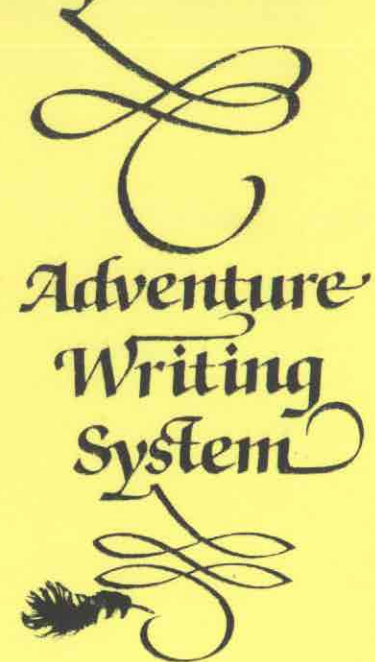


© 1986 Gilsoft International Ltd.

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

# The Professional



Adventure  
Writing  
System

## Introducción





P A W S

## SISTEMA PROFESIONAL PARA LA CREACION DE AVENTURAS GRAFICAS

Versión en Castellano

(c) programa: 1986/89 Gilsoft International Ltd.

(c) 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 éste 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 distinguirse 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 tu 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.

## FINAL DEL CAMINO

Esperamos que el manual anterior te haya dado una pequeña idea de las facilidades que ofrece el PAW y de lo complejo que es.

Ahora te toca a tí profundizar más en él por medio del uso.

La Guía Técnica te dará una especificación exacta de todo lo que contiene PAW, además de varios ensayos sobre otros temas que te serán de mucha utilidad a la hora de crear tus propios juegos.

En la cinta, además de PAW y todos sus Overlays, encontrarás una base de datos llamada Start que te dará un comienzo rápido.

En la otra cara están los 22 sets de caracteres y al final encontrarás una magnífica "Demo" que debe ser cargado usando la opción J del menú principal (después de haber salvado tu base de datos, por supuesto).

Verás en esta Demo ejemplos de vocabulario, manejo de tablas, complicados dibujos con subrutinas, manejo de objetos (lleno y vacío), e incluso un simpático PSI al cual deberás aprender a hablar entre comillas (decir "como te llamas").

Estudia a fondo este breve ejemplo tomado nada menos que de la AVENTURA ORIGINAL, te dará muchas ideas de cómo debes planear tu juego.



ANDRES R. SAMUDIO - 1989



Pero este efecto se cancela la próxima vez que Proceso 1 y Proceso 2 sean ejecutados, porque PAW le resta a la bandera los 128 que le debía.

Esto es para asegurar que los Procesos 1 y 2 actúen con normalidad hasta que se les ordene de una manera específica (con la bandera 58 en 128) que se comporten especialmente.

Suponemos que deducirás hábilmente que si pones 58 a 0, el efecto se anula.

A medida que vayamos introduciendo mejoras recibirás pronta documentación sobre ellas.

## INDICE

Introducción	Pag. 5
Conceptos	Pag. 7
Sección de datos	Pag. 7
Paginación	Pag. 7
Parser	Pag. 9
Menú	Pag. 9
Escribiendo una Aventura	Pag. 12
Enfoque y Guión	Pag. 12
Diseño	Pag. 13
Empezando a teclear	Pag. 16
Localidades	Pag. 16
Colores	Pag. 16
Conexiones	Pag. 20
Probando el juego	Pag. 21
Objetos	Pag. 23
Vocabulario	Pag. 27
Palabras para objetos	Pag. 30
Diagnóstico	Pag. 31

## INDICE

Tabla de Respuestas	Pag. 36
Situación de un objeto	Pag. 42
Mensajes	Pag. 46
Tabla de Procesos	Pag. 50
Objetos dentro de objetos	Pag. 56
El Pajarito	Pag. 60
Sub-procesos	Pag. 62
El Ferrito	Pag. 68
Mejoras por ti mismo	Pag. 78
Uso de Overlays	Pag. 80
Compresión de Texto	Pag. 82
Editor de Caracteres	Pag. 83
Editor Gráfico	Pag. 85
Ultima Hora (muy importante)	Pag. 99
Final del Camino	Pag. 103

En versiones anteriores, el marcador controlado por NEWTEXT se hubiera seteado automáticamente, pero esto se ha cambiado para permitir, en caso de que haya múltiples acciones PARSE, encontrar el último comando de una cadena sin que la bandera se setee.

Por lo tanto, en todos tus juegos previos debes añadir una acción NEWTEXT justo después de una orden PARSE (que es el exacto lugar donde el Proceso irá si no encuentra una sentencia válida en la cadena), si quieres mantener la compatibilidad.

El Verbo y Adverbio actuales no se borran (las banderas 33 y 36 permanecen seteadas) cuando se ejecuta la acción PARSE en una cadena. Esto significa, que si un Verbo (o un Nombre Convertible en verbo), es omitido de la primera frase en la cadena, el Verbo usado será el de la frase que inició o disparó la acción PARSE (usualmente un ...Decir a...)

Esto es un cambio menor, pero significa que el Verbo Actual se mantiene cuando la cadena sufre un PARSEado múltiple.

## UNA BANDERA MAS

¿Recuerdas que te insistíamos mucho en que respetaras las banderas hasta la 60, porque se usarían en futuras expansiones?

Pues el caso ha llegado con esta nueva versión.

Nos congratulamos en anunciar que la bandera 58 ha alcanzado el estatus de "Bandera del Sistema".

Si la seteas a 128 (es decir, el bit 7), PAW empezará a buscar pareja para palabras en una tabla de Procesos, cosa que antes, como recordaréis, sólo hacía en la tabla de Respuestas.

Esto facilita lo visto anteriormente en los PARSE múltiples, permitiendo acciones para un PSI en la tabla de Procesos 2. Estamos seguros, de que con tu habilidad habitual, le encontrarás otros usos.

## DRIVERS DE IMPRESORAS

El Driver de Impresora queda ahora limitado a 48 bytes en la dirección 29587 (PRTADD). Aunque la memoria ha quedado sensiblemente reducida, se ha hecho para poder dar más facilidades extras.

### EL PARSE

En ésta versión se han hecho algunos sutiles cambios en la cadena que maneja el PARSE.

Casi todos están encaminados a facilitar e implementar el manejo de los PSI. Pero, aunque no los uses, debes de saber que existen, pues pueden afectar algunas partes del juego.

1-El conducto PARSE, mantiene ahora la "posición actual" dentro de una cadena en la SL que está siendo usada.

Por lo tanto, un segundo conducto PARSE, continuará a partir de esa posición.

En las versiones anteriores, un segundo conducto PARSE, hubiera dado como respuesta la misma SL que el primero.

Por lo tanto la frase ...Dile al Archivero "coge la espada y mata al dragón" es aceptada y ejecutada en cada una de sus acciones.

2-La acción PARSE, no afectará al indicador que marca si la "línea de comando" está vacía o llena, que estaba controlado por la acción NEWTEXT.

Esto significa que una orden del tipo...Decir al Archivero "Coge la llave", ir al Norte.....será en esta versión ejecutada correctamente, es decir, primero lo que has dicho al PSI y luego tu acción de irte al Norte.

### Introducción

Bienvenido al mundo del escritor de aventuras...

El P.A.W. o Professional Adventure Writer, te ofrece todas las facilidades para producir aventuras gráficas de alta calidad en código maquina.

PAW te proporciona una base estructurada para poder escribir el juego, pero sigue siendo tarea del escritor el tener un guión imaginativo e ideas originales.

Los manuales que vienen con PAW cubren todos los aspectos de su uso. Este manual primero servirá como guía para crear una aventura y recomendamos su lectura detenida y el ejercicio de sus ejemplos antes de intentar hacer un juego propio.

El segundo manual da una información detallada de todo el sistema y debe ser usado como libro de referencia al escribir una aventura.

El equipo de Aventuras A.D. te desea buena suerte y espera tus interesantes producciones.



## Empezando

La cinta contiene el programa principal en el lado 1, éste programa es el mismo para los usuarios de 48K y 128K.

Inserta la cinta y teclea:

LOAD '' luego ENTER y PLAY en el cassette.

En el 128K usa la opción LOAD del menú inicial. Si cargas PAW en el modo 48K en un 128K, este se comportará como si fuera la versión de 48K. Es preferible cargar PAW en el modo 128K.

En la pantalla inicial de carga se mostrará la versión actual (una letra seguida por dos dígitos, p.e. A01).

También se mostrarán dos direcciones en decimal que serán necesarias si quieres escribir tus propias adiciones al PAW, sea en BASIC o en CODIGO MAQUINA. Para detalles consulta el manual técnico.

Al apretar cualquier tecla aparecerá el menú principal...

En este manual cualquier orden que sea requerida por PAW se pondrá entre corchetes, p.e. [ A 1 CAJA ] y se debe teclear exactamente todo lo que haya dentro de los corchetes, *incluidos los espacios*, pero no los corchetes.

También las teclas especiales se muestran en mayúsculas, p.e. ENTER, CAPS SHIFT, etc. pero no las teclees completas, solo pulsa la tecla requerida o la combinación de teclas.

Cualquier comunicación sobre PAW debe ser dirigida a :



## Notas adicionales para la última versión del PAW.

La versión Castellana de Paw se corresponde con la versión A18 inglesa, siendo por lo tanto la más avanzada de las existentes.

Las diferencias con respecto a todo lo que se ha explicado en estos densos y sapientísimos manuales son:

### OVERLAYS PARA EL USUARIO

La última versión de PAW tiene una característica muy importante que permite, y de hecho facilita, la incorporación dentro del sistema de cualquier rutina o utilidad hecha por separado : son los overlays del usuario.

Estos añadidos deben estar escritos en Assembler y pueden tener hasta 5 K de longitud.

Para ellos se ha añadido en el menú principal la opción Z, que te pregunta el nombre de la base de datos a cargar.

Puedes elegir entre cualquier letra (A - Z) y PAW empezará a buscar en el "periférico escogido" por una base de datos con esa extensión.

### PERIFERICO ESCOGIDO

(Current Device)

PAW ahora te preguntará siempre en qué periférico vas a Salvar/Cargar tu base de datos. Puedes hacerlo en Disco, Cassette o Microdrive.



Parecería que no hay forma de borrar esto sin continuar haciendo PLOTS cada vez más lejos. Pero de hecho, [GRAPH] que es (CAPS SHIFT + 9 en un 48K) borrará el siguiente o próximo comando y podrás borrar también el PLOT de esta manera.

A tener en cuenta es que, al final de una edición, es posible dejarse órdenes por encima del puntero de cadena que sean inútiles. Puedes borrarlas todas usando DELETE NEXT (GRAPH) por un tiempo.

Ahora vuelve al menú principal y selecciona la tabla de Respuestas (los de 48K deberán cargar un OVERLAY). Corrige la entrada \* \* para que ponga MODE 3 3 LINEA 16 antes de GOTO 2.

La acción MODE 3 selecciona uno de los modos de operar la pantalla. En este caso quedará un área de gráfico fijo, ningún texto la mueve; el segundo 3 le informa a PAW de que no debe cambiar el color del borde y que debe imprimir "CONT.." cuando la pantalla esté llena de texto.

LINEA 16 le dice a PAW dónde va a empezar la primera línea de texto.

Finalmente usamos TEST para probar la aventura (de nuevo los de 48K deben de cargar un OVERLAY) para verla en acción.

El dibujo solamente será pintado la primera vez que visites la localidad, pero no en visitas siguientes. Esto es lo que se llama el modo normal para gráficos. Es posible de todos modos seleccionar ON y OFF, siendo ON la opción para siempre poner los gráficos y OFF nunca poner los gráficos.

Se seleccionan con: PICS NORM  
PICS ON  
PICS OFF

Esas órdenes no fueron explicadas cuando hablamos de la tabla de Respuestas, pero están todas en el manual técnico. (Que también habrás de leer, je, je).

Debes decidir si permites al jugador cambiar entre estas opciones o no, y forzarle a usar un solo método desde el comienzo.

## Conceptos

Es una buena idea introducir algunos conceptos importantes utilizados por PAW.

### Secciones de datos

Durante muchos años ha sido de práctica común en los ordenadores que utilizan disco como soporte principal, el partir los programas largos en dos o mas secciones que se cargan desde el disco cuando se van a utilizar. Este sistema no ha sido muy utilizado en cintas por su misma naturaleza como contenedor de datos en forma seriada.

PAW usa una forma muy simple para dejar el máximo de memoria libre, tanto en 48K como en 128K. Utiliza la memoria disponible para ir guardando porciones de datos hasta el momento en que esa memoria va siendo usada y sobrescrita. Los datos entonces se van obteniendo desde cinta según se requieran.

En un 128K no hay necesidad de preocuparse por las secciones de datos hasta que las últimas 16K son usadas (i es decir, hasta que se lleva escrita una aventura de cerca de 92K !).

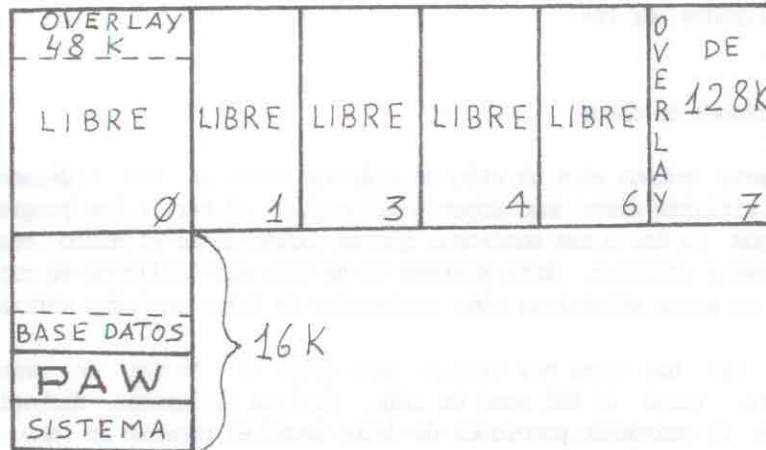
En 48K las secciones se empezarán a necesitar en el momento de usar los gráficos o al llevar escritas ya 16K de programa.

La muestra en este manual no necesita secciones extra de datos así que su uso se explicará en detalle mas adelante.

### Paginación de memoria

El 128K usa un sistema llamado paginación para obtener su memoria extra; el diagrama 1 ayudará a visualizar su composición. En él se vé que la parte superior se puede mover como una diapositiva en un proyector para que cada una de las páginas de 16K se pueda acomodar en un ordenador que solo puede ver 64K cada vez. (ver diagrama No. 1)

DIAGRAMA 1



Nota : las páginas 2 y 5 son el área de memoria donde se localiza el PAW y por ello no están disponibles.

La zona de memoria marcada como secciones de datos será usada por PAW si es necesario. (ver sección anterior).

#### Base de datos (BD)

PAW guarda el juego en una base de datos (una colección de tablas e información que definen el juego que se está escribiendo). Inicialmente la bd es muy pequeña y solo contiene las palabras y órdenes mas comunes en estos juegos.

Esta base de datos va poco a poco usando el área de memoria que se muestra como libre en el diagrama 1.

PAW también puede hacer uso de otras páginas, pero en un 48K esas páginas extra no están disponibles. Lo que significa que si se usa un 128K para escribir la aventura y se quiere que ésta corra tanto en 128 como en 48K, no se debe usar otra página sino la principal (la Ø).

Pinta el hierbajo con un PLOT 72,72 y ahora pongamos líneas entre los siguientes puntos:

68,85	77,73	74,89	81,73	79,94	87,72
87,94	92,70	94,86	95,70	99,79	98,69

Volvamos al comienzo de la cadena con (CAPS SHIFT + 8) y usemos el comando NEXT (CAPS SHIFT + 6) para ir hasta justo después del PLOT. Usemos DELETE para quitarlo.

OJO: esta acción extraña significa que ahora la primera línea está en 0,0 y nos permite colocar el dibujo con exactitud.

OJO: si tratas de corregir el dibujo otra vez (AMEND) saldrá el mensaje " Out of range" que significa fuera de límites, porque PAW no puede dibujar "fuera de la pantalla". El puntero de cadena deberá estar situado exactamente antes de la primera orden LINEA. Así que insertemos el PLOT otra vez mientras editas la cadena, borrándolo de nuevo al final.

Corrige el dibujo 4 de nuevo [A 4 ENTER] y hagamos un PLOT en el punto 217,69. Ahora con [SS] y [G] activaremos un bucle GOSUB y te pedirá el número de la localidad que vas a usar, tecléa [0 ENTER] para usar nuestro hierbajo que hemos definido en esa localidad.

Lo siguiente que se te pedirá, será la escala en que vas a dibujarlo, esta escala va de 8 en 8 (o sea en octavos) desde su tamaño original. Por ejemplo, si tecléas [4 ENTER] será 4/8 (4 octavos) igual a la mitad.

PLOT 21,58	GOSUB 0	escala 5
PLOT 60,74	GOSUB 0	escala 3
PLOT 103,61	GOSUB 0	escala 4

Y para demostrar errores:

PLOT 128,170 GOSUB 0 escala 0

Supongo que sabrás que escala 0 significa tamaño completo (no 0 octavos). El error que se ha generado ha dejado el puntero de cadena antes del comando que causó el error: en este caso el GOSUB.



**INVERSE** hace que cualquier PLOT y LINEA reseteen o apaguen los pixels afectados en vez de setearlos o encenderlos, por lo tanto:

MOVIDA RELATIVA 141,95    LINEA 141,107  
MOVIDA RELATIVA 146,108    LINEA 146,96

ahora quita el INVERSE con [SS] + [I].

#### LA CERCA

Vamos a usar una técnica importante de desdibujar el perímetro de un diseño de sombreado (SHADE):

Tinta negra [SS] + [X], [0 ENTER].

PLOT 8,104    LINEA 79,104    LINEA 79,88    LINEA 8,88  
PLOT 247,88    LINEA 208,88    LINEA 208,104    LINEA 247,104

Usemos el diseño de SHADE número 7 en 246,102 y 11,102.

Vamos a desdibujar la parte de arriba de la cerca para crear algunas pequeñas puntas: [SS] + [I] (INVERSE).

PLOT 247,104    LINEA 208,104    PLOT 79,104    LINEA 8,104

Luego quita el INVERSE.

Para terminar el efecto:

PLOT 8,101    LINEA 79,101    PLOT 208,101    LINEA 247,101

Vamos a añadir ahora unos matorrales de hierba usando la sub-rutina que mencionamos anteriormente. Esto nos economizará memoria y tiempo. Con ENTER vamos al submenú. Como la localidad 0 es la pantalla del título vamos a usarla en nuestro juego de demostración para que su cadena de dibujo contenga nuestros matorrales de hierba.

Usemos [A 0 ENTER] para corregir el dibujo de la localidad 0. Nos damos cuenta de que el número está seguido por la letra S, que indica que es una sub-rutina (no se pintará).

#### **Parser**

To Parse: clasificar una palabra o analizar una frase en sus términos gramaticales.

PAW convierte lo tecleado por el jugador en series de *sentencias lógicas* (SL) para las cuales ya hay una respuesta definida.

Esto se hace extrayendo frases de la cadena de *input*, de una en una y dejando que el intérprete las descifre.

Las frases pueden ir separadas por cualquier signo de puntuación y por las conjunciones *y*, *entonces*, o *luego* (o la que se haya definido).

Cuando ya no hay mas frases en la actual cadena de input, se pide otra al jugador.

Una frase debe consistir al menos en un **verbo** (indica acción); opcionalmente dos **nombres** (objetos o personajes); posiblemente asociados a **adjetivos** (que los describen); de **adverbios** (que modifican al verbo); de una **preposición** (que muestra la relación de un nombre con otra palabra); y también puede contener una "cadena entre comillas para hablar con otros personajes"

#### **Menú**

Del menú principal, mostrado al pulsar cualquier tecla desde la página de título, se puede acceder a cualquier opción del PAW con sólo teclear una sola letra (en MAYUSCULAS) seguido de ENTER.

El menú está dividido en dos partes y la opción E permite cambios entre ellas. La parte inicial muestra todas las funciones necesarias para crear un juego. La segunda (accesible con E), contiene todo lo necesario para Salvar, Cargar, Verificar, Probar el juego, etc. y también el diseñador de caracteres y el compresor de textos (se explicarán mas adelante).

Si te encuentras con la pregunta: ¿carga de sección de datos?, es porque la parte a usar no está presente en ese momento y requiere que se cargue de memoria externa (ver secciones de datos). Entonces se requerirá teclear si o no, ENTER. En el momento actual no la necesitaremos = no, ENTER.

#### LINEA DE EDICION

Similar a la de input en basic; se puede usar las teclas de cursor para moverse a derecha e izquierda y **DELETE** para borrar lo que está a la izquierda del cursor.

**EDIT** debe ser tecleado dos veces, o apretado hasta que se autorrepita para limpiar cualquier cosa que se haya tecleado.

**CURSOR Y ABAJO (6)** debe ser tecleado dos veces o apretado hasta que se repita para abandonar el texto presente, la diferencia es que aquí se deja el texto que se estaba escribiendo en la base de datos y con **EDIT** se borra todo.

#### Memoria libre

Con **F** se muestra la memoria disponible en cada página. En 48K se referirá solo a la página 0.

Además, se muestra también la localidad y el mensaje mas alto usados hasta el momento. La razón de esta información tan útil se explica mas adelante. Apretando cualquier tecla se vuelve al menú.

#### Salvando, cargando y verificando la base de datos

Como es obvio que no se puede terminar un juego de una sola sentada y existe el peligro de pérdida de información por cualquier fallo del sistema, estas opciones permiten pasar a cinta o a disco un fichero con toda la información.

Con **S** se pedirá un nombre de fichero, te sugerimos ponerle un nombre que sea significativo y que contenga el número de la versión o la fecha.

Normalmente cada **PLOT** y **LINEA** lo que hace es setear los pixels a los cuales afecta, pero esos mismos comandos si se insertan mientras la acción **OVER** está activada, lo que hacen es setear todos los pixels reseteados y resetear todos los pixels seteados. Es decir, encienden los apagados y apagan los encendidos.

Se parece mucho a la acción que hacía **SPACE** cuando mirabamos el Editor de Caracteres. Es importante recalcar que el estado de **OVER** (y el Inverso que veremos más tarde) están codificados como parte de la orden; para obtener el efecto, tienes que insertar la orden mientras esté activo, no puedes cambiar un **PLOT** o **LINEA** insertado previamente sin haberlo borrado con anterioridad.

**OVER** se cancela con **START** (CAPS SHIFT + 8), **PREVIOUS** (CAPS SHIFT + 7) y **DELETE**.

Ahora con el **OVER** seleccionado haz lo siguiente:

LINEA 115,140    MOVIDA RELATIVA 144,166    LINEA 171,140

y después **[SS] + [O]** para quitar el **OVER** otra vez. Debes de tener dos líneas dibujadas con pixels en parte seteados y en parte reseteados.

El último toque del Pabellón:

MOVIDA RELATIVA 142,136    LINEA 142,96    LINEA, 145,96    LINEA 145,135

mueve la línea a 143,133 y haz un **FILL**.

La parte trasera de la balastrada debe ser un poco más pequeña:

MOVIDA RELATIVA 125,107    LINEA 161,107  
MOVIDA RELATIVA 162,96    LINEA 126,96

mueve la línea a 128,98 y haz un **SHADE** con el diseño 15. Mueve la línea a 150,98 y haz un **SHADE** con el diseño 15 otra vez.

Para que quede mejor, deberíamos hacer destacar el palo central removiendo una línea de pixels a cada lado. Hagamos **[SS] + [I]** (**INVERSE**) y la letra **I** aparecerá en la línea de estado para indicarte que el **INVERSE** está activado.



PLOT 206,96 LINEA 206,109 LINEA 162,109 LINEA 162,96

mueve ahora la línea a 163,97 y haz un SHADE con el diseño 15 (el que habías diseñado en la sección sobre el Editor de Caracteres), se hace con [SS] + [S], [15 ENTER 15 ENTER].

PLOT 125,96 LINEA 125,109 LINEA 81,109 LINEA 81,96

mueve la línea ahora a 82,97 y utiliza el diseño de SHADE número 15 otra vez.

Vamos ahora con los palos del Pabellón de Música:

PLOT 103,96 LINEA 103,136 LINEA 106,136 LINEA 106,96

mueve la línea a 104,121 y con [SS] y [F] accedemos al comando FILL: éste llena un área previamente definida completamente con pixels seteados en un forma muy similar a SHADE.

MOVIDA RELATIVA 184,96 LINEA 184,136 LINEA 181,136 LINEA 181,96

mueve la línea a 183,122 y haz otro FILL con [SS] + [F].

Veamos ahora el techo del Pabellón:

MOVIDA RELATIVA 207,143 LINEA 206,140 LINEA 202,136 LINEA 86,136  
LINEA 83,139 LINEA 80,143 LINEA 207,143

mueve la línea a 205,141 y haz un SHADE con el diseño 15.

MOVIDA RELATIVA 144,166 LINEA 220,139

MOVIDA RELATIVA 65,139 LINEA 144,166

mueve la línea a 144,164 y haz un FILL.

#### CREANDO UN EFECTO DE VOLUMEN EN EL PABELLON DE MUSICA:

Selecciona OVER con las teclas [SS] + [O], aparecerá una letra O en la línea de estado para indicar que el OVER está activado.

La bd se salvará en varias partes. PAW salva dos ficheros por página, cambiando la última letra del nombre a A, B, C, etc para cada uno usado.

Con H se verificará la bd salvada comparándola con la que hay en memoria. Si se detecta un error, entonces hay que volver a salvar la bd en una cinta diferente si es necesario.

Con J se pide el nombre de la bd que se desees cargar de nuevo dentro de PAW. *Cualquier base de datos que esté presente dentro de PAW se borrará para dejar paso a la nueva.*

Si se produce un error de carga, se corromperá toda el área donde se aloja la bd y la única opción que se debe hacer es continuar con J hasta que la bd se haya cargado satisfactoriamente. Cualquier otra opción producirá daño al programa propio de PAW y se necesitará una recarga completa.

Regla: salva tu base de datos regularmente y en cintas diferentes para tener así un resguardo en caso de producirse el clásico corte de luz u otro accidente cualquiera.

### Enfoque previo

Es muy importante un buen planteamiento si se quiere crear una aventura con un nivel aceptable. De nada sirve sentarse a la máquina y empezar a teclear esperando la inspiración. Lo único que se conseguirá es un desorden de números y palabras y sin otro recurso que volver a empezar todo de nuevo.

Para ilustrar el método recomendado, crearemos y desarrollaremos una simple aventura desde el inicio de la idea hasta el resultado final.

!!!Recuerda salvar tu bd con regularidad !!!

### El guión inicial

Esa idea inicial es siempre la parte mas difícil al crear cualquier cosa. Una historia original puede dar al juego un interés mucho mayor que la vieja historia de rescatar a la consabida princesa.

Guiones hay de todas clases y de todo tipo, pero aconsejamos que si intentas basar la historia en una película, libro o comic, y luego intentas usarla comercialmente, te asegures antes de obtener el permiso del autor o de los actuales dueños de copyright.

En nuestro ejemplo tomaremos los problemas que pueden ocurrirle a un pasajero en su vuelta a casa:-

Mientras estaba en la parada del autobus, el billete de subida fue arrebatado por un golpe de viento y luego cogido y llevado por un pajarito hasta un cercano parque.

El ordenador juega la parte del pasajero a quien se debe dirigir para encontrar el BILLETE antes de que llegue el autobus.



Volviendo con lo nuestro, entonces desde el inicio, CAPS SHIFT + 6 hasta que llegemos a la orden PLOT, que era, como recordarás, 191,72, con la cual se inició la base del Pabellón de Música.

Usa DELETE para borrarlo de la cadena y entonces haz un PLOT 207,72 que PAW insertará en la cadena de dibujo como si fuera el puntero. Si usamos NEXT, o sea (CAPS SHIFT + 6), la línea de la base aparecerá dibujada.

En vez de usar NEXT cada vez para llegar al final de la cadena hay otra forma mucho más fácil: cuando estés al principio de una cadena muy larga si tecléas ENTER terminará la edición y te llevará al sub-menú. Entonces tecléa [A 4 ENTER] para corregir el dibujo otra vez, y ya tendrás el puntero al final de la cadena.

Es importante que te des cuenta de que la base entera se mueve como una sola unidad, porque antes usamos una MOVIDA RELATIVA, cuando la pintamos originalmente.

### LA BALAUSTRADA.

Seleccionemos tinta blanca [SS] + [X] [7 ENTER], y papel transparente (es un color especial que significa no modificar el color), lo hacemos con [SS] + [C], [8 ENTER]. Luego:



LA ESCALERITA: Mover la línea a 144,78.

Con [SS] + [R] (MOVIDA RELATIVA): Una MOVIDA RELATIVA es la que mueve el punto como PLOT y como la MOVIDA ABSOLUTA, pero a un pixel que está a una distancia fijada del punto actual, en vez de hacerlo a una posición absoluta X, Y. Es usado para mantener juntos los grupos de órdenes que dibujan un solo objeto en un gráfico. La razón de esto la veremos en un momento. Fijemos la línea a 112,78.

[SS] + [R] (MOVIDA RELATIVA) para mover a 111,81. Fijemos la línea a 143,81 y continuamos por los siguientes grupos de coordenadas:

MOVIDA RELATIVA a 144,84    LINEA a 112,84

MOVIDA RELATIVA a 111,87    LINEA a 143,87

MOVIDA RELATIVA a 144,90    LINEA a 112,90

MOVIDA RELATIVA a 111,93    LINEA a 143,93

Ahora vamos a examinar las facilidades de edición que podemos usar para corregir errores, aparte del DELETE ya conocido. Como ejemplo, vamos a mover toda la base de ladrillos que acabamos de crear, dos fronteras de caracteres a la derecha. El borrarla con DELETE toda entera para volverla a pintar, sería un poco palizas.

Vamos a explicar cómo guarda PAW la cadena de dibujo. Mientras tú pintas tu obra maestra, PAW va añadiendo cada orden a la cadena. El sitio donde la añade se llama el "puntero de cadena" y de momento el puntero de cadena es el final de la cadena.

Es posible para PAW retroceder a través de las órdenes hasta cualquier punto de la cadena. Para ello se usa el cursor de la derecha (CAPS SHIFT + 8) una sola vez, no hay que preocuparse si desaparece, puesto que el dibujo todavía está ahí. Pero aunque el dibujo está todavía en memoria, PAW sólo lo pinta hasta donde esté el puntero, el cual estará ahora al comienzo de la cadena.

El diagrama 7 te facilitará la visión de como se sitúa la cadena en la memoria.

Puedes adelantar al siguiente comando con cursor abajo (CAPS SHIFT + 6) y puedes retrasar al anterior comando con cursor arriba (CAPS SHIFT + 7). Esto es lo que se llama "Comando Previo", no borra la orden, sólo mueve el puntero una orden hacia atrás.

## Diseño del juego

Ahora que la idea está mas clara es importante hacer un boceto del área de juego, en nuestro ejemplo sería algo parecido al diagrama No. 2.

Es de notar que la zona de juego debe estar cerrada de una manera lógica, o el jugador no entenderá por qué no puede ir en una dirección si no hay nada que le impida el paso.

Para el ordenador, una aventura consiste en un determinado número de apartados o localidades que el jugador debe visitar, es ahora el momento de decidir qué zonas del boceto se tomarán como localidades y numerarlas individualmente.

Se debe intentar hacer una escala consistente o lógica (a menos que el juego sea ilógico intencionadamente), evitando que con un solo paso se vaya de una localidad a otra que aparentemente queda a varias millas de distancia.

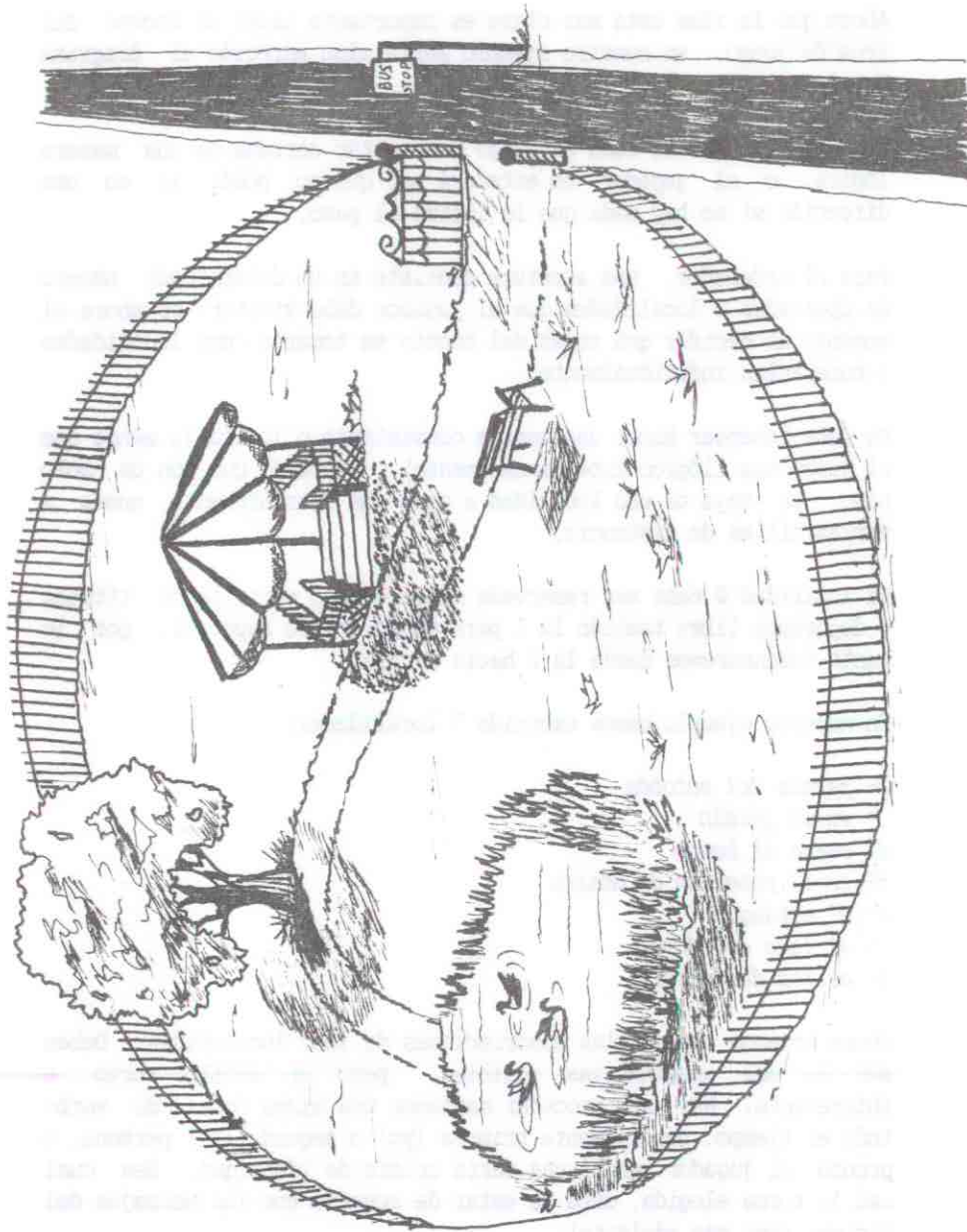
La localidad 0 debe ser reservada siempre como pantalla de títulos y dejaremos libre también la 1 para darle un uso especial, por lo tanto comenzaremos desde la 2 hacia arriba.

En nuestro ejemplo hemos escogido 7 localidades:

- 2- parada del autobús
- 3- en el jardín
- 4- cerca al banco
- 5- en el pabellón de música
- 6- el estanque
- 7- al lado del árbol
- 8- arriba del árbol

Ahora podemos empezar las descripciones de las localidades. Deben ser lo mas imaginativas posibles, pero en estilo corto e interesante. Hay que procurar mantener una misma forma de verbo todo el tiempo, generalmente primera (yo) o segunda (tu) persona, o pronto el jugador tendrá una seria crisis de identidad. Sea cual sea la forma elegida, debe de estar de acuerdo con los Mensajes del Sistema (ver mas adelante).

Diagrama 2



Para evitar problemas de mezcla de colores debemos hacerla respetando exactamente la frontera de los dos caracteres: es decir, de 3 de altura por 16 de ancho. Seleccionemos el papel rojo con [SS] + [C] y [2 ENTER].

Y la tinta negra con [SS] + [X] y [0 ENTER].

Movemos la línea a 191,72 y hacemos un PLOT con [SS] + [P].

Con [SS] + [L] (o el botón de fuego del joystick) dibuja lo siguiente:

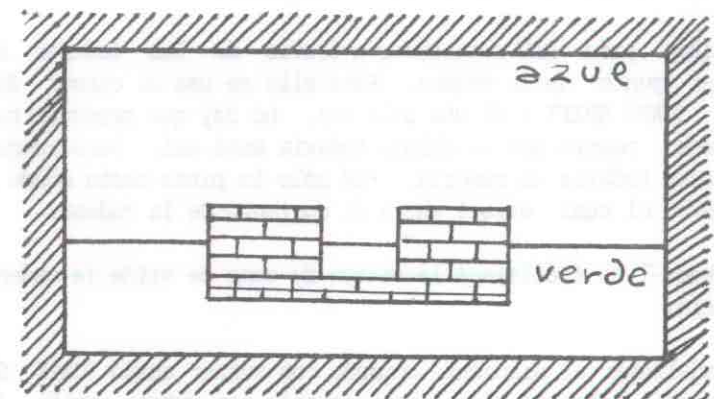
64,72 64,95 111,95 111,75 144,75 144,95 191,95 191,72

Luego, movemos la línea a 189,73 y hacemos un SHADE usando el diseño 14, por ejemplo:

[SS] + [S], [14 ENTER 14 ENTER] y crearemos el efecto de ladrillos.

Tu dibujo debe verse ahora como el del diagrama 6.

Diagrama 6





5/ A veces, si quedan sitios sin rellenar, se puede hacer poniendo el punto de comienzo dentro de la zona vacía.

6/ Hay 16 diseños que se pueden cambiar usando el Editor de Caracteres. (Ver Editor de Caracteres).

7/ Si se cambia un diseño, este se cambiará en todos los dibujos en los que haya sido usado. Por eso recomendamos usar la mayoría para un uso general y definir unos cuantos para usos muy especiales.

8/ Por supuesto, se pueden mezclar diversas combinaciones de diseño.

Ahora vamos, en nuestro dibujo, con el cielo y con la tierra.

Ponemos el color del papel moviendo la línea a 8,56 y apretando [SS] + [A] para una MOVIDA ABSOLUTA. Esta Movida Absoluta no afecta al pixel como lo hacía PLOT.

Ahora, si tecleamos [SS] + [Y] quitaremos la retícula, pues mientras esté activa NO se ven los colores.

Para poner la hierba de color verde, apretaremos [SS] + [C] (para el papel), y como verde es el color 4 teclearemos [4 ENTER] (fíjate como en la línea inferior de la pantalla el número del papel cambia a 4).

Ahora hay que mover la línea a 246,87 y apretar [SS] + [B] (de BLOCK): esto coloreará el rectángulo del cual la línea forma la diagonal con el actual color para el papel y para la tinta.

El azul del cielo: haz [SS] + [A] o una Movida Absoluta hacia 247,88. Seleccionemos ahora papel azul con [SS] + [C] y [1 ENTER], movámonos a 8,167 y hagamos otro BLOCK con [SS] y [B]. Veremos como el área cuya diagonal estemos señalando se tinte del color que hayamos elegido.

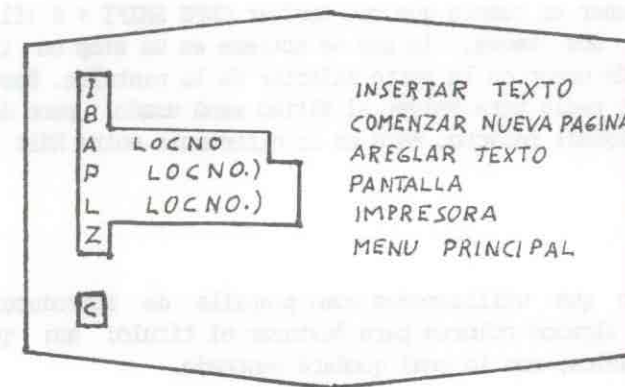
La base del Pabellón: con ladrillos rojos.

Para claridad y ver mejor los posibles movimientos haremos el mapa del diagrama 3.

Diagrama 3



Pantalla de muestra del menú



## EMPEZANDO A TECLEAR

### LOCALIDADES

Selecciona la opción L en el menú principal. Un pequeño submenú te mostrará las opciones para insertar o corregir (amend). Como todos los menús se parecen, examinaremos estas opciones para familiarizarnos con ellos. (Ver el esquema de menú del dibujo anterior)

Cualquier cosa tecleada aparecerá en la parte inferior de la pantalla, como siempre. Las zonas en inverse nos indican lo que hay que teclear para obtener las funciones que figuran a la derecha.

Por ejemplo: La opción P nos dará una descripción de la localidad elegida, en este caso será la Ø que siempre existe para simplificar el trabajo interno del PAW, y podremos ver la inclusión de colores, etc., dentro del texto, para destacar ciertas palabras. Ese texto debe ser reemplazado por nuestra introducción, así que debe ser corregido.

Con cualquier tecla volvemos al submenú y tecleando A Ø (no olvidar el espacio) veremos el texto de la localidad elegida, pero con el cursor al final para una posible edición. Para limpiar TODO el texto teclea EDIT (cap. sh. + 1) dos veces y obtendrá una localidad limpia para trabajar.

Nota: tener en cuenta que con teclear CAPS SHIFT + 6 (flecha hacia abajo), dos veces, lo que se obtiene es un stop en input, con mensaje de error en la parte inferior de la pantalla. Basta apretar cualquier tecla para volver al último menú usado, pero dejando el texto original intacto. Esta es la diferencia entre Edit y el 6.

### COLORES

El texto que utilizaremos como pantalla de introducción va a incluir algunos colores para destacar el título; así que tecléa once espacios, con lo cual quedará centrado.

Los bordes o marcos alrededor de los gráficos están de moda, así que vamos a poner uno. Primero hay que mover el final de la línea a 248,55 y hacer un PLOT en ese punto.

Luego vamos a movernos a 248,168 y fijar la línea, luego a 7,168 y continuamos fijando la línea lo mismo que en 7,55 y volvemos a 248,55. Nos quedará, pues, un cuadrado.

Es de notar que hemos ido por la línea que divide las celdas de caracteres. Esto se hace para minimizar el problema de dos colores por carácter que permite el Spectrum. Es conveniente que leas bien el manual y veas cual es éste problema.

Para hacer el borde mucho más atractivo usaremos la opción SHADE. Movemos el final de la línea a 248,53 y apretamos las teclas [SS] + [S] para tener un SHADE.

La parte inferior de la pantalla se cambiará y te preguntará el número de diseño que desees. De momento tecleemos [12 ENTER], y a continuación se te pedirá otro segundo diseño.

Este diseño se pondría encima del primero, añadiéndose a este, o sea, que se pueden mezclar dos diseños para formar uno más complicado.

De momento solamente queremos uno, y teclearemos [12 ENTER] otra vez. Automáticamente el borde quedará con su relleno. Notarás que el punto no se ha movido y que la línea sale desde el mismo sitio desde donde estaba antes del relleno.

### CARACTERISTICAS DE LA ORDEN SHADE (SOMBREADO):

- 1/ Es un comando muy rápido.
- 2/ Se pueden sombrear incluso áreas con perímetro muy difícil.
- 3/ Se puede colar a través de cualquier pixel que hayas dejado como "agujero" en tu dibujo.
- 4/ El área a sombrear debe estar definida como mínimo por una línea de un solo pixel, o por el borde de la pantalla como en nuestro dibujo.



El movimiento es pixel a pixel, pero se puede acelerar a ocho pixels cada vez si se mantiene apretado CAPS SHIFT al mismo tiempo que las teclas de movimiento o mientras se maneja el Joystick.

La línea que se usa para dibujar es del tipo llamado "elástico", porque se comporta como un tenso elástico mientras se mueve por la pantalla, con ello se consigue una mayor exactitud al posicionarla para el dibujo.

En nuestro juego de ejemplo vamos a usar gráficos a media pantalla, por lo que necesitamos dejar varias líneas libres en la parte inferior de la pantalla para el texto.

Con [SYMBOL SHIFT] más [Y] activamos la Reticula para mostrar las fronteras entre caracteres.

Ahora hay que mover la línea a X=0,Y=47 y luego mover el "punto" al final de la línea con [SYMBOL SHIFT] y [P] (Plot) para que el dibujo comience en ese pixel.

Lo siguiente es pasar la línea a X=255,Y=47 usando [A] sólo una vez, lo que de paso sirve para demostrarte que la pantalla no tiene fronteras en los bordes y que al salir por un lado, entrarás por el otro (esta opción, al parecer insignificante, te ahorrará mucho tiempo en tus dibujos).

Ahora pasamos a dibujar la línea, lo que se llama "fijarla". Para ello apretamos [SHYMBOL SHIFT] y [L] (De Línea) o usando la opción "fuego" del Joystick.

Si se hace un error, se puede borrar una orden previa con DELETE (CAPS SHIFT y 0 en un 48 K), lo cual te llevará hasta el inicio del dibujo si así lo deseas.

NOTA : Mientras no se especifique lo contrario, todas las órdenes de inserción en la cadena de dibujo (como PLOT o LINE) requieren el uso de SHIMBOL SHIFT, por lo cual lo abreviaremos a SS.

NOTA : Todas las coordenadas se darán en forma X,Y. Por ejemplo: 255,47 en vez de continuar con el engorroso X=255,Y=47.

Tecleando EXTRA MODE (que es Symbol Shift + Caps Shift en el 48K), los números del 0 al 7 daran el papel ó color de fondo. Si nosotros queremos que el título tenga un fondo rojo apretaremos el 2.

Tecléa el título, que en nuestro caso es "EL BILLETE", y ahora debemos volver al color original de papel, así que tecléa EXTRA y 0 para obtener el negro.

Necesitamos poner una línea en blanco entre el título y el texto, pero no podemos teclear ENTER porque esto termina la edición, por lo tanto, necesitamos UN CODIGO EXTENDIDO DE CONTROL DE PANTALLA (CECP); estos son códigos de 0 a 7 que tienen varios usos dentro del PAW.

CECP 7 da automáticamente un newline (INTRO ó nueva línea).

Para usar un CECP hay que hacer una pequeña trampa con el Editor de la siguiente forma: Primero tecléa EXTRA y selecciona el color blanco (7), luego tecléa DELETE (borrar) una sola vez; con ello borrarás el control del papel pero dejara el número 7, por lo tanto, como dijimos antes, el cursor saltará al principio de una nueva línea. Haz esto otra vez para dejar una línea en blanco y tecléa el resto de la introducción. Ten en cuenta que PAW hace un formateo automático del texto.

TEXTO: [Mientras esperaba el autobús una rafaga de viento se llevó mi billete. ¿Puede Vd. ayudarme a encontrarlo?].

Cuando hallas terminado tecléa ENTER para finalizar la edición y luego cualquier tecla para volver al submenú.

Con la opción P se puede ver el texto todavía sin formatear, pues esto sólo se ejecuta mientras se juega el juego.

Con L se imprimirá el texto. Si parece que el ordenador se cuelga tecléa BREAK (CAPS SHIFT + ESPACIO en 48K) para volver al menú, y entonces busca en la guía técnica más información sobre la impresora.

Con I se crea una nueva localidad. No hace falta teclear un número detras porque PAW automáticamente le asigna el próximo que esté libre.

De momento dejaremos la localidad 1 sin usar, así que sólo tecléa I + ENTER y ahora tendrás una localidad virgen para usar. Con ENTER y cualquier tecla volverás al submenú.

Ahora tecléa I + ENTER otra vez para insertar la localidad 2 y luego poder teclear el texto que la describe.

Debido al formateo automático hay que teclear siempre los espacios aunque sea al final de la línea porque si no el programa tomará dos palabras como si fuera una sola larga. Los espacios sobrantes serán suprimidos luego durante el juego.

**REGLA:** Tecléa siempre el espacio entre dos palabras o entre un punto final y el comienzo de la próxima frase, aunque ese espacio esté al comienzo de una nueva línea de pantalla. Luego será suprimido por el formateador si es necesario.

**TEXTO LOCALIDAD 2:** [Estoy en la parada del autobús en una calle de dirección Norte - Sur. Al Oeste queda un parque cuya verja de hierro está abierta].

Vuelve al submenú con ENTER (desde ahora omitiremos el uso de la palabra ENTER) suponiendo que con tu inteligencia habitual has captado la onda.

Ahora es el momento de demostrar cómo PAW previene y avisa cuando una orden no es válida: tecléa I 3 ENTER (juraíto que es la última vez que te lo recuerdo) y como resultado tendrás una parpadeante interrogación después del 3. Esto se debe a que PAW se ha dado cuenta de que no se necesita ningún número después de la opción insertar.

Por otra parte PAW posicionará siempre el cursor tan cerca al problema como pueda, en este caso con un solo DELETE quitaremos el número (no te preocupes por el espacio que queda pues PAW ignora cualquier espacio superfluo).

Ahora ENTER proporciona la localidad 3 en blanco.

**TEXTO LOCALIDAD 3:** [La hierba sobre la cual camino está muy bien cuidada. Hacia el Norte hay un cómodo banco y hacia el Este queda un estanque].

#### 1- Línea superior:

Tinta seleccionada.  
Papel seleccionado.  
Estado de Parpadeo (flash)  
Estado de Brillo (bright)

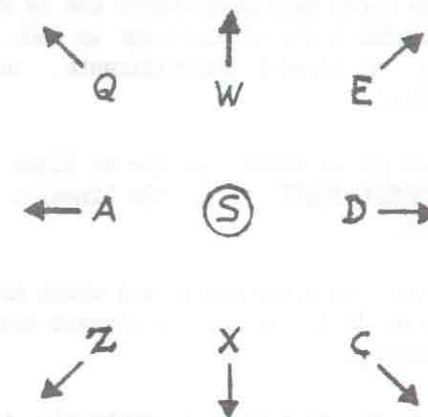
#### 2- Línea inferior:

Coordenadas x,y de la posición de dibujo.  
Número de Localidad editada.

A veces se da otra información adicional y ya la explicaremos en su momento.

Si miras con atención, notarás un solo pixel parpadeante en la parte inferior derecha, es lo que se llama "el punto" y muestra la posición de comienzo para cualquier gráfico.

Tecleando [E], verás que empieza a aparecer una línea cuya parte inicial estará en "el punto" y cuyo final podrás mover usando las teclas alrededor de la letra S como indica el esquema:



Si quieres hacer uso del Joystick, puedes insertarlo en el puerto 2 (Plus 2 e Interface 1). Los usuarios de un interface tipo Kempston deben apretar [SYMBOL SHIFT] y [J] para activarlo.

En la línea inferior aparecerá una J que indica que está activo y el Joystick controlará el final de la línea.



Como ejemplo para uso de los gráficos, lo que haremos será dibujar el gráfico del Pabellón de Música como se vería desde la localidad 4 (el camino).

Como queremos que PAW dibuje el gráfico cuando visite la localidad, tendremos que asignarle a nuestro gráfico un valor para Papel y para Tinta.

Usaremos un fondo negro (o sea un papel negro) y una tinta amarilla: Tecléa [A 4 0 6 ENTER]. Si ahora vuelves a usar [P] verás que en la entrada para la localidad 4 dirá:

Localidad 4    Papel: 0    Tinta: 6

Si aún no estás contento y quieres cambiarlo, tendrías que usar la opción de corregir (por ejemplo: A 4).

Seleccionemos la opción G del menú.

Los usuarios de 48K se habrán sorprendido al ver que no necesitaban cargar otro overlay, esto es porque Caracteres, Colores por Defecto, y Gráficos están todas en el mismo.

Si usas la opción Arreglar limpiará todos los overlays que PAW tenga en memoria y si deseas usar otra opción, excepto CARGAR/GRABAR o Memoria Libre, necesitarás cargar el overlay que las contiene.

Pero continuemos con los gráficos.

Para cada dibujo del juego hay varias opciones que permitirán: Arreglar, Imprimir en pantalla, Copiar a una impresora, calcular el tamaño de un dibujo o hacer un Volcado (Dump) de la pantalla.

Como nosotros vamos a hacer el dibujo de la localidad 4, hemos de teclear [A 4 ENTER]. La pantalla se limpiará dejando sólo dos líneas de información en la zona baja.

La información proporcionada, de izquierda a derecha es la siguiente:

TEXTO 4: [Estoy en un camino de grava que va de Este a Oeste, muy cercano a un cómodo banco. Hacia el Sur hay un cuidado césped y hacia el Norte hay un pabellón de música].

TEXTO 5: [Estoy en el pabellón de música. Al Sur hay un camino de grava].

TEXTO 6: [El sol crea bellos efectos de luz en la superficie del estanque, mecida por una suave brisa. Hay un camino al Norte que termina en un lloroso sauce. Al Este queda un cuidado césped].

TEXTO 7: [Estoy al lado de un sauce llorón. Al Sur queda el estanque].

TEXTO 8: [Estoy sentado en una rama del sauce, con una visión panorámica del parque lejos al Este, más allá de la verja del parque, puedo ver la parada del autobús].

Usa P para chequear lo que has tecleado y cuando la pantalla se llene de texto aparecerá el mensaje "más..." en la parte inferior de la pantalla. Si entonces pulsas cualquier tecla (excepto BREAK, ESPACIO o N) aparecerá otra pantalla llena de texto y así hasta llegar al final donde aparece el mensaje "PULSE CUALQUIER TECLA".

BREAK, ESPACIO y N causarán un BREAK error y te permiten salir del listado.

Vale. Ahora tenemos varias localidades pero no hay modo de llegar de una a otra.

Eso nos deja dos opciones por ver; la opción B con la cual se inicia una nueva página de memoria en los 128K. De momento no tocarla, porque el juego de práctica cabe sobradamente en la página 0.

Sobre la opción B, trataremos más extensamente en la guía técnica.

Con la opción Z, como agudamente habrás deducido, volveremos al menú principal.

## CONEXIONES

Desde el menú principal, tecleando (C) vamos a un menú similar al de las localidades. Date cuenta que las entradas pueden ser corregidas con (A), presentadas en pantalla con (P), o sacadas por impresora con (L).

La razón de esto es que PAW ya ha insertado una línea en blanco en la tabla de conexiones cada vez que tú has puesto una nueva localidad. Si usas la opción (P) para mirar a la tabla veras que ya hay entradas en blanco para localidades 0 a 8.

Si miramos otra vez nuestro mapa veremos las conexiones que se requieren entre localidades. Desde la localidad 2 (parada autobús) necesitaremos solamente el movimiento OESTE que nos llevará a la localidad 4.

Por lo tanto, hay que teclear (A 2) para corregir la entrada de la localidad 2, y luego teclear (OESTE 4) lo cual instruye a PAW que, cuando el jugador teclee la palabra OESTE y esté en la localidad 2, debe ser movido a la localidad 4.

Si usamos ahora la opción (P) para examinar la entrada que acabamos de hacer (date cuenta de que con P 2 se imprimirán solamente las entradas a partir de la localidad 2 hacia arriba), entonces encontrarás que la entrada esté más o menos:

Localidad 2 OESTE to 4  
etc.

Esto es porque PAW sabe que O es un sinónimo de OESTE (un sinónimo es una palabra que significa lo mismo). PAW usará siempre el sinónimo más corto.

Para la localidad 3, necesitamos 3 conexiones: NORTE (a 4), OESTE (a 6), NOROESTE (a 7).

## EL EDITOR GRAFICO

El sistema usado por PAW para dibujar es el llamado de "Línea y Relleno" que es muy eficiente en el uso de la memoria para el tipo de gráficos que se necesitan en los juegos de aventuras.

La forma como actúa es, en vez de guardar la imagen que has pintado en forma de pantalla (screen) como hacen muchos programas comerciales, lo que hace es que guarda una lista de los comandos que usaste para dibujarlo.

De esta forma, incluso las imágenes o los dibujos más complejos consumirán solamente 2K de memoria en comparación a las 7K que lleva una pantalla normal. De hecho, encontrarás que se puede hacer algo incluso con unos 100 bytes, cuando tengas práctica, por supuesto.

La lista de comandos que se han guardado es lo que se llama una "cadena de dibujo", y hay una cadena de dibujo para cada localidad que insertes usando la opción de Localidad del menú principal. Es decir, PAW te reserva ya un espacio para cada pantalla que tú has creado cuando has hecho el texto. Si las dibujas o no ya es cosa tuya. Por lo tanto, si solamente dibujas unas cuantas, las otras cadenas de dibujo estarán vacías.

Cada dibujo, (y por lo tanto cada localidad) tiene un color definido de Papel y de Tinta. Seleccionando la opción [D] del menú se te presentará un sub-menú en el cual puedes arreglar estos valores. Usa [P] para ver que ya existe una entrada para las 9 localidades que hemos creado en el juego de demostración.

Todas ellas estarán de momento marcadas como sub-rutinas, lo cual dice a PAW dos cosas:

- 1/ No dibujes éste gráfico cuando describas una localidad.
- 2/ Este es un sub-dibujo que puede ser usado en otros gráficos.

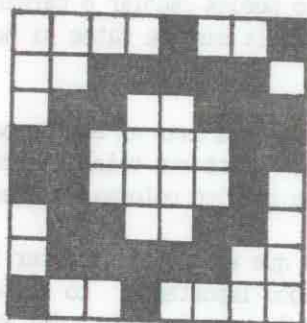
La facilidad de tener varios sub-dibujos es muy similar a la idea que usábamos antes con las tablas de sub-procesos. Luego veremos un ejemplo de ellos.



Usa las teclas del cursor (CAPS SHIFT 5 hasta 8 en un 48K) para mover el cursor rojo parpadeante alrededor del entramado, y la tecla de ESPACIO para cambiar la situación de un pixel. Es decir si el pixel está encendido (negro) lo pondrá apagado; y si está apagado (amarillo o blanco) lo encenderá. Intentalo y verás lo que queremos decir.

El sombreado o el diseño que requerimos es el que muestra el Diagrama 5. Cuando hayas terminado usa [R] para volver a dibujar los dos cuadros donde se muestran los sombreados y verás la nueva forma de entramado. Finalmente pulsa ENTER para terminar la edición.

Diagrama 5



La cara B de la cinta contiene 22 diferentes sets de caracteres, que pueden ser cargados en los sets de caracteres del 1 al 5, después de haberlos insertado, por supuesto.

Después de la inserción de un set, la opción de usar ése set como set principal o set por defecto se ofrece en el menú de Colores de Fondo.

Las conexiones que quedan son las siguientes:

Localidad 4 (N 5), (E 2), (S 3) (SO 6) (O 7)  
 Localidad 5 (S 4), (SO 7)  
 Localidad 6 (N 7), (NE 4), (E 3)  
 Localidad 7 (ARRIBA 8), (NE 5), (E 4), (SE 3), (S 6)  
 Localidad 8 (BAJAR 7)

Ahora corrige las entradas de la localidad 0 para que cualquier movimiento nos lleve a la localidad 2, donde comenzamos el juego (hay una forma mejor de hacer esto, pero necesitamos para ello usar una tabla que todavía desconocemos, y que ya te mostraremos más adelante).

Ahora solamente falta teclear (A 0) ENTER (N 2) ENTER, y ya nos indica que de la localidad 0 al NORTE iremos a la 2.

Comprueba todas estas entradas mirando el mapa hasta que estés seguro de que todo sea correcto. Recordamos que usando la Z se puede volver al menú principal y que desde el menú principal con la opción S se puede grabar en cinta la base de datos actual.

También puedes, con la opción E, examinar el resto del menú que no cabe en la pantalla.

## JUGANDO EL JUEGO

Ha llegado el momento de intentar probar nuestro juego. La opción requerida para ello es (T) en el menú principal. Te preguntará si quieres diagnóstico o no, por el momento sólo tecleemos N y ENTER, porque no sabemos todavía para qué sirven los diagnósticos.

Ahora debe de aparecer la pantalla con el título que tecleamos al comienzo. La línea de INPUT se usa para poner las órdenes para que el PAW las interprete y ejecute tus acciones de acuerdo con la información que le hayas dado cuando escribiste el juego.

De momento, solamente le hemos dicho hacia donde nos puede llevar cuando una dirección se teclea. Eso es lo que intentaremos probar.

Ahora empecemos el juego correctamente tecleando NORTE desde la localidad 0 y, por supuesto, ENTER.

Recordemos que con DELETE podemos corregir cualquier error que hayamos cometido en la línea de INPUT.

Veremos entonces que la pantalla se limpia y que la descripción para la localidad 2 aparece. Si ello no ocurre, probablemente la entrada en las conexiones esta equivocada. No te preocupes, porque se puede volver otra vez al editor tecleando ABANDONAR o RETIRAR (que es el comando que el PAW conoce desde el principio) y respondiendo S a la pregunta que el PAW hace y contestando N a la siguiente pregunta de que "si quieres probar otra vez" el juego o no.

Vuelve a las conexiones y corrige lo necesario.

Ahora hay que probar moverse entre las localidades viendo todos los movimientos posibles (toma nota de cualquier movimiento erróneo para poderlo corregir cuando vuelvas al Editor).

También es tiempo de hablar de algunas órdenes que PAW ya tiene dentro y que conoce.

Por ejemplo, tecleando M o MIRAR se repite otra vez la descripción de la localidad, lo que es muy útil para el jugador si hay un montón de texto en una descripción que ya ha hecho SCROLL.

Tecleando I o INVENTARIO harás una lista de los objetos que lleves en este momento, por ahora llevarás solamente un objeto desde el comienzo, pero no podrás hacer nada con él.

Por cierto, en este momento cabe destacar que, puesto que PAW acepta hasta 120 palabras juntas, se puede hacer un recorrido por todo el juego tecleando todas las direcciones de una vez. Inclusive durante el recorrido puedes hacer un inventario.

## EL EDITOR DE CARACTERES

Seleccionando la opción Q del menú principal (los usuarios de 48K tendrán que cargar un overlay) tendremos el Editor de Caracteres.

Este sub-menú te permite cambiar la forma de los caracteres que se muestran en la pantalla. Se pueden tener hasta cinco sets de caracteres diferentes en memoria y en cualquier momento cambiarlos entre ellos usando ESCC 0-5 o una acción CHARSET en Procesos o Respuestas.

Los sets de caracteres están numerados de 0 a 5, set 0 es el set normal y no puede ser cambiado excepto por los caracteres que tengan valores de 0 a 15, que son los diferentes tipos de sombreados, y también los del 144 al 185 que son los GRAFICOS DEFINIDOS por el USUARIO normales del Spectrum.

Si usas en este momento [P] para mirar a la tabla, encontrarás solamente éstos caracteres presentes. Tendrás que insertar un set en blanco antes de que lo puedas cambiar o cargarlo. Esto se hace para conservar memoria en la base de datos si no estás cambiando sets de caracteres.

De momento, solamente vamos a usar el editor para cambiar una de las formas de sombreado. Estos son solamente caracteres normales con los cuales el sistema gráfico colorea un área de la pantalla.

[A 0 15 ENTER] permitirá que edites el carácter 15 del set 0. Como éste es un sombreado poco importante, lo alteraremos para que represente el trabajo de la reja de hierro del Pabellón de Música.

Cada carácter de PAW está definido por un entramado de 8 pixels por 8 pixels. La parte izquierda superior de la pantalla mostrará una versión agrandada del diseño tal y como es en este momento. En la parte media de arriba y en la derecha te mostrará cómo se ve cuando se use como un sombreado (de ambas formas: la normal y la inversa), mientras que la parte de abajo de la pantalla muestra un sumario de las órdenes a tu disposición y el set de caracteres actualmente editado.



## COMPRESION DE TEXTO

La opción K del menú principal (los usuarios de 48K tendrán que cargar un overlay en este punto) preguntará si quieres comprimir la base de datos, cualquier tecla que teclées aparte de la S de "si" te mandará de vuelta al menú principal.

El compresor de texto reducirá la cantidad de memoria que se necesita para el texto de tu juego agrupando las letras más comunes en "tokens". Esto puede tomar desde un minuto hasta una hora dependiendo del tamaño de tu juego. En este juego de demostración que estamos haciendo llevará más o menos un minuto y salvará unos 900 bytes.

La única diferencia que encontrarás es que, cuando estás editando un texto que ya existe, el cursor pegará saltos de dos, tres, y a veces cuatro y cinco caracteres de una sola vez. Lo mismo pasará al borrar. Si se hace alguna corrección hay que teclear todas las letras separadas, puesto que ya serán comprimidas la próxima vez que uses el compresor. Es de notar que el compresor usa los tokens normales del Spectrum, lo que producirá agrupaciones de letras, pero no de las palabras claves después de que uses el compresor.

Muy importante es que no uses los tokens, o sea, las palabras clave, si luego vas a comprimir la base de datos, (es decir, no pongas tokens dentro de tu texto si después vas a comprimir). Y para saber lo que son tokens mira el manual de tu ordenador.

## OBJETOS

Objeto es cualquier cosa que el jugador puede manipular dentro del juego, por ejemplo: una manzana que puede comer, una llave que puede usar para abrir una puerta, una espada para pelear, etc.

En nuestro juego, que es bastante simple, usaremos los siguientes objetos (no todos ellos tendrán una función en el juego finalizado):

Objeto 0	una antorcha encendida.
Objeto 1	una bolsa.
Objeto 2	un emparedado.
Objeto 3	una manzana.
Objeto 4	un billete de autobús.
Objeto 5	una piedra.
Objeto 6	un anorak.
Objeto 7	una antorcha apagada.

Muy Importante: La antorcha se trata como dos objetos separados. Cambiaremos de uno al otro cuando el jugador la apague o la encienda.

Otro punto a destacar es el hecho de que PAW, al hacer un inventario, muestra la descripción de los objetos eliminando la primera palabra ("una"). Para evitar esto, se deberá insertar al comienzo de cada descripción un código de cambio de color, por ejemplo [EXTRA y 0] para poner el paper negro. Este truco evita que PAW se coma la primera palabra.

La opción 0 del menú principal sirve, como muy agudamente habrás deducido, para poner las descripciones de los objetos, de una forma muy similar a como hemos puesto las descripciones de las localidades.

Veremos que también existe un objeto 0 si se usa la opción P, para hacer una lista de ellos.

Por lo tanto hay que corregir esta entrada del objeto 0 tecleando (A 0) y usando un color de tinta diferente. Por ejemplo, si queremos usar el color Cyan, hay que pasar al modo extendido o extra en el 128 y luego teclear al mismo tiempo CAPS SHIFT + 5.

No te olvides de volver al color blanco otra vez, esto se logra con el modo extendido y luego CAPS SHIFT y 7 al mismo tiempo. Esto debe hacerse al final de cada texto.

Volvamos al menú principal para que le podamos decir a PAW más cosas sobre nuestros objetos. Con la opción I tenemos donde está cada objeto inicialmente, es decir, cuando la aventura comienza.

Vemos que no hay opción de insertar puesto que eso ya queda hecho automáticamente por PAW cuando en la tabla 0 se ha puesto un objeto. Date cuenta que corregir (Amend) (A) tiene dos parámetros; el número del objeto y su posición.

Esta posición tiene varios valores especiales que son muy importantes (son localidades que no existen);

252 es para objetos no creados, por ejemplo, todavía no existen dentro del juego.

253 tiene todos los objetos llevados encima (puestos) por el jugador.

254 tiene todos los objetos llevados por el jugador, pero no puestos encima.

Por ejemplo, si queremos hacer de la antorcha encendida un objeto no creado hay que teclear [A 0 252] y como antes estaba como objeto llevado, el mensaje AMEND que significa corregido será impreso para mostrar que PAW ha hecho la corrección.

Tecleemos ahora las siguientes posiciones iniciales, pero sin poner los comentarios que van después del punto y coma (;).

Luego, cuando hayas terminado, usa la opción P para corregirlos.

Una vez que una base de datos se ha cargado, se te presentará un menú como siempre. Si hay algún error puedes volver al menú principal, volver a seleccionar la opción y probar de nuevo.

Es importante que tengas en cuenta que cualquier overlay cargado previamente será borrado si ocurre un Error de Carga, así que a menos que tengas suficiente memoria para mantener los overlays principales, no podrás hacer nada, excepto cargar un nuevo overlay o hacer un Save (Grabar) o un Load (Cargar) de la base de datos. Lo cual siempre está a tu disposición.

Si no tienes un contador de revoluciones o quieres hacer las cosas más sencillas se puede pasar cada una de las bases de datos a una cinta aparte. Son sencillamente bases de datos de tipo CODE.



## EL USO DE LOS OVERLAYS

El usuario de un 128K probablemente no necesita usar paginación todavía, pero puede encontrar beneficioso también este capítulo.

Ya explicamos antes el concepto de las paginaciones. Para poder pasar los gráficos y la compresión del texto, por ejemplo, los usuarios de 48K tendrán que cargar un overlay o página aparte.

PAW hará casi todo el trabajo para tí. Si seleccionas la opción de menú que necesitas, se te pedirá que confirmes si quieres cargar o no un overlay. Pulsando cualquier tecla que no sea la S o la Y (de yes) te mandará al menú principal. Si continúas, PAW imprimirá el nombre del overlay que está buscando en la pantalla.

Los cinco overlays se encuentran al final del programa principal, que es donde la cinta quedará situada después de cargar el PAW. Si tienes un contador de revoluciones, es importante que lo pongas a cero en este punto, hagas una lectura de donde comienza cada overlay y así puedes ir directamente a cada uno de ellos.

Los cinco overlays están en el orden que mostramos más abajo, y también describiremos las opciones que contienen cada uno:

- PAWOVR 1 Lleva el Interpreté, el probador del juego, Grabar y Cargar la aventura ya terminada.
- PAWOVR 4 Lleva las tablas de Procesos y Respuestas, el Vocabulario, las Conexiones, y Palabras.
- PAWOVR 5 Lleva los Mensajes, Localidades, Peso de Objetos, Objetos-Inicialmente-En y Colores de Fondo.
- PAWOVR 2 Lleva un Compresor.
- PAWOVR 3 Lleva un Editor de Caracteres y el Editor Gráfico.

Es de notar que el Grabar, Verificar y Cargar una base de datos, junto con la Memoria Libre, están siempre a tu disposición, puesto que son parte del menú principal.

- Objeto 1 2; porque la bolsa se encuentra en la parada del autobús.
- Objeto 2 254; porque el jugador lleva el emparedado.
- Objeto 3 254; y la manzana.
- Objeto 4 8; significa que el billete está arriba del árbol.
- Objeto 5 3; una piedra que está en la hierba.
- Objeto 6 253; porque el jugador lleva puesto encima el anorak.
- Objeto 7 254; y lleva consigo la antorcha apagada.

Ahora vamos a decirle a PAW más cosas acerca de los objetos; por ejemplo, su peso relativo, o si son capaces de contener a otros objetos, o si el jugador puede llevarlos puestos encima.

Volvemos al menú principal y tecleamos la opción X, con ello pasamos al menú de peso de objetos, que también nos permite poner los otros dos atributos (contenedor y llevable encima).

Como siempre, PAW ya ha creado una entrada en esta tabla para todos los objetos que se han insertado previamente. De hecho ya le ha dado a cada objeto el peso de una unidad, no contenedor, y no llevable. Esto lo asigna a cada objeto por defecto (por defecto significa el que PAW le da, si tu no le das ningún valor).

Necesitamos probablemente corregir el anorak y la bolsa, puesto que el jugador es capaz de llevar puesto encima el anorak (de hecho el jugador está haciendo esto desde el principio cuando comienza el juego) y la bolsa debe ser capaz de contener otros objetos.

Además, el anorak y la bolsa son mucho más pesados (relativamente) que los otros objetos y hay que cambiarles su valor a 3 unidades cada uno.

Cuando se teclea A para corregir aparecen 3 valores; uno es el número del objeto, otro es el peso unitario del objeto, y finalmente el tercero son los otros atributos, que pueden ser:

- 0 = ninguno.
- 1 = contenedor para otros objetos.
- 2 = el jugador puede ponérselo y quitárselo.
- 3 = un contenedor que puede ser puesto y quitado (ejemplo: unos pantalones que tienen un bolsillo).

Volvamos a nuestra bolsa, (objeto 1) que es un contenedor y pesa 3 unidades. Entonces necesitamos teclear [A 1 3 1] (NO TE OLVIDES DE PONER LOS ESPACIOS).

El anorak no tiene bolsillos (por lo menos en este juego no los tiene), pero puede ser llevado encima y quitado, por lo tanto hay que teclear [A 6 3 2].

Usemos la opción P para examinar las entradas que hemos hecho, y deben de ser:

```
Objeto 0  Pesa 1
Objeto 1  Pesa 3  C ;C significa que es un contenedor. (1)
Objeto 2  Pesa 1
Objeto 3  Pesa 1
Objeto 4  Pesa 1
Objeto 5  Pesa 1
Objeto 6  Pesa 3  WR ;WR significa que se puede poner y quitar.(2)
Objeto 7  Pesa 1
```

Ahora podemos probar la aventura otra vez para asegurarnos de que todos los objetos estén donde tienen que estar. Pero todavía no se puede hacer nada con ellos, necesitamos decirle a PAW qué palabra describe cada objeto.

También es el momento de hacer un SAVE de la Base de Datos, pero teniendo el cuidado de que sea en una nueva sección de la cinta y usando un diferente número de versión.

- 2/ El pájaro debería de verdad volar, escapándose de tí si intentas coger el emparedado mientras está presente. Por ejemplo, se supone que estaría picoteando el emparedado y cualquier pájaro se las piraría.
- 3/ "DESATA \_" y "ATA \_" deben tener un mensaje más o menos parecido a "¿Atar qué? y ¿A qué?", porque la forma en que lo pusimos con NOTDONE era una salida bastante fácil.
- 4/ Si el jugador intenta teclear PONER un objeto DENTRO DE LA BOLSA y la bolsa no está presente, tal como lo tenemos de momento lo único que pasará es que dejará caer el objeto, ¿por qué?. Arréglalo.
- 5/ Nada se ha hecho todavía con la antorcha. Las siguientes entradas te permiten encenderla y apagarla (pero debes de tener la palabra encender y apagar en el Vocabulario):

```
ENCIENDE  ANTORCHA  CARRIED  7
                               SWAP  7  0
                               OK

APAGA      ANTORCHA  CARRIED  0
                               SWAP  0  7
                               OK
```

Debes mirar todos los condaccs que no hemos puesto aquí en la guía técnica, y leer con atención el capítulo sobre claridad y oscuridad. A lo mejor se podría añadir a esta demostración un sótano debajo del pabellón de música. El movimiento debería ser puesto también en la tabla de Respuestas con una entrada del tipo de: (suponiendo que 9 sea la nueva localidad).

```
BAJAR  _  AT      5      ;¿Está el jugador en el Pabellón?
        SET      0      ;Si la bandera 0 = 255 = Es de noche
        GOTO     9      ;Nueva localidad
        DESC
```

No debes olvidarte de poner una entrada para SUBIR que limpie la bandera otra vez a 0.

- 6/ ¿Qué pasa si el jugador intenta subir al árbol? y cómo sería otra manera de poner esto?. Como clave te diremos que solamente hay una cosa que se puede subir en esa localidad.



DECIR PAJARO SAME 12 38  
MESSAGE 8  
DONE

DECIR \_ MESSAGE 23 ;Que pregunte: ¿Quién?  
DONE

No ponemos nada para la preposición A, esto permite al jugador recortar su orden si así lo quiere. En líneas generales no se debe de hacer una comprobación para una Sentencia Lógica extendida, a menos que sea necesario para diferenciar entre dos frases similares.

Como prueba final, ahora debes jugar la aventura intentando atar al perro, hablarle y todas esas chorradas. Por ejemplo, cuando estemos en el camino cerca del banco del parque, con la correa y el perro, intenta PONER LA CORREA AL PERRO Y ATARLO AL BANCO.

Luego para desatarlo digamos: DESATA PERRO.

Si estamos arriba del árbol con la bolsa, digamos PON TODO EN LA BOLSA Y SUELTA. BAJA Y MIRA DENTRO DE LA BOLSA.

Para hacer que el perro se siente tecléa: DECIR AL PERRO "SIENTATE".

Para hacerle volver: DECIR AL PERRO "VEN AQUI".

¿Funciona?... Malegro.

#### MEJORAS POR TI MISMO

Antes de que continuemos con los gráficos, aquí hay unos cuantos puntos que convendría arreglar en el juego de demostración, para que sirvan como práctica en el uso del sistema.

- 1/ EXAMINA debe responder a todos los objetos aunque sea con una respuesta general como "No veo nada especial en el \_".

#### VOCABULARIO

Tiene un menú más complejo que los que hemos visto hasta ahora, pero la mayoría de las entradas son sólo recordatorios de las opciones a tu disposición.

El Vocabulario es una lista de las palabras que PAW puede reconocer si el jugador las teclea durante el juego. Por lo tanto, cualquier palabra que no esté en esta tabla no tendrá ningún efecto.

Hemos dejado un vocabulario inicial que ya contiene los verbos, nombres, etc., más usados en las aventuras.

Cada entrada para una palabra consiste en un máximo de cinco letras que, o bien cubren la palabra completa, por ejemplo NORTE, o son las cinco primeras letras de otra palabra más larga, por ejemplo describir, seguido del valor para esa palabra y del tipo de palabra (por ejemplo: Nombre, Verbo, etc.).

El usar solamente cinco letras para guardar una palabra reduce la cantidad de memoria requerida para guardar el vocabulario entero; también reduce la longitud de los INPUTS del jugador, y acelera la búsqueda de palabras por el propio PAW. Además, cinco letras son suficientes para establecer una diferencia entre la mayoría de las palabras usadas corrientemente.

El menú permite la INSERCIÓN y el BORRADO (delete) de palabras; el LISTADO de entradas hechas para cada tipo de palabras, y la inspección de sinónimos.

Si tecleamos S OESTE veremos que PAW muestra el sinónimo O también.

Si tecleamos P 2 tendremos una lista de todos los nombres que hay actualmente en PAW (los números que corresponden a cada tipo de palabra se muestran al lado derecho del menú).

También encontraras que PAW tiene ya en su Vocabulario todas las direcciones del compás y algunas otras útiles.

Necesitamos ahora aumentar el número de NOMBRES para poder poner las palabras que representan nuestros objetos.

Todos los nombres con un valor menor de 50 son nombres propios, por ejemplo: nombres de personajes o lugares, pero más específicamente, para PAW son nombres que no se pueden cambiar por por la terminación verbal lo o la.

Explicándolo mejor: para mayor rapidez PAW detecta cuales palabras son nombres propios (menos de 50) y cuales pertenecen a objetos manejables (mayores de 50).

En la frase "coge el martillo, golpea a Manolo en la cabeza y déjalo", si has dado a Manolo un valor menor de 50 y a martillo uno mayor, PAW entiende que el la final del verbo se refiere al martillo.

Otro ejemplo: en la frase "dale a Carlos una tortilla y quitasela", entenderá que el la final se refiere a la tortilla.

El lo o la también puede ponerse antes del verbo con iguales resultados.

Otro hecho importante es que las palabras con valor inferior a 20 son nombres, y que si PAW no encuentra un verbo en la frase los convertirá temporalmente en verbos.

Por ejemplo: la palabra NORTE es un nombre y puede ser usado en la frase "ir al norte"; pero también puede ser tecleada sola, en cuyo caso como tiene un valor inferior a 20 se comportará como un verbo (o sea, que te vas al norte).

Finalmente, las palabras con un valor menor de 14 se tomarán siempre como palabras-movimiento (cualquier palabra que sea una dirección) y sirven para determinar el mensaje que PAW imprimirá si no puede hacer nada con esa frase (por ejemplo, determinará si te responde con un "no puedo" o "no puedo ir en esa dirección").

Hay que notar que esto de "menor de 14 es palabra-movimiento" se aplica tanto a verbos como a nombres convertibles en verbos.

Puesto que todos nuestros objetos son manejables, les daremos un valor mayor de 49.

ATAR \_ NOTDONE ;Para asegurarse de que no pueda

DESATAR PERRO LET 34 55 ;La bandera 34 es el Nombre de la  
SL

DESATAR CORREA AT 4 ;Donde está el banco  
EQ 14 2 ;Si ya está el perro atado  
CLEAR 14 ;La bandera indica que está libre  
MESSAGE 25 ;Díselo al jugador  
CREATE 5 ;Hay que volver a crear la correa  
GET 5 ;Dársela al jugador, cogerla  
DONE

DESATAR \_ NOTDONE ;Es para asegurarse de que no pueda  
desatar otra.

CREATE

Es una acción que debe ser seguida por un número de objeto. Ello causa que el objeto aparezca en la posición donde está el jugador.

GET

Es una acción que si está seguida por un número de objeto, intenta coger el objeto especificado.

Usamos estas acciones en vez de poner el objeto simplemente en 254, porque puede haber problemas debido al peso o al demasiado número de objetos que se lleven, y entonces estos problemas deben de producir información.

Finalmente pondremos las entradas que permiten hablar al perro. Hemos también incluido algunas entradas necesarias para que se te permita hablar al pájaro, pero que él te ignore.

DECIR PERRO SAME 13 38 ;Que esté aquí  
PROCESS 5 ;Se le manda a esa tabla para que  
haga el trabajo.  
DONE



Ahora una entrada relativamente simple para que se haga cargo de:  
PONER LA CORREA AL PERRO:

PONER	CORREA	PREP	ENCIMA			;Para estar seguros si el jugador teclea eso
	NOUN2	PERRO				
	CARRIED		5			;Que el jugador tenga la correa
	SAME		13	38		;Que esté en la misma localidad que el perro
	LET		14	1		;Ahora el perro tiene la correa
	DESTROY		5			;por lo tanto, el jugador no la puede tener
	MESSAGE		21			;Se le dice.
	DONE					

En las entradas que siguen, introduciremos un nuevo concepto. Se trata de la modificación de la Sentencia Lógica actual.

Nosotros queremos que el juego entienda ambos, ATAR EL PERRO AL BANCO y ATAR LA CORREA AL BANCO puesto que son la misma cosa. Pero tenemos que correa y perro tienen diferente valor como palabras, así que la entrada de ATAR PERRO irá primero en la tabla (puesto que su valor es más bajo que correa). Por lo tanto, hay que convertir el Nombre en CORREA (55) y permitir que PAW lleve a cabo una entrada que equivalga a ATAR CORREA. Un sistema similar hay que utilizar para DESATAR.

Esto es bastante importante, y lo utilizarás en tus juegos con bastante frecuencia. Hay que poner las entradas:

ATAR PERRO LET 34 55 ;La bandera 34 es el Nombre de la SL

ATAR	CORREA	PREP	A			
	NOUN2	BANCO				
	AT		4			;Donde está el banco
	SAME		13	38		;El perro está aquí
	EQ		14	1		;Tiene la correa puesta
	PLUS		14	1		;Ahora atado al banco
	MESSAGE		22			;Se informa al jugador sobre ello
	DONE					

ANTORcha	50
BOLSA	51
EMPAREDado	52
MANZAna	53
BILLEte	54
PIEDRa	55
ANORak	56

Usa la opción I para insertar estas 7 palabras como nombres (tipo 2). Por ejemplo, tecléa [ I ANTOR 50 2 ], etc.

Con P 2 se puede comprobar que estos nombres están ahora en el Vocabulario.

También necesitaremos usar adjetivos para diferenciar entre las dos antorchas. Con P 3 podemos ver los adjetivos que PAW ya conoce.

Si no están ENCENDIDO y APAGADO insértalas procurando darles un número alto, pues son poco usados, y así PAW no tendrá que pasar a través de ellos cada vez que busque una palabra más común, por ejemplo un nombre o un verbo.

Todos los números desde el 2 al 254, están disponibles para cualquier tipo de palabra y no hay limitación en el uso de sinónimos.

Si tratas de insertar una palabra que ya esta presente, PAW te lo avisará, lo mismo que si tratas de borrar una que no esté presente.

Ten en cuenta que PAW solo tomará las primeras 5 letras cuando te refieras a una palabra e ignorará el resto.

Luego volveremos al Vocabulario, pero ahora vamos a decirle a PAW qué palabras definen nuestros objetos. Porque hasta ahora hemos descrito cada objeto, cuanto pesa, y dónde comienza, pero no le hemos dicho a PAW qué palabra identifica cada objeto.

Volver al menú principal.



## PALABRAS PARA LOS OBJETOS

La opción W es una tabla donde las palabras que están en el vocabulario se asocian con un objeto particular. En esta tabla solamente se puede corregir, imprimir en pantalla, o imprimir en impresora, puesto que PAW ya ha insertado una entrada en blanco para cada objeto cuando se hizo su descripción en la tabla de texto para objetos.

Así pues, tecleando P aparecerán 8 entradas en blanco para nuestros objetos.

La tabla de asociación de palabras y objetos permite poner un nombre y un adjetivo asociados al número del objeto. Nuestros objetos requerirán las siguientes entradas:

Objeto 0 ANTORCHA ENCENDIDA  
Objeto 1 BOLSA \_  
Objeto 2 EMPAREADO \_  
Objeto 3 MANZANA \_  
Objeto 4 BILLETE DE AUTOBUS \_  
Objeto 5 PIEDRA \_  
Objeto 6 ANORAK \_  
Objeto 7 ANTORCHA APAGADA

Este signo especial " \_ " significa que no hay palabras, o sea que entenderá cualquier palabra. Siempre hay que teclear \_ (este signo) si no hay un adjetivo que describe al nombre. Suponemos que con tu sagacidad habitual habrás notado que no se puede insertar un objeto, sino solamente corregirlo (A).

La primera de todas es la entrada que hace que los objetos que se dejen en el árbol caigan al suelo. Esto debe ir entre la entrada que se hace cargo de poner objetos dentro de la bolsa y la entrada normal de "DEJAR \_". Entonces [I PONER \_ 1] nos situará en esa posición. La entrada es:

PONER	_	AT	8				;Que el jugador esté en la rama
		WHATO					;Ya hablaremos de ello
		LT	51	255			;¿Es un objeto válido?
		EQ	54	254			;¿Es un objeto que llevas encima?
		MESSAGE	15				;Avisa que está al pie del árbol
		PUTO	7				;La pone ahí.
		DONE					

Esto es un ejemplo de cómo se crea una acción automática, viene a ser lo mismo que un AUTOG, etc.

### WHATO

Es una acción que mira el primer Nombre de la Sentencia Lógica actual en la tabla Objeto-Palabra, y la convierte en el número de un objeto. Este número se pone entonces en la bandera 51. La bandera 51 siempre contiene el número del último objeto que se ha usado, o el último del que PAW tiene referencia, y siempre que esté seteada, las banderas asociadas (números 54 y 57) también se setean.

La bandera 54 lleva la localización actual del objeto. Es conveniente que mires el manual técnico para las banderas 51, 54 y 57.

### PUTO

Es una acción que cambia de localidad al objeto últimamente tratado, hacia la localidad que se especifique. O sea, es una acción que cambia el objeto de que se trate hacia la nueva localidad especificada.

El Mensaje 15 contiene una raya. Siempre que PAW encuentra una raya en el texto (sea mensaje o sea localidad) la reemplaza con el objeto actual. Es decir, el mensaje se cambia para tener en cuenta el último objeto que se esté usando.

La última entrada se usa para obviar el vocabulario tan limitado que entiende el perro, haciéndole que para cualquier otra frase mueva la cola, por eso se pone la última con \_ \_ .

Recordar que PAW va buscando en toda la tabla de Procesos, y si no encuentra una igual a la que busca caerá en esta última entrada de " \_ \_ ".

### PARSE

PARSE le permite a PAW continuar buscando conductos si no encuentra una Sentencia válida. Hay que tener cuidado aquí, puesto que la actual Sentencia Lógica puede estar un poco enrevesada (el parser intentará sacar algún significado de todos modos) así que normalmente se debe insertar algún mensaje como "Parece no entender" o algo similar, y después un DONE para volver a la acción anterior.

Si se forma una Sentencia válida PAW empezará a buscar las siguientes entradas por una similar, como hacía en la tabla de Respuestas. PARSE debe ser solamente usada en un sub-proceso que sea llamado desde la tabla de Respuestas, puesto que no tiene ningún significado en otra tabla.

Date cuenta de como las entradas VEN y AQUI se hacen cargo de una cantidad de frases que el jugador puede usar después de que el perro esté sentado, porque ambas tienen, una delante y otra detrás, la palabra comodín, o sea, la raya.

Y por último, la entrada " \_ \_ " coge cualquier sentencia que se haya metido en la cadena y para la cual el perro no tenga una respuesta específica.

Selecciona la tabla de Respuestas para insertar algunas entradas extras que controlen el habla y el dejar caer objetos del árbol.

### DIAGNOSTICO

Con T del menú principal se selecciona la opción de probar el juego, o sea, hacer un test de la aventura.

Primero te pregunta si quieres o no quieres diagnóstico, si tecleas "si", aparecerá el título y la introducción, y en la parte de abajo se pide un INPUT al jugador.

El que hayas contestado que quieres diagnóstico no tiene efecto aparente todavía, pero si tecleas ENTER antes de cualquier otra cosa, el cursor desaparece y aparecerá una línea parecida a:

Flag 38 = 0 ?

con flag se refiere PAW a una bandera, en este caso la 38, que es la que lleva el número de localidad presente.

PAW contiene 256 de estas banderas, y en cada una se puede situar un número de 0 a 255. Se usan para indicar el estado en alguna parte del juego.

Por ejemplo, si decides que la bandera 11 se ponga en 1 cuando la puerta del parque esté cerrada, y haga un clear a 0 cuando esté abierta, esto servirá de marcador para que luego puedas dejar pasar o no al jugador cuando teclée ENTER.

Es decir, si la bandera estuviera a 0 podría pasar, y si la bandera estuviera a 1 le darías el informe de que la puerta está cerrada.

PAW tiene para su uso específico varias banderas (de la 0 a la 10 y de la 29 a la 59). Por eso, si el valor que aparece al lado de la bandera 38 es igual a 0, PAW sabe que tu localidad actual es 0 en este caso.

Para ver cómo funciona esto, teclea ENTER de nuevo (con ENTER cambiamos el uso entre la opción diagnóstico o continuar con el INPUT si no has tecleado ningún otro comando) y vámonos hacia el principio del juego usando la palabra NORTE o cualquier otra dirección que hayas puesto.



De nuevo, antes de cualquier otra cosa teclea ENTER y verás que la bandera ha cambiado a 2, esto es, por supuesto, porque estás en la localidad 2.

Flag 38 = 2 ?

Puedes mirar a los valores de cualquier otra bandera tecleando su número antes de apretar el ENTER, por ejemplo: 100 ENTER y entonces puedes ver el valor de esa bandera.

Flag 100 = 0 ?

Pero hay otra opción mucho más potente que te deja poner el valor que quieres que tenga una bandera. Ello se hace poniendo el signo = frente al número. Por ejemplo, si en este momento tecleas = 10 ENTER verás que la bandera 100 se ha hecho = a 10.

Flag 100 = 10 ?

La bandera 100 no hace nada en este juego y su valor no es importante, pero si te pones a practicar por tu cuenta ten cuidado de NO cambiar los valores de ninguna otra bandera de momento, porque te puedes tropezar con alguna otra bandera que SI sea importante, y el juego se te haga un enredo. Vuelve a la línea de INPUT pulsando ENTER y veamos que más puede hacer PAW.

Ahora si que seremos capaces de manipular los objetos en el juego. De momento la bolsa estará en la parada del autobús. Con nosotros llevaremos el emparedado, la manzana, una antorcha apagada, y el anorak puesto.

Usa el diagnóstico para ver el valor de la bandera 1 (se hace con ENTER 1 ENTER), y vemos que tiene el valor de 3, que es el número de objetos que se llevan en las manos, pero no puestos encima. Volvamos a la línea de INPUT y tecleemos "coger saco" y PAW imprimirá el mensaje "Ahora cojo el saco".

Es lo que se llama AUTO-REPORTING ( informe automático de cualquier acción que hayas hecho).

El número de cualquier inserción o corrección tiene un valor máximo de 255. O sea, que no debes de insertar más de 256 entradas con el mismo tipo de palabra (lo cual sería de todos modos bastante poco manejable), si quieres retener la habilidad de seguir insertando en cualquier parte de la lista.

El último cambio en la tabla de Procesos es insertar un sub-proceso que luego llamaremos desde la tabla de Respuestas para que se haga cargo de hablar al perro. El mecanismo es muy simple. Si el jugador incluye una frase "(" ")" en su sentencia de INPUT, entonces el parser guardará memoria del lugar donde esté y seguirá descifrando la frase entre comillas.

Hay una acción llamada PARSE que instruye a PAW para que decodifique la cadena de palabras que el jugador haya tecleado, haciendo entonces la Sentencia Lógica. Lo lógico es hacer esto en un sub-proceso porque PAW tratará de buscar una similitud entre la nueva Sentencia Lógica y el resto de la frase.

Entonces comencemos una nueva tabla de Proceso (que debe ser la 5) e insertemos las siguientes entradas:

*	*	PARSE			;Convierte una cadena a SL.
		MESSAGE 16			;No es una frase válida así que..
		DONE			;El perro no te entiende.
SIENTATE	_	ZERO 14			;¿El perro no está amarrado?
		SET 14			;Ahora está sentado quieto.
		MESSAGE 24			;Díselo al jugador (si está en el mismo lugar que el perro)
		DONE			
VEN	_	EQ 14	255		;El perro debe de estar sentado.
		CLEAR 14			;Ahora está normal.
		MESSAGE 18			;El perro te sigue.
		DONE			
_	AQUI	EQ 14	255		
		CLEAR 14			
		MESSAGE 18			
		DONE			



Similarmente a como hicimos con el pájaro, se requieren entradas en la tabla de Procesos 1, para informarle al jugador que el perro anda por ahí rondando:

```

- PERRO SAME 13 38 ;¿El perro está en la misma localidad?
  MESSAGE 17 ;Díselo al jugador.
  EQ 14 1 ;¿con la correa?
  MESSAGE 19 ;si, díselo al jugador.

- PERRO SAME 13 38
  EQ 14 2 ;¿Está el perro amarrado al banco?
  MESSAGE 20

- PERRO SAME 13 38
  GT 14 2 ;255 es mayor que 2 así que
  MESSAGE 24 ;Dile al jugador que el perro está
  sentado.

```

Ya que estamos en la tabla de Procesos 1, modifiquemos la entrada \* para que también tenga [LET 13 2] (antes del GOTO), esto hace que el perro empiece en la parada de autobús.

Ahora, para que el perro asuste al pajarito, necesitamos una entrada extra en la tabla de Procesos 3. Pero debe de ir delante de la entrada que decide que el pájaro deje caer el billete y después de la entrada que hace que el pájaro vuele. Así nos aseguramos que el pájaro volará con el billete si lo tiene y lo deja si no lo tiene.

Por lo tanto, necesitamos insertar otra entrada antes de la sexta. Usaremos [I \_ PAJARO 6] para hacerlo.

```

- PAJARO SAME 12 13 ;El pájaro y el perro estén en la misma
  localidad.
  LET 12 8 ;Sólo en el pabellón de música
  LET 5 3 ;Muévelo a la rama, espera 3 frases.
  AT 5 ;¿Está el jugador en el Pabellón de
  Música.
  MESSAGE 13 ;Dile que el pájaro se ha ido..

```

Esta última orden ha hecho que la posición del saco haya cambiado de la localidad 2 que era la parada del autobús a la localidad 254 (que es una localidad especial para los objetos que puedan ser llevados pero no puestos).

Fíjate que no hay ningún cambio en la tabla donde se localizan los objetos al comienzo, sino que lo que ha cambiado es la copia que se ha hecho de la base de datos cuando se empieza el juego.

Si ahora miras el valor de la bandera 1 (fíjate también como cuando se selecciona diagnóstico estará en uso la última bandera que hayas mirado) verás que su valor ha aumentado a 4.

Ahora prueba "quitar anorak" y recibirás el mensaje "No puedo quitarme el anorak, mis manos están llenas". Esto es porque PAW inicialmente (si tu no has dicho lo contrario) sólo permite al jugador llevar 4 objetos al mismo tiempo, y en algunos casos esta limitación puede impedirle al jugador quitarse vestido, etc., (de hecho, quitarse una prenda es cambiar la posición de ese objeto de la localidad 253, que es llevado encima, a la localidad 254 que es llevado en la mano).

Si, por ejemplo, dejamos el saco primero y tecleamos "quitar anorak" veremos que si se puede. Si miras otra vez a la bandera 1 te darás cuenta de que todavía tiene el número 4, esto es porque al quitarte el anorak has aumentado el número de objetos que llevabas en tus manos.

Ahora como prueba teclea las siguientes órdenes y fíjate qué hace cada una (usando las banderas):

```

Coger Bolsa
Quitar Anorak
Poner Anorak
Coger Manzana
Coger Billete de autobús

```

Hay que darse cuenta de que todas las respuestas, excepto la de la última orden, mencionan los objetos por su nombre, esto es porque estaban todos a la vista y por lo tanto el jugador sabía que existían (estaban en su propia localidad). Pero para un jugador que no conociera el juego, el billete todavía no existiría, y si al teclear "coger billete" ya en la respuesta lo mencionamos por su nombre, le estamos dando una pista muy importante. Y es que el PAW es muy cuco.

Si tratas de poner cualquier cosa dentro del saco descubrirás que PAW lo que hace es dejar caer los objetos al suelo, esto es porque no le hemos dicho a PAW todavía qué cosas pueden ser puestas dentro de la Bolsa. Solamente le hemos dicho de momento que es un contenedor, en el próximo capítulo trataremos de este tema.

Finalmente encontraremos un terrible "error" en nuestro juego: si tecleamos "coger la puerta", se nos dará la respuesta de "No hay ninguna de esas aquí" o "No está eso aquí". No debería decir eso, porque en la descripción hemos dicho que hay una puerta.

Este es un problema muy frecuente, y ocurre porque le hemos dicho a PAW que hay una manzana, un emparedado, etc., pero no le hemos dicho que existe una puerta, y si usas coger o dejar, etc, con cualquier palabra que no esté en el vocabulario entonces PAW asume que es un objeto que "no está aquí". Sin embargo, una vez que esa palabra entre en el vocabulario, PAW sabrá que no es un objeto (siempre que no haya una entrada para esa palabra en la tabla de relación-objeto-palabras) y dará la respuesta "no puedo hacer eso" que es la correcta.

Así que volvamos al menú principal tecleando FIN, ENTER, SI, ENTER, NO, ENTER y seleccionaremos de nuevo la opción de Vocabulario V.

Pongamos todos los nombres de las cosas que hayamos usado en la descripción, aunque no sean objetos, tales como:

Puerta	57	Camino	60	Arbol	63
Reja	58	Banco	61	Rama	63
Hierba	59	Estanque	62	Hoja	63

La bandera 14 contendrá varios indicadores:

- 0 si el perro está libre para andar por ahí.
- 1 si el perro tiene la correa alrededor del cuello.
- 2 si el perro está atado al banco.
- 255 si el perro está sentado muy tranquilo.

Desde la tabla de Procesos, con [B] empezamos una nueva tabla de Procesos (debe ser la tabla número 4) y ponemos una sola entrada:

PERRO NOTSAME	13	38	;¿No está el perro donde está el jugador?
LT	14	2	;¿Todavía se puede mover?
NOTAT	8		;¿El jugador no está arriba del árbol?
COPYFF	38	13	;Mueve el perro a la localidad del jugador.
MESSAGE	18		;Y dile, que lo atosigue.

Suponemos que eres capaz de entender qué es NOTSAME, NOTAT y COPYFF, en caso negativo, en la guía técnica te explicaremos para que sirva cada una.

LT

Es una condición que es positiva o favorable si la bandera especificada contiene un valor MENOR QUE el valor especificado. En éste caso [LT 14 2] solamente será positivo o acertado si la bandera 14 tiene un valor menor de 2.

Ahora vamos a la tabla de Procesos 2:

PERRO	PROCESS	4
-------	---------	---

es decir, que estás usando la tabla de Procesos 2 para enviar a PAW a la tabla de Procesos 4.

La entrada "\_Perro", (si usamos [P] para mirar a la tabla), debe estar antes de la entrada del pajarito (si no es así, es que has puesto las palabras del vocabulario de una forma diferente de la que dijimos). Es para asegurar que el perro se mueva a la nueva localidad del jugador antes de que el pájaro sea comprobado.

Mensaje 18

El perro me sigue moviendo la cola.

Mensaje 19

El perro va arrastrando la correa.

Mensaje 20

El perro está atado al banco con una correa.

Mensaje 21

Confiadamente, el perro me deja ponerle la correa alrededor del cuello.

Mensaje 22

He amarrado el perro al banco.

Mensaje 23

¿A quién se lo digo?

Mensaje 24

El perro se sienta tranquilo.

Mensaje 25

He desatado el perro del banco.

No hay verdadera necesidad de hacer la rutina para el perro en una tabla de Procesos separados, porque solamente hay una entrada, pero lo haremos por si quieres ampliar el juego más tarde.

La bandera 13 contendrá la localidad actual del perro.

Es importante que árbol, rama y hoja reciban el mismo número, porque no intentamos que sean manipulables, sino que solamente las entienda el PARSER. Si fueran manipulables, tendríamos que darles números diferentes. Esto es una consideración importante de diseño.

Ahora puede ser el momento de hacer otro test del juego para ver que la orden "coger puerta" produzca una respuesta correcta.

De momento hemos creado las localidades, las hemos conectado entre si, hemos creado y descrito los objetos, les hemos asignado una palabra desde el vocabulario, un punto de comienzo en el juego, un peso relativo, si eran llevables o quitables, y si eran contenedores.

El próximo capítulo tratará de cómo se crean algunos problemas y otros personajes para hacer que el juego sea más interesante.



## PROCESOS Y RESPUESTAS

### LA TABLA DE RESPUESTAS ( R )

La tabla de Respuestas (R) del menú principal es una forma especial de tabla de Procesos.

Una tabla de procesos puede verse como un lenguaje de programación secuencial (se ejecuta un comando en cada turno). Las órdenes que se llevan a cabo se llaman "CondAccs" porque se pueden dividir principalmente en dos grupos: condiciones y acciones.

Antes mencionamos que el PARSER de PAW divide las sentencias en frases, las cuales son organizadas entonces en lo que se conoce como SL (sentencias lógicas).

En el caso de direcciones como NORTE (que es una sentencia lógica por si sola), PAW usa la tabla de conexiones para descubrir hacia donde se ha de mover el jugador.

ANTES, sin embargo, PAW mira en la tabla de RESPUESTAS para ver si hay alguna entrada en ella que pueda asociar con la SL.

Es muy importante que quede claro el concepto de que cada frase que el jugador teclée y PAW entienda, debe tener su correspondiente entrada en la tabla de Respuestas, excepto por la mayoría de los movimientos, que han sido ya puestos en la tabla de Conexiones.

La parte más importante de una SL es el verbo, que muestra la intención de la frase; la siguiente en importancia es el *primer nombre*, que muestra el sujeto de la SL, por ejemplo en "coger manzana", coger es el objetivo y manzana es el sujeto.

Si ahora seleccionamos R del menú principal aparecerá el submenú de esa tabla.

Tecleando P para mirar la tabla e ignorando de momento cualquier otra entrada, vamos a considerar la entrada que pone:

I \_ INVEN

El jugador se puede desembarazar del maldito perro poniéndole la cadena y amarrandola al banco. Además, el jugador puede "hablar" al perro, lo cual le dará otro medio de desembarzarse de él diciéndole que se SIENTE o que se QUEDE.

Antes de examinar las entradas en las tablas de Procesos y Respuestas, necesitamos tener el control del perro por medio de las siguientes palabras en el vocabulario y mensajes en la tabla de Mensajes.

Verbos		Nombre	
ATAR	34		
DESATAR	35		
SENTAR	36		
QUEDAR	36		
VEN	37	AQUI	37

Los números te demuestran que sentarse y quedarse son lo mismo, y que también usamos un Nombre y un Verbo con el mismo número.

Mensaje 15

El \_ cae al suelo al pie del árbol.

No se debe cambiar el color de este mensaje, y el que pongamos una raya baja en él tiene un propósito especial, del cual ya hablaremos más adelante.

Los demás mensajes que hablen del perro pueden ser puestos de cualquier otro color, por ejemplo magenta (MODOS EXTENDIDO, CAPS SHIFT + 3) no olvidando nunca volver al blanco al final de cada uno de ellos. El color del mensaje le permite al jugador saber exactamente a quién se refiere cada mensaje.

Mensaje 16

El perro me mira con amor.

Mensaje 17

El perro está aquí.

COGER BILLETE SAME	12	38	;¿Está el pájaro en la misma localidad?
ISAT	4	252	;¿Con el billete en el pico?
CLEAR	5		; Esto lo fuerza a volar
NOTDONE			; "No puedo hacer eso".

Esta entrada se disparará antes de la entrada GET \_ y previene la respuesta "No hay uno de esos aquí" si el pájaro está presente con el billete.

#### **CLEAR**

Es una acción que va seguida por un número de bandera, y que pone esa bandera a 0. En este caso hará que el pájaro salga pitando, simulando su miedo a una gran mano que descende sobre él para coger su preciada posesión.

#### **NOTDONE**

Es una acción similar a DONE, pero que engaña a PAW pensando que no has hecho nada y por lo tanto hace imprimir el mensaje "No puedo hacer eso".

Y ahora, vamos al momento de la verdad. Hay que probar el juego para ver si el pájaro vuela del pabellón de música a la rama. Juega un rato para ver si el pájaro continúa con su existencia vagabunda. Después trata de dejar caer el emparedado en la misma localidad. Date cuenta, de que si no coges el billete antes de que el pájaro se pise, el pájaro lo cogerá otra vez.

Y menos mal que acabamos con el bendito pájaro. Es complicado, pero te enseñará un montón sobre el PAW.

#### **EL PERRO**

Este perro lo ponemos para complicar el juego un poco más. Simplemente seguirá al jugador donde quiera que vaya y espantará al pajarito. Como no se espera que el perro se suba al árbol, debemos impedir que el jugador pueda tentar al pájaro con el emparedado desde la rama. Para hacer eso dispondremos que cualquier objeto que se suelte desde la rama del árbol caiga hasta el suelo.

Las dos palabras (porque \_ es una palabra), indican el verbo y el nombre respectivamente. Como I es un nombre convertible en verbo (como vimos en la sección de Vocabulario) esto significa que si se teclea por sí solo, es decir, si es la única palabra que ha tecleado el jugador en la frase, será tomada como un verbo en la SL. La línea ( \_ ) indica que el nombre no es importante en esta entrada, significaría como el "no palabra" que vimos en la tabla de objetos y palabras.

Lo que esto significa es, que si el jugador teclea I sólo, PAW lo equipará con la primera entrada de la tabla de Respuestas que tenga la I, y ejecutará la orden que se describa al lado de ella.

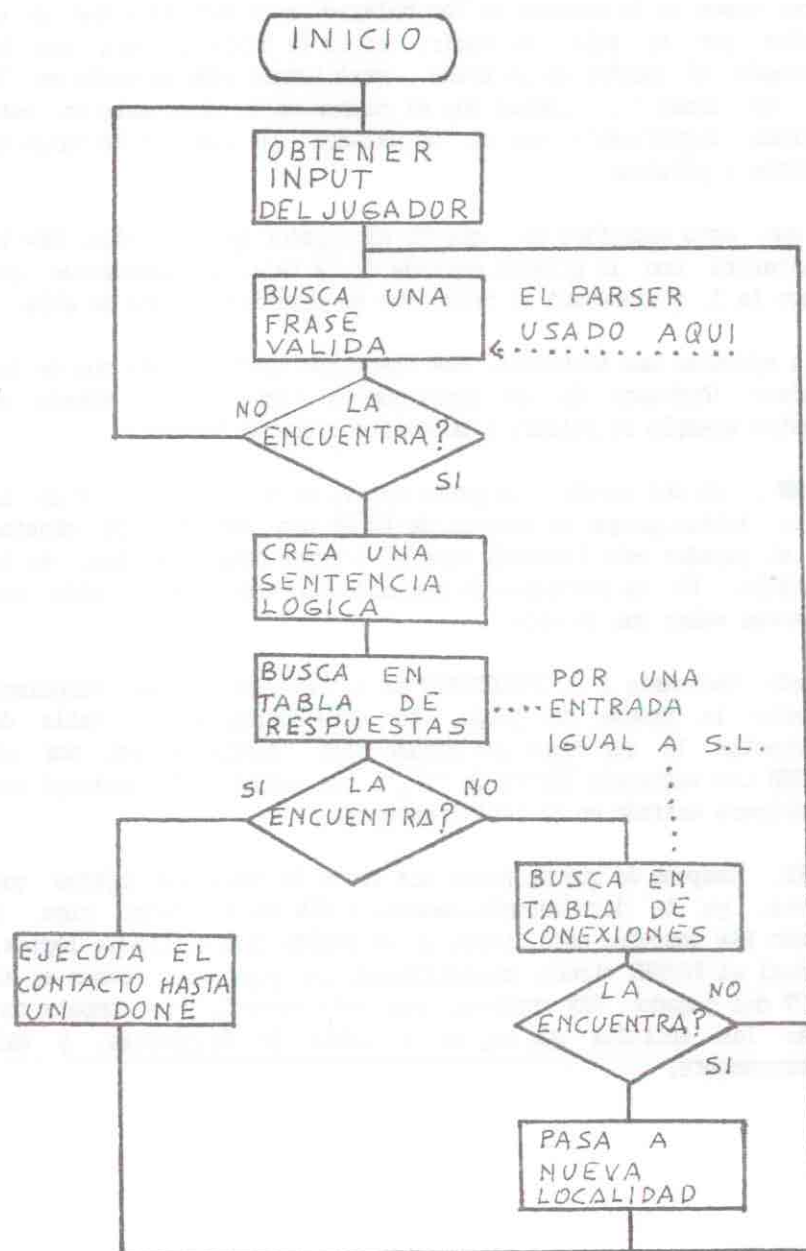
Para ejecutar una sentencia, PAW tiene que ejecutar cada uno de los condacc (órdenes) en una lista que ya tiene. Ahora viendo en nuestro ejemplo la palabra I solamente tiene un condacc.

**INVEN** : es una acción (la parte activa de un condacc!). Y es la parte activa porque se encarga de hacer una lista de los objetos que el jugador esté llevando consigo o tenga puestos encima, en la pantalla. No te preocupes de momento como hacer esto, sólo nos interesa saber que lo hace.

Cuando tecleaste I (o INVENTARIO si lo has puesto como sinónimo) durante la prueba del juego, fue esta entrada en la tabla de Respuestas la que causó que pasase algo, porque se creó por el PARSEER una sentencia lógica de "I \_", que entonces PAW emparejó con la primera entrada en la tabla-respuesta.

INVEN, después de que ha hecho una lista de todos los objetos que lleves, ya le dice automáticamente a PAW que ha hecho algo, y cuando PAW descubre esto, busca en el PARSEER otra sentencia lógica, la cual el PARSEER ofrece, decodificando las siguientes frases en el INPUT del jugador. PAW entonces coge esta nueva SL y la compara con todas las entradas que hay en la tabla de Respuestas, y así sucesivamente.

Diagrama 4



Así completamos las rutinas de control del pajarito. Pero necesitamos una entrada en el Proceso 2 para llamar a todas esas rutinas en cada ciclo, así que busquemos en la tabla 2 y pongamos la entrada que llama a todo el subproceso, es decir, la entrada "madre".

#### PAJARO PROCESS 3

lo que hará que PAW ejecute nuestros controles de pájaro cada vez que pase por ahí.

Ahora debemos asegurarnos de que el pájaro empiece en la localidad correcta y que el jugador sabe que el pájaro está ahí cuando la localidad se describe (o empezará a ver mensajes acerca de un pájaro dando la paliza sin haber tenido ninguna descripción de que existe tal pájaro).

Así que seleccionemos el Proceso 1 (que ya sabemos que se llama después de cada descripción de localidad) y vamos a corregir la entrada existente con \*\* poniéndole un [LET 12 8], lo cual hará que el pájaro esté en la rama al principio del juego. La entrada modificada debe de ser:

```

* * AT      0
    ANYKEY
    LET      12  8 ;El pájaro está en la rama (localidad 8)
    GOTO     2
    DESC
  
```

Y también hay que poner la siguiente entrada en la tabla de Procesos, para que le diga al jugador que hay un pajarito, y que tiene el billete.

```

- PAJARO  SAME  12  38
          MESSAGE  9
          ISAT   4  252
          MESSAGE 10
  
```

Por último seleccionaremos la tabla de Respuestas e insertaremos la entrada:



```

- PAJARO EQ      5   3 ;¿Acaba de volar el pájaro?
  SAME      12  38 ;¿Está en la localidad del jugador?
  AT        5     ;¿o en el pabellón de música?
  MESSAGE   11     ;Ha aterrizado en la hierba.

```

```

- PAJARO EQ      5   3 ;¿Acaba de volar el pájaro?
  SAME      12  38 ;¿Está en la localidad del jugador?
  AT        8     ;¿o está en la rama?
  MESSAGE   12     ;Aterrizó en la rama.

```

Ahora, si el pájaro tiene el billete en su pico debemos decirselo al jugador.

```

- PAJARO EQ      5   3 ;¿Acaba de volar el pájaro?
  SAME      12  38 ;¿Está en la localidad del jugador?
  ISAT      4  252 ;¿El billete no ha sido creado?
  MESSAGE   10     ;Tiene el billete en su pico.

```

#### ISAT

Es una condición seguida de un objeto y el número de una localidad y es positiva si el objeto está en la localidad especificada.

Finalmente, si el emparedado está en la misma localidad que el pájaro, el pájaro dejará el billete para coger el emparedado. Esta entrada no tiene nada que ver con la bandera 5, así que será buscada cada vez que PAW vaya al Proceso 2. Por lo tanto, si el jugador deja caer el emparedado después de que el pájaro haya llegado, la secuencia correcta aún se llevará a cabo.

```

- PAJARO COPYOF  2   11 ;Emparedado
  SAME      11  12 ;¿En la misma localidad que el pájaro?
  ISAT      4  252 ;¿Lleva el billete en el pico?
  COPYFO    12   4 ;Empieza a caer el billete.
  SAME      12  38 ;¿Está el jugador también por ahí?
  MESSAGE    6     ;Diselo

```

#### COPYFO

Es una acción que copia el contenido de una bandera específica a la localidad actual del objeto específico. También hay acciones COPYFF y COPYOO que son fáciles de comprender.

Este bucle (loop) se muestra en el diagrama 4 en forma de una carta de flujo que se puede seguir desde el cuadrado marcado con "comienzo". El bucle es bastante más complejo que lo que el diagrama te pueda parecer y de hecho se da una versión más completa en la guía técnica, pero de momento el diagrama 4 nos servirá.

Te aconsejamos fervientemente que vuelvas a leer los párrafos anteriores y estudies el diagrama, hasta que entiendas como PAW trabaja sobre una sentencia lógica antes de funcionar.

Vamos a considerar la segunda entrada de Respuestas:

Hacer I INVEN

como habrás deducido hábilmente, ésta entrada toma en cuenta otra forma de la cual el jugador puede pedir una lista de objetos.

Veamos ahora la entrada que pone QUIT\_ (para hacer esto basta con apretar cualquier tecla excepto BREAK, ESPACIO o N):

```

FIN _ QUIT
      TURNS
      END

```

Bien, FIN es un verbo en el vocabulario, así que como la mínima frase que PAW considerará válida es un verbo, si FIN se teclea solo entonces ya el PARSER generará una SL de "FIN \_".

En su búsqueda a través de la tabla de Respuestas, PAW encontrará la entrada que hemos visto más arriba y empezará a ejecutar los condacc.

QUIT es una condición (la parte condicionante de la palabra condacc). Una condición solo decide si PAW continúa para llevar a cabo el siguiente condacto de la lista, es decir, no toma de por sí ninguna acción, sino una decisión.

QUIT lo que hace es determinar si el siguiente condacto puede ser ejecutado, preguntándole al jugador "¿estás seguro?". Si él contesta NO, entonces QUIT avisa a PAW que ha hecho "algo" y éste irá en busca de otra SL (dejando en paz QUIT).

Esta es una forma un poco diferente de tratar las condiciones en PAW, pero QUIT es una condición bastante especial, como verás en el futuro. Si el jugador teclea "SI" entonces QUIT no hace nada, y deja que PAW busque el siguiente condacc de la secuencia, que entonces será TURNS.

**TURNS** es una acción que imprime "has hecho x movidas o has dado x órdenes" en la pantalla. Donde x es el número de frases que el jugador ha tecleado desde el principio del juego. Aparte de este hecho, como tú no le has dicho a PAW que deje de buscar condaccs, entonces éste continuará buscando END.

**END** es una acción también especial que imprime "¿apetece otro juego?, o ¿quieres jugar otra partida?" en la pantalla. Si el jugador teclea "SI" entonces END reiniciará el juego con todos los objetos en sus posiciones iniciales, etc. Si el jugador dice "NO" entonces generará un error "OK", lo cual te hará volver al menú editor de PAW.

Si el menú editor de PAW no está presente (por ejemplo, un juego ya terminado), entonces el computador hace un reset.

**MUY IMPORTANTE:** Debes tener siempre una acción END en alguna parte del juego, porque si no, no podrás volver a la sección editora tan fácilmente cuando estés probando el juego.

Tendrás que hacer uso de la tecla BREAK, la cuál solamente trabaja mientras que PAW esta procesando, y como no es fácil pillar a PAW haciéndolo, porque es muy rápido, te será muy difícil.

Las otras entradas que ya están presentes en la base de datos, se hacen cargo de varios comandos estandarizados que los jugadores normalmente necesitarán en la aventura.

Los condaccs usados en las otras entradas se discuten más abajo.

Tú te estarás preguntando porque éstas entradas están en una tabla y no forman ya de por sí parte de PAW, si son tan necesarias en cada juego.

## DESTROY

Es una acción que pone un objeto especificado en la localidad 252 (localidad no creada).

Ahora vamos a crear dos posibles movimientos para el pájaro.

Si el pájaro está en el pabellón de música y la bandera 5 ha llegado a 0, entonces hay que mover el pájaro, setear la bandera 5 a tres otra vez y decirle al jugador que el pájaro se ha ido, si está en la misma localidad. Y lo mismo si el pájaro está en la rama.

PAJARO	EQ	12	8 ;¿Está el pájaro en la rama?
	ZERO	5	;¿Es tiempo de volar?
	LET	12	5 ;Mover el pájaro al pabellón
	LET	5	3 ;Tres frases antes de moverse.
	AT	8	;¿Está el jugador también aquí?
	MESSAGE	14	;Dile que el pájaro ha volado.

PAJARO	EQ	12	5 ;El pájaro está en el pabellón
	ZERO	5	;¿Es tiempo de volar?
	LET	12	8 ;Mover el pájaro a la rama
	LET	5	3 ;Deja 3 frases antes de moverlo
	AT	5	;¿Está el jugador también aquí?
	MESSAGE	14	;Dile que el pájaro ha volado.

## EQ

Es una condición que debe ser seguida por un número de bandera y otro valor, y que será positiva si la bandera contiene ese valor. En este caso (EQ 12 8) está comprobando si el pájaro está en una localidad específica.

## LET

Es una acción que también es seguida por una bandera y un valor. Y pone ese valor en la bandera.

Ya hemos tenido en cuenta el vuelo del pájaro. Ahora vamos a poner sus llegadas, y si llega a una localidad en la que esté el jugador, hay que decirselo.



Primero hay que determinar si el pájaro va a salir pitando, esto pasará cuando la bandera 5 tenga valor 0 (empezará a contar desde 3). Entonces, si el billete está en la misma localidad que la del pájaro, será destruido (puesto que la localidad 252 indica que el pájaro lo tiene) y si el jugador está en la misma localidad que el pájaro, se le dirá que el pájaro se ha llevado el billete.

Tienes que darte cuenta de que el pájaro sigue con sus ciclos de movimientos aunque el jugador no lo vea. De hecho en el mundo de PAW un árbol se cae aunque no haya nadie para verlo. Así de real es.

PAJARO COPYOF	4	11	;Copia la localidad del objeto 4 (el billete) a la bandera 11.
SAME	11	12	;y mira si está en la misma localidad que el pájaro.
ZERO	5		;¿Va a volar el pájaro?
DESTROY	4		;El pájaro coge el billete.
SAME	12	38	;¿Está el pájaro en la misma localidad que el jugador?
MESSAGE	7		;Diselo al jugador.

Fíjate, no hay acción DONE porque queremos que PAW haga cada entrada, una detrás de otra, siempre que se cumplan los requisitos de las condiciones de ellas. La tabla anterior te muestra como se pueden mezclar condiciones y acciones para crear nuevas condiciones.

#### COPYOF

Es una acción que debe ser seguida por el número de un objeto y una bandera. Lo que hace es que copia la localidad actual del objeto que se especifique, en la bandera especificada. La usamos en esta situación para ver si el billete está en la misma localidad que el pájaro.

#### SAME

Es una condición que compara el contenido de dos banderas, y que es favorable o positiva si ambas son del mismo valor.

Pues bien mi querido saltamontes, la razón es que, aparte del hecho de que es mucho más fácil ponerlas en una tabla que en propio juego, **TU JUEGO A LO MEJOR NO LAS NECESITA**, y de esta forma las puedes borrar cuando quieras.

DESC es una acción. De hecho, es la que se usa en la entrada "M \_" de la tabla. Y esta acción hace que PAW abandone la búsqueda por la tabla de respuestas y reDESCRIBA la localidad actual del jugador. También tiene varios sinónimos: mirar, describir, etc., en el Vocabulario.

SAVE Y LOAD son dos acciones que permiten hacer un SAVE y un RELOAD del estado actual del juego en cinta. La posición actual del juego incluye también cualquier otra pieza de información que se necesite para restaurar exactamente el juego en el mismo estado en que estaba, ello incluye los valores de las banderas, las posiciones de los objetos, y mucha más información.

No debes confundir aquí los anglicismos SAVE y LOAD que hemos dejado en el vocabulario, con las acciones SAVE y LOAD. Nosotros hemos incluido también otros sinónimos, como GRABAR y CARGAR, en el Vocabulario, pero todos ellos usarán también las acciones SAVE y LOAD en la tabla de Respuestas.

Es de notar que ambos SAVE y LOAD ya de por sí hacen una acción DESC cuando han terminado. Lo cual significa que *cualquier condacc que siga, será ignorado igualmente en el INPUT del jugador.*

RAMSAVE Y RAMLOAD son dos acciones similares a SAVE y LOAD, excepto que ellos usan un "buffer" (área de memoria libre) para guardar la posición del juego. Esto significa que no tienes que andar toqueteando cintas.

Solamente una posición puede ser guardada, y como la guarda en la memoria, suponemos que sabes que si apagas el ordenador desaparecerá. Aunque de todos modos, ésto debes hacérselo ver claramente al jugador en las instrucciones.

También el área del buffer se pierde cuando vuelves al editor, porque tú puedes desear cambiar algún diseño en el juego entre dos textos.



El número ligado a RAMLOAD es un parámetro, y le dice al condacc cuantas de las banderas debe restaurar de un RAMSAVE previo. Esto permite que la puntuación y otras cosas se mantengan, aunque el jugador "haga trampas" en una parte muy difícil del juego usando constantemente RAMSAVE y RAMLOAD.

Ambas acciones son seguidas automáticamente por una acción DESC, puesto que a diferencia de SAVE y LOAD, ellas, por sí solas continuarán hasta el siguiente condacc.

Si a PAW se le acaban los condaccs en una lista sin haberle aclarado que ya ha hecho (done) algo, sencillamente "caerá" hasta el final, y al darse cuenta de esto, continuará en su búsqueda de otro SL.

También dijimos que QUIT era una condición bastante especial. Para comenzar, es la única condición que busca información del jugador, y por otra parte, también le informa a PAW de que algo ha sido hecho si el jugador teclea "NO" (es decir, que no quiere abandonar el juego), lo cual hace que PAW comience a buscar otro nuevo SL.

**MUY IMPORTANTE :** una condición normal, si "falla", sencillamente hará que PAW continúe buscando en la tabla de Respuestas otra condición que haga juego con la SL.

Los otros conductos que están usados se considerarán ahora en relación con las entradas de las cuales son parte. Para simplificar nuestra explicación, debemos considerar la posición de un objeto en el juego.

Puede estar en cuatro lugares:

**AQUI:** es la localidad actual del jugador (es el valor que se guardaba en la bandera 38, si recuerdas bien). Tiene un valor de 255.

**LLEVADO PERO NO PUESTO:** Localidad 254, que es una localidad imaginaria donde se ponen todos los objetos que el jugador lleve.

**PUESTOS ENCIMA:** Localidad 253, que es una localidad imaginaria donde todos los objetos que el jugador lleve puestos encima son guardados.

Es decir, las sub-rutinas pueden ser anidadas hasta un valor de 10. Si se intenta ir más lejos saldrá la frase "Limit Reached", o sea Límite Alcanzado.

En este momento no vamos a usar un sub-proceso muy complicado, solamente uno simple, y lo haremos para tener en cuenta todas las chorraditas del pájaro.

Usando [B] se comienza un nuevo sub-proceso y PAW automáticamente te localizará el siguiente número libre. De momento deberíamos estar comenzando el proceso 3. Como solamente hay unas pocas entradas en la tabla, usaremos el mismo par de palabras ("PAJARO"), aún cuando hay que tener en cuenta que al escribir tu juego debes de usar otras palabras que te recuerden lo que hace cada entrada.

La bandera 11, será una bandera de "trabajo", porque contiene un valor que se usa solamente como comparación.

La bandera 12, llevará el número de la localidad actual del pájaro.

La bandera 5, es una bandera especial, porque si tiene un valor diferente de 0, disminuirá 1 cada vez que PAW ejecute una acción, es decir, cada vez que PAW revise la tabla de Proceso 2. Es lo que se llama una bandera auto-decreciente.

En este caso usaremos la bandera 5 para contar el número de turnos que han pasado en el juego. Un turno del juego es toda una vuelta del circuito grande en el diagrama 4, y de momento, esto pasa cada vez que el jugador tecléa una frase.

El pajarito cambiará de localidad cada tres frases, lo que creará una especie de apariencia de acción independiente de lo que tecléa el jugador.

Ahora, vamos a insertar las siguientes entradas (sin los comentarios, como hicimos antes). En cada entrada tenemos una explicación de su propósito y también explicaremos cualquier nuevo conducto que se use:

### Mensaje 13

El pajarito ve al perro y sale pitando.

### Mensaje 14

El pajarito se va.

Luego pondremos los mensajes que vaya a usar el perro, pero ahora debemos seleccionar la tabla de Procesos y prepararnos para otras de las virguerías de PAW.

### SUB-PROCESOS

En una aventura de verdad que contenga varios PSI y un montón de acciones, las tablas de Procesos 1 y 2 pronto acabarán llenas, y se hace cada vez más difícil trabajar de esta forma. Entonces viene el momento de dejar que las otras tablas de Procesos comiencen a actuar. Estas pueden ser llamadas desde las tablas de Procesos 1 y 2 o de la tabla de Respuestas, y usadas como una extensión de la tabla desde la cual fueron llamadas.

El llamar a otro proceso hace que PAW guarde en memoria dónde estaba en el momento de ser llamado, y continuará actuando en la misma forma en que lo estaba haciendo. Por ejemplo, si se llamó desde una tabla de Respuestas, PAW tratará de hacer coincidir la Sentencia Lógica actual con cualquier otra entrada. Pero si es llamado desde una tabla de Procesos, sea la 1 o la 2, PAW sencillamente ejecutará cada entrada. (ESTO ES MUY IMPORTANTE).

Por otra parte, cuando algo es DONE en el Proceso llamado, PAW volverá a la tabla original y continuará trabajando. Por lo tanto se pueden hacer acciones bastantes complejas mediante el uso de los sub-procesos. Los usuarios que sepan algo de programación reconocerán fácilmente que son sencillamente sub-rutinas.

Cuando PAW está en un sub-proceso, es posible que sea llamado para actuar en otro sub-proceso (¿un sub-sub-proceso?), y así se puede continuar con sub-sub-sub hasta un valor de 10.

NO PRESENTES: ¡En cualquier otro sitio!. Esto también incluirá la localidad 252, que es una localidad imaginaria donde todos los objetos que aún no existen son guardados.

### Resumiendo:

Aquí	=	Localidad 255
Llevados	=	Localidad 254
Puestos encima	=	Localidad 253
No presentes	=	Localidad 252 = No creado todavía.

Vamos a ver otras dos entradas en la tabla de Respuestas:

COGE TODO DOALL 255 (Parámetros: Localidad donde estás: Aquí)

COGER \_ AUTOG  
DONE

éstas dos entradas son las que permiten al jugador coger cualquier objeto. Coger un objeto significa cambiar su localidad desde AQUÍ (255) a LLEVADO (254).

Ignoremos por ahora la entrada "coge todo" y vamos a estudiar la entrada "COGER \_".

Como dijimos antes, esa línea baja significa "cualquier palabra", así que no importa cual nombre teclee el jugador en una frase que contenga el COGER. La entrada COGER \_ siempre casará (esto es lo que se llama "cargando" o "preparando" una entrada).

Veamos la frase "coger la manzana": la será ignorada por PAW porque no está en el Vocabulario, pero la SL será "coger manzana", y esto "cargará" la entrada "GET \_" cuando PAW encuentre este conducto.

AUTOG es una acción que AUTOMáticamente cogerá cualquier objeto que haya sido especificado por el nombre.

Es donde la tabla de Palabras-Relacionadas con Objetos viene a funcionar. AUTOG mira a través de esa tabla buscando una entrada que sea similar al nombre de la SL, cuando la encuentre (manzana en nuestro ejemplo) entonces sabe el número del objeto al cual se refiere (que en el caso de la manzana es 3).



Se asegura entonces de que la localidad actual de ese objeto sea AQUI que es (255) y la pasa a LLEVADA que es (254) y entonces imprime el mensaje "Ahora tengo el objeto \_." donde la línea baja es reemplazada por la descripción del objeto actual, por ejemplo el que acaba de marcar el AUTOG.

Si no encuentra ninguna entrada, entonces hay cinco posibilidades.

1/ El jugador ha intentado coger un objeto que ya lleva o tiene puesto. En cuyo caso el mensaje "Ya tengo el \_" se imprime.

2/ El jugador ha intentado coger un objeto que está en la localidad AQUI NO. En cuyo caso se imprimirá el mensaje "Aquí no hay uno de esos".

3/ El jugador ha intentado coger algo que no es un objeto, pero que tiene una palabra en la tabla de Vocabulario (por ejemplo: puerta en nuestro juego de demostración). Esto producirá el mensaje "No puedo hacer eso".

4/ El jugador ha usado una palabra que no está en el Vocabulario. Eso hace que el PARSER cree una SL de "GET \_" lo cual "carga" nuestra entrada de GET \_ de todos modos. AUTOG asume que se trata de un nombre que describe un objeto (que puede o no puede existir), y entonces el mensaje "No hay uno de esos aquí" se imprime.

5/ El jugador no puede llevar más objetos, o el peso de éste objeto sobrepasa el límite permitido para el jugador. En cuyo caso también aparecerá un mensaje que le informa de ello.

Si el AUTOG funciona, entonces PAW mira al siguiente condacc.

DONE solamente le dice a PAW que la entrada ha terminado y que debe de ir y buscar otra SL.

Ahora miraremos a la entrada GET ALL. Como habrás supuesto, ésta entrada lo que hace es intentar coger todos los objetos que estén en la localidad donde estás tú.

En el vocabulario insertemos también los nombres pajarito o pájaro y perro. Supondremos que el pájaro es 22 y el perro es 21, perro = 21, pájaro = 22, porque sus valores como palabras serán usados para poner las entradas en el orden correcto en las tablas de Procesos.

Después, insertemos los siguientes mensajes. Podemos usar tinta verde (en MODO EXTENDIDO, CAPS SHIFT + 4) para los mensajes, no olvidando volver a poner tinta blanca al final (MODO EXTENDIDO, CAPS SHIFT + 7).

MODO EXTENDIDO + CAPS SHIFT + NUMERO = EL COLOR DE LA TINTA

Mensaje 6

El pajarito deja caer el billete para coger el emparedado.

Mensaje 7

El pajarito se lleva velozmente el billete.

Mensaje 8

El pajarito me ignora.

Mensaje 9

Un pajarito anda rondando por aquí.

Mensaje 10

El pajarito tiene un billete en su pico.

Mensaje 11

Un pequeño pajarito se posa en la hierba.

Mensaje 12

El pajarito está ahora en la rama del árbol.



## EL PAJARITO

El juego de práctica que hemos hecho es bastante fácil de resolver (bastante tontito, la verdad). Así que le vamos a añadir un poco de complejidad creando dos caracteres que vayan vagando por nuestro pequeño mundo. Estos caracteres son llamados PSI, o Caracteres Seudo Inteligentes, porque es obvio que aunque no pueden pensar, deben de parecerle al jugador como si lo hiciesen.

Un PSI consiste principalmente en una colección de mensajes, banderas y entradas en tablas de procesos. Verás que con unas pocas entradas simples se pueden crear efectos sorprendentemente real.

La creación de un complejo PSI puede tomar mucho más tiempo, pero en general sigue los mismos principios que los que vamos a hacer con los nuestros: un pajarito y un perrito.

El pájaro se pone para complicar el escenario de la siguiente manera: Cogera el billete al principio del juego (normalmente se le daría una localidad no usada como contenedor de los objetos que lleve el pájaro, pero nosotros usaremos la localidad 252, puesto que tenemos solamente un PSI que puede tener un objeto).

De modo que el jugador debe convencer al pajarito para que deje caer el billete. El intentar quitárselo o cogerlo dará el mensaje "No puedo hacer eso", y que el pajarito se vaya volando hacia otra parte.

El pajarito también dará la paliza volando entre el pabellón de música y una ramita del árbol, y esto lo hará a intervalos regulares. La única manera de convencer al pajarito para que deje caer el billete de autobús es dejar caer el emparedado en la misma localidad.

Lo primero que hay que hacer es ir a la tabla de Objetos Inicialmente En [la tabla I] y cambiar el objeto 4 a no creado, es decir, darle al objeto 4, que es el billete, el valor de 252.

Esto hace que el objeto no exista como un objeto, porque lo lleva el pajarito en el pico.

Explicaremos el mecanismo:

Si el jugador teclea la frase COGE TODO, el PARSER creará una sentencia lógica (SL) de "COGE TODO" y entonces hará pareja con la entrada presente en esta tabla y PAW pasará a ejecutar la acción DOALL.

DOALL es una acción que debe ser seguida con un parámetro. El parámetro nos da el número de la localidad que usaremos.

Lo que hace DOALL es buscar la lista de localidades en la cuál se encuentra cada objeto, intentando encontrar entradas que sean iguales para el parámetro que se ha dado (que en este caso es 255, una localidad especial que significa que se use la localidad en la cual el jugador esté presente).

Cuando encuentra uno similar, o el mismo, entonces se va a la tabla objetos relacionados con palabras para encontrar la palabra del Vocabulario que describe el número del objeto.

Esta palabra se pone en la actual SL (reemplazando el nombre todo) y una bandera se resetea para indicar que DOALL está activo.

Luego PAW sigue mirando el resto de la tabla de Respuestas buscando una entrada que haga juego con la SL recientemente modificada.

Esta entrada será COGE \_ (de la cuál hemos hablado antes), que cogerá el objeto que PAW haya buscado en la tabla de Objetos.

Cuando haya terminado de hacer lo anterior, PAW se dará cuenta de que todavía permanece activo el DOALL, o sea, la bandera está seteada y vuelve otra vez a la entrada COGE TODO.

En realidad lo que hace es saltar directamente a la acción DOALL y buscar otro objeto para generar una nueva SL, y así sucesivamente para todos los objetos que estén en la localidad especificada.

Cuando se acaban los objetos, entonces la bandera se resetea para demostrarle a PAW que no está activo el bucle y se le dice a PAW que busque otra SL.

Esto podría parecer un modo bastante complejo de ejecutar esa acción, pero si examinamos a las entradas similares que tratan de PONERSE, QUITARSE y DEJAR, veremos que utilizan el mismo mecanismo.

AUTOD, AUTOW y AUTOR trabajan de una manera muy similar a AUTOG, mientras que DOALL sencillamente se limita a cambiar la localidad actual para ejecutar las acciones.

Por ejemplo, usa 254 (LLEVADAS) como el parámetro en la acción DROP y también en la acción WEAR (es decir, DOALL busca todos los objetos que lleves cuando intentas dejarlos caer o ponértelos encima); y utiliza la localidad 253 (LLEVADO PUESTO ENCIMA) cuando se trata de quitarse todo.

Si lo anterior te parece un poco complicado, no te preocupes de momento, porque DOALL es uno de los conductos más complejos de PAW, y poco a poco lo irás entendiendo.

Sería interesante que en este momento hagas un test de la aventura, y trates todos los comandos de "dejar todo", "coger todo", etc., para que el mecanismo se te haga más claro.

## MENSAJES

Antes de que continuemos con la tabla de RESPUESTAS vamos a meter algunas entradas en otra tabla que necesitaremos. Así es que, desde el menú principal, teclea [ M ] para mensajes.

La tabla de mensajes es poco complicada. El submenú es muy parecido a los de localidades y descripciones de objetos. El objetivo de los mensajes es contener todo el texto que se necesita para responder a las entradas del jugador, es decir, todo lo que está sucediendo durante el juego, (excepto por los mensajes que ya PAW pone por su cuenta, como "Yo no puedo hacer eso", etc.). Si tecleas la opción [ P ] verás que ya hay una entrada presente.

Ahora vamos a poner todos los comandos para cuando el jugador intente examinar las cosas en el juego (una de las órdenes más frecuentes y más importantes de toda la aventura).

Tecleemos [ I COGER TODO 0 ENTER ]:

PREP FUERA NOUN2 BOLSA DOALL 1

Todo esto se puede obviar usando los verbos sacar y meter (pero es importante que lo veamos de esta forma por si hay que hacer alguna entrada complicada como esta).

También debemos poner una entrada que le permita al jugador EXAMINAR DENTRO DE LA BOLSA o MIRAR DENTRO DE LA BOLSA, y será:

MIRAR	BOLSA	PREP	DENTRO
		MESSAGE	5
		LISTAT	1
		DONE	

## LISTAT

Debe ser seguido por un número de localidad y listará todos los objetos presentes en esa localidad. Si no hay objeto presente dirá "ninguno"; así que si tecleamos con la bolsa vacía, la respuesta será "En la bolsa hay: nada", lo cual es correcto, en oposición con LISTOBJ que veíamos que no ponía nada porque tiene un uso mucho más frecuente.

Hagamos un test de la aventura para ver que de verdad podemos poner y sacar todo de la bolsa.



Combinado con la entrada previa tiene el efecto de hacer que PAW pare de buscar en las tablas de conducto a menos que la Sentencia Lógica fuera PONER \_ EN o DENTRO DE LA BOLSA donde la raya baja sería cualquier objeto.

#### AUTOP

Debe ser seguida por el número de una localidad. Nosotros hemos puesto la localidad 1 aparte con un propósito especial desde el comienzo de este ejemplo. Esto es, la hemos usado como si estuviera dentro de la bolsa. Por lo tanto AUTOP, de la misma forma que AUTOD, busca en la tabla de Objetos-Palabra por un nombre que haga pareja con el primer nombre de la Sentencia Lógica. Cuando ha encontrado uno, lo pone en la localidad que le hemos dado, respondiendo "He puesto el o la \_ en o dentro de la bolsa".

La entrada DEJA TODO ya existente, también se hace cargo de PONERLO TODO EN LA BOLSA, porque no especifica que EN LA BOLSA sea parte de la Sentencia Lógica, por lo tanto se disparará en ambas ocasiones, y en ambos casos 254 es la localidad desde la cual vendrán los objetos.

Ahora para COGER un objeto FUERA DE LA BOLSA (o sacar un objeto de la bolsa), debemos de teclear una entrada similar, que haga nula la entrada ya presente de COGER \_ . Así que tecleemos [I COGER \_ 0 ENTER], con ello nos aseguramos que será puesta antes.

PREP FUERA NOUN2 BOLSA PRESENT 1 AUTOT 1 DONE

#### AUTOT

Debe ser seguido por un número de localidad, que debe ser de la cuál venga el objeto que se va a sacar. (En éste caso la 1).

Por otra parte, para hacer la versión SACAR TODO necesitamos también otra entrada. De momento COGER TODO causa un DOALL 255, lo cual es la posición actual del jugador. Para que saquemos todo de la bolsa necesitamos generar un contador de los objetos que estén dentro de ella (localidad 1), así que hay que insertar una entrada nueva de coger todo, que salte por encima de la existente.

El que un objeto sea examinable, solamente requiere que el escritor haya dejado un mensaje que dé más información sobre ése objeto, por ejemplo, en el caso de la manzana diríamos "La manzana está madurita y muy sabrosa".

Cambia el texto del mensaje 0 (A 0 ENTER) e inserta los siguientes mensajes para poder examinar todas las otras cosas del juego.

#### Mensaje 1

Es un emparedado delicioso de jamón y queso.

#### Mensaje 2

El billete tiene un número y el nombre de la compañía de transportes para aventureros impreso.

#### Mensaje 3

El banco está anclado firmemente a una base de concreto.

En nuestro juego de práctica solamente usamos cuatro objetos, pero en los juegos más complicados hay que poner muchos detalles para muchas cosas, aunque no sean muy importantes en el juego, porque ésto le da un toque de realismo que hace que el jugador se sienta atraído o envuelto por la atmósfera.

Así que volvamos a la tabla de RESPUESTAS (R) y a trabajar.

Vamos a comenzar con la manzana. La frase que el jugador tecleará será "EXAMINAR LA MANZANA" (o EXAMINA MANZANA si es bastante vago), y esto producirá una SENTENCIA LOGICA de "EXAMINAR MANZANA". Así que necesitaremos poner una entrada con estas dos palabras.

Tecléa (I EXAMINAR MANZANA ENTER), PAW ignorará las letras que sobren (que sean más de cinco), e imprimirá la entrada en la parte de arriba de una pantalla limpia y esperará a que tu teclées la lista de conductos para esa entrada.



Pero nosotros sólo vamos a permitir que el jugador examine la manzana si está AQUI, LLEVADA o PUESTA ENCIMA (por ejemplo, en un bolsillo), porque la mayoría de la gente no tiene la habilidad de mirar a través de las paredes para saber si la manzana está en otra localidad.

Estos tres parámetros AQUI, LLEVADO, o PUESTO ENCIMA todos se conocen como PRESENTE, y por ello basta con poner la condición PRESENT seguida del número del objeto que estemos considerando (por ejemplo, la manzana, es el objeto 3).

Si el objeto está presente, entonces podemos hacer aparecer nuestro mensaje 0, que describe el objeto usando la acción MESSAGE, que debe ser seguida del número del mensaje que quieras que aparezca. Finalmente lo acabaremos todo con la acción DONE para decirle a PAW que ya hemos completado lo que queríamos hacer.

Entonces teclea (PRESENT 3 MENSAJE 0 DONE ENTER) y PAW concestará que el mensaje está "insertado".

Ahora vamos otra vez al submenú con cualquier tecla, y tecleemos (P EXAMINAR) para examinar nuestra nueva creada entrada.

Por cierto, aquí conviene decir que si quieres ver la tabla desde el principio basta con teclear P. Si tecleas P seguida por un VERBO, verás todas las entradas que pertenecen a ese verbo, y si tecleas P seguida por un VERBO y un NOMBRE verás todas las entradas del verbo y el nombre. Nuestra entrada debe de parecerse a:

```
MIRAR-EXAMINAR  MANZANA  PRESENT 3
                  MESSAGE 0
                  DONE
```

Aquí tenemos un ejemplo clásico de la tabla de Respuestas, porque si la condición PRESENT 3 falla, es decir el objeto no está presente, entonces PAW continuará buscando por otra entrada diferente para hacer coincidir su SENTENCIA LOGICA.

En este caso encontrará la siguiente entrada, que es MIRAR \_, y ésta entrada producirá una descripción de la localidad en que estamos actualmente (por medio de la acción DESC), después de que la acción PLUS haya añadido 128 a la bandera 29.

Podemos usar exactamente el mismo sistema que hicimos con coger y dejar todo. Poner es un sinónimo de dejar, por lo que la Sentencia Lógica que estamos buscando debe ser PONER \_ (es decir, cuando el jugador está tratando de poner o dejar algo), pero si el jugador incluye EN LA BOLSA como parte de su frase, haremos que PAW ponga el objeto DENTRO DE LA BOLSA.

Esto significa que hay que pasar por alto la entrada PUT \_, que ya está presente, si las palabras extras incluidas en la Sentencia Lógica especifican que el objeto es la bolsa.

Para insertar esta entrada antes de una que ya esté presente, hay que hacer lo siguiente: normalmente PAW insertará otra entrada con el mismo valor de palabra después de cualquier entrada que ya esté presente. Es posible forzarlo a que lo haga antes especificando un número después de la inserción. (Ojo: muy importante).

Intenta [I PUT DEJA \_ 0 ENTER], esto instruye a PAW para que ponga la entrada antes de la entrada número 1, puesto que le hemos dado un valor 0 antes de ENTER, y como la entrada número 1 es el PUT \_ que ya existe, PAW la pondrá antes de ella.

Ahora pongamos los contactos que necesitamos y que son:

```
PREP  EN NOUN2 BOLSA PRESENT 1 AUTOP 1 DONE.
```

Esto es un ejemplo de cómo nos aseguramos de que ciertas partes de la frase sean lo que nosotros queremos.

## PREP

Es una condición que debe ser seguida por una preposición del vocabulario. Las preposiciones son palabras que se usan antes de un Nombre para mostrar su relación con otras palabras de la frase. En este caso la condición será favorable o acertará si el jugador ha usado en (o dentro) como parte de la frase.

## NOUN2

Es una condición que debe ser seguida por un nombre del vocabulario. Será positiva si el jugador ha usado bolsa en la frase como segundo nombre.

## ANYKEY

Imprime "Pulse tecla para continuar" en la parte inferior de la pantalla, y espera a que pulses una tecla antes de permitirle a PAW continuar.

## GOTO

Debe ser seguido por un número de localidad y mueve al jugador a esa localidad (efectivamente, pone la bandera 38, la bandera que marca la localidad actual del jugador, al valor dado). Como no hace nada más, debe ir seguida por un DESC para que PAW describa la nueva localidad.

Esta entrada entonces lo que hace es que la pantalla del título aparezca, luego espera por el toque de tecla para pasar al juego en la localidad correcta.

Ahora es el momento de ir a la tabla de conexiones y quitar la entrada que pusimos desde la localidad 0 porque ya no se necesita.

Hagamos un test del juego para ver las dos entradas en acción. Si tecleásemos de una sola vez toda la frase:

OESTE, OESTE, ARRIBA. COGER EL BILLETE. ABAJO, ESTE Y ESTE.

tendríamos el juego terminado, con su mensaje de si quieres volver a jugar otra vez. Si no es así, vuelve a chequear las entradas en las tablas de Procesos 1 y 2.

## OBJETOS DENTRO DE ALGO

Vamos a ver la habilidad de la bolsa para contener objetos. Ello requiere algunas entradas en la tabla de Respuestas, porque vamos a permitir al jugador que MIRE DENTRO DE LA BOLSA, así que necesitamos un nuevo mensaje: "En la bolsa hay:". Por lo tanto, selecciona la opción de MENSAJES e INSERTALO (debe ser Mensaje 5).

Otra vez a la tabla de RESPUESTAS.

Vamos a dotar al jugador de la opción de decir "Poner todo dentro de la bolsa", además de "poner (objeto) en la bolsa".

Esto forzará a PAW a dibujar cualquier gráfico que exista en esa localidad y al mismo tiempo hacer una descripción. Sobre esto entraremos con más detalle cuando hablemos de los gráficos.

Si asumimos que la manzana está realmente presente, entonces PAW continuará con el siguiente condacc y desplegará nuestra descripción de la manzana (MENSAJE 0), luego, la acción DONE le dice a PAW que vaya y busque otra Sentencia Lógica, porque ya ha hecho algo (esto previene que continúe y pase lo que pasó antes, es decir que coja la siguiente entrada que es LOOK \_).

Si hay una entrada que es incorrecta, se puede corregir tecleando por ejemplo, A MIRAR MANZANA. Sin embargo, en las tablas de Procesos puede haber más de una entrada que tenga los mismos valores de palabras, y entonces todas ellas se verán presentadas una a una para que se pueda corregir.

Si no quieres corregir una en particular, sencillamente teclea ENTER y déjala tal como está.

Para borrar una entrada completamente de una tabla, lo que hay que hacer es quitar todos sus condaccs.

Por ejemplo, corrige la entrada con A, pulsa la tecla EDIT dos veces para limpiar el buffer y pulsa ENTER. Con eso la entrada quedará completamente borrada.

Pero de momento, si el jugador trata de EXAMINAR cualquier otra cosa excepto la manzana (o cuando trata de EXAMINAR LA MANZANA y ella no está presente) entonces, como vimos antes, se le recompensará con una descripción de su localidad actual.

Vamos a insertar las entradas para poder examinar el emparedado, el billete de autobús y el banco.

Las entradas las listaremos como las verías si usaras la opción P después de que fueran tecleadas, y pondremos al lado algunos comentarios solamente para referencia. Pero no te olvides de que hay que meterlas como hiciste para introducir EXAMINAR MANZANA anteriormente.



MIRAR EMPAREDADO PRESENT 2 ;Que el emparedado esté aquí  
MESSAGE 1 ;Describelo  
DONE

MIRAR BILLETE PRESENT 4 ;El billete está aquí  
MESSAGE 2  
DONE

MIRAR BANCO AT 4 ;AT porque el banco no es un objeto  
MESSAGE 3 ;y entonces se chequea la localidad  
DONE

AT es una condición que debe de ser seguida con un número de localidad y que será válida (es decir, dejará que PAW continúe al siguiente conducto), si el jugador está en esa misma localidad.

Esto lo tuvimos que usar con el banco porque no era un objeto, pero, al ser parte de la descripción, al volver a la localidad 4 siempre estará ahí. Entonces lo que comprobamos es que el jugador esté en la localidad 4.

Usa ahora la opción T para comprobar tu aventura y ver que lo anterior funcione.

#### LAS TABLAS DE PROCESOS ( P1 : P2 : ETC. )

Vamos a estudiar ahora la opción más potente que tiene el menú principal: las tablas de Procesos.

Dijimos antes que la tabla de Respuesta era una opción especial de las tablas de Procesos, y de hecho lo es. Si tú seleccionas la opción P del menú verás un submenú muy similar al de la tabla de Respuestas, excepto que tiene dos opciones extra.

Fíjate que el título dice "Procesos 2", esto es porque hay más de una tabla de procesos en PAW, de hecho puede haber 254 tablas de procesos como veremos.

#### ZERO

Es la primera condición que hemos encontrado hasta ahora que chequea el estado de una bandera. Zero 0 será positiva si la bandera cero contiene 0, lo cual significa que hay luz.

#### ABSENT

Chequea que el objeto 0 no esté presente (es el opuesto a la condición PRESENT, todas las condiciones tienen su opuesto. Por ejemplo, AT tiene su opuesto en NOTAT, etc). Como la siguiente \* \_ va a hacer un listado de los objetos, si el objeto 0 (la fuente de luz) está presente, no queremos que la primera entrada de \* \_ sea positiva también (la cual tiene en cuenta una hipotética situación, en la cual haya luz y el objeto 0 también esté presente, y entonces se haría una lista de los objetos dos veces).

#### LISTOBJ

Hará una lista de todos los objetos que estén presentes en la localidad actual. Si ninguno está presente no hace nada. Sería un poco tonto decir "Puedo ver nada".

Nota Importante: Repasa bien lo anterior, puesto que es una característica de PAW que necesitarás usar con frecuencia en tus juegos.

Ahora vamos a poner una forma más adecuada de pasar de la pantalla de introducción al principio del juego en la parada de autobús. Lo haremos de la siguiente forma:

```
* * AT 0
      ANYKEY
      GOTO 2
      DESC
```

Inserta esta entrada en Procesos 1 (asegúrate de que todavía está seleccionado) usando [I \* \* ENTER]. Esto nos introduce dos nuevos conductos.



## NEWLINE

Imprime espacios hasta el final de la línea actual. Esto permite que el mismo color de Papel continúe hasta el final de la línea sin tener que teclear todos los espacios. Su principal objetivo aquí es asegurar de que cualquier texto que se imprima lo sea en la siguiente línea, porque PAW no pone automáticamente una línea en blanco después de la descripción de las localidades. La guía técnica nos mostrará como usar este efecto para modificar una descripción de localidad, para reflejar los cambios que haya.

De ahora en adelante las dos entradas deben de ser consideradas como una pareja, su propósito final es hacer una lista de los objetos que estén en la localidad actual. Veamos como:

PAW usa la bandera 0 para determinar si hay luz para que el jugador pueda ver (esto no ha sido utilizado de momento en nuestra demostración), si no, hay luz, la bandera tendrá un valor diferente de 0 y PAW contestará "está muy oscuro para ver" en vez de hacer una descripción de la localidad. En este caso los objetos que están presentes no deben de aparecer en lista puesto que se supone que el jugador no los puede ver.

El objeto 0 siempre es tomado por PAW como el objeto que provee luz (objeto 0 = fuente de luz), esto es por lo que en nuestro juego de demostración, el objeto 0 es una antorcha encendida. Si este objeto está presente cuando en el juego se supone que está oscuro (o sea, que la bandera 0 no sea 0) entonces, la sola presencia del objeto 0 tendrá preferencia sobre el hecho de que no haya luz, y los objetos deben ser descritos.

Estos dos últimos párrafos son muy importantes y debes releerlos hasta estar seguro de que los entiendes.

Así pues, vemos que las dos entradas nos dan un ejemplo de cómo se usa PAW para crear una situación OR. Por ejemplo, haz una lista de los objetos si hay luz o (OR) el objeto 0 está presente.

De momento hay dos tablas de Proceso en la base de datos y PAW, tal como hacía en el de Respuestas, buscará a través de ellas, pero las buscará no después de haber obtenido una sentencia lógica, sino de la siguiente forma:

Proceso 1: El proceso 1 es chequeado inmediatamente después de que PAW ha descrito una localidad. Esto permite que la información se imprima solamente cuando el jugador llegue a esa localidad o cuando haya obtenido una redesccripción.

Proceso 2: El proceso 2 es revisado justo antes de pedir una nueva sentencia lógica al PARSE. Es usado como si fuera el "turno" de PAW para jugar.

Estudia detenidamente las formas en que PAW busca en la tabla de Proceso 1 y en la tabla de Proceso 2, ES MUY IMPORTANTE.

La diferencia principal es que en estas tablas, PAW no intenta buscar una entrada similar a una sentencia lógica, sino que EJECUTA CADA ENTRADA.

De momento, mientras jugábamos en nuestro juego de demostración, hemos tenido que terminar siempre tecleando FIN. Ahora en la "Historia Original" lo que teníamos que hacer era ayudar al pasajero a encontrar el billete de autobús antes de que éste llegase. Es obvio que podemos poner una entrada en la tabla de Respuestas para que si el jugador teclea "coger billete" (y este estuviese presente), se termine el juego.

```
COGER      BILLETE  PRESENT  4 ; El billete está ahí
                                TURNS
                                END
```

pero pensamos que sería mucho más conveniente terminar el juego cuando el jugador vuelva a la parada del autobús.

Para hacer eso, primero necesitamos un mensaje que describa la llegada del autobús. Así que vamos al menú principal, seleccionemos la opción mensaje y pongamos el siguiente mensaje:

#### MENSAJE 4

El autobús llega. Le doy el billete al conductor, quien sonríe y dice: "siento haber llegado tarde, supongo que no ha tenido que esperar mucho".

Volvamos ahora a la tabla de Procesos. Aún cuando las palabras no tienen ningún significado en esta tabla, puede ser útil poner las palabras adecuadas con lo que la entrada hace.

La entrada que determina el final del juego será llamada "AUTOBUS" (debemos comenzar con \_ porque PAW solamente permite el nombre Autobús en una posición que corresponda al nombre, o sea, primero el Verbo y luego el Nombre).

Así que tecléa [ I \_ AUTOBUS ENTER ].

Las condiciones para terminar el juego son: que el jugador se encuentre en la parada del autobús (localidad 2) y que lleve el billete del autobús (objeto 4).

La primera condición será AT 2, la segunda será CARRIED 4, así que la entrada final será:

```
AT 2 CARRIED 4 MESSAGE 4 TURNS END
```

Si se usa P, se verá que la entrada aparece de la siguiente forma:

```
-   AUTOBUS  AT      2
      CARRIED  4
      MESSAGE  4
      TURNS
      END
```

Como PAW buscará siempre en esta tabla antes de que haya un nuevo Input (una nueva Sentencia Lógica) por parte del jugador, cada vez que se den las condiciones descritas por CARRIED se ejecutarán las acciones Message Turns y End. Y esto es independiente de cualquier cosa que el jugador teclée para llegar a la localidad donde está la parada de autobús.

Seleccionemos ahora proceso 1 tecleando [ S 1 ENTER ] y usemos [P] para examinar las entradas que ya están presentes:

```
*   -   NEWLINE
      ZERO      0
      ABSENT    0
      LISTOBJ
*   -   PRESENT  0
      LISTOBJ
```

Un asterisco "\*" significa "cualquier palabra" como significaba "\_", pero con una diferencia: siempre que PAW inserta entradas en una tabla de Procesos (y ello incluye la de Respuestas), las insertará según el orden de valor, primero el Verbo y segundo el Nombre (por ejemplo, todas las entradas que se refieran a un mismo verbo irán una detrás de otra en orden ascendente según su valor de nombre).

PAW considera que la raya baja "\_" tiene un valor de 255 (por lo tanto, será siempre la última entrada). Sin embargo, considera que asterisco "\*" tiene el valor de 1 (por lo tanto será siempre la primera entrada).

La posición de las entradas en una tabla de procesos es muy importante. Por ejemplo, las dos entradas que hemos mencionado arriba, siempre deben de estar en el mismo orden que las hemos dado. Ellas deben ser ejecutadas inmediatamente después de que se ha impreso la descripción para una localidad y por ello es por lo que usamos el asterisco, para que estén al principio de la tabla (usamos la raya baja "\_" como si fuese un nombre para poder meter alguna entrada antes de ella, como veremos en su momento).

Veamos porqué se ponen esas dos entradas. Como PAW ejecuta siempre todas las entradas que hayan en Proceso 1 y Proceso 2 (suponemos que te darás cuenta de que lo hará de cualquier forma, porque una entrada que lleve \* \_ hará pareja con cualquier Input del jugador, es decir, con cualquier Sentencia Lógica) la acción NEWLINE siempre será ejecutada.