EXPLORING THE POTENTIAL OF GAME ENGINES FOR REAL-TIME LIGHT SIMULATION

Peter Ferschin[1], Monika Di Angelo[2], Georg Suter[1]
[1]Vienna University of Technology, Institute for Architectural Sciences,
Digital Architecture and Planning
[2]Vienna University of Technology, Institute for Computer Aided Automation,
Automation Systems Group

ABSTRACT

We explore the potential of game engines in the area of real-time light simulation as a feasibility study. We especially consider needs of architects, light planners, and clients like enhanced workflow, tighter integration of software packages, and real-time rendering of scenes. Furthermore, we investigate the publishing possibilities for client side interactive exploration of lighting scenarios. As a proof of concepts we tested the workflow on a chosen scenario suitable for diverse lighting situations, the ProjectSpace Kunsthalle.

Keywords: Real-Time Simulation, Light Simulation, Game Engines

INTRODUCTION

Light simulation is often incorporated in standard 3D modeling software. Even though this is done at a very advanced level in some of them (e.g. Maya), they are not specifically geared towards real-time simulation. Nevertheless, they support advanced features of *qualitative* light simulations like reflection, transparency, shadow, indirect lighting, etc. *Quantitative* aspects are usually handled by dedicated lighting planning software (e.g. DIALux), some of them even in real-time. They incorporate calculations for luminance, contrast, glare, etc. to meet recommended standards like "Lighting of indoor workplaces" EN12464-1. Real-time rendering engines (like LinceoVR) on the other hand deliver qualitative light simulations in real time, but do not consider quantitative aspects and come without a 3D modeler.

When it comes to publishing a model for client presentations, one is left with a collection of powerful software packages that are rarely easily combined, even though most of them support import from and export to other common tools.

There is a definite need for a tighter workflow. A workflow for light planning basically consists of three parts: 1) design of the building including the design of the lighting concept, 2) simulation of the lighting situations and 3) visualization of the building with lighting (see Figure 1).

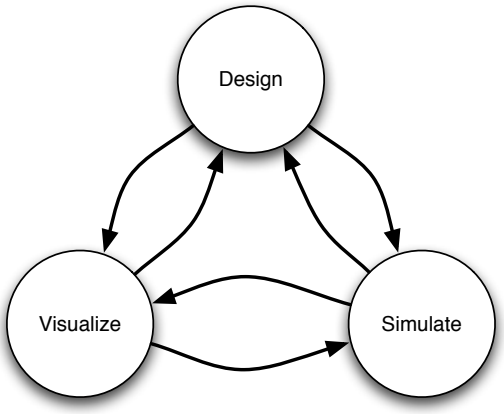

Figure 1: Basic workflow for light planning

This workflow is dramatically enhanced when it is iterative, easily coupled and iterations are done fast. "Performing visual-based analysis in the early stage of the design process" (Indraprashta et al., 2008) is essential. In a very tight coupling, the goal would be real-time visualization of light simulation.

Real-time digital techniques for design, simulation and representation (especially visualization) in architecture are available. To take advantage of state-of-the-art technology, a feasible workflow is needed. "The complexity of fully engaging with these opportunities rises alongside the laymen's experiences and expectations. If Architects do not want a filtering VR-expert between them and their clients, they need to rise to the challenge. The current generation of Architecture students is doing just that" (Lowe, 2009).

Recent technical developments of game engines, which contain increasing support for physical simulation in real time, make an investigation of their

application for scientific research interesting. Beside mechanical forces like gravity or even dynamic forces, light and sound are often calculated in real time in a dynamically changing three-dimensional space. Natural phenomena like wind, rain and snow are also integrated into the environment. Additionally, day light simulation can be handled interactively, for example letting the client set specific daytimes or the speed of the sun. Artificial light sources can be operated by the client as well.

Game engines provide many advantages. They allow for a "design in context", where a user explores the design as an interaction between himself, the 3D model and the environment. It so places "context over object", thus enabling interactive and participatory design (Moloney, 2002). Furthermore, in the interactive exploration "movement is not forcefully prescribed", but invites the user to walk along his own ways, adding "tactical solidity" and "integrated audio" (Hoon et al., 2003). With its real-time visualization "this form of presentation makes sense to every spectator very quickly and in a definite way, thus it makes collaboration easy especially in interdisciplinary teams." It is so "helping to make decisions in the process of panning" (Zeile et al., 2007).

There are also limitations to the employment of game engines (Hoon et al., 2003) as an interactive real-time renderer with tight workflow: not only the 3D geometry, but also the entire environment must be modeled, thereby increasing the complexity furthermore. Because of the necessity for real-time calculations, limitations have to be put on the geometry. Rather the overall implementation should always keep the requirements for real-time applications in mind. Last but not least, at present there is a definite lack of standardization. Nevertheless, the many advantages are worth the effort.

The choice between several existing 3D real-time game engines depends on the selection criteria. In a university environment, free software is always a preferred choice, as to not burden students with additional costs. For our purpose ease of workflow is an essential criterion, which basically means, straight-forward import of so-called assets (like 3D geometry, textures, animations, sounds). Also the possibility for an "import of external lights" is desirable (Koehler et al., 2008).

LIGHT SIMULATION

Resorting to computer-aided light simulation especially because of its 3D visualization possibilities becomes more and more common for planning and presenting lighting and luminaires in architecture. It allows for faster evaluation of lighting concepts while inspecting the building from various viewpoints. Two basic approaches to light simulation have been established: qualitative and quantitative light simulation, which are not very well integrated yet.

In quantitative light simulation, emphasis is laid on physically correct calculations of light distribution and intensity, thus enabling tests for compliance with existing standards. Typically, there is an extensive database of luminaires with technical data for each, provided by the manufacturer, that is available on-line. The software tools use this data for their calculation, combine it with the individual 3D geometry of the building; add in effecting materials (e.g. reflection) and deliver (a more or less time consuming) calculation result. Free software, which falls into that category, includes RADIANCE, RELUX and DIALux, to name the most prominent ones. RELUX only allows for very simple geometry, comes with a command line interface, and employs backwards ray tracing. RADIANCE also lacks a user-friendly free GUI, which is a major drawback, especially because of its complexity. But it is still interesting since it serves as a base for several other (mostly not for free) applications. DIALux on the contrary comes with a handy GUI and is considered standard among the existing light planning software. It is well suited for educational purposes as well.

Qualitative simulation on the other hand puts emphasis on (physically correct) photo realistic rendering quality without providing quantitative information that can be checked against standards. Well known examples oft software tools in this category are e.g. Maxwell, Fry, and VRAY.

REAL-TIME VISUALISATION

The significant advance in computing power due to faster processors, multi-core processing and high speed calculations on graphics cards by using GPU computations have led to the availability of software packages which allow a real-time simulation of lighting effects or at least perform the same task at much higher speeds. As many of the performance increases were also motivated by the increasing visual quality of video games we will give a brief comparison between dedicated real-time visualization software and game engines that allow the modification of its contents for architectural visualization.

Real-time Rendering Software

Real-time visualization software tries to minimize the time between design changes and the light simulation as qualitative output. Mostly the design changes consist in parameter changes of the light sources or the material parameters. Sometimes simple modifications on the geometry of the scenes are also allowed. Integration to 3D modeling software is done either by file format exchange or by even tighter integration using plug-ins for the modeling environment.

Presentation is mostly possible with interactive walkthroughs, often also for customer presentations on large projection environments. Examples for such applications are LinceoVR and Octanerender.

Game Engines

Recently, a trend towards employing game engines for dynamic and interactive visualizations in architecture can be noticed. One of the reasons might be, that they represent a *comprehensive framework* for real-time visualization. The advantage of a framework lies in its *ease of workflow*, where the specialized tools blend in together, digital content can be assembled, enriched with high *interactivity* and fine-tuned with purpose-tailored *programming*. As game engines can also be seen as software development environments with already existing frameworks for three-dimensional simulation, their scripting possibilities and integration of programming languages allow the extension, customization and behavior of the pre-defined scene (objects, cameras, lighting, etc.).

Finally, the customized presentation can be *published* for a variety of platforms (PC, mobile phone, game consoles, web, etc.) and distributed to the clients.

Among others the following game engines have been used in architectural visualizations: Unreal, Cryengine, Unity, Quest3D, Silex3d, Blender.

In our project we have chosen the Unity game engine for the following reasons:

- simple user interface
- easy workflow for design changes
- tight integration to 3d modeling
- workflow enhancement by scripting
- free entry version available
- artist friendly

Although some of the software packages mentioned above might provide better visual results or allow the real-time simulation of very complex scenes, Unity seemed to be the best choice for our needs. The overall advantage of Unity lies in its ease of use and its flexibility of extending the workflow. Unity seems also very appropriate for the use in an academic environment as it supports Windows and Macintosh platforms and has a student friendly license policy.

TEST ENVIRONMENT

As a reference and test model an exhibition space, the ProjectSpace of the Kunsthalle in Vienna, was chosen. Besides of the convenient location on the Karlsplatz, near the Vienna University of Technology, it is one of the landmarks at this area, well observed by local inhabitants, students, tourists and the passing traffic. The design was done by the architect Krischanitz as a glass box that allows the view of the historic surrounding from inside as well as the exhibited objects from the outside.

As the exhibition space might be used at day and night time, a variation in the light design is always an interesting challenge for each exhibition. Furthermore the exhibition "ArchDiploma", that shows an overview of the best Master Projects in Architecture over a period of two years is taking place, with always changing exhibition concepts. Despite the simplicity of the geometry of the building, the challenge of light simulation lies in the interplay between inner lighting and daylight simulation.

Figure 2: Photo Kunsthalle Day

Figure 3: Photo Kunsthalle Night

LIGHT SIMULATION WITH DIALux AND UNITY

Our main intention of this project was to establish a tight integration between design, simulation and visualization. For the geometric design and basic material definitions Cinema4D was used. Its ability to export 3DS files allowed to import the 3D model into DIALux. The next step was to define the luminaires that should be used for the exhibition. In our case we used the integrated plug-in from Zumtobel to obtain suitable light sources. Then the light simulation is performed in DIALux with various settings, where at the end of the calculation the result of the calculated light distribution can be viewed interactively in a 3D view window.

After viewing the light distribution in its qualitative state (Figure 5 and 6) images or videos can be saved to show the result of the light calculation to the customer. Our intention in this project was to enhance the presentation methods that can be shown to a client by allowing interactive walkthroughs of pre-calculated light settings as well as enhancing the visual display of using photographic representations of the surrounding historic environment.

Unfortunately, DIALux does not allow the saving of the content of the 3D view as textured 3D model, which would be very suitable for our purposes. Therefore we had to overcome this limitation by using the 3D capture functionality of Adobe Acrobat Extended. Hereby it is possible to perform a kind of 3D „screenshot“ as so-called 3D capture of objects in another application by saving the display list of OpenGL objects, which requires to use DIALux settings that use OpenGL as display driver.

After capturing the 3D model, Acrobat allows to save it as a 3D-PDF file, from which the content can be extracted as U3D file, which later can be converted to a FBX file, the preferred file format to integrate textured 3D models into the Unity environment.

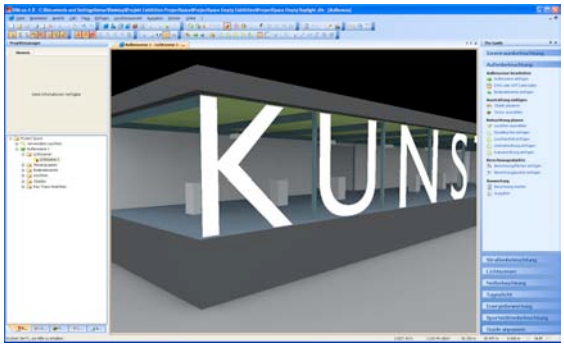

Figure 4: Daylight Simulation in DIALux

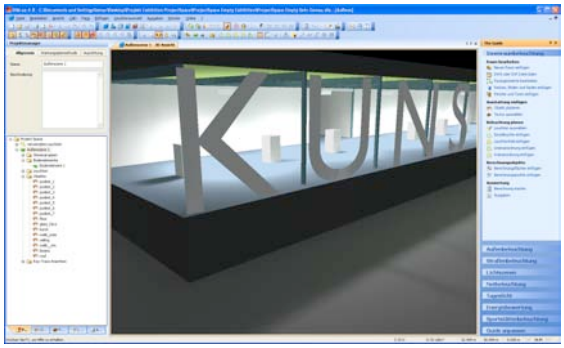

Figure 5: Simulation of Exhibition Lights

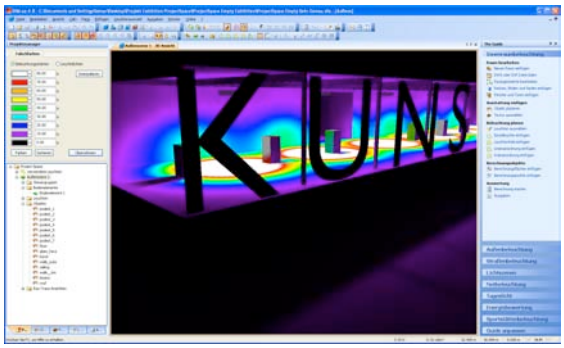

Figure 6: False Color Visualization of Light Distribution

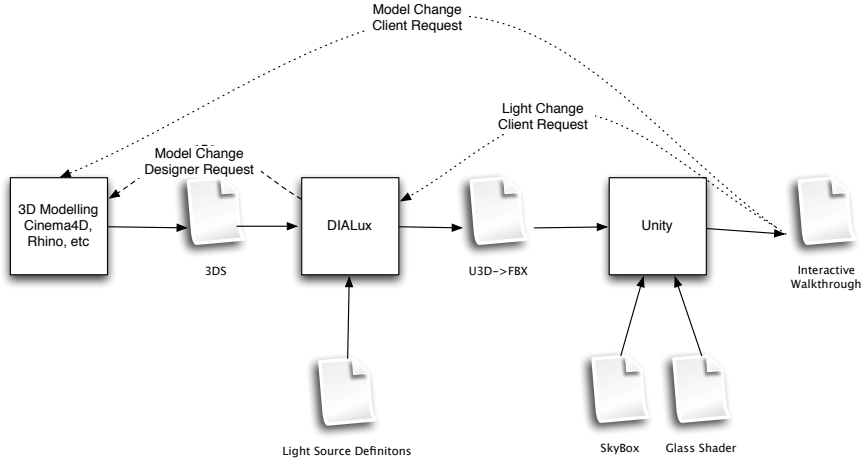

Figure 7: Workflow for Simulating Light with DIALux and Unity

The diagram in figure 7 describes the workflow between the chosen design software (e.g. Cinema4D), DIALux as light simulation program and Unity as visualization environment, with the possibility to transfer the data in the most appropriate file formats, and how the client's (dotted line) or designer's (dashed line) feedback might alter parameters of the entire setting.

At importing the converted FBX file into Unity some post-processing needed to be done, to assign the correct shaders to objects with light distribution as textures on the mesh surface. As the light distribution was pre-calculated – which is also called light-baking in real-time visualization – a lightmapped-diffuse shader was assigned to meshes with diffuse materials where the pre-calculated textures were copied into the light map channel. The materials of the glass façade were manually tagged as glass, so the import process could assign a special glass shader that also allows reflection mapping.

To enhance the visual display in Unity a panoramic image was generated of photos of the surrounding environment of the exhibition space. The panoramic image was then converted into 6 quadratic images covering the 6 faces of a cube. This so-called cube map could later be used to enable the reflections in the glass façade and further more to generate an image based environment as a skybox.

Figure 8: Environment with Cube Maps

Although panoramic photography is an established technology, usual panoramic images are shot with a fixed view center. In our case the images outside the exhibition building had to be combined with different view points which caused problems for objects near the camera. Despite these problems it was possible to generate a usable cube map as most of the environmental buildings were at a distance without too much perspective distortion.

Figure 9: Walkthrough in Unity with Light Distribution Inside, Reflections on the Façade, and Skybox.

In figure 9 the combined 3D scene is shown: the background shows a section of the skybox, and in the facing façade the mirror reflection can be seen, which is generated in real-time from the photographed cube map. The inside shows the light mapped calculations that have been imported from DIALux.

Additionally a terrain, enhanced with grass textures, was generated so that an interactive walkthrough is possible inside the exhibition space as well as in the immediate surrounding to allow a view from outside into the building.

DISCUSSION

In our work we showed how the processes of design, simulation and visualization could be integrated in the context of light simulation. From our experience this integration can be achieved with three strategies:

1) a fully integrated application
2) a central application with plug-in modules
3) separate applications with data exchange

In light simulation, qualitative results can be obtained very well with strategies 1) and 2). If quantitative results are necessary, at the present situation strategy 3 seems to be the most practical solution.

Additionally, one should consider the roles of the involved persons as the designer/architect might have a focus on qualitative light design, were the dedicated light planner will possess the necessary knowledge of quantitative parameters and the standards that have to be met at certain use cases. The role of the client should not be neglected and strategies should be provided for solutions that can integrate both qualitative and quantitative considerations into easy accessible and navigable applications.

In our project we have chosen strategy 3 to allow the inclusion of a quantitative analysis and to provide a virtual experience for the client with an interactive walkthrough. Unfortunately, the workflow was not as tight as desired, which was mainly caused by lack of data exchange possibility between DIALux and Unity. Nevertheless, we could demonstrate that accurate light simulation solutions can be integrated into a real time visual framework and the resulting application shows a lot of potential for future extensions.

One possible workflow improvement would be to allow data export from DIALux from the 3D view, that already contains distributed light solutions as light-mapped polygons. Possible export formats would be the COLLADA or the FBX file format. Additionally, improvements to the various material concepts could help for a better integration between each software package. As it is possible to integrate libraries for light parameters from various manufacturers – as in DIALux – a similar concept for materials would improve the workflow to a large extent.

In the area of interactive presentation of simulation results it would be desirable to switch between different lighting concepts or to activate individually each pre-defined light source. This should be possible to implement in Unity, as it contains already the concept of light-mapped shaders. Different light settings could be realized by switching between different light-maps and by using multilayered materials.

Another interesting extension would be to use image-based lighting with high dynamic range images (HDRI), which is currently not possible in DIALux, but could be integrated into Unity by writing special shaders.

The improvement of generating skyboxes from the surrounding environments, maybe even as image-based light-source, seems another interesting research direction.

ACKNOWLEDGEMENTS

Thanks to the Department of Architectural Theory for providing the basic 3D model of the Kunsthalle, and to Marcus Bleckert for his support in processing the panoramic images.

REFERENCES

Koehler, T. et al. 2008. An Evaluation of Contemporary Game Engines: In: Architecture in 'Computro' (26th eCAADe C. Proc.).

Hoon, M., Kohoe, M. 2003. Enhancing Architectural Communication with Game Engines. In: Proc. ACADIA 22.

Moloney, J. 2002. Game Engines and Virtual Design Studios. In: Architecture in 'Computro' (Proc. 23th eCAADe).

Inanici, M. 2001. Application of the State-of-the-Art Computer Simulation and Visualization in Architectural Lighting Research. In: Proc. 7th IBPSA.

Indraprashta, A., et al. 2008. Constructing Virtual Urban Environment Using Game Engine Technology. In: Architecture in 'Computro' (Proc. 26th eCAADe).

Zeile, P., et al. 2007. Illumination of urban space using digital simulation methods, exemplified on the UNESCO World Heritage Site of Bamberg. In: Proc. ASCAAD'07, Egypt.

Lowe, R. 2009. Computer Game Modding for Architecture. In: Proc. CAADRIA'09, Taiwan.

Roy, G. 2000. A Comparative Study of Lighting Simulation Packages Suitable for use in Architectural Design. Tech.rep. School of Engineering, Murdoch University.

[cryengine] mycryengine.com
[linceo] linceovr.seac02.it
[octane] www.refractivesoftware.com
[quest3d] quest3d.com
[silex3d] Silex3D.com
[unity] unity3d.com
[unreal] www.udk.com
[blender] www.blender.org
[zumtobel] www.zumtobel.com
[maxwell] www.maxwellrender.com
[vray] www.chaosgroup.com
[fry] www.randomcontrol.com/fryrender
[radiance] radsite.lbl.gov/radiance/
[relux] relux.biz
[dialux] www.dialux.com
[acrobat] www.adobe.com
[u3d] www.ecma-international.org
[fbx] usa.autodesk.com
[collada] www.khronos.org
[cinema4D] www.maxon.net