

# Interactive Virtual Gallery

Tomas Ullrich<sup>1</sup>, Jiri Zara<sup>2</sup>

<sup>1</sup> Charles University in Prague, the Faculty of Mathematics and Physics  
<sup>2</sup> Czech Technical University in Prague, the Faculty of Electrical Engineering

## Abstract

*Interactive virtual gallery* is a project designed to help *gallery visitors* and to facilitate the work of *gallery designers* preparing art exhibitions and other displays.

It gives *gallery visitors* the possibility to see exhibitions in virtual reality.

It enables *gallery designers* to model a 3D gallery, to design an exhibition (to lay out exhibits) and to obtain some information needed for a real gallery construction.

## 1 Introduction

Application *Interactive Virtual Gallery* [1] consists of a few modules:

- A) A **gallery editor** for designing the geometry of an exhibition place and modeling the 3D gallery,
- B) A **texture and 3D object manager** for editing lists of textures and 3D objects that can be inserted into the gallery
- C) An **exhibition editor** for laying out exhibits in the gallery

All modules work separately and they save/load all information to/from a joint database. This application is accessible through the Internet but it is also designed in such a way that it can be installed locally using the local installed resources.

*Interactive virtual gallery* is especially designed to facilitate the work of gallery designers. However, gallery visitors will find this tool useful, too. Before visiting a gallery in the real world, they will look up its 3D model on the Internet that has been previously created by this program. Once they find it, they will be able to walk through the gallery using the navigation in virtual reality. After that, it will be easier to find their way around the real gallery.

The *gallery editor (module A)* enables users to design a multistorey gallery with many rooms. The gallery can accommodate any 3D objects like furniture or spotlights. When the user is satisfied with the gallery design, he/she commands the editor to create a 3D model of the gallery. This model is stored in a VRML file format [2] and is used later in the *exhibition editor*.

The *texture and 3D object manager (module B)* maintains the lists of textures and 3D objects created by the user.

The *exhibition editor (module C)* uses one of the 3D models created by the *gallery editor*. The model is displayed in a VRML browser using the principle of virtual reality, where the user can place exhibits. There are two kinds of exhibits: 2D i.e. pictures or paintings and 3D i.e. statues or sculptures.

The following text is structured as:

- 2/ *Related Work*
- 3/ *Design of the virtual gallery*
- 4/ *Implementation details*
- 5/ *Gallery editor features*
- 6/ *Texture and 3D object manager features*
- 7/ *Exhibition editor features*
- 8/ *Conclusion*

## 2 Related Work

If we search through the Internet for any related work, we discover, that virtual galleries are usually static. That means the user is unable to change the scene, he/she can't move any exhibits, he/she can only watch. Such galleries can be divided into the following categories:

### Pictures only, 2D world

This kind of a virtual gallery is widely spread. Each individual room has its own HTML page. Pictures placed there are ordinary images inserted into the page by HTML tag <IMG>. These galleries could be found for example at [8,9].

### Pictures with "QuickTimeVR" models

This category is very similar to the previous one, but we can find *QuickTimeVR* models there. This virtual tour uses the principle of panoramic virtual reality. Pictures and video playback are used to create the vision of a 3D world. Individual pictures are linked into a looped video record. Using this technology we can display a statue, which the user will be able to turn left or right by playing the video backward and forward.

An example is Prague National Museum's Internet page [10].

### Static galleries, 3D world

This category contains 3D models of galleries usually produced in VRML. There are pictures hanging on walls

<sup>1</sup> tull6627@barbora.ms.mff.cuni.cz (or tom\_ullrich@hotmail.com)

<sup>2</sup> zara@fel.cvut.cz

and some 3D objects making the scene more real (e.g. a spot-light or any kind of furniture). The scene is static, no changes can be made – no exhibits can be removed or added. These galleries can be found on pages [11,12].

An interesting application that produces these 3D gallery models is described in [14].

### Dynamic galleries facilitating change

Dynamic virtual galleries enable users to change the scene. There are very few of these galleries on the Internet. These galleries usually don't let users manipulate with exhibits. The users are only able to add some predefined objects like furniture or wall textures. This gallery can be found on *3D-AnyWhere* pages [13].

## 3 Design Of The Virtual Gallery

The section *Modules description* gives an account of all modules (figure 1), especially of the gallery construction. The next section *Project realization* describes the way this project was built.

### 3.1 Modules description

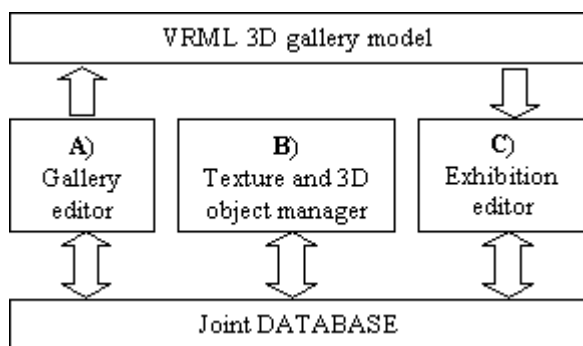


Figure 1: Modules architecture

Each gallery, this program can produce, makes up an imaginary building, which consists of several floors. In the **gallery editor, module A**, (figure 2), each floor is laid out as a ground plan with separate rooms. The rooms can occupy any space, the walls don't have to be mutually perpendicular. The user can define the appearance of each wall, floor and ceiling surface by setting appropriate texture or color. He/she can place any 3D objects stored in the database into the gallery (e.g. chair, table, case, pedestal). There can be windows and doors inserted in each wall according to user's wish. Doors of different rooms (even with different levels of height) can be connected or they can be replaced with stairs (figure 3). The user is also supposed to place some exhibition panels (called *holders*), on which exhibits are going to be placed later in the *exhibition editor*.

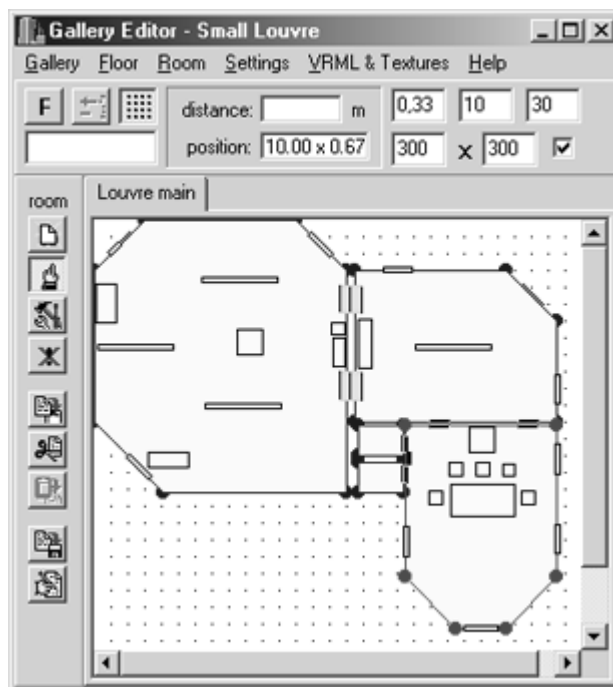


Figure 2: Gallery editor

There are two kinds of holders:

- vertical* for holding 2D exhibits (e.g. pictures)
- horizontal* for holding 3D exhibits (e.g. statues)

Vertical holders can be placed on each wall and any upright side of a 3D object (like a brick exhibition panel).

Horizontal holder is automatically set on each floor but it can also be placed on the top of a 3D object (e.g. on a pedestal).

The user can define viewpoints, which are used later on in *exhibition editor*. They determine fixed points of view for virtual gallery visitors (called avatars). The viewpoints facilitate visitors' walk round the gallery. The user can also determine the order in which he/she will walk through the rooms and set the viewpoint sequence accordingly.

The gallery geometry can be saved to and loaded from the database. The user can also export it to and import it from a special file format called *gal*. This feature is needed when a gallery is transferred from one designer's personal computer to another's.

*Database manager* is a facility that supervises and maintains the database. It is used both by the *gallery editor* and the *texture and 3D object manager*. *Gallery editor* maintains the exhibit lists whereas **texture and 3D objects manager, module B**, is responsible for editing lists of textures and 3D objects.



Figure 3: Multistorey gallery with stairs

**Exhibition editor, module C**, (figure 4) controls a concrete gallery, which has been previously created in the *gallery editor*. In the 3D model of this gallery, designers are able to add or remove exhibits, change their position or turn them (3D objects only).

Any created exhibition can be saved and loaded from the database. Visitors can walk through the gallery and see all exhibits placed there.

A search button enables designers and visitors to look up any picture placed in the gallery. After the picture is found, the virtual visitor's viewpoint is set two meters in front of this picture.



Figure 4: Exhibition editor

### 3.2 Project realization

There are two different ways, how this project could be realized:

- the application can be accessible only *through the Internet*
- it can be a *standalone application*
  - The main advantage of the Internet access version is the fact, that no special software installation on client side is needed, e.g. the database server. Users install only

a special plug-in into their Internet browser, which enables the rendering of 3D data.

This solution brings about some problems: there might be many Internet users all willing to make their own galleries, but there is only one database to share. This means some access restriction. But how will they share the lists of textures and exhibits? Who will be responsible for the maintenance? And there is another problem caused by the security mechanism built in the programming language. The possibility of writing data into a file on a distant computer can't be taken for granted.

b) On the other hand, if the application is installed locally, no problem with sharing the database will arise as no security mechanism prevents the application from writing into a local file.

The biggest disadvantage of the b) possibility is the fact that only a limited number of people will have the access to this application. The possibility to walk through the virtual gallery before visiting its real copy will be denied to the majority of the public.

The designers, as well, will have problems to distribute their galleries to other designers or visitors.

The solution we used is a kind of a compromise between the a) and b) possibility. The application was created in such a way that it enables the Internet access possibility, but it can also be used locally with the help of local resources.

The difference between the designers and the visitors is in the range of database access rights. The visitors have the right to read only (corresponding to SQL command SELECT), while the designers have all rights. There is no protection in the *exhibition editor* prohibiting visitors from moving pictures from one place to another. However, if a visitor tries to move a picture, the database will signal an error message saying: "*database command failed, access denied*".

## 4 Implementation Details

MySQL database server has been chosen for this project deliberately, because it is widely spread and very reliable.

The gallery editor and the textures and 3D objects manager are programmed in Borland C++ Builder™.

The exhibition editor consists of two parts:

- the *main control center*
- the *3D scene* itself

Both are included in one Internet HTML page, which is the main "executable" file. The control center is a JAVA applet. The 3D scene management is programmed in VRML - Virtual Reality Modeling Language [5]. This is an effective language for 3D scene representation, in particular on the Internet. The 3D scene management uses the ECMAScript [4], which is very similar to JavaScript, known from HTML. The communication between the JAVA applet and the VRML world is executed by the EAI interface [3] (External Authoring Interface, figure 5).

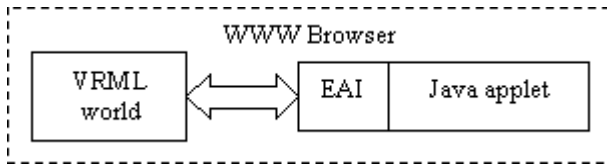


Figure 5: EAI interface function

To present a VRML file, a VRML Internet browser must be installed. We recommend the *Cortona VRML client* VRML browser for this project [6].

## 5 Gallery Editor Features

Gallery editor allows designing a static model of the gallery. Here is the list of functions for the use of designers:

1. Load/Save one gallery from/to a database after successful login to the database
2. Create a new / remove gallery
3. Add/remove floor to/from the gallery
4. Export/Import gallery to/from *gal* file for transferring gallery from one computer to another (involves exporting/importing database mapping to needed textures and 3D objects)
5. Set basic dimensions and supporting info for each floor (e.g. grid, zoom)
6. Create a new room with polygonal layout, define ground and ceiling elevation
7. Change room position, remove room
8. Change room parameters (e.g. enlarge a wall, delete a wall)
9. Copy, cut and paste any room(s) to a different or same floor
10. Set colors/textures for ground/wall/ceiling in the room
11. Insert VRML doors and windows into walls, change their position and delete them
12. Import and displace non-interactive VRML objects into a room (e.g. furniture, a pedestal), delete them
13. Create and displace vertical and horizontal holders
14. Define viewpoints with appropriate parameters
15. Connect doors (having the same type) from two rooms in order to make one door-alley
16. Connect two rooms from different floors with stairs
17. Change room order, in which viewpoints will be created
18. Export resulting gallery to a VRML file for the use of *exhibition editor* (create ceiling on/off)
19. Add, remove an exhibit (picture or an VRML object, e.g. a statue), change its parameters (scale, dimensions)

## 6 Texture And 3D Object Manager Features

Textures and 3D objects manager maintains the lists of textures and non-interactive 3D objects. These are the manager's features:

1. Load/Save texture and 3D-object lists from/to a database
2. Add/Remove a texture to the list from a image file (JPG, GIF and PNG formats recommended)
3. Add/Remove a 3D object to the list from a VRML file (usually WRL files)
4. Set 3D object type: window/door/other

## 7 Exhibition Editor Features

Exhibition editor is a highly interactive module working in a 3D virtual environment. The main purpose of its use is to insert exhibits into a gallery space. Paintings (2D images) can be placed on vertical holders, while 3D objects can be on the horizontal ones. A user can change position of exhibits interactively. The list of available exhibits is displayed using small previews in a special "selection area" (figures 7 and 8).

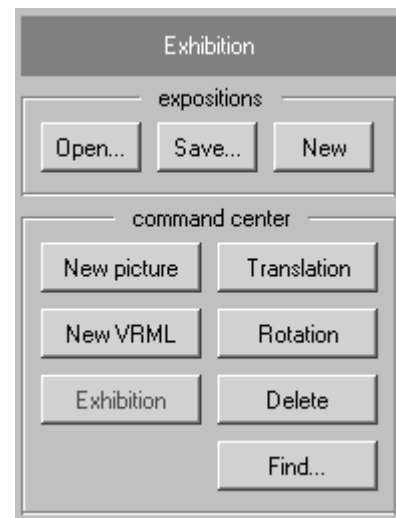


Figure 6: Exhibition editor, control center

Exhibition editor features follow:

1. Choose gallery to load to the 3D scene after successful login to database
2. Select an exhibit from the list and put it on a holder in 3D space
3. Change position of already inserted exhibit in 3D space
4. Remove an exhibit from 3D space and put it back to the list of available exhibits
5. Jump to selected predefined viewpoint
6. Find a position (in 3D space) of a picture exhibit specified by its name or appearance
7. Save/Load/Delete created exhibition to/from database
8. Remove all exhibits from gallery (in 3D space)



Figure 7: Exhibition editor, a list of available pictures with preview

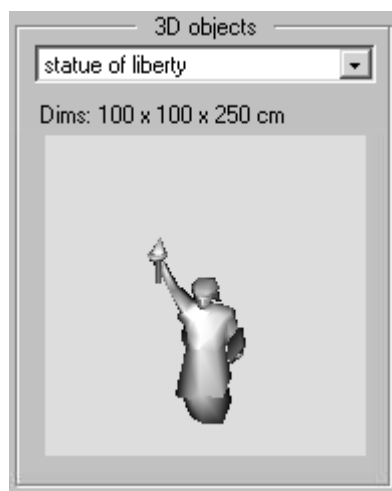


Figure 8: Exhibition editor, a list of available 3D objects with preview

## 8 Conclusion

Interactive Virtual Gallery is a powerful and useful tool for cultural heritage presentation and administration. This tool enables designers to decide the ideal position of a picture in the virtual reality world. In reality, each picture will go straight to its right place where it can be seen in its best, it will perfectly match other pictures, the wall, the room and the overall space. Once a picture is placed on the wall, this will certainly be its ideal position (as it was determined by the *Interactive Virtual Gallery*) and the picture will never have to be shifted to another place for some reason or other.

The structure of the application has been designed with the aim to implement it step by step with possibility to extend its functionality in the future and to add various new features depending on users' wishes and needs.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Spring Conference on Computer Graphics, Budmerice, Slovakia  
Copyright © ACM 2002 1-58113-608-0/02/009...\$5.00

## References

- [1] *Interactive virtual gallery* home page:  
<http://www.novacompc.cz/tom>
- [2] VRML specification:  
ISO/IEC 14772-1:1997 – *Information Technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 1: Functional specification*, 1997
- [3] EAI specification:  
ISO/IEC 14772-2:1999 – *Information Technology – Computer graphics and image processing – The Virtual Reality Modeling Language (VRML) – Part 2: External Authoring Interface*, 1999
- [4] ECMA specification:  
ISO/IEC 16262:1998 – *Information Technology – ECMA Script language specification*, 1998
- [5] VRML & EAI:  
<http://www.web3d.org/WorkingGroups/vrml-eai/>
- [6] Parallel Graphics:  
<http://www.parallelgraphics.com>,  
<http://www.outline3d.com>
- [7] QuickTime:  
<http://www.quicktime.com>
- [8] The Art On The Net Gallery:  
[http://www.art.net/TheGallery/the\\_gallery.html](http://www.art.net/TheGallery/the_gallery.html)
- [9] Louvre Museum - home page:  
<http://www.louvre.fr>
- [10] Prague National Museum home page:  
<http://www.nm.cz>
- [11] Virtual Reality Matrix, 3D Art Gallery pages:  
<http://www.vr-matrix.com/3d-gallery/index.html>
- [12] Wanadoo – home page:  
<http://perso.wanadoo.fr/woldenberg.arts/html/frameoilus.htm>
- [13] 3D Anywhere - home page:  
<http://www.3danywhere.com/3da-adv-simulation.html>
- [14] S. Hrk: Virtual Art Gallery. In Proceedings of the 5<sup>th</sup> Central European Seminar on Computer Graphics, pages 185—194, Vienna University of Technology, 2001  
<http://www.cg.tuwien.ac.at/studentwork/CESCG-2001/SHrk/>