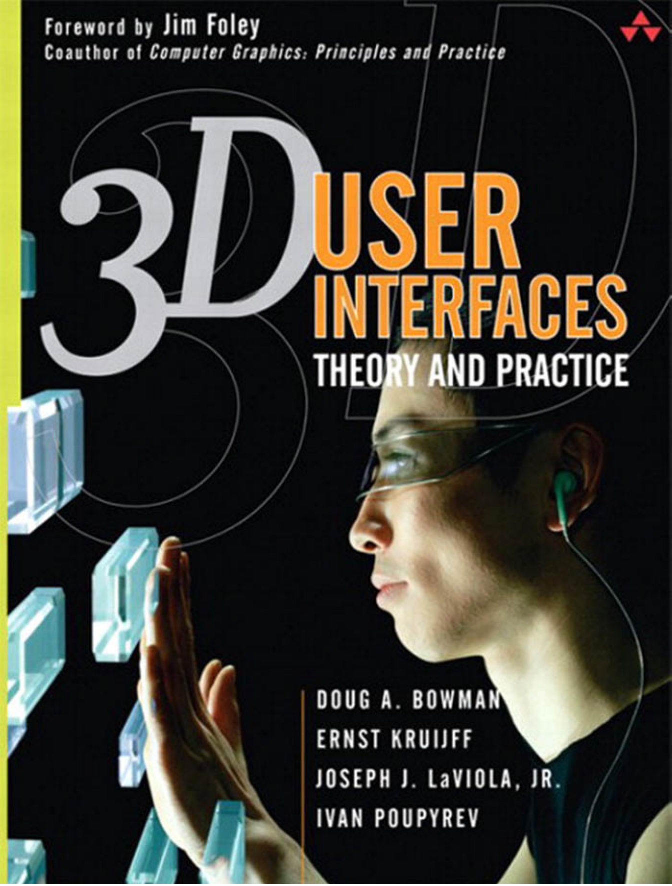
Foreword by Jim Foley
Coauthor of *Computer Graphics: Principles and Practice*

3D USER INTERFACES

THEORY AND PRACTICE



DOUG A. BOWMAN
ERNST KRUIJFF
JOSEPH J. LaVIOLA, JR.
IVAN POUPYREV

3D User Interfaces

This page intentionally left blank

3D User Interfaces

Theory and Practice

Doug A. Bowman
Ernst Kruijff
Joseph J. LaViola, Jr.
Ivan Poupyrev

Addison-Wesley

Boston • San Francisco • New York • Toronto • Montreal
London • Munich • Paris • Madrid • Capetown
Sydney • Tokyo • Singapore • Mexico City

The publisher offers discounts on this book when ordered in quantity for bulk purchases and special sales. For more information, please contact:

U.S. Corporate and Government Sales
(800) 382-3419
corpsales@pearsontechgroup.com

For sales outside of the U.S., please contact:

International Sales
(317) 581-3793
international@pearsontechgroup.com

Visit Addison-Wesley on the Web: www.awprofessional.com

Library of Congress Cataloging-in-Publication Data

3D user interfaces : theory and practice / Doug A. Bowman . . . [et al.].
p. cm.
Includes bibliographical references and index.
ISBN 0-201-75867-9 (hardbound : alk. paper)
1. User interfaces (Computer systems) 2. Three-dimensional display systems. I. Bowman, Doug A., 1971–

QA76.9.U83A14 2004
005.4'37—dc22

2004008403

ISBN 0-201-75867-9

Text printed in the United States on recycled paper at Courier Westford in Westford, Massachusetts.

3rd Printing March 2011

For Dawn, Drew, and Caroline, my true joys on this earth.

—Doug

To my family, for their faith, affection, and support.

—Ernst

To my family, with love—they are my life.

—Joe

To my parents, for bringing me up.

—Ivan

This page intentionally left blank

Contents

This page intentionally left blank

Foreword

Three-dimensional user interfaces are finally receiving their due! Research in 3D interaction and 3D display began in the 1960s, pioneered by researchers like Ivan Sutherland, Bob Sproull, Fred Brooks, Andrew Ortony, and Richard Feldman. Although many commercially successful 3D applications exist—computer-aided design and simulation, radiation therapy, drug discovery, surgical simulation, scientific and information visualization, entertainment—no author or group of authors has written a comprehensive and authoritative text on the subject, despite a continuing and rich set of research findings, prototype systems, and products.

Why is that? Why is it that this book by Doug Bowman, Ernst Kruijff, Joe LaViola, and Ivan Poupyrev is the first thorough treatment of 3D UIs?

Perhaps it was our digression during the last 20 years to the WIMP GUI. After all, the Windows, Icons, Menus, and Pointers GUI is used very widely by millions of users. Mac OS and Microsoft Windows users know it well, as do many UNIX users. Indeed, every user of the Web works with a GUI, and this year there are many hundreds of millions of them. Two-dimensional GUIs will be with us for a long time. After all, a lot of the workaday world with which we deal is flat—not just our Web pages but our documents, presentations, and spreadsheets too. Yes, some of these can be extended to 3D, but most of the time, 2D is just fine, thank you very much. Furthermore, pointing and selecting and typing *are* relatively fast and relatively error-free—they work, and they work well.

Perhaps it is that not as many people use 3D GUIs as use the 2D WIMP GUI, and so they are not thought to be as important. But the above list of 3D applications involves multibillion-dollar manufacturing industries, such as aerospace and automotive, and equally large and even more important activities in the life-saving and life-giving pharmaceutical and health care industries.

Perhaps it was that we needed the particular set of backgrounds that Doug, Joe, Ivan, and Ernst bring to the table. Doug comes out of the GVU Center at Georgia Tech, where he worked on 3D UIs with Larry Hodges and others and learned the value of careful user studies and experimentation, and he is now a member of an influential HCI group at Virginia Tech; Joe works at Brown with Andy van Dam, a long-time proponent of rich 3D interaction; Ivan comes from the HIT Lab at the University of Washington, where he worked with Tom Furness and Suzanne Weghorst, and now works with Jun Rekimoto at Sony CSL; and Ernst works with Martin Goebel in the VE Group at Fraunhofer IMK in Germany.

Whatever the case, I am excited and pleased that this team has given us the benefit of their research and experience. As I reviewed the draft manuscript for this book, I jotted down some of the thoughts that came to my mind: comprehensive, encyclopedic, authoritative, taxonomic; grounded in the psychological, HCI, human factors, and computer graphics literature; grounded in the personal research experiences of the authors, their teachers, and their students.

I myself have long preached the importance of integrating the study of the computer with the study of the human. Indeed, this is the key premise on which I built the GVU Center at Georgia Tech. This book certainly follows that admonition. There are numerous discussions of human issues as they relate to 3D navigation and interaction, drawing on references in psychology and human factors.

This is indeed a book for both practitioners and researchers. The extensive literature reviews, examples, and guidelines help us understand what to do now. Combined with the research agenda in Chapter 13, The Future of 3D User Interfaces, the material also helps us have a sense of what it is that we do not yet know.

I particularly commend to readers the Chapter 11 discussion of evaluating 3D UIs. We in the computer graphics community have tended to design devices and techniques and then "throw them over the wall" to the user community. This is not the route to success. Careful study of user needs coupled with evaluation as part of the ongoing design cycle is much more likely to lead to effective techniques. The authors, all of whom have grappled with the difficult task of designing 3D interfaces, know from first-hand experience how crucial this is. Their section 11.4, on the distinctive characteristics of the 3D interface evaluation process, is a wonderful codification of that first-hand knowledge.

Thanks to Doug and Ernst and Joe and Ivan!

Jim Foley
GVU Center
College of Computing
Georgia Tech
March 2004

Preface

An architect sits in her home office, putting the final touches on the design of the new entrance to the city park. A three-dimensional virtual model of the park appears in front of her on the desk's surface. She nudges a pathway slightly to the right to avoid a low-lying area, and then makes the model life-size so she can walk along the path to view the effect. "Those dark colors on the sign at the entrance are too foreboding," she thinks, so she quickly changes the color palette to brighter primary colors. She looks up and notices that the clients are arriving for the final design review meeting. They are located in other offices around the city, but they can all view the 3D model and make suggested changes, as well as communicate with one another. "What's the construction plan?" asks one of the clients. The architect starts an animation showing the progress of the project from start to finish. "That first step may not work," says the client. "The excavation is much too close to the existing playground. Let me show you." He looks out his window, which has a view of the park, and overlays the virtual construction plan on it. "You're right," says the architect, "let's plan to move the playground slightly—that will be much cheaper than changing the construction site." After viewing the effects of the change, all agree that this plan will work, and the meeting adjourns.

This scenario and others like it illustrate the enormous potential of 3D environments and applications. The technology to realize such a vision is available now, although it will certainly be improved. But the scenario also leaves out a great deal of information—information that is crucial to making this dream a reality. How did the architect load the park model, and how does she manipulate her view of it? What technique is used to change the pathway? How can multiple clients all manipulate the model at the same time? How do the participants appear to each other in the virtual space? How is the speed and playback of the animation controlled? How did the client instruct the system to merge the real and virtual scenes?

These questions all relate to the design of the *user interface* (UI) and *interaction techniques* for this 3D application, an area that is usually given only a cursory treatment in futuristic films and books. The scenarios usually either assume that all interaction between the user and the system will be "natural"—based on techniques like intuitive gestures and speech—or "automatic"—the system will be so intelligent that it will deduce the user's intentions. But is this type of interaction realistic, or even desirable?

This book addresses the critical area of *3D UI design*—a field that seeks to answer detailed questions, like those above, that make the difference between a 3D system that is usable and efficient and one that causes user frustration, errors, and even physical discomfort. We present practical information for developers, the latest research results, easy-to-follow guidelines for the UI designer, and relevant application examples. Although there are quite a few books devoted to UIs in general and to 2D UI design in particular, 3D UIs have received significantly less attention. The results of work in the field are scattered throughout numerous conference proceedings, journal articles, single book chapters, and Web sites. This field deserves a reference and educational text that integrates the best practices and state-of-the-art research, and that's why this book was created.

How This Book Came to Be

The story of this book begins in April 1998, when Ivan Poupyrev and Doug Bowman were doctoral students at Hiroshima University and Georgia Tech respectively, working on 3D interaction techniques for object manipulation in virtual environments (VEs). We started a lively email discussion about the design and usability of these techniques and about 3D UIs in general. Ivan, who was at the time a visiting research student at the University of Washington, suggested that the discussion would be even more profitable if other researchers in this new area could join in as well, and so the 3DUI mailing list was born. Since that time, over 100 researchers from around the globe have joined the list and participated in the discussion (to see an archive of all the list traffic or to join the list, check out http://www.3dui.org). Joe LaViola and Ernst Kruijff were two of the first people to join the list.

In August of that same year, Doug forwarded to the list a call for tutorials for the upcoming IEEE Virtual Reality Conference. After some discussion, Joe, Ivan, and Ernst agreed to join Doug to organize a tutorial on "The Art and Science of 3D Interaction." The tutorial was a big hit at the conference in Houston, and the four of us continued to present courses on the topic at ACM Virtual Reality Software and Technology 1999, IEEE VR 2000, and ACM SIGGRAPH 2000 and 2001.

After developing a huge amount of content for the notes supplements of these courses, we decided it would be silly not to compile and expand all of this information in book form. Furthermore, there was no way to include all the information available on 3D UIs in a one-day course. And that's why you're holding this book in your hands today—a book containing information on 3D UIs that can't be found in any other single source.

What's in the Book

The title of this book emphasizes that we have written it for both academics/researchers and practitioners/developers—both those interested in basic research and those interested in applications. Most chapters of the book integrate both theory and practical information. We intend the book to be used both as a textbook (see suggestions below) and as a reference work.

Theory-related content includes the following:

- Sections on the psychology and human factors of various 3D interaction tasks
- Information on different approaches for the evaluation of 3D UIs (Chapter 11)
- Results from empirical studies of 3D interaction techniques
- A research agenda for 3D interaction (Chapter 13)
- Lists of recommended further reading at the end of most chapters
- A comprehensive bibliography of important research articles

Practice-related content includes the following:

- Principles for choosing appropriate input and output devices for 3D systems (Chapters 3 and 4)
- Details and helpful tips for the implementation of common 3D interaction techniques
- Guidelines for the selection of interaction techniques for common 3D tasks
- Case studies of 3D UIs in real-world applications

The book is organized into five parts. Part I introduces the topic of 3D UIs. Part II discusses the input and output device technology used in the development of 3D UIs, with an emphasis on the impact of these devices on usability and performance. Part III presents a wide range of 3D interaction techniques for the common tasks of navigation, selection and manipulation, system control, and symbolic input. In Part IV, we discuss the design, development, and evaluation of complete 3D UI metaphors and applications. Finally, Part V considers the future, with chapters on 3D interaction in augmented reality applications and a research agenda for 3D UIs. The appendix includes information on required mathematical background and is followed by a bibliography of 3D UI references.

Throughout the book, we offer several special features. First, most chapters contain numerous *guidelines*—practical and proven advice for the designer and developer. Guidelines are indicated in the text like this:

> Follow the guidelines in this book to help you design usable 3D UIs.

We also include implementation details for many of the most common and useful interaction techniques. We describe these algorithms using a combination of textual and mathematical descriptions (to avoid a bias toward any particular development tool or programming style).

How to Use the Book and Related Material

If you are a 3D UI developer: Professional developers can use the book for inspiration and guidance in the design, implementation, and evaluation of applications with 3D UIs. In the design process, developers can consider overall UI metaphors from Part IV, choose specific interaction techniques from Part III, and match these with appropriate input and display devices from Part II. The design guidelines from all of these sections should help developers make rational, informed decisions. The implementation of the 3D UI can benefit from the textual and mathematical descriptions of interaction techniques we provide in Part III. Finally, developers can choose evaluation methods and assess the usability of their applications based on the information in Chapter 11.

If you are a teacher: The book can also be used as a textbook in several different types of university-level courses. A graduate course on 3D UI design could use it as a primary textbook. A more generic virtual environments course could use Parts I, II, and III of this book as an introduction to the basic technology and techniques used in VE interaction. An undergraduate HCI course could pull information from Parts I and IV in a module on 3D interfaces and their differences from traditional UIs. Implementation of common techniques from Part III could enhance a course on interactive 3D graphics.

If you are a researcher: This book can serve as a comprehensive reference guide for researchers engaged in 3D UI design or evaluation, the investigation of 3D applications, or the use of VEs or augmented reality. The research agenda in Chapter 13 also provides researchers and research students with a list of important questions to be addressed in the field. It could even be used as the starting point for a PhD student looking for a topic related to 3D UIs.

3D UI design is a fast-moving and evolving field. Therefore, we are committed to updating the material in this book. One way we will do this is through the book's official Web site at http://www.3dui.org. This site will contain information and links related to the latest 3D UI research and applications, organized in the same manner as the book so you can easily find new information about the topics in a particular part or chapter. The site will also allow you to join the 3DUI mailing list. We also ask for your help in keeping the book up to date. Send us your comments, clarification questions, or links to additional information by visiting the Web site above and using the online feedback form. Or email us directly at 3dui@3dui.org. Your comments will help us update the Web site, as well as future editions of this book.

Acknowledgments

This book would not have been possible without the hard work, support, and intelligence of a large group of people.

First, we offer our gratitude to the reviewers who gave their time and energy in improving the quality of the book. Their comments and suggestions have made the book more complete, more readable, and more useful. Thanks to Ben Shneiderman, Harry Hersh, D. Jay Newman, Jeff Pierce, Dieter Schmalstieg, and Bob Zeleznik for providing this invaluable service. Special thanks go to Jim Foley for his encouragement and support.

Next, we would like to thank our editor at Addison-Wesley, Peter Gordon, for his invaluable advice and encouragement. The rest of the staff, including Bernie Gaffney, Amy Fleischer, Julie Nahil, Heather Mullane, and Curt Johnson have also been extremely helpful. Thanks also to Simone Payment and Carol Lallier for their competent and professional work during the production phase.

All of us would like to personally thank our colleagues in the 3D UI community for their fruitful discussions and collaborations. They include Mark Mine, Robert Lindeman, Matthew Conway, Ken Hinckley, Shumin Zhai, Kiyoshi Kiyokawa, Chris Shaw, Mark Billinghurst, Rudy Darken, Pablo Figueroa, and Bernd Fröhlich.

Portions of this material are based upon work supported by the National Science Foundation under Grants No. DUE-0127326 and IIS-0237412. Any opinions, findings and conclusions, or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation (NSF).

Doug Bowman: I would like to thank my wife, Dawn, for her unfailing love and support, as well as my extended family and friends, especially those at Grace Covenant Presbyterian Church. Much gratitude is due to Joe, Ivan, and Ernst for seeing this project through and for all their years of friendship and collaboration. Thanks also go to my colleagues and students at Virginia Tech, including Chris North, Ron Kriz, Mehdi Setareh, Walid Thabet, Thomas Ollendick, David Cox, Debby Hix, John Kelso, Joe Gabbard, Chad Wingrave, Jian Chen, Nicholas Polys, Wendy Schafer, and Marcio Pinho. Past colleagues at Georgia Tech also deserve thanks. They include Larry Hodges, Drew Kessler, David Koller, Donald Johnson, Donald Allison, Brian Wills, Jean Wineman, Jay Bolter, Elizabeth Davis, Albert Badre, and Ben Watson.

Ernst Kruijff: First of all, my thanks go to Doug, Ivan, and Joe for their great cooperation, help, and extensive discussions. Thanks also go to my

parents, my brother, and my sister-in-law for their support, each in their own way. Furthermore, my thanks go to all of my colleagues at the VE group of IMK, especially Martin Goebel for being a great chief, Gerold Wesche, Andreas Simon, Gernot Goebbels, Stefan Conrad, Aeldrik Pander, and Steffi Beckhaus for their past and current cooperation and help during the making of this book and its instigators, the courses we (the authors) gave together. My thanks also go to my past colleagues at Bauhaus University, especially the members of igroup, in particular Holger Regenbrecht. Furthermore, thanks to all the students who helped in many projects, especially Stefan Hansen, Arnold Mueller, Jakob Beetz, and Hartmut Seichter. Finally, my thanks go to Dieter Schmalstieg for being patient with my PhD efforts.

Joe LaViola: I would like to thank my PhD thesis advisor, Andries van Dam, for his forgiveness and patience and for putting up with someone as hardheaded as me. Thanks also to my mom and dad, Jamie, Nick, Heidi, and my friends Don Carney and Dave Gondek for their love and support throughout the writing of this book. My colleagues Robert Zeleznik, Daniel Keefe, Daniel Acevedo Feliz, Andrew Forsberg, Loring Holden, Tim Miller, Steve Dollins, Lee Markosian, David Karelitz, Tim Rowley, Christine Waggoner, and all the members of the Brown University Computer Graphics Group past and present also deserve thanks. Finally, thanks to my coauthors for their hard work and dedication.

Ivan Poupyrev: I would never have been able to work on this book without the help and support I received from many people I was fortunate to meet. I am deeply indebted to Professor Tadao Ichikawa, who supervised my PhD thesis at Hiroshima University. I am also thankful to Professor Masahito Hirakawa for his help and support, as well as to my fellow students and researchers Bryn Holmes, Olivier Liechti, and Numada Tomokazu. I am grateful to the Japanese government for providing me with the Monbusho Scholarship for conducting graduate studies in Japan.

While working on my PhD thesis, I had the exceptional opportunity to spend almost three years at the Human Interface Technology Laboratory (HITL) at the University of Washington. The HITL experience made an enormous impact on me and formed me as a researcher. For this, I will always be grateful to Suzanne Weghorst, the Director of Research at the HITL, who invited me in summer 1995 to join HITL first as a summer intern and then as a visiting researcher; and Professor Tom Furness III, Director of HITL, who believed in me and provided invaluable moral and financial support. I was very fortunate to meet and work with Mark Billinghurst; our collaboration and friendship extended well beyond my stay at the HITL. I am also deeply thankful to Edward Miller for his help in developing a number of HITL 3D UI projects, in particular the V3D interface toolkit, as well as to Jerry Prothero and Hunter Hoffman for late-night discussions of chess, psychology, and the meaning of life; Tony Emerson for finding everything that I ever needed; Ann Elias for all her generous help while I at was the HITL; Mark Phillips, Jennifer Feyma, and Chris Airola for being friends and giving me life outside the lab.

The augmented reality chapter of this book includes a significant amount of work that I did while at the ATR Media Integration and Communication Research Labs in Kyoto. I am very thankful to all the amazing people I met and worked with there: Ryohei Nakatsu, Jun Ohya, Nobuji Tetsutani, Jun Kurumisawa, Tatsumi Sakaguchi, Keiko Nakao, Lew Baldwin, Desney Tan, Sidney Fels, Michael Lyons, Tal Shalif, Parham Zolfaghari, Christa Sommerer, and many others.

Finally, I am thankful to my family and friends in Russia, particularly my parents and brother, Pavel Poupyrev; my first research advisers, Vladimir Lischouk, Dinara Gazizova, and Vladimir Lukin; as well as to Sergei Tsimbalist, Vadim Malishev, Vitaly Lazorin, Constantin Guzovski, and Elena Krupskaya.

It is impossible to mention everyone in Japan, the United States, and Russia who helped, supported, and inspired me. Without this help, support, and inspiration I would not be able to take even my current modest steps toward designing and exploring future interaction technologies. My deepest gratitude and appreciation goes to all of you.

CHAPTER 11
Evaluation of 3D User Interfaces

Most of this book has covered the various aspects of 3D UI design. We have addressed questions such as, How do I choose an appropriate input device? How do I support wayfinding in large-scale environments? and What object manipulation techniques provide precise positioning? However, one of the central truths of human–computer interaction (HCI) is that even the most careful and well-informed designs can still go wrong in any number of ways. Thus, evaluation of UIs becomes critical. In fact, the reason we can provide answers to questions such as those above is that researchers have performed evaluations addressing those issues. In this chapter, we discuss some of the evaluation methods that can be used for 3D UIs, metrics that help to indicate the usability of 3D UIs, distinctive characteristics of 3D UI evaluation, and guidelines for choosing evaluation methods. We argue that evaluation should not only be performed when a design is complete, but that it should also be used as an integral part of the design process.

11.1. Introduction

Evaluation has often been the missing component of research in 3D interaction. For many years, the fields of VEs and 3D UIs were so novel and the possibilities so limitless that many researchers simply focused on developing new devices, interaction techniques, and UI metaphors—exploring the design space—without taking the time to assess how good the new designs were. As the fields have matured, however, we are taking a closer look at usability. We must critically analyze, assess, and compare devices, interaction techniques, UIs, and applications if 3D UIs are to be used in the real world.

11.1.1. Purposes of Evaluation

Simply stated, evaluation is the analysis, assessment, and testing of an artifact. In UI evaluation, the artifact is the entire UI or part of it, such as a particular input device or interaction technique. The main purpose of UI evaluation is the identification of usability problems or issues, leading to changes in the UI design. In other words, design and evaluation should be performed in an *iterative* fashion, such that design is followed by evaluation, leading to a redesign, which can then be evaluated, and so on. The iteration ends when the UI is "good enough," based on the metrics that have been set (or, more frequently in real-world situations, when the budget runs out or the deadline arrives!).

Although problem identification and redesign are the main goals of evaluation, it may also have secondary purposes. One of these is a more general understanding of the usability of a particular technique, device, or metaphor. This general understanding can lead to *design guidelines* (such as those presented throughout this book), so that each new design can start from an informed position rather than from scratch. For example, we can be reasonably sure that users will not have usability problems with the selection of items from a pull-down menu in a desktop application, because the design of those menus has already gone through many evaluations and iterations.

Another, more ambitious, goal of UI evaluation is the development of *performance models.* These models aim to predict the performance of a user on a particular task within an interface. For example, Fitts's law (Fitts 1954) predicts how quickly a user will be able to position a pointer over a target area based on the distance to the target, the size of the target, and the muscle groups used in moving the pointer. Such performance models must be based on a large number of experimental trials on a wide range of generic tasks, and they are always subject to criticism (e.g., the model doesn't take an important factor into account, or the model doesn't apply to a particular type of task). Nevertheless, if a useful model can be developed, it can provide important guidance for designers.

11.1.2. Terminology

We must define some important terms before continuing with our discussion of 3D UI evaluation. The most important term (which we've already used a couple of times) is *usability.* We define usability in the broadest sense, meaning that it encompasses everything about an artifact and a person that affects the person's use of the artifact. Evaluation, then, measures some aspects of the usability of an interface (it is not likely that we can quantify the usability of an interface with a single score). Usability measures (or metrics) fall into several categories, such as system performance, user task performance, and user preference (see section 11.3).

There are at least two roles that people play in a usability evaluation. A person who designs, implements, administers, or analyzes an evaluation is called an *evaluator.* A person who takes part in an evaluation by using the interface, performing tasks, or answering questions is called a *user.* In formal experimentation, a user is sometimes called a *subject.*

Finally, we distinguish below between *evaluation methods* and *evaluation approaches.* Evaluation methods (or techniques) are particular steps that can be used in an evaluation. An evaluation approach, on the other hand, is a combination of methods, used in a particular sequence, to form a complete usability evaluation.

11.1.3. Chapter Roadmap

We begin by providing some background information on usability evaluation from the field of HCI (section 11.2). We then narrow the focus to the evaluation of 3D UIs (specifically the evaluation of immersive VEs), looking first at evaluation metrics (section 11.3) and then distinctive characteristics of 3D UI evaluation (section 11.4). In section 11.5, we classify 3D UI evaluation methods and follow that with a description and comparison of two comprehensive approaches to 3D UI evaluation—testbed evaluation and sequential evaluation. Finally, we conclude with a set of guidelines for those performing evaluations of 3D UIs (section 11.7).

11.2. Background

In this section, we describe some of the common tools and methods used in 3D UI evaluation. None of these tools or methods is new or unique to 3D UIs. They have all been used and tested in many other usability evaluation contexts. We present them here as an introduction to these topics for the reader who has never studied HCI. For more detailed information, you can consult any one of a large number of introductory books on HCI (see the recommended reading list at the end of the chapter).

11.2.1. Tools for Evaluation Design and Implementation

The tools presented below are useful for designing, organizing, and implementing usability evaluations of 3D UIs.

User Task Analysis

A user task analysis (Hackos and Redish 1998) provides the basis for design in terms of what users need to be able to do with the application. This analysis generates (among other resources) a list of detailed task descriptions, sequences, and relationships, user work, and information flow. Typically, a user task analysis is provided by a design and development team, based on extensive input from representative users. Whenever possible, it is useful for an evaluator to participate in the user task analysis.

Scenarios

The user task analysis also shapes representative user task scenarios by defining, ordering, and ranking user tasks and task flow. The accuracy and completeness of a scenario directly affect the quality of the subsequent formative and summative evaluations because these methods typically do not reveal usability problems associated with a specific interaction within the application unless it is included in the user task scenario (and is therefore performed by users during evaluation sessions). Similarly, in order to evaluate how well an application's interface supports high-level information gathering and processing, representative user task scenarios must include more than simply atomic, mechanical- or physical-level tasking; they should also include high-level cognitive, problem-solving tasking specific to the application domain. This is especially important in 3D UIs, where user tasks generally are inherently more complex, difficult, and unusual than in many GUIs.

Taxonomy

Taxonomy is defined as the science of classification, but it has also come to mean a specific classification scheme. Many different types of taxonomies have been used in 3D UI research, including multidimensional

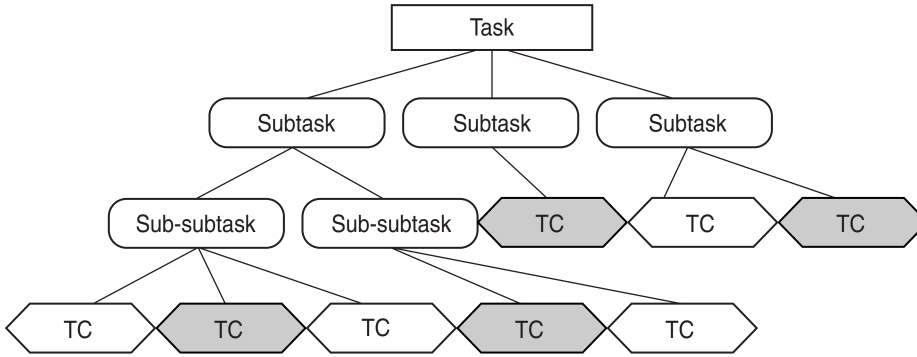


Figure 11.1 *Generic technique-decomposition taxonomy. The shaded technique components can be combined to form a complete interaction technique for the top-level task.*

design spaces (Card et al. 1990) and metaphor-based classifications (Poupyrev, Weghorst et al. 1997). The main goal of all of these types is to organize a particular set of objects so that they can be thought about systematically. Here we focus on a specific type of taxonomy—the technique-decomposition taxonomy.

The concept of technique decomposition is that each interaction task can be partitioned (decomposed) into subtasks. Similarly, we can decompose the techniques for a particular task into subtechniques, which we call *technique components* (Bowman and Hodges 1999). Each technique component addresses a single subtask (Figure 11.1). We can think of each subtask as a question that must be answered by the designer of an interaction technique, and the set of technique components for a subtask as the set of possible answers for that question.

The set of technique components for each subtask may be built in two ways. First, we can decompose existing techniques and list the components for each subtask. Second, we can think of original technique components that could be used to accomplish each subtask in the taxonomy.

Such technique-decomposition taxonomies have several advantages. Most relevant to the topic of this chapter, the taxonomy can be used as a guide for the evaluation of techniques. In other words, we can perform summative evaluations (Hix and Hartson 1993) that compare technique components rather than holistic techniques. This means that the results of our evaluation will be more precise—we will be able to claim, for example, that object-manipulation techniques that use the virtual object as the center of rotation are more precise than those that use the virtual hand as the center of rotation. Of course, this increased precision comes with a cost—we must perform more complex and time-consuming evaluations. The taxonomy can also be used to design new techniques. For example, the four shaded components in Figure 11.1 could be combined to create a complete interaction technique.

Prototyping

In order to perform a usability evaluation, there must be something to evaluate. In some cases, the full-fledged, final application is available to be evaluated, but more often, evaluation is (or should be) performed earlier in the design cycle so that most problems can be caught early. Thus, many evaluations use some form of prototype.

Prototypes are generally classified based on their level of *fidelity*—that is, how closely the prototype resembles and acts like the final product. Somewhat surprisingly, a great deal of useful usability information can be gleaned from the evaluation of a low-fidelity prototype such as a paper-based sketch, a storyboard, or a static mockup of the interface. In general, the fidelity of the prototype will increase with each successive evaluation.

One important prototyping method for 3D UIs is the so-called Wizard of Oz (WOZ) approach. A WOZ prototype appears to have a large amount of functionality, even though that functionality is not actually present. A human controls the prototype (like the wizard behind the curtain), making it appear more intelligent or high-fidelity than it actually is. For 3D UIs, this prototyping method can be quite useful because the actual implementation of many 3D interaction techniques and UI metaphors can be very complex. For example, one may not want to go to the trouble of implementing a full-fledged speech interface if it is only one of the options being considered. By developing a simple keyboard-based interface, an evaluator can mimic the actions that would be taken by the system when a user speaks a particular word or phrase and can thus determine the usability characteristics of the actual speech interface.

For more detailed information on prototyping in general, see Hix and Hartson (1993).

11.2.2. Evaluation Methods Used for 3D Interfaces

From the literature, we have compiled a list of usability evaluation methods that have been applied to 3D UIs (although numerous references could be cited for some of the techniques we present, we have included citations that are most recognized and accessible). Most of these methods were developed for 2D or GUI usability evaluation and have been subsequently extended to support 3D UI evaluation.

Cognitive Walkthrough

The *cognitive walkthrough* (Polson et al. 1992) is an approach to evaluating a UI based on stepping through common tasks that a user would perform and evaluating the interface's ability to support each step. This approach is intended especially to gain an understanding of the usability of a system for first-time or infrequent users, that is, for users in an exploratory learning mode. Steed and Tromp (1998) have used a cognitive walkthrough approach to evaluate a collaborative VE.

Heuristic Evaluation

Heuristic or *guidelines-based expert evaluation* (Nielsen and Molich 1992) is a method in which several usability experts separately evaluate a UI design (probably a prototype) by applying a set of heuristics or design guidelines, that are either general enough to apply to any UI or are tailored for 3D UIs in particular. No representative users are involved. Results from the several experts are then combined and ranked to prioritize iterative design or redesign of each usability issue discovered. The current lack of well-formed guidelines and heuristics for 3D UI design and evaluation make this approach more challenging for 3D UIs. Examples of this approach applied to 3D UIs can be found in Gabbard, Hix, and Swan (1999); Stanney and Reeves (2000); and Steed and Tromp (1998).

Formative Evaluation

Formative evaluation (both formal and informal; Hix and Hartson 1993); is an observational, empirical evaluation method, applied during evolving stages of design, that assesses user interaction by iteratively placing representative users in task-based scenarios in order to identify usability problems, as well as to assess the design's ability to support user exploration, learning, and task performance. Formative evaluations can range from being rather informal, providing mostly qualitative results such as critical incidents, user comments, and general reactions, to being very formal and extensive, producing both qualitative and quantitative (task timing, errors, etc.) results.

Collected data are analyzed to identify UI components that both support and detract from user task performance and user satisfaction. Alternating between formative evaluation and design or redesign efforts ultimately leads to an iteratively refined UI design. Most usability evaluations of 3D UIs fall into the formative evaluation category. The work of Hix and her colleagues (1999) provides a good example.

Summative Evaluation

Summative or *comparative evaluation* (both formal and informal; Hix and Hartson 1993; Scriven 1967) is an evaluation and statistical comparison of two or more configurations of UI designs, UI components, and/or UI techniques. As with formative evaluation, representative users perform task scenarios as evaluators collect both qualitative and quantitative data. As with formative evaluations, summative evaluations can be formally or informally applied.

Summative evaluation is generally performed after UI designs (or components) are complete and as a traditional factorial experimental design with multiple independent variables. Summative evaluation enables evaluators to measure and subsequently compare the productivity and cost benefits associated with different UI designs. Comparing 3D UIs requires a consistent set of user task scenarios (borrowed and/or refined from the formative evaluation effort), resulting in primarily quantitative results that compare (on a task-by-task basis) a design's support for specific user task performance.

Many of the formal experiments discussed in Part III of this book are summative evaluations of 3D interaction techniques. For example, see Bowman, Johnson and Hodges (1999) and Poupyrev, Weghorst, and colleagues (1997).

Questionnaires

A *questionnaire* (Hix and Hartson 1993) is a written set of questions used to obtain information from users before or after they have participated in a usability evaluation session. Questionnaires are good for collecting demographic information (e.g., age, gender, computer experience) and subjective data (e.g., opinions, comments, preferences, ratings) and are often more convenient and more consistent than spoken interviews.

In the context of 3D UIs, questionnaires are used quite frequently, especially to elicit information about subjective phenomena such as presence (Witmer and Singer 1998) or simulator sickness/cybersickness (Kennedy et al. 1993).

Interviews and Demos

The *interview* (Hix and Hartson 1993) is a technique for gathering information about users by talking directly to them. An interview can gather more information than a questionnaire and may go to a deeper level of detail. Interviews are good for getting subjective reactions, opinions, and insights into how people reason about issues. *Structured* interviews have a predefined set of questions and responses. *Open-ended* interviews permit the respondent (interviewee) to provide additional information, and they permit the interviewer to ask broad questions without a fixed set of answers and explore paths of questioning that may occur to him spontaneously during the interview. Demonstrations (typically of a prototype) may be used in conjunction with user interviews to aid a user in talking about the interface.

In 3D UI evaluation, the use of interviews has not been studied explicitly, but informal interviews are often used at the end of formative or summative usability evaluations (e.g., Bowman and Hodges 1997).

11.3. Evaluation Metrics for 3D Interfaces

Now we turn to metrics. That is, how do we measure the characteristics of a 3D UI when evaluating it? We focus on the general metric of usability. A 3D UI is usable when the user can reach her goals; when the important tasks can be done better, easier, or faster than with another system; and when users are not frustrated or uncomfortable. Note that all of these have to do with the user.

We discuss three types of metrics for 3D UIs: system performance metrics, task performance metrics, and user preference metrics.

11.3.1. System Performance Metrics

System performance refers to typical computer or graphics system performance, using metrics such as average frame rate, average latency, network delay, and optical distortion. From the interface point of view, system performance metrics are really not important in and of themselves. Rather, they are important only insofar as they affect the user's experience or tasks. For example, the frame rate probably needs to be at real-time levels before a user will feel present. Also, in a collaborative setting, task performance will likely be negatively affected if there is too much network delay.

11.3.2. Task Performance Metrics

User task performance refers to the quality of performance of specific tasks in the 3D application, such as the time to navigate to a specific location, the accuracy of object placement, or the number of errors a user makes in selecting an object from a set. Task performance metrics may also be domain-specific. For example, evaluators may want to measure student learning in an educational application or spatial awareness in a military training VE.

Typically, speed (efficiency) and accuracy are the most important task performance metrics. The problem with measuring both speed and accuracy is that there is an implicit relationship between them: *I can go faster but be less accurate,* or *I can increase my accuracy by decreasing my speed.* It is assumed that for every task, there is some curve representing this speed/accuracy tradeoff, and users must decide where on the curve they want to be (even if they don't do this consciously). In an evaluation, therefore, if you simply tell your subjects to do a task as quickly and precisely as possible, they will probably end up all over the curve, giving you data with a high level of variability. Therefore, it is very important that you instruct users in a very specific way if you want them to be at one end of the curve or the other. Another way to manage the tradeoff is to tell users to do the task as quickly as possible one time, as accurately as possible the second time, and to balance speed and accuracy the third time. This gives you information about the tradeoff curve for the particular task you're looking at.

11.3.3. User Preference Metrics

User preference refers to the subjective perception of the interface by the user (perceived ease of use, ease of learning, satisfaction, etc.). These preferences are often measured via questionnaires or interviews and may be either qualitative or quantitative. The user preference metrics generally contribute significantly to overall usability. A usable application is one whose interface does not pose any significant barriers to task completion. Often, HCI experts speak of a *transparent* interface—a UI that simply disappears until it feels to the user as if he is working directly on the problem rather than indirectly through an interface. UIs should be intuitive, provide good affordances (indications of their use and how they are to be used), provide good feedback, not be obtrusive, and so on. An application cannot be effective unless users are willing to use it (and this is precisely the problem with some more advanced VE applications—they provide functionality for the user to do a task, but a lack of attention to user preference keeps them from being used).

For 3D UIs in particular, *presence* and *user comfort* can be important metrics that are not usually considered in traditional UI evaluation. Presence is a crucial, but not very well understood metric for VE systems. It is the "feeling of being there"—existing in the virtual world rather than in the physical world. How can we measure presence? One method simply asks users to rate their feeling of being there on a 1 to 100 scale. Questionnaires can also be used and can contain a wide variety of questions, all designed to get at different aspects of presence. Psychophysical measures are used in controlled experiments where stimuli are manipulated and then correlated to users' ratings of presence (for example, how does the rating change when the environment is presented in mono versus stereo modes?). There are also some more objective measures. Some are physiological (how the body responds to the VE). Others might look at users' reactions to events in the VE (e.g., does the user duck when he's about to hit a virtual beam?). Tests of memory for the environment and the objects within it might give an indirect measurement of the level of presence. Finally, if we know a task for which presence is required, we can measure users' performance on that task and infer the level of presence. There is still a great deal of debate about the definition of presence, the best ways to measure presence, and the importance of presence as a metric (e.g., Usoh et al. 2000; Witmer and Singer 1998).

The other novel user preference metric for 3D systems is user comfort. This includes several different things. The most notable and well studied is so-called *simulator sickness* (because it was first noted in flight simulators). This is symptomatically similar to motion sickness and may result from mismatches in sensory information (e.g., your eyes tell your brain that you are moving, but your vestibular system tells your brain that you are not moving). There is also work on the physical aftereffects of being exposed to 3D systems. For example, if a VE misregisters the virtual hand and the real hand (they're not at the same physical location), the user may have trouble doing precise manipulation in the real world after exposure to the virtual world. More seriously, activities like driving or walking may be impaired after extremely long exposures (1 hour or more). Finally, there are simple strains on arms/hands/eyes from the use of 3D devices. User comfort is also usually measured subjectively, using rating scales or questionnaires. The most famous questionnaire is the simulator sickness questionnaire (SSQ) developed by Kennedy and his colleagues (1993). Researchers have used some objective measures in the study of aftereffects—for example by measuring the accuracy of a manipulation task in the real world after exposure to a virtual world (Wann and Mon-Williams 2002).

11.4. Distinctive Characteristics of 3D Interface Evaluation

The approaches we discuss below for usability evaluation of 3D UIs have been developed and used in response to perceived differences between the evaluation of 3D UIs and the evaluation of traditional UIs such as GUIs. Many of the fundamental concepts and goals are similar, but use of these approaches in the context of 3D UIs is distinct. Here, we present some of the issues that differentiate 3D UI usability evaluation, organized into several categories. The categories contain overlapping considerations but provide a rough partitioning of these important issues. Note that many of these issues are not necessarily found in the literature, but instead come from personal experience and extensive discussions with colleagues.

11.4.1. Physical Environment Issues

One of the most obvious differences between 3D UIs and traditional UIs is the *physical* environment in which that interface is used. In many 3D UIs, nontraditional input and output devices are used, which can preclude the use of some types of evaluation. Users may be standing rather than sitting, and they may be moving about a large space, using whole-body movements. These properties give rise to several issues for usability evaluation. Following are some examples:

- In interfaces using non-see-through HMDs, the user cannot see the surrounding physical world. Therefore, the evaluator must ensure that the user will not bump into walls or other physical objects, trip over cables, or move outside the range of the tracking device (Viirre 1994). A related problem in surround-screen VEs (such as the CAVE) is that the physical walls can be difficult to see because of projected graphics. Problems of this sort could contaminate the results of a usability evaluation (e.g., if the user trips while in the midst of a timed task) and more importantly could cause injury to the user. To mitigate risk, the evaluator can ensure that cables are bundled and will not get in the way of the user (e.g., cables may descend from above). Also, the user may be placed in a physical enclosure that limits movement to areas where there are no physical objects to interfere.

- Many 3D displays do not allow multiple simultaneous viewers (e.g., user and evaluator), so equipment must be set up so that an evaluator can see the same image as the user. With an HMD, for example, this can be done by splitting the video signal and sending it to both the HMD and a monitor. In a surround-screen or workbench VE, a monoscopic view of the scene could be rendered to a monitor, or, if performance will not be adversely affected, both the user and the evaluator can be tracked (this can cause other problems, however; see section 11.4.2 on evaluator considerations). If images are viewed on a monitor, then it is difficult to see both the actions of the user and the graphical environment at the same time, meaning that multiple evaluators may be necessary to observe and collect data during an evaluation session.
- A common and very effective technique for generating important qualitative data during usability evaluation sessions is the "think-aloud" protocol (as described in Hix and Hartson [1993]). With this technique, subjects talk about their actions, goals, and thoughts regarding the interface while they are performing specific tasks. In some 3D UIs, however, voice recognition is used as an interaction technique, making the think-aloud protocol much more difficult and perhaps even impossible. Post-session interviews may help to recover some of the information that would have been obtained from the think-aloud protocol.
- Another common technique involves recording video of both the user and the interface (as described in Hix and Hartson [1993]). Because 3D UI users are often mobile, a single, fixed camera may require a very wide shot, which may not allow precise identification of actions. This could be addressed by using a tracking camera (with, unfortunately, additional expense and complexity) or a camera operator (additional personnel). Moreover, views of the user and the graphical environment must be synchronized so that cause and effect can clearly be seen on the videotape. Finally, recording video of a stereoscopic graphics image can be problematic.
- An ever-increasing number of proposed 3D applications are shared among two or more users (Stiles et al. 1996; Normand

et al. 1999). These collaborative 3D UIs become even more difficult to evaluate than single-user 3D UIs due to physical separation of users (i.e., users are in more than one physical location), the additional information that must be recorded for each user, the unpredictability of network behavior as a factor influencing usability, the possibility that each user will have different devices, and the additional complexity of the system, which may cause more frequent crashes or other problems.

11.4.2. Evaluator Issues

A second set of issues relates to the role of the evaluator in a 3D UI usability evaluation. Because of the complexities and distinctive characteristics of 3D UIs, a usability study may require multiple evaluators, different evaluator roles and behaviors, or both. Following are some examples:

- Many VEs attempt to produce a sense of *presence* in the user—that is, a feeling of actually being in the virtual world rather than the physical one. Evaluators can cause breaks in presence if the user can sense them. In VEs using projected graphics, the user will see an evaluator if the evaluator moves into the user's field of view. This is especially likely in a CAVE environment (Cruz-Neira et al. 1993) where it is difficult to see the front of a user (e.g., their facial expressions and detailed use of handheld devices) without affecting that user's sense of presence. This may break presence, because the evaluator is not part of the virtual world. In any type of VE, touching or talking to the user can cause such breaks. If the evaluation is assessing presence, or if presence is hypothesized to affect performance on the task being evaluated, then the evaluator must take care to remain unsensed during the evaluation.
- When breaks in presence are deemed very important for a particular VE, an evaluator may not wish to intervene at all during an evaluation session. This means that the experimental application/interface must be robust and bug-free so that the session does not have to be interrupted to fix a problem. Also, instructions given to the user must be very detailed, explicit, and precise, and the evaluator should make sure the user has a complete understanding of the procedure and tasks before beginning the session.

- 3D UI hardware and software are often more complex and less robust than traditional UI hardware and software. Again, multiple evaluators may be needed to do tasks such as helping the user with display and input hardware, running the software that produces graphics and other output, recording data such as timings and errors, and recording critical incidents and other qualitative observations of a user's actions.
- Traditional UIs typically require only a discrete, single stream of input (e.g., from mouse and keyboard), but many 3D UIs include multimodal input, combining discrete events, gestures, voice, and/or whole-body motion. It is much more difficult for an evaluator to process these multiple input streams simultaneously and record an accurate log of the user's actions. These challenges make multiple evaluators and video even more important.

11.4.3. User Issues

There are also a large number of issues related to the user population used as subjects in 3D UI usability evaluations. In traditional evaluations, subjects are gleaned from the target user population of an application or from a similar representative group of people. Efforts are often made, for example, to preserve gender equity, to have a good distribution of ages, and to test both experts and novices if these differences are representative of the target user population. The nature of 3D UI evaluation, however, does not always allow for such straightforward selection of users. Following are some examples:

- 3D UIs are still often a "solution looking for a problem." Because of this, the target user population for a 3D application or interaction technique to be evaluated may not be known or well understood. For example, a study comparing two virtual travel techniques is not aimed at a particular set of users. Thus, it may be difficult to generalize performance results. The best course of action is to evaluate the most diverse user population possible in terms of age, gender, technical ability, physical characteristics, and so on, and to include these factors in any models of performance.
- It may be impossible to differentiate between novice and expert users because there are very few potential subjects who could be considered experts in 3D UIs. Most users who could be considered experts might be, for example, research staff, whose participation

in an evaluation could confound the results. Also, because most users are typically novices, the evaluation itself may need to be framed at a lower cognitive and physical level. Evaluators can make no assumptions about a novice user's ability to understand or use a given interaction technique or device.

- Because 3D UIs will be novel to many potential subjects, the results of an evaluation may exhibit high variability and differences among individuals. This means that the number of subjects needed to obtain a good picture of performance may be larger than for traditional usability evaluations. If statistically significant results are required (depending on the type of usability evaluation being performed), the number of subjects may be even greater.
- Researchers are still studying a large design space for 3D interaction techniques and devices. Because of this, evaluations often compare two or more techniques, devices, or combinations of the two. To perform such evaluations using a within-subjects design, users must be able to adapt to a wide variety of situations. If a between-subjects design is used, a larger number of subjects will again be needed.
- VE evaluations must consider the effects of cybersickness and fatigue on subjects. Although some of the causes of cybersickness are known, there are still no predictive models for it (Kennedy et al. 2000), and little is known regarding acceptable exposure time to VEs. For evaluations, then, a worst-case assumption must be made. A lengthy experiment (anything over 30 minutes, for example, might be considered lengthy, depending on the specific VE) must contain planned rest breaks and contingency plans in case of ill or fatigued subjects. Shortening the experiment is often not an option, especially if statistically significant results are needed.
- Because it is not known exactly what VE situations cause sickness or fatigue, most VE evaluations should include some measurement (e.g., subjective, questionnaire-based [Kennedy et al. 2000], or physiological) of these factors. A result indicating that an interaction technique was 50% faster than any other evaluated technique would be severely misleading if that interaction technique also made 30% of subjects sick! Thus, user comfort measurements should be included in low-level VE evaluations.

- Presence is another example of a measure often required in VE evaluations that has no analogue in traditional UI evaluation. VE evaluations must often take into account subjective reports of perceived presence, perceived fidelity of the virtual world, and so on. Questionnaires (Usoh et al. 2000; Witmer and Singer 1998) have been developed that purportedly obtain reliable and consistent measurements of such factors.

11.4.4. Evaluation Type Issues

Traditional usability evaluation can take many forms. These include informal user studies, formal experiments, task-based usability studies, heuristic evaluations, and the use of predictive models of performance (see section 11.3 for further discussion of these types of evaluations). There are several issues related to the use of various types of usability evaluation in 3D UIs. Following are some examples:

- Evaluations based solely on heuristics (i.e., design guidelines), performed by usability experts, are very difficult in 3D UIs because of a lack of published, verified guidelines for 3D UI design. There are some notable exceptions (Bowman 2002; Conkar et al. 1999; Gabbard 1997; Kaur 1999; Kaur et al. 1999; Mills and Noyes 1999; Stanney and Reeves 2000), but for the most part, it is difficult to predict the usability of a 3D interface without studying real users attempting representative tasks in the 3D UI. It is not likely that a large number of heuristics will appear, at least not until 3D input and output devices become more standardized. Even assuming standardized devices, however, the design space for 3D interaction techniques and interfaces is very large, making it difficult to produce effective and general heuristics to use as the basis for evaluation.
- Another major type of usability evaluation that does not employ users is the application of performance models (e.g., GOMS, Fitts's law). Very few models of this type have been developed for or adapted to 3D UIs. However, the lower cost of both heuristic evaluation and performance model application makes them attractive for evaluation.
- Because of the complexity and novelty of 3D UIs, the applicability or utility of automated, tool-based evaluation may be greater than it is for more traditional UIs. For example, several issues

above have noted the need for more than one evaluator in a 3D UI usability evaluation session. Automated usability evaluations could reduce the need for several evaluators in a single session. There are at least two possibilities for automated usability evaluation of 3D UIs: first, to automatically collect and/or analyze data generated by one or more users in a 3D UI, and second, to perform an analysis of an interface design using an interactive tool that embodies design guidelines (similar to heuristics). Some work has been done on automatic collection and analysis of data using specific types of repeating patterns in users' data as indicators of potential usability problems (e.g., Siochi and Hix 1991). However this work was performed on a typical GUI, and there appears to be no research yet conducted that studies automated data collection and evaluation of users' data in 3D UIs. Thus, differences in the kinds of data for 3D UI usability evaluation have not been explored, but they would involve, at a minimum, collating data from multiple users in a single session, possibly at different physical locations and even in different parts of the 3D environment. At least one tool, MAUVE (Multi-Attribute Usability evaluation tool for Virtual Environments) incorporates design guidelines organized around several VE categories: navigation, object manipulation, input, output (e.g., visual, auditory, haptic), and so on (Stanney et al. 2000). Within each of these categories, MAUVE presents a series of questions to an evaluator, who uses the tool to perform a multicriteria, heuristic-style evaluation of a specific 3D UI.

- When performing statistical experiments to quantify and compare the usability of various 3D interaction techniques, input devices, interface elements, and so on, it is often difficult to know which factors have a potential impact on the results. Besides the primary independent variable (e.g., a specific interaction technique), a large number of other potential factors could be included, such as environment, task, system, or user characteristics. One approach is to try to vary as many of these potentially important factors as possible during a single experiment. This "testbed evaluation" approach (Bowman, Johnson et al. 1999; Snow and Williges 1998) has been used with some success (see section 11.6.1). The other extreme would be to simply hold as many of these other factors as possible constant and evaluate only in a particular set of circumstances. Thus, statistical 3D UI experimental evaluations may be either overly simplistic or overly complex—finding the proper balance is difficult.

11.4.5. Miscellaneous Issues

- 3D UI usability evaluations generally focus at a lower level than traditional UI evaluations. In the context of GUIs, a standard look and feel and a standard set of interface elements and interaction techniques exist, so evaluation usually looks at subtle interface nuances or overall interface metaphors. In 3D UIs, however, there are no interface standards, and there is not even a good understanding of the usability of various interface types. Therefore, 3D UI evaluations most often compare lower-level components, such as interaction techniques or input devices.
- It is tempting to overgeneralize the results of evaluations of 3D interaction performed in a generic (nonapplication) context. However, because of the fast-changing and complex nature of 3D UIs, one cannot assume anything (display type, input devices, graphics processing power, tracker accuracy, etc.) about the characteristics of a real 3D application. Everything has the potential to change. Therefore, it is important to include information about the environment in which the evaluation was performed and to evaluate in a range of environments (e.g., using different devices) if possible.

11.5. Classification of 3D Evaluation Methods

A classification space for 3D UI usability evaluation methods can provide a structured means for comparing evaluation methods. One such space classifies methods according to three key characteristics: *involvement of representative users*, *context of evaluation*, and *types of results produced* (Figure 11.2).

The first characteristic discriminates between those methods that *require* the participation of representative users (to provide design or use-based experiences and options) and those methods that do not (methods not requiring users still require a usability expert). The second characteristic describes the type of context in which the evaluation takes place. In particular, this characteristic identifies those methods that are applied in a generic context and those that are applied in an application-specific

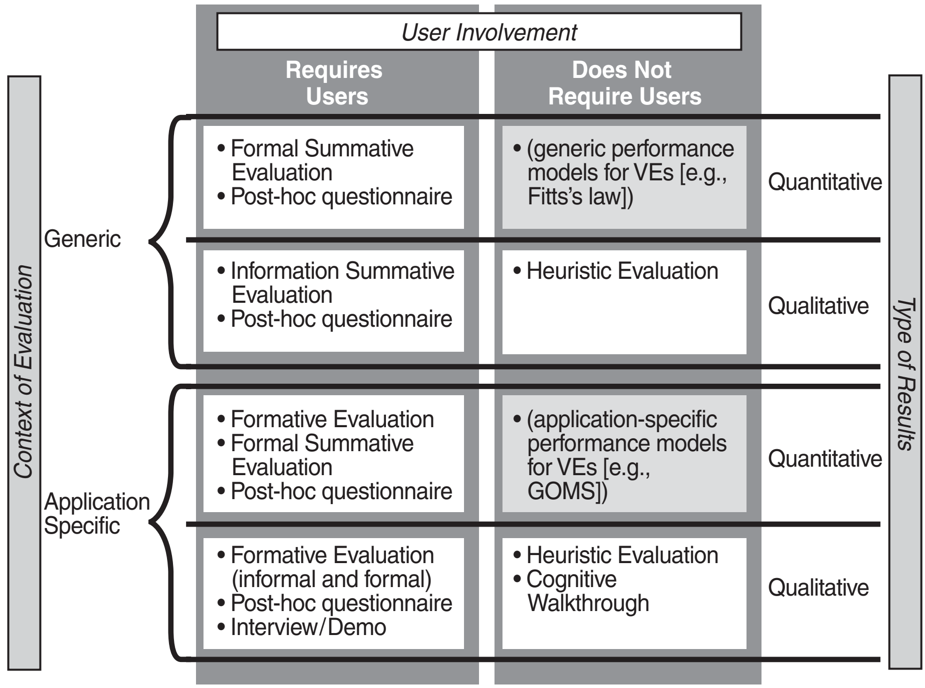


Figure 11.2 *A classification of usability evaluation methods for 3D UIs. (Image reprinted by permission of MIT Press and Presence: Teleoperators and Virtual Environments)*

context. The context of evaluation inherently imposes restrictions on the applicability and generality of results. Thus, conclusions or results of evaluations conducted in a generic context can typically be applied more broadly (i.e., to more types of interfaces) than results of an application-specific evaluation method, which may be best suited for applications that are similar in nature. The third characteristic identifies whether or not a given usability evaluation method produces (primarily) qualitative or quantitative results.

Note that the characteristics described above are not designed to be mutually exclusive, and are instead designed to convey one (of many) usability evaluation method characteristics. For example, a particular usability evaluation method may produce both quantitative and qualitative results. Indeed, many of the identified methods are flexible enough to provide insight at many levels. These three characteristics were chosen (over other potential characteristics) because they are often the most significant (to evaluators) because of their overall effect on the usability process. That is, a researcher interested in undertaking usability evaluation will likely need to know what the evaluation will cost, what the impact of the evaluation will be, and how the results can be applied. Each of the three characteristics addresses these concerns: degree of user involvement directly affects the cost to proctor and analyze the evaluation; results of the process indicate what type of information will be produced (for the given cost); and context of evaluation inherently dictates to what extent results may be applied.

This classification is useful on several levels. It structures the space of evaluation methods and provides a practical vocabulary for discussion of methods in the research community. It also allows researchers to compare two or more methods and understand how they are similar or different on a fundamental level. Finally, it reveals "holes" in the space (Card et al. 1990)—combinations of the three characteristics that have rarely or never been tried in the 3D UI community.

Figure 11.2 shows that there are two such holes in this space (the shaded boxes). More specifically, there is a lack of current 3D UI usability evaluation methods that do not require users and that can be applied in a generic context to produce quantitative results (upper right of the figure). Note that some possible existing 2D and GUI evaluation methods are listed in parentheses, but few, if any, of these methods have been applied to 3D UIs. Similarly, there appears to be no method that provides quantitative results in an application-specific setting that does not require users (third box down on the right of the figure). These areas may be interesting avenues for further research.

11.6. Two Multimethod Approaches

A shortcoming of the classification discussed in section 11.5 is that it does not convey "when" in the software development lifecycle a method is best applied or "how" several methods may be applied. In most cases, answers to these questions cannot be determined without a comprehensive understanding of each of the methods presented, as well as the specific goals and circumstances of the 3D UI research or development effort. In this section, we present two well-developed 3D UI evaluation approaches and compare them in terms of practical usage and results.

11.6.1. Testbed Evaluation Approach

Bowman and Hodges (1999) take the approach of empirically evaluating interaction techniques outside the context of applications (i.e., within a generic context rather than within a specific application) and add the support of a framework for design and evaluation, which we summarize here. Principled, systematic design and evaluation frameworks give formalism and structure to research on interaction; they do not rely solely on experience and intuition. Formal frameworks provide us not only with a greater understanding of the advantages and disadvantages of current techniques, but also with better opportunities to create robust and well-performing new techniques based on knowledge gained through evaluation. Therefore, this approach follows several important evaluation concepts, elucidated in the following sections. Figure 11.3 presents an overview of this approach.

Initial Evaluation

The first step toward formalizing the design, evaluation, and application of interaction techniques is to gain an intuitive understanding of the generic interaction tasks in which one is interested and current techniques available for the tasks (see Figure 11.3, area labeled 1). This is accomplished through experience using interaction techniques and through observation and evaluation of groups of users. These initial evaluation experiences are heavily drawn upon for the processes of building a taxonomy, listing outside influences on performance, and listing performance measures. It is helpful, therefore, to gain as much experience of this type as possible so that good decisions can be made in the next phases of formalization.

Taxonomy

The next step is to establish a taxonomy (Figure 11.3, area 2) of interaction techniques for the interaction task being evaluated. These are technique-decomposition taxonomies, as described in section 11.2.1. For example, the task of changing an object's color might be made up of three subtasks: selecting an object, choosing a color, and applying the color. The subtask for choosing a color might have two possible technique components: changing the values of R, G, and B sliders or touching a point within a 3D color space. The subtasks and their related technique components make up a taxonomy for the object coloring task.

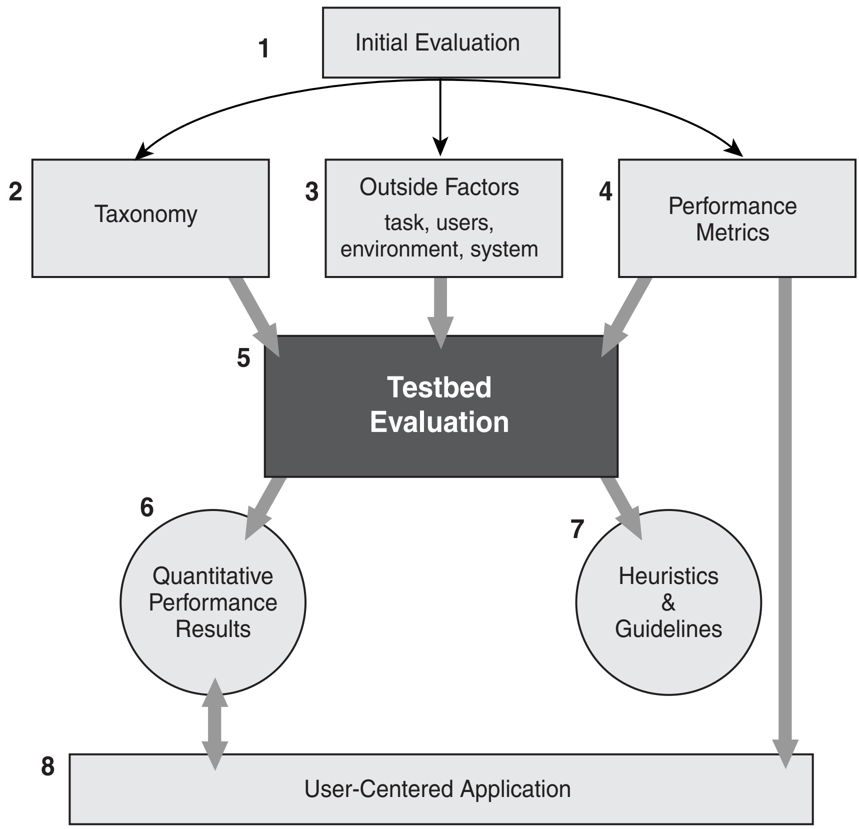


Figure 11.3 *Testbed evaluation approach. (Image reprinted by permission of MIT Press and Presence: Teleoperators and Virtual Environments)*

Ideally, the taxonomies established by this approach need to be correct, complete, and general. Any interaction technique that can be conceived for the task should fit within the taxonomy. Thus, subtasks will necessarily be abstract. The taxonomy will also list several possible technique components for each of the subtasks, but they do not list every conceivable component.

Building taxonomies is a good way to understand the low-level makeup of interaction techniques and to formalize differences between them, but once they are in place, they can also be used in the design process. One can think of a taxonomy not only as a characterization, but also as a design space. Because a taxonomy breaks the task down into separable subtasks, a wide range of designs can be considered quickly, simply by trying different combinations of technique components for each of the subtasks. There is no guarantee that a given combination will make sense as a complete interaction technique, but the systematic nature of the taxonomy makes it easy to generate designs and to reject inappropriate combinations.

Outside Factors

Interaction techniques cannot be evaluated in a vacuum. A user's performance on an interaction task may depend on a variety of factors (Figure 11.3, area 3), of which the interaction technique is but one. In order for the evaluation framework to be complete, such factors must be included explicitly and used as secondary independent variables in evaluations. Bowman and Hodges (1999) identified four categories of outside factors.

First, task characteristics are those attributes of the task that may affect user performance, including distance to be traveled or size of the object being manipulated. Second, the approach considers environment characteristics, such as the number of obstacles and the level of activity or motion in the 3D scene. User characteristics, including cognitive measures such as spatial ability and physical attributes such as arm length, may also contribute to user performance. Finally, system characteristics, such as the lighting model used or the mean frame rate, may be significant.

Performance Metrics

This approach is designed to obtain information about human performance in common 3D interaction tasks—but what is performance? Speed and accuracy are easy to measure, are quantitative, and are clearly important in the evaluation of interaction techniques, but there are also many other performance metrics (Figure 11.3, area 4) to be considered. Thus, this approach also considers more subjective performance values, such as perceived ease of use, ease of learning, and user comfort. The choice of interaction technique could conceivably affect all of these, and they should not be discounted. Also, more than any other current computing paradigm, 3D UIs involve the user's senses and body in the task. Thus, a focus on user-centric performance measures is essential. If an interaction technique does not make good use of human skills, or if it causes fatigue or discomfort, it will not provide overall usability, despite its performance in other areas.

Testbed Evaluation

Bowman and Hodges (1999) use testbed evaluation (Figure 11.3, area 5) as the final stage in the evaluation of interaction techniques for 3D interaction tasks. This approach allows generic, generalizable, and reusable evaluation through the creation of testbeds—environments and tasks that involve all important aspects of a task, that evaluate each component of a technique, that consider outside influences (factors other than the interaction technique) on performance, and that have multiple performance measures. A testbed experiment uses a formal, factorial, experimental design and normally requires a large number of subjects. If many interaction techniques or outside factors are included in the evaluation, the number of trials per subject can become overly large, so interaction techniques are usually a between-subjects variable (each subject uses only a single interaction technique), while other factors are within-subjects variables. See the case studies below for examples of testbed experiments.

Application and Generalization of Results

Testbed evaluation produces a set of results or models (Figure 11.3, area 6) that characterize the usability of an interaction technique for the specified task. Usability is given in terms of multiple performance metrics with respect to various levels of outside factors. These results become part of a performance database for the interaction task, with more information being added to the database each time a new technique is run through the testbed. These results can also be generalized into heuristics or guidelines (Figure 11.3, area 7) that can easily be evaluated and applied by 3D UI developers.

The last step is to apply the performance results to 3D applications (Figure 11.3, area 8) with the goal of making them more useful and usable. In order to choose interaction techniques for applications appropriately, one must understand the interaction requirements of the application. There is no single “best” technique, because the technique that is best for one application may not be optimal for another application with different requirements. Therefore, applications need to specify their interaction requirements before the most appropriate interaction techniques can be chosen. This specification is done in terms of the performance metrics that have already been defined as part of the formal framework. Once the requirements are in place, the performance results from testbed evaluation can be used to recommend interaction techniques that meet those requirements.

Case Studies

Although testbed evaluation could be applied to almost any type of interactive system, it is especially appropriate for 3D UIs because of its focus on low-level interaction techniques. Testbed experiments have been performed comparing techniques for the tasks of travel (Bowman, Davis et al. 1999) and selection/manipulation (Bowman and Hodges 1999).

The travel testbed experiment compared seven different travel techniques for the tasks of naïve search and primed search. In the primed search trials, the initial visibility of the target and the required accuracy of movement were also varied. The dependent variables were time for task completion and subjective user comfort ratings. Forty-four subjects participated in the experiment. The researchers gathered both demographic and spatial ability information for each subject.

The selection/manipulation testbed compared the usability and performance of nine different interaction techniques. For selection tasks, the independent variables were distance from the user to the object, size of the object, and density of distracter objects. For manipulation tasks, the required accuracy of placement, the required degrees of freedom, and the distance through which the object was moved were varied. The dependent variables in this experiment were the time for task completion, the number of selection errors, and subjective user comfort ratings. Forty-eight subjects participated, and the researchers again obtained demographic data and spatial ability scores.

In both instances, the testbed approach produced unexpected and interesting results that would not have been revealed by a simpler experiment. For example, in the selection/manipulation testbed, it was found that selection techniques using an extended virtual hand performed well with larger, nearer objects and more poorly with smaller, farther objects, while selection techniques based on ray-casting performed well regardless of object size or distance. The testbed environments and tasks have also proved to be reusable. The travel testbed was used to evaluate a new travel technique and compare it to existing techniques, while the manipulation testbed has been used to evaluate the usability of common techniques in the context of different VE display devices.

11.6.2. Sequential Evaluation Approach

Gabbard, Hix, and Swan (1999) present a sequential approach to usability evaluation for specific 3D applications. The sequential evaluation approach is a usability engineering approach and addresses both design and evaluation of 3D UIs. However, for the scope of this chapter, we focus on different types of evaluation and address analysis, design, and prototyping only when they have a direct effect on evaluation.

Although some of its components are well suited for evaluation of generic interaction techniques, the complete sequential evaluation approach employs application-specific guidelines, domain-specific representative users, and application-specific user tasks to produce a usable and useful interface for a particular application. In many cases, results or lessons learned may be applied to other, similar applications (for example, 3D applications with similar display or input devices, or with similar types of tasks). In other cases (albeit less often), it is possible to abstract the results for general use.

Sequential evaluation evolved from iteratively adapting and enhancing existing 2D and GUI usability evaluation methods. In particular, it modifies and extends specific methods to account for complex interaction techniques, nonstandard and dynamic UI components, and multimodal tasks inherent in 3D UIs. Moreover, the adapted/extended methods both streamline the usability engineering process and provide sufficient coverage of the usability space. Although the name implies that the various methods are applied in sequence, there is considerable opportunity to iterate both within a particular method as well as among methods. It is important to note that all the pieces of this approach have been used for years in GUI usability evaluations. The unique contribution of Gabbard, Hix, and Swan's (1999) work is the breadth and depth offered by progressive use of these techniques, adapted when necessary for 3D UI evaluation, in an application-specific context. Further, the way in which each step in the progression informs the next step is an important finding: the ordering of the methods guides developers toward a usable application.

Figure 11.4 presents the sequential evaluation approach. It allows developers to improve a 3D UI by a combination of expert-based and user-based techniques. This approach is based on sequentially performing user task analysis (see Figure 11.4, area labeled 1), heuristic (or guideline-based expert) evaluation (Figure 11.4, area 2), formative evaluation (Figure 11.4, area 3), and summative evaluation (Figure 11.4, area 4), with iteration as appropriate within and among each type of evaluation. This

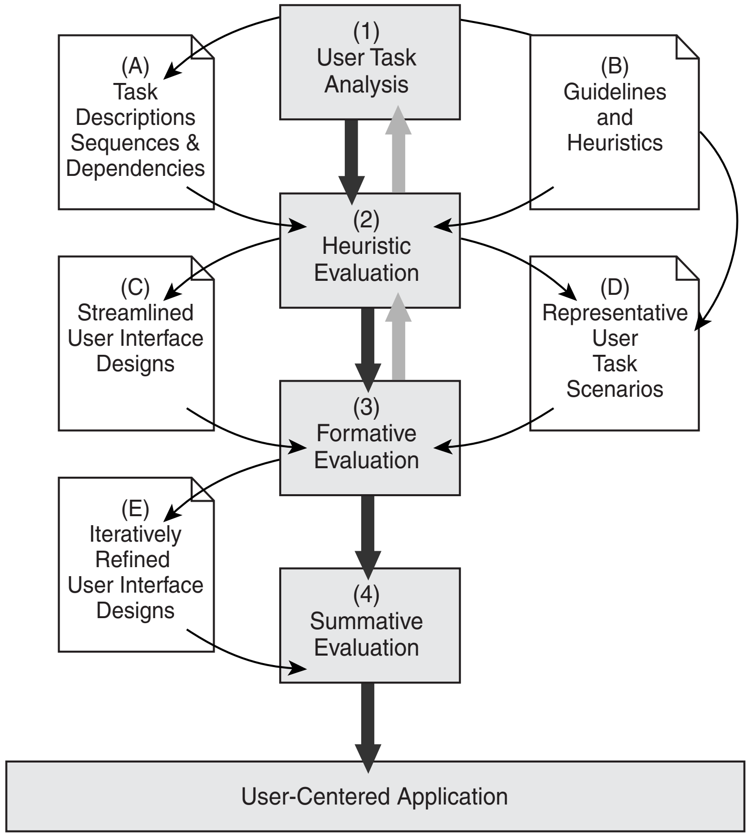


Figure 11.4 *Sequential evaluation approach. (Image reprinted by permission of MIT Press and Presence: Teleoperators and Virtual Environments)*

approach leverages the results of each individual method by systematically defining and refining the 3D UI in a cost-effective progression.

Depending upon the nature of the application, this sequential evaluation approach may be applied in a strictly serial approach (as Figure 11.4's solid black arrows illustrate) or iteratively applied (either as a whole or per-individual method, as Figure 11.4's gray arrows illustrate) many times. For example, when used to evaluate a complex command-and-control battlefield visualization application (Hix et al. 1999), user task analysis was followed by significant iterative use of heuristic and formative evaluation and lastly followed by a single, broad summative evaluation.

From experience, this sequential evaluation approach provides cost-effective assessment and refinement of usability for a specific 3D application. Obviously, the exact cost and benefit of a particular evaluation effort depends largely on the application's complexity and maturity. In some cases, cost can be managed by performing quick and lightweight formative evaluations (which involve users and thus are typically the most time-consuming to plan and perform). Moreover, by using a "hallway methodology," user-based methods can be performed quickly and cost-effectively by simply finding volunteers from within one's own organization. This approach should be used only as a last resort or in cases where the representative user class includes just about anyone. When used, care should be taken to ensure that "hallway" users provide a close representative match to the application's ultimate end users.

The individual methods involved in sequential evaluation are described earlier in the chapter (user task analysis in section 11.2.1 and heuristic, formative, and summative evaluation in section 11.2.2).

Case Studies

The sequential evaluation approach has been applied to several 3D UIs, including the Naval Research Lab's Dragon application: a VE for battlefield visualization (Gabbard et al. 1999). Dragon is presented on a Responsive Workbench that provides a 3D display for observing and managing battlespace information shared among commanders and other battle planners. The researchers performed several evaluations over a nine-month period, using one to three users and two to three evaluators per session. Each evaluation session revealed a set of usability problems and generated a corresponding set of recommendations. The developers would address the recommendations and produce an improved UI for the next iteration of evaluation. The researchers performed four major cycles of iteration during the evaluation of Dragon, each cycle using the progression of usability methods described in this section.

During the expert guideline-based evaluations, various user interaction design experts worked alone or collectively to assess the evolving user interaction design for Dragon. The expert evaluations uncovered several major design problems that are described in detail in Hix et al. (1999). Based on user task analysis and early expert guideline-based evaluations, the researchers created a set of user task scenarios specifically for battlefield visualization. During each formative session, there were at least two and often three evaluators present. Although both the expert guideline-based evaluation sessions and the formative evaluation sessions were personnel-intensive (with two or three evaluators involved), it was found that the quality and amount of data collected by multiple evaluators greatly outweighed the cost of those evaluators.

Finally, the summative evaluation statistically examined the effect of four factors: locomotion metaphor (egocentric versus exocentric), gesture control (controls rate versus controls position), visual presentation device (workbench, desktop, CAVE), and stereopsis (present versus not present). The results of these efforts are described in Hix and Gabbard (2002). This experience with sequential evaluation demonstrated its utility and effectiveness.

11.6.3. Comparison of Approaches

The two major evaluation methods we have presented for 3D UIs—testbed evaluation and sequential evaluation—take quite different approaches to the same problem: how to improve usability in 3D applications. At a high level, these approaches can be characterized in the space defined in section 11.5. Sequential evaluation is done in the context of a particular application and can have both quantitative and qualitative results. Testbed evaluation is done in a generic evaluation context and usually seeks quantitative results. Both approaches employ users in evaluation.

In this section, we take a more detailed look at the similarities of and differences between these two approaches. We organize this comparison by answering several key questions about each of the methods. Many of these questions can be asked of other evaluation methods and perhaps should be asked prior to designing a usability evaluation. Indeed, answers to these questions may help identify appropriate evaluation methods given specific research, design, or development goals. Developers should attempt to find valid answers to these and related questions regarding different usability evaluation methods. Another possibility is to understand the general properties, strengths, and weaknesses of each approach so that the two approaches can be linked in complementary ways.

What Are the Goals of the Approach?

As mentioned above, both approaches ultimately aim to improve usability in 3D applications. However, there are more specific goals that exhibit differences between the two approaches.

Testbed evaluation has the specific goal of finding generic performance characteristics of interaction techniques. This means that one wants to understand interaction technique performance in a high-level, abstract way, not in the context of a particular application. This goal is important because, if achieved, it can lead to wide applicability of the results. In order to do generic evaluation, the testbed approach is limited to general techniques for common, universal tasks (such as navigation, selection, or manipulation). To say this in another way, testbed evaluation is not designed to evaluate special-purpose techniques for specific tasks, such as applying a texture. Rather, it abstracts away from these specifics, using generic properties of the task, user, environment, and system.

Sequential evaluation's immediate goal is to iterate toward a better UI for a particular application, in this case a specific 3D application. It looks very closely at particular user tasks of an application to determine which scenarios and interaction techniques should be incorporated. In general, this approach tends to be quite specific in order to produce the best possible interface design for a particular application under development.

When Should the Approach Be Used?

By its non-application-specific nature, the testbed approach actually falls completely outside the design cycle of a particular application. Ideally, testbed evaluation should be completed before an application is even a glimmer in the eye of a developer. Because it produces general performance/usability results for interaction techniques, these results can be used as a starting point for the design of new 3D UIs.

On the other hand, sequential evaluation should be used early and continually throughout the design cycle of a 3D application. User task analysis is necessary before the first interface prototypes are built. Heuristic and formative evaluations of a prototype produce recommendations that can be applied to subsequent design iterations. Summative evaluations of different design possibilities can be done when the choice of design (e.g., for interaction techniques) is not clear.

The distinct time periods in which testbed evaluation and sequential evaluation are employed suggests that combining the two approaches is possible and even desirable. Testbed evaluation can first produce a set of general results and guidelines that can serve as an advanced and well-informed starting point for a 3D application's UI design. Sequential evaluation can then refine that initial design in a more application-specific fashion.

In What Situations Is the Approach Useful?

Testbed evaluation allows the researcher to understand detailed performance characteristics of common interaction techniques, especially user performance. It provides a wide range of performance data that may be applicable to a variety of situations. In a development effort that requires a suite of applications with common interaction techniques and interface elements, testbed evaluation could provide a quantitative basis for choosing them, because developers could choose interaction techniques that performed well across the range of tasks, environments, and users in the applications; their choices are supported by empirical evidence.

As we have said, the sequential evaluation approach should be used throughout the design cycle of a 3D UI, but it is especially useful in the early stages of interface design. Because sequential evaluation produces results even on very low-fidelity prototypes or design specifications, a 3D application's UI can be refined much earlier, resulting in greater cost savings. Also, the earlier this approach is used in development, the more time remains for producing design iterations, which ultimately results in a better product. This approach also makes the most sense when a user task analysis has been performed. This analysis will suggest task scenarios that make evaluation more meaningful and effective.

What Are the Costs of Using the Approach?

The testbed evaluation approach can be seen as very costly and is definitely not appropriate for every situation. In certain scenarios, however, its benefits can make the extra effort worthwhile. Some of the most important costs associated with testbed evaluation include difficult experimental design (many independent and dependent variables, where some of the combinations of variables are not testable), experiments requiring large numbers of trials to ensure significant results, and large amounts of time spent running experiments because of the number of subjects and trials. Once an experiment has been conducted, the results may not be as detailed as some developers would like. Because testbed evaluation looks at generic situations, information on specific interface details such as labeling, the shape of icons, and so on will not usually be available.

In general, the sequential evaluation approach may be less costly than testbed evaluation because it can focus on a particular 3D application rather than pay the cost of abstraction. However, some important costs are still associated with this method. Multiple evaluators may be needed. Development of useful task scenarios may take a large amount of effort. Conducting the evaluations themselves may be costly in terms of time, depending on the complexity of task scenarios. Most importantly, because this is part of an iterative design effort, time spent by developers to incorporate suggested design changes after each round of evaluation must be considered.

What Are the Benefits of Using the Approach?

Because testbed evaluation is so costly, its benefits must be significant before it becomes a useful evaluation method. One such benefit is generality of the results. Because testbed experiments are conducted in a generalized context, the results may be applied many times in many different types of applications. Of course, there is a cost associated with each use of the results because the developer must decide which results are relevant to a specific 3D UI. Second, testbeds for a particular task may be used multiple times. When a new interaction technique is proposed, that technique can be run through the testbed and compared with techniques already evaluated. The same set of subjects is not necessary, because testbed evaluation usually uses a between-subjects design. Finally, the generality of the experiments lends itself to development of general guidelines and heuristics. It is more difficult to generalize from experience with a single application.

For a particular application, the sequential evaluation approach can be very beneficial. Although it does not produce reusable results or general principles in the same broad sense as testbed evaluation, it is likely to produce a more refined and usable 3D UI than if the results of testbed evaluation were applied alone. Another of the major benefits of this method relates to its involvement of users in the development process. Because members of the representative user group take part in many of the evaluations, the 3D UI is more likely to be tailored to their needs and should result in higher user acceptance and productivity, reduced user errors, and increased user satisfaction. There may be some reuse of results, because other applications may have similar tasks or requirements, or they may be able to use refined interaction techniques produced by the process.

How Are the Approach's Evaluation Results Applied?

The results of testbed evaluation are applicable to any 3D UI that uses the tasks studied with a testbed. Currently, testbed results are available for some of the most common tasks in 3D UIs: travel and selection/manipulation (Bowman, Johnson et al. 2001). The results can be applied in two ways. The first, informal technique is to use the guidelines produced by testbed evaluation in choosing interaction techniques for an application (as in Bowman, Johnson et al. 1999). A more formal technique uses the requirements of the application (specified in terms of the testbed's performance metrics) to choose the interaction technique closest to those requirements. Both of these approaches should produce a set of interaction techniques for the application that makes it more usable than the same application designed using intuition alone. However, because the results are so general, the 3D UI will almost certainly require further refinement.

Application of results of the sequential evaluation approach is much more straightforward. Heuristic and formative evaluations produce specific suggestions for changes to the application's UI or interaction techniques. The result of summative evaluation is an interface or set of interaction techniques that performs the best or is the most usable in a comparative study. In any case, results of the evaluation are tied directly to changes in the interface of the 3D application.

11.7. Guidelines for 3D Interface Evaluation

In this section, we present some guidelines for those wishing to perform usability evaluations of 3D UIs. The first subsection presents general guidelines, and the second subsection focuses specifically on formal experimentation.

11.7.1. General Guidelines

> Begin with informal evaluation.

Informal evaluation is very important, both in the process of developing an application and in doing basic interaction research. In the context of an application, informal evaluation can quickly narrow the design space and point out major flaws in the design. In basic research, informal evaluation helps you understand the task and the techniques on an intuitive level before moving on to more formal classifications and experiments.

> Acknowledge and plan for the differences between traditional UI and 3D UI evaluation.

Section 11.4 detailed a large number of distinctive characteristics of 3D UI evaluation. These differences must be considered when designing a study. For example, you should plan to have multiple evaluators, incorporate rest breaks into your procedure, and assess whether breaks in presence could affect your results.

Choose an evaluation approach that meets your requirements.

Just as we discussed with respect to interaction techniques, there is no optimal usability evaluation method or approach. A range of methods should be considered, and important questions such as those in section 11.6.3 should be asked. For example, if you have designed a new interaction technique and want to refine the usability of the design before any implementation, a heuristic evaluation or cognitive walkthrough fits the bill. On the other hand, if you must choose between two input devices for a task in which a small difference in efficiency may be significant, a formal experiment may be required.

Use a wide range of metrics.

Remember that speed and accuracy alone do not equal usability. Also remember to look at learning, comfort, presence, and other metrics in order to get a complete picture of the usability of the interface.

11.7.2. Guidelines for Formal Experimentation

Design experiments with general applicability.

If you're going to do formal experiments, you will be investing a large amount of time and effort, so you want the results to be as general as possible. Thus, you have to think hard about how to design tasks that are generic, performance measures to which real applications can relate, and a method for applications to easily reuse the results.

> Use pilot studies to determine which variables should be tested in the main experiment.

In doing formal experiments, especially testbed evaluations, you often have too many variables to actually test without an infinite supply of time and subjects. Small pilot studies can show trends that may allow you to remove certain variables because they do not appear to affect the task you're doing.

> Look for interactions between variables—rarely will a single technique be the best in all situations.

In most formal experiments on the usability of 3D UIs, the most interesting results have been interactions. That is, it's rarely the case that technique A is always better than technique B. Rather, technique A works well when the environment has characteristic X, and technique B works well when the environment has characteristic Y. Statistical analysis should reveal these interactions between variables.

Recommended Reading

Many entry-level HCI textbooks, such as the following, provide an excellent introduction to usability evaluation and usability engineering:

Hix, D., and H. Hartson (1993). *Developing User Interfaces: Ensuring Usability Through Product & Process*, John Wiley & Sons.

Rosson, M., and J. Carroll (2001). *Usability Engineering: Scenario-Based Development of Human Computer Interaction*, Morgan Kaufmann Publishers.

Acknowledgment

Much of the content in this chapter comes from a 2002 article by Doug Bowman, Joseph Gabbard, and Deborah Hix that appeared in the journal *Presence*: "A Survey of Usability Evaluation in Virtual Environments: Classification and Comparison of Methods." (*Presence: Teleoperators and Virtual Environments*, 11[4], 404–424).

We thank the coauthors and the MIT Press for their generous permission to reuse the material here. This content is © 2002 MIT Press.

Index

Numbers

A

B

C

D

G

H

J

K

L

M

N

O

P

S

T

U

V

W

X

Y

Z