



## Multimedia for Music Study on the Web: Director from Macromedia

Dave Headlam



KEYWORDS: multimedia, director, web

ABSTRACT: The advent of powerful, widely accessible, and financially viable personal computers with network connections on the World Wide Web has led to exciting possibilities for creating musically meaningful presentations for teaching, performing, and researching music and disseminating them easily to colleagues, classes, or even to a potentially vast mass audience. These presentations can combine sound, images, animation, and interaction for stimulating and interesting results. The downsides of learning different programming languages, understanding Web page design and networking, dealing with multiple platforms (principally MacIntosh and Windows), and the problems of generating sound on different computers have a solution in the multimedia authoring program Director and its Web counterpart Shockwave, from Macromedia. This paper outlines the format of Director in its Web version with an emphasis on its potential for sound, and includes some demonstration files, and closes with some considerations of technique and content in web authoring for educational purposes.

- 
- [1. Introduction](#)
  - [2. Educational and Research Writings on Music Using Computers](#)
  - [3. Files on the Web: Quality, Quantity, and Convenience](#)
  - [4. Sound on the Computer from the Web](#)
  - [5. Web Authoring: Browsers and HTML](#)
  - [6. Shockwave plug-in](#)
  - [7. Director](#)
  - [8. Music Analysis on the Web with Minimal Scripting: An Example](#)
  - [9. Scripting](#)

10. [Teaching Music Theory: A Scripted Example](#)

11. [Streaming Audio](#)

12. [More on Shockwave: Web Programming](#)

13. [Conclusion](#)

---

## 1. Introduction

[1.1] With the advent of powerful, widely accessible, and financially viable personal computers with network connections on the World Wide Web, musicians of widely varying interests are presented with an intriguing prospect: the possibility of creating musically meaningful presentations for teaching, performing, and researching music and disseminating them easily to colleagues, classes, or even to a potentially vast mass audience. These presentations can combine sound, images, animation, and interaction for stimulating and interesting results. The frustrating downside to this picture is, of course, the necessity of learning different programming languages, understanding Web page design and networking, dealing with multiple platforms (principally MacIntosh and Windows), and, worst of all, solving the problems of generating sound on different computers.

[1.2] Fortunately, a solution to these obstacles exists in the multimedia authoring program Director and its Web counterpart Shockwave, from Macromedia. This program, currently at version 6.0, is available in an educational form for around \$300 and is well worth the time and effort to learn. In this paper, I will outline the format of Director in its Web version with an emphasis on its potential for sound, and include some demonstration files. I have used Director only on the Macintosh, and this article is written from that perspective, but the program exists in almost the same form in Windows versions, with differences noted clearly in the documentation. The Web compiler, Shockwave, allows authors to compile Director files for use on the Web, and a Shockwave plug-in for Web browsers allows users to access the Director program in a suitable format. <sup>(1)</sup> I have used Shockwave only with the browser Netscape Navigator, but other browsers also support Shockwave.

## 2. Educational and Research Writings on Music Using Computers

[2.1] The computer digital revolution is often compared to the development of the printing press in the 1500s. The printing press allowed for wide dissemination of reading material as well as musical scores. While books were illustrated, combining the visual with the printed word, for musicians books and scores separated the visual from the aural. An educational industry has grown up around attempts to recombine these two aspects, usually with value placed on the ability to conjure sound from image using internal musical production and by developing musical memory. The computer, however, allows authors to combine sound, text, and images, thus creating a true musical book.

[2.2] Those who believe in the split between the visual and aural might be skeptical of the computer's sound production capabilities, regarding them as a crutch similar to using a piano for sight-singing or part-writing. Those who regard it as a benefit when the visual and aural are joined in a musical environment will find truly new uses for computers in musical production, education, and research. The trick, I believe, is to use the computer in ways that it is uniquely suited for, and not to translate books or musical scores into computer programs with the same goals and perceived limitations. Several interesting uses of computers, in Computer Aided Instruction (CAI) and with Hypercard Stacks have and are emerging, as well as some innovative Web page designs, and it is my hope that musicians will be stimulated by this discussion of Director to keep developing new tools and paradigms in this area. <sup>(2)</sup>

## 3. Files on the Web: Quality, Quantity, and Convenience

[3.1] The first consideration for Web authoring is the tradeoff between the quality of the images or sound viewed and the quantity and size of files required. These are in direct proportion: the larger the file, the greater the amount of information, and the better the resulting quality. However, on the Web, quantity is also inversely proportional to download time and ease of use. Thus, the larger the file, the longer is the time it takes for the client computer to receive the information from the server, and the more taxing the requirements of the downloaded file on the client computer in terms of memory and

processing time.

[3.2] At least five solutions have arisen in response to this situation: 1) use of small files with often ingenious solutions to the limitations and/or acceptance of a lower quality; 2) development of compression programs that allow files to be sent in a smaller, compressed form from the server and then be expanded on the client computer; 3) breaking down of files into smaller components that are downloaded on demand; 4) linking to external media such as images, sound, and video (instead of copying them into a program so they are internal or inline), which are downloaded separately and on demand; and 5) streaming of files that only need to be partially downloaded before functioning, particularly with sound and video files, which tend to be large.<sup>(3)</sup> Director's Web design incorporates all of these solutions but allows for less reliance on the first solution than has been the case on the Web.

#### 4. Sound on the Computer from the Web

[4.1] As those with experience creating or receiving sound on personal computers from the Web can attest, it is often an apparently chancy and unpredictable procedure. Why is this the case? Transmitting images and texts on Web pages works almost 100% of the time, but sound requires a few conditions be set. Director's Shockwave plug-in handles the sound playback requirements for the most part, but it is useful to know a bit about sound files on the Web to use sound efficiently.

[4.2] Sound files have several forms. First, as relatively large files, they contain numbers that are representations of actual sound signals, carrying information about amplitude and time dimensions created from an Analog to Digital converter.<sup>(4)</sup> Sound files are in two sections, a header containing information relating to the control and format of the data, and the data itself. The header features control settings such as whether the sound is mono or stereo, what the maximum amplitude is, what the sample rate is, what the bit rate is, etc.. The remainder of the file, the raw sound data, is interpreted according to the settings in the header. To hear the sound, the numbers must be fed into a Digital-to-Analog converter to convert numbers to electrical signals that can be processed into sound, similarly to a home stereo setup with a CD player. This setup is included in most computers sold within the last few years, which all have CD-ROMS, but the client computer and its browser must also have some sort of program or plug-in to interpret the sound file properly and link up with the computer's sound hardware. Using Netscape, useful plug-ins related to sound are Quicktime from Apple (**Example 1**), and LiveAudio; these programs can handle the usual .AIF (MacIntosh) and .WAV (Window) sound file formats, as well as others. Director has its own sound format, .SWA, which is incorporated into its Shockwave plug-in.

[4.3] When discussing digital sound quality, reference is made to bits, bytes, and samples: the more bits, bytes, and samples in the representation, the better the sound quality and the larger the file. The standard Web sound quality, 8-bit and 22050 (or less) sample rate, is intolerable for musicians, but it is used because a mono sound file of 8-bit, 22050 samples (uncompressed) takes about 22 Kilobytes (KB) per second or 1.3 Megabytes (MB) per minute, while a CD-quality stereo 16-bit, 44100 sample file takes 166.4 KB per second or about 10 MB per minute in size. A mono 16/44100 file, perfectly acceptable, is 88.2 KB/s or about 5 MB/m, but a good compromise file size is mono 16/22050, which requires 44 KB/s or 2.5 MB/m. Even though sound files are compressed, with ratios of 3:1 or 4:1 common, they are still relatively large. With download times of, on average: 1 KB/second for a 14.4 Kilobit (Kb) modem, 3 KB/second for 28.8 Kb modem, 5 KB/second for a 64 Kbs connection, and about 30 KB/second for higher speed connections, such as the T1 connection often found in universities, sizes of sound files are a constant consideration.

[4.4] To get sound files to a more manageable size, several compression procedures are used, which cut down on the amount of numbers used to represent the signal. Audio compression is a complex topic beyond the bounds of this article, but Director and Sound Edit 16 have a particularly good compression feature in Shockwave Audio (discussed below), which allows for an acceptable quality of 16 bit/22050 sample, and provides for at least a 6:1 compression ratio (compressed file is 1/6 the size of the original).

[4.5] Other common types of sound files are Midi (Musical Instrument Digital Interface) files, that contain no information about the actual sound and so are quite compact, and files like Mod files that contain mostly Midi-type information but also a few sound samples.<sup>(5)</sup> Midi files have the advantage of being small, as they contain only information about frequency, duration, attack, articulation, etc. The disadvantage is that the resulting sound is entirely the result of the client computer

system, which can be good or bad, depending on the presence of synthesized sound sources. One useful development is the inclusion of a software synthesizer in Apple's Quicktime 2.5, so that Midi files can play directly through the MacIntosh's Sound Manager.<sup>(6)</sup> Director can use the same Midi tools as Apple's Hypercard, as part of its overall compatibility with Hypercard Xcommands. Mod files and others with a few sounds and lots of control instructions have been staples of computer games for years; they allow the author to control the resulting sound and have small file sizes. There is no record of Director uses of Mod files, at least that I have found, but they might be useful when specific sounds are desired for extensive sounds.

## 5. Web authoring: Browsers and HTML

[5.1] Web Browsers all read HTML (Hypertext Markup Language), a language consisting of a simple set of controls over data. Rather than include graphics and sound files within one file, HTML files include only instructions and text; other elements are called when needed from their resting places in the same folder as the HTML file. With this architecture, HTML files can be quite small and thus transfer quickly over the internet.

[5.2] HTML is basically a labeling language, but has associated scripting and programming languages like CGI, Perl, Java, and JavaScript. These allow for more complex interactions between the browser and the file, and for more interesting Web pages than the standard static page available with simple HTML commands. Currently Java applets, small programs that are compiled and run on the client computer, and Director Shockwave movies are the most common applications used for creating Web pages that combine sound, image, animation, and text. When, as is the case with Director movies, files sent over the Internet have more complex control sections, and use different proprietary command structures, the browser may need plug-ins to run or display them. Director's plug-in is called Shockwave; by using Shockwave from within Director, self-contained file is created which consists of controls and resources (such as images, movies, etc.) that are either placed within the file or are outside and are called when required (like HTML). The Director movie is called from within an HTML file using an EMBED command (described below). Linked sounds (.SWA files) and images (.GIF files) do not require a separate HTML file, but currently linked video must be called from within a separate HTML file.

[5.3] Readers running Netscape should look at their plug-ins. Go to the Finder and look inside the Netscape folder for the plug-ins folder and look inside at the plug-in programs available (Example 1 above). You will probably see Quicktime and LiveAudio—other plug-ins may be downloaded from various places on the Web. In the next section of this paper, you will put the Shockwave plug-in this folder. Back to Netscape. In Netscape, open up Options; then General Preferences, and then helpers (**Example 2**). A list of helper applications that call plug-in programs appears; it will probably also include Quicktime and LiveAudio. Click on one plug-in line and then edit to see the details (**Example 3**): the code names for the application and the file extensions or suffixes, such as .mid for Midi files and .dcr, .dir, .dvr for Director. These extensions are very important, as they tell the browser what type of file needs to be interpreted.

## 6. Shockwave plug-in

[6.1] Director's plug-in is called Shockwave, and it is used to interpret the control codes of Shockwave files, which are identified with .dcr suffixes. Before continuing with this article, the reader should get a shockwave plug-in and install it. Go to this URL address: <http://www.macromedia.com/shockwave/download> and download the plug-in, following instructions. Another address, <http://www.macromedia.com/support/shockwave>, has answers to questions. The plug-in and supporting files come in a compressed form and require Stuff-it Expander to expand. If you have this program, it will expand the file automatically (or you may have to double-click on the downloaded file). An installer program will appear; quit the browser and double-click on it to install the Shockwave plug-ins. Start up Netscape, and set the .dcr type in Options: General Preferences: Helpers: New (Example 3). If you don't have Stuff-it Expander, there is a Catch-22 because Stuff-it Expander itself is often presented for download in a compressed state. Here are two addresses to help getting Stuff-it Expander:

<http://www.aladdinsys.com/consumer/expander2.html>, and  
<http://www.aladdinsys.com/support/faqs/mexp2.html>.

[6.2] You will need to restart Netscape, but first increase the amount of memory (**Example 4**) that Netscape allots, since the program will now have to open up Director movies as well. Set the memory to at least 15000 KB (15 MB). Restart Netscape. Go to the Options folder and increase the cache size to at least 10 MB (**Example 5**). Finally, if you haven't already, extend the window to the right so that it is as wide as your screen. The accompanying movies to the rest of this article will appear in a new upper window which you can move around to be able to see the corresponding text while you look at the movie or image.

## 7. Director

[7.1] Director, as the name implies, uses the metaphor of a movie for file creation.<sup>(7)</sup> A director movie that is itself an introduction to Director is provided in **Example 6**. Go through the movie and refer to it for the following. Each element in the movie, an image, text, sound, or video, is called a cast member; when it sits in reserve in a cast library; and a sprite; when it is actually onstage;—the screen space allotted for the movie—and part of the action. The sprite corresponding to a cast member is independent and can be altered in many aspects (size, color, etc.), without affecting the original cast member, which acts as a template. Up to 32,000 cast members can be ready for use in a movie (using linked libraries), with up to 120 sprites possible on the stage at one time. Successive events—combinations of sprites—happen in frames; placed within a score; which controls the action. Each element, the frame, the score, the sprites, and cast members, can all have scripts or sets of commands associated with them, activated automatically within the movie or by the user. Pre-set scripts called behaviors; are included, but these can be altered and new scripts written. The user interacts by clicking on sprites, which can be buttons or hot; spots; as with Web pages, it is possible to indicate to the user where to click by changing the cursor to a hand; at those places.

[7.2] Director's design gives the author a choice of two levels for creating presentations: 1) using prepared scripts called behaviors; and 2) writing new scripts. Script is a general term denoting a programming language that is created to use common language references, such as the text of castmember 'x' to 'I love ice cream,' rather than the more complex allotment of variables, types, and memory locations found in C or C++ or other similar programs. While easier to use than programming languages, scripting retains certain features, such as "if then . . ."; and "while" loop constructions, and the ancestor - parent - child metaphor of object-oriented programs. Director's programming language is called Lingo. The standardized behavior scripts allow for application of scripts and interaction without independent programming, but those interested in programming can add their own scripts to govern the actions of sprites, etc. in the course of the movie. Since the behavior scripts have only appeared in the latest version of Director, 6.0, anyone who has used Director has already learned scripting, which was required of the author in earlier versions.

[7.3] Movies with behavior scripts only are generally created manually, using the changing relationships of sprites in the flow of frames implied in the movie metaphor. Cast members—images, text, etc.—are either imported or created within Director, using its Paint and Field tools, and put in a list of cast members, then dragged and arranged on a stage as sprites. Successive movie frames; are similarly arranged, with a few behavior scripts to create animations. The introductory movie shown above has few scripts, consisting mostly of 1) "wait" type commands, which stops the movie by causing it to replay the same frame until the user clicks; and 2) "go to" type commands that send the movie to a different frame when the user clicks. Another movie (see **Example 7**) with minimal scripting is one I created for a multi-media class at Eastman, with general information on computers and multi-media. The only scripts are "go to" types and "wait until user does something" types.

[7.4] When the movie is run, the frames run sequentially at speeds of 1 to 30 frames per second, allowing for animation or sequences of events similar to a movie reel. In frame-by-frame animation (see **Example 8**), the frame succession is clear when the tempo, or frame rate, of the movie is slowed. Frames can be looped to wait for user-activated buttons or hot; spots; these interactive elements allow the user to jump around between frames and control the pace or order of the movie, depending on its design. A simple interactive movie (see **Example 9**) allows users to alternate between images. Simple movies with animation are small, easy to create and compile, and can be found in many places on the Web.

## 8. Music Analysis on the Web with minimal Scripting: An Example

[8.1] As a simple example of music analysis using a Director movie with minimal scripting, I offer a study of Los muertos illvan alas de musgo, a madrigal by George Crumb (see **Example 10**). The Director movie of this analysis was created with 1) a Finale file of the score, which was imported into Director as the notes only, and the staves only, the two are combined in the movie, but having them as separate cast members allows for manipulations (such as coloring) on the notes alone; 2) colored cast members taken from the notes cast member; 3) a separate analytical stave; and 4) a sound file of the song, in this case a performance conducted by the author.

[8.2] The tricky part was to get the scrolling of the score to coincide with the music: the corresponding cast members of the score and notes were lined up and placed in successive frames on the stage to line up approximately with the flow of the music. As the score scrolls by, colored cast members are placed on the score, then moved up to the analytical layer to show the analysis develop in real time. The advantage is that the harmonies (pitch-class sets) are coordinated with colors, and that the student can listen to the score and start, stop, and pause the music and score for study.<sup>(8)</sup> The scripts are minimal for this movie, which is based on the frame-by-frame succession that is the essence of Director movies, and consist only of restart and pause controls, which are toggled when the user clicks on the stage, and the stop/restart button on the stage.

## 9. Scripting

[9.1] Scripted movies have the advantage that a small number of elements may be manipulated in different ways by controls in the script, so that the size of the file is smaller than if the elements were copied and manipulated manually. Director's language, Lingo, is somewhat similar to Hypercard's Hyperscript and has features found in many scripting languages. Every element of a movie has a controlling script which may be defined using program handlers that begin with the word on: from the movie itself (on startMovie), to frames (on enterFrame, on exitFrame), to cast members, sprites, and interactive buttons and hot spots (on mouseUp, on mouseDown). In addition to the program handlers provided within Director, users may write their own handlers, which are similar to functions in programs like C and Pascal, and either perform a task or take a value, change it, and return a value. Variables may be used locally, or defined as global and used in different handlers.

[9.2] The script of Lingo can be divided into at least nine categories: 1) decision scripts: mathematical symbols, numbers, operators such as = or <> (not equals); "if - then — else if - then — else" loops; "repeat while" or "repeat with" loops; 2) function scripts: handlers which perform a task, or take a value, perform an operation, and return a value; 3) classification scripts: define local variables, global variables, lists, put and set commands which assign values; 4) control and program handler scripts: scripts that run the program and connect with the computer, such as abort, exit, start, alert, mouseDown, KeyDown, button scripts, menus, sorts; 5) location scripts: setting the location of sprites; on the stage: top, bottom, right, left, location of mouse when clicking on stage; 6) attributes scripts: setting other attributes of elements than location: color, blend, height, number, name, type; 7) text-related scripts: identifying characters, words, lines, whether text field is editable by user, length; 8) connection scripts: connecting to other programs, files, desk accessories, resedit manipulations, Xlibraries, File Input/Output (I/O), etc.; 9) creation scripts: object-oriented programming elements: ancestors, parents, children; 10) Web and network-related scripts: controlling Web and networking features.

## 10. Teaching Music Theory with Director: A Scripted Example

[10.1] A typical freshman homework assignment using a scripted Director movie is given (see **Example 11**), with the first eight bars of Schubert's Moment Musicale op. 94/6 as the passage. The movie involves 1) the use of a notation program such as Finale to notate the musical passage, a reduction of the passage as the answer (if desired), and a blank stave of the same size as the reduction for the students to enter their answers to a counterpoint-type question; 2) a Roman Numeral, Arabic number, and note head repository for analysis; 3) color flags and text field to label non-chord tones; and 4) a sound file of the passage, the movie includes a sound file of the outer voices for the answer. For the first question, the passage appears on the stage, and the student can first listen to the passage, then arrange Roman Numerals and Arabic numbers in the proper place, and then move color flags over non-chord tones and label them. For the second question, on a blank stave, the student can arrange notes and numbers to show underlying counterpoint and/or a reduction of the harmonies. When done, if desired, the student can check the answer. To hand the assignment in, the student makes snapshots of the screen (command-shift-3 with any Macintosh after Operating System 7.01) showing the answer to question 1 then question 2; the snapshot saves a picture of the screen as a PICT file to the desktop. The student then labels the two pictures with his/her



name, and puts them in a homework folder.

[10.2] A non-working version of the movie showing the separate elements is included (see **Example 12**). To create the movie, the main cast members, the score and staves, are exported from Finale as PICT files (an image format that Director supports). First, import the files into Director as cast members, and size the stage at 432 X 455 pixels (for a 17 inch monitor, 464 X 310 for 15 inch). For Question 1, put the notation cast member on the stage. To create the Roman numeral, Arabic number, and note repository, use fields and type in the numerals and numbers, but use texts for the quarter note heads and sharp, natural, and flat signs (using a font like Petrucci). Fields are more flexible, but require that the user has the same fonts as the author—so they should be set up so that the font doesn't really matter; texts are treated like bitmap images so the user won't need the Petrucci or other notational font. To complete the cast, using the Paint tool, make a small colored circle to use for non-chord tones. Arrange the Roman Numerals and Figures and symbols on the stage. For each sprite, check the moveable box so that the user can move the symbol. Create a field for the student to add the names of non-chord tones, and fields explaining what the student is to do. Finally, add navigational buttons, to move to different frames. When allowing for interaction, use “go to the frame” in the frame script, as this allows the movie to keep running and accepting mouse clicks etc. (the use of “pause” stops the movie). For Question 2 and the answer, follow a similar procedure.

[10.3] A working version of the movie showing its scripts is also included (see **Example 13**)—the main movie script is given in the top right hand field, changing scripts according to what sprite is clicked are given in the lower right field. These fields may be moved by dragging with the mouse.

[10.4] When scripting a movie, it is easiest to place all handlers in “Movie” scripts, under a general global line, so that the global variables are available to all handlers, with calls to the handlers in the scripts of the various corresponding sprites, frames, etc. This avoids the common problem of undefined global variables—a situation which causes errors not always clearly articulated by Director's automatic de-bugging features. The Movie script; sets up several groups of variables, beginning with IH, IHH, etc., which are variables set to the initial locations of many sprites (necessary for the script when they move and recreate themselves as the user performs the analysis). The next variables (scroll down in the upper right hand box to see these) are for two lists, then for counters also used in the move/recreate scripts. Global variables for the sound follow. The “on startMovie” handler sets up the initial values for locations and counter global variables. It also initially sets all sprites visible, as some are made invisible in the course of the movie, but toward the end another handler called “nonvissprites1” makes the sprites corresponding to question 2 invisible, since the user will presumably start with question 1. (If the user goes to question 2 first, a script along the way will make the relevant sprites visible.) It also erases all cast members past a certain number: this is the number of functioning cast members, until users start to perform the analyses, then more cast members are created. Every time the movie is restarted, the excess cast members are deleted.

[10.5] The essential script for all the moving sprites (the roman numerals, etc.) is “dupandmove,” placed within the movie script. Move one of the Roman Numerals to show this script, then look in the lower right field. The “if .. then” script is a test: if the sprite has been moved from its original location, then it will not duplicate itself; this is so that users can adjust the Roman Numerals, etc. they have already placed in the score. If the sprite has not been moved, then the handler “dupandmove” is invoked. This script, shown in the upper right hand field, duplicates the cast member corresponding to the sprite, assigning the duplicate all the attributes of the original (all the “set the” lines in the script) so that there is an apparently endless supply of numerals and figures. In fact, the duplicate replaces a dummy sprite that is sitting unnoticed on the stage, starting with sprite 120 and moving backwards by the incremented value of a counter variable x. There is a total of 120-(the number of functioning cast members) of these dummy sprites, allowing for around 60 copies of the Roman Numerals, etc. The script places the duplicate at the same place as the original, while the original is free to move away.

[10.6] The trick in the script is to duplicate the cast member (the Roman numeral, etc.) only the first time the sprite is clicked—after it is moved, the user should be able to move it again without creating another duplicate. As noted above, this is the purpose of the location test beginning the script in each of these moving sprites, as seen in the lower field on the stage. If the sprite is at the original location, it will create a duplicate; if it has been moved once, it will not.

[10.7] The handler “dupandmove” also includes “puppetSprite” commands. To alter the condition of a sprite from the script, it is necessary to use a “puppetSprite spriteNumber, TRUE or FALSE” command. All elements, sounds, transitions, palettes,

etc. may be changed with Lingo, with the corresponding ‘puppet command (“puppetSound,” etc.). When changing any sprite attributes, it is also necessary to end the handler with “upDateStage” or to use “go to the frame” in the frame script to register the change—this movie uses the latter.

[10.8] A problem with the “puppet” commands within Director lead to the “visible” sprite handlers, “vissprites” and “nonvissprites.” To see these handlers, click on one of the navigation buttons: “question 2” or “Answer.” A channel or sprite that is puppeted will persist on the stage even when it is not in the score; thus, it needs to be made invisible when not required. The “vissprites” and “nonvissprites” handlers set the sprites corresponding to question 1 visible or invisible, while the “vissprites1” and “nonvissprites1” handlers set the visibility of sprites corresponding to question 2. Two lists are created to catalogue the sprites from these two questions: Q1List and Q2List. At the end of the “dupandmove” handler, the duplicated sprite number is put into the corresponding list, and the numbers in the list are used in the “visible” handlers. These handlers also include the sprite numbers corresponding to the original movable sprites (the Roman numerals, etc.), which also must be made visible or invisible, since they are also puppeted.

## 11. Streaming Audio

[11.1] Director movies may also include sound files.<sup>(9)</sup> With Director on the Web as Shockwave, having relatively large sound files embedded in the program causes slow download problems. One solution is to use Macromedia’s streaming audio format, which comes with an efficient compression of at least 6:1. Using this compression, sound files embedded in the movie are smaller, and can be used normally. When files are still quite large, it is also possible to link to external sound files and play them during the movie. This is the technique in the assignment analysis movie: the sound is played with the “Listen” button, which activates a specially prepared compressed sound file, “schubert.swa,” that sits outside the movie and is called when needed. The handler is set in the movie script, and the sound is called by the button script, with the handler “playswa.” With this streaming audio, a buffer is created and the downloading sound file is played when the buffer is full, generally about 5 seconds. As the file continues to download, it simultaneously plays.

[11.2] Returning to the homework assignment, to complete the steps for creating the movie, prepare a sound file of the passage, using a program like Sound-Edit 16 (also made by Macromedia), to create a mono file, 16-bit and 22050 samples. Use the SWA Compression Xtra in Soundedit and the SWA Settings Xtra for Director to compress the sound. The sound is given a corresponding dummy cast member, called “SWAHolder,” and the name of the sound file is assigned in the “on startMovie” handler. The sound handlers then take this sound file and play it, with a few configuration features. The sound can be heard any number of times by pressing the corresponding button. A second sound in this example is a reduction to the two outer voices. Both of these sounds were created first as Midi files, then altered to .AIF format sound files using Quicktime’s MoviePlayer, then Sound Edit 16 to change the sounds into compressed .SWA files.

## 12. More on Shockwave: Web Programming.

[12.1] Shockwave, both a process and plug-in for putting Director movies on the Web, was introduced in October 1995, and is available for Windows or Macintosh computers, and for Web browsers Netscape Navigator and Microsoft Internet Explorer. Since Shockwave introduces a new MIME type, .DCR, the server must be configured, and the client browser must be configured with the plug-in and helper type (as discussed above).<sup>(10)</sup> To create Shockwave movies, Director uses Xtras contained in folders called “media,” which contain Xtras related to streaming audio, and net support, which contain Xtras related to Shockwave movies on the Web. Director 5.0 had an Xtra called “Afterburner” to create shocked movies, but Director 6 has a built-in “Create Shockwave Movies” choice in the file menu, which creates a file ending with .DCR. Shocked movies are placed within HTML documents. The basic EMBED tag is: `<EMBED SRC=“movieName.drc” WIDTH=widthInPixels HEIGHT=heightInPixels>`.

[12.2] The first rule of Web programming with Shockwave is to start with the smallest movie possible: if the movie is large, divide it into smaller movies that call one another, or begin with an introductory movie that consists of only an image which displays while other movies load (using the “GoToNetMovie” script). The second rule is be clear on pathnames and file names. Director’s lingo, the `pathName`, is disabled in Shockwave, and the movie and supporting files all need to be in the same folder or directory and properly labelled. Files names have different restrictions on different platforms: UNIX servers



are case-sensitive; Windows machines may have trouble with names longer than eight characters with a three character extension or names with spaces. These two and MacIntosh may have trouble with special characters in file names.

[12.3] As with sound, color and graphics are complex topics unto themselves when it comes to size, quality, and compression. The safest route is to use the System Color Palettes and the smallest bit size for images that provides a clear image: the difference between 8 bit and 16 bit color is noticeable, but the former is acceptable. Use text fields if possible, with standard fonts: the client computer must have the same fonts to get the same text shape and size.<sup>(11)</sup> Director automatically substitutes between standard MacIntosh and Windows fonts. Text created in the text cast members is bitmapped and takes more space, but the client does not need the same fonts, and so the effect can be assured.

[12.4] When putting Director movies into shocked form on the Web, important considerations are: 1) data transmission rates; 2) New Lingo commands; 3) Director features disabled; and 4) miscellaneous issues that affect playing Director movies in a browser. Data transmission on the Web is, from a hardware point of view, standardized, with user modem speeds of 14.4 kilobits (Kb) per second (= 1.8 KB/s), 28.8 Kb/s (3.6 KB/s), 33.6 Kb/s (4.2 KB/s), and 56 Kb/s (7 KB/s). Higher speeds come from ISDN lines, ranging from 64 Kb/s (8 KB/s) to 1.5 Megabits (Mb)/s (184 KB/s), and T1 lines at 1.544 Mb/s (193 KB/s), Ethernet at 10 Mb/s (1.25 MegaBytes (MB)/s), and T3 lines at 44.736 Mb/s (5.592 MB/s). Over the internet, however, these theoretical speeds are affected by the data transmission capacity of server computers and by how busy the server and the lines between server and client are at that moment—a very unpredictable value. A given speed estimate for general use is about 1 KB/s. By taking the size of the movie and supporting files, approximate download times can be calculated. The most important advances for data transmission are streaming versions of data: Shocked movies and audio are both streaming, but streaming video is not incorporated into Director, although Quicktime movies, which are linked to from within shocked movies, are now streaming.

[12.5] Other concerns to users of Director are the features not available in Shockwave form, and the new Lingo scripts required by Net form. Features not available are mostly those related to operating systems: custom menus, printing, saving, restarting, resources, file I/O etc., but also include movies within windows. The Web-related Lingo commands are in three categories: those that start a process, those that monitor it, and those that send or receive a result once the process is done. This three-step set-up accommodates the essentially asynchronous Internet, where the time to send and receive data is unpredictable. Finally, a concern at all times on computers is the amount of free memory: Shocked movies on top of Web Browsers take up enormous resources, and the use of multimedia files can easily (on the MacIntosh at least) cause system crashes. While new versions of operating systems and internet protocols help this situation, it is always a concern when sending large and complex files over the Web.

### 13. Conclusion

[13.1] As I noted at the outset, the availability of powerful computers linked on the Web offers many possibilities, but also pitfalls. The most basic opposition here is technique versus content. It is easy (remarkably easy!) to become absorbed in the how rather than the why and what when working with computers. This situation is the result of many factors. First, although the Web is meant to be platform independent, in reality, each of the major systems, from Apple, Microsoft and the IBM clones, Sun, and SGI, has its own characteristic ways of interacting with the Web. Thus, authors must learn different programming languages and codes. Currently, for instance, Microsoft has its own brand of HTML, different for its browser, called Internet Explorer, than that used on the browser, Netscape Navigator, so that some web pages or features cannot be read by both programs. Although Director is useful since its Shockwave movies are playable on MacIntosh or Windows machines, in the latter case, users must have sound cards properly installed and overcome some difficulties in the plug-and-play; features to hear musical files. As well, servers are often Unix based machines. In my own efforts, I discovered that, while upper case or lower case letters in file names and links are equivalent on the MacIntosh, on the UNIX server a reference to a file with a different configuration of lower or upper case letters failed to find the file, and I had to redo some links and code.

[13.2] Second, since in the marketing of computer programs, prices must go down with successive generations of more powerful systems, companies dole out versions of software yearly that users are required to purchase to keep up. Although Director is useful, for instance, the initial cost is high (around \$800) until the educational versions come out, usually a year or

two later. The educational version of Director 5, for around \$75, was available only a few months before Director 6.0 came out. After the initial outlay, while upgrades are not too expensive, the price dissuades many authors. In my case, I learned the Shockwave version of Director 5.0 by cobbling together elements from books and the Web; in Director 6.0, these elements have been integrated into the program—but with some elements now changed from the Afterburner; generation. Such changes make working in multi-media exasperating.

[13.3] Third, to create worthwhile educational content for the Web, instructors have to re-imagine their course material, or invent new courses, appropriate for the new medium. Putting class notes and schedules in electronic form is convenient, but hardly represents a new use for computers. What is the criteria for useful Web texts? At Eastman, only a relatively small proportion of students have web access, so it is difficult to do tests with large groups. I have just begun to use web pages in my own teaching, but in the process of thinking about using the web and computers, a few important questions have come to mind and, I believe, need to be considered by all web authors for educational materials:

- 1) Do the materials use the capabilities of the computer to combine sound and image? Image and text alone work quite well in a book, and sound alone is easily available on CD systems. Combining images and text, with instant access to both, is a strength of the computer, and should be the goal of instructional materials.
- 2) Are the learning materials goal-directed? I'm not a big fan of pages full of multiple menu possibilities on successive pages or teaching materials organized like a game. Part of teaching is instructing students in a systematic and goal-oriented way, as their lack of experience with the material is the reason they're in the class to begin with: it is not appropriate to assume that they will make the best decisions in the order of material. Web materials should be efficiently and clearly presented, with the next step indicated.
- 3) Do the assignments or tasks on the Web allow students to learn at their own pace? This question requires a well-thought out interface, with lots of looping connectors so that students can go back over material and review/rediscover ideas and concepts as needed. This potential is also one of the great strengths of the computer.
- 4) Does the program or file point beyond itself, causing the student to make larger connections? Self-contained drill type programs might be of some use in a directed environment, but the ideal materials should both present the immediate topic and point (with links) to topics of larger implication.

[13.4] To the problems of multi-media authoring noted above—learning programming languages, Web pages, problems with multiple platforms and generating sound—Director offers an all-in-one solution, and is worth consideration by all those interested. Although it has a few quirks, Director is relatively easy to learn and get fast results with scripts.<sup>(12)</sup> As for the often frustrating and time-consuming process of developing content for the Web, Director can help with the technique, but the content—from imagination and an understanding of the medium—is the result of hard work and time spent in the environment. As operating systems and programs become ever more closely tied to the Web, with teaching and learning in the education business inevitably following, all of us will need to use computers in the most efficient and effective ways possible.

---

**Dave Headlam**  
**Eastman School of Music**  
**26 Gibbs Street**  
**Rochester, New York 14604**  
[dhlm@theory.esm.rochester.edu](mailto:dhlm@theory.esm.rochester.edu)

---

## Footnotes

1. The author is here defined as the creator of the program or file which is then utilized by the user. The author puts the file

on a server computer, and the user accesses the file on a client computer which is connected to the server in some way, most likely via the internet.

Computer programs such as word processors, webbrowsers, and operating systems are often designed with smaller programs in self-contained modules which can be added on easily. For instance, the MacIntosh operating system has system extensions and control panels, Director has Xtras, and Netscape has plug-ins, called helpers in the Preferences menu. These modules, put in an appointed folder, can be called upon when necessary to run certain parts of the program or to recognize and run related types of programs. Procedures made available by Xtras are called X-commands (Xcmd), and other names beginning with X. Netscape's plug-ins allow it to recognize file types, called Mime (Multipurpose Internet Mail Extensions) types, that are sent over the internet. Director's web plug-in for Netscape is called Shockwave.

Web browsers are programs designed to recognize and display different types of files, particularly Mime (Multipurpose Internet Mail Extensions) types, which are sent over the internet. Netscape Navigator is a well-known browser, that can display text, images, and animations written in different codes, such as HTML (Hypertext Markup Language). Browsers can be extended by plug-ins when new codes are introduced.

[Return to text](#)

2. Computer-aided Instruction (CAI) programs have developed along with personal computers. Some existing programs are MacGamut, Practica Musica.

Hypercard is a MacIntosh program using a stack of cards as the metaphor. The best known examples of its use are found in studies of works such as Stravinsky's Rite of Spring and Beethoven's Ninth Symphony, as well as the more recent Ragtime, put out by the Voyager Company. For recent examples of teaching uses, see the article by Elizabeth West Marvin and Alexander R. Brinkman in this issue of MTO. See also the Voyager web site and accompanying software, CDLink, which allows for Hypercard Stacks online.

For some Web pages designed for interactive activities and teaching, see the following <http://sparky.parmly.luc.edu/sharc> for Greg Sandell's timbre research, as well as <http://locutus.esm.rochester.edu/~cscotto/jrt121/> by my colleague, Ciro Scotto, as well as my own <http://locutus.esm.rochester.edu/~dhlm/acoustic.html>.

[Return to text](#)

3. In the past on the Internet, a sound or other file had to download completely and then begin playing, which can take a lengthy period of time. Streaming files are created with enough control information put at the beginning of the file so that the file can be processed and begin playing or presenting almost immediately. The remainder of the file continues to download into a buffer simultaneously. The problem is that download times are unpredictable on the Web and may change even during one file's download. Thus, a sound may begin playing, then stop, while enough information flows into the buffer to continue the play. Director's streaming audio allows authors to set the initial time before the playing begins (default 5 seconds), which allows some leeway, but there is no fail-safe solution.

[Return to text](#)

4. Soundwaves can be described digitally in a Cartesian graph type format, with x-values as time, measured in small chunks called samples. Samples of sounds are taken in fractions of seconds, usually in 1/22050 or 1/44100's of a second, or 44,100 samples = 44,100 times a second. The sample rate is twice the highest frequency possible in the sound represented, thus 44,100 samples per second allows for a frequency of 22050 Hz, or cycles per second, above most people's range of hearing; 22050 samples per second allows for 11025 Hz frequencies, within the range of hearing, but is acceptable as a compromise (the piano spans 27.4- 4186 Hz).

Computers store information in binary numbers, with each bit set to either 0 or 1, and in bytes of 8-bits each to store numbers or words, etc: 8-bits = 1 byte, 16 bit = 2 bytes. A kilobit (Kb) = 1000 bits, a Kilobyte (KB) = 1000 bytes, a Megabyte (MB) = 1,000,000 bytes. In the Cartesian graph representation of a soundfile, y-values are the amplitude levels, measured in divisions relating to the bits; present in the sound representation, usually 8 bits (=256 levels) or 16 bits (=16536

levels). The sound wave is much more faithfully reproduced by the many more levels in 16 bits, but for size reasons 8-bit sound has been used on the Web. The size of sound files (uncompressed) depends on the bits and samples: 8 bits = 1 byte, so that 8 bit, 22050 sample sound takes about 22 KB per second; CD quality stereo 16 bit, 44,100 sample sound takes 2 2 44,100 = about 166.4 KB per second.

An Analog-to-Digital converter converts a sound signal (trumpet note, traffic noise) from its electrical representation into a series of numbers representing the time and amplitude values of the signal. A Digital-to-Analog converter reverses the process, creating sound from a series of binary numbers.

[Return to text](#)

5. Midi file information can be found at <http://www.eeb.ele.tue.nl/midi/index.html>, but there are thousands of sites with MIDI files and info. Mod file information can be found at: <http://members.aol.com/Edgeman101/musica.html>.

[Return to text](#)

6. Information and downloads related to Quicktime 2.5 can be found at: <http://quicktime.apple.com>.

[Return to text](#)

7. Although useful by itself, Director may also be ordered in a Studio, bundled with a sound editing program, SoundEdit 16, an image editor, xRes 3, and a 3D tool, Extreme 3D 2, for a complete multimedia package.

[Return to text](#)

8. In the original CD-ROM version of this movie, the sound is a quicktime movie, and can be accessed at any point, as well as stopped, started, and paused. Director's streaming audio unfortunately only allows for the latter.

[Return to text](#)

9. In non-web uses, sounds can also be placed in the two sound channels; when the movie plays the frame with the sound file in it, it will play the sound file. To coordinate sound with animation, it is easier to import the sound as a quicktime movie; quicktime movies can be controlled virtually frame by frame, and so a quicktime movie sound can also be started and stopped and accessed at any point easily to coordinate with animation. Unfortunately, this is not yet possible with linked quicktime sounds or Director's streaming audio.

[Return to text](#)

10. On servers, a file called "mime.type" (UNIX) contains many file types; add the line: application/x-director dcr, dir, dxr.

[Return to text](#)

11. The standard system fonts: Arial, Courier New, Symbol, Times New Roman, and Wingdings on Windows and Chicago, Courier, Geneva, Helvetica, Monaco, New York, Palatino, Symbol, and Times on the Macintosh. If you choose a standard Macintosh system font, Director automatically substitutes a standard Windows system font when someone runs the movie on a Windows computer--and vice versa.

[Return to text](#)

12. Director has a few quirks that should be noted. First, the first frame of a movie often doesn't work properly when sounds or scripts are present, and it is best to begin in frame 2, leaving frame 1 either empty, or with the same stage as frame 2 without the sounds, scripts, or markers, so that the movie begins and passes through at least one unscripted frame before the action begins. Second, the last frame of a movie needs to be considered carefully, otherwise the movie will just end; often, the ;last; frame loops, waiting for a user click. Third, repeat loops and pauses shut out the user (and, in Web pages, the browser as well) and should be used carefully; options should be given to stop any lengthy activity. Fourth, when using text, use ;field; cast members for general text but use a standard font, since client computers will only show fonts that they have mounted; for specific fonts or text arrangements, use ;text; cast members, which are read as bitmaps. Fifth, when done making a movie, it is often convenient to get rid of unused Cast members by opening the Find command, which has a setting ;not used in movie.; The list of members includes movie and frame scripts however, which must be kept. It is thus best to name these scripts to avoid a tedious search to be sure which cast members may be deleted. There are also a few general

quirks of computer programs to remember. For instance, when using linked media, make the link to a copy of the image or sound, etc., because any changes made within, in this case, Director, will change the source media.

For the best source on Director, go to <http://www.macromedia.com/>, for music-related shockwave sites go to <http://www.teleport.com/~arcana/shockwave>; a few helpful books are: Gary Rosenzweig, *The Comprehensive Guide to Lingo* (Ventana, 1996), and Cathy Clarke, Lee Swearingen, and David K. Anderson, *Shocking the Web* (Berkeley, CA: Macromedia Press, 1997).

[Return to text](#)

---

## Copyright Statement

**Copyright © 1997 by the Society for Music Theory. All rights reserved.**

[1] Copyrights for individual items published in *Music Theory Online (MTO)* are held by their authors. Items appearing in *MTO* may be saved and stored in electronic or paper form, and may be shared among individuals for purposes of scholarly research or discussion, but may *not* be republished in any form, electronic or print, without prior, written permission from the author(s), and advance notification of the editors of *MTO*.

[2] Any redistributed form of items published in *MTO* must include the following information in a form appropriate to the medium in which the items are to appear:

This item appeared in *Music Theory Online* in [VOLUME #, ISSUE #] on [DAY/MONTH/YEAR]. It was authored by [FULL NAME, EMAIL ADDRESS], with whose written permission it is reprinted here.

[3] Libraries may archive issues of *MTO* in electronic or paper form for public access so long as each issue is stored in its entirety, and no access fee is charged. Exceptions to these requirements must be approved in writing by the editors of *MTO*, who will act in accordance with the decisions of the Society for Music Theory.

This document and all portions thereof are protected by U.S. and international copyright laws. Material contained herein may be copied and/or distributed for research purposes only.

Prepared by Jon Koriagin and Tahirih Motazedian, Editorial Assistants