

A Study on the Game Programming Education Based on Educational Game Engine at School

Jongho Jeon¹, Kwanwoong Kim² & Soonyoung Jung¹

¹ Department of Computer Science Education, Korea University, Seoul, Korea

² Department of Game Development, Chunnam Techno University, Jeollanam-do, Korea

Correspondence: Soonyoung Jung, Department of Computer Science Education, Korea University, Seoul, 136-701, Korea. Tel: 82-234-855-151. E-mail: jsy@korea.ac.kr

Received: September 2, 2012

Accepted: October 16, 2012

Online Published: November 27, 2012

doi:10.5539/jel.v1n2p282

URL: <http://dx.doi.org/10.5539/jel.v1n2p282>

Abstract

It is believed that the game programming education at school should be conducted in consideration of an individual student's ability, an elementary programmer. Language to be used in the programming education also need to be associated with the ones that are actually used in the game industry. Lately, many researches on the educational programming are being carried out and for this reason, the language used in such game programming education, which is targeting students, should be carefully determined. In addition, it is suggested that the game programming education should not be limited to a student's understanding the knowledge of the programming language but should have the students learn how the knowledge of the programming language is actually adopted to the practical game development. In order to solve the problem, this study proposed the game programming education framework of programming language knowledge to be conducted in association with the game programming skills which are actually applied to the game development according to the stages of design, generation and evaluation of the game development. Afterwards, this study designed and developed an educational game engine for the game programming skills training. The game programming education based on educational game engine was performed to the students. As a result, some games have been developed by the students which were most impossible under the current knowledge acquisition centered curriculum.

Keywords: game programming education, educational game engine, 3D game engine

1. Introduction

The recent trends in the game development field are categorized by three characteristics. To begin with, time for the development of a game is getting really short. Second, up-to-date techniques such as 3D, network and multi-media are being applied to the game development. Third, a game engine is used to develop a game of high completeness in the aspect of stability. Hence, it is inevitable that education time should be constantly increased in order to conduct the game programming education according to the recent game development trends. However, at school, there is a certain amount of time for the education and this makes it very difficult to teach students the game programming with the latest technologies such as 3D, network and multi-media applied

In order to solve the problem, this study proposes a game programming education based on educational game engine which will be proper enough to be applied at school that programming subject is on the curriculum.

Among massive knowledge of the game programming language, the proposed education method has only focused on vitally necessary knowledge to develop a game according to the game development stages by using an easily understandable programming language of the educational game engine as well as a support function. For the verification of the proposed education method, it was examined if a game is eventually developed.

2. Background

2.1 Game Programming Education

The purpose of the game programming education is considered to 'make a student equipped with an ability to develop a game by understanding and using the programming language which is necessary for the game development.'

In general, the current programming education has solely engaged in learning knowledge on the programming

language. Thus, students frequently fail to understand how the programming language knowledge would be adopted to the actual game development (Perkins & Martin, 1986). Therefore, in order to work on the problem, Davies (1993) suggested that the curriculum should be divided into programming language knowledge education and programming skills training. In other words, the programming language knowledge will be much more easily learned if the programming skills education is combined (Widowski & Eyferth, 1986).

For the stages of design, generation and evaluation in the program development, Anthony Robins (2003) proposed a programming framework as he organized learning elements which had been categorized into the programming language knowledge, the programming strategies and the mental model.

2.2 Educational Game Engine

Game engine means what is embodied in the type of various software libraries or modules required for development of the game (Junsik Park, 2008). Game engine is classified into rendering engine, animation engine, sound engine, artificial intelligence engine and server engine, etc. according to its technical characteristics (Heejung Kang, 2004). Game companies are using game engine to secure the rapid changes of game technology and competitiveness of game in general (Gangwook Lee, 2003). That is, they use game engine to develop the game with high completeness saving development cost in shorter time as the size of game becomes large scale.

Even until now from the paste, the education for the game developers has been carried out focusing on the acquisition of programming knowledge. However, the game companies prefer the developers who have the experience of creating one completed game by performing the practical project that develops the game using game engine rather than such fragmentary acquisition. Using game engine for game programming education students can understand more the principles of programming logic than programming syntax (Cristiane C. H. et al., 2010).

For the educational game engine are SAGE(Simple Academic Gaming Engine) to teach the students the computer science and the game programming and also, Alice which has been created for introducing Object Oriented Programming.

For the educational game engine, the programming language needs to be simplified as much as possible so that the students would understand commands of the game programming language as they check them with their own eyes (Du Bouley, 1988). Add to this, for those who are not familiar with the game development, the language offered by the educational game engine should meet requirements such as 1) readable syntax, 2) high level language except difficult concepts, 3) small size grammar for fast and easy learning, 4) core concept for programming and lastly, 5) programming language connection between academy and industry (Kolling, 1999; McIver & Conway, 1996).

3. Method

3.1 Design of Game Programming Education Framework

In this study, a game programming education framework with the educational game engine applied to programming framework of research by Anthony Robins (2003) was developed.

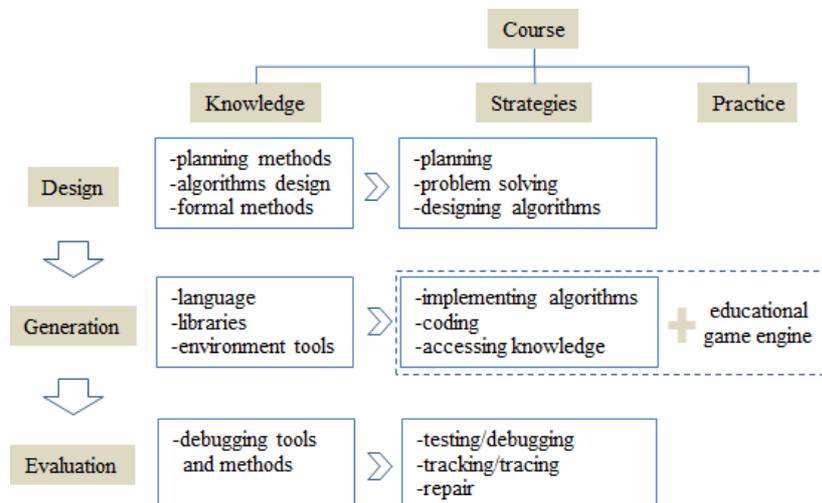


Figure 1. Game programming framework with the educational game engine

This framework has been rather focused on programming skills than programming language knowledge. While the students learn to understand the game development stages, it is likely for them to know how the programming language knowledge will be used in the real field. Add to this, since the students easily acquire only the programming language knowledge which is necessary for the game development by using the educational game engine, they are able to come up with a new game in a limited time.

3.2 Design and Implementation of Educational Game Engine

3.2.1 Design

The use of game engine itself should be easy and required basic functions of game engine. Also the functions should be provided to enable the constant extension in the open architecture method considering various education situation (Scott Nykl et al., 2008).

Basic essential functions include:

- Provision of educational API full sources except for engine core
- Provision of strong & easy program environment through class coupling like LEGO block
- Supporting API in constant newest technology through extensible structure
- Provision of independent file system & resource managing ability
- Provision of game server-related template class & API
- Supporting production of contents such as 2D, 3D & multimedia API

The education of the educational game engine is performed independently along the practice of game programming and the completed portfolio that all contents of education are integrated should be produced. The interface should be easy along the level of teachers or students and various learning exercise should be provided.

The biggest feature of educational game engine is the provision of the functions that is suitable for being used in site of school practice. First, it provides the educational level API separately for the teachers and students who feel difficulties in using game engine and the functions that can be utilized by creating the educational framework that is suitable for the level of students. Second, the product of completed game contents that actually operates can be developed by using the educational API and various interfaces. It provides the functions so that the independent unit of practice for each game developing phase aiming at the product of completed game contents. Third, it provides various templates and learning exercises for the teachers and students who feel lack of practical data of game programming.

The functions that are optimized for practice of game programming include:

- Provision of educational level of API that is suitable for practice
- Provision of low level of API for production of portfolio
- Provision of framework for prompt practice & basic library
- Inducement of independent type of practice along each phases
- Provision of student-centered SDK that is composed of large volume of learning exercises
- Suggestion of specimen model for production of game tool & various types of template
- Provision of various utility modules required for contents development

3.2.2 Implementation

Microsoft Visual C++ 8.0 was used as the development tool. In order to operate educational game engine, it should support above Microsoft Windows XP and above Shader 2.0 for graphic card and above Net Framework 3.5, Microsoft DirectX 9.0c for the environment.

Educational game engine basically provides Core API as the essential function for game programming and Engine API as the upper level of concept to deal with that easily. Educational game engine provides performance-based API that is enormous and difficult to understand after simplification to essential shape through educational level layer. The learners who develops the actual game contents can use the functions easily by using App Framework and various supporting tools.

The class was produced based on object-oriented technology of pure C++. All functions of game contents is realized as the class and the users can use the required class by definition or succession, overriding.

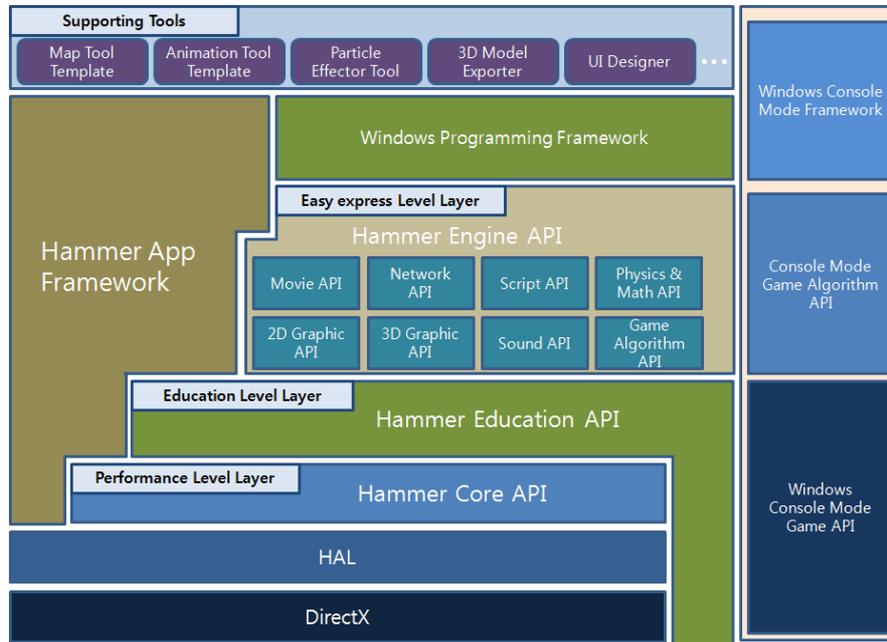


Figure 2. Architecture of educational game engine

Educational game engine provides rendering component, physics component, artificial intelligence component, network component, character & animation editor, geography & game world editor, special effect & particle editor and GUI editor.

The functions of main components are the same as follows.

Table 1. The functions of main components

Component	Function
HAL	is the layer to support hardware independency of engine between DirectX that is the lowest layer of engine and Core API that is performance level layer. Various cross platforms can be realized through this layer.
Core API	is Core API of engine and provides the optimized API that can raise performance for development of game contents.
Education API	simplifies the parts required for practice education purpose among API's that are difficult to understand and enormous and provides in type of separate educational API.
Engine API	provided API that is specialized for game and high level of algorithm as Console Mode Game Algorithm API. It provides the easy interface and high level of functions through Core API. It provides the high level of API required for operating 2D/3D graphic API and sound multimedia.
Windows Programming Framework	is the lightened framework that is provided for working on low level of framework and it can excludes high level of scene node control & updater structure and produce the framework specialized for its own application.
App Framework	is the framework that has the unique structure of engine's own and it provides the high level of interface and functions so that it can access the engine. It supports all precision frameworks to compose prompt framework for practice during curriculum and produce the product with high performance.
Supporting Tools	provides various tool templates to enable the independent tool production as well as the fixed engine tool. Especially, the special interface production tool called UI Designer that cannot be seen in the existing engine provides the functions that can control interface and window frame easily with the simple images.

3.3 field Experiment

In order to verify the applicability of the educational game engine-based game programming education in the field, the study conducted a 34 week of field experiment to students from the game development department of a college. For an actual game development, they were told to do in a small group. The game development was carried out in pursuit of the students' general comprehension in the game development as they gained the game programming knowledge and understood the game development strategies. For 34 weeks, the students were given 3-4 hour of theoretical classes and educational game engine-based practical classes.

4. Result

After the 34 week of education to verify the applicability of the educational game engine based game programming education, all the students from the game development department of a college appeared to create simple games which were still considered to be equipped with main functions that would be required for the development of a game. Comparing to the previous programming language knowledge based curriculum which had failed to have the students develop a game in 34 weeks, the achievement is believed to be significant.

Through the game development, the students were able to learn about the game engine, understanding how they could adopt the game programming knowledge to the game development process and strategies.

Despite all the different themes of the games that the students had created, most of them were discovered to use functions such as the geographical editing, the effect editing and AI engine of the educational game engine, realizing the multiple character processing, the character state controlling and the character controlling in the third person.

Table 2. Screen shots of games, all the games have the similar function, developed by students

	Function	Screen shots
Game1	Online action RPG Using server engine Multi-users enable	
Game2	Action RPG Multi-users enable	

5. Conclusion

This study proposed a game programming education framework focused on acquisition of programming skills rather than programming language knowledge and composed of programming knowledge, strategies, practice using educational game engine based on the game development stages. According to the experimentation conducted to the students of the game development department at a college, the students who were only able to fragmentarily understand the programming language knowledge and an individual function are now capable of creating a highly-improved game with various knowledge on the game programming language applied.

This study expects such results to be used to upgrade the current game programming language centered curriculum to a curriculum centered on acquisition of the programming skills which is practical to the actual game development. While the results of the study are well adopted to the school education, the students, elementary programmers, would be able to utilize a tool such as the educational game engine.

In a follow-up study, efficient uses of educational methods according to various purposes and functions of games should be investigated.

References

- Alice (2012). An Educational Software that teaches students computer programming in a 3D environment. Retrieved from <http://www.alice.org/>
- Anthony Robins, et al. (2003). Learning and Teaching Programming: A Review and Discussion. *Computer Science Education*, 13(2), 137-172. <http://dx.doi.org/10.1076/csed.13.2.137.14200>
- Cristiane C. H., Luciano S., Fafael A. S., Juliano S., Manuel F. P. L., Luis N. M. B., & Ismar F. S. (2010). Teaching Programming Principles through a Game Engine. *CLEI Electronic Journal*, 13(2). Retrieved from <http://www.clei.cl/cleij/papers/v13i2p3.pdf>
- Davies, S. P. (1993). Models and theories of programming strategy. *International Journal of Man-Machine Studies*, 39(2), 237-267. <http://dx.doi.org/10.1006/imms.1993.1061>
- Du Boulay, B. (1988). Some difficulties of learning to program. In E. Soloway & J. C. Spohrer (Eds.), *Studying the novice programmer* (pp. 283-299). Hillsdale, NJ: Lawrence Erlbaum.
- Gangwook Lee. (2003). Study of game engine developmental element & procedure using project management methodology (pp. 14-66). Ms thesis, Sangmyung University. Retrieved from <http://www.riss.kr/link?id=T8989317>
- Heejung Kang. (2004). Ajou Engine: 3D Game engine architecture centering on rendering and scene module (pp. 5-10). Ms thesis, Ajou University. Retrieved from <http://www.riss.kr/link?id=T11077356>
- Junsik Park. (2008). Study of network game engine technology for distributed processing (pp. 3-6). Ms thesis, Dankook University. Retrieved from <http://www.riss.kr/link?id=T11201522>
- McIver, L., & Conway, D. (1996). Seven deadly sins of introductory programming language design (pp. 309-316). Proceedings of the 1996 International Conference on Software Engineering: Education and Practice. <http://dx.doi.org/10.1109/SEEP.1996.534015>
- Kolling, M. (1999). The design of an object-oriented environment and language for teaching. Phd thesis, University of Sydney. Retrieved from <http://129.12.3.236/papers/1999-mik-thesis.pdf>
- Perkins, D. N., & Martin, F. (1986). Fragile knowledge and neglected strategies in novice programmers. In E. Soloway & S. Iyengar (Eds.), *Empirical studies of programmers, First Workshop* (pp. 213-229). Norwood, NJ: Ablex. Retrieved from <http://dl.acm.org/citation.cfm?id=28896>
- SAGE. (2012). SAGE: A Simple Academic Game Engine. Retrieved from <http://larc.csci.unt.edu/sage/>
- Scott Nykl, Chad Mourning, Mitchell Leitch, David Chelberg, Teresa Franklin, & Chang Liu (2008). An Overview of the STEAMiE Educational Game Engine (pp. 21-25). 38th ASEE/IEEE Frontiers in Education Conference. <http://dx.doi.org/10.1109/FIE.2008.4720454>
- Widowski, D., & Eyferth, K. (1986). Comprehending and recalling computer programs of different structural and semantic complexity by experts and novices. In H.P. Willumeit (Ed.), *Human decision making and manual control* (pp. 267-275). Amsterdam: North-Holland Elsevier.