Chapter 8

Pedagogical Mini-Games Integrated into Hybrid Course to Improve Understanding of Computer Programming: Skill Building Without the Coding Constraints

Walter Nuninger
University of Lille, France

Jean-Marie Châtelet
University of Lille, France

ABSTRACT

To improve learning efficiency in Computer Programming courses, a voluntary decision was to fully integrate different learner-centered pedagogical devices. The result is the development of a set of pedagogical serious mini-Games (mGs) in synchronous time in the classroom for a decided scenario of the hybrid course. Supported by a Learning Management System, the innovation results in a common flexible and modular framework for mGs, taking into account a really short duration and higher constraints of the training. The expected outcome is to make future end users (who will not be IT developers) aware of the potential of the underlying transversal skills developed while building up universal algorithms, stressing functional analysis regardless of specific expertise required for a given coding. The challenge is to make their knowledge ownership easier, to prevent rejection, to incent involvement and collective intelligence and further Agile method adoption with a concern for quality.

DOI: 10.4018/978-1-5225-1034-5.ch008

Copyright © 2017, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
INTRODUCTION

In order to improve the efficiency and effectiveness criteria of the training of heterogeneous groups in Computer Programming, trainers use a learner-centered pedagogical toolbox. Over the years, the trainers have integrated into the courses tutorials and practical lessons with additional digital material provided to the group. But, although the huge development of IT tools with new media and means to access contents give new training opportunities, on the ground a real inequity is observed due to cultural and social background, leading to a discrepancy in pre-requisite and prior experience. The challenge for society is to make true the motto that people, at any age, should be able to “read, write, count and code” (Bardeau & Danet, 2014). Today, the issue for Higher Education Institutions (HEIs) in a tremendously changing world is to provide the set of Digital Competences (EC, 2014) dealing with Computer Programming (CP): some of the students are not able to code, even if they are expected to be IT independent-users, satisfying the pre-requisite of the European Computer Driving Licence (ECDL) before entering the Master’s degree. But, this is not enough to be able to benefit from Information & Communication Technology (ICT) in their learning but also in their everyday life. In addition, some parts of the skills are transverse, dealing with problem solving methods, functional analysis, analysis and data mining, modeling database, collective work; a set of expectations useful to develop requirement specification in the company, identify new opportunities, communicate with software developers although they are just end-users; but proficient ones as expressed in DIGCOMP (EC, 2014). Obviously for trainees, teaching CP is not only teaching coding (i.e. learning how to code in a specific technical language) as the challenge of training CP is mostly: how to functionally build on a universal solution to a given problem, further coded in a specific context in order to satisfy the need. For most students, programming is coding, a mysterious language to master (they think); so they can spend lot of useless time making a code run instead of finding the right reply to the key question; leading to low motivation in the end. The aim of CP training is to make these specific signs understandable as there is no mystery in it, but some rules to understand and key basis to make one’s-own; the ones you will meet in any language and make you understand them all after a period of adaptation. So, the challenge for trainers is to make learners aware of the real goal, make the basic knowledge their own and find in practice some situation to develop the skills related to problem solving and collective work. In this context, the CP hybrid course is initiated, in the classroom, with a set of Pedagogical Serious mini-games (mGs). The lessons are also supported by a Learning Management System (LMS) to facilitate the immediate use of specific software for numerical computation (Scilab) in addition to upstream autonomous learning of some knowledge. The aim is to initiate learning of functional analysis (IDEF0) and algorithm design with focus on: input/output, function, control and support (the code), splitting the expertise into a specific code (which is not the aim) and the necessity of a universal solution. This one is further coded in practice as the technical solution of an upstream built-on algorithm. Within the mGs, the focus is put on function requiring input to produce output. Algorithm design requires a new talent (or skill) to assemble already given functions (or ones to be developed later) in order to solve the challenging situation. This leads to the necessity to master variables (name and content) of different type and format (string, float for instance). Basics are: read() and write() function, test, loop and then additional more complex structure like table (matrix), files…

In the following, first the context of the CP course is presented with the trainer’s motivations to face new difficulties. A course dedicated to three kinds of audience among them (Master’s degree): apprentices in Initial Vocational Training (IVT), learners in Continuous Vocational Training (CVT) and students in Initial Training (full time). Second, the standard of the mG structure is presented with a short synthesis
Integrated Brain and Body Exercises for ADHD and Related Problems with Attention and Executive Function
