Introduction

Programming as a medium and design tool in its own right has become commonplace in today’s architectural design. Movement towards Building Information Management (BIM), parametric design and rapid prototyping all suggest a strong coupling between the actual architectural data and the algorithms that produce them. The Build The Code Workshop held from the 7th to the 10th May 2012 in Bialystok has gone in exactly this direction (also see Outcomes):

- Automatization of common work tasks by means of Visual Basic for Applications (VBA) for AutoCAD was shown, using sketched designs of the course participants as input.
- The course goal was to have an algorithm design a building, according to the (observed or real) rules governing it. The so-generated city of buildings was discussed afterwards, leading to the question of “what is creative?”

The workshop’s main layout was to give first two days of theoretical work, followed by practical application in the abovementioned manner. Beside the actual teaching, parameters regarding the quality were measured, based on previous work done (see Measuring Quality).

Workshop Outline

Students learned how to build up architecture in a structured, floor-by-floor manner. As general concepts, variables (Integer, Double, Boolean, String, Arrays, Objects), Conditionals (if, Then, Else), Loops (While...) were taught, furthermore, Iterations and Recursions were used for building up geometry. From the results (see Fig. 1), we observed that the students thought mostly additively (no boolean interaction/subtractions) when building up architecture. The linear fashion of building floor-by-floor was not always followed, and furthermore, we have seen also approaches which would not do qualify as “well”, “floor” or anything out of the usual (modernist) repertoire, which is good.

The question remains whether this type of generation is producing „architecture” or is merely automatizing. Coding as a medium/tool for creative exploration was covered in an ending discussion on “what is creative?” As the architecture faculty of Bialystok is yet lacking an algorithmic design course and/or specialization, this might contribute to future teaching efforts to be conducted by the course participants.

Measuring Quality

During the actual lecturing work, the performance of the lecturers was evaluated using „mood charts”. The students had to chart their mood, the quality of presentation and comprehensibility over time, using a fivefold scale (very good, good, average, poor, bad). Participation in the survey was done anonymously, on a voluntary basis. Every student that wanted to participate left his/her „mood chart” on his place when leaving the class. The lecturers could then collect the samples and correlate the so-indicated values to the subject brought at that instance (e.g. [1a] topic: function, average mood: good, average comprehension: good, average presentation: excellent).

From a conceptual point of view, this feedback can give answers to a large body of question which lecturers may pose themselves, e.g. (and also referring to Fig. 2):

- Was the topic presented clearly and precisely?
- Was the presentation too fast/too slow?
- Has the mood of the class shifted in the course of the presentation, and why (lunch break? general comprehensibility?)
- Was the sequence in which the (closely related?) topics were brought adequate.

The mood chart results (see Fig. 3) indicate a general overload of information, going hand in hand with poor mood and presentation on the second day of the theoretical part. More time should have been invested in bringing the subjects, perhaps at the cost of an additional break in between (workshop days were from 9am to 5pm with lunch break of an hour). Clearly, the subject was also itself considerably hard to learn, since it involved a lot of mathematics (on the one hand), and learning the syntax of programming (string).

We also observed that, in contrast to similar workshops conducted before (Wurzer, Alacam and Lorenz, 2011), there was no mood decline after lunch break. The reason for that may be cultural or based on the „studio-like” environment in which the course was performed, leading to a more relaxed atmosphere than in a classical lecture hall. The architecture faculty of Bialystok is yet lacking an algorithmic design course and/or specialization, thus, such a „studio-embedded” programming course might contribute to the teaching efforts already in place.

References


Acknowledgements

Academia depends on collaboration. In this context, we wish to thank Alexander Asano-wicz, Adam Jakimowicz, Malgorzata Stolarewicz and all course participants for their effort. Furthermore to the vice-dean for teaching and students’ affairs, Jadwiga Zarnowiecka, for exami- ning the outcomes during the end presentation.

Academia also depends on financing. Our travel has been made possible by Vienna UT’s foreign relations department, under the ERASMUS lifelong learning programme of the European Union.

Figure 1: Programming Exercise. Building a city out of housing blocks created by the course participants in VBA for AutoCAD 2012 (Student Edition). The created designs were programmed after a two-day theory workshop (1).

Figure 2: Feedback Loop. Students give a vector of subjective impressions, which are correlated with the topic presented at that time. The results contribute to the next iteration of the lecture.

Figure 3: Mood Charts. Actual measurements of student mood, comprehensibility and quality of presentation for day 1 (upper figure) and day 2 (lower figure) of the introductory programming course.