



Proyecto Grandia En Español



Índice

1-Historia De La Traducción

2-Instrucciones

3-Problemas Conocidos Y Incompatibilidades

4-Créditos

Historia De La Traducción

Todos os preguntaréis como empezó esta traducción, la verdad es que ni yo mismo lo sé ya que empezó hace muchos años de la mano de Rod Traducciones.

Uno de los encargados de la traducción por aquél entonces era pokechar, con el cual hablé previamente antes de comenzar a hackear el juego.

El caso es que este grupo de traducción intento hackear el juego con relativo éxito, ya que se quedaron a las puertas debido a su horrible complejidad en el sistema de punteros de la cual más adelante hablaré.

En ese grupo de traducción había una chica que colaboraba, ella es la actual coordinadora y jefa de este proyecto, con el nick de Sailor_Venus.

Como dije antes este grupo desistió del intento y pasaron los años, pero ella no se había olvidado del proyecto, con todo su empeño no desistió y un buen día posteó en chrono traducciones un post de ayuda.

Yo en un principio me mostré distante, ya que es públicamente sabido que no era un romhacker de traducciones del inglés al español, si no que traducía del japonés al inglés.

Pasó algo de tiempo y tome amistad con ella, amistad que me llevó a ver la maravillosa persona que es y como a mi no me costaba nada, pues decidí empezar a hackear este juego, que fue, es y será una auténtica pesadilla mundialmente conocida para los romhackers.

El hackeo de las fuentes fue relativamente fácil al principio, aunque me quedaron sin hackear un set el cual estaba comprimido. Poco después le empecé a meter mano al sistema y el engine del juego, el cual estaba horriblemente porteado (aparte de ser un engendro gracias a que en la versión americana le quitaron cosas).

No me costo mucho, así que empecé a colgar los scripts de las técnicas y demás en el foro, los cuales fueron merendados por los traductores en un santiamén.

Al ver lo rapidillo que iba decidí meterle mano a los scripts de la historia: los diálogos.

Pronto iba a saber el por qué todos los grupos que intentaron meterle mano al grandia desistieron, sus intenciones se esfumaron al ver la complejidad del sistema de punteros, el cual a veces llega a ser un tanto incomprensible y caótico.

No elegí precisamente el mejor archivo de dialogo para empezar, ya que tenía códigos de dialogo entre medias y dificultaba muchísimo la lectura de código.

Intentando re calcularlos me di cuenta de que no había manera, el audio se me desincronizaba, o apuntaba a donde no debían los punteros, total un desastre.

Es ahí cuando encontré a Butz De Dark Devils, y ahí estuvimos varios días mirando el código y intentando verle los pies y la cabeza al asunto, algo conseguimos pero no lo suficiente.

Desesperado ya, empecé a buscar información, alguna pista que me dijera que estaba pasando con los punteros de ese juego, y ahí es cuando encontré al genio Agemo, os sonará ya que ha desarrollado algunas herramientas para psx.

El realizó una traducción en su idioma (mandarín) y había desarrollado una herramienta la cual desmembraba el juego y encima comprendía byte por byte todo lo que hacia el juego. Lamentablemente estaba en su idioma, y ni siquiera era compatible con nuestros ordenadores, cosa que nos hizo hundirnos más en nuestra frustración.

No tardé en encontrar a alguien que también se encontraba en nuestra misma situación, era Mentz un italiano el cual llevaba 7 años traduciendo el grandia y se había quedado “colgado” con varios problemas que no podía arreglar. Decidió echarnos una mano después de que le arreglase un problemilla en el sistema de combate el cual le había dejado parado el proyecto por varios años.

El estuvo desarrollando una herramienta parecida a la de Agemo, solo que no funcionaba nada pero nada bien y la verdad es que era horriblemente imprecisa. Pero nada nos apartó de nuestro objetivo, y los dos empezamos a desarrollar más su herramienta, ya que lo básico si lo tenía, que era la comprensión de la rutina de punteros (que después tuvimos que depurar), la cual explicare ahora:

Los punteros eran llamados por un código alrededor del offset 0x60 y el cual escaneaba y los diferenciaba por sectores, mas o menos el código quedaba así:

```
($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($30)($00)($A0)($10)
($40) ($00)($B1) ($10)($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF) ($FF)
```

(\$30)(\$00)---> Es la ID o sector del consecuente puntero que se encuentra justo al lado.

Estos punteros apuntaban al dialogo, y lo seccionaban en sectores, los cuales a su vez tenían llamadas y códigos especiales de creaciones de distintos tipos de parámetros.

Dependiendo de que tipo de archivo fuese (de evento o dialogo normal) funcionaba de una manera u otra, dando a conocer otro problema que te encontrabas; la sincronización de eventos.

Si un sector por lo que fuese en bytes (y script) era mas pequeño de lo que originariamente era, sin importar que el puntero estuviese cambiado (apuntando a donde debía), funcionaba, pero se podía accionar en cualquier parte del archivo, ejemplo claro es hablar con un cartel y recibir una escena de evento.

Eso se podía arreglar introduciendo bytes de basura, en el mismo sector o en los sectores de antes, para que se sincronizara. Creo que esto es por culpa de que en alguna parte del archivo (no muy visible) hay un checksum que chequea los sectores bytes por bytes y devuelve un valor, que a su vez es guardado en la ram y va en ejecución, si no es el que debe la consola intenta auto arreglando, haciendo que el evento se ejecute justo en esa posición de la memoria, independientemente de donde este en el archivo.

Descubierto esto, y tras muchas pruebas Mentz y yo logramos automatizarlo todo y encontrar la manera de arreglarlo, pero no sabíamos que nos quedaba encontrarnos con un error bastante molesto; los saltos de sector al revés.

Y si, en algunos archivos compilados nos dimos cuenta al testear, o que los personajes empezaban a hablar cosas que no tenían que decir ellos, o que tocabas puntos de guardar y hablaban como personajes. Nos quedamos horripilados al ver en el código como los punteros apuntaban correctamente, pero siempre salía el bloque de la frase anterior y no la que estaba apuntando.

Esto era debido a una llamada especial, imposible de saber cuando se realizaba a simple vista, la cual hacia que cuando apuntase un puntero a una determinada parte, al valor en memoria del puntero le fuese restado la diferencia de bytes del bloque anterior, lo que hacia que apuntase a un sitio y saliese otro mensaje de la parte de arriba del script.

En consecuencia, otro error más en el cual pasamos innumerables horas testeando y tratando de verle los pies y la cabeza, hasta que nos dimos cuenta que era un set de bytes, en los que apuntaban nuestros punteros, los cuales debían tener otro valor (osea eran punteros que apuntaban a punteros dentro del texto, los cuales te enviaban un valor de donde se encontraba el sector del texto). Probablemente esta es la primera vez en mi vida que veo un sistema de redireccionamiento de punteros tan katano como ese, no me extraña que los romhackers le cogieran manía al juego.

Después de esto empecé a colgar más scripts de dialogo, los cuales fueron merendados por seres de otros planetas con nicks como monicapo, Auron, Butz, etc....

Una vez arreglado este error, (después de muchas horas) nos pusimos a testear nuestros resultados, mientras que yo buscaba una solución (la cual no la tuvo) de las imágenes de combate que estaban comprimidas.

Hablé con cyberwarriorx, un romhacker que había estado metiéndole mano a la versión de saturn, le envié archivitos pero al parecer la compresión de esta versión era mucho mas difícil que la de saturn. Así que poco se pudo hacer.

Un buen día gemini me pidió que le enseñase los archivos, y dijo que quería probar, desconozco lo que vio en el debugger, pero salio echando pestes del juego y dijo que estaba extremadamente complicado el realizar una herramienta para la compresión y descompresión ya que incluso le había costado horrores tracear en la memoria la rutina.

Lamentablemente la idea de modificar las imagenes comprimidas se tuvo que descartar y eso nos llevo al testeo de los cds, fase crucial donde las haya, ya que ayudo muchísimo a desarrollar la herramienta de Mentz que mencioné antes, como también a arreglar bugs por supuesto.

Este proyecto no hubiese pasado de la fase de testeo sin la ayuda de auron_antonio, un nuevo romhacker el cual dedico todo su tiempo y esfuerzo en jugarlo una y otra vez, llegando a haber del primer CD hasta 5 fases de testeo diferentes.

Está claro que sin su ayuda esto no hubiese llegado a buen puerto, o por lo menos no con los fallos tan minimizados como ahora los tiene.

La historia de esta traducción ha sido una historia de sacrificio y dedicación, como también de frustración a veces, pero todo esto en conjunto es el parche que os presentamos.

Esperamos que os guste tanto o mas como nos ha gustado a nosotros ver como nos superábamos cada día más y más.

Saludos.

Instrucciones

Esta traducción traduce todos os textos y algunas imagenes al castellano, exceptuando los vídeos por supuesto.

Tanto para el Cd1 como para el Cd2 necesitareis tener las imágenes de vuestros cds con el tamaño exacto de bytes, estos son los tamaños:

CD1

Primeramente,necesitareis un disco del Grandia (CD1) en pal, el código del serial es SLES_023.97.

La imagen tendrá que estar en formato .bin y tener obligatoriamente (693,162,624 bytes).

CD2

Primeramente,necesitareis un disco del Grandia (CD2) en pal, el código del serial es SLES_023.97.

La imagen tendrá que estar en formato .bin y tener obligatoriamente (678,079,248 bytes).

Preparación De Imagen

Ya sea que lo tengáis en una imagen, o en cd, podéis hacer el .bin.Si no tiene los dichos bytes el .bin, entonces el parche fallara.

Para crear un .bin adecuado. cogemos un programa llamado isobuster (más adelante daré el link) y metemos el cd en el lector.

Si tenemos la imagen en otro formato que no sea .bin, pues arrastramos la imagen hasta el isobuster.

Bien, donde pone track 1,clicamos con botón derecho y seleccionamos lo siguiente:

Extract Track 01----->Extract RAW data (2353 bytes/block) (*.bin *.iso)

El archivo lo guardamos con el nombre "Grandia.bin". Y esperamos.

Una vez concluido,ya tenemos nuestra imagen adecuada a la que aplicarle el parche.

Este proceso se deberá repetir con ambas imágenes.

Aplicación Del Parche

Los parches están en formato .ppf, es necesario aplicarlos con la herramienta ppf-Omatic.

El funcionamiento es muy básico ya que se selecciona el parche y la iso a parchear, después se acepta con el botoncito y listo.

Descarga De Aplicaciones

Isobuster: <http://www.isobuster.com/isobusterdownload.php>

PPF O-Matic: <http://chronoxtradus.hostoi.com/download.php?view.7>

Problemas Conocidos Y Incompatibilidades

Sí señor, acabamos de llegar a la sección que mas me gusta de este pdf, que es el de los petas del grandia, variados ya que es un port horrorosamente infernal del juego original de saturn, y como muchos ports tiene su porron de incompatibilidades.

PSP

Un eboot de este juego no se salva ni por los pelos, es posible experimentar cuelgues donde a simple vista en la consola o en el emu no los haya, ya que la psp maneja de otra manera distinta la ram, a los emuladores o la psx original.

Claro esta que se ha minimizado los bugs, pero pueden existir, recomiendo hacer algunos cambios de ID del eboot, ya que se ha reportado que es posible autoarreglar esos fallos **que son de la psp y no de la traducción en sí**, y así poder seguir jugando.

Emulador, Consola Y Errores Generalizados

El error más común en el emulador es el de basura en los diálogos, por ejemplo te encuentras al final de las lineas trozos de letras. Esto es causado por un error en el plugin gráfico, si esto sucede se debe cambiar de plugin dependiendo del emulador, **ya que con el Epsxe 1.6 usando el plugin gráfico Pete's OpenGL2 Driver 2.9 esto no sucede.**

Es posible que os encontréis durante el juego en un momento en el que pongáis el menú y veáis que el fondo del menú ha cambiado de color o simplemente esta de un color anormal. Esto se debe a una alta corrupción en la ram. Hay un error en las tiendas de limpiado de ram, justo en el menú. Hasta ahora no se ha arreglado, pero quizás en alguna otra revisión del parche se haga. Durante el testeo también nos encontramos con algún error de corrupción del puntito de los dialogos, esto también se arregla reseteando el emulador/consola.

Hemos intentado minimizar estos errores, pero todavía se que es posible que durante una partida os encontréis alguno. Si es así guardando la partida normalmente, reseteando el emulador y cargándola hará que vuelva a estar todo tal y como debería estar.

No es nada recomendable el uso de los savestates, ya que corrompen la ram y crean infinidad de bugs y otros problemas deribados de la misma corrupción.

Aparte de esto mencionado, el hardware original lo tolera perfectamente.

Créditos



Testeadores

auron_antonio

Flik

snake128

TearsGC

monicapo

Butz

bigsefirot

Especial mención a auron_antonio, ya que sin su empeño esta traducción nunca hubiese salido adelante.

Traducción dedicada a todos los fans y usuarios de Chrono Cross Traducciones, porque nuestro deseo es capaz de llevarte mas allá de cualquier límite:

<http://chronoxtradus.hostoi.com/news.php>

