Student Project – Portable Real-Time 3D Engine

Christoph Müller and Fabian Gärtner

Furtwangen University, Germany

Abstract

FUSEE ("Furtwangen University Simulation and Entertainment Engine") is a student project at Furtwangen University to develop a cross-platform real-time 3D engine from scratch. While about 20 students have been involved in the project's development so far, several more students have used the engine in class or during their own research projects while learning how a contemporary engine works behind the scenes. Furthermore, students can learn how programming languages such as C# and new trends such as HTML5 can be used for writing cross-platform applications. This paper describes the engine and its core functionality as well as the benefits the project provides for education in real-time 3D software development compared to classes where other engines are used.

Categories and Subject Descriptors (according to ACM CCS): K.3.2 [Computers and Education]: Computer and Information Science Education—Computer science education

1. Introduction

Computer science classes at Furtwangen University provide basic and advanced knowledge in programming and graphical computing. To improve the students' comprehension and motivation in this area, practical projects such as writing simple video games or 3D applications are part of various courses. Unfortunately, many of the available real-time 3D engines are not suitable for undergraduate education because they are either written in a complex programming language such as C++ or are not free of charge and therefore not available to students. However, there are some exceptions to the rule such as the proprietary Unity engine [Uni]. With Unity, students can easily get in touch with a contemporary 3D engine by using a simple graphical interface and easy-to-learn programming languages such as C#. This allows for teaching various basic elements in computer science, for example, hierarchical scene management, animations, or shader programming. Unfortunately, Unity does not require an understanding of important aspects in graphical computing such as affine transformations or procedural generation of mesh geometry. Furthermore, there is no way of gaining insight into the rendering process as the engine is closed source and everything is done automatically behind the scenes. This results in application of more traditional education techniques where those subjects are explained in theory and this means a lower level of comprehension. With this in mind, a new educational and scientific student project called "FUSEE" was founded at Furtwangen University in 2012.

2. The Project FUSEE

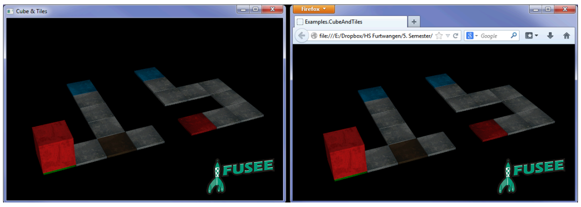

Figure 1: *The same FUSEE example running on Microsoft Windows 7 and in Mozilla Firefox 26 without plugins.*

At Furtwangen University, a group of eight undergraduate students in Computer Science in Media started working on the FUSEE engine during a one-year research project. The idea behind FUSEE was to create an open-source 3D engine to prove that state-of-the-art programming languages and techniques can be applied to cross-platform game programming. C# was chosen as the main programming language because it is considered to be learnt easily by people with basic knowledge in computer science. The students implemented basic and advanced features, such as scene and shader management, lighting, keyframe animations, sound and network functionality, and text rendering. In addition, they created simple examples which showcase the functionality of FUSEE, such as a puzzle game (see Figure 1) or a visualization of the solar system. The source code was published on a website [Fur] along with several tutorials.

DOI: 10.2312/eged.20141030

2.1. Cross-Platform Implementation

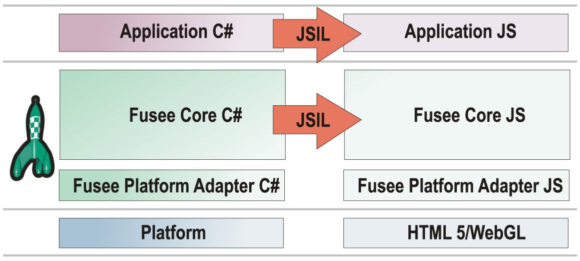

Figure 2: *FUSEE's workflow: Only specific parts are cross-compiled from C# into JavaScript with JSIL while others are manual written implementations.*

Similar projects such as the "Simple Academic Game Engine" [Par] do not allow for writing applications which run on more than one operating system. In contrast, the students' main focus while developing FUSEE was always on cross-platform compatibility. Therefore, the students avoided using libraries that are available on specific operating systems or platforms only. This allows FUSEE applications to run on all major operating systems (i.e. Windows, Linux and Mac OS X) as well as on mobile devices. Another objective was to enable FUSEE applications to run natively in web browsers (see Figure 1). While several other engines require a plug-in to display content in a web browser, the students preferred using modern techniques such as HTML5, JavaScript, and WebGL for this purpose. This required a clean separation between cross-platform and platform-specific parts to allow for cross-compiling huge parts of the engine into JavaScript. For FUSEE, the students decided to use JSIL [Gad], a free open-source cross compiler for translating CIL byte code (i.e. .NET applications) into JavaScript.

However, important parts of a real-time 3D engine are platform-dependent implementations, for example, the access to OpenGL or DirectX as the underlying graphics library – those components cannot be cross-compiled. This problem was solved in FUSEE by exposing facade objects and classes to the user/application code. These objects delegate functionality requests to platform-dependent implementation objects. An anonymous factory pattern is used for decoupling the facade objects from the platform-dependent code, thus hiding the platform-dependent objects from the facade objects. In FUSEE, the cross compiler only translates the engine's core and the user-written application from C# into JavaScript as shown in Figure 2. Afterwards, manual implementations (so-called "Platform Adapters") are added as JavaScript files. The scripts implement the same methods as in C# but use WebGL rather than OpenGL. The same solution is used for FUSEE's audio, input, text, and network libraries. This technique also enables applications written with FUSEE to run on different operating systems and on mobile devices as already mentioned.

2.2. FUSEE in Education

When designing FUSEE, the objective was to allow other students easily getting in touch with real-time 3D programming and gaining insight in what is happening under the hood of contemporary 3D game engines. Therefore, FUSEE is not a full-featured interactive game editor with some programming functionality but rather a software library for developing real-time 3D applications without steep learning curves. Students can access the functionality at various levels. For example, beginners could start at a high level by using the scene management, a wide range of prefab shaders, and other functionality, which would allow them developing visible objects and hierarchies of positions and rotations. Then again, experienced programmers are free to implement their own shaders and materials, create procedural geometry, handle affine and projective geometry, and much more. This can all be done without worrying about the underlying platform, the rendering API, or the operating system.

Besides learning background knowledge about real-time 3D programming, students also get involved with development processes applied when creating complex software libraries. They learn about source-code management tools, release planning, creating documentations, collaboration tools, and agile processes such as SCRUM in a targeted approach. These are valuable experiences directly applicable in their future education and in working life.

Early in 2013, another student project was started in cooperation with the Fraunhofer IPM Institute. This project uses FUSEE for visualizing huge sets of 3D laserscanning data. FUSEE has also been the subject of three students' theses during the last year. Other theses, research projects, and courses using FUSEE for educational purposes will start in 2014. So far, more than 30 students have directly or indirectly been involved in the development of FUSEE.

3. Conclusion

The one-year research project ended recently but the students continued their work. This shows the students' high motivation for an innovative project that requires a lot of skill and creativity. FUSEE also has a great potential for all students who want to learn how a contemporary engine works behind the scenes. Additionally, this paper shows that education on graphical computing can be greatly improved by including outstanding student projects in courses.

References

[Fur] Furtwangen University: FUSEE. [Online. 01 Feb. 2014]. URL: http://www.fusee3d.org/. 1

[Gad] Gadd K.: JSIL - .NET to JavaScript Compiler. [Online. 01 Feb. 2014]. URL: http://www.jsil.org/. 2

[Par] Parberry I.: A Simple Academic Game Engine. [Online. 01 Feb. 2014]. URL: http://larc.unt.edu/sage/. 2

[Uni] Unity Technologies: Unity3D. [Online. 01 Feb. 2014]. URL: http://unity3d.com/. 1