Yib’s Guide to MOOing

Getting the Most from

Virtual Communities on the Internet

Elizabeth Hess

Table of Contents

Foreword

Yib, the author of this volume, needs little introduction to those who participate in the virtual world of LambdaMOO. She is a familiar figure to both newbies and experienced MOOers as a wizard (administrator) on various MOOs, as a provider of advice and support to those interested in learning about MOOing, as a creative and industrious programmer, and as a witty and wise contributor to the social life of the communities in which she participates. Instead, I will use this foreword to say a few words about MOOs and about the origins of this excellent introduction to them.

For those unfamiliar with these virtual worlds, a MOO is a type of program that permits multiple users (or players or participants), typically from widely dispersed sites, to access a shared database simultaneously, via telnet or a client program, and to communicate and interact synchronically and asynchronically. The environment is characterized by a spatial metaphor and an architectural motif. Because of these features, Pavel Curtis, the founder of LambdaMOO, has described it as "an electronically- represented 'place' that users can visit."

A MOO is a world of words. Participants describe themselves (or, more precisely, the characters they control) and the objects they create, and it is the text of these descriptions that players see when they "look" at one another and at the other objects they encounter within the virtual environment. Users act and interact with one another by typing – the objects they create and manipulate and the messages they send and receive appear as words scrolling down a screen. The users who create, name, and describe objects may also program them to "do" something when an appropriate command is given, each object "responding" to a command implied in its description.

In MOOs, players communicate and interact with one another in various ways. They "talk" by typing a command and a message appears as text on their computer monitors. Players can also depict themselves as gesturing or emoting by typing a command and a message that displays nonverbal actions. To participate even minimally in a virtual world like LambdaMOO, it is necessary to learn the basic commands that enable communication and interaction in it. And to partake more fully of the possible experiences there, a player needs to master a more advanced set of commands.

For many years, I have taught courses about the ways in which people behave online, especially in virtual environments like MOOs, and as part of those courses, I have asked my students to enter into the life of LambdaMOO. They are expected to establish characters there and to learn about the community by becoming participant-observers in it. They are asked to communicate and interact with other players, and, as do other participants, they often form friendships with other MOOers. Some even become avid programmers. For these experiences, the students must learn the basics of MOOing and that is how *Yib's Guide to MOOing* came into existence. At my invitation, Yib began to participate in the online meetings of my classes, responding to students' questions about how to communicate and navigate in

LambdaMOO. She also helped them when they were logged into the community, responding to a barrage of requests for help.

In the summer of 1999, Yib audited my class in person. As Yib puts it, “I taught them about MOOing; they taught me about anthropology.” As questions about MOOing came up, Yib would answer them briefly in class, then, later, she would post a more in-depth answer to a mailing list that the students could read at their leisure. This collection of essays was eagerly read by those taking the courses; it became the students’ MOO bible. The topics ranged from technical questions to social conventions to historical background about some of the major developments in the social and political life of the community. One day I said to her, “You know, you have a book to write,” and she said, “Yes, I do.” The result is *Yib’s Guide to MOOing.*

For both newbies and veteran players, *Yib’s Guide to MOOing* offers the clearest and most comprehensive explanation of the concept of a MOO and overview of MOO commands and customs. It has illuminated the world of MOOing for my students and me, and facilitated our participation in and enjoyment of the life of LambdaMOO. We have all benefited greatly from her contribution. And now the *Guide* is available to a wider audience. I fully expect that others will get as much satisfaction and pleasure from her efforts as we have. Everyone interested in participating in LambdaMOO and in virtual worlds like it owes Yib a debt of gratitude.

Tower
February 25, 2003

Acknowledgements

A MOO is a gift economy, and myriad people have given unstintingly of their time, energy, and talent over the years to make MOOs what they are today. This book is my tribute to those who have made meaningful the phrase *user extensible*.

Any mistakes are solely my own.

The following people and MOO players have been of particular help:

Judy Anderson (yduJ)
Pavel Curtis
David Jacobson
Peter "Euphistopheles, Source Error Supreme" St.John
Chris Stacy

Audrey
Barth
Bartlebooth
Bear(tm)
Bits
Boo
Doci
eep
Klaatu
mockturtle
Nim
Ostrich
Pax
Shmool
Tartan_Guest
Tower
Werebull

The LambdaMOO wizards

B.

Introduction

A MOO is a computer program that enables people to interact with one another in a variety of ways through the use of text. One logs on (as one might log on to any computer system), initiates action with any of a very wide variety of typed commands, and reads the results as text displayed on one's screen. Often, the commands one types cause text to appear on the screens of other people who are using the program at the same time, and, similarly, commands that others type may cause text to be displayed to one's own screen – giving rise to interpersonal interaction.

The text that one sees immediately after logging on usually includes the description of a location, often a room within a building (though not necessarily so). One facet of MOOs is that each one is a textually represented world, such as one might encounter in a play, a novel, or a non-fiction work about a particular time and place. Within this represented world, called the *virtual reality* or *VR*, are many rooms and other places that one can explore using commands such as "`north`", "`out`", and "`enter cottage`", and *objects* (presented as textual descriptions) that one can manipulate using commands such as "`open box`", "`look in box`", and "`take surprise prize from box`". When you type one of these commands, appropriate text concerning the object will appear on your screen. Each person logged on to the MOO also appears as an object represented within the MOO's frame story or *theme*. Many of the available commands are used to depict one's represented self as saying or doing various things within the represented scene. It's something like improvisatory theater, except that you type what you would say or do instead of saying things out loud or actually gesturing – with the added fillip that because it's typed text rather than physical action, you can as easily depict yourself as walking across the ceiling as dropping a handkerchief.

Like persons performing in an improvised theater scene, the users of a MOO – usually called *players*, for historical reasons – can interact with one another on a variety of levels. They can behave *as if* the stage set, props, and costumed characters are what they appear to be. In the theater, one might refer to this as acting "in character". On a MOO, we would say that an action or statement was "consistent with the virtual reality." At any time, though, an actor can break from her role and speak or act instead as the *person* performing the role. In a rehearsal, she might do this to discuss a point of how the scene should be played; while not on stage, she might ask another player if he would like to go out for coffee later. During a performance, she might peek into the audience to see how full the house is. On a MOO, everyone is both a participant in the virtual reality (a *player*) and a person who is logged onto and using the MOO computer program (a *typist*).

Actions (typed commands) which are not consistent with or not intended to function as part of the MOO's virtual reality are referred to as *non-VR* or *meta-VR* actions. These actions, while outside the MOO's frame story, virtual reality, or theme, are still done within the MOO itself, that is, while logged on and typing commands, and they are as much a part of MOOing as actions that are consistent with the virtual reality. Some examples would include typing a command to see who else was logged on to the MOO at a particular point in time, a command to change the description of one's represented self, or a command to add a new room or object to those already available for use by oneself or by other players. MOO's are *user-extensible.* That the players within a MOO are themselves able to modify and extend the virtual reality is a central feature of what MOOs are for and about, and sets MOOs apart from chat rooms, web sites, and other virtual environments.

Much of this book concerns itself with how to investigate and manipulate the MOO's virtual reality underpinnings. Let me begin, then, by introducing a few terms. The MOO *server* is the program that runs on a host computer, whose job it is to handle the connection process, receive typed commands from users, cause those typed commands to be executed appropriately (much more on this later), and display resulting text to users' screens. The MOO *database* is a structured record of all the represented items within the virtual reality and all their technical underpinnings as created and extended by MOO users. A MOO's database is divided into *objects*, which are data constructs that are used both to represent things in the virtual reality and in the meta-VR. By this I mean that an object might represent a couch that you would sit on, in the context of the virtual reality – or it might not represent anything in the VR but serve as a repository of commands for extending the VR. Every object has associated with it a set of attributes, called *properties*, which are named pieces of data that contain information about that object (e.g. its name, its location, etc.). Some properties are common to all objects, while others are specific to certain kinds of objects: for example, objects that represent players (people) have a property associated with them indicating gender, while objects that represent pieces of furniture typically do not. In addition, objects also have associated with them sets of coded instructions, called *verbs*, that control their behavior. By creating new objects within the MOO, and adding interesting properties and useful verbs to these objects, *every* player can expand the virtual reality. Players adept at programming can even create virtual tools designed to modify the virtual reality.

The objective of this book is to provide a comprehensive yet comprehensible explanation of the many commands, both VR and non-VR, available to users of MOOs, in hopes of helping people learn the ropes as quickly as possible and get the most from their MOOing experience. Part I of the book focuses on commands that are common to all MOOs based on a particular common starting database called "LambdaCore". Part II provides information specific to LambdaMOO, which is the original and largest MOO in existence today (having between 3000 and 4000 registered users at any given time).

A note about typography: this is partly a book about words that people type into a computer program and words that a computer program prints on people's screens.

In the beginning, for clarity, if I specify something that you are to type in, I will put it on its own line, thus:

```
connect Guest
```

In the interest of space, though, I sometimes place text that you are to type (or text that the computer displays) within a paragraph, thus: `connect Guest`.

I use angle brackets <> as a placeholder for specific information that you fill in yourself. For example, if I told you to type:

```
say Hi, my name is <your name>.
```

and your name were Sally, then you would type:

```
say Hi, my name is Sally.
```

A note about the “Spivak” pronouns I use in this book: The mathematician Michael Spivak developed a set of gender-neutral singular pronouns for use in his books. They are popularly used on LambdaMOO when one doesn’t know a person’s gender, or when one is referring to a generic person such as “the reader” or “a MOOer”. They correspond to the familiar pronouns as follows:

E:	He/She
Em:	Him/Her
Eir:	His/Her
Eirs:	His/Hers
Emself:	Himself/Herself

Any other terms may be found in the glossary at the end of the book.

One last bit of reassurance: a MOO is a very rich interactive environment, providing thousands of different commands that enable people to use the MOO in many different ways. Because the commands are so many and so varied, the initial learning curve can seem quite steep, and people who aren’t comfortable using computer programs whose interface is strictly text may feel daunted. Please be assured that MOOs are intended to be arenas of exploration, and that while you may find yourself confused or uncertain on some occasions, there is nothing you can do that will mess things up irrevocably. You will learn as much (and hopefully more) from your own experimentation as you will from this book. Go ahead: Try things!