



A Low-Cost Test Environment for Usability Studies of Head-Mounted Virtual Reality Systems

Ahmed Seffah

Dept. of Computer Science
Concordia University
1455 Maisonneuve West,
Montreal
Qc, Canada H3G 1M8
seffah@cse.concordia.ca

Jonathan Benn

Dept of Computer Science
Concordia University
1455 Maisonneuve West,
Montreal
Qc, Canada H3G 1M8
Jonathan.benn@gmail.com

Halima Habieb Mammam

Dept. of Computer Science
Concordia University
1455 Maisonneuve West,
Montreal
Qc, Canada H3G 1M8
mammam@encs.concordia.ca

Abstract

There is a need to develop new usability testing environments and methodologies for unconventional interactive systems. Pursuant to that need, we developed a low-cost test environment for a Head-Mounted Display (HMD)-based, virtual reality system called Osmose. Osmose was difficult to test for many reasons, one of which was its style of interaction. We began setting up the testing environment about two weeks before the start of the usability testing. We learned many lessons throughout the experience. This paper summarizes the study findings, both methodological – how to setup and conduct a usability lab for such an environment – as well as conceptual –the human experiences and behavioral patterns involved in using an immersive environment.

Keywords

Immersive systems, head-mounted display, usability testing, sensing-based interaction, breathing interaction, virtual reality

Introduction

In the field of usability testing, it is known today that the current stationary and mobile usability testing labs, although powerful, suffer from a critical problem (Dumas and Redish, 1999). There is an implicit assumption that the participant (i.e., the representative end user) will be sitting at a desktop computer with a mouse and keyboard for input and a monitor for output. However, the range of computerized systems is far greater than just desktop applications. In particular, unconventional input/output systems (such as augmented reality and virtual reality) and mobile applications (such as pagers, cellular phones, and personal digital assistants) are difficult or impossible to test using such labs.

For Virtual Reality (VR), many difficult design questions arise (Sherman and Craig, 2003):

- How do we record what the participant is seeing when the computer's monitor is really a helmet?
- How do we record the participant's interaction with the system when the participant is able to freely move about the entire world?

Furthermore, usability testing in a virtual environment typically involves dealing with a very large range of measures:

- objective measures (task demands, task results and correlated measures, error numbers, achieved task levels, etc.)
- subjective measures (on-line evaluations, post test evaluation questionnaires, explanation of high stress events)
- psycho-physical measures (detection of the stimulus, recognition of the stimulus, stimuli discrimination, magnitude)
- physiological measures (heart rate, blood pressure, respiration rate, ECG)
- task performance and learning efficiency

Since usability testing techniques originally were developed to assess desktop applications, there is a definite need today to question their applicability for immersive environments. This is due to the fact that immersive environments and virtual world are highly different from GUIs (Sadowski, 2002). A little work has been done on this issue. New methodologies can be adapted from the old (as is the case for stationary virtual reality systems), or entirely new methodologies may need to be developed, as is probably the case for mobile applications (Poupyrev, et al., 1997).

The purpose of this research is to illustrate how to take a Virtual Environment (VE) and create a test environment/system that will enable researchers to record and analyze the entire user experience and interaction, which allows that VE to be used for usability evaluations.

In this report, we describe how we developed a new testing environment and methodology for performing usability tests on Osmose (Davis and Harrison, 1996), a Head-Mounted Display (HMD)-based, stationary, virtual reality system. Our goal was to create a testing environment that was low-cost, using only off-the-shelf components in new ways. We describe the new technique at a level of detail sufficient to allow others to repeat what we have done. While this paper only describes our experience, we hope that it is useful to people performing usability tests on other HMD-based virtual reality systems. In particular, the low cost of our solution (perhaps \$3,000 CAD, not including the computers) makes it broadly applicable.

Osmose: What make it Different and Difficult to Test?

Osmose is an immersive, interactive, virtual-reality environment installation with 3D computer graphics and interactive 3D sound, a head-mounted display and real-time motion tracking based on breathing and body movements. The first virtual space encountered is a three-dimensional Cartesian Grid, which functions as an orientation space. After a short time, the grid changes to a clearing in a forest. A dozen world-spaces exist in Osmose, most based on metaphorical aspects of nature, including the Clearing, Forest, Tree, Leaf, Cloud, Pond, Subterranean Earth, and Abyss. The substratum, Code, contains much of the actual software used to create the work. A superstratum, Text, is a space consisting of quotes from the artist

and excerpts of relevant texts on technology, the body, and nature. Code and Text function as conceptual parentheses around the worlds within.

Osmose was created by Char Davies, a pioneer of VR and one of Canada's most awarded digital artists. Osmose has been widely exhibited across the world, in places such as the Museum of Modern Art (San Francisco), the Australian Center for the Moving Image (Melbourne), the Barbican Centre (London), the Museum of Monterrey (Mexico), the National Gallery of Canada (Ottawa), and the Museum of Contemporary Art (Montreal).

Since Osmose was designed to be available for public display, it has a built-in mechanism to ensure that each participant's session lasts about 15 minutes.

Turning on Osmose is a step-by-step process that takes approximately 30 minutes. Turning it off is simpler and only takes about 5 minutes.

Osmose was assembled from commercially available components, including the following:

- A server (Figure 1), made up of two storage racks containing four personal computers (PCs), two with the Windows operating system, one with Macintosh, and one with Linux. The racks also contain sophisticated multimedia processing hardware that provides a variety of audio-visual outputs in various formats.



Figure 1: Test Back-End (left) and Osmose Server (right)

- A helmet (Figure 2), also known as an HMD, containing two low-resolution stereoscopic monitors offering a wide 180-degree field of view, speakers, and a magnetic sensor that detects the helmet's position and orientation. Figure 3 shows an example of what a participant might see while wearing the helmet.



Figure 1: Osmose Helmet

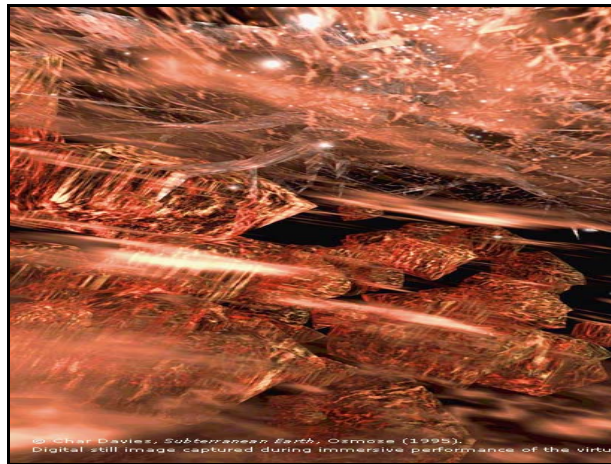


Figure 3: Subterranean Earth Environment from Osmose

- A vest (Figure 4 and Figure 5), which measures participant's breathing (by measuring chest expansion via a pressure gauge) and the orientation of their upper body.



Figure 4: Osmose Helmet and Vest



Figure 5: Osmose Magnetic Sensor Controller

Most of the components of the vest and helmet are connected to the server by wire. The exception is the helmet's magnetic sensor, which is wireless, but short-ranged. Its signal is captured by the magnetic sensor controller that is placed above the participant's head (see Figure 4). Hence, the participant does not need to be at the same location as the server. However, the participant cannot stray very far from the magnetic sensor controller.

Osmose is a public exhibit that has had to serve dozens of different people each day. This has given it particularly stringent usability requirements and has made it an excellent candidate for usability testing.

Typical Usability Testing

During a typical usability test, testers ask participants to perform a series of tasks and then record their experiences (Dumas and Redish, 1999). Testers record three types of data:

- the participant's interaction with the system
- what the participant is seeing
- the participant's speech and facial expressions.

While testing these is normally a challenge for desktop applications, Osmose is more difficult to test for the following reasons:

- The style of interaction, instead of being through a mouse and keyboard, is through unusual mechanisms: breathing and movement of the head and body. Therefore, it is more difficult to record the participant's interaction with the system since we cannot simply record mouse clicks and keystrokes.
- Participants are relatively mobile. Instead of being confined to a chair in front of a computer monitor, they have a certain degree of freedom (they may walk around within limits, and look in many directions). This makes participants harder to record.
- Recording what participants see is a challenge, since they are seeing two images originating from inside a helmet, rather than a single image coming from a standard computer monitor.
- Obtaining all of the required information will result in multiple data streams (e.g., participants, what they are saying, and what they are seeing). The design challenge is how to inexpensively assemble these data streams so that they can be reviewed in a simultaneous and synchronized way.

Methods

To meet the need for analyzing multiple data streams, we developed a new test environment for performing empirical usability tests on this unconventional interactive system. While the new usability test environment might not be generally applicable to all VR systems, it could, nonetheless, serve as an example of how to bridge the gap between the old and new interactive technologies.

Specific Goals and Testing Setup

In creating the new test environment and methodology, we had the following major goals:

- Acquire first-hand user experience on an unconventional, non-interactive system that is not based on the WIMP (Window, Icon, Mouse, Pointer) metaphor.
- Design and validate a testing environment that was low-cost, using only off-the-shelf components in new ways.
- Master the new technique, and the process used to conduct the study, at a level of detail sufficient to allow others to repeat our experiment.
- Develop a specific infrastructure and practical guidelines for usability testing of unconventional computerized systems.

Our test plan was to take two weeks in late January 2005 to set up the test environment and methodology for a usability test of Osmose. We planned to use the infrastructure to perform the usability test between February and April 2005.

Participant Interaction with Osmoses:

- To look in any direction, the participants simply turn their heads. They can look down, through their own body, up, and all around.
- To move forward, backward or to the side, participants bend their upper body at the waist. The direction leaned in is the direction that the participants move. The greater the angle of the lean, the greater the speed they will move in the virtual world. Whereas in conventional VR, the body is often reduced to little more than a probing hand and roving eye, immersion in Osmose depends on the body's most essential living act, that of breath—not only to navigate, but more importantly—to attain a particular state-of-being within the virtual world.
- To move up, participants inhale. To move down participants exhale. As a result, deep breathing can cause them to bob up and down in the virtual world.

While it's not necessary for participants to move their feet in order to move in the virtual world, they have the tendency to shuffle their feet or move around anyway. If participants move too far from that magnetic sensor controller, it will degrade the accuracy of the magnetic sensor and cause the virtual world to jitter.

Testing Environment

In creating the test environment, we used the following equipment:

- two computers
- a digital camera
- a wireless microphone
- a web camera
- software, such as Morae (by Tec Smith) and Premiere (by Adobe)

We began setting up Osmose's usability testing environment about two weeks before our first test. One of the first things we did in preparation for the tests was to obtain off-the-shelf software called Morae. Morae is an application designed to perform usability tests in a traditional testing environment. In other words, it expects a participant sitting at a desktop computer in a stationary setting. It records the participants' interaction with the system by recording the computer's screen output, as well as mouse clicks and keystrokes. It records participant's speech and facial expressions via a web-cam and a microphone. Morae then combines all of the different data streams into one video file. It also offers important and useful features such as real-time remote viewing and annotation of a recording.

Figure 6 shows the test environment we set up to test Osmose. In the diagram, monitor (A) showed what participants could see with their right eye, while monitors (B) and (C) showed their left eye view. The speakers (D) played Osmose's musical accompaniment.

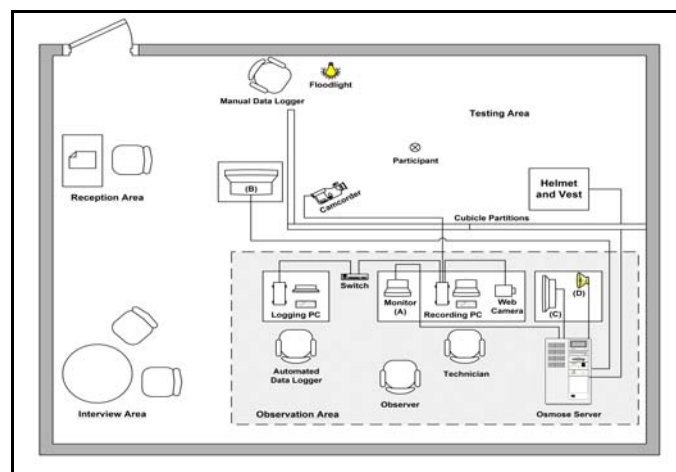


Figure 6: Osmose Test Environment

We could not install Morae on the Osmose server to record the participant's interaction with it. This was because we were not allowed to disrupt Osmose's carefully custom-designed software by installing new software. Since Morae only functions on Windows, but the Linux PC ran Osmose's graphics, we couldn't install Morae on the appropriate computer anyway. Hence, we were not sure how we could effectively use Morae in this situation.

We turned our attention to how to record the participant's interaction with Osmose. The fact that participants had to stay near the magnetic sensor controller (in Figure 3) made our work easier. We marked an X on the ground where participants were supposed to stay during the test (see the label "Participant" in Figure 6).

To record participants, we placed a digital camcorder facing the X on the ground (Figure 7). After some pre-tests, putting ourselves in place of the participants, we could see that even though participants could move around during the test, they couldn't move enough to leave the field of view of the camera. What's more, since interaction with Osmose is limited to upper body motion, recording the participants' upper body also allowed us to record their interaction with the system. This solved problems 1 and 2 as described in the previous section.



Figure 7: Camcorder for Observing Participants

Even though darkness was required for a good Osmose experience, we were able to use a floodlight and the camcorder's low light mode in order to overcome that limitation. Some pre-testing allowed us to adjust the lighting to produce optimal results. We had overcome the lighting problem.

One question that remained was how to store information recorded by the camcorder. We could have recorded the data onto mini-DV tapes. The trouble with that option was that if we also separately recorded what the participant saw then we would have two un-synchronized, separate data sets. We wondered how we could stitch together these two data streams. One technique would have been to buy or rent very expensive TV broadcasting hardware, which would have allowed us to merge multiple input signals and combine them into one stream. However, this was too expensive (on the order of \$100 000 CAD to purchase). We had to think of other options. This was where Morae came in. We realized that we could use a computer to inexpensively assemble all of the different streams of data.

We used the Premiere application to display the camcorder's output on the Recording PC's monitor—thus using the camcorder as an electronic eye rather than a recording device. Then, we set up a webcam to observe monitor C (in Figure 6), which allowed us to see and record what the participant would be seeing (see Figure 8 for a picture of the webcam setup). Pre-tests showed us that the recorded video quality was low, but that it was still good enough for our purposes. We didn't need to be able to see tiny details. We only needed to know what was happening in Osmose's virtual world at any given time.

Finally, by running Morae Recorder, we were able to record what the Recording PC was showing on its monitor (thus capturing the camcorder's output), and simultaneously record from the webcam. This was all saved in a single synchronous video file, which was exactly what we needed. This technique solved our remaining problems.

There was one last thing we wanted to record, which was what participants would be saying during the test. We attached the wireless microphone transmitter to Osmose's vest, and then plugged the receiver into the Recording PC's sound card. Morae was able to take the input from the wireless microphone and integrate it into the single recorded file. We also connected a set of headphones to the wireless receiver so that the testers could hear the participant clearly, if necessary, even through the cubicle partitions that muffled the participant's voice. This way, the data loggers could clearly hear the participant, without the participant being able to see them.

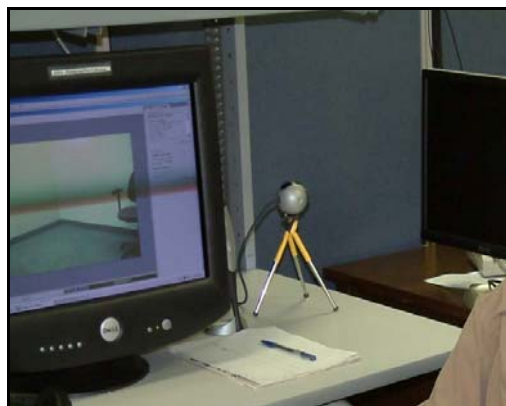


Figure 8: Recording PC and Webcam

We needed to know how large the Morae Recorder files would be. We wanted that size to be less than 650 MB, so that we could burn any recording onto a CD-ROM. We knew that a typical test session would normally last about 15 minutes, since Osmose automatically ended sessions after 15 minutes. However, we supposed that a session potentially could last up to 40 minutes (especially if we extended the recording to before and after the Osmose session was started, to record the process of putting on and taking off the helmet and vest).

We ran through Morae's various recording quality options, and for each set of options, recorded for 1 minute. Multiplying the resulting data file's size by 40 let us know approximately how large the file would be after 40 minutes. Every combination proved to be adequate, so we opted for the most portable one: Morae's basic built-in full-motion video compression scheme at a 15 frame per second (fps) capture rate with 1024x768 resolution on the monitor, and 640x480 resolution on the webcam. We did find that we could get much better compression using other schemes, but we opted for portability since the better compression was not needed.

We also wanted to enable Morae's remote viewer capability. This would allow a data logger, located at the Logging PC, to embed annotations inside the Morae recordings being made on the Recording PC. To accomplish this, we installed Morae Remote Viewer on the Logging PC, and then connected the Recording PC to the Logging PC via Ethernet (RJ-45) cables and a network switch. As a final note, even though Osmose did play music, the music did not in any way affect interaction with Osmose. Therefore, we judged that it was not necessary to record the music for the purpose of this usability test.

Throughout the process of setting up the usability test environment, we considered what sort of testing procedure would be required and how many people would be required to conduct each test. The details of the procedure and the roles and responsibilities of each tester evolved naturally as we built and tested the system. In addition, refinements continued during the first several live test sessions.

We used the following equipment to set up the usability test (Figure 6):

- **Camcorder:** a high-quality digital video camera with low light capability. Mounted on a tripod.
- **Computer (“Recording PC”):** a PC equipped with dual 2 GHz Xeon processors, 1 GB RAM, Windows 2000 Professional operating system, Parhelia/P650/P750 Dual Head 128 MB graphics card, and Sound Blaster Live sound card.
- **Computer (“Logging PC”):** a PC equipped with a 2.8 GHz Pentium 4 processor, 512 MB RAM, Windows XP Home Edition operating system, and Radeon 9200 128 MB graphics card.
- **Floodlight:** a bright lamp designed to provide ambient lighting.
- **Headphones:** standard cheap headphones.
- **Computer Monitors:** We used three different monitors. They allowed test conductors, located in various places, to observe what the participant was seeing from inside the helmet. In Figure 6, Monitor (A) showed what participants could see with their right eye, while monitors (B) and (C) showed their left eye view.
- **Morae Recorder:** software by TechSmith that records a computer’s screen, mouse clicks, keystrokes, input from a webcam, and input from a microphone. Morae Recorder combines all of that information inside a single synchronized file. This software was installed on the Recording PC.
- **Morae Remote Viewer:** software by TechSmith that connects with a computer running Morae Recorder, and allows the data logger to see what is being recorded in real-time and to add annotations. This software was installed on the Logging PC.
- **Network Switch:** a computer network apparatus used to connect the Recording PC with the Logging PC.
- **Premiere:** software by Adobe that can display input from a digital video camera on a computer screen. This software was installed on the Recording PC.
- **Speakers:** These allowed the testers to hear what the participant was hearing from inside the Osmose helmet.
- **Webcam:** a standard, cheap web camera mounted on a miniature tripod.
- **Wireless Microphone:** a microphone composed of two parts: a wireless transmitter and a receiver. The receiver can simultaneously send its sound output to two different sources: directly into a computer’s sound card, and to a speaker system (e.g., headphones).

Results

During the usability test setup process, the major problem we encountered was getting the video footage of the participant, which was being displayed by Premiere, to be screen-captured by Morae. We originally tried to accomplish this on the Logging PC, but Morae did not function correctly. Morae could not capture Premiere’s live video feed. In the video file, the feed would manifest itself as featureless grey or as a hole in the screen capture.

When we tried to do exactly the same thing on the Recording PC, that computer was able to record Premiere’s live video feed—but only if we were logged in as the administrator. While this result was very mysterious at the time, a later investigation of TechSmith’s customer support website revealed the probable source of this problem: “Some systems have graphics hardware which can perform a “hardware overlay” of video on the screen which bypasses the normal Windows display memory that Morae software captures. The best solution is to disable hardware acceleration in the application that is playing the video, or disable hardware acceleration system wide (Morae, 2006).” Hence, it seemed that the Logging PC’s graphics card used a hardware overlay to play the Premiere video, but the Recording PC did not use it when logged into the administrator account.

We ran into a minor problem with Morae Remote Viewer. By default, its Include Audio and Picture-In-Picture feature was turned on. Without this feature, Remote Viewer would only show the screen capture from the Recording PC. With this feature turned on, Remote Viewer also showed the webcam output (picture-in-picture) and outputted the participant’s recorded audio. Leaving this feature turned on increased the network load and added a delay to the Remote

Viewer connection with the Recording PC. This was very confusing for the data logger because it would desynchronize what was happening at one moment in the lab with the audio-video signal that the Logging PC was showing.

Users' Interaction Behaviors

User's interactions in Osmose can be classified into two main categories: i) navigation and viewpoint control and ii) object interactions, such as picking, grabbing, rotating, and moving objects. The interaction behavior in this study was only related to navigation and viewpoint control, since there was no object interaction.

Studying the interaction behaviors of Julie and Charles, two typical people representing our subjects (Figure 9 and 10) were identified using the inquiry-based approach proposed by Tahir (1997). We gathered the following data points:

- The average number of interactions per minute related to head was higher comparing with other methods of interaction. This was due to the fact that head movement to look around is a real world interaction habit.
- Both Julie and Charles were first freely and rapidly navigated through the space to have an overview of the environment. During this period, they used a lots of head movements. However, the average rate of interactions related to head decreased over time as their focus to a certain point increased.
- Julie was quite comfortable with breath interaction, whereas Charles found the breathing unnatural and awkward.
- As you can see in Figure 10, the graph of total number of interactions per minute for both Julie and Charles has a shift. This could indicate the moment when they learned how to interact with the system. When their brain and body movements adapted to the new environment, and when an unnatural movement such as leaning or breathing to move around became more natural.
- This shift happened sooner for Julie than for Charles, which indicated Julie's learnability was higher than Charles' for this activity.

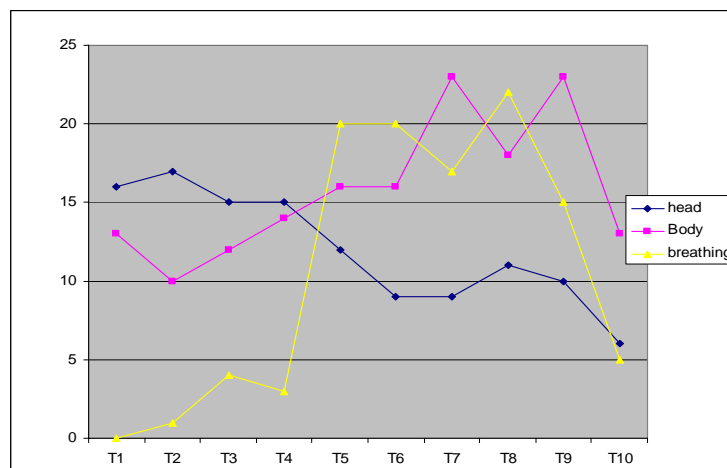


Figure 9. Julie's interaction behaviors

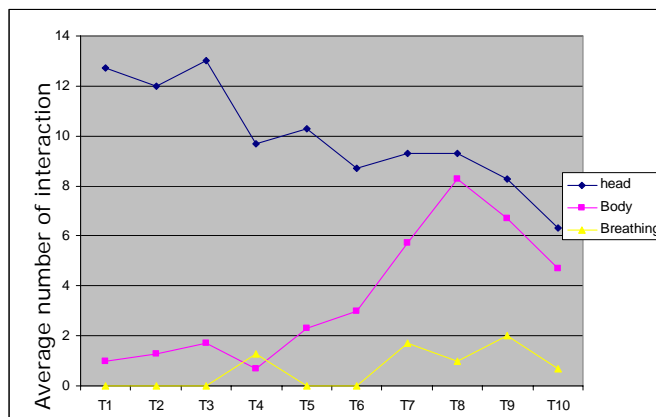


Figure 10. Charles' interaction behaviors

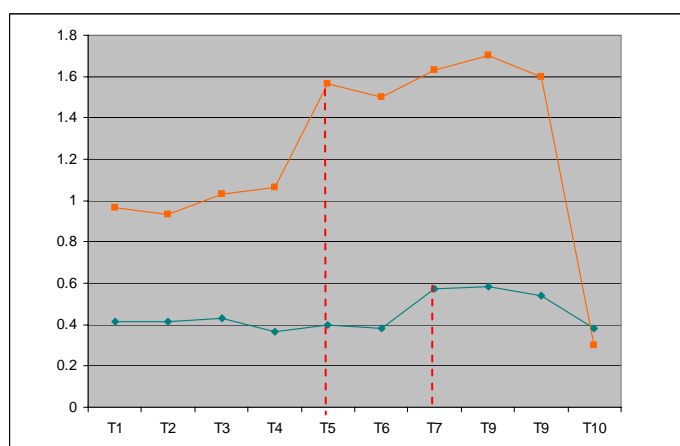


Figure 11. Comparison between Charles' and Julie's interaction behaviors

Our experience with Osmose was sufficient to allow us to make a few preliminary conclusions regarding the design of immersive systems. Osmose was designed to be easy to move and deploy in art galleries and show rooms. It is packed in three boxes for transportation. As a result, rather than using bulky wearable devices (e.g., the HMD), we can use lightweight wearable devices and bulky stationary devices. Osmose had the capacity to be immersive. This will probably be a useful trait although it is as-yet unclear how to accomplish this or what the benefits will be. It was clear that several elements of Osmose's construction made it less immersive. The breathing sensor constricted the participant's chest and made it hard to breathe. The helmet was very heavy and would typically hurt the participant's head or nose. Participants would often get tangled in wires. All of these elements contributed to making the experience less immersive, since each of these problems could remind participants that they were wearing devices and not truly inside the VR.

Recommendations

We gave a detailed description of how we successfully set up a usability test environment and methodology for an HMD-based VR system. We learned the following from this experience:

- It is possible to set up a usability-testing environment for an unconventional virtual reality system, using only low-cost, reusable, off-the-shelf components.
- It is important to begin preparing the testing environment in advance of the testing (two weeks was enough in our case). This gives time to work out bugs.

- Pre-testing, putting testers in place of the future participants should be used to develop a testing procedure and to find faults in the testing environment. Don't assume that any component or, especially, any component combination is working correctly until it has been tested. This allowed us to catch problems, like Morae Recorder's failure to capture a live video feed and the synchronization problem with Morae Remote Viewer, before it was too late.
- Having clearly defined roles and responsibilities help ensure that no element of the testing process is neglected.
- Have a defined procedure for conducting the usability test. This will ensure that the testers work smoothly together. However, the procedure should not be written in stone and should be open to opportunistic adaptation. Run through the procedure at least once in a pre-test to make sure that nothing has been forgotten and that the procedure runs smoothly.
- Once usability testing begins, regularly perform reviews of the research results. Doing so can help you optimize your test environment and methodology, and catch critical errors that were not apparent during pre-testing.

Conclusion

As highlighted in this paper, there is a need to develop novel usability testing environments and methodologies because computerized interactive systems are evolving and the current labs are limited. The new techniques will be necessary for testing a new generation of interactive systems, such as augmented reality, virtual reality, and mobile devices. It is true that today's VR provides very limited tactile feedback, almost no proprioceptive feedback (as would be the case for walking on a sandy beach or on rough terrain), rare opportunities to smell, and little mobility. However, in the coming years, some or all of these limitations will disappear. As they disappear, usability labs will need to keep pace and continue to adapt their techniques. This experience report was a contribution toward a reengineering and upgrade of the current testing lab.

We customized existing techniques and technologies useful for desktop applications to our unconventional system under test. In so doing, we learned many important lessons on the practicalities of creating usability test environments for unconventional systems. It is our hope that our experiment will be useful to people performing usability tests on similar systems.

The amount of detail supplied by this report should make it relatively easy to repeat our experiment. We provided the final result of our experimental procedure and our knowledge of the required roles and responsibilities. In particular, the low cost (perhaps \$3 000 CAD, not including the cost of the computers) of our solution makes it broadly applicable.

Our materials were off-the-shelf, and therefore, reasonably easy to find. We also learned the nature of the barriers that Osmose had that often prevented its users from becoming immersed in the VR, and thus, if we wanted to implement the same thing, we would know how to do it better. The main barriers to immersion were the heavy helmet, the breathing sensor that restricted breathing, and the wires that participants could tangle themselves in.

Practitioner's Take Away

- There is a need today to rethink and develop new usability testing environments and methodologies for unconventional interactive systems such as virtual reality, immersive environments and mobile applications
- For Virtual Reality (VR) many difficult design questions arise, such as how do we record what the participant is seeing when the computer's monitor is really a helmet? How do we record the participant's interaction with the system when the participant is able to freely move about the entire world?
- We provided a detailed description of how we successfully set up a usability test environment and methodology for an immersive interactive virtual-reality environment installation with 3D computer graphics and interactive 3D sound, a head-mounted display and real-time motion tracking based on breathing and body movements.

Acknowledgements

We would like to thank everyone who participated in the setup and execution of the Osmose usability tests: Dr. Sudhir Mudhur, Ashraf Gaffar, Fanglei Wang, Harkirat Padda, Homa Javahery, Muhammad Faraz Anwar, Naouel Moha, and Yojana Joshi.

References

- Davies C. (2004) Virtual Space, in F. Penz, G. Radick, and R. Howell, Eds., *Space in Science, Art and Society* (pp. 69-104). Cambridge: Cambridge University Press.
- Davies C. & Harrison, J. (1996) Osmose: Towards Broadening the Aesthetics of Virtual Reality, *ACM Computer Graphics* [special issue on Virtual Reality], vol. 30, 25-28.
- Dumas J.S. & Redish, J. C. (1999) *A Practical Guide to Usability Testing*. Bristol, England: Intellect Books.
- Hix, D., & Gabbard, J. (2002) Usability Engineering of Virtual Environments. In K. Stanney (Ed.), *Handbook of Virtual Environments: Design, Implementation, and Applications* (pp. 681-699). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Immersence. Osmose Description. Accessed on Dec 2006 from <http://www.immersence.com/osmose/osmose.htm>.
- Kruger, R., Carpendale, M.S.T., Scott, S.D., & Greenberg, S. (2003) How People Use Orientation on Tables: Comprehension, Coordination and Communication. In *Proceeding of GROUP 03*, (pp. 369-378).
- MacIntyre, B., Gandy, M., Dow, S., & Bolter, J. D. (2004) Tools: DART: a toolkit for rapid design exploration of augmented reality experiences. In *Proceedings of the 17th annual ACM symposium on User interface software and technology* (pp. 197-206).
- McCoy, J. M. & Evans G. W. (2002) The Potential Role of the Physical Environment in Fostering Creativity, *Creativity Research Journal 2002*, Vol. 14, Nos. 3 & 4, 409-426.
- Poupyrev I., Weghbrst S., Billinghamurst M., & Ichikawa T. (1997) A Framework and Testbed for Studying Manipulation Techniques for immersive VR. In *ACM VRST 97* (pp 2-28).
- Ryall, K., Forlines, C., Shen, Meredith, C. & Morris, R. (2004) Tabletop design: Exploring the effects of group size and table size on interactions with tabletop shared-display groupware. In *Proceedings of the 2004 ACM conference on Computer-supported Cooperative Work* (pp. 284-293).
- Shen C., Vernier, F. D., Forlines, C. & M. Ringel. (2004) DiamondSpin: An Extensible Toolkit for Around-the-Table Interaction, presented at *ACM Conference on Human Factors in Computing Systems (CHI)*.
- Sherman W.R., & Craig, A. B. (2003) *Understanding Virtual Reality*. San Francisco: Morgan Kaufmann Publishers.
- Sadowski, W., & Stanney, K. (2002) Presence in Virtual Environments. In K. Stanney (Ed.), *Handbook of Virtual Environments: Design, Implementation, and Applications* (pp. 791-806). Mahwah, New Jersey: Lawrence Erlbaum Associates.
- Tahir, M. F. (1997) Who's on the other side of your software: Creating user profiles through contextual inquiry. In *Proceedings of Usability Professionals Association Conference*.
- TechsSmith. (2007) Morae Observational Software. Accessed on Nov, 2007 from <http://www.techsmith.com/morae.asp>.

About the Authors



Ahmed Seffah

is an associate professor of human-centered software engineering at Concordia University. His research is at the crossroads for software engineering and human-computer interaction, as well as their applications to environmental challenges.



Jonathan Benn

is a software engineer at Nuance Communications, Montreal Branch. He graduated from Concordia University with a bachelor's and master's of software engineering.



Halima Habieb-Mammar

is a postdoctoral fellow at Concordia University. Her current research interests include user and usability requirements, accessibility and universal usability for multiple user interfaces. She received her Ph.D. (2004) in Computer Science from the French Engineering University.