Martin Skrodzki

RIKEN iTHEMS, Wako, Japan

Corresponding author contact: mail@ms-math-computer.science

Online Geometry Visualization Frameworks – A Case Study

Motivation

The *JavaView* visualization framework was designed at the end of the 1990s as a software that provides—among other services—easy, interactive geometry visualizations on web pages. Subsequently, it was widely used in geometry groups around the globe. However, as *JavaView*'s easy web exports was based on *Java Applets*, the deprecation of this technology disabled one main functionality of the software. Starting from this case study, the work [1] identifies general elements in the development of a geometry software framework. This poster summarizes the main findings.

Goals for Geometry Visualization Software

Interactivity:
Provide interactive, online visualization of mathematical content.

Accessibility: Provide a system-independent framework that takes as much technicality away from the content creators as possible.

Communication: Introduce a unified file format for easy exchange of research and visualization data.

Up-to-dateness: Keep the software up to the available hardware as well as to changes in the underlying programming language.

Environmental Limitations

- A given setup can only achieve a level of **security** according to the weakest link in its chain.
- The **programming language** of the software provides certain functionalities, but also comes with constraints.
- **Portability** is a factor that is also affected by the rise of new architectures.
- **Performance** is mostly a question of the desired use case.
- Similarly, the ease of use of a software comes with its **user base**.

Frequent Questions in the Development of Geometry Software identified in the Case Study

How to react when a main use case of an application is put to the test?
A frequent and thorough reorientation of the software project can be necessary to ensure that new goals and target areas are identified and focused on.

What aspects of software can be maintained by container technology?
It is important to provide an executable environment for the computations, even if the underlying software setup changes. By using available techniques such as *Docker* or *VirtualBox*, it is possible to provide an out-of-the-box running environment even for outdated software.

How does the choice of the programming language affect a software framework?
The underlying programming language evolves and corresponding problems and the necessity for adjustment will occur. The translation into another language can be a valid option to handle these challenges. Possibly, other features of the original language be employed to circumvent occurring problems.

What other basic building blocks of a software framework are subject to change?
In particular a visualization framework has to cope with more than changes in the programming language and related software components—like the operating system. Missing reaction to these changes easily results in frustration on the side of the users who then migrate to other systems that better satisfy their demands.

How does a closed- or open-source policy affect the development of a framework?
The availability of software as open-source clearly affects the number of available developers, collaborators, and users willing to interact with the software. However, a closed-source software with a loose release cycle allows for closer monitoring of the developed functionality and can provide better quality and more homogeneous code.

How to adjust to competitors coming into the field?
As new developments arise in a field, new niches open and new competitors rush in to fill these. Established visualization frameworks thus have to cope with other software coming in and competing for the same user base.

References

[1] Martin Skrodzki. *How the Deprecation of Java Applets Affected Online Visualization Frameworks – A Case Study*. In: VisGap – The Gap between Visualization Research and Visualization Software, edited by C. Gillmann, M. Krone, G. Reina, and T. Wischgoll, *The Eurographics Association*, 2020.