Educational computer games development: methodology, techniques, implementation

Olga Shabalina ,Pavel Vorobkalov ,Alexander Kataev ,Alla Kravets

Volgograd state technical university, Russia

Abstract

Educational games development process includes all the stages of software development process. But there are important specific problems of educational games development to be solved related to the way of integration of educational content to the game and technologies of its implementation. In this paper we describe the approach which we propose as a method for achieving this aim. We give evidence of how that method should be used and an example of how we have implemented this approach specifically in developing a computer game to teach object oriented architecture.

Keywords: educational games development, methodology, programming paradigms, programming languages.

1. Introduction

Educational games development process includes all the stages of software development process. But there are important specific problems of educational games development to be solved related to the way of integration of educational content to the game and technologies of its implementation.

Educational games should certainly encapsulate their educational goals, but must also meet the requirements of regular games: have attractive gameplay and scenarios, high quality sounds and graphic effects, etc. Underestimation of these characteristics might cause a loss of educational effect of the game by reducing player engagement and therefore motivation to learn. One of the main problems that educational games developers face is developing games that offer effective learning and which at the same time are really attractive. Game attractiveness can be achieved in different ways but one of the most powerful factors of increasing game attractiveness is an engaging game story.

Educational games are developed by some commercial companies, working in game industry, and also by people belonging to education – academicians, students and just game hobbyists. Active scientific research in the field of educational games development and wide use of educational games show growing interest, and provide the necessity of further investigations in this field.

2. Approaches of educational games development

Different ways are used in educational games for the integration of learning process into the game context. Integration approach considerably depends on peculiarities of knowledge domain, which is supposed to be learnt through the game.

One of frequently used approaches of educational games development is simulation [1][2]. Such games are based on

simulation of real work of professionals in different fields. Design of simulators includes development of realistic model of a work process in a corresponding subject domain, implemented in a particular game.

One of modifications of simulation approach is a case-study approach, based on successful real-world examples (cases) and realized mostly in role-playing or adventure games. Design process of games that implement case-study approach includes development of game plot based on story, development of virtual environment and technique of its presentation [3]. One of the most important components of such games is a corresponding decision model which should be used for estimation of player's decisions. Development of games of this kind is similar to development of an expert system in a corresponding domain.

Considered approaches to educational games development are based on virtual reality. But there are knowledge domains dealing with abstract concepts (mathematics, programming, data bases, etc.). The content of these domains have no direct real interpretation and simulation-based approaches can't be used for the development of corresponding games. Thus new approaches of integration of educational content to a game need to be developed for using by game developers.

3. Methodology

In [4] we suggested an approach to development of educational games for learning subjects dealing with abstract concepts. The approach is based on the idea of using some basic paradigms forming the subject content that can be interpreted as axioms, rules and limitations of a virtual world where virtual habitants – game characters can live. A player is matched with a main game character and is immersed in a new unknown world. The player must investigate this world and master it, for living in the world he must develop and improve his character's skills. Correspondingly the player obtains knowledge and skills of learning subject interpreted in a game context. He just can't exist in the world without knowing learning subject as a game process is combined with a learning course.

The realization of the approach for each educational game depends on the desired learning outcome that that will be integrated into the game. Integration of learning outcomes can be presented as a sequential combination of two information spaces –the knowledge space and the game space. The first step is to develop a representation of the learning outcome components in the game. Then, depending on the particular educational aim, the game concept should be created; learning elements should be interpreted in the game context and combined in the game scenario. Thus, development of educational games demands development of the game components (learning material, game resources, educational game engine) while taking into account their joint use.

4. Techniques

The implementation of the approach meets such poorly structured problems of learning content development as the interpretation of the subject domain in a game context and the development of a game scenario.

The subject domain introduction in a game requires elaboration of learning content presentation methods and editing tools. Developers can either use existing content creation tools or develop specialized editors. Players can obtain learning content from dialogues with non-player characters or while exploring the game world (in a similar way to collecting information in regular games). Dialogues with game characters can be voiced. Usu-

ally, the game interface can contain game menus, indicators, dialogues and information message tools.

Normally games include a small set of user interface components, and interface abilities are limited (i.e. the game might not have window position and size changing options, scrollbars in text output windows, etc). The size of text blocks and a set of text formatting options are minimized; sheets, pictures and hypertext are not common in games. Game interfaces usually correspond to a given game genre. However, educational games may need to provide the possibility of representing much larger blocks of text in comparison to common games. Moreover the text can be formatted (i.e. can include tables, diagrams, formulas, etc.) Realization of such an interface is not a trivial task; developers might meet some technical problems caused by limited computer resources, and also game usability requirements. This problem can be solved using external programs (different viewers and browsers) for presentation of learning information in a game. But in this case opening the external viewers will cause hiding of the game window which could divert the player's attention. Also the format of existing viewers might not correspond to the game style, and learning information may not be thought by a player as a part of the game.

Using the approach also requires elaboration of methods of game tasks fulfilling support: solution input, checking and visualization, internal representation of the task and solution. If game tasks can be realized through game dialogs (for instance in a game for training communication skills), they do not require additional educational game engine support, as this function is always realized in some way practically in common game types. When the text of a task solution includes formulas, diagrams etc., it is necessary to develop specialized editors, able to work with rich-text information blocks. It would be useful for game developers to analyze existing methods of editing of certain types of text and then choosing the most normal method for a potential customer. For instance, if the text of a game solution includes math formulas, MathML or TeX languages can be used.

Depending on the input language it can be possible to use existing software components for solution input. This approach guarantees the input method will work correctly and allow a gamer to be trained in the skills needed for using the applications component itself (e.g.MathCad, MsWord, Visual Studio etc). On the other hand integration of existing applications to a game might be a nontrivial task because of some specific features of the I/O interaction devices in the game application. Depending on the task type and method of solution checking different ways of visualization can be used. One of the easiest ways is matching each task solution with an appropriate animation model. Another approach, suitable only for executables is the direct demonstration of solution results.

5. Implementation. A game for learning OOD and the C# programming language

We have used suggested approach to develop a game for learning object-oriented paradigm and the programming language C# [5]. The idea of the game is based on of a virtual world and its inhabitants who are able to develop their skills. The main character in the game is "Professor Kamaev", who occasionally destroys his laboratory and due to an explosion is turned into a small child's toy – a transformer, made from 16 small magnets. He is required to adapt to his new appearance to survive in the world of big things and strange creatures born after his experiments. He must reach and recover his la-

boratory in order to become a human again.

While playing the player develops the game world, and adds new characters. In terms of object-oriented paradigm he creates new classes and new objects of these classes. Description of the game world is represented in the game as a class diagram. The final version of the diagram, created by the player corresponds to simplified version of the set of relations between classes used in the game. Some fragments of the game programming code are also available to the player in order to show him how the game is realized. Thus the player can understand how the object-oriented paradigm is used for real task development.

Fig. 1: Transformation of "Professor Kamaev" to a small toy Kammy

6. Conclusions and discussion

In this paper we have presented the results of our research around development of educational games. We have seen both through experience and with reference to the literature that the main problem for educational game developers in total is to provide an attractive game (one which is challenging and engaging and exhibits common game characteristics) which at the same time adequately integrates the learning objectives. This problem demands answering of such questions as how to integrate learning and game components, how to implement the integration approach and how to develop games using this methodology.

In our paper we try to answer these questions concerning educational games: we describe the design methodology and consider different techniques that can be used for implementation of our approach.

We tested the game "Kammy" on a target group of 1st-year students studying programming at CAD Department of Volgograd State Technical University that graduates software engineers. We received evidence that using our game helps to understand serious SE subjects. And besides students mentioned that the game itself was attractive and motivated them to continue playing and therefore continue learning. Thus we have received evidence that our approach allows the creation of effective and attractive learning games.

7. Acknowledgement

The authors would like to thank the head of CAD Department of Volgograd State Technical University Prof. Kamaev for the permission to use his name as a name for the main game hero ("Professor Kamaev") in the "Kammy" game.

8. References

[1] J. Nutaro, "Building Software for Simulation: Theory and Algorithms, with Applications in C++", Wiley, 2010.

[2] J.A. Sokolowski, C.M. Banks, "Principles of Modeling and Simulation", Hoboken, NJ: Wiley. 2009.

[3] T. Hainey, T. Connolly and L. Boyle "Development and Evaluation of a Game to Teach Requirements Collection and Analysis in Software Engineering at Tertiary Education Level", *Proc. Of the 3rd European Conference on Games-Based Learning*, pp. 145-153, 2009.

[4] O. Shabalina, P. Vorobkalov, A. Kataev, A. Tarasenko, "3I-Approach for IT Educational Games Development", *Proc. Of the 3rd European*

Conference on Games-Based Learning, pp. 339-344, 2009.

[5] O. Shabalina, P. Vorobkalov, A. Kataev, “Educational games development: integration of training and game components”, *Open education: Scientifically-practical Journal*, №2 (85), Part 2, pp. 290–294, 2011.