A Game-Engine Based Virtual Museum Authoring and Presentation System

Victor Mateevitsi
University of Peloponnese
Terma Karaiskaki
22100 Tripoli, Greece
mvictoras@gmail.com

Michael Sfakianos
University of Peloponnese
Terma Karaiskaki
22100 Tripoli, Greece
msfakianos@gmail.com

George Lepouras
University of Peloponnese
Terma Karaiskaki
22100 Tripoli, Greece
+302710372201
gl@uop.gr

Costas Vassilakis
University of Peloponnese
Terma Karaiskaki
22100 Tripoli, Greece
+302710372203
costas@uop.gr

ABSTRACT
In this paper we present a system that facilitates virtual museum development and usage. The system is based on a game engine, ensuring thus minimal cost and good performance, and includes provisions that enable museum curators design the virtual museum without any specialized knowledge. Besides visual and auditory information, museum curators may also provide metadata which provide additional information to the visitor, while they can be also exploited for searching for exhibits with certain properties. A guide is also included in the museum, to present additional information to the visitors and aid them throughout their tour.

Categories and Subject Descriptors
H.5.1 [**Multimedia Information Systems]** - *Artificial, augmented, and virtual realities*; H.5.2 [**User Interfaces**] - *Graphical user interfaces (GUI),* I.3.6 [**Methodology and Techniques**] - *Interaction techniques*

General Terms
Design, Management, Human Factors.

Keywords
Virtual museum; game engines; authoring environment; museum guide.

1. INTRODUCTION
The term VM – Virtual Museum – that was first used by Tsichritzis and Gibbs [1], describes a museum designed in the nominal world of a computer, giving the visitor the illusion of being present in an actual museum. The benefits of a Virtual Museum are numerous with the greatest being the ability to experience a visit regardless the distance to the actual museum. Therefore the development of this technology has met vast evolution over the past years.

In the current paper the term is used to portray an interactive system based on three dimensional graphics and designed to accomplish the same objectives of a "real-world" museum. The development of a virtual environment is a difficult and time-consuming process, requiring highly skilled people of diverse expertise. Usually this task is implemented in the Virtual Reality Modeling Language (VRML), which is very hard to code and deliver all the required functionality, or using specialized environments (e.g. VirTools, Vizard), which are costly and require high programming skills. In this paper we assert that Game Engines can be used instead of VRML and dedicated environments, thus providing the same results. The use of a Game Engine makes the development easier and provides a familiar Virtual Environment for users, which can run on simple Personal Computers without exaggerated system requirements. These advantages of game engines have been well-documented in a number of publications in the last few years, reporting experiences from development efforts regarding virtual reality systems with different foci: [8] discusses a game-engine based infrastructure for remote virtual exploration on PDAs; [9] reports on using a game engine to implement an environment for building musical compositions in a 3D environment in real time; notably, [10] gives an overview of game engine features that facilitate the development of first-person virtual environments and surveys projects that have adopted game engines for developing virtual environments. Moreover a Game Engine provides an expandable environment and usually supports most popular programming languages. The familiarity of users – museum creators and visitors alike- with the game engine environment is owing to the extended use of these engines in the game area and can be considered as a valuable asset, since the time to learn the operational environment is minimized and the user's skills and attention can be directed solely to the achievement of the task and not to the environment idiosyncrasies [7], [11].

Our goal is to create a game engine-based tool that will allow the creation of virtual museums with limited programming and designing skills; museum curators, who are mainly concerned with the arrangement of exhibits, so as to more effectively communicate the *museum message* to visitors, are a major target group of the presented tool. In this context, the tool user must be capable of defining the building that the exhibition will be presented in and, thereafter, easily place and arrange exhibits. The user should be also able to provide additional multimedia information about each exhibit of the museum, making the visiting experience more vivid and realistic. Finally, the tool must allow its user to provide metadata (attributes) for the exhibit, including short and extended descriptions, classifications, keywords etc, which make the visit more informative and facilitate the visitor in locating certain exhibits or exhibits with specific properties. Since the metadata

applicable to different museums vary considerably (e.g. the attributes applicable to a mineralogy museum exhibit are radically different than those applicable to a painting museum exhibit), the tool user should also able to define not only metadata values for each exhibit, but the metadata schema as well.

The remaining of this paper is structured as follows: related work is surveyed in section 2; section 3 presents the tool design principles and decisions, while section 4 discusses the system architecture and provides details on the constituent modules. Section 5 focuses on the user interface, and section 6 concludes the paper and outlines future work.

2. RELATED WORK

There are numerous research activities and papers, focusing on the design process of a VM. Most of the projects try to minimize the gap between the experienced and inexperienced user, with the creation of authoring tools – Graphic User Interfaces (GUI) – that aim to ease and speed up the development of a VM.

Hendricks [2] proposes a system for specifying interactions in Virtual Reality Environments. Traditionally, interactions are specified by the programmer during the implementation phase. Non-computer experts lack of the knowledge and ability to specify and modify these interactions. Creating authoring tools specific for a type of environment reduces significantly this problem. The system consists of a GUI that helps a non-experienced user to create the interactions needed. This system however focuses only on interaction specification, while other aspects of virtual museum development are not considered.

CiVedi – Customized Virtual Environment for Database Interaction – [3] is a scalable system providing a flexible and customizable virtual environment for displaying multimedia content. The system supports the display of multimedia contents through a flexible and customizable virtual environment. The media objects are either stored into a database or dynamically collected from online digital libraries. This proposal is mainly targeted for generic multimedia content, and does not take into account requirements specific to virtual museums, including exhibit descriptions and classifications, and most importantly, customization of the museum building and placement and organization of exhibitions.

The ARCO project – Augmented Representation of Cultural Objects – [4] aims at developing the whole chain of technologies required to help museums create, manipulate, manage and present digitized cultural objects in virtual exhibitions. Dynamic content creation can be achieved, through predesigned visualization templates, which allow designers to create virtual exhibitions very efficiently. The ARCO project adopts X-VRML as its base technology, allowing for content distribution through the WWW, but at the same time imposing considerable requirements on the client that will host the virtual exhibition. Additionally, interaction in X-VRML is not always as easy to specify and customize as in generic programming languages.

The Matthew – Museum-oriented Authoring Tool - THEsis Work – system [5] is a visual authoring tool that helps inexperienced users setting up an exhibition with user defined constrains. Moreover, the system sketches a placement algorithm for the exhibition objects of the museum. The Mathew system, however, does not directly support the formulation of exhibition: the curator can specify constraints on how objects should be grouped and sorted, and the system then interprets these constraints and creates the exhibition, catering for issues such as maximization of area coverage. This procedure is counter-intuitive for museum curators and the result may not be the desired one.

The SAGRES system [6] is an environment built on Web technology that facilitates the presentation of information that resides in a museum adapted to the individual characteristics of each visitor. The added value of the system is a software agent, a virtual guide, which assists visitors via the monitoring of their actions. The agent uses a human-computer interaction paradigm, improvising its actions while helping the visitor, presenting thus a behavior similar to that of a real world human. The SAGRES system however does not support an immersive or semi-immersive museum visit; the software agent aims at supporting the user in locating exhibits of interests, which are displayed as a list and the user can then view details on each exhibit.

In [7] the authors present a case study based on an already developed version of a virtual museum and a newly implemented version that uses game technologies. The authors conclude that the application of game technologies in the context of edutainment is prominent, since these technologies deliver adequate quality for the majority of the target audience of a virtual museum, with the extra advantage of the reduced need for development and system resources. In this work, however, museum creators had to have programming knowledge to specify interactions, while a number of activities had to be manually performed on system configuration files, thus museum curators could not directly use this system. The work reported in this paper extends the work presented in [7] by providing a graphical user interface through which museum curators may perform all activities related to the construction of the virtual museum; it also removes any need for interaction programming taking advantage of the functionalities offered by the game engine. Finally, this work introduces the virtual guide which assists the user in his/her navigation within the museum.

3. TOOL DESIGN

The existing literature and an informal study of similar systems helped in deriving the requirements and the corresponding set of system specifications.

From the designer's point of view it is of the essence to create a system which allows a non-expert to easily develop a virtual museum. To this end, introducing a new exhibit should be a simple and easy to execute procedure. A step by step approach in importing the 3D object along with corresponding documentation was adopted, to make effortless for a novice user to input an exhibit in the virtual museum. This includes placement of the exhibit in the virtual space and arrangement of its presentation properties, as well as producing all the functionalities required by end-users.

These functionalities comprise the generation of documentation, the guiding and the provision of support during the visit in the virtual environment. Generation and presentation of documentation can carried out automatically by the system. A

virtual guide in the form of an avatar can 'read' the exhibit properties from a database and depending on the user's profile can present the corresponding information. The virtual guide can also aid the users during their visit, by answering questions regarding exhibits and offering directions in the virtual space. For example, at any point of the visit the user can ask the guide for the location of a specific exhibit or even for other exhibits similar to the one she sees. The guide will respond with a list of exhibits and the user can select from this list. The guide then can either present a museum map or navigate to show the exhibit to the user.

4. SYSTEM OVERVIEW

After surveying the characteristics, capabilities and restrictions of 11 game engines, both commercial and open source, the Torque Game Engine that was chosen for the development of the project. Some of the advantages of the specific engine are its extended documentation, its built-in world editor and the ability to program in C++ when the engine's scripting language does not provide the required functionality or performance.

Three major phases can be identified in the life cycle of a virtual museum built using the presented system: content production, content management, and visualization.

Content production consists of the creation of (a) digital representations of museum exhibits (b) the building that will host the museum and (c) other digital content that will be used for presentation and visualization purposes (background sounds, narrations, functional items such as stands and display cases for placing exhibits, decorative items etc). In order to create the 3D models of the exhibits, several programs can be used, like Milkshape 3D, with the sole requirement of saving to a format compatible with the Torque Game Engine; as for the museum's building, a variety of programs can be used as well, while in the development of the presented work the Quake Army Knife (QuArK) was employed. The last is a level designer for first person shooting games and it can export building models that are supported by Torque Game Engine.

Content management consists of the population of the content database with the digital exhibits and associated resources, and the establishment of relationships between resources when necessary. For instance, exhibits may be linked with the showcases they should be displayed in, with keywords and taxonomy branches and so forth. An XML storage schema was chosen for this purpose, since it offers a number of advantages: first, the files may be edited with a variety of tools ranging from simple text editors for mini-editing to customized tools tailored to the specific XML schema. Second, XML parsers and XML to memory structure mapping toolkits are widely available and can be employed for reading the data into the visualization engine. Third, interoperability with databases and data exchange with other museums is promoted.

Finally, visualization of the Virtual Museum is performed by the Game Engine itself. Since the behavior of the Torque Game engine is highly customizable, this ability has been exploited to specify the way exhibits will appear on the user interface as well as the interactions available to museum visitors for each exhibit. This has been accomplished by creating script files, which are hooked into the Game Engine.

4.1 Torque Game Engine Overview

The Torque Game Engine consists of different modules that cooperate to provide a smooth experience to the user. The modules communicate each other via an IMC (Inter-Module Communication) module. The adoption of a modular architecture promotes software testability and maintainability, since each module can be tested and maintained separately from the others, while it also allows for replacing a specific module implementation with another if this is required (e.g. when specialized functionality [e.g. driving a particular 3D display] or higher performance is required); note that such replacements will typically be performed as installations of commercial, off-the-shelf (COTS) products and will not encumber the process of developing or visiting the virtual museum.

The modules comprising the Torque Game Engine are briefly presented in the following paragraphs:

4.1.1 Scripting Language Interpreter Module

System interaction with the "real world" (i.e. museum visitors) is defined by the programmer through scripts written in TorqueScript – an object-oriented language through which all system aspects are efficiently accessible and customizable. Using TorqueScript the programmer can define how the system will respond to user input, "spontaneous" system actions (e.g. proposing a specific exhibit for viewing), pre-configured sequences (e.g. a proposed path in the museum) and so forth.

4.1.2 Graphics Module

The Graphics Module renders the Virtual World for presentation to the user. It arranges for the creation of the visual image, taking into account the capabilities of the output device (e.g. standard 2D monitor, stereoscopic features, hardware acceleration etc). An important feature of the Torque Game Engine is the *continuous level of detail* it provides for objects [12]. This feature enables the use of very accurate artifact representations when the visitor stands close to an exhibit, while smaller and more efficient resolutions are used for distant objects and transitions between the different resolutions are smooth.

4.1.3 Sounds Module

The Sounds Module is responsible for playing back the audible content, such as background music, sound effects and synthesized speech. It retrieves sound resources, performs all necessary decoding and finally mixes the content for playback. Similarly to the graphics module, it takes into account the capabilities of the underlying hardware.

4.1.4 Physics Module

The Physics module is equipped with mathematical models and functions that enable it to accurately reproduce the real-world effect of physical laws on objects, including gravity, waves, wind, external stimuli (e.g. forces, blasts) etc. The Torque game engine employs sophisticated models, making the final user experience highly realistic.

4.1.5 Core Module

The Core Module is the "supervisor" of the other modules. It coordinates module execution and communication, and monitors the execution flow of the virtual environment.

4.2 System Overview

The overall system architecture is illustrated in Figure 1, where we can identify three main components, namely the authoring tools, the database and the system logic. These components are briefly presented in the following paragraphs.

4.2.1 Authoring tools

A set of GUI-based tools is available to help the museum developer create the Virtual Museum easily and quickly. The user can add, edit and delete exhibits from a Virtual Museum through a graphical user interface. The interface allows also the user to enter exhibit metadata (name, description, or other attribute values), while exhibit placement in the museum building is also performed visually by clicking and dragging within the museum area. Digital content producers (photographers, 3D modelers, sound composers etc) use the authoring tools to create the necessary digital exhibit representations and additional multimedia resources, and museum curators are responsible for adding metadata and organizing the collection presentation. Museum curators may also define which interactions are available for each exhibit.

Figure 2 illustrates the phase of exhibit metadata editing.

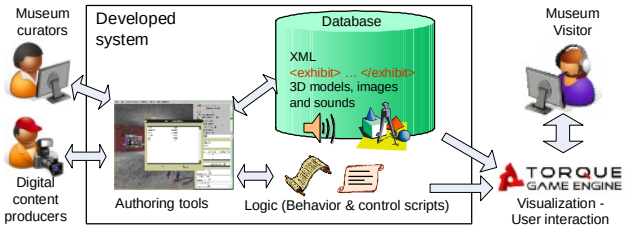

Figure 1. Overall system architecture

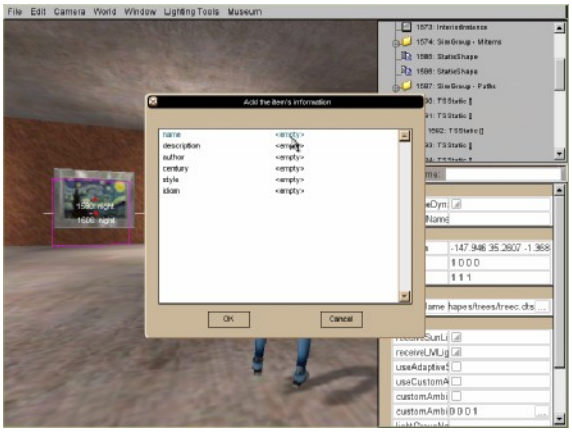

Figure 2. Adding information about an exhibit

4.2.2 System Logic

System logic defines the way that the system loads, executes and interacts with the user. It has been defined through TorqueScript files, and native code execution was used when the required functionality (e.g. XML parsing) was not available in TorqueScript. The first notable functionality implemented through the TorqueScript files is the monitoring of user behavior within the museum and the execution of relevant actions. In particular, when the visitor approaches an exhibit, the exhibit name and description appear for the visitor to read; moreover, an icon becomes visible through which the visitor may access additional information on the item (detailed descriptions, information regarding the creators or any other metadata entered by the museum curators).

The second notable functionality controlled through TorqueScript logic is that of the *museum guide*, an agent visualized as an avatar, which assists the visitor in locating exhibits, gathering and displaying additional information, while it may also offer guided tours. The museum guide is described in more detail in section 5.

4.2.3 Database Module

The Database Module is responsible for the storage and retrieval of multimedia resources and associated metadata for museum exhibits and the presentation in general. An interface to the database module has been built into the Torque game engine, and it communicates with the other Torque engine modules (submission of data retrieval requests and returning of results) via the IMC. The actual storage schema for the managed data (e.g. RDBMS, XML files, XML database or even plain text files) has been kept internal to the database module; this allows for using the most appropriate storage schema in a transparent manner. Currently, metadata are represented as XML files, while multimedia resources are stored as separate files, in the format they will be finally used by the Torque Game engine; the latter choice allows the rapid retrieval and direct usage without the need of format conversions or transcoding. The XML files describing the individual exhibits and museum objects contain links to the associated multimedia resources in the form of path names. Storage of multimedia resources as separate files finally allows their in-place editing with the appropriate tools (e.g. 3D modelers or sound editors), without the need for export and import.

The same interface is employed by the authoring tools to store, retrieve and update the database contents.

One of the major design decisions for our system was that it should be able to accommodate the metadata needs of any museum. Since these needs are quite diverse among museums due to the different nature of their exhibits and/or differences in the information that they decide to present to the visitors, it was decided that each museum should be able to tailor the metadata schema to its own, particular needs. The authoring tools environment encompasses this functionality, by allowing the museum curator to specify which attributes (metadata elements) apply to the museum exhibits.

Figure 3 illustrates the metadata schema definition process. Besides the user-defined metadata elements, the following three attributes are compulsory for the metadata schema:

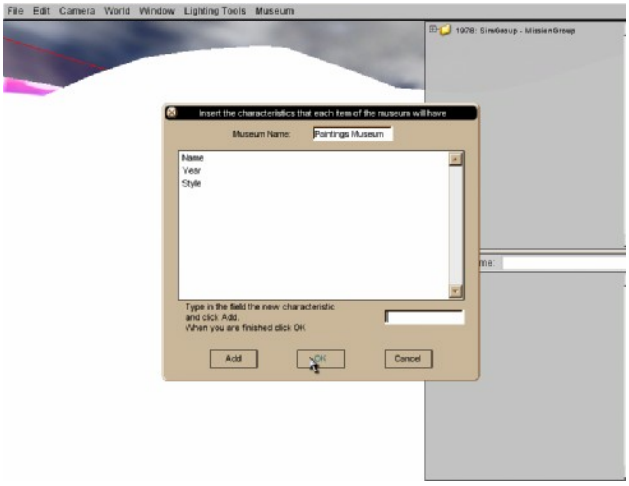

Figure 3. Specifying the metadata attributes for a museum

- **id**: the id of the exhibit that identifies it uniquely. It is system-assigned and used internally by the system for maintaining associations between museum areas, exhibits and available interactions.
- **name**: a name for the given exhibit. It is displayed when the visitor approaches an exhibit and in result lists when the user poses queries to the system.
- **description**: A short description of the exhibit, displayed when the visitor moves close to an exhibit.

Figure 4 depicts an excerpt from the XML file describing exhibits in a paintings museum.

```
<?xml version="1.0" ?>
<objects>
  <object id="12">
    <name>Guernica</name>
    <description>Guernica is a painting by Pablo Picasso, ….</description>
    <author>Pablo Picasso</author>
    <century>20th</century>
    <style>Surrealistic</style>
    <idiom>War</idiom>
  </object>
…
```

Figure 4. XML description excerpt

5. USER INTERFACES

Our system provides two different types of user interfaces. The first one is the creators' interface and is used during the museum development phase by digital content producers who populate database with multimedia content and museum curators who organize this content into exhibitions and provide the exhibit metadata. The second one is the visitors' interface, which is used by visitors to navigate through the museum, visualize its content and access the exhibit metadata. In the following paragraphs, the main characteristics of these interfaces are discussed.

5.1 Creator Interface

Initially the creator begins with an empty world; at this stage, the terrain of the world must be set up (see Figure 5), the territory and scenery must be designed and the building in which the virtual exhibits will be placed must be created. This task will be undertaken by a digital content producer, although a preconfigured world can be drawn from a library, skipping the world setup step altogether. Following the formulation of the world, or even in parallel to it, digital content producers can populate the multimedia database with the digital representations of the exhibits and other pertinent multimedia content.

Once the world has been completed and digital representations of exhibits have become available, museum curators can begin arranging the exhibits into exhibitions. In order to place an exhibit within the museum area, the museum curator needs simply to choose the relevant menu command and select the area in which the exhibit will be displayed. The placement procedure is performed visually, by clicking and dragging and scaling is allowed to facilitate the maintenance of real-word size ratios. Curators may place exhibits on walls, stands, showcases or even on the room floor, opting for the arrangement that will better serve visitors in (a) navigating to the exhibit [e.g. an ancient earring exhibit may not even draw the visitor's attention unless placed in a show case] and (b) examining all interesting aspects of the exhibit [e.g. placing a statue close to a wall clearly inhibits the inspection of all its sides].

Museum curators also define the exhibit metadata (cf. Figure 2). This can be performed either before or after the placement of an exhibit in the virtual museum world. Defining metadata for exhibits non-placed exhibits effectively contributes to the creation of a library, from which exhibits can readily be drawn and placed into a virtual museum.

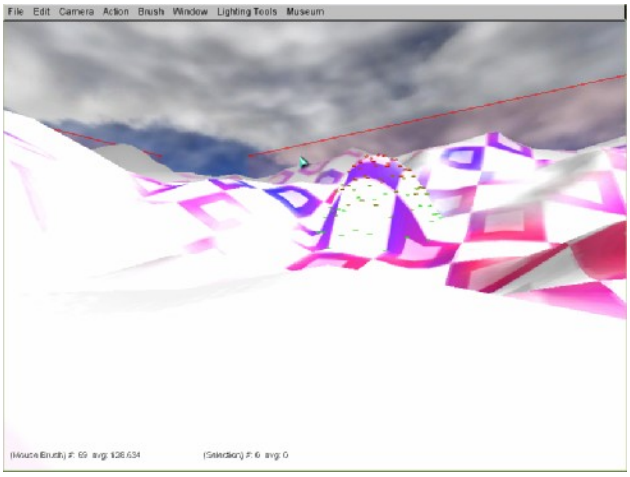

Figure 5. A new empty world.

5.2 Visitor interface

The visitor interface is the interface presented to museum visitors. One of the choices offered by the Torque game engine regarding this interface was the choice of the perspective, which may be either first person (the visitor "sees" through the eyes of the avatar s/he controls) or a third person perspective (a camera follows the user avatar and displays both the avatar and its surroundings). We adopted the first person perspective as the default in our museum, since it has been found to give visitors a better sense of immersion and personal involvement. The visitor is allowed to switch to the third person perspective, if s/he wants to see a more generic view of the avatar surroundings.

Another feature of the Torque game engine that was maintained (i.e. not restricted) in the visitor interface was the existence of a

second camera, which is allowed to move independently of the user's avatar. This camera can be used for two purposes: first, it allows the visitor to move and examine the virtual museum without loosing his/her current position, since the avatar stands still while the second camera is moving. Second, since the user avatar obeys the law of gravity, it always remains on the floor, thus exhibits that are placed high in the museum building will appear distorted due to the high angle at which they are viewed. Note that placement of exhibits in high positions cannot always be avoided, if the real-world ratios of objects need to be maintained. For example, in a museum hosting a digital representation of the Ishtar gate, under the first person perspective the visitor would view the decorations at the top under an angle of 80,7 degrees, when standing at a distance of (the equivalent of) 2m from the exhibit, which will lead to a highly slanted view. (The height of the Ishtar gate is 14m while the avatar eyes are assumed to be at 1.70m.) Moving further from the exhibit will reduce the angle, but will also reduce the level of detail available to the visitor. By employing the second camera, the visitor may close up to the target area and view it without any slant or distortion.

When the visitor moves close to an exhibit, the exhibit's description is automatically displayed at the top of the screen (see Figure 6). Another option available to the visitor at this point (i.e. when close to an exhibit) is to find exhibits that are related to the current exhibit, as such a relation is indicated through metadata values (e.g. same creator, same name and so forth). In order to perform this search, the user simply chooses the attribute of interest (Figure 7), and the system responds with a list of matching exhibits. A "General search" is also available, where the user types in the value that the selected attribute must match.

Figure 6. Zooming into an exhibit

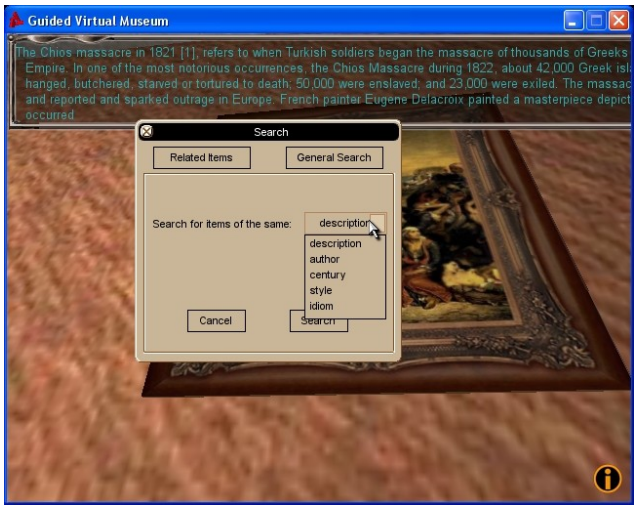

Figure 7. Searching for related exhibits

Once the visitor has chosen an exhibit from the result list, the museum guide undertakes the task of assisting the visitor to navigate to the chosen exhibit. The museum guide is an avatar that in these cases firstly approaches the visitor and then starts moving towards the location of the selected item; the visitor must simply follow the guide in order to reach the selected exhibit. This approach was preferred against teleporting the visitor to the selected exhibit, since it resembles more closely the real-world method and gives visitors a better sense of involvement. Figure 8 depicts a screenshot where the guide has leaded the visitor to the selected exhibit.

Figure 8. The guide has leaded the visitor to the selected exhibit

6. CONCLUSIONS

In this paper we have presented the design rationale, the overall architecture and detailed components and interface descriptions of an integrated environment for authoring and navigating into virtual museums. The presented environment is based on a game engine, which guarantees a small cost and good performance, while the interfaces have been appropriately customized to

facilitate the tasks of both authors and visitors. Compared to previous approaches that have implemented virtual museums using game engines (in particular with [7] for which more details are available), the proposed environment has a number of important advantages, including (a) removing any need for programming (b) all virtual museum creation functionality is available through user-friendly GUIs that can be directly used by museum curators (c) a virtual guide has been added to assist the user in navigating within the exhibition (d) better accuracy for displayed artifacts, through exploiting the advanced capabilities of the game engine graphics subsystem.

Future work will focus on enhancing the intelligence of the museum guide, enabling it to deduce the preferences and likings of the visitor and suggest further exhibits for viewing. Dynamic formulation of exhibition areas based on user queries as well as pre-configured tours will be also investigated.

7. REFERENCES

[1] Tsichritzis D, Gibbs S. 1991 *Virtual museums and virtual realities*. In: Proceedings of the International Conference on Hypermedia and Interactivity in Museums, Pittsburgh, PA..

[2] Hendricks, Z., Marsden, G., and Blake, E. 2003. A meta-authoring tool for specifying interactions in virtual reality environments. In Proceedings of the 2nd international Conference on Computer Graphics, Virtual Reality, Visualisation and interaction in Africa (Cape Town, South Africa, February 03 - 05, 2003). AFRIGRAPH '03. ACM, New York, NY, 171-180. DOI= http://doi.acm.org/10.1145/602330.602362

[3] Mazzoleni, P., Bertino, E., Ferrari, E., and Valtolina, S. 2004. CiVeDi: a customized virtual environment for database interaction. SIGMOD Rec. 33, 3 (Sep. 2004), 15-20. DOI= http://doi.acm.org/10.1145/1031570.1031574

[4] Wojciechowski, R., Walczak, K., White, M., and Cellary, W. 2004. Building Virtual and Augmented Reality museum exhibitions. In Proceedings of the Ninth international Conference on 3D Web Technology (Monterey, California, April 05 - 08, 2004). Web3D '04. ACM, New York, NY, 135-144. DOI= http://doi.acm.org/10.1145/985040.985060

[5] Grimaldi, A. Catarci, T. 1999. The Matthew system for creating virtual museums. In IEEE International Conference on Multimedia Computing and Systems (Florence, Italy, June 06 – 11, 1999), ISBN: 0-7695-0253-9

[6] Moraes, M. C., Bertoletti, A. C., da Rocha Costa, A.C. 1999. Virtual Guides to Assist Visitors in the SAGRES Virtual Museum. 19th International Conference of the Chilean Computer Science Society, p. 35-42.

[7] Lepouras, G., Vassilakis, C. 2005. Virtual Museums for All: Employing Game Technology for Edutainment, Virtual Reality Journal, Vol. 8, 2005, pp. 96-106.

[8] Chiara, R. D., Erra, U., Petta, A. Scarano, V., Serra L. 2007. An infrastructure for remote virtual exploration on PDAs. 11th International Conference Information Visualization (IV'07)

[9] Bertacchini, F., Bilotta, E., Gabriele, L., Mazzeo, V., Pantano, P., Rizzuti, C., Vena, S. 2007. ImaginationTOOLS™: Made to Play Music. K.-c. Hui et al. (Eds.): Edutainment 2007, LNCS 4469, pp. 369–380.

[10] Trenholme, D., Smith, S. 2008. Computer game engines for developing first-person virtual environments. Virtual Reality

[11] Robillard G, Bouchard S, Fournier T, Renaud P. 2003. Anxiety and presence using VR immersion: a comparative study of the reactions of phobic and non-phobic participants in therapeutic virtual environments derived from computer games. CyberPsychol Behav 6(5):467–475

[12] Garage Games, 2008. Torque Game Engine Advanced: Comparison. http://www.garagegames.com/products/torque/tgea/featurecomp