An Approach to Teaching Introductory Programming for IT Professionals Using Games

Tzvetomir I. Vassilev, University of Ruse, Ruse, Bulgaria

ABSTRACT

This paper addresses the difficulties of teaching introductory programming to students in the field of information technologies and computer science. It reviews several existing approaches to make learning programming more attractive: using games or programmatically controlled hardware devices. The author believes that programming moving virtual actors motivates most of the students to learn programming. Moreover, seeing moving visual objects on the computer screen makes it easier for the students to perceive the mistakes in their code and correct them. However, writing such programs without any additional tools is not a task for a novice programmer. That is why this paper proposes an approach for teaching programming for IT professionals using a new game library, specially developed for training, which controls virtual models moving in a virtual environment. The system uses appealing 3D graphics to attract attention. Several sample programs, illustrating main features of this approach, are presented. Screenshots of the sample programs and results of teaching a pilot group of students are given at the end of the paper. Initial results show that the students enjoyed the course, most of them liked the game library approach better than the traditional one and do recommend using it in the future. More than half of the trainees improved their results.

Keywords: Game Programming, Programmatically Controlled Hardware Devices, Serious Games, Teaching Introductory Programming

INTRODUCTION

In the last twenty years there has been a noticeable interest in young people to study computing specialities like Computer Science, Computer Engineering, Information Systems, Information Technologies, Software Engineering, etc. There are two main reasons for this: learning more about computers that penetrate all human activities nowadays and finding a satisfactory job. Even in the years of world economy recession there have been very well paid jobs for good IT specialists. However, learning programming well turns out to be a hard task. In general programming courses are considered very difficult by many students and often have the highest dropout rates (Robins et al., 2003). It usually takes about 10 years of experience to turn a novice into an expert programmer. Introductory programming used to be taught using procedural languages, but more universities acknowledge that in the recent years object-oriented program-

DOI: 10.4018/ijhcitp.2015010103
ming (OOP) has become the most influential programming paradigm (Kölling, 1999-1). It is widely used in industry and almost every university teaching any computing speciality has it in its curriculum (Kölling, 1999-1). Currently introductory programming is mainly taught using C++ or Java (Pendergast, 2006) programming languages. There are also attempts of doing this with C# (Corral et al., 2014). Some research (Kölling, 1999-2) argues that the software development environment used for teaching is also important and should meet some requirements like easy to use, integrated tools, object support, etc. Our conviction is that the course content, used examples and assignments are more important than the selection of procedural or object-oriented approach for teaching introductory programming.

OOP is suitable to teach using real world objects, like actors, robots with their activities being methods like move, turn, jump, stop, etc. This paper reviews such approaches (Dann et al., 2013, Harms, 2011, Corral et al., 2014) and points out their strong and weak sides. It also proposes to develop a dedicated game library, whose main objective is to be used for teaching introduction to programming with C++. It can also be used for teaching OOP principles for students, who have already passed the course introduction to programming. The library was developed using the Panda3D game engine and utilizes appealing 3D graphics to attract attention.

The rest of the paper is organised as follows. The next section describes the related work like using games and programmatically controlled hardware devices to teach programming. The following two sections describe the proposed approach for teaching programming in C++ using a game library and give results. The last section concludes the paper and gives ideas for future work.

RELATED WORK

There have been many attempts to improve the education of IT professionals. Figuera et al. (2011) describe an educational project based on Problem Based Learning for Telecommunications students. Ryoo et al. (2008) propose a Problem Based Learning curriculum centred on game development to deliver basic object-oriented programming concepts to software engineering students. Minovic et al. (2012) analysed how to apply modern information technologies for developing game-based learning platforms for education. During analysis, they found that more research is needed in order to improve application of games in education of IT professionals. Herranz Sánchez et al. (2014) define a gamification framework to be used by software engineering professionals.

The idea of using games for teaching programming is not new. Adams (1998) describes a final project for CS1 students, in which they have to build player classes in C++ to play a game, however the main class for playing the game is provided. Becker (2001) and Lorenzen and Heilman (2002) use game programming examples and assignments as tools to increase student motivation in CS1 and CS2 courses. Leutenegger and Edgington (2007) use 2D game development in their introductory programming courses. They begin with Flash & Action Script and later switch to C++ with OpenGL. Chen and Cheng (2007) designed and implemented of an object-oriented programming laboratory course. In this course, the students are required to implement a small-to-medium scale interactive computer game in one semester, making use of a game framework.

Lu Yan (2009) experimented with using other two game-like environments for teaching Java, called Greenfoot and BlueJ. The Greenfoot system is a framework and environment to create interactive, simulation-like applications in a two-dimensional plane. It allows implementation of and interaction with objects in the context of scenarios. BlueJ has a GUI-centric design and encourages students to define classes and their relationships with an UML-like notation (Lu Yan, 2009). First the students play with objects and their properties and then they look at the source code and modify it. This approach is strongly object-oriented, and it creates objects and animations in a two-dimensional world.
A Notation for the Task-Oriented Modeling of Business Processes
www.igi-global.com/article/notation-task-oriented-modeling-business/68170?camid=4v1a

Engaging the Students in Activity Based Learning for Future Employability
Margaret Ross, Geoff Staples and Mark Udall (2011). International Journal of Human Capital and Information Technology Professionals (pp. 38-48).
www.igi-global.com/article/engaging-students-activity-based-learning/60526?camid=4v1a