Your Title Here

COMP160 - Software Engineering Essay

1608305

March 20, 2017

Please include an abstract of at most 100 words (these do not count towards your word count).

1 Introduction

Recently in the games industry there have been disconcert over games that have been ported from consoles to PC. The console version works as expected but the PC versions suffer big performance losses and restrictions such as frame rate caps. This paper will be looking into why this is happening and what actions developers can take to stop this happening with future ports. The research conducted for this paper will be looking into techniques used for producing portable code and the difficulties with developing it.

2 In computing

Porting software to work on different hardware is common practice in computing. The process is used in many fields of computing one of the biggest being robotics. [1][2] During the design process portability and maintainability should be a big factor to be considered. [3][4] Modular designs and flexible architecture techniques can be used to make porting the software later much more painless. [5] Other techniques like abstracting the functionality and then adding to it has proven to be one way of making highly portable software. [6] If the program has been made using an abstract model of computation it can then be mapped to different systems by attaching annotations. [7] This is especially useful with multicore processor systems as it helps to get the most out of the cores.

Some common pitfalls with portable software design is using language extensions, although these may work on one system they are painfully nonportable.[8] When developing for one system in mind then language extensions can make life easier but if considering porting then avoiding any language extensions will help in the long term.

3 In games industry

In the games industry games are often developed for consoles first as that is usually where the biggest market is. This means the games are optimized to work best on these specific systems. Many games have been designed with portability in mind so porting them isnt such a big task. But when a development company initially only had plans for a console release then porting to a PC can become quite the task. When developing specifically for console programmers can be lazy and use work arounds that work on the hardware they are developing for but not for others.

Using tick update every frame and locking the frames to a certain amount is an example of this. it works fine on the console but when you run the same software on a pc that is capable of much higher frame-rates it must also lock at the same amount or it will break the game.

Hardware goes out of date relatively fast but developers will stay developing for the hardware for a long time. [4] This is especially true for the games industry where the hardware in consoles becomes outdated rapidly but developers are still developing for them ten years after their release. The old technology means that sometimes developers will cut corners for their games to have favourable performance. Architectural differences often require different programming styles and performance optimization techniques. [9] Using programming styles that differ from normal programming standards can make software substantially harder to port. [10] Keeping to normal programming standards and having maintainable code is key when thinking about porting a game.

4 Conclusion

The developers that have had the most trouble with porting their games are the ones that initially designed their games without portability in mind. Games that have been successful on one platform but ported to another as an afterthought are the ones that have the most problems. First time developing for a different platform proves to cause a lot of developers problems for these reasons.

When a game hasnt been designed with porting in mind developers can still produce a good port but it will be likely to take much longer. The problem with some developers is that they have not given enough time or funding for the port to be made to a good quality.

Steams new refund system should help eliminate bad ports with users now able to refund games easily when they are not up to standard making it harder for developers to make money from poor quality products.

References

[1] F. Weisshardt, J. Kett, T. d. Freitas Oliveira Araujo, . A. Bubeck, and A. Verl, "Enhancing software portability with a testing and evaluation platform," in *ISR/Robotik 2014; 41st International Symposium on Robotics*, June 2014, pp. 1–6.

[2] R. Smith, G. Smith, and A. Wardani, "Software reuse in robotics: Enabling portability in the face of diversity," in *IEEE Conference on Robotics, Automation and Mechatronics, 2004.*, vol. 2, Dec 2004, pp. 933–938 vol.2.

[3] V. K. Myalapalli, J. K. Myalapalli, and P. R. Savarapu, "High performance c programming," in *2015 International Conference on Pervasive Computing (ICPC)*, Jan 2015, pp. 1–6.

[4] B. Sedov, A. Syschikov, and V. Ivanova, "Technology and design tools for portable software development for embedded systems," in *Proceedings of 16th Conference of Open Innovations Association FRUCT*, Oct 2014, pp. 86–93.

[5] B. DeAbren, "Test software design techniques for reuse and portability," in *2000 IEEE Autotestcon Proceedings. IEEE Systems Readiness Technology Conference. Future Sustainment for Military Aerospace (Cat. No.00CH37057)*, 2000, pp. 334–338.

[6] D. L. Brittain, "Portability of interactive graphics software," *IEEE*

Computer Graphics and Applications, vol. 10, no. 4, pp. 70–75, July 1990.

[7] A. Colbrook, W. E. Weihl, E. A. Brewer, C. N. Dellarocas, W. C. Hsieh, A. D. Joseph, C. A. Waldspurger, and P. Wang, “Portable software for multiprocessor systems,” *Computing Control Engineering Journal*, vol. 3, no. 6, pp. 275–281, Nov 1992.

[8] P. F. Dubois, T. Epperly, and G. Kumfert, “Why johnny can’t build [portable scientific software],” *Computing in Science Engineering*, vol. 5, no. 5, pp. 83–88, Sept 2003.

[9] H. Su, N. Wu, M. Wen, C. Zhang, and X. Cai, “On the gpu-cpu performance portability of opencl for 3d stencil computations,” in *2013 International Conference on Parallel and Distributed Systems*, Dec 2013, pp. 78–85.

[10] D. Hardy and J. L. Sancey, “Specification and-production techniques for portable software components,” in *International Symposium on Switching*, vol. 2, May 1990, pp. 183–189.