

Pablo Muñoz Sánchez (conhecido como Pablito's)

Copyleft (CC) 2005 Sayans Traduções Tradução por Marciso Gonzalez (RiFF) Edição/Revisão por Israel Crisanto (Fallen_Soul) Copyleft(CC) 2005 Translations Center

Esta obra está sobre uma licença de *Reconhecimento-NãoComercial-SemObraDerivada* da Creative Common. Isso que dizer que pode copiar, distribuir e comunicar publicamente a obra segundo as condições desta licença. Para ver uma cópia desta licença, http://creativecommons.org/licenses/by-nc-nd/2.1/es/.

Agradecimentos

Quero mostrar meus agradecimentos à todas aquelas pessoas que me ajudadaram durante o meu longo e duro cominho como romhacker. Sem elas, nem o presente manual nem minhas traduções poderiam ter visto a luz. Assim, desejo mostrar meus mais sinceros agradecimentos a Andrés Botero Castro(KaOSoFt) por ter revisado tão exaustivamente este manual com o propósito de que esteja livre de qualquer erro. Qualquer outra falha não será mais que responsabilidade minha. Por último, gostaria de dar o reconhecimento que merecem todas as pessoas que escreveram todos os documentos que recomendo ao longo deste manual e a todos aqueles programadores que tem facilitado tanto o trabalho de um romhacker criando os programas utilizados no ROMHacking.

Correio eletrônico e página WEB

Pode entrar em contato comigo através deste e-mail: pms_sayans@hotmail.com Visite minha página na web para mais informações: http://sayans.todotradus.com/

Nota de abertura

A imagem usada foi pega de <u>http://www.centenarioelquijote.com/</u> respeitando sua licença Creative Commons.

Índice de Conteúdo

PRÓLOGO	
CAPÍTULO 1: ASPECTOS TRADUTOLÓGICOS	5
1.1. Requisitos para traduzir	
1.2. O papel do contexto	
1.3. A invisibilidade da tradução	6
1.4. A necessidade de adaptações	
1.4.1. Conselhos Ortográficos	8
1.5. Os jogos de palavras	9
1.6. Dicionários e outras fontes de informações	
1.6.1. Dicionários recomendados.	
1.7. Erros mais comuns nas traduções	
CAPÍTULO 2: INTRODUÇÃO AO ROMHACKING	
2.1. Programas necessários e outros requerimentos	13
2.2. Conceitos básicos: código hexadecimal e tabelas	13
2.2.1. Código hexadecimal	
2.2.2. Tabelas	
2.3. Procura por textos em um arquivo binário	14
2.4. Fazer uma tabela	
2.5. Para saber mais	
CAPÍTULO 3: EDIÇÃO DE TEXTO	
3.1. Programas necessários e outros requerimentos	19
3.2. Manuseio do Translhextion	19
3.3. Completando a tabela	21
3.4. Edição do texto	
CAPÍTULO 4: EDIÇÃO DE GRÁFICOS	
4.1. Programas necessários e outros requerimentos	25
4.2. Editando gráficos	
4.2.1. Gráficos visíveis	
4.2.2. Gráficos que utilizam um tamanho diferente do padrão	
4.3. Alterar a largura de uma fonte de largura variável(VWF)	
4.4. Alterar a largura de uma fonte de altura variável(VWF)	
CAPÍTULO 5: PROCURA POR TEXTOS COMPRIMIDOS	
5.1. Programas necessários e outros requerimentos	
5.2. Compressão DTE e MTE	
5.2.1. Averiguar a compressão DTE e MTE	
5.2.2. Alterar a compressão DTE e MTE original	
5.3. Procurar texto em modo 16 bits	
5.4. Para saber mais	

CAPÍTULO 6: PONTEIROS	
6.1. Programas necessários e outros requerimentos	
6.2. Introdução aos ponteiros	
6.3. Como encontrar a tabela de ponteiros	44
6.4. Outras formas de achar ponteiros	
6.5. Para saber mais	
CAPITULO 7: SCRIPTS	
7.1. Programas necessários	
7.2. Scripts	
7.2.1. Definição	
7.2.2. Vantagens e inconvenientes	
7.3. Extrair textos	50
7.4. Inserir textos	51
7.5. Recalcular ponteiros	
7.6. Para saber mais	53
CAPÍTULO 8: A TRADUCÃO DE JOGOS DE PSX	55
8 1 Programas necessários e outros requerimentos	
8.2 Criação de uma imagem ISO	
8.2. Enação de arguivos na ISO	
8.4. Procura de arquivos com o texto a traduzir	
8.5. Procura e edição de gráficos	
8.5.1 Gráficos em formato PAW	50 50
8.5.1. Graficos em formato TIM	
8.5.2. Oraneos em formato Thvi	
CAPÍTULO 9: INTRODUÇÃO AO ASM	
9.1. Programas necessários e outros requerimentos	63
9.2. Instruções básicas	63
9.3. Procurar e modificar textos que não se encontram do modo normal	64
9.4. Introdução avançada: bucles.	
9.5. Rastrear códigos: outras modificações ASM	71
9.6. Para saber mais	
APENDICE: CRIAÇÃO E APLICAÇÃO DE PATCHS	
1.1. Programas necessários e outros requerimentos	
1.1.1. Formato IPS	
1.1.2. Formato PPF	
1.2. O checksun.	
1.2.1. Corrigir o checksun de jogos de SNES	
1.2.2. Corrigir o checksun de jogos de Mega Drive	
1.3. Patchs IPS.	
1.3.1. Criar patchs	
1.3.2. Aplicar patchs	79
1.4. Patchs PPF	
1.4.1. Criar patchs	
1.4.2. Aplicar patchs	80

PRÓLOGO

Muito já aconteceu desde que, em meados do ano de 1997, um grupo de aficcionados tornaram público um patch cujo propósito era modificar uma quantidade considerável de bytes de um arquivo, de tal modo que, em vez de mostrar na tela alguns caracteres ilegíveis para muito desconhecedores do idioma japonês, mostrasse o texto perfeitamente em inglês. Tais aficionados, sob o nome de RPGe, haviam traduzido o jogo de Final Fantasy V do japonês para o inglês, depois de modificar exaustivamente os arquivos binários pertencentes a este jogo. Naquela época a emulação de consoles como o NES, Megadrive/Genesis e do SNES começava a fazer seus anseios e sonhos de desfrutar de jogos que nunca sairiam do japonês(e do inglês no caso da Espanha) deixava pouco a pouco se conceber como tal. Assim, jogar Final Fantasy V deixava de ser uma fantasia graças ao aparecimento dos emuladores (muito distantes de serem perfeitos, isso sim) a barreira lingüística sugere um novo impedimento na hora de passar um bom tempo. O RPGe havia dado, um passo a mais ao permitir ao jogador compreender os indecifráveis diálogos japoneses. O ROMHacking havia nascido.

Este manual não pretende oferecer ao leitor uma quantidade enorme de informação desnecessária para traduzir um jogo. Portanto, segue com um ritmo rápido evitando um excesso de verbosidade, mas não passando por cima de detalhes relevantes no princípio que podem ser fundamentais em um projeto. Tão pouco tem como objetivo fazer uma retrospectiva de toda a história do ROMHacking, nem discutir sua legalidade, muito menos de teorizar uma realidade prática confundindo ao leitor. Por último, já que por experiência própria sei que pode acontecer, o leitor não deve cair no erro de pensar que é inútil experimentar algo que pode falhar e que precisa de tempo, assim como ignorar aquilo que já sabe. É por isso que lhe digo agora que ainda que frustração e o desânimo nos invadam, e ainda que nunca consiga um resultado satisfatório, sempre haverá aprendido algo que, ainda que passem os anos, estará sempre na memória. Simplesmente precisa dos estímulos apropriados para invocar tudo o que foi aprendido.

Espero que com êxito de sem deixar nada no tinteiro, tudo o que aprendi durante todos esses anos no mundo do ROMHacking. Muitos programas que facilitam o trabalho aparecerão desde a primeira versão deste tutorial, assim como uma infinidade de outros tutoriais que nos tem sido fruto da experiência. Não há o que duvidar de que é este manual que nos dará a clave da vitória diante de um caminho que reúne dificuldades dadas em outros vídeo-games.

Prólogo

Espero que meu tutorial sirva de algo na emuscene e que, sobre tudo, anime muita gente a traduzir seus jogos preferidos neste momento em que esta escasso os "jovens" talentos. Sem mais o que dizer, demos início por fim a este manual.

Pablo Muñoz Sánchez Almería (Espanha), 4 de Julho de 2005

CAPÍTULO 1: ASPECTOS TRADUTOLÓGICOS

Há de se supor que quem vá traduzir um jogo deve ter a ilusão de vê-lo traduzido. Que seja você o autor da tradução, pode ser uma razão para levar em conta, se bem que o primordial é que traduza um jogo para você mesmo e depois para os demais. É freqüente cair no erro de pensar que você possui a necessidade de traduzir para alguém quando, ao tratar-se de algo altruísta, não há quem deva sentir mais satisfação do fruto colhido do que você mesmo. E para isto, algo é preciso: Qualidade.

1.1. Requisitos para traduzir

Ainda que tratar isso tudo como uma afeição, e embora nada deveria reprovar isso mesmo em princípio, devemos levar em conta que é preciso tentar "reexpressar" o sentido original no texto em português, não só "traduzir". O que quer dizer com isso? Que ainda que conheça perfeitamente o idioma estrangeiro do qual se vai traduzir (algo que dúvido, pois nem mesmo um nativo conhece todo seu idioma), é preciso ter cuidado ao escrever o novo texto em nosso idioma. E não só por ortografía como muita gente pode pensar (é horrível ver erro em qualquer texto, mesmo que pareça trivial) senão pela expressão. Sou de opinião que só se deve traduzir a língua materna, porque erros sérios de expressões nesta se consideram inaceitáveis.

O primeiro requisito para se traduzir não é saber um idioma estrangeiro, senão conhecer a língua do texto meta. Até mesmo sem ter nem idéia de chinês se pode escrever um texto em nossa língua materna que tenha sentido. Outra coisa é que seja fiel ao original. Obviamente, conhecer a língua de origem e sua cultura é também um principio básico para a tradução, embora sempre se possa consultar um dicionário e outros recursos para nos ajudar, o que tem seu inconvenientes como veremos mais tarde. Para adquirir um alto nível de competência tanto em sua língua de origem com na língua meta, recomendo encarecidamente algo que esta fazendo agora mesmo: ler. Não só aumentará sua bagagem cultural (indispensável para traduzir), mas com o tempo aprenderá centenas de expressões e palavras que certamente lhe pareciam estranhas em dado momento. Lembro que já faz muitos anos, quando traduzia o Secret of Mana, a expressão "what on earth" me chocava muito. Seu sentido literal pouco tem haver com "mas que demônios", seu equivalente em espanhol (entre outras expressões similares).

1.2. O papel do contexto

A situação em que se da lugar um ato comunicativo, quer dizer, o contexto, é fundamental pra compreender completamente qualquer palavra do enunciado. Conversando como um amigo, e com ânimo de oferecer um exemplo mais claro, proponho explicar a importância do contexto para compreender uma mensagem com a seguinte questão: é igual citar Deus quando alguém se surpreende ao observar uma proeza, quer dizer Deus em um momento de prazer? Eu creio que não. Pois que "God" em inglês corresponde a ambos os casos como o "Deus" em português. Mas isso nem sempre acontece assim.

É por isso que antes de traduzir um jogo é preferível jogá-lo antes e, ainda mais importante, traduzir enquanto se joga (uma prática provavelmente mais importante). Eu mesmo não traduzo sem antes joga-los, e segundo traduzia me enterrava de coisas que poderiam fazer variar minha tradução, sobre tudo o registro dos personagens. (digo, sua forma de falar). O exemplo mais claro se encontra no Final Fantasy IV, já que mudei muitas coisas enquanto revisava a tradução. Sem impedimento, no Phantasy Star IV, como conhecia com perfeição toda a história e a personalidade das personagens ao tratar-se de um de meus jogos preferidos, não me surgiram muitas dúvidas na hora de dar um certo "toque" aos diálogos de Alys e Raja (dois dos mais carismáticos de vários protagonistas do jogo).

Por outro lado, vale lembrar que ler o texto de um jogo é muito diferente de ler uma novela. Um jogo, é igual a um filme, é caracterizado por possuir imagens e sons. Se um personagem dizer "it's hot, isn't it? [Está calor, não está?] quando tal indivíduo se encontra na neve, produzirá em alguns jogadores ao menos uma leve risada. Este caso não parece estabelecer nenhum problema de tradução (sempre e quando não tenha um duplo sentido), mas em determinados caso as imagens do jogo possuem um importância fundamental na hora de criar jogos de palavras que só dão dor de cabeça ao tradutor. É nestes casos que a criatividade do tradutor cobra sua verdadeira capacidade.

1.3. A invisibilidade da tradução

Uma tradução de qualidade será aquela que precisamente não se deixe notar, quer dizer, aquela que não pareça uma tradução. Citamos muito, nossos personagens de cinema favoritos, mas por que nunca pensamos que tais falas estão escritas iguais ao script? Acaso pensamos que "mosquis" ou "multiplicar por zero" são resultado de uma tradução? Isso é o que devemos pensar, que nossa tradução não se manifeste em nossos consumidores como tal. E para isso o requisito

básico é dar naturalidade a nossas traduções.

Por exemplo, em inglês se usa bastante a passiva coisa que em espanhol não ocorre com tanta freqüência. "A present was given to me" se pode traduzir para "Me foi dado um presente". A equivalência é exata (não vou entrar em discussões lingüística) e em um exame de inglês certamente o professor consideraria certa a tradução. Sem problema, se analisamos bem a frase: "quem disse isso?" "quando lhe dão algo você se expressa assim?". É certo que pode dar ao caso, mas soa muitíssimo mais natural "Me deram um presente". Ainda que precise levar sempre em conta o contexto da frase com já foi dito anteriormente, soaria ainda melhor "Me deram um presente".

Outra coisa que gostaria de falar é da repetição de informação. Vejamos um exemplo. "Welcome to Aeido. Aeido is a big city!". O lógico seria traduzir esta frase para "Bem-vindo a Aiedo. Aiedo é uma grande cidade!" Transmite o mesmo sentido que a frase em inglês, mas acredito que "Bem-vindo a grande cidade de Aiedo!" soa muito melhor. Talvez não se expressa exatamente igual, mas nem toda frase deve ter uma equivalência exata nem seguir ordem sintática do texto original. Observemos esse outro exemplo: "Chaz: It's Great! -Alys: I don't think so...". Uma boa tradução seria: "Chaz: É genial! -Alys: Eu acredito que não...", mas sinceramente creio que "Chaz: É genial! -Alys: pois eu acredito que não..." soaria ainda melhor porque é muito mais natural. Não há com o que se preocupar desde que haja exatamente a mesma informação em um texto traduzido, pois às vezes tirar ou acrescentar elementos ao texto original favorece a tradução. Em qualquer caso, toda decisão deve estar justificada, já que somos tradutores e como tais não temos o direito de fazer o que nos der na telha com o texto.

Quanto à tradução de nomes próprios, não há o que ter medo ao trocá-los ou adaptá-los sempre e quando sua troca é justificada. Esta é uma questão um tanto delicada, já que os mais puristas se posicionam contra. Bem, eu proponho o seguinte: o que soa melhor, Cecil ou Cecilio? Anna ou Ana? Eu me calo com Cecil e Ana respectivamente, se bem que é certo que já tenho um contexto criado em mente (mas de um jogador reconheça tais nomes). Qualquer que seja a decisão, as explicações estariam contidas no arquivo "leia-me" que sempre acompanha as traduções. Vale lembrar que ver um nome estrangeiro provoca discussões em quem está jogando a tradução, mas também pode dar um ar exótico a ela. Da mesma maneira, adaptar todo nome que aparece em um texto pode implicar numa falta de credibilidade com respeito ao texto original e diminuir a beleza de sua respectiva cultura (já que o idioma esta extremamente unido a cultura). Não devemos esquecer que a atenção dirigida a uma tradução influi muito na mesma.

1.4. A necessidade de adaptações

A tradução audiovisual, e em geral a de jogos de video-games neste caso, se caracteriza por impor ao tradutor uma série de limitações. Em princípio, não se pode adicionar mais texto do que há originalmente em uma conversação. Esta construção poderá ser burlada mediante o uso de diferentes técnicas que serão vistas mais adiante neste manual. Apesar que, em teoria sempre se poderá ampliar o espaço original, levar tal afirmação a prática pode resultar numa tarefa nada simples que pode desesperar até a pessoa mais paciente dada sua complexidade. Especial importância é cobrada na hora de traduzir os menus que geralmente aparecem em muitos jogos(especialmente RPGs) e cujos impedimentos se baseiam na limitação de caracteres impostos por uma janela imóvel. Uma solução para tal problema é empregar as abreviações, se bem que não possuem um grande prestígio por diminuir o impacto visual. Ainda assim, até as traduções oficiais já vêm marcadas por tal visão.

Quando traduzimos e vemos que esta limitação de espaço nos obriga a aderir ao espaço marcado, devemos tratar de reexpressar o mesmo conteúdo da mensagem traduzida, mas de uma forma diferente. Aí que, como já disse anteriormente, considere que um exercício de tradução requer de vez em quando uma reexpressão de sentido. Por exemplo, se não podemos colocar "Não funciona" por exceder um caractere, uma possível solução seria reexpressar o mesmo sentido em "Não serve". Se bem que pode parecer que "servir" é de uma qualidade estética menor que "funcionar", mas cumpre perfeitamente sua obrigação e nenhuma crítica deverá cair ao tradutor por optar por esse sinônimo. Em caso que se deva de prescindir informação, se tratará de manter a essência da mensagem. Por exemplo, "Mas o que disse?" passaria para "Que disse?" se assim for preciso. Em efeito, perde-se a raiz, mas continua com o mesmo sentido. Por desgraça, há muitas ocasiões em que não há remédio que elimine informação, ainda que por sorte muitas mensagems dispõe de informação supérflua que não acrescenta nada de novo a conversação.

1.4.1. Conselhos ortográficos

Eis aqui uma série de conselhos a levar em conta na hora de traduzir um jogo:

- É melhor colocar duas frases curtas em duas linhas do que uma em uma e meia:
 - *x* Cecil, vieram lhe ver ontem. Nãotinham boa cara.Não tinham boa cara.

8

- É inadmissível usar abreviações, por exemplo o vc no lugar de você:
 - *x* Vc poderia ajudar-me? Poderia ajudar-me?
- Sempre que se pode e não seja uma cifra grande e complicada se deve escrever um número.
 - xMe deram 8 anéis.Me deram oitos anéis.
 - ✓ Faz 1000 anos. Faz mil anos.
 - x Custa 27 Moedas. Custa vinte e sete moedas.
- As reticências não devem ser utilizadas para indicar que haja mais texto para ser mostrado na próxima tela. Em tal caso pode ser conveniente adaptar a mensagem para que caiba em um só tela, se possível:

X	Cecil me disse que viria	Cecil me disse que viria
	logo para irmos ao bosque	logo para irmos ao bosque

... com os outros. conosco.

1.5. Os jogos de palavras.

Um dos grandes desafios que se apresentam aos tradutores são os jogos de palavras. Para isto retome á fundo na informação que neste momento cabe para um projeto universitário.

Adrián Fuentes Luque defendeu uma tese que tratava sobre a recepção do humor em um estúdio que eles fizeram do filme "Sopa de Ganso" dos celebres irmãos Marx. O filme não era mais do que um desafio de titãs para os tradutores. Um dos fragmentos mais interessantes do filme é o seguinte:

- FIREFLY (Groucho Marx): I suggest that we give him ten years in Leavenworth, or eleven years in Twelveworth.
- CHICOLINI (Chico Marx): I tell you what I'll do. I'll take five and ten in Woolworth.

CAPÍTULO 1: Aspectos tradutológicos

Aqui o humor se baseia na decepção do nome próprio *Leavenworth* (uma prisão federal dos Estados Unidos) em duas partes. *Eleven* [onze] e *worth* [valor], pelo que se estabelece numa relação fonética com os elementos *Twelve* [doze] e *worth de Twelveworth*(uma palavra inventada por Groucho) e com *Woolworth* (uma cadeia de tendas americanas pela sua campanha de five and ten). Ainda mais, um jogo de palavras com a seqüência de números de *ten years* in *Leavenworth, or eleven years in Twelveworth* e em *five and ten in Woolworth*. A solução que se propõe na versão dublada em espanhol do filme é a seguinte:

- FIREFLY (Groucho Marx): Sugiro condena-lo a dez anos de cárcere ou onze anos de prisão.
- CHICOLINI (Chico Marx): Sabe o que farei? Tomarei umas férias no campo.

Se tem bem pouco a ver de certo modo a ver versão original com a dublada, as estratégias de traduções empregadas são notáveis. Em primeiro lugar, deve destacar que toda referência cultual desapareceu sem mais remédio. Contudo, nesta tradução optou-se pelo absurdo com meio de humor. Não é preciso mais que ler a frase de Chicolini para soltar uma gargalhada, tal como disseram os participantes do estúdio. E disso se trata: o dialogo original tinha como propósito divertir o expectador, algo que pode ser comprovada na versão em espanhol. De fato, a versão legendada esta marcada por uma literatura que nenhum expectador achou graça na cena, seno que muitos acharão estranha.

1.6. Dicionários e outras fontes de informação

É mais provável que não seja uma pessoa bilíngüe e não conheça com perfeição o idioma estrangeiro que irá traduzir. Por isso, não tem que se "envergonhar" de usar dicionários. Todo tradutor usa, mas sempre com especial cuidado. Não precisa abusar e menos ainda confiar neles totalmente. Por exemplo, sei que existe muitas traduções de jogos em que se traduz "pendant" por "pendente". A mim também já aconteceu; e conto adiante da minha história. Ainda que sabendo que "pendant" era "pendurado", coloquei minha confiança nos dicionário e optei por usar "pendente". Sim, efetivamente, o dicionário colocava isso. Tem que usar o senso comum e ver se o que diz o dicionário pode ser considerado "verdadeiro". Muitas vezes é melhor analisar o contexto (espero que com isso fique totalmente clara sua função) e ver que palavra ou expressão é a que melhor encaixe e usar o dicionário para assegurar-se e contatar possibilidade.

O dicionário bilíngüe (aquele em que se busca uma palavra em um idioma e vem seu significado em outro) não é sempre a única solução, às vezes é recomendável recorrer ao dicionário monolíngüe (a explicação vem dada no mesmo idioma, por exemplo, qualquer dicionário de espanhol que usamos quando não entendemos alguma palavra complexa). Si se traduz desse idioma estrangeiro não deve haver problema para compreender a explicação. A propósito, hoje em dia muitos dicionários estão disponíveis em CD-ROM para usa-los nos computadores, coisa que recomendo encarecidamente, já que assim é muito mais fácil e cômodo de buscar palavras e até permite várias opções de busca. Não só se pode usar dicionários; pergunte a alguém cuja língua mãe é a que nós ainda tratamos de aprender, pode ser uma boa solução, igual a internet. Contarei outra de minhas experiências.

Um vez, encontrei numa série de anime legendada em inglês que eu traduzia ao espanhol uma expressão que não entendi: "Serves you right!". Tal expressão era falada em um momento delicado e não podia inventar a expressão. Como o dicionário não me ajudava, fui ao canal de IRC onde conversava um grupo de tradutores ingleses e, sem preguiça ou timidez, os perguntei o que significada tal expressão. Em dez segundos me responderam, e pela explicação que me deram, cheguei a conclusão de que deveria traduzi-la para "Você merece!".

1.6.1. Dicionários recomendados

Não tenho em minhas mãos todos os dicionários que existem evidentemente, ainda assim os farei uma recomendação. Como dicionário bilíngüe inglês-espanhol e espanhol-inglês, eu uso o Gran Dicionário Oxford(Terceira Edição). Vem com um CD-ROM (realmente facilita a árdua tarefa de procurar palavras na versão impressa). Eu gosto, é muito completo, ainda que o dicionário de Collins não parece ser uma má opção. Como dicionário monolíngüe inglês uso a versão em CD-ROM do Merriam-Webster's 11th(o th em sobre escrito) Collegiate Dictionary. A verdade é que é muito completo e sempre é encontrado tudo o que se procura. Se traduz do francês, eu uso como dicionário bilíngüe a versão impressa (não encontrei nenhum decente para computador) do Gran Diccionario Larousse e como monolíngüe a versão em CD-ROM do Le Petit Robert, Se alguém traduz do alemão, usar dicionários da editora Langenscheidt parece ser a melhor solução.

CAPÍTULO 1: Aspectos tradutológicos

1.7. Erros mais comuns nas traduções

Fiz aqui uma pequena lista de erros que tenho visto em algumas traduções de ROMs:

- Actually: Não, não é "atualmente". Eu sempre opto traduzi-lo por "de fato", "a verdade é que..." ou expressões parecidas.
- Bastard: Sim, pode se referir a um filho bastardo, mas se emprega também como insulto.
 Supondo que "corno" é muito forte para colocar em um jogo, cabe ao tradutor decidir coloca-la ou não.
- A ordem do adjetivo: Em inglês se antepõe o adjetivos do substantivo. "He's a strong man" deve ser traduzido para "é um homem forte" e não por "é um forte homem". Em muitos casos pode se traduzir dessa maneira, mas deve ter cuidado.
- *Eventually*: Não se deve traduzir por "eventualmente", e sim por "finalmente", "ao final" ou expressões similares.
- Now: Igual ao anterior, é um dificuldade. Eu traduzo por "Bueno".
- Pendant: Como foi dito mais acima, não deve traduzi-lo para "pendante" e sim por "pendurar"
- Surgir efeito: O correto é "Surtir efeito".
- Virtually: Não tem nada haver com virtual. Se deve traduzir para "praticamente" ou similares.
- *What on earth?*: "O que acontece na terra?" Não, deve traduzi-la para "Que demônios?" ou expressões parecidas.
- Why: Nem sempre é "por que", já que se pode empregar como interjeição de surpresa, é dizer, que se deve traduzir para "anda", "vá"...
- Ya: Contração de "you", não tem nada a ver com o "ya" espanhol.
- Yo: É um tipo de saudação muito coloquial (é como dizer "como vai" entre amigos).

CAPÍTULO 2: INTRODUÇÃO AO ROMHACKING

Neste capítulo serão explicados os paços básicos necessários para começar a editar o texto de um jogo qualquer. Em outras palavras, verão o processo utilizado para procurar texto em um arquivo binário de um jogo (geralmente se trata de uma ROM).

2.1. Programas necessários e outros requerimentos

- *TaBular*: Programa que simplifica o trabalho de fazer uma tabela manualmente. Mais adiante veremos o que é um tabela.
- *Editor de texto* (Bloco de Notas do Windows por exemplo): Servirá para modificar a tabela de uma forma rápida.
- SearchR X: é capaz de procurar texto em arquivos muito grandes (como ISOs de PSX) sem necessidade de carrega-lo na memória RAM, algo que pode deixar lento o processo de abrir o arquivo. Além disso, permite fazer uma busca de 16 bits.
- O arquivo que se deseja traduzir. Tomarei como exemplo a ROM de Secret of Mana, já que é uma ROM que não oferece problemas na hora de procurar o texto.

2.2. Conceitos Básicos: código hexadecimal e tabelas

Não quero me alargar mais que o necessário gostaria de esclarecer o melhor possível estes dois termos já que qualquer tradução que façamos, faremos uso freqüentes deles.

2.2.1. Código hexadecimal

Hexadecimal faz referência a um sistema de numeração que, tal como indica seu nome, toma inteiro de base 16. Então, o número 10 em decimal equivale a 0A em hexadecimal. Cada número em hexadecimal são representados por dois dígitos que podem ir de 00 à FF (255 em decimal). Abaixo podemos ver uma série de números decimais com seus correspondentes valor em hexadecimal.

Sistema decimal: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18... Sistema hexadecimal: 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F, 10, 11, 12...

CAPÍTULO 2: Introdução ao ROMHacking

A pesar de que talvez o correto seja dizer "valor hexadecimal", tende-se a ampliar mais ao termo "código hexadecimal" para falar sobre números hexadecimais ou simplesmente "código hex".

Aparte do sistema decimal e hexadecimal existem mais dois: o sistema binário e o octal. Devido que é possível que devamos trabalhar com o sistema binário, procederei com sua explicação. neste sistema cada número é composto de 8 bits, quero dizer, oito dígitos, que pode ser 0 ou 1. Por exemplo, 00000011 em binário corresponde ao número 3 em decimal. No caso que precise fazer conversões de um sistema para outro, existem muitas calculadoras que nos podem ser de grande ajuda, como a que o Windows traz (vale lembrar que deve ativar o modo científico).

2.2.2. Tabelas

Uma tabela é um arquivo de texto com extensão ;tbl no qual se especifica a equivalência dos caracteres que u jogo usa (a, b, c...) em hexadecimal. Também se utiliza tal termo para referir-se ao conjunto de equivalências que estão em uma tabela sem fazer referência ao arquivo com extensão .tbl. As tabelas são usadas por programas como Thingy ou Translhextion com o objetivo de mostrar neles os caracteres latinos (no caso de traduzir do inglês) que previamente especificaremos. Desta maneira se poderá ver e editar o texto do jogo em questão mais comodamente.

O valor de cada igualdade é a seguinte: (Valor hexadecimal)=(caractere). Por exemplo, algumas igualdades da tabela do Secret of Mana são:

82=a 9A=z 9C=B

•••

2.3. Procura por textos em um arquivo binário

Para isso o primeiro que temos que fazer é executar o SearchR X e carregar a ROM do Secret of Mana. Uma vez carregada , nos situamos na caixinha de texto e escrevemos uma palavra que apareça no jogo, como por exemplo "Sword". Note que essa palavra possui maiúsculas e minúsculas. Clicamos no botão *Buscar* e quando terminar olhe para a área de baixo onde tem os *Valores*. Veremos que existem várias entradas dentre elas aparecem A=9B e a=81 como mostra a seguinte imagem:

8	Search	irx – – ×
Archivo Opciones Ayuda		
Offset [dec]	Valores	Vista Previa
24832	A=9B a=81 1=??	#Cave#in#Gloves#Sword#Axe#Spear
592040	A=9B a=81 1=??	###Revived#Mana#Sword#####h####
603030	A=9B a=81 1=??	<pre>#the##real#Mana#Sword####Huh##</pre>
603206	A=9B a=81 1=??	ned#to#the#Mana#Sword#####Here
607039	A=9B a=81 1=??	revive#the#Mana#Sword####Go#to
610795	A=9B a=81 1=??	#with##the#Mana#Sword#####The#v
615375	A=9B a=81 1=??	#about#the#Mana#Sword#and#you#w
616386	A=9B a=81 1=??	<pre>wer#of#the#Mana#Sword##has#seep</pre>
627913	A=9B a=81 1=??	a#restored#Mana#Sword##can#beat
660168	A=9B a=81 1=??	U#have#the#Mana#Sword###YOU#mus
660468	A=9B a=81 1=??	ic##then###Mana#Sword#will#appe
669562	A=9B a=81 1=??	###wielding#The#Sword##they####
670911	A=9B a=81 1=??	will#master#the#Sword##by#joini
687995	A=9B a=81 1=??	using#the##Mana#Sword#####The#s
688785	A=9B a=81 1=??	##b#b#b#########Sword#power#at#
697122	A=9B a=81 1=??	#Shadow#X##Evil#Sword#Tomato#Ma
697635	A=9B a=81 1=??	#Wolf#Lord#Doom#Sword#Terminato
1		
Búsqueda Relativa Búsqu	eda a 16bit	
. Duoda		Búsqueda terminada en 0 min 1161
		sec.
Busca: Sword	Tiempo mínimo: 0 m	in. 2,678 sec.
De: 0	a: 2097152	O Hex
De. U	2037132	Dec
		Buşcar Parar

O que aparece em *Offsets* é irrelevante neste caso (um offset indica a posição do texto dentro do arquivo binário). O que surge em *Vista Prévia* nos permite ver o texto usando a equivalência dos valores como se de uma tabela se trata. Pelo que parece, tais equivalência são as que são usadas no jogo, á que todos os texto sem que "Sword" aparece usam os mesmo valores e estão rodeados por outras palavras em inglês. Se queremos nos assegurar disso, podemos dar um duplo clique na tela do programa onde mostra tudo e aplicar *ASCII* para que mostre o texto com os valores da possível tabela:

8		ç	:ATre	idue	cion	esW	tilida	des)	Sear	chR	(ASe	eret	of M	ana (V1.0)) (E).s	iiie 💌
-																	
Offset [dec]	Hex	ade	ecin	nal													Vista Previa 📐 🔺
0000024832	AD	97	8F	92	84	00	9B	98	85	00	AD	90	85	81	92	00	Sword#Axe#Spear# -
0000024848	B1	88	89	90	00	9B	92	92	8F	97	00	9C	8F	8F	8D	85	Whip#Arrow#Boome
0000024864	92	81	8E	87	00	Α4	81	96	85	8C	89	8E	00	00	00	C2	rang#Javelin####
0000024880	93	80	94	81	8E	87	8C	85	84	C8	00	80	83	81	8E	C2	s#tangled###can#
0000024896	94	80	8D	8F	96	85	C8	00	C2	93	80	95	8E	83	8F	8E	t#move###s#uncon
0000024912	93	83	89	8F	95	93	С8	00	C2	93	80	86	92	8F	93	94	scious###s#frost
0000024928	89	85	84	С8	00	C2	93	80	90	85	94	92	89	86	89	85	ied###s#petrifie
0000024944	84	C8	00	C2	93	80	83	8F	8E	86	95	93	85	84	C8	00	d###s#confused##
0000024960	C2	93	80	93	90	81	83	85	84	80	8F	95	94	C8	00	C2	#s#spaced#out###
0000024976	93	80	90	99	87	8D	89	9A	85	84	C8	00	C2	93	80	82	s#pygmized###s#b
0000024992	81	92	92	85	8C	85	84	C8	00	C2	93	80	94	92	81	8E	arreled###s#tran
0000025008	93	86	8F	92	8D	85	84	C8	00	C2	93	80	8D	8F	8F	87	sformed###s#moog
0000025024	8C	85	84	C8	00	C2	93	80	90	8F	89	93	8F	8E	85	84	led###s#poisoned
0000025040	C8	00	C2	93	80	85	8E	87	95	8C	86	85	84	C8	00	80	###s#engulfed###
0000025056	93	85	85	93	80	94	88	85	80	92	85	81	90	85	92	C8	sees#the#reaper#
0000025072	00	80	86	85	81	92	93	80	Α1	8E	8F	8D	85	C8	00	80	##fears#Gnome###
0000025088	86	85	81	92	93	80	AD	99	8C	90	88	89	84	C8	00	80	fears#Sylphid###
0000025104	86	85	81	92	93	80	ΑF	8E	84	89	8E	85	C8	00	80	86	fears#Undine###f
0000025120	85	81	92	93	80	AD	81	8C	81	8D	81	8E	84	8F	C8	00	ears#Salamando##

CAPÍTULO 2: Introdução ao ROMHacking

Agora que sabemos a tabela do jogo, basta lembrar o valores de "A" e "a" para fazer a tabela. Porém, é muito provável que tenha sido capaz de encontrar o texto no caso de ter tentado com outro arquivo que não seja o Secret of Mana. O mais provável é que o texto esteja comprimido; no capítulo 5 são dados mais detalhes sobre a procura de textos comprimidos.

2.4. Fazer uma tabela

Para isso devemos usar o TaBuLar. O que temos que fazer é colocarmos o cursor na casa do valor hexadecimal do caractere "A", que no caso do Secret of Mana é 9B. Os números e letras da esquerda são para o primeiro dígito do código hexadecimal(9) e os de cima do segundo (B). No menu *"Insert"* colocamos em *"Capital English Alphabet"*. Agora colocamos o valor que corresponde ao "a" e fazemos o mesmo, só que dessa vez colocamos *"Lowercase English Alphabet"*. Salvamos a tabela no formato do Thingy e pronto. O resultado deverá ser como o da imagem a seguir:



Desta maneira temos uma tabela com os valores hexadecimais do alfabeto latino, ainda que falte alguns outros caracteres, como os sinais de pontuação. No próximo capítulo veremos de qual maneira de acha o valor hexadecimal de tais caracteres.

2.5. Para saber mais

 The Definitive Guide to ROM Hacking Tables (por InVerse): Excelente documento no qual se explica detalhadamente todos os passos para fazer uma tabela. Além de explicar com fazer as tabelas de jogos japoneses. CAPÍTULO 2: Introdução ao ROMHacking

CAPÍTULO 3: EDIÇÃO DE TEXTO

Neste capítulo veremos como editar o texto de uma ROM a partir da base que espero que tenha se consolidado depois de ler o capítulo anterior. Editar o texto será uma tarefa fundamental durante o processo de tradução, veremos ainda mais de um modo de traduzir o texto de uma ROM.

3.1. Programas necessários e outros requerimentos

- *Translhextion*: À parte de ser um potente editor hexadecimal, considero que é o melhor programa para a extração e inserção de scripts (que veremos mais adiante) apesar de algumas carências.
- *ThingyV*: É a versão modificada por Vegetal Gibber da Vegetal Translations que acrescentou algumas características novas ao clássico Thingy, tais como suporte para arquivos grandes (como ISOs de PSX). E uma na opção no caso de precisar usar duas tabelas.
- O arquivo que deseja traduzir. Tomarei como exemplo a ROM do Secret of Mana..

NOTA: Verifique que a ROM não tenha o atributo Somente Leitura, pois se não, ao salvar as modificações acontecerá um erro. Para comprovar , de um clique com o botão direito sobre o arquivo da ROM e vá em Propriedades. Assegure-se de que a opção Somente Leitura NÃO esteja marcada, tal como mostra é mostrado na imagem:

Atributos: Somente leitura Oculto

3.2. Uso do Translhextion

No capítulo anterior fizemos uma tabela que incluía todo o alfabeto latino, mas que não estava completa faltando caracteres como os sinais de pontuação. É necessário completá-la antes de começar a traduzir, para isso devemos fazer uso do programa Translhextion Devido que vamos usá-lo constantemente, explicarei todas as suas funções ainda que pareça exaustivo.

CAPÍTULO 3: Edição de texto

52		[C:\Trad	uecior	nes\U	Itilidad	es\Sea	urehl	ocisi	eere	t of l	Mana	e91.	0) (E) . .sm	e] - '	Translhexti	an e	
File	Selection	n Offset	Inser	t Se	earch	Bookm	arks	Scri	ipt	Optic	ons	View	He	lp				
	Table Te	disar	00 4C	4C	83 OI 64 40	0 4C	86 64	00 4C	4C 00	89 38	00 4C	4C 03	8C 38	00 4C	4C 06	# \#c\ o\#\##	#f\#i\#l\# ###\######	-
	L hingy Vi	ew Active	09 38	38 4C	4C 00 1B 30	C 38 3 4C	4C 1E	0F 38	38 4C	4C 21	12 38	38 4C	4C 24	15 38	38 4C	###### ######	########### ###########	
C	Shift-JIS	 #	4C 20	2A 64	38 40 4C 30	2 2D 2 38	38 4C	4C 3F	11 38	64 4C	4C 42	14 38	64 4C	4C 45	17 38	#######	############ ###########	
0	EUCJIS	□ >	38 4C	4C 5A	4B 31 38 40	3 4C 2 9B	4E 00	38 6B	4C FF	51 FF	38 FF	4C FF	54 FF	38 FF	4C FF	###### ######	########## A\########	
00	000080	4C 40 97 1E	0E 4C	4C 13	2C 0: 01 40	L 4C D BF	F7 0E	1B 4C	4C 63	C9 10	00 5C	4C 18	BA 00	00 C1	4C 4C	###### i#####	####ú\##\# .######\##	
00	0000A0	A5 00 AB 20	4C 24	B9 10	00 81	B DA B FA	48 AB	08 6B	C2 6B	10 08	E2 C2	20 20	A9 8F	7E 5A	48 A2	K\##\k Q#####	###ñ###0## #Q###ñ#o#H	
00	0000C0	7E A9	07 8F	00 F6	28 50 CF 71	0 97 E AF	1D 12	C0 A2	08 7E	E2 09	20 10	A9 8F	00 12	8F A2	FD 7E	#0#\## #\o###	i#,###O∖o# U#H###o#H#	
00	0000E0	20 AA 20 AF	0F 13	A9 A2	FF 81 7E C	F 1F 9 07	01 07	00 F0	A9 14	06 20	22 D5	97 0F	1D AF	C0 3C	C2 A2	#P#O#o #U#H#ú	##\0##i#,ñ ######U#H	
00)00100)00110	7E 22	0F C1	00	C2 AI E2 21	7 5A 0 C2	A2 10	7E AF	28 62	5C A1	0C 7E	80 C9	00 02	28 F0	5C 45	###\ñU #\####	#H#### \## ñ#U#G#ú###	
00)00120)00130	A9 01 10 AF	8F 62	62 A1	A1 71 7E C1	E 8F 9 02	10 F0	1D 2C	00 A9	80 02	39 8F	08 13	E2 1D	20 00	C2 AF	0#o#G# #U#G#ú	0##\ ####ñ ###0#0##\U	
00	000140 000150	10 1E 7E C9	00 51	89 F0	01 F 07 AI) 18) 12	A9 1D	00 89	8F 20	10 F0	1D 03	00 20	AF E1	70 0F	A1 A9	##\i## #ú###S	#0\o##\U#G ##i#####0	
00 00	000160 000170	00 8F A5 29	62 29	A1 07	7E 81 8D 01	3 OB 5 1D	A2 C9	00 07	00 D0	DA 06	2B A9	A9 01	00 8F	48 62	AB A1	∖o#G#k K###m#	#H\\##0\#Q #ú###0#o#G	
00)00180)00190	7E A9 0C 04	00 1D	EB AD	AD 04 03 11	4 1D D AC	29 01	02 1D	F0 48	03 AB	82 AF	91 70	00 A1	A9 7E	01 C9	#0\#S# ###S##	#####bq\0# R###QU#G#ú	
00 00	0001A0	7D D0 C9 7D	06 D0	22 06	00 01 22 01	9 CC 0 0 9	80 CC	76 80	C9 65	00 C9	D0 50	17 90	B9 19	00 C9	00 60	####\# ú####\	# #ú\###\\ ## #ú#p#ú#	
00 00	0001C0	90 02 70 A1	A9 FC	53 DA	48 31 10 81	3 E9 3 48	50 A9	0A 00	48 48	A9 AB	7E AD	48 04	AB 1D	68 29	8D 80	p#0### #G###	###P0##Q#m #0\#QS###	
00 00	0001E0 0001F0	F0 OE 5D A1	C2 7E	20 A9	AF D! 01 10	5 A1 C 04	7E 1D	8D A9	$01 \\ 04$	1D 1C	E2 00	20 1D	A9 A9	00 80	8F 2C	##ñ#U# #G#O##	G#n####0\0 ##0##\#0 #	
00)00200)00210	00 1D D0 0D	F0 A9	1B 20	1C 0) 1C 0-	1D 4 1D	1C 1C	04 12	$^{1\mathrm{D}}_{\mathrm{1D}}$	AF A9	87 01	A1 1C	7E 10	89 1D	01 A9	\####\ ##0###	####UgG#i# ####0####0	
<u>∩∩</u>	100220	100 8F	13	1D	00_21	R AR	28	6B	BB	E8	ΑF	03	1D	00	88	<u>\</u> _#*\#	∩####II##∖k_	b v
Offs	et 0=0×0	Bits=010	01100	Table	e={4⊂)	Unsig	ned: I	B:76,	W:3	2844,	L:12	7510	1260	P	NSI	/ OVR / L	Size: 2097152	

- No menu *File* é onde abriremos os arquivos para modificações, as demais opções não necessitam de explicação.
- No menu Selection Selection podemos encontrar os típicos comandos para copiar, cortar e colar o texto.
- No menu *Offset* Offset está o super útil *Jump* o que nos servirá para ir a posição ou endereço (offset) que queiramos. Se colocarmos um endereço em hexadecimal será necessário colocar um x antes do endereço (por exemplo, x10EE). Note que não é preciso colocar os zeros da esquerda.
- No menu Insert não tem muitas opções úteis para dizer a verdade.
- No menu Search introduziremos o texto para fazer uma busca utilizando a tabela. É muito sensível como veremos. Também pode procurar por texto em ASCII (texto que não precisa de tabela para mostrar corretamente os caracteres). As demais funções são interessantes, mas para isso já temos o SearchR X.
- No menu *Bookmark* podemos inserir na tabela, *marcadores de offset*, quer dizer, uma espécie de acesso direto para uma posição determinada do arquivo binário. É muito útil colocar marcadores em diversas partes do texto que pode ter no jogo, como os menus, créditos ou o texto principal. Translhextion suporta até 9 marcadores.
- No menu *Script* podemos encontrar tudo o que é relacionado aos scripts. Veremos mais detalhes sobre scripts mais tarde.
- No menu Options encontramos as Preferences do programa. Recomendo trocar os { } por <>

como pode ser visto na imagem seguinte porque os scripts (logo veremos o que eles são) extraídos com outros programas usam <>, o facilitará a compatibilidade. Além disso, se deseja que ao clicar em um arquivo com o botão direito sai no menu contextual a opção *Abrir com Translhextion*, ativada a opção *Integrate into Windows Explorer Context Menu*

Preferences											
Automatically adjust number of bytes per line. OK Number of bytes to display per line: Cancel											
Display length of offset in how many characters: 8											
Display values at caret as: Character set Image: Character set Image: Character set Image: Charac											
Separated-byte format: Left Enclosure: < Right Enclosure: >∏ ☑ Make default											
 Reset changed-byte color to unchanged color upon saving a file Set read-only mode upon opening a file 											
Integrate into Windows Explorer Context Menu Path and filename of preferred text editor:											
notepad.exe Path and filename of preferred Internet browser:											
lexplore											

3.3. Completando a tabela

Para isso devemos saber o texto exato de algum diálogo, então jogaremos um pouco o Secret of Mana para ver e trabalhar com um dos primeiros textos. É escolhido o que se mostra na imagem abaixo:



CAPÍTULO 3: Edição de texto

Nossa tarefa será procurar na ROM a frase *Be quiet*. Para isso, executamos o Translhextion e carregamos tanto a ROM como a tabela que previamente já criamos. Vamos ao menu *Search* e escolhemos a *Find Using Table*. Procuramos *Be quiet* e em poucos segundos estaremos diante do texto. Todos os # que são vistos ao redor do texto são os códigos hexadecimais que não constam na tabela. Nosso objetivo é procurar os valores que nos faltam, não precisa descobri pra que serve TODOS os #.

Bem, coloque o curso no # que esta logo depois de *ELLIOTT* e olhamos seu código hexadecimal a esquerda. Nos baseando no texto que vimos antes deduzimos que C5 equivale a (:) e que C8 a (!). Também há outros # interessantes no texto, como o que está ao lado do primeiro ponto de exclamação. Este código, cujo valor é 7F, indica quebra de linha. Também é importante o código que está atrás do último ponto de exclamação. já que aparentemente representa o código do fim da mensagem. Porém, temos que ressaltar que Secret of Mana possui vário código de fim de mensagem é o 00. De fato, muitos jogos usam este valor como código de fim de mensagem. Por outro lado, parece evidente que o código 80 representa um espaço. Aqui pode ver uma imagem de área da ROM de Secret of Mana.

File Selection Offset Insert Search Bookmarks Script Options View Help Tables Houghtar 87 80 89 8E 80 81 80 90 8C 81 83 85 7F 80 ing in a place# like this####### Image BB 85 80 94 88 89 93 CA 28 00 51 17 0E 00 like this####################################	[G:\Trad	lucciones/Utilidades/SearchRX(Secret of Mana (V1.0) (E).smc] - Translhextion 😑 🙃	J) 🗶
Bits Built Hundbar 87 80 89 8E 80 81 80 90 8C 81 83 85 7F 80 ing in a place# Image: Thingy View Active 88 85 94 88 89 93 CA 28 00 51 17 0E 00 like this####################################	File Selection Offset	t Insert Search Bookmarks Script Options View Help	
○ Shift-JIS # 04 C8 08 95 90 C8 32 04 C6 08 30 0C 01 28 ####up################################	Thingy View Active Japanese	87 80 89 8E 80 81 80 90 8C 81 83 85 7F 80 ing in a place# 8B 85 80 94 88 89 93 CA 28 00 51 17 0E 00 like this####################################	-
00092900 9F A6 A6 A3 A9 AE AE C5 AD 88 88 C8 80 9C 85 80 ELLIOTT#Shh# Be 00092910 91 95 89 85 94 C8 7F 80 AE 88 85 80 9F 86 84 85 quiet## The Eller 00092920 92 80 97 89 8C 8C 88 85 93 7F 80 r rill find us# 00092930 88 85 92 85 80 88 85 28 80 95 93 7F 80 r	O Shift-JIS □ # O EUC-JIS □ >	04 C8 08 95 90 C8 32 04 C6 08 30 0C 01 28 ####up######## 09 31 00 88 08 31 00 88 08 80 9B 81 81 81 81 88 C8 34 00 9D 32 05 40 32 06 80 28 08 52 ab###C#################################	
00092950 AE A2 B3 C5 B3 85 81 88 BF 80 B1 85 C2 92 85 80 THY¥Yeah# ###re 00092960 8E 8F 93 95 90 95 93 85 84 80 94 85 not# supposed to 00092970 80 82 85 80 85 92 85 BF 7F 80 AE 88 85 92 85 BF 7F 80 81 94 80 81 92 85 86 93 94 80 81 94 81 84 80 92 85 80 92 85 80 92 85 80 92 85 80 92 81 94 <	00092900 9F A6 00092910 91 92 80 00092920 92 80 00092930 88 80 00092930 82 80 00092950 AE A2 00092950 AE A2 00092960 BE 87 00092970 80 82 93 000929970 80 80 000929900 84 94 000929900 84 94 000929900 85 81 000929900 80 94 0009292900 93 58 93 000929290 93 58	6 $A6$ $A3$ $A9$ AE AE $C5$ AD 88 88 $C8$ 80 $9C$ 85 80 $guitt##$ 5 89 85 94 $C8$ $7F$ 80 AE 88 85 80 $9F$ $8C$ 84 80 97 80 $8C$ $8C$ 80 87 $8C$ 84 80 95 97 80 r r $rill$ $rind$ $uitt##$ $rill$ $rill$ $rind$ $uitt##$ $rill$	
000929F0 85 81 92 81 82 84 80 93 81 99 eard Grandma say 00092A10 7F 80 93 87 80 85 94 88 89 82 97 80 81 82 87 90 81 82 87 90 81 82 87 90 81 82 87 90 81 82 87 90 81 82 87 90 81 82 87 90 81 82 87 90 81 82 86 93 t a shiny# objec 00092A10 94 80 81 92 50 32 06 40 34 00 82 93 t near the falls 00092A40 A9 AE A2 B3 C5 AE 88 81 94 80 84 80 82 OTHY#That must b 00092A50 85 81 93 95 92 85 87 80 81 81 84 <td>000929F0 85 87 00092A10 94 80 00092A20 94 80 00092A30 BF 28 00092A30 BF 28 00092A30 BF 28 00092A30 BF 28 00092A40 A9 AE 00092A50 85 80 00092A60 94 92 00092A70 96 85 00092A90 06 51 00092A90 06 D0 00092A90 A3 C2</td> <td>1 92 84 80 A1 92 81 80 A1 90 81 93 81 99 eard Grandma say 0 93 87 80 85 93 81 90 81 80 85 95 # something abou 0 81 80 93 81 92 85 93 81 85 95 # something abou 0 81 80 93 81 92 85 80 86 81 82 84 85 80 t a shiny# objec 0 81 92 80 94 88 85 80 86 81 80 85 1 a shiny# objec 0 81 80 85 80 86 81 80 82 84 80 80 A ####################################</td> <td></td>	000929F0 85 87 00092A10 94 80 00092A20 94 80 00092A30 BF 28 00092A30 BF 28 00092A30 BF 28 00092A30 BF 28 00092A40 A9 AE 00092A50 85 80 00092A60 94 92 00092A70 96 85 00092A90 06 51 00092A90 06 D0 00092A90 A3 C2	1 92 84 80 A1 92 81 80 A1 90 81 93 81 99 eard Grandma say 0 93 87 80 85 93 81 90 81 80 85 95 # something abou 0 81 80 93 81 92 85 93 81 85 95 # something abou 0 81 80 93 81 92 85 80 86 81 82 84 85 80 t a shiny# objec 0 81 92 80 94 88 85 80 86 81 80 85 1 a shiny# objec 0 81 80 85 80 86 81 80 82 84 80 80 A ####################################	

Bem, o que temos que fazer agora é abrir a tabela como o Bloco de notas do Windows (ou qualquer outro editor de texto) e acrescentar esses valores, embora o código de quebra de linha se escreve com o um asterisco (*) e de fim de mensagem com uma barra (/). Desta forma, a tabela deveria ficar assim:

🗭 Bloco de notas														
Arquivo	Editar	Formatar	Exibir	Ajuda										
/00 *FF C5=: C8=! 80= 82=a 83=C														

Agora precisa procurar por outros diálogos para acrescentar virgulas e outros caracteres, embora que outro método eficaz de encontrar vário códigos hexadecimais e suas equivalências é ver seu valor logo no jogo. Lembra que para editar os códigos hexadecimais que aparecem a esquerda tem que usar o TAB.

-	Į¢	:\Tradu	ecio	nesV	Utilid	ade	s\Sea	rchR	XASe	ere	t of P	lana	(91.0)) (E)).sme	: 제]	Translhextion 😑 🗉 😠
File	Selection	Offset	Inse	ert	Searc	:h	Bookn	narks	Scr	ipt	Optic	ons	View	He	lp		
	Table Tool	ien u Astius	87 8B 40	80 85 01	89 80 10	8E 94 44	80 88 2F	81 89 27	80 93 8D	90 CA 06	8C 28 50	81 00 32	83 51 00	85 17 03	7F 0E 32	80 00 04	ing in a place* like this##\###\ #######m###\###
Ē	Japanese	W ACUVE	32 95	00 99	Č0 93	57 C8	00 7F	Č5 32	Å2 04	85 C8	99 08	Č8 80	80 B1	32 81	04 89	Č8 94	###\##\#Hey# ### #Guvs# * #### #ait
	Shift-JIS ELIC-JIS	□ # □ >	04 09	C8 31	08	95 88	90 08	C8 31	32 00	04 88	C6 08	08 80	30 9B	0C 81	01 81	28 81	####up######## ####\h##\h# Aaaa
000	92900	B5 B6	88 B7	C8 B8	34 C0	00 C2	9D C3	32 C4	05 C5	A0 8	32 88	06 C8	80 80	28 9C	08 85	52 80	aah##\C##F## ### ########hh# Be
000	92910 92920	91 95 92 80	89 97	85 89	94 8C	C8 8C	7F 80	80 86	AE 89	108 8E	85 84	80 80	9F 95	8C 93	84 7F	85 80	quiet#* The Elde r jill find us*
000	92930 92940	88 85 85 80	92 8D	85	80 84	81 C8	8E 28	84 00	80 7F	88 32	85 06	C2 40	8C AE	8C A3	80 A7	82 A9	here and he#11 b e mad##*###TIMO
000	92950	AE A2 8E 8F	94 95	C5 7F	80 80	93	95	88 90	BF 90	80 8F	93 93	85	84	92 80	94	80 8F	THY#Yeah# #e#re not* supposed to
000	92970	C2 93	80	80	80	87	88	85 8F	93 22	94	80	AE 81	88 92	85 8F	92	85 8E	#s a ghost aroun
000	929A0	A9 AE	AE 8C	C5	A2	81 75	C8	80 82	9E	8F	80	99	8F	95 95	80	92 97	OTT#Ha# Do you r
000	929C0	88 81	94	CA 81	80	ÅÅ 80	85	8F 8F	90	8C	85	80	93	81	99	7F	hat# People say*
000	929E0 929F0	93 C8 85 81	28	00	7F 80	32 A1	05	40 81	80 8E	9Č 84	95 8D	94 81	80	A3 93	80	88 99	s##*### But I h
000	92A00 92A10	7F 80 94 80	93 81	8F 80	8D 93	85	94 89	88 8E	89 99	8E 7F	87 80	80 8F	81 82	82 8A	8F 85	95 83	* something abou t a shinv* objec
000	92A20 92A30	94 80 BF 28	8E 00	85 7F	81 32	92 05	80 C0	94 32	88 06	85 40	80 34	86 00	81 80	8C AE	8C A3	93 A7	t near the falls ##*######\ TIM
000	92A40 92A50	A9 AE 85 80	A2 93	В3 8F	C5 8D	AE 85	88 7F	81 80	94 8B	80 89	8D 8E	95 84	93 80	94 8F	80 86	82 80	OTHY#That must b e some* kind of
000	92A60 92A70	94 92 96 85	85 80	81 94	93 8F	95 80	92 86	85 89	C8 8E	7F 84	80 80	B1 89	85 94	80 C8	88 08	81 28	treasure#* #e ha ve to find it###
000	92A80 92A90 92AAN	00 51 06 D0 A3 C2	32 34 8D	05 00 80	D0 88 86	32 08 81	06 50 8C	D0 57 8C	32 00 89	00 C5 8E	00 A2 87	08 85 C8	32 8C 32	05 90 05	D0 C8 80	32 80 32	\########\\##### ###\h###\#Help# T#m falling### #
∙ Offse	et 600329=1	0×92909	Bits=	=100	0100	ο ι	Jnsign	ed: B	:136	,w:3	4952	,L:21	6062	5800	ļ,	ANSI	/ OVR / L Size: 2097152

CAPÍTULO 3: Edição de texto



Por último, lembre-se que se fazer alterações na tabela deve carregá-la novamente no Translhextion para vê-los.

3.4. Edição do texto

Já estamos capacitados para poder editar o texto de um arquivo. Editar o texto com o Translhextion não tem mistério, se bem que é surpreendente que tem havido muitas pessoas que tem problemas para entender o modo de edição. A única coisa que tem que fazer é situar-se no caractere a partir do qual se deseja traduzir e pulsar a tecla <ENTER>, Deste modo aparecerá uma janela de edição do Translhextion.

Eu tenho outra forma editar o texto com o Translhextion baseada na maneira em que se faz no Thingy. Vá ao dialogo que queira traduzir, situe-se no último caractere que deseja editar e vá até ao primeiro caractere do dialogo ou frase que deseja traduzir selecionado o texto (com a tecla shift).Preste atenção nos bytes selecionados e aperte a tecla <ENTER>. Quando terminar de traduzir volte a teclar <ENTRE>, ainda recordando dos bytes selecionados se não queira problema. Se sobrar caracteres, simplesmente preencher os bytes restantes com espaços. Lembrando de fazer copias de seguranças com freqüência dos arquivos que edita é sempre um bom costume. Levando em conta o número máximo de caracteres que pode ter uma linha e quantas linhas pode ter um caixa de texto, já que se os exceder pode provocar uma série de conseqüências.

CAPÍTULO 4: EDIÇÃO DE GRÁFICOS

É mais que provável - especialmente se, além disso, já tentou traduzir um jogo - que já se tenha perguntado durante a leitura deste manual que os acentos não aparecem em canto nenhum dos jogos. Talvez se quer tenha pensado na possibilidade de que não exista nenhum problema relacionado com isso, algo também bastante óbvio. Bem, o que acontece é que nos jogos em inglês (ou japoneses, a questão é que não estejam em português) não existem tais caracteres, simplesmente porque não são utilizados. Por tanto, se queremos escrever uma letra tão representativa de nosso idioma como o "ã", antes devemos inseri-la na fonte que é utilizada pelo jogo.

Os editores gráficos permitirão realizar essa tarefa que requer trocar alguns gráficos por outros que não utilizemos na ROM, como por exemplo o apostrofo ('). Depois de tudo, toda a fonte da ROM se baseia em um conjunto de gráficos, assim, a ROM se limitará em mostrar os gráficos que estão armazenados em um determinado lugar do arquivo. Durante todo este manual me referirei a cada quadradinho que aparece em um editor gráfico com *tile* já que é esse o nome comumente dado pelos *expert* no assunto (é preciso ter em mente que muitas palavras são plágios tomados desnecessariamente do inglês).

Por outro lado, tenho que dizer que trocar gráficos não é como retocar imagens com um programa como pode ser o Paint do Windows, pois todos os gráficos de um arquivo aparecem "desordenados". A primeira vez que carregar uma ROM em um editor gráfico te dará a impressão de que está tudo corrompido quando ver todos os gráficos da maneira que aparecem. Algumas vezes é mais fácil encontrar a fonte do jogo, mas outras vezes não será tão simples porque podem estar comprimidas ou usar um formato de tiles desconhecido, para citar algumas possibilidades.

4.1. Programas necessários e outros requerimentos

- *Tile Layer Pro*: Suporta quase todos os formatos de ROMs e possui um muito útil clipboard no qual se pode arrastar os tiles "desordenados" de maneira que a edição de gráficos seja feita mais fácil e comodamente.
- Tile Molester: Suporta ainda mais formatos que o Tile Layer Pro e permite "arrumar" gráficos como veremos mais em seu momento, embora para mim é muito mais fácil editar os gráficos com o Tile Layer Pro. Foi programado em Java, por isso talvez seja necessário instalar as bibliotecas de execução do Java, pois sem elas não será possível executa-lo.

CAPÍTULO 4: Edição de gráficos

 O arquivo que se deseja alterar os gráficos. Tomarei neste caso como exemplo a ROM de Zelda do SNES e a ROM francesa de Final Fantasy Mystic Quest.

4.2. Editando gráficos

4.2.1. Gráficos visíveis

O que faremos primeiro será executar o Tile Layer Pro e abri a ROM de Zelda de SNES. Em seguida devemos clicar em *Format* no menu *View* e selecione o modo gráfico adequado. O lógico será utilizar o modo SNES neste caso, será? Nada mais distante da realidade. Resultado irônico, pois em algumas exceções é preciso escolher um formato que não corresponde com o sistema do console. O mais normal é que um jogo de SNES use o formato do Game Boy para a fonte de uma ROM, inclusive o modo *1bpp*. Por tanto, é recomendado optar por estes dois modos antes que nenhum outro para visualizar a fonte.

Uma vez dito isso, agora é só ir deslocando para baixo no editor.Enquanto faz isso você poderá ver de vez em quando entre muita desordem alguns gráficos do jogo em questão. Continuando encontrará lá pela metade da ROM de Zelda de SNES uma imagem como o abaixo:



Se quer trocar, por exemplo, o apostrofe(') que esta situado depois da flechinha que assinala à direita e antes do primeiro pedaço do coração pelo "ã" basta selecionar o tile em questão e, na janela cujo nome é *Tile Editor* podemos altera-lo de acordo com nosso gosto. Para desenhar um "ã", a melhor opção é pegar o tile a - que é igual ao ã salvo pelo acento - arrastando o tile a posição do apostrofe e assim edita-lo economizando esforços.

Quando aparecia tal caractere na tela enquanto se jogava o que aparecerá agora será o que nós tínhamos desenhado. Se editar um tile cujo valor não tem na tabela, simplesmente some o tile com o último tile que tenha na tabela. Para que fique mais claro, vamos ao seguinte exemplo. O símbolo que fecha parênteses ")" é o último caractere e este possui o valor hexadecimal de 46. Se editar esse estranho símbolo que aparece justamente antes do rosto do Link, o "e" comercial (&), contamos desde o 46 um a um até chegar a esse tile e obteremos como resultado o 49.

4.2.2. Gráficos que utilizam um tamanho diferente do padrão

É muito provável que não tenha encontrado a fonte de uma maneira clara, mas tenha certeza que diante dela. Você a verá tão desordenada que será praticamente impossível edita-la com comodidade. Isto acontece, por exemplo, se tentarmos editar a fonte do Sercret of Mana com o Tile Layer Pro como se pode ver nesta imagem (que foi tirada como o Tile Molester):



Para poder editar fontes que tenham tal inconveniente deveremos usar o Tile Molester. Supondo que já tenha instalado as livrarias de execução do Java, o executaremos. Para isso, abra uma janela do MS-DOS (no Windows XP esta no *Iniciar > Todos os programas > Acessórios > Prompt de Comando*) vá até a pasta (lembre de digitar *cd <pasta>* para andar pelas pastas) onde esta o Tile Molester e digite *Java -jar tm.jar* como se vê na imagem:

CAPÍTULO 4: Edição de gráficos



Claro que, também se pode criar um arquivo .bat com tais comando para agilizar sua execução por precisar de apenas um duplo clique sobre ele. Simplesmente bastará escreve-los em um arquivo de texto e salva-lo com extensão .bat.

Não há com o que se preocupar caso demore para carregar, já que isso é algo normal. Uma vez carregado, abrimos a Rom de Secret of Mana e vamos a posição onde vimos a fonte com o Tile Layer Pro. Embora a fonte de Secret of Mana pode ser vista de leve no modo do Game Boy (codec *2bpp* no Tile Molester, ela é melhor vista em *1bpp*. A grande variedade de codecs do Tile Molester pode ser escolhida no menu *Codec* do menu *View*. É conveniente ativar o Tile *Grid*) do menu *View* para ver um grade que separa os tiles, embora neste caso não nos será de grande utilidade. Agora só nos cabe usar as teclas direitas e esquerdas do teclado mantendo apertada a tecla Shift para trocar as colunas. Também podemos trocá-las em *Image* > *Canvas Size*.



Vejamos outro exemplo. Desta vez carregaremos a ROM francesa do Final Fantasy Mystic Quest de SNES. Se usarmos o Tile Molester e avançarmos pela ROM nós encontraremos a fonte, se bem que neste caso é certo que se visualizará melhor usando o codec *2bpp*. Use o *Tile Grid* que será muito útil nesse caso. A fonte se encontra muito obscura, para tentarmos vê-la melhor selecione o *Mode 2 Dimension* no menu *View*. A fonte fira muito mais legível mas ainda assim não poderá ser edita com facilidade. Para poder vê-la como vemos a fonte de Secret of Mana o que devemos fazer é trocar as colunas do *Canvas Size* para 32 como se vê na imagem:



4.3. Alterar a largura de uma fonte de largura variável(VWF)

Para explicar esta separação tomaremos de novo como exemplo o Zelda de SNES. Este jogo utiliza o que se denomina *VWF (Variable Width Font,* quero dizer, *Fonte de largura Variável*), o que quero dizer é que no jogo não se apresentará o sinal de exclamação (¡) como pode ser vista no editar, sinal que mostrará os pixels que ocupa. Para isso, a ROM contem uma área ou tabela como normalmente é denominado na qual se armazenam os valores de largura de cada caractere. Deste modo, se substituir o 'W' pelo sinal (¡), este será mostrado na tela junto com vários pixels vazios ao redor, já que utiliza a largura do 'W', cirando um feito nada vistoso:



Vejamos como podemos resolver esse problema. Abrimos o Tile Layer Pro, carregamos a ROM do Zelda nos dirigimos ao local onde está a fonte. Prestamos atenção nos pixels que cada tile ocupa. Vamos pegar o tile da letra 'A'. Na janela *Tile Editor* podemos ver claramente de quantos pixels é formado este tile:

CAPÍTULO 4: Edição de gráficos



Podemos observar se contarmos os pixels da esquerda para a direita que são 6 pixels os que formam o tile. Muitas vezes o último pixel de um caractere é um espaço, ainda que Zelda não é o caso. Do contrário, em vez de 6 pixels falaríamos de 7. Praticamente todos os demais caractere ocupam os mesmos pixels, salvo a letra 'I' ou o 'j' para citar exceções.

Agora executamos o Translhextion e carregamos a ROM de Zelda. O que vamos fazer é procurar os códigos que representam o número de pixels que se mostram por tile. Como já foi dito que cada caractere se compõe em geral de 6 pixels, converteremos tal cifra para hexadecimal, quero dizer, 06. E damos um *Find* no menu *Search*(lembre de ativar a procura por hexadecimais) e buscamos 06 06 06 06. A razão disso é que na suposta tabela de valore de largura de cada caractere "abcd" se representaria com esses valores tendo em conta a largura dos tais caracteres.

É muito provável que se encontram nesta seqüência, ainda se não é o caso deveríamos provar a busca por 07 07 07 07 (como foi dito anteriormente, alguns caracteres possuem um pixel a mais que serve como espaço). Lembre-se que deve haver muitos códigos parecidos ao redor, já que nem sempre vamos encontrar o que realmente procuramos de primeira. Uma vez encontrado o que procurávamos, veremos que há alguns 03 (3 são os pixels do 'i') entre os 06. Se contarmos quantos caracteres existem entre o 'I' e o 'A', ambos inclusive, veremos que existem 9. bem, se contamos 9 a partir do primeiro 06 nos daremos conta de que, efetivamente, correspondem ao alfabeto. Agora fazemos o mesmo, mas entre o sinal de exclamação (¡) e o 'A', que é o nos interessa. Se fomos substituir o 'W' (que possui 7 pixels) por esse sinal, o que devemos fazer é procurar a largura do 'W', que esta a 22 caracteres do 'A'. O sinal (¡) possui 3 pixels, assim que trocarmos o valor original do 'W', que é 07, por 03. Todo isso pode ser visto mais claramente na seguinte imagem:

	[C:\Traduccion	esWilldadesScare	child?Legend of Zela	da, The (U).andJ - T	ranshextion						
File Selection	n Offset Insert	t Search Bookmarl	ks Script Options	View Help							
00074C80 00074C90 00074C80 00074C80 00074C80 00074C80 00074C80 00074C80 00074020 00074010 00074010 00074010 00074050 00074050 00074050 00074080	$ \begin{array}{ccccc} CC & FD & CC & F\\ CC & FD & CC & F\\ 1C & C9 & 13 & C\\ E2 & 30 & 4C & 8\\ 7F & C9 & 59 & F\\ 1C & 0A & AA & F\\ 06 & 06 & 06 & 07 & C\\ 06 & 03 & 05 & C\\ 06 & 03 & 05 & C\\ 04 & 04 & 06 & C\\ 04 & 04 & 06 & C\\ 04 & 08 & 88 & 4\\ 00 & 00 & 40 & C\\ 8B & 4B & AB & C\\ 8B & 4B & AB & C\\ 8B & 26 & 07 & F\\ 64 & 03 & AE & T\\ \end{array} $	$ \begin{array}{ccccc} FD & CC & FD & CC & FD \\ FD & CC & FD & CC & F3 \\ FD & CC & F0 & FC & 9 & 334 \\ C9 & E2 & 30 & 20 & 5E & C \\ 344 & C9 & E2 & 30 & A \\ C2 & 30 & 20 & 5E & C \\ 364 & C9 & E2 & 30 & A \\ C0 & 00 & 00 & 00 & 00 \\ 06 & 04 & 06 & 06 & 0 \\ 06 & 04 & 06 & 06 & 0 \\ 06 & 04 & 06 & 06 & 0 \\ 018 & 06 & 06 & 06 & 0 \\ 018 & 06 & 06 & 06 & 0 \\ 018 & 08 & 08 & 08 & 0 \\ 018 & 08 & 08 & 08 & 0 \\ 018 & 08 & 08 & 08 & 0 \\ 018 & 00 & 00 & 7F & B \\ C2 & 20 & AD & 20 & 0 \\ 039 & 50 & CB & 8D & 2 \\ 039 & 50 & CB & 8D & 2 \\ 039 & 50 & CB & 8D & 2 \\ 030 & 1C & BF & 00 & 1 \\ \end{array} $	$ \begin{array}{ccccc} \text{PD} & \text{CC} & \text{FD} & \text{CC} & \text{FD} \\ \text{D} & \text{CC} & 20 & \text{B8} & \text{CA} \\ \text{B} & 00 & \text{FO} & 0\text{A} & \text{C9} \\ \text{O} & \text{C2} & 10 & \text{AE} & \text{D9} \\ \text{9} & \text{OC} & \text{8D} & 2\text{F} & 01 \\ \text{B} & \text{AD} & \text{D6} & 1\text{C} & \text{8D} \\ \text{B} & \text{AD} & \text{D6} & 06 & 06 \\ \text{O} & 6 & 06 & 06 & 06 \\ \text{O} & 6 & 06 & 06 & 06 \\ \text{O} & 6 & 06 & 06 & 06 \\ \text{O} & 6 & 08 & 08 & 08 \\ \text{8} & 04 & 08 & 08 & 08 \\ \text{8} & 04 & 08 & 08 & 08 \\ \text{8} & 04 & 08 & 08 & 08 \\ \text{9} & \text{C2} & 01 & 00 & 00 \\ \text{F} & \text{DF} & \text{EF} & \text{F7} & \text{FB} \\ \text{7} & \text{F0} & 12 & \text{AC} & 22 \\ \text{4} & 07 & 9\text{C} & 20 & 07 \\ \text{4} & 27 & \text{FC} & 10 & \text{A8} \\ \end{array} $	$ \begin{array}{cccc} {\rm FD} \ {\rm CC} \ {\rm FD} \ {\rm CC} \ {\rm FD} \ {\rm CD} \ {\rm DD} \ {\rm DD} \ {\rm G3} \ {\rm 00} \ {\rm FD} \ {\rm 0D} \ {\rm D5} \ {\rm 1C} \ {\rm BF} \ {\rm 00} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 12} \ {\rm C2} \ {\rm 30} \ {\rm AD} \ {\rm DD} \ {\rm D5} \ {\rm 10} \ {\rm 60} \ {$	İýİýİýİýİýİýİ Iýİýİýİ, EÅ0- .E. 3.É; 3.Éc. 3 AOL EAO Å.©Ù. 2. .EY3.A0©. ∕.Å0- A ≇AO °E-Ö. Ö. 						
00074090	02 AF 24 0	17 1 7F 30 C	2 7F 9F 31 C2	7F F8 8F 24	.@\$0Å~.1Å~è.	\$					
00074 Imd <t< td=""></t<>											
00074 00074 00074 00074 00074 00074	Match case Di	Dup Down	ata Type D Text I Hex	OK Cancel	-&iP¢¥ Z@ .¿∕Å~)ÿe.`)ĕ .e.³.Jâ¥ ¿YBÉ ¿	i)\$).					
Offset 478431	=0x74CDF Bits=0	00000110 Unsi	gned: B:6,W:1542,L:1	01058054 ANSI /	/ OVR / L Size: 10490	188					

Uma vez terminado, salvamos as alterações. Se tudo for alterado corretamente, agora aparecerá o tile como deveria, como podemos ver nesta imagem:



4.4. Para saber mais

Console Graphics Document (por Klarth): Vem explicando o formato de cada console. Bastante útil se você tentar analisar uma fonte comprimida que possui partes descomprimidas como a do Phantasy Star II (que utiliza uma compressão RLE).

CAPÍTULO 4: Edição de gráficos

- *Editing fonts without the aid of a Graphics Editor!* (por Neil_): Não só te servirá para editar gráficos sem a ajuda de editor gráfico (algo que pode ser muito útil para copiar a fonte de um jogo para outro sempre que elas usarem o mesmo modo gráfico e paleta), todavia que explica no que se baseia uma tile.
- *Title Screen Hacking Made Easy* (por InVerse): Se está interessado em trocar os gráficos da tela de apresentação, este documento pode ser útil.
CAPÍTULO 5: PROCURA POR TEXTOS COMPRIMIDOS

É possível que com o método normal de procura que vimos no segundo capítulo deste manual não encontrarmos nada. O que é normal, sobre tudo em jogos programados pela companhia Squaresoft, pois o texto está comprimido. Adiante veremos o que podemos fazer quando se está diante desta situação.

5.1. Programas necessários e outros requerimentos

- *Martial*: Programa que se encarrega de procurar a tabela DTE ou MTE mais optimizada para nossos scripts.
- O arquivo que se deseja traduzir. Neste caso tomaremos como exemplo a ROM do Final Fantasy III(USA).

5.2. Compressão DTE e MTE

5.2.1. Averiguar a compressão DTE e MTE

Muitos jogos de SNES empregam as técnicas DTE (Dual Tile Encoding) e MTE (Multi Tile Encoding) para comprimir o texto. Um código DTE comprime em um byte dois caracteres (daí o seu nome), pelo que, supondo, a palavra "Hola" ocuparia dois bytes em vez de quatro. "Ho" ocuparia um byte e "la" outro. Um código MTE é igual a um DTE coma diferença de que pode comprimir mais de dois caracteres em um só byte, ainda que às vezes o número de caracteres comprimidos é tão grande que são codificados em dois bytes. Um bom exemplo disto podemos encontrar na ROM de Lufia de SNES.

Agora veremos como decifrar um texto que está comprimido por meio de DTE, já que é mais simples aprender com DTE pois ainda que o texto comprimido com MTE se baseie nos mesmo princípios, fazer uma tabela com códigos MTE é mais pesado e demorado. A ROM de Final Fantasy III de SNES (versão USA) utiliza esse tipo de compressão, deste modo o primeiro passo será executar o SearchR X e carregar a ROM. O que temos que fazer é procurar silabas fáceis, como pode ser "cas", "ca" ou "lo". Se não encontrarmos nada, deveremos tentar com outras silabas que nos vier a mente (claro, devem formar parte de uma palavra inglesa). Com um pouco de sorte logo encontraremos algo.

CAPÍTULO 5: Procura por textos comprimidos

Para termos certeza de que são os valores que procuramos, podemos editar o texto que aparece e ver se há alguma coisa diferente quando jogamos. Para isso, devemos olhar em que posição está o texto encontrado e assim ir até ela com o Translhextion por meio da função *Jump* to. Este método possui o inconveniente de que pode levar muito tempo, mas se o texto está com esse tipo de compressão é a tarefa mais fácil para se dar cabo dela.

Porém, há um método mas fácil e quase sempre funciona, que é aproveitar-se do texto que está em maiúscula. O que quer dizer isso? Que devido a sua pouca freqüência nos textos é mais provável que todas as palavras que todas as palavras que estejam em maiúsculas não estejam comprimidas. O inconveniente é que devemos recordar de um texto que apareça em maiúsculo e que se possa visualizar no emulador para verificar as mudanças.

No Final Fantasy III (USA) os nomes das personagens não protagonistas estão em maiúsculas. Além disso, existe um diálogo justamente no inicio, que trás a tradução, onde aparecem dois desse nomes em maiúsculas: VICKS e WEDGE. Por tanto, o seguinte a se fazer é executar o SearchR X para procurar por VICKS:

8	Search	RX 😑 🗆 🗷
Archivo Opciones Ayuda		
Archivo Opciones Ayuda Offset [dec] 293766 852480 852569 852640 853496 1103052 1103110 1103181 1103295	Valores A=80 a=?? 1=?? A=20 a=?? 1=?? A=20 a=?? 1=?? A=20 a=?? 1=?? A=80 a=?? 1=?? A=80 a=?? 1=?? A=80 a=?? 1=?? A=80 a=?? 1=??	Vista Previa DUIN#####WEDGE#VICKS######## #W###########WICKS#T###### #W#############WICKS#AF####### ##################UICKS#AF###################################
Búsqueda Relativa Búsqu Busca: VICKS De: 0	eda a 16bit Tiempom ínimo: 0 mi a: 3146240	n. 3,517 sec. ○ Hex ⓒ Dec Buscar Parar

A equivalência do primeiro resultado tem toda a pinta de ser a tabela dos menus porque está junto a outros nomes, mas fazer e completar tal tabela não é o objetivo deste trabalho. Se nos fixarmos nas equivalências do resto dos resultados, todos são 20=A. Ainda não se vê texto ao lado de VICK, resulta estranho que sempre mostra a mesma equivalência.

Agora devemos fazer a tabela das maiúsculas (lembra que 20 equivale a 'A') tal como já explicamos no segundo capítulo fazendo uso do TaBuLar. Como também já comprovamos antes, justamente depois de 'Z' maiúsculas vem o 'a' minúsculo. Assim que trás o valor de 'Z' clicamos em *Insert Lowecase Alphabet* nos menu *Insert*. Feito isso, salve a tabela e voltamos para o Translhextion. Carregamos a ROM de Final Fantasy III (USA) e tabela que acabamos de fazer e damos um *Find using Table* nos menu *Search*. Se procurarmos por WEDGE no lugar de VICKS, nós encontraremos o que pode ser visto na seguinte imagem:

9	ļ	C:\Tradu	iccio	nes	Wtilid	lade	s\Sea	irchl	OC/F	inal	Fante	isy 3	(91.	0) (U).sm	ej - T	ranslhextic	in e	
File	Selection	Offset	Inse	rt	Seard	h	Bookm	arks	Scr	ipt	Optic	ons	View	He	lp				
	. Datitia Dout	1017	22	2Å	32	61	7F	33	85	89	B4	81	85	81	C7	47	VICKS#	#T######n	-
	T1 - 12	A. 17	18	12 3F	13	36	24	23	26	24 BC	61 4B	7F 63	27	A6 C9	8C 3.4	96 XB	##### <mark>\</mark>	<mark>╘ⅅĠ╘</mark> ##Н### #⋤ _⋑ # _` ######	
Ľ	I hingy Viev	N Active	97	D2	ÉÉ	84	83	87	3E	66	7F	55	54	54	54	7F	######	##e#######	
	Japanese		Α6	86	3A	3F	BA	93	83	80	36	ЗA	93	D8	81	85	ye##af	####Wa####	
	Shift-JIS	#	F7	42	67	16	18	12	00	35	28	22	24	32	61	7F	#M#i##	###VICKS## ######:#	
	EUC-JIS	\square >	24	23	26	24	61	00 7F	2E	ČÅ.	38	34	3B	45	52	67	#WEDGE	######1VE# ##P#hahlv#	
000	D0280	01 67	43	4Ĕ	3D	40	42	9Ã.	ĀB	ČĂ	D3	83	80	ΒĔ	FE	47	##judg	i#####,##n	
000	D0290	3C A3	D8	92	BE	92	4B	3D	87	4C	65	7F	16	18	12	00	c####	rd#s######	_
000	DU2AU	35 28	-22 - AD	2A 4C	32	61 DE	75	20	B5	81	AD FD	86	50	48	D4	47	VICKS#	#A####wo#n	
000	D02D0	41 80	85	89	SF	16	18	12	13	Å9	85	A6	8C	4C	41	80	h#####	########sh#	
000	D02D0	3F 4B	42	3Ē	8C	59	54	92	Ĉ5	8B	93	2C	3A	40	42	BA	frie##	#####Magi#	
000	D02E0	F0 20	4B	46	48	89	- 8C	DE	EC	42	3E	C8	97	94	4E	47	#Armo#	###ie###un	
000	DU2FU	DD 93	57	6 24 61 7F 2D 48 4D 81 90 50 48 4B 4B 52 EDGE##Not##worry															
000	D0310	65 7F	33	3 85 7F 32 45 3A 99 7F 22 CA 50 47 92 94 ##T##Sla##C#wn##															
000	D0320	85 93	85	5 3A 3D D1 48 3B 86 85 93 D8 88 A7 B2 9F ###ad#ob########															
000	D0330	4C 3C	42	42 48 CB 81 D7 4E 40 41 4D FB 32 41 3E 63 scio###ught#She#															
000	DU340	A7 AB	48 45 EA F5 B0 DD C8 65 16 18 12 00 36 24 ##ol##########WE																
000	D0350	4B C3	81	85	7F	3E	BD	4D	65	01	2C	49	99	92	E4	5E	r####e	#t##Mo####	
000	D0370	16 OC	12	ŌŌ	7F	2B	9F	Č2	F7	48	66	81	85	7F	36	Å6	#####L	###o####W#	
000	D0380	92 C5	83	80	2C	34	40	42	01	7F	7F	7F	7F	D1	AA	4E	####Ma	gi ###### u	
000	D0390	SC AA	81	85	8D	BU	EC	81	90	34	40	40	30	BU 01	FC 1.4	AA	C##### ###1~#	###a#sc### ####aio###	
000	DOGRO	4C 42	2 46 49 F9 B2 A8 B7 8C 96 7F 3E 51 42 9E FB simp####################################																
000	D03C0	11 FF	12 00 55 54 54 54 AC A8 C8 91 3A 4F 80 49 ##################																
000	D03D0	3A 4C	4C	AA	67	7F	28	4B	9F	66	01	40	4E	47	49	<u>C7</u>	ass###	Ir###gunp#	
000	DU3EU	DD 4B	66	88	B5	8E	BA	3A 2D	46	71	3E	9A 40	AR A	AF 07	01	7F	#r####	#am#e#####	
000	D03F0	8F 3D	01	14	05	AD	DF	81	3E	30	41	40	48	45	48	40	#4####	##fechnolog	
ŏŏŏ	D0410	A3 89	42	40	47	4C	67	7Ē	11	FF	12	ŐÖ.	7F	7F	7F	7F	##igns	##########	
000	D0420	21 4E	84	83	87	80	A 6	80	DE	Δ2	8D	D7	01	7F	7F	50	Bu####	#########	
4	<u> </u>																		• •
Selec	ed: Offset 852501=0xD0215 to 852505=0xD0219 (5 byte(s)) ANSI / OVR / L Size: 3146240 🥖																		

Parece que, efetivamente, depois de 'Z' vem o 'a', porque somos capazes de ver alguns textos legível entre tanto #. Se não fosse assim, seria questão de substituir alguns códigos por outros e verificar o resultado jogando o jogo no emulador para ver o que aparece na tela.

Já demos um grande passo, pois encontramos o texto, embora agora vem o que não é difícil, mas que leva tempo: completar a tabela.É preciso ter em mente que um código hexadecimal equivale a dois caracteres e não a um devido a compressão DTE, embora ainda nos falte colocar os sinais de pontuação, não podemos esquecer que um # também pode equivaler a um só caractere. Como sabemos que depois de cada nome de personagem há dois ponto (:) e um espaço, é fácil verificar tais valores. Basta ir até onde esses valores estariam, e acrescenta-lo na tabela.

CAPÍTULO 5: Procura por textos comprimidos

Logo perceberemos que o código de quebra de linha é o 01 e o de fim de caixa de texto é o 00. De fato, o de quebra de linha é tirado por dedução após analisar o final o fim da caixa de texto, já que é bastante freqüente em uma ROM que 00 seja o código de fim de texto e 01 o de quebra de linha. Uma vez encontrados e colocados todos esses valores na tabela será mais deduzir os valores das DTEs, então voltamos a carregar a tabela com os novos valores acrescentados, nós encontraremos isso:

-		C:\Tradu	lecia	nes	Wtilia	ladi	es\Se	urchl	RXNF	inal	Fanta	isy 3	(V1.	0) (U).sm	e] - T	ranslhextic	an e	
File	Selection	Offset	Inse	ert	Searc	:h	Bookn	narks	Scr	ipt	Optic	ons	View	He	lp				
	lable bui	160	22 18	2A 12	32 13	61 36	7F 24	33 23	85 26	89 24	B4 61	81 7F	85 27	81 A6	C7 8C	47 96	VICKS: #####W	T######n EDGE: H###	-
	Thingy View	w Active	42	3E D2	99 EE	88	94 83	24 87	4C 3E	BC 66	4B 7F	63 55	86 54	C9 54	Å4 54	AB 7F	##ie## ######	#Es#r#### ##e# ####	
	Japanese change	- +	<u>A6</u>	86	34	1	BA	93	83	80	36	34	93	D8	81	85	ye##af	####Wa####	
Ĭŏ	SHIKVIS	= +	47	42	97	4D	18 - 18	86	9E	-35 F8	28 B1	Č0	42	32 4F	3E	5F	M#1## T#nk#t	##\VICK5: ######ive#	
	EUCOIS		24	23	26	24	61	7F	2F	CA	3B	3Å	3B	45	52	67	*WEDGE	: P W bably#	
000	D0280	01 67 3C A3	43 D8	4E 92	3D BF	40	42 4B	9A 3D	AB 87		D3 65	83 7 F	80	BE 1.8	FE 12	47	*#judg	i########n ~d#e#_###\	
000	D02A0	35 28	22	2Å	32	61	- 7F	20	BŚ	81	AD	86	50	48	$\overline{D4}$	47	VICKS:	A####wo#n	
000	D02B0	66 81	AD	4C	67	DE	4B	3C	87	87	FB	CF	52	63	86	4C	###s## ~~~	rc####y##s ########	
000	D02D0	3F 4B	42	3E	8C	59	54	92	Č5	8B	93	2C	3A	40	42	BA	frie##	######Maqi#	
000	D02E0	F0 20	4B	46	48	89	80	DE	EC	42	3E	C8	97	94	4E	47	#Armo#	###ie###un	
000	D02F0 D0300	24 23	26	24	46	42 7F	47 2D	4E 48	4D	4C 81	90	16 50	48	4B	4B	36 52	### mi EDGE:	nu#s####\W Not##worrv	
ŏŏŏ	D0310	65 7F	33	85	7Ē	32	45	ЗĂ	99	7Ē	22	ČĂ	50	47	92	94	# T# S	la# C#wn##	
000	D0320	85 93 40 30	85	3A 48	3D	D1 81	. 48 D7	3B 4 F	86	85	93 4D	D8 FB	88	A7 41	82 3 F	9F 63	###ad#	ob######### #uab+#She#	
000	D0340	A7 AB	48	45	EA	F5	ВÓ	DD	Č8	65	16	18	12	õõ	36	24	##ol##	#######\WE	
000	D0350	23 26 40 C2	24	61	61 7F 36 3E 63 A7 88 49 49 CA EE DA 3F DGE: We###pp###f														
000	D0360 D0370	46 C3	12	00		2E	3 9F	4D C2	F7	48	66	81	85	7F	36	A6	###\ L	#t#*no### W#	
000	D0380	92 C5	83	80	2C	34	40	42	01	7F	7F	7F	7F	D1	AA	4E	####Ma	gi* ##u	
000	D0390	3C AA	81 BA	45	8D 34	-B0 -B5	- EC - 98	81 8F	90 8C	3A D4	40	40	30	BU 01	FC 14	AA 05	C##### ###1a#	###a*sc### ####aic*##	
ŏŏŏ	D03B0	4C 42	46	49	F9	B2	A8	B7	8Č	96	7F	3Ē	51	42	9Ē	FΒ	simp##	#### exi##	
000	D03C0	11 FF	12	00	55	54 25	54	54 4B	AC	A8 66	C8	91	3A 4 F	4F	80	49	###\##	######av#p Tr####avp	
000	DO3E0	DD 4B	66	88	B5	8Ē	BA	3A	46	7F	ЗĒ	9A	8Å	ÅF.	Ōí.	7F	ass## #r####	#am e###*	
000	DO3F0	7F 9B	99	A1	DC	94	89	3D	42	4C	30	48	4F	87	AA	98	#####	#discov###	
000	D0400 D0410	A3 89	42	$\frac{14}{40}$	47	40	67	81 7F	3E 11	FF	12	47	48 7F	45 7F	48 7F	40 7F	# d *### ##igns	##ecnnolog # ###\	
nnn	D0420	21 4E	84	83	87	80	A6	80	DE	Α2	8D	D7	01	7F	7F	50	R11####	###### * v	
4	L OFOCOL	0.00070	Dik		0101		F1 1		D. 00	o	1500		00/7			ALC T		Ci 0141041	
Offse	et 852601=	UXD0279	Bits	=11001010 TUnsigned: B:202,W:15306,L:993672138 ANSI / OVR / L Size: 3146240											1				

O que devemos fazer é ir pouco a pouco clareando os #. O código DTE que está justamente onde está o cursor na imagem tem a pinta de corresponder a "ro", olhamos seu valor hexadecimal e o acrescentamos na tabela da seguinte maneira: CA=ro. Para começar a traduzir só nos resta completar a tabela com os valores dos códigos DTE. Por ser uma tarefa um pouco árdua recomendo por em prática o truque de colocar uma sucessão de códigos hexadecimais para averiguar seus valores como vimos no terceiro capítulo. Porém recomendo por um espaço (neste caso o 7F) entre código e código porque pode ficar difícil ler tantos caracteres juntos e por tanto averiguar quais caracteres correspondem a cada código DTE.

Sem problema, o jogo deve ter em algum canto a lista de compressão com todos os caracteres comprimidos, bem juntos ou bem separados por algum código. O que temos que fazer é procurar os valores dos códigos de forma seqüencial usando uma tabela que não contenha nenhum valor DTE ou MTE. Por tanto, se possuirmos os valores de, por exemplo 80 ("e") e 81 ("t"), vamos ao inicio do Translhextion, carregamos a tabela sem nenhum valor DTE e lhe damos um *Find using Table*. Se procurarmos "e t" acharemos o seguinte:

9 2	Į.	C:\Tradu	iccia	nesi	Wtilia	lade	sSea	urehl	00F	inali	Fanta	isy 3	(91)	0) (U).sm	e]-1	Iransthextic	in (00	×
File	Selection	Offset	Inse	ert	Searc	h I	Bookm	arks	Scr	ipt	Optic	ons	View	He	lp					
	labia bool	160	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	###### ######	********** ******	# #	•
~	Thingy View	w Active	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	FF FF	######	########## ##########	# #	Н
	Japanese cและแต	• •	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	######	*******	¢	
ŏ	FUCAIS		FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	FF	######	********	* #	
000	015190	FF FF	FF	FF	FF	FF	FF	FF FF	FF	FF	FF	FF	FF	FF	FF	FF	######	########## ******	¢ *	
000	0E1A0	3E 7F	7F	4D	61	7Ē	4D	41	4D	ŽĒ	41	3Ē	4 <u>C</u>	7Ē	3Ē	4B	e t:	tht hes e	r	
000	0E1B0 0E1C0	7F 3A 48 7F	4B 7F	3E 41	42 7F	47	48 4B	4E 7F	3D 47	7F 7F	3A	50 4D	4D	4C 48	3A 7F	47	o h o	pud w san r n atto :	n i	
000	0E1D0	66 7F	4F	3E 3B	47	40	41	3A 75	7F 3E	46	33	41 4D	4C	$\frac{4D}{4B}$	48	47	# veng	ha mThstor	n I	
000	0E1E0	3E 3A	28	<u>7</u> E	3E	3D	7Ē	3Ē	7F	52	41	42	42	4 <u>C</u>	3E	4 <u>C</u>	eal ed	f yhiise:	S	
000	0E200 0E210	48 4B 7F 28	45 3A	7F 7F	4D	3C 3E	47 7F	3E 45	49	4C 3E	47 3A	3D 4C	45 4E	3E 4B	4C 4E	3E 7F	ori c Ia te	ne#sndles: lpeasuru	•	
000	0E220	3A 45	7F	49	40 4 P	7F	48	46	7F	3D	3F	7F	7F	40 215	48	50	al pg	om [°] df gov	7	
000	0E240	3E 4D	7F	4B	47	$\frac{40}{4D}$	46	7F	46	3Å	28	63	45	42	41	48	et rnt	m mal#liho	5	
000	0E250 0E260	48 3F 3C 3A	38 4B	48 3A	41	-7F -63	7F 4D	47 3A	3E 4E	3E 4D	3D 3E	3E 45	4C 5E	48 7F	40 3F	41	ofYoh caran#	needesogi tautel# fo	n 5	
000	0E270	4D 42	36	3E	45	48	3E	5E	45	3D	47	48	34	3C	3C	3E	tiVelo	e#ldnoacce	•	
000	0E290	42 45	45	52	3C	48	65	7F	3C	41	40	48	40	3E	3Ē	67	illyco	# chgogee	¢	
000	0E2A0 0E2B0	JEZAO BB AB 35 00 00 7E 84 18 4E FF E1 C3 82 21 0A 1A ##V\##################################																		
000	0E2C0	8F 17	38	60	E4	0E	D0	7C	FE	FF	57	EC	FF	3F	00	00	##Y###	#######f\\ *\\\\\\	*	
000	0E2E0	03 20	D2	19	01	00	öö	CŨ	06	10	30	61	00	00	õõ	ŏó	#A##*\	*****Q:\\\\	÷.	
000	0E2F0 0E300	00 00 61 00	00	00	00	00 68	00 0D	00 38	00 5C	00 86	00 5E	00 D3	00 2E	00 91	00 2E	00	: \##\#	*Y####0#0;	÷	
000	0E310	12 7B	FD	00	FF	80	FB	38	FF	00	00	00	00	00	00	00	###\##	#¥#\\\\\\\\ ########	*	
000	0E320	30 38	20	78	30	78	30	38	30	38	30	38	28	7C	ññ	20	07##0#	070707##\;	¢ ¢	
4 Cale		577/0 0		0.1-	F77/			141	. de m C							NICT		Since 01.440	10	-
Selec	Cted: Offset 57760=UXE1AU to 57763=UXE1A3 (4 byte(s)) ANDLY OVR / L Dize: 3146240																			

Por sorte encontramos o procurávamos. Simplesmente devemos fazer a tabela da seguinte maneira:

80= e

81 = t

82=:

83=th

84=t

85=he

...e assim sucessivamente. Ficando com o endereço da posição do local em que começa esta lista de compressão, chamada tabela de DTE e MTEs (usar um *Bookmark* do Translhextion é mais que recomendado), porque precisaremos dela mais tarde.

CAPÍTULO 5: Procura por textos comprimidos

5.2.2. Alterar a compressão DTE e MTE original

Agora veremos como podemos nos aproveitar da compressão para tornar a tradução um pouco mais fácil. Lembre-se que isto o obrigará a ter duas tabelas (a original e a modificada com as trocas que fizemos) e a jogar uma ROM diferente da qual se está traduzindo para ler o texto original porque este ficará ilegível após trocar as DTE e MTEs. Tudo isso será desnecessário se optarmos por usar scripts como veremos no próximo capítulo deste manual.

O que temos que fazer é trocar a tabela DTE e MTE original, porque, por exemplo, trocaremos "st" por "si", sílaba que iremos utilizar muito em nossa tradução. Para isso, devemos ir ao local onde estavam todos as DTE juntas como foi visto anteriormente. O que é preciso fazer é substituir "st" por "si" e salvarmos a troca. A partir deste momento, em vez de aparecer "st" irá aparecer "si" na tela. Devemos repetir esse procedimento até que não reste mais valore de DTE. Enquanto realizar essas trocas, é mais que recomendado fazer uma tabela com os novos valores e não deixa-lo para o final.

Isso levará o texto da ROM aparecer ilegível, daí a necessidade de ir jogando uma ROM sem modificações para saber o que aparecera realmente na tela. Para traduzir você pode usar o Thingyv, já que ele permite carregar duas tabelas por vez. Basta usar o TAB para mudar entre uma e outra.

SSe trabalharmos com scripts nos será muito mais fácil optimizar uma tabela de DTE ou MTE de acordo com os scripts traduzidos. Ainda que tudo relacionado com scripts será tratado nos capítulo 7, creio que seja conveniente explicar isso neste momento.

O programa Martial se encarrega de analisar as repetições mais freqüentes de dois ou mais caracteres juntos por todo o script. Vejamos como optimizar um script mediante este programa. Na guia *Input* devemos especificar o script ou scripts que queremos analisar, além da longitude dos caracteres juntos que se vai procurar. Como queremos buscar DTEsm temos que designar i, valor de 2 na caixa de texto de Sample *Length(s)*. Passemos para a guia *Output* indicaremos um valor de 300 em *Only Show Top X Results*. Este alto valor se justifica muitos das possíveis DTEs não nos servirão por diversos motivos, como que haja um caractere próprio do script (por exemplo, <o>) na possível DTE. Nesta guia devemos especificar os arquivos onde serão salvas as resultados e a nova tabela DTE, algo que nos poupará um esforço considerável. Na terceira guia ativaremos as opções *Ignore Carriage Return-Line Feeds*; uma vez tudo pronto poderemos clicar o botão Analyze. Depois de alguns estantes, aparecerão todos os resultados na tela:

🐴 Martial 0.03	
Input Output Formatting Analyze Results	
Filename:C:\Emuladores\Traducciones\Traducciones\Final Fantasy IV\Scripts traducidos\ff4Text1.txt Total characters: 80958 Characters actually counted: 80958 Number of different strings: 1232 Sample lengths requested:2 Elapsed time: 00:00:01	
0skipped for containing a carriage return linefeed- 1: total space = 3614; hits = 1508; string = ;; hex = 0A20; len = 2 2: total space = 3016; hits = 1508; string = c; hex = 6520; len = 2 3:skipped for containing a carriage return linefeed- 4: total space = 1932; hits = 936; string = a; hex = 6520; len = 2 5: total space = 1932; hits = 936; string = a; hex = 6520; len = 2 7: total space = 1978; hits = 989; string = a; hex = 6720; len = 2 8: total space = 1978; hits = 989; string = c; hex = 7320; len = 2 8: total space = 1890; hits = 945; string = s; hex = 7320; len = 2 9: total space = 1832; hits = 916; string = s; hex = 6573; len = 2 10: total space = 1642; hits = 826; string = c; hex = 6572; len = 2 11: total space = 1648; hits = 826; string = c; hex = 6572; len = 2 12: total space = 1648; hits = 826; string = c; hex = 6572; len = 2 13: total space = 1640; hits = 810; string = c; hex = 6573; len = 2 14: total space = 1596; hits = 754; string = c; hex = 2632; len = 2 15: total space = 1596; hits = 754; string = c; hex = 6432; len = 2 16: total space = 1508; hits = 754; string = c; hex = 6455; len = 2 17: total space = 1508; hits = 754; string = c; hex = 2645; len = 2 18: total space = 1508; hits = 754; string = c; hex = 2645; len = 2 19: total space = 1304; hits = 652; string = c; hex = 2045; len = 2 19: total space = 1304; hits = 652; string = c; hex = 2045; len = 2 19: total space = 1304; hits = 655; string = c; hex = 2045; len = 2 20: total space = 1304; hits = 655; string = c; hex = 2045; len = 2 21: total space = 1304; hits = 655; string = c; hex = 2045; len = 2 21: total space = 1304; hits = 659; string = c; hex = 2045; len = 2 22: total space = 1306; hits = 578; string = u; hex = 5665; len = 2 22: total space = 1306; hits = 578; string = u; hex = 5665; len = 2 22: total space = 1306; hits = 659; string = u; hex = 5765; len = 2 22: total space = 1306; hits = 578; string = u; hex = 5765; len = 2 23: total space = 1306; hits = 578; string = u; hex = 5765; len = 2 24: tot	
Reset << Back	<u>A</u> nalyze

Agora é só dar uma olhada nos resultados e modificar na tabela gerada aqueles valores que nos façam falta, tais como (.<) de acordo com os resultados que aparecem na tela. Lembre-se que ponto e virgula que aparece atrás de cada valor não é colocado na tabela.

5.3. Procurar texto em modo 16 bits

Que o texto se encontra em modo 16 bits não significa realmente que está comprimido, sendo que um caractere ocupa dois bytes em vez de um, ao contrário da compressão DTE. Sinceramente, desconheço a razão disto. Normalmente nestes texto existe um código "normal"(aqui é assim mesmo) junto a outro que sempre se repita (geralmente 00).

Não existem muito jogos com o texto codificado em 16 bits, ainda que, supostamente o Super Metroid de SNES seja um deles. O Zelda de SNES usa este modo para a descrição dos objetos se me lembro bem. Em qualquer caso, é utilizado por um jogo de PC, o Grand Thelf Auto 2. Os jogos de PC só possui este modo ou igual que usam uma tabela ASCII.

É muito fácil localizar o arquivos que contem os textos deste jogo, já que a pasta TEXT e o arquivo que se chama spanish.gxt dentro dela dizem tudo. Desta vez vamos usar o Translhextion para procurar o texto, porque o SearchR X é muito lento e na mostra direito os resultados que se procura em modo 16 bits.

CAPÍTULO 5: Procura por textos comprimidos

Dito isso, executamos o Translhextion e abrimos o arquivo mencionado. No *Search Relative* clicamos em *Search* e digitamos, por exemplo, a*r*m*a*s. O asterisco atua como código comum quando se procura o texto; esse será o código que se repete como já foi dito no início. Não precisa que façamos a tabela com o TaBuLar desta vez, já que podemos salvar a tabela diretamente com o Translhextion. Espero que tudo fique mais claro com esta imagem.

ici	Juegos/GTA Vice City/TEXT/sp	anish.gzifj - Translu	rexiten	
File Selection Offset Insert	Search Bookmarks Script O	ptions View Help		
	Scan Relative	₩ 72	2 00 .^.A.:	s.e.g.ª.r. ▲
Fieldstein		44	A 00 s.e.a.	.q.u.e .e.lª.
Find what:		Scan 72	700 1.t.i.	m.o.!
armas			9 00 .e.s.	t.et.i.
			300 p.o.?.	.C.h.i.c.
Direction		20	00 a.r.o.:	sd.e
O Up	1.00 × 1	5E ST	200l	^.B.o.n.
Down NUTE: Scan	relative is always case sensitive.	ΔĒ	00 [.n.e.	n.a.l
a*r*m*a*s 34890 A=??	a=61 1=??	Generate Table	00 E.s.t.	es.o.y. ? Bue
a*r*m*a*s 35554 A=??	a=61 1=??		00 n.o.,.	.n.e.c.e.
a*r*m*a*s 39506 A=?? a*r*m*a*s 62882 A=??	'a=61 1=??	Save Results ^o 57	700 s.i.t. 500 g.n. 4	oa.l.g. c.a.b.r.!
a*r*m*a*s 125040 A=??	?a=611=??	ίΞ.	00 no.)	b.s.t.i.n.
a*r*m*a*s 170850 A=?; a*r*m*a*s 184294 A=??	? a=61 1=?? ? a=61 1=??	51	JUU a.d.o. LOO s.u.e.	.q.u.e l.t.ea.
a*r*m*a*s 187360 A=??	? a=61 1=??	5I	00 l.g.o.	.d.em.
arrimians 187434 A=?: arrimians 203844 A=??	? а=61 1=?? ? а=61 1=??	13	3 00 1.e.r.(1 00 a.d.a.	d.aC.
a*r*m*a*s 212688 A=??	?a=611=?? 💌	51	100 q.u.e.	.c.o.j.a.
		59	700 su.: 900 g.i.e.	nb.u.g. .r.e.c.i.
00066780 62 00 69 0		0 00 61 00 75	00 b.i.r.	sa.u.
0006CFA0 6D 00 65 0	0 6E 00 74 00 65 00 2	9 00 63 00 61 0 00 75 00 6E	LUU t.o.m. 200 m.e.n.	t.1.C.a. t.eu.n.
0006CFB0 20 00 70 0	0 61 00 6C 00 6F 00 2	0 00 64 00 65	00 .p.a.	l.od.e.
0006CFD0 65 00 67 0	0 5F 00 5C 00 2C 00 2 0 70 00 72 00 65 00 2	0 00 73 00 69	00 .g.o.	1.,S.1. r.eq.u.
0006CFE0 65 00 20 0	0 6C 00 61 00 20 00 7	2 00 61 00 61	200 e. 1.	ar.a.n.
0006CFF0 75 00 72 0 0006D000 72 00 6D 0	0 61 00 20 00 64 00 6 <mark>0 61 00 73</mark> ∎00 20 00 6	5 00 20 00 <mark>61</mark> 2 00 60 00 61	1 00 u.r.a. 1 00 rma:	.a.e <mark>a.</mark> s hla
•				<u>ه ج</u>
Selected: Offset 446462=0x6CFF	E to 446470=0×6D006 (9 byte(s))		ANSI / OVR / L	Size: 468412 🥢

Como não incluímos nenhuma maiúscula procura, só serão salvos os valores das minúsculas. No caso de querer acrescentar os valores das maiúsculas, simplesmente procure por uma palavra que possua uma. Não deve esquecer que, como o código que sempre se repete é 00, terá que acrescentar 00 a cada valor, por exemplo, o 'a' não deve aparecer na tabela como 61=a, e sim como 6100=a.

5.4. Para saber mais

Infelizmente para nós romhackers e felizmente para os programadores, não existe apenas texto comprimidos com DTE e MTE. Se não encontrar o texto o mais certo é que o esteja comprimido por meio dos algoritmos Huffmam ou LZ77 (ou variantes deste) ou outros. Eu nunca trabalhei com nenhum desses tipos de compressão, de modo que não posso oferecer muitos detalhes. Por tanto, proponho ler uns documentos muitos interessantes que podem servir de guia, ainda que, lembrando que não é algo fácil e que requer tempo para compreender a compressão utilizada no jogo. Além disso, em alguns caso necessitará que saiba programar para elaborar uma aplicação que descomprima e volte a comprimir os textos.

- Comprendre la Compression de Huffman (por Ti Dragon y Meradrin): Se até agora não conseguiu muito com o que foi explicado sobre a compressão Huffman, porventura encontrará uma explicação melhor neste tutorial.
- *Compression Tutorial* (por Jay): Explica detalhadamente os diferentes tipos de compressão que se pode encontrar. Muito recomendado.
- Descompresión del Intro del Secret of Mana (por Dark-N): Excelente guia que explica detalhadamente passo a passo como funciona a compressão LZSS (o mais bem uma variante desta) da introdução do Secret of Mana. Todavia, existem algumas partes que são um pouco obscuras até para o autor.
- Entender los Algoritmos de Compresión de Huffman y LZ (por Dark-N): Um estupendo documento em espanhol que fala sobre estas compressões. Também explicam com exemplos para favorecer a compressão do documento.
- Huffman Decoder (creio que seja Bongo): Um programa escrito em C para descomprimir textos que se encontram com esta compressão, ainda que supondo que varia de acordo com o jogo. Inclui o código fonte em C e também uma interessante rotina em *assembly* de SNES.

CAPÍTULO 6: PONTEIROS

Neste capítulo explicarei tudo relacionado com os ponteiros, se bem que será dito nada relacionado sobre a modificação destes. Normalmente se modifica quando utilizamos scripts para traduzir, o que foi preferido deixar a explicação para o próximo capítulo.

6.1. Programas necessário e outros requerimentos

- *WindHex32*: editor hexadecimal alternativo ao Translhextion. Durante o capítulo veremos porque utiliza-lo.
- Lion Pointer Calculator: ainda não foi mencionado neste manual, este programa será muito útil para calcular ponteiros de diferente sistemas (consoles).
- Arquivo que tenha ponteiros. A modo de exemplo utilizaremos a ROM de Final Fantasy III(USA) de SNES, ainda que será mencionado outros jogos.

6.2. Introdução aos ponteiros

Um ponteiro é um endereço que indica desde onde se deve começar a ler o texto de um jogo. O jogo deixa de ler o texto quando encontrar um código de fim de mensagem; esse código é que se pode encontrar ao final de um dialogo, normalmente representado pelos códigos hexadecimais 00 ou FF.

Se já tentou trocar o lugar do código de fim de mensagem, com muita sorte terá conseguido mudar o espaço original dos diálogos, quero dizer, a quantidade máxima de caracteres que podem ser usados em um diálogo. Isso acontece na ROM de Zelda de SNES, mas infelizmente se trata de uma exceção a regra. A resposta para a pergunta de qual é a utilidade dos ponteiros é bem simples: alterando os ponteiros podemos utilizar mais texto que o original. Tentarei explicar seu funcionamento mediante o seguinte exemplo:

Welcome. #We have some problems with that new king. #Hello, my name is Lorena.#

Se optarmos pelo método de trocar o código de fim de mensagem (o # que aparece depois de cada ponto), trocando somente a primeira oração teríamos:

CAPÍTULO 6: Ponteiros

Bem-vindo.#have some problems with that new king.#Hello, my name is Lorena.#

Se tivermos um jogo, como esse texto e o modificarmos tal como se vê neste segundo texto, comprovaríamos de onde deveria aparecer *Welcome* agora se mostra *Bem-vindo*. Sem problema, a segunda oração não mostrará *have some problem with that new king*., sendo simplesmente *ido*. Isso acontece porque há um ponteiro que aponta (daí o nome de ponteiro) ao caractere do começo do segundo diálogo. Por tanto, para alterar o tamanho de um diálogo sem alterar o resto será necessário alterar os ponteiros e não o código de fim de mensagem.

6.3. Como encontrar a tabela de ponteiros

Uma tabela de ponteiros se refere a um lugar do arquivo em que se encontram todos os ponteiros. Estes estão normalmente de forma ordenada, ou seja, que o primeiro ponteiro corresponde ao primeiro diálogo segundo aparece o texto internamente no arquivo, o segundo ponteiro ao segundo diálogo e assim sucessivamente. Existe uma forma padrão para verificar isso, ainda que com o tempo espero que não seja necessária a técnica que proponho adiante. Novamente, tomaremos como exemplo a ROM do Final Fantasy III (USA). Tenho que dizer que os ponteiros são geralmente de dois bytes, mas existem de três e até de quatro (o Suikoden de PSX e o Phantasy Star IV de Megadrive utilizam ponteiros de quatro bytes).

Primeiramente devemos carregar a ROM no Translhextion e ir ao primeiro caractere do primeiro diálogo que existe na ROM (não tem que ser necessariamente o primeiro que aparece no jogo), que neste caso é "VICKS: There's the town...". Se nos fixarmos na posição do primeiro caractere (o 'V') veremos que está situado na posição 0D0200 em hexadecimal. Agora temos que aplicar a seguinte fórmula:

Ponteiro do diálogo X=(Posição do primeiro caractere do diálogo)-(header da ROM se existir)->(deixar os últimos quatros caracteres)->(agrupados em pares)->(invertidos).

Vejamos por partes par vê-lo de uma maneira mais clara:

 O header de uma ROM (se possuir) pode variar segundo o seu formato, ou seja, uma ROM de SNES possui 512 bytes, igual a uma ROM de Megadrive, mas uma ROM de NES no formato .nes possui um tamanho de 16 bytes. Sigamos como o de antes. A posição do primeiro caractere do diálogo é, como verificamos antes 0D0200.

- A 0D0200 ficamos com 200 neste caso porque 512 (o número de bytes do header de uma ROM de SNES) é 200 em hexadecimal. Desta maneira, obteremos 0D0000.
- 3. Deixamos os últimos quatro dígitos, pois de 0D0000 pegamos 0000.
- 4. Agrupamos os últimos quatro dígitos em pares, ou seja, 0000 passa a 00 e 00.
- 5. Por último, invertemos os dígitos; ao ser os dois códigos hexadecimais iguais não importa, mas se tivéssemos 01 45 esta seqüência numérica passaria a 45 01. É preciso fazer esse passo porque o SNES e outros consoles lêem os ponteiros ao contrario (para explicar-lo de uma maneira bem simples).
- 6. Agora o que é preciso fazer, é ir até o início da OM para procurar esses valores. Antes de vaze-lo, repetiremos o processo com o diálogo seguinte, porque 00 00 é algo muito comum em qualquer ROM. Na realidade, quase todos os jogos possuem esse valores como primeiro ponteiro. Volte a repetir o processo com o local do diálogo seguinte, nos daremos conta de que 69 00 é o ponteiro do segundo diálogo.

Agora vamos ao Translhextion, carregamos a ROM e vamos a *Find* no meu *Search* com a opção *Hex* marcada. O que devemos fazer é procurar 00005900, mas então veremos uma das falhas que o Translhextion possui. Este não procura corretamente quando é acrescentado vários 00 no início da cadeia de busca. Como resultado disto deveremos usar um editor hexadecimal distinto, como por exemplo o WindHex32. Vamos em *Hex Search* no menu *Seach* e em seguida estaremos no local onde é mostrada na figura:

🗟 WindHex for Windows - "Final Fantasy 3 (V1.0) (U).smc"
File Edit Tools Search Option Help
▲ 000CE800 26060000000005900A000FF004E017401 &
000CE810 C4011C027302B402EB021B033D037103
000CE820 9C03B903F80314043004490450045E040.I.P.^.
000CE830 9504DA0420055F059505CE05E3051606
000CE840 26065D066D06A306AA06B606FB060C07 &].m
000CE850 9E07DC0726085B08A608E7081F09E209
000CE860 FE09300A6A0A7B0A9D0AA60AC80ACF0A 0.j. {
000CE870070B7B0B820B9E0BB00BC50BE90B0B0C . {
000CE880 1A0CBA0CD70CE90CF70C160D510D650D
000CE890 800D900DFB0D260E6E0E8D0EA50EC40E
000CEBA0 DB0E160F270F330F5B0F9B0FCC0F2E10'.3.[
000CE8B0 4F109110BB10F0101011201150115511 0
000CEBC0 5F1164118B11BC11E211F3116A128212 .dj.
000CE8D0 AA12D21229133A13471356137B13D313).:.G.V.{
000CE8E0 45148B14DD1423154A15641583159415 E # .J.d
000CE8F0 BB1500161E163D164B16A31616178717
000CE900 9417FF172B1854186318B618C0180C19+.T.c.
000CE910 23199B19D719021A1D1A571A751A821A #W.u
▼000CE920 B01ABE1A191B201B341B511B651B771B
Dec: 000 Hex: 00 Bin: 00000000 Offset: 000CE804 [026%] Asc 'A'=41

CAPÍTULO 6: Ponteiros

Tudo aponta que estamos diante da tabela de ponteiros. Em efeito, os resto dos códigos são muito parecidos, ou seja, que respeitam a ordem que se deve seguir nos ponteiros: o códigos da esquerda não vária de ponteiro a outro como faz o código da direita e aumenta em um quando o código da direita supera o valor de FF. Por outro lado, parece que o jogo possui uma rotina especial para detectar se o código da esquerda passa de FF, já que volta a 00, mas na realidade é como se carrega 01 00 00 (como se fosse um ponteiro de três bytes).

6.4. Outras formas de achar ponteiros

Se mediante o método explicado não se encontrar os ponteiros, pode haver várias razões. As seguintes são as diferentes possibilidades que eis encontrado ao longo de vários anos dedicados ao ROMHacking:

- Que os ponteiros não sejam de 2 bytes. Deixamos só os dois últimos códigos de cada posição. Isto se deve a que, normalmente, a maioria dos jogos usam ponteiros deste tipo. Sem embargo, outros jogos, como Final Fantasy V de SNES, utilizam ponteiros de 3 bytes. Outros, como o Suikoden de PSX, utilizam os ponteiros de 4 bytes. Isso só facilita a tarefa de utilizar textos de diferentes partes do arquivo com o que trabalharemos. Incluído há ponteiros de um só byte como os que se encontram no Phantasy Star II de Mega Drive. Este utiliza uma técnica baseada em somar ao ponteiro a direção que se utiliza no diálogo prévio ao que o ponteiro em questão aponta. Por ultimo, resta dizer que se nos encontrarmos com ponteiros de 3 bytes em vez de deixar só os últimos quatro dígitos, devemos deixar seis, assim como oito no caso de que se trate de ponteiro de 4 bytes.
- Que os ponteiros estejam espalhados pelo arquivo e não em uma tabela de ponteiro. No momento só os vi na ROM de Phantasy Star IV. Ao se ro ponteiro de 4 bytes, pode colocar o texto onde se queira na ROM. O único modo de os achar é procurar a direção de cada diálogo pela Rom e comprovar se realmente é um ponteiro o que encontrarmos. Por exemplo, se temos um diálogo que começa na direção 004588AB, devemos procurar a mesma seqüência na ROM (lembre-se que é possível que seja necessário subtrair os dois bytes do header da ROM se esta possuir uma) o só a metade (88AB).

- Que o jogo use uma maneira distinta de carregar o texto. o Final Fantasy VI de PSX possui uma rotina que carrega algumas vezes textos do menu do objetos usando ponteiros e outras vezes não, ou ao menos é o que parece. Essa é a razão pela que decidi não traduzi-lo devido aos problemas que acarreta modificar os textos. Em Phantasy Star IV acontece algo parecido, pois há um suposto ponteiro nos objetos que eu nuca consegui encontrar. Em Phantasy Star III utiliza também um sistema pouco comum de ponteiros que não descobri pois preciso trabalhar mais nele.
- Que o jogo tenha um formato estranho de ponteiros. Por exemplo, o Secret of Mana. No principio os ponteiros parecem seguir um padrão, mas logo percebemos que são diferentes e começam a não corresponder ao padrão. Além de, cada diálogo deste jogo esta rodeado de códigos. Magno do *Traducciones Magno* [http://www.nekein.com/magno/] deverá saber mais sobre isto, pois ele também traduziu o Secret of Mana.
- Que a tabela de ponteiros não começar em 00 00: como dissemos antes, a tabela de ponteiros geralmente começa por 00 00, mas outras vezes começa por outros valores diferentes (não digo arbitrariamente porque eles terão suas razões de ser). Teremos de ser espertos para encontra-la; se sabemos assembly (quero dizer, conhecer e interpretar as operações do processador do jogo com o que trabalhamos) e analisarmos o códigos do jogo, tarde ou cedo chegaremos na tabela dos ponteiros. Se não, que é o mais provável, o melhor é começar a examinar próximo do bloco de texto, já que a tabela pode estar perto deste, normalmente um pouco mais acima, ainda que às vezes esteja embaixo do bloco de texto.
- Que a tabela de ponteiros possua ponteiros desordenados. Pode ocorrer que encontre a tabela, mas que dá a impressão de não ser, porque todos os ponteiros estão desordenados (quero dizer, que não seguem a ordem do texto que há o arquivo). Comprove os ponteiros a ver si se correspondem com seus endereços. Ainda que nunca topei com nenhum jogo que utilize tais ponteiros, eu li que existem alguns nesse formato.

CAPÍTULO 6: Ponteiros

Que simplesmente na haja ponteiros. Pelo melhor que a principio aparenta ser, é improvável o paradoxo de sua verdadeira causa. Nunca é demais trocar o código de fim de mensagem para passar o texto e ver o que se mostra no dialogo "modificado", ou seja, o seguinte ao código de fim de mensagem. Isto acontece no Zelda de SNES e freqüentemente nos objetos (a parte de magias, habilidades, técnicas...) de jogos como o Phantasy Star II e IV. Algumas vezes nem sequer há o código de fim de mensagem , outras sim; a questão é que existe um número de caracteres fixos em todos os objetos, se ocupam ou não todos os caracteres disponíveis, como acontece no Final Fantasy III (USA) de SNES. Essa técnica é conhecida como *Constant Lenght Text*(Texto de Comprimento Constante). Um exemplo desta pode ser a seguinte, no que existeum máximo de oito caracteres por arma: *Sword###Spear###Bow####*.

6.5. Para saber mais

- AnusP's Advanced Hacking Tutorial (por AnusP): Outro documento muito interessante sobre ponteiros. Na realidade, eu aprendi com este.
- Comment trouver les pointeurs 24bits SNES (por Elfe Noire): O próprio titulo diz tudo. Nunca é demais saber um pouco sobre este tipo de ponteiros, pois você pode topar com ele.
- The Mad Hacker's Guide to Pointers (por The Mad Hacker): Interessante, completo e útil documento que explica detalhadamente como encontrar os ponteiros e quais tipos podem ser encontrados normalmente em uma ROM. Esta baseado em ROMs de NES.
- The Madhacker's Guide to NES Pointers: Appendix B (por Gil_Galad): Explicação mais técnica e complementar ao que é explicado no documento original do MadHacker.
- Using Pointers Effectively (por Ghideon Zhi): Ainda que não seja descrito nesta parte do manual a modificação de ponteiros, que será visto no próximo capítulo, pode ser recomendado ler-lo para conhecer a maneira de aumentar espaço na ROM usando ponteiros.

CAPÍTULO 7: SCRIPTS

7.1. Programas necessários

- *Vrecalc*: Um simples mas prático utilitário de MS-DOS que é capaz de economizar muito tempo e esforço recalculando para nós os ponteiros.
- Tabela e ROM de Final Fantasy IV traduzida por J2e: a tabela é realmente tediosa de completar, por isso a inclui junto com esse manual. A ROM será por sua conta conseguir.
- *PSPad*: Excelente e gratuito editor de textos para Windows. Ainda que não o menciono em qualquer parte deste capítulo, o recomendo para traduzir scripts.

7.2. Scripts

7.2.1. Definição

Um script - termo em inglês para se referir aos textos de um jogo - é um arquivo de texto em que se encontra parte (ou todo) o texto do jogo, para ser traduzido mais comodamente com qualquer editor de texto. Este arquivo de texto será criado depois de extrair o texto do próprio arquivo que o contém e voltará a inseri-lo mais tarde no mesmo para que a tradução deste modo no arquivo em questão. Sem problema, os usos mais comuns que se dá aos scripts (aparte da flexibilidade que oferecem), são os de trocar mais tarde os ponteiros para inserir os textos e ter um texto com mais qualidade e dispor de mais espaço que o texto original.

7.2.2. Vantagens e inconvenientes

Evidentemente, nem tudo são rosas. Vejamos o que nos oferecem os scripts. Por um lado temos comodidade flexibilidade na hora de traduzir o texto. Mas traduzir o texto sem jogar, pode levar a traduzir coisas indevidamente como se viu no primeiro capítulo deste manual, pois não são raras as vezes que é difícil distinguir o significado de uma palavra ou expressão em concreto ou saber se "you" se utiliza em singular ou plural. Em outras palavras, quem não conhece o contexto da mensagem, cuja importância ficou clara no primeiro capítulo.

CAPÍTULO 7: SCRIPTS

Também pode ser tedioso inserir os scripts e recalcular ponteiros. De toda maneira, a pesar de tudo isto, é muito melhor utilizar os scripts para ter uma melhor qualidade de texto (ao se recalcular ponteiros, o espaço original aumenta) e mais comunidade si o jogo for traduzido por mais de uma pessoa. Temos que lembrar que o tradutor deve ter todas as facilidades possíveis, pois traduzir algo cheio de códigos estranhos entre os textos e que por sua vez tenha que respeitar os espaço original e tira a vontade de qualquer um.

7.3. Extrair o texto

Vamos usar como exemplo a ROM do Final Fantasy IV traduzida por J2e. O primeiro que devemos fazer é executar o Translhextion. Carregamos a ROM e a tabela correspondente e procuramos o primeiro código que aparecer no jogo:



Não apareceremos no primeiro diálogo do bloco de texto desse local da ROM pois este diálogo não é o primeiro que aparece internamente na ROM. Por tanto, retrocederemos até chegar no primeiro de todos, "You inventory is full". É recomendável criar o *bookmark*(no menu *Bookmark* do Translhextion) para mais tarde voltar ao local mais rapidamente. Se você compreendeu sem problema o que são as tabelas de ponteiros, poderá comprovar claramente que os ponteiros desse bloco de texto estão justamente encima.

No momento sabemos onde começa o texto. Agora que temos que fazer é ir selecionando o texto mantendo pressionada a tecla Shift até encontrar o último texto de diálogo desse bloco de texto tal como se pode ver na imagem:

-		[C	Mra	duca	aione	s\Tr	radug	cier	nes\F	inal	Fant	asy l	VVII 4	fig]	- Tra	insthe	extion		
File	Selection	Offset	Inse	ert :	Seard	h B	Bookm	narks	Scr	ipt	Optic	ns	View	He	lp				
	Table Tool	160	DF FF	01 22	FF 2C	C4 DF	C9 FF	DF DE	E6 E6	A8 E3	C4 E7	E6 31	30 DD	2A 30	C1 E7	C1 FF	the* o .* The	verworld. crvstals	· 🕒
	Thingy View Japanese Shift-JIS	w Active	C9 FF FF DD	DF 31 20 20	FF DD C5 DD	DD 2F 29 2C	30 DF 2D DD	E6 E2 DE 2C	DF FF DD DD	DD 31 E2 33	2A 2C 31 01	E3 DF DF FF	01 E6 33 22	FF DF 01 2C	29 FF FF DF	DF 29 16 FF	have a en tak y * Rub ahahah	lready* b en there icante!* aha!* The	e b H
	EUCJIS 8F510 8F520 8F540 8F550 8F550 8F550 8F580 8F580 8F580 8F580 8F520 8F520 8F520 8F520 8F550 8F550 8F600 8F620 8F620 8F640 8F640 8F640 8F640 8F640 8F640 8F640 8F640	> FF 31 E2 33 2C FF C5 FF DF E7 31 2C FF 11 E6 DF DF E7 DF E7 11 2C FF 11 E6 DF DF E1 FF 11 E6 DF 2C 15 C4 E2 31 2C C4 E2 31 2C C4 E2 31 2C C4 E2 C1 C1	DDFF2017332D253F79F60102AEDFF30F720174200	E69 DFFF00 E67 2FF20 E28 F720 E230 2020 2020 2020 2020 2020 2020 20	C9FF3DD32FF222C1DE7173DFF6F2220DD71533DFF6FF2FF6FF2FF6FF2FF	DF15912F2F2F2DDFF911592303F2F8FF2201844 FF	E7FD FFD 2C1 2C1 DE66 E62 DFFD 2C5 DFFD 2C2 FFD 2C2 FFD DC2 SC5 FFD DC2 SC5 FFD DC2 SC5 FFD DC2 SC5 FFD DC2 SC5 FFD DC2 SC5 SC5 FFD SC5 SC5 SC5 SC5 SC5 SC5 SC5 SC5 SC5 SC5	F2FD2FF441FFFF4003120F3557C07121F	A8222FF97EFDD2231233DFFDF16FF4F6661FF	2D0120000000000000000000000000000000000	300 FF2DF2DD17F5200 DD17F2DD5F200 FF2DF200 FF00 FF00 FF00 FF00 FF00 F	30F1FF13A0FFFF730EF437775D00F2DFFFFFF7300FFFF75000000000000000000000000	FF2D2C2C2C2C2C2C2C2C2C2C2C2C2C2C2C2C2C2C	E7F2F64221328DD2CEFF442FE295214	31 29 20 20 20 20 20 20 20 20 20 20 20 20 20	2D3 C31 C24 FF0 2C4 FF0 2C4 S00 S01 S02 S02 S05 S05 S05 S05 S05 S05 S05 S05 S05 S05	Dwarve 11 be* the G n!* Ev h I am us!\'I this k suffe nal da Cierra 1 be* er Fab Cryst ready n.*Yan God!* orce w lled! he cas only j their* * If w ked no	s will st killed b iant Cann en in dea * victori hose who to wield* atana wil r in* ete rkness!'\ : Then th coming af ul's* Win al 's* Win al 's* Win Damcyan' al has* a been stol g: !Go Our main as all* k Those in tle* have ust begun training e're atta w	i yo t o Pers ert: s t l ert: s i t t c
000 000 1	8F6A0 8F6B0	FF FF	FF	FF	FF	FF	FF FF	FF	FF	FF FF	FF	FF	FF	FF	FF	FF FF	Último (es repe	caracter, lo dem tido	nás + +
Selec	ted: Offset	et 525824=0x80600 to 587402=0x8F68A (61579 byte(s)) ANSI / OVR / L Size: 1573376																	

Manual de tradução de ROMs: O fascinante mundo do ROMHacking

Nos situamos no último caractere (que deve ser um fim de mensagem), e clicamos em *Dump Script* no menu *Script*. O quadradinho de *Separated-Byte format* não os recomendo que marque se seu jogo utilizar muita DTE/MTE porque pode trazer problemas na hora de inserir os script. Além disso, deixará praticamente uma tradução mais cômoda, porque todos os DTE e MTE terão < e > ao redor deles. Já só resta clicar em OK; se tudo está acontecendo bem, terá um bonito arquivo de texto com o texto inteiro ou parte dele do jogo com o que está trabalhando.

7.4. Inserir textos

Uma vez terminado de traduzir os script, ou simplesmente queremos comprovar como está nossa tradução da ROM, temos que inseri-los novamente na arquivo do qual os extraímos. A tarefa no início é bem simples ainda que podemos encontrar problemas se não a fizermos da maneira certa. A única coisa que devemos fazer é clicar em *Replace Script* no menu *Script* do Translhextion, escolher os mesmos parâmetros que escolhemos para extrair e clicar em OK. Se não tiver nada errado com os scripts, depois de alguns segundos o script estará reinserido no arquivo desejado, ainda que às vezes por motivos desconhecidos o programa pode dar erro. Em tal caso, seremos nós os responsáveis de encontrar o que causou a falha no script modificado.

CAPÍTULO 7: SCRIPTS

7.5. Recalcular ponteiros

O nome deste tópico pode sugerir que deveria estar no capítulo dedicado aos ponteiros, creio que seja mais adequado explica-lo agora. Espero que o leitor compartilhe da mesma opinião uma vez que tenha escolhido pela tradução de algum jogo que tenha que recalcular ponteiros.

Bem, uma vez que tenhamos inseridos os scripts traduzidos, copiamos a ROM modificada para a pasta onde esta o Vrecalc. Como o programa é de MS-DOS e é mais seguro que tenhamos que escrever os mesmo parâmetros uma ou outra vez, será melhor criar um arquivo de texto com a extensão .bat (quero dizer, não o salve como texto) e escrever em hexadecimal o que se diz na documentação do programa do Vegetal Gibber (todas as explicações são dele, não minhas, ainda que adapteis algumas):

VRECALC(arquivo)(p.texto)(p.ponteiros)(f.ponteiros)(tam.ponteiros)(separador)[prim.ponteiro]

(arquivo) = Nome de arquivo da ROM.

(p.texto) = Posição inicial do texto.

(p.ponteiros) = Posição inicial da tabela de ponteiros.

(f.ponteiros) = Posição final da tabela de ponteiros.

(tam.ponteiros) = Tamanho em bytes de cada ponteiro (2-4).

(separador) = Valor hexadecimal do código de fim de mensagem.

[prim.ponteiro] = Valor do primeiro ponteiro (opcional). Se não se especificar se tomara como referência o valor original do primeiro ponteiro da tabela a ser recalculada.

Assim, em nosso arquivo .bat deveríamos escrever o seguinte:

VRECALC.EXE ff4.smc 80600 80200 805FF 2 00

Agora bastará executar o arquivo .bat dando um duplo clique sobre ele. Se foi feito tudo certo, terá todos os ponteiros recalculados sem problema algum. Se não, revise o que pode ter falhado. Temos que ter cuidado de que o texto inserido não utilize o mesmo código de fim de mensagem (por exemplo, 00) para outro propósito, como pode representar o nome de um personagem, por exemplo, 0002. Se for assim, o programa falhará devido que não distingue entre o nome e o código de fim de mensagem.

7.6. Para saber mais

Ainda que não são propriamente documentos sobre scripts, mas sim sobre ROM Hacking em geral acredito ser oportuno estes tutoriais:

- Guide de la Traduction (compilada por la T.R.A.F.): Neste tutorial em francês são reunidos os documentos mais famosos escritos em francês sobre o ROM Hacking. E o mais interessante de todos é o de ASM do Skeud.
- *Romhacking Profissional* (por xxcunix). Documento em português, um pouco confuso mas que abrange muitos dos temas de ROM Hacking.

CAPÍTULO 7: SCRIPTS

CAPÍTULO 8: A TRADUÇÃO DE JOGOS DE PSX

Decidi dedicar um capítulo especifico sobre a tradução de jogos de PSX porque seguindo o método "tradicional" nós encontraremos várias barreiras, que seja como e4ncontrar os textos em um CD ou bem como modificar os gráficos. Muita gente segue empenhada em trabalhar cobre a imagem ISO de um jogo de PSX (um arquivo que contém todos os dados do CD), ainda que eu sou partidário de que não há nada melhor que q trabalhar com arquivos pequenos ainda que tenha que ter mais cuidado de saber em que arquivo está os dados que modificamos.

8.1. Programas necessário e outros requerimentos

- ISOBuster: Excelente programa para manejar imagens de CD. Será necessário para criar a imagem ISO do jogo que vamos traduzir.
- *CDMage*: Também permite manipular imagens de CD ainda que com mais limitações que o ISOBuster. Porém, possui a opção de poder substituir arquivos dentro de uma ISO, o que nos permitirá inserir os arquivos que traduzimos na ISO novamente se algum corromper os dados.
- *TIM Viewer*: Com este programa seremos capazes de manipular sem problema aqueles gráficos que se encontra no formato TIM.
- *ePSXe*: Emulador de PSX com o qual podemos comprovar nossos avanços graças a possibilidade de carregar imagens ISO.
- O jogo para traduzir: Se tomará como exemplo o Suikoden de PSX em sua versão PAL.

8.2. Criação de uma imagem ISO

O primeiro que temos que fazer para trabalhar é criar a imagem ISO do jogo. Para isso, executamos o ISOBuster com o CD do Suikoden inserido em nosso leitor CD/DVD. O que temos que fazer pe dar um clique com o botão direito sobre o ícone em que colocou o CD e dar-lhe a *Extrair Imagem>Raw*(*.bin, *iso). E continuando damos o nome ao arquivo que vamos criar (por exemplo, *Suikoden.bin*). Depois de vários minutos teremos em nosso HD a imagem ISO do Suikoden (ainda que se denomine imagem ISO, ela está no formato .bin). Quando o processo de extração finalizar, ele nos perguntará o nome de um arquivo .cue; o mais recomendado é dar o mesmo nome que a imagem ISO.

🔯 IsoB						. 🗆 🖂
Archivo	Extraer CD <imagen></imagen>	Datos de usuario (*.tao, *.iso)				
E: [0,1] †	Crear archivo cuesheet Archivo MD5 Checksum	De RAW a usuario (*.tao, *.iso) RAW (*.bin, *.iso) t t t t t t				۲
	Extraer desde hasta	Nombre	LBA	Tamaño	Tamaño (Bytes)	Modificade
	Buscar Sistema(s) de archivos Find missing files and folders Reprodudr audio Vista sectores Realizar una comprobación de superficie Propiedades	DATA DEMO MAIN.EXE OPEN15MX.STR PSX.EXE ■ PSX.EXE ■ SYSTEM.CNF	LBA 18854 23 17976 231784 231434 18346 17975	2.00 KB 2.00 KB 740.00 KB 37.55 MB 580.00 KB 616.00 KB 0.07 KB	1 amano (bytes) 2.048 2.757.760 33.473.728 533.920 630.784 68	Modinead 07/02/19 07/02/19 07/02/19 25/08/19 07/02/19 07/02/19
CD : [00 -	251210] Longitud : 251211	Carpetas : 21	Objetos en la ca	rpeta seleccionada	a : 7	

CAPÍTULO 8: A tradução de jogos de PSX

8.3. Inserção de arquivos na ISO

Ainda que agora não podemos inserir nenhum arquivo modificado, simplesmente porque não os temos, acredito ser conveniente explicar neste momento a maneira de inserir arquivos modificados na ISO que criamos. Para isso devemos fazer uso da última beta do CDMage.

A tarefa a ser realizada é bem simples. Basta abrir o arquivo .cue que foi criado no processo de criação da imagem ISO, selecionar o arquiv que queremos substituir, dar um clique com o botão direito do mouse sobre ele e ir em *Import File*. Selecionamos o arquivo modificado de nosso HD e em alguns segundos as trocas nesse arquivo estarão também dentro da ISSO, agora modifica. Devemos recordar que o arquivo a ser inserido deve ter o mesmo tamanho do original, já que assim não perderão dados, o que pode deixar inutilizada a ISO.

log CDmage B5 - C:\Suikoden.cue							(- 🖂
File Edit Action Player Display View Help									
] 🗋 🖨 🖬 🖻 💣 🎯] 🖻 🏦 🗸 🖛	• R 6 C		Ð 💭 🖸	9 🎫 😆					
ISO9660 Image tree Session 1 filesystem 💌	Data track folde	rs and files				18		63	74 80
😰 Image [Suikoden.cue]	Name		Size	LBA	Туре	Date and time	Timezone	Flag	s
	🚞 DATA			18.654	Carpeta de archivos	07/02/1997 4:29:38	0:00	D	
E S DATA	🗀 DEMO			23	Carpeta de archivos	07/02/1997 4:47:50	0:00	D	
	MAIN.EXE		757.760	17.976	Aplicación	07/02/1997 12:56:22	0:00	F	
CENIO CENIO	🛛 🖬 OPEN15MX	.STR 39.	.473.728	231.784	Archivo STR	25/08/1995 3:35:12	0:00	F	
	PSX.EXE		593.920	231.494	Aplicación	07/02/1997 12:57:56	0:00	F	
	🖬 SLES_005.2	Large Ico	ns	346	Archivo 27	07/02/1997 12:57:04	0:00	F	
	SYSTEM.CI	Small Icor	ns	975	Velocidad de marcado	16/12/1996 5:54:56	0:00	F	
		List							
		 Details 							
		Extract F	iles						
		Compare	Files						
		Find File.							
		Import Fi	e						
		Browse F	ile						
		Show Erro	or Log						
		Show Fou	und Files Lo	g					
		Show Diff	fereces Log	.					
		Show Bro	wser						
7 Objects 41.456.260 bytes in 2 folders and 5 fi	les				Idle				//

8.4. Procura de arquivos com o texto a traduzir

A maioria das pessoas agora explicaria como procurar o texto dentro da imagem ISO que acabamos de criar, mas eu proponho outra solução. O que devemos fazer agora é localizar os arquivos que contém os dados que nos interessam, ou seja, gráficos e textos. O Suikoden é um bom exemplo, pois possui uma quantidade notável de arquivos. É inevitável começar a procurar arquivo por arquivo ainda que fazendo uso do sentido comum podemos economizar muito tempo, sobre tudo porque o nome de um arquivo normalmente têm muito em comum com o seu conteúdo. De todas as formas, não confie sempre no conteúdo de um arquivo já que é possível que o texto se encontre repetido em mais de um deles.

O mais normal é que se encontrem todos os arquivos com os textos em uma mesma pasta e que, além disso, em casos de RPGs, os textos dos combates, dos menus e das conversações se encontram o arquivo diferente aos dos diálogos do jogo. Se nos fixarmos na estrutura de arquivos do Suikoden, tudo aponta que o texto se encontra em alguma parte da pasta DATA. Além disso, podemos pensar de maneira lógica, e dado que há muitas pastas com o nome de [XX]_AREA.[Y] (onde XX é o número e Y uma letra), tudo parece indicar que o jogo carrega o texto de uma pasta diferente segundo o local em que nos encontramos jogando. Assim, a cidade de Greminster que é onde começamos, pertencerá a primeira, enquanto as outras cidades ou aldeias como dos anões correspondem a outras pastas.

Para procurar o texto usaremos o método explicado neste manual, ou seja, mediante o SearchR X. Basta carregar os arquivos diretamente do CD, ainda que seja preferível copia-los para o HD porque antes ou depois teremos que faze-lo. O que temos é descartar extensões de arquivos (ou simplesmente arquivos se prefere dizer assim) cuja função é nos deixar saber que não tem nada haver com o texto. Por tanto, devemos ignorar arquivos com extensão .str já que são arquivos de vídeos do PSX assim como a extensão .xa, que corresponde aos arquivos de som; Se nos aventurarmos na pasta 01_DATA.A veremos que existem extensões .va, .vb, .bin e .8. Como já dissemos que a primeira área deve conter os textos de Gregmunster vamos procurar "Gregminster" em todos os arquivos, não sem antes analisar as extensões que temos.

Ainda que muitos os desconheçam, os arquivos .va y .vb são arquivos de sons. Isso acontece porque há um programa que se encarrega de converter o som desses arquivos em formato .wav.Em caso de desconhecer, o único que haveria passado é que teríamos procurado por textos nestes arquivos e nos daríamos conta de que neles não se encontra nada.

CAPÍTULO 8: A tradução de jogos de PSX

O seguinte seria procurar nos arquivos .8, também sem resultado. Então só restaria os arquivos com extensão .bin que tem toda a pinte de ter o que procuramos. Se procuramos "Gregminster" no arquivo VA1.bin encontraremos o texto que procuramos. O mesmo acontece se fazermos o dito no arquivos VA2.bin, jpa que além disso os valores para "a" e "A" são os mesmo em ambos os arquivos. Em boa hora, encontramos a chave: o textos de Suikoden se encontra nos arquivos .bin de cada pasta como nome AREA. Já só falta criar uma tabela básica, abrir os arquivos em questão com o Translhextion e comprovar que estávamos certo.

S	E:\DATA	.01_A	RE	A.A'	\VA	I.BII	- [ا	Tra	nslh	exti	on									
File	Selection	Offse	et	Inse	rt	Searc	h E	Bookn	narks	Scr	ipt	Optic	ons	View	He	lp				
Tab	le Toolbar			27	5C	FD	00	00	00	30	31	FD	00	00	00	00	00	#\w##\\\I	G#\\\\\	•
	T11 - 121	A 10		00	00	00	00	00	00	00	00	00	00	84	EB 00	04	80	####\\\\\	///####	
I ≚.	Thingy Viel	N Activ	e	04	00	BC	FB	04	80	~~~	FB	0.0	80	00	00	00	00	****	· · · · · · · · · · · · · · · · · · ·	
	Japanese			ŏŏ.	ŏŏ	õõ	õõ	ŏŏ	ŏŏ	õõ	õõ	ŏŏ	ŏŏ	1Č	B2	04	80	1111111	\\\1###	
	Shift-JIS	= #		04	80	33	53	1C	1C	10	24	15	1C	1C	10	29	1F	####I#11	tell yo	
C	EUC-JIS	\square		23	1F	1D	15	24	18	19	1E	17	10	01	19	1E	24	u somethi	ng *int	
0.00				15	23	24	19	1E	17	56	10	2E	1F	10	29	1F	25	eresting	f Do you	
000	3EC50	10 1	.В	1E 22	11	27	10	27	18	11	24	10	01	22	25	1E	15	know what	at *rune	
000	3EC50	20 1	0	12	29	23	12	12	10	23	11	10	10	10	10	10	12	Crystals	s are# *	
0000	3EC20	22 1	5	14	19	12	10	15	10	20	뷺	27	15	22	56	00	00	redible r	ower#\\	
000	3EC90	01 3	Ď	20	19	13	29	ĩŏ	23	24	15	27	ÔŎ.	01	06	ŏŏ	53	*Spicy st	ew*Pablito	#
000	3ECA0	23 1	ō	14	19	11	22	29	ōō	24	ĒĊ	04	80	90	ĒĊ	04	80	s diarv\t	;#######	
000	3ECB0	9C E	C.	04	80	33	24	53	23	10	23	1F	10	12	15	11	25	####It#s	so beau	
000	3ECC0	24 1	.9	16	25	1C	10	18	15	22	15	56	10	01	31	22	15	tiful her	re# * <mark>Gre </mark>	
000	3ECD0	17 1	D	19	1E	23	24	15	22	10	27	11	23	10	26	19	22	gminster	was vir	
000	3ECE0	24 2	5	11	10	10	29	10	01	14	15	23	24	22	1F	29	15	tually *c	lestroye	
000	3ECFU	14 1	.U.	14	25	22	19	15	17	10	24	18	15	10	3D 1.6	15	26	d during	the Sev	
000	3ED00	15 I 25 I	2	12	43	15	22	19	10	41	56	10	10	12	15	24	10	entrear (ar or 5	
0000	3FD20	1 1 1	F	27	10	19	24	53	23	10	23	11	10	12	15	11	25	now it#s	so beau	
000	03ED30 24 19 16 25 1C 56 10 01 28 1C 1C 10 24 18 11 1E tiful# *All than																			
000	3ED40	1B 2	3	10	24	1F	10	24	18	15	10	2F	1D	20	15	22	1F	ks to the	e Empero	
000	3ED50	22 5	6	10	01	2C	1C	15	23	23	10	18	19	1D	57	10	2C	r# *Bless	s him# B	
000	3ED60	1C 1	5	23	23	10	18	19	1D	57	00	00	00	41	1F	27	56	less him#	t///Wow#	
000	3ED70	10 3	D.	1F	10	24	18	19	23	10	19	23	10	31	22	15	17	So this	is Greg	
000	003ED80 10 19 16 23 24 15 22 55 10 01 24 18 10 33 1D minster# *the 1m																			
000	3ED90	20 1	. Б П	22	13	븙	10	10	13	10	20	10	14	16	10	20	10	perial ca	apital#	
0000	SEDRO	11 2	ŭ.	10	16	22	11	10	10	10	29	12 01	18	11	10	15	10	*Sure is	uillere	
0000	3EDC0	24 1	F	27	1E	10	19	1E	10	24	18	15	10	12	1F	1F	ÎĔ	town in t	the boon	
000	3EDD0	19 1	5	23	56	Ô0	ôó	õõ	ôŏ	41	19	24	18	10	23	1F	10	ies#\\\\\	Jith so	
000	3EDE0	1D 1	1	1Ē	29	10	13	1F	25	1Ē	24	22	29	10	12	25	1D	many cour	ntry bum	
000	3EDF0	20 1	в	19	1E	23	10	01	27	11	1E	14	15	22	19	1E	17	pkins *wa	andering	
1		••••		**			~ *			~ ·			10				10		• •	• •
Selec	ted: Offset	25722	9=0	0x3E	CCD	to 25	7239	9=0x	3ECD	7 (11	byte	e(s))					AN	ISI / OVR / L	Size: 452536	

8.5. Procura e edição de gráficos

A localização e edição de gráfico em jogos de PSX é muito mais complicada que em jogos de outros consoles como o SNES e ou o Mega Drive. Nem dizer que em caso de dados comprimidos sua localização pode implicar em esforços monstruosos por parte do RomHacker. Os jogos de PSX podem ter gráficos em formato RAW, ou também em TIM, ainda que seja freqüente que o modo do Game Boy do Tile Layer Pro funcione em remakes como o do Final Fantasy VI. Adiante começaremos a estudar como trabalhar com estes formatos.

8.5.1. Gráficos em formato RAW

Pelo que podemos comprovar, estes gráficos se encontram em todos os jogos 2D como os dois primeiros Suikoden ou o Breath of Fire III. A parte difícil, ao igual que com o texto, é localizar os arquivos onde estão os gráficos. Um dado a destacar é que nos arquivos MAIN.EXE e SLES_XXX que podem estar no diretório principal do CD sempre há dados bastante importantes. Assim, nosso primeiro passo será comprovar se a fonte de Suikoden se encontra em um destes arquivos. Para isso, nos valeremos novamente do Tile Molester. Tenho que dizer que foi Vegetal Gibber quem me mostro o modo de visualizar uma fonte de PSX com este programa. Vamos provar primeiro com o arquivo SLES 005.27, que no qual deve estar ao fonte.

Assim, excutamos o Tile Molester como dissemos antes. Só falta saber qual codec temos que utilizar: o *4bpp linear, reverse-order*. Porém, lembrando que dependendo do jogo pode ser melhor optar por outro tipo de codec. Se dermos uma olhada rápida no arquivo e não encontrarmos nada, isto acontece porque devemos visualizar os gráficos no formato RAW em modo 2D, tal como foi feito na hora de modificar a fonte de Final Fantasy V de SNES. Por tanto, antes de dar uma segunda olhada, devemos clicar em *View > Mode > 2-Dimensional*. Deste modo, se avançarmos pela ROM chegará um momento em que poderemos distinguir a fonte de Suikoden sem nenhum problema. Uma vez modificada basta inserir novamente o arquivo SLES_005.27 na ISO da maneira que vimos antes para poder apreciar as trocas.



CAPÍTULO 8: A tradução de jogos de PSX

8.5.2. Gráficos em formato TIM

Para modificar gráficos em formato TIM teremos que utilizar a valiosa ferramenta TIM Viewer, que não só é capaz de visualizar imagens TIM e escanear arquivos normais em busca de imagens TIM, e ainda mais permite exporta-la para BMP e de BMP para TIM. Para ver como é fácil editar imagens em formato TIM, tomaremos como exemplo a versão de PSX do Final Fantasy IV (PAL).

Executamos o TIM Viewer e abrimos as imagens TIM com as que queremos trabalhar. É possível que se encontrem ocultas dentro de outros arquivos, porque a opção *Scan RAW File* mos será mais útil em muitas ocasiões, ainda que não é o caso.

Em Final Fantasy IV de PSX os TIMs que nos interessa estão na pasta CARDTIM. Vamos abrir no TIM Viewer o arquivo SAVEMOD.TIM, que é no que se encontra a fonte do jogo entre outras coisas. O primeiro que temos que fazer será converter a imagem TIM para o formato BMP, clicando em *Image > Convert to BMP*. Uma vez selecionamos a pasta destino, teremos em nosso HD a imagem que víamos no TIM Viewer mas em BMP. Agora editamos ainda que seja para destroçar literalmente a imagem a base de pixels negros por todos os lado com algum editor de imagens como o Paint. Uma vez que tenhamos salvo as alterações, vamos ao TIM Viewer, carregamos o BMP editado e convertemos para TIM outra vez:



Supondo que criamos a imagem ISO do Final Fantasy IV, inserimos o arquivo modificado com o CDMage e executamos o ePSXe. Se carregarmos a ISO modificada, poderemos ver que fizemos tudo com perfeição, já que podemos ver o seguinte:



8.6. Para saber mais

- Extracción y utilización de CLUTs (paletas PSX) en el editor gráfico Tile Molester (por Vegetal Gibber): Acredito que o nome já diz tudo. Se está interessado nas paletas de PSX, asseguro que este documento será de grande utilidade.
- La FAQ de Moogle sur le hack PSX (por Moogle). Documento que trata sobre a inserção de arquivos maiores que os originais em uma imagem ISO. No momento eu ainda não comprovei o método. Esta em francês.

CAPÍTULO 8: A tradução de jogos de PSX

CAPÍTULO 9: INTRODUÇÃO AO ASM

Este capítulo pretende apenas introduzir o leitor de maneira básica a linguagem Assembly (ASM) do processador do SNES, o 65c816, ainda que pode servir para trabalhar com outros processadores de características parecidas. Nem muito menos pretendo explicar tudo o que faz cada instrução (opcode) do processador, só explicarei alguma coisas que podem ser úteis na hora de traduzir um jogo. Já que muita informação disponível sobre o tema, se bem que é muita técnica.

9.1. Programas necessário e outros requerimentos

- Geiger's Snes9x Debugger: Emulador e depurador de SNES com capacidade de escrever em um arquivo log todas as instruções que se processam durante o tempo que especificarmos. Além disso, mostra valores de cada registro do processador de SNES durante o tempo todo.
- *Lunar Address*: Nos permitira converter uma posição do SNES para PC (a da ROM) e viceversa, ainda que veremos seu uso esta no manual.
- SNES Professional ASM Development Kit: Imprescindível para realizar pequenas modificações ASM dentro da ROM de SNES.
- O jogo com que se vai trabalhar. Neste caso tomarei como exemplo a Rom traduzida por J2e do Final Fantasy IV.

9.2. Instruções básicas

- LDA (LoaD Accumulator), LDX (LoaD X register) y LDY (LoaD Y register)

Digamos que o acumulador é uma variável que serve para guardar um valor que será utilizado posteriormente. Esse valor se especifica justamente depois da instrução, por exemplo, LDA #\$20 carregará o valor hexadecimal 20 (#\$ indica que o código esta em hexadecimal) em A. O mesmo ocorre nos registradores X e Y, ainda que estes servem normalmente como contadores para fazer bucles (logo veremos o que são).

- STA (STore Accumulator), STX (STore X register) y STY (STore Y register)

Como talvez possa ser adivinhado, estas instruções se encarregam de guardar os valores do registro especificado (A, X ou Y) na direção especificado). Assim, STX \$000A guardaria os valor de X na direção 000A.

– JMP (Jump)

Esta é uma instrução mais que útil já que pode meio dela podemos escrever nossas rotinas em outro local da ROM.

9.3. Procurar e modificar textos que não se encontram do modo normal

Com só estas três instruções já somos capazes de fazer algo muito útil: procurar textos curtos que normalmente não se encontra por métodos normais. Por exemplo, os típicos HP e MP dos RPGs que nunca se encontram, somente são encontrados com este modo, ao menos no jogos da Square como os Final Fantasy ou o Chrono Trigger, Vamos tomar como exemplo a ROM de Final Fantasy IV japonesa já com a patch da tradução do grupo J2e aplicada.

Primeiro vamos procurar os HP e MP do menu principal para modifica-los por PV e PM respectivamente. Para isso, abrimos o SNES Professional ASM Development Kit, e clicamos em *File > New Project* e abrimos a ROM do Final Fantasy IV. Se abrirá uma janela do projeto com o nome da ROM aberta. Se dermos um duplo clique no nome aparecerá dois novos elementos, *Code*, que mostra o código ASM do jogo, e *Data*, que mostra os conteúdo da ROM como um editor hexadecimal.A janela que nos interessa é a do Code.

Se dermos uma olhada na tabela do Final Fantasy IV veremos que 49=H, 4E=M e 51=P. Bom, é provável que o código que carrega HP e MP seja o seguinte:

LDA #\$49 ; H STA \$xxxx LDA #\$51 ; P STA \$xxxx LDA #\$4E ; M STA \$xxxx LDA #\$51 ; P STA \$xxxx

Claro, LDA poderia ser LDX ou LDY, ao igual que STA poderia ser STX ou STY. Bem, o que devemos fazer agora é procurar o LDA #\$49, para isso clicamos em Go > Find. Veremos que se encontra rapidamente o seguinte:

SNES Professional ASM Development Kit	
File Edit View Go Labels Dump Help	
	a 🚜 🗞 PC+ PC− 📮 📮 📮 A [®] X [®] ?
Code	
0189D8: A949 0189DA: 9D4000 0189DD: A951 0189DF: 9D4200 0189E2: 9D8200 0189E5: A94E	LDA #\$49 STA \$0040,x LDA #\$51 STA \$0042,x STA \$0082,x LDA #\$4E
Find Image: Constraint of the state of t	STA \$0080,x LDA #\$C7 STA \$004E,x STA \$008E,x LDY #\$0002 LDA (\$48),y JSR \$81BD STA \$0014,x XBA STA \$0016,x STA \$0000,x STA \$0000,x STA \$0000,x STA \$0000,x STA \$0000,x STA \$0

Ainda que não é exatamente a rotina que tínhamos escrito, tudo indica que estamos diante a rotina que procurávamos. De fato, está "optimizada" devido a que, em vez de carregar o caractere 'P' (51=P) duas vezes, ele carrega apenas uma vez mas o guarda em dois locais diferentes. Por sorte isso não nos impõe um impedimento, já que em PV (lembre-se que 57=V) e PM o 'P' também se repete, ainda que em locais diferentes. Segunda as direções que aparecem, podemos deduzir em posição são guardados os calores de HP e MP.

\$0040 \$0042 \$0080 \$0082

H P M P

Deste modo e utilizando a lógica devemos trocar os valores que são carregados e as direções onde são salvas no mesmo espaço que temos. Uma solução possível é a seguinte:

\$0040 \$0042 \$0080 \$0082 P V P M

Assim, vamos modificar o código da seguinte maneira (para editar código basta dar um clique sobre o menu e apertar a tecla INTRO quando for fazer a modificações desejadas):

CAPÍTULO 9: Introdução ao ASM

Code				_ 🗆 🔀
0189D8:	A957	LDA #\$57	17	
0189DA:	9D4200	STA \$0042,x	v	
0189DD:	A951	LDA #\$51	D	
0189DF:	9D4000	STA \$0040,x		
0189E2:	9D8000	STA \$0080,x		
0189E5:	A94E	LDA #\$4E	M	
0189E7:	9D8200	STA \$0082,x		
0189EA:	A9C7	LDA #\$C7		
0189EC:	9D4E00	STA \$004E,x		
0189EF:	9D8E00	STA \$008E,x		
0189F2:	A00200	LDY #\$0002		
0189F5:	B148	LDA (\$48),y		
0189F7:	20BD81	JSR \$81BD		
0189FA:	9D1400	STA \$0014,x		
0189FD:	EB	XBA		
0189FE:	9D1600	STA \$0016,x		-
0.0.0.0.0	-000			

Uma vez feitas as modificações pertinente, salvamos o arquivo. Já só resta comprovar se o que fizemos serviu para algo. Se executarmos o jogo no emulador e abrirmos o menu acabaremos com as dúvidas:



De fato, as modificações que foram feitas surtiram efeito. Devemos nos sentir orgulhosos, pois fizemos nossa primeira modificação ASM com alguns conhecimentos mais que básicos. Quem disse que deveria temer o ASM? O próximo que vamos modificar é o LEVEL que aparece no menu e que tão pouco se encontra por meios normais. Como veremos, traduzir LEVEL por NIVEL, pode estabelecer um problema ao não haver espaço suficiente na rotina original.

Sabendo desta vez que a rotina que procuramos esta "optimizada" da maneira que vimos antes, e que 4D=L, 46=E e 57=V, devemos esperar uma rotina como a seguinte:

LDA #\$4D ; L STA \$xxxx,x STA \$xxxx,x LDA #\$46 ; E STA \$xxxx,x STA \$xxxx,x LDA #\$57 ; V STA \$xxxx,x

Se procurarmos LDA #\$4D encontraremos a dita rotina. Fazemos o mesmo que antes, deduzimos em que posição são armazenados os valores de LEVEL:

\$000A \$000C \$000E \$0010 \$0012

L E V E L

Por tanto, os valores que devem ser carregados e as direções nas quais devem ser armazenados no caso de NIVEL são as seguintes:

\$000A \$000C \$000E \$0010 \$0012

N I V E L

Infelizmente, desta vez temos um problema, e é que o 'L" e o 'E' são carregados somente uma vez e são armazenados em dois locais diferentes, mas em NIVEL não existe nenhum caractere que se repita, porque não teríamos espaço suficiente para escrever nossa rotina. A solução para tal impedimento é procurar um local na ROM onde não exista nada, escrever ali nossa rotina e dirigir a rotina original para a nossa rotina.

Normalmente todas as ROM possuem áreas cheias de 00 ou FF, ou seja, que estão vazias. Para encontrar um área vazia, bastar procurar uma série de 00 ou FF na janela de Data. Não obstante, é também bastante freqüente encontrar um pouco de espaço inútil no final da ROM. Por exemplo, a posição 2FFD00 há um poço de espaço que nos será útil.

CAPÍTULO 9: Introdução ao ASM

O que devemos fazer é escrever uma rotina que carregue os valores "NIEL" (lembre que 4F=N e 4A=I) e os armazene em suas posições correspondente para que depois de finalizar a rotina vá diretamente a posição em que se carrega o caractere 'V' (0189D3), já que tanto em "LEVEL" como em "NIVEL" está na mesma posição e além disso é a última instrução que tem que ver com carregar "LEVEL".

Para isto, vamos para a posição 2FFD00 na janela de Códigoe escrevemos nossa rotina da seguinte forma:

Code					🛛 🔀
2FFD00:	A94F	LDA	#\$4F	N	
2FFD02:	9D0A00	STA	\$000A,x		
2FFD05:	A94A	LDA	#\$4A	1	
2FFD07:	9D0C00	STA	\$000C,x	1	
2FFD0A:	A946	LDA	#\$46	F	
2FFD0C:	9D1000	STA	\$0010,x		
2FFD0F:	A94D	LDA	#\$4D		
2FFD11:	9D1200	STA	\$0012,x	L	
2FFD14:	5CD38901	JMP	\$0189D3		
2FFD18:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD1C:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD20:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD24:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD28:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD2C:	FFFFFFFF	SBC	\$FFFFFF,x		
2FFD30:	FFFFFFFF 	SBC	\$FFFFFF, x		-

No final da rotina temos que colocar u JMP para a posição que carrega o caractere 'V' para que o jogo siga executando o código sem problemas. Já que só nos resta ir na rotina original e fazer um JMP para a posição da nossa nova rotina:

Code			🛛 🔀
0189C3:	5C00FD2F	JMP	\$2FFD00 🔺
018907:	009D	BRK	#\$9D
018909:	1200	ORA	(\$00)
0189CB:	A946	LDA	#\$46
0189CD:	9D0C00	STA	\$000C,x
0189D0:	9D1000	STA	\$0010,x
0189D3:	A957	LDA	#\$57
0189D5:	9D0E00	STA	\$000E,x
0189D8:	A957	LDA	#\$57
0189DA:	9D4200	STA	\$0042,x
0189DD:	A951	LDA	#\$51
0189DF:	9D4000	STA	\$0040,x
0189E2:	9D8000	STA	\$0080,x
0189E5:	A94E	LDA	#\$4E
0189E7:	9D8200	STA	\$0082,x
0189EA:	A9C7	LDA	#\$C7
Como vemos com a ilustração, a instrução JMP ocupa mais que a instrução LDA, parte do código que tem continuando é trocado. Isso em principio não deve importar, pois não deve haver nenhuma rotina que salte o código como tínhamos trocado. Depois destas modificações ASM, já só falta salvar as modificações e comprovar que tudo saiu bem:

Dark Knight Cecil NIVEL10 PV 200/200 PM 0/200	Item Magic Equip Status Form Change Config Save
	TIME 0:06
	o Gil

9.4. Instruções avançadas: bucles

Um bucle faz referência a um conjunto de instruções que se executem X vezes segundo uma ou várias condições. Por exemplo, um detonador pode contar de 10 a 0 para fazer explodir uma bomba. Cada segundo que passa o detonador comprova se já chegou ao número 0; se não é assim, não faz nada, mas se chegar em 0, executa a ação e deve explodir a bomba. Adiante explicarei os opcodes que veremos normalmente em um bucle, ainda que exista mais:

- INC (INCrement accumulator), INX (INcrement X register) e INY (INcrement X register)
 Se incrementa o valor da variável em questão (Acumulador, X e Y respectivamente) em um. É bastante útil para fazer um bucle se repita um número de vezes desejado.
- DEC (DECrease accumulator), DEX (DEcrease X register) e DEY (DEcrease X register)
 Diminui o valor da variável em questão (Acumulador, X e Y respectivamente) em um.
- ASL (Arithmetic Shift Left)

Sem entrar em nenhuma explicação técnica, digamos que o que ele faz é multiplicar por dois o calor de acumulador.

CAPÍTULO 9: Introdução ao ASM

- LSR (Logical Shift Right)

Este opcode faz o contrário do anterior, ou seja, divide entre dois o valor do acumulador.

- CMP (CoMPare accumulator), CPX (ComPare X register) e CPY (ComPare Y register)
 Compara o valor da variável em questão com o valor especificado. Por exemplo, CPX #\$4B
 compara X com o valor 4B. Esta comparação será muito importante para as instruções que varemos adiante.
- BCC (Branch if Carry Clear), também conhecido como BLT (Branch if Less Than)
 Se o valor da variável que se comprar é menor que o valor especificado na comparação saltará a posição especificada. Por exemplo:

CMP #\$20	; A = #\$16
BCC \$7800	; Como o valor de A é menor que #\$20, saltará a posição \$7800

BCS (Branch if Carry Set), também conhecido como BGE (Branch if Greater than or Equal)
 Se o valor da variável que se compara é maior ou igual ao valor especificado na comparação, será saltada a posição especificada. Por exemplo:

CMP #\$20	; A=#\$23
BCS \$7800	; Como o valor de A é maior que #\$20 será saltada a posição \$7800

- BEQ (Branch if Equal)

Se o valor da variável que se compara é **igual** ao valor especificado na comparação será saltada a posição especificada. Por exemplo:

CMP #\$20	; A= #\$24
BCS \$7800	; Como o valor de A não é igual a #\$20, não será saltada a posição
	; indicada e se executa a próxima posição.

- BNE (Branch if Not Equal)

Se o valor da variável que se compara **não é igual** ao valor especificado na comparação será saltada a posição especificada. Por exemplo:

CMP #\$20	; A= #\$08
BCS \$7800	; Como o valor de A não é igual a #\$20, será saltada a posição \$7800

- BRA (Branch Always)

Independentes de que se faça uma comparação ou não, e sim importar o resultado deste, sempre se saltará a posição especificada.

- JSR e JSL (Jump to SubRoutine)

Sua função é parecida com a de JMP. Salta a posição especificada e executa o código até que se encontre o opcode que faz regressar a posição seguinte ao JSR ou JSL correspondente. Bastante útil quando uma mesma rotina for utilizada várias vezes. A diferença entre JSR e JSL é que JSL é utilizado para especificar posições mais longínquas, pois ocupa mais bytes e demora mais tempo para ser executado.

- RTS y RTL (Return To Subroutine)

É o opcode que faz terminar uma rotina. A diferença entre RTS e RTL é a mesma que entre JSR e JSL e devem ser usados adequadamente.

9.5. Rastrear códigos: outras modificações ASM

Depois dessa enxurrada de novas instruções da impressão de que nos encontramos em ponto já bastante avançado, algo que não passa de uma ilusão para nós. Porém, e como já vimos anteriormente, com alguns conhecimentos básicos se pode conseguir fazer algo. Desta vez vamos ver como modificar algo "sério" como é o que proponho para continuar. O segredo sempre esta em usar o sentido comum, o ridículo e clássico método de prova e erro e, como não, o aprendido até o momento.

Resulta que no Final Fantasy IV, dependendo de se estamos em uma cidade ou no mapamúndi ou em uma masmorra, temos a possibilidade de salvar ou não a partida. Assim, "Save" aparecerá em cinza em uma masmorra enquanto que aparecerá em branco no mapa-múndi. O problema aparecerá quando nomearmos "Guardar" em lugar de "Save"(não tem nenhum problema pois já espaço suficiente; temos que procurar com a tabela dos diálogos, não como a do menu), pois só aparecem em cinza o quatro primeiros caracteres como podemos ver na imagem:



CAPÍTULO 9: Introdução ao ASM

Tudo parece indicar que existe uma rotina que, segundo se está na masmorra ou não, faz que apareça em cinza quatro caracteres. Com nosso conhecimento parece que embarcar na aventura de procurar e além disso modifica-la pode resultar em um tarefa impossível, mas logo comprovaremos que não é tão difícil, se bem que também não é fácil.

O que temos que fazer pe criar dois logs com todas as instruções executadas no momento de abrir o menu. O primeiro log será criado em um lugar onde se pode salvar e o segundo em um lugar onde seja impossível salvar. Para isso, devemos fazer uso do Geiger's Snes9x Debugger. A verdade é que prefiro usar a versão de LordTech Del Snes9x mas como é uma versão antiga é possível que não emule alguns jogos. Vem, o primeiro coisa que temos que fazer é desativar o som clicando em *Sound* > *Playback Rate* > *No Sound* e ter ativado a na janela de depuração *Trace Once, Squelch* e *Auto Usage Map*. Para executar o jogo, basta clicar em *Run* e trocar a janela do emulador.

Para o que temos que fazer adiante é possível que seja preciso um pouco de prática. Consiste em clicar no botão que despregue o menu principal enquanto se joga e nesse preciso instante ativar a opção CPU. Uma vez ativada, deve-se trocar a janela do emulador e desativar a opção CPU nada mais irá apareer no menu. Se fizermos tudo certo, teremos um log de uns 149kb mais ou menos no diretório em que está ROM que carregamos. Agora basta fazer está operação em um lugar onde se pode salvar a partida e em outro que não. Lembre-se que depois de criar um log é recomendável fechar o emulador e começar de novo para que o código já executado não volte a ser escrito. Se tudo sair bem, ao abrir o arquivos log (é conveniente que chame um de Branco.log e o outro de Cinza.log segundo sua correspondência) as primeiras instruções devem ser iguais.

Enquanto escrevo estas linhas me dei conta que não era necessário explicar nada referido aos bucles, salvo as instruções *JSR* e derivadas. Queria complicar menos o leitor o menos possível, mas o fiz, feito está. Bom, em qualquer caso, sigamos com o que queria explicar. Abrimos o arquivos Branco.log com o programa PSPad e comparamos com o Cinza.log mediante *Herramientas* > *Diferencias en texto* > *Comparar com archivo....* Se dermos um olhada rápida veremos que tudo coincide, menos algumas parte. Usando o senso comum nos damos conta de que a rotina que procuramos deve estar no Cinza.log porque por defeito todos os caracteres aparecem brancos. Assim, resta saber que código é o que corresponde a rotina em que estamos procuramos.

Tudo aponta que em \$21/F030 está a solução. Há uma rotina que simplesmente guarda o valor do acumulador em 5 posições e além disso, ela tem haver com o que se mostra na tela nesse momento, já que \$7E:xxxxsó indica isso (ou ao menos assim ocorre nos jogos com os que foi abordado). Por si for pouco, estas direções seguem a estrutura que vimos ao modificar LEVEL por NIVEL.

🛡 PSF	0ad - [(C:\Em	ulad	ores	\Tradu	iccior	nes\Utilia	lades\Snes9	K Trac	er (Win)\Blanco.l	log]														- 🔀
📝 Arc	hivo Pi	royecto	os Ec	litar	Buscar	Ver	Formato	Herramientas	Scripts	HTML	Configuracio	ones Vent	ana A	yuda												а×
00	n 🔁 -	G (00	۰ (🗁 👻	- 1		🔎 💅 👳	() ii	i 🛛	🖻 🗸 🖌	1		1	1	1	12.83	- 4	4	•		ML GRG Gag 🌑 🤻	G 🖸 🕯		8	
Blanco	.log ¹ B	lanco.	log ²																							
		1.7			1.75 I. I.		a					×.	L o	n 💷 8				- \ T-) COY T	(14:-)) (- 1		
U:NE	mulador	es\lrac	Juccio	nes\l	Jtilidade	s\5nes	SK Tracer	(Winj\Blanco.ld	g			U DEGA	2 Q		≌> €.			sin	auucu	aones	Vullidade	es onesox mac		s.log		
1905	\$01/0	BJZE	E8			INX				A:/D9D	X:C6/0	Y:FF91	20	1902	\$01/8	32E 1	28			INX			A:/D9L	X:C6/0	1:FF91	P A
1906	\$01/0	0321	00	F C		DD2	CF.C	1000101			X.C672	V.FF01	5	1903	\$01/0	220 1	-0 -0 -	c		TINA	CF C	1000101	A. 7000	V.C672	V.FF01	
1907	\$0171	0000	00	20		DRA	9110	[\$0310]		A. / D.5.D	A.00/2	1.1151	-	1904	\$01/0	550 1	50 E	0		DRA	910	[40310]	A. /DSL	A.00/2	1.0051	£.
1908	\$01/1	8332	F۵			PT.X				A: FF00	XICAFE	Y:FFD0	P	1905	\$01/8	332	FD			PT.X			A . FF00	X:COFF	Y:FFD0	р.
1010	\$01/1	8333	2B			PT.D				A: FF00	X:0000	Y:FFD0	p	1007	\$01/8	333	2B			PLD			A: FF00	X:0000	Y:FFD0	p.
1011	\$01/	8334	AB			PLB				A: FF00	X:0000	Y:FFD0	P	1908	\$01/8	334	AB			PLB			A:FF00	X:0000	Y:FFD0	P
1912	\$01/	8335	60			RTS				A:FF00	X:0000	Y:FFD0	P	1909	\$01/8	335 (60			RTS			A:FF00	X:0000	Y:FFD0	P:
1913														1910												
1914	\$01/1	8934	AD	02 :	1A	LDA	\$1A02	[\$7E:1A0	2] 3	A:FF00	X:0000	Y:FFD0	P	1911	\$01/8	934 1	AD 0	2 17	4	LDA	\$1A02	[\$7E:1A02]	A:FF00	X:0000	Y:FFD0	P:
1915	\$01/	8937	D0	0E		BNE	\$0E	[\$8947]		A:FF01	X:0000	Y:FFD0	P	1912	\$01/8	937 1	D0 0	E		BNE	\$0E	[\$8947]	A:FF00	X:0000	Y:FFD0	P:
1916	\$01/	8947	60			RTS				A:FF01	X:0000	Y:FFD0	P	1913	\$01/8	939 1	A9 2	4		LDA	#\$24		A:FF00	X:0000	Y:FFD0	P:
														1914	\$01/8	93B 3	22 3	0 F(21	JSL	\$21F03	0[\$21:F030]	A:FF24	X:0000	Y:FFD0	P:
														1915												
														1916	\$21/F	030 (8D 3	1 C2	7	STA	\$CA31	[\$7E:CA31]	A:FF24	X:0000	Y:FFD0	P :
														1917	\$21/F	033 (8D F	1 C9	Э	STA	\$C9F1	[\$7E:C9F1]	A:FF24	X:0000	Y:FFD0	P:
														1918	\$21/F	036 (8D 3	3 CI	1	STA	\$CA33	[\$7E:CA33]	A:FF24	X:0000	Y:FFD0	P:
														1919	\$21/F	039 (8D 3	5 CI	7	STA	\$CA35	[\$7E:CA35]	A:FF24	X:0000	Y:FFD0	P:
														1920	\$21/F	03C I	8D 3	7 CI	1	STA	\$CA37	[\$7E:CA37]	A:FF24	X:0000	Y:FFD0	P :
														1921	\$21/F	03F (6B			RTL			A:FF24	X:0000	Y:FFD0	P:
														1922												
													- 1	1923	\$01/8	93F (60			RTS			A:FF24	X:0000	Y:FFD0	P:
1917	000 (700							-	1924	000 (0)					705						-
1918	\$01/1	8752	20	11 3	94	JSR	\$9411	[\$01:941	1] .	A:FF01	x:0000	Y:FFD0	P	1925	\$01/8	/82 .	20 1	1 94	1	JSR	\$9411	[\$01:9411]	A:FF29	x:0000	Y:FFD0	P
1919	002 11	0755	20			TCD	0004E	1001-004			¥.0000	V. FEDO		1926	001 /0			E 01		TCD	CODAE	1001-004E1	3.0001	¥.0000	V. FEDO	
1920	\$0170	6/25	20	42 (02	USK	\$0241	[\$01:624	r] ,	A:0001	X:0000	1:000	P	1927	\$01/8	/15.	20 4	r 04	2	USR	\$024r	[\$01:0241]	A:0001	X:0000	1:1100	Ρ:
1921	\$01/1	8758	40	64 (04	.TMD	\$9464	1\$7E • 946	41		x.0000	V.FFDO	D	1928	\$01/8	758 .	40 6	4 04	1	.TMP	\$9464	1978-94641	7.4301	x.0000	V.FEDO	р.
1022	\$01/0	9464	20	RA /	81	JSR	\$8183	1\$01:818	21	4301	x.0000	Y.FFD0	p.	1020	\$01/9	464	20 8	A 81		JSR	\$8182	[\$01:8184]	A-4301	x.0000	Y.FFD0	p.
1923	φ01/.	5101	20	OA (0.510	VOIDA	[001.010	. I.		A.0000	1.1100	* 🗆	1021	\$01/J		20 0	n 0.		ODIC	VOIDA	[001.0104]	A. 1501	A.0000	1.1100	- 'œ
1925	\$01/	9467	A5	88		LDA	\$88	[\$00:018	81	A:4301	X:0000	Y:FFD0	P	1932	\$01/9	467 3	A5 8	8		LDA	\$88	[\$00:0188]	A:4301	X:0000	Y:FFD0	P:
1926	\$01/	9469	29	7F		AND	#\$7F		1	A:4380	X:0000	Y:FFD0	P	1933	\$01/9	469	297	F		AND	#\$7F		A:4380	X:0000	Y:FFD0	P:
1927	\$01/	946B	8F	00 2	21 00	STA	\$00210	0[\$00:210	0]	A:4300	X:0000	Y:FFD0	P	1934	\$01/9	46B (8F 0	0 21	L 00	STA	\$00210	0[\$00:2100]	A:4300	X:0000	Y:FFD0	P:
1928	\$01/	946F	48			PHA			- i	A:4300	X:0000	Y:FFD0	P 🚩	1935	\$01/9	46F 4	48			PHA			A:4300	x:0000	Y:FFD0	P : 🚩
<													>	<												>
	1:1/	2177			-		\$ 36 \$	24	TeX			Código de	página	ANSI (M	indows)											

Só que há algo que não se encaixa, e que para economizar tempo, explicarei agora. Resulta que como a primeira letra de "Save" é maiúsculas e portanto, mais alta que resto, deve colocar em cinza tanto na parte acima como na parte abaixo. Daí a explicação do STA \$C9F1 [\$7E:C9F1].

Bom, é hora de modificar esta rotina por uma que coloque em cinza todas as letras de "Guardar". Basta acrescentar três instruções STA como a que vimos antes, ainda que como o 'd' de "Guardar"é também mais alta que o resto das letras será necessário acrescentar um STA extra como o que é utilizado para o 'S' de "Salve". Abrimos o SNES Professional ASM Developmente Kit, carregamos a ROM do Final Fantasy IV e procuramos a instrução STA \$C9F1. Parece que estamos com sorte, pois temos espaço de sobra para inserir nossa rotina, que deve ficar como se mostra na imagem:

CAPÍTULO 9: Introdução ao ASM

Code			_ 🗆 🔀
21F02D:	206BFF	JSR \$FF6B	A
21F030:	8D31CA	STA \$CA31	
21F033:	8DF1C9	STA \$C9F1	
21F036:	8D33CA	STA \$CA33	
21F039:	8D35CA	STA \$CA35	
21F03C:	8D37CA	STA \$CA37	
21F03F:	8D39CA	STA \$CA39	
21F042:	8DF9C9	STA \$C9F9	
21F045:	8D3BCA	STA \$CA3B	
21F048:	8D3DCA	STA \$CA3D	
21F04B:	6B	RTL	
21F04C:	FFFFFFFF	SBC \$FFFFFF,x	
21F050:	FFFFFFFF	SBC \$FFFFFF,x	
21F054:	FFFFFFFF	SBC \$FFFFFF,x	
21F058:	FFFFFFFF	SBC \$FFFFFF,x	
21F05C:	FFFFFFF	SBC \$FFFFFF,x	•

Espero que não haja nenhuma confusão na hora de escrever novas direções, simplesmente estão baseadas no resto. Agora já só falta comprovar se tanto esforço valeu a pena. A prova final consiste em salvar as alterações e executar a ROM modificada no emulador. Prova superada?



9.6. Para saber mais

Apesar de que há muitíssima informação disponível na internet sobre ASM, prefiro não citar documentos avançados sobre tal, como aqueles que tratam da inserção de rotinas de fontes variáveis ou DTE. Só irei citar aqueles documentos que considero "essências" para ter uma forte base de ASM. Se quer uma recomendação, leia primeiro o log do canal #leetasm do IRC no que o mestre LordTech explica as bases do ASM. De fato, eu me iniciei no ASM graças a esse log.

 Assembly Programming for the Sega Megadrive (por The Sega Programming Network): Incompleto, mas muito útil para conhecer as instruções básicas do Motorola 68000, o processador do Megadrive.

- Aulas de Assembly para NES (Motorola 6502) (por Odin): Documento em português que explica de forma amena as base de ASM do NES.
- *Curso de ASM* (por Dark-N): Dos poucos documentos em espanhol que tratam o assunto ASM.
 Não se pressupõem nenhum conhecimento sobre o tema, o que facilitara um aprendizado de forma progressiva segundo se entra em detalhes técnicos.
- Log del canal #leetasm (por, entre otros, LordTech): A melhor forma de aprender sobre ASM do SNES. Ler este log será um prazer para os olhos que anseiam por conhecimento.
- Ze Skeud's guide sur l'asm (por Skeud): Ainda que pressupõe conhecimentos prévios de ASM, trata algumas rotinas tomadas de diferentes consoles que pode servir de apoio ao aprendizado de ASM.

Apêndice: Criação e aplicação de patchs

APÊNDICE: CRIAÇÃO E APLICAÇÃO DE PATCHS

Devido que a tarefa de criar e aplicar patches pode acontecer em qualquer momento, decidi proceder sua explicação ao final deste manual neste apêndice. Antes de nada, devo declarar que para ROMs são utilizadas patches no formato IPS e para as imagens ISO de PSX são utilizados o formato PPF. Assim pois, começo a explicação.

1.1. Programas necessários e outros requerimentos

1.1.1. Formato IPS

- Para criar o patch: Snestool, o arquivo original e o arquivo modificado.
- Para aplicar o patch: Snestool e o arquivo original.
- Para corrigir o checksun: WindHex 32 para SNES e GenRomSuite para Megadrive.

1.1.2. Formato PPF

- Para criar o patch: Criar PPF, a Isso original e a ISO modificada.
- Para aplicar o patch: PPF-O-MATIC e a ISO original.

1.2. O Checksun

O "Checksun" é um valor calculado a partir de todos os bytes de um arquivo. Por tanto, se modificar o código ao traduzir, o checksun não corresponderá ao original, assim, teremos que recalcula-lo. Não é uma explicação muito técnica mas creio que cumpre seu trabalho. Os emuladores não tem problema com isso, mas os "copiadores", ou seja, aqueles aparatos que nos permite jogar a ROM em um console real. Explicarei como corrigi-lo em jogos de SNES e de Megadrive; não deve ser difícil encontrar informação sobre outros consoles na Internet caso seja necessário.

1.2.1. Corrigir o checksun de jogos de SNES

Há vários programas que permitem faze-lo, mas eu gosto do WindHex32 por sua facilidade. A única coisa que temos que fazer é carregar a ROM e clicar em *Tools > Repair SNES Checksun* como mostra a imagem a seguir e depois salvar a imagem:

Apêndice: Criação e aplicação de patchs

🔁 WindH	ex for Windows - "ff4.smc'	,		
File Edit	Tools Search Option Help			
- 000000	Table Editor	Ctrl+T	00	
000000	Table Maker	Ctrl+M	10	
000000	Tile Editor	Ctrl+L	0 0 0 0	
000000	Snes To Hex	Ctrl+E	10	
000000	Create an IPS File	F5	10	
000000	Patch Current File	F6	0	
000000.	Increase / Decrease ROM Size	Ctrl+N	10 10	
000000	Repair SNES Checksum	Ctrl+P	10	
000000	Compare Files	Ctrl+C	10	
000000	String Length Calculator	F7	0	
- 0000012	0 0000000000000000000000000000000000000	000000000000000000000000000000000000000	00	
Dec: 192 Hex	c: C0 Bin: 11000000 Offset: 00000	000 [000%] /	Asc 'A'=41 Repair the checksum in a SNES file	

1.2.2. Corrigir o checksun de jogos de Megadrive

O mais seguro é que nuca faça falta, porque os emuladores Kega Fusion e Gens possuem a opção de corrigir ao carregar a ROM, ainda que francamente nunca consegui corrigir o checksun com nenhum emulador, apesar de aparecer uma mensagem que informa que foi corrigido.

O que temos que fazer é ir ao menu *Editor de Encabezados* e carregar a ROM. Ao menos que o checksun esteja correto, aparecerá uma mensagem em que perguntará se quer corrigi-lo. Não há que preocupar-se com a mensagem de advertência que aparece depois.

GenRomSuite	: - [Editor de encabezados - CATraducciones\Traducciones\Phantasy Star II\Copia (3) de pstar2 😑 💷 🗙
Archivo ?	
Utilidades	CSeleccionar ROM en formato binario (.bin)
<mark>wa</mark> ze ⊄=	C:\Traducciones\Traducciones\Phantasy Star II\Copia (3) de pstar2esp.bin
<mark>\$</mark> 8	CInformación del encabezado
Convertidor de	Nombre domestico: PHANTASY STAR 2
ROMS	Nombre comercial: PHANTASY STAR 2
ROM	Código del juego: GM 00005501-01
	1989.JUN
Editor de encabezados	Cor
	El checksum de esta ROM no es correcto, ¿Deseas arreglarlo?
* 👯	'stick MS
Lanzadera de	Sí No ivador
emuladores	lado
and A	Lector de CD Disquetera
IPS	Zonas de destino
Parcheador IPS	💿 Modelo viejo
de ROMs	Modelo nuevo
	Guardar Guardar como Ejecutar Cerrar
ROM C:\Traduccio	nes\Traducciones\Phantasy Star II\Copia (3) de pstar2esp.bin lista para su edición.

1.3. Patches IPS

1.3.1. Criar patches

Simplesmente clique em *Create IPS*, aperte a tecla ENTER, selecione a ROM original, aperte novamente ENTER e por último selecione a ROM modificada para apertar peça última vez ENTER.

1.3.2. Aplicar patches

Clique em *Use IPS*, aperte ENTER, selecione a patche, aperte ENTER e por último selecione a ROM original para apertar pela última vez o ENTER.

FFMQFRA .SMC MYSTIC'I.SMC SNESTOOL.EXE524800 524800 23978Snes-Tool Version 1.2 (c) The M.C.A./EliteSnes-Tool Version 1.2 (c) The M.C.A./EliteSplitIse IPSSend & lunIoinGreate IPSSloWrom fixIse IPSSloWrom fixIse IPSSloWrom fixIse IPSSloWrom fixIse IPSSloWrom fixIse IPSSloWrom fixIse IPS <t< th=""><th>C:\EMULAD~1\TRADUC~1</th><th>\UTILID~1\Snestool\Snestool.exe</th><th>_ 🗆 ×</th></t<>	C:\EMULAD~1\TRADUC~1	\UTILID~1\Snestool\Snestool.exe	_ 🗆 ×
Info:	FFMQFRA .SMC 524 MYSTIC~1.SMC 524 SNESTOOL.EXE 23	Snes-Tool Version 1.2 Comparison Split Use IPS Join Create IF SloWrom fix Cd 3 to f fix for Ntsc Fig to gd fix for Pal remove Int Gdd header Felete hea Path: C:\EMULAD^I\TRADUC^I\TRADUC^I\Traduc	S) The M.C.A./Elite S send & Fun PS slowrom Off Fig clinfiguration A 3 programinfo Cro Cuit Programinader Repair Header NUTILID ¹ SNESTOOL

1.4. Patches PPF

1.4.1. Criar patches

Abrimos o Crear PPF e selecionamos a Isso original, a modificada e especificamos o arquivos PPF a criar. Além disso, também podemos especificar uma descrição do patche, ainda que está limitada a 50 caracteres. Uma vez tudo isso pronto, basta clicar em *Crear* e esperar alguns minutos já que o processo leva tempo (com um Celeron 1,73 GHz demorou de 4 a 5 minutos).

😫 Crea PPF v.0.5 - YURSOFT	🗆 🔀
BIN original C:\Suikodenor.bin BIN parcheado C:\Suikoden.bin PPF C:\Suikoesp.ppf	Crear Salir Estado
Fileid.diz Descripcion Traducido por Pablito's	Creado por YURSOFT Agradecimientos a: Paradox, por crear el formato PPF

1.4.2. Aplicar Patches

Tão fácil como especificar a ISO original e o patche no programa PPF-O-MATIC e clicar em *Apply*. Se aparecer uma mensagem de erro selecione o patche, assegure-se de que tudo esta correto.

🎎 ::[: PPF-O-M	ATIC v3.0 :]::	
—Select File) MATIC-
	V E	RSION 3.0
TSO File	C:\Suikodenor.bin	8
Ratch	C:\suikoesp.ppf	8
🖌 Appl	y 🕕 About	🚫 Quit
📃 Apply Undo-		
Filename	: C:\suikoesp.ppf	<u>^</u>
Filelength	: 197710 Bytes	
Enc. Method	: 1 (PPF2.0)	
Description	:	
File TD dia	: 196626 Bytes	
	. Nope	
		v .