

# Game Development: a Teaching Challenge

## ABSTRACT

Cogaen is a component-based game engine that addresses the issues of diversity and continuity of development interests as part of a game development curriculum.

## Author Keywords

Game education, game technology

## The Teaching Challenge

Game development is a particular challenge within the field of software engineering. For one thing, specific knowledge about a variety of technical domains is necessary to achieve a satisfactory result. Furthermore, game development involves the creation of assets to a considerably larger extent than traditional software projects. Creative steps actually precede the implementation of the game, as in the case of game and level design, and even during the programming phases, specific skills are required for the production and integration of game assets.

Academic institutions that educate game developers are thus confronted with a threefold mission: provide students with technical expertise in multiple areas; familiarize them with the creative process and the workflow/integration of assets; and provide them with experience working in dynamic team situations, as this cannot be effectively taught from the chalkboard. At first glance, the ideal educational environment would seem to be a mid-to-large-scale project that incorporates all three elements. However, such game projects are problematic for a number of reasons:

- Courses often consist of a fairly heterogeneous group of students with different backgrounds and interests. Hence, such groups rarely fulfill the requirements of a development team.
- The broad scope of computer games makes it difficult to complete a project within a semester, especially since at the same time content has to be taught and the acquired knowledge should be applied to the project. Projects over multiple terms are problematic as well as this usually entails the integration of new students who are unfamiliar with the project.
- Developing their own reusable software packages for computer games is more appealing to students. It provides them with profound, lasting knowledge which surpasses mere tool application. However, employing these tools in various student projects runs the risk of severely delaying the project or, even worse, failing.

In addition to the implementation of individual games, a major focus in the academic education of game developers is the development of domain-specific software packages and tools. In particular, this field of activity provides a basis for research. Game projects that meet the expectations of students and their future employers are hardly realizable in an academic setting. Experience has shown that the curriculum is limited due to time constraints (and, naturally, the demands of other coursework) and most projects are unfortunately abandoned after one term regardless of their status. This is a particularly irritating problem for projects that focus on the development of tools since it makes reusability virtually impossible; an insufficiently documented tool is useful only to the original developer.

## Cogaen – a Component-based Game Engine

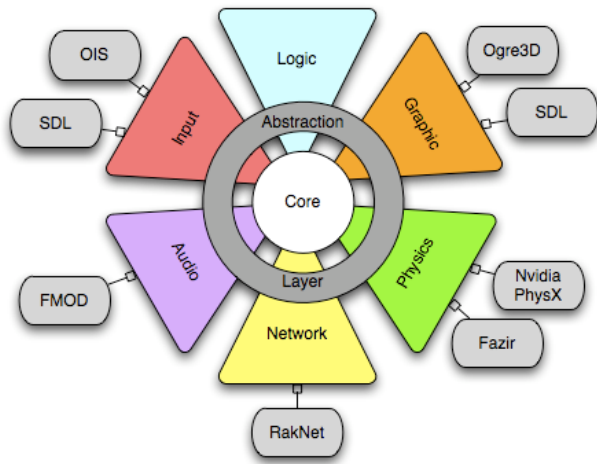
The component-based game engine Cogaen was developed in order to enhance the continued development of game engineering education in the Digital Media program at the Upper Austria University of Applied Sciences. The primary objective was to facilitate the development process of computer games with the aid of reusable components..

Further revisions added the subsystem concept to the Cogaen architecture, which provides the capability to develop and use larger parts independently of each other. These so called subsystems take over specific tasks in a game. Such a subsystem can be implemented and tested separately and if required it can be easily exchanged by another subsystem that performs the same task.

Figure 1 shows the basic structure of a Cogaen application which is constructed by including a subset of all available subsystems. The core of Cogaen is responsible for the central administration of the subsystems. In addition to the subsystems, Cogaen also offers all the benefits of component-based game development.

**Breaking New Ground: Innovation in Games, Play, Practice and Theory. Proceedings of DiGRA 2009**

© 2009 Authors & Digital Games Research Association (DiGRA). Personal and educational classroom use of this paper is allowed, commercial use requires specific permission from the author.



**Figure 1:** Modular Structure of the Cogaen framework

Cogaen represents more than a mere development framework, but also employs a multi-faceted didactic model to accommodate a variety of development interests. This model has been examined during three use-case scenarios which have demonstrated both its effectiveness but also areas for its improvement. In particular, although students who were involved in the development of the framework were very motivated to utilize its potential, persuading new students of the benefits of this modular approach has taken considerable effort.

## REFERENCES

1. El-Nasr, M.S. "Learning through game modding" in *Computers in Entertainment* vol. 4, no. 7 (New York NY, 2006), ACM.
2. Hoetzlein, R.C. and Schwartz, D.I. "GameX: a platform for incremental instruction in computer graphics and game design" in *SIGGRAPH '05: ACM SIGGRAPH 2005 Educators program*, no. 36 (New York, NY, 2005), ACM.
3. McGill, M. "Critical Skills for game developers: an analysis of skills sought by industry" in *Proceedings of the 2008 Conference on Future Play* (New York, NY, 2008), ACM, pp. 89-96.
4. Patterson, S. "An Object-Composition Game Framework" in *Game Programming Gems 3* (Hingham, MA, 2002), Charles River Media, pp.15-25.
5. Rene, B. "Component Based Object Management" in *Game Programming Gems 5* (Hingham, MA, 2005), Charles River Media, pp.25-37.
6. Stoy, C. "Game Object Component System" in *Game Programming Gems 6* (Boston, MA, 2006), Charles River Media, pp.393-404.
7. Varanese, A. *Game Scripting Mastery*. Premier Press, Cincinnati, OH, 2003.
8. Volk, D. "How to embed a game engineering course into a computer science curriculum" in *Proceedings of the 2008 Conference on Future Play* (New York, NY, 2008), ACM, pp. 192-195.