

Chapter 6

Leveraging Open Source Technology in 3D Game Engine Development

Tim Stowell

Utah State University, USA

Jon Scoresby

Utah State University, USA

K. Chad Coats

Utah State University, USA

Michael Capell

Utah State University, USA

Brett Shelton

Utah State University, USA

ABSTRACT

Market 3D engines have all the capabilities needed for developing full-featured 3D simulation and game environments. However, for those in education and small-business, it remains a formidable task to acquire the resources needed to purchase or create a development platform with cutting-edge capabilities. Leveraging existing and open-source software libraries can greatly enhance the main application development, freeing developers to focus more on the application concept itself rather than the needed supporting pieces. This chapter explores the nuances of successfully mixing core code with these third-party libraries in creating a fully functioning development environment. Many steps with accompanying checks-and-balances are involved in creating a game engine, including making choices of what libraries to use, and integrating the core code with third-party libraries. By offering insights into our open source driven process, we help inform the understanding of how game engines may be generated for other educational and small-budget projects.

DOI: 10.4018/978-1-60960-565-0.ch006

INTRODUCTION

Market 3D engines have all the capabilities that one would need to build full-featured 3D environments. Those that develop these market 3D engines often invest vast resources of money and people and, therefore, market 3D engines cost a significant amount to license. Educational institutions rarely have the funds to purchase a current top-of-the-line engine, choosing instead to research alternate avenues. These avenues focus mainly on current open source engines or older engines that could be purchased for a lower price (Cruz, 2006). These engines would not support many current development needs, or were too dated to use for cutting edge applications. Choosing instead to create a 3D engine from scratch allows the flexibility to customize functionality and development structure.

Our group teamed up with the Institute of Emergency Services and Homeland Security (IESHS) to develop software they could use in their training. Generally, our task is to build an incident command training simulation for firefighters. The simulation has been generated as a multiuser first-person perspective training program. Even though the simulation is designed for multiple user participation, the training is meant for the incident commander of that emergency situation. Currently the training will take place in the IESHS computer lab, but future development will allow this training to be expanded to firefighters to participate in the simulation via the Internet. The simulation is partitioned into two primary roles, a facilitator and client user. The *facilitator* starts and records the simulation whereas the *client* is meant for the individual firefighters participating in the simulation. The facilitator can also load previously recorded scenarios for learning. Part of the development is to expand this training to EMT, police officers, and other emergency personnel. To encompass all of these areas, we use the moniker Hazard Emergency & Accident Training (HEAT) to describe

the needs of our game engine. As described by Shelton et al. (2010), the HEAT engine aims to be a full-featured 3D training environment that includes single-player and multi-player scenarios, interactive menus triggered by key events, and support for specific instructional attributes such as automated assessment and replay. As such, it requires a number of pieces that many 3D game engines also share, including:

- 3D graphical capabilities
- VOIP (Voice Over Internet Protocol)
- Sound and physics

Developing software for any one of the above components is a large task in and of itself. To save time and money, it is often desirable to use “off-the-shelf” software components that already include the needed functionality. However, there are many issues to consider when making the choice to use available libraries because of potential issues that may exist within open code (Shelton, et al., 2010). For the HEAT engine, we used the following process when building necessary functionality of high-end systems, with each point to be described in more detail throughout the paper:

1. Perform background research into the library functionality
2. Make choices based on that research
3. Start developing core code (code developed in-house for the main project, not relying on off-the-shelf components)
4. Begin an iterative process of integration of core-code with 3rd party libraries

We next discuss the process we used for searching and deciding on what open source libraries to use to develop the 3D engine. Reading the forum discussion boards helped in making decisions—what has worked for others, and which library code to avoid (see Figure 1). It is difficult to contrast our particular open source approach with existing ones. There may be documentation

30 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/leveraging-open-source-technology-game/54358

Related Content

Load Frequency Control Strategy for Islanded Microgrid Based on SCQ() Algorithm

Qiang Wang and Zhenwei Huang (2024). *International Journal of Gaming and Computer-Mediated Simulations* (pp. 1-16).

www.irma-international.org/article/load-frequency-control-strategy-for-islanded-microgrid-based-on-scq-algorithm/339198

Enhancing Student Authenticity and Engagement Through Gamification and Problem-Based Learning in Higher Education

Nitin Vihari Poluru (2025). *Enhancing Engagement With Gamification: Education, Business, and Healthcare Perspectives* (pp. 29-52).

www.irma-international.org/chapter/enhancing-student-authenticity-and-engagement-through-gamification-and-problem-based-learning-in-higher-education/364653

World of Race War: Race and Learning in World of Warcraft

Alfred Weiss and Sharon Tettegah (2012). *International Journal of Gaming and Computer-Mediated Simulations* (pp. 33-44).

www.irma-international.org/article/world-race-war/74833

Sorting Out the Virtual Patient: How to Exploit Artificial Intelligence, Game Technology and Sound Educational Practices to Create Engaging Role-Playing Simulations

Thomas B. Talbot, Kenji Sagae, Bruce John and Albert A. Rizzo (2012). *International Journal of Gaming and Computer-Mediated Simulations* (pp. 1-19).

www.irma-international.org/article/sorting-out-virtual-patient/74790

Towards a Role-Playing Game Procedural Dungeon Generation Strategy to Help Developing Working Skills

Esteban A. Durán-Yañez, Mario A. Rodríguez-Díaz and César A. López-Luévano (2023). *Research Anthology on Game Design, Development, Usage, and Social Impact* (pp. 944-965).

www.irma-international.org/chapter/towards-a-role-playing-game-procedural-dungeon-generation-strategy-to-help-developing-working-skills/315523