DOWNLOAD PDF MACROMEDIA DIRECTOR MX AND LINGO

Chapter 1 : Lingo (programming language) - Wikipedia

Lingo is the language that makes Director work in ways that outshine the simple flashes and buttons of old Internet content. Director Lingo can be combined with other applications to create truly interactive web content and endless possibilities for designers.

History[edit] Lingo was invented by John H. Thompson at MacroMind in , and first released with Director 2. Jeff Tanner developed and tested Lingo for Director 2. Dave Shields tested and documented Object-based Lingo for Director 3. Lingo was quickly adopted by burgeoning multimedia community during the s and the already popular Director product. The Journeyman Project is a prominent example of this. Features[edit] Lingo is embedded into Adobe Director, and can be added as scripts on objects or on the timeline. Lingo is an object-oriented programming OOP language, and supports Smalltalk -like verbose syntax, OO dot syntax, and inheritance. Verbose syntax[edit] When Lingo was created, a verbose syntax was designed to mimic spoken language, so it would be easy for new users. Users could write HyperTalk -like sentences such as: The equivalent in new scripting style would be: The syntax in prior versions would be like: Each type of script may be added to certain types of compatible objects. Cast scripts work only with their member, not all events can be used with them. Behavior scripts are attached to a sprite or inserted into a frame. Frame behaviors can be used to create a pause or delay within a certain frame in the score. Behaviors make it easy to program in an object-oriented way, as you can directly see the relationship between the programming and the item they are attached to. They can also control or interact with other sprites, making them a true object. Movie scripts are not attached to sprites nor can they be instantiated as Objects. They are available throughout the program movie and are especially useful for holding global handlers and initializing global variables at the start or end of the movie. Parent scripts are used to birth create instances of an object into a variable using the new command. These objects can control sprites and other media remotely, without being attached to any one sprite, may be used to control data or other non-displayed items, and are useful for recursion routines such as pathfinding. A Parent script can be used to create or destroy an object at any time, freeing them from the confines of the score that a Behavior is limited to. Behavior and parent scripts encourage good object-oriented programming. Movie scripts are not as OOP-oriented. However, they can still be used to make black-box handlers, where other objects can input raw data and receive answers back, without knowing the inner workings of the box. Inheritance[edit] Lingo supports object inheritance by a slightly idiosyncratic system: Properties and methods of the ancestor are inherited by the parent. Behavior scripts are also a kind of ancestor of the sprites to which they are attached, since properties and methods of the behavior can be accessed by reference to the sprite itself. In this case, it is a kind of multiple inheritance, as one sprite may have several behaviors. XObjects[edit] Lingo 3. XObject API was openly available to developers and media device producers, which added to the popularity and versatility of Lingo. Macromind was very active in positioning the XObject API as standard for external media devices to collaborate through Lingo; and its interest as a standard achieved a lot of involvement from prominent and burgeoning media product companies through an ad hoc group called the Multimedia Association. The standardization with COM helped attract developers to creating a market for such plug-ins. Imaging Lingo[edit] Imaging Lingo was introduced with Director 8. There are some similarities to functions of image applications like Photoshop , that make it easy to create dynamic, code-based visual effects. Image manipulation was also added into ActionScript 3. As this included more sophisticated commands, Director was also updated to allow conversion between the BitmapData object and its own Image objects. Lingo was updated substantially to support the new 3D objects and now includes a full-featured set of 3D commands. Other languages[edit] These other languages are perhaps not as well known as the Macromedia language. A language called Lingo was released for software development under Windows. This version was designed as a compilable high level programming language. This language was named Lingo [7] and is significant because its makers successfully obtained a trademark in the UK. This language is still in production.

DOWNLOAD PDF MACROMEDIA DIRECTOR MX AND LINGO

Chapter 2 : Macromedia Director MX | Dr Dobb's

Find helpful customer reviews and review ratings for Macromedia Director MX and Lingo: Training from the Source at calendrierdelascience.com Read honest and unbiased product reviews from our users.

Features[edit] Director applications are authored on a timeline , similar to Adobe Flash. Director supports graphical primitives and playback controls such as video players, 3D content players, and Flash players. Director includes a scripting language called Lingo , and plug-in applications called Xtras , which are similar in functionality and design to ActiveX. Director supports a graphical user interface framework with basic controls, and allows interaction with external files and certain Windows APIs. Director supports many different image, audio, and video formats. Lingo programming language Director includes a scripting language called Lingo , and a suite of 2D image manipulation tools, referred to as "imaging Lingo". This subset of Lingo allows authors to perform advanced operations such as to bitblit. While a vast majority of users rely on the score timeline for the development of their work, a number of expert developers create stunning projects, such as games, that take advantage of the speed of imaging Lingo. These advanced projects typically use only 1 frame on the score timeline using Lingo to control animation and interaction. The 3D features were quite advanced for the time, unusual for an authoring environment. The 3D capability includes the ability to create geometry on the fly from code, hardware accelerated model display, and advanced lighting features. It also supports vector graphics and 3D interactivity through a Shockwave 3D file object. Xtras[edit] One of the most powerful aspects of Director is its extensibility, which is achieved through plug-in applications named Xtras. For example, there are Xtras for OS desktop manipulations creating folders, files, icons, shortcuts, registry editing and Shell control, dedicated text processing RegX , PDF readers, and many more. With Xtras, Director can be extended to support additional media types beyond those that the stock version of the software allows. These can be created by users or purchased from third party vendors. With the change in new versions of Director, Xtra developers need to modify their products to maintain ongoing support. With changing industry trends, many third-party Xtra developers have discontinued products and dropped support due the cost of development without the significant return. Publishing[edit] For online distribution, Director can publish projects for embedding in websites using the Shockwave plugin. Shockwave files have a. Other publishing options include stand-alone executable file called projectors, supported on Macintosh and Windows operating systems, and with Director 12, output for iOS. Early versions also supported execution of the 3DO console. The Director score timeline can also be exported as a non-interactive video formats, such as a QuickTime or sequence of images. Comparison with Flash[edit] The differences between Director and Flash have been the subject of much discussion, especially in the Director development community. Extensibility is one of the main differences between the two, as are some of the sundry codecs that can be imported. Director has tended to be the larger of the two, but that footprint has been part of its weakness. The download footprint of the Director Shockwave plugin was significantly larger than the Shockwave Flash download footprint. Additionally, Macromedia partnered with distributors such as Dell, Apple, etc. At that point in time â€” , broadband internet access was not the norm for most users, and the fivefold difference in size was significant. Animations were initially limited to the black and white of early Macintosh screens. The name was changed to "Director" in , with the addition of new capabilities and the Lingo scripting language in A Windows version was available in the early s. From to a competing multimedia authoring program appeared called mTropolis from mFactory. In mTropolis was purchased and buried by Quark, Inc.

Chapter 3 : Buying guide and subscription Plan | Adobe Director

Page 1. Lingo Dictionary Macromedia Director MX Page 2. This guide contains links to third-party Web sites that are not under the control of Macromedia, and Macromedia is not responsible for the content on any linked site.

Cons Steep learning curve. You can even create network connections and manage streaming audio and video using Flash Communication Server MX. Director comes bundled with a free copy of the Personal edition. And, when you double-click on any SWF-based element in Director, it will now launch the Flash authoring tool automatically. If you edit the Flash source file, you will find the SWF updated automatically when you return to Director. This is a nice feature; it eliminates a couple of steps when you edit Flash content. The combination of Lingo and ActionScript is extremely powerful. For those considering branching out from Flash into Director, however, the complex, proprietary Lingo scripting language and the overall complexity of Director will remain barriers. It makes more sense to expect a Flash expert and a Director expert to cooperate, rather than for one person to master both tools. On the other hand, MX gives experienced Director developers more access to the Flash universe. Objects Revealed As suggested above, Lingo objects can be quite complex. In particular, the 3-D objects that you import into Director may be composed of multiple objects and subobjects, each with its own properties and subproperties, all arranged in branching hierarchies. Figure 1 [click for larger image] The Object Inspector outlined in red provides visibility into various types of objects, including Lingo objects, lists, 3-D objects, and on-Stage objects "sprites". Director MX has taken a giant step forward in providing visibility into such complex systems. The new Object Inspector shown in Figure 1 lets you explore the structure of Lingo objects, objects on the Stage sprites , lists, and 3-D objects. It replaces the Watcher window of previous versions, but is much more powerful because it lets you see the structure of an object just by clicking on it. You can use the Object Inspector to familiarize yourself with the structure and properties of an imported 3-D object before you start doing any programming, for instance, or even before you click "play" to see its animation. The Object Inspector also displays changes in object properties while the movie plays, making it a valuable real-time debugging tool. Previously, getting such information about objects within Director took much more effort. You can change values in the Object Inspector directly, too, which is useful for debugging. The Watcher pane has a tabbed interface that lets you organize variables without taking up a lot of screen space. The Variable pane now contains three tabs, allowing you to display all, local, or global variables. Director MX saves screen real estate by morphing the Script window into a Debugger window whenever you choose to debug an error or when a breakpoint is triggered. In Script mode not Debugger mode , new buttons are visible to enable line numbering, automatic syntax coloring, and auto formatting, and to help you find Lingo commands, either alphabetically or according to categories. Among the important additions are several improvements that make it easier to create content that is accessible to people with hearing, physical, and visual impairments. For instance, the Speech Xtra plug-in converts text to speech without requiring a screen reader. At first glance, you might think that Director MX is just a "face-lift" for Director 8. In reality, it adds substantial new muscle, particularly for Lingo programmers. Its new Flash integration makes Director a possible next step for Flash developers, particularly those interested in creating advanced interactive games, sales presentations, or courseware.

Chapter 4 : Adobe Director - Wikipedia

The Paperback of the Macromedia Director MX and Lingo: Training from the Source by Gross at Barnes & Noble. FREE Shipping on $ or more!

Shockwave 3D Cubic VRs flying puppet: Very creative shockwave interactive animated stories 10 ways: Two very cool interactive image experiences lecielestbleu. Fun shockwave, must see flying giraffe submarine: Check out Twister Housing Virtual Tour: Allows change of floor tiles, wall painting, open doors. Science Alberta Foundation - Wonderville 3D: Read more about the development of this project here. Forums The following are Adobe hosted Director related forums. Director - General forum for all Director discussions. Director Basics - for novice users of Director. This forum covers discussions of installation, troubleshooting, and general usage. Director Lingo - for intermediate and advanced users of Director to discuss issues and questions regarding Lingo scripting functionality and usage. Other Forums Mediamacros Forum: Main forum of Russian Director community. There are other lists and groups but all are relatively inactive. Open-source community project, the goal of which is to develop the largest single repository of completely free, commercial quality, reusable software components for Director. Discuss any Lingo question that would be interesting to advanced Macromedia Director developers and share your Lingo scripts. Where anyone can ask any Director-related questions, and, share resources that will benefit the list. This list is for both web and graphic designers to share ideas, inspiration, professional knowledge, and methods for improving our work. This is ideal for artists, web developers, web designers, IT professionals, web site owners. German Mailing list, managed by Thomas Biedorf. Mailing list aimed is to promote cooperation between French-speaking users. Shockwave interactive tests, tricks and workarounds by Adam Kane. Director Developers Developers Dispatch: Personal blog covering Flash, Flex, Apollo and Shockwave. Mark Whybird - "This blog: Random things I find, and amazing sights I see. My interests span medical fascinations, computer coding with some bias for Macs , education, and the strange and weird things people do. There could be any of that and more in here. Art Related Blog - "I like to write about interesting art projects. Director Books I have co-written a book on Director so have to list it first. It was structured to be a book for absolute beginners so is not suited to everyone. It is a good idea to look around at various books on the market and see which suits your needs the best.

Chapter 5 : Publish interactive content for web and mobile | Adobe Director

Macromedia Director MX and Lingo: Training from the by Gross, Phil Email to friends Share on Facebook - opens in a new window or tab Share on Twitter - opens in a new window or tab Share on Pinterest - opens in a new window or tab.

Over time, you will adopt your own method. For instance, some programmers create one movie script member to hold all their movie handlers. Others break them up into several script members by category. Some people even create one script member per handler. If you do a lot of programming, you will find that your style adjusts over time. There is no right or wrong way to go about doing it. However, some basic programming guidelines can help you get started. The Lingo Programmer Lingo programmers are really of two types. The first type is someone who has a background in computer science or engineering. These programmers probably know languages such as C or Pascal, and took courses such as "Data Structures" and "Linear Algebra" in college. The second type is far more common. This is the graphic artist or multimedia producer. These programmers may have used presentation tools before, even Director, but have never used a programming language before. They have explored the basic range of Director and want to go beyond the basics. They are now ready to start learning Lingo. For both types of programmers, starting to learn Lingo can be difficult. For experienced programmers, Lingo takes care of much of the tedious work that they were used to in the past, but gives them control over graphics elements and the user interface. For graphic artists, Lingo can seem like lines and lines of text that stand between them and their end products. The important point to remember is that programming is an art. Programming languages, such as Lingo, provide a wide canvas for programmers to express themselves. Two programmers given the same task are almost certain to write two different programs. For experienced programmers, this means that programming in Lingo provides another type of canvas on which they can create. As an experienced programmer myself, I will even venture to say that Lingo will enable them to be more creative than before. For graphic artists, this means that Lingo is a new brush with which they can paint. I know many artists who have become Lingo programmers and used it as a way to create the art they envision. Programming as Problem Solving Programming is just problem solving. If you want to make a Director movie, think of it as a problem. Your goal is to find a solution. As with all problems, more than one step is usually needed to solve the problem. Then, you need to come up with a plan for solving it. To solve a programming problem, first define it. What, exactly, do you want to have happen? Saying "I want to animate a sprite" is not a well-defined problem. Saying "I want to move a sprite from the left side of the Stage to the right side" is better. What you really should be going for is something like "I want to move a sprite from position 50, to position , over the period of 5 seconds at the rate of 10fps. Imagine that your goal is to move a sprite pixels in 5 seconds at 10fps. At 10fps, 5 seconds will be 50 frame loops. So, you want to move the sprite 10 pixels per frame loop. Only by clearly defining the problem can you start to envision a solution. Solving Smaller Problems The key to writing a program in any language is being able to break it down. You start off with a large concept, such as "A quiz that teaches children about geography. You can bet that Lingo has no quizKidsOnGeography command. So, you break it into smaller parts. Maybe you want to have each question of the quiz show a map of the world with one country lit up. Then, three choices are presented as to what the country might be. So, now forget about the whole program and start to concentrate on just asking one question. But this part is also too big to tackle all at once. However, it has smaller parts. How does the map display a country? How do the three choices appear? How do you make sure one of the three choices is correct? A very small part of this might be just having a Lingo program that selects a random country out of a list of names. Now that you have broken the problem down this far, you might begin to program. The result is a handler that selects a random country and outputs it to the Message panel. Then, you continue to identify and solve other small problems in this way. Before you know it, you have a working program. The concept of breaking big problems into smaller ones is the most important aspect of programming. If you ever get stuck while programming in Lingo, it is likely to be because you have not broken the problem down into small enough pieces. Take a step back from what you are doing and decide how you can break it down before continuing. NOTE Did you read the last paragraph? It just might be the most important one in the book. Check

it out again even if you have. Make it your personal mantra while using Lingo. Many Ways to Do Things The first thing to realize about programming in just about any language is that there are usually many ways to accomplish the same task. This is particularly true of Lingo. For instance, suppose you want the movie to loop on the current labeled frame. You could use a go to the frame command in an on exitFrame handler. Or you could use go loop. Or, you could use go marker 0. All of these will work. They have differences when used in other situations, but for a single labeled frame, they will all behave exactly the same way. So why use one method over the other? You might choose one method because you use it more often, so you want to be consistent. You might choose one because you are using that same method elsewhere, where the slight differences in the commands will bring about different behavior. Or, you might just like how one sounds. In some cases, one solution is more elegant than another. For instance, there are several ways to change the volume of a sound. One method changes the volume setting for the entire movie, not just that one sound. So you may find that advantageous in some cases and a problem in others. In some cases, that technique is not compatible with certain sound cards. So you fall back on another technique which will work better in your situation. Calling one of these methods the "right" method is subjective. Instead, work toward getting your program to correctly solve the problem. Setting Up Your Script Members If you are creating a small Shockwave movie or projector, a useful way to organize your scripts is to have one movie script member and several behavior members. In fact, you might not need to have any movie scripts at all. Well-written behaviors can eliminate the need for movie scripts. If your movie requires mostly buttons, some behavior scripts can handle it all. Each behavior can be attached to a button and tell the movie what to do when it is clicked. A larger Lingo project might require a few movie scripts. Movie scripts are needed for items placed in the on startMovie handler, for instance, where commands need to be executed when the movie is initially run. Movie scripts can also hold handlers that are used by more than one behavior. If you want various behaviors to play a random sound, for instance, you might want them all to call your on playRandomSound handler that is stored in a movie script member. This saves you from having to include a similar handler in many different behavior scripts. It is always a good idea to keep movie scripts together in the Cast. The same goes for behaviors. There are exceptions, of course. Sometimes, you might want to place behaviors near the bitmap members that they usually control. Commenting Your Code Director is equipped with an automatic script color function. This function color-codes different types of words in your scripts.

Chapter 6 : Macromedia Director MX and Lingo: Training from the Source

Covers the latest version of Director: Macromedia Director MX; available in December for $1, (for new users) and $ (for users upgrading from versions or) Manageable, step-by-step projects are based on the curriculum developed and tested for use by Macromedia's authorized trainers.

Lingo is the programming language used in Director movies. Unless you stick to bare bones, PowerPoint-like presentations or linear animations, you need to learn Lingo to use Director to its full capabilities. Lingo code is stored in cast members called scripts. There are three different types of script members: In addition, other cast members, such as bitmaps, can have scripts embedded inside them. These are usually referred to as cast scripts. The difference between script types is not in what they look like or how they behave, but in when they act. Here is a summary: Cannot be assigned to specific sprites or frames. Controls the sprite or frame that it is assigned to. Only affects that one cast member, but affects every sprite instance of the cast member. A movie script is a global presence in a movie. If a movie script produces a system beep whenever the mouse is clicked, this script sounds the beep whenever the mouse is clicked anywhere in the movie. Thus the name movie script: It acts on the entire movie. A behavior script does nothing until it is placed on a sprite or in a frame script channel. When a behavior script is placed on a sprite, the Lingo commands inside the script are active only as far as the sprite is concerned. If you have a behavior that plays a beep when the mouse is clicked, for example, and you apply that behavior to a sprite, the beep sounds only when users click that sprite. Behavior scripts are sometimes called sprite or Score scripts for this reason. They act only on a sprite in the Score to which they are assigned. Behavior scripts can also be assigned to the frame script channel of the Score. When they are, they act like movie scripts, but only for the frame or frames to which they are assigned. Behaviors used this way are sometimes called frame scripts. Parent scripts are a different type of script. You would use parent scripts if you like object-oriented programming. This means that the code and the data exist in special objects that can be duplicated and modified. Cast scripts, on the other hand, are easy to use. You can create one by selecting a member, such as a bitmap, and clicking the Script button at the top of the Cast panel. This opens the Script panel and enables you to add a script to that particular member. Cast scripts act only on that one cast member. If you place a script with a cast member that makes the system beep when users click the mouse, that action affects only that one cast member when it is on the Stage. If you use that cast member more than once in the Score, the script that is a part of that cast member is active in all those places. Behaviors can accomplish the same tasks and are much more flexible. However, they do come in useful when you want to create some quick buttons without crowding a Cast with both the button members and the scripts that are assigned to them.

Chapter 7 : Macromedia Director MX Lingo

Learn how to write Lingo scripts to bring your Macromedia Director MX capabilities to their full potential. You will explore the various types of Lingo scripts and see when each should be used.

Chapter 8 : Dean's Director Resources - Lingo, 3D, Shockwave, Xtras

Learn how to write Lingo scripts to bring your Macromedia Director MX capabilities to their full potential. You will explore the various types of Lingo scripts and see when each should be used. This chapter is from the book Lingo is the programming language used in Director movies. Unless you.

Chapter 9 : Macromedia Director MX - Free download and software reviews - CNET calendrierdelascience

The calendrierdelascience.com file in the installation dir of Director initializes variables when you start the author aplication, not when you rewind-play the movie. It is a important point. Like Show 0 Likes (0).