Macromedia Director as a Prototyping and Usability Testing Tool

by *Stephanie Ludi*

Efforts to understand user requirements commonly focus on the functionality and features of a product. However, it is important to analyze other product attributes, such as usability. A product may meet all of its functional requirements, but can fail if it has an interface that is difficult to navigate and learn. To address this problem, it is important to get feedback from users as early in the development life cycle as possible. A common technique is to develop a prototype or mockup of a product's interface to present to users. In *Usability Engineering*, Jakob Nielsen writes, "User testing with real users is the most fundamental usability method and is in some sense irreplaceable, since it provides direct information about how people use computers and what their exact problems are with the concrete interface being tested" [4].

A common testing technique is to present a prototype to a user and ask them to think out loud as they encounter new screens. The user is then asked to perform a set of structured tasks that correspond to the desired functionality of the product. Data can be gathered to measure response rates and the selection of items on the screen. Informal techniques such as surveys can also provide meaningful data regarding user preferences on layouts and confidence levels in completed tasks.

Usability engineers use a variety of tools to develop prototypes, including the old-fashioned paper and pencil method, and a combination of HTML, Javascript, Visual Basic, and Java. An alternative tool useful for building prototypes for usability testing is Macromedia Director. This article explores Macromedia Director by providing an overview of how to use the tool. While prototyping and usability testing are important throughout the development process, Macromedia Director will mainly be presented as a tool to be used during the analysis and design phases.

Director is a multimedia-authoring tool for PCs, Macintoshes, and the Web [1]. Multimedia-rich programs can be developed with little programming skill. Alternatively, advanced users can tap into Director's complexities with some programming experience. Traditionally, Director has been used to create presentations, games, and training software as stand alone executables or as programs to be played Web browsers via the Shockwave plug-in. Director is also useful for the prototyping and usability testing stages of software development.

Director is a visual development environment that accepts several types of multimedia and text files. The executables that are created can be saved as either .EXE files or Java files. Because Director can export projects to Java, it can be used to save time in creating a project in other Java development environments. Although HTML can be used for prototyping, Director supports more file formats than most Web browsers and allows complex scripting. Visual Basic (VB) is another common tool for developing prototypes. However, VB can only be used on a Windows platform, whereas Director is suitable for use on both Macintosh and Windows platforms. Another advantage of Director over other tools is that text and multimedia content can be linked to the project allowing content files to be easily maintained separate from the Director project. In addition, linkage based includes make reusing content files in the final programming environment feasible.

While Director will not eliminate standard development environments or programming languages, it will enhance the prototyping and usability testing experience by allowing developers to gather feedback from prospective clients and users early in product development. Early prototyping will allow developers to identify and fix defects early in development [3]. Miscommunication of customer requirements can cost time and resources if not addressed early in the product's development. The earlier in development that one can uncover defects, the lower the cost to repair the defect [5].

Director's metaphor

Director uses the metaphor of directing a film. There are several tools and windows that enable you to have complete directorial control of all facets of the project. The window where the project exists is called the stage. Figure 1 shows several of the typical windows used in Director.

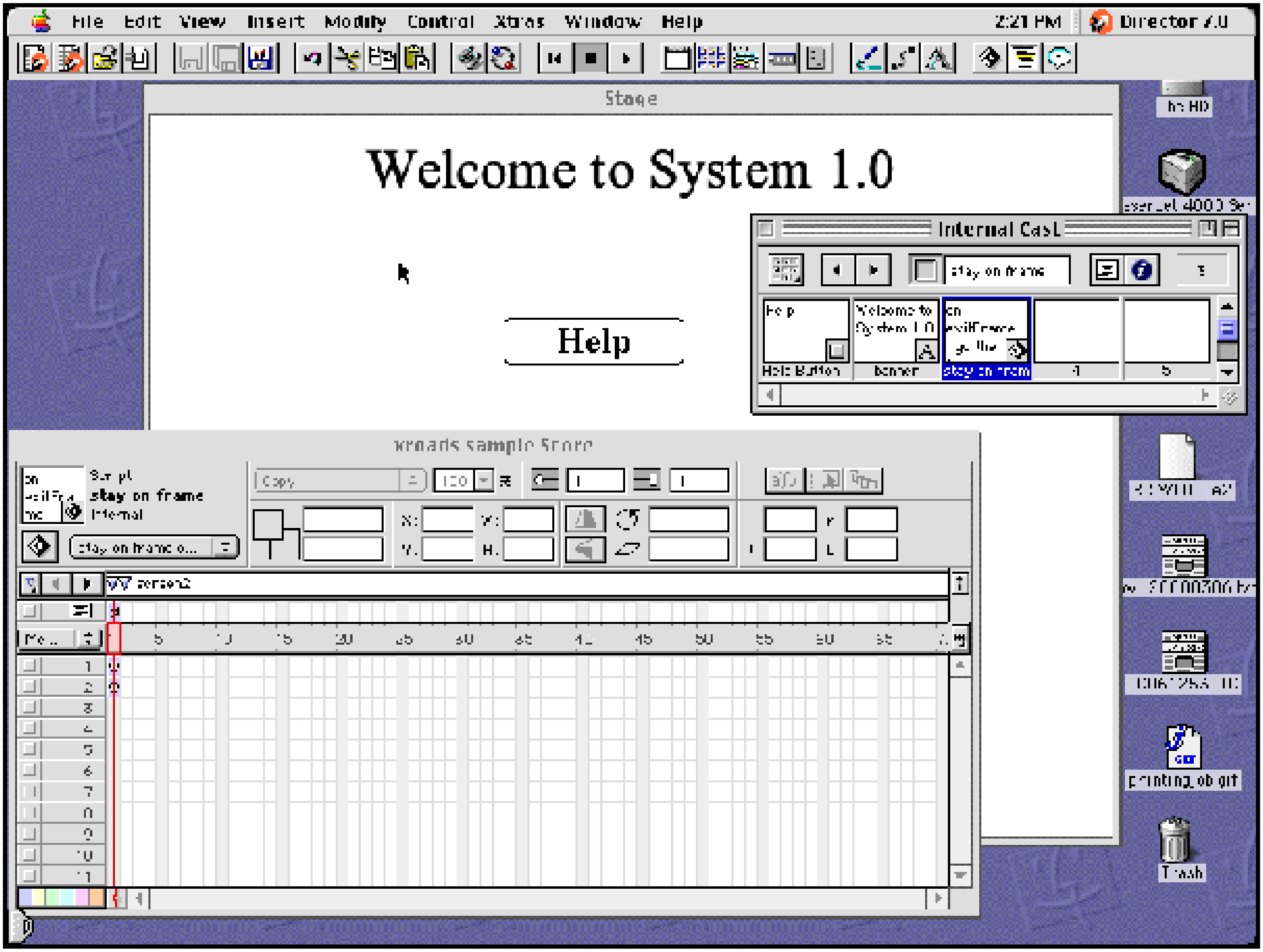

Figure 1. Director 7 environment.

The program environment consists of a series of frames (**screens**) and the cast members are the media elements and content in the frames. Each frame is a screen in your program. The user can interact with your program by clicking on, dragging, or otherwise interacting with the cast members that you placed in the frames. Each cast member has a unique ID and can be named for easier referencing as shown in Figure 2. These cast members can be placed on many frames or can be placed on a single frame multiple times. A collection of cast members, called the **Cast**, can even be shared among Director projects.

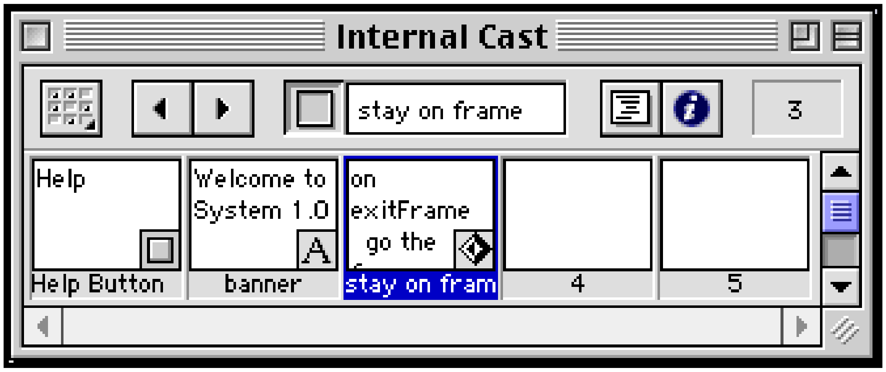

Figure 2. Cast for sample program containing three cast members.

You direct the pace and flow of your project with the Score window, shown in Figure 3. This window consists mainly of a large table, where the columns are the frames of your project and the rows are reserved for the cast members. Each cast member that is placed in a frame resides in a separate table cell. This arrangement allows for finer control over the manipulation of cast members within a frame.

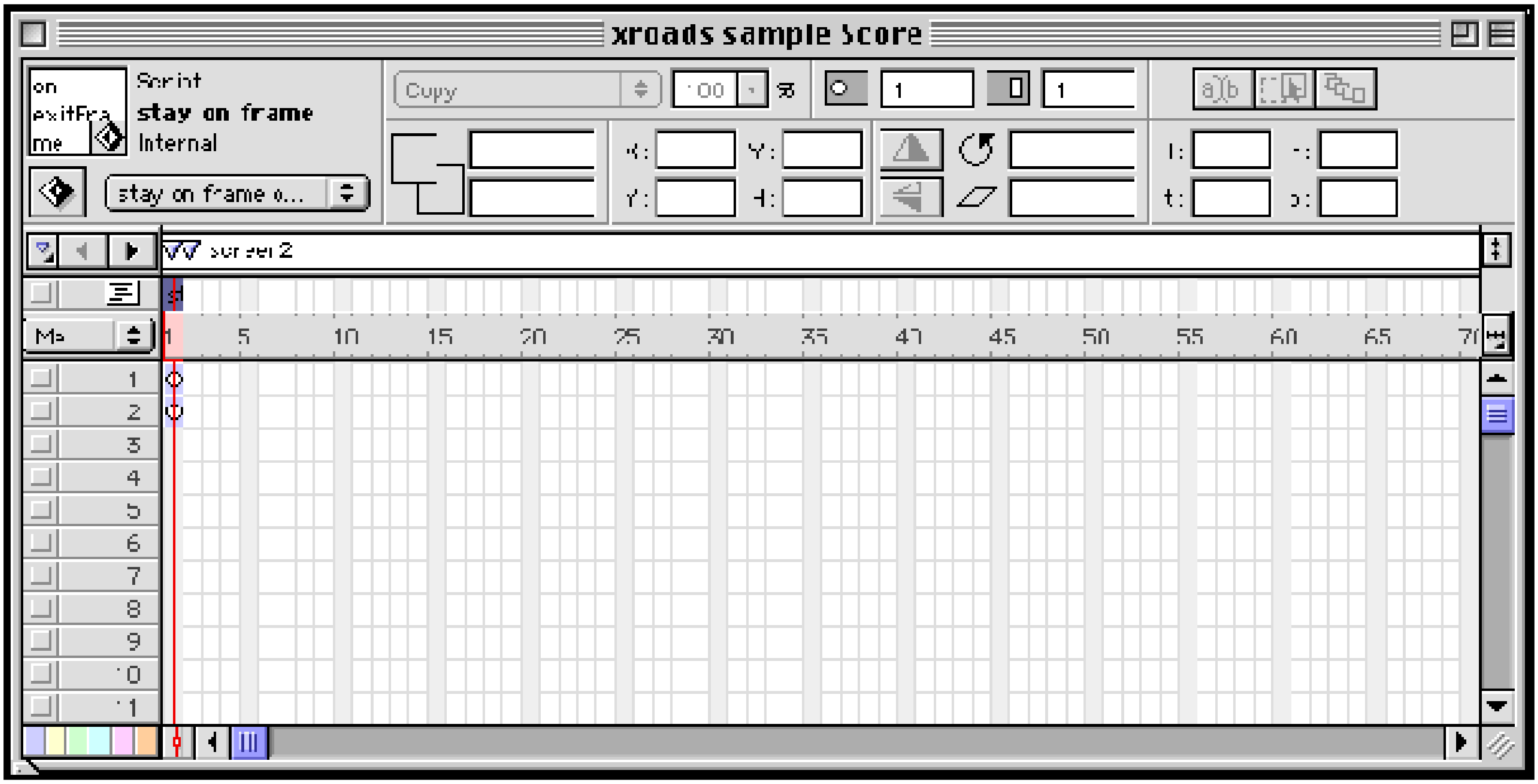

Figure 3. Score for sample program.

As the Director metaphor is not a traditional programming metaphor, it takes some time tp get acquainted with the system. A learning curve does exist, but many resources exist online, in books, and within Director itself. Although people learn at their own pace, Director can be learned and applied in a short amount of time, from hours to days, once the metaphor and a small set of Lingo commands are understood.

Prototyping

Coding the product prototype in the development language can be potentially useful, but the creation of a prototype in another format can save time on the project. Director contains wizards, called **Behaviors**, to do common tasks, an easy to learn scripting language, called **Lingo**, and drag and drop capabilities for placing elements on the screen. The combination of these features allows the user to create and change prototypes faster than using the development language.

Constructing prototypes is simple. Director supports a variety of content file types including: GIF and JPEG images, AIFF and WAV audio files, ASCII and RTF text files, Quicktime, and Flash files. Embedding files within the prototype executable is supported, but the files can also be kept external to the prototype file. With such support, you can revise the files in their respective programs (e.g. Photoshop) and the changes will automatically be reflected in the prototype. Linking files externally also allows you to reuse the same content in your prototype as for the product itself. Developing several variations of a prototype is also made easier by linking the content files. Simple bitmap and vector drawing programs, in addition to a text editor, are available in the Director environment as well.

By incorporating multimedia elements, along with built-in interface components, the interface can be quickly prototyped. Buttons, text fields, check boxes, and pull-down menus are built-in and can be easily customized to meet your needs. You can also tap into different event handlers with Lingo in order to specify what types of user actions are responded to by the system. Several event handlers exist for user actions(e.g. mouseUp, keyDown) and the state of the system(e.g. enterFrame, idle).

Using the built-in wizards (Behaviors), you can quickly add navigation to your project, control the playback of multimedia elements, animate objects, and customize your interface components without having to program. You can also add navigation capabilities to maneuver among the frames of your project using Behaviors. Another Behavior allows you to add tool tips to buttons. Tool tips are the explanatory text that pops up when you place your mouse over a button.

Using Director's scripting language, Lingo, provides the ability to link the activation of interface components to static screens allowing the opportunity to walk the customer through the interface, while presenting the flow of the product. Such walk throughs provide opportunities for feedback on the requirements of the product. Figure 4 shows an example of a Lingo script, which responds to mouse events over an addressbook icon. The script for the addressbook icon responds to three different interactions: mouseUp, mouseEnter, and mouseLeave. When the user clicks on the addressbook icon (on mouseUp), text and timing information is inserted into a global variable, the timer is reset, and the user is taken to the next screen of the prototype. When the mouse pointer is over the addressbook icon (on mouseEnter), an enlarged and animated version of the icon appears (as is shown in the lower-left-hand corner of the screen). When the mouse is no longer over the addressbook icon (on mouseLeave), the animated icon disappears, reverts back to the original size, and the animation is rewound to the beginning. The script is not long, but it demonstrates how an icon in an interface can behave in a prototype and how information about the user session can be captured.

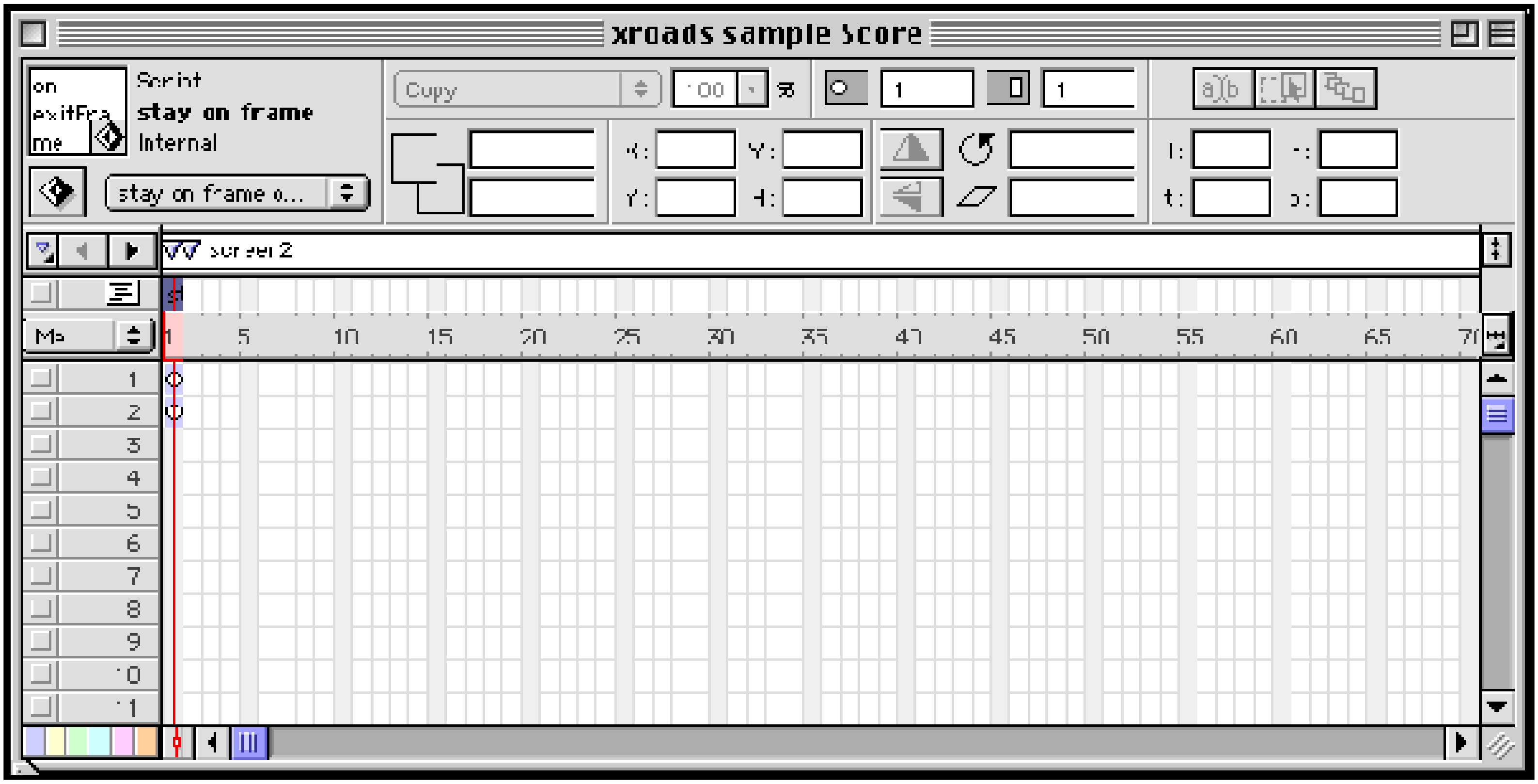

Figure 4. Script corresponding to the Addressbook icon.

By investing some more time, you can also create prototypes that can simulate the product. Allowing the customer to try a simulation of the product can generate more meaningful feedback in terms of some non-functional requirements (e.g. performance) and functional requirements that delves below the superficial appearance and flow. For example, data input and output issues can easily be addressed using such prototypes.

Sharing your prototype with your customer is facilitated by the ability to save Director programs to a stand-alone executable, called a **Projector**. Director also provides capabilities to save the program as Shockwave or Java files. Shockwave and Java versions of your prototype are cross-platform, but the Shockwave Plug-in is needed to view the Shockwave files. There may be other compatibility issues related to different content files types that you wish to utilize in your prototype that need to be investigated. For example, you cannot use your Flash files if you are exporting your project to Java. If you did plan on utilizing Flash files, you need to save your project as a stand-alone executable or to Shockwave.

Using Lingo Scripts

An advantage that Director provides is the ability to easily keep track of user and task information. User and timing information can be gathered with Lingo scripts, then later exported to a text file, which encapsulates the user's session data. Director provides the capability of gathering feedback from multiple users and saving the data in separate files for each user. Analyzing the feedback can be useful in finding use patterns and in comparing performance and other requirement data. An efficient way to consistently gather user information for your project is to create Lingo scripts that keep track of information. To time a task, a built-in timer can be started with the `startTimer` command. This command will keep time in ticks, where each tick is 1/60 of a second (i.e. 60 ticks is one second). To capture the time it takes for a task to be completed by a user, you only need to place the current state of the timer in a variable. In the following script, the mouseUp event is handled when the OK button is pressed, thus completing a hypothetical task.

```
on mouseUp

global textinfo -- the variable used to store information to be
exported

put " time= " after textinfo
put the timer after textinfo -- the value of the timer is appended

end
```

The text in the `textinfo` variable can then be manipulated so that you can add more information that can later be saved to the file. Once the user session is completed, you can save the captured information to an external text file, which can later be imported into a spreadsheet or other data analysis program. You can save all of the information at once from a user session by placing a text export script in the Quit button. The script will run when the user releases the mouse button (the mouseUp event) on the Quit button. The following script will take the information in the global `textinfo` variable and send it to a text file named `person 1`.

```
on mouseUp
global textinfo -- a global variable that holds all the text to be
sent to the file
global filename --the filename selected for the text file

put "person 1" into filename
setPref filename, textinfo --creates the file and sends the contents
of textinfo

end
```

The text in the file, `person 1`, for a hypothetical user would appear as:

```
User1
Enter Name Task; time= 200; completed;
New File Task; time= 430; completed;
Enter Product Order; time= 1244; completed
Enter Billing Information; time= 10238; not completed
```

While this article scratches the surface of Macromedia Director, you may find that Director has the features and flexibility that you need to prototype you next project. If you intend on conducting usability tests with your prototype, then Director is also a viable tool. Although Director is expensive ($399 for the educational version), the tool is most useful if you intend to use the behaviors and the scripting capabilities. You can download a 30-day trial version and various lesson files are available online [2].

References

1

Macromedia Director, http://www.macromedia.com/software/director/

2

Macromedia Director Support Center, http://www.macromedia.com/support/director/

3

Mayhew, D. *The Usability Engineering Lifecycle: a Practitioner's Handbook for User Interface Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1999.

4

Nielsen, J. *Usability Engineering*. Academic Press Professional, Inc.,Chestnut Hill, MA, 1993.

5

Pressman, R. *Software Engineering: a Practitioner's Approach*. McGraw Hill, New York, NY, 2000.

Further Reading

1

Director Online User Group, http://www.director-online.com/Display.cfm

2

Macromedia Director Online Tutorial, http://www.sci.sdsu.edu/multimedia/workshops/dir/index.html

3

Director 7 Tutorials and References, http://www.fbe.unsw.edu.au/learning/Director/

4

Epstein, B. *Lingo in a Nutshell*. O'Reilly & Associates, Cambridge, MA, 1998.

5

Gross, P., Elley, F., and Tucker, K. *Director 7 and Lingo Authorized*. Macromedia Press and Peachpit Press, Berkeley, CA, 1999.

6

Manuel, M. *Macromedia Director Workshop*. Hayden Books, Indianapolis, IN, 2000.