Paper for CAA98
26th Conference on Computer Application in Archaeology
Barcelona 25, 26, 27 and 28 March 1998

Brogni A., Bresciani E., Bergamasco M., Silvano F.

PERCRO Laboratory
Scuola Superiore S.Anna

University of Pisa
Egyptology

Pisa - Italy

AN INTERACTION SYSTEM FOR THE PRESENTATION OF A VIRTUAL EGYPTIAN FLÛTE IN A REAL MUSEUM

Abstract

This paper addresses a new paradigm of virtual interaction with virtual reconstruction of archaeological artefacts. At present the use of synthetic images for the presentation of archaeological works is performed by the use of computer screens, projectors or retroprojectors capable of providing immersive experiences in the virtual reconstruction scenery. More sophisticated system exploit the so called CAVE, in order to allow a deeper sense of immersion in the reconstructed site.

The new paradigm of visual interaction we have considered is related to the representation of virtual archaeological artefacts on a portable flat screen; this is held by the human observer and pointed along the line of sight to the spatial real position where the object would be located in the museum. It is assumed that the virtual artefact occupies a physical position in the real three-dimensional space and that the observer acquires its 3Dimages through a graphical window located on the screen.

The real time representation of the exact perspective and dimensions of the virtual object on the video screen is controlled by monitoring the position and orientation of the screen itself by a tracking sensor.

During the exploration of the virtual space, the observer can obtain some historical and archaeological information about the artefact and its production phases, by touching the screen on graphic buttons located on the side of the window.

The system includes a SiliconGraphics Indigo2 R10000 High Impact workstation, connected with a commercial flat screen (weight 3.5 Kg, width 55 mm), with touch-screen incorporated. The tracking sensor is a Fastrak Polhemus, which can detect electromagnetic fields emitted from a source, and describes position and orientation giving analog signals; this sensor is physically attached to the portable screen.

The software application comprehends the following modules:

- graphical interface for the consultation and exploration system; this module was realised by programming with the Xforms library and C++ language;
- 3D model of the artefact, created by the PowerAnimator of Alias|Wavefront graphical software;
- sensor driver, for the communication of the position and orientation data;
- the OpenGL library in C++ language has been used.

Experimental results were obtained by utilising the above system for the representation of an Egyptian blown glass flûte, decorated with cut gold leaf and coloured paints; this flûte dates back to the III sec. A.D. and it was discovered in a tomb of the necropolis of Sedeinga, Sudan, in the 1970. The 3D shape of the flûte has been reconstructed from a series of photographies and measurements. The decorations describe four human figures and some animals, which have been painted on the glass, warping the pictures on the model, after some deformation to correct the optical distortions from the cylindrical chalice to the planar photography.

Introduction

In the last few years we have witnessed a constant evolution in the relationship between the technical and the archaeological sciences, which from the simple data recording is developing in a closer relation.

The work described intends to develop a new system for a fuller enjoyment of works of art by trying to make the most of the new interaction technology. We have developed and carried out the plan, on a new paradigm of visual interaction between the user and the virtual artefact, and we have used a portable flat screen like a real window on the virtual world.

The O_x.y.z datum-system is connected to the window, and therefore to the observer and his point of view. The position and orientation of the screen are monitored by a 3D tracker and we can reproduce them in the virtual ambient. The screen is held by the user and pointed along the line of sight to the real position, where the artefact would be located, so that the user is immediatly informed about the physical dimension of the artefact.

The system also allows total access to the information about the archaeological artefact, by means of an ambient with text-windows and buttons (Graphic User Interface), which allows us to interact with the application and choose the consultation.

At present this application is used for the representation of an Egyptian glass flûte, but it's a suitable platform for every artefact, and the virtual ambient could even be a tomb, or an ancient palace.

The visitor gives his orders by touching the screen on graphic buttons, located on the side of the screen, which are easy to hit with the same fingers holding the screen. At the same time the tracking sensor gives all the information about the movement in the real space relative to the central system, which can prepare the new image for the screen according to the new point of view.

Unlike the traditional 3D browser, where the user chooses what to look at and the browser moves the virtual camera, in this system the user actually walks in the space, and the point of view of the camera follows his movements.

The blue glass flûte from the Schiff Giorgini Collection

The subject of our virtual reconstruction is a blue glass flûte from the Schiff Giorgini Collection, now in the Egyptological Museum of the University of Pisa, Italy.

The flûte is 20,1 cm high, and it consists of a base, a stem and a cup.

The flûte was discovered in 1970, along with an identical object of the same origin and form currently preserved in the Museum of Khartoum.

The flûte was retrieved from a tomb in the Sedeinga necropolis, situated between the second and third Nile cataract, and subsequently donated to the University of Pisa by its discoverer, Michela Schiff Giorgini.

Tomb WT8, where the flûte was located, belongs to the final period of the necropolis, i.e. the second half of the III century B.C., and hosts the remains of an individual whose rank, judging from the richness and quality of the funerary apparel, had to be rather high.

The burial place did in fact contain some 3000 glass fragments, from about 30 very differently shaped vessels. Both the flûtes were found broken into a large number (about 70) of fragments; they were probably purposely shattered during the funerary rites celebrated in the tomb during the burial of the dead.

The Greek inscription along the rim of the chalice and the peculiar workmanship, suggesting the work of very skilled craftsmen, emphasise a strong Alexandrine influence, albeit supplemented by undeniable indigenous motifs and themes of a genuine Pharaonic tradition.

3D shape reconstruction

We have used the Alias|Wavefront PowerAnimator 8.5 and StudioPaint3D 4.0 software to model the flûte. We decided to build the model as a composition of many meshes in order to obtain a good level of realism and high visualisation performances.

The flûte is irregularly shaped: the base, as well as the stem, are uneven and they aren't symmetric. The artefact shows the traces of having been reassembled from many fragments and it leans on one side. The modelling of the artefact was piecemeal and we achieved a satisfactory degree of accuracy by correcting every little detail. We divided the flûte in eight parts, called base, stem, bottom, low decoration, middle decoration, high decoration, top and inside. This subdivision allowed us to work textures better, by working on the decoration in detail and using just the material colour for the other parts.

The top side was separated because it presents a surface with acute angles and a very sharp circular relief. We deemed it more convenient to model it alone, because the final model would be a polygonal model and it's important to use a large number of polygons just where we need them to keep the total number low enough for a good application performance in real time.

Our solution for a real time rendering was the use of the VRLib graphic library, in progress in our laboratory, based on the OpenGL library. We chose the RTG format, a typical format used in 3D commercial game animations, to save the final model because it allows us to save an ASCII text with information about the vertices, the material colour and to connect texture file to the objects.

The measurements of the flûte, for reconstruction, were taken from many photographs, three for every 45 degree angle, one for the top, the middle and the bottom respectively. In this way we could adjust the right images and obtain the model of the flûte, by working with the advance functions for the surface manipulation of PowerAnimator.

We built the base like a revolution shape and we completed it by moving the single vertices of the mesh, to obtain a good reproduction.[ALIAS1_98]

The stem is uneven, so we preferred to model it with a different technique: we built many circles, on different heights, with different radii, following the geometric data from the pictures, then we linked them together with only one surface; we therefore obtained a strange cylinder, unsymmetrical and irregular, like in the real flûte.

All the other parts are built like revolution solids, from convenient curves, and fitted together to look like a single object. From the NURBS mesh, we converted it in a polyset and saved it in the RTG format.[ALIAS4_98]

Since we work with real time applications, and we need to keep the total number of polygons displayed on the screen at minimum, we used the division to dedicate more polygons to the complex side, like the top or the base, and less to the regular parts, like the middle, which is almost a cylinder.

We obtained the final number of polygons after many framerate tests with different polyset; we opened an OpenGL window 800x600 pixel on the screen, like the final window on the application, and we displayed moving objects. For the test we used a SiliconGraphics Indigo2 R10000 HighImpact workstation with 64 Mbytes RAM.

The speed parameters are the following:

No. of polygons	geometric definition	framerate
1106	very poor	86 (frame/sec)
2078	poor	72
3997	good	36,5
6284	very good	18,1
8248	excellent	10,5

The final mesh consists of 3997 polygons:

base	912,
stem	304,
bottom	336,
low decoration	224,
middle decoration	192,
high decoration	192,
top	1316,
inside	521.

Besides the flûte we also built other models, because we needed a pedestal for the flûte and a little slab for the planar vision of the decoration. The pedestal consists of 240 polygons, just enough for a good definition but not too numerous to slow down the system .

We attached the decoration textures on a slab, consisting of 304 polygons, to give the user a complete view of the scene represented.

Texture mapping

The first step was fixing the right value for the material colour and the transparency; by loading the photographs on the computer with a scanner, we could measure the chromatic RGB value (Red Green Blue) for the ground colour. We matched these values with the light in the virtual ambient to define the final RGB value.

We fixed the transparency value we did by putting a little rod inside the flûte and changing the value and choosing the best condition. We can control the transparency value by a fourth number, called alpha, to add on RGB, with a range from 0 (completely opaque) to 1 (transparent).

We used a different technique for the decorations because in this instance transparency is needed just where the flûte is unpainted. We therefore built a transparency map, on a grey scale, to fix the value in detail; we saved the map as a file and we linked it to the RGB file of the model.

We had to reconstruct the picture of the decorations, and we did so by manipulating the photographs. By photographing a cylinder we obtained a distorted image and we used just the little central strip; with a technique, called warp, we cut the strip and we stretched it in detail to offset the distortion.

Reconstructing the decorations, we kept in mind their real production phases and we obtained many textures for each phase, by following the published studies on them.

A big problem was mapping the texture on the polyset. On a NURBS model we can project the texture around the mesh and obtain a good mapping. When we change the model, we transform it into a polyset, losing every connection. For this reason we need to build a new map associating the right coordinate of the texture to the single vertices.[ALIAS3_98]

We removed the old texture coordinates, and by using a cylindrical projection we created the UV coordinate on the polyset too. Then we mapped the texture; to limit the break effect of the texture rolled up on a cylinder , we normalised the coordinates, with values from (0,0) to (1,1).

Rendering

Since we needed to work in real time, we chose the RTG format and the rendering of the scene depended just on the OpenGL platform and on the hardware we used.

Regarding the texture, we chose the ppm format to save them on, because it allows a good image definition with reasonably small files.

Hardware description

The workstation is a SiliconGraphics Indigo2 with IP28-195 MHz audio processor, MIPS R10000 CPU and MIPS R10010 floating point FPU; it has a parallel pipeline structure, with 32 Kbytes data cache, 32 Kbytes instruction cache, 1 Mbyte secondary unified instruction/data cache and a 64 Mbytes RAM. The graphic card is an High Impact with 1 Mbyte texture memory and 8 Mbytes DRAM and it has 1280 x 1024 as max resolution and a 32 bit double buffer with a 24 bit Z-buffer for a resolution of 1024 x 768.

The system interacts with the real ambient through two different inputs: the tracking sensor and the touch screen. The sensor (an electromagnetic Polhemus Fastrack sensor) provides information about the user's position in the real space and the perspective from which he/she is looking at the virtual object.

With a C++ driver we read the value from the sensor on the serial port of the workstation and we process the data with an appropriate software. With another driver we read the data from the touch screen.

The output is displayed on an LCD flat screen, with low energy cost (slightly less than 15 Watt) and without geometric distortion for a wide angle of vision.

Software Description

The application works on the UNIX platform, a SiliconGraphics IRIX 6.2 system; the whole programming has been performed on this platform. The application was realised by programming with C++ language, and by using many libraries, Xforms, OpenGL, GLUT and VRLib.

The main building block of the system is the graphical interface, which shows every window or button and regulates the interaction with the sensor and calculates the correct image for the animation. For this we used the Xforms library, programming in XWindow ambient.[ZHAO_97]

The management of every high level graphic function is controlled by the VRLib library developed in our laboratory (PERCRO Lab - Scuola S.Anna - Pisa - Italy) to achieve a very high performance in virtual reality applications. This library allows us to visualise in a simple manner a complex scene with many objects present, and to manipulate at the same time the models; it thus becomes possible to move the camera in the virtual ambient in real time.

We connect the VRLib to the graphical hardware by means of the OpenGL library, by exclusively managing the several objects, with the exception of their visualization. The images rendering management is in fact peculiar to each hardware platform, and for this reason each individual operator should use his particular system. In our case, this was the GLUT library, for the sole purpose of open an OpenGL window into the screen.[DAVIS_97]

The GLUT (OpenGL Utility Toolkit) is a low level GUI, purposely developed for the OpenGL.[KILGARD_97]

Our application had to be very easy to use and with a large diffusion potential. For this purpose we decided to use a library compatible with every platform, UNIX or otherwise.

The VRLib works on any platform, while it is possible to download from the Net of the free version of the suitable library for every platform.

Porting on PC platform

To reduce the cost of the system, we developed the final application to be totally compatible for a WindowNT platform. We installed the same libraries on the PC, an Xserver, a free Xwindow emulator for the Xforms, and we used a commercial 3D graphic card: the performances are good enough for a fluent animation on the screen.

At present the application works on both platforms, UNIX and WindowNT: all that remains is to define a variable compiling the code.

The application

The system needs a very user-friendly interface, so that anybody can easily use it in a museum exhibition. We decided to use a few, large buttons on each window, with very clear functions.

The application includes the possibility of changing the consultation language (English and Italian are currently available). The user can choose the language at the start of the exploration and he can change it at any time: changing the language means translating every information text and every label on the buttons.

Another important thing was to give a lot of historical and archaeological information about the artefact, as well as some photographs and maps; we placed the appropriate buttons to obtain this information.

It is possible to obtain information about the production phases also, with text and images: the images are obtained by rendering the models with the appropriate textures for each phase.

There is a main menu, which the user use between the consultations; the buttons are on the sides of the screen: on the left are the buttons to modify the system, as well as the language choice, the help, the quit and the bibliography information buttons; on the right are the interaction buttons for the virtual exploration or the production steps. In accordance with the archaeologists, we divided the production steps into five phases, as described in the published researchs on it.

The virtual exploration is possible through a full screen window and the user can choose to look at the model of the flûte or at the model of the planar slab. This second possibility is important because it allows to see another aspect of the artefact. To complete the visual information on the flûte, which is almost a cylinder, we decided to "unwrap" the decorations. For different objects, for example a sarcophagus, it is possible to use this exploration to have a closer look at the mummy or other artefacts found in the same tomb.

Software development

We developed the whole application with C++ language to have good real time performances.

The code consists of many files with different functions. The files for the description of the GUI (Graphic User Interface) are kept together and there is a static part of the code, which is loaded for the sole purpose of building the interface, at the start.

Some files contain the operative functions, for the movement (sensor), the graphic rendering (VRLib, OpenGL and GLUT) and the internal functions.

There are also data files with English and Italian texts, photographs and 3D models, from which some functions load the necessary data and many of these callback functions are linked to the buttons.

Three functions manipulate the sensor, by activating it, by reading its position and orientation, and by deactivating it before closing the application.

The loading phase of the 3D models is very slow, therefore we prefered to do it before everything else to have it ready when starting the application, allowing the user to immediately explore the virtual ambient.

There are many windows which the application uses just once, like the title, the presentation and the first language choice. We thus decided to clear this window from the memory after its use, in order to achieve a better performance.

Problems and solutions

The first problem we met was being to able to change the language at any time: we have defined a global variable, LANG, with value 0 or 1, readable from every functions in the code. When the user changes the language, he alters the variable LANG and starts a function that translates everything in the right language.

To open and manipulate an OpenGL window we used the GLUT, the VRLib and the Xforms together; at the beginning the system is controlled by an Xforms function (fl_do_forms), but during the virtual exploration the control passes to the GLUT with its control function.

A function, called idle function, which is present in both libraries, works autonomously every time the system isn't performing other operations. We used this function to read the sensor data and check the buttons; when it can, the idle function reads the sensor, checks the buttons with an Xforms function (fl_check_forms) and decides the right image to send to the screen.

A very delicate module is the control of the sensor, and to limit the problem we activate it on the first exploration of the user and we only deactivate it when the user closes the application.

During the idle function operation, the position and the orientation of the sensor are controlled, and we can assign the six global variables (XPOS, YPOS, ZPOS, YAW, PITCH, ROLL) for the visualisation module.

The electromagnetic sensor, like the Polhemus Fastrak, is often distorted and inaccurate; the errors were extremely sensitive to the distance between the source and the receiver (beyond 1.5 meters the scatter becomes macroscopic) and to the ambient electromagnetic environment.[BRYSON_92]

Therefore we had to control the working conditions: the short range of action around the virtual artefact and the use of a LCD monitor, with low emission, allow directly to use the value from the control driver. The control system of the Polhemus has an EPROM memory, with a low pass filter programmable for the position and orientation values; we used those filters with the default settings.

We added a linear filter to take the average of the last three values: the sensor framerate is 120 frame/sec.: thus our filter didn't present a delay of the image, nor any sliding effect.

During the virtual exploration, it is possible to retrieve particular information about the figures in the decoration of the flûte. By touching a button, the virtual exploration stops and a window opens with a photograph and an explanation text.

We defined a function that calculates the point of view of the user and chooses the right explanation; the function uses the data sensor, the position of the camera and the laying vector of the virtual camera in the virtual ambient; it calculates the intersection point from the camera and the bounding box of the model by choosing one of the four explanations of the figures.

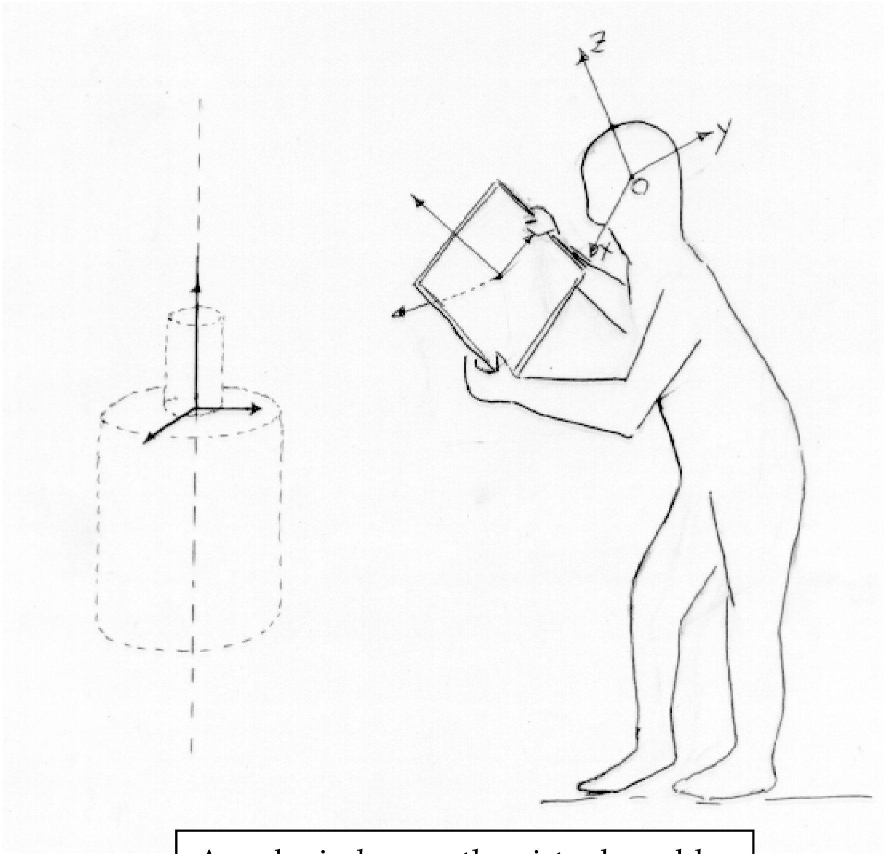
Conclusions and project development

In this paper we have presented a global system for a new visual interaction between the virtual artefact and the user.

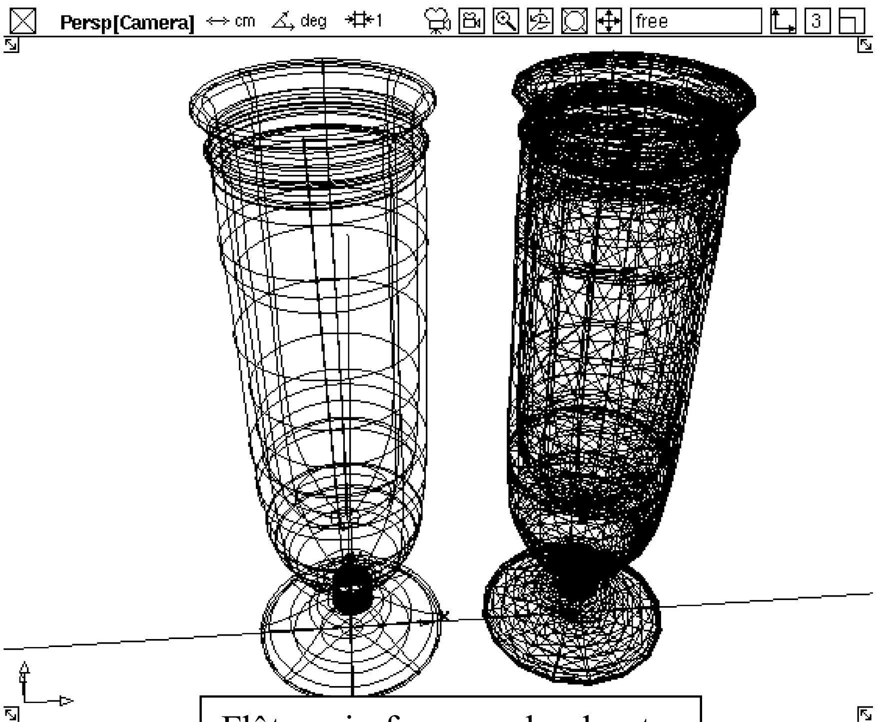
We have proposed a system to enjoy of works of art by using the new technology and giving visual and textual information about the artefact. The application results very easy to use: the user can walk in the real space and look at the virtual object, by keeping the screen only.

The future technical development of this system will aim at bringing the prices down, with a perfect porting on a PC platform, which is cheaper than a workstation, and at improving the conditions for a station in a real museum.

We are also planning to realise a multi-user system, in which many people will be able to move freely, unencumbered by cables, with a radio connection, and they will be able to use the application and the models together.



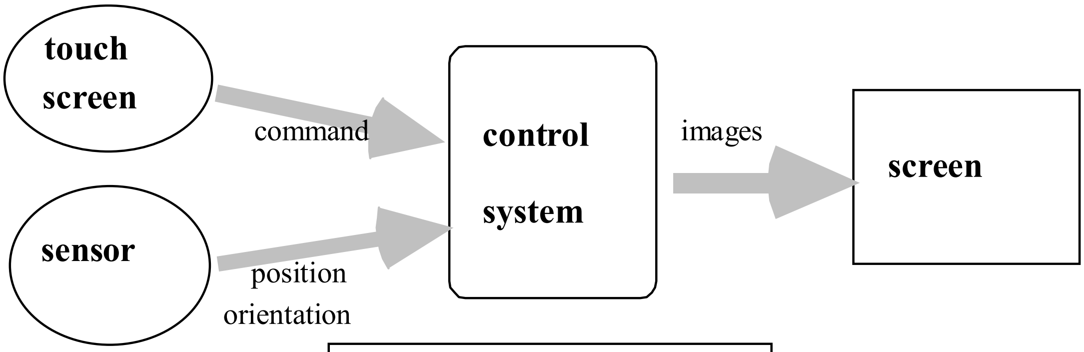
A real window on the virtual world



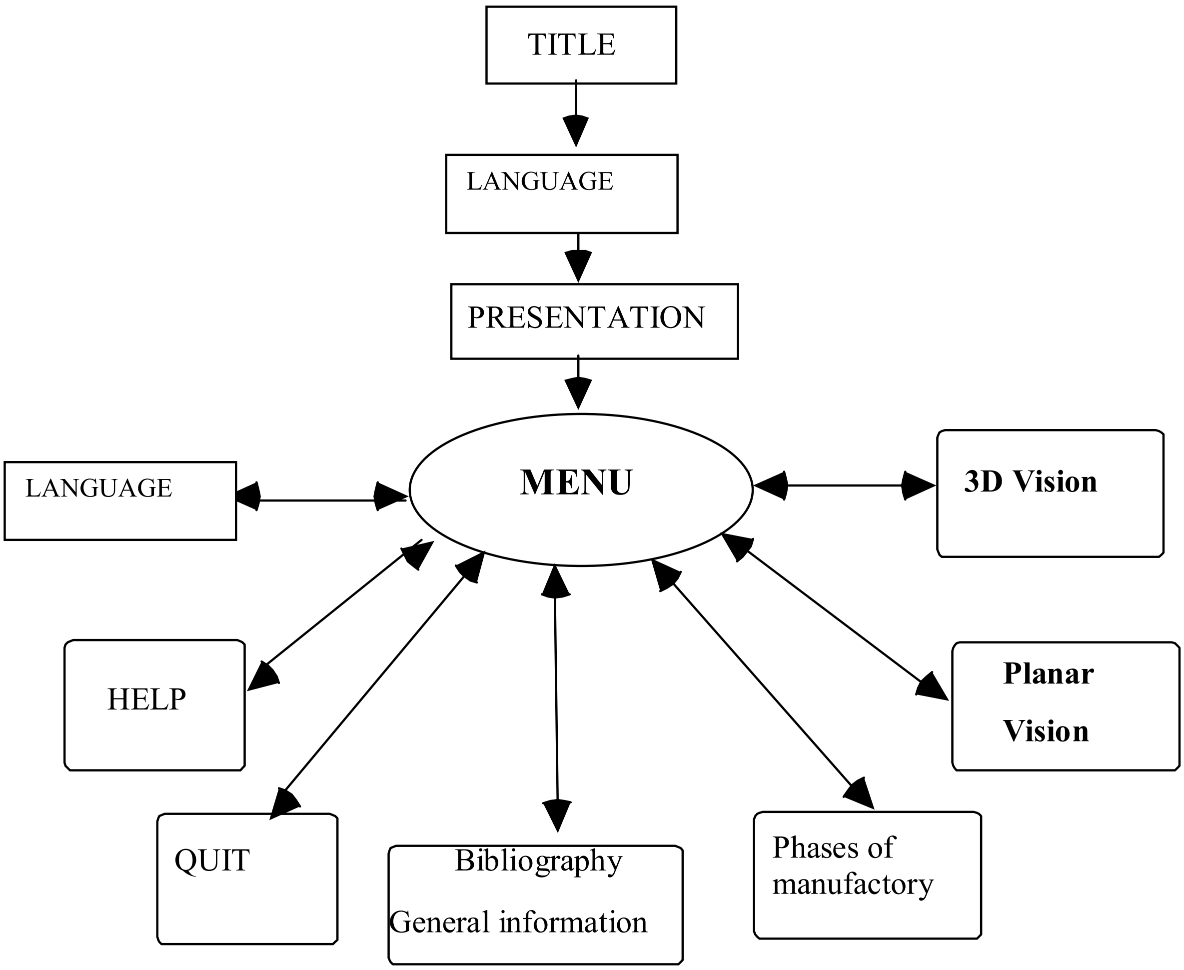
Flûte: wireframe and polyset

The textures of the decorations

Rendering of the flûte model



The interaction system



The application

References

[ALIAS[1]_98] AA.VV. - *NURBS modelling in Alias* - Alias|Wavefront Silicon Graphics 1998

[ALIAS[2]_98] AA.VV. - *Rendering in Alias* - Alias|Wavefront - Silicon Graphics - 1998

[ALIAS[3]_98] AA.VV. - *Polygonal modelling in Alias* - Alias|Wavefront Silicon Graphics – 1998

[ALIAS[4]_98] AA.VV. - *Games Data Transfer* - Alias|Wavefront - Silicon Graphics - 1998

[BRESCIANI_92] E. Bresciani, F. Silvano - *La collezione Schiff Giorgini* - Ed. Giardini - 1992

[BRILL_91] R.H. Brill - *Scientific investigations of some glasses from Sedeinga* JGS 33 - 1991

[BRYSON_92] S. Bryson - *Measurement and calibration of static distortion of position* data from 3D trackers - RNR Tecnical Report - 1992

[BURDEA_94] G. Burdea - P. Coiffet - *Virtual Reality Technology* Wiley - Interscience - 1994

[DAVIS_97] T. Davis - *The GLUT Library* - Innovation[3] - summer 1997

[DELORENZO_96] A. De Lorenzo - *Toy Story Dossier* - Computer Grafica - giu./sett. 1996

[FORSBERG_97] A.S. Forsberg, J.J. LaViola, L. Markosian, R.C. Zeleznik *Seamless interaction in virtual reality* Computer Graphics and Application – IEEE - nov./dec. 1997

[FORTE_96] M. Forte - A. Siliotti (a cura di) *Archeologia: percorsi virtuali nelle civiltà scomparse* - Mondadori - 1996

[KILGARD_97] M.J. Kilgard - *The OpenGL Utility Toolkit (GLUT) Programming Interface* API Version 3 - Silicon Graphics, Inc. - 1997

[KING_97] C.F. King - *Medical Volume Visualization* - Innovation[3] - summer 1997

[LECLANT_73] J. Leclant - *Glass from the Meroitic necropolis of Sedeinga* - JGS - 1973

[MATTHIAE_95] P. Matthiae - *Ebla - Un impero ritrovato* - Einaudi - 1995

[MAZZEO_97] F. Mazzeo - *Architettura al calcolatore* - Computer Grafica - ott./nov. 1997

[ROEHL_97] B. Roehl - *Virtual Archaeology* - Innovation[3] - summer 1997

[SHERIDAN_92] T.B. Sheridan - *Telerobotics, Automation and human supervisory control* The MIT Press - 1992

[SIMS_97] D. Sims - *Archaeological Models: pretty pictures or research tools ?* Computer Graphics and Application - IEEE - jan./feb. 1997

[WALTZ_97] M. Waltz - *New immersive display heat up high-end visualization* Innovation[3] - summer 1997

[ZHAO_97] T.C. Zhao - *Forms Library - A Graphical User Interface Toolkit for X* Department of Physics, University of Wisconsin- Milwaukee 1997.