

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 Autobus en una posición que corresponda al nombre, o sea, primero el Verbo y luego el Nombre).

Así que teclea [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 teclee para llegar a la localidad donde está la parada de autobús.

Seleccionemos ahora proceso 1 tecleando [S 1 ENTER] y usamos [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.

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 cuál 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 (asegurate de que todavía está seleccionado) usando [I * * ENTER]. Esto nos introduce dos nuevos conductos.

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 estamos 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 MESSAGE 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 **POWER** (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 conductos 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.

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.

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 cuál "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 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 teclea una frase.

El pajarito cambiará de localidad cada tres frases, lo que creará una especie de apariencia de acción independiente de lo que teclea 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:

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

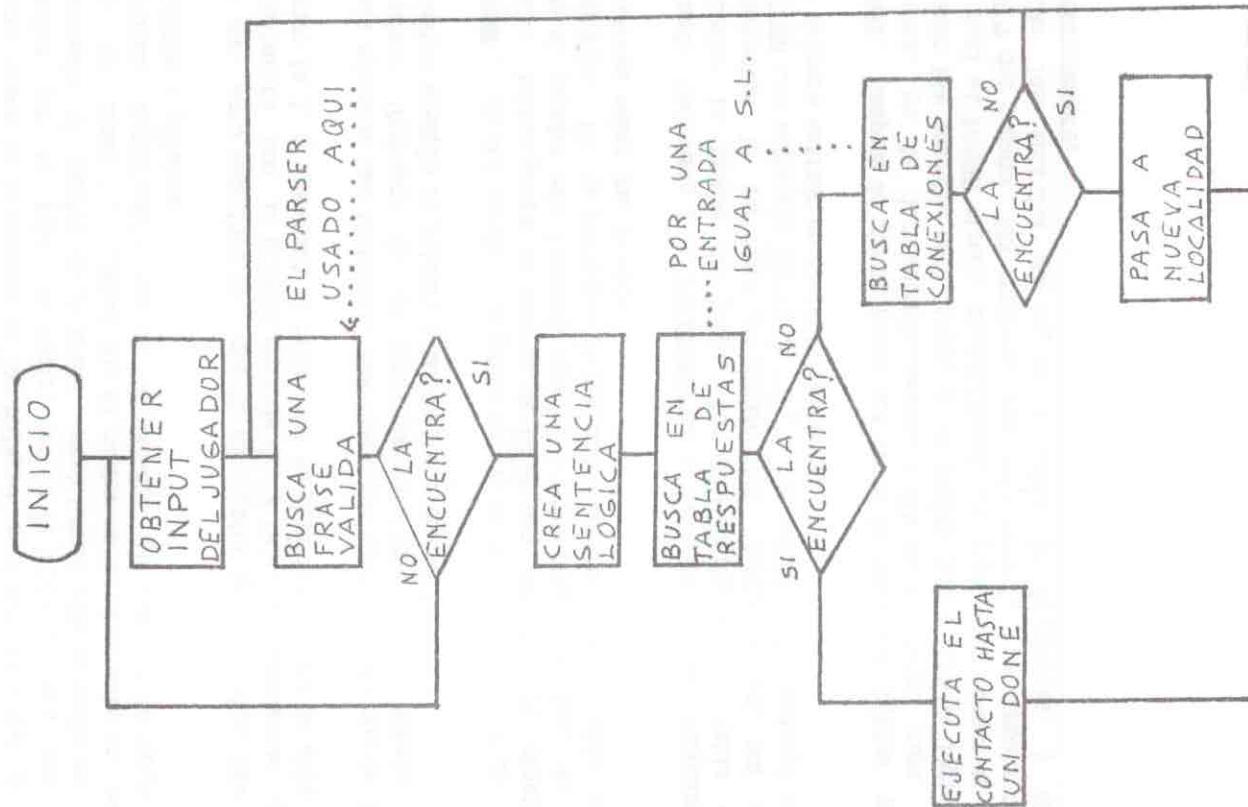
Es una condición que debe ser seguida por un número de bandera y otro valor, y que será positiva si la bandera contiene ese valor. En este caso (EQ 12 8) está comprobando si el pájaro está en una localidad específica.

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.

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 empieza 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:

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 sí 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 teclee 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 amarrándola al banco. Además, el jugador puede "hablar" al perro, lo cual le dará otro medio de desembarazarse 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

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 (MODO 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.

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.

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.

SIEMPRE	-	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		

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  EMPAREJADO _
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.

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 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 ésa tabla para que
haga el trabajo.
DONE

```

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].

Usamos 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 NOTDNE 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 condacs 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.

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.

Ahora empecamos 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 llesves 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.

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 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.

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 teclar el texto que la describe.

Debido al formato automático hay que teclar 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 (juraito 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)

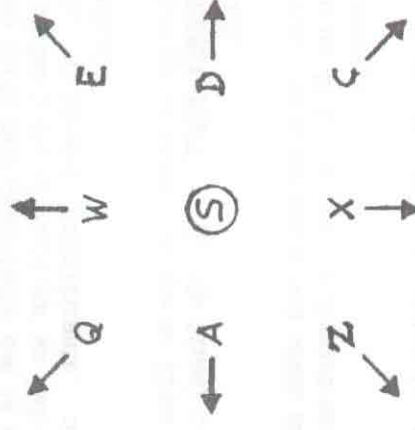
2- Línea inferior:

- 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.

Teclando [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.

EMPEZANDO A TECLLEAR

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 teclleada aparecerá en la parte inferior de la pantalla, como siempre. Las zonas en inverse nos indican lo que hay que tecllear 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 teclleando 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 tecllea EDIT (cap. sh. + 1) dos veces y obtendrá una localidad limpia para trabajar.

Nota: tener en cuenta que con tecllear 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 teclléa 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 teclleemos [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.

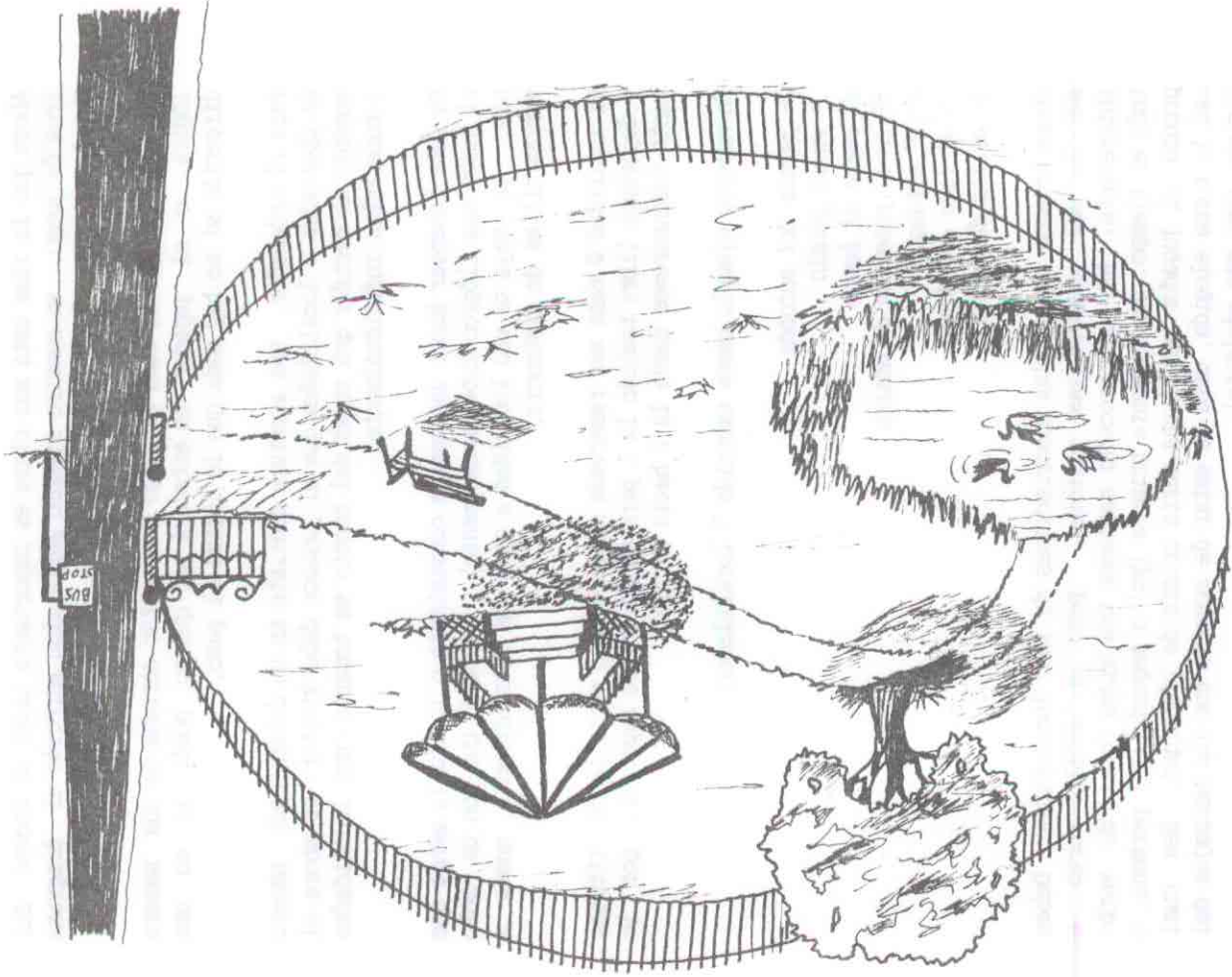
De momento solamente queremos uno, y tecllearemos [12 ENTER] otra vez. Automáticamente el borde quedará con su relleno. Notarás que el punto no se ha movido y que la línea sale desde el mismo sitio desde donde estaba antes del relleno.

CARACTERÍSTICAS 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.

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:

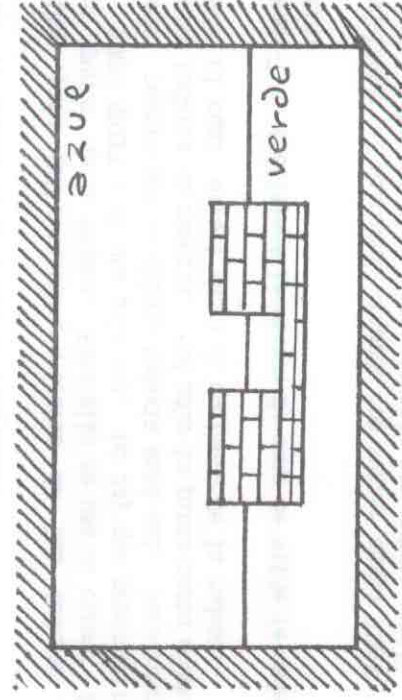
64,72 64,95 111,95 111,75 144,75 144,95 191,95 191,72

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



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 empezarlo 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 autobús.



Volviendo con lo nuestro, entonces desde el inicio, CAPS SHIFT + 6 hasta que lleguemos 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:

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 hacia **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.

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



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.

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

QJO: 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 COSUB y te pedirá el número de la localidad que vas a usar, tecléa [0 ENTER] para usar nuestro hieabajo 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 \emptyset 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.

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 PAV 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 brdenes 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 0).

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).

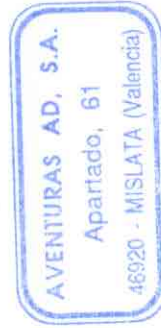
También se mostrarán dos direcciones en decimal que serán necesarias si quieres escribir tus propias adiciones al PAW, sea en BASIC o en CODIGO MAQUINA. Para detalles consulta el manual técnico.

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.

PERIFÉRICO 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.

INDICE

Tabla de Respuestas	Pag. 36	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.
Situación de un objeto	Pag. 42	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.
Mensajes	Pag. 46	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...)
Objetos dentro de objetos	Pag. 50	
El Pajarito	Pag. 56	
Sub-procesos	Pag. 60	Esto es un cambio menor, pero significa que el Verbo Actual se mantiene cuando la cadena sufre un PARSEado múltiple.
El Ferrito	Pag. 62	
Mejoras por ti mismo	Pag. 68	
Uso de Overlays	Pag. 78	
Compresión de Texto	Pag. 80	UNA BANDERA MAS
Editor de Caracteres	Pag. 82	Recuerdas que te insistíamos mucho en que respetaras las banderas hasta la 60, porque se usarían en futuras expansiones?
Editor Gráfico	Pag. 83	Pues el caso ha llegado con esta nueva versión.
Ultima Hora (muy importante)	Pag. 85	Nos congratulamos en anunciar que la bandera 58 ha alcanzado el estatus de "Bandera del Sistema".
Final del Camino	Pag. 99	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 recordarán, sólo hacía en la tabla de Respuestas.
	Pag. 103	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.

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 distinguirte de esa plaga moderna: los piratas.

3- te invitamos a reflexionar sobre lo que te ha costado el programa, su gran potencial y los beneficios que puedes obtener con él. Por cada copia ilegal que 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 tú 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.

ANDRÉS R. SAMUDIO - 1989

