

Real Virtualities - Architecture 2.0

Using Second Life for an Architectural Design Course

Jochen Hoog, Manfred Wolff-Plottegg (Й. Хоог, М. Вольф-Плоттегг)

Institute for Architecture and Design, Vienna Technical University, Austria

<http://www.gbl.tuwien.ac.at>

Introduction

Technology to create three dimensional virtual environments has become extremely powerful and affordable within the last years. Virtual worlds provided through the WWW, like Second Life, There, Open Croquet or multi player computer games are constantly growing. The conceptional design of these new virtual computational spaces could open new opportunities and possibilities for architects; but the specific rules and characteristics of this kind of spatial software environments are not yet discussed or even explored by architects. It is a new field to be reflected and occupied by architects.

The use of an avatar based multi-user virtual environment (MUVE) like Second Life as a place, hyper media and collaborative platform offers new ways of re-thinking the process of designing architecture.

Virtual worlds combine aspects of e-Learning, collaborative work, design with algorithms, CAAD and user generated persistent environments. Using MUVE's for the architectural design process shifts the way how architecture is being generated and designed. We want to address this issue with research, by introducing students to "spatial" software environments, like Second Life (SL). "(...) the visual culture of a computer age is cinematographic in its appearance, digital on the level of its material, and computational (i.e., software driven) in its logic." (Manovich 2001).

The virtual environment should act like a program, a place, and as multimedia. The goal of the design course is to explore the possibilities of creating virtual architecture, which means to generate, to program, and to script virtual environments.

The main focus of this paper is to stress and to describe the differences of the student's results compared to common architectural design courses.

Method

The design course "Second Life - Architecture 2.0" (winter semester 2007/2008) as well as the currently running course "Scripting Architecture" (summer semester 2008) are meant to be practice based research. Second Life is used to be the software environment where the projects have to be realized. The following methods and results describe the course given in ws 2007/8 where the projects are finished.

Compared to traditional design courses we did not give any architectural functional specifications, except for, that the students had to explore and to push the limits of the given virtual environment of Second Life. Beyond that there was no specific architectural design task. It was up to the students to develop an individual design approach, to define a project theme, a title, and a program. In that way the instructions to students were shifted from functionally focused instructions to operational ones, thus challenging the way of architectural thinking. Also the question if the results should be interpreted as real or virtual architecture was opened to discussion. We tried to use the computer as a media and software environment to create new architectural algorithmic design approaches – to use SL to teach architectural design.

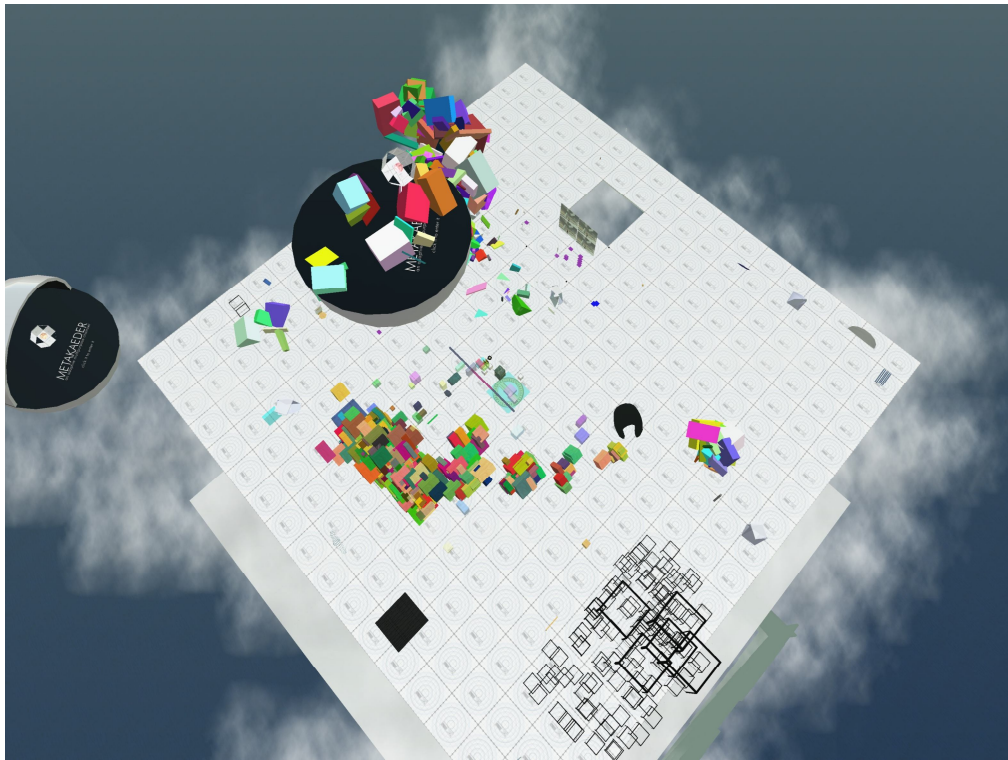


Figure 1. TU Vienna Island in Second Life

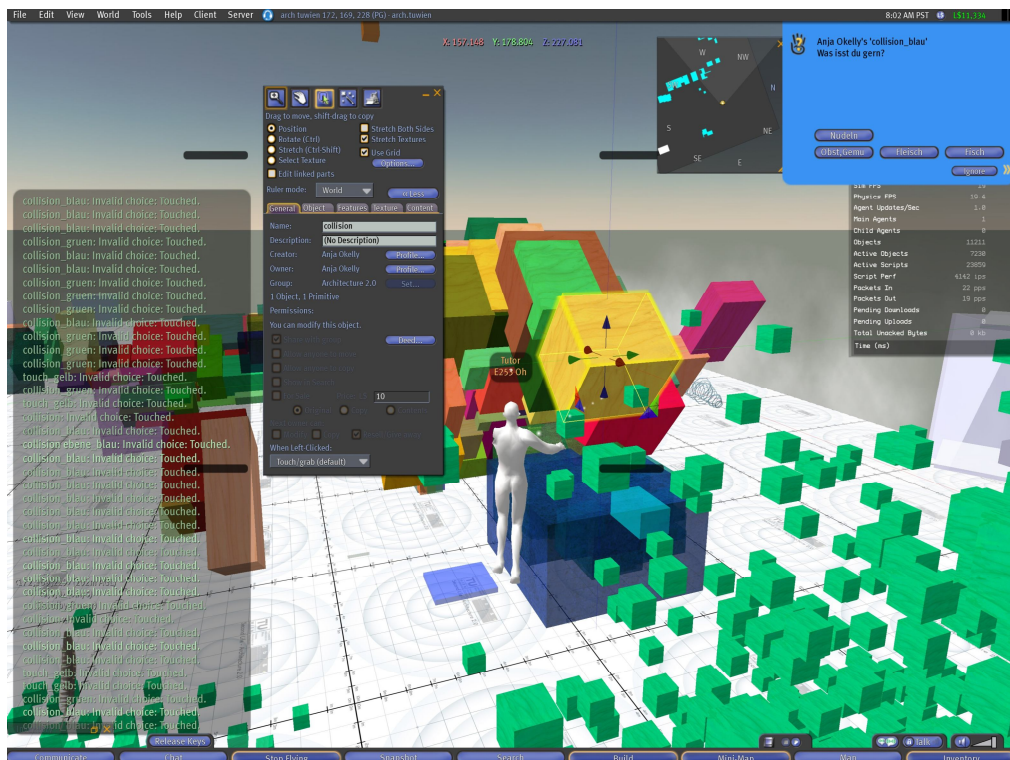


Figure 2. Object property panel, SL GUI

The environment of SL should be used as a methodical, conceptional approach for an autonomous architecture, like the “binary house” (“Binäre Haus” – cf. Plottegg 1988(1996)).

“The result of an algorithmic design process is open. The input and the rule may be well-known, but the result, the output is for now unknown. The breakup of common, historic, traditional algorithms, which evidently direct to pauperization, can bring up new frontiers and solutions (limes-pictures), even for architects.” (Plottegg 1988(1996) – translated by the author)

“Der Ausgang einer algorithmischen Prozedur ist offen. Der input kann bekannt sein, die Handlungsanweisung kann bekannt sein, aber das Ergebnis der output kann vorerst unbekannt sein. Die Auflösung bekannter, historischer, traditioneller Algorithmen, die nachweislich zur Verelendung führen, kann zu neuen Limesbildern führen, zu von Architekten selbst nicht geahnten (Limesbildern und) Lösungen.“ (Plottegg 1988(1996))

The students were given a short introduction how to use SL, how to move the avatar, how to script, and to create simple objects. A “group” (a sort of newsletter) in SL was used to provide information, to communicate asynchrony, or just to see who is online. Additionally a blog (<http://today.tuwien.ac.at/secondlife/>) was installed as an online platform, where every student had the possibility to present results or to provide information. We requested the students to look for well finished scripts or projects in SL, to reuse them for their projects and to share everything they found or made on the blog or within SL. The idea was to think and act “open source”.

The course was given bi-weekly. One week as a face-to-face presentation, the other week online e-Learning lessons only within SL were offered.

Technical Framework

The virtual island in Second Life (<http://slurl.com/secondlife/arch%20tuwien/88/108/216/>), which is indeed server space, is a white plain (256 x 256 m) where the students have unlimited access. SL is a virtual and persistent online platform, perceptible through the internet by avatars. Currently, SL provides simple geometric objects (prims), which can be (only) generated within SL. There is no way to import or export objects. All the objects have properties like size, rotation, position, and behaviours like locked – unlocked, physical – non physical, temporary, and phantom. It is possible to add colour, textures and assign scripts to objects (prims). The scripting language (Linden Script Language) is event driven and C/ java style. An island in SL is limited to 15'000 prims, and the amount of running scripts defines the frames per second (FPS) which is noticeable as the speed or bandwidth of the island. SL provides chat, asynchrony text messaging and VoIP.

Student´s Projects Results

The course “Second Life - Architecture 2.0” was given to fourteen architectural graduate students, who had mostly no experience in scripting, and one student of computer science. In the current course “Scripting Architecture” twenty people participate.

Despite the initial concerns about Second Life, due to the media hype, the students accepted SL and got used to it very fast. Right from the beginning they worked on the island as virtual neighbours. The island became the site, where all the projects were built and presented. The blog was used to provide information, links to tutorials or interesting topics, while it was used by the students to present screenshots or conceptual descriptions of their projects. The face-to-face presentations were better accepted than the online meetings in SL.

During the presentations the lecturer was on the island with his own avatar, like all the students, testing and exploring their results in real-time, while discussing the individual projects. The online presentations were slightly less popular; however it was always possible for everybody to take a stroll through the island, seeing who was working on his project. The

students worked on the same ground and helped each other in case of problems. At the end of the semester the island got slower, because of the amount of prims (11'326) and running scripts (23'974).

Most of the projects dealt with issues on interaction. All objects within SL were capable to interact with each other (collision detection). They communicated via channels and could exchange simple states or properties. They could also send text messages to avatars or could be programmed to be controllable by avatars.

Some student results:

Umadam Voxeln (Lukas Antoni) (Fig. 3)

The project provides a kind of small reactive landscape, which interacts with the user by building up some small hills. The user in SL finds only a black rectangle. When he enters it he gets pushed up by columns, which changes also its colour; by walking around the user forms a kind of dynamic landscape. When two or more users enter it, they compete against each other.

Invisible Architecture (Natalia Dziadus) (Fig. 4)

The project deals with different properties of an object concerning the visibility, size, and walkthrough behaviour. A big amount of cubes are hidden invisible on the island. If the user enters the invisible barrier, he sees a randomly changing environment of cubes, changing their colour, visibility and the possibility to enter it or not. The user is inside a constantly changing cubic environment. The project deals with questions of different possible perspectives of the avatar within SL and changing properties.

Virale Partikelwolke (Marie Prammer and Iris Ivera Cerny) (Fig. 5)

The students want to create a kind of living virus, which sticks to existing objects and avatars, by reproducing itself. The result is in that way very successful because the running virus is so agile and it reproduces itself so fast, that the island collapses. The students were warned of Second Life, and they managed to get control over their living result.

Dream Object (Anja Fiebig) (Fig. 6)

The avatar is in a dialog with objects, which seem to react to the answers he gives. The project has a starting spatial arrangement and is constantly changing over time, due to the answers of all visitors.

The other projects deal with creating paths of avatars, reacting walls, multiplication of fractals or providing space. It has to be mentioned here that the fact that all objects have rights of ownership causes some problems. It was hard to help the students with their scripts, when they forgot to share it with the group or the public. Also the idea of open source was not as well accepted as intended. No project dealt with sound, or with commercial concepts.

(See all the results at: <http://twoday.tuwien.ac.at/secondlife/>)



Figure 3. "Umadum Voxeln" by Lukas Antoni

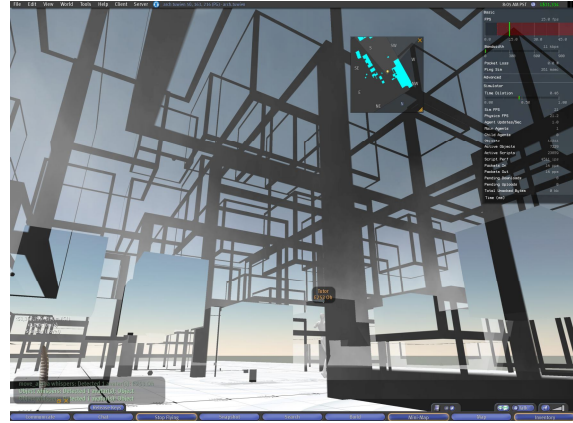


Figure 4. "Invisible Architecture" by Natalie Dziadus

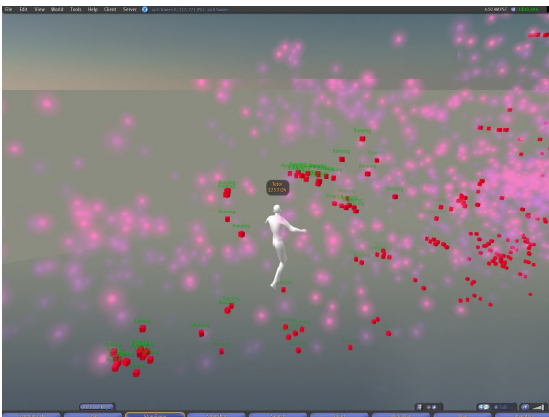


Figure 5. "Virale Partikelwolke"
(Marie Prammer and Iris Ivera Cerny)

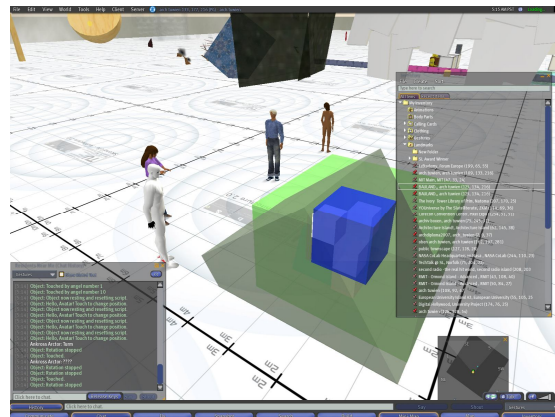


Figure 6. Conversation with students

Discussion

Concept – Pseudo-Script – Code - Running Code

The course is meant to be a design approach into a new architectural field, by doing project based research. The results show architectural ways of entering 3d virtual worlds. The goal is to explore the new role of an architect in a virtual environment, to open the black box of programming, to foster the understanding of the computer, and to learn the basic principles of a computer language, situated in a 3D multi-user virtual environment. Despite the fact that SL looks like old computer games, it offers simple and quick ways to script its environment. The students have to really focus on the code that makes things going, instead of using predefined manipulations, like in common CAAD software or 3d modelling programs.

The results are active virtual 3D environments, which can be experienced by the avatar. They are "built", like real architecture and not visualized by well rendered perspectives, drawings, or models. Still they are virtual.

One of the most interesting results is the spatial unintended overlapping of different results. Because of the narrowness of the island the projects sometimes penetrate each other, building an unexpected new situation, which clearly never happens in other courses. The results interact with each other, thus the user on the island experiences one big environment.

The students are real virtual neighbours, class mates, users and constructors at the same time and the same place. Remarkably, most of them chose a special area on the island and stay there with their projects during the whole semester.

Second Life provides a stable environment to be used for students to gain experiences with algorithmic architecture. The virtual architecture has a rule-based logic inherent in code. It is constantly in flux.

Conclusion

The focus of the course was on how to use the end-user scripting possibility within Second Life, which means to learn the scripting language, to understand the principles of algorithmic design and to get involved with multi-user virtual environments as a kind of hyper media. Beside the possibility of software as a set of instructions that direct the computer to do a specific task, SL provides a 3D virtual world, with own (computational and network based) rules, with atmosphere in which people immerse, spending real time.

The architectural design becomes an ongoing process, initiated by the student, performed by the computer perceptible through the internet - a step towards autonomous architecture. The output is an interactive virtual hyper media, acting and happening in real time, to be experienced inside a virtual environment, thus also effect and change the role of the traditionally more passive spectator to an active part of the virtual environment. The results are combining concepts of architecture, computer science, art and New Media.

The first experience that students have, if they start working with code, is the fact that they want something and get something completely else. That is a simple but important step to learn for students, starting to work with algorithmic design approaches: a simple rule is capable to create complex outputs. The process of having an idea, writing a pseudo-script, translate it to real code and then to see what the program is doing, helps the students to really focus on the design process. It forces the designer to understand what he does and what the computer does. "As soon as the designer furnishes the machine with instructions for finding a method of solutions, the authorship of the results becomes ambiguous." (Negroponte 1969).

The writing of a pseudo-code constrains the student to abstract and to distil the idea. He has to concentrate on a few thoughts, breaking them down to rule-based logical algorithms. Then the code must be written, and the project can be tested in real-time.

The focus of the design is shifted towards the conceptional phase, and more time must be spent to get the code running. Software is complex, error-prone and difficult to visualize.

If the code is running, the result is there. The running code produces ongoing outcomes and the students have to think of conditions to limit or to stop their own objects. This is quit new for architects.

Compared to traditional design courses the discussions about the individual projects were more focused on the principal design ideas. We observed that it is really hard not to transfer ideas of the real world into the virtual.

The dropout quote was slightly smaller; the students show a high level of interest. The present course "Scripting Architecture" is even more visited.

Acknowledgements

We have to thank the Institute of Architecture and Design, TU Vienna for sponsoring the Island and Lukas Ofner, who helped as a research assistant.

References

Negroponte, N.: 1969, Towards Theory of Architecture Machines, Journal of Architectural Education (1947-1974), Vol. 23, No. 2 (Mar., 1969), pp. 9-12.

Manovich, L.: 2001, The Language of New Media, The MIT Press, p. 180.

Plottegg, M.: 1988(1996)), Architektur Algorithmen, Das binäre Haus & die Interaktion / Theorie, <http://plottegg.tuwien.ac.at/buchaa.htm#ToC19>.