MacPRO
o suplemento dos power users
Ano 1- Nº 7
Parte integrante da revista Macmania
Não pode ser vendido separadamente

ProNotas

QuickTime no Java, enfim

A Apple acabou de disponibilizar o **QuickTime for Java**, que vai permitir criar programas Java capazes de utilizar os recursos do QuickTime, como a capacidade de editar e criar exibir filmes, capturar áudio e vídeo e mostrar animações 2D e 3D. Em síntese, o que a Apple disponibilizou foi um programa que permitirá a qualquer usuário de Macintosh ou Windows o poder de rodar applets Java que utilizem recursos QuickTime. Para usufruir disso, o usuário deverá ter previamente instalados no seu micro o MRJ e o QuickTime. Usuários de Windows deverão ter pré-instalados o Windows Java Runtime Environment, atualmente na versão JRE 1.1.x or JRE 1.2, que deve ser baixado do site da Sun. Pecezistas e macmaníacos deverão também ter instalados em seus micros o QuickTime 3.0.2. Finalmente, todos deverão baixar o instalador do QuickTime for Java, que é pequeno e, compactado, ocupa 799 KB. Assim que você autorizar a instalação, um script colocará na pasta Extensions o arquivo **QTJava.zip**, que ocupa 1,1 MB. O **QTJava.zip** é uma extensão com função dupla: uma camada dá acesso ao QuickTime Applications Programming Interface (API) e a outra funciona como integradora entre as aplicações Java e o QT. Esta última integra a JVM ao QuickTime para que possam compartilhar eventos e espaço de exibição.
QuickTime for Java: http://qtj.apple.com/pub/QTJava.sit.hqx

Atualizado o software do FireWire

A Apple postou um update do seu driver do **FireWire**, voltado para os G3 da série azul e branca e para as placas FireWire PCI da empresa. A versão 2.0 instala duas extensões no Mac, Firewire Support e FireWire Enabler. A Apple diz que o update melhora a qualidade do vídeo capturado nas novas máquinas através do dispositivo DV.
FireWire 2.0: http://asu.info.apple.com/swupdates.nsf/artnum/n11316

Mac controla NT

Foi lançado o **netOctopus 3.0**, o programa de gerenciamento de sistemas da Netopia. Oferece, na nova versão, compatibilidade cross-plataform, auditoria do bug do ano 2000 e ferramentas de relatório HTML. Segundo a empresa, o produto permite que administradores de sistemas utilizem um Mac para gerenciar estações Windows NT. O cliente nativo Windows de 32 bits possibilita executar instaladores e alterar as configurações de PCs. A administração de TCP/IP Windows e as funções de gerenciamento de inventário de intranet também foram melhoradas na nova versão. A Netopia garante que todos os recursos do netOctopus podem operar automaticamente. O preço é baseado no número de máquinas de uma rede, girando em torno de US$ 55 a US$ 65, dependendo da quantidade de cópias.
Netopia: www.netopia.com

Free? Public? Open?

Mac OS X Server versus open source: Um guia para a sua desorientação

por Rainer Brockerhoff

Uma das surpresas do anúncio do Mac OS X Server foi o projeto **Darwin**. Através de duas páginas na Internet – www.apple.com/darwin e www.publicsource.apple.com – a Apple está publicando o código-fonte de toda a infra-estrutura do Mac OS X Server. Isso compreende a versão Apple do kernel Mach 2.5, a camada UNIX BSD 4.4, conforme a adaptação da Apple, e alguns componentes adicionais, como drivers de porta serial e acesso a drives HFS. Publicaram também um conjunto de links para componentes de domínio público, como o servidor Apache e diversas ferramentas UNIX. E o mais importante: um documento chamado **Apple Public Source License** (APSL) explica as condições sob as quais a Apple permite o download e utilização desses códigos fontes.

Houve manifestações de apoio e de censura à política de public source da Apple

Num golpe de relações públicas, Eric S. Raymond, presidente da Open Source Initiative (www.opensource.org) e autor do influente texto "The Cathedral And The Bazaar" (www.tuxedo.org/~esr/writings/cathedral-bazaar), participou do lançamento do Mac OS X Server e falou da sua aprovação da iniciativa da Apple. Segundo ele, a APSL foi redigida com a sua colaboração.

Imediatamente, houve grandes manifestações de apoio por parte dos desenvolvedores Apple e de censura por parte de outros líderes do movimento "open source". Richard M. Stallman, famoso fundador do Projeto GNU e líder da "Free Software Foundation" (http://gnu-dist.gnu.org/philosophy/free-software-for-freedom.html) pronunciou-se fundamentalmente contrário aos aspectos filosóficos da APSL. Um site dedicado a desenvolvedores do GNU/Linux (http://slashdot.org) publicou milhares de comentários, na maioria negativos, sobre a iniciativa da Apple. Mais importante, Bruce Perens, autor primário da definição oficial de "open source", publicou comentários indicando que a APSL, na redação atual, não se encaixa na sua definição, mas fazendo sugestões para que a Apple fizesse adaptações. *(Leia a definição em português em* http://das.harvard.edu/~pvs/osd/po-osd.html *e* http://das.harvard.edu/~pvs/osd/po-osd-rationale.html*).*

Bom, para quem, como eu, sempre conviveu com a realidade de desenvolver para uma plataforma não-aberta e dedicada ao espírito do resguardo da propriedade intelectual através de segredos industriais (parcialmente desvendados para seletíssimas pessoas através das famosas NDAs, declarações de não-divulgação), tudo isso ficou muito complicado. Vamos ver se consigo explicar um pouco as informações que consegui levantar.

As bases históricas são as seguintes:

• No início da informática, software era grátis. A IBM e outros fabricantes dos pesadões e milionários mainframes forneciam o software necessário para operar seus equipamentos sem custo adicional, geralmente em forma de código-fonte... Também, era tão pouco para os padrões atuais! Mais tarde, quando os softwares ficaram mais complexos e começaram a ser vendidos, primeiro restringiram os fontes aos seus clientes universitários e finalmente os trancaram a sete chaves. Quando surgiram os computadores pessoais, esse ciclo foi repetido. ▶

Free? Public? Open?

continuação

É interessante notar que foram Bill Gates e Paul Allen, fundadores da Microsoft, que iniciaram a era do software comercial quando começaram a vender o seu primeiro BASIC em fita de papel e iniciaram vigorosos esforços para combater o que consideravam pirataria.

• Quando a AT&T foi autorizada a entrar no mercado de hardware e software, ela passou a vender o seu sistema operacional UNIX. Grupos originários do MIT também começaram a formar empresas de software, e Richard Stallman, um dos últimos hackers que resistiu a essa comercialização, fundou a Free Software Foundation e começou a escrever, quase sozinho, o GNU (abreviação de "Gnu's Not UNIX"), que seria um sistema operacional padrão UNIX, mas completamente livre, grátis e acompanhado do seu programa fonte. Stallman ainda desenvolveu a licença de uso para o GNU (geralmente chamada GPL) e um conceito de propriedade intelectual, o *"copyleft"*, para garantir que seu trabalho não pudesse ser cooptado pelo inimigo – as empresas de software comercial.

• Mesmo com os incríveis poderes intelectuais de Stallman e seus companheiros, o GNU não chegou a ser lançado como sistema operacional completo e utilizável. Linus Torvalds, programador finlandês, conseguiu fazer a parte que faltava, o *kernel*, que ele chamou de Linux.

O conjunto do Linux com as ferramentas GNU teve enorme popularidade e, hoje em dia, o Linux (mais propriamente chamado de "GNU/Linux") tem uma grande comunidade de usuários que dão suporte, programam modificações, ajudam na depuração e fornecem software sob a GPL ou licenças derivadas.

• Como parte dessa popularização surgiram empresas como a Red Hat, que conseguiram tirar proveito comercial da distribuição do GNU/Linux, sem porém deter direitos sobre a propriedade intelectual. A tensão entre os radicais da linha Stallman, dos pragmáticos seguidores de Torvalds e dos interesses comerciais fizeram surgir entidades, como a Open Source Initiative, que tentam formalizar as definições do que, afinal de contas, quer dizer "free" ou "open" ou "public" software. A maior ambiguidade é que o termo *free* não quer dizer necessariamente "grátis" – o software "free" pode ser vendido, mas o que é considerado fundamental é a liberdade do usuário de pegar o código-fonte, adaptá-lo ao seu bel-prazer e redistribuí-lo (grátis ou não) do modo que quiser, desde que não negue as mesmas liberdades aos seus próprios usuários.

Analisando a APSL

De volta à APSL. Essa licença diz, muito resumidamente, que qualquer pessoa ou empresa pode fazer download dos fontes que a Apple está publicando e usá-los ou modificá-los para estudos e pesquisas particulares.

Vamos supor que você pegou o software e introduziu pequenas alterações, digamos, para fazê-lo funcionar com um mouse de dois botões. Se você usar o software modificado na sua empresa ou redistribuí-lo para outros interessados, você tem que se sujeitar às outras exigências da APSL; em especial, você só pode redistribuir suas alterações se mantiver intactas as provisões originais da APSL; deve publicar os fontes alterados na Internet e notificar a Apple do local de publicação; e ceder à Apple o direito de reutilizar as suas alterações. Além disso, se houver reclamação judicial de algum terceiro alegando que os fontes publicados violam sua propriedade intelectual, a Apple pode tentar contornar essa violação, em último caso revogando a APSL para a parte afetada do código-fonte. *(Nota: para referência exata, leia o texto original da APSL. Esta é apenas a minha interpretação pessoal.)*

Foram Bill Gates e Paul Allen que iniciaram a era do software comercial

A Apple poderia estar querendo sugar o trabalho de voluntários?

Há objeções vindas de diversas partes. Os mais radicais se rebelam contra o fato da Apple ter publicado apenas uma parte do Mac OS X Server e contra a intenção da Apple de reincorporar as modificações em versões futuras (e pagas!) desse software. A cláusula de revogação é considerada por eles uma prova da intenção da Apple de sugar o trabalho de voluntários em depurar o seu software e depois poder retirá-lo do domínio público com uma pretensa desculpa de violação de direitos.

Outros se preocupam que, na hipótese do fechamento da Apple, uma interpretação literal da APSL exigiria o congelamento completo do software, impedindo modificações posteriores.

A objeção de Bruce Perens é a mais ponderada. Segundo ele, a APSL não define detalhadamente o que acontece em caso de reclamação judicial. Uma interpretação ampla permitiria à Apple retirar de circulação todo o software do projeto Darwin, não apenas a parte afetada pela reclamação. Segundo ele, a Apple se mostrou interessada nas suas sugestões e pretende publicar, periodicamente, versões mais eficazes da APSL, de acordo com as sugestões recebidas.

Qual é a vantagem, afinal?

OK, mas agora você deve perguntar: afinal de contas, qual é a vantagem que levamos com o projeto Darwin?

Para a Apple, há poucas desvantagens, supondo que esses probleminhas com a APSL sejam resolvidos. Ela está divulgando uma certa parte do seu sistema, mas na verdade apenas uns 10%, porque quase tudo já estava publicado como parte dos projetos Mach e BSD. Uma certa parte dos fontes foi alterada primeiro pela NeXT, e depois pela Apple, para adaptar os fontes às suas plataformas. Essas alterações são pequenas mas valiosas.

A grande vantagem é que reduzem-se muito os custos de suporte da Apple. O Mac OS X Server hoje roda em poucos modelos de Mac, o suporte a periféricos é restrito e faltam no pacote vários tipos de softwares servidores. Se a comunidade "open source" aderir, produzirá contribuições importantes para adaptar as camadas básicas do sistema a outros ambientes – inclusive ambientes não-PowerPC – e a periféricos muito diversos. Acho que é esse o motivo do cuidado da Apple em restringir o seu suporte apenas aos novos G3.

A Apple tem liberdade de incorporar essas modificações em versões futuras do Mac OS X, não apenas da versão Server. Isso irá enriquecer muito o produto. A Apple, enquanto isso, pode se concentrar em desenvolver suas tecnologias reservadas: a interface de usuário, o QuickTime, o WebObjects. E, se alguém portar o Darwin para outra plataforma de hardware – Pentium, Alpha, o que for – a Apple tem a opção de subitamente implementar essas tecnologias na nova plataforma com pouquíssimo esforço. E, sutilmente, essa liberação do fonte também afasta a atenção de correntes anti-monopolistas.

A Open Source Initiative busca definir o que quer dizer "free", "open" ou "public"

E o que vale para nós?

Para nós, desenvolvedores, isso também é interessante. Uma das grandes dificuldades que enfrentamos hoje é que o coração do Mac OS é de acesso restrito. Vai ser muito mais fácil debugar, afinar desempenho e investigar técnicas novas com o fonte do sistema básico aberto a todos, mesmo se não tivermos interesse em introduzir alterações. Mesmo a Apple vai ter incentivos para definir melhor as APIs das suas tecnologias reservadas, para evitar os efeitos colaterais e truques não-implementados de hoje... Nada de ter "ganchos" reservados para o QuickTime, por exemplo, fazendo coisas que outros desenvolvedores não podem entender ou duplicar.

Para o usuário, a médio prazo, o efeito só será benéfico: software melhor e mais barato, tanto da Apple quanto de terceiros. M

RAINER BROCKERHOFF
rainer@ez-bh.com.br

“A Apple não está sendo insincera”

Entrevista a Rainer Brockerhoff

Consegui entrevistar Bruce Perens, principal autor da definição oficial de “open source”, que gentilmente me autorizou a publicar vários comentários adicionais sobre a controvérsia da APSL.

Rainer: Por que algumas correntes da comunidade “open source” reagiram com toda essa paranóia à iniciativa da Apple? Se não tivessem mencionado o termo, isso não seria, de qualquer modo, interessante para os usuários e desenvolvedores Apple?

Bruce Perens: A Apple não está sendo insincera, como alguns querem insinuar. Publicar os fontes de um modo interessante apenas aos atuais desenvolvedores Apple ou NeXT não teria atendido seus propósitos. Eles sentiram como o Windows está perdendo o “mind share”, ou interesse, dos desenvolvedores, que estão debandando para o Linux e entrando para a comunidade. É claro que a Apple gostaria de capturar esse interesse, e é difícil ganhar de um software “free”, então eles estão se juntando a nós. Temos nossos padrões estabelecidos para software e precisamos de algumas pequenas alterações para que a APSL funcione dentro da nossa comunidade. Eles estão muito interessados em trabalhar conosco, portanto não acho que essas mudanças sejam problemáticas.

Rainer: Eu li comentários de que a APSL é especialmente desvantajosa para desenvolvedores de outros países, que ao aceitá-la automaticamente se sujeitariam às leis americanas de exportação e patentes de software, e também levariam desvantagem no cancelamento do projeto. É verdade?

Bruce: Não é o cancelamento do projeto que é problemático, mas o cancelamento da nossa licença de continuar a usar o software... O código-fonte publicado poderia continuar a ser usado sem o apoio da Apple; afinal de contas, não precisamos do suporte de uma empresa para usar o Linux. A Apple precisa modificar a APSL para que o cancelamento seja menos ambíguo. Do jeito que está, o nosso direito de distribuir ou usar o software poderia, possivelmente, ser completamente cancelado. No caso de uma questão judicial, queremos que eles removam apenas o mínimo necessário para evitar o uso da patente contestada. A licença é pior para estrangeiros porque a maioria dos países não tem legislação sobre patentes de software, como temos aqui. É o mesmo caso da exportação de software criptográfico ou que possa ter aplicações militares... A APSL força programadores de outros países a aceitarem essas restrições. Outras empresas que publicaram licenças similares delegaram ao governo a aplicação das leis, mas a Apple decidiu incorporá-las na sua licença. Muitos aqui acham que essas leis foram uma má decisão do nosso governo e não gostariam de vê-las exportadas ou incluídas em licenças, onde elas podem persistir mesmo que consigamos derrubá-las por aqui.

Rainer: É estranho ver essa discussão de minúcias e vírgulas numa licença que supostamente “libera” um software. Não parece um contra-senso considerar a hipótese de a Apple querer perseguir ou prejudicar pessoas que irão ajudá-la? Não seria um desastre de relações públicas?

Bruce: Não acho que as pessoas que estão na Apple fariam algo tão hostil. Fazemos contratos para que o espírito de um acordo continue o mesmo depois que as pessoas que fizeram o acordo saiam dos seus cargos. Se não tivermos uma boa licença agora, caso futuramente a Apple fosse adquirida pela Microsoft ou outra empresa, as coisas poderiam mudar substancialmente. O público é muito volúvel e nem sempre podemos contar com a repercussão pública para nos proteger. Então gostaríamos que essa licença, que na verdade é um contrato da Apple com todos nós, fosse muito precisamente redigida. Não supomos que coisas ruins irão acontecer, mas queremos garantir que não possam acontecer.

Rainer: Minha leitura pessoal da APSL é que eles estão mais se prevenindo contra ações dos próprios acionistas (que podem argumentar que estão divulgando segredos industriais indevidamente) do que querendo restringir os direitos da comunidade de desenvolvedores.

Bruce: Eles estão se resguardando contra ações de violação de patentes e das restrições de exportações do governo; estão deixando uma porta de escape para sair de uma ação retirando a licença. Os desenvolvedores precisam de uma garantia de que a licença não será cancelada por outros pretextos, o que o texto atual não exclui.

Rainer: Mas a Apple teria elementos para ir atrás de uma firma fundo-de-quintal em outro país? Ou poderiam fazer outras pressões?

Bruce: Acho que bastaria que ameaçassem fazer isso... E não deve ser fácil ser responsável por uma ameaça de restrições comerciais ao seu país!
Mas é melhor evitar problemas e não violar a lei desde o princípio, porque quem o fizer poderá ser escolhido como exemplo para os outros, mesmo sendo pequeno.

Rainer: Eu li a GPL e não achei as cláusulas de cancelamento substancialmente diferentes da APSL, exceto que a APSL tenta, logicamente, preservar os interesses da Apple em primeiro lugar.

Bruce: A GPL apenas diz que você não pode redistribuir o software contestado. A Apple, além disto, pode forçar você a destruir as cópias que já tem. No caso de uma ação, o governo ou seja quem for teria que ir atrás de cada licenciado GPL individualmente, o que é muito mais difícil.

Procurei também me informar sobre a posição de Eric Raymond. Eric foi um dos responsáveis por convencer a Netscape a abrir o fonte do seu browser, no ano passado.
Quando lhe mandei as mesmas perguntas que mandei a Bruce Perens, ele respondeu muito concisamente: segundo ele, a paranóia de alguns membros da comunidade “open source” era inevitável e não há desvantagem especial para desenvolvedores fora dos Estados Unidos. Ele também não vê diferença significativa entre as licenças GPL e APSL, levando em conta que a Apple tentou se proteger especificamente. M

CineMotion

Nunca foi tão fácil e barato fazer uma imagem de vídeo ficar parecida com um filme a 24 quadros por segundo

por João Velho

A empresa de software **DigiEffects** continua investindo a sério na área do *film look* para vídeo, e para essa linha de produtos lançou recentemente o CineMotion, uma nova coleção de plug-ins para o After Effects.

Criado para expandir os recursos de simulação de filme telecinado do After Effects, o CineMotion funciona como um complemento do CineLook, também da DigiEffects. Enquanto o CineLook atua intensivamente sobre parâmetros da imagem de vídeo como granulação e cor, o CineMotion explora parâmetros temporais da seqüência de frames.

Plug-ins oferecem novos recursos para simular o "film look" no vídeo digital

Film Motion em destaque

O CineMotion é constituído de dez plug-ins, com destaque para o Film Motion. Sua história começa na ISFX, que também atua no mercado de plug-ins para o After Effects. Depois de desenvolvido, o Film Motion foi oferecido à DigiEffects, que o adquiriu e entrou com os outros nove plug-ins para fechar o pacote. Todo o segredo do Film Motion está em imitar o processo *3:2 pulldown (leia o box na próxima página)*, que ocorre quando um filme é transferido para vídeo através de telecine. Em geral, essa transferência produz uma sensação de movimento de imagem mais suave do que a obtida em cenas captadas originalmente em vídeo.

Além dos presets para placas e sistemas não-lineares, há nove ajustes no Film Motion, com destaque para o que determina o modo de simulação da conversão para 24 quadros por segundo. Os oito modos oferecidos por esse ajuste definem o algoritmo usado para a simulação. De acordo com o modo escolhido, técnicas de *motion blur* e *frame blending* permitem uma afinação precisa do resultado da simulação.

Dependendo das características do movimento interno da imagem original, os presets do Film Motion podem não ser suficientes para se obter o melhor efeito. Em algumas situações de movimento linear, por exemplo, o render da imagem pode gerar eco ou batimento. Nesses casos, a solução é experimentar vários modos de telecine com ajustes diferentes até chegar a um resultado aceitável.

A interface do Film Motion dá uma idéia da sofisticação dos seus recursos, com nove ajustes diferentes

Os filtros do CineMotion

Os outros plug-ins do CineMotion afetam novos aspectos não-temporais da imagem, a exemplo do CineLook. O filtro Adaptive Noise introduz ruídos em áreas de pequenos detalhes com ajustes por canais de cor RGB, monocromático e alpha. O Banding Reducer procura áreas de baixo detalhe ou de tonalidades parecidas e rearranja os pixels dessas áreas para obter um efeito mais suave. O filtro Grain Reducer minimiza o grão e o ruído de uma cena captada em filme ou vídeo, e para isso usa até dez algorítmos com diferentes métodos de redução. Para diminuir o *flicker* provocado por áreas de alto detalhe, o Interlace Aliasing Reducer suaviza a imagem onde há efeito de alta freqüência. O LetterBox é útil para criar rapidamente aquelas barras horizontais em cima e em baixo, imitando a relação de aspecto de 16:9 do cinema.

Fechando o pacote, mais quatro filtros poderosos e similares entre si. Os dois primeiros, Selective HSB Noise e Selective RGB Noise, adicionam ruído para cada canal em áreas específicas da imagem, selecionadas pelos métodos HSB ou RGB. Os outros, Selective HSB Posterize e Selective RGB Posterize, reduzem o número de cores da imagem controlando quantos níveis de detalhe serão permitidos para cada canal, também baseados em HSB ou RGB.

Conclusão

Quando usado junto com o CineLook, a DigiEffects alerta para aplicar o Film Motion primeiro na ordem do render. À exceção do FilmMotion, os outros acrescentam técnicas boas para coisas que já podiam ser feitas razoavelmente bem com outras ferramentas. O problema é o tempo de render para avaliar cada tentativa, que exige a CPU mais rápida possível, no mínimo um G3, ou então a placa aceleradora ICE. Mas a estrela principal do pacote, o próprio FilmMotion, faz a diferença e justifica os US$ 295 que constam como preço de lista. Os usuários registrados do CineLook ganham um desconto, com o preço final caindo para US$ 195. Sem dúvida, uma pechincha que vale a pena. **M**

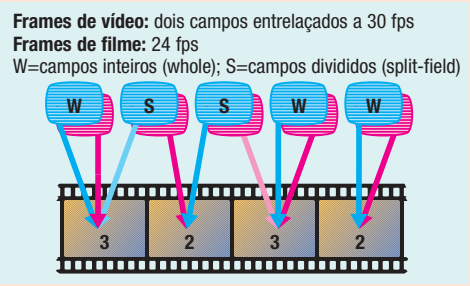
JOÃO VELHO
jvelho@cyberhome.com.br
É sócio da Digiworks, empresa de criação de projetos de animação, vinhetas e pós-produção de vídeo digital.

CineMotion

DigiEffects: www.digieffects.com
Preço: US$ 295 (upgrade US$ 195)

Destelecinando o vídeo

O processo **3:2 pulldown**, ilustrado ao lado, é uma técnica de telecinagem para fazer a conversão dos 24 quadros (frames) por segundo do filme de cinema para 30 quadros/60 campos por segundo da TV e do vídeo. O termo "3:2" se refere ao padrão em que ocorre essa conversão, com o mapeamento repetido de cada 4 quadros de filme em 10 campos (5 quadros) de vídeo.
Pela técnica 3:2, alternadamente, um quadro de filme é reproduzido em 2 ou 3 campos de vídeo. O resultado é que de cada 5 quadros de vídeo, 2 são do tipo "campo dividido" (*split-field*) e 3 do tipo "inteiro" (*whole*). A ordem dos quadros com campo dividido ou inteiro no início de um take determina se ele está numa das 5 possibilidades de fase do 3:2 pulldown.
O plug-in Film Motion faz um pouco o raciocínio inverso do telecine, e simula uma conversão do material original de vídeo a 30 fps para 24 fps depois de telecinado.

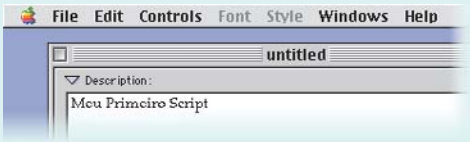


Alô, mundo dos scripts!

Curso de AppleScript, parte 2

por Maurício L. Sadicoff

Hoje você vai dar seus primeiros passos no AppleScript, como prometi no mês passado. Para começar, já que vamos aprender a escrever, talvez seja bom termos um dicionário. Que tal? Pois bem, como sou bonzinho vou contar onde está o dicionário... Adivinhe? Dentro do seu computador!
"Como é que é? Quer dizer que, além desse tal de AppleScript que tenho aqui no meu Mac, esse povo da Apple ainda me deu de presente um dicionário?" É isso mesmo. Tá vendo só quanta coisa a Apple faz pra você? E você, que achava que tudo que vinha de Cupertino (a cidade-sede da Apple na Califórnia) era iMac...
"Tá, tá, mas cadê o dicionário?" Está em cada programa. Continue lendo, que antes de terminar eu explico melhor.
Para começar, se vamos mexer com AppleScript, é bom abrir o **Script Editor**, o programinha que permite que você crie seus belos e poderosos scripts. Se você leu a coluna do mês passado, já sabe onde fica; se não leu ou não se lembra, dê uma olhada na pasta Apple Extras, dentro do seu hard disk.
A primeira coisa que você vê é uma janela um pouco diferente das janelas que costuma encontrar. Essa janela tem um espaço reservado para uma *Description* que, pra você que não fala francês, significa descrição. Está lá para que, quando fizer um script, você escreva um pouco sobre o que vai fazer com ele, de tal forma que, se algum louco resolver olhar o seu script, ou se você resolver mexer nele (no script, não no louco) algum tempo depois, o louco ou você vai poder entender o que diabos está acontecendo.
Então, como, além de aprender onde estão os dicionários, você quer escrever seu script, digite uma descrição... Sugerimos algo bem original, como você pode ver no screen shot:

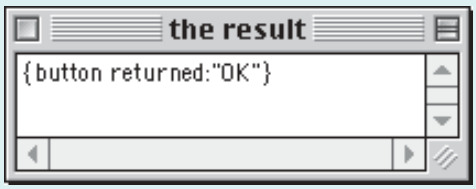


Você, a essa altura, já notou também os quatro botões: Record, Stop, Run e Check Syntax. Não se preocupe com eles agora. Na hora certa eu explico para que nós, rascunhadores de script, precisaremos deles.

Meu primeiro script

Como todo grupo de seres humanos com algo em comum, programadores (e rascunhadores de script, por extensão) têm algumas tradições. Uma delas é que, sempre que você aprender uma linguagem de programação, seu primeiro programa vai ser pra dizer "alô, mundo" *(hello world)*. Então, digite no espaço embaixo dos botões: `display dialog "Alô, mundo dos scripts!"`

Agora, para ver no que dá, aperte o botão Run (eu disse que avisaria quando você ia usar, né?).
"Uooooooooou! Daonde saiu essa janela?
Fui eu que fiz?"
Foi você mesmo, Joãozinho! Viu só que barato? Fácil assim, né? E você achava que devia dar um trabalhão programar...
Vou contar um segredo. Programadores são preguiçosos por natureza. São gente que tem tanta paúra de trabalho que inventaram uma máquina que faz o trabalho por eles!
Agora é que entra a primeira diferença entre quem usa Mac OS 8.5 e quem usa os sistemas anteriores. A nova versão do Mac OS tem algumas melhorias significantes no AppleScript, que facilitam o processo de *debugging* dos scripts. Debugging é um termo em inglês que denota o processo de encontrar e eliminar bugs nos programas e que foi conseqüentemente estendido para os scripts.
Se você está em qualquer versão do Mac OS anterior à 8.5, tanto faz apertar OK ou Cancel. No Mac OS 8.5, se você apertar o botão OK, recebe como resposta uma janela extra com a frase {button returned: "OK"}.
Essa janela, maravilha das maravilhas, conta para você, que está testando o script, que o botão apertado foi o OK. Pode não parecer claro agora para que isso serve, mas mais na frente você vai ver como ajuda.



Agora, antes que eu me esqueça, vou mostrar como fazemos para encontrar o dicionário. Cada programa tem seu próprio dicionário. "O que diabos é esse dicionário?", você pergunta. É a descrição dos termos de AppleScript que um programa suporta e dos termos que o programa permitirá que você use quando escrever um script. É claro que nem todos os programas suportam AppleScript, mas a grande maioria dos programas que você usa tem suporte a ele. Pois é, isso estava lá o tempo todo e você não sabia.
Para ver o dicionário do Netscape ou do Internet Explorer, por exemplo, basta usar o comando Open Dictionary do menu File no Script Editor e escolher a aplicação que você quer, seja o Netscape ou o Explorer. Se você abriu o Netscape, isso é o que vai ver:

No mês que vem, vamos escrever scripts mais complicados, que fazem mais do que mostrar uma janela. Quem sabe, eu posso resolver até explicar um pouco mais sobre dicionários... M

MAURÍCIO L. SADICOFF
Está fazendo Mestrado em Engenharia de Computação na Flórida. Ele é alto, moreno e tem olhos castanhos. Seus hobbies são futebol, equitação e discutir tópicos avançados de programação na MacDev-BR. Cartas para a redação!