



## General Outline of the Workshop

16 April 2011 (updated in May and June)

After a bit of confusion, it was decided that this workshop will be in English. Its three weekends will be spread out a bit.

Generally speaking, what we'll be doing in this block seminar (really: lab workshop), is to study the algorithmic and generative approach to graphic design on an introductory level. This is a very abstract formulation, the ceiling or top level only of something much more concrete. In fact, during our sessions, we will be mainly concerned with practical issues of programming using the Processing system. The immediate goal is that you become familiar with the language and system of Processing. You develop skills, a basic understanding, and a certain versatility. You should start to enjoy thinking algorithmically, and writing program code. And that's what turning vague ideas into precise programs is all about.

In focussing on the language of Processing as a special case, you will gain insight into programming of digital media in general. The course will constitute a practical approach to design aspects of visual digital media. We will cover some parts 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 was published by MIT Press, 2007.)

As a participant, you will be asked to work on a series of introductory design exercises stepwise building up from almost trivial to more involved designs. Each of the meetings concentrates on one important aspect of programming. Two weekends will be dedicated to basic structures (static and dynamic) of programs; the third will be on object-oriented programming.

The inventors of Processing, the designers Casey Reas and Ben Fry, call pieces of code, „sketches“. They want that programming should develop into a skill considered to be analogous to what is known in visual design as sketching with pencil and paper. The language and system, *Processing*, is their suggestion to go this direction. When you sketch, you use your previous results to come up with better, more advanced, more complex designs. 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.

In summary, this course and workshop is about a new way of thinking (the algorithmic way: vague and precise, at the same time). The course expects bloody beginners more than proud cracks. Those are most welcome who are afraid of programming, those also who believe they can never do this. You can! You will enjoy it.

summer term 2011 | 2 sws | lab & workshop in 3 blocks | for students of Digital Media (Bachelor or Master), as well as others  
M-107 or independent study possible for credit, based on work in this course and special arrangement  
limited to 12 participants | register by email | bring your laptop | download Processing from [www.processing.org](http://www.processing.org)  
meetings: 16 April (11-18), 17 April (10-17), 15 May (10-18), 18 June (11-18), 19 June (10-18), 25 June (11-18), at HfK 3.09.100

Frieder Nake | [nake\(at\)informatik.uni-bremen.de](mailto:nake(at)informatik.uni-bremen.de) | (0421) 218 3525 Linzer Str. 9a, room 3015  
office hours Wednesday 12 to 14 Linzer Str. & Tuesday at HfK upon arrangement | [www.agis.informatik.uni-bremen.de](http://www.agis.informatik.uni-bremen.de)

### Conditions for credit

You may gain 4 credit points from this workshop when you do an independent study. If you choose to do so, we draw up an agreement. It is about your topic, the module, the required results, and dates.

In general, what you will be required to do is this.

*You choose a specific topic. You develop a sketch of some complexity up to running stage. You present the sketch in class at our last meeting. You write a short paper (5 pages) about your work.*

The topic you choose (and on which we agree) must be aesthetically interesting. It must allow for an interactive exchange. Its code must be well parameterized, concentrating the outside influences in one block of code. It must be well documented and structured. It must run safely.

If you want to gain credit, you should follow this advice:

1. Send me an email before 1 May 2011 indicating what you want your topic to be. Try to give it a title, tell in a few lines what you have on your mind.
2. I will react with questions and suggestions.
3. Send me a one-page outline of your project until 13 May 2011.
4. From this, we draft the agreement.
5. Your presentation and submission of the paper will be on 25 June 2011 (or another day around this date).

## Schedule

The following schedule lists the dates, times, and topics of our meetings. The actual depth and breadth of treating topics and subtopics will largely depend on you, the participants themselves. Insofar, this schedule is only a first indication of possibilities.

All meetings are at HfK, room 3.09.100.

Date	Topic	Remark
5 April 2011 13 to 14	Preliminary meeting, only for organizational matters	
Weekend 1 16 April, 11-18 17 April, 10-17	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. Advanced structures of programs. Simple animation. Structuring programs. Functions, parameters. Array. Saving images.	Randomness. Geometric transformations, matrices, stack. Lots of examples, individual work
Weekend 2a 15 May, 10-18	Interactive programs. Input. Recursion. Interactive devices: mouse, keyboard. Event handling. File import & export. – Ideas and agreements for project work	
Weekend 3 18 June, 11-18 19 June, 10-18	Arrays, data types, and data structures. Interaction once more Functional abstraction in detail. Declaration and use of functions, parameter matching, side effects. Some 3D graphics examples	
Weekend 2b 25 June, 11-18	Objects and classes. Object-oriented programming Conclusion & future work	
Wednesday 13 July, 12-15	Extra meeting: presentations for credit. (Deadline for reports is 20 July.)	

Weekends are scheduled as Saturday from 11 to 18, and Sunday from 10 to 18, including breaks. We may need extra time for presentation of individual projects done as independent work for credit. becomes possible. This is left open. The weekends will be Saturdays from 11 to 18, and included.