Easy access classes for three-dimensional generative design

Using a collaborative environment for e-learning

Thomas Grasl, Christoph Falkner, Christian Kühn
Institute for Architecture and Design, Vienna University of Technology, Austria
http://www.gbl.tuwien.ac.at

Part of an EU funded project to develop a "VIrtual campus for virtual space design Provided for European Architects (VIPA)" was the implementation of a practical run at the Vienna University of Technology. Therein we attempted to evaluate some of the concepts and technologies which were intended for the e-learning platform. After briefly introducing the didactical background, this paper concentrates on the technological setup accompanying the course. Especially the use of Croquet as an immersive three-dimensional environment to teach generative design is highlighted; its strengths and weaknesses in supporting our envisioned didactical concept are analysed.
The practical run and its evaluation by the participating students are described, as well as some of the student work performed during and after the course.
Concluding remarks elaborating on problems encountered in the software setup and in our didactical concept, followed by the description of future work to amend the above mentioned pitfalls, will mark the end.

Keywords*: architecture; collaborative environment; croquet; generative design; learning platform, virtual space design.*

Situating the project

e-Learning has long since lost its magical aura; apparently it's the content and not the media (Rovai, 2002) that has the greatest impact on our ability to learn. In this spirit was our attempt, granted to us by the European Union, to create a novel learning platform trimmed towards architecture students. The primary target of the ongoing project is to establish a "VIrtual campus for virtual space design Provided for european Architects (VIPA)", which can later be extended to serve an arbitrary number of interested faculties.

The target audience was defined to be students interested in architecture and several neighbouring disciplines such as virtual space design, gaming design, generative design and 3D modelling. We were determined to implement ideas form the various learning theories such that they could unfold their greatest effect (Ally, 2004). Behaviourist ideas were used for learning hard facts and to implement self enclosed learning units recalling prerequisites, cog-

nitive theory guided the design of most autonomous units, so far primarily organized as a linear sequences of information, and constructivist methods were to be used while designing some of the more open, unguided knowledge repositories and especially the practical examples in the collaborative three-dimensional environment.

Furthermore we incorporated ideas put forward by constructionism as described by Papert (1991) and Resnick (1994), as these seemed to fall together nicely with the character of most of the material being taught on the platform. Especially the constructionist learning-by-doing approach was intended to guide the more computationally orientated courses, which were envisioned to play a major role within the VIPA curriculum. In order to provide an easier entry for students and educators it was decided to start the VIPA project primarily based on blended learning courses, leaving the option open to integrate full e-learning courses at a later stage.

Setup

For the first test-run the software setup was based on two open-source packages. For the more straightforward requirements of an e-learning platform a Moodle environment was installed. It was to deal with the user and curriculum management, to provide basic tools for creating content, revisions and examinations as well as serve as a communication centre and a resource repository.

Additionally we were looking for an immersive and collaborative three-dimensional environment, which could be used for interactive demonstrations, as a mode of communication and as a development platform at once. A market research executed by one of the consortium partners uncovered an array of possible products, most of which are mainly intended as gaming engines. However our previously compiled quality criteria laid high emphasis on three points. The licensing costs were to be held low, as the platform was to be scalable and participation should not be hindered by financial considerations. One of the programming languages used should be suitable for novice programmers, which most of our prospective students would be. The package and especially the integrated development environment (IDE) should not be overly complex as to allow the students to install and work with the software on their private computers. After weighing available products by the above mentioned criteria, Croquet established itself as the best choice.

Compared to most products we encountered Croquet boasted a whole array of advantages. Croquet is a collaborative, multi-user online application, thereby enhancing the sensation of collaborative learning by transmitting the well known computer lab situation, several users sharing a computer and

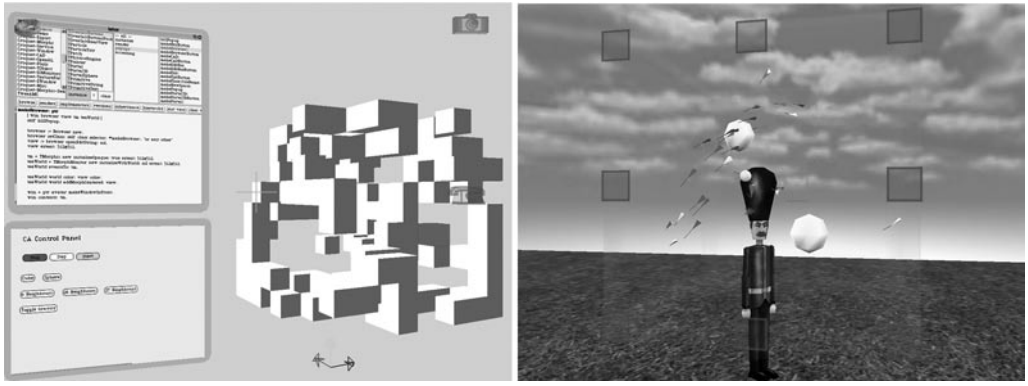


Figure 1
Croquet: CA application with code editor and two avatars viewing flocking boids

thereby augmenting the learning experience, into virtual space (Lombardi, 2005). The possibility for several users to manipulate the code for the three-dimensional environment from within the environment in a collaborative manner was especially compelling. The whole product is built upon Squeak, an implementation of Smalltalk, an object-orientated programming language well suited for beginners. Indeed, Croquet, as well as Squeak, is primarily targeted at the educational sector. Squeak comes with an integrated development environment, with which one can not only delve into the details of Croquet, but of the whole Squeak platform, right down to the most basic kernel classes. Another benefit, and a major point of consideration while implementing such a project, is the level of support which can be expected from the user community and other web-based resources. And finally, Croquets distribution is based on an open source licence.

Both Croquet and Squeak pride themselves as constructivist learning environments. This claim is strongly supported by the openness of the architecture and the ease of Squeak as a programming language. These were also the main paths upon which we wanted to guide our students towards an understanding of generative systems.

We had intended three levels of interactivity. The first levels should require no prior knowledge of Squeak nor of the system to be studied, the student interacts with existing applications via a user interface, initially allowing only for interaction anticipated by the programmer. The second level involves the manipulation of certain elements of the application. For example, the cellular automata application consists of numerous classes, for a novice programmer however it is sufficient to know where and how to manipulate the rules governing the automatons behaviour and maybe redesign the visual representation of the cell in the three-dimensional world. This can be done by generating a new subclass of the class cell and by overriding some of the base class behaviours. The student requires a certain level of Squeak knowledge, though the classes should be written in such a way that the essential functionality can be changed without prior knowledge of the system as a whole. Finally, the third level requires a sound understanding of Squeak and Croquet. An intermediate programmer might decide to create a new overall structure for the individual cells or indeed it might become necessary to extend the CA engine itself, allowing for new levels of interactivity. Alternatively the student might develop a new application altogether, thereby simultaneously allowing the VIPA platform to progress.

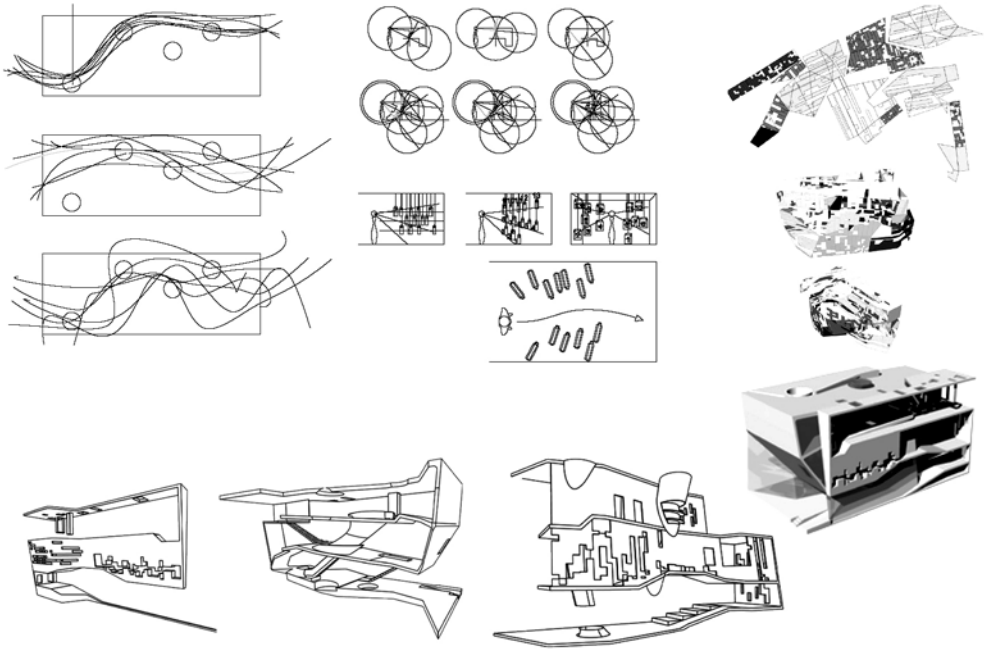
Practical run

Within this larger framework an actual course at the Vienna University of Technology was chosen within which a test-run was performed. The course "CAD and planning methods", held between December 2005 and February 2006, is part of a conglomerate of courses forming a module directed at students interested in generative and computational design. It is weighted with 2.4 ECTS points and is not compulsory. The main focus was on teaching a methodical approach to design as well as necessary insights into the history and workings of several generative systems.

Though the 12 participating students where expected to realize a project utilizing the newly learned methods within the course, it was not primarily a design course. Instead it aimed at providing the students with a set of design methods, to be used during other courses of the module or in later semesters. Assignments were worked out together with the student once an interest in a specific field could be made out. The only criteria were that the problem must be clearly defined in a first step and the project must result in a generated solution in two or three dimensions. Figure 2 shows the diploma project of a student who absolved the course previously and then went on to utilise the ideas and methods for his final project.

Not all chapters of the course were transformed into e-learning suitable elements using Moodle and/

Figure 2
Diploma project: "Concept Store" by Paul Kweton



or Croquet. The main topics presented in this fashion were cellular automata, pathfinding algorithms and autonomous steering behaviours based on Greg Reynolds Opensteer library. Other fields, not implemented for this practical run, included genetic algorithms and neural networks. The primary aim of the course was to introduce the student to systematic, computational design approaches. Emphasis was laid on developing a methodical approach to solving problems with the intent to strengthen this strain of thought, be it in a computational environment or not. Of course trying to express ones ideas in terms of a computer program lends itself well to such a target. It was however left to the individual student whether he/she takes advantage of a programming language to implement their project. In retrospect it becomes clear that the programming orientated group made better progress in systematic design, thus perhaps suggesting to make the use of a programming environment compulsory for future courses.

Evaluation

At the end of the course an evaluation questionnaire, prepared by the consortiums e-learning expert, was distributed amongst the students. The main topics of interest were "Didactical and pedagogical criteria", "Technical criteria", "Administrative, contextual and organisational criteria", "Content criteria" and "Design and usability criteria".

From the didactical point of view most participants missed a clear definition of requirements; both the overall assignment and the autonomous self-assessment units must be defined more rigidly to provide the student with a clearer guide-line. In contrast however, the motivation to participate in the course and experiment with the platform received good ratings, with the sole critic that communication among the students and with the tutors could be better.

The evaluation of the technical criteria is more differentiated. Robustness was, mainly due to the

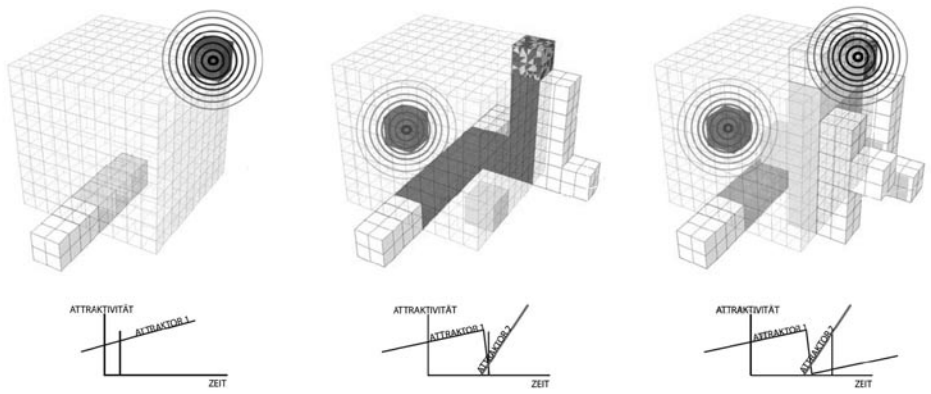


Figure 3. Student project: "The extended volume" by Elke Kronsteiner

early release of Croquet adopted, seen to be extremely bad. The conformance to standards, such as a consistent style of programming, received mixed ratings. It seems this point was influenced by the participants own programming skills.

In the administrative and organisational category the response was uniformly positive with respect to the clear definition of the user groups and the context to other training units, such as "Programming in Smalltalk".

The assessment rules were not defined clear enough, the students were unsure of the extent their final deliverable must have and which criteria will be used in the evaluation.

Most participants were very satisfied with the content of the course and the selected teaching method and deem the course with its current preparation of content suitable for e-learning. However, that there is room for improvement on the technical presentation of content is shown by the moderate ratings given for "Efficient selection of used media".

Overall the formal instructions for use of the platform were seen to be comprehendible. Also the individual support was evaluated positive as was the possibility to adapt the 3D laboratory tools. Most participants would like to implement the ideas and/or tools, they were confronted with during the course, in future projects.

Conclusion

Problems encountered

On the technical side most problems encountered were a result of the chosen platform. This is not to say that a wrong decision was made, but as an early adopter of technological innovation stability and usability can pose an initial problem. Development had started using the Croquet release 0.3, while many of the short-comings were recognised immediately we had hoped for a more rapid progress of Croquet. The main points of concern were the accessibility of the platform for a novice user used to Microsoft Windows functionalities and the possibilities to provide a coherent interface within the Croquet environment. The squeak developers follow a rather different philosophy concerning user interface design than most current software packages; this can become daunting for new users. Some of the pitfalls can however be amended by simply changing the default behaviour of the system. More serious were the shortcomings Croquet displayed for creating and using interactive menus within the three-dimensional environment, thus making it difficult to provide our envisioned first level of access, which is entirely dependent on this functionality.

In the mean time Croquet release 1.0 beta has been published. Many problems have been amended; especially stability and performance have been

improved considerably. However before extending our efforts adapting Croquet we are going to wait for further releases, covering more of our requirements at a satisfactory level. Especially some of the pitfalls concerning the user-interface have also been acknowledged by the developers (Smith, 2006) and are being worked on.

A didactical difficulty encountered is interconnected with Croquets programming language Smalltalk/Squeak. Squeak, although suited for novice programmers, has not reached very broad distribution. Understandably most students would like to see their efforts learning a programming language awarded with a more universal tool. As developers of the VIPA platform it is clear to us, that the user interface will never be able to handle all necessary interactions, nor should it, as programming is an essential element of the didactical concept. It is thus necessary to enhance the attraction of Squeak. We are convinced that as the platform develops and the number of courses using Croquet/Squeak grows, there will also be more incentive for students to learn Squeak. Additionally to Croquet Squeak also offers compelling possibilities in two-dimensional and non graphic applications which could also be used in a number of courses.

Future Work

In order to enable the students a gentler approach to the matter at hand, it would be advantageous to reorganize the module in a more linear manner. Thereby allowing for more time to get acquainted with the platform and encouraging incremental steps in programming skills. The repository of tools and lectures must also be extended to reach broader acceptance amongst the students and the faculty. Additionally we have realised that self contained exercises offered parallel to the lectures, constitute well defined short-term learning goals, thereby enhancing motivation and simultaneously supporting the learning curve in an autonomous manner.

As an attempt to extend the tools in the repository and to evaluate an alternative to the Croquet platform the next test run will be performed using Blender enhanced by the newly developed Verse module. It will provide similar features to Croquet, albeit in a less immersive environment. Additionally we are hoping to benefit from the greater stability and possibilities offered by a renowned 3D modelling package. While a collaborative and immersive setting still has a high priority on our list of quality criteria, we are willing to incorporate other aspects into the overall package as well.

Acknowledgements

Much of the basis upon which this project was founded was created as a joint effort by the VIPA consortium. Accordingly we would like to thank the members: The architecture departments at the universities of Aalborg (DK), East London (UK), Lubljana (SL) and Vienna (A) as well as our industry partners adm (Atelier für digitale Medien) (A), E. Gabriel - Immersive Systeme (D) and the FH Joanneum (A). Of course we would also like to acknowledge the support we have received by the European Commission.

References

Ally M.: 2004, Foundations of educational theory for online learning, in T. Anderson and F. Elloumi (eds), Theory and Practice of Online Learning, Athabasca University, Athabasca, CDN.

Lombardi M.: 2005, Croquet Learning Environments, http://www.opencroquet.org

Papert S.: 1991, Situating Constructionism, in I. Harel and S. Papert (eds), Constructionism, Ablex Publishing, Norwood, NJ.

Resnick M.: 1994, Turtles, Termites, and Traffic Jams, The MIT Press, Cambridge, MA.

Rovai, A.: 2002, Building sense of community at a distance, International Review of Research in Open and Distance Learning

Smith D. et al: 2006, Filters and Tasks in Croquet, http://www.opencroquet.org