PROGRAM BY DESIGN: FROM BOOTSTRAP TO UNIVERSE AND BEYOND

Viera K. Proulx [1)]

Abstract

This workshop covers all three components of the Program By Design curriculum: from Bootstrap designed for children in grades 6-8, to HtDP component that targets secondary school and early university grades, up to HtDC component that is the bridge to full scale programming proficiency. In the context of designing interactive graphics-based games, possibly played by players on different computers, children and young adults are taught systematic design process combined with introduction to algebra and formal reasoning. Supporting libraries and software allows the students to focus on the creative part of the design process. This program has been used successfully with thousand of students in a number of different settings and in a number of different countries.

Keywords program by design, game programming, problem solving

1. Program By Design

1.1. Workshop goals

The goal of this workshop is to give the participants the opportunity to learn about the three different components of the *Program By Design* curriculum, find out about the wealth of the supporting materials (software, libraries, curriculum guides, lab materials, online textbooks, and lecture notes, sample assignments, and more), but also experience first-hand the pedagogy of the *design recipe* the way students learn to use it. The pedagogy empowers the student to be the creator of the program and provides a framework for constructive creativity regardless of the underlying context.

1.2. The Curriculum

The original goal of this curriculum has been to provide secondary school students with a context for understanding and exploring the meaning of algebraic formulas – through a very simple functional programming language, with examples and problems that model the behaviour of objects in an interactive graphics-based game. The common core of this curriculum is the use of the *design recipe* – a description of the steps of the design process – that teaches students how to solve complex problems in a systematic way. Over the past fifteen years the curriculum has evolved in a number of directions.

[1] Northeastern University, Boston, MA, USA

The Bootstrap component, the first stepping stone designed for younger children, has been used with thousands of children, the curriculum materials have been designed to allow for easy adoption by teachers with a minimal training, the supporting software has been carefully designed to eliminate any stumbling blocks that do not serve the learning process. The materials are available of the website bootstrapworld.org [1], including the web-based IDE.

The main component, *How to Design Programs*, was originally called *TeachScheme!* - where the exclamation point at the end meant *not* – we are **not** teaching Scheme. The first edition of the textbook for the curriculum, *How to Design Programs* was published in 2001. The project was conceived by Matthias Felleisen with Shriram Krishnamurthi, Robby Findler, and Matthew Flatt as the collaborators. Through workshops and online support hundreds of secondary school teachers became the adopters of this program, seeing their student's problem-solving skills rapidly improving. The current version is supported by the second edition of the textbook available online [3], and the software that has been extended to teach students how to design games where the players communicate between different computers [4]. It has been adopted by secondary schools and universities worldwide, and translated into several languages (Spanish, German, Chinese, Polish, possibly more). Additionally, Picturing Programs [6] textbook and software provides a variant that presents the mathematical concepts in the context of picture manipulation.

The third component, *How to Design Classes* [2], takes the next step in the systematic approach to program design by introducing the class-based object-oriented programming in the context of Java programming language. Again, the supporting libraries [5] allow students to design interactive games, this time adding sound effects and musical background, or building GUI controls. The challenge here was to develop the software infrastructure that would support the pedagogy of the design recipe, as well as the support for the interactive game design where students could focus on the systematic design of the game behaviour.

1.3. Workshop Format

The workshop will consist of three 30 minutes segments, each covering one component of the curriculum. The first 15 minutes will introduce the curriculum, the key pedagogical points, the materials available. The second 15 minutes will be a hands-on session where the participants can experience the pedagogy and the supporting infrastructure first-hand.

References

[1] Bootstrap, "Bootstrap curriculum and materials," rev. 2014. [Online]. Available: http://www.bootstrapworld.org. [Accessed 27 4 2014].

[2] M. Felleisen, et. al., "How to design classes," rev. 2012. [Online]. Available: http://www.ccs.neu.edu/home/matthias/ HtDC/htdc.pdf. [Accessed 27 4 2014].

[3] M. Felleisen, et. al., "How to design programs, 2nd ed.," rev. 2012. [Online]. Available: http://www.ccs.neu.edu/home/matthias/ HtDP2e/index.html. [Accessed 27 4 2014].

[4] DrRacket, "DrRacket: a programmable programming language," 2014. [Online]. Available: racket-lang.org. [Accessed 27 4 2014].

[5] Javalib, "Java libraries for HtDC curriculum," rev. 2013. [Online]. Available: http://www.ccs.neu.edu/javalib/. [Accessed 27 4 2014].

[6] S. Bloch, "Picturing progams," rev. 2010. [Online]. Available: picturing.programs.org [Accessed 27 4 2014].