilities, new working relationships with potentially unfamiliar teammates from other cultures, and more (cf. Scacchi, 2007).

Consequently, game modding appears to be (a) emerging as a leading method for developing or customizing game software; (b) primarily reliant on the development and use of OSS extensions as the ways and means for game modding; and (c) overlapping a large community of OSS projects that develop computer game software and tools that has had comparatively little study. As such, the research questions that follow then are why do these conditions exist, how have they emerged, and how are they put into practice in different game modding efforts.

This paper seeks to examine what is known so far about game mods and modding practices. The research method in this study is observational and qualitative. It seeks to snapshot and highlight current practices that can be associated with software engineering and game studies, as well as how these practice may be applied in CSS versus OSS game modding. Numerous examples of different game mods and modding practices

10 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage: www.igi-global.com/chapter/modding-open-source-approach-extending/74668

Related Content

Monitoring Social Distancing Using Artificial Intelligence for Fighting COVID-19 Virus Spread
Hashem Alyami, Wael Alosaimi, Moez Krichenand Roobaea Alroobaea (2021). *International Journal of Open Source Software and Processes (pp. 48-63).*
www.irma-international.org/article/monitoring-social-distancing-using-artificial-intelligence-for-fighting-covid-19-virus-spread/286652

Cross-Project Software Refactoring Prediction Using Optimized Deep Learning Neural Network With the Aid of Attribute Selection
Rasmita Panighrahi, Sanjay Kumar Kuanarand Lov Kumar (2022). *International Journal of Open Source Software and Processes (pp. 1-31).*
www.irma-international.org/article/cross-project-software-refactoring-prediction-using-optimized-deep-learning-neural-network-with-the-aid-of-attribute-selection/300756

Improving Logging Prediction on Imbalanced Datasets: A Case Study on Open Source Java Projects
Sangeeta Lal, Neetu Sardanaand Ashish Sureka (2016). *International Journal of Open Source Software and Processes (pp. 43-71).*
www.irma-international.org/article/improving-logging-prediction-on-imbalanced-datasets/181326

Implications of Using Corpus Tools in Primary and Secondary Education
(2020). *Computer Corpora and Open Source Software for Language Learning: Emerging Research and Opportunities (pp. 179-190).*
www.irma-international.org/chapter/implications-of-using-corpus-tools-in-primary-and-secondary-education/256703

The University Cloud Library Model and the Role of the Cloud Librarian
Sangeeta Dhamdhereand Ramdas Lihitkar (2015). *Open Source Technology: Concepts, Methodologies, Tools, and Applications (pp. 308-319).*
www.irma-international.org/chapter/the-university-cloud-library-model-and-the-role-of-the-cloud-librarian/120922