Panel on Open Source Software for RIS

Pablo Figueroa*
Universidad de los Andes, Colombia

ABSTRACT
We organize at SEARIS a panel about the use of open source software (OSS) and in particular open source game engines for research in Realtime Interactive Systems (RIS). Panelists identified two types of researchers, application oriented and development oriented, and they agree on the importance of OSS for development oriented researchers and for the future research on RIS, so novel ideas could be built on top of existing solutions. This paper mentions some of the key ideas of this panel.

Index Terms: K.5.1 [Information Systems]: Multimedia Information Systems—Artificial, augmented, and virtual realities;

1 INTRODUCTION
Research in Realtime Interactive Systems (RIS) is a daunting task, given the amount of issues that are involved in any complex system, and the high expectations from researchers and public in general due to the quality that current RIS have, specially video games. This year, at the Software Engineering and Architectures for Realtime Interactive Systems Workshop (SEARIS), we had a Panel on Open Source Software for RIS as a way to understand how important is the concept of Open Source in RIS research, current trends, and future areas of research. Selected experts in the IEEE VR and Open Source communities accepted our invitation as panelists, in order to present and discuss their points of view. Carolina Cruz-Neira, from University of Louisiana at Lafayette, presented her experience of more than 20 years in the field of VR Development, in particular with VR Juggler [1], an open source, VR application development framework. Perry McDowell, from the Naval Postgraduate School and head of the Delta3D project [2], presented the advantages of open source game engines for the development of VR simulators. Luciano Soares, Project Manager at TecGraf (PUC-Rio), presented his experience as a developer of JINX [3], an open source VR framework, and as a current developer in the Oil & Gas Industry. Evan Suma, Senior Research Associate at the Institute for Creative Technologies of the University of Southern California, presented his experience as a VR researcher and developer of the widely used FAAST toolkit [4].

Our panelists had the following questions as a motivation for our discussion:

- How to deal with obsolecense in OSS (Or what to do when your favorite piece of OSS is no longer supported?)
- What are your favorite OSS based solutions, and why are they special?

2 MAIN THOUGHTS
Researchers in the VR community have used and created several open source tools, some more successful than others. There is therefore a widespread experience in the use of open source technologies, yet there is a lack of standard tools that the community as a whole use for their work. The following are some of the main ideas that panelists discussed, either in pro or against OSS ¹

2.1 It is Difficult to Make Money from VR Software
Several commercial VR tools have died in the past years for several reasons, and there are very few survivors. One reason may be the relatively small size of our community. In order to get profit from a piece of software, it is necessary to sell a large amount of copies or to have an expensive price per copy. However, since our community is small, it is difficult to support such developments.

Although OSS does not offer a real alternative for this issue, since it is also difficult to survive by offering services in a small community, it is at least an alternative that allows future developments to be based on existing solutions and therefore facilitate advances in this field.

2.2 Most of the Time You Don’t Modify OSS
Most researchers only use OSS technology and do not modify source code in order to accomplish their goals. For example, users studies in VR and integration solutions do not usually require modifications to source code in existing tools, just the use of their standard functionality. In these cases, the main potential of OSS is not used, and researchers are faced to a complexity in the infrastructure they use that they do not need.

While this may be a deficiency in many current OSS packages, it is only based on the fact that most OSS tools do not include easy installation and usage processes, in order to facilitate their use. Instead, researchers are faced with a complex interface that allows them to do many things they do not use. Commercial software seem in this case easier to use and more user friendly.

2.3 The Field is Young, not Settled
Although RIS applications have existed for more than 40 years, there is still a high level of innovation in topics of research. There are research topics that did not exist 5 years ago and nowadays are very important, so it is difficult to have a common set of tools that could support research over time, no matter if it is open source or commercial.

This is indeed a very difficult problem in our field, since it is difficult to have a piece of software that can both support future requirements and facilitate changes. OSS have a slight advantage over commercial software, since developers have access to source code in case of changes, but as we have seen few researchers make use of this potential.

¹As a disclaimer, there are several opinions of my own in the following paragraphs, which do not represent or compromise our panelists’ positions.
2.4 There is no More Profit in Vanilla

Years in advances in VR and Game development have created common solutions to typical problems and an expected state of the practice. Infrastructures for device handling, real time and realistic render engines, and physics libraries are some examples of existing solutions to particular problems in RIS, which can be integrated to new solutions in order to provide an expected level of behavior. These solutions become the vanilla ice cream, the expected set of functionality in any solution, which define the minimum set of functionality.

Developers expect to count with the vanilla layer as foundation for their solutions. OSS solutions can offer most basic foundations, but they are not usually integrated, so developers should do such an extra work (There are some exemptions, such as Delta3D). This vanilla layer is usually easier to use in commercial products, such as Unity, and therefore developers could prefer such solutions. This is not really an inherent drawback on OSS, but instead a missing feature that should be addressed. However, OSS offers such vanilla layer at no cost, in order to create solutions with extended functionality.

2.5 Application Developer vs. System Developer

Panelists identified 2 roles that VR researchers could play: application oriented people, interested in application development that could be evaluated by users, and system oriented people, interested in the creation of new VR systems, hardware, or software tools. Application oriented people usually find support in high level development tools for their work, whereas system oriented people usually require functionality beyond the capabilities of current tools.

OSS tools can provide an important support for system developers, since it is possible to modify them as required. However, OSS tools do not usually provide an important advantage to application developers, since these people do not require to modify functionality, but instead to build on top of existing functionality. Moreover, application developers usually prefer tools with large communities, easy to use, and well supported, which may not be common features in OSS solutions.

2.6 Lifespan of VR Applications is 0

A major problem with current practices in VR development is life span of applications. It is difficult to keep over time a working VR application, usually due to the complex setup and execution process of such applications which is only known by its authors. It is also extremely difficult to share applications between VR installations and labs, due to difficulties in sharing setup knowledge and in replicating the exact hardware and software platform.

In this regard, OSS has a disadvantage over commercial software: installation, setup, and execution of OSS solutions are usually more complex than the same processes in their commercial counterparts. Commercial products should be easy to install and use, on top of all other requirements that final users may have, so they have a chance to be sold. On the other hand, OSS solutions usually concentrate on functionality, not ease of use. They do not usually share installation and setup processes, and in VR solutions that means expertise in several tools in order to integrate a solution. Although commercial products may create similar problems when they have upgrades, OSS solutions have to do a major effort in user friendliness and resilience over time.

2.7 Not Invented Here Syndrome

System Developers in the RIS community tend to create their own tools and solutions, and not reuse code from third parties. This approach is usually based on the assumption that it will be more productive in the long run to have control on the entire solution than learning how to modify an existing software, due to either unexpected shortcomings, lack of support, or suboptimal code in an existing piece of software. The drawback of this approach is the never ending creation of new solutions, with lots of already existing functionality and little innovation.

Commercial solutions seem to have an advantage in this regard: since RIS researchers do not have control over the source code of a commercial solution, they tend to expect more from these tools, bend to the specific way to develop in such tools, and reuse such solutions as they are. Since OSS solutions are usually more difficult to install and since researchers can see code, with all its complexity and possible shortcomings, it is easier to think that they could do better in an in-house solution.

2.8 VR Software is too Complex

VR applications comprise multiple issues at once, for example graphics, networking, synchronization, physics, input and output devices at different update rates. A VR developer should take all this into account in any development, which is a daunting task, and also organize the solution in an understandable way.

Commercial products have a slight advantage over OSS solutions in this regard, since by definition they hide information inside their implementation, and therefore final developers do not have to handle all that information at once and at the same level of abstraction. Although this is not an intrinsic issue with OSS, it is necessary to find out ways to hide complexity in OSS for RIS applications.

2.9 Most Productive in the Least Amount of Time

VR researchers and in particular Application Developers would like to develop a solution in the least amount of time possible. For that reason, they pick tools and environments that promise more productivity, with no considerations if these solutions are open source or not.

Unfortunately, OSS solutions are not usually considered very time efficient, due to more extensive installations procedures and learning curves. Moreover, there are some common tasks that commercial software tend to facilitate, and these functionality accelerate the most traditional developments. Therefore, developers tend to classify commercial solutions as more productive.

2.10 Unity is the Flavor of the Week

There are several researchers in our community that are using game engines for their research, and in particular Unity [5]. Unity is a relatively new engine, with a very aggressive commercial strategy: everyone can have a free version for development, and there are other versions for extended functionality. It is arguably the most popular game engine nowadays, with support for several platforms.

Despite its popularity and accessibility, Unity is still a closed solution, which may not be available in the future. In this scenario, all developments on top of Unity could be left non-operational, which may affect their lifespan.

2.11 Interactive Technologies are Created by Non-Technical People

Several VR installations are now used by non-technical people, interested in the development of artistic setups. For these type of developers, it is necessary to rely on easy to use development environments, which can be used both in standalone setups and in the special VR equipment. Moreover, it is important to count with learning resources, so non-technical people can learn what is necessary for their developments with minimum support from staff.

Again, OSS solutions are in disadvantage over more popular commercial solutions. First of all, non-technical people are not interested or technically capable of changing source code, so they do not need or use such functionality. Second, more widespread development environments are commercial, so non-technical people will prefer them because their support, learning resources, and larger communities.
2.12 Other Issues

There is a widespread necessity for easier to use tools, due to the necessity to handle increasing levels of complexity in RIS development. Although there are some attempts in OSS to facilitate development, such as the editor in Delta3D, it is necessary to increase efforts in this direction.

It is also an issue the amount of resources that are invested in game technology in our days. Although it is possible to use game technologies for several RIS developments, it is also difficult to accommodate requirements to such tools. It is also very difficult to create compelling RIS tools that show the same level of quality and capabilities than tools for game development.

Finally, it was again noted that VR technology often lacks a clear separation between interface and application. The lack of separation and the tight coupling of the interface with the user makes it difficult to create portable applications and to use common technologies, which may be resilient to novel interface technologies and graphics, to name just two elements in RIS applications. Although it could be easier to incorporate such changes in OSS than in commercial solutions, this issue still affects the lifespan of RIS solutions, and therefore the requirements for OSS tools.

3 Conclusions

Open source technologies offer some advantages for the development of RIS applications and research in RIS, although they also have some drawbacks. This panel have identified some key points in the use of OSS for RIS research, which may be useful for future directions of research.

Acknowledgements

I would like to thank the panelists, the participants in our panel, and the SEARIS organization. Particular thanks to Roland Blach, for his useful comments.

References