

GAMING TECHNOLOGY IN CULTURAL HERITAGE SYSTEMS

Tim Horz

Albert Pritzkau

Christof Rezk-Salama

Severin S. Todt

Andreas Kolb

Computer Graphics Group
University of Siegen
Hoelderlinstr. 3
57076 Siegen, Germany

{tim.horz, albert.pritzkau}@student.uni-siegen.de
{rezk, todt, kolb}@fb12.uni-siegen.de

KEYWORDS

Serious Gaming, Methodology, Rendering

ABSTRACT

This paper describes the design and implementation of an interactive walk-through of a reconstructed German stronghold, the Dillenburg. The application is currently in use at the local museum. Applying technologies and algorithms primarily used for the development of realistic 3D computer games, we present a system that can be categorized as a *Serious Gaming* environment, a term that has recently come into existence. More specifically, it can be called a *Cultural Heritage Game*. With this paper we want to give a programmer's view on the topic of interactive cultural heritage systems.

INTRODUCTION

In recent years the markets for high-quality 3D computer games and commodity graphics hardware have grown immensely. Every new generation of graphics cards allows for the next generation of 3D games, which in turn demands the following generation of graphics hardware. During this development, the amount of innovative techniques and algorithms needed for a realistic real-time rendering of virtual environments keeps growing.

The technology and specific know-how gained from years of computer game development, not limited to 3D games, can be used for creating applications *outside* a gaming environment. Recently, this notion has been labeled as *Serious Gaming* (Blackman, 2005; Raybourn and Bos, 2005). As can be seen below, numerous projects show the major interest in cultural heritage applications, all of them using game technology in some way or another. Those and future interdisciplinary projects of that kind can be called *Cultural Heritage Games*.

Our *Cultural Heritage Game* presents an information and learning experience that helps to revive a part of history. As a cultural heritage undertaking funded by

a non-profit organization and implemented by undergraduate students, our project stands outside the usual scope of business models and profit interests.

Throughout this paper we discuss aspects concerning real-time graphics programming applied to the 3D model extracted from a previous cultural heritage reconstruction project.



Figure 1: The Installation at the Museum

RELATED WORK

Generally, our work can be categorized as a computer aided cultural heritage project. Numerous comparable projects exist throughout Europe, some of which inspired our design decisions in different ways. Most notably, the virtual walk installation at *Trento* (Conti et al. 2006), the guided tour at *Sagalassos* (Pollefeys et al. 2001) and the (web-based) visit to the *Piazza dei Miracoli* (Carrozzino et al. 2005) gave us insight into cultural heritage projects. Those projects provide some good concepts and ideas that inspired some of our solutions. Content related aspects of the projects were

not adapted, since our work is focused on the application of game technology to existing 3D cultural heritage content.

The starting point to our work was, in fact, a previous project, aiming at the virtual reconstruction of the long-gone German stronghold Dillenburg (Todt et al. 2007). The project yielded a detailed 3D model used to generate an off-line rendered documentary movie of the stronghold that was totally destroyed in 1760. The movie and some rendered flights around individual buildings can be viewed by visitors of the *Museum Wilhelmsturm*, located at the archaeological site of the Dillenburg stronghold. The multimedia installation is integrated seamlessly into the exhibition. Visitors can browse a DVD containing the available renderings using a touchscreen interface (see Figure 1). The ambitious project was inspired and funded by the local cultural heritage association (the *Museumsverein Dillenburg*). From this project, we use the reconstructed model to implement an interactive walk-through. For human-computer interaction we take advantage of the already present touchscreen interface and use it as the input device for the entire simulation.

THE VIRTUAL MUSEUM

When creating a virtual museum application, most design decisions are determined by the setting, audience and objective of the corresponding exhibition.

In general, the heterogeneous audience of a museum, ranging from school classes to interested senior citizens, implies many options the system has to offer. For example, children might get uninterested if the simulation does not provide the visual quality that they are used to from modern computer games, while seniors might reject the notion of interactivity and prefer watching the original narrated movie rather than exploring the stronghold themselves. Furthermore, a good and easy means to navigate through the virtual environment is necessary, as well as an information system to present the most important and interesting facts about the site. Some of these aspects can be addressed by finding similar problems in computer game development, while others need a more tailored solution.

IMPLEMENTATION

Within the scope of this kind of project, it is not feasible to implement a custom game-engine from scratch. Since licensing a commercial game-engine was also not an option, we decided to combine various free and/or open-source software and adapted it to our needs. For the most important piece, the graphics engine, we chose the *Ogre-3D-engine*, after analyzing different open-source competitors. The *Ogre* engine convinced us with its ease-of-use, its extensibility and, equally important, the very active community that contributes to a wiki-

portal and a Q&A-forum. Through an extension called *OgreNewt* we have simple access to the functionality of the free *Newton* physics engine. Similarly, sound support is added via *FMOD*, and keyboard and mouse input is handled by the *Open Input System (OIS)*. Direct access to the touchscreen is provided by the manufacturer *Elo*. As an embedded HTML-rendering component we use TerraInformatica's *HTMLLayout*. Another plugin called *WMvideo-Plugin*, created and supported by the *Ogre* community, enables the system to show movies on arbitrary surfaces. Figure 3 sketches the collaboration of all these APIs.

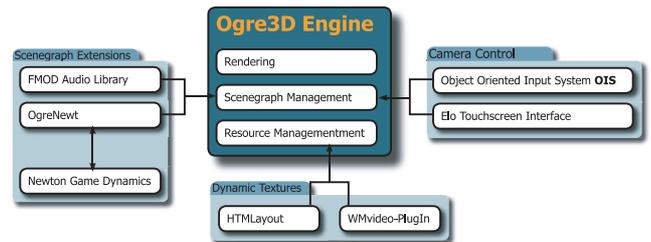


Figure 3: System Overview

Rendering

Nowadays, products that use real-time three-dimensional rendering, most notably computer games, have to meet a simple but challenging criterion: the consumers have to feel the rendering looks *good*, i.e. convincing and realistic. Being a subjective factor as it may, the perception of quality is the driving force in computer graphics technology, since visual quality is a selling point for each generation of graphics hardware. Reversely, consumers get used to better and better quality. If a product does not meet the visual standards the users have grown accustomed to, it may be disregarded completely. This notion can be especially observed with an important audience of cultural heritage sites: children and young adults. Any approach to a virtual museum has to meet this quality challenge. Therefore, we chose to apply a state-of-the-art rendering technique to ensure the good looks of our simulation. When designing the system we also took advantage of high-end computer graphics hardware, an *nVidia GeForce 8800 GTX* with 756 MB of video memory.

The preceding reconstruction project mentioned above yielded a complex model of the Dillenburg stronghold, completely textured with decal maps, normal maps and gloss maps. An 8-minute flight around and inside the model was rendered off-line, using various global illumination and compositing techniques. With our interactive version we try to offer an equally impressive rendering of the same model in real-time. Due to the complexity of the Dillenburg model we use an environment map



Figure 2: left: the Manor-house inside the Stronghold, right: Overview Map with Teleportation Points

(representing the sky) to light the entire scene, rather than using one or various point-light sources. However, traditional (even HDR) environment maps cannot be used for a complete and realistic real-time lighting of an object, as they only represent the light information for perfectly reflective surfaces. *Prefiltered environment maps* provide a powerful lighting and rendering approach to compete the challenge (Kautz and McCool 2000). This technique approximates glossy reflection with a set of cube maps generated from the original environment maps. Using ray casting techniques, the diffuse part of the lighting is precalculated and stored in an irradiance map. Likewise, a set of maps corresponding to different predefined specular lobes are generated. In the application, the information is combined on a per pixel basis using programmable graphics hardware as follows:

$$I = K_d \cdot I_{irr} + K_s \cdot ((1 - g) \cdot I_{specLow} + g \cdot I_{specHigh}) \quad (1)$$

In Equation 1 I is the final lighting for the pixel. K_d simply denotes the color value determined by the decal texture. I_{irr} corresponds to the diffuse term in the Phong illumination model and is determined by a texture lookup inside the irradiance environment map. The factor K_s represents the shininess of the material, passed to the graphics hardware as a uniform parameter. The specular term is approximated by the linear interpolation of two predefined specular lobes, denoted as $I_{specLow}$ and $I_{specHigh}$. After experimenting with some values, we chose $I_{specLow} = 20$ and $I_{specHigh} = \infty$, as they yield the best visual results. The interpolation value g is looked up inside the gloss-texture and is equivalent to the specular exponent of the Phong model. Note that to avoid a blueish touch on non-reflective surfaces (caused by the blue sky), we desaturate the irradiance value beforehand.

Realistic shadowing is achieved by a combination of two types of precalculated (baked) shadow-maps. On the one hand traditional baked shadows are used. On the other hand we generated ambient occlusion maps to improve realism and plasticity of the objects (Landis, 2002; Zhukov et al., 1998). As the original model had been created with *Autodesk® Maya®*, we used the baking technique provided by this modeling software which allows textures containing lighting information to be created fast and easy. After the rendering process both the ambient occlusion and the shadow maps were combined into one texture. To ensure a high quality we decided to use one grayscale 1024x1024 pixel texture per building or wing of a building. With a total of 130 textures, the lightmaps for the entire reconstruction consume 130 MB of video memory. The shadowing information is combined with Equation 1 as follows:

$$I_{shadowed} = s \cdot I \quad (2)$$

Here, the factor s denotes a lookup inside the combined shadow and ambient occlusion map. Figure 4 shows the different textures used for rendering on the most prominent building of the site, the manor-house. Another screenshot showing the same building inside the simulation can be seen in Figure 2, left.

Interaction

The creators of the *Sagalassos* project argue that offering a completely free walk through a virtual museum is not desirable, as users might not get to see all the important buildings or even get bored. Therefore, they only offer virtual guided tours to their reconstruction. Our system however allows the visitor to walk freely through the environment, while we address the inherent problems as well.

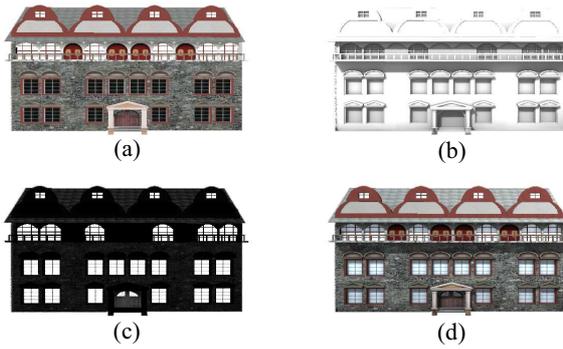


Figure 4: (a) Decal Map, (b), Combined Ambient Occlusion and Shadow Map, (c) Reflectivity (Gloss) Map, (d) Combined Result Illuminated by Prefiltered Environment Maps

An examination considering different input tools (e.g. keyboard and mouse, joystick, or a gamepad) led us to the decision to rely on the touch-screen monitor which was already part of the configuration. In contrast to other options as used in computer games it associates the visual experience and the navigational task in a manner best suited especially for the untrained user.

As for navigating through the virtual environment, we considered several strategies to account for visitors with gaming experience and unexperienced users alike. Finally, we devised two ways of interaction. Inspired by the keyboard-and-mouse input known from games, the camera can be moved by clicking on arrow symbols that control lateral and longitudinal motion as well as yaw and pitch, that way allowing complete freedom. This method of interaction is clearly targeted at more experienced users. Navigation is only limited by collision geometry, which is a sparse 3D model used for the collision detection by *OgreNewt* and the *Newton* physics engine.

The other method is based on a list of predefined viewpoints stored for each building. When the user points at a certain location on the screen, a camera-to-world ray and its intersection with the geometry is computed. If it hits a building, the system checks the associated list of predefined viewpoints and navigates to the one closest to the current position. Should the ray hit the ground, the camera simply moves to the indicated position. This approach allows the user to directly specify the building they want to examine, without getting distracted by the task of getting there on their own.

Supplemental Material

In any real museum, visitors need additional information, usually in the form of information plaques. A virtual environment has to offer some kind of information system as well.

The creators of the *DentroTrento* project describe a hotspot system that marks areas of interest inside their simulation. Likewise, our system allows content creators to place information panels at arbitrary world positions, but instead of only rendering some sort of hotspot icon the information is brought across directly inside the simulation: the information panels display freely configurable HTML content. That way users can easily obtain further information on the area they are looking at without changing into a different view (e.g. a menu). This is achieved by an HTML rendering extension to the *Ogre* 3D engine we developed, using *TerraInformatica*'s free *HTMLLayout* engine. The extension interprets HTML content and renders the result to a texture, which is then applied to the face of an information display inside the scene. The content can be handled in the same way known from any other browser, so when the visitor points to a hyperlink by touching the screen, the system opens the referred page and shows it on the information display. We also implemented basic support for augmenting the HTML content by video files. Note that those HTML-signposts have only one associated predefined viewpoint, which represents the optimal camera position needed to read and interact with the HTML content. When a user points to such a signpost, the mechanism described above will move the camera to that optimal position.



Figure 5: An HTML-signpost

RESULTS AND CONCLUSION

It is important to find new ways of applying the specific know-how from computer games to different applications and to provide highest rendering quality and state-of-the-art interaction techniques to the field of *Serious Gaming*. Using a variety of 3D game related algorithms, techniques and third party software we developed a system that resembles basic structures of a computer game.

We solved the problems inherent in cultural heritage simulations by applying concepts that exist in countless computer games, e.g. the overview map and the arrow-

based navigation. Innovative techniques and ideas have been implemented within the project, most notably the HTML-signposts and the viewpoint-based interaction system.

However, contrary to a computer game development project, our work required less time and fewer resources than a current state-of-the-art computer game, since only a small part of a game's functionality had to be implemented. For example, the system does not offer "opponents", so no artificial intelligence is necessary. Scalability was also not an issue, since the application is used on only one target machine, which was a high-end desktop PC with the most powerful graphics hardware available at the time. We could harness all the power available without ever having to implement options for toggling to a low-detail version of the model or a stripped-down rendering process. In that regard, our project is more related to game console development rather than to a PC game. Furthermore, the lack of typical tasks like level, character, and sound design sets our project somewhat apart from a "normal" 3D game.

FUTURE WORK

Our system can be adapted and improved to facilitate other reconstructed cultural heritage sites. With *Ogre*'s many import/export tools it is possible to convert data from a wide range of modeling software, while our modularized design facilitates the adaptation to other specific needs.

However, even with only the current Dillenburg model in mind the application in its current state holds still room for improvement. It would especially benefit from some "eye-candy" such as particle and animation systems for effects like torches, fountains or movable doors. The *Ogre* engine supports all this in a very straightforward manner, so after incorporating those features into the system it would all become a question of content creation. This next step would also benefit from the use of more recent developments, most notably geometry shaders.

Further down the road the system could be extended to support predefined animated guided tours (comparable to the "virtual guide" used by the *Sagalassos* project), selectable on the overview map. Furthermore, the HTML-signposts have a lot of potential, since the (now static) HTML-pages could just as well be generated by some kind of CMS, maintained by the museum staff and scientists. Added to that, combining the sound-system with the signposts would make it possible to have a narrator read the information back to the user, thus further enhancing the experience for the audience. Another useful addition could be internationalization support, allowing translated versions of both written and audible text sources inside the virtual museum, just

like usually offered by many of its real-world counterparts.

Acknowledgements

Our thanks go to the *Museumsverein Dillenburg* which funded parts of our project.

Special thanks to Matthias Hoffmann, Jeong Il-Sin, Jan Kunze, Michael Schantin, Jens Schlemper, Sascha Schlude, and Tim Wenclawiak, the members of the preceding project which yielded the final Dillenburg model. Their results were the very basis to our work.

REFERENCES

- Blackman S., 2005. *Serious games...and less!* *SIGGRAPH Comput Graph*, 39, no. 1, 12–16.
- Carrozzino M.; Brogi A.; Tecchia F.; and Bergamasco M., 2005. *The 3D interactive visit to Piazza dei Miracoli, Italy*. In *ACE '05: Proceedings of the 2005 ACM SIGCHI International Conference on Advances in computer entertainment technology*. ACM Press, New York, NY, USA, 192–195.
- Conti G.; Piffer S.; Girardi G.; de Amicis R.; and Ucelli G., 2006. *Dentro Trento: a virtual walk across history*. In A. Celentano (Ed.), *AVI*. ACM Press, 318–321.
- Kautz J. and McCool M., 2000. *Approximation of Glossy Reflection with Prefiltered Environment Maps*. In *Proc. Graphics Interface*. 119–126.
- Landis H., 2002. *Production-Ready Global Illumination*. In *ACM SIGGRAPH Course Notes 16*. 87–102.
- Pollefeys M.; Gool L.V.; Akkermans I.; Becker D.D.; and Demuyneck K., 2001. *A Guided Tour to Virtual Sagalassos*.
- Raybourn E.M. and Bos N., 2005. *Design and evaluation challenges of serious games*. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. ACM Press, New York, NY, USA, 2049–2050.
- Todt S.; Rezk-Salama C.; Horz T.; Pritzkau A.; and Kolb A., 2007. *An Interactive Exploration of the Virtual Stronghold Dillenburg*. In *Proc. Eurographics Cultural Heritage*. 213–218.
- Zhukov S.; Iones A.; and Kronin G., 1998. *An Ambient Light Illumination Model*. In *Proc. Eurographics Rendering Workshop*. 45–56.