Processing: Programming Instead of Drawing

Experimental Use of an Open-source Programming Language within the Architecture Curriculum

Pia Fricker 1, Christoph Wartmann 2, Ludger Hovestadt 3
1-3 Chair for Computer Aided Architectural Design CAAD, Prof. Dr. Ludger Hovestadt, Institute of Building Technology, Department of Architecture, Swiss Federal School of Technology, Zurich, Switzerland
http://www.caad.arch.ethz.ch
1 fricker@arch.ethz.ch, 2 wartmann@arch.ethz.ch

Abstract. The following paper essentially focuses on the innovative use of an open-source programming language, called ‘Processing’, in the architecture curriculum and the development of a line of teaching beginning with Processing and ending with object-oriented programming in Java. This represents one creative possibility through which students are able to overcome the typically difficult step of learning a programming language and simultaneously learn how to apply it as a design tool.

Keywords: CAAD curriculum; CAAD research; User Participation in Design; Programming instead of Drawing.

Introduction - Context

In addition to the CAAD coursework required of bachelor students, our Chair offers two electives for master students and a one-year continuing education degree program Master in Advanced Studies in Computer Aided Architectural Design (MAS CAAD). Different experiments on the application of current CAD CAM techniques were conducted within the framework of these courses, spanning from the one to one construction of a computer-aided design to the programming of structures to inform design. One particular focus was dealt with in the course ‘Programming Instead of Drawing’, which has produced extraordinary results over the past few years.

In order to further the continuity and credibility of our course, over the past two years we have established the learning of the Processing programming language as a comprehensive theme from the bachelors to the masters program, all the way to the MAS. This concentration on one programming language allows our chair to professionally refine our programming expertise as well as develop a library tailored to the needs of architecture students.

Processing was our language of choice due to its simplicity, the excellent documentation available, as well as the ambition of its academic user community.
Course Content

The central task of all of the ‘Programming Instead of Drawing’ courses that we offer is integration of a programming language in order to conceive programmed design based on parameters as opposed to conventional raster-based ones: The design is no longer drawn but rather developed through a program.

The goal of the course is to familiarize the students with the novel computer-aided design process and verify its area of application within the architectural profession. The programming of the design idea allows them to generate an arbitrary number of solution possibilities and, as a result, represents a totally new manner in which to design.

Bachelor course

Within the bachelor course of studies, this theory is illustrated through a number of theoretic and practical examples. For the large majority of students, the subsequent short exercise represents their first contact to a programming language.

The presentation puts great emphasis on the pragmatic and immediately useful application of new information technologies. As a result, the conceptual formulation of the task is abstract and simply presented and has for its primary goal the realization of a simple concept, i.e. creating a program that distorts an image. What is stunning is that students are able to complete this task only after a 90-minute introduction into the computer language. The students are given a maximum of 3 weeks time to complete the presented task, during which they are able to ask questions at regularly offered consultation hours at the chair. The didactic structure of the task focuses on acquainting students with the process of acquiring their own knowledge via on-line tutorials. The lectures illustrate basic principles and demonstrate how different forums function.
Master selective course: Project ‘Programming an Artificial Landscape’. A program was developed based on a landscape design, which generates the topography as well as the path network and planting plan through parameters defined by the application of an agent system.

Master selective course: Project ‘tetris pavilion’ based on Conway’s Game of Life. In this course a spatial structure was programmed composed of simple geometric shapes (modules and empty spaces), which when combined can be allowed to grow into various structures. Student: Florian Poppele, fall semester 07.
Section 13: CAAD Curriculum 2

Master elective course
The master elective offered over one semester takes more time on the one hand, and on the other, can build on the basic understanding of the bachelor’s course. At the end of the semester, concepts are realized that focus on architectural issues such as the facade and open space planning but also comprehensive design tasks.

In order to feature the pragmatic application of Processing, the course revolves around the development of a project, which is to be subsequently programmed. Using examples from contemporary architecture, students are demonstrated the necessity of programming based on different design concepts. In particular projects that our chair has worked on with different architecture offices, such as Herzog & de Meuron, Daniel Libeskind, and Andrea Deplazes, demonstrate the spectrum and point out the necessity of programming complex structures.

MAS course
The MAS course structure is similar to that of the basic training of the master elective. The essential difference lies in the intensity and length of the course. Structurally, the course is divided into three components. In the first week, students are familiarized with the main features of Processing within the framework of an intensive mountain retreat week. The second five-week module is devoted to object-oriented programming. The course ends by making the transition to Java DIE (Eclipse), where full-fledged Java programs and applets are created. The final thesis project is also concerned with complex issues in architecture and prepares students for the subsequent Java course.

Processing
One may ask, why invest in already another programming language? Where does its greatest potential lie, and why is it especially appropriate for the integration of teaching and research within the course of studies in architecture?

Processing is essentially only a Java class language with its own IDE (integrated development environment). It was developed from a continuously growing group of people at MIT’s Media Lab and made available to the open source community.
In this manner it has continually been developed, expanded, and improved. This development was initiated by the idea to offer students, artists and designers without previous programming experience a simple introduction into programming. As a result, Processing allows ideas to be realized and prototypes to be created relatively efficiently. By utilizing a procedural introduction, the learning curve is very steep. Within a pair of seconds, graphics, text, etc. can be embedded and displayed in real time.

Once this first small hurdle has been overcome, Processing becomes a modular programming language. After a couple of hours of procedural programming, one quickly reaches an intellectual boundary after which encapsulation is required. This allows for an introduction into modular programming.

The logical next step is to better describe and model as well as integrate these concepts in other projects without requiring additional programming effort. To this end, Processing’s object-oriented programming (OOP) has the great advantage that its syntax is the same as Java’s.

**Conclusion and Outlook**

The combination of Java and Processing is ideal for implementation in teaching and research and opens a large spectrum of application areas in the field of architecture. Through the use of agent systems, many conventional architectonic problems can be seen in a new light, leading to interesting solution possibilities. The promptness and free scalability of this technique allow the program to be used independently of the computer system or the user’s previous programming experience. The results of the student projects are striking, especially since only a few possessed programming skills prior to taking the course. In order to expand the application, we are currently developing another course called ‘Processing-Pro’, conceived as the continuation of the diploma elective and focusing on deeper problems and application areas within the field of architecture.

In the research sector, various projects (Processing mobile, Wiring, Arduino, etc.) are being worked on in parallel, for example concerning the development of applications for a Java-compatible mobile phone in the project ‘Processing mobile’. The project ‘Wiring’ offers another perspective through which various hardware components can be linked to each other to create new physical installations.

**Acknowledgements**

We would like to thank all of the students participating in the Processing courses since 2006 for their hard work as well as Prof. Dr. Ludger Hovestadt, head of the CAAD Chair at ETH Zurich. Special thanks to all the assistants of the chair who have been involved in the courses and to Georg Munkel and Karsten Droste for their great support.

**References**

Shoham, Y.: 1993, Agent-oriented programming, Artificial Intelligence, 60(1).
http://www.processing.arch.ethz.ch.