

Superglús - Guia Técnica

Programa: Yokiyoki/Uto

Manuales : Uto/YokiYoki

(Basándose en el manual de PAW PC)

Basado en el software PAW:

Programa: T. J. Gilberts, Graeme Yeandle y P. Wade

Índice de contenido

Introducción	5
El Intérprete.....	6
Inicialización	6
Describir Localidad.....	6
Búsqueda en el Proceso 1	6
Búsqueda en el Proceso 2	6
Tomar una Orden	6
Búsqueda en la Tabla de Respuesta (Proceso 0)	7
Búsqueda en la tabla de conexiones.....	7
Ejecutar una tabla de procesos.....	9
Usando el Interprete	10
Los Contactos	11
Condiciones	11
De localidad	11
De localidad de objeto	12
De valor de los flags	12
De vocabulario	12
Aleatorias	12
Acciones	13
Posición de objetos	13
De objetos y flags	17
De flags	19
De flags relativos al jugador	20
De modos de pantalla y formato	20
De escribir flags	21
De control de pantalla.....	21
De listado de objetos	21
Acciones de Control	22
De salida de proceso.....	22
De grabación/carga de partida.....	23
De pausa.....	23
De control del parser (analizador de frases)	24
De tipo variado.....	24
De estilos, gráficos y sonidos.....	26
Descripción detallada del fichero fuente	27
La sección de control (CTL).....	27
La sección de Vocabulario (VOC).....	27
La sección de mensajes del sistema (STX)	27
La sección de mensajes (MTX)	27
La sección de textos de objeto (OTX).....	27
La sección de textos de localidad (LTX)	27
La sección de conexiones (CON)	27
La sección de definición de objetos(OBJ).....	28
Las tablas de procesos (PRO).....	28
La tabla de respuestas es el proceso 0	28
Proceso 1	28
Proceso 2.....	28
Proceso 3 (y siguientes)	28
Sintaxis del fichero fuente.....	28
CTL.....	28
VOC.....	28

STX, MTX, OTX & LTX	29
CON	30
OBJ	30
PRO	31
El compilador pgc.....	35
Uso del compilador	35
Uso del ensamblador glulxa	35
Código ensamblador generado	35
Errores del compilador	35
Errores léxico-sintácticos.....	36
Errores semánticos	36
Errores generales	38
Ensayos	39
El Parser	39
Objetos	40
Luz y Oscuridad.....	41
Pseudo-Inteligencias	41
Sumario	44
Los Mensajes del Sistema	44
Función de los flags de Superglús	44
Los CondActos	46
Condiciones:.....	46
Acciones: (la segunda columna es el tipo de acción)	47

Índice de ilustraciones

Ilustración 1- Bucle principal8	
Ilustración 2 - Ejecución de una tabla	9

Índice de tablas

Tabla 1 - abreviaturas	11
Tabla 2- condiciones de localidad	11
Tabla 3 - condiciones de localidad de objeto	12
Tabla 4 - condiciones de valor de los flags	12
Tabla 5- condiciones de vocabulario	12
Tabla 6- condiciones aleatorias	13
Tabla 7- acciones de posición de objetos	15
Tabla 8- acciones automáticas	17
Tabla 9- acciones de objetos y flags	18
Tabla 10- acciones de flags	19
Tabla 11- acciones de flags relativos al jugador	20
Tabla 12 - acciones de modo de pantalla y formato	20
Tabla 13- acciones de escribir flags	21
Tabla 14- acciones de control de pantalla	21
Tabla 15- acciones de listado de objetos	21
Tabla 16- acciones de control	22
Tabla 17- acciones de salida de proceso	22
Tabla 18- acciones de grabación/carga de partida	23
Tabla 19- acciones de pausa	23
Tabla 20- acciones del analizador de frases	24
Tabla 21- acciones varias	25
Tabla 22- acciones de estilos, gráficos y sonidos	26
Tabla 23- condiciones y acciones con sus parámetros	33
Tabla 24- mensajes ejemplo del PSI	42

Tabla 25- resumen de las condiciones	47
Tabla 26- resumen de las acciones	51

Introducción

Superglús puede dividirse en cuatro áreas principales:

1.El Fuente

El fuente (ficheros .SCE) es una colección de tablas, que contienen toda la información para definir una aventura.

2.El Compilador

El compilador comprueba el fichero fuente para encontrar errores y convertirlo en un fichero de salida Superglús (.ULA).

3.El Ensamblador

Una vez Superglús ha convertido el fichero de la base de datos en un fichero .ULA, la aplicación Glulxa (Glulx Assembler) convertirá dicho fichero en el definitivo fichero .ULX, un formato de juego de aventuras portable para la máquina virtual Glulx.

4.El Intérprete

Para poder ejecutar nuestra aventura, una vez esta está en formato ULX, necesitaremos un intérprete de máquina Glulx. Podéis encontrarlos en el if-archive (<ftp://ftp.gmd.de/if-archive/programming/glulx/interpreters/glulxe/>) para multiples plataformas (DOS, Windows, Linux, Amiga, mac, etc.). Esto quiere decir que aunque vosotros hagáis el juego en, por ejemplo, Windows, después se podrá ejecutar en todas esas plataformas.

El interprete realiza las tareas usuales de describir las localidades, atender a las entradas de órdenes de los jugadores, y responder a las mismas.

El Intérprete

Así funciona un intérprete de Superglús:

Inicialización

Los flags se ponen a cero excepto los siguientes: flag 37, el numero de objetos que el jugador puede llevar, se pone a 4, el flag 52, el número máximo de peso que el jugador puede llevar, se pone a 10. Los flags 46 & 47, el pronombre actual, se pone a 255 (sin pronombre). flag 1 se pone al numero de objetos que el jugador lleve (en las manos, no puestos). Nótese que borrar los flags tiene el efecto de que los juegos siempre comienzan en la localidad cero. Esto es porque el flag 38, la localidad actual del jugador, es ahora cero.

Describir Localidad

Si el flag 2 no es cero es decrementado. Si esta oscuro (el flag 0 no vale 0) y el flag 3 no es zero se decrementa en uno también. Si está oscuro el flag 0 no vale 0 y no hay ningún objeto fuente de luz, el flag 4 es decrementado.

Se borra la pantalla si el modo actual de pantalla (contenidos del flag 40) es par.

Si está oscuro, y no hay objetos con luz, se muestra el mensaje del sistema 0 (referido como SM0 - "Esta muy oscuro para ver nada"). En otro caso la descripción de la localidad se muestra (sin retorno de carro al final, el cursor queda justo detrás de la ultima letra o carácter de la misma).

Búsqueda en el Proceso 1

El diagrama de flujo 2 en la siguiente sección describe el proceso de ejecución de una tabla de proceso. El proceso 1 se usa principalmente para poner entradas que añadirán información extra a la descripción de la localidad actual. Por ejemplo, detalles sobre puertas abiertas, objetos presentes, etc.

En este momento entramos en el bucle principal, que controla cada fase de tiempo (es decir, cada vez que una frase se obtiene del jugador o que ocurre un timeout mientras se espera).

Búsqueda en el Proceso 2

Esto contiene el control principal de los turnos de Superglús en el juego. Se usa para implementar movimientos y acciones de PSIs, eventos no controlados por el jugador como por ejemplo puentes que se hunden, y demás.

Tomar una Orden

Los flags 5 a 8 son decrementados si no valen 0. Si esta oscuro y el flag 9 no es 0 es decrementado. Si esta oscuro, el flag 10 no es cero, y además no hay objetos que den luz se decrementa el flag 10.

Se llama al parser para extraer una frase y convertirla en una sentencia lógica. Si el buffer de input esta vacío, se le pide al jugador una nueva orden. En ese caso se mostrará el prompt, que será el mensaje del sistema cuyo numero esté en flag 42 (si el valor es cero se seleccionarán los mensajes del sistema 2, 3,4 y 5 con un ratio de posibilidades de 30:30:30:10 respectivamente).

Si la opción de timeout (exceso de tiempo de espera) está seleccionada, lo cual se hace poniendo en el flag 48 un valor superior a cero, entonces la rutina de esperar una orden del jugador puede tener limite de tiempo. En caso de que se pase del tiempo, el SM35 ("El tiempo pasa...") se muestra y se reprocesa el proceso 2.

Se extrae una frase convirtiéndola en una SL (sentencia lógica) convirtiendo las palabras presentes, que hay en el vocabulario, en su valor numerico y poniéndolas en los flags correspondientes (aquellos que guardan el numero del verbo, nombre, adjetivo, etc. de la SL actual).

Si no se encuentra ninguna frase válida el SM6 ("No entiendo") es mostrado, y se llama de nuevo al proceso 2.

Búsqueda en la Tabla de Respuesta (Proceso 0)

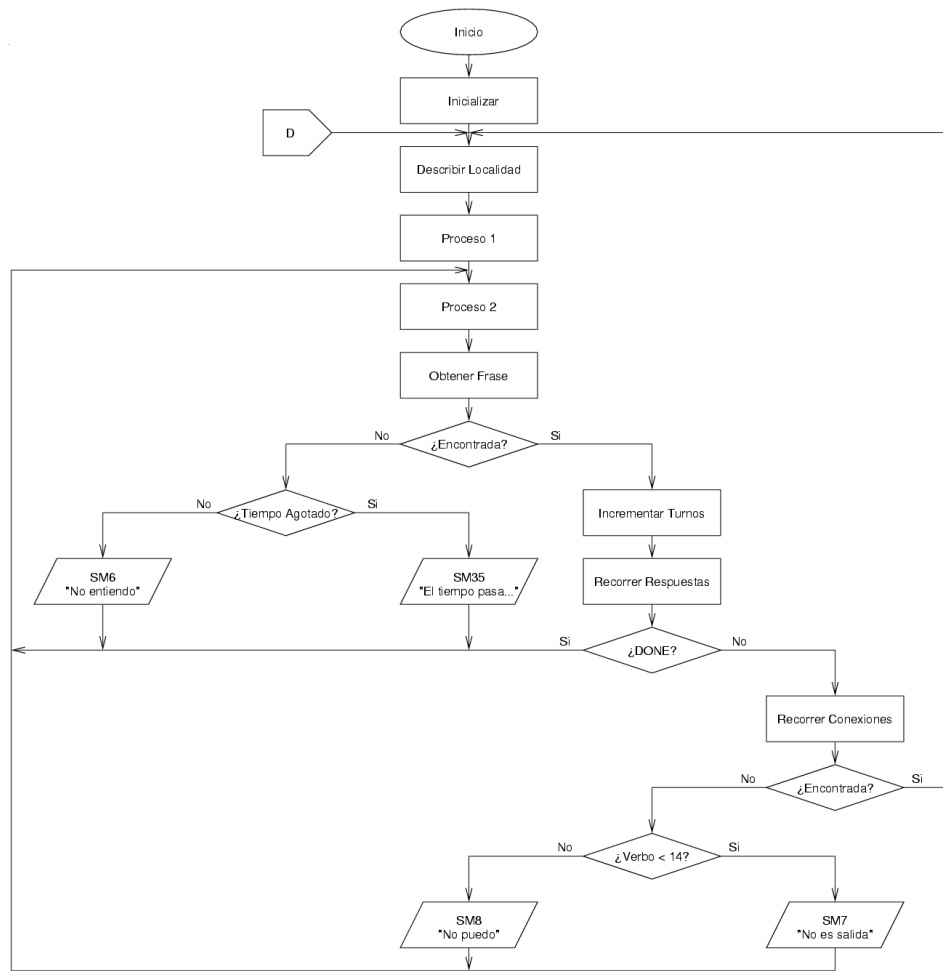
Los turnos (flags 31 y 32), que son el numero de frases válidas extraídas por el parser, se incrementa en uno. Pese a permitir los flags mantener valores altos se utiliza el viejo sistema de mantener dos valores de 8 bits de PAW en cada flag para formar un valor de 16 bits por compatibilidad [*].

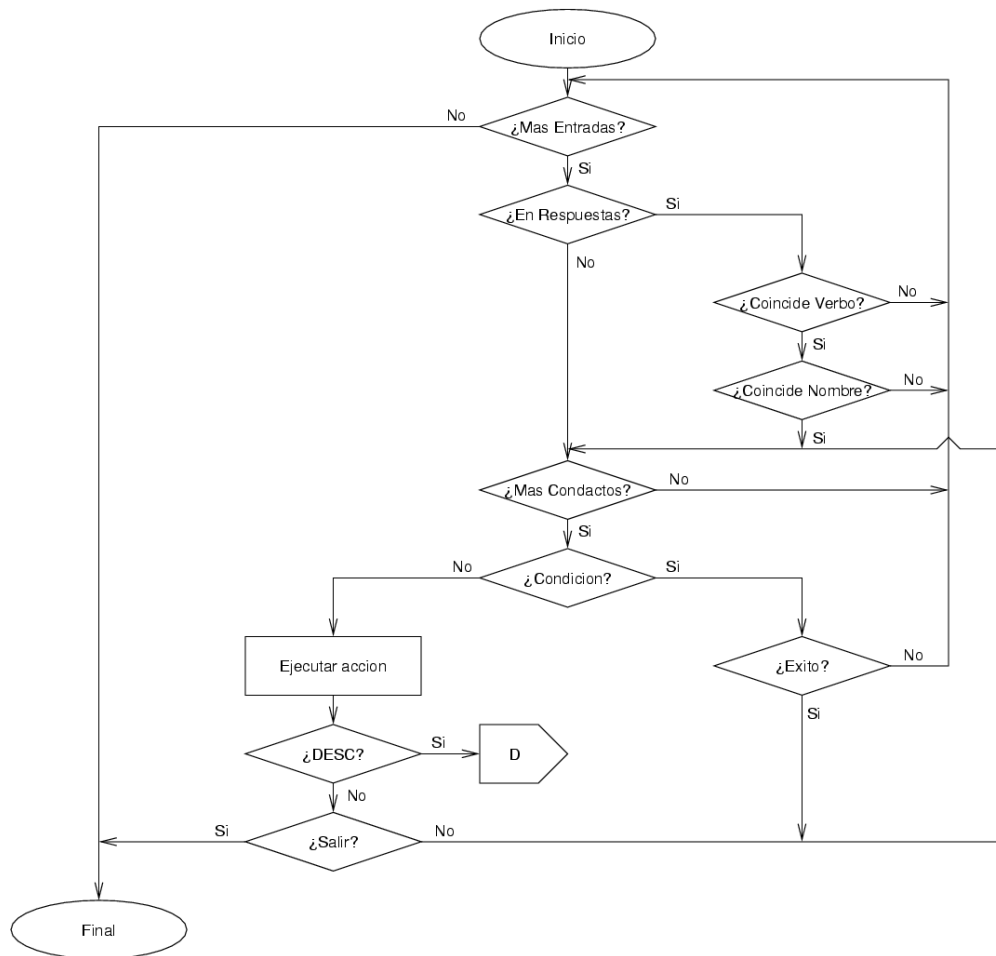
La tabla de respuestas (Proceso 0) es ejecutada entonces, buscando una entrada que concuerde con el verbo y nombre1 de la actual SL, usando las rutinas de proceso que se describen mas adelante.

Si se tiene éxito ejecutando una acción (por ejemplo si Superglús ejecuta cualquier acción distinta de NOTDONE) entonces vuelve y salta de nuevo a ejecutar el proceso 2.

Búsqueda en la tabla de conexiones

Se realiza una búsqueda en las conexiones entre localidad para la localidad actual, buscando una palabra que concuerde con el verbo de la actual SL. Si se encuentra una, entonces la localidad actual del jugador (flag 38) toma el valor del número asociado con la palabra. Entonces se ejecuta un retorno a la zona donde se describe la localidad actual. En caso contrario Superglús escribe el SM7 ("No se puede ir en esa dirección") si el valor numérico del verbo de la SL actual es menor o igual que 14, o el SM8 ("No puedes hacer eso"). En ambos casos se fuerza una ejecución adicional del proceso 2.





Ejecutar una tabla de procesos

La tabla de respuestas es el proceso 0, así que esta sección va también referida a ella.

Básicamente, Superglús mira cada entrada en la tabla hasta que esta se acaba. Suponiendo que haya alguna entrada, se asegurara de que el verbo y el nombre de la misma concuerden con la SL. El uso del carácter '_', tanto como verbo como como nombre, hará que case con cualquier palabra en la SL. Por tanto, entradas con el valor "_" (guión bajo – espacio – guión bajo) causaran la ejecución de dicha entrada independientemente de lo que haya en la SL.

Una vez en una entrada Superglús ejecutará cada contacto uno a uno; en caso de ser una condición, si se cumple, seguirá con el siguiente contacto. En caso contrario dejará de ejecutar la entrada actual y seguirá evaluando las siguientes entradas (si hay mas). Hay una excepción a esto, el comando QUIT, que en caso de fallar saldrá completamente de la tabla (esto no se muestra en el diagrama 2 por claridad).

Los contactos pueden dividirse en cinco tipos:

- 1.DESC; que finaliza completamente la ejecución de todas las tablas (incluso si se esta en un nivel de llamadas a proceso profundo) y saltará a describir la localidad actual.
2. END; (contacto que es un grupo en si mismo) que saldrá completamente de la ejecución de todas las tablas y realizará un salto a la inicialización el juego, o bien hacia salir del intérprete.
- 3.Salida; cualquier acción que produzca la parada del procesamiento de la tabla de procesos actual volviendo a la tabla desde la que se llamo, o de vuelta al bucle principal en el caso de la tabla de respuestas, el proceso 1 o el 2). Por ejemplo INVEN, DONE etc.

4.Salida condicional; cualquier acción que detendrá el procesamiento de la tabla de procesos actual y salir a la tabla que llamo a este proceso. (o de vuelta al bucle principal si es la tabla de respuestas, proceso 1 o proceso 2) si falla al hacer su función requerida. Por ejemplo GET, PUTIN, etc. Si no fallan, siguen con el siguiente conducto.

5. Normal; cualquier acción que lleva a cabo su función, y permite a Superglús continuar ejecutando el siguiente conducto en la entrada actual . Por ejemplo COPYFF, PLUS, etc.

Puede parecer que QUIT actúa como una acción de tipo 4, pero es realmente una condición. El listado de conductos al final de este manual muestra de que tipo es cada uno.

Usando el Interprete

Cada intérprete Glulx tiene un modo distinto de ejecutar una aventura incluida en un fichero ULX, así que os sugiero os miréis la documentación del mismo.

Los Conductos

Aquí sigue una descripción detallada de cada conducto. Están divididos en grupos de acuerdo a su función dentro de Superglús; cosas como flags, objetos,. Etc. Y dan algunas pistas sobre su posible uso.

Se usarán algunas abreviaturas:

locno	Es una localidad válida.
locno+	Es una localidad válida, o bien 252 o "_" (que significa "no creado") o 253 o "WORN" (localidad de los objetos "puestos"), o 254 o "CARRIED" (para los objetos que llevamos), o 255 o "HERE", que se convierte a la localidad actual del jugador.
mesno	Es un número de mensaje válido.
sysno	Es un número de mensaje de sistema válido.
flagno	Es cualquier flag (0-255).
procno	Es un número de proceso válido.
word	Es una palabra del tipo requerido, que esté presente en el vocabulario o bien "_", que asegura que no haya ninguna palabra. <i>Cuidado: no es un comodín que hace que cualquier palabra case, sino una que asegura de que no hay dicho tipo de palabra en la frase.</i>
value	Es un valor de 0 a 255.

Tabla 1 - abreviaturas

Condiciones

De localidad

AT locno	Se cumple si el jugador está en la localidad locno.
NOTAT locno	Se cumple si el jugador no está en la localidad locno.
ATGT locno	Se cumple si el jugador está en una localidad cuyo numero es mayor que locno.
ATLT locno	Se cumple si el jugador está en una localidad cuyo numero es menor que locno.

Tabla 2- condiciones de localidad

De localidad de objeto

PRESENT objno	Se cumple si el objeto está en la localidad actual, o bien lo llevamos o bien lo llevamos puesto.
ABSENT objno	Se cumple si el objeto ni está en la localidad actual, ni lo llevamos ni lo llevamos puesto.
WORN objno	Se cumple si llevamos puesto el objeto.
NOTWORN objno	Se cumple si no llevamos puesto el objeto.
CARRIED objno	Se cumple si llevamos el objeto en las manos.
NOTCARR objno	Se cumple si no llevamos el objeto en las manos (independientemente de que lo llevamos puesto, este en la localidad actual o en otro sitio)
ISAT objno locno+	Se cumple si el objeto esta en la localidad indicada.
ISNOTAT objno locno+	Se cumple si el objeto no está en la localidad indicada.

Tabla 3 - condiciones de localidad de objeto

De valor de los flags

ZERO flagno	Se cumple si el flag tiene valor cero.
NOTZERO flagno	Se cumple si el flag no tiene valor cero.
EQ flagno value	Se cumple si el flag tiene valor 'value'.
NOTEQ flagno value	Se cumple si el flag no tiene valor 'value'.
GT flagno value	Se cumple si el flag tiene un valor mayor que 'value'.
LT flagno value	Se cumple si el flag tiene un valor menor que 'value'.
SAME flagno1 flagno2	Se cumple si ambos flags tienen el mismo valor.
NOTSAME flagno1 flagno2	Se cumple si ambos flags no tienen el mismo valor.

Tabla 4 - condiciones de valor de los flags

De vocabulario

Estas condiciones permiten condicionar más la orden dada por el jugador, añadiendo adverbios, adjetivos, etc. Se recomienda que solo se use la palabra específica (o la ausencia de palabra usando "_") en caso de ser estrictamente necesario, para permitir más flexibilidad a la hora de que el jugador teclee la orden (y evitar el síndrome de la palabra exacta).

ADJECT1 word	Se cumple si el adjetivo del primer nombre de la SL es el indicado.
ADVERB word	Se cumple si el adverbio de la SL es el indicado.
PREP word	Se cumple si la preposición de la SL es el indicado.

NOUN2 word	Se cumple si el segundo nombre de la SL es el indicado.
ADJECT2 word	Se cumple si el adjetivo del segundo nombre de la SL es el indicado.

Tabla 5- condiciones de vocabulario

Aleatorias

Las siguientes condiciones son para condiciones aleatorias. Puedes usarlas para poner una posibilidad de que el jugador caiga de un árbol durante un rayo en una tormenta, para que un puente se caiga, etc. No abuses de esta posibilidad, permite siempre al jugador evitar esas posibilidades para no caer en muertes súbitas e inexplicables que restan jugabilidad.

CHANCE porcentaje	Se cumple si el tanto por ciento es menor o igual a un número del 0 al 100 obtenido aleatoriamente. De este modo un CHANCE 50 se cumplirá con una posibilidad del 50%, y un CHANCE 70 con un 70% de posibilidades. Para posibilidades menores al 1% se pueden concatenar varios CHANCE, de modo que CHANCE 1 seguido de CHANCE 1 tendrá una posibilidad de producirse de $1/100 \times 1/100 = 1/10000 = 0.0001 \%$, o por ejemplo CHANCE 50 seguido de CHANCE 1 será como $50/100 \times 1/100 = 50 / 10000 = 5 / 1000 = 0,5\%$
TIMEOUT	Esta condición se cumplirá si el ultimo input del jugador finalizo en TIMEOUT. Por ejemplo, una entrada en el proceso 2: <div style="text-align: center;"> TIMEOUT MESSAGE 0 </div> (en la que el mensaje 0 dice "Despiértate dormilón")
QUIT	Se muestra el SM12 ("Estás seguro?") y se llama a la rutina de pedir un texto al jugador. La condición se cumple si el jugador escribe un texto cuya primera letra sea la primera letra del SM30 ("S"). En caso contrario las acciones NEWTEXT y DONE se ejecutan.

Tabla 6- condiciones aleatorias

Acciones

Posición de objetos

GET objno	<p>Si llevamos el objeto puesto o en las manos, el mensaje SM25("Ya tengo _.") sale, y las acciones NEWTEXT y DONE se ejecutan.</p> <p>Si el objeto no esta en la localidad actual sale el mensaje SM26("No hay uno de esos por aquí") y se ejecutan NEWTEXT y DONE.</p> <p>Si el peso total de los objetos llevados y puestos por el jugador excede del máximo que puede llevar (flag 52) entonces el SM43("Eso pesa demasiado") aparece, y se ejecutan NEWTEXT y DONE.</p> <p>Si el jugador lleva el máximo numero de objetos que puede llevar (flag1 es mayor o igual que el flag 37) el mensaje "No puedo llevar mas cosas" aparece, y se ejecutan NEWTEXT y DONE. Además cualquier acción DOALL es cancelada.</p> <p>En otro caso el objeto pasa a estar 'llevado', se incrementa el flag 1 en 1 y se muestra el SM36 ("Coges _").</p>
DROP objno	<p>Si el objeto lo llevamos puesto sale el SM24 ("No puedo, lo llevo puesto") y NEWTEXT y DONE se ejecutan.</p> <p>Si el objeto está en la localidad actual (pero no lo llevamos en las manos ni puesto) se muestra el SM49 ("No tengo el _.") y NEWTEXT+DONE.</p> <p>Si el objeto no esta en la localidad actual el SM28 ("No tengo uno de esos") sale, y se ejecutan NEWTEXT y DONE.</p> <p>En otro caso el objeto pasa a estar en la localidad actual, se decrementa en 1 el flag 1 y el SM39("Dejas _.") se muestra.</p>
WEAR objno	<p>Si el objeto esta en la localidad actual (pero no puesto o en las manos) el SM49("No tengo _.") sale y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto lo llevamos puesto sale el SM29("Ya lo llevo puesto") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no lo llevamos en las manos sale el SM28("No tengo uno de esos") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no es una prenda el mensaje SM40("No puedo ponerme _") aparece y se ejecutan NEWTEXT y DONE.</p> <p>En caso contrario el objeto pasa a estar 'puesto' , se decrementa el flag 1, y sale el SM37("Ahora llevo puesto _").</p>

REMOVE objno	<p>Si el objeto lo llevamos o esta en la localidad actual (pero no puesto) sale el SM50("No llevo puesto _.") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no lo llevamos puesto aparece el SM23("No llevo puesto uno de esos") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no es una prenda aparece sale el SM41("No puedo quitarme _.") y se ejecutan NEWTEXT y DONE.</p> <p>Si el máximo numero de objetos que se pueden llevar en las manos es excedido (flag1>=flag37), sale el SM42 ("No puedo quitarme _. Mis manos están llenas.") y se ejecutan NEWTEXT y DONE.</p> <p>En otro caso el objeto pasa a estar 'llevado', se incrementa en 1 el flag 1 y se muestra el SM38("Me he quitado _").</p>
CREATE objno	El objeto es colocado en la localidad actual y el flag 1 se decrementa en 1 si el objeto estaba antes en las manos del jugador (carried).
DESTROY objno	El objeto es situado en la localidad 252 (No creado) y el flag 1 se decrementa en 1 si el objeto estaba antes en las manos del jugador (carried).
SWAP objno1 objno2	Los dos objetos intercambian su lugar. Atención: el flag 1 no se modifica, dado que en cualquier caso no va a cambiar el numero de objetos. El objeto actualmente referenciado pasa a ser el objno2.
PLACE objno locno+	El objeto es situado en la localidad indicada, se decrementa el flag 1 si el objeto estaba llevado, se incrementa el flag 1 si el objeto es puesto en la localidad 254 (CARRIED).
PUTO locno+	<p>El objeto actualmente referenciado (el que esta en el flag 51) se cambia para estar en la localidad indicada. En el flag54 queda su antigua localidad.</p> <p>Se decrementa el flag 1 si el objeto estaba llevado, se incrementa el flag 1 si el objeto es puesto en la localidad 254 (CARRIED).</p>
PUTIN objno locno	<p>Si el objeto lo llevamos puesto sale el SM24 ("No puedo. Llevo puesto _.") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto esta en la localidad actual (pero no puesto o en las manos) sale el SM49 ("No tengo _.") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no esta en la localidad actual, ni lo llevamos en las manos entonces sale el SM28 ("No tengo uno de esos.") y se ejecutan NEWTEXT y DONE.</p> <p>En otro caso el objeto se pone en la localidad indicada, se decrementa el flag 1 y se muestra el SM44 ("El _ esta ahora en", seguido de la descripción del objeto coincidente con locno, y luego se muestra el SM51(".").</p>

TAKEOUT objno locno	<p>Si el objeto lo llevamos puesto o en las manos sale el SM25 ("Ya tengo _.") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto esta en la localidad actual sale el SM45 ("_ no esta en "), seguido de la descripción del objeto cuyo numero coincide con locno, seguido del SM51(".") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no esta en la localidad actual ni en la localidad indicada sale el SM52("No hay uno de esos en"), seguido de la descripción del objeto cuyo numero coincide con locno, seguido del SM51(".") y se ejecutan NEWTEXT y DONE.</p> <p>Si el objeto no lo llevamos puesto o en las manos, y el peso total de los objetos llevados y puestos por el jugador mas el del objeto en cuestión exceden el máximo peso que el jugador puede llevar (flag 52) sale el SM43 ("Pesa demasiado") y se ejecutan NEWTEXT y DONE.</p> <p>Si el jugador ya lleva el máximo numero de objetos que puede llevar (flag1>=flag37) sale el SM37 ("No puedo llevar mas cosas") y se ejecutan NEWTEXT y DONE. Adicionalmente se interrumpe cualquier bucle DOALL.</p> <p>En otro caso el objeto pasa a ser llevado, se incrementa en 1 el flag 1 y sale el SM36 ("Coges _").</p> <p>Nota: No se comprueba, por parte de PUTIN y TAKEOUT, que el objeto referenciado por locno (generalmente un contenedor) esta presente. Esto debe ser comprobado por el programador.</p>
DROPALL	<p>Todos los objetos que el jugador lleve o lleve puestos son creados en la localidad actual (es decir, como si se dejara caer todos los objetos). Y el flag 1 se pone a cero. Este conducto se incluye por compatibilidad con antiguos sistemas de desarrollo (N.del T. ¿Quill?). Nótese que DOALL 254 llevará a cabo un verdadero DROP ALL, teniendo en cuenta cualquier acción especial incluida.</p>

Tabla 7- acciones de posición de objetos

Las seis acciones siguientes son versiones automáticas de GET, DROP, WEAR, REMOVE, PUTIN y TAKEOUT. Son automáticas en el sentido de que no se necesita especificar el objeto, sino que se convierte el actual nombre (y adjetivo) de la SL actual en un objeto y lo trata.

La búsqueda se hace por un objeto buscado en diferentes localidades en un orden de preferencia determinado. Esta búsqueda priorizada permite a Superglús saber qué objeto esta implicado si existe mas de un objeto con el mismo nombre (si el adjetivo no esta especificado) dependiendo de si el objeto esta puesto, en las manos, en la localidad actual o en un contenedor.

AUTOG	<p>Prioridad:</p> <ul style="list-style-type: none"> - Localidad actual - Llevado - Puesto <p>Se supone que el jugador pretenderá coger antes un objeto que este en la localidad actual. Cuando se encuentra un objeto se le pasa a la acción GET, si no se encuentra ninguno el SM8 ("No puedes hacer eso") aparece. En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p>
AUTOD	<p>Prioridad:</p> <ul style="list-style-type: none"> - Llevado - Puesto - Localidad actual <p>Se supone que el jugador pretenderá dejar antes un objeto que lleve. Cuando se encuentra un objeto se le pasa a la acción DROP, si no se encuentra ninguno el SM28 ("No tengo uno de eso") aparece. En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p>
AUTOW	<p>Prioridad:</p> <ul style="list-style-type: none"> - Llevado - Puesto - Localidad actual <p>Se supone que el jugador pretenderá ponerse antes un objeto que lleve. Cuando se encuentra un objeto se le pasa a la acción WEAR, si no se encuentra ninguno el SM8 ("No puedo hacer eso") aparece. En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p>
AUTOR	<p>Prioridad:</p> <ul style="list-style-type: none"> - Puesto - Llevado - Localidad actual <p>Se supone que el jugador pretenderá quitarse antes un objeto que lleve puesto. Cuando se encuentra un objeto se le pasa a la acción AUTOR, si no se encuentra ninguno el SM23 ("No puedo hacer eso") aparece. En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p>

AUTOP locno	<p>Prioridad:</p> <ul style="list-style-type: none"> - Llevado - Puesto - Localidad actual <p>Se supone que el jugador pretenderá poner antes un objeto que lleve encima que uno que lleve puesto. Cuando se encuentra un objeto se le pasa a la acción PUTIN, si no se encuentra ninguno y además no hay ningún objeto en el juego que corresponda se escribe el SM8 ("No puedo hacer eso") , si hay algún objeto en el juego que corresponda en algún lugar aparece el SM28 ("No tengo uno de esos"). En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p>
AUTOT locno	<p>Prioridad:</p> <ul style="list-style-type: none"> - En contenedor marcado por locno - Llevado - Puesto - Localidad actual <p>Se supone que el jugador pretenderá sacar antes un objeto que este en el contenedor. Cuando se encuentra un objeto se le pasa a la acción TAKEOUT, si no se encuentra ninguno y además no hay ningún objeto en el juego que corresponda se escribe el SM8 ("No puedo hacer eso") , si hay algún objeto en el juego que corresponda en algún lugar aparece el SM28 ("No hay _ en "), seguido de la descripción del objeto cuyo numero concuerda con locno, seguido del SM51("."). En cualquier caso las acciones NEWTEXT y DONE se ejecutan.</p> <p>Nota: No se comprueba en AUTOP o AUTOT que el objeto locno está presente, es trabajo del programador.</p>

Tabla 8- acciones automáticas

De objetos y flags

COPYOO objno1 objno2	El objeto objno2 pasa a estar en la misma localidad que el Objno1. El objeto actualmente referenciado pasa a ser objno2.
COPYOF objno flagno	<p>La localidad del objeto es copiada en el flag indicado. Esto puede usarse para examinar la localidad donde esta el objeto, comparar con otro flag, etc.</p> <p>Por ejemplo:</p> <pre> COPYOF 1 11 SAME 11 38 </pre> <p>puede usarse para comprobar que el objeto 1 esta en la misma localidad que el jugador, aunque lo cierto es que ISAT 1 255 sería una mejor opción.</p>
COPYFO flagno objno	El objeto indicado pasa a estar en la localidad indicada por el flag. Cualquier intento de copiar un flag que contenga 255 dará error, sin embargo darle un valor de una localidad que no existe no tendrá ningún efecto peligroso (aparte de ser imposible ver dicho objeto).

WHATO	Se buscará el objeto referenciado por el nombre y adjetivo 1 de la SL actual, en el orden de prioridad: llevado, puesto, localidad actual. Esto es porque se presupone que el uso de WHATO estará relacionado con objetos que lleve el jugador, mas que con objetos puestos o en la localidad actual. Si se encuentra un objeto su numero se incluye en el flag 51, junto con los parámetros de dicho objeto en los flags 54-57. Esto permite crear otras auto-acciones (el tutorial da un ejemplo de cómo 'tirar objetos desde un árbol').
WEIGH objno flagno	<p>El peso verdadero del objeto es calculado, esto es, si el objeto es normal se da su peso, pero si es un contenedor se calcula el peso de lo que hay dentro mas el peso del propio objeto. Se permite el calculo de contenedores dentro de contenedores. Nota: los contenedores con peso cero pesan cero, aunque tengan cosas dentro. (una interesante manera de hacer una bolsa mágica donde se puede meter de todo o algo parecido).</p> <p>El peso nunca pasará de 255, aunque haya más.</p>

Tabla 9- acciones de objetos y flags

De flags

SET flagno	El flag pasa a valer 255.
CLEAR flagno	El flag pasa a valer 0.
LET flagno value	El flag pasa valer el valor indicado.
PLUS flagno value	El flag es incrementado en value, pero si sobrepasa 255 se quedará en 255.
MINUS flagno value	El flag es decrementado en value, pero si baja de 0 se quedará en 0.
ADD flagno1 flagno2	El flag flagno2 pasa a valer la suma de flagno2 y flagno1,pero si sobrepasa 255 se quedará en 255.
SUB flagno1 flagno2	El flag flagno2 pasa a valer la resta de flagno2 menos flagno1,pero si baja de cero se queda en cero.
COPYFF flagno1 flagno2	El valor de flagno1 es copiado en flagno2.
RANDOM flagno	El flag indicado pasa a tener un valor aleatorio entre 1 y 100. Esto puede ser útil para tomar decisiones aleatorias de una manera mas flexible que con la condición CHANCE.

MOVE flagno	<p>Esta es una acción muy potente diseñada para manipular PSIs. Permite que la SL actual se use para comprobar sobre la tabla de conexiones para la localidad dada en el flag indicado. Si se encuentra el verbo se cambia el valor de flagno al de la localidad de destino, y se pasa al siguiente conducto. Si no se encuentra, o la localidad indicada no es válida, Superglús salta a la siguiente entrada. Puedes considerar que Superglús lleva a cabo las siguientes entradas imaginarias al salir de la tabla de respuestas si no hubo acción ejecutada.</p> <pre> _ _ MOVE 38 ;Intenta mover al jugador DESC ;Describe la nueva localidad. _ _ LT 33 14 ;palabra de movimiento? SYSMESS 7 ;"No puedo ir por ahí.." DONE _ _ SYSMESS 8 ;"No puedo hacer eso" </pre> <p>Esta capacidad puede ser usada para dar a los personajes una movimiento aleatorio en direcciones válidas, cambiando el verbo de la SL actual a un valor aleatorio de palabra de movimiento y usando MOVE para decidir si el personaje puede o no ir en esa dirección.</p>
-------------	---

Tabla 10- acciones de flags

De flags relativos al jugador

GOTO locno	Cambia la localidad del jugador, es equivalente a LET 38 locno.
WEIGHT flagno	<p>Calcula el peso real de todos los objetos llevados y puestos por el jugador, incluido lo que hay en los contenedores . El máximo peso calculable es 255, si hay mas peso se obtendrá 255.</p> <p>Puede usarse para asegurarse de que el jugador no lleva demasiado peso al cruzar un puente frágil, etc.</p>
ABILITY value1 value2	Esta acción fija los valores del flag 37 y flag 58, es decir, el numero máximo de objetos que el jugador puede llevar y el máximo peso que el jugador puede llevar. No se hacen comprobaciones sobre si el jugador lleva ya más del máximo. Equivale a LET 37 value1 y LET 52 value2.

Tabla 11- acciones de flags relativos al jugador

De modos de pantalla y formato

MODE value	<p>Hay dos modos de pantalla, controlados por el flag 40 y establecidos mediante la orden MODE.</p> <p>Modo 0</p> <p>La pantalla se borra antes de describir localidad.</p> <p>Modo 1 (o cualquier otro valor distinto de cero)</p> <p>La pantalla no se borra antes de describir localidad.</p>
------------	--

PROMPT sysno	Provoca que el mensaje sysno sea usado para mostrarlo como mensaje antes del input del jugador. Un valor de cero (valor por defecto) hará que Superglús seleccione un de entre los SM2, SM3, SM4 y SM5 aleatoriamente con un ratio de probabilidad de 30:30:30:10 respectivamente. Nótese que esto no afecta a los textos mostrados por END o QUIT.
TIME duración opción	<p>Permite que la entrada de datos del jugador pueda tener un timeout tras una duración especificada en segundos, es decir, el proceso 2 se ejecutará de nuevo si el jugador no teclea nada en determinado tiempo. Esta acción altera los flags 48 y 49. 'opción' permite que pueda ocurrir también durante un ANYKEY. Para calcular el numero para usar como 'opción' simplemente suma los números de las opciones siguientes que quieras usar:</p> <p>1 - Solo cuando este esperando la entrada de datos del jugador vacía. 4 - También en los ANYKEY</p> <p>Ejemplo. TIME 6 5 (opción = 1+4) permitirá 6 segundos de inactividad independientemente de si se esta pidiendo una entrada al jugador o un ANYKEY. Sin embargo TIME 5 3 (opción = 1+2) permite solo que se salte si el jugador no ha tecleado al menos un carácter pero no en los ANYKEY o si el jugador ha tecleado algo (aunque no haya llegado a pulsar enter).</p> <p>Time 0 0 impide los timeouts (valor por defecto).</p>

Tabla 12 - acciones de modo de pantalla y formato

De escribir flags

PRINT flagno	<p>El valor decimal del flag es mostrado en pantalla, sin espacios por delante o por detrás. Es una acción muy útil: digamos que el flag 100 contiene el numero de monedas que lleva el jugador, entonces una entrada en una tabla de procesos:</p> <pre>MES 10 ;"llevas "</pre> <pre>PRINT 100</pre> <pre>MESSAGE 11 ;" monedas de oro."</pre> <p>puede usarse para mostrarle al jugador el dinero que lleva.</p>
URNS	Se usan los SM17-20 "Llevas jugados x turno(s)." , donde x es flag 31 + 256 x flag 32.
SCORE	Se muestran los SM21-22 "Tu puntuación es de un x%", donde x es el flag 30.

Tabla 13- acciones de escribir flags

De control de pantalla

CLS	Borra la pantalla.
NEWLINE	Escribe un retorno de carro (salta a la linea siguiente de escritura).
MES mesno	Escribe el mensaje especificado.

MESSAGE mesno	Escribe el mensaje especificado y después ejecuta una acción NEWLINE
SYSMESS sysno	Escribe el mensaje del sistema especificado.

Tabla 14- acciones de control de pantalla

De listado de objetos

LISTOBJ	Si hay objetos presentes entonces se muestra el SM1("Puedes ver:"), seguido de una lista de todos los objetos presentes en la localidad. Si no hay objetos presentes no se escribe nada.
LISTAT locno+	Si hay objetos presentes en la localidad indicada se les lista, en otro caso se muestra el SM53("nada."). Nótese que generalmente deberá precederse esta orden de un mensaje del tipo "En la bolsa hay ". Es posible crear una alternativa a la acción INVEN que se describe justo después usando LISTAT, indicando la localidades WORN(253) o CARRIED (254).
INVEN	Esta acción no se ve afectada por el flag que indica listado 'continuo' por compatibilidad con sistemas de creación de aventuras anteriores (N.del T.:¿Quill?). Se muestra el SM9("Llevo:"). Si no se llevan objetos se muestra el mensaje SM11("Nada de nada"). En otro caso se muestra la descripción de cada objeto en una línea separada. Si un objeto lo llevamos puesto su descripción va seguida del SM10 ("puesto/a"). Se ejecuta la acción DONE.

Tabla 15- acciones de listado de objetos

Acciones de Control

DESC	Cancela cualquier bucle DOALL, cualquier subproceso en ejecución, y salta a describir la localidad actual.
END	Se escribe el SM13 ("¿Quieres jugar de nuevo?") y se llama a la rutina de input. Cualquier DOALL y llamadas a subprocesos son cancelados. Si la respuesta del jugador no empieza por el primer carácter del SM31 se hace un salto a la inicialización (se reinicia el juego). En otro caso el interprete glulx se cierra.

Tabla 16- acciones de control

De salida de proceso

DONE	Esta acción salta al final de una tabla de procesos indicando a Superglús que una acción ha sido realizada. Es decir, no se ejecutan mas conductos o entradas. Se realizara entonces el retorno al proceso que llamo al proceso actual, al bucle principal o al punto de inicio de un bucle DOALL activo.
NOTDONE	Esta acción salta al final del proceso actual y le indica a Superglús que no se ha realizado ninguna acción. Es decir, no se ejecutarán mas entradas o conductos del proceso actual. Se realizara entonces el retorno al proceso que llamo al proceso actual, al bucle principal o al punto de inicio de un bucle DOALL activo. Esto hará que Superglús escriba uno de los mensajes "No puedes" si es necesario (por ejemplo, si no hay una conexión valida para el verbo de la SL actual).

OK	Muestra el SM15 ("OK") y se ejecuta la acción DONE.
----	---

Tabla 17- acciones de salida de proceso

De grabación/carga de partida

SAVE	Esta acción graba la partida. En detalle, es el intérprete el que pide un nombre de archivo al jugador. Si el fichero existe es borrado. Se crea el fichero y se graba la partida en él. Si hay algún error sale el mensaje del sistema SM56("Error de E/S") y se ejecutan ANYKEY y DESC. En caso de grabarse DESC se ejecuta.
LOAD	Carga la partida. En detalle, se obtiene un nombre de fichero (que el intérprete pide al jugador), si no se encuentra el fichero se muestra el SM54 ("Archivo no encontrado.") y se ejecutan ANYKEY y DESC. Si falla al cargar sale el SM55 ("Fichero corrupto.") y el juego es reiniciado.
RAMSAVE	De una manera similar a SAVE esta orden graba la partida, pero no a un fichero de disco sino a un buffer de memoria. Evidentemente esto es volátil y si se acaba el juego se pierde la grabación.
RAMLOAD flagno	<p>Esta acción es la pareja de RAMSAVE, restaura el buffer de memoria. El parámetro establece el último flag a ser recuperado, y puede utilizarse para preservar valores en el restaurado de partida.</p> <p>Por ejemplo:</p> <pre> RAMLO _ COPYFF 30 255 RAMLOAD 254 COPYFF 255 30 DESC </pre> <p>Puede utilizarse para mantener la puntuación actual, de modo que el jugador no pueda usar RAMSAVE/RAMLOAD como una manera de conseguir un 100%.</p> <p>Nota: A diferencia de con SAVE y LOAD las acciones de RAM permiten la ejecución del siguiente conducto. En general deben ir seguidas de DESC.</p>

Tabla 18- acciones de grabación/carga de partida

Estas acciones pueden usarse para implementar un comando OOPS, común en otros sistemas para llevar el juego al punto anterior (generalmente a un fracaso o muerte), creando una entrada en el proceso 2 que automáticamente haga un RAMSAVE en cada turno.

De pausa

ANYKEY	Se muestra el SM16 ("pulse una tecla para continuar") y se espera la pulsación de una tecla o que ocurra un timeout (si esta habilitado el timeout durante los ANYKEY).
PAUSE value	Pausa el juego durante value / 50 segundos. Sin embargo, si el valor es cero la pausa es por 256/50 segundos (05:06 minutos)

Tabla 19- acciones de pausa

De control del parser (analizador de frases)

PARSE	<p>Esta acción ha sido desarrollada para manejar conversación con PSIs. (una frase entrecomillada). Cualquier cadena es convertida en la SL actual, borrando la SL que hubiera. Si no hay frase o no es valida Superglús seguirá con el siguiente conducto. En caso contrario se pasa a la siguiente entrada pero con la nueva SL. Dado que sobrescribe a anterior SL debe ser utilizado sólo en un subproceso. Dicho proceso tendrá el aspecto de:</p> <pre> _ _ PARSE ;haz esto siempre MESSAGE x ;"No te entiende" DONE word word CondAct list ;Cualquier frase que el PSI entienda _ _ MESSAGE x ;"No te entiende muy bien" habrá una o mas entradas de llamada que serán algo como: DECIR name SAME pos 38 ;Esta aquí el PSI? PROCESS y ;Decodificar conversación DONE ;la SL ha sido destruida, así que siempre DONE DECIR name MESSAGE x ;"no esta aquí!" DONE </pre>
NEWTEXT	<p>Fuerza la perdida de cualquier frase que pudiera quedar en la linea de input. Se usa para prevenir a un jugador de seguir haciendo una orden compleja si falló en una anterior.</p> <p>COGER ESPADA Y MATAR OGRO dado que atacar un ogro sin la espada puede ser peligroso.</p>

Tabla 20- acciones del analizador de frases

De tipo variado

BELL	Hace pitar el teclado (Ctrl-G)
PROCESS procno	Esta acción es muy potente, transfiriendo la atención de PAW al proceso especificado. Nótese que esta es una llamada real a subrutina y cualquier salida del proceso llevara el control al conducto que siga a PROCESS. Se pueden hacer llamadas anidadas (llamar a un proceso dentro de otro y así sucesivamente).

DOALL locno+	<p>Otra potente acción que permite hacer ordenes del tipo "TODO" (COGER TODO, etc.):</p> <ol style="list-style-type: none"> 1.Se intenta encontrar un objeto en la localidad locno. Si no se encuentra se cancela el DOALL y se ejecuta un DONE. 2.El objeto es pasado como nombre a la SL actual (y el adjetivo, si lo hay). Si el nombre y adjetivo1 concuerdan con el nombre y adjetivo 2 de la SL se salta al paso 1. Esto implementa el "Verbo TODO EXCEPTO objeto". 3.El siguiente conducto o entrada en la tabla es ejecutado. Esto en realidad convierte un "verbo Todo" en "verbo objeto", como si el jugador lo hubiera puesto. 4.Cuando se hace un intento de salir de la tabla actual, si el DOALL está activo, entonces en lugar de salir se volverá al punto 1, pero empezando por el siguiente objeto en la localidad. <p>La ramificación principal de la búsqueda hace que los objetos de igual nombre y distinto adjetivo, deban ser comprobados en orden ascendente de número de objeto, o uno de ellos podría pasarse por alto.</p> <p>Al usar DOALL para implementar cosas como ABRIR TODO debe tenerse en cuenta que las puertas deben estar consideradas objetos, y que podría intentarse abrir algunos objetos que no son puertas, por ejemplo:</p> <pre> ABRIR TODO Abres la puerta norte. Abres la puerta sur. Abres la caja de bombones. </pre>
RESET locno+	<p>Esta acción esta pensada para permitir la implementación de juegos multi-parte donde los objetos no son llevados a su localidad inicial.</p> <p>Todos los objetos que pueden estar presentes en ambas partes deben estar presentes (y con la misma descripción). Los demás pueden ser reutilizados en cada parte como se desee.</p> <p>Cualquier objeto presente en la localidad actual es llevado a la localidad locno, y la localidad actual del jugador es cambiada a locno. Los demás objetos se llevan a sus localidades de inicio. No hay efecto en los flags. Se ejecuta DESC al final.</p>
EXTERN value	Llama a una rutina externa. No implementado en Superglús (aun).
TRANSCRIPT value	Si value es un valor distinto de cero, el intérprete pide al jugador un nombre de archivo donde guardar la transcripción de la partida, es decir, todo el texto que aparezca desde este momento, tanto el escrito por el jugador como el escrito por el juego. Si value es 0, deja de escribir la transcripción. Este conducto es útil para el testeo de juegos.

Tabla 21- acciones varias

De estilos, gráficos y sonidos

PAPER value	<p>Cambia el color de fondo de la ventana de gráficos, según la siguiente tabla:</p> <p>0: negro</p> <p>1: azul</p> <p>2: rojo</p> <p>3: magenta</p> <p>4: verde</p> <p>5: turquesa</p> <p>6: amarillo</p> <p>7: blanco</p> <p>8: negro</p> <p>9: negro</p>
CHARSET value	Cambia el estilo de escritura de texto a partir de este punto al estilo Glk dado por el valor.
PICTURE value	Muestra el gráfico indicado por value en la ventana gráfica.
GRAPHIC value	Activa o desactiva la visualización de gráficos. Si value es 0, se activa; si vale cualquier otro valor, se desactiva.
BEEP value value	Reproduce el efecto de sonido cuyo identificador viene dado por el primer valor. El segundo valor se ignora y se mantiene por compatibilidad con PAW.
SOUND value	Activa o desactiva el sonido en el juego. Si value es 0, se activa; si vale cualquier otro valor, se desactiva.

Tabla 22- acciones de estilos, gráficos y sonidos

Descripción detallada del fichero fuente

El fichero fuente consiste en una serie de secciones interrelacionadas describiendo la aventura:

La sección de control (CTL)

Esta sección le dice al compilador cual va a ser el carácter que va a ser usado como la palabra nula. En Superglús dicho carácter se da por supuesto que es el guión bajo, subrayado (underline/underscore). Siempre debe ser el guión bajo, esta sección se ha mantenido por compatibilidad con el PAW. No se aceptará otro carácter.

La sección de Vocabulario (VOC)

Cada entrada en esta sección contiene una palabra (o los diez primeros caracteres de una palabra), un numero de palabra y un tipo de palabra. Las palabras con igual tipo y valor son consideradas sinónimas. Los caracteres mas alla del decimo son ignorados con lo cual da igual que el jugador ponga *contabiliza* que *contabiliz*, y asi mismo es indiferente como lo ponga el programador al hacer la aventura.

La sección de mensajes del sistema (STX)

Esta sección contiene mensajes usados por el interprete, que son numerados desde el numero 0. La descripción del intérprete, en capítulos anteriores, indica cuando son usados dichos mensajes. Adicionalmente pueden incluirse más mensajes a favor del jugador.

La sección de mensajes (MTX)

Esta sección contiene el texto de cualquier mensaje que el programador requiera para su aventura, partiendo del mensaje 0.

La sección de textos de objeto (OTX)

Esta sección, que tiene una entrada para cada objeto, contiene el texto que se usara como descripción de dichos objetos. Los objetos se numeran del 0 en adelante. El objeto 0 se da por supuesto que es una fuente de luz.

La sección de textos de localidad (LTX)

Esta sección tiene una entrada para cada una de las localidades del juego. La entradas se numeran de 0 en adelante. La localidad 0 es en la que empieza el juego.

La sección de conexiones (CON)

Esta sección tiene una entrada por cada localidad y cada una puede estar bien vacía (nula) o bien con movimientos. Un movimiento consiste en un verbo de movimiento seguido de un numero de localidad (a la que se accede desde dicha localidad poniendo el verbo de movimiento).

Una típica entrada será:-

```
SUR 6
ESTE 7
SALIR 6
NORTE 5
```

que significa que SALIR o SUR nos llevan a la localidad 6, ESTE a la localidad 7 y NORTE a

la 5. si hay verbos sinónimos no es necesario repetirlos, así si E y ESTE son sinónimos bastará con poner uno de ellos.

Nota 1. Cuando la aventura este en juego sólo el verbo de la SL causará el movimiento.

Nota 2. Si un movimiento es ejecutado en la tabla de respuestas usando la acción GOTO, entonces puede no ser necesaria en la tabla de conexiones, a no ser que dicha salida pueda ser realizada por un PSI mediante la orden MOVE.

La sección de definición de objetos(OBJ)

Esta sección tiene una entrada para cada objeto que especifica:-

- 1.La localidad inicial del objeto.
- 2.El peso del objeto.
- 3.Si el objeto es un contenedor.
- 4.Si el objeto es una prenda.
- 5.El nombre y adjetivo asociados con el objeto.

Las tablas de procesos (PRO)

Estas secciones son el corazón del código de Superglús:

La tabla de respuestas es el proceso 0

Cada entrada contiene un verbo y un nombre que casaran o no con la SL actual, seguidas de un numero determinado de conductos.

Proceso 1

Ejecutado por Superglús tras la descripción de localidad.

Proceso 2

Ejecutado por Superglús cada turno.

Proceso 3 (y siguientes)

Procesos opcionales para ser usados por el programador a los que se puede llamar con la acción PROCESS.

Sintaxis del fichero fuente

El fichero fuente consiste en una serie de secciones que deben estar presentes en el orden correcto. Cualquier cosa tras un punto y coma que no sea parte de un texto es ignorada, considerándose un comentario.

Las líneas en blanco sólo se permiten en las tablas de procesos.

En las definiciones del fichero fuente habrá:

- 1.w-s significa espacio en blanco, es decir, espacios o el carácter TAB (tabulador)
2. Las cosas que vayan entre corchetes [] son opcionales.
- 3.EOL significa Fin de Linea

CTL

La sección de control consiste en dos líneas tal que así:

```
/CTL [comentario] EOL
NULLWORD [comentario] EOL
```

Nótese que NULLWORD es el carácter que se usará después como palabra nula. Debe ser obligatoriamente el carácter de guión bajo "_". Esta sección se mantiene por compatibilidad con el PAW.

VOC

La sección de vocabulario empieza con la línea:

```
/VOC [comentario] EOL
y puede ir seguida de cualquier numero de líneas tal que así:
[w-s] PALABRA w-s VALOR w-s TIPO_PALABRA [w-s[:comentario]] EOL
Notas
```

- 1.sólo los 5 primeros caracteres de la palabra son tenidos en cuenta
- 2.Son convertidas a mayúsculas, solo A-Z y 0-9 son caracteres válidos en una palabra.
- 3.No se admiten palabras duplicadas
- 4.El valor esta en el rango 1-254
- 5.El tipo de palabra debe ser uno de estos:

- VERB
- NOUN
- ADJECTIVE
- ADVERB
- PREPOSITION
- PRONOUN
- CONJUNCTION

[N del T: Por error en versiones anteriores de Superglús se puso CONJUGATION en lugar de CONJUNCTION. Aunque la correcta es CONJUNCTION y ya ha sido corregido se ha permitido CONJUGATION para que los juegos antiguos puedan compilarse]

- 6.En el intérprete los nombres <20 pueden ser usados como verbos si no hay verbo en la SL. Los verbos <14 son usados como verbos de movimiento.

STX, MTX, OTX & LTX

La sección de mensajes del sistema empieza con la línea:

```
/STX [comentario] EOL
```

La sección de mensajes empieza con la línea:

```
/MTX [comentario] EOL
```

La sección de textos de objeto empieza con la línea:

```
/OTX [commentario] EOL
```

La sección de textos de localidad empieza con la línea:

```
/LTX [commentario] EOL
```

En cada sección cada entrada consiste en:

```
/n [w-s [commentario]] EOL
seguido de cualquier numero de líneas de texto que no empiecen por /
```

Nota

1.n debe ser un numero consecutivo y empezar en 0

2.El límite en STX es 60-255

3.El límite en MTX es 1-255

4.El límite en OTX es 1-255

5.El límite en LTX es 1-252

Las líneas de texto deben contener solo caracteres ISO-8859-1 imprimibles.

El compilador une las líneas consecutivas no vacías con un espacio.

Por ejemplo:

Yo

soy grande.

Como tu.

Será escrito por el intérprete :

Yo soy grande. Como tu.

El compilador convierte cualquier línea en blanco en saltos de línea.

Yo soy

grande.

Como

tu.

Será escrito como:

Yo soy grande.

Como tu.

Del mismo modo:

Yo soy grande.

Como

tu.

Será escrito como:

Yo soy grande.

Como tu.

CON

La sección de conexiones empieza con la línea:-

```
/CON [comentario] EOL
```

Cada entrada consiste en:-

```
/n [w-s [comentario]] EOL
```

seguida de cualquier número de líneas:

```
[w-s] PALABRA w-s LOCNO {w-s[:comentario]} EOL
```

Nótese:

1.Debe haber el mismo número de entradas que en LTX

2.n debe empezar por 0 y ser consecutivo

3.PALABRA debe estar en el vocabulario como verbo (o nombre < 20).

4.LOCNO debe estar en LTX.

OBJ

La definición de objetos empieza con la línea:

```
/OBJ [comentario] EOL
```

Cada línea consiste en:-

```
/n w-s LN w-s WT w-s CONT w-s WR w-s NOMBRE w-s ADJETIVO [w-s[:com]] EOL
```

Nótese

- 1.n debe empezar por cero y ser consecutivo
- 2.Debe haber el mismo numero de entradas que en OTX
- 3.LN es la localidad de el objeto cuando el juego comienza y debe existir en LTX, o ser 252-254, WORN, CARRIED o el carácter NULLWORD.
 - 252 y el carácter NULLWORD indican 'no creado'
 - 253 y WORN significan 'puesto'
 - 254 y CARRIED significan 'llevado'
- 4.WT es el peso del objeto en el rango 0-63.
- 5.CONT especifica si el objeto es un contenedor. Puede ser Y o la NULLWORD.
- 6.WR especifica si el objeto es una prenda. Puede ser Y o la NULLWORD.
- 7.NOMBRE es un nombre del vocabulario que refiere el objeto o la NULLWORD.
- 8.ADJETIVO es un adjetivo en el vocabulario o la NULLWORD.
- 9.Si un objeto empieza estando 'puesto' entonces es obligatorio que sea prenda (WR=Y)
- 10.No puedes especificar un adjetivo si no especificas un nombre.

PRO

La sección de procesos consiste en un numero de tablas que empiezan por la línea:

```
/PRO [w-s] n [w-s[comentario]] EOL
```

Nótese

- 1.n es el número de proceso, debe ser consecutivo y empezar por 0
- 2.Debe siempre especificarse un mínimo de 3 procesos, del 0 al 2.
- 3.Las líneas en blanco entre entradas o w-s son ignorados.

Cada entrada empieza con:

```
V w-s N w-s PALABRA_CLAVE [w-s PARAM1 [w-s PARAM2]] [w-s[:comentario]] EOL
```

Seguida de cualquier numero de :-

```
w-s PALABRA_CLAVE [w-s PARAM1 [w-s PARAM2 ]] [w-s[:comentario]] EOL
```

Nótese

- 1.V debe ser un verbo en el vocabulario, un nombre <20 o la null word.
- 2.N debe ser un nombre en el vocabulario o la null word.
- 3.PALABRA_CLAVE debe ser una CONDICION o ACCIÓN.
- 4.Cualquier cosa que empiece en la columna uno es considerado que es el principio de la siguiente entrada.

Las CONDICIONES y ACCIONES y sus parámetros son:-

Condiciones	Acciones
-------------	----------

AT	locno		INVEN
NOTAT	locno		DESC
ATGT	locno		QUIT
ATLT	locno		END
PRESENT	objno		DONE
ABSENT	objno		OK
WORN	objno		ANYKEY
NOTWORN	objno		SAVE
CARRIED	objno		LOAD
NOTCARR	objno		URNS
CHANCE	porcentaje		SCORE
ZERO	flagno		CLS
NOTZERO	flagno		DROPALL
EQ	flagno	valor	AUTOG
GT	flagno	valor	AUTOD
LT	flagno	valor	AUTOW
ADJECT1	adjective		AUTOR
ADVERB	adverb		PAUSE valor
TIMEOUT			GOTO locno
ISAT	objno	locno+	MESSAGE mesno
PREP	preposition		REMOVE objno
NOUN2	noun		GET objno
ADJECT2	adjective		DROP objno
SAME	flagno	flagno	WEAR objno
NOTEQ	flagno	valor	DESTROY objno
NOTSAME	flagno	flagno	CREATE objno
ISNOTAT	objno	locno+	SWAP objno objno
			PLACE objno locno+
			SET flagno
			CLEAR flagno
			PLUS flagno valor
			MINUS flagno valor
			LET flagno valor
			NEWLINE
			PRINT flagno
			SYSMESS smesno

COPYOF	objno	flagno
COPYOO	objno	objno
COPYFO	flagno	objno
COPYFF	flagno	
LISTOBJ		
RAMSAVE		
RAMLOAD	flagno	
BELL		
ADD	flagno	flagno
SUB	flagno	flagno
PARSE		
LISTAT	locno+	
PROCESS	prono	
MES	mesno	
MODE	valor	
TIME	valor	valor
DOALL	locno+	
PROMPT	smesno	
WEIGH	objno	flagno
PUTIN	objno	locno
TAKEOUT	objno	flagno
NEWTEXT		
ABILITY	valor	valor
WEIGHT	flagno	
RANDOM	flagno	
WHATO		
RESET	locno	
PUTO	locno+	
NOTDONE		
AUTOP		
AUTOT		
MOVE		
TRANSCRIPT	value	
PAPER	value	
CHARSET	value	
PICTURE	value	

GRAPHIC	value	
BEEP	value	value
SOUND	value	

Tabla 23- condiciones y acciones con sus parámetros

Nótese

- 1.flagno y valor están en el rango 0-255.
- 2.porcentaje esta en el rango 1-99.
- 3.objno debe estar definido en OTX.
- 4.mesno debe estar definido en MTX.
- 5.smesno debe estar definido en STX.
- 6.prono debe ser el numero de una tabla de procesos
- 7.locno debe estar definido en LTX.
- 8.locno+ debe estar definido en LTX o ser 252-255 o la null word, 'WORN', 'CARRIED' o 'HERE'.
- 9.adjective debe ser un adjetivo en el vocabulario o la null word
- 10.adverb debe ser un adverbio en el vocabulario o la null word
- 11.preposition debe ser una preposición en el vocabulario o la null word
- 12.noun debe ser un nombre en el vocabulario o la null word

El compilador pgc

Uso del compilador

Para compilar un fichero fuente en formato de Superglús, ejecuta el siguiente comando:

```
> pgc unidad:\ruta\al\fichero\fuente.sce
```

Si el compilador encuentra el fichero fuente, lo lee y si no hay errores genera el fichero "unidad:\ruta\al\fichero\fuente.ula" en el mismo directorio donde se encuentre el fuente.sce. Este nuevo archivo es un archivo de texto con el código ensamblador completo de tu nueva aventura. Por sí sólo no es jugable, hay que ensamblarlo para generar un ejecutable de máquina Glulx que se pueda jugar en cualquier(*) intérprete de esta plataforma. Ese es el cometido del ensamblador glulxa.

Si se quiere y se tienen los conocimientos precisos para ello, este fichero en ensamblador se puede retocar como se desee (aunque no se recomienda, porque es muy fácil equivocarse y que el juego no funcione).

Uso del ensamblador glulxa

El programa glulxa.exe, que forma parte de la distribución de Superglús, es el encargado de convertir el fichero .ula generado por el compilador en un ejecutable de máquina Glulx que se pueda jugar en cualquier intérprete de este sistema. Para realizar esta conversión, es decir, ensamblar el juego, se debe introducir el siguiente comando:

```
> glulxa -i entrada.ula -o salida.ulx
```

donde "entrada.ula" es el nombre del fichero que generó el compilador, y "salida.ulx" es el nombre con el que se creará el fichero compilado. Normalmente no debería haber errores de ensamblado, porque se supone que el compilador genera código ensamblador correcto.

Código ensamblador generado

En el código ensamblador que se genera se incluyen en los lugares adecuados los archivos de librería .pgl que contienen las rutinas de tiempo de ejecución del juego, el "intérprete PAW". Si quieres hacer modificaciones al intérprete para todos tus juegos, es mejor que lo hagas en esas rutinas de tiempo de ejecución, que se encuentran en el directorio (carpeta) "pgl" que está en el lugar donde esté el compilador (es decir, si tu compilador pgc.exe está, por ejemplo, en C:\WINDOWS\COMMAND, las rutinas de tiempo de ejecución seguramente estén en C:\WINDOWS\COMMAND\PGL), y así tus cambios se darán para todos tus juegos.

Al igual que se indicó antes, sólo es recomendable tocar estos archivos si se tienen los conocimientos necesarios (básicamente, ensamblador de Glulx y sintaxis de los archivos .ula para glulxa).

Errores del compilador

A menudo el compilador parará en medio del proceso de compilación, dándote algún mensaje indicador del error que ha encontrado, así como del número de línea donde se dio cuenta de que había tal error. Especifico "donde se dio cuenta" porque muchas veces lo que ocurre es que malinterpreta el error, lo toma por otra cosa, y no es hasta varias líneas después que se da cuenta de que existe un error. Por eso es conveniente buscar el error en la línea indicada, y si allí no se encuentra, buscarlo en las líneas anteriores.

Existen básicamente tres tipos de errores:

- Errores léxico-sintácticos: se encontró un carácter o palabra que no debería estar en ese contexto.

- Errores semánticos: se encontró una palabra o valor que está fuera del rango permitido en esa posición.
- Errores generales: otros errores que no son ninguno de los anteriores, y que generalmente no se deben al código fuente, sino a otras circunstancias como falta de recursos, por ejemplo.

A continuación se listan todos los errores que el compilador puede darte, clasificados en esas tres categorías:

Errores léxico-sintácticos

"Error sintáctico: se esperaba XXXXX en la línea NNNNN."

donde XXXXX puede ser `"/ctl"`, `"/voc"`, `"/otx"`, `"/ltx"`, `"/con"`, `"/stx"`, `"/mtx"`, `"/obj"`, `"/pro"`, `"_"`, `"una palabra"`, `"un número"`, `"número"`, `"final de fichero"`.

Se ha encontrado un símbolo del lenguaje en el lugar donde debería aparecer otro. Busca secciones que te falten o que estén mal escritas.

Errores semánticos

"Error de vocabulario. Palabra duplicada en línea NNNNN."

Has repetido la misma palabra con la misma función. Elimina una de las repeticiones.

"Error semántico. El tipo de palabra no es válido en línea NNNNN."

Has escrito mal un tipo de palabra. Repásalo, y recuerda que una palabra puede ser: adverb, verb, pronoun, adjective, noun, preposition o conjunction.

"Error semántico. Texto no consecutivo en línea NNNNN."

Te has saltado un número de texto en una de las secciones de textos (`/STX`, `/MTX`, `/LTX`, `/OTX`). Recuerda que en cada una de ellas los textos se deben empezar a numerar con el 0 y ser consecutivos.

"Error semántico. Demasiados textos en línea NNNNNN."

Has intentado definir más textos de los que el sistema permite. Este error es poco probable que salga, puesto que el límite de mensajes es de más de dos mil millones (excepto en el caso de las localidades, que como máximo puede ser 251, y de los objetos, que como máximo puede ser 255).

"Error semántico. pocos textos de sistema en línea NNNNN."

Has olvidado poner uno o más textos de sistema. Recuerda que en Superglús deben existir como mínimo los 60 primeros mensajes del sistema.

"Error semántico. Conexiones desordenadas. Línea NNNNN."

En la sección `/CON` deben aparecer todas las localidades, empezando por la 0 y en orden, aunque no tengan conexiones con otras. Repásalo.

"Error semántico. "XXXXX" no es una dirección. Línea NNNNN."

En la sección de conexiones (`/CON`), has intentado conectar una localidad con otra por medio de una palabra que no es de movimiento. Seguramente te habrás confundido al escribirla.

Repásalo.

"Error semántico. "XXXXX" no es un valor válido para la posición. Línea NNNNN."

En la sección de definición de objetos (/OBJ), has intentado darle a un objeto una posición inicial no válida. Recuerda que tiene que ser un número de localidad válido, o bien la palabra CARRIED, o bien la palabra WORN, o bien un guión bajo ("_") para indicar que el objeto no está creado al inicio del juego.

"Error semántico. Se encontró "XXXXX" en lugar de "y", "Y" o "_". Línea NNNNN."

Recuerda que en la sección de definición de objetos (/OBJ), para los atributos de "contenedor" y "prenda", los únicos valores válidos son los mencionados.

"Error semántico. "XXXXX" no es una palabra del vocabulario. Línea NNNNN."

En la sección de definición de objetos (/OBJ), se ha intentado poner como nombre o adjetivo de un objeto una palabra que no existe en el vocabulario. Si te has confundido al escribirla, corrígela. Si te habías olvidado de incluirla en la sección /VOC, ponla.

"Error semántico. "XXXXX" no es un nombre del vocabulario. Línea NNNNN."

En la sección de definición de objetos (/OBJ), se ha intentado poner como nombre de un objeto una palabra que, si bien existe en el vocabulario, no está definida como nombre (noun). Si te has confundido al escribirla, corrígela. Si te habías olvidado de incluirla en la sección /VOC, ponla.

"Error semántico. "XXXXX" no es un adjetivo del vocabulario. Línea NNNNN."

En la sección de definición de objetos (/OBJ), se ha intentado poner como adjetivo de un objeto una palabra que, si bien existe en el vocabulario, no está definida como adjetivo (adjective). Si te has confundido al escribirla, corrígela. Si te habías olvidado de incluirla en la sección /VOC, ponla.

"Error semántico. Número de objeto no válido u orden incorrecto. Línea NNNNN."

En la sección de definición de objetos (/OBJ), se ha encontrado un número de objeto que no corresponde con el orden, o bien un número demasiado alto de acuerdo con los textos de objeto definidos en la sección /OTX. Recuerda que en la sección de objetos deben aparecer tantos como textos se hayan definido, empezando por el 0 y en orden ascendente hasta el último. Repásalo.

"Error semántico. "XXXXX" no es un conducto. Línea NNNNN."

Durante la compilación de un proceso se esperaba encontrar un nombre de conducto y se ha encontrado otra cosa. A menudo este error da como línea causante una en la que no se aprecia la falta. En este caso mira en las líneas anteriores y busca parámetros mal puestos o ausentes en otros conductos.

"Error semántico. Número de proceso no válido u orden incorrecto. Línea NNNNN."

Los procesos deben empezar por el número 0 (tabla de respuestas) y seguir consecutivamente (el 1, el 2, etc.). Posiblemente te hayas saltado un proceso. Repásalo.

"Error semántico. "XXXXX" no es un verbo del vocabulario. Línea NNNNN."

Durante la compilación de un proceso se esperaba encontrar un verbo al principio de una

entrada y se ha encontrado otra cosa. A menudo este error da como línea causante una en la que no se aprecia la falta. En este caso mira en las líneas anteriores y busca parámetros mal puestos o ausentes en otros conductos.

"Error semántico. El valor debe estar entre 0 y 255. Línea NNNNN."

Se ha pasado como parámetro de un conducto un número fuera del rango de 0 a 255. Repásalo.

"Error semántico. El valor debe estar entre 1 y 99. Línea NNNNN."

Se ha pasado como parámetro de un conducto que esperaba un porcentaje un número fuera del rango de 1 a 99. Repásalo.

"Error semántico. Número de objeto no válido. Línea NNNNN."

A un conducto que recibe como parámetro un número de objeto se le ha pasado un número que no corresponde con un objeto definido. Repásalo y comprueba que el objeto al que se refiere se ha definido (en las secciones /OTX y /OBJ).

"Error semántico. Número de mensaje no válido. Línea NNNNN."

A un conducto que recibe como parámetro un número de mensaje se le ha pasado un número que no corresponde con un mensaje definido. Repásalo y comprueba que el mensaje al que se refiere se ha definido (en la sección /MTX).

"Error semántico. Número de mensaje del sistema no válido. Línea NNNNN."

A un conducto que recibe como parámetro un número de mensaje del sistema se le ha pasado un número que no corresponde con un mensaje del sistema definido. Repásalo y comprueba que el mensaje del sistema al que se refiere se ha definido (en la sección /STX).

"Error semántico. Número de proceso no válido. Línea NNNNN."

A un conducto que recibe como parámetro un número de proceso (PROCESS) se le ha pasado un número que no corresponde con un proceso definido. Repásalo y comprueba que el proceso al que se refiere se ha definido (como sección /PRO).

"Error semántico. Número de localidad no válido. Línea NNNNN."

A un conducto que recibe como parámetro un número de localidad se le ha pasado un número que no corresponde con una localidad definida. Repásalo y comprueba que la localidad a la que se refiere se ha definido (en la sección /LTX).

Errores generales

"Error: comando no válido, falta el fichero de entrada"

Al intentar ejecutar el compilador, éste no ha recibido un nombre de fichero como parámetro, o bien no se ha podido abrir el fichero.

"Error: no hay suficiente memoria. Línea NNNNN."

Al llegar a la línea NNNNN del archivo fuente, no se ha podido reservar memoria para seguir compilando. Puede indicar que la aventura es demasiado grande o que hay algún problema con el compilador.

"Error: error genérico."

Este tipo de error se da cuando hay algún problema interno con el compilador. Se sugiere que se informe a los desarrolladores.

Ensayos

Aquí comienza una serie de temas que son de difícil explicación en otros puntos dado que usan varias partes de PAW:

El parser da una pequeña lección de castellano.

Los Objetos serán mas flexibles después de leer esto.

Luz y Oscuridad da algo de luz sobre dicho asunto

PSIs añade algo de carácter a tus juegos.

El Parser

El parser trabaja buscando en cada linea de ordenes del jugador palabras que estén en el vocabulario, extrayendo frases que puede convertir en sentencias lógicas (SL).

Cuando se extrae una frase, las tablas de respuestas y conexiones son examinadas para ver si la SL es reconocida. En caso contrario el SM 8 ("No puedo hacer eso") o el SM 7 ("No puedo ir en esa dirección") son mostrados dependiendo del valor del verbo (>13 o no). Después se solicita nuevas órdenes.

También se solicitan nuevas ordenes si alguna acción falla de alguna manera (por ejemplo, al intentar coger un objeto demasiado pesado), o si el autor lo fuerza con una acción NEWTEXT. De otro modo los resultados podrían ser catastróficos para el jugador, por ejemplo, en el caso de la orden COGE EL HACHA Y ATACA AL TROLL probablemente no se quiera atacar al troll si no se ha conseguido caer el hacha.

Si la SL se ejecuta con éxito entonces otra frase es extraída o un nuevo texto es solicitado si no hay mas texto en el buffer.

Las frases están separadas por las conjunciones ("Y" y "ENTONCES" habitualmente) y por cualquier signo de puntuación.

Se puede usar un pronombre ("LO", normalmente) para referirse al nombre/adjetivo usado en la orden anterior, incluso si ha sido en otra entrada anterior del jugador. Los nombres cuyas palabras correspondientes tengan valores menores de 50 se consideran nombres propios y no afectan a este pronombre. Sin embargo, debido a una limitación del intérprete de Superglús, el jugador debe escribir el pronombre separado del verbo al que acompañe, así: "SACA LA ESPADA Y BLANDE LA" (nótese que "LA" está separado por un espacio de "BLANDE"). Se espera poder resolver esta inconveniencia en alguna versión del futuro cercano de Superglús.

El formato de la SL es el siguiente:

Verbo(Adverbio)(Nombre1(Adjetivo1))(preposición)(Nombre2(Adjetivo2))

donde los tipos entre paréntesis son opcionales, es decir, la frase mínima es un verbo (o un nombre convertible a verbo, que es un nombre cuyo numero es <20, el cual si no hay verbo en la frase será convertido a verbo, por ejemplo. NORTE). Si se omite el verbo la SL entenderá que el verbo que se usó para la anterior SL es el correcto, es decir, COGER ESPADA Y ESCUDO funcionará correctamente. El pronombre actual será reemplazado por ESPADA en dicho ejemplo. No cambiará hasta que el verbo o nombre convertible cambie.

Nótese que la frase no tiene por que ser estrictamente escrita por el jugador en este formato. Por ejemplo:

```
COGE LA ESPADA PEQUEÑA RAPIDAMENTE
RAPIDAMENTE COGE LA ESPADA PEQUEÑA
RAPIDAMENTE LA ESPADA PEQUEÑA COGE
```

Son ejemplos equivalente que producen la misma SL, aunque la ultima opción es rara de encontrar en un castellano que no sea lírico.

Un frase real puede ser :

COGE TODO. ABRE LA PUERTA Y VE AL SUR, ENTONCES COGE EL CUBO Y MIRA DENTRO DE EL.

Se convertirá en cinco SL:

COGE TODO

ABRE PUERTA (porque LA no está en el vocabulario)

SUR (porque VE y AL no están en el vocabulario)

COGE CUBO

MIRA DENTRO (preposición) CUBO (de EL, pronombre)

Nótese que DOALL no generará el objeto descrito por el nombre 2 (Adjetivo 2) de la SL. Eso facilita un metodo simple de implementar EXCEPTO. Por ejemplo, COGE TODO EXCEPTO EL PEZ. Tiene el también el efecto secundario de no permitir PON TODO EXCEPTO EL PEZ EN EL CUBO, dado que tiene tres nombres.

Objetos

Los guiones de subrayado (_) en los textos son convertidos durante el juego en el último objeto referenciado por GET, DROP, DESTROY, etc. Esto es principalmente debido al hecho de que aunque GET, DROP, etc. indican su éxito (o fallo) pueden ser usados para examinar objetos y otros informes automáticos. El flag 53 controla el modo en el que los objetos son mostrados en las acciones LISTOBJ y LISTAT. Si el flag vale 64 (es decir, el bit 6 está activado) los objetos serán listados sin saltos de linea entre ellos, formando una frase correcta en castellano.

El formato es el siguiente:

SM53 ("nada") - solo puede aparecer con LISTAT

objeto SM48(" ")

objeto SM47(" y ") objeto SM48(" ")

objeto SM46(", ") objeto SM47 (" y ") objeto SM48(" ")

Además, el bit 7 del flag 53 se activará (el flag valdrá mas de 127) si se mostro al menos un objeto. Esto te permite decidir si se quiere añadir una acción NEWLINE.

Una acción LISTAT generalmente irá precedida de un mensaje.

La descripción de objeto es construida a partir de la descripción completa dada en la tabla de textos de objeto. El formato preferido para dicha descripción es:

Artículo.indefinido nombre (adjetivo) . texto extra.

donde , el artículo indefinido es "Un, Una, Algunos, etc.", y los adjetivos y nombres tienen letras minúsculas (por ejemplo "Una llave pequeña" o "Una naranja. Mas bien podrida").

Superglús extrae una descripción del objeto de dos modos:

1.Para GET, DROP etc (es decir, "_") el artículo indefinido se salta, y la descripción es escrita hasta el primer punto y seguido. Por ejemplo "He cogido: llave pequeña"

2.Para una lista compuesta de objetos se cambia la letra del artículo indefinido a minúsculas. Y la descripción es escrita hasta el primer punto y seguido. Por ejemplo: "En la bolsa hay una llave pequeña".

Obviamente, si no usamos "_" o listas compuestas, en tal caso se es libre de escribir los objetos como uno quiera.

[N. del A.: la descripción dada corresponde al comportamiento del PAW PC original. Sin embargo, en Superglús en ambos casos se elimina la primera palabra de la descripción del objeto. Se espera corregir esta inconveniencia para una versión próxima de Superglús.]

[N. del T.: alguna facilidad que tiene el lenguaje inglés, como que el artículo indefinido sea neutro y se use igual en femenino como en masculino, no la tiene el castellano. Así nos vemos obligados a poner los mensajes de GET, DROP etc. al estilo del punto 1 anterior o bien a poner el artículo indefinido en minúsculas para que Superglús entienda que no hay artículo indefinido y muestre todo el texto del mensaje (hasta el punto y seguido) "Coges una llave pequeña").]

Importante: Si un objeto va a ser un contenedor, debe haber una localidad sin usar con el mismo numero que el objeto, dado que Superglús la usará como "dentro de" de dicho objeto. Por ejemplo el objeto cuyo numero es 4 si es un contenedor obliga a crear la localidad 4 y dejarla sin usar como localidad normal.

Luz y Oscuridad

La oscuridad es algo común en las aventuras de hoy en día, usado correctamente puede dar cierta sensación de realismo.

En Superglús, se puede 'crear' oscuridad dando al flag 0 cualquier valor distinto de cero. Esto debe hacerse cuando el jugador se mueve dentro y fuera de 'la oscuridad', es decir, el movimiento debe hacerse con un GOTO en la tabla de respuestas para permitir la ejecución de la acción CLEAR y SET.

Si al jugador se le va a dar una fuente de luz entonces el objeto 0 es la mejor manera de hacerlo. Una fuente de luz no tiene por que ser una antorcha o lámpara, con un poco de imaginación puede ser unas gafas de infla-rojos o un haz láser.

Pongamos por ejemplo la creación de un ciclo de día y noche, sobre 24 rodajas de tiempo que pongamos que equivalen a una hora.

Las entradas necesarias en el proceso 2 son:

—	—	EQ	5	0	;Fin de ciclo
		LET	5	24	;iniciar de nuevo el ciclo
—	—	EQ	5	18	;Puesta de sol
		SET	0		
		MESSAGE	x		
—	—	EQ	5	6	;Salida del sol
		CLEAR	0		
		MESSAGE	y		

Importe: si parte del juego es bajo tierra o dentro de un edificio no olvides comprobar si el jugador puede realmente ver si sale el sol o se pone desde donde está, antes de poner los mensajes.

Pseudo-Inteligencias

Lo mas importante a recordar es que un personaje (o PSI) es una palabra en el vocabulario (habitualmente un nombre con valor inferior a 50 para que sea nombre propio), algunos flags, una serie de mensajes y algunas entradas en una o mas tablas de procesos. Un flag guarda donde están, los mensajes dan información sobre sus movimientos y acciones y las tablas de procesos lo atan todo junto.

Bien, imaginemos un personaje llamado Gervasio que puede moverse de manera independiente . Es descrito en el vocabulario como GERVA (palabra con valor 25, nombre). El flag 20 contiene su localidad actual. El proceso 3 llevara toda conversación que se pueda tener con él, mientras que el process 4 manejará sus movimientos y acciones. Las siguientes entradas le permitirán moverse cuando tu le pidas que lo haga. Tras un corto tiempo se 'aburrirá' y se desvanecerá en una nube de humo.

Mensaje 1	Gervasio no parece entender lo que le dices.
Mensaje 2	¡No hay nadie aquí con ese nombre!
Mensaje 3	Gervasio contesta "Hola"
Mensaje 4	Gervasio va por ese camino como si no tuviera otra cosa que hacer.
Mensaje 5	Gervasio el mago está aquí.
Mensaje 6	Gervasio ignora 'educadamente' lo que dices.
Mensaje 7	Gervasio gira su rostro para mirarte y en su profunda voz anuncia: "Estoy cansado de todo esto, me voy a un juego mas grande" y con un "puff" desaparece en medio de una nube de humo azul.

Tabla 24- mensajes ejemplo del PSI

Primero: la presencia de Gervasio en la localidad debe anunciarse, así que en el proceso 1, que es llamado tras la descripción de cada localidad, comprobamos si está aquí (es decir, que el flag 20 es igual al flag 38).

Nótese que nos aseguramos que no estamos en la localidad 0 porque es la localidad inicial de presentación y en ese momento seguro que el flag 20 y el 38 valen los dos 0.

```

-      -      SAME      20  38  ;Comprobar que Gervasio está aquí
NOTAT      0      ;No estamos en la localidad 0
MESSAGE    5      ;Decir que Gervasio está aquí

```

Para manejar conversaciones con Gervasio necesitamos dos entradas en la tabla de respuestas tal que así: -

```

DECIR      GERVA PREP      A      ;Esto puede omitirse para
;                                permitir que las sentencias cortas
;                                verbo+nombre sean entendidas

      SAME      20  38  ;      Asegurarnos de que Gervasio está aquí
PROCESS    3      ;      Manejar conversación
DONE      ;      Prevenir que se ejecuten mas acciones con
la
;      SL que se le dice al jugador

```

```

DECIR      -      PREP      A      ;de nuevo opcional
MESSAGE    2      ; no hay nadie con ese nombre
DONE

```

Las siguientes entradas en el proceso 3:

```

-      -      PARSE      ;Esta orden siempre se ejecuta para convertir
;                                Las ordenes de entrada en una SL
MESSAGE    1      ;PARSE llega aquí si falla al tratar de formar SL
DONE      ;Nótese que la SL se ha corrompido y que no ;
; debe ejecutarse mas en la tabla

```

```

HOLA -      MESSAGE    3      ;Asumiendo que HOLA es un verbo del Vocabulario
DONE

```

```

-      -      LT      34  14  ;se le ha dado una orden de movimiento a Gervasio?
MOVE      20      ; comprobar que hay una salida en esa dirección
MESSAGE    4      ;Decirle al jugador que lo hizo si la había
DONE

```

```

-      -      MESSAGE    6      ;Te ignora

```

Obviamente harían falta muchas más entradas para dar a Gervasio la apariencia de alguien que entiende el habla, pero con unas pocas entradas inteligentes puede dar una amplia variedad de respuestas.

Finalmente; para dar a Gervasio una oportunidad de desaparecer cuando se aburra, necesitamos una entrada en el proceso 4 tal que:

```
—      —      EQ      20      2 ;En la localidad 2?
          CHANCE  10          ;10% posibilidad
          SET      20          ;La localidad 255 no existe
          AT        2          ;estamos donde el estaba?
          MESSAGE   7          ;PUFF! - Decir al jugador que desaparece
```

Y una entrada en el proceso 2 para llamar regularmente al proceso 4.

De este modo un personaje convincente puede ser construido. Un PSI añade una gran sensación de realismo a los juegos, especialmente si se requiere interacción con él.

Sumario

Los Mensajes del Sistema

SM0 - se usa en lugar de la descripción de localidad cuando esta oscuro.

SM1 - es mostrado por LISTOBJ si hay al menos un objeto presente.

SM2 to SM5 - son los cuatro prompt de la entrada de ordenes del jugador que son mostrados aleatoriamente a no ser que el flag 42 tome el valor de un numero de mensaje válido.

SM6 - es mostrado por el parser cuando es incapaz de entender nada en la orden del jugador.

SM7 - es mostrado si no se realizo acción (o se ejecuto NOTDONE) en la tabla de respuestas y el verbo es < 14.

SM8 - es mostrado si no se realizo acción (o se ejecuto NOTDONE) en la tabla de respuestas y el verbo es > 13.

SM9 to SM11 - son mostrados por la acción END.

SM12 - mostrado por la acción QUIT.

SM13 y 14 - son mostrados por la acción END.

SM15 - mensaje para la acción OK.

SM16 - mensaje para la acción ANYKEY.

SM17 to SM20 - Son los mensajes para la acción TURNS.

SM21 and SM22 - Son los mensajes para la acción SCORE.

SM23 and SM29 - son los primeros de muchos mensajes producidos por acciones de manipulación de objetos.

SM30 - La respuesta positiva esperado por END y QUIT.

SM31 - La respuesta negativa esperado por END y QUIT.

SM32 - Se produce cuando se ha llenado la pantalla de texto.

SM33 - el carácter del input.

SM34 - libre.

SM35 - Mostrado cuando ocurre un timeout.

SM36 to SM45 - son mas mensajes producidos por las acciones de manipulación.

SM46 - el enlace entre los objetos de una lista de objetos.

SM47 - el enlace final entre los dos últimos objetos de una lista de objetos.

SM48 - La terminación de una lista de objetos (mostrada por LISTOBJ y LISTAT, así que cuidado).

SM49 and SM50 – más mensajes de objetos.

SM51 – terminación para las frases compuestas de PUTIN/TAKEOUT (y AUTOP/AUTOT).

SM52 – mensaje de último objeto.

SM53 - mensaje para la acción LISTAT si no hay objetos que listar.

SM54 a SM60 son mensajes usados por las acciones SAVE y LOAD.

Función de los flags de Superglús

Los flags normales están libres para el uso en los juegos. Los flags de auto-decremento (2 a 10) están libres para uso, pero se debe estar seguro de para qué se usan. Otros flags solo deberían ser modificados usando la acción adecuada, o siendo muy cuidadoso con sus contenidos.

flag 0 Cuando no es cero indica que el juego está oscuro (ver también el objeto 0).

flag 1 Guarda la cantidad de objetos que el jugador lleva (no puestos).

Los siguientes nueve flags son decrementados si no valen 0 por Superglús:

flag 2 Cuando se describe una localidad.

flag 3 Cuando se describe una localidad y esta oscuro (flag 0 no vale 0).

flag 4 Cuando se describe una localidad y esta oscuro (flag 0 no vale 0) y el objeto 0 esta ausente.

flag 5 a 8 Cada turno (es decir, cada frase / timeout).

flag 9 Cada turno que esta oscuro.

flag 10 Cada turno que esta oscuro y el objeto 0 esta ausente.

flag 11 a 28 están libres para tus propios juegos.

flag 29 guarda los indicadores de control de gráficos en la versión para Spectrum de PAW.

flag 30 La puntuación.

flag 31/32 (LSB/MSB) guarda el numero de turnos que lleva jugando el jugador (realmente es el numero de frases extraídas del jugador).

flag 33 guarda el verbo de la SL actual.

flag 34 guarda el primer nombre de la SL actual.

flag 35 guarda el adjetivo de la SL actual.

flag 36 guarda el adverbio de la SL actual.

flag 37 guarda el numero máximo de objetos que el jugador puede llevar (inicialmente 4). Se cambia con la acción ABILITY.

flag 38 guarda la localidad actual del jugador.

flag 39 guarda la linea superior de la pantalla en la version Spectrum de PAW.

flag 40 guarda el modo de pantalla, se cambia con la acción MODE.

Bit 0. Impide que se borre la pantalla en un DESC.

flag 41 guarda la linea que separa grafico de texto en la version para Spectrum.

flag 42 guarda el prompt a usar (un numero de mensaje del sistema- 0 selecciona uno de cuatro mensajes aleatoriamente). Se cambia con la orden PROMPT.

flag 43 guarda la preposición de la SL actual.

flag 44 guarda el segundo nombre de la SL actual.

flag 45 guarda el adjetivo para el segundo nombre de la SL actual.

flag 46 guarda el nombre para el pronombre de la SL actual.

flag 47 guarda el adjetivo para el pronombre de la SL actual.

flag 48 guarda la duración del timeout.

flag 49 guarda los indicadores de control del timeout.

Bit 7 - Activo si ocurrió un timeout el ultimo turno.

Bit 2 - Activado si el timeout puede ocurrir en ANYKEY.

Bit 0 - Activar para que el timeout solo pueda ocurrir al inicio del input.

Se cambia usando TIME (al igual que el flag 48), TIMEOUT comprueba el bit 7 de este flag.

flag 50 guarda el objeto para el bucle DOALL. Es decir, el value que sigue en DOALL.

flag 51 guarda el ultimo objeto referenciado por GET/DROP/WEAR/WHATO etc. Este es el numero del objeto actualmente referenciado que será escrito en lugar del guión bajo en los textos.

flag 52 guarda la fuerza del jugador (el máximo peso de los objetos llevados o puestos, inicialmente 10).

flag 53 guarda los indicadores de escritura de objetos.

Bit 7 -Activar si cualquier objeto se ha impreso como parte de LISTOBJ o LISTAT.

Bit 6 -Activar para forzar listado de objetos continuo. Es decir, LET 53 64 hará que Superglús liste los objetos todos en la misma linea formando una frase válida.

flag 54 guarda la localidad actual del objeto actualmente referenciado.

flag 55 guarda el peso del objeto actualmente referenciado.

flag 56 es 128 si el objeto actualmente referenciado es contenedor.

flag 57 es 128 si el objeto actualmente referenciado es prenda.

flag 58 & 59 no deben usarse, dado que pueden ser usados para extensiones de Superglús.

Los flags del 60 a 255 están disponibles para tu propio uso.

Los CondActos

Condiciones:

AT	locno		Cierto si el jugador esta en la localidad especificada.
NOTAT	locno		Cierto si el jugador no está en la localidad especificada.
ATGT	locno		Cierto si el jugador está en una localidad cuyo número es mayor que el especificado.
ATLT	locno		Cierto si el jugador está en una localidad cuyo número es menor que el especificado.
PRESENT	objno		Cierto si el objeto está en la localidad actual del jugador, puesto o llevado.
ABSENT	objno		Cierto si el objeto no está ni en la localidad actual del jugador, ni puesto ni llevado.
WORN	objno		Cierto si el objeto lo lleva puesto el jugador.
NOTWORN	objno		Cierto si el objeto no lo lleva puesto el jugador.
CARRIED	objno		Cierto si el objeto lo lleva el jugador (pero no puesto).
NOTCARR	objno		Cierto si el objeto no lo lleva el jugador, o bien lo lleva puesto.

ISAT	objno	locno+	Cierto si el objeto se encuentra en la localidad indicada.
ISNOTAT	objno	locno+	Cierto si el objeto no se encuentra en la localidad indicada.
ZERO	flagno		Cierto si el flag tiene el valor 0.
NOTZERO	flagno		Cierto si el flag no tiene el valor 0.
EQ	flagno	0-255	Cierto si el flag tiene el valor especificado.
NOTEQ	flagno	0-255	Cierto si el flag no tiene el valor especificado.
GT	flagno	0-255	Cierto si el flag tiene un valor mayor que el dado.
LT	flagno	0-255	Cierto si el flag tiene un valor menor que el dado.
SAME	flagno	flagno	Cierto si los dos flags tienen el mismo valor.
NOTSAME	flagno	flagno	Cierto si los dos flags tienen valores distintos.
ADJECT1	word		Cierto si el adjetivo del primer nombre de la SL es la palabra dada.
ADVERB	word		Cierto si el adverbio de la SL es la palabra dada.
PREP	word		Cierto si la preposición de la SL es la palabra dada.
NOUN2	word		Cierto si el segundo nombre de la SL es la palabra dada.
ADJECT2	word		Cierto si el adjetivo del segundo nombre de la SL es la palabra dada.
CHANCE	0-99		Posibilidad aleatoria de éxito
TIMEOUT			Ultima entrada del jugador paso en <i>timeout</i>
QUIT			Preguntar al jugador si desea terminar y actuar en consecuencia.

Tabla 25- resumen de las condiciones

Acciones: (la segunda columna es el tipo de acción)

GET	4	objno		Intenta coger el objeto.
DROP	4	objno		Intenta dejar el objeto.
WEAR	4	objno		Intenta ponerse el objeto.
REMOVE	4	objno		Intenta quitarse el objeto.
CREATE	5	objno		Intenta colocar el objeto en la localidad actual del jugador.
DESTROY	5	objno		Intenta colocar el objeto en la localidad 255 (objeto inexistente).

SWAP	5	objno	objno	Intercambia las posiciones de los dos objetos.
PLACE	5	objno	locno+	Pone el objeto indicado en la localidad indicada.
PUTO	5	locno+		Pone el objeto actualmente referenciado en la localidad indicada.
PUTIN	4	objno	locno	Intenta meter el objeto en el contenedor correspondiente a la localidad.
TAKEOUT	4	objno	locno	Intenta sacar el objeto del contenedor correspondiente a la localidad.
DROPALL	5			Mueve todos los objetos que lleve el jugador a la localidad actual del mismo.
AUTOG	4			Saca de la SL el objeto referenciado y trata de cogerlo.
AUTOD	4			Saca de la SL el objeto referenciado y trata de dejarlo.
AUTOW	4			Saca de la SL el objeto referenciado y trata de ponérselo.
AUTOR	4			Saca de la SL el objeto referenciado y trata de quitárselo.
AUTOP	4	locno		Saca de la SL el objeto referenciado y trata de meterlo en el contenedor correspondiente a la localidad.
AUTOT	4	locno		Saca de la SL el objeto referenciado y trata de sacarlo del contenedor correspondiente a la localidad.
COPYOO	5	objno	objno	Pone el segundo objeto en la misma localidad que el primero.
COPYOF	5	objno	flagno	Copia posición del objeto a flag.
COPYFO	5	flagno	objno	Pone el objeto en la localidad dada por el flag.
WHATO	5			Convierte Nombre1(Adjetivo1) en objeto actual.
WEIGH	5	objno	flagno	Peso de los objetos es almacenado en flag.
SET	5	flagno		Da al flag el valor 255.

CLEAR	5	flagno		Da al flag el valor 0.
PLUS	5	flagno	0-255	Sumar valor a flag.
MINUS	5	flagno	0-255	Restar valor a flag.
LET	5	flagno	0-255	Da al flag el valor indicado.
ADD	5	flagno1	flagno2	Los contenidos de flag 1 son sumados a flag 2.
SUB	5	flagno1	flagno2	Los contenidos de flag 1 son restados de flag2.
COPYFF	5	flagno1	flagno2	Da al flag 2 el valor del flag 1.
RANDOM	5	flagno		Poner a un valor aleatorio de 0 a 99.
MOVE	5	flagno		Ajusta el valor del flag de acuerdo al verbo de la SL y las conexiones de la localidad (permite movimientos de PSIs).
GOTO	5	locno		Mueve al jugador a la localidad indicada.
WEIGHT	5	flagno		El peso de los objetos llevados y puestos es puesto en el flag.
ABILITY	5	0-255	0-255	Indica el peso y máximo numero de objetos del jugador.
MODE	5	0-255		Controla si se limpia la pantalla antes de describir las localidades: con 0 limpia, con otro valor no.
PROMPT	5	sysno		Texto en el input, 0 es aleatorio.
TIME	5	0-255	0-255	Controla las opciones y el tiempo del temporizador.
PRINT	5	flagno		Muestra valor del flag.
URNS	5			Muestra los turnos consumidos por el jugador.
SCORE	5			Muestra la puntuación obtenida por el jugador.
CLS	5			Limpia la pantalla de texto.
NEWLINE	5			Imprime un retorno de carro.
MES	5	mesno		Mensaje sin nueva linea.
MESSAGE	5	mesno		Mensaje con nueva linea.
SYSMESS	5	sysno		Mensaje del sistema sin nueva linea.
LISTOBJ	5			Lista los objetos de la localidad actual.

LISTAT	5	locno+		Lista objetos en una localidad especificada.
INVEN	3			Imprime el inventario del jugador.
DESC	1			Imprime la descripción de la localidad actual.
END	2			Sale de tabla para reiniciar el juego.
DONE	3			Sale de la tabla indicando que se ha hecho algo y que ya se ha terminado el procesamiento de la orden.
NOTDONE	3			Sale de la tabla indicando que no se ha terminado el procesamiento de la orden.
OK	3			Imprime VALE y ejecuta un DONE.
SAVE	1			Guarda la partida en disco.
LOAD	1			Carga la partida de disco.
RAMSAVE	5			Graba la partida en memoria.
RAMLOAD	5	flagno		Carga la partida de la memoria.
ANYKEY	5			Espera la pulsación de una tecla.
PAUSE	5	0-255		Pausa el programa por n/50 segundos.
PARSE	5			Convierte las ordenes del jugador en una SL valida.
NEWTEXT	5			Fuerza la perdida de las frases que queden del jugador.
PROCESS	5	prono		Ejecuta sub-procesos.
DOALL	5	locno+		Genera Nombre y adjetivo para cada objeto de la localidad.
RESET	1	locno		Mueve el jugador y los objetos presentes, reinicia los demás a su posición de inicio. Usar para encadenar juegos con LOAD.
TRANSCRIPT	5	0-255		Activa o desactiva la transcripción de la partida a disco.
EXTERN	5	0-255		Ejecuta un comando externo (no implementado).
PAPER	5	0-255		Pone el color de fondo de la ventana gráfica.
CHARSET	5	0-255		Cambia el estilo del texto.

PICTURE	5	0-255		Muestra un gráfico.
GRAPHIC	5	0-255		Activa o desactiva los gráficos.
BEEP	5	0-255	0-255	Hace sonar un efecto.
SOUND	5	0-255		Activa o desactiva el sonido.

Tabla 26- resumen de las acciones