

Chapter 10

Using Game Frameworks to Teach Computer Programming

Alberto Simões

Instituto Politécnico do Cávado e do Ave, Portugal

ABSTRACT

Teaching computer programming is an important task in the formation of computer scientists. Being a subject taught in the first years of student degrees, need to properly motivate students, so they try, at home, to learn by themselves, complementing that way their classes. This chapter proposes an approach to computer programming teaching based on the construction of videogames, using state of the art game frameworks. The author will show how the task of writing a game using a common framework deals with the basic programming concepts that are usually taught on a first course on computer programming, namely on object oriented programming languages like C# or Java: algebraic operations with variables, methods declaration, objects definition, objects hierarchy and multidimensional arrays. As it will be shown, even the common order of concepts presentation during the course can be kept, although applying them to computer games instead of the usually requested exercises.

INTRODUCTION

Teaching computer programming is a complex task. Not just because the need of complex reasoning needed for a student to learn how to develop an algorithm and code it, but also because of the lack of catchy feedback.

We all know that currently teenagers are in the smartphone era, used to touch in an icon and see a nice application appear. Later, in the computer programming classes, they are asked to implement an algorithm to sort numbers or strings, and print them in the console. This makes the student struggle, on whether programming is too hard, or the teacher too lame.

More recently, teachers use visual tools, like Forms. Nevertheless, we are still speaking on standard grey windows, feeling like accounting applications.

DOI: 10.4018/978-1-5225-1034-5.ch010

Trying to tackle this problem, some teachers already try to ask students to perform more interesting tasks, like the generation of images, web pages and, a minority, computer games. I am involved in teaching at two different universities in Portugal, and in both there are approaches related to programming games.

For example, in the last years, at University of Minho (Braga, Portugal), students are asked to implement a game in their functional course, when learning Haskell. The projects were first based in an implementation of Carcassonne (Schend, 2007), using mostly the console; in the next year the project was slightly different, with the implementation of LightBot (Yaroslavski, 2014) game, simulating it using X3Dom (Behr, Eschler, Jung, & Zöllner, 2009); and this year, the project was the implementation of the Sokoban (Falcon Corporation, 2016) game, using the Gloss Haskell library (Lippmeier, 2010) for simple game development. Although these projects (mostly the latest two) resulted in some interesting projects, by better students, the whole class was dedicated to implement simple functions to process strings (filters, from a board and a move, generating a new board) and only in the final assignment students really see something moving. Therefore, when those better students get to the final assignment, their motivation raises exponentially, but those we have difficulties never get to see anything fancy, and get demotivated. I still defend this approach to be better than in previous years, but it is not the best (but probably the possible when learning a language like Haskell).

In the other hand, at Instituto Politécnico do Cávado e do Ave (Barcelos, Portugal), there is a grade dedicated to the development of computer games. When teaching to those students, that are highly motivated for computer games development, I used some game programming frameworks, and it seems that their use might be a good approach to teach object oriented programming basics to first years students. And this statement is not based only on the eye candy feedback students receive, but because it is easy to focus on standard learning concepts when developing a computer game.

In this chapter I will focus on my point of view on how teaching object oriented programming can be done resourcing to a game-programming framework. To illustrate how I see it, I will use two different programming languages (although based on a same paradigm and a virtual machine): C# and Java. And for each one I chose a game development framework: the MonoGame Framework (Chamillard, 2015) (the open-source fork of XNA (Sung, 2013)) for the C# language, and LibGDX Framework (Cook, 2015) for the Java language. This way I will be able to demonstrate how basic OO programming concepts can be taught using different languages, and different platforms: from Windows native applications to Android applications.

The remaining of this chapter will be divided in the following sections:

- **Game Frameworks:** In this section I expect to answer a set of questions regarding technology. First, make a distinction between game engines and game-programming frameworks, explaining why the second are best suited to teaching programming basics. Then, I will list a set of game frameworks that are being used nowadays, and focus on their similarities and differences.
- **The Game's Main Loop:** One of the big advantages on using a game engine is that it hides the main program structure. There is no need to write a main method, explain why it should or not be static (depending on the language of your choice) or even discussing if it is an object or not. Also, the basic *'out of the box'* game already runs!
- **First Exercises:** Usually a programming language is taught starting with the concept of variables and doing some arithmetic and conditional expressions for different simple tasks. This kind of approach is also possible when writing game. Just add a sprite to the empty game and then make it move randomly, or in specific directions. Later, for conditional expressions, keyboard events

14 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

www.igi-global.com/chapter/using-game-frameworks-to-teach-computer-programming/163708

Related Content

Computers and the End of Progressive Education

David Williamson Shaffer (2009). *Digital Simulations for Improving Education: Learning Through Artificial Teaching Environments* (pp. 68-85).

www.irma-international.org/chapter/computers-end-progressive-education/8510

Designing Educational Games: A Pedagogical Approach

Stephen Tangand Martin Hanneghan (2010). *Design and Implementation of Educational Games: Theoretical and Practical Perspectives* (pp. 108-125).

www.irma-international.org/chapter/designing-educational-games/42449

Assassin's Creed and Transmedia Storytelling

Connie Veugen (2016). *International Journal of Gaming and Computer-Mediated Simulations* (pp. 1-19).

www.irma-international.org/article/assassins-creed-and-transmedia-storytelling/147349

Becoming Citizens Through Game-Based Learning: A Values-Driven, Process Approach to Citizenship Education

Yam San Chee, Swee Kin Lokeand Ek Ming Tan (2009). *International Journal of Gaming and Computer-Mediated Simulations* (pp. 32-51).

www.irma-international.org/article/becoming-citizens-through-game-based/3954

A Proposed Framework for Studying Educational Virtual Worlds

Ricardo Javier Rademacher Mena (2010). *Design and Implementation of Educational Games: Theoretical and Practical Perspectives* (pp. 65-77).

www.irma-international.org/chapter/proposed-framework-studying-educational-virtual/42446