Processing. Three Weekends.
Introduction to Processing as the Language for Visual Designers
Frieder Nake & Stefan Kreitmayer

PROCESSING

Outline. Schedule. Conditions
October 2009

Generally speaking, what we'll be doing in this block seminar (really: lab workshop) of three weekends, is to study the algorithmic approach to graphic design. This is a very abstract formulation, the ceiling or top level only of something much more concrete. In fact, during those six prolonged sessions, we will mainly be occupied with practical issues of programming using one particular system. The immediate goal is that you become familiar with the language and system, Processing. You develop skills, basic understanding, a certain versatility, and you should start to enjoy thinking algorithmically, and writing program code.

In focussing on the particular language, i.e. Processing, you will gain insight into programming of digital media in general. The course will constitute a practical approach to the design aspects of visual digital media. We will cover a good deal of the contents of the book

> *Processing. A programming handbook for visual designers and artists*, by Casey Reas and Ben Fry, the two designers of the system. (The book is publ. by MIT Press 2007.)

As a participant, you will be asked to work on a series of introductory design exercises building up stepwise from almost trivial to more involved designs. Each of the three weekends concentrates on one important aspect of programming. These will be: basic concepts of data structures and control stuctures; advanced data structures and control structures; interactive programs.

Each of the three Sundays will in part be dedicated to a restricted project in visual design. The inventors of Processing, the designers Casey Reas and Ben Fry, call pieces of code, „sketches". Programming in Processing should, therefore, be considered analogous to sketching in visual design. What you come up with in such an exercise is a sketch. When you sketch, you use your previous results to come up with better, more advanced, more complex designs. The sketching helps you first to develop basic, but then more and more advanced techniques until you reach a point where you switch to more difficult materials and tools.

Parallel to our simple design work, we will inject systematic studies of algorithmic thinking. We do so by identifying components and structures of algorithmic descriptions. For this goal, we will often follow the Reas & Fry book. Other sources will be introduced into the course as well. In closing, we may make an attempt to pull together the practical design and technical programming experience in a theoretical perspective on digital media. But emphasis will always be on the practice of algorithmic design work.

Some of you are taking courses in parallel where, I hope, you can use what you learn here. To our good advantage, Stefan Kreitmayer was endowed with the job of a tutor to improve your learning situation. This will allow us to support you much more directly and immediately.

It was decided that you do not get credit for this course as it is not really a design course although offered at HfK. We will live with this condition, I guess. The path to an independent study is still open to you.

Time Schedule

The following schedule gives the dates and rough purpose of each of the anticipated meetings. In the actual depth and breadth of treating topics and subtopics, much will depend on you, the participants themselves. Insofar, this schedule is no more than a first indication of possibilities.

Date	Topic	Remark
21 & 22 Nov 2009	Basic structures of programs: data & control flow. Variables, constants. Data types. Coordinates. Arithmetic. Declarative and operational statements. Assignment, condition, iteration. State machine. Basic drawing functions. General structure of a sketch.	We start immediately into first examples. During the first day, you already write your first program. A first small course project on Sunday.
28 & 29 Nov 2009	Advanced structures of programs. Animated elements. Structuring programs. Functions, parameters. Array. Saving images.	Randomness. Geometric transformations, matrices, stack. Second course project
5 & 6 Dec 2009	Interactive programs. Input. Recursion. Interactive devices: mouse, keyboard. Event handling. File import & export. Wrapping up, conclusions. Critique. Definition of independent study projects for credit.	Object-orientation. Final course project

On each of the six days, we will meet from 10 to 17. This time-span will be divided into the morning session from 10 to 12, lunch break from 12 to 13, and two afternoon sessions from 13 to 15, and 15:30 to 17, with a break between. We will, however, be flexible as conditions require.

On Sundays afternoons, you will be asked to work on assignments (short course projects).

Please, be on time! Since the building must be closed on weekends, we must organize getting in and out.