Using 3D Design software, BIM and game engines for architectural historical reconstruction

Stefan Boeykens
K.U.Leuven Department of Architecture, Urbanism and Planning, Belgium

The use of digital tools has become a tremendous aid in the creation of digital, historical reconstructions of architectural projects. While common visualization techniques focus on pre-rendered graphics, it is possible to apply Game Engines for real-time architectural visualization. This article presents the results of different reconstruction case studies, using a variety of design applications and techniques. Specific focus is placed on the use of Building Information Modeling software and on the inherent complexity and limitations in the process of translating an active, evolving model into an environment suitable for use in a real-time system.

Introduction

In digital, historical reconstructions, many different techniques and tools can be applied. While traditional approaches focused mostly on the creation of 2D CAD drawings, current reconstructions increasingly apply 3D Modeling techniques. Initially, the intent of applying 3D software was mostly oriented towards pre-rendered visualizations. Regular visualization techniques have been used for quite some time and they still pose interesting approaches. Caderon et al. [1] even suggest following cinematic techniques. While common visualizations focus on pre-rendered graphics, it is possible to apply Game Engines, for which Koehler and Dickmann [2] provide an overview. Hoon and Kehoe [3] and Pelosi [4] further discuss the use of such engines in architectural visualization.

In the course of our teaching and research efforts at the K.U.Leuven Department of Architecture, Urbanism and Planning in Leuven (Belgium), we have collected experience with several visualization and modeling

techniques, including the use of gaming engines, through different case studies. Table 1 displays a list of digital reconstructions that have been elaborated within our research group, partly through master thesis work carried out by our students of architecture.

Table 1 List of Case Studies

Town hall and Historic City Center (Leuven, Belgium)	Chapel N.D. Du Haut by Le Corbusier (Ronchamp, France)
Technical School RITO by H. Van De Velde (Leuven, Belgium)	Castle (Horst, Belgium)
Hunting Residence Mary of Hungary (Mariémont, Belgium)	Béguinage Church and Site (Hasselt, Belgium)
Castle (Boussu, Belgium)	Broodhuys (Brussels, Belgium)
Indochina University by E. Hébrard (Hanoi, Vietnam)	Church of Saint-James (Leuven, Belgium)
Maison du Peuple by V. Horta (Brussels, Belgium)	The dome of the Palace of Justice by J. Poelaert (Brussels, Belgium)
Palais Stoclet by J. Hoffmann (Brussels, Belgium)	Saint Walburgis Church (Antwerp, Belgium)

In these reconstruction cases, the 3D model was not the final product, but it aided the historical study, providing more than just visualization. The choice of projects to reconstruct was led by considerations of historical and architectural importance, possible accessibility of the (remains of the) project and access to historical sources.

To fully exploit these models requires them to be repurposed for different contexts, such as rendering, real-time exploration but also accurate building documentation. This motivated us to more actively develop working methods applying parametric techniques and Building Information Modeling, preferably as a combination of both.

Modeling for different purposes

Modeling buildings for heritage purposes requires the construction of accurate geometry, faithful to the historical context. Different techniques can be used. On the one hand, there are the measurement and surveying techniques. Image based photogrammetry and terrestrial laserscanning techniques are widely used and discussed in heritage literature and research, such as the works of Boulaassal et al. [5] or Somers et al. [6]. Most of these techniques generate mesh-based, static geometry with possible texture or color information. This is sufficient for archival, visualization and measurement purposes. However, in the context or our reconstruction cases, we rather have control over the model as to embed information and repurpose it for different contexts.

While modeling qualitative geometry for use in regular visualization already poses an elaborate effort, preparing models for different uses is even more complex. In practice it is unfeasible to create different models independently from each other, so a balance has to be found between the desired usage and the available model data. Polygon reduction algorithms can remove details, but avoiding excessive geometry at the modeling stage is preferable.

Using Parametric methods for modeling

The use of parametric methods is one usable approach to tackle this complexity, as illustrated by Chevrier et al. [7] and [8]. They apply the *MEL* scripting language in the *Maya* software from Autodesk and special attention is paid to a hierarchical description of elements, respecting their historical meaning and context, e.g. with the different architectural parts of cupolas and arches.

Different software tools are available, providing different means to apply parametric techniques. Table 2 gives a possible categorization of techniques into visual/non-visual approaches and into generic/specific systems for 3D design and modeling.

Table 2 Parametric Modeling Approaches with example implementations

	CAD/3D Systems	Generic
Visual Graphical	AutoCAD Dynamic Blocks Revit Families Bentley Generative Components Rhino3D + Grasshopper Paracloud SideFX Houdini MCAD (e.g. Inventor, SolidWorks, CATIA)	Geogebra Cabri/Cabri3D WireFusion Puredata+Gem * MAX/MSP/Jitter *
Textual Code-based	AutoCAD scripting (AutoLisp, .NET, VB) ArchiCAD GDL * or API 3ds Max MAXscript or API Maya MEL or API SketchUp Ruby or API VectorWorks VectorScript Rhino3D RhinoScript or Python	Flash/ActionScript * Processing Java, C++, .NET with added graphics libraries

* hybrid graphical and textual systems

While systems such as Bentley *Generative Components* and McNeel *Grasshopper* for Rhinoceros3D currently attract a large community of followers among architects and designers, in general almost any system is usable for parametric design and scripting. Most parametric systems are rather generic in nature and can be used for many different purposes. However, within the context of this article, we stress the importance to plan for the generation of models with different purposes and levels-of-detail in mind.

Integration in a Building Information Modeling workflow

While there is much to be said for the geometry-based model creation methods with most 3D CAD software or Animation programs, they are hard to integrate properly in a workflow around *Building Information Modeling* (BIM). The growing importance and acceptance of the BIM methodologies in practice, as described by Eastman et al. [9], but also in research and education is mostly seen with new constructions, focusing on the design process and building construction phases. While not trivial in the context of historical reconstructions, the BIM approach allows the same model to be used for technical drawings in 2D and an optimized 3D model in varying levels of detail for different visualization outcomes.

Parametric Object Libraries in BIM

Building Information Modeling applications seldom provide sufficient library content for the recreation of historical models and require an extensive set of parametric, custom objects for historically accurate models, which can serve multiple purposes.

An approach not identical but related to our work, is described by Zarzycki [10], where building details are created parametrically. They did this mostly in the context of teaching parametric design approaches, but the end result is quite similar, as the objects at stake are fully exploiting parametric behavior and support different purposes and variations.

We started elaborating parametric objects within BIM software in the course of the (ongoing) reconstruction of a synagogue in Vienna (Austria). In previous work done at TU/Wien by Martens [11], comparable reconstructions have been made, but custom library objects were not tackled thus far and this was seen as an interesting investigation path.

A single parametric object in the *ArchiCAD* BIM software is written in the Basic-like *GDL* language [12]. Such objects allow multiple representations, as they not only contain 3D, but also 2D and master fragments, all from a shared set of freely definable parameters. Consecutive updates to the library object definition can be reflected when the model is reloaded, retaining the instance parameters and position, even over multiple files or projects. Moreover, as the object can interrogate the context in which it is visualized, it can be made scale-sensitive and aware of some aspects of the project in which it resides. This allows the same "master model" to control different outputs, including the internal drawings and models, as well as exported geometric models.

However, this is not without limitations. GDL objects are mostly isolated entities and are not able to react to parameters of other placed

objects. They can only access some properties of the project in general, such as orientation and display scale. While the use of "*Solid Element Operations*" provides Boolean operations between the 3D geometry from different objects, which stay dynamic after model changes, there are no inter-object relations.

Example of parametrically adjusting a Library Object

The example of Figure 1 displays a column object from a regional ArchiCAD library. The model detail of this particular element can be controlled with the "resolution" parameter, which is set on an object-by-object basis.

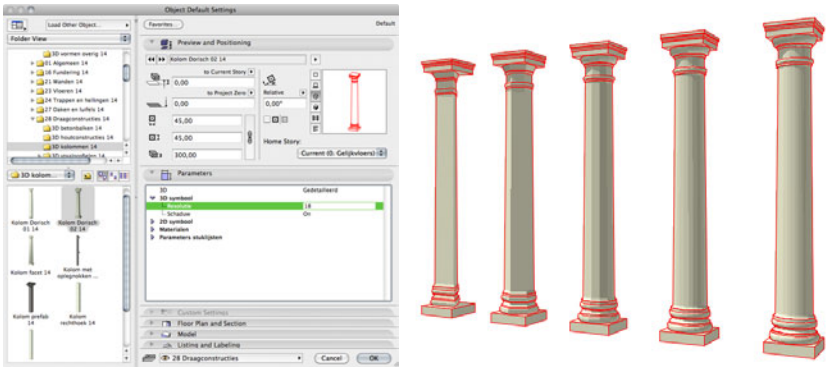

Fig. 1 Parametric Column object in ArchiCAD

The five variations generate from 252 till 14784 faces, depending on the 'resolution' setting, which is how GDL controls faceting of circular shapes. With proper normals defined, even the first column is suitable for a real-time environment. Similar comparisons can be made for other basic library objects, such as barrel vaults or cupolas. However, for photorealistic rendering, the higher resolutions are preferred.

The development of custom library objects for single-use is more work than directly modeling a static object, but writing them more generic allows reuse for other projects or for other instances in the same project.

Combining Parametric Design tools and BIM

While BIM software is inherently parametric, most current attention to parametric design and development is focused on non-BIM software. Especially Grasshopper and Generative Components have generated a large community of adopters, from both academics and architectural practice, as reflected in the *SmartGeometry* organization [13]. However,

some people attempt to connect both worlds, where parametric software is used to generate models to be further applied in a BIM workflow.

A promising example of integrating parametric modeling into several modeling tools can be seen in the *ANAR+* modeling library [14]. This library is written for *Processing* [15], an Open Source environment for the coding of interactive applications. The ANAR+ project presents the user an object-oriented programming library of classes, which can be called upon in a custom script. While the resulting geometry could be visualized using common rendering tools, they also support exchange to different systems in the form of generated scripts. They even support ArchiCAD by exporting GDL scripts. This is quite important, as exported parametric objects still retain (some of) their parametric properties. Instead of exporting the generated static geometry, the actual object definition is recreated in the native software environment.

Another nice example looking at integrating parametric software and BIM are the "*GeometryGym*" tools [16]. These are additional components for Grasshopper, to investigate the creation of shapes and models as *IFC* compatible files. While this does not retain the parametric definition from the first example, it does look at having more proper geometry into BIM projects. Thus far, most IFC-compatible systems, however, only support a limited subset of the possible geometric constructs in IFC, e.g. extrusions and boundary representations (breps).

Model Updates and workflow

The exported model does not need to stay static. An important threshold to tackle is the consecutive updates to the model, which will occur. Rather than waiting for the model to be finalized, it is possible, at least with some combinations of applications; such as *ArchiCAD* and *Cinema4D* or *Revit* and *3ds Max*, to continuously update the CAD (or BIM) model when exported into another application.

While this seems trivial, it becomes even more important when the software in which the model is hosted is used as an intermediate for even more model exports, e.g. to a real-time engine, which we will discuss further on.

Collaboration and custom tagging using BIM?

With the current options of collaborative editing of projects with the latest releases of BIM software, there are more options than ever to work on the same model with different people. It stays important, however, to embed enough information in the model. Not only the object's position and parameters need to be set, but also indication on the function and

composition of the model. This is much more detailed than with regular 3D modeling, where only geometry and surface materials matter.

Current BIM software has only a small set of "tagging" options available, such as defining main attributes of the object (e.g. through layers) or by defining some preset attributes, such as the "status" (new, existing, to be demolished) and the project phase. This is reasonably fine for current projects, but is far too rough for historical purposes. The software developers should provide more means to "tag" objects with proper information or metadata and allow end users to define custom tagging categories. If this were to be combined with the possibility to visualize the tags for different views, color-coded models could be more easily extracted from the same BIM model.

Figure 2 gives an example of color-coding for the purpose of displaying accuracy/certainty about particular parts of the model.

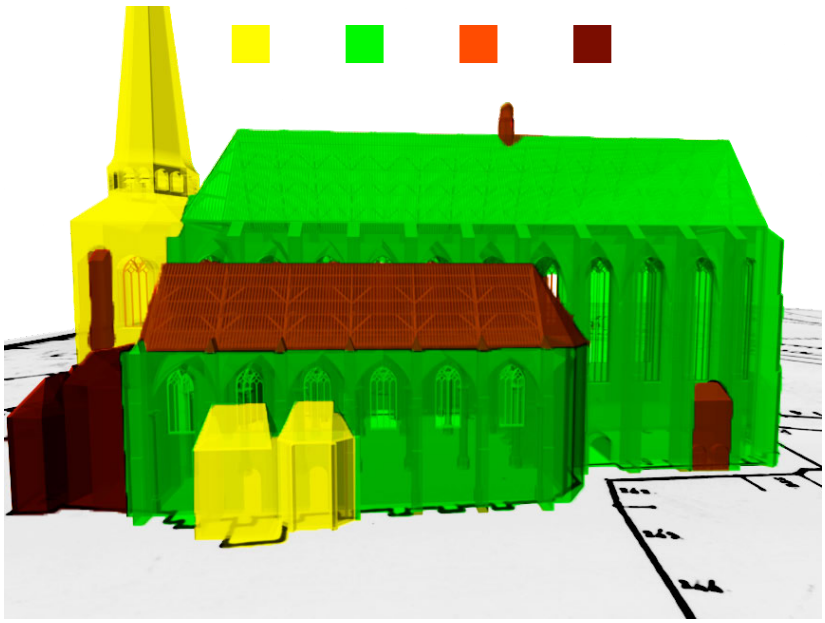

Fig. 2 Color-coded picture of St.-Walburgis Church reconstruction

This particular color-coding example was realized using adjusted versions of the main 3D model, although this disconnects representations from the model information. In a true BIM-approach, there would be a single model, where objects have both their regular material appearance and several optional functional appearances. In current BIM software, this can be only partially emulated, using views and possibly layers, but an

object can only live on one particular layer. *ArchiCAD* uses layers mainly for assigning functions to objects and some tagging options are under development. Some software, such as Revit, does not even provide layers, except upon export to DWG files. In *SketchUp* you can only use the main materials or the color-by-layer option, which is quite limited if you need to define multiple attributes visually.

Game Engines and interactive simulation

Made popular by computer games, real-time environments have the important advantage of allowing free dwelling through a building, shown by Uddin and Yoon [17]. Indraprastha and Shinozaki [18] and Johns and Lowe [19] extend this to an urban level or site, while Hernandez et al. [20] embed certain forms of interactivity. In the context of historical reconstruction, Doms and Kadar [21] show how it is possible to visit a virtual model of a site and experience first-hand how the project is constructed. Aspects of space and time can be properly presented. The use of hyperlinks makes it also possible to attach information to objects or locations, although care has to be taken not to interrupt navigation and to provide means of continuing the dwelling of the model. The use of Game Engines is also motivated by the visual quality and the flexible interaction that can be obtained, when compared to purpose-built systems for virtual visits, which offer only a fixed functionality and limited additional control.

A very elaborate example is the *Half-Life2* model of the Kauffmann house by Frank Lloyd Wright, created by user KasperG [22], visible as a screen capture movie on the *Digital Urban* blog [23]. Here, a rather faithful reconstruction was made, including full access to most of the house and the environment, with included lighting and some interactivity.

Gaming technology commonly simulates navigation into virtual worlds with various degrees of precision. The avatar in games is restricted in movement by collision detection and gravity, but usually as idealized figures with extraordinary navigation skills far beyond the physical realms. It is possible to define less-capable avatars, to simulate navigation with disabilities, as illustrated by Graf and Yan [24] or Yan and Liu [25], focusing on Interactive Building Design. Through the simulation of a virtual wheel chair, the accessibility of a design can be witnessed first hand. By rolling through the scene, using the integrated collision detection, possible problems can be checked. In addition, suitable first and third person perspectives allow the shift of perspective from user to the overview for the designer.

From 3D or BIM models to Game Engines – workflow considerations

There are different options to translate a 3D or BIM model into a real-time environment. Some solutions are integrated directly in the modeling software, such as the ArchiCAD *Virtual Building Explorer* or *VR4MAX* which integrates with 3ds Max. Some systems are independent authoring environments for interactivity, but have direct exporters available for different programs. Dedicated Game systems provide basic modeling tools inside an authoring environment and import meshes from external software. Pelosi [4] discusses some aspects of current game engines, how they relate and can be compared them with 3D and BIM software. This pinpoints some of the problems and issues, e.g. on navigation and representation. BIM proposes several options to control the output of the 3D geometry and is a valid approach to provide multi-purpose output from the main building model. In regular 3D modeling software, geometry is controlled directly and cannot easily be represented differently.

Examples from Case Studies

For the "Béguinage Church" and some other older case studies, we used *VRML* models with embedded hyperlinks to resources such as documents, images and movies. While quite dated, the VRML standard still has wide support in current 3D applications, although the lack of modern shader-based techniques is noticeable. In another case study, an *AutoCAD* model was loaded in 3ds Max to be transferred as a "static mesh" in the *Unreal Editor*. However, additional effort was required to properly define materials and collisions and it was a quite complex process. At that time, it was not suitable for common use or for frequent model updates.

Based on partial drawings and on-site measurements, the "Stoclet Palace" case study from Figure 3 was modeled using *SketchUp*. Special attention was paid to use textures based on photographs or books.

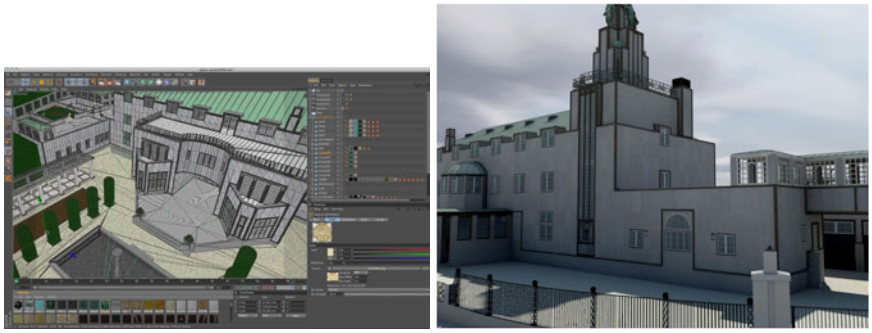

Fig. 3 Stoclet Model in Cinema4D and rendering

While earlier reconstructions have used Game Engines, they always posed serious effort to translate even a modest CAD or BIM model. With each subsequent modification, the whole process had to be repeated and most of the additional configuration and tweaking was lost. Exporting a 3D model is only the first step, as modifications to the main building model will occur frequently. By supporting repeated exports of the adjusted geometry, while maintaining the interactivity and configuration inside the real-time engine, a better workflow is supported.

We recently started using the *Unity3D* Game Engine [26], which is a cross-platform system, available in free and non-free versions. Goldstone [27] gives a complete overview. Games can be exported as standalone applications for *OSX* and *MS Windows*, for consoles, such as *XBox* and *Wii* and smartphones running *iOS* or *Android*. Importantly, web-applets for online use are supported as well. "Assets" are referenced external files, such as scripts, textures and models. Together with the internal Game Objects, they are assembled into different scenes or levels. As illustrated in Figure 4, the *FBX* format is ideal to translate geometry and material information into the game engine. However, the biggest workflow supporting aspect is the fact that the model is referenced instead of being fully embedded and it can be reloaded after model changes.

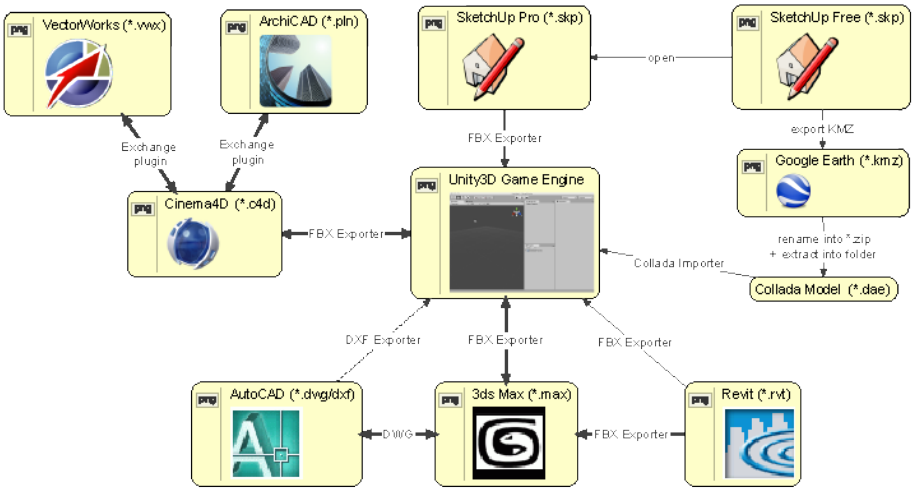

Fig. 4 Unity3D Game Engine and preferred integration with CAD/BIM software

Updating a model leaves material adjustments, model placement and script attachments in place, maintaining the interactivity to the model. When this is combined with linking CAD/BIM and visualization software,

a very flexible system is realized. It is possible to export one version of the BIM model with lower resolution and less detail into the real-time environment, while at the same time hosting a higher-resolution version for photorealistic rendering. All models can be updated at any time.

Lighting and shadows?

While only Unity3D Pro supports real-time shadows, external shadow baking or integrated light baking are supported, as illustrated in Figure 5, which also displays *Ambient Occlusion* (AO) for added realism.

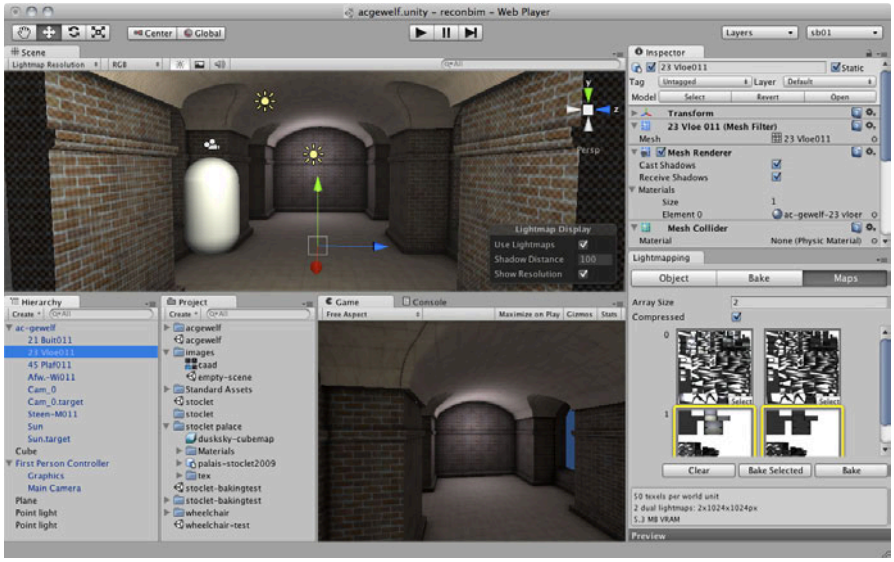

Fig. 5 Baking lighting in Unity3D

Accessibility Simulation

Figure 6 shows a wheel-chair model from SketchUp, that is attached to a moving camera, with adjusted movement, controlling the steepest slope or threshold and the speed the avatar can obtain.

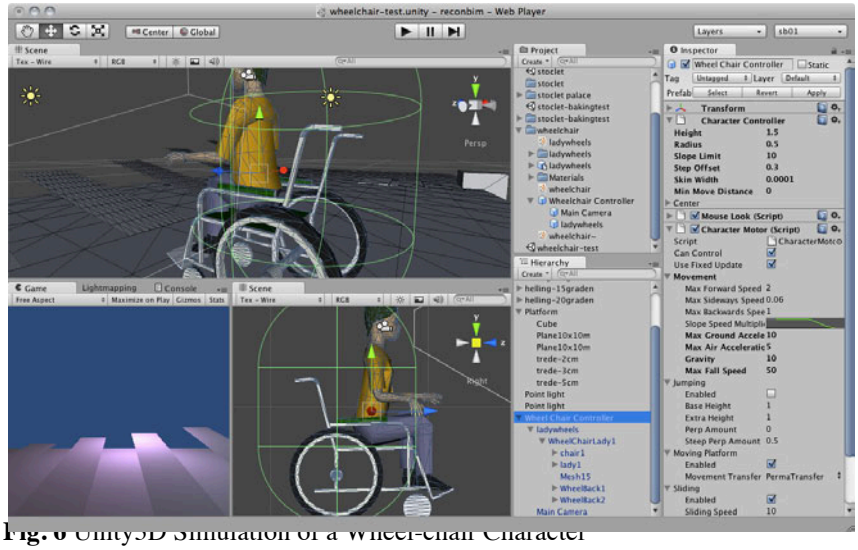

Fig. 6 Unity3D Simulation of a Wheel-chair Character

Export into real-time scene for web browsers

Figure 7 shows a 7 MB webplayer applet, generated by Unity3D from a 40 MB FBX model and about 8 MB of textures. More importantly, however, considering workflow is that the model can be further edited, improved and re-imported into the game system, with all material adjustments and applied interactivity intact.

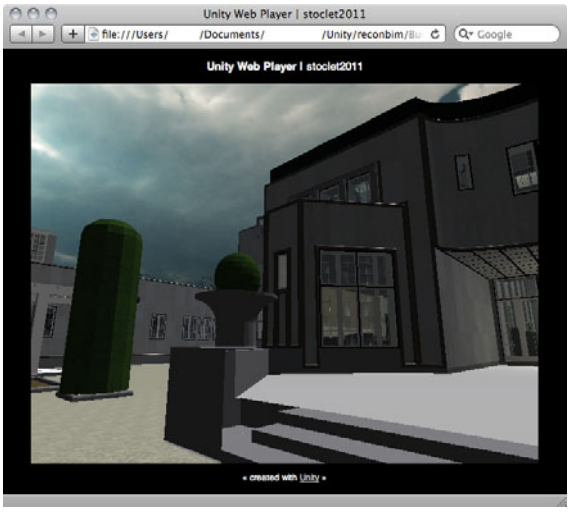

Fig. 7 Unity3D Webplayer applet, with loaded Stoclet model

Considerations and issues

Table 3 summarizes the most important problems or difficulties and some possible solutions, based on our work in different reconstruction cases.

Table 3 Considerations for better workflow between 3D/BIM and Game Engines

Problem/Issue	Possible solution(s)
Large CAD/BIM models make non-responsive scenes	Prepare "views" in BIM software to filter exported geometry. Scale-sensitive objects can generate smaller objects. Use "views" to split project in different parts.
No support for "instancing"	Convert repeating objects separately, to utilize instancing.
Bad textures or materials	Use materials in CAD/BIM software as placeholders and use advanced shaders to replace them. Use low-res versions of high-res textures in the CAD system.
Model out of scale	Use reference objects of known sizes (e.g. 1x1x1m cube). This is of utmost importance for navigation.
(Seemingly) Missing Faces	Flipped normals (reversed faces) become invisible in the visualization software. Proper modeling is the only solution. Generating 2-sided faces doubles geometry, so best avoided.
No lights or shadows	Baking shadows in textures is not trivial in large scenes. Real-time shadows are GPU intensive and not always supported.
Overlapping geometry	E.g. inner sides of objects, back of furniture placed against wall. Professional game designers strip away this geometry, but this is not suitable for multi-purpose architectural modeling. Current engines have less trouble with (some) extra geometry.

While not all of these considerations can be dealt with in every project, insight into possible problems can help with optimizing work on the modeling and management of complex and detailed 3D Models of architectural projects and reconstructions.

Conclusions and Future work

Using Parametrically controlled geometry on the one hand and BIM software on the other hand, it is possible to prepare a master model for historical reconstruction, while at the same time maintaining several derived models for different purposes. More work still has to be carried out on the creation of suitable parametric object libraries for BIM software, with these multiple purposes in mind. While regular 3D modeling software still has many advantages for complex geometry generation, BIM software can support both 2D and 3D documents from a single model. Other aspects to be further optimized pertain the preparation of an export methodology, where object instancing and proper object tagging is taken into account.

Acknowledgements

The work that is presented here could not be done without the effort of several researchers and master and post-graduate students of architecture. Thanks to Prof. Em. Herman Neuckermans for his continued support during these reconstructions. Figure 2 was adapted from a picture by M. De Vocht, while Figures 3 and 7 use the original model from K. Lesaffre.

References

1. Calderon C, Nyman K, Worley N (2006) The Architectural Cinematographer: Creating Architectural Experiences in 3D Real-time Environments. In: International journal of architectural computing 4(4): 71-90
2. Koehler T, Dieckmann A, Russell P (2008) An Evaluation of Contemporary Game Engines. In: Architecture in Computro (26th eCAADe Proceedings) Antwerpen (Belgium): 743-750
3. Hoon M, Kehoe M (2003) Enhancing Architectural Communication with Gaming Engines. In: Connecting >> Crossroads of Digital Discourse. In: ACADIA 2003 Proceedings, Indianapolis (Indiana): 349-355
4. Pelosi AW (2010) Obstacles of utilising real-time 3D visualisation in architectural representations and documentation In: Proceedings of the 15th CAADRIA Hong Kong: 391-400
5. Boulaassal H, Landes T, Grussenmeyer P (2008) Automatic extraction of planar clusters and their contours on building façades recorded by terrestrial laser scanner, In: Digital Heritage, VSMM2008 Proceedings,, Limassol (Cyprus), Archeolingua: 8-15
6. Somers M, McGovern E, Mooneya K (2008) Semi-automated building surface type extraction from terrestrial laser scanner data, In: VSMM 2008 Digital Heritage – In: Digital Heritage, VSMM2008 Proceedings, (Cyprus), Archeolingua: 16-23
7. Chevrier C, Perrin JP (2009) Generation of architectural parametric components: Cultural heritage 3D modelling, In: Joining Languages, Cultures and Visions: CAADFutures 2009, PUM: 105-118
8. Chevrier C, Charbonneau N, Grussenmeyer P, Perrin JP (2010) Parametric Documenting of Built Heritage: 3D Virtual Reconstruction of Architectural Details, In: International Journal of Architectural Computing 8(2): 135-150
9. Eastman C et al (2008) BIM handbook: a guide to building information modeling for owners, managers, designers, engineers and contractors, Wiley, New York
10. Zarzycki A (2010) Exploring Parametric BIM as a Conceptual Tool for Design and Building Technology Teaching, In: Proceedings of the Symposium on Simulation for Architecture and Urban Design (SimAud), Orlando, Florida (USA): 105-108

11. Martens B, Peter H (2002) Developing Systematics Regarding Virtual Reconstruction of Synagogues, In: Thresholds - Design, Research, Education and Practice, in the Space Between the Physical and the Virtual. ACADIA 2002 Proceedings, Pomona (California): 349-356.
12. Nicholson-Cole D (2000) The GDL Cookbook. Nottingham: Marmelade Graphics; 3d edition.
13. SmartGeometry, http://www.smartgeometry.org (checked on 4 April 2011)
14. LaBelle G, Nembrini J, Huang J (2009) Programming framework for architectural design ANAR+: Object oriented geometry, In: Joining Languages, Cultures and Visions. CAADFutures 2009, Montreal (Canada). PUM: 771-785.
15. Reas C, Fry B (2007) Processing: A Programming Handbook for Visual Designers and Artists, Boston: MIT Press.
16. GeometryGym, http://ssi.wikidot.com (checked on 4 April 2011)
17. Uddin MS, Yoon SY (2002) Peter Eisenman's House X, Scheme G: 3D Game Engine for Portable Virtual Representation of Architecture, In: Connecting the Real and the Virtual - design e-ducation (20th eCAADe Proceedings) Warsaw (Poland): 526-531
18. Indraprastha A, Shinozaki M (2008) Constructing Virtual Urban Environment Using Game Technology, In: Architecture in Computro (26th eCAADe Proceedings) Antwerpen (Belgium): 359-366
19. Johns R, Lowe R (2005) Unreal editor as a virtual design instrument in landscape architecture studio, Proceedings of the 6th international conference for information technologies in landscape, Dessau, Germany: 330–336
20. Hernandez LA, Taibo J, Blanco D, Iglesias JA, Seoane A, Jaspe A, Lopez Rocio (2007) Physically Walking in Digital Spaces - A Virtual Reality Installation for Exploration of Historical Heritage, International Journal of Architectural Computing 5(3): 487-506
21. Doms O, Kadar M (2008) Real time interaction with cultural heritage objects in virtual 3D, In: Digital Heritage, VSMM2008 Proceedings, Limassol (Cyprus), Archeolingua: 311-316
22. Kauffmann House, Half-Life 2 model by Kasperg, 2007 http://twhl.info/vault.php?map=3657 (checked on 4 April 2011)
23. Screen capture movie of Kauffmann House on Digital Urban Blog, 2006 http://www.digitalurban.org/2006/08/frank-lloyd-wright-architectual.html (checked on 4 April 2011)
24. Graf, R. and Yan, W. "Automatic Walkthrough Utilizing Building Information Modeling to Facilitate Architectural Visualization", In: Architecture in Computro (26th eCAADe Proceedings) Antwerpen (Belgium): 555-560.
25. Yan W, Liu G (2007) BIMGame: Integrating Building Information Modeling and Games to Enhance Sustainable Design and Education, In: eCAADe 2007 Proceedings, Frankfurt, Germany: 211-218.
26. Unity3D Game Engine http://www.unity3d.com (checked on 4 April 2011)
27. Goldstone W (2009) Unity Game Development Essentials, Packt Publishing