Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Computers & Graphics 28 (2004) 477–484

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Magic wand and the Enigma of the Sphinx

Tolga Abacı, Rachel de Bondeli, Ján Cíger*, Mireille Clavien, Fatih Erol, Mario Gutiérrez, Stéphanie Noverraz, Olivier Renault, Frédéric Vexo, Daniel Thalmann

VRlab, Swiss Federal Institute of Technology (EPFL), IN-J Ecublens, 1015 Lausanne, Switzerland

Abstract

This paper presents an evaluation of the benefits and user acceptance of a multimodal interface in which the user interacts with a game-like interactive virtual reality application "The Enigma of the Sphinx". The interface consists of a large projection screen as the main display, a "magic wand", a stereo sound system and the user's voice for "casting spells". We present our conclusions concerning "friendliness" and sense of presence, based on observations of more than 150 users in a public event.

Keywords: Magic wand; Virtual reality; Multimodal interaction; Non-obstructive interface

1. Introduction

Virtual reality is a field which is traditionally associated with head-mounted displays, data gloves, motion trackers, plenty of wires everywhere and a usually steep learning curve for the users. Our work presents a different possibility—a very minimalistic but "user-friendly" approach, accessible even to the non-trained general public.

This work is an experiment for testing a multimodal and non-obstructive interface for virtual reality applications. We want the user to be immersed and provide a natural interface to interact with the game-like application, without resorting to complex and obstructive hardware, such as HMD or data gloves. Our emphasis is not on advanced visual effects or extended "gameplay". What we propose is a different approach to immerse the user into a virtual environment and let him/her interact with it.

The application was implemented using a generic in-house development framework for interactive VR applications. This framework incorporates generic functionality to bring together several components and devices required for implementing multimodal interfaces. The technology being tested here can be used not only to create more entertaining games, but also to implement serious applications for training, visualization and manipulation of complex data.

2. Background

Multimodal interfaces are an approach trying to merge several input (and output) modalities, such as speech, gestures, pen input, sound, video, haptics or various other devices. They enable the user to interact with the virtual environment in a similar way to how he communicates in everyday life—for example "Move **that** box to the door!", where the box is selected by hand gesture or pointing.

*Corresponding author. Tel.: +41-21-693-5248; fax: +41-21-693-5328.

E-mail addresses: tolga.abaci@epfl.ch (T. Abacı), rachel.cetre@epfl.ch (R. de Bondeli), jan.ciger@epfl.ch (J. Cíger), mireille.clavien@epfl.ch (M. Clavien), fatih.erol@epfl.ch (F. Erol), mario.gutierrez@epfl.ch (M. Gutiérrez), stephanie.noverraz@epfl.ch (S. Noverraz), olivier.renault@epfl.ch (O. Renault), frederic.vexo@epfl.ch (F. Vexo), daniel.thalmann@epfl.ch (D. Thalmann).

0097-8493/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cag.2004.04.003

Probably the first multimodal application was MIT's famous "Media room", described in the work of Bolt [1] from 1980. It implemented the "put that there" interface by tracking the directions of the user's hands and using a hardware-based speech recognition system.

The work of Nijholt and Hulstijn [2] describes a multimodal interface to a virtual character (speech and keyboard input). Krum and Omoteso [3] make a comparison between the multimodal (gestures performed by the "gesture pendant" combined with speech) and classical (keystrokes) interfaces used in a GIS environment. They conclude that actually many users found the multimodal interface much easier to use than the keystrokes.

Quickset, described in [4], is a 2D map application with a pen and speech interface. The user can create and manipulate virtual objects on the map for a variety of applications: military simulation and training, 3D terrain visualization, disaster management, etc.

The multimodal scientific visualization tool [5] is a visualization environment for exploring scientific data such as fluid flow simulations. The interface is composed of a pair of data-gloves (using magnetic trackers) and voice recognition (approx. 20 commands). The system provides a variety of navigation, manipulation and picking techniques. Our work uses a less obstructive device for posture recognition and we do not require very complex interaction techniques.

In [6], authors describe a multimodal testbed composed of a virtual environment called MDScope and a graphical front-end (VMD). The system is designed to simulate the interaction of biomolecular structures. The interface consists of voice (spoken commands) and gesture recognition (3D finger pointing and simple hand gestures are extracted with two fixed cameras).

BattleView [7] is a virtual battlefield application for supporting planning and decision making developed by NCSA. In this system, 3D pointing and simple hand gestures recognition—using a fixed single camera—are used in combination with speech recognition, using IBM ViaVoice. A multimodal integration module combines the recognizer streams. As will be explained later, in our system we use a similar approach to integrate the data coming from the multimodal interface components (device aggregator).

The "magic wand" multimodal interface we are using was described recently in [8]. It replaces the more traditional 3D mouse and buttons with a magnetically tracked wand and speech recognition, which are used in a mutually complementary way.

3. Enigma of the Sphinx

Our system was developed as a demonstration of the research done in our laboratory for the general public attending the events held at the 150th anniversary of the Swiss Federal Institute of Technology in Lausanne (EPFL), which took place from 2 to 4 May 2003. During these 3 days, approximately 150 users tested the system and many more visitors saw the demo.

The plot of the game is very simple. It is set in ancient Egypt, where the Sphinx has got a problem—its nose disappeared. It is up to the user to solve this puzzle and recover the missing nose from a maze hidden inside the large pyramid.

From the user's point of view, there are two main parts in the application:

- *The flying part*: the user is asked to find the Sphinx, fly to it using a virtual flying carpet and listen to the introduction of the story. After his visit to the Sphinx, the user is supposed to find the entrance of the pyramid, land there and enter the labyrinth. Figs. 1 and 2 show this part of the application.
- *The maze part*: inside the pyramid, consisting of four "mini-games" hidden in separate rooms, which have to be completed in order to win the game. The user has to "walk" through the maze to find three virtual characters, complete the tasks they ask him to do in order to get three objects which are keys to open the door to a room with the missing nose.

The goal of the game is not to challenge users with difficult riddles, but to encourage them to explore the virtual environment. Throughout the game, 2D graphical cues are visible on screen—a map of the labyrinth with the user's current position, the available keywords, the objects already collected, etc. In addition, the user has five "lives" (the possibility to fail five times).

The user controls the application with a simple multimodal interface consisting of the "magic wand" and speech recognition system described in detail in [8].

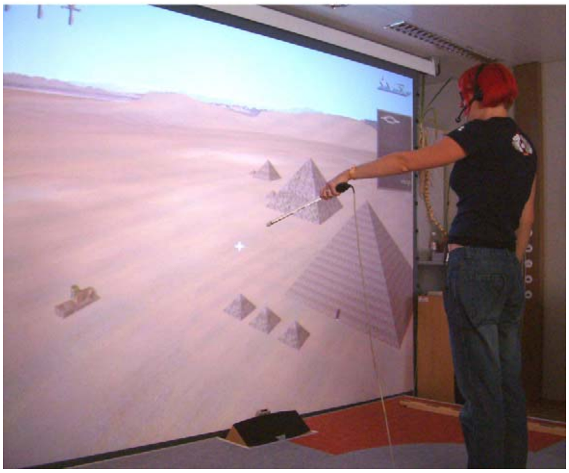


Fig. 1. Flying.

Fig. 2. Crying Sphinx.

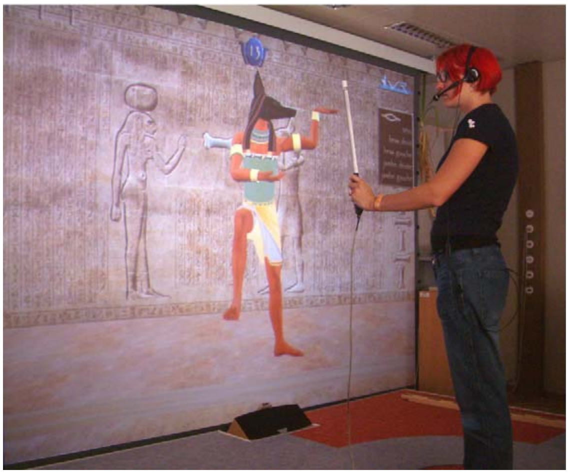
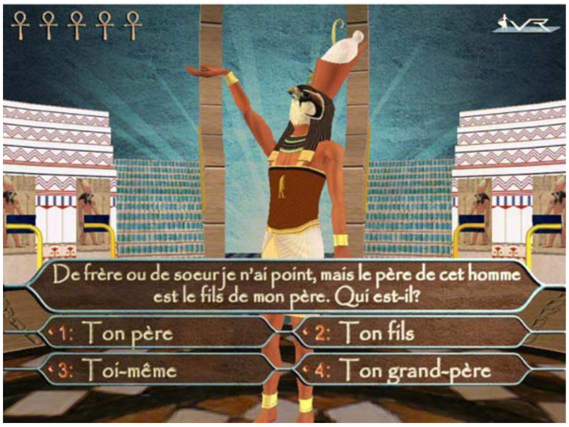


Fig. 3. Anubis.



De frère ou de soeur je n'ai point, mais le père de cet homme est le fils de mon père. Qui est-il?

1: Ton père

2: Ton fils

3: Toi-même

4: Ton grand-père

Fig. 4. Horus.

The interaction varies depending on the part of the game the user is solving.

- While flying, the user only points the wand in the direction he wants to fly and uses the voice keyword "fly" to activate flying towards the target, and "land" to land either in front of the Sphinx or in front of the pyramid entrance.
- Inside the pyramid, the user in general uses the "magic wand" as a joystick and navigates around the labyrinth by moving the wand left, right, forward and up (to stop). The "mini-games" have their special interaction paradigms.

The "mini-games" consist of three interactions with the virtual characters—Anubis, Horus and Sobek, the Egyptian gods. Each of them presents a different challenge and uses a different mean of interaction.

- Anubis (Fig. 3) needs to have his posture changed into the one engraved in the wall behind him. The user achieves this by selecting the parts of the body by voice (for example "left arm" or "head") and moving the "magic wand". This moves the selected body part, until the user finds the proper position and the part is locked in place. The game continues until the user either moves Anubis into the proper posture for winning the game or until time runs out.
- Horus (Fig. 4) presents the user with a riddle. The user must choose from four possible answers, only one of which is correct. The selection is made purely by voice, by saying the number corresponding to the answer. The "magic wand" is not used at all.
- Sobek (Fig. 5) is Cleopatra's aerobics trainer. He asks the user to follow a simple "aerobic" routine with the magic wand. The user has to reproduce the precise movements of the wand at the right time. Speech recognition is not used in this game.
- The final part of the game is solving the riddle on the door. The three key objects obtained after winning the three "mini-games" with the three virtual characters have to be placed into the correct slots on the door in order to unlock it.

  The user has to move the "magic wand" into the position where he wants to put the objects and say its name aloud. If the position (left, center, right) for the object is correct the object floats into place, otherwise nothing happens. After placing all three objects correctly, the door opens and the audience sees the happy Sphinx dancing with the nose back in place (Fig. 6).

4. Game implementation

4.1. Smart proxy concept

Multimodal interfaces usually employ several user interaction devices. The capabilities of the hardware can

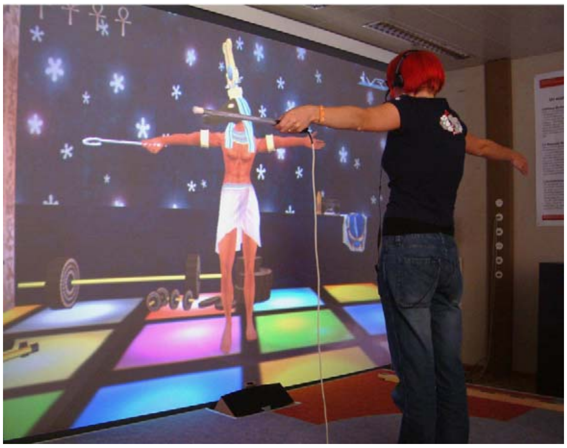


Fig. 5. Aerobic with Sobek.

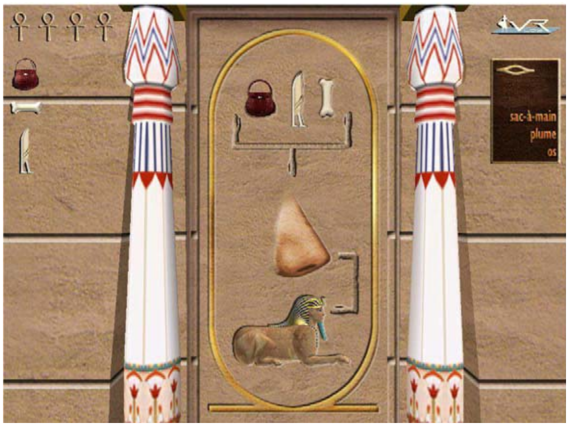


Fig. 6. Riddle on the door.

be quite different (for example, the interface of the "Enigma of the Sphinx" combines speech-recognition and motion tracking technologies). Therefore, it is beneficial to have a generic architecture where various hardware can be used in a complementary fashion.

Input devices may operate through various hardware channels such as joystick ports, serial interfaces, USB, etc. Speech recognition is more complex. It is actually implemented in software that can run on a different computer. It is not exactly a hardware device, although we treat it as such for the purpose of this project.

Our implementation of the generic architecture consists of a common network protocol and "smart proxies" that communicate using it. "Smart proxies" are a simple solution to unify different kinds of equipment. They are small programs that run on the computers to which the hardware is attached. Their main role is to communicate locally with the hardware and convert the data to the common network protocol. This "hides" the differences in the hardware capabilities and allows the development of device-independent applications, with all the hardware-related complexity concealed inside the "smart proxies".

4.2. Architecture

The virtual environment and the application "Enigma of the Sphinx" are implemented using an integrated framework VHD++, developed in the collaboration of VRlab, EPFL and MIRALab, University of Geneva. It was described in [9].

Fig. 7 is an overview of the system architecture. There are five main parts:

(1) the speech recognition, based on the Sphinx II engine from Carnegie Mellon University;
(2) the "magic wand", which tracks the position and orientation of the wand and recognizes postures;
(3) the device aggregator, which combines the data from both the speech recognition and the "magic wand" proxies;
(4) the game logic, which contains a finite state machine controlling the game;
(5) the virtual environment, which contains the graphics module, sound module and other supporting modules (data loaders, animation engines, etc.).

The game logic contains the finite state machine controlling the application. It is shown in the schematic diagram in Fig. 8.

The flow of control is as follows:

Edge 1: Application starts in flying mode: desert landscape, user drives "magic carpet" by pointing with "magic wand" and using voice commands, camera management also controlled by magic wand.

Edge 2: flying mode continues while the user is far from the Sphinx or the large pyramid.

Edge 3: If flying near the Sphinx for the first time, display animation of crying Sphinx (Fig. 2) to explain goal of the game.

Edge 3a: When Sphinx animation finishes, return to flying mode.

Edge 4: If flying near the large pyramid after having visited the Sphinx, enter the maze. The game switches to labyrinth mode.

Edge 4a: While the user is far from the gods rooms, navigation through the labyrinth continues without change.

Edge 5: If the user is near Anubis's room and this game has not been won yet, system enters Anubis game.

Edge 5a: If the user wins this game, or refuses to play again after failing, and still has at least one life left, game re-enters Labyrinth Mode.

Edge 5b: If the user loses this game and has no lives left, the game ends, displaying the Game Over sequence.

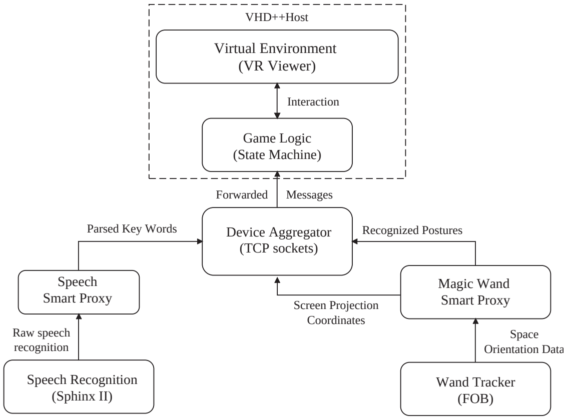


Fig. 7. System architecture.

Fig. 9. Dancing Sphinx.

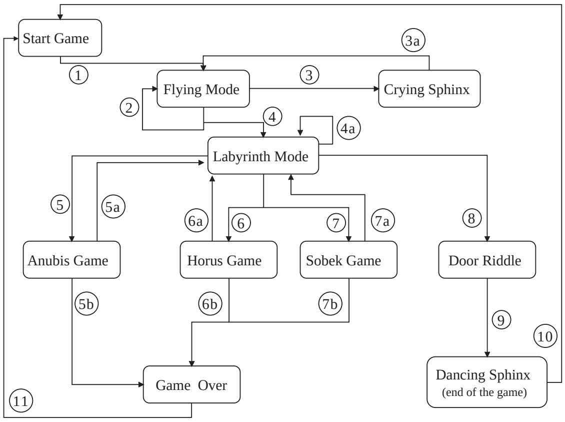


Fig. 8. Game logic states.

Edges 6, 6a, 6b, 7, 7a and 7b are similar to state 5 triplet and correspond to other two "mini-games" (Horus and Sobek).

Edge 8: If the user is near the door to the room hiding the nose and has already won the three "mini-games" to obtain the required objects, system switches to Door Riddle (final "mini-game").

Edge 9: Once the user has placed three objects in their corresponding slots, game ends, by showing Dancing Sphinx animation (Fig. 9).

Edge 10: Dancing Sphinx animation can be stopped at any time to restart game.

Edge 11: After Game Over screen, application can be restarted from state 1.

The game logic uses three sub-modules (services in VHD++ terminology)—the flying service, labyrinth service and camera control service. Each of them has an important role in the application.

The flying service simulates the flying carpet. Internally, it uses a simple physical model, allowing for realistic acceleration and braking and proper collision handling.

The labyrinth service handles navigation inside the labyrinth, as well as interaction with the three virtual characters to win the objects needed for the final riddle.

Anubis is controlled by the "magic wand", each recognized posture from the wand corresponds to one pre-recorded animation of the selected limb. The correct posture of the limb triggers a sound effect as a cue for the user.

The riddle with Horus is very simple, only voice input is used. To avoid problems with wrong answers being triggered by noise, the user is prompted to confirm his answer by saying "yes" or "no".

"Aerobics" with Sobek mainly uses the "magic wand". No speech input is necessary. The virtual character moves on the screen and the user has to reproduce the same motion in predefined time intervals (see Fig. 5). If the posture of the "magic wand" is not correct at the end of each interval (does not match that of Sobek's wand), the game is lost. The animations are again pre-recorded.

The last puzzle on the door is activated only after all three objects have been collected. Approaching the door activates the last "mini-game". Only "left", "forward" and "right" postures are used, together with the names of the objects. The game logic checks which object name was spoken and whether the posture of the wand is correct for that object (the postures are pre-defined to match the positions of the carvings on the door—Fig. 6).

The camera control service is used to "drive" the camera in first-person view, which is used in the application. It has several modes:

- In the flying mode, the camera position is fixed to the position of the "flying carpet", but the user is free to rotate the camera using the "magic wand". This technique is described in detail in [8].

- The labyrinth mode uses a human walking model to produce a realistic-looking "walking" of the user inside the labyrinth. The camera is attached to the "head" of an invisible avatar which "walks" in the labyrinth.
- There are several "cut-scenes" during the game, used to explain the plot and to get the camera in the required position for the interactions with the virtual characters, in order to properly see them. These are implemented by animating the motion of the camera—e.g. the "landing" of the "flying carpet" or the activation of the "mini-games".

The game logic also makes use of an overlay screen with orthographic projection, which displays the 2D graphical cues.

5. Results

During 3 days of demonstration, we had the opportunity to observe users' reactions to the multimodal interface we have described. People of all ages—ranging from 6 to 50 years old—played the game.

5.1. Design and believability

Sound plays a key role in the sense of immersion. Thanks to a well-designed sound, in particular good quality voices (performed by real actors), we could achieve a very high degree of believability (rather than realism, which we consider less important).

The design of the environment (landscape and labyrinth) contributed greatly to the sense of presence [10,11]. The public was especially impressed by the labyrinth's design. Its beauty encouraged the user to explore and lessened his fear of getting lost (the map helped as well).

Concerning the design of the virtual humans, their god-like looks (human bodies with animal heads) allowed us to avoid strict realism (getting towards symbolic actors) and to conceal some problems. For example, the lack of facial animation and expressions, which could have shocked the users, was dissimulated by the use of animal features. As for body animation, unrealistic movements are usually more visible when they are played on human-like shapes than on non-human shapes. Thanks to the symbolic features of our virtual humans, few people noticed these problems (feet sliding, bad transitions, etc.).

5.2. Display and immersion

The 2D graphical interface was intended to help the user, and testing it with the public gave us many hints for further improvement: since we developed the game on standard computer screens, we considered the whole surface of the screen for the user interface, therefore scattering 2D graphical elements such as the map, the keywords, etc. along the borders. Then, when playing on the larger projection screen, the user had to either step back or turn their head to be able to read the information, therefore disrupting the sense of presence.

The height and posture of the user and audience seemed to be relevant as well, as they could change the degree of immersion. We should explore the possibility of adapting the virtual camera position and angle, depending on height.

Indeed, the immersion of the audience seemed good, sometimes even better than that of the users themselves, and they participated actively in the various stages of the game. One explanation for this could be that they were sitting, and therefore their eye-level was better centered on the screen. The fact that they did not have to concentrate on the map of the labyrinth and were able to enjoy the scenery instead helped as well.

We also tested the use of stereographic display (shutter-glasses), and noticed that in general, the effect was very impressive when the virtual objects were close (inside the labyrinth). However, in open spaces (landscapes), the results were disappointing and even disturbing (mostly because of the flicker of the shutter-glasses). Due to these unsatisfactory results observed during the development phase, we gave up on using this feature for the public presentation.

5.3. Interaction and intuitiveness

Concerning interaction, the use of the "magic wand" and keywords seemed rather intuitive. Playing with them was very natural for most of the users, in particular for children, who generally understood right away how to use the flying carpet and easily found their way inside the labyrinth, whereas adults had more difficulties.

Our guess is that children are more used than adults to playing in immersive environments and more at ease when using new devices, because on the one hand, they were "born with computer technology", and on the other hand, they are used to learning new things every day. Besides, playing video games might help developing some skills such as spatial orientation (required to interpret a 2D map while navigating in a 3D world).

As a whole, we observed that children were also more patient, whereas adults expected an immediate response from the system (both from the wand and voice recognition) and often complained about the delay. In general, adults had the tendency not to listen to the rules and explanations we gave them and seemed more affected and stressed by the presence of an audience (they were also much less enthusiastic when we asked for volunteers). We thought that using a “magic wand” would be intuitive for everyone. However, some adults (whom we suspect had already had difficulties learning the use of today’s standard devices such as mouse and keyboard) felt apprehensive in front of yet another new device, feared the training would be long.

We also noticed the following behaviors/expectations:

- One of the most commonly observed “problems” was the absence of a backwards-pointing posture. We thought that pointing the wand backwards would be unintuitive and we did not implement recognition for such a posture. However, we found that many users were trying to go back by pointing backwards, instead of stopping and turning either left or right as we expected them to do.
- Many users expected the interface to react proportionally to the velocity of their gestures. Also, the “magic wand” was not able to react as fast as some people expected, in particular when a repeated movement to the left or right was required. We implemented an “auto-repeat” feature in the wand but apparently, it was not fast enough for some users. Some of them tried to move the wand violently, hoping this would make the system react faster.
- One problem concerning naturalness was the fact that the most comfortable or natural posture for some users was to keep the “magic wand” in a vertical or upwards-tilted position—in our system, holding the wand in a vertical position is interpreted as a “stop” command (to keep moving, the wand must be kept horizontal). It seemed that for some people, pointing forward to keep walking was not so natural.

5.4. Diversity of interaction paradigms

The pointing mode seems to be more intuitive for navigation. Inside the labyrinth (walking), some expected the same navigation mode as in the beginning of the game (flying). During some parts of the game as well, users tended to point when a direction was needed (for example, in the riddle “mini-game”, they pointed at the answers, or in the final game, for placing objects in the door slots).

While “walking” inside the labyrinth, users tended to use the voice as well—in addition to the wand direction they often used keywords like “stop” or “turn”. Further possibilities of mixed interactions should be implemented for a more effective multimodal interface.

As a whole, switching between the navigation modes (pointing and postures) was rather confusing for the users. The interface was not consistent enough, because the same input mode did not behave the same way all the time.

6. Conclusions and future work

Adding spatial sound could enforce the user’s sense of presence inside the building. Graphical improvements, such as shadows and bump-mapping, would also increase the presence in the virtual environment (tend to photorealism, or “perceptual realism” [11]). Solving the feet sliding problem in motion capture, as well as adding projected shadows, facial animation and lip-synch would further enhance the general believability.

We should explore the possibility of adapting the virtual camera position and angle, depending on the height of the user (without having to use an HMD). For future immersive applications, more attention should also be paid to the location of the 2D graphical elements on the screen (place them closer to the centre, within a certain focus angle). Peripheral vision is a key issue, which should be studied more closely and treated differently. It could also be interesting to test stereographic display using other devices than Shutter-Glasses, for example simpler devices such as polarized or colored (green/red) glasses.

In general, the “magic wand” could be improved by adding “backwards” to the postures repertoire, by making it react to the velocity of the motion—recognizing gestures rather than postures, and finally by implementing some kind of memory, so that a given order could be stored until a new decision is required (e.g. walking forward until either a wall is encountered or a different order given—“backwards”, “left”, “right”, “stop”, etc.). This could also be solved by increasing the threshold of the forward posture and reducing that of the neutral (vertical) posture. This way the order to stop would only be given when a very well-defined vertical position is assumed.

Finally, what could make the interface more intuitive for a large number of people is its flexibility, for example by customizing the voice keywords or redefining the meaning of the “magic wand” actions. There could be more ways to give the same order, to satisfy a larger range of user preferences and skills. However, for one given user, the interface should stay consistent throughout the application.

The demo was well received by the public attending the 150th anniversary of EPFL, and was featured in the main regional newspaper “24Heures”. The “Enigma of the Sphinx” demonstrates that multimodal interfaces do not need to be complex and obstructive to achieve “friendliness” and good sense of presence in virtual environments.

Acknowledgements

Authors want to thank their colleagues Nicolas Elsig for the design of some of the virtual characters and Bruno Herbelin for proofreading the text.

References

[1] Bolt RA. Voice and gesture at the graphic interface. ACM Computer Graphics 1980;14(3):262–70.

[2] Nijholt A, Hulstijn J. Multimodal interactions with agents in virtual worlds. In: Kasabov N, editor. Future directions for intelligent information systems and information science, studies in fuzziness and soft computing. Wurzburg: Physica-Verlag; 2000. p. p148–73.

[3] Krum D, Omoteso O, Ribarsky W, Starner T, Hodges LF. Speech and gesture multimodal control of a whole earth 3d virtual environment, 2002. URL citeseer.nj.nec.com/article/krum02speech.html.

[4] Cohen P, Johnston M, McGee D, Oviatt S, Pittman J, Smith I, Chen L, Clow J. Quickset: multimodal interaction for distributed applications. In: Fifth ACM International Conference on Multimedia; 1997. p. 31–40.

[5] Laviola J. MSVT: a virtual reality-based multimodal scientific visualization tool. In: IASTED International Conference on Computer Graphics and Imaging; 1999. p. 221–5.

[6] Sharma R, Huang T, Pavlovic V, Zhao Y, Lo Z, Chu S, Schulten K, Dalke A, Phillips J, Zeller M, Humphrey W. Speech/gesture interface to a visual computing environment for molecular biologists. In: International Conference on Pattern Recognition (ICPR); 1996. p. 964–8.

[7] Pavlovic V, Berry G, Huang T. A multimodal human–computer interface for the control of a virtual environment. In: American Association for Artificial Intelligence, Spring Symposium on Intelligent Environments; 1998.

[8] Cíger J, Gutiérrez M, Vexo F, Thalmann D. The magic wand. In: Proceedings of Spring Conference on Computer Graphics 2003, Budmerice, Slovak Republic; 2003. p. 132–8.

[9] Ponder M, Molet T, Papagiannakis G, Magnenat-Thalmann N, Thalmann D. VHD++ development framework: towards extendible, component based VR/AR simulation engine featuring advanced virtual character technologies. IEEE Computer Society Press, Proceedings of Computer Graphics International (CGI), 2003, pp. 96–104.

[10] Usoh M, Catena E, Arman S, Slater M. Using presence questionnaires in reality. PRESENCE: Teleoperators and Virtual Environments 2000;9:497–503.

[11] Lombard M, Ditton T. At the heart of it all: the concept of presence. Journal of Computer-Mediated Communication 3(2). Available from: http://www.ascusc.org/jcmc/vol3/issue2/lombard.html.