

SELFRUNNING SOFTWARE ACCESS FOR THE EXPLORATION OF VIRTUALLY RECONSTRUCTED SYNAGOGUES

Bob Martens, Prof.Dr.⁽¹⁾

b.martens@tuwien.ac.at, www.tuwien.ac.at

Herbert Peter, Dipl.-Ing.⁽²⁾

h.peter@akbild.ac.at, www.akbild.ac.at

⁽¹⁾ Vienna University of Technology, ⁽²⁾ Academy of Fine Arts Vienna

ABSTRACT: *This contribution focuses on the so-called “Virtual Building Explorer (VBE)”, an adapted game-engine, which is now being used for communicating architectural information as it provides an interactive display of 3D scenes. The paper discusses the handling of this software and the creation of self-running applications. It also offers a comparison with other applications with a similar 3D visualisation performance profile. The aim is to test suitability of the software packages for establishing 3D walkthrough models of sacred buildings, based on existing 3D models of destroyed Viennese synagogues. The paper explores and explains the conditions for data conversion and any follow-up uses. While certain restrictions exist, potential benefits for the virtual reconstruction context can be identified.*

KEYWORDS: *Virtual reconstruction; destroyed synagogue; Virtual Building Explorer (VBE); real-time walk-through; perception.*

1. Introduction

What are the modern-day options of “re-creating” a building structure which no longer exists as a result of purposeful destruction? For more than 10 years, academic research has been undertaken into the virtual reconstruction of synagogue buildings in Vienna. Research must probably start with the question as to what type of information about the building has survived (planning and design documents, photographs, descriptions, etc.). Subsequently, a three-dimensional model will be produced. In the process, contradictory information needs to be checked for plausibility and/or information gaps filled in a plausible manner. A decision needs to be made about whether it is sufficient to present the respective building structure without its urban context or whether the immediate surroundings should be presented in a simplified manner (Martens and Peter, 2009). Another important issue is the level of detail used in the modelling.

The modelling process and the concomitant interpretation of available information have been comprehensively published (Martens and Peter, 2002 / 2010). This paper is concerned with possibilities of interactive viewing of an existing 3D virtual reconstruction, by using a self-running application instead of the original modelling software (ArchiCAD).

A particularly interesting form of visualisation of virtual building models became possible in 2005, when a Swedish software company (Zermatt VR Software) licensed its “game-engine” to Graphisoft, who implemented the *Virtual Building Explorer* (VBE) as a communications module in ArchiCAD in order to allow planners improved visual control of their building concepts and to offer the developers of the future structures a virtual walkthrough. With self-running VBE-based applications, users can easily navigate through a building by means of the mouse. “Global Illumination” gives the models realistic ambient lighting. Togeth-

er with still-renderings and digitally created scale models, the VBE is one further way of utilising an existing three-dimensional CAD-model.

Section 2 explains how the VBE applications are created and what the connected requirements are, and section 3 takes a closer look at the performance range of the VBE from the user's perspective. The respective advantages and disadvantages of related software packages are assessed in section 4, and the section 5 presents the conclusions.

2. Requirements for the Creation of a VBE Application

Which requirements need to be fulfilled in the preceding 3D modelling process to facilitate the creation of a self-running *Virtual-Building-Explorer* application? An important factor is the size of the initial model. The higher the number of polygons, the larger the storage capacity and performance requirements for the computer running the VBE application. The 3D data are imported directly from ArchiCAD through an integratable add-on and saved as a VBE data model. The model is then opened in the actual VBE authoring environment (Fig. 1) for fine-tuning of the display quality (global illumination, shadows, transparency, etc.). This is when the actual compilation takes place in this software environment. It must be noted that “global illumination” may well take a few hours in case of a large model. For the final saving as a self-running application the user can choose between the Macintosh and Windows OS or save the model for both operating systems if required.

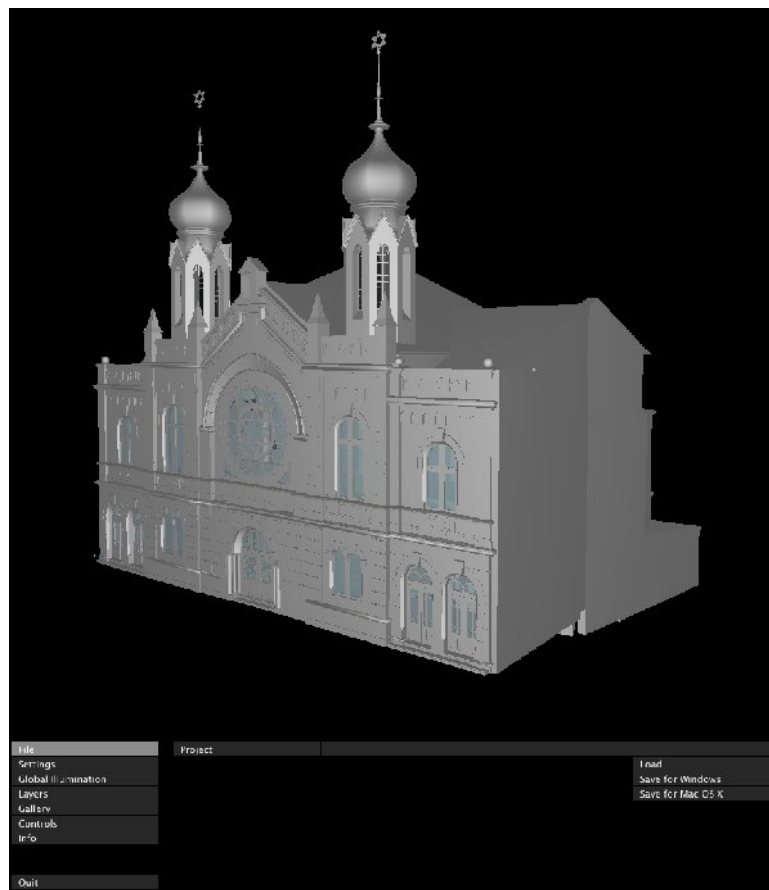


Fig. 1: 3D model of the synagogue at Kluckygasse (Vienna) during editing in the VBE authoring window.

It must be assumed that models of average complexity – i.e. in the 20-80 MB range – will not create serious problems. During modelling, the degree of abstraction (level of detail) needs to be given some thought. The exact adjustment of the VBE quality requires a high level of experience. Computation time will depend mainly on the use of global illumination, since the light model is calculated for each individual case.

The artefacts of which the 3D model is composed are usually placed on a kind of “greenfield”, and the creation of a surrounding environment – albeit at a high level of abstraction – is recommended (Fig. 2). During the VBE application generation a virtual baseplate is created automatically. Crossing over the “edge” may lead to the user “falling” into empty space, and he will not be able to get back to the building in “walking mode”. In order to prevent that from happening, it is advisable to provide for a structural boundary.

Normally, users will move through spaces devoid of people. Any desired humanoid figures would have to be modelled separately. They would be immobile. This can be compared to viewing physical models, where human figures are often placed inside the model.

The use of “realistic” surface patterns – so-called textures or shaders – can enhance the quality of the virtual model (they may also impair it, if they produce the impression of a multiplied pattern, or “tile effect”). The multiplication of a surface texture unit aims at giving the surface the appearance of a homogenous material. The edges of the pasted images need to be designed for seamless replication. Most software applications offer the choice of mirroring or sequentially arranging patterns. Usually, a texture does not consist of a single image of the desired material, but of at least three images which are superimposed on the surface to create the desired colour, reflectivity and bump. Creating a realistic materiality in terms of colour effect, light reflectivity (reflection, mirroring, gloss, etc.) is a real challenge. Navigation should be fast and stutter-free (= computing power), and the impression of the space should be as realistic as possible. Not unlike the developments with digital data formats in music and photography, we are ready to accept certain quality deficits in exchange for compact file sizes or other advantages.

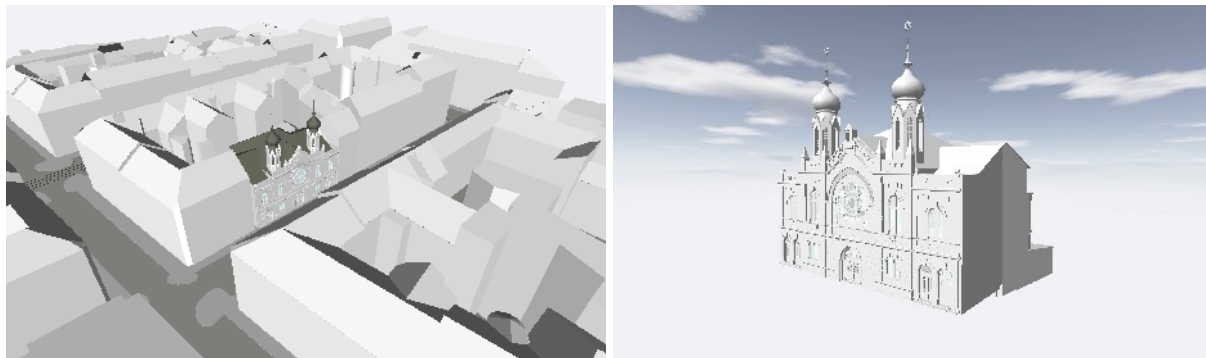


Fig. 2a-b: model representation with (ArchiCAD model) and without urban context (VBE application).

3. VBE from the End User's Perspective

Der *Virtual Building Explorer* is a self-running software application for the interactive communication of 3D objects and situations. This means that no additional software needs to be installed; a simple double click will start the presentation. Even an older computer will be able to cope with the software and the required starting time seems appropriate. It is, of course, interesting to know to which extent the software

is self-explanatory. This question was studied with the help of eight master students who work on themes relating to virtual reconstruction. They were sent a demo VBE application of a sacred building and asked to explore it on their own and subsequently report on this experience in writing. The idea was to encourage free exploring, and “stumbling” was not considered a failure. We were also interested in the students’ feedback on using this tool for the presentation of a (no longer existing) sacred structure. The following explanation is not meant to be a mere “how to” guide, but also intends to convey the practical experiences made with the tool.

3.1 Getting started and navigation

It takes only a short orientation phase to master the controls which are easy to use and intuitive. It is worthwhile, however, reading the information about the escape menu provided on the start-up window, as it facilitates getting help later on if needed. Users who would like to get the full flavour of navigation functions will have to use that menu, because it provides for a range of navigation options – from fly-mode to crouching-mode just about anything is possible. As an occasional user one will frequently consult the key shortcut menu.

The handling brings back memories of game-engine environments, since the letters W-S-A-D or the arrow keys are used for navigating. If the user feels unsure about that, the escape key will call up a menu with explanations at the bottom of the screen which shows all navigating options. Beginners will move very cautiously at first, but as nothing can be destroyed the initial inhibitions will soon be forgotten.

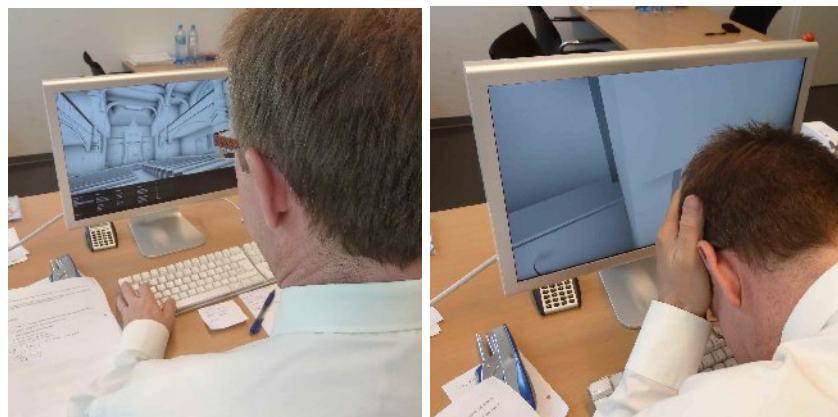


Fig. 3a-b: Using the Virtual Building Explorer (VBE) at the desktop computer: Just view it! Disorientation in virtual space can be a challenge and sometimes gives users a headache.

The most common modes of locomotion through the virtual model will be “walk-through” and “fly-through”. Spatial boundaries and objects (walls, furniture etc.) stop the user in walk-through mode and the “collision protection” function means that the user is forced to a halt. Therefore, doors must be built into walled-in spaces. Only if parts from the ArchiCAD door library are used in the 3D model, will the VBE tool interpret them as “doors”. The VBE will not, however, allow the user to pass through any door-like elements created with the help of other modelling tools. Stairs are correctly identified, and the viewer can walk up or down the stairs.

In fly-mode, the entire space is available for navigation and the user can move through obstacles such as walls like a ghost. A novice will probably sooner or later find him- or herself flying into nirvana and have to restart the VBE application to get back to the structure (Fig. 3a-b). Another “tragicomic” situation may arise if users (in walk through mode) are not perfectly familiar with the virtual structure and mistakenly

leave the building, reach the edge of the baseplate and “fall off the edge of the world”. This can be prevented by erecting a wall or fence around the site.

What is a “classical tour” (Fig. 4) through a Jewish sacred building like? Viewers will first get to the central aisle (ground floor), facing the *bimah* (also called *almemor*) and the Torah shrine behind it as they progress. Depending on the religious principles, the bimah – i.e. the rostrum where the Torah is read during liturgy – will be freestanding or very close to the Torah ark behind it. The structure and dimensions of the main room require the viewer to move his or her head to get enough information about the space. Walking straight ahead will soon have to stop and viewers need to consider how they want to continue the tour. Probably they will want to turn around and perhaps sit down for a short while. Or they may become interested in the side parts of the structure and sit down again. Depending on what they are interested in they may also want to visit the gallery or the winter praying room, if the synagogue has one. This will conclude the tour in every sense of the word. There will also be areas that are not publicly accessible and that fact should also be reflected in the virtual model, since the fly-mode gives the user virtual omnipresence and the possibility to satisfy any voyeuristic navigation wishes without restriction.

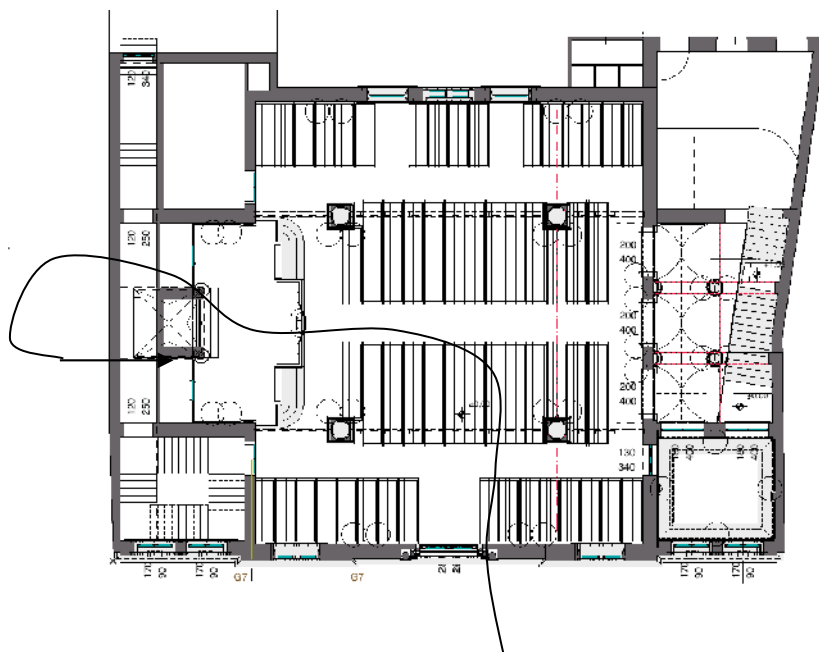


Fig. 4: A “classical” tour of the synagogue at Kluckygasse.

3.1.1 Mapmode: superimposed floor plan

For better orientation, the 3D presentation can be overlaid by a line drawing of the floor plan (Fig. 5). This feature is also known from computer gaming and provides simultaneous navigation and orientation without having to abandon the main scene. The arrow indicates the current direction of view. In shooting games, this type of orientation map is indispensable for “looking around corners”. Mapmode facilitates anticipation of what lies ahead in large and ramified structures. Full-screen superimposition soon becomes irritating, however, and unfortunately there is no way of reducing the map size. There is also an unfortunate association with shooting games which seems very unsuitable in the context of a religious building.

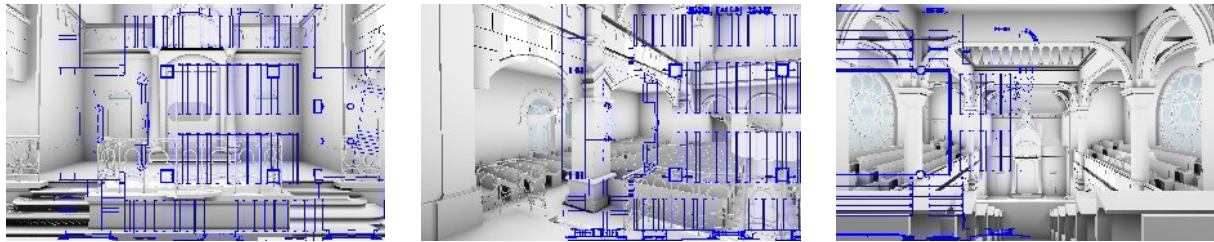


Fig. 5: Superimposition of 3D view and the floor plan showing the user's current position.

3.1.2 Unrestricted movement for the viewer

The movement through the structure is a kind of “digital endoscopic exploration”. The speed cannot be adjusted precisely to that of a pedestrian, cyclist or car driver, since the mouse controls the viewer's movement: forward and back, also in combination with moving to the side. Up or down are also available options. The eye level and the viewing angle can be accurately pre-set by keyboard entry. The use of other user interfaces – e.g. WiiView remote controller etc., could also be considered (Fritz, et al 2009).

Some viewers have the impression of looking into a virtual world through special (blinker) glasses. Usually the perspective is at eye level of a walking person. In fly-mode the viewer will of course lose touch with the ground. In other words, the overview perspective provided by fly-mode is useful but not particularly realistic. The control options, which are accessible through drag and drop or the right-click context menu are convenient. The software also permits one-hand (real-time) navigation by means of the keyboard (acceleration/deceleration of the navigation speed, etc.). In combination with mouse control of the direction of view, the tool creates a realistic impression of movement through the structure. Intuitive handling is not possible in any situation, though. When the escape menu is open, movement with the arrow keys is still possible, but mouse control of the direction of view is disabled. Users who are not aware of this will be puzzled why they are no longer able to “pan the camera” with the mouse.

| | | | | |
|---------------------|-----------|----------|---------------|-----------|
| File | Menu | Escape | Fly Mode | F |
| Settings | | | | |
| Global Illumination | Movement | W S A D | Info Tool | I |
| Layers | Move Fast | Shift | Measure Tool | M |
| Gallery | Move Slow | Control | Outlines | O |
| Controls | | | | |
| Info | Crouch | Alt | Sun Shadows | F3 |
| | Jump | Spacebar | Screenshot | F5 |
| | Lift | Y | Parallel View | F8 |
| Quit | Lower | H | Map Mode | Backspace |

Fig. 6: The options under the Controls menu.

3.1.3 Layers as information structuring tools

The layer structure enables a better understanding of the view and the structural elements it contains. Hiding individual layers serves to highlight particular elements. This option has to be taken into account when creating the model to enable the targeted hiding and showing of layers in the VBE. As the layer names are taken over from ArchiCAD, choosing clear names for the individual layers is recommended. When layer composition is appropriately planned, the layering will also allow for presenting different epochs (with different building states).



Fig. 7: Example of a layer structure (synagogue at Kluckygasse).

3.1.4 The VBE Gallery to assist exploration

The option to save animations is useful. The “autoplay” function may be used for info stalls and similar presentations: whenever there is no visitor present who actively moves through the digital model, the software will run a predefined animated sequence. A passive viewer will thus get an impression of the model without having to worry about navigation.

As there is a gallery, an integration of defined camera positions (reference points and viewpoints) would be desirable. This would not have to be restricted to predefined relevant viewpoints, but, in case of a large and complex model, should also include individual positions in different spaces or on different levels. In this manner, the viewer would not constantly be forced to move towards the starting position (= the camera position defined in ArchiCAD).

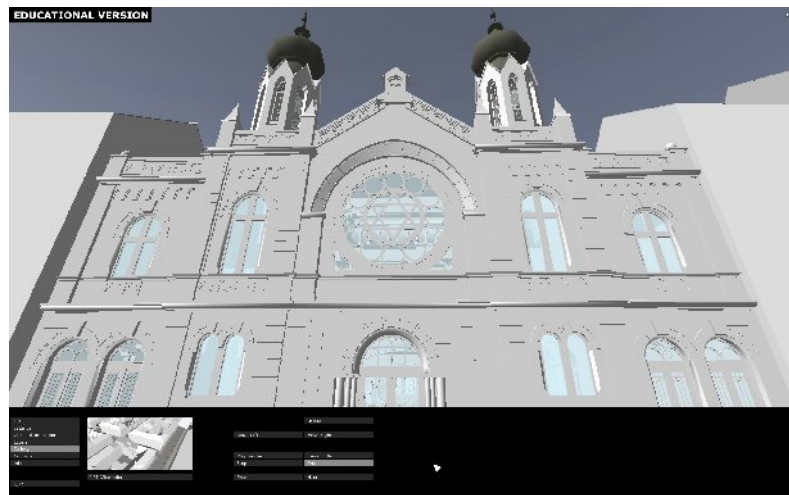


Fig. 8: Within the VBE gallery, pre-defined viewpoints can be retrieved.

3.1.5 Settings: less may be more

Settings include a large number of options. It might be advisable to restrict options radically for certain types of users and only enlarge the range when required. Are there many viewers who would like to start the session with dealing with render modes, changing backgrounds and adjusting transparency? A suggestion for the info tool (Fig. 9) would be to include descriptions of elements, e.g. what a *bimah* is or what the significance of the “Holy Ark” is (vgl. Pauwels, et al, 2010). The measuring tool is easy to use and provides dimensions of elements in the model. It is more difficult, however, to get exact measurements, since the snap points of structural elements such as ceilings, walls or roofs defined in the ArchiCAD software environment are not exported to the VBE application.



Fig. 9: Additional information about model elements can be displayed.

Unfortunately, changed settings cannot be saved, which means that they have to be repeated with every new start of the VBE tool. Another impractical point is the fact that it seems impossible to switch to another software while the VBE is running, e.g. in order to write a text. It would be helpful to be able to minimise the VBE or switch to the desktop.

After having looked at the creation of a VBE application and providing an overview of the way the *Virtual Building Explorer* (VBE) works, the next section will compare it with other software environments.

4. Alternative Software Environments

What are possible alternatives to the *Virtual Building Explorer*? This section explores other applications, without claiming to be exhaustive, which provide a stand-alone mode of display of 3D models. The claim to be independent of software may seem paradox at first. What we mean by that is that no proprietary software is required for viewing the model. A paragon example would be the *universal PDF format*, which was introduced in 1993 and enables document viewing across multiple platforms.

With the development of *3D-PDF*, Adobe provides a way to display 3D designs, also including embedded animation sequences. The main area of use, however, seems to be the engineering sector with technical publications (assembly plans) for engineering products. Architectural designs are a niche market, including e.g. the documentation of building components. The creation of a 3D-PDF document is relatively time-consuming and can only be done under Windows (while the finished file can also be displayed under Mac OS). A selected perspectival view of the model is linked to the .U3D file (previously saved from the ArchiCAD model). The user interface is easy to understand and offers various display and lighting options. The 3D model can be explored in walk-through or fly-mode. On the downside, loading and navigation speed (depending on model size) seem particularly slow.

QuicktimeVR has also been around for many years. Since 1994, it has been enabling users to “walk into” virtual spaces. It is a useful tool for giving a viewer the impression of being “inside” a virtual structure. The software uses 360° panorama images for navigation and the creator controls which part of the space is shown. The computation time for a panorama image has been reduced by several factors owing to mod-

ern computer processing power and does not represent a major problem for a mid-range computer. On the downside, the linking of individual viewpoint (by means of an authoring tool) is time-consuming. As soon as several panoramas have been linked through hot spots, the user moves from point to point by mouse-click.

Google SketchUp also offers navigation through a model. Again, there is a choice of walk-through or fly-mode. The direction of movement can be controlled with the arrow keys and by holding down additional keys. Unfortunately, walkthroughs start to stutter soon, even with smaller data models. The use of larger and more detailed imported models is not really advisable for the time being. An attractive option is viewing the model in a *GoogleEarth* environment (including annotations!), although the viewer can only navigate outside of the structure and has no possibility of entering a building. SketchUp models can also be used in game engines by means of a plug-in function (Lowe, 2009).

Blender Game Engine is an open source application used for modelling and visualisation (animation). The package includes an integrated, programmable game engine, which can be used from within the programme or on a player tool which comes with the software. The setup of the engine is not self-explanatory but can be quickly learned with the help of user-tutorials which are available on the internet free of cost. For the present case, two data formats (.3DS, Wavefront-.OBJ) were tested as interfaces between ArchiCAD and Blender. The Blender model can be explored in walk-through or fly-mode. Fly-mode is controlled by moving the mouse left or right, up or down. The speed of moving forward or backward is controlled by turning the scroll wheel in the desired direction. The more the wheel is scrolled, the more quickly will the camera move through the virtual space. In both modes, Blender supports shading and texture mapping, but not illumination. In practice, it has been found that a 10 MB (ArchiCAD) model of a synagogue will grow to approx. 100 MB in the export file and end up as a 60MB file in Blender. Hence, a high performance CPU and appropriate RAM size are required for smooth navigation.

5. Conclusions and Outlook

This paper studies the possibilities for making a 3D CAD design available to a wide range of viewers with the help of special software viewing tools. The designs used in this context are virtual reconstruction models of destroyed synagogue buildings. It must be stated that this exploration does not aim at physical reconstruction of the building stock, as almost all of the former sites are occupied with other structures or functions, and there would not be sufficient demand on the part of potential users. The Virtual Building Explorer (VBE) can be used to generate self-running applications from ArchiCAD models and make those models available in a slimmed down format.

The VBE tool, which is discussed in detail, has many features known from game engines. The fact that users may be reminded of shooter or horror games while looking at reconstructed sacred structures may be an issue of concern. It needs to be studied whether this would make the tool more or less appropriate for the public at large. Further consideration should be given to the target user group for such applications. Architects, archaeologists or (art) historians would perhaps be able to distance themselves from such associations. If the target user group includes wide sections of the population, certain restrictions in the viewing format might be considered, particularly in the case of models of sacred buildings. On the upside, the software tool does not require a long training period. Another question worth exploring is whether the virtual environment might give rise to boredom after spending some time there.

Since the (virtual) reconstruction of synagogue buildings is underway in many European countries, an exchange of virtual environments would be useful. The idea of creating a repository of self-running applications seems particularly appealing.

Acknowledgements

The authors are indebted to Georgi Buyukliev, Dimitri Inci, Hubert Jagsch, René Kommer, Maciej Lazewski, Julia Neuruhner, Robert Wieczorek and Katharina Wolf (Master students of Architecture at Vienna University of Technology) for their individual explorations and feedback reports. We would also like to thank Dr. Wolf-Michael Tschuppik for his feedback and comments.

References

- Fritz, R., Hsiao, C.-P. and Johnson, B. (2009). Gizmo & WiiView: Tangible User Interfaces Enabling Architectural Presentations, ACADIA 09: reForm - Building a Better Tomorrow [Proceedings of the 29th Annual Conference of the Association for Computer Aided Design in Architecture], Chicago (Illinois), 278-280.
- Lowe, R. (2009). Computer Game Modding for Architecture. Proceedings of the 14th International Conference on Computer Aided Architectural Design Research in Asia, Yunlin (Taiwan), 163-172.
- Martens, B. and Peter, H. (2002). Developing Systematics Regarding Virtual Reconstruction of Synagogues Thresholds. Design, Research, Education and Practice, in the Space Between the Physical and the Virtual [Proceedings of the 2002 Annual Conference of the Association for Computer Aided Design In Architecture] Pomona, (California) 24-27 October 2002, 349-356.
- Martens, B. and Peter H. (2009) Die zerstörten Synagogen Wiens - Virtuelle Stadtspaziergänge [City Guide to appear in English – autumn 2011]. Mandelbaum Verlag, Vienna.
- Martens, B. and Peter, H. (2010.). Displacing the frontiers of reconstructed cultural heritage: representation of the non-existing within an urban context. Proceedings of the 15th International Conference on Computer Aided Architectural Design Research in Asia. Chinese University of Hong Kong, 63-72.
- Pauwels, P.; De Meyer, R. and Van Campenhout. J. (2010). Visualisation of Semantic Architectural Information within a Game Engine Environment. Proceedings of the 10th International Conference on Construction Applications of Virtual Reality (Makanae, K. et. al. eds.), Sendai International Center, 219-228.

Web Links

- <http://www.graphisoft.com/products/virtual-building-explorer/>
- <http://www.adobe.com/manufacturing/3dpdfsamples/3dsolutions/>
- <http://www.apple.com/quicktime/technologies/qtvr/>
- <http://sketchup.google.com>
- <http://www.blender.org>