EDUCATIONAL GAMES FOR LEARNING PROGRAMMING LANGUAGES

Olga Shabalina, Pavel Vorobkalov, Alexander Kataev, Alexey Tarasenko

Abstract: A concept of educational game for learning programming languages is presented. The idea of learning programming languages and improving programming skills through programming game characters' behavior is described. The learning course description rules for using in games are suggested. The concept is implemented in a game for learning C# programming language. A common game architecture is modified for using in the educational game. The game engine is built on the base of the graphical engine Ogre3D and extended with game logic. The game has been developed as an industry level commercial product and is planned for sale to educational institutions.

Keywords: educational games, serious games, digital game-based learning, e-learning, course notation, game engine, learning engine.

ACM Classification Keywords: K.3.1 Computer Uses in Education; K.3.2 Computer and Information Science Education

Conference: The paper is selected from Third International Conference "Modern (e-) Learning" MeL 2008, Varna, Bulgaria, June-July 2008

Introduction

Teaching students in technical universities faces the problem of low motivation of students. One of the effective ways of raising motivation and attraction of learning process is using games. Use of computer games for non-entertainment purposes (so called Serious Games), and its sub-field Digital Game-Based Learning (DGBL), which means the use of computer games specifically for educational purposes [1], are an emerging area of research and interest to this area is growing rapidly. Combining content with computer games is a newest trend in e-learning [2]. The number of gaming-related courses has increased in recent years. Educational games are used in different areas: marketing, finance, economics, management, innovation, knowledge management, etc. The main objectives of some European projects concern educational games design and implementation.

Game-based approach is also very effective in training employees of different companies. Evidence suggests that adults learn more and retain more in courses that incorporate such game elements as competitive scoring, increasingly difficult player levels and fantasy role playing [3].

By now the potential of digital game-based learning remains still largely unrealized [4], and design and implementation of new games in education process is in great demand.

Game Concept

Game-based approach achieves high learning results in areas where interdisciplinary knowledge is necessary and where skills such as critical thinking and problem solving in a group are of importance. To a great extent it can be referred to the computer science disciplines. Specialists in this field are called for, and their training is difficult and includes skill gaining. One can mention such subjects, demanding skill gaining, as algorithmization and programming, artificial intelligence, computer graphics and so on.

Saying 'educational game' we mean a learning system, which realizes some or all components of learning process (learning theory, gaining skills and experience, estimation of knowledge level) in a game context. But unlike learning system educational game should not use direct teaching approaches, but should provide getting new knowledge through playing game. Educational game concept depends on a learning course. So at first one should design a learning concept and after it design a gameplay.

Computer science students start the learning process from learning programming languages. This subject is very good for teaching through games. Such games can be used by both beginners and those who want to improve their skills. While playing learner gets theoretical knowledge and an experience of programming. If he is really interested in this game he will go to external information (books, Internet) and in that way study advanced programming.

We consider that the main objective of learning programming in games is awaking interest in programming through game. Thus the game concept should be based on three components:

- learners must get the course information through it's interpretation in game world;
- learners must see the result of his programming in a game context;
- results of programming must influence game results.

The concept is based on a Role Playing Game (RPG) genre. The RPG genre has been chosen because it enables a player to improve a game character, players often associate game characters with themselves. The main game character is a transformer, who lives in a virtual world. A player can program different transformer shapes and his behavior. Different situations in the game require different characteristics of the main character. Transforming shape allows players to get optimal character parameters for playing.

At the beginning a player programs the basic character abilities (moving, jumping, and attacking). The results of programming are used by the player to control character behavior in the game. Then the player can program controlling algorithms for intelligent behavior of the character (overcoming different obstacles, going out labyrinths and so on). Effectiveness of the code determines effectiveness of character behavior. Unreasonable or foolish behavior prompts player to improve program code. Thus the player obtains knowledge and gains skills while playing.

The components of learning process are realized in the game context as follows:

- getting theory with the help of game characters;
- gaining skills through programming transformer shapes and behavior;
- playing result depends on the result of programming so a player can estimate his knowledge level when playing.

A game story and scenario are developed by a game designer. The game scenario consists of game levels. Each level includes several game quests. In educational game quests include some course information and assignments for getting practice in programming. Each assignment solution needs knowledge of related course chapters, on the other hand information and assignments are associated with a game story. Thus levels completing forces player to learn new course chapters and solve level assignments, so the game scenario relates to the learning course (Figure 1).

For example, games usually start from input by a player some information about himself. And as each programming language course begins from learning data types, we use the input process for learning basic data types (Figure 2). A player fills not only the information fields, but he must also choose appropriate data type for each field from the list. This list includes data types, their purpose and range. This assignment is used for explaining a concept of "class". For explaining a concept of object-oriented design the real game class hierarchy is used.

The correlation between game levels and course chapters provides using the game not only for individual learning but also for using in university or at school. Teachers might use the game level by level according to their studies, so learners would complete playing the game when they complete studying the course.

For learning in the game a game-related course description has been elaborated. The course is divided into elements – information blocks of two types: theory information blocks and assignment blocks. Each block relates to one or several chapters of the course.

Information blocks can be of two types: displayable and reference blocks. Displayable information blocks are parts of the game script and are displayed for a player. Reference information blocks contain additional information on the corresponding chapter and are available for the player on demand. A set of displayable and reference information blocks form full learning course theory.

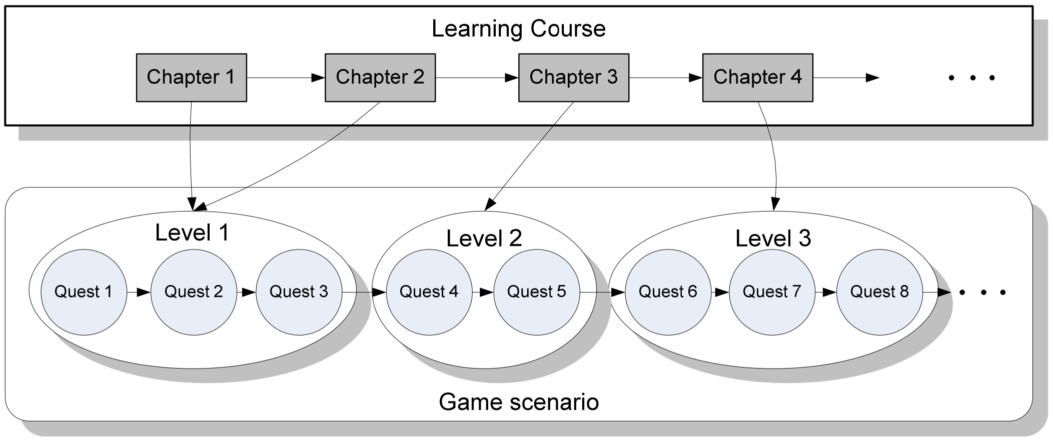
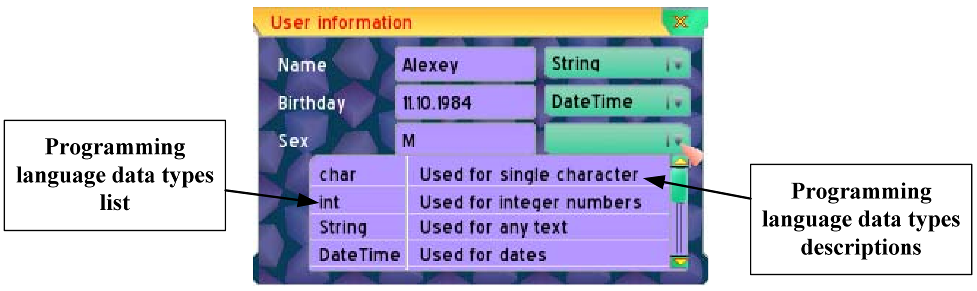


Figure 1: Game script and course chapter

Each assignment block includes assignment description, solution description, and solution interpretation. Assignment description is a displayable information block with a quest description. Assignment solution requires writing some program code. Solution description includes solution testing rules or/and solution code examples. Solution interpretation includes right and wrong solutions visualization techniques.

Two ways of solutions check have been defined: verification and running. Verification means checking source code using corresponding rules. Running means executing source code and checking the result. We use verification for checking assignments that have simple solutions for a player (for example, programming transformer basic moves). Real implementation methods for these moves in the game are more complicated. Interpretation of right solutions of verifiable type uses the real methods. Methods, showing that the assignment hasn't been executed, are called for interpretation of wrong solution of verifiable type (for example if you try to make main character to move left and the corresponding solution code is identified as wrong, the transformer will fall). For visualization of runnable solution the code of solution is being executed and the results are visualized.


User information
Programming language data types list
Programming language data types descriptions

Name	Alexey	String
Birthday	11.10.1984	DateTime
Sex	M	

char	Used for single character
int	Used for integer numbers
String	Used for any text
DateTime	Used for dates

Figure 2: Using input process for learning data types

Course description is made by instructional designer and game coder. The instructional designer describes a course in the natural language, and then the game coder translates it into engine input language. The course description is made in three stages. First stage (logical level description) should be made by the instructional designer. This level describes blocks' content, their sequence and links. Second stage (extended logical level) is made both by instructional designer and a game coder. The game coder describes the solution testing types and solution visualization techniques. The last stage (implementation level) is made by the game coder. For coding information blocks and links mark-up language is used. Blocks sequences, assignment interpretation and solution blocks are coded in script language.

For marked up text presentation HTML support is applied. For programming assignments solutions .NET Framework is used. The choice of this framework is caused by its feature to use different programming languages.

Implementation

The educational game concept has been implemented in the learning game for C# programming language, which is the main language of .NET Framework. System architecture is based on common game engine architecture, but it is extended for using in educational games (Figure 3) and consists of two high-level subsystems: a game engine and a learning engine.

The game engine is built on the base of the graphical engine Ogre3D [5] and enlarged with game logic and advanced user interface. Script core provides game engine modules interaction. Using scripts simplifies game logic programming and avoids recompiling game engine after learning course modification.

The learning engine completes game engine functions with learning process support, which includes information course blocks and assignments control. Learning engine modules and routines are exposed to scripting core for organizing course sequence control and assignments solutions check from game scripts.

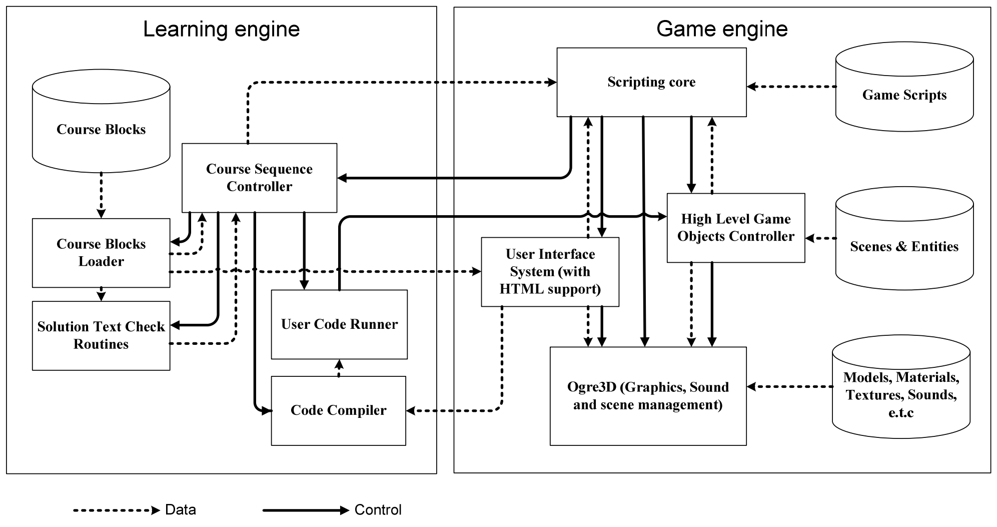
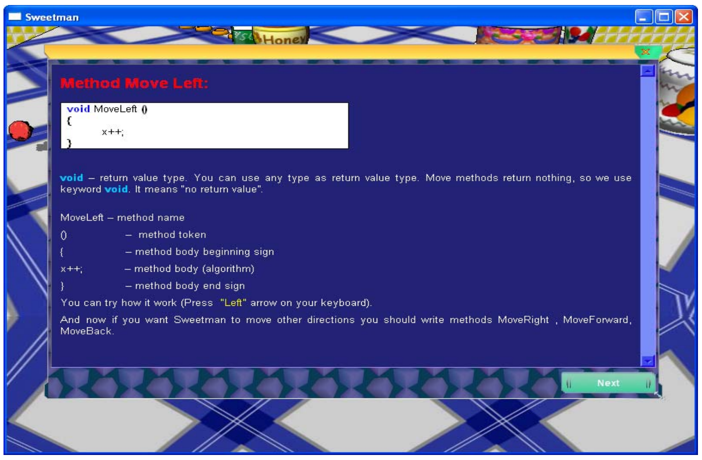

Learning engine
Game engine
Course Blocks
Course Sequence Controller
Course Blocks Loader
Solution Text Check Routines
User Code Runner
Code Compiler
Scripting core
Game Scripts
High Level Game Objects Controller
Scenes & Entities
User Interface System (with HTML support)
Ogre3D (Graphics, Sound and scene management)
Models, Materials, Textures, Sounds, e.t.c
Data
Control

Figure 3: System architecture



Method Move Left:

```
void MoveLeft ()
{
        x++;
}
```

void – return value type. You can use any type as return value type. Move methods return nothing, so we use keyword void. It means "no return value".

MoveLeft – method name

```
()          –  method token
{           – method body beginning sign
x++;        – method body (algorithm)
}           – method body end sign
```

You can try how it work (Press "Left" arrow on your keyboard).

And now if you want Sweetman to move other directions you should write methods MoveRight , MoveForward, MoveBack.

Figure 4: Screenshot «Visualization of Assignment Description Block»

Game implementation is based on free for commercial use libraries and utilities. Graphical engine Ogre3D is chosen because it supports modern technologies, has extensible architecture, and is allowed for commercial usage. Lua is used as a scripting language because of handy syntax, easy integration process and fast interpreter. The .NET Framework built-in C# compiler is used to compile assignment solutions. Encrypted zip archives are used to store course blocks. The engine is developed in C++ language.

Practical Usage and Examples

The C# learning course description has been created. The course includes a set of blocks, described in a game-related notation. An example of assignment description block is shown in Figure 4. The assignment is intended for learning functions. A player is shown an example of function definition (move left method implementation) and is suggested to teach his character to move, i.e. to implement the rest functions for moving right, forward and back.

Conclusion

To develop learning games for other programming languages it is necessary to describe a learning course according to the course description rules. The concept and game engine can also be used for other educational games development. In this case the ways of solution check should be defined according to appropriate learning course and proper tools should be implemented.

We consider that this approach allows gaining and improving knowledge and skills in computer science, and raising the motivation to study.

Bibliography

1. Serious Games Pathfinder [online] http://www.minkhollow.ca/KB/PF/index.html#defn

2. Marc Prensky. Digital Game-Based Learning. Paragon House Publishers, 2004.

3. Michael Totty. Business Solutions: Better Training Through Gaming. The Wall Street Journal, April 25, 2005.

4. John Kirriemuir. Video Gaming, Education and Digital Learning Technologies. Relevance and Opportunities. In D-Lib Magazine, February 2002.

5. Ogre3D Engine Official Site [online] http://www.ogre3d.org

Authors' Information

Olga Shabalina – PhD, assistant professor; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: Shabalina@cad.vstu.ru

Pavel Vorobkalov – PhD student; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: pavor84@mail.ru

Alexander Kataev – PhD student; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: kataevav@mail.ru

Alexey Tarasenko – Second Higher Education student; CAD department, Volgograd State Technical University, Lenin av., 28, Volgograd, Russia; e-mail: volgatav@mail.ru