Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 75 (2015) 257 – 266

Procedia Computer Science

2015 International Conference on Virtual and Augmented Reality in Education

Adopting a Game Engine for Large, High-Resolution Displays

Anton Sigitov[a]*, David Scherfgen[a], André Hinkenjann[a], Oliver Staadt[b]

[a]*Bonn-Rhein-Sieg University of Applied Sciences, Grantham-Allee 20, 53757 Sankt Augustin, Germany*
[b]*University of Rostock, Institute of Computer Science, Albert-Einstein-Str. 22, 18051 Rostock, Germany*

Abstract

The steadily decreasing prices of display technologies and computer graphics hardware contribute to the increasing popularity of multiple-display environments, like large, high-resolution displays. It is therefore necessary that educational organizations give the new generation of computer scientists an opportunity to become familiar with this kind of technology. However, there is a lack of tools that allow for getting started easily. Existing frameworks and libraries that provide support for multi-display rendering are often complex in understanding, configuration and extension. This is critical especially in educational context where the time that students have for their projects is limited and quite short. These tools are also rather known and used in research communities only, thus providing less benefit for future non-scientists. In this work we present an extension for the Unity game engine. The extension allows – with a small overhead – for implementation of applications that are apt to run on both single-display and multi-display systems. It takes care of the most common issues in the context of distributed and multi-display rendering like frame, camera and animation synchronization, thus reducing and simplifying the first steps into the topic. In conjunction with Unity, which significantly simplifies the creation of different kinds of virtual environments, the extension affords students to build mock-up virtual reality applications for large, high-resolution displays, and to implement and evaluate new interaction techniques and metaphors and visualization concepts. Unity itself, in our experience, is very popular among computer graphics students and therefore familiar to most of them. It is also often employed in projects of both research institutions and commercial organizations; so learning it will provide students with qualification in high demand.

Peer-review under responsibility of organizing committee of the 2015 International Conference on Virtual and Augmented Reality in Education (VARE 2015)

Keywords: large-high-resolution displays; rapid prototyping tool; Unity; tools for education

* Anton Sigitov. Tel.: +49-2241-865-266; fax: +49-2241-865-8266.
E-mail address: Anton.Sigitov@h-brs.de

Peer-review under responsibility of organizing committee of the 2015 International Conference on Virtual and Augmented Reality in Education (VARE 2015)
doi:10.1016/j.procs.2015.12.246

1. Introduction

The significant progress in the areas of display and graphics hardware technologies are reflected in a price decrease for visual output devices and graphics cards as well as in an enormous efficiency increase of the latter. Thus it became possible to build affordable large, high-resolution displays (LHRD). In general, LHRDs differ from mainstream desktop displays in two aspects: physical size and resolution. They can be defined as a combined visual output perceived as a single, continuous visual space that provides significantly more pixels and is distinctly larger by comparison with a normal display. LHRDs are usually built from an array of projectors or LCD displays. Both technologies have their advantages and disadvantages. For example, the individual tiles of an LCD-based display are disjoint because of bezels. This results in a discontinuous or distorted image output. Opposed to LCD-based displays, projector-based displays don't suffer from this problem. Though, they have to struggle with a colour and brightness inconsistency as well as permanently changing alignment, which requires complex and frequent calibration. The main properties of LHRDs are[1]: size, pixel density, resolution, brightness, contrast, viewing angle, bezels, display technology, and form factor. These properties are also applicable for normal desktop displays. In case of LHRDs they vary stronger, though. The application field of LHRDs is very broad. They are used for visualization of common office and multimedia applications, thus foster collaborative work. They enable the visualization of complex datasets, affording a better insight into complex relationships of data entities. Moreover, they enable a full size visualization of industrial constructions and machines, therefore ensure better spatial impression. The typical applications of LHRDs are: command centrals, vehicle design, geospatial imagery, scientific visualization collaboration and tele-immersion, education and training, immersive applications, and public information displays. The survey by Ni et al.[2] gives an extensive overview on LHRD technologies, applications, and challenges.

With regard to steadily growing popularity of LHRDs in different domains, we believe it to be of great importance to provide an opportunity for upcoming generations of computer scientists to become familiar with that type of displays, so they can gain an essential competence for their future career. Developing applications for LHRDs, students will recognize that common, desktop-oriented interaction devices and techniques are not suitable for this kind of system. Thus, they will be forced to experiment with design, and research for better solutions. However, in order to create this opportunity, specific tools have to emerge which will allow a quick and easy start on application development for LHRDs. These tools are necessary in the first place because of the limited time students have at their disposal during semester projects. Most of currently available solutions are too complex in configuration and extension, and offer scarce support. Thus, students have to spend a lot of time learning how to utilize them, before being able to proceed with their ideas. Moreover, none of those tools provide any modules for rapid prototyping, like a visual scene editor, animation libraries, input device libraries, graphical user interface libraries, etc. Therefore, students are often forced to combine multiple libraries which stem from different developers, and which do not necessarily work together flawlessly. Consequently, many projects remain unfinished, and interest in the technology decreases due to negative experience.

Establishing a common ground on the basis of a well-known framework, which provides better support for rapid prototyping, better support on the part of the software developer, and a large developer community will contribute to fluent project progress. As a result, the students' experience may become improved in a positive way, better outcomes may be achieved, and the technology may be promoted. Moreover, having a unified basis will result in less heterogeneous source code throughout the projects, making them more appropriate for communication and advancement.

In this paper we present an extension for the Unity game engine[3] that allows to run Unity applications on LHRDs. Our experience shows that many media informatics students are intimate with Unity and have utilized it for their hobby projects beforehand. The game engine itself provides a lot of useful tools for swift content creation. Thus, we think it could be the common ground we are looking for. Although Unity's primary focus lies on digital games, it is often used for the development of applications in different other domains, like architecture, engineering and construction, simulation in the medical and security fields, serious gaming, and so on. Currently, we intend to utilize Unity for rapid prototyping only, in order to evaluate ideas on interaction and visualization concepts. We hope to produce completed software products for LHRDs later on, though.

The paper is organized as follows: First, we present related work, which contains information on distributed rendering frameworks. Then, in section 3, we describe our large, high-resolution display, called HORNET, which