Emulation as a strategy for the preservation of games: the KEEP project

Dan Pinchbeck, David Anderson, Janet Delve, Getaneh Otemu, Antonio Ciuffreda
University of Portsmouth
School of Creative Technologies, Middle Street, Portsmouth, PO1 2DJ
dan.pinchbeck@port.ac.uk

Andreas Lange
ComputerSpieleMuseum
Marchlewskistr. 27 D- 10243 Berlin
lange@computerspielemuseum.de

ABSTRACT
Game preservation is a critical issue for game studies. Access to historic materials forms a vital core to research and this field is no different. However, there are serious challenges to overcome for preservationists in terms of developing a strategic and inclusion programme to retain access to obsolete games. Emulation, as a strategy already applied by major developers and the gaming community, is introduced and the KEEP project, designed to create an open emulation access platform is described.

Author Keywords
Games, preservation, emulation, archiving

WHY GAME PRESERVATION MATTERS TO GAME STUDIES

The preservation of digital games is of vital importance to game studies. As with any other field, a record of the historical development of the medium and access to specific artifacts within this should not be underestimated. This extends beyond classic or important titles, to the vast numbers of less well known or critically lauded games released over the last thirty years.

Preservation of games tends to be piecemeal. National libraries and archives do not currently have systematic strategies for collection and certainly not for preserving access and runtime functionality. Private collectors and fansites have played a major role in providing access but once again, these are not systematic and access remains a problem. This paper splits the issue into two major components: the necessity for creating archives of digital games; and the barriers to successfully preserving access to these.

Software development is rarely carried out from the ground up, that is, without the re-use and adaptation of existing technologies and techniques, and this is particularly evident in the games industry. Dynasties of build engines, middleware and plug-ins form an essential map of the history of the medium. As an example of this, consider the Source engine, used for titles such as Half Life 2 [24]. Source arose from the GoldSrc Engine, itself derived from the original Quake engine [10]. It is thus linked into an engine dynasty with the entire Quake series, dating back to 1996. Quake 4, however, was built using the idTech 4 engine, part of another dynasty which originates in the original Doom engine [11]. Adoption of both engines has been limited in comparison to a third dynasty, the Unreal engine, whose most recent 'children' include Bioshock [1], Unreal Tournament 3 [5], Gears of War [6] and Turok. What makes one engine more frequently adopted than another should be an interesting question for scholars interested in the development of games from a variety of perspectives from development processes, economics and licensing to design, functionality and specific approaches graphics, audio, multiplayer options and artificial intelligence.

To extend this example, now consider an approach a scholar may take to addressing this question. On one hand, a film researcher interested in the influence of A Trip to the Moon [17] could rely on secondary data: interviews, pieces written by other scholars or journalists or critics, examples of derivative material. But it is a highly questionable approach to not actually engage with the primary data itself: to not examine A Trip to the Moon as a piece of media. Likewise, an understanding of the Unreal engine is always going to be limited if the actual objects built using the technology are not examined. Recent publications have argued that the playing of games constitutes an essential aspect of game studies [2], and beyond this there is the under-represented but equally vital process of reverse engineering build tools and component parts to understand how they function. A robust understanding of why and how the Unreal 1.5 engine was selected, adapted and applied, and what impact this process had upon the final object that is Deus Ex [12] becomes extremely difficult without access to this primary data. On one level, verifying secondary data

Breaking New Ground: Innovation in Games, Play, Practice and Theory. Proceedings of DiGRA 2009

is critical to avoid error, on the other, we cannot assume a comprehensive exposition of the object will be available – indeed, this is a completely unrealistic assumption to make.

The requirement to preserve access to both games themselves and their build tools (and raw data) extends far beyond the technology driven research suggested by this. It has been argued, for example, that any cultural interpretation of game elements should rest within the ludic and technological constraints and circumstances that underpin the objects themselves – that there is room for structuralism as well as culturalism in game studies, regardless of the dubious outcomes of the so-called ludology/narratology debates. For example, the visualization of agents in games is impacted by the difficulties in real-time graphical representation. Hard armor is less processor intensive than soft, flowing cloth; a leather catsuit is much easier to deal with than Jedi robes. Games are, after all, products of economy, whether the constraints are monetary or technological. When asking why games look as they do, then, understanding what they are capable of, in system terms, is at least as important as any cultural reading.

Running alongside this is the content-driven evolution of a medium, including archetypal mechanisms of play as well as symbolic and semantic representations of game elements. Searching for an explanation of why Birth of a Nation was shot in black and white and not engaging with the brute fact that only black and white film was available is self-evidently ridiculous; a similar process of understanding historical constraints upon design is critical for game studies. Content and construction are fundamentally interwoven in this medium and in order to properly understand historical objects, we must preserve access to both the primary data and build tools. This is therefore in addition to the power of using historical games as tools in game education. Assigning students the task of developing according to the constraints of an older, simpler system and giving them access to the objects developed in this period to understand how solutions were reached has clear pedagogic power. This should be especially relevant now, with the rise of mobile gaming suddenly undercutting the graphics arms-race and returning us to games that have more in common with Manic Miner [4] than Assassin's Creed [23]. Likewise, in a medium where the shoulders of giants form the basic building blocks of development, understanding how evolution has occurred by assessing historical artifacts allows us to not only trace clear problem/solution pathways, but to question assumed methodologies by identifying their origins and processes of adoption. In short, to move forwards with any aspect of game studies without keeping the preservation of, and access to, historical artifacts is near-sighted, self-defeating and, considered against the core activities of other fields of media, art and cultural studies, palpably absurd.

SPECIFIC ISSUES IN GAME PRESERVATION

Having said that, games are particularly difficult to preserve and it is perhaps no surprise that so little has been achieved when the full complexities of what it means to archive a game are considered, let alone the technical problems with retaining runtime functionality. In this section, we will offer an illustration of just how complex capturing a full game actually is, and offer some indication of the quantity of supplementary data that may be of interest surrounding an archived game, before moving on to how the KEEP project, in particular, is tackling the second problem.

Assuming for a moment, that the core technical issue of platform obsolescence can be bypassed, what exactly does it mean to archive a game such as S.T.A.L.K.E.R.: Shadow of Chernobyl [7]? The game was notoriously buggy when released and six separate patches were subsequently released to fix most major issues and add additional functionality that was omitted in the initial release. On top of this, the game could be purchased in disc format, or digitally downloaded, each of which required separate patches. Localisation meant translation of the substantial text-based dialogue trees. A Collector's Edition box contained not only the standard game manual but additional print such as a Zone Map and Survival Guide, and a DVD containing supplementary images, text and video files. The multiplayer option meant the establishment of servers, both official and unofficial, which contain data about the history of the online aspect of the game, not to mention potential information about how these online games have been played and whether they fit any generalisable pattern of, say, deathmatch behavior (itself an understudied and important question in game studies). Alongside all of this official data, S.T.A.L.K.E.R. has been adapted and altered by the modding community, adding new assets, tweaking and fixing code and, in the case of Kanyhalos' Oblivion Lost [13], subject to major revision. If all this wasn't enough, the community of gamers has also added reviews, discussions, walkthroughs, forum arguments, cheats and hacks, not to mention that the proprietary XRay engine developed for the game has evolved along with the sequels, Clear Sky [8] and the forthcoming Call of Pripyat [9]. This is not only a vast body of data surrounding a single object (which, in itself requires 10Gb of hard disc space to store), but it raises profound questions of what should be prioritized in terms of preservation. For games ported to several platforms the problem increases for each variation on the game. It is necessary, for example, to preserve all versions of the release – digital and DVD-ROM based – in the original form, or final, patched, version? If the patches are deemed important, as they presumably should be for any scholar interested in the shift in development practice towards releasing clearly unfinished games, then how are these to be stored and what relationship should they have to the artifact itself? Archivists are faced with a stark choice: collect and archive everything, including multiple versions of the same object; or make decisions about excluding material strategically to make the process more feasible,

and risk consigning what may one day be important data to the rubbish bin. Finally, there is the question of the XRay engine itself. Unlike many FPS games, which fall into dynasties of engines, GSC Gameworld created a proprietary engine for the game, including features that do not exist in these others, such as the dynamic A-Life engine. The issues with intellectual property will be covered in a later section, but for the moment, it is worth noting that alongside final products and supplementary data, the tools and build data for games is of equal value to future researchers. Understanding, for example, how the X-Ray engine functions; issues and advantages in developing using it compared to other build engines; its use of middleware and plug-ins; its handling of AI and rendering; the relationship between scripted sequences, sandbox design and diegetic and gameplay construction, all have a profound value to scholars as well as future educators and developers. Put another way, being able to access pre-compiled game data enables a far deeper, greater understanding of the game as a media artifact than simply playing or studying the final build. Thus, game preservation should not only aim to capture games and their surrounding data, but, wherever possible, the tools and assets used to create them.

Technically, the problems do not get any easier. S.T.A.L.K.E.R., like any other game, is reliant upon an operating system with the correct system specifications to run. The issue here is self-evident: operating systems, along with hardware components, are superseded and become obsolete. This is as true for consoles as home computers.

Obsolescence is a major issue in the preservation of any digital artifact. Historically, the primary solution has been to migrate such artifacts to current platforms to enable continuing access to them. Migration effectively means altering the code of an object to enable it to be rendered on a non-obsolete platform [25]. However, migration inevitably accelerates the process of bit-rot, or data degradation, meaning that the life-span of migrated objects is generally reduced. Further, migration is highly inefficient, as every time it is required, every object must be migrated individually [21, 22]. Equally, the process of migrating an artifact such as a game is substantially more complicated than a simpler object such as an electronic document, or even audiovisual data. Put simply, migration is of limited value in digital preservation generally, and of extremely limited value in the preservation of games.

EMULATION AND GAMES

The alternative to migration is emulation, meaning that the environment used to run the game in its original format is recreated virtually on a contemporary platform. The focus therefore is on the platform or enabling technologies required the run the object, not on the object itself This is left untouched, which has advantages in terms of not contributing to data degradation. It also means that dealing with obsolescence becomes an issue of creating new emulators for platforms as they become obsolete, so large numbers of objects can be served by more generic emulators, streamlining the process of preserving access.

A number of issues have been put forward as arguments against emulation as a core strategy for preservation. For example, Phelps & Watry [18] have contended that a major block is that emulation prevents searching within documents. Whilst this is a concern, it demonstrates the slant in digital preservation towards textual material and has far less problematic implications for games. The complexity of emulation systems (particularly for users, where lack of technical knowledge may be prohibitive) is a bigger problem. Do if the fact that the overwhelming majority of current emulators rely upon specific platforms which are equally vulnerable to obsolescence, and potential issues with stacking emulators to reach an object. In other words, emulating an environment from which to emulate another environment within which to run an object is theoretically possible but remains largely untested. Finally, as Bearman [3] notes, many of the target environments for emulation are locked within copyrights even after obsolescence. This is particularly true for games console emulators. It is clear that emulation is certainly not a ‘magic bullet’ solution, as Bearman caricatures Rothman’s approach. However, the notion of migrating games, even without considering the general problems with migration, is clearly not feasible. Even the most cursory conversation with developers about the technicalities of porting games would make that explicit.

It is not surprising, therefore, that emulation has been adopted by the games preservation community more widely than for other digital objects, with a large number of solutions already existing in the public domain. The MAME architecture, which enables the emulation of large numbers of arcade games, is well established and well-known [16]. There are a substantial number of console emulators also in existence, although the majority of these rely on hacked BIOS to function and therefore infringe copyright law. Emulation is also used actively by legitimate platform developers: Sony’s PlayStation 3 contains a PSX emulator, as does the PSP, and the Nintendo Wii Store offers access to a wide selection of previous console titles via emulation and the purchase of a bespoke hardware controller. Ensuring backwards compatibility through emulation makes sound financial sense as it extends the shelf-life of intellectual property and it seems likely that this policy will continue.

The biggest problem, however, with all current emulators is their own obsolescence, as each is built for a specific platform and thus vulnerable to this being superseded. It is this issue that the KEEP project aims to tackle directly.

THE KEEP ARCHITECTURE

In January 2009, the KEEP (Keep Emulation Environments Portable) project was launched. Funded through the European Commission’s Framework 7 program, KEEP is being developed by an international consortium: the

national libraries of France, Germany and the Netherlands; Tessella (UK/NL) and Joguin SAS (FR), software developers specializing in preservation; project consultants CrossCzech (CZ); the ComputerspieleMuseum (DK); the University of Portsmouth (UK) and the European Game Developers Federation. The first phase of the project, lasting until 2012, aims to develop a prototype of an emulation access platform to enhance the preservation of digital objects, with a particular focus on digital games.

Unlike current emulator systems, KEEP is not built upon a specific platform technology, but a virtual machine. This follows the conceptualization of such a system by Rothenberg [22] and Lorie [14]. The OLONYS system, developed by Joguin SAS, is a series of virtual machines stacked in order of complexity that will interface with, and support a modular emulation framework [15]. Thus, at root, KEEP is far more future proof than current solutions. It also has the advantage of offering multiple emulation solutions to any given artifact within a single, user-friendly interface, allowing both bespoke manual configuration of the emulation process and a more automated and simple means of accessing artifacts for users not requiring this. On one hand then, KEEP benefits from less reliance upon any given platform and a modular architecture that enables independently developed emulators to function within its framework (meaning that existing components can be integrated). On the other, it enables archivists and researchers to bypass the traditionally complicated process of installing and running emulation software.

Alongside the emulation access platform, KEEP will also develop a transfer tools framework to enable new objects to be integrated with the system. The existing archives of the Bibliothèque nationale de France, Koninklijke Bibliotheek, Deutsche Nationalbibliothek and Computerspiele Museum will form the initial core of the available KEEP archive, but this framework will establish a means for new objects to be added. Further, research is currently being carried out to supplement current metadata standards for archiving with emulation metadata to ensure high compatibility with existing international archives. Part of this process is evaluating and enhancing existing metadata systems to ensure maximum compatibility with games.

KEEP's focus is on retention on existing emulation work and enabling archives and users to transfer objects to the KEEP architecture as seamlessly as is possible. Whilst conceptually and technologically advanced, it is, at root, a deeply pragmatic solution to an extremely difficult problem.

If successful, the impact of KEEP upon game studies will be profound. Although initially limited to localized access in three European countries, a second phase of the project (once proof of concept is established) may be to roll the architecture out to other archives internationally, and to explore potential public release of the system so individual users can transfer their obsolete media into KKEP and retain access. This is one means of circumventing the normal copyright problems, as KEEP therefore functions as an enabling technology for 'home archiving' to supplement access to archived artifacts held in national storage.

In terms of both research and education, a modular and open emulation platform goes some way to addressing the difficulties of archiving the large bodies of material surrounding most games. For example, the Dioscuri emulator, which emulates x86 hardware and can run Windows98 [26], has the potential to enable game patches or build engines to be run within this native environment. In other words, emulating hardware to provide native access to the functionalities of obsolete operating systems provides access to many of the other tools and data surrounding a game, rather than simply providing access to the final object itself. Equally, a modular architecture means that a variety of emulators, each with particular strengths and functionalities may be selected to access a particular object. So a user seeking to simply emulate S.T.A.L.K.E.R. to access the game may opt for an emulator that sacrifices additional O/S functionality for increased performance, whereas another looking specifically at the codebase behind the game may choose to dispose of advanced graphics emulation in favour of alternate functionality.

There is a caveat in all of this, of course. The emulators themselves need to be written, and the KEEP consortium is working closely with the existing emulator community to try and maintain a high level of adaptability with the emulation access platform and current emulators. The metadata extensions require a careful balancing act between what is both pragmatic and compatible for archives to implement, and the high-level information required for the modular emulation architecture to function intelligently. There are outstanding issues with copyright protection in regard to games that a legal study is exploring. Nor does KEEP directly solve the issue of the large potential body of supplementary information surrounding each game. However, it is the first systematic, large-scale attempt to solve the technical problems with access retention for this medium and, as such, presents a major potential benefit to the games research and education community.

CONCLUSION

Game studies requires systematic archiving of historical titles. Otherwise it runs a serious risk of data loss. Personal collections, fan archives such as Abandonia or Home of the Underdogs and the rerelease of old IP through digital distribution such as Playstation Store cannot and should not be counted on to ensure access is protected for obsolete titles. The preservation of games is a very difficult issue, partially due to the large bodies of secondary artifacts surrounding each release; partially due to big fixes and patches, and partially due to the technological challenges of ensuring run-time functionality. Emulation is the only real solution to this challenge, but an open access architecture based upon a virtual machine is the only means of future

proofing these emulators from the same cycle of obsolescence as faces the original media. Hardware emulation and a modular framework not only enables original titles to be run, but offers access to codebases, build engines, middleware and game assets, all of which, it has been argued, are of potential equal value to future scholars.

The KEEP project, like emulation in general, is not a 'magic bullet' solution. It cannot ease the secondary artifact burden, nor can it resolve the problems with the continuation of copyright post-obsolescence. What is does offer, however, is the best current solution to retaining access to obsolete games into the future. For this reason, it should be of interest to any researchers interested in the past, or the future, of our medium and field.

REFERENCES

1. 2K Boston + 2K Australia. *Bioshock*. 2K Games: PC. 2007

2. Aarseth, E. Playing Research: Methodological approaches to game analysis, *Melbourne DAC - the 5th International Digital Arts and Culture Conference* (2003).

3. Bearman, D. Reality and Chimeras in the Preservation of Electronic Records. In *D-Lib Magazine* vol 5 no 4, (April 1999)

4. Bug-Byte. *Manic Miner*. Bug-Byte: ZX Spectrum, 1983

5. Epic Games. *Unreal Tournament 3*. Midway: PC, 2007

6. Epic Games. *Gears of War*. Microsoft Game Studios: Xbox360, 2006

7. GSC Game World. *S.T.A.L.K.E.R.: Shadow of Chernobyl*. THQ/GSC: PC, 2007

8. GSC Game World. *S.T.A.L.K.E.R.: Clear Sky*. Deep Silver: PC, 2008

9. GSC Game World. *ST.A.L.K.E.R.: Call of Pripyat*. PC, Q42009

10. id Software. *Quake*. id Software: PC, 1996

11. id Software. *Doom 3*. Activision: PC, 2003

12. Ion Storm. *Deus Ex*. Eidos Interactive: PC, 2000

13. Kanyhalos. *Oblivion Lost*. Uncommercial modification for S.T.A.L.K.E.R.: Shadow of Chernobyl. PC, 2007. Available from www.oblivion-lost.net

14. Lorie, R.A. *Long-term archiving of digital information.* IBM Research report, IBM Almaden Research Center, San Jose, Almaden, 2000

15. Joguin, V. Emulating emulators for long-term digital objects preservation: the need for a universal machine. *Emulation Expert Meeting 2006*, (The Hague, The Netherlands).

16. *MAME: Multiple Arcade Machine Emulator* (1997-2009). Information available at http://mamedev.org. Retreived 21/7/2009.

17. Méliès, G. *A Trip to the Moon*. 1902

18. Phelps, T.A., & Watry, P. A no-compromises architecture for digital document preservation. *ECDL*, 2005: 266-277.

19. Propaganda Games. *Turok*. Touchstone: PC, 2008

20. Raven Software. *Quake 4*. Activision: PC, 2004

21. Rothenberg, J. *Avoiding Technological Quicksand: Finding a Viable Technical Foundation for Digital Preservation*. A Report to the Council on Library and Information Resources. Washington: Council on Library and Information Resources, 1999

22. Rothenberg, J. Using Emulation to Preserve Digital Documents. The Netherlands: Koninklijke Bibliotheek, 2000

23. Ubisoft Montreal. *Assassin's Creed*. Ubisoft: PC, 2008

24. Valve. *Half Life 2*. VU Games: PC, 2004

25. van der Hoeven, J., vanWijngaarden, H. Modular emulation as a long-term preservation strategy for digital objects. In: *Proceedings of the 5th International Web Archiving Workshop (IWAW05) held in conjunction with the 8th European Conference on Research and Advanced Technologies for Digital Libraries* (Vienna, Austria, September 2005

26. van der Hoeven, J.; Lohman, B.; and Verdegem, R. Emulation for digital preservation in practice: The results. *International Journal of Digital Curation* vol.2, no. 2 (2007), pp123– 132.