



Romhacking: localización de videojuegos clásicos en un contexto de aficionados

Pablo Muñoz Sánchez
Universidad de Granada

Resum: L'objectiu d'aquest article és presentar a la comunitat acadèmica el concepte de *romhacking* i descriure la metodologia seguida pels *romhackers* per localitzar videojocs clàssics. També presenta una reflexió sobre la tasca que duen a terme els *romhackers*.

Paraules clau: Enginyeria inversa, localització, *romhacking*, traducció d'afficionats, videojocs

Resumen: Este artículo tiene como objetivo presentar a la comunidad académica el concepto de *romhacking* y describir la metodología seguida por los *romhackers* para localizar videojuegos clásicos. Asimismo, se presenta una reflexión acerca de la tarea que llevan a cabo los *romhackers*.

Palabras clave: Ingeniería inversa, localización, *romhacking*, traducción de aficionados, videojuego

Abstract: This paper aims at presenting to the academic community the concept of romhacking and the methodology followed by romhackers to localize classic video games. Furthermore, it presents a reflection on the work carried out by romhackers.

Keywords: Fan translation, localization, reverse engineering, romhacking, video games

1. Introducción

Gran parte de los videojuegos que aparecen hoy día en el mercado tiene su origen en Japón y Estados Unidos. Por desgracia, es posible que un videojuego no salga más allá de tales países a pesar de los grandes beneficios que pueden obtener en los mercados internacionales (Díaz Montón, 2006: 4). La situación empezó a cambiar notablemente a mediados de los noventa (Bernal Merino, 2006: 23-24), aunque no todos los videojuegos aparecidos en otros países pasan siempre por un proceso de localización. Además, la comunidad de aficionados aún lamenta la ausencia en el mercado europeo de algunos títulos que gozan de buena crítica en Japón y Estados Unidos, como por ejemplo la primera entrega de *Xenosaga* para la plataforma PlayStation 2.

Desde la aparición de los primeros videojuegos en la década de los sesenta ha habido muchos cambios tecnológicos y han surgido nuevos géneros de videojuegos, lo que no implica necesariamente que los videojuegos actuales ofrezcan un mayor entretenimiento que los de

antes. De hecho, tal y como comenta Gieske (2002: 9-10), parece haber resurgido un interés especial por los videojuegos clásicos debido, sobre todo, a la nostalgia que tienen las generaciones pasadas de jugadores como las que describe Herz (1997). Así, autores que escriben sobre la historia de los videojuegos como Burnham (2001) y Sellers (2001) defienden que estos productos de entretenimiento encontraron su época dorada en los años noventa y que hoy en día la calidad de un producto se basa más en los apartados técnicos que en la diversión que son capaces de ofrecer. En efecto, basta consultar los comentarios sobre videojuegos clásicos que se vertieron en la BBC (2001) para ver la realidad de esta controversia.

Entre los que defienden que cualquier tiempo pasado fue mejor se encuentran personas que siguen disfrutando de sus videojuegos favoritos de su infancia gracias al uso de los llamados «emuladores», que son programas que permiten ejecutar software en una plataforma diferente de la cual se programó originalmente dicho software. Gracias a Internet se puede conseguir fácilmente un archivo que contiene todos los datos de un videojuego —lo que se conoce como imagen ROM (del inglés *Read Only Memory*) o simplemente ROM— y poder disfrutar de él sin necesidad de poseer el producto original y un sistema que sea capaz de ejecutarlo. A pesar de que el fin ideal de los emuladores es preservar el patrimonio cultural de los videojuegos como apunta Esposito (2004), también ofrecen la ventaja de poder disfrutar de títulos que jamás vieron más luz que la del país en que se desarrollaron.

Por otro lado, la posibilidad de jugar a videojuegos que de otra manera no se podrían adquirir —obviando la práctica de la importación— no implica superar felizmente la barrera de no disponer de conocimientos del idioma en que se lanza un videojuego al mercado. Así pues, surge la necesidad de localizar videojuegos que ya llevan cierto tiempo en el mercado para poder disfrutarlos plenamente, lo que anima a una serie de aficionados a usar ciertas técnicas de ingeniería inversa para analizar el funcionamiento interno de un videojuego y así localizarlo de forma que otros puedan jugarlos sin necesidad de saber idiomas.

Partiendo de la experiencia del autor en el campo de esta forma de ingeniería inversa conocida como *romhacking*, este artículo pretende ofrecer una visión general de los rasgos que caracterizan esta manera de localizar videojuegos que llevan cierto tiempo en el mercado y que siguen teniendo aficionados. Así pues, se pretende dar un paso en más en la investigación sobre la traducción de videojuegos, cuya importancia se ha visto incrementada poco a poco en los últimos años con la publicación de varios trabajos dedicados a este campo de la traducción (Chandler, 2005; Vela Valido, 2005; Bernal Merino, 2006; Dietz, 2006; Heimburg, 2006; Mangiron y O'Hagan, 2006).

2. Concepto de *romhacking*

En su sentido más amplio, se denomina *romhacking* a todo el proceso de modificar una ROM para alterar principalmente sus gráficos, diálogos, niveles —escenarios de un videojuego con unas características y objetivos propios— y eventos —acontecimientos que se dan en un videojuego cuando se cumplen ciertas condiciones—. Esto se consigue, por un lado, con la ayuda de un editor hexadecimal, es decir, un programa para editar el contenido de un archivo de ordenador que no sea texto (por ejemplo, el archivo ejecutable de una aplicación), y por otro, mediante programas específicos para, entre otras cosas, buscar texto codificado o modificar gráficos. Dicho esto, no se puede afirmar que *romhacking* haga referencia meramente al proceso de localización de un videojuego, sino que cualquier tipo de modificación que se efectúe en una ROM, ya sea para traducir diálogos o para alterar gráficos, debe englobarse bajo el concepto de *romhacking*. Así, por ejemplo, existen editores de ROMs como Lunar Magic cuyo objetivo es modificar el contenido de los niveles del videojuego Super Mario World de la consola Super Nintendo.

El término *romhacking* encuentra su origen en el hecho de que ningún videojuego se diseña para que pueda modificarse tras ponerse en venta, a excepción de las actualizaciones disponibles para algunos juegos que añaden nuevas características o contenidos extra. Además, los kits de desarrollo de software (SDK por sus siglas en inglés) utilizados por las compañías no están disponibles para el público en general, lo que hace imposible editar el código. Por tanto, la ROM de un videojuego se debe modificar mediante ingeniería inversa por alguien ajeno a la programación del mismo —un *hacker*— para conseguir el efecto deseado. Hay que precisar que, en este contexto, *hacker* debe entenderse tal y como lo define el *Merriam-Webster's 11th Collegiate Dictionary* en su tercera acepción: «*An expert at programming and solving problems with a computer*». De hecho, el perfil de un *romhacker* comparte ciertas similitudes con los ingenieros de localización profesional, pues tal y como comenta Esselink (2002: 4):

[A] localization engineer doesn't need to be a programmer, a translator, or a designer. Most importantly, a localization engineer should be someone who is not afraid of technology, who loves to experiment and troubleshoot, and who is extremely flexible.

Este término es rechazado por algunos miembros de la comunidad del *romhacking*, ya que los *hackers* suelen asociarse erróneamente con el uso de determinadas técnicas ilegales para desproteger programas de sus claves y así evitar tener que comprarlos. No obstante, el término *romhacking* ha sido utilizado durante demasiado tiempo como para acuñar uno nuevo que desvincule a esta comunidad de aficionados de cualquier actividad ilegal.

Por otro lado, en inglés se utiliza a veces el término *fan translation* como sinónimo de *romhacking*, aunque el primero denomina a todo tipo de traducción realizada por aficionados. Sin embargo, se ha de rechazar tal comparación puesto que, mientras que *romhacking* engloba tanto el proceso como la obtención del producto —decodificación, traducción y recodificación del texto—, *fan translation* sólo evoca al producto final y no se dice nada de cómo se obtiene. No hay que olvidar que, debido a que lo que se altera es algo en principio no previsto para su modificación, toda persona involucrada en el proceso del *romhacking* debe considerarse como aficionado y no como profesional.

En cuanto a la grafía del término, no hay aparentemente consenso entre la comunidad de aficionados. Mientras que algunos prefieren escribir *romhacking*, otros utilizan *ROMhacking* o *ROM Hacking*. De hecho, ROMhacking.net, uno de los principales sitios de referencia para el *romhacking*, utiliza el término *ROMhacking* en la sección de *About this Site* de su sitio web mientras que en su foro utiliza *Romhacking*. La propuesta de utilizar *romhacking* se fundamenta en que, aunque ROM deba escribirse en mayúsculas cuando se usa de forma aislada, *romhacking* debe concebirse como sustantivo y como tal no pude escribirse con unas letras en mayúsculas y otras en minúsculas. Además, las personas que se dedican al *romhacking* reciben el nombre de *romhackers*, sin uso de mayúsculas. Otras lenguas como el español o el portugués intentan evitar el anglicismo y se habla de *traducción de ROMs* y *tradução de ROMs* respectivamente, aunque el término *romhacking* sigue estando muy presente.

3. El proceso de *romhacking*

A continuación se describen las etapas del proceso de *romhacking*, aunque dado el carácter de este artículo sólo se presenta lo esencial del proceso sin ahondar en una visión más práctica con ejemplos. Además, hay que tener en cuenta que hay ROMs que no se pueden traducir mediante la siguiente metodología por usar técnicas internas de compresión de datos. Aquellos que deseen más información sobre este tema pueden consultar el manual de *romhacking* de Muñoz Sánchez (2007) o el wiki de Romhack Hispano (2007), que detalla de una manera más amplia y práctica todas las tareas que se llevan a cabo para traducir una ROM y en el que se basa esta sección. También es preciso añadir que todos los programas descritos pueden descargarse de la sección de utilidades de las páginas ROMHacking.org y Fortaleza Romhack, así como aclarar que están disponibles sólo para el sistema operativo Windows, pues si bien

hay herramientas para Linux, las más usadas sólo están disponibles para el primero. En cualquier caso, no hay ningún programa que sea de pago.

Conviene aclarar que cuando se habla de «hexadecimal» se hace referencia a un código basado en el sistema hexadecimal de base 16 (de 0 a 9 y luego de la A a la F) y que un byte está compuesto por dos dígitos hexadecimales (de 00 a FF ambos inclusive). Por ejemplo, 8 se expresa en decimal como 08 y 11 como 0B.

Por último, hay que decir que todas estas etapas se llevan a cabo normalmente por un único *romhacker*, aunque es frecuente que en la fase de traducción participe otro tipo de personas con pocos o nulos conocimientos de *romhacking* pero sólidos conocimientos lingüísticos para acelerar el proceso.

3.1. Decodificación del texto

Lo normal es que el texto de una ROM esté codificado y no se presente en el mismo formato de un archivo de texto estándar como el usado por el Bloc de Notas de Windows. Para saber de qué manera está codificado el texto, se debe hacer uso de un programa como SearchR X o Search Relative. Gracias a estas utilidades se puede averiguar el valor hexadecimal de los caracteres codificados que han de especificarse en un archivo denominado «tabla» para que un editor hexadecimal diseñado especialmente para el *romhacking* permita visualizar correctamente el texto de una ROM. Esto se consigue buscando una frase que aparezca en el transcurso del videojuego.

Una tabla es un archivo de texto con extensión .tbl en el que en cada línea se especifican las equivalencias de los caracteres que se usan en los diálogos de un juego en su código hexadecimal, aunque este término también se utiliza para referirse al conjunto de equivalencias que hay en una tabla sin hacer referencia exclusivamente al archivo con extensión .tbl. Dado que la realización de una tabla resulta un poco tediosa, existen programas como TaBuLar o TAG que simplifican mucho tal labor.

3.2. Edición de gráficos

Es muy frecuente que un videojuego que nunca ha salido de Japón o Estados Unidos no permita utilizar acentos u otros caracteres como la cedilla (ç), pues no se utilizan en el idioma del país originario. Por tanto, si se quiere usar, por ejemplo, la eñe del español, hay que insertarla previamente en la fuente que utiliza la ROM. Los editores gráficos permiten realizar esta tarea, que consiste en cambiar unos gráficos por otros que no se vayan a utilizar en la traducción, tales como el apóstrofe ('). Todas las fuentes de un videojuego están compuestas por un conjunto de píxeles —unidad gráfica mínima de una imagen digital—, por lo que el videojuego se limitará a mostrar los gráficos almacenados en una parte determinada de la ROM. Por otro lado, editar gráficos de una ROM es diferente a editar una imagen con un programa de retoque fotográfico, pues la mayoría de los gráficos de una ROM aparecen desordenados, es decir, es necesario buscar la fuente del juego entre todos los datos que componen la ROM. Además, dependiendo del tipo de consola para la que se programe un videojuego, los gráficos se almacenan en formatos diferentes. Por suerte, los programas que existen para editar gráficos de ROMs, como por ejemplo Tile Layer Pro, permiten visualizar la gran mayoría de formatos existentes.

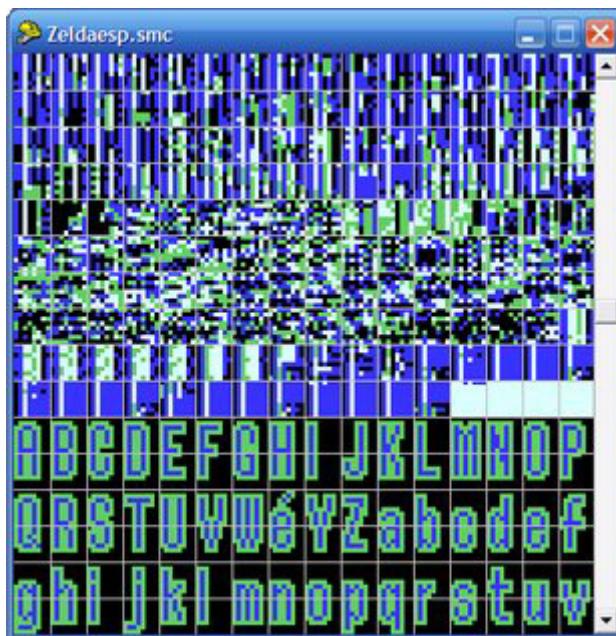


Figura 1. Fuente del videojuego *The Legend of Zelda: A Link to the Past* visualizada en Tile Layer Pro

3.3. Traducción del texto decodificado

3.3.1. Mediante un editor hexadecimal

Una vez creada la tabla de una ROM y alterados los gráficos en caso de ser necesario, ya es posible editar texto con la ayuda de un editor hexadecimal como WindHex o Translhextion. Cuando se edita el texto es preciso tener mucho cuidado de no sobreescribir códigos que forman parte de las instrucciones internas del juego y que pueden confundirse con texto traducible. De lo contrario, es posible que el juego no funcione correctamente.



Figura 2. Edición de texto en Translhextion del videojuego *Phantasy Star IV*

Si el texto es menos largo que el original, basta con usar espacios para reemplazar los caracteres no necesarios. No obstante, es posible modificar el código del juego para que no haya que ceñirse al espacio original. Para ello, los *romhackers* han de modificar los llamados «punteros», conjunto de códigos que indica desde dónde se debe empezar a leer internamente el texto de un diálogo en concreto. Éste deja de leer texto cuando encuentra un código de fin

de mensaje que se suele encontrar al final de un diálogo. Según el caso, un puntero puede estar formado por dos, tres o cuatro códigos, aunque también es posible que un puntero sea representado únicamente por un código. Desgraciadamente, y a pesar de que hay programas dedicados a la modificación de punteros de forma automática, muchas veces es difícil hallar la denominada «tabla de punteros», es decir, la zona de una ROM en la que se encuentran todos los punteros. Éstos normalmente están de forma ordenada, o sea, el primer puntero corresponde al primer diálogo según aparece el texto en la ROM, el segundo puntero al segundo diálogo, y así sucesivamente.

3.3.2. Mediante extracción y reinserción de archivos de texto

Es posible extraer el texto de una ROM a un archivo de texto conocido como *script* —anglicismo empleado para referirse al guión del juego— gracias al programa Translhextion y a otros más específicos para la extracción de texto como Romjuice. Para ello, generalmente hay que especificar dónde comienza y acaba el texto que se quiere extraer de la ROM y el nombre del archivo en que se guardará. De este modo, se puede traducir de una manera más cómoda y fácil así como distribuir los archivos de texto entre varios traductores. Una vez estén traducidos los *scripts*, bastará ir a la posición inicial del texto en la ROM e insertar el *script* traducido con un programa que encargue de recodificar el texto extraído como por ejemplo Translhextion o Atlas.

3.3.3. Mediante el uso de kits de traducción

Algunos *romhackers* explotan sus habilidades informáticas para programar kits de traducción que faciliten la labor del traductor tal y como ocurre en la localización profesional (Esselink, 2002: 4; Brown, 2006). Por un lado están los editores visuales de *scripts* que facilitan la labor del traductor.

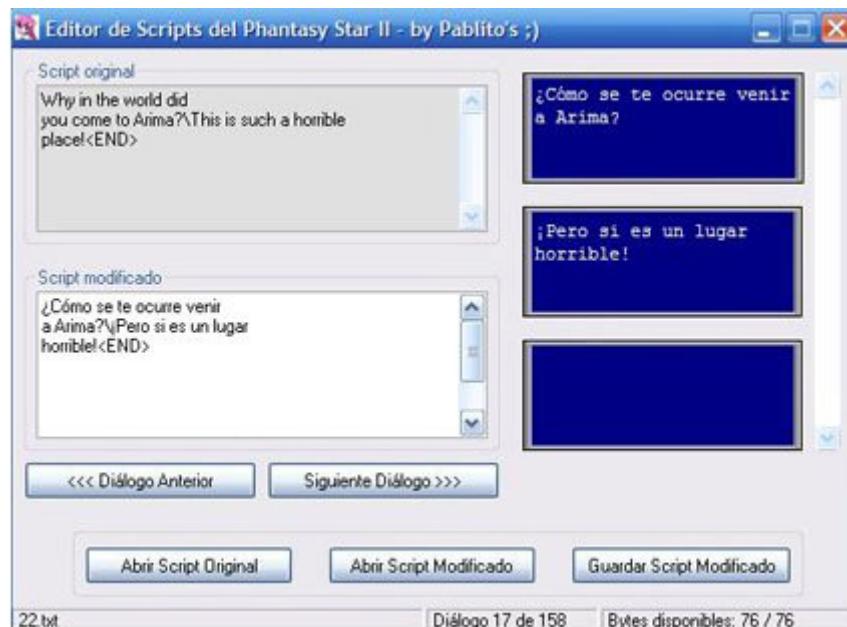


Figura 3. Editor de scripts para traducir los textos del videojuego *Phantasy Star II*

Por otro lado, existen «kits todo en uno» que permiten traducir una determinada ROM sin la necesidad de ningún otro programa.

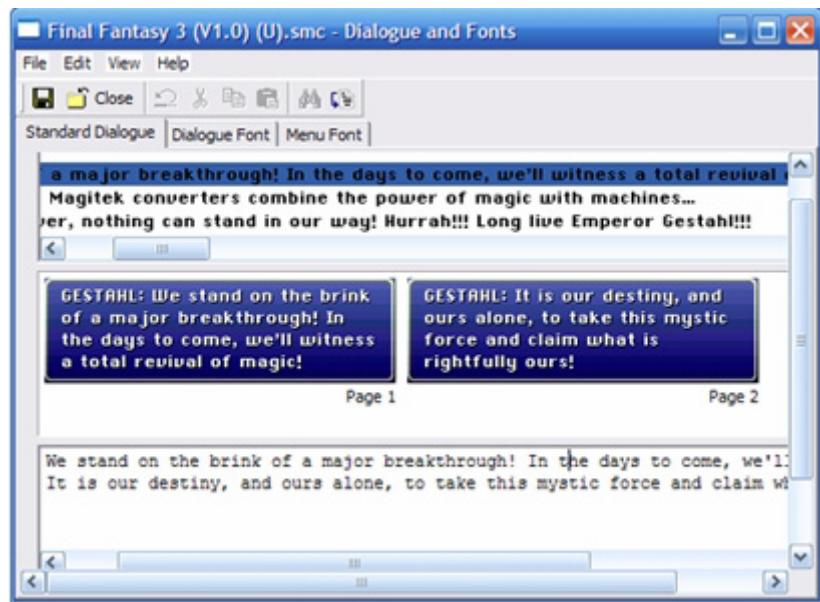


Figura 4. «Kit todo en uno» para traducir los textos del videojuego *Final Fantasy VI*

3.4. Creación y aplicación de un parche

Cuando la traducción esté terminada y se quiera distribuir entre el público por Internet, el paso final es crear un parche que contenga todos los cambios efectuados en una ROM. Un parche es un archivo que contiene los datos modificados y las direcciones internas de la ROM en que se deben escribir los mismos para que, al aplicarlo en la ROM original, se genere como resultado la ROM traducida tal y como el creador del parche tiene. Para ello se necesita un programa como SnesTool, cuya función principal reside en comparar los datos de la ROM modificada con los de la original para crear un parche. Para generar una ROM traducida, es necesario disponer del parche adecuado y de la ROM original y del programa SnesTool para aplicar el parche. Un parche contiene sólo los cambios entre una ROM modificada y una original, por lo que el tamaño del parche será menor que el de la ROM.

4. Conclusión

Traducir un videojuego es una decisión compleja y costosa que ni los desarrolladores ni los distribuidores se plantean salvo que haya ganancias aseguradas. La industria de la localización de videojuegos pasó por alto traducir algunos títulos en la pasada década, lo que, quizás, podría haber sido un error dada la cantidad de aficionados que aún siguen esperando a que sus juegos favoritos, olvidados por las nuevas generaciones de jugadores, sean traducidos por gente con los conocimientos y ánimos necesarios para ello. No obstante, bien es cierto que el mercado de los videojuegos ha crecido de forma exponencial en la última década y por tanto la situación que goza en la actualidad difiere notablemente con respecto a hace algunos años.

El *romhacking* se diferencia del proceso de localización profesional en que se trabaja sobre un producto ya terminado y no se facilita ningún tipo de ayuda por parte del equipo programador. Por ello, si bien hay mucha gente devota a determinados videojuegos con conocimientos lingüísticos y ganas de traducir, las trabas que supone editar una ROM mediante ingeniería inversa provocan un desánimo entre los traductores aficionados, ya que para participar en un proyecto de *romhacking* no basta saber traducir. Por otro lado, las ventajas que tiene modificar algunos aspectos de un videojuego como el tamaño de los menús o conocer de antemano el contexto de la mayoría de los diálogos conducen a una reflexión acerca del método de trabajo del traductor de videojuegos profesional. En este sentido, aunque los archivos que traducen los

traductores profesionales ofrecen una pequeña aclaración del contexto de la situación comunicativa, cabe plantearse si es mejor que un traductor haya jugado previamente a un videojuego antes de traducirlo como hacen los *romhackers* para ahorrar tiempo en el proceso de revisión y de control de calidad.

Newman (2004: 159) ya habla del talento con el que cuentan algunos aficionados para crear productos de entretenimiento digitales simplemente por hobby. Los *romhackers* hacen su trabajo por esta razón y por superar retos que van más allá de traducir un texto complejo. Aunque los *romhackers* no posean un título académico que certifique que son capaces de localizar un videojuego, demuestran habilidades que quizás podrían servir en el mundo profesional. Sólo el tiempo podrá decir si las empresas buscarán apoyo entre esta comunidad de aficionados como comenta Dans (2006). Por el momento, la labor del *romhacker* debe ser vista como lo que es: una práctica que se realiza por afición y que alarga e incluso revive la vida de muchos de los grandes clásicos de la historia de los videojuegos.

Agradecimientos

El autor desea agradecer los comentarios de Miguel Bernal Merino sobre un borrador de este artículo.

Referencias

- Barreras, P. (2003). "Localización de videojuegos". *Meristation*. Disponible en http://www.meristation.com/v3/des_articulo.php?id=4425&pic=GEN
- BBC News (2001). *Your Say on Classic Videogames*. Disponible en <http://news.bbc.co.uk/1/hi/sci/tech/1562572.stm>
- Bernal Merino, M. (2006). "On the Translation of Video Games". *The Journal of Specialised Translation* 6: 22-36. Disponible en http://www.jostrans.org/issue06/art_ernal.pdf
- Brown, K. (2006). "Creating an effective localization kit". *Multilingual Computing and Technology*, 17 (5).
- Burnham, V. (2001). *Supercade: A Visual History of the Videogame Age*. Cambridge, MA: The MIT Press.
- Chandler, H. M. (2005). *The Localization Handbook*. Massachusetts: Charles River Media.
- Dans, Enrique. (2006). "Cuando ruge la marabunta. Las empresas buscan el apoyo de los aficionados de todo el mundo". *PC Actual* 189: 32-36.
- Díaz Montón, D. (2006). "Entresijos de la localización". *Meristation*. Disponible en http://www.meristation.com/v3/des_articulo.php?id=cw443a3eb5db755&idp=&pic=HRD
- Dietz, F. (2006). "Issues in localizing computer games", en Dunne, K. J. (ed.) *Perspectives on Localization*. Amsterdam y Filadelfia: John Benjamins, 121-134.
- Esposito, N. (2004). "Émulation et conservation du patrimoine culturel lié aux jeux vidéo". *Proceedings of ICHIM 04 (Digital Culture and Heritage)*. Disponible en <http://www.utc.fr/~nesposit/publications/esposito2004emulation.pdf>
- Esselink, B. (2002). "Localization Engineering: The Dream Job?". *Tradumàtica* 1. Disponible en <http://www.fti.uab.es/tradumatica/revista/articles/besselink/besselink.PDF>

- Gieske, J. (2002). *Avoid Missing Ball for High Score*. Disponible en http://www.computerspielmuseum.de/_DOCS/jeremy_gieske.pdf
- Heimburg, E. (2006). "Localizing MMORPGs", en Dunne, K. J. (ed.) *Perspectives on Localization*. Amsterdam y Filadelfia: John Benjamins, 135-154.
- Herz, J.C. (1997). *Joystick Nation*. Londres: Abacus.
- Mangiron, C. y O'Hagan, M. (2006). "Game Localisation: Unleashing Imagination with 'Restricted' Translation". *The Journal of Specialised Translation* 6: 10-21. Disponible en http://www.jostrans.org/issue06/art_ohagan.pdf
- Merriam-Webster 11th Collegiate Dictionary*.
- Muñoz Sánchez, P. 2007. *Manual de traducción de videojuegos: el fascinante mundo del ROM Hacking*. Disponible en http://sayans.romhackhispano.org/documentos/manual_de_traducion_de_videojuegos.pdf
- Newman, J. (2004). *Videogames*. Nueva York y Londres: Routledge.
- Raymond, E. S. (ed.) (1996). *The New Hacker's Dictionary*. Tercera edición. Massachusetts: The MIT Press.
- Scholand, M. (2002). "Localización de videojuegos". *Tradumàtica* 1. <http://www.fti.uab.es/tradumatica/revista/articles/mscholand/mscholand.PDF>
- Sellers, J. (2001). *Arcade Fever: The Fans Guide to the Golden Age of Video Games*. Filadelfia: Running Press.
- Vela Valido, J. (2005). "La localización de videojuegos", en Reineke, D. (dir.) *Traducción y localización. Mercado, gestión y tecnologías*. Las Palmas: Anroart. 253-284.

Páginas sobre romhacking

- Charnego Translations. <http://www.isdifferent.com/~charnego/>
 DeJap. <http://www.dejap.com/>
 Fortaleza Romhack. <http://fortaleza.romhackhispano.org/>
 ROMhacking.org. <http://www.romhacking.net/>
 RPGOne Translations. <http://www.rpgone.net/>
 Sayans Traductions. <http://sayans.romhackhispano.org/>
 Traducciones Magno. <http://magno.romhackhispano.org/>
 Vegetal Translations. <http://vegetrans.jandar.net/>