

# THE PROFESSIONAL ADVENTURE WRITING SYSTEM

## GUÍA TÉCNICA

### P A W S SISTEMA PROFESIONAL PARA LA CREACION DE AVENTURAS GRAFICAS

Versión en Castellano

© programa: 1986/89 Gilsoft International Ltd.  
© versión castellana: Aventuras A. D., S. A.

DERECHOS DE TRADUCCION, ADAPTACION Y REPRODUCCION RESERVADOS  
Prohibida la reproducción total o parcial del sistema

Los juegos producidos con PAW, quedan libres de toda restricción,  
agradeciéndose se cite la fuente.

#### **AGRADECIMIENTOS**

Este proyecto, labor de 4 meses, nunca se hubiese podido hacer sin la ayuda desinteresada de:

Manolo, por la Demo

Alicia, por los dibujos

Carlos, por arreglillos finales

Juanjo, por la revisión ortográfica

Tim, por ser el padre de la criatura original

Eva, por pasarlo al editor de textos, impresión y archivos.

A todos ellos, miembros del equipo de Aventuras A. D., hago llegar mi agradecimiento.

en Valencia, a inicios de 1989  
Andrés.

Amigo aventurero que has adquirido este Creador:

1. agradecemos tu confianza en nosotros.
2. te enviamos tu número de código: 840734

Indispensable para hacernos cualquier consulta técnica sobre el uso del PAW o sobre dudas al planear tu aventura y, lo más importante, para distinguirte de esa plaga moderna: *los piratas*.

3. te invitamos a reflexionar sobre lo que te ha costado el programa, su gran potencial y los beneficios que puedes obtener con él. Por cada copia ilegal que tú facilites, tendrás un competidor más (le ha salido gratis y en el fondo se reirá de ti).
4. suerte, si participas en nuestro concurso, envíanos buenas aventuras.

EQUIPO DE AVENTURAS A. D., S. A

## INDICE

Introducción	Pag. 5
El intérprete	Pag. 7
1. Inicialización	Pag. 7
2. Descripción de localidad actual	Pag. 7
3. Tabla de procesos 1	Pag. 8
4. Tabla de procesos 2	Pag. 8
5. Coger la frase	Pag. 8
6. Tabla de respuestas	Pag. 9
7. Tabla de conexiones	Pag. 10
Cómo se busca en una tabla de procesos	Pag. 13
Los conductos	Pag. 15
Condiciones	Pag. 15
Acciones	Pag. 20
Descripción detallada de la base de datos	Pag. 48
Descripción detallada del Editor	Pag. 56
Descripción detallada del Editor Gráfico	Pag. 74
Errores y su significado	Pag. 80
PAW y las impresoras	Pag. 83
Los Ensayos	Pag. 85
1. Consideraciones para los 128K	Pag. 85
2. Las fronteras de Color	Pag. 86
3. El Parser	Pag. 89
4. La Pantalla	Pag. 91
5. Objetos	Pag. 92
6. Aventuras múltiples	Pag. 94
7. Luz y Oscuridad	Pag. 96
8. Comando EXTERN y carga de Pantallas	Pag. 98
9. PSI (personajes seudo inteligentes)	Pag. 99
Sumario	Pag. 104
1. Órdenes del Editor Gráfico	Pag. 104
2. Mensajes del Sistema	Pag. 105
3. Funciones de las Banderas	Pag. 107
4. Los conductos	Pag. 111

### INTRODUCCION

PAW puede ser dividido en tres partes principales:

#### **1/ La Base de Datos**

La base de datos es una colección de tablas organizadas de manera lógica, que contienen toda la información para definir un juego de aventuras.

Esta base de datos está cargada inicialmente con un mínimo de entradas que probablemente se necesitarán en cualquier juego.

Normalmente hay un entrada en cada tabla para simplificar la forma en que opera el sistema.

La base de datos puede utilizar páginas de RAM adicionales en un Spectrum de 128K.

#### **2/ El Editor**

Consiste en un sistema de menús en dos niveles (esto quiere decir que la mayoría de las opciones del menú principal presentarán un submenú del cual se pueden seleccionar otras opciones).

Todas las órdenes pueden entrar, o bien desde una línea de mando como parte de un submenú, o como respuesta a una pregunta del ordenador.

Esta combinación de menú y línea de mando la consideramos la más apropiada y eficiente tanto para el principiante como para el veterano.

### **3/ El Intérprete**

Es el corazón real de PAW. En el siguiente capítulo trataremos de él con más detalle, pero debes saber lo que hace de una manera general: *recibe órdenes del jugador y usa la información contenida en la base de datos para decodificarla y responder a ellas.*

Por el tamaño de PAW es imposible tenerlo enteramente en la memoria.

En un 128K la mayoría del código se sitúa como overlays en otra página de RAM y es transferida al área principal cuando es requerida.

Un sistema similar se usa en el 48K, en el cual se mantienen tres de los overlays en memoria libre hasta que el área de memoria se sobrescribe.

Si en un 128K la página extra de memoria es inicializada, o en 48K la zona de memoria sobrescrita, cualquier intento de seleccionar una opción que no esté actualmente en el overlay usado, tendrá como resultado que PAW intentará cargar el overlay requerido de cassette o de disco.

El menú principal y los sistemas de overlay ocupan aproximadamente 4K, que no pueden ser usadas por la base de datos de PAW.

El intérprete consume unas 10K, lo cual nos deja aproximadamente 27K para la base de datos, más 5 páginas de 16K cada uno en un Spectrum de 128K (un total de más de 117K).

A esto se le puede sacar aún más partido con la inclusión de un compresor de texto que ocupará unos 222 bytes de memoria y que hace una compresión de más o menos el 40% de efectividad.

En un juego sin gráficos, en un Spectrum de 48K esto nos dará una suma de 8K más de espacio para juego (es decir más o menos 35K en un 48K).

## **EL INTERPRETE**

La descripción siguiente del Intérprete debe ser leída al mismo tiempo que se estudia la tabla de flujo de las páginas.

### **1-LA INICIALIZACION DEL SISTEMA**

Los colores de fondo y el set de caracteres son seleccionados. La pantalla no se limpia puesto que siempre ocurrirá al describir la localidad 0.

Las banderas son todas puestas a 0 excepto:

- La bandera 37, contiene el número de objetos llevados, que se pondrá en 4.
- La bandera 52, lleva el máximo peso permitido, que se pone a 10. Las banderas 46 y 47 que llevan el pronombre actual, que se pondrán en 255 (no hay pronombre).
- La bandera 1, que lleva el número de objetos llevados pero no usados.

Hay que tener en cuenta que con la limpieza de banderas, el juego siempre comenzará en la localidad 0. Esto es porque la bandera 38 que lleva a la localización actual del jugador está ahora en cero.

### **2-DESCRIPCION DE LA LOCALIDAD ACTUAL**

Si la bandera 2 no está a cero será decrementada en 1.

Si está oscuro (que la bandera 0 no es cero) y la bandera 3 no es cero entonces la bandera 3 es decrementada. Si está oscuro y la bandera 4 no está a cero y el objeto 0 (la fuente de luz) está ausente, la bandera 4 es decrementada.

Se hace una limpieza de pantalla (clear) si el contenido del modo de pantalla (el que se contiene en la bandera 40) no está a 1.

Si está oscuro y el objeto 0 está ausente, entonces el Mensaje del Sistema 0 (el que se refiere al mensaje "está muy oscuro para ver") se imprime.

Si no, cualquier gráfico presente para la localidad es dibujado y la descripción de la localidad en texto aparecerá sin hacer un NEWLINE.

### **3-BUSQUEDA DE LA TABLA DE PROCESOS 1**

La tabla de flujo número 2 y la próxima sección describirán cómo se busca a través de una tabla de Procesos.

La tabla de Procesos número 1 se utiliza principalmente para que contenga aquellas entradas que van a añadir información extra a la descripción actual de la localidad.

Por ejemplo, detalles sobre puertas abiertas, objetos presentes, etc. También contendrá cualquier acción PROTECT cuando se use el modo de pantalla número 4, para señalar el número de la línea por debajo de la cual hará un scroll el texto.

Ahora entraremos en el bucle principal del Intérprete, que actuará en cada paso de tiempo (siempre que se haya extraído una frase del jugador o se haya alcanzado el tiempo válido de un timeout) y también con las respuestas a las órdenes del jugador.

### **4-BUSQUEDA DE LA TABLA DE PROCESOS 2**

Esta tabla contiene el control principal del turno de PAW durante el juego. Se usa para implementar los movimientos y las acciones de los PSI's. También se usa en los eventos incontrolados, como son puentes que se derrumban, cosas que caen, etc.

### **5-COGER LA FRASE**

Si las banderas 7 a la 8 no están a cero son decrementadas. Si está oscuro (la bandera 0 no está a cero) y la bandera 9 no está a cero, entonces también es decrementada. Si está oscuro y la bandera 10 no está a cero, será decrementada si el objeto 0 está ausente.

Ahora el parser va a extraer una frase y convertirla en la sentencia lógica (SL).

Si el buffer de input esta vacío, se busca un nuevo input del jugador por medio de la impresión de la frase adecuada y una posterior llamada a la rutina de input. El mensaje será el que contenga la bandera 42. Si tiene un valor igual a 0 los mensajes 2, 3, 4 y 5 serán seleccionados con una frecuencia de 30:30:30:10 respectivamente.

Si se selecciona la opción tiempo muerto (timeout), haciendo que la bandera 48 tenga un valor más grande que 0, entonces se ejecutará durante la rutina de input. En este caso el mensaje del sistema MS35 "El tiempo pasa..." aparece y se vuelve otra vez a la búsqueda en la tabla de Procesos 2.

Una frase es sacada y convertida en una sentencia lógica *por medio de la conversión de cualquier palabra en ella presente, que esté en el vocabulario, a su número de palabra y poniéndola luego en la bandera requerida.*

Si no se encuentra una frase válida, entonces el mensaje del sistema MS6 "No entiendo" aparece y vuelve a buscar al Proceso 2.

### **6- BUSQUEDA EN LA TABLA DE RESPUESTAS**

El turno es llevado por las banderas 31 y 32, y es el número de frases válidas que han sido sacadas por el parser. Este turno entonces se aumentará en 1.

Se usan dos banderas para permitir un total de 65.536 turnos, 256 por 256, (si el jugador no ha resuelto el juego para entonces, es porque está tarado).

En la tabla de Respuestas se busca una entrada que haga juego con el Verbo y el primer Nombre de la Sentencia Lógica actual.

Si es positivo se lleva a cabo la acción (excepto si la acción es NOTDONE) y luego se vuelve a la búsqueda en la tabla de Procesos 2.

## 7-BUSQUEDA. EN LA TABLA DE CONEXIONES

Es importante destacar que la tabla de conexiones (al revés que ocurría en el QUILL) se ejecutará *después de haber visto las tablas de Procesos o las tablas de Respuestas*. Por ello, cualquier conexión que hayas puesto como un Proceso o como una tabla de Respuestas puede ser omitida en las conexiones.

En la tabla de conexiones hay una búsqueda para la localidad actual de una palabra que haga juego con el verbo presente en la sentencia lógica actual.

Si se encuentra una, entonces la localidad del jugador (llevada por la bandera 38) se pone para que sea el número que esté asociado con esa palabra. Luego se ejecuta la descripción de esa localidad.

Si no encuentra la palabra, PAW imprimirá el Mensaje del Sistema MS8 "No puedo ir en esa dirección", si el Verbo que se busca tiene un valor menor de 14, o imprimirá el mensaje del sistema MS7, "No puedo hacer eso", si es superior.

De cualquier manera vuelve a hacer una búsqueda en la tabla de Procesos 2.

Tabla de flujo 1

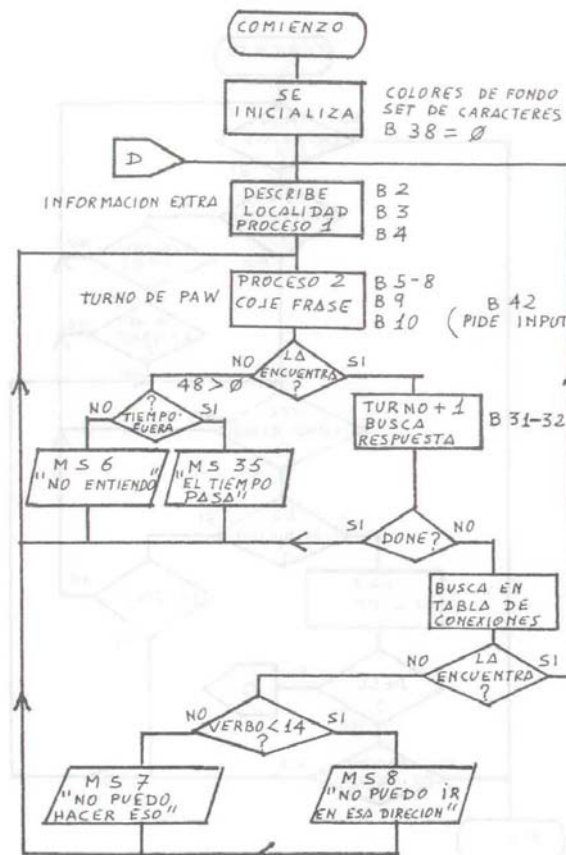
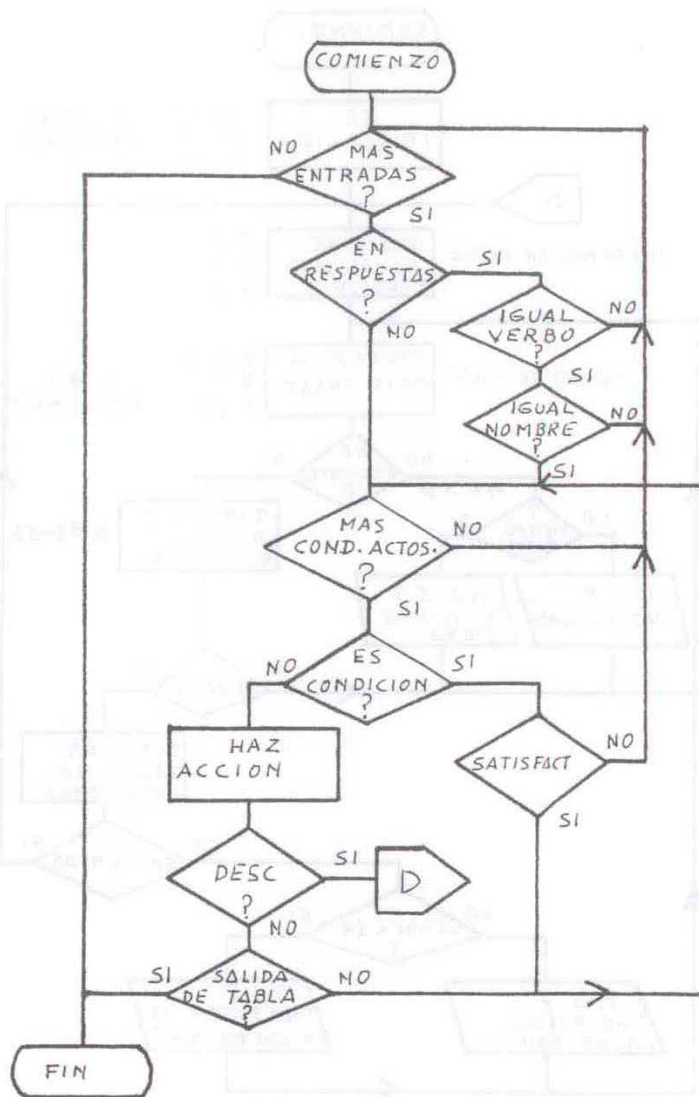


Tabla de flujo 2



## BUSQUEDA EN UNA TABLA DE PROCESOS

La tabla de Respuesta es una tabla de Procesos (y cualquier subproceso que se llame desde la tabla de Respuestas recibirá el mismo tratamiento), por lo cual siempre será buscada en esta sección de PAW. La diferencia es, por supuesto, que PAW intenta buscar una similitud entre el Verbo y el Nombre 1 de la SL y la tabla de Respuestas.

Pero esencialmente, lo que PAW hace es mirar una por una cada entrada de la tabla, hasta que se acabe (no PAW sino la tabla). Si encuentra una entrada en la tabla de Respuestas se asegurará de que el Verbo y el Nombre sean los mismos a los de la SL.

Por ello el uso de palabras como "\*" y "\_" bien sea como Verbo o como Nombre, hará que sea positiva esa palabra para cualquier parte de la SL. Por lo tanto, una entrada en la tabla de Respuestas como "\* \*", "\_ " o cualquier combinación de las mismas, disparará cualquier entrada, no importa lo que se haya tecleado.

Este hecho debe ser también usado en cualquier tabla que vaya a ser llamada, bien sea por una tabla de Respuestas o una tabla de Procesos. De todos modos, en una tabla de Procesos, el Verbo y el Nombre serán similares siempre a lo que teclees en la sentencia lógica.

PAW entonces va a examinar cada uno de los conductos. Si es una condición válida PAW buscará el siguiente conducto. Si no, se saldrá de la actual lista de conductos y buscará otra entrada en la tabla (una excepción de esta regla es el conducto QUIT, el cual, si falla, se saldrá completamente de la tabla). Esto no lo mostramos en la tabla de flujo 2 para una mayor claridad.

Si se trata de una acción, entonces se llevará a cabo.

Las acciones pueden ser divididas en cinco grupos principales:

1/ DESC; que causará una salida completa de todas las tablas (aunque esté en un subproceso de nivel 10) y saltará a describir la localidad actual.

2/ END; (es ella sola un grupo) y causará la salida total de todas las tablas y un salto a inicializar un nuevo juego.

3/ EHT (SALIDA); es cualquier acción que interrumpe el procesamiento de la tabla actual y causa su salida hacia la tabla de llamada (o devuelta al bucle principal si estamos en una tabla de Respuestas o de Procesos, 1 y 2) por ejemplo: INVEN, DONE, etc.

4/ SALIDA CONDICIONAL; es cualquier acción que interrumpa el procesamiento de la tabla actual y cause su salida a la tabla de llamada (o hacia el bucle principal si estamos en una tabla de Respuestas o de Procesos 1 y 2) pero *sólo* si falla de ejecutar la función requerida. Por ejemplo: GET, PUTIN, etc. Si no, se continuará con el siguiente Conducto.

5/ NORMAL; es cualquier acción que ejecuta su función, y permite a PAW continuar buscando en el siguiente conducto en la entrada actual. Por ejemplo: COFYFF, PLUS, etc.

Parecería que la acción QUIT actuaría como una de tipo 4, *pero* todavía tenemos que tener en cuenta que es una condición. Por lo tanto, es un Conducto.

Al final de este librito haremos un sumario de qué acción pertenece a cada tipo para una consulta rápida.

## LOS CONDUCTOS

Ahora seguirá una descripción detallada de cada conducto. Están divididos en grupos, de acuerdo con el tema del que se encargan dentro del PAW; como banderas, objetos, etc. También damos algunas claves sobre su posible uso.

En las descripciones se usan varias abreviaturas que es importante conocer:

**locno.** es el número de una localidad válida.

**locno+** es un número válido de localidad:

252 (no creado)

253 (llevado puesto)

254 (llevado)

255, el cual se va a convertir en la localidad actual del jugador.

**mesno.** es un mensaje válido.

**sisno.** es un Mensaje del Sistema válido.

flagno, es cualquier bandera (de 0 a 255).

**Procno.** es un número válido de subproceso.

**palabra** es una palabra del tipo requerido que está presente en el vocabulario, o “ \_ “ que ahora significa **no palabra** (que no significa cualquier palabra como hasta ahora).

**valor** es un valor de 0 a 255.

## CONDICIONES

Hay cuatro condiciones que tratan de *probar la localidad del jugador*:

**AT locno.**

Satisfactorio si la localidad actual es la misma que pone en locno.

**NOTAT locno.**

Satisfactorio si la localidad actual es diferente a lo que pone en locno.

**ATGT locno.**

Satisfactorio si la localidad actual es mayor que locno.

**ATLT locno.**

Satisfactorio si la localidad actual es menor que locno.

Hay ocho condiciones que tratan de la localidad actual de los objetos:

**PRESENT objno.**

Satisfactorio si el objeto de que trata objno está siendo llevado, usado o en la localidad actual.

**ABSENT objno.**

Satisfactorio si el objeto que trata objno no está siendo llevado, ni está puesto, ni está en la localidad actual.

**WORN objno.**

Satisfactorio si el objeto de que trata objno está siendo llevado puesto.

**NOTWORN objno.**

Satisfactorio si el objeto que trata objno no es llevado puesto.

**CARRIED objno.**

Satisfactorio si el objeto que trata objno se lleva.

**NOTCARR objno.**

Satisfactorio si el objeto que trata objno no se lleva.

**ISAT objno. Locno+**

Satisfactorio si el objeto de que trata objno está en el locno de la localidad.

**ISNOTAT objno. Locno+**

Satisfactorio si el objeto de que trata objno no está en el locno de la localidad.

Hay ocho condiciones que tratan de los valores y de la comparación de banderas:



**ZERO flagno.**

Satisfactorio si el número de la bandera indicado por flagno es igual a cero.

**NOTZERO flagno.**

Satisfactorio si el número de la bandera indicado por flagno no es igual a cero.

**EQ flagno. valor**

Satisfactorio si la bandera indicada por flagno tiene un valor igual al indicado.

**NOTEQ flagno. valor**

Satisfactorio si el número de la bandera indicada por flagno no es igual al valor.

**GT flagno. valor**

Satisfactorio si el número de la bandera indicado por flagno es mayor que el valor.

**LT flagno. valor**

Satisfactorio si el número de la bandera indicada por flagno es menor que el valor.

**SAME flagno1 flagno2**

Satisfactorio si el valor de la bandera indicado por flagno1 tiene el mismo valor que el valor de la bandera indicado por flagno2.

**NOTSAME flagno1 flagno2**

Satisfactorio si el valor de la bandera indicado por flagno1 no tiene el mismo valor que el valor de la bandera indicado por flagno2.

Hay cinco condiciones para iniciar una búsqueda en una SL (sentencia lógica).

Es mejor usar estas condiciones solamente cuando se necesita una palabra específica (o la ausencia de palabra si usamos "\_") para diferenciar claramente una situación. Esto permite una mayor flexibilidad de las órdenes entendidas por medio de la entrada.

**ADJECT1 palabra**

Satisfactorio si el adjetivo del primer Nombre encontrado en la SL actual es igual a palabra.

**ADVERB palabra**

Satisfactorio si el adverbio de la SL actual es igual a palabra.

**PREP palabra**

Satisfactorio si la preposición en la SL actual es igual a palabra.

**NOUN2 palabra**

Satisfactorio si el segundo Nombre de la SL actual es igual a palabra.

**ADJECT2 palabra**

Satisfactorio si el adjetivo del segundo Nombre en la SL actual es igual a palabra.

La siguiente condición es utilizada para efectos al azar. Se puede usar para que las cosas ocurran como si fuera por casualidad.

Por ejemplo, la caída de un árbol sobre el jugador durante una tormenta, o un puente que se derrumba, etc. Recomendamos no abusar de esta facilidad y dar siempre al jugador un aviso que le ayude a prevenir el problema y a solucionarlo.

## **CHANCE porcentaje**

Satisfactorio si el porcentaje es menor o es igual a un número aleatorio entre 1 y 100 (ambos inclusive). Ello significa que una condición con CHANCE 50 permitirá a PAW seguir buscando en el próximo contacto solamente si el número aleatorio generado era entre 1 y 50, o sea, un porcentaje del 50% de que salga satisfactorio.

Una condición para probar el "tiempo muerto" o "fuera de tiempo".

## **TIMEOUT**

Será válida si la última petición de INPUT para el jugador fue permitida en fuera de tiempo, es decir, sin relación con lo que teclee.

Por ejemplo, una entrada en la tabla de Procesos 2 que diga TIMEOUT MESSAGE 0 (donde el mensaje 0 diga "Venga, despierta") podría ser creada, quizá con una CHANCE o una cuenta de "fuera de tiempo" para hacerla menos monótona.

Hay un verdadero contacto:

## **QUIT**

EL mensaje del sistema número 12 "¿Estás seguro?" se imprime, y la rutina de INPUT es llamada. Será válido si el jugador responde con la primera letra del Mensaje del Sistema 30 ("S") a esta petición. Si no, entonces las acciones NEWTEXT y DONE son ejecutadas.

## **ACCIONES**

Hay diecinueve acciones que se encargan de la manipulación de las posiciones de los objetos. Fijaros que digo de las posiciones, no de los objetos:

### **GET objno.**

1. Si el objeto de que trata el objno es llevado o está puesto encima, el Mensaje del Sistema número 25 "Ya tengo el \_\_\_" se imprime y se hacen las acciones NEWTEXT y DONE.
2. Si el objeto al que se refiere objno no está en la localidad actual, el mensaje del sistema número 26 "No hay uno de esos aquí" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el peso total de los objetos llevados o llevados puestos encima por el jugador, más el objeto a que se refiere objno pasa del peso máximo permitido (el que lleva la bandera 52), entonces el Mensaje del Sistema 43 ("El o la \_ pesa mucho para mí") se imprime y las acciones NEWTEXT y DONE son ejecutadas.
4. Si el número máximo de objetos ya se lleva (la bandera 1 es mayor o del mismo valor que la bandera 37) el Mensaje del Sistema número 27 "Ya no puedo llevar más cosas" se imprimirá y las acciones NEWTEXT y DONE se ejecutan. Además, cualquier bucle DOALL es cancelado.
5. Si no se da ninguno de los anteriores, la posición del objeto marcado como objno se cambia a llevado, la bandera 1 se incrementa y el Mensaje del Sistema 36 "Ahora tengo \_" se imprime.

### **DROP objno.**

1. Si el objeto a que se refiere objno se lleva puesto encima, entonces el Mensaje del Sistema 24 "No puedo. Llevo puesto \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
2. Si el objeto a que se refiere objno está en la localidad actual (pero ni se lleva puesto encima, ni es llevado), el Mensaje del Sistema 49 "No tengo \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el objeto a que se refiere objno no está en la localidad actual, entonces el Mensaje del Sistema 28 "No tengo eso" se imprime y las acciones NEWTEXT y DONE se ejecutan.
4. Si no se dan los casos anteriores, la posición del objeto al que se refiere objno se cambia a la localidad actual. La bandera 1 se decrementa y el Mensaje del Sistema 39, "He dejado caer \_" se imprime.

### **WEAR objno.**

1. Si el objeto al que se refiere objno está en la localidad actual (pero no se lleva, ni está puesto encima) el Mensaje del Sistema 49 "No tengo \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
2. Si el objeto al que se refiere objno es llevado puesto encima, el Mensaje del Sistema 29 "Ya llevo puesto \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el objeto al que se refiere objno no se lleva, el mensaje del sistema número 28 "No tengo uno de esos" se imprime y las acciones NEWTEXT y DONE se ejecutan.
4. Si el objeto al que se refiere objno no se ha marcado por medio de una bandera como "usable" (con la opción WR en el menú Pesos de Objetos), entonces el Mensaje del Sistema número 40 "No puedo ponerme \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
5. Si no se dan ninguno de los casos anteriores, la posición del objeto a que se refiere objno se cambia a "llevado puesto encima", la bandera 1 se decrementa y el Mensaje del Sistema 37 "Ahora llevo puesto \_" se imprime.

### **REMOVE objno.**

1. Si el objeto al que se refiere objno se lleva o está en la localidad actual (pero no se lleva puesto encima) entonces el Mensaje del Sistema número 50 "No llevo puesto" se imprime y las acciones NEWTEXT y DONE se ejecutan.
2. Si el objeto al que se refiere objno no está en la localidad actual, el Mensaje del Sistema número 23 "No llevo puesto uno de esos" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el objeto al que se refiere objno no está puesto en una bandera como objeto llevable encima (y por lo tanto como objeto quitable), entonces el Mensaje del Sistema número 41 "No me puedo quitar \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
4. Si ya se lleva el máximo de objetos (y la bandera 1 es mayor, o lo mismo que la bandera 37) el Mensaje del Sistema 42 "No puedo quitarme \_. Mis manos están llenas" se imprime y las acciones NEWTEXT y DONE se ejecutan.
5. Si ninguno de los anteriores se da, la posición del objeto que es indicado por objno se cambia a llevado, la bandera 1 se incrementa y el Mensaje del Sistema 38 "Me he quitado \_" se imprime.

### **CREATE objno.**

La posición del objeto a que se refiere objno se cambia a la posición de la localidad actual, la bandera 1 se decrementará si el objeto es llevado por el jugador.

### **DESTROY objno.**

La posición del objeto a que se refiere objno se cambia a "no creado" y la bandera 1 se decrementa si el objeto es llevado por el jugador.

### **SWAP objno1 objno2**

Las posiciones de dos objetos son cambiadas. La bandera 1 no se toca para nada. El objeto al cual se refiere actualmente, es decir, el último referido y que PAW conserva en memoria en la bandera 51, se sitúa en el objno número 2 (el que está marcado en la bandera 51).

### **PLACE objno. locno+**

La posición a la que se refiere el objno se cambia a la localidad a la que se refiere locno. La bandera 1 se decrementa si era un objeto llevado. Y se incrementa si el objeto se pone en la localidad 254 (llevado).

### **PUTO locno+**

La posición del objeto al que se refiere actualmente (el objeto a cuyo número esté en la bandera 51) se cambia para ser el número de localidad de locno. La bandera 54 permanece como su vieja localidad. La bandera 1 se decrementa si el objeto era llevado, y se incrementa si el objeto se pone en la localidad 254 (llevado).

### **PUTIN objno. locno.**

1. Si el objeto con número de locno. no existe o no está marcado como un contenedor (opción C en el menú de Objetos Peso) entonces se genera un error de "Argumento Ilegal".
2. Si el objeto al que se refiere objno. es llevado puesto encima, entonces el Mensaje del Sistema 24 "No puedo, llevo puesto \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el objeto al que se refiere objno. está en la localidad actual (pero ni se lleva puesto encima, ni se lleva) el Mensaje del Sistema número 49 "No tengo \_" se imprime, y las acciones NEWTEXT y DONE se ejecutan.
4. Si el objeto al que se refiere objno. no está en la localidad actual, pero tampoco está llevado, entonces un Mensaje del Sistema número 28 "No tengo uno de esos" se imprime, y las acciones NEWTEXT y DONE se ejecutan.
5. De cualquier otra forma, la posición del objeto al que se refiere objno. se cambia a la localidad a la que se refiere locno. La bandera 1 se decrementa y el Mensaje del Sistema 44 "El \_ está en la", viene entonces una descripción de la localidad del objeto y luego el Mensaje del Sistema 51 ". ", se imprime.

### **TAKEOUT objno. locno.**

1. Si el objeto al que se refiere locno. no existe o no está marcado como un contenedor (la opción C en el menú de objetos y pesos) entonces se genera un error de "Argumento Ilegal".
2. Si el objeto al que se refiere objno. se lleva puesto encima o es llevado, el Mensaje del Sistema número 25 "Ya tengo \_" se imprime y las acciones NEWTEXT y DONE se ejecutan.
3. Si el objeto al que se refiere objno. está en la localidad actual, el Mensaje del Sistema número 45 "\_ no está en la", viene una descripción del objeto que va a ser usado como localidad, y

luego el Mensaje del Sistema 51 "." se imprime y las acciones NEWTEXT y DONE se ejecutan.

4. Si el objeto al que se refiere objno. no está en la localidad actual y no está en la localidad locno., entonces el Mensaje del Sistema 52 "No hay uno de esos en", viene una descripción de el objeto localidad y luego el Mensaje del Sistema 51 se imprime y las acciones NEWTEXT y DONE se ejecutan.
5. Si el objeto al que se refiere locno. no se lleva o no está puesto y el peso total de los objetos llevados por el jugador, más el objeto al que se refiere objno. excede del peso máximo llevable (bandera 52), entonces el Mensaje del Sistema 46 "El \_ pesa mucho para mí" se imprime y las acciones NEWTEXT y DONE se ejecutan.
6. Si el máximo número de objetos ya se lleva (o sea, que la bandera 1 es mayor o igual que la bandera 37), el Mensaje del Sistema 27 "No puedo llevar más cosas" se imprime y las acciones NEWTEXT y DONE se ejecutan. En adición, cualquier bucle DOALL se cancela.
7. De otro modo, la posición del objeto al que se refiere objno. cambia a llevado, la bandera 1 se incrementa, y el Mensaje del Sistema 36 "Ahora tengo" se imprime.

**Nota Importante:** No se hace ningún chequeo ni para PUTIN ni para TAKEOUT de qué objeto utilizado cano locno. esté actualmente presente. Esto debes hacerlo tú si es necesario.

## **DROPALL**

Todos los objetos que sean llevados o puestos encima se crearán en la localidad actual (lo que dará el efecto de que todos los objetos han sido dejados caer) la bandera 1 se pondrá a 0. Este comando se ha incluido para compatibilidad con los antiguos sistemas, (léase QUILL). Es de notar que un DOALL 254 llevará a cabo un verdadero DROP ALL, y además tendrá en cuenta cualquier otra acción especial que se incluya.

Las siguientes seis acciones tratan de versiones *automáticas* como COGER, DEJAR, LLEVAR PUESTOS, QUITAR DE ENCIMA, PONER DENTRO y SACAR DE, (GET, DROP, WEAR, REMOVE, PUTIN and TAKEOUT). Son automáticos por el hecho de que en vez de necesitar que se especifique el número del objeto, ellos convertirán el primer Nombre con su Adjetivo en el objeto actualmente indicado o referenciado, buscándolo en la tabla de Objetos Palabras.

La búsqueda se hace por un objeto que esté en una de las siguientes localidades *en orden descendente* de prioridad. (Mirar las descripciones individuales).

Esta búsqueda con prioridades permite que PAW "sepa" de qué objeto se trata, si más de un objeto tiene el mismo nombre descriptivo (cuando el jugador no haya especificado un Adjetivo), en la localidad actual llevado o puesto encima o en un contenedor (si se trata del comando TAKEOUT).

## **AUTOG**

Una búsqueda del número del objeto representado por Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabra según su prioridad de localización que será: aquí, llevado, puesto, etc.

Es más lógico que el jugador esté tratando de coger un objeto que esté en la localidad actual, que uno que lleve o tenga puesto encima, por eso el orden de prioridades.

Si el objeto es encontrado, su número se pasa a la acción GET. De otra forma, si es un objeto que existe en cualquier otra parte del juego o si el Nombre1 no está en el Vocabulario, el Mensaje del Sistema número 26 "No hay uno de esos aquí" se imprime. Si no, el Mensaje del Sistema número 8 "No puedo hacer eso" se imprime (significa que no es un objeto válido pero que existe en el juego). De todas formas las acciones NEWTEXT y DONE se ejecutan.

## **AUTOD**

Una búsqueda por el número del objeto representado por Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabras según la prioridad de localidad: llevado, puesto encima, aquí.

Por ejemplo, lo más probable es que el jugador esté intentando dejar caer (DROP) un objeto que lleve, que lleve puesto encima o esté aquí.

Si se encuentra un objeto, su número se pasa a la acción DROP. De otra forma si hay un objeto que existe en cualquier parte del juego, o si el Nombre1 no está en el Vocabulario, el Mensaje del Sistema 28 "No tengo uno de esos" se imprime. Si no, el Mensaje número 8 "No puedo hacer eso" se imprime, es decir (no es un objeto válido pero sí que existe en el juego). De cualquier forma las acciones NEWTEXT y DONE son ejecutadas.

### **AUTOW**

En una búsqueda del número del objeto representado por Nombre (Adjetivo) 1 se hace en la tabla de objetos en orden de prioridad de: llevado, puesto encima, aquí.

Es más probable que el jugador esté intentando ponerse encima (WEAR) un objeto que lleve en sus manos, que uno que ya lleve puesto o esté aquí.

Si el objeto se encuentra, su número se pasa a la acción WEAR. De otra manera, si hay un objeto en existencia en cualquier parte del juego o si el Nombre1 no estaba en el Vocabulario, entonces el Mensaje del Sistema número 28 "No tengo uno de esos" se imprime.

Si no, el Mensaje del Sistema número 8 "No puedo hacer eso" se imprime. No es un objeto válido, pero sí existe en el juego. De cualquier modo, las acciones NEWTEXT y DONE se ejecutan.

### **AUTOR**

Una búsqueda por el número del objeto representado por el Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabra con una prioridad de: puesto encima, llevado, aquí.

El jugador, lo más probable es que esté tratando de quitarse (REMOVE) un objeto que lleve puesto, que uno que lleve en sus manos o que esté aquí.

Si un objeto se encuentra, su número se pasa a la acción REMOVE. Si hay un objeto en existencia en cualquier parte del juego o si el Nombre1 no estaba en el Vocabulario, entonces el Mensaje del Sistema 23 "No llevo uno de esos" se imprime. Si no, el Mensaje del Sistema número 8 "No puedo hacer eso" se imprime (no es un objeto válido pero existe en el juego). De cualquier forma, las acciones NEWTEXT y DONE se ejecutan.

### **AUTOP locno.**

Una búsqueda del número del objeto representado por el Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabras en el siguiente orden de prioridad: llevado, puesto, aquí.

Es más factible que el jugador esté intentando poner un objeto que lleva en sus manos dentro de otro, que uno que lleva puesto encima o está aquí.

Si el objeto se encuentra, su número se pasa a la acción PUTIN. Si hay un objeto en existencia en cualquier parte del juego o si el Nombre1 no está en el Vocabulario, entonces el Mensaje del Sistema 28 "No tengo uno de esos" se imprime.

Si no, el Mensaje del Sistema número 8 "No puedo hacer eso" se imprime. No es un objeto válido, pero existe en el juego. De todos modos, las acciones NEWTEXT y DONE son ejecutadas.

### **AUTOT locno.**

Una búsqueda por el número del objeto representado por el Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabras según la siguiente prioridad: un contenedor, llevado, puesto, aquí.

El jugador es más probable que esté intentando sacar un objeto que esté dentro de un contenedor que un objeto que sea llevado, puesto encima, o aquí.

Si un objeto se encuentra, su número se pasa a la acción TAKEOUT.

Si hay un objeto en existencia en cualquier parte del juego o si el Nombre1 no está en el Vocabulario entonces el Mensaje del Sistema "No hay uno de esos en él", viene luego una descripción del objeto que sirve de localidad, y luego el Mensaje 51 del sistema "." se imprime.

Si no, el Mensaje del Sistema 8 "No puedo hacer eso" se imprime. (No es un objeto válido pero existe en el juego). De cualquier modo, las acciones NEWTEXT y DONE se ejecutan.

**Nota Importante:** No se hace ningún chequeo, ni por AUTOP ni por AUTOT, de que el objeto que se va a usar como contenedor, o sea el de locno. esté actualmente presente. Esto debe ser hecho por ti, si así es requerido.

### **COPYOO objno1 objno2**

La posición del objeto al que se refiere objno2 se hace la misma que la posición en que estaba el objeto objno1.

El objeto que se toma actualmente como referencia se pone en objno2.

Hay cuatro acciones que permiten dar varios parámetros a los objetos: ponerlos en banderas, setearlos de banderas. Son usados para comparaciones o manipulaciones.

### **COPYOF objno. flagno.**

La posición del objeto al que se refiere objno. se copia en el flagno. de la bandera. Esto puede ser usado para examinar la localización de un objeto en comparación con otro valor de la bandera.

Por ejemplo: [COFYOF 1 11 SAME 11 38] puede ser usado para chequear que el objeto 1 esté en la misma localidad del jugador, aunque ISAT 1 255 sería mucho mejor.

### **COPYFO flagno. objno.**

La posición del objeto al que se refiere objno. se pone como contenido del flagno. de la bandera. Una intención de copiar de una bandera que contenga 255 resultará en un error de "Argumento Inválido".

El poner un objeto en una localidad inválida será aceptada puesto que no acarrea ningún peligro para las operaciones de PAW. Esto debes tenerlo en cuenta (y no hacerlo) tú.

### **WHATO**

Es un comando muy importante. Una búsqueda para el número del objeto representado por el Nombre (Adjetivo) 1 se hace en la tabla de Objetos Palabras en la siguiente prioridad: llevado, puesto, aquí.

Esto es, porque se supone que cualquier uso del comando WHATO se referirá más que todo a objetos llevados que a cualquier objeto que se lleve puesto o esté aquí.

Si un objeto se encuentra su número se pone en la bandera 51, lo mismo que con cualquier parámetro del objeto al cual se refiera actualmente en las banderas 54 a 57.

Esto te permite crear otras acciones de tipo automático (en el manual de introducción dábamos el ejemplo de esto dejando caer objetos desde el árbol).

### **WEIGH objno. flagno**

El verdadero peso al que se refiere objno. se calcula (por ejemplo: si es un contenedor, cualquier objeto que lleve dentro hará aumentar su peso, no te olvides de que puede haber varios contenedores

uno dentro de otro y todos sus pesos se pueden añadir hasta el nivel 10) y el valor que dé, se pone en el flagno. de la bandera. El máximo es 255.

Si el objeto al que se refiere objno. es un contenedor de peso cero, el flagno. de la bandera se pondrá a cero, puesto que cualquier objeto que esté en un contenedor que pese cero se considera que tiene un peso de cero (Sería un dispositivo antigraavedad en una aventura de ciencia ficción).

Ahora diez acciones que manipulan las banderas:

### **SET flagno.**

Pone el número de flagno. de la bandera a 255.

### **CLEAR flagno.**

Pone el número de flagno. de la bandera a 0.

### **LET flagno. valor**

Pone el número de flagno. de la bandera a su valor.

### **PLUS flagno. valor**

Le añade a flagno. el número que se ponga en valor. Si excede de 255 la bandera se setea a 255.

### **MINUS flagno. valor**

El número de la bandera en flagno. se decrementa por el número que se ponga en valor. Si el resultado es negativo, la bandera se deja a 0.

### **ADD flagno1 flagno2**

Se le añade a la bandera contenida en flagno2 el contenido de la bandera de flagno1. Si el resultado excede de 255 la bandera se pone a 255.

### **SUB flagno1 flagno2**

Se le resta al contenido de la bandera en flagno1 el contenido de la bandera flagno2. Si el resultado es negativo la bandera se pone a 0.

### **COPYFF flagno1 flagno2**

El contenido de la bandera flagno1 se copia en la bandera flagno2.

### **RANDOM flagno.**

La bandera cuyo número especifique flagno. se pone a un número que es resultado de una secuencia pseudo aleatoria de 1 a 100. Esto puede ser útil para permitir que se tomen decisiones aleatorias de una manera más sencilla que con la condición CHANCE.

### **MOVE flagno.**

Esta es una potente acción para la manipulación de los personajes pseudo inteligentes (PSI). Trabaja permitiendo que el Verbo actual de la sentencia lógica sea usado para buscar en la tabla de conexiones en busca de la localidad que se dé en la bandera cuyo flagno. se especifique.

Si el Verbo se encuentra, entonces el flagno. especificado cambia a ser el número de localidad asociado con él, y el próximo contacto se considera.

Si el Verbo no es encontrado, o el número de localidad original no era válido, entonces PAW se ocupa de la siguiente entrada en la tabla (si está presente).

De esta forma se podría considerar que PAW ejecuta las siguientes entradas imaginarias a su salida de Respuestas si no ha ejecutado ninguna acción:



_	_	MOVE	38		;Intenta mover al jugador
		DESC			;Describe su nueva localidad
_	_	LT	33	14	;¿Es una palabra usada para moverse?
		SYSMESS	7		;"No puedes ir a esa localidad"
		DONE			
_	_	SYSMESS	8		;"No puedo hacer eso"

Esta orden puede ser usada para dar a los personajes movimientos aleatorios en direcciones válidas, poniendo para ello el Verbo de la SL en una palabra de movimiento aleatoria y dejando que la orden MOVE decida si el personaje puede ir en esa dirección.

Hay que tener en cuenta, que cualquier movimiento especial (o movimientos) que se manejen en la tabla de Respuestas para el jugador, deben manejarse separadamente para un PSI.

Hay tres acciones para manipular las banderas que se usan para controlar al jugador:

### **GOTO locno.**

Cambia la localidad actual al número dado por locno. Efectivamente, setea la bandera 38 al valor que se dé en locno.

### **WEIGHT flagno.**

Calcula el peso verdadero de todos los objetos llevados y puestos encima por el jugador (también los de cualquier contenedor, que por supuesto tendrá el peso de su contenido hasta un máximo de 255), este valor se pone en la bandera especificada por flagno. Se usa para asegurar que un jugador no esté, por ejemplo, llevando mucho peso para cruzar un puente sin que éste se derrumbe, etc.

ABILITY valor1 valor2

Setea la bandera 37 (que tiene el número máximo de objetos llevados) al valor dado por valor1, y la bandera 52 (el máximo peso de objetos que el jugador puede llevar en cualquier momento), o sea, su fuerza, al valor dado por valor2.

*No se hace ninguna prueba* para chequear que el jugador ya no lleve con anterioridad un peso mayor que el permitido, porque GET y los demás, (que si tienen en cuenta estos valores), seguirán todavía trabajando correctamente y prevendrán que el jugador lleve más objetos, aún si has seteado el valor a un número más bajo que el número de objetos que ya lleva en este momento.

Hay siete acciones que se usan para la manipulación de banderas, el modo de pantalla, el tipo de formato, el INPUT, etc.:

### **MODE modo opción**

Hay cinco modos de pantalla, todos ellos controlados por la bandera 40 y elegidos usando la acción MODE de la siguiente forma:

#### **MODO 0**

Si un gráfico (diferente de una subrutina) está presente en la base de datos, la pantalla se limpia a los colores por defecto, y el dibujo se pinta. Cuando se pulsa una tecla, los colores originales son restaurados y la localidad es descrita.

#### **MODO 1**

Sólo texto. Scroll de pantalla entera. El texto va saliendo en la pantalla de forma continua. CLS no es activo durante la descripción.

## **MODO 2**

Si un gráfico (diferente de una subrutina) está presente en la base de datos, la pantalla se limpia a los colores por defecto y el dibujo se imprime (como en el Modo 0). Luego, los colores originales son restaurados, y el área que está debajo de la línea dada por la bandera 41 se limpia y el texto se imprime a partir de esa misma línea.

## **MODO 3**

Es como el Modo 2, pero el gráfico no hace Scroll.

## **MODO 4**

Es un modo para texto solo, que protege la descripción de la localidad como si fuera un gráfico, haciendo que el resto del texto haga un Scroll por debajo. Probablemente se necesitará una acción PROTECT en el punto adecuado en la tabla de Procesos 1 (¿después de LISTOBJ?) porque si no, sólo se protegería hasta la penúltima línea de texto.

En todos los modos gráficos dados arriba hay también opciones:

**Opción 2:** fuerza al borde *a no ser igual que el color del papel*; es la mejor opción para seleccionar en el modo 2 y el modo 3, porque si el borde se resetea en cuanto el dibujo está pintado del todo, causa un molesto efecto de parpadeo.

**Opción 1:** hace que el mensaje del sistema MS32 (Más...) aparezca cuando se llena el área de pantalla (a diferencia de "¿Scroll?" no se puede hacer un break, al menos mientras estés en el editor).

**Opción 3:** por supuesto, da las dos opciones anteriores y opción 0 no da ninguna.

Por ejemplo, con el MODO 4 1, tendremos una ventana de texto fijo con un mensaje de "Más...".

## **LINE lineno.**

Especifica la línea donde se inicia el texto en las pantallas partidas (con gráficos).

En caso de usar el Mode 2, especifica la línea superior del texto a limpiar. Cuidado, al usar INPUT en la parte baja de la pantalla, deja por lo menos cuatro líneas libres en la parte superior (no olvidarse de las líneas usadas por el marcador y también la línea en la que aparece el mensaje de contestación del ordenador).

## **GRAPHIC opción**

Es la que especifica el modo de cómo PAW trata los dibujos y por medio de la cual se pueden modificar. Hay tres opciones válidas:

- 0 Opción normal, cualquier gráfico que esté presente en la localidad es dibujado cuando el jugador entra por primera vez en esa localidad. A partir de aquí solamente serán dibujados si los gráficos se ponen artificialmente o se pide una dibujada temporal (ver más adelante).
- 1 Apagado, cualquier gráfico es ignorado por PAW aunque se pida una redescipción.
- 2 Encendido, cualquier gráfico que esté presente en la localidad se dibuja cada vez que el jugador entre en esa localidad.

Hay una opción 3, (sería la opción 2 + 1) pero la opción 1 APAGADO tendría prioridad siempre, por lo cual es redundante.

El dibujo actual para una localidad puede ser *redibujado añadiéndole 128 a la bandera 29*, esta es una dibujada temporal, que puede ser usada cuando el jugador quiera que aparezca el dibujo en el Modo Normal. Por ejemplo, habría que poner en la tabla de Respuestas, como respuesta a su petición: **PLUS 29 128 DESC.**

## **PROMPT sysno.**

Hace que el mensaje del sistema que especifica sysno. aparezca cada vez que PAW obtenga una línea de mando por el jugador. Serla la respuesta del ordenador. Si se deja sin ningún valor, el valor de 0 (por defecto) hará que PAW seleccione uno de los mensajes del sistema 1, 2, ,3, 4 o 5 en una proporción de: 30:30:30:10 respectivamente. Hay que hacer notar que esto no afecta a los mensajes que son impresos por medio de los contactos END o QUIT.

### **TIME duración opción**

Permite que los INPUT aparezcan después de un tiempo cuya duración se especifica en intervalos de 1,28 segundos.

Por ejemplo, la tabla de Procesos 2 se buscará otra vez si el jugador no teclea nada durante un tiempo especificado.

Opción, permite que esto también pueda ocurrir durante los mensajes de "PULSAR CUALQUIER TECLA" y "Más...".

Para calcular el número que se debe de usar para opción, debes sumar los números que mostraremos al lado de lo que se requiera, hasta que tengas la combinación ¡ideal!

1. Mientras se espera para la primera letra del INPUT solamente.
2. Mientras se espera para teclear cuando sale un mensaje de "Más..."
3. Mientras se espera a teclear después de una acción PULSA CUALQUIER. TECLA.

Por ejemplo, TIME 5 6 (6= opción 2+4) permitirá 6.4 (5 x 1,28) segundos de inactividad para el jugador después de aparecer el mensaje PULSA CUALQUIER TECLA o "Más..." antes de que se toque cualquier tecla para continuar. Y la orden TIME 5 3 (3 = opción 1 + 2) permitirá este periodo de inactividad solamente antes de la aparición del primer carácter de INPUT y después de "Más...".

TIME 0 0 interrumpirá todos los tiempos muertos (es el elegido por defecto).

### **INPUT opción**

Esta acción permite cambiar la forma de cómo opera la rutina de INPUT por medio de opciones.

Algunas combinaciones no se usan, pero son el producto de la forma como trabaja PAW. Para calcular el número que se va a usar, sólo hay que sumar los números que se dan para cada una de las opciones hasta que se tenga la combinación requerida.

1. hará que el mensaje requiriendo acción por parte del jugador, los marcadores y el cursor se pinten en la parte inferior de la pantalla. Esta es la forma preferida de INPUT para los juegos con tiempo muerto, donde pueden ocurrir entre tecleos, puesto que de esta forma no se deja la línea parcialmente impresa, si ocurriese un tiempo muerto mientras se está imprimiendo. (Nos referimos a impresión en pantalla).
2. hará que PAW imprima una copia de la línea de INPUT siempre que se apriete la tecla ENTER. Se usa normalmente con el número 1 para dejar que la línea final de INPUT aparezca en pantalla y así que el jugador sepa lo que ha tecleado.
3. Causará la reimpresión de cualquier texto que haya sido puesto cuando el INPUT se imprime después de un tiempo muerto. Lo mismo que el anterior, se debe usar principalmente con el número 1 donde puede ocurrir un tiempo muerto entre tecleos.

Por defecto PAW asume el INPUT 0 (no opciones).

### **PROTECT**

Se usa casi siempre en conjunción con el Mode 4. Esta acción lo que hace es que pone la línea a imprimirse como tope de la pantalla para el scroll de texto.

Debe ser usada solamente dentro del *Proceso 1* (lo puedes poner en cualquier otro sitio, pero no tiene ningún efecto) y después de cualquier texto que se quiera proteger una vez haya sido impreso. Suele

ser usualmente antes o después de una acción LISTOBJ, puesto que los objetos son usualmente lo último que se imprime.

Puede usarse en otros modos, por ejemplo, para permitir que un gráfico y una parte de una descripción se protejan dejando que el resto del texto haga un scroll por debajo.

Hay tres acciones que imprimen valores que llevan las banderas en la pantalla:

### **PRINT flagno.**

El contenido en decimal de la bandera a que se refiere flagno. se imprime con los colores actuales *sin dejar espacios ni antes ni después*. Esta es una acción muy eficaz.

Por ejemplo, si la bandera 100 contiene el número de monedas que lleva el jugador, entonces una entrada en la tabla de Procesos del tipo MES 10 "Tienes (espacio)" y ahora [ PRINT 100 MESSAGE 11 ] "(espacio) monedas de oro.", se usaría para que se imprimiera el número de monedas que tiene.

### **TURNS**

Los mensajes del sistema del 17 al 20 se imprimen, pueden ser del tipo de "Has hecho X movidas" o "Has dado X orden (es)". Aquí X sería el resultado de la bandera  $31 + 256 * \text{la bandera } 32$ .

### **SCORE**

Es el mensaje que hace imprimir la puntuación. Se imprime por medio del mensaje del sistema 21 y 22 en X%, siendo X el valor de la bandera 30.

Y ahora tenemos trece acciones que tratan de la forma y control de la pantalla:

### **CLS**

Limpia la pantalla a los colores de fondo definidos actualmente. También deja la posición actual de PRINT y de SAVEAT a 0,0.

### **NEWLINE**

Imprime espacios hasta el final de la línea y luego resetea los colores y el set de caracteres a los colores de fondo.

### **MES mesno.**

Imprime el mensaje dado por mesno. en los colores actuales. MESSAGE mesno.

Imprime el mensaje dado por mesno. en los colores actuales, *pero luego ejecuta una acción NEWLINE*.

### **SYSMESS sysno.**

Imprime el mensaje del sistema al que se refiere sysno. en los colores actuales.

### **PICTURE locno.**

Dibuja los gráficos definidos por locno. sin importar que sea una subrutina o un dibujo principal.

Nota: no se hace un clear de la pantalla y de los colores como cuando se describe una localidad.

Puede ser usado para añadir (o para quitar, si se usa PRINTAT y MES para quitar con espacios) partes de un dibujo.

*Hay que tener en cuenta que el dibujo será dibujado a partir del último punto usado por el dibujo que se pintó con anterioridad.*

#### **PAPER n**

Donde n va de 0 a 9. Cambia los colores de fondo del papel.

#### **INK n**

Donde n va de 0 a 9. Cambia los colores de fondo de la tinta.

#### **BORDER n**

Donde n va de 0 a 7. Cambia los colores del borde de la pantalla.

#### **CHARSET valor.**

Cambia el set de caracteres que existe usando el valor que se dé si es válido. Si no es válido no se ejecuta ninguna acción.

Los siguientes tres controles de pantalla permiten que una sección de texto sea impresa en la pantalla separada o fuera de la actual posición de impresión. Puede ser usado, por ejemplo, para crear una línea de información en la parte superior del dibujo actual, o para crear una sección de texto en una pequeña ventana dentro de un dibujo, etc, etc.

#### **SAVEAT**

Vacía o limpia el buffer actual restaurando por lo tanto los colores de fondo. Luego SALVA la posición actual de impresión por encima de cualquier otra posición salvada anteriormente.

#### **BACKAT**

Vacía o limpia el buffer actual restaurando por lo tanto los colores de fondo. Luego restaura la posición de impresión que ha sido salvada previamente por un SAVEAT.

#### **PRINTAT lineno. colno.**

La posición de impresión actual se cambia a un valor específico. También esto vaciará o limpiará el buffer actual y restaurará los colores temporales a los colores de fondo.

Tres acciones de hacer listados de objetos en pantalla. Los primeros dos están controlados por el valor de la bandera 53 tal como se describe en el capítulo dedicado a objetos. Y son:

#### **LISTOBJ**

Si cualquier objeto está presente, entonces el mensaje del sistema número 1 "Además puedo ver" se imprime, seguido de una lista de los objetos presentes en la localidad actual. Si no hay ningún objeto, entonces no se imprime nada.

#### **LISTAT locno.**

Si hay algunos objetos presentes, se listan. Si no, el mensaje del sistema número 53 "Nada" se imprime. Hay que tener en cuenta que usualmente hay que preceder esta acción con un mensaje del tipo de "En la bolsa hay", etc.

Sería posible crear una alternativa a la acción de INVEN que viene después usando 253 y 254 como parámetros en un LISTAT.

## **INVEN**

El Mensaje del Sistema número 9 "Tengo: \_" se imprime. Si no hay ningún objeto que lleves o que tengas puesto se imprime el mensaje del sistema número 11 "Nada en absoluto".

De otra forma el texto que define cada objeto que se lleva o se tiene puesto encima se imprime en una línea separada.

Si un objeto se lleva puesto, su texto irá seguido del Mensaje del Sistema 10 "Encima" o "Puesto", alineado del lado derecho de la pantalla si no cabe en la misma línea. La acción DONE es ejecutada luego.

Dos acciones hacen que PAW salga completamente de la tabla de Respuestas o de Procesos, son:

## **DESC**

Cancelará cualquier bucle DOALL, cualquier llamada a subproceso y hará un salto para describir la localidad actual.

## **END**

El Mensaje del Sistema 13 "¿Quieres jugar otra vez?" se imprime y se llama a la rutina de INPUT. Cualquier bucle DOALL y cualquier llamada de subprocesos son canceladas. Si la respuesta no empieza con el primer carácter que lleve el Mensaje del Sistema 31, se hace un salto a inicialización. Si no, se hace un salto al editor si está presente, o un NEW de BASIC si no está presente.

Tres acciones de salida de tabla:

## **DONE**

Esta acción salta al final de la tabla de Procesos e indica a PAW que ya se ha ejecutado una acción. Significa que no se consideran más entradas. Una vuelta se hará a la tabla de Procesos previa o al punto de inicio de cualquier bucle DOALL que esté activo.

## **NOTDONE**

Esta acción ejecuta un salto al final de la tabla de Procesos y le indica a PAW que ninguna acción ha sido hecha. Quiere decir que no se consideran más conductos ni entradas. Causará una vuelta a cualquier tabla de Procesos previa o a un punto de partida de cualquier bucle DOALL que esté activo. Esto hará que PAW imprima uno de los mensajes "No puedes" si es necesario, es decir, si no hay más acciones y no hay entradas presentes en las conexiones para el Verbo actual.

## **OK**

El Mensaje del Sistema número 15 "Vale" o "Bien" se imprime y la acción DONE se ejecuta. Cuatro acciones permiten que el estado actual del juego sea CARGADO y GRABADO:

## **SAVE**

El mensaje usual de empezar la cinta (start tape) se imprime al final de la pantalla. Cuando se pulse una tecla, la posición del juego se SALVA a cinta, luego, la acción DESC se ejecuta. Si se tecléa BREAK durante este proceso se ejecuta un salto a inicialización.

Un SAVE incluye toda la información requerida para permitir volver a jugar el juego en el punto exacto donde se estaba antes de ese SAVE, incluyendo los valores de las banderas, las posiciones de los objetos, las banderas que se hacen cargo de los dibujos, etc.

## **LOAD**

Carga datos (que deben ser una posición de juego) de cinta, luego, la acción DESC se ejecuta. Si se pulsa la tecla de BREAK durante una carga o se detecta un error de carga se hace un salto a

inicialización. Si la base de datos que se intenta cargar no es una posición del juego, normalmente ocurrirá un error de cinta.

### **RAMSAVE**

Es una forma similar de SAVE, pero que graba toda la información relevante del juego no a cinta, sino a un buffer en la memoria del ordenador. Este buffer, por supuesto, es volátil y se destruye cuando se apaga el ordenador. Esto debe de hacerse saber al jugador. Además, también se limpiará el buffer cuando vuelves a la sección de editor del PAW. Te aviso esto en caso de que cambies el diseño de tu juego.

### **RAMLOAD flagno.**

Esta acción es la contraria de RAMSAVE y permite que el buffer que antes se salvó sea restaurado. El parámetro dado por flagno. especifica la última bandera que debe ser cargada y se usa para mantener valores por encima de un restore. Por ejemplo una entrada de:

```
RAMLO _ COPYFF 30 255 (30 = score)
        RAMLOAD 254 (carga sin cambiar hasta bandera 254) COPYFF
        255 30 (haz la 30 = 25 y luego describe)
DESC
```

podría ser usada para mantener la puntuación actual y evitar que el jugador pueda usar el RAMSAVE/LOAD como una opción muy fácil de sacar un 100%. Como verás, lo que hace es cargar la puntuación en una bandera y luego devolverle a esa bandera la puntuación.

**Nota Importante:** a diferencia de SAVE y de LOAD las acciones que van a la RAM sí permiten que el próximo conducto se ejecute. Por ello, deben de ir normalmente seguidas por una acción DESC para que el juego se restaure en una posición idéntica.

Estas acciones pueden ser usadas para hacer el comando OOPS que se está poniendo de moda en otros sistemas. Se trata de poder volver a la jugada anterior si has metido la pata. Se hace creando una entrada en la tabla de Procesos 2 (o en la tabla de Respuestas) que haga un SAVE a la RAM automático cada vez que el jugador entre su movida.

Ahora dos acciones que permiten hacer una pausa en el juego durante un tiempo o hasta que se pulse una tecla:

### **ANYKEY**

El Mensaje del Sistema 16 "Aprieta una tecla para continuar" se imprime al final de la pantalla y se inicia una búsqueda en el teclado hasta que se detecte la presión de una tecla o hasta que haya pasado el tiempo que se ha usado como tiempo muerto, si es que se ha usado tiempo muerto.

### **PAUSE valor**

Pausa por un valor/50 segundos (Pause 50 = 1 segundo). Sin embargo, si el valor es cero entonces la Pausa es por 256/50 segundos. Hay que destacar que el teclado se desconecta durante la duración de una Pausa.

Dos acciones que controlan el parser:

### **PARSE**

Esta acción se diseñó para uso en lenguaje con los PSIs. Cualquier frase que se encuentre entre comillas (cualquier cadena) que esté presente en la frase actual tecleada por el jugador se convierte automáticamente en una Sentencia Lógica, por encima de cualquier SL ya existente que haya formado esa frase.

Si no hay una frase presente o no es válida, PAW buscará en el siguiente conducto. De otra manera la siguiente entrada se considera con la nueva SL del discurso hecho al PSI.

Por el hecho de que tiene prioridad sobre la SL actual, debe ser usada solamente en un tabla de subrespuestas en la forma de:

* *	PARSE			; Siempre hay que hacer esta entrada
	MESSAGE	x		; "Ellos no entienden"
	DONE			
palabra	palabra		CondAcc list	;Cualquier frase que nuestro PSI entienda
_ _	MESSAGE	x		;como arriba u otro mensaje diferente

También hay algunas otras formas de hacerlo:

DECIR nombre	SAME	pos 38	;¿Están ellos ahí?
	PROCESS	y	;Decodifica el discurso
	DONE		;Puesto que la SL se ha destruido hay que hacer siempre un DONE
DECIR nombre	MESSAGE z		; "Ellos no están aquí"
	DONE		

### NEWTEXT

Fuerza que se pierda cualquier frase que permanezca en la línea actual de INPUT. Se debe usar para prevenir que el jugador continúe con un INPUT dañado, sin tener un nuevo INPUT, si algo sucede durante una situación.

Por ejemplo, la acción COGER (GET) ya de por sí lleva un NEWTEXT si por alguna razón falla en coger el objeto que se requiere. Ello se usa para prevenir el desastre en una frase como:

COGE LA ESPADA Y MATA A MANOLO CON ELLA

por si se lleva a cabo un ataque a Manolo sin tener la espada, lo cual es muy peligroso.

Una acción para el sonido:

### BEEP duración tono

Ambos, la duración y el tono, van desde 0 a 255. La duración se hace en unidades de centésimas de segundo.

El valor del tono se debe obtener tomando el valor del comando BEEP en BASIC, añadiéndole 60 y luego dividiéndolo por 2.

Otras acciones de difícil clasificación:

### PROCESS procno.

Esta poderosa acción transfiere la atención de PAW hacia el número de tabla de Proceso especificado por procno. Este subproceso exhibirá las mismas características que la tabla que lo llamó. Por ejemplo, si ha sido llamado desde la tabla de Respuestas, PAW seguirá buscando la correlación del Verbo y el primer Nombre de la SL y las palabras que hayan sido entradas en la tabla principal.

Hay que darse cuenta de que es una verdadera llamada a una subrutina y de que cualquier salida desde la tabla nueva (por ejemplo causada por un DONE, OK, etc) llevará de nuevo el control al siguiente conducto que esté en lista después de la acción PROCESS.

Un subproceso puede ser llamado (anidar) a otros procesos hasta un profundidad de 10 en cuyo punto se creará un error de "Límite Alcanzado".



### **DOALL locno+**

Otra acción muy poderosa que permite la mejora de cualquier comando de tipo "ALL" (todo).

1. Se intenta encontrar un objeto en la localidad a la que se refiere locno. Si no se encuentra (no es satisfactorio) el DOALL se cancela y una acción DONE se ejecuta.
2. Si se encuentra un objeto, el número del objeto se convierte en el Nombre1 de la SL (y el Adjetivo1 si está presente) por medio de una referencia a la tabla de Objetos Palabras. Si Nombre (Adjetivo) 1 es similar o hace juego con Nombre (Adjetivo) 2 entonces se hace una vuelta al paso 1. Esto es para facilitar el "Verbo TODO EXCEPTO objeto" (coge todo excepto...).
3. El siguiente conducto y/o entrada en la tabla se considera entonces. Esto convierte efectivamente una frase del tipo "Verbo todo" en una del tipo "Verbo objeto" la cual será entonces procesada por la tabla como si el jugador lo hubiese tecleado de la última forma.
4. Cuando se hace un intento de salida de la tabla actual, si el DOALL todavía está activo (por ejemplo, no ha sido cancelado por una acción), entonces la atención del PAW vuelve de nuevo al bucle DOALL desde el paso 1. La búsqueda continuará desde el siguiente objeto (el que tenga un número más alto).

La consecuencia de este método de búsqueda a través de la tabla de Objetos Palabras es que los objetos que tengan el mismo Nombre y Adjetivo en su descripción, (y en donde el juego sólo sabe a cual objeto se refiere por su presencia en esa localidad) son buscados en un orden ascendente de número de objeto, si no algunos de ellos se perderían.

Si se usa un DOALL para cosas como abrir todo, hay que tener en cuenta el hecho de que, en el funcionamiento de la mayoría de las puertas, se han puesto banderas y que entonces habría que convertir esas banderas en objetos para poderlas incluir en un bucle DOALL.

### **RESET locno+**

Esta acción se diseñó para permitir juegos de partes múltiples donde los objetos que no van a ser llevados a la siguiente parte se resetean a sus localidades de inicio.

Todos los objetos que se puedan llevar a la siguiente parte o entre partes deben estar presentes (y con las mismas descripciones) en cada parte. Los demás pueden ser usados de cualquier forma en cada una de las partes.

Cualquier objeto que esté presente en la localidad actual es movido a la localidad designada por el locno. Y la localidad actual pasa entonces a ser la señalada por locno.

Cualquier otro objeto es seteado a la localidad que se dio en la tabla de Objetos Inicialmente En.

No tiene ningún efecto en las banderas.

La acción DESC se ejecuta cuando se completa.

Un ejemplo de cómo se usa este comando se dará en el capítulo donde se habla de los juegos de partes múltiples.

### **ENTERN valor**

Llama a una rutina externa que traiga el parámetro dado por valor. Hay un capítulo entero que trata de este comando.

## **DESCRIPCION DETALLADA DE LA BASE DE DATOS**

La base de datos consiste en un número de tablas interrelacionadas y también contiene un área de información variada; por ejemplo, valores de colores de fondo, números de objetos llevables, set de caracteres, etc.

En un ordenador de 128K las principales áreas de datos y punteros están presentes para cada área de RAM. Efectivamente cada página de RAM es una base de datos independiente.

Las tablas presentes (en el orden en que aparecen en la memoria) son:

## **UDG's**

Esta tabla, que tiene 152 bytes de longitud, contiene los gráficos definidos por el usuario. Los gráficos definidos por el usuario pueden ser cambiados con el Editor de Caracteres que se describe más tarde. Se utilizan códigos del 144 al 162.

## **SOMBREADOS**

Esta tabla tiene 128 bytes de longitud, y contiene las tramas usadas por el Editor Gráfico para sombreados. Estas pueden ser cambiadas con el Editor de Caracteres como se describirá más adelante. Códigos del 0 al 15.

Las dos tablas de UDG son cargadas y salvadas como un par y se llaman set de caracteres 0. (aunque la ROM dé los códigos del 32 al 127).

## **MISCELANEA**

Son varias pequeñas tablas y algunos valores sueltos, 50 bytes en total.

## **EL SET DE CARACTERES**

Esta tabla está vacía hasta que se inserte un set de caracteres usando el Editor de Caracteres. Como es bien sabido, cada set de caracteres usará 763 bytes.

## **DICCIONARIO**

Esta tabla contiene solamente 1 byte hasta que el compresor se utilice por primera vez sobre la base de datos. Después contendrá 222 bytes de un diccionario de expansión para permitir que los tokens 165 a 255 puedan ser convertidos en las agrupaciones de letras que ellos representan.

## **LAS TABLAS DE PROCESOS**

Son el corazón de la base de datos puesto que desde ellas se controla el juego.

## **LA TABLA DE RESPUESTAS**

Cada entrada contiene el valor de palabra del Verbo y del Nombre que va a ser aparejado con la SL, seguido por cualquier número de conductos.

Cuando se juega la aventura, si hay alguna entrada en la tabla que hace juego, o es igual, al Verbo y al Nombre1 de la SL, entonces el conducto es válido y se ejecuta.

Todos los conductos y lo que hacen, se ha especificado completamente en la descripción del intérprete.

El orden de entrada en la tabla es por valor ascendente de Verbo.

En las entradas que tienen el mismo valor de Verbo, el orden es por valor ascendente de Nombre.

Entradas que tienen el mismo valor de Verbo y de Nombre se ordenan por el orden en que fueron insertadas en la base de datos.

Muy importante saber que la palabra "\_" tiene un valor (como palabra) de 255 mientras que la palabra "\*" tiene un valor (como palabra) de 1.

Véanos un ejemplo del orden de las tablas (con los valores de las palabras mostrados entre paréntesis):

*	(1)	*	(1)
MIRA	(30)	ARRIBA	(10)
MIRA	(30)	ABAJO	(11)
MIRA	(30)	_	(255)
COGER	(100)	*	(1)
COGER	(100)	LLAVE	(16)

COGER	(100)	LAMPARA	(26)
COGER	(100)	LAMPARA	(26)
_	(255)	_	(255)

Las otras tablas de procesos llevan el mismo formato, pero las palabras son ignoradas por PAW mientras busca en ellas. **A menos que el proceso haya sido llamado desde una tabla de Respuestas usando una acción PROCESS**, en cuyo caso funcionará como una verdadera tabla de Respuestas, aunque sea una tabla de Procesos.

### **PROCESOS 1**

Es buscada por PAW inmediatamente **después** de la descripción de una localidad. Ello permite que se añada cualquier información adicional que forme parte de la localidad, pero que pueda ser variable.

### **PROCESOS 2**

Es buscada por PAW en cada turno. Es decir, después de cada frase sacada del INPUT del jugador, o después de cualquier tiempo muerto durante el INPUT.

El Verbo y el Nombre que se usan en cada entrada en las tablas de Procesos 1 y 2 (y en cualquier subproceso llamado desde ellas) no tiene ningún significado, puesto que son ignorados, pero pueden ser usados por el escritor para tener una idea de la función que ejecuta cada entrada dentro de la tabla. Cada tabla de proceso tiene una cabecera de siete bytes (21 bytes son usados en una base de datos vacía, puesto que tres procesos -incluyendo la tabla de Respuestas- están ya presentes). Cada entrada subsiguiente usa 5 bytes y cada conducto usa 1, 2 o 3 bytes dependiendo del número de parámetros.

**NOTA muy importante:** Si una palabra se borra del vocabulario y no tiene sinónimos, entonces todas las entradas en las tablas de Procesos y de Respuestas que contengan esa misma palabra serán borradas. Cuidado pues con los borrados del vocabulario.

### **LA TABLA DE OBJETOS RELACIONADOS CON TEXTO**

Esta tabla, que tiene una entrada para cada objeto, contiene el texto que se imprime cuando el objeto es descrito. Cada entrada usa 3 bytes, más la longitud del texto.

Un objeto es cualquier cosa en una aventura que puede ser manipulado y los objetos son numerados desde el 0 hacia arriba. El objeto 0 es para el intérprete una fuente de luz.

Siempre que se inserta un nuevo objeto, una entrada como no-creado se hace para ese objeto en la tabla de INICIALMENTE EN, otra entrada de "\_ \_" en la tabla de Palabras Objeto y un peso de una unidad. Sin ser contenedor y sin ser llevable ni quitable se inserta también en la tabla de Objetos Peso.

### **LA TABLA DE TEXTO DE LOCALIDADES**

Esta tabla, que tiene una entrada para cada localidad, contiene el texto que se imprime cuando una localidad se describe. Cada entrada usa 3 bytes más la longitud del texto. Las entradas se numeran desde 0 hacia arriba y la localidad 0 es la localidad en la cual se comienza la aventura. Siempre que se inserta una nueva localidad, una entrada en blanco para esa localidad también se hace en la tabla de conexiones.

### **LA TABLA DE TEXTOS DE MENSAJES**

Esta tabla contiene el texto para cualquier mensaje que se necesite en la aventura. Lo mensajes son numerados desde 0 hacia arriba y cada uno usa 3 bytes más la longitud del texto.

## LOS MENSAJES DEL SISTEMA

*Esta tabla contiene los mensajes usados por el Intérprete.* Cada entrada usa 3 bytes más la longitud del texto. La descripción del Intérprete muestra cuándo se usan esos mensajes. Además se pueden insertar mensajes extra por el escritor si así lo requiere el juego.

## LA TABLA DE CONEXIONES

Esta tabla tiene una entrada para cada localidad y cada entrada puede estar vacía o contener un número de "pares de movimiento".

Un par de movimiento consiste en una palabra con el valor de un Verbo (o un Nombre de conversión a Verbo) del vocabulario seguido por el número de una localidad.

Significa que cualquier Verbo (o Nombre de conversión a Verbo) con ese valor causa un movimiento hasta esa localidad.

Una entrada típica sería: SUR 6 ESTE 7 SALIR 6 NORTE 5 lo que significa que SUR o SALIR (o sus sinónimos) producirán un movimiento a la localidad 6. ESTE (o su sinónimo) a la localidad 7 y NORTE (o su sinónimo) a la localidad 5. Cada entrada usa 3 bytes más 2 bytes por cada par de movimiento.

**Nota 1.** El par de movimiento contiene el valor de la palabra, no la palabra actual. Esto significa que si el valor de una palabra se borra del vocabulario (borrando todas las palabras que lleven ese número) todos los pares de movimiento que contengan ese valor también serán borrados.

**Nota 2.** Cuando la aventura completa se está jugando, lo único que producirá movimiento será el Verbo de la SL.

**Nota 3.** Si se ejecuta un movimiento desde una entrada en la tabla de Respuestas usando la acción GOTO, entonces no se necesita poner ningún par de movimiento en la tabla de conexiones, a menos que esa entrada sea necesaria para un PSI que se puede mover incondicionalmente.

## EL VOCABULARIO

Cada entrada en la tabla usa 7 bytes de memoria y contiene una palabra (o los primeros cinco caracteres de esa palabra si tiene más de cinco letras), un tipo de palabra desde el 0 al 6 y un valor de palabra que va desde el 2 al 254. Los tipos de palabra son:

- 0 - Verbo
- 1 - Adverbio
- 2 - Nombre
- 3 - Adjetivo
- 4 - Preposición
- 5 - Conjunción
- 6 - Pronombre

Las palabras que tienen el mismo valor de palabra son sinónimos. Las entradas se ordenarán según el orden ascendente de valor de palabra.

Si tienen el mismo valor, las entradas que tengan menos letras (o más espacios) vendrán primero por ejemplo:

E  
EXAM  
EXAMI(NA)

cuando las entradas de la misma palabra tienen el mismo valor y también tienen el mismo número de letras entonces se ordenan cronológicamente.

**Nota 1.** Siempre que el editor tiene que convertir un valor de palabra a una palabra, tomará la primera que tenga ese valor.

**Nota 2.** Los Verbos y los Nombres de Conversión a Verbo con valores menores de 14 deben reservarse para palabras-movimiento.

## **LA TABLA DE LUGAR INICIAL DE OBJETOS**

Esta tabla tiene una entrada de 1 byte para cada objeto. Especifica la localidad en la cual el objeto se sitúa al principio de la aventura. Un objeto también puede comenzar la aventura siendo llevado puesto, llevado, o no creado.

## **LA TABLA OBJETO-PALABRA**

Esta tabla tiene una entrada de 2 bytes para cada objeto, y lleva el valor del Nombre y del Adjetivo asociado con ese particular objeto.

## **LA TABLA DE PESO DE OBJETO Y DE ATRIBUTOS**

Esta tabla tiene una entrada de 1 byte para cada objeto. El bit 7 (128) se usa para decidir si el objeto puede ser puesto encima y quitado (por ejemplo con la opción WR). El bit 6 (64) se usa para mostrar que el objeto es un contenedor (con la opción C) y los bits 0 a 5 definen el peso del objeto (dando desde 0 a 63).

## **AREA GRAFICA**

El área gráfica de la base de datos crece hacia ABAJO desde la parte superior de la memoria:

## **LAS BANDERAS DE LOCALIDAD**

Esta tabla tiene una entrada de 1 byte para cada gráfico. El bit 7 especifica si el gráfico puede ser dibujado cuando se alcanza esa localidad (si la localidad no es una subrutina). Los bits 0 a 5 describen el PAPEL y la TINTA de comienzo para el dibujo. El bit 6 queda sin usar.

## **LA TABLA DE GRAFICOS**

Cada entrada en la tabla usa 3 bytes más la longitud de la cadena. Siempre habrá un número de entradas en la tabla igual al de las localidades creadas para la aventura. La cadena de dibujo está codificada como una variedad de comandos de diferentes longitudes, lo cual minimiza la cantidad de memoria necesaria para producir el dibujo.

## **LOS PUNTEROS**

Tiene los principales punteros de la base de datos.

## **DESCRIPCION DETALLADA DEL EDITOR**

Cada opción de menú se va a describir en el orden que aparece:

## **I. VOCABULARIO**

1. Insertar = I palabra No. (número) y tipo.  
No. va desde 2 a 254  
TIPO, va de 0 a 6

Si PALABRA no está ya presente en el Vocabulario, se inserta con un valor de No. y un tipo de tipo.

2. Borrar = D palabra

Si palabra está presente en el Vocabulario, ella, su tipo y su valor son borrados. Si hay sinónimos de esa misma palabra borrada presentes en el Vocabulario, no sucede nada.

Sin embargo, si no hay sinónimos presentes entonces:

- a) todas las entradas en las tablas de procesos que usen ese valor de palabra serán borradas también.
- b) si el valor de la palabra es menor de 14, entonces todos los movimientos en las tablas de conexiones que usen ese valor de palabra también serán borrados.
- c) todas las entradas en la tabla de Objeto-Palabra que -usen ese valor de palabra se pondrán a 0.

3. Mostrar sinónimo = S palabra

Si palabra está presente en el Vocabulario, ella y cualquier otra palabra que tenga el mismo valor y el mismo tipo aparecerá en pantalla. (Hay que destacar que para los Nombres convertibles en Verbos, los que son menores de 20, cualquier Verbo que sea sinónimo se mostrará y viceversa).

4. Imprimir = P (tipo) o L (tipo)

Imprimir se puede hacer en la pantalla con P, o por la impresora usando L.

Si se especifica el tipo, solamente esa clase de palabras aparecerán en el listado.

Nota 1. Hay que tener cuidado con usar la opción de BORRAR, puesto que puede afectar a las tablas de Procesos, de Conexiones y de Objetos-Palabra. También es una opción que puede llevar bastante tiempo (incluso minutos) si la base de datos es muy larga y hay que borrar muchas entradas que llevan esa misma palabra.

Nota 2. Los Verbos y los Nombres que tengan un valor menor de 14, son palabras de movimiento para el Intérprete, y todos ellos causarán que el Mensaje del Sistema número 7 "No puedo ir en esa dirección" se imprima, en vez del Mensaje número 8 "No puedo hacer eso".

Nota 3. Los valores de palabras del 2 al 254 pueden ser usados para cualquier tipo de palabra. **Las palabras no serán sinónimos si tienen un diferente tipo.** Esto permite que más de 250 palabras de cada tipo existan en el Vocabulario.

## II. EL TEXTO DE LAS LOCALIDADES

1. Insertar = I

Si ya se ha insertado el número máximo de localidades permitidas, entonces se causará el error "Límite Alcanzado".

Si no, la siguiente localidad libre en la página de RAM que se esté usando (que se muestra en la opción Memoria Libre), se adjudicará y una entrada libre **se hace para esta localidad en las banderas de localidad de gráficos, en las tablas de dibujo, en las tablas de conexiones, y en las tablas de texto para localidades.**

El proceso continúa entonces con una llamada automática a la rutina de arreglo (Amend) para permitir al escritor corregir el texto vacío que se ha dejado libre.

2. Comenzar Nueva Página = B

Si todas la páginas de esa RAM han sido usadas o el máximo número de localidades ha sido ya insertado, el error "Límite Alcanzado" se genera.

Si no, en la siguiente página de RAM disponible se crea una base de datos vacía y en ella se asignará la siguiente localidad que esté libre. El proceso continúa entonces con una llamada automática a la rutina de corrección para permitir al escritor corregir el texto nulo que se ha entrado en esa nueva página.

### 3. Arreglar = A locno.

El texto que ya existía para el número que especifique locno. se copia en el buffer de entrada y se imprime en la parte de abajo de la pantalla para su arreglo. Cuando se tecléa ENTER la entrada que ya existe es reemplazada con los contenidos que se hayan puesto en el buffer de entrada.

### 4. Imprimir = P (locno.) o L (locno.)

La impresión se hace en la pantalla usando P o en la impresora usando L. El proceso de impresión comienza por el texto de la localidad que se haya especificado con locno. o al principio si locno. no se ha especificado.

Nota 1. El comienzo de la aventura es *siempre* en la localidad 0.

Nota 2. Hay un límite de 252 localidades.

Nota 3. Es imposible comenzar una nueva página si ya se han usado todos los números de localidades posibles.

## III. CONEXIONES

### 1. Arreglar = A locno.

La entrada existente para la localidad que especifique locno. se decodifica, se copia en el buffer de entrada y se despliega en la parte de abajo de la pantalla para su arreglo.

Al teclear ENTER el buffer de INPUT puede haber quedado vacío o contener un par consistente en una palabra y el número de la localidad.

La Palabra debe ser un Verbo (o un Nombre convertible a Verbo) que esté presente en el vocabulario y locno. debe estar presente en la tabla de texto para localidades.

Si no existe ningún error de sintaxis, la entrada que ya está presente se reemplaza con una copia codificada del buffer de INPUT (es decir, las palabras han sido cambiadas a sus valores de palabras).

### 2. Imprime = P (locno). o L locno.

Imprimir se puede hacer en pantalla con P, o con la impresora usando L. La impresión empieza en la localidad que especifique locno. o desde el comienzo si locno. no ha sido especificado.

Nota 1. Un texto para la localidad debe estar presente antes de que las conexiones se puedan hacer.

Nota 2. Cualquier Verbo (o Nombre convertible a Verbo) en el Vocabulario puede ser usado en la tabla de Conexiones, no solamente las palabras especiales para movimiento.

Nota 3. Cuando una entrada es decodificada (con arreglar o imprimir) los valores de palabra son cambiados al primer Verbo (o Nombre de conversión a Verbo) del Vocabulario que tenga ese número de palabra.

## IV. GRAFICOS

### 1. Arreglar = A picno.

Con este comando se expande la base de datos que contiene los gráficos para dejar espacio libre al final del dibujo requerido.

Luego entraremos en el bucle principal del Editor de Gráficos que se describe más abajo. Cuando se tecléa RETURN cualquier espacio libre que aún quede es eliminado. Es importante tener en cuenta que al contrario de cuando se edita texto, aquí la base de datos misma cambia, por lo tanto no se puede abandonar la edición con CAPS SHIFT + 6 (de todos modos no podrías, porque lo que hace es un comando del tipo NEXT).

## 2. Tamaño = S

El número de bytes entre el principio de la base de datos del dibujo y el principio de la siguiente es calculado y se imprime en la pantalla.

## 3. Imprimir, Copiar y Volcado = P picno., C picno. y D picno.

Picno. debe ser especificado.

Si P es seleccionado se pide un nombre de fichero.

El gráfico pedido se dibuja en la pantalla.

Si C fue seleccionado, se llama a la rutina de COPY de la impresora (ver la sección sobre impresoras para detalles).

Si se seleccionó D, un fichero del tipo SCREENS se SALVA a cinta con el número de fichero que se le ha dado (es un bloque de código de pantalla: 16334,6912), este fichero puede entonces ser usado como pantalla de carga para el juego o editado usando otro paquete de utilidad gráfica.

Nota 1. No se puede recargar un fichero de tipo SCREENS en la base de datos.

## V. COLORES POR DEFECTO.

### 1. Arreglar = A picno. (Papel Tinta)

Una bandera se setea para indicar que el dibujo al que se refiere picno. es una subrutina ***a menos que el valor del PAPEL y de la TINTA sean especificados***, en cuyo caso estos son guardados como los colores por defecto del dibujo. ***Cualquier localidad en la aventura que no requiera un dibujo debe ser Arreglada como una subrutina.***

### 2. Imprimir = P o L

Se puede imprimir por pantalla usando P, o por la impresora usando L. Si la localidad no es una subrutina, los colores se imprimen.

## VI. MENSAJES

### 1. Insertar = I

El siguiente número de mensaje que esté libre se presenta y una entrada en blanco se guarda para ese número en la tabla de textos de mensaje, en la página más alta de RAM usada. Entonces se hace una llamada automática a la rutina de arreglo para que el usuario pueda corregir esa entrada en blanco.

### 2. Comenzar nueva Página = B

Al siguiente número de mensaje libre se le da una entrada en blanco (nula), en la siguiente página de texto de la RAM, ***siempre que haya sido inicializada*** en el menú de localidad de texto.

Si no ha sido así, el error de "página no inicializada" se genera. El proceso continúa con una llamada automática a la rutina de arreglo para permitir al usuario arreglar la entrada en blanco.

### 3. Arreglar = A mesno.

El texto que exista para el mensaje designado con mesno. se copia en el buffer de INPUT y se imprime en la parte baja de la pantalla para su arreglo. Cuando se tecléa ENTER el texto que existe se reemplaza con el que contenga el buffer de INPUT.



4. Imprimir = P (mesno.) o L (mesno.)

La impresión puede ser hecha a la pantalla usando P, o a la impresora usando L. La impresión empieza con el texto del mensaje especificado por mesno., o al principio si mesno. no ha sido especificado.

Nota 1. Hay un límite de 255 mensajes, pero como los Mensajes del Sistema también pueden ser utilizados, ello nos da un total de 510 mensajes (aunque algunos ya están usados).

Nota 2. No se puede comenzar una nueva página de mensajes sin primero inicializar esa página insertando una localidad en ella.

## VII. MENSAJES DEL SISTEMA

1. Insertar = I

El siguiente número de mensaje del sistema que esté libre es rellenado con una entrada en blanco en la tabla de mensajes del sistema en la página de RAM 0. Una llamada automática se hace entonces a la rutina de arreglo para permitir al escritor arreglar la entrada en blanco.

2. Arreglar = A mesno.

El texto existente para los mensajes del sistema, (o sea mesno.), se copia en el buffer de INPUT y se pone en la parte de abajo de la pantalla para arreglar. Cuando se tecléa ENTER el texto existente es reemplazado por el que contenga el buffer de INPUT.

3. Imprimir = P (mesno.) o L (mesno.)

La impresión se hace a la pantalla usando P, o a la impresora usando L. La impresión empieza por el texto especificado con el Mensaje del Sistema mesno., o al principio de los mensajes si mesno. no es especificado.

**Nota 1.** La descripción del Intérprete nos muestra dónde y cuándo se imprimirán esos mensajes. Pueden ser cambiados para segunda persona o primera persona según las preferencias, pero hay que tener cuidado de mantener su significado original.

**Nota 2.** Los mensajes 30 y 31 no son en realidad mensajes, sino que contienen las respuestas positivas y negativas usadas en los contactos QUIT y END. Por lo tanto, hay que ser muy cuidadosos al cambiarlos, puesto que la acción END es la única y principal forma de volver al Editor desde el Intérprete.

Tampoco hay que hacerlos de más de una sola letra de longitud, porque si no el compresor los convertirá a tokens.

**Nota 3.** El Mensaje del Sistema 10 "Puesto" o "Llevado encima", tiene ya su longitud calculada por el comando INVEN. Esto significa que *no debe incluir ningún código de control y debe ser vuelto a teclear si se ha usado el compresor para quitar cualquier tokenización en él. Además, debe de comenzar siempre con un espacio para que el cálculo de su lugar en la pantalla funcione.*

Nota 4. El Mensaje del Sistema número 34 (el que contiene el cursor), sí que puede contener cualquier número de caracteres de control, pero no debe moverse la posición en que se imprime más de un espacio o el INPUT se hará bastante confuso.

Cualquier texto que continúe estará en los mismos colores que el que se haya forzado al final de este mensaje.

**Nota 5.** Se pueden insertar hasta 202 mensajes más para tu propio uso, pero como no hay una acción equivalente a la acción de MESSAGE, una acción NEWLINE debe ejecutarse al final, si se necesita.

**Nota 6.** El PAW para el CPM usa los mensajes números 54 a 60. Esto hay que tenerlo en mente si se intenta transferir un juego de un ordenador a otro. MUY IMPORTANTE PARA LOS QUE PLANEAN COMERCIALIZAR LOS JUEGOS.

## VIII. EL TEXTO DE LOS OBJETOS

### 1. Insertar = I

Se escoge el siguiente objeto libre y se hace una entrada en blanco para él en la tabla de objeto texto. Una entrada de "objeto no-creado" se hace también para ese objeto en la tabla de Lugar Inicial de Objeto.

Una entrada de "\_ \_" en la tabla de Objeto Palabra.

Se le da el peso de 1 unidad (sin dar más atributos), en la tabla de peso de objetos.

Luego el procesador continúa con una llamada automática a la rutina de arreglo para permitir al usuario corregir la entrada en blanco que ya se ha creado en la tabla de texto de objetos.

### 2. Arreglar = A objno.

El texto que existe para el objeto que especifica objno. se copia en el buffer de INPUT y se muestra en la parte de abajo de la pantalla para su corrección. Cuando se aprieta ENTER, el texto que existe es reemplazado por los contenidos del buffer de INPUT.

### 3. Imprimir = P (objno.) o L (objno.)

La impresión se hace a la pantalla usando P, o a la impresora usando L. La impresión comienza con el texto marcado por objno., o al principio si no se ha especificado ninguno.

**Nota 1.** El objeto 0 se considera siempre por el Interpretador como una fuente de luz.

**Nota 2.** Hay un límite de 255 objetos.

## IX. LA TABLA DE LUGAR INICIAL DE OBJETOS

Es donde se especifica la localidad en la cual se sitúa el objeto al comienzo de la aventura.

### 1. Arreglar = A objno. locno.

La entrada existente para el objeto que especifica objno. se reemplaza con la localidad que especifique locno. Esta locno. debe, o bien estar presente en la tabla de texto de localidades, o ser una de las localidades especiales, por ejemplo (252 no creada), (253 puesto encima) o (254 llevado).

### 2. Imprimir = P o L

Se imprime a la pantalla usando P, o a la impresora usando L.

**Nota 1.** Debe estar presente el texto para los objetos antes de que pueda ponerse en qué localidad se va a situar.

## X. LA TABLA DE RELACION OBJETO PALABRA

### 1. Arreglar = A objno. Nombre Adjetivo

La entrada existente para el objeto que especifica el objno. se reemplaza con los valores de Nombre y Adjetivo que estén presentes en el Vocabulario (o se puede usar también un subrayado para que la entrada se haga en blanco).

### 2. Imprimir = P o L

Se puede imprimir en la pantalla usando P, o en la impresora usando L.

## XI. LA TABLA DE RELACION OBJETO PESO

### 1. Arreglar = A objno. Peso Opción

La entrada existente para el objeto especificado por objno. se reemplaza con el peso y la opción que se haya especificado. El peso puede variar desde 0 hasta 63.

La opción especifica otros atributos que puede tener el objeto, y son los siguientes:

0 - No atributos.

1 - Es un contenedor.

2 - Se puede llevar encima y quitar de encima.

3 - Es un contenedor que se puede poner y quitar de encima. (Por ejemplo, una mochila).

## 2. Imprimir = P o L

Imprimir a la pantalla usando P, o a la impresora usando L. El peso puede estar seguido por una "C" para indicar que es un contenedor o por una "WR" para indicar que se puede llevar encima y quitar (wearable/removeable).

Nota 1. Solamente se puede seleccionar un objeto con atributo de contenedor si hay una localidad que tenga el mismo número. Esto es así, porque PAW trata esta localidad como si fuera "la parte de dentro del objeto contenedor".

Nota 2. Un contenedor de peso 0 no tendrá como peso final la suma de todos los objetos que contenga, sino que su peso será siempre 0. Esto puede ser usado para crear un saco mágico o un transportador por levitación, etc.

## **XII. LAS TABLAS DE PROCESOS (incluyendo la de RESPUESTAS)**

Solamente se pueden escoger dos opciones del submenú principal de Procesos:

### 1. Comenzar una nueva Tabla = B

Con esto, el siguiente número de proceso que esté libre se adjudica y se crea una tabla nula con este número. El Proceso continúa con una llamada a la opción de Selección de tablas pero con el número de la nueva tabla.

### 2. Selección de Tabla = S No.

La tabla que se especifique por No. se hace la tabla actual; se selecciona y entonces el submenú aparece de nuevo pero con este número de tabla.

Las otras opciones se encuentran tanto en los submenús de RESPUESTAS y los de PROCESOS, y son:

### 1. Arreglar = A Verbo Nombre (n)

Si el número de entrada (n) no se especifica, la primera entrada que aparezca en la tabla con ese Verbo y ese Nombre se copia en el buffer de INPUT y se muestra en la parte de abajo de la pantalla para su corrección. Si no, el número especificado se usa (desde 0 hasta el número de entradas que hayan sido insertadas menos 1).

Si al apretar ENTER el buffer de INPUT está vacío, la entrada correspondiente (sin tener contactos) se borra. Si contiene cualquier número de contactos válidos no se borrará. Si no existen errores de sintaxis, entonces la tabla existente es reemplazada por los contenidos del buffer de INPUT.

Cualquier siguiente entrada en la tabla que tenga los mismos valores (por ejemplo el mismo Nombre y el mismo Verbo) se muestra entonces para que llegue su turno de corregirla de la misma manera.

### 2. Insertar = I Verbo Nombre (n)

El Verbo y el Nombre pueden ser rayas, asteriscos o palabras que estén en el vocabulario. El valor del número que tenga el Verbo y el Nombre (la raya tiene un valor de 255 y el asterisco el valor de 1) se usa para encontrar el orden correcto dentro de la tabla para la nueva entrada que va a ser creada.

Si hay ya una entrada que tenga ese mismo Verbo y Nombre y no especifica el número (n), entonces la nueva entrada será creada después de la que ya existe. De otra forma (si se especifica el número), la

entrada será creada antes del número especificado. *El orden en que estén las entradas puede ser muy importante para la marcha del juego.*

Una entrada en blanco se crea (con inserción), en el lugar apropiado y se hace una llamada automática a la rutina de arreglo para permitir al usuario corregirla de cualquier forma.

Los contactos que pueden ser usados han sido mostrados en la descripción del Intérprete y en el sumario al final de esta guía.

### 3. Imprimir = P (Verbo (Nombre)) o L (Verbo (Nombre))

La impresión puede ser en pantalla usando P, o a la impresora usando L.

La impresión empieza con la primera entrada que tenga los valores de palabra de Verbo y Nombre. Si Verbo y Nombre no se especifican, entonces un número de palabra de valor 0 se supone que es el comienzo.

Por lo tanto, apretando P o L solamente sin decir ningún número, comenzarás la impresión siempre al principio de la tabla.

**Nota 1.** Para borrar una entrada; arréglala para que no quede ninguna condición ni ninguna acción.

**Nota 2.** Hay un límite de 255 tablas de proceso. Y las llamadas en un bucle anidado pueden hacerse hasta un máximo de 10, después de lo cual ocurre un error.

## OPCIONES EXTRAS (E)

Con esto se seleccionan las otras opciones de que dispone el menú.

## XIII. PROBANDO LA AVENTURA

El mensaje "¿Necesitas diagnóstico?" se imprime, y cualquier contestación que no comience con la letra "S" se tomará como negativa. Si esto sucede, se ejecuta un salto hacia el Intérprete.

Si se requirió o se pidió diagnóstico, entonces mientras se está probando la aventura, y cuando el Intérprete esté pidiendo una rutina de Input, basta con apretar la tecla ENTER para que aparezca el valor de una bandera en la zona inferior de la pantalla.

También se puede teclear el número de cualquier otra bandera para ver su contenido o "=" seguido de un número para hacer que la bandera que esté en pantalla adquiera ese valor.

Tecleando ENTER otra vez volvemos a la rutina de Input.

La mejor manera de volver de nuevo al Editor desde el Intérprete (que es el que está actuando mientras se prueba la aventura) es por medio de una acción END en la tabla de Procesos.

## XIV. MEMORIA LIBRE

Con esta opción se imprime el número de bytes que quedan por usar en cada página. Las páginas que no hayan sido inicializadas todavía, se mostrarán como "sin usar". Si la página 7 contiene actualmente algún Overlay también se informa de este hecho.

Además de la memoria libre, también se da una lista del último mensaje y de la última localidad usados en cada página para un mejor control.

Suponemos que te das cuenta de que en un Spectrum de 48K solamente aparecerá la página 0, puesto que es la única que hay.

## XV. COLORES DE FONDO

Los colores para el BORDE, PAPEL y TINTA se pueden poner a cualquier valor que sea válido.

Se recomienda usar tinta 9 (que corresponde a contraste), pero hay que tener en cuenta que la tinta 9 se comporta de una forma diferente en la parte inferior de la pantalla.

Además, si cualquier otro set de caracteres se ha insertado en la base de datos, se podrá en estos momentos seleccionarlo como el set primario.

## **XVI. CARACTERES**

Un carácter consiste en una parrilla de 8x8 píxeles y define la forma en que las letras, números, puntuación, gráficos definidos por el usuario y tramas aparezcan en pantalla.

Inicialmente, sólo estará presente un set de caracteres: el que se contiene dentro de la ROM del Spectrum. Este set también contendrá las 16 tramas de sombreados (desde el código 0 hasta el 15) y los 19 gráficos definidos por el usuario (código 144 al 162) y se denomina set 0.

**Nota:** En la versión castellana se ha definido otro set diferente al de la ROM del Spectrum por considerarlo mucho más adecuado.

### **Insertar I**

Si aún no se ha alcanzado el máximo número de sets posibles, con esta opción se localiza el siguiente set disponible.

Una entrada de 768 bytes se inserta en la tabla de caracteres y una copia del diseño para los 96 caracteres (códigos 32 al 127) en la ROM se copia en esa entrada. De otra forma ocurrirá un error de "límite alcanzado".

### **Arreglar A un carácter de un set**

El carácter especificado del set especificado se presenta en pantalla para su arreglo.

Solamente los caracteres del 0 al 15 y del 144 al 162 pueden ser arreglados desde el set 0, mientras que solamente los caracteres del 32 al 127 pueden ser arreglados en cualquier otro set. Esto es importante.

El Editor crea entonces tres áreas en la pantalla:

- |                                       |   |
|---------------------------------------|---|
| 1/ La zona de enrejado o cuadrículado | Es donde se muestra de una forma bastante o cuadrículado magnificada (x8) la forma en que está constituido un carácter. Se presenta en forma de cuadraditos amarillos y blancos, donde los bits que estén seteados se mostrarán en negro. También se muestra la posición actual del cursor por medio de un cuadradito rojo parpadeante. |
| 2/ Los modelos de muestra             | Hacia el lado derecho de la parrilla hay dos dibujos que muestran el carácter que se está modificando, de ambas formas, la normal y la inversa. Estas muestras no cambian mientras se está trabajando en el cuadrado mayor, pero se pueden ver los resultados al apretar la tecla R para Redibujar.                                     |
| 3/ El área de muestra de estatus      | Aquí se muestra el carácter actual, el set y un sumario de todas las órdenes disponibles  |

Para modificar un diseño, usa las teclas del cursor (CAPS SHIFT 5 al 8 en un Spectrum de 48K) para mover el cursor parpadeante.

El estado en que se encuentra el bit debajo del cursor puede ser cambiado en cualquier momento usando la tecla ESPACIO. Con ENTER se devolverá el carácter corregido de nuevo a la base de datos.

Con Q abandonará la edición dejando el carácter inalterado.

### **Impresión y Copia de sets = P o C**

Con la opción P los gráficos definidos por el usuario, los sombreados y los caracteres insertados son impresos en la pantalla.

Si se seleccionó C, entonces se hace una copia a la impresora. Ver la sección sobre impresoras para más detalles.

### **Load y Save de los sets = L set o S set**

Si un set ha sido insertado, entonces se pide un nombre de fichero. La décima letra es convertida al signo " " y el set es cargado o grabado según se especifique.

El cargar o el grabar el set 0 actúa también en los 286 bytes de los gráficos definidos por el usuario y los sombreados.

**Nota 1.** No se puede corregir un carácter de un set, ni tampoco cargar un set, hasta que éste haya sido insertado.

**Nota 2.** Las bases de datos para cualquier otro set (excepto 0), son de tipo estándar de 768 bytes.

Pueden ser cargadas y definidas desde cualquier otro diseñador de caracteres, mientras el nombre de la base de datos termine con el símbolo en la posición 10.

## **XVII. COMPRESOR DE TEXTO**

La pregunta ¿Comprimir base de datos (S/N) ? aparece en pantalla. Si la respuesta no empieza con "S" se asume que es negativa y causa una vuelta al menú principal.

De otra forma un diccionario de 222 bytes se inserta en la base de datos.

Se busca entonces en la base de datos por 90 agrupaciones de letras de las más comunes y éstas son reemplazadas con un token de un solo byte de los usados en las posiciones 165 a 255.

Un contador de pantalla muestra el número de tokens que aún quedan.

Este método de compresión reduce cualquier texto del juego en aproximadamente un 40%. Esto en un juego de texto solo de 128K, nos dará un equivalente de 160K de memoria libre.

**Nota 1.** Cuando se hace una edición de texto después de una compresión, el cursor saltará a veces 2, 3, 4 o 5 caracteres debido a la tokenización. Las correcciones deben volver a ser tecleadas completamente de nuevo, puesto que serán recomprimidas la próxima vez que el compresor se use.

## **XVIII. GRABAR/SALVAR BASE DE DATOS**

El área de la base de datos se pasa a cinta como una secuencia de ficheros, dos por página. La décima posición del nombre del fichero se marcará con las letras de la A a la L secuencialmente.

## **XIX- VERIFICAR BASE DE DATOS**

Los ficheros de una base de datos grabada/salvada previamente en cinta son repasados para buscar diferencias con la base de datos actualmente en memoria.

## **XX. CARGAR BASE DE DATOS**

Los ficheros de una base de datos que ha sido grabada/salvada previamente en cinta se cargan, borrando cualquier base de datos que esté actualmente en memoria.

### **Muy Importante**

Si se pulsa la tecla BREAK u ocurre un error de carga de cinta durante la carga, entonces la base de datos que esté actualmente en la memoria se corromperá y no debe ser usada, puesto que puede a su vez dañar el Editor y el Intérprete.

En estas circunstancias, la única opción que se puede usar con seguridad es: "cargar base de datos" y esto es lo que se debe hacer hasta que una base de datos sea cargada satisfactoriamente.

## XXI. GRABAR/SALVAR AVENTURA

El Intérprete y la base de datos son pasados a cinta como una secuencia de ficheros usando el nombre de fichero que se haya especificado.

Están guardados en cinta de tal forma que la aventura hará una auto-carga cuando se cargue desde BASIC usando LOAD"", sin que el Editor de PAW esté presente.

## XXII. VERIFICAR AVENTURA

Verifica que la aventura haya sido grabada/salvada correctamente.

## EL EDITOR GRAFICO

### Descripción detallada del Editor Gráfico

Esta sección de PAW permite llevar a cabo varias operaciones en forma de una cadena de dibujo para cada localidad. Al editar, la cadena se sitúa en la memoria de la siguiente forma:



## CARACTERISTICAS

Se usa una línea "elástica".

El "punto" es el punto base de esa línea y muestra el último punto al que se ha movido.

El punto final de la línea elástica muestra la próxima posición del "punto" o el lugar de comienzo de un Fill/Sombreado, etc.

Hay cuatro grupos de órdenes. Todos requieren el uso de SIMBOL SHIFT (a menos que se especifique lo contrario).

### 1) ORDENES DE DIBUJO

A = ABS MOVE (MOVIDA ABSOLUTA) (3 Bytes)

Mueve el "punto" a la posición x, y del final de línea, seteando sólo los atributos.

Esto se codifica como si fuese un PLOT con el Inverse y el Over encendidos.

P = PLOT (3 Bytes)

Setea el píxel al final de la línea de acuerdo con su situación de Inverse y Over.

Luego mueve el "punto" a esa posición.

R = REL MOVE (MOVIDA RELATIVA) (3 Bytes)

Mueve el "punto" al final de la línea sin afectar la pantalla. Codificada como una retirada relativa desde el punto antiguo (Movida Relativa).

L = LINEA (3 Bytes)

Dibuja (o fija) una línea recta desde el "punto" al final de la línea, de acuerdo con su situación de Inverse y Over. Luego mueve el "punto" al final de la línea. La línea está codificada como una retirada relativa desde el punto antiguo.

F = FILL (RELLENO) (3 Bytes)

El área desde el final de la línea (Relativa) se rellena usando píxeles sólidos.

Funciona pasando un entretejido sobre una rutina de sombreado (Shade). Así que también se aplica aquí lo que se hable sobre SHADE.

S = SHADE (SOMBREADO) (4 Bytes)

El área desde el final de la línea (Relativo) se sombrea con uno de entre los varios diseños escogibles. La base de datos contiene 16 (0-15) y pueden ser cambiados mediante el Editor de Caracteres.

La trama usada para sombrear viene determinada de la siguiente forma:

- Se te preguntan 2 números de trama desde el 0 al 15. Si solamente quieres usar una, entonces debes repetir el mismo número para ambas tramas.
- Las 2 tramas especificadas sufren entonces un proceso de OR (se colocan una encima de la otra).
- Si el INVERSE estaba encendido (ON), entonces la trama resultante será invertida, (es decir, se cambiarán todos los píxeles que estén seteados a píxeles reseteados).

**Nota 1.** El sombreado primero irá en una dirección hacia abajo en la pantalla, y luego de nuevo en dirección hacia arriba. Para una mayor velocidad cuando se está sombreado hacia abajo, no volver nunca hacia arriba hasta que se llegue al final. Cualquier área sin sombrear debe de hacerse otra vez por separado. Esto se puede evitar escogiendo cuidadosamente el punto inicial.



**Nota 2.** Si el área que va a ser sombreada es muy compleja (en el sentido de que ya tiene muchos píxeles seteados en ella), entonces no se podrá sombreada. Esto se ha tenido que hacer así para que el programa pueda "detectar" si el área elegida ya ha sido previamente sombreada o no. Por eso un área solamente se puede sombreada una sola vez.

No debes sombreada un área y luego intentar llenar el fondo con el comando fill; para ello se debe usar la opción INVERSE.

T = TEXTO (4 Bytes)

Si se han insertado sets de caracteres en la base de datos, entonces se pide un número de set. Luego se pide un código de carácter comprendido entre 32 a 162. Este carácter será situado en la pantalla en el cuadrado de caracteres (que se mostrará con el comando GRID), en el cual esté situado la punta de la línea.

De esta forma podemos poner letras en un dibujo usando el Editor de Caracteres, lo que consume menos memoria que hacerlo por medio de muchas líneas.

B = BLOQUE (5 Bytes)

Hace que un bloque de los colores, que estén seleccionados en ese momento, llene un rectángulo del cuadrado de atributos que tengan la "línea" por diagonal.

## 2) ORDENES PARA EL COLOR

X = TINTA (1 Byte)

Da a la tinta usada el valor que haya sido seleccionado. TINTA 8 (como en BASIC) hace que la tinta sea tomada de los atributos existentes en la pantalla.

C = PAPEL (1 Byte)

Pone el color del papel en el valor que hayamos seleccionado. El PAPEL 8 se comporta lo mismo que en el BASIC.

V = FLASH (1 Byte)

Se pide el nuevo valor de Flash o parpadeo (puede ser 0, 1 u 8).

Z = BRILLO (1 Byte)

Se pide el nuevo valor para Brillo (0-1-3).

I = INVERSE

El estado de Inverse se pone en ON o en OFF.

O = OVER

El estado de Over se pone en ON o en OFF.

Los comandos Inverse y Over no usan memoria, pero su estado es codificado como parte de cualquier futura instrucción que sea afectada por ellos.

## 3) ORDENES DE SUBROUTINA

G = GOSUB (2 Bytes)

Este comando pide un número de dibujo, desde 0 hasta donde haya número de localidad disponible. Luego pide un valor de una escala para ese dibujo, esta escala

está dividida en octavos desde 0 a 7.  
 0 significa 8/8 = no escala.  
 4 significa 4/8 = la mitad.

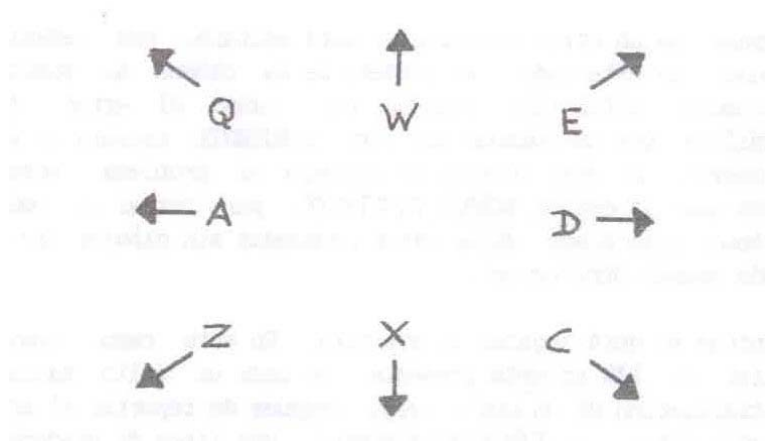
- Ojo Notas:**
- a) Escala solamente afecta a algunos comandos, y son: MOVIDA RELATIVA, LINEA, FILL o RELLENO, SHADE o SOMBREADO. No son afectados ni relocalizados: MOVIDA ABSOLUTA, PLOT, BLOCK y TEXTO, por ello no deben ser usados estos comandos en subrutinas (aunque con la práctica se pueden llegar a hacer funcionar algunas veces).
  - b) Sólo se pueden anidar llamadas de subrutinas hasta un nivel de diez (anidado quiere decir llamar una subrutina desde dentro de otra subrutina).
  - c) Escala no afecta al comando GOSUB. Por ejemplo: si GOSUB es usado en una subrutina la cadena dibujada será de tamaño fijo y no en escala.
  - d) El llamar a la misma rutina con la cual se está trabajando causará un error de "Límite Alcanzado". El mismo error se dará cuando se ha alcanzado el nivel de 10 subrutinas anidadas.

#### 4) ORDENES DE EDICION

- (CAPS SHIFT + 8) COMIENZO Lleva el puntero al inicio de la cadena.
- ↓ (CAPS SHIFT + 6) SIGUIENTE Ejecuta el siguiente comando de la cadena, si no lo hay será ignorada
- ↑ (CAPS SHIFT + 7) PREVIO Mueve el puntero hacia atrás una orden y pone al día la pantalla.
- BORRAR (CAPS SHIFT + 0 en 48k) Borra el comando previo de la cadena y pone al día la pantalla.
- BORRAR SIGUIENTE (GRAFICS) (CAPS SHIFT + 9 en 48K) Borra el siguiente comando, si lo hay.

- Y = RETICULA Pone una retícula del tamaño de un carácter, TINTA 0, PAPEL 7 y PAPEL 6. Esto permite conocer exactamente las fronteras de color, cosa a tener muy en cuenta cuando se colorean dibujos.
- J = JOYSTICK Pone o quita la opción de Joystick Kempston.

**TECLAS DE MOVIMIENTO:** Las teclas alrededor de la (S) mueven el final de la línea píxel a píxel. Esto se puede acelerar a movimientos de 8 en 8 usando el CAPS SHIFT:



## COMANDOS DEL JOYSTICK

El joystick debe enchufarse en el Port 2 del Interface 2 o del Spectrum Plus 2. Alternativamente se puede enchufar en un Interface Kempston, en cuyo caso debe usarse SYMBOL SHIFT + J. CAPS SHIFT también acelerará la velocidad de movimiento del joystick. El botón de fuego del joystick actuará como SYMBOL SHIFT y L para dibujar una línea.

## MENSAJE DE ERROR DEL EDITOR Y SU SIGNIFICADO

Debido a la complejidad de PAW, muchas órdenes pueden generar errores.

Cuando un error es descubierto, un mensaje que lo describe se imprime en la parte inferior de la pantalla. PAW, entonces espera a que se pulse una tecla antes de retornar directamente al menú en el cual ocurrió el error.

Hay dos excepciones a esto:

- Después de un error mientras se está editando una cadena de dibujo. En este caso, el puntero de la cadena se posiciona justamente antes del comando que causó el error. (Esto significa que un comando del tipo SIGUIENTE, causará el error de nuevo). Si eres incapaz de corregir el problema, entonces debes usar la opción BORRAR SIGUIENTE, para borrar el comando erróneo. Esto puede dejar otros comandos sin dibujar (lo cual puede causar otro error).
- Mientras se está jugando la aventura. En este caso, como el Editor de PAW no está presente, se hace un salto hacia la inicialización de un nuevo juego, después de reportar el error. De otra forma, si PAW está presente, una línea de diagnóstico se imprime en la parte inferior de la pantalla. Esta línea muestra la tabla de Procesos, Verbo y Nombre de la entrada, y el conducto en el cual ocurrió el error. A continuación se da la oportunidad de ver un diagnóstico de bandera, lo que ayuda a confirmar dónde ocurrió el error. Cuando ENTER se tecléa, el reporte normal de errores se imprime y se hace una vuelta al menú principal de PAW.

Cuando uses tus propios programas junto con el comando EXTERN, cualquiera de los errores típicos del BASIC puede ocurrir. Hay que hacer notar, sin embargo, que la línea y el número donde ocurrió el error no se imprimen, por lo tanto hay que probar el programa a fondo antes de que la aventura esté presente.

### Errores propios del PAW:

BREAK	Indica que la tecla BREAK ha sido presionada durante una operación o mientras el juego estaba jugándose.
STOP en INPUT	Indica que el CAPS SHIFT + 6 ha sido teclado.
Tape loading error	Igual que en BASIC. Recordamos que un error de este tipo durante la carga de una base de datos, significa que la base de datos está corrompida y solamente se deben de intentar hacer nuevas cargas de bases de datos.
Base de Datos full	Indica que no hay suficiente espacio en la base de datos para lo que se intenta hacer.
Limite Alcanzado (Limit Reached)	Nos avisa de que el máximo número de localidades, mensajes, objetos, tablas de Proceso o páginas de RAM, ya están presentes. También puede indicar que se ha alcanzado un máximo de subrutinas o de subprocesos = 10. Además, puede ocurrir si se intenta hacer un comando DOALL mientras hay

un bucle DOALL activo.

- Integer out of range** Indica que, mientras se está dibujando un gráfico, un comando del tipo LINEA se ha salido de su rango. Esto pasa normalmente debido a un cambio de posición del "punto" de comienzo mientras se está editando.
- Out of memory** Esto indica que la entrada para un proceso o una conexión es muy larga para el buffer de PAW (bastante difícil que ocurra). También puede ocurrir cuando se intenta comenzar una nueva página de RAM en un Spectrum de 48K, o cuando se intenta cargar un juego de 128K en un Spectrum de 48K, o si no se ha dejado suficiente espacio para trabajo, (por ejemplo, si se ha escrito un programa muy grande para ser usado con el comando EXTERN).
- Invalid Argument** Este error se imprimirá cuando un comando descubra un valor ilegal. Por ejemplo: ocurre si se intenta cargar la bandera 38 (la que indica la localidad) a una localidad ilegal. También se puede generar si se intenta poner algo dentro (PUTIN/ TAKEOUT) de un objeto que no es un contenedor, o también si se intenta poner un objeto en la localidad 255 (sólo admite COPYFO). (Véase el punto 2 mencionado anteriormente para la información de diagnóstico).
- Nota 1.** Durante un Input, el Spectrum emitirá un RASP si la pantalla y/o el buffer de input está lleno.
- Nota 2.** Si una entrada ya está presente en la base de datos y es muy grande para que quepa en una sola pantalla, el Spectrum emitirá un RASP cuando la entrada se sitúe en la parte inferior de la pantalla para corregirla (amending).
- Nota 3.** Si una entrada anormalmente larga se inserta en la tabla de conexiones usando abreviaciones. Por ejemplo: N 1-0 6-S 4, etc., y las abreviaciones son después borradas del Vocabulario. Entonces nos encontraremos con que quedan las palabras normales: NORTE, OESTE, SUR, etc., las cuales pueden ser muy largas para el buffer de input. Si esto sucede, un mensaje de "out of memory" se producirá. El remedio es reinsertar las abreviaciones en el Vocabulario.

## PAW Y LAS IMPRESORAS

PAW imprime usando el canal número 3 (usualmente "P").

Si se está usando una impresora térmica tipo ZX (o su equivalente en Alphacom) toda la impresión se dirige a las rutinas requeridas de la ROM.

Hay que tener en cuenta que las impresoras térmicas solamente trabajarán en modo 48K en un Spectrum de 128K, y que imprimirán exactamente lo que está en la pantalla, incluyendo los nuevos sets de caracteres, etc. La opción "C" en los caracteres y menús gráficos harán un COPY normal de la pantalla.

En un Spectrum de 128K, el canal usado para imprimir por defecto es el puerto de RS232, y PAW tendrá esto en cuenta. Toda la impresión se dirige a través del port, pero el set de caracteres se filtrará de la siguiente manera:

- Caracteres del 0 al 31, serán suprimidos excepto por el 7 y el 13 que imprimirán un CR (13) y 6 (TAB) que se imprimirá como un espacio.
- Caracteres del 32 al 127, se imprimirán de una manera normal.
- Caracteres del 128 al 162, se imprimirán como un símbolo de interrogación ("?").
- Caracteres 163 y 164, serán expandidos a las palabras clave PLAY y SPECTRUM.
- Caracteres del 165 al 255, se expandirán hacia sus valores normales en una base de datos incompresión, o hacia el agrupamiento de letras correcto si la base de datos está comprimida.

Si tienes un Interface 1, es posible usar el RS232 para imprimir abriendo el canal "T" con el comando:  
OPEN #3,"t"

antes de cargar el PAW . El canal entonces operará como se describió arriba para el RS232 en un 128K.

En general, es posible usar cualquier Interface de impresora que use el canal 3 correctamente. Si todo falla, y eres bueno en código máquina, puedes escribir un driver específico de impresora. PAW imprime una dirección cuando carga, (llamada: PRTADD).

Esto nos da las direcciones de los vectores dentro de PAW para el sistema de impresoras, definido de la siguiente forma (el carácter está en A):

PRTADD	DEFW	PRTRET	; Dirección del driver de impresora.
COPVEC	JP	DOCOPY	;Salto a la rutina de COPY de pantalla.
PRTRET	RET		;Pon aquí el driver...

Hay espacio libre hasta 29504 para expandir.

Debes grabar tu código de driver desde la dirección correcta, o poniéndole un número de fichero que tenga una "A" en la posición 10.

Junto con él debes grabar un fichero sin bytes con el mismo nombre pero terminado en "B".

Hay que redireccionar entonces el canal número 3, (hacia cualquier otra parte), para hacer creer a PAW que hay una impresora no térmica conectada.

Entonces debe usarse la opción "carga base de datos" del menú principal para cargar tu driver (PAW cree que es una base de datos y cargará ambos ficheros, uno de los cuales será tu driver).

## ENSAYOS

Vamos a tocar ahora nueve puntos que combinan varias características de PAW, difíciles de catalogar. Son:

1. Consideraciones para los 128K
2. Fronteras de color
3. El parser
4. La pantalla
5. Objetos
6. Juegos de múltiples partes
7. Luz y sombras
8. EXTERN y SCREENS
9. PSIs (personajes seudo inteligentes)

### 1. Consideraciones para los 128K.

Cuando se carga el programa, solamente la página 0 está siendo usada. Quedan unos 25.500 bytes libres. Las páginas 1, 3, 4, 6 y 7 están sin usar, pero pueden contener 16K cada una.

El texto para Mensajes y Localidades, Conexiones, Gráficos y los Colores por Defecto, pueden usar las páginas 1 a 7. Todas las demás tablas usarán solamente la página 0. Todas las entradas para la misma localidad estarán contenidas en la misma página.

Cuando se inserta un texto de localidad, se hacen automáticamente entradas en blanco en las tablas de Conexión, Gráficos y Colores por Defecto para esa localidad.

La opción I del menú de Texto de Localidades siempre insertará en la página más alta que esté siendo usada.

La opción B de ese menú se usa para comenzar una nueva página, es decir inserta en la siguiente página.

El usuario debe decidir dónde usar B para comenzar una nueva página, pero sugerimos que uses antes el compresor para ganar toda la memoria posible antes de comenzar otra página, porque luego no se

pueden insertar mensajes extra en una página cuando ya se ha comenzado a insertar en otra página, solamente se pueden corregir las entradas ya existentes.

La base de datos inicial, que viene con PAW, tiene la localidad 0 en la página 0 y hay que decidir pronto si la localidad 1 debe ponerse en una nueva página. Sobre todo teniendo en cuenta que todas las otras tablas solamente pueden utilizar la página 0.

Nosotros recomendamos que todos los mensajes y las localidades se pongan en la página 1 y superiores, puesto que la página 0 pronto se llenará de Vocabulario, etc.

Lo mismo ocurre con la tabla de Textos para Mensajes. I insertará siempre en la página más alta seleccionada. B comienza otra página, sólo si ésta ha sido inicializada poniendo una localidad en ella previamente.

Si deseas que tu juego sirva también para un Spectrum de 48K, entonces no debes usar ninguna otra página excepto la número 0.

## 2. Las Fronteras de Color

Probablemente uno de los aspectos más difíciles de los gráficos en el Spectrum es el problema de los atributos de color.

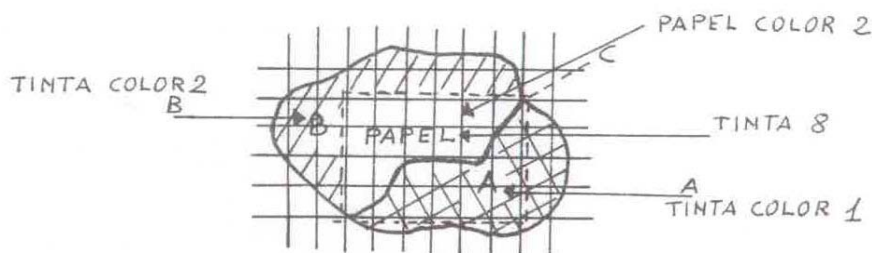
Esencialmente, sólo se pueden utilizar dos colores en cualquier grupo de píxeles de 8x8. Estas áreas se muestran claramente mediante la opción RETICULA en el Editor Gráfico.

Los dos colores son el PAPEL y la TINTA, lo que significa los colores de los píxeles apagados y los píxeles encendidos respectivamente. Ambos PAPEL y TINTA pueden tener hasta ocho valores para cada grupo de píxeles de 8x8.

Las rutinas de RELLENO y SOMBREADO lo que hacen es rellenar un área definida con una trama de píxeles seteados y reseteados. Por lo tanto, si usas un color de TINTA o/y PAPEL diferente para dos RELLENOS o SOMBREADOS adyacentes, obtendrás un efecto bastante desagradable. Ello se debe a que los colores del segundo RELLENO afectan a los píxeles en los grupos donde estaban seteados por el primer RELLENO.

Esta desventaja se puede solucionar de cuatro formas diferentes:

- 1/ Ignorándolos. Es el sistema más sencillo y usado en la mayoría de los dibujos. El efecto extraño se minimiza sencillamente no teniendo dos áreas adyacentes con sombreado o relleno diferentes.
- 2/ Usando el sombreado a tope. El sistema de sombreado puede ser usado solamente con dos colores. Ello basta para crear dibujos bastante atractivos variando la densidad del sombreado.
- 3/ Teniendo mucho cuidado con la posición de los colores, como se explicó en el manual de introducción. En él se ve que todas las áreas han sido definidas respetando las fronteras de los atributos. Pero esto tiene el efecto de que se crean dibujos de tipo bastante cubista.
- 4/ Alternar el color. Esto permite tener dos áreas adyacentes de diferente color sólido (no sombreado). Parece complicado, pero el principio es bastante sencillo: el diagrama lo muestra de una manera más simple que cualquier explicación.



## EXPLICACION DEL DIAGRAMA.

Primero se dibuja la parte de fuera del área A y del área B.

El área A se rellena entonces usando una TINTA del primer color.

El área B se rellena usando una TINTA del segundo color.

Entonces se usa el comando BLOCK en el rectángulo indicado por (C) teniendo cuidado de que el PAPEL que se use sea el del segundo color y que la TINTA sea igual a 8, esto deja el área A y el área B sin ningún cambio.

Y por este método el coloreado del PAPEL hace que la confluencia parezca continua.

## 3. EL PARSER

El parser actúa buscando en una línea de Input, (que puede ser hasta 125 letras), palabras que estén en el Vocabulario, extrayendo de esta forma "frases" que se puedan convertir en Sentencias Lógicas (SL).

Cuando una frase ha sido extraída, las tablas de Respuesta y de Conexiones son revisadas para ver si la Sentencia Lógica es reconocida en ellas.

Si no, entonces el Mensaje del Sistema número 8 "No puedes hacer eso" o el Mensaje del Sistema número 7 "No puedes ir en esa dirección" se imprimirá, dependiendo siempre del valor del Verbo. Por ejemplo, si el Verbo tiene un valor inferior a 14, se usará el Mensaje del Sistema 7, porque los Verbos inferiores a 14 tienen un sentido de dirección. Luego se pide otro nuevo Input.

También se pedirá un nuevo input si cualquier acción falla de alguna otra forma, por ejemplo, un objeto que sea muy pesado, o si el escritor lo fuerza con una acción NEWTEXT.

A veces este salto a pedir otro nuevo input se hace para evitar resultados verdaderamente catastróficos. Supongamos que el jugador haya tecleado "COGE EL HACHA Y ATACA AL TROLL" y no haya ningún hacha a mano; en este caso se haría un salto para impedir que el jugador pueda atacar al troll con las manos vacías, con el consiguiente mordisco en la cabeza y desmembramiento sangriento.

Si la Sentencia Lógica (SL) se ejecuta satisfactoriamente, el parser extrae otra frase, y si no la hay, requiere un nuevo texto.

Las frases pueden ir separadas por las conjunciones "Y" o "ENTONCES" y por cualquier otro signo de puntuación.

**Nota Muy Importante:** En la versión inglesa se utilizaba el pronombre "IT" para referirse al último Nombre o Adjetivo que ha sido usado en la frase previa, aunque fuese un input separado. Esto permitía la facilidad de poder decir "coge la manzana y come it = (ella)".

En la versión castellana se ha tenido el cuidado de modificar esta zona, para que el parser haga una búsqueda a la inversa de las terminaciones de cualquier Verbo, buscando las terminaciones LA, LO, LOS y LAS, permitiendo así que el jugador pueda decir "COGE LA MANZANA Y COMETELA". Esto facilita mucho la forma de jugar, puesto que no tienes que repetir dos veces manzana.

El parser sabrá que todos los Nombres que tengan un valor menor de 50 los utilizará el escritor para nombres propios, y por lo tanto no se verán afectados por esta regla.

Por ejemplo, en la frase "COGE LA MANZANA DE EVA Y COMETELA", el parser sabe que se refiere a la manzana si está situada con un número mayor de 50 y Eva con un número menor de 50. La mínima unidad que forma una frase es un Verbo (o un Verbo de conversión, que es un Nombre que tenga un valor de palabra menor de 20, en cuyo caso, si no se encuentra Verbo en esa frase, ese Nombre será convertido en Verbo, es el ejemplo de: NORTE, SUR, etc.).

En caso de que el Verbo sea omitido en una frase, entonces la SL asumirá que se está utilizando un Verbo previamente usado. Por ejemplo: "COGE LA ESPADA Y EL ESCUDO" trabajará perfectamente, porque el parser sabe que lo que estás intentando hacer con el escudo es cogerlo.

Si hay dos o más objetos a los que se refieren las terminaciones "COGELA", "COGELO", etc., se asume por el parser que te refieres siempre al primer nombre. Es decir, funcionará siempre con el primer nombre. Por ejemplo, si dices: "COGE LA ESPADA Y LA DAGA Y TIRALA", como el primer nombre es espada lo que tirarás será la espada.

Examinamos una frase muy larga:

"COGE TODO. ABRE LA PUERTA Y DIRIGETE AL SUR, LUEGO COGE LA MALETA Y MIRA DENTRO DE ELLA".

Esa frase se dividirá en varias SL, como sigue: "COGE TODO",

"ABRIR PUERTA" (porque el artículo no está en el Vocabulario)

"SUR" (porque dirígete no está en el Vocabulario),

"COGE MALETA"

"MIRA DENTRO ELLA" (ese ella se asume que es maleta, puesto que es el último Nombre del que se ha hablado).

#### 4. LA PANTALLA

Los diferentes sets de caracteres se seleccionan en el texto primero seleccionando un color de 0-5 y luego haciendo un solo DELETE (BORRAR), lo que generará una función ESCCs de 0-5 y solamente tendrá efecto si el correspondiente set ha sido insertado.

ESCAPE 7 ejecutará un verdadero salto de línea (pero no resetea los colores, puesto que es parte de un mensaje).

ESCAPE 6 funciona como si fuera un TAB, pero debe ser usado con mucho cuidado, puesto que cuando se haga la impresión en pantalla puede ser suprimido por el formateador de texto.

Los colores y los sets de caracteres seleccionados desde dentro del texto estarán activos hasta que se haga una acción NEWLINE. Por ejemplo, para imprimir el número de monedas que lleva el jugador (supongamos que los vaya controlando por la bandera 100), si el mensaje 1 tiene un código de control de PAPEL ROJO y ejecutamos entonces una entrada en la tabla de Respuestas o de Procesos del tipo de:

```
MONEDAS _      MES      1 ;      ; "[color ROJO] llevas"
                PRINT    100
                MESSAGE  2      ; "monedas"
                DONE
```

Resultará que el mensaje entero se imprime de color rojo.

Es lo que se llama un mensaje compuesto, puede, y de hecho debe ser usado para crear efectos mucho más atractivos en la pantalla.

Hay que hacer notar que la orden MES deja que el próximo mensaje se imprima justo después de él; es decir, no lleva implícito un NEWLINE como la orden MESSAGE.

#### 5. OBJETOS

La raya baja en el texto será convertida durante el juego a la descripción del último objeto al que se refiera una acción COGER, DEJAR, DESTRUIR, etc.

Esto se hace principalmente por el hecho de que los Verbos COGER, DEJAR, etc., hacen una respuesta automática de su fallo o de su éxito. Por ejemplo, cuando tecleas "COGER CUCHILLO", el parser responde "YA HAS COGIDO EL CUCHILLO". Pero también puede ser usado para examinar objetos y multitud de otras funciones automáticas.

La bandera 53 se usa para controlar la forma en que los objetos se exponen en pantalla cuando las acciones LISTOBJ y LISTAT son usadas. Si esa bandera 53 está seteada a 64, (es decir, tiene el bit 6 seteado, lo cual se obtiene con un comando LET 53 64), entonces todos los objetos serán listados sin líneas entre ellos, formando una sentencia válida; es lo que se llama listado compuesto.



En cuanto a los Mensajes del Sistema implicados en estos procesos tenemos:

MS53 ("nada") - sólo puede ocurrir con la orden LISTAT

OBJETO y MS48 (".")

OBJETO y MS47 ("y") OBJETO MS48 (".")

OBJETO y MS46 (",") OBJETO MS47 ("y") OBJETO MS48 (".")

Además, el Bit 7 de la misma bandera (53), estará seteado (la bandera será mayor de 127, o sea 128) si cualquier objeto se imprime. Esto te permitirá determinar si quieres o no un NEWLINE para resetear los colores.

Una acción del tipo LISTAT debe ser precedida usualmente por un mensaje.

La descripción de OBJETO se reconstruye a partir de la descripción completa que se ha puesto en la tabla de texto de objetos.

En nuestra versión, y debido a que el artículo THE inglés tiene cuatro acepciones en castellano (el, la, un y una) hemos tenido que hacer una importante modificación:

Como bien sabes, cada vez que se imprima un mensaje que incluya el carácter de subrayado (SIMBOL SHIFT + 0). en lugar de éste, aparecerá la descripción del último objeto referenciado.

Así, al teclear "Coger botella" teniendo en la tabla de Respuestas una entrada del tipo de:

```
COGER _   AUTOG
          DONE
```

daría como resultado "Ya tienes el botella" (suponiendo que tengas la botella en tu poder).

Para evitar el cambio de género de los objetos podemos utilizar el siguiente truco:

En los Mensajes que hagan uso del subrayado, intentaremos evitar el uso de artículos *antes* del mismo. Por ejemplo: "Ya tienes\_" (sin espacio entre "tienes" y el guión).

Luego en la descripción de los objetos " la botella".

El primer carácter ( ) deberá ser el 128, o sea, un blanco en modo gráfico. (CAPS + 9, 8 y CAPS + 9, otra vez).

El segundo carácter será un carácter normal de espacio, y a continuación el texto.

De esta manera, el mensaje impreso será: "Ya tienes *la* botella", y habrás logrado la diferenciación entre objetos masculinos y femeninos.

**Nota Importante:** Si el objeto va a ser un contenedor, debes dejar una localidad sin usar con *el mismo número* para que PAW la use como la parte "interior" del objeto.

Por ejemplo, el objeto número 1 necesitará la localidad número 1. No te olvides de marcarlo como contenedor "C" en la tabla de Pesos.

## 6. AVENTURAS MULTIPLES

Si deseas crear aventuras más largas, con un área mucho más grande de juego, sin sacrificar la calidad de la descripción, puedes dividir el juego en secciones mucho más pequeñas.

Esto es mucho más fácil si se ha planeado el juego con áreas de unión entre zonas. Es lo que se llama un cuello de botella.

Por ejemplo, un juego en el cual la primera parte termine en que debes subirte a un barco y echarte a navegar. Así, el juego es mucho más fácil de dividir.

Para que la puntuación, los turnos que has hecho, y otra información, se puedan llevar de una parte del juego a la siguiente, debes usar acciones que te permitan CARGAR/GRABAR (load/save) las posiciones del juego.

Para poder CARGAR una posición de juego en un juego diferente de aquel en el cual fue GRABADO, necesitas tener el mismo número de localidades y de objetos en cada parte.

Además, todos los objetos que pueda llevarse el jugador de una parte a otra deben tener la misma descripción en todas las partes.

Supongamos que tenemos un juego de 120 localidades que vamos a partir por la mitad. Ello requiere 60 localidades en cada parte.

De hecho, la localidad 60 existirá en ambos juegos como localidad de transición (es decir, donde el jugador empieza la segunda parte y termina la primera) y también necesitamos una bandera libre, (digamos la 26), que será usada para indicar de qué parte del juego viene la posición que se ha GRABADO.

Así pues, cuando el jugador complete la primera parte, se mueve a la localidad 60 y la bandera 26 se setea a 1 para marcarlo.

La primera parte podría ser de la siguiente forma:

Localidad 60

Final de parte 1 Prepara una cinta para grabar tu posición.

(Puedes grabar más de una copia si lo deseas).

Por favor CARGA (load) parte 2 y sigue las instrucciones que se dan en la pantalla.

### Procesos 1

Fin _	AT	60		; ¿Final del juego?
	LET	26	1	; Posición válida de la parte 1
	SAVE			

### Y en la parte 2:

Localidad 0

Parte 2 Preparado para cargar de cinta la posición grabada.

Localidad 60

Pon aquí, cualquier introducción que se requiera de la Parte 2.

### Procesos 1

INICIO _	AT	0		;¿Es el comienzo?
	LOAD			; Entonces debe ser otra localidad
INICIO _	NOTEQ	26	1	;No es una posición válida de la Parte 1
	GOTO	0		;Pide otra localidad
	DESC			
INICIO _	AT	60		;Acabas de cargar una posición válida
	ANYKEY			; Esperar hasta que se lea introducción
	RESET	1		;Para comenzar el juego en la localidad 1

La acción RESET hace una descripción de la nueva localidad de comienzo automáticamente, después de poner todos los objetos que no estaban siendo llevados, o puestos encima o estaban en la localidad 60, en sus localidades de comienzo.

Hay que tener en cuenta que se debe insertar una acción de tipo CLEAR para las banderas entre la acción ANYKEY y RESET, puesto que las banderas no se ven afectadas por la orden RESET.

## 7. DIA Y NOCHE (LUZ Y OSCURIDAD)

La oscuridad se está haciendo muy común en las aventuras actuales, y de hecho, usada correctamente puede añadirle una sensación de realismo bastante fuerte al juego.

Dentro de PAW la oscuridad se crea **SETEANDO LA BANDERA 0 A UN VALOR DIFERENTE DE 0**. ES DECIR:

BANDERA 0 IGUAL a 0 = LUZ

BANDERA 0 DIFERENTE a 0 = OSCURIDAD

Estos cambios deben hacerse siempre que un jugador se mueva hacia o desde una zona de oscuridad, y este movimiento debe hacerse mediante una acción GOTO en la tabla de Respuestas, para permitir el poder SETEAR o CLEAR de la bandera.

Si el jugador tiene una fuente de luz, entonces esta fuente de luz SERA SIEMPRE EL OBJETO 0 (Fuente de luz = Objeto 0).

Una fuente de luz no tiene por qué ser siempre una antorcha o una llama, si tienes un poco de imaginación, pueden ser unas gafas infrarrojas o un láser de luz amplia.

Vamos a tomar como ejemplo la creación de un ciclo de día y de noche, usando 24 espacios de tiempo (que asumimos que son equivalentes a 1 hora cada uno).

Las entradas requeridas en la tabla de Procesos 2, son:

–	CICLO	EQ	5	0	; Final del ciclo
		LET	5	24	; Nuevo comienzo del contador
–	NOCHE	EQ	5	18	; Caída de la noche
		SET	0		; es decir, la bandera 0 no es = 1, por lo tanto no habrá luz

MESSAGE x

–	DIA	EQ	5	6	; Salida del sol
		CLEAR	0		; Hacemos la bandera 0 = 0 = luz
		MESSAGE	y		

Es importante que tengas en cuenta que si el juego tiene una parte bajo tierra, o dentro de un edificio, hay que determinar si el jugador actualmente puede ver la salida del sol o la puesta de sol desde donde está, antes de imprimir estos mensajes.

## 8. EL COMANDO EXTERN Y LA CARGA DE PANTALLAS

El comando EXTERN puede ser usado para llamar tus propios programas en código máquina o a una sección de un programa en BASIC. Esta opción puede ser utilizada solamente en el juego final, porque las 4K de memoria ocupadas previamente por el menú principal se harán usables. (Las otras 2K que quedan es el lugar de trabajo de PAW, así que ni tocarlo).

La dirección importante es el valor que se imprime cuando PAW se carga (llamado EXTVEC). Debes usar siempre el valor que se da en la copia de PAW desde la cual grabaste el juego final, o tendrás el caos asegurado.

### PARA LOS ENTENDIDOS EN CODIGO MAQUINA:

Para que el comando EXTERN pueda ejecutar rutinas de código máquina, debes hacer un POKE de tres bytes en el EXTVEC con un JP hacia tu dirección de comienzo. El registro IX debe ser preservado, puesto que está apuntando "hacia" la bandera 0. Todos los demás pueden ser destruidos, pero SP debe de ser preservado.

El registro A, de entrada contiene el parámetro hacia el comando EXTERN. Si grabas los dos bytes en EXTVEC + 1, ellos apuntarán hacia la rutina que llama el programa de BASIC. El número de línea debe de ponerse en A.

### PARA LOS USUARIOS DE BASIC:

Un POKE a EXTVEC, 195 (después de haber cargado el Intérprete) permitirá que cualquier acción EXTERN ejecute la línea de BASIC (100 + parámetros), por ejemplo, EXTERN 10 irá a la línea 110 en BASIC. Para volver al siguiente condacc debe ponerse un comando de tipo STOP en BASIC.

No se puede usar el CLS en BASIC (o en código máquina), pero la dirección EXTVEC + 3 contiene una rutina que ya lo hace, por ejemplo: RANDOMIZE USR extvec+3 hará un clear de la pantalla. Además, EXTVEC +12 contiene el número de línea donde se imprimen los nombres de las bases de datos y EXTVEC +9 contiene el comando que carga la base de datos y empieza el juego. Como ejemplo, tu siguiente programa (sin los REM's; debe ser cargado en una cinta virgen por medio de SAVE "el nombre del juego" LINE 10:

```
10 REM cargar el juego
15 BORDER 0:PAPER 0:INK 0:REM para que el nombre sea invisible
20 CLEAR 28671:REM este es el máximo valor de la RAMTOP
25 LET extvec=????:REM cualquiera que sea el extvec
30 LOAD " " SCREENS:PRINT AT 19,0;:REM impide que se dañe el dibujo
40 LOAD " " CODE:REM carga el fichero del Intérprete
50 POKE extvec, 195:REM pone en marcha el EXTERN
60 POKE extvec+12,20: REM los nombres de la línea 20
70 RANDOMIZE USR (extvec+9) :REM carga la base de datos y comienza el juego
100 REM es la línea llamada por medio de EXTERN 0
110 PRINT "hola desde BASIC":STOP
355 REM es la línea que va a ser llamada por EXTERN 255
```

Las líneas 100-355 pueden contener comandos del tipo GOTO si es necesario.

Graba entonces tu SCREEN\$ inmediatamente después del programa en la cinta. Luego, desde dentro del mismo PAW graba la aventura (usando la opción A).

**Importante:** deben quedar por lo menos 2.000 bytes libres para el área de trabajo de PAW, o tu programa de BASIC empezará a desaparecer. La RAMTPOP puede ser bajada para que contenga tu código máquina, pero deben quedar 2K libres.

## 9. PERSONAJES SEUDO INTELIGENTES (PSI)

El punto fundamental a tener en cuenta cuando se maneja un personaje seudo inteligente, es que todo el proceso será llevado por: Una palabra del Vocabulario (usualmente un Nombre con un valor menor de 50 para que sea Nombre propio). Algunas banderas. Una serie de mensajes. Algunas entradas en una o más tablas de procesos.

La bandera debe mostrar en todo momento dónde está el personaje, los mensajes nos dan información sobre sus acciones y las tablas de Procesos son las que mueven todo el cotarro.

Así que imaginemos un personaje llamado "Archivero", que puede andar por todo el juego independientemente.

Hay que describirlo en el Vocabulario como ARCHIVERO (esperamos que por ahora ya sepas que el parser sólo aceptará las primeras cinco letras, pero ahora usaremos la palabra completa para facilitar la comprensión), y le damos un valor de 25 y la categoría de Nombre.

La bandera número 20 se usa para mantener su localidad.

Escogeremos la tabla de Proceso 3 para las entradas que tengan que ver con lo que se le habla y las Repuestas.

La tabla de Proceso 4 nos servirá para controlar sus movimientos y sus acciones.

Las siguientes entradas le permitirán a él moverse por todo el juego cuando así se lo pida el jugador. Y haremos que después de algún tiempo el Archivero se aburra y desaparezca en una nube de humo.

### Mensaje 1

El Archivero parece no entenderte.

### Mensaje 2

No hay nadie con ese nombre por aquí.

### Mensaje 3

El Archivero te dice "Hola" con una tenebrosa voz.

### Mensaje 4

El Archivero va a su marcha, puesto que no encuentra ninguna cosa mejor que hacer.

### Mensaje 5

El astuto Archivero está aquí.

### Mensaje 6

El Archivero pasa olímpicamente de lo que tú dices.

### Mensaje 7

El Archivero te mira y con su horrorosa voz dice "Estoy aburrido de todo esto, me voy a un juego más entretenido", y se desvanece en una nube de humo verde.

Esos son los mensajes con los que trabajaremos.

Primero, la presencia del Archivero en una localidad debe ser anunciada, por eso en la tabla de Procesos 1 (la cual es llamada inmediatamente después de describir una localidad) chequearemos si él está aquí.

Por ejemplo: que la bandera 20 (su localidad) sea la misma que la bandera 38 {nuestra localidad}.

Es importante que nos demos cuenta de que no debe estar en la localidad 0, puesto que ésta es la pantalla de introducción.

Entonces, la tabla de Procesos 1 queda de la siguiente forma:

ARCHIVERO _	SAME	20	38	; Confirmar que esté aquí
	NOTAT	0		;Que el jugador no esté en la localidad 0
	MESSAGE	5		; Donde se avisa que está aquí

Para poder hablar con el Archivero necesitamos dos entradas en la tabla de Respuesta, de la siguiente manera:

DECIR ARCHIVERO PREP	A			;esto último (PREP A) puede ser omitido si solamente queremos que acepte la forma de hablar tipo indio "Decir Archivero", si ponemos PREP A también se puede decir "Decir al Archivero" (lógicamente, debes tener A y AL como sinónimos).
	SAME	20	38	;para asegurarnos de que el Archivero esté aquí
	PROCESO	3		;es el que se encarga de hablar
	DONE			;esto se pone para impedir que la acción caiga a una nueva SL
DECIR _	PREP	A		;esto es opcional
	MESSAGE	2		;que dice que no hay nadie con ese nombre
	DONE			

Por su parte en la tabla de Procesos 3, hay que poner las siguientes entradas:

* *	PARSE			;es la que convierte cualquier línea de "input" a una SL
-----	-------	--	--	--

	MENSAJE	1		;es donde caerá la acción si PARSE no encuentra una frase válida
	DONE			;esto es porque la SL no es la adecuada y no se deben ejecutar más entradas
HOLA _	MENSAJE	3		; suponiendo que HOLA sea verbo en el Vocabulario
	DONE			;para que al decir "HOLA" al Archivero funcione
_ _	LT	34	14	;¿se le dijo al Archivero una palabra que indica movimiento?
	MOVE	20		;ver si hay una conexión para esa palabra
	MENSAJE	4		; venir aquí y decirle al jugador que se ha hecho eso
	DONE			
_ _	MESSAGE	6		; donde te ignora

Es obvio que se deben poner muchas más entradas para darle al Archivero más originalidad y vocabulario. Pero ya te habrás dado cuenta de que con entradas de este tipo bien hechas, se le puede dar a un personaje una apariencia bastante normal (aunque el Archivero no es normal). Finalmente, para darle al Archivero la oportunidad de que desaparezca cuando está aburrido vamos a la tabla de Procesos 4 donde pondremos:

ARCHIVERO _	EQ	20	2	;¿Está en la localidad 2?
	CHANCE	10		;le damos un 10% de oportunidad
	SET	20		;Localidad 255 no existe, y al setear la 20 la hacemos = 255
	AT	2		;¿y donde estaba él?
	MESSAGE	7		;es donde el bordejo desaparece.

También, en la tabla de Procesos 2 haremos una entrada para que regularmente llame a la tabla 4, sería:

ARCHIVERO \_ PROCESO 4

De ésta forma, se puede hacer un PSI muy convincente.

Tengamos en cuenta que le añade al juego mucho realismo, especialmente si se requiere interacción con ellos para poder resolver los problemas que en el juego se pongan.

### \*\*\* SUMARIO \*\*\*

## 1. ORDENES DEL EDITOR GRAFICO

Ocasionalmente notaremos que al mover el cursor a velocidad rápida puede desaparecer; esto puede ser debido a que la frecuencia de barrido del monitor coincide con la velocidad de impresión de la línea en pantalla.

Se mueve el final de la banda elástica con:

Q	W	E	un píxel por movida. También se puede usar
A	D		el Interface 2/Plus 2, joystick port 2,

Z X C

o el Interface Kempston (SIMBOL SHIFT + J)

Se puede acelerar el movimiento de 8 en 8 píxeles con CAPS SHIFT.

Los comandos de edición:

Cursor Derecha	(CAPS SHIFT & 8)	Comienzo de la línea de dibujo.
Cursor Abajo	(CAPS SHIFT & 6)	Nueva orden
Cursor Arriba	(CAPS SHIFT & 7)	Comando previo
DELETE	(CAPS SHIFT & 0)	Borra el comando previo
GRAFICOS	(CAPS SHIFT & 9)	Borra el siguiente comando

Los siguientes comandos requieren SIMBOL SHIFT a menos que se especifique lo contrario:

I	Causa un Inverse
O	El Over
Y	El Grid, o sea, la Retícula
J	El joystick Kempston

Todos pueden estar encendidos o apagados si se teclean una o dos veces.

Las siguientes órdenes se insertan dentro de la Base de datos:

A	Movida Absoluta (un PLOT con I u O seleccionado)
B	Bloque en un rectángulo del área de atributos. Diagonalmente definido por línea.
F	Rellena el área desde el final de la línea (Ver nota a pie de página)
G	Gosub al número de localidad con escala
X	Tinta (pide un número para tinta)
L	Fija la línea (dibuja ya la línea elástica)
C	Selecciona el Papel
E	Hace un PLOT de un punto al final de la línea
R	Movida Relativa al final de la línea
S	Sombreado de un área de la pantalla desde el final de la línea y con la trama elegida V Selecciona el Parpadeo
Z	Selecciona el Brillo
T	Pone texto en la pantalla, selecciona el carácter y el set

**Nota:** RELLENO (Fill) y SOMBREADO (Shade) no son completamente iguales. El Rellenado es un Sombreado con todos los píxeles seteados.

ENTER se teclaea para finalizar la sesión de edición.

## 2. MENSAJES DEL SISTEMA

Aviso: La mayoría de los Mensajes del Sistema contienen un ESCAPE 7 para permitir un espaciado correcto en la pantalla. (ESCAPE 7 causará una línea en blanco).

- MS0 - usado en vez de la descripción de la localidad cuando está oscuro.
- MS1 - impreso por LISTOBJ si hay por lo menos un objeto presente.
- MS2 al MS5 - son cuatro formas de input (petición de movida al jugador) que se seleccionan aleatoriamente a menos que la bandera 42 se ponga como un mensaje válido.
- MS6 - es producido por el parser cuando no se encuentra ninguna otra frase entendible.

- MS7 - es producido si no se lleva a cabo ninguna acción (o una acción NOTDONE fue ejecutada) en la tabla de Respuesta cuando el Verbo sea menor > 14.
- MS8 - se produce si no se lleva a cabo una acción (o la acción NOTDONE fue ejecutada) en Respuesta si el Verbo es mayor < 13.
- MS9 al MS11 - se imprimen por medio de la acción INVEN.
- MS12 - impreso por QUIT (abandonar).
- MS13 y MS14 - impresos por la acción END.
- MS15 - el mensaje que responde a la acción OK.
- MS16 - el mensaje de la acción ANYKEY.
- MS17 al MS20 - son mensajes que corresponden a la acción TURNS.
- MS21 y MS22 - mensajes que corresponden a la acción SCORE.
- MS23 al MS29 - son los primeros de los muchos mensajes producidos por acciones que manipulan los objetos.
- MS30 - la respuesta positiva que se espera por la acción END y QUIT.
- MS31 - la respuesta negativa que se espera por la acción END y QUIT.
- MS32 - se produce cuando aparece una pantalla llena de texto.
- MS33 - es donde se marca el input.
- MS34 - el cursor.
- MS35 - aparece mientras se toma un tiempo muerto.
- MS36 al MS45 - son más mensajes producidos por las acciones que manipulan los objetos.
- MS46 - es el lazo de unión entre objetos cuando se están listando continuamente.
- MS47 - es el enlace final entre los dos últimos objetos cuando se está haciendo un listado.
- MS48 - es la terminación de un listado de objetos. Se usa tanto para LISTOBJ como para LISTAT, así que ten cuidado con él, (generalmente suele ser un punto).
- MS49 y MS50 - aún más mensajes para manejo de objetos.
- MS51 - es la terminación para una sentencia compuesta en PUTIN/TAKEIT (y AUTOP/AUTOT).
- MS52 - es un mensaje final de objetos.
- MS53 - es el mensaje que aparece cuando durante una acción LISTAT no se encuentran objetos.

Desde MS54 hacia arriba, están libres para tu propio uso. Pero PAW en otras máquinas futuras puede usar más mensajes, esto hay que tenerlo en cuenta si se intenta en el futuro transferir la aventura a otra versión. Por ejemplo, PAW en CP/M usa los mensajes 54 a 60 (pero PAW en CP/M de momento no lleva gráficos).

NOTA: Mejor no tocar los Mensajes del Sistema, sino a partir del 60 hacia arriba.

### 3. LAS FUNCIONES DE LAS BANDERAS

Las banderas normales están libres para poder usarlas de cualquier forma en el juego. Las banderas que se auto decrementan automáticamente (de la 2 a la 10) también pueden ser usadas libremente, pero asegúrate de que sabes en qué situaciones se decrementan antes de usarlas.

Hay otras banderas que solamente se pueden setear mediante la acción apropiada, pero sin embargo, se puede hacer un test de su contenido en cualquier momento.

Bandera 0 255 oscuridad = no se describe la localidad

0 luz = si se describe la localidad

Si obj 0 presente = loc. se describe aunque 0 = 255

Bandera	1	Indica objetos llevados (no puestos).
Bandera	2	< cuando describe localidad.
Bandera	3	< cuando desc loc. y oscuro (0 no 0).



Bandera	4	< cuando desc loc. y oscuro y obj 0 ausente	
Bandera	5 a 8	< con cada turno, (cada frase o timeout)	
Bandera	9	< con cada turno y oscuro (0 no 0)	
Bandera	10	< con cada turno y oscuro y obj 0 ausente	
11 a 28		LIBRES PARA USAR	
Bandera	29	Lleva el control de dibujos	(GRAFICO)
		Bit 7 (128) fuerza el dibujo	(MIRAR)
		Bit 6 (64) para que siempre se dibuje	(PICS ON)
		Bit 5 (32) para no dibujos	(PICS OFF)
Bandera	30	Lleva la puntuación.	
Bandera	31 y 32	(LSB/MSB) Turnos o número de frases del input del jugador.	
Bandera	33	El Verbo de la SL actual	
Bandera	34	El primer Nombre de la SL actual.	
Bandera	35	El Adjetivo del primer Nombre.	
Bandera	36	El Adverbio de la SL actual.	
Bandera	37	El número máximo de objetos llevables.	
		Por defecto = 4	(ABILITY)
Bandera	38	La localidad actual.	
Bandera	39	La línea superior del texto.	(LINEA)
Bandera	40	Controla el display de pantalla.	(MODE)
		0 = Si hay dibj. en base de datos (no subrutina) se hace un CLS a los "Colores por Defecto" y se dibuja. Luego, al apretar tecla = se restituyen los colores originales y se hace DESC.	
		1 = Sólo Texto. Hay Scroll completo de pantalla. CLS no está activado al DESC.	
		2 = Si hay dibujo en base de datos (no subrutina) se hace un CLS a los "Colores por Defecto" y se dibuja. Los colores originales se restituyen, el área bajo la línea dada por Ban 41 hace un CLS y se imprime texto desde esa misma línea.	
		3 = Como en dos. Pero el dibujo no hace Scroll.	
		4 = Para Texto sólo. Pero protegiendo el DESC como si fuera dibujo.	
		Bit 7 (128) fuerza el que no haya cambio de Border	
		Bit 6 (64) produce el mensaje "Hay más..." cuando se llena la pantalla.	
Bandera	41	Lleva el número de línea para partir Dibujo/Texto. Este es el usado por la acción PROTECT. Por defecto, (o si recibe valor < de 4 o > 24), tomará el valor de 12.	
Bandera	42	Lleva el mensaje con el que se contesta al input del jugador, (normalmente se selecciona de forma aleatoria entre los Mensajes del Sistema 1 a 4).	(PROMPT)
Bandera	43	Lleva la Preposición de la SL actual.	
Bandera	44	Lleva el Segundo Nombre de la SL actual.	
Bandera	45	Lleva el Adjetivo del Segundo Nombre.	
Bandera	46	Lleva el Pronombre del Nombre actual (La o Lo)	
Bandera	47	Lleva el Pronombre del Adjetivo actual (La o Lo)	
Bandera	48	Lleva la duración de TIEMPO MUERTO.	
Bandera	49	Lleva los controles de TIEMPO MUERTO (INPUT+TIME)	
		Bit 7 (128) = Seteado si el Tiempo Muerto ocurrió en el último turno.	

El Tiempo Muerto chequea este Bit.

Bit 6 (64) = Seteado si hay base de datos disponible para recordar (No es de uso del escritor)

Bit 5 (32) = Seteado para provocar una autollamada del Buffer de Input durante el Tiempo Muerto.

Bit 4 (16) = Seteado para imprimir el Buffer en Exit. (Es para ser usado con el Bit 3).

Bit 3 (8) = Seteado para coger el Input de la parte inferior de la pantalla.

Bit 2 (4) = Seteado para que el Tiempo Muerto ocurra en ANYKEY (Pulsar cualquier tecla)

Bit 1 (2) = Seteado para que el Tiempo Muerto ocurra en More... (Hay más...)

Bit 0 (1) = Seteado para que el Tiempo Muerto ocurra sólo al comienzo del input.

Bandera	50	Lleva el Número del Objeto para un bucle de DOALL. (Es el valor que sigue a DOALL).
Bandera	51	Lleva el último objeto a que se refiere un GET/DROP/WEAR/WHATO, etc.
Bandera	52	La fuerza del jugador (el máximo peso de los objetos llevados más los puestos encima) Inicialmente es 10. (ABILITY)
Bandera	53	Es la bandera para imprimir los objetos en INVENTARIO. Bit 7 (128) = Seteado para que cualquier objeto se imprima en un LISTOBJ o un LISTAT. Bit 6 (64) = Seteado para que los objetos se impriman en una línea continua. Por ejemplo, LET 53 64 hará que se listen en una sola línea formando una frase.
Bandera	54	Lleva el número de la loc. del objeto actualmente referido.
Bandera	55	Lleva el Peso del objeto actualmente referido.
Bandera	56	Estará a 128 si el objeto actualmente referido es un contenedor.
Bandera	57	Estará a 128 si el objeto actualmente referido se puede llevar puesto encima.
Bandera	58 y 59	NO DEBEN USARSE PUES SE USARAN EN FUTURAS EXPANSIONES.
Bandera	60- 255	ESTAN LIBRES PARA USAR

### Significado de las abreviaturas:

locno.	es un número de localidad válido.
locno+	también permite el uso de: 252 (no creado); 253 (llevado puesto encima); 254 (llevado) y 255, las cuales son convertidas en la localidad actual del jugador.
mesno.	es un mensaje válido.
sysno.	es un Mensaje del Sistema válido.
flagno.	es cualquier bandera de 0 a 255.
procno.	es un número de subproceso válido.
palabra.	es una palabra de un tipo requerido, que esté presente en el Vocabulario, o que significa no palabra (no que no se ha encontrado una, que sería lo normal).

Las acciones pueden ser divididas en 5 grupos:

1. Son acciones que ejecutarán una salida total de todas las tablas, y harán un salto a describir la localidad actual.
2. Usualmente, solamente la acción END también hará una salida total de las tablas, pero inicializará un nuevo juego.
3. Se trata de acciones que paran el proceso en la tabla actual y hacen una salida a la tabla desde la cual se llamaron.
4. Salida condicional, es cualquier acción que para el proceso en la tabla actual y hace una salida a la tabla desde la cual se llamó *si falla ejecutar la función requerida*.
5. Normal, cualquier acción que hace su función y permite a PAW continuar buscando por el siguiente conducto en la entrada actual.

### Condiciones:

AT	locno.	;Se asegura de que el jugador esté en una localidad específica.
NOTAT	locno.	
ATGT	locno.	;una localidad más alta que la especificada
ATLT	locno.	;una localidad más baja...

PRESENT	objno
ABSENT	objno
WORN	objno
NOTWORN	objno
CARRIED	objno
NOTCARR	objno
ISAT	objno locno+
ISNOTAT	objno locno+

ZERO	flagno
NOTZERO	flagno
EQ	flagno 0-255
NOTEQ	flagno 0-255
GT	flagno 0-255
LT	flagno 0-255
SAME	flagno flagno
NOTSAME	flagno flagno

ADJECT1	palabra
ADVERB	palabra
PREP	palabra
NOUN2	palabra
ADJECT2	palabra

CHANCE	0-99	; posibilidad aleatoria de que suceda
TIMEOUT		;en el último input del jugador hay un Tiempo Muerto
QUIT		

### ACCIONES:

GET	objno	tipo 4
DROP	objno	" 4

WEAR	objno	" 4	
REMOVE	objno	" 4	
CREATE	objno		
DESTROY	objno		
SWAP	objno objno		
PLACE	objno locno+		
PUTO	locno+		
PUTIN	objno locno.	tipo 4	
TAKEOUT	objno locno.	" 4	
DROPALL			
AUTOG		" 4	
AUTOD		" 4	
AUTOW		" 4	
AUTOR		" 4	
AUTOP	locno.	" 4	
AUTOT	locno.	" 4	
COPYOO	objno objno		
COPYOF	objno flagno		;Copia la posición del objeto a la bandera
COPYFO	flagno objno		
WHATO			;Convierte el Nombre1(Adjetivo1) al objeto actual
WEIGH	objno flagno		;El Peso del objeto se pone en la bandera
SET	flagno		
CLEAR	flagno		
PLUS	flagno 0-255		;Añade el valor a la bandera
MINUS	flagno 0-255		
LET	flagno 0-255		
ADD	flagno1 flagno2		;el contenido de la bandera1 se añade a la bandera2
SUB	flagno1 flagno2		
COPYFF	flagno1 flagno2		
RANDOM	flagno		;Pone un número aleatorio de 0 a 99
MOVE	flagno		;Ajusta el contenido de la bandera de acuerdo al Verbo de la SL y de la entrada en la tabla de conexiones para las localidades, (controla el movimiento de un PSI).
GOTO	locno.		
WEIGHT	flagno		;El peso de los objetos llevados o puestos encima se ponen en la bandera.
ABILITY	0-255 0-255		;Pone los objetos llevados y la fuerza.
MODE	0-4 0-3		
LINE	0-20		;Pone línea de pantalla para los juegos que llevan dibujo y texto
GRAPHIC	0-3		
PROMPT	sysno		;Ejecuta un Prompt en el input. Si se utiliza 0 es aleatorio
INPUT	0-7		
TIME	0-255 0-7		
PROTECT			;Protege el texto en la pantalla desde la línea de impresión actual del scroll
PRINT	flagno		; Muestra el contenido de la bandera en la pantalla
URNS			
SCORE			
CLS			

NEWLINE			
MES	mesno		;mensaje sin dejar línea en blanco
MESSAGE	mesno		;mensaje con línea en blanco
SYSMESS	sysno		;mensaje del sistema sin línea en blanco
PICTURE	locno		;Muestra el dibujo (sin CLS).
PAPEL	0-9		
TINTA	0-9		
BORDER	0-7		
CHARSET	0-255		;Selecciona el set de caracteres (si está insertado)
SAVEAT			;Salva la posición actual para imprimir
BACKAT			;vuelve a ella
PRINTAT	0-20	0-31	;Pone una nueva posición de imprimir
LISTOBJ			;Hace un listado de los objetos que están en la localidad actual
LISTAT	locno+		;Hace un listado de los objetos que están en la localidad especificada
INVEN	Tipo 3		
DESC	Tipo 1		
END			;Tipo 2. Produce una salida de todas las tablas para recomenzar el juego
DONE	Tipo 3		
NOTDONE	" 3		
OK	" 3		
SAVE	Tipo 1		
LOAD	" 1		
RAMSAVE			
RAMLOAD	flagno		
ANYKEY			
PAUSA	0-255		; Detiene el programa por un número de segundos igual a n/50
PARSE			;Convierte una cadena que se haya teclado en el input en una SL
NEWTEXT			; Fuerza la pérdida de lo que queda de la frase.
BEEP	0-255	0-255	
PROCESS	procno		;Ejecuta una subrespuesta/proceso
DOALL	locno+		;Genera el Ncmbre (Adjetivo) 1 para cada objeto que esté en la localidad marcada por locno.
RESET	Tipo 1	locno.	;Mueve al jugador y a los objetos presentes. Resetea los demás a la posición inicial (es el usado en juegos de partes múltiples con el comando LOAD, ver el ensayo correspondiente).
EXTERN	0-255		;Hace una llamada a un programa externo.

© 1986 Gilsoft International Ltd.

Published by Gilsoft International Ltd.,  
2 Park Crescent. Barry, South Glamorgan. CF6 8HD  
Telephone: Barry (0446) 732765  
Distribuidor Oficial para España : A.D.S.A

All rights reserved, unauthorised copying, hiring or lending strictly prohibited